

Міністерство освіти і науки України
Чорноморський національний університет імені Петра Могили

В. Я. Кутковецький

Розпізнавання образів

Навчальний посібник

Миколаїв – 2017

УДК 004.93'1(075.8)

К 95

Рекомендовано до друку вченою радою ЧНУ ім. Петра Могили (протокол № 4 від 8 грудня 2016 року).

Рецензент:

Мусієнко М. П., д-р техн. наук, професор, зав. кафедри інформаційних технологій і програмних систем, декан факультету комп'ютерних наук, Чорноморський національний університет ім. Петра Могили.

К 95

Кутковецький В. Я. Розпізнавання образів : навчальний посібник / В. Я. Кутковецький. – Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2017. – 420 с.

ISBN 978-966-336-384-4

Викладені основні методи й алгоритми розпізнавання образів в інтелектуальних системах.

Розглянуті питання: обробки ознак образів; методології групування образів; теорії тілесних та об'єктних кластерів на базі розробленої в Україні аналітичної геометрії n-вимірних тілесних кутів; дискримінантного аналізу; логічного апарату класифікації образів; розробленої в Україні теорії однозначних нечітких систем; нейронних мереж (НМ); стохастичних методів; лінгвістичних (структурних) методів розпізнавання образів. Приділена увага розпізнаванню ознак товарів та покупців. Теоретичний матеріал супроводжується навчальними завданнями. Наведена інформація про розроблені в Україні нейрони та НМ, які захищені патентами, що належать ЧНУ ім. Петра Могили

Навчальний посібник призначений для студентів, аспірантів і науково-технічних робітників, фахівців різних спеціальностей, які займаються створенням програмного та апаратного забезпечення сучасних інтелектуальних систем.

УДК 004.93'1(075.8)

© Кутковецький В. Я., 2017

© ЧНУ ім. Петра Могили, 2017

ISBN 978-966-336-384-4

ЗМІСТ

1. Загальні відомості про розпізнавання образів	9
1.1. Розпізнавання образів, ознаки, дії, історія.....	9
1.2. Ієрархія інтелектів	10
1.3. Класи задач розпізнавання образів.....	11
1.4. Проектування системи РО.....	12
1.5. Визначення ознак образів.....	14
1.5.1. Технічні датчики ознак.....	14
1.5.2. Людина як датчик реальних та абстрактних ознак.....	17
1.5.3. Ознаки маркетингового аналізу.....	19
1.5.4. Ознаки ризиків фінансових операцій.....	20
1.5.5. Визначення інтелекту, психічних даних та характеру людини.....	21
1.5.6. Застосування класифікаторів.....	23
1.5.7. Ознаки соціологічного стану	25
1.5.8. Ознаки лінгвістичного аналізу текстової інформації та мовлення.....	27
1.6. Обробка сигналів	29
1.6.1. Початковий вибір числових змінних образу.....	29
1.6.2. Попередня обробка числових сигналів: нормалізація, квантування, фільтрація, диференціювання, фрагментизація, об'єднання, згладжування ознак	31
1.6.3. Контрольні карти процесів.....	33
1.7. Приклад системи розпізнавання образів	33
1.8. Класифікація систем розпізнавання образів.....	34
1.9. Необхідність зовнішніх та внутрішніх знань.....	36
2. Теорія аналітичної геометрії n-вимірних тілесних кутів.....	39
2.1. Функції аналітичної геометрії n-вимірного простору.....	39
2.2. Візуальна аналітична геометрія об'єктів n-вимірного простору	43
2.3. Аналітичні та геометричні особливості n-вимірних тілесних кутів..	47
2.3.1. Контекстні метрики та числові значення змінних по осях координат	47
2.3.2. Порядкові номери тілесних кутів та їх позначення.....	52
2.3.3. Сусідство тілесних кутів та об'єктів.....	54
2.3.4. Початкові дані аналітичної геометрії n-вимірних тілесних кутів	58
2.3.5. Точки, лінії, площини, вектори та геометричні фігури n-вимірних тілесних кутів	61
2.3.6. Система координат тілесних кутів на двовимірній площині	63
2.3.7. Годографи, ізолінії та геометрична, текстова, математична, акустична і мовна інформація n-вимірних тілесних кутів.....	65
2.3.8. Двовимірна мапа тілесних n-вимірних кутів	68

2.4. Напрямки використання аналітичної геометрії <i>n</i> -вимірних тілесних кутів.....	72
2.5. Теорія тілесних кластерів <i>n</i> -вимірного простору.....	74
3. Обробка ознак та групування образів.....	81
3.1. Прості методи класифікації образів	81
3.2. Перетворення ознак образів у коди.....	87
3.3. Перетворення аналогового сигналу у код.....	90
3.4. Відстань Евклідова, Манхеттена, Чебишева, Махаланобіса, Хеммінга. Близькість по Хеммінгу та за скалярним добутком	91
3.5. Групування об'єктів з цифровими ознаками по середньоквадратичному розкиду.....	93
3.6. Напрямки групування об'єктів по аналізу даних в магазині	95
3.7. Метод ієрархічних потоків.....	102
3.8. Ринкова ціна на товар	105
3.9. Групування об'єктів розділом та об'єднанням множин.....	106
3.10. Дискримінантний аналіз.....	108
3.11. Системи автоматичного регулювання та захисту	109
3.12. Особливості комп'ютерного керування	110
4. Моделі представлення знань.....	114
4.1. Знання і їх представлення в системах розпізнавання образів.....	114
4.2. Формальні системи.....	115
4.3. Твердження та логічні формули	117
4.4. Таблиця істинності для відомої логічної формули.....	118
4.5. Таблиця істинності для сукупності тверджень.....	121
4.6. Логіка предикатів	123
4.7. Однозначні нечіткі системи.....	130
4.7.1. Теорія однозначних нечітких систем.....	130
4.7.2. Переваги однозначних нечітких систем	135
4.7.3. Чайові в ресторані	136
4.7.4. Оплата працівника	141
4.7.5. Апроксимація «невідомої» функції $F = x_1 \sin x_2$	142
4.7.6. Соціальний стан на підприємстві.....	145
4.7.7. Математична апроксимація графіка.....	146
4.8. Продукційна база знань	149
4.8.1. Форми продукційних правил в базах знань	149
4.8.2. Структурна схема продукційної бази знань.....	152
4.9. Семантичні мережі	153
4.9.1. Базові компоненти семантичної мережі та їх призначення.....	153
4.9.2. Апарат успадкування властивостей об'єктів семантичної мережі	156
4.10. Фреймові моделі	156
4.10.1. Загальні положення теорії фреймів.....	156
4.10.2. Типові процедури роботи із фреймами	158

4.10.3. Ієрархічність фреймової моделі.....	159
4.11. Експертні системи	161
4.11.1. Призначення експертних систем	161
4.11.2. Створення експертної системи	162
4.11.3. Звичайні та спеціальні мови програмування експертних систем	164
4.12. Бази даних	165
4.12.1. Загальні відомості про бази даних	165
4.12.2. Концептуальна модель реляційної бази даних	170
4.13. Бази знань.....	174
4.14. Сценарії	175
5. Стохастичний аналіз.....	177
5.1. Початкові відомості з теорії стохастичних рішень	177
5.2. Ймовірності помилок класифікації одновимірних образів двох класів.....	178
5.3. Визначення коефіцієнта правдоподібності з врахуванням ризику рішення для одновимірних образів двох класів.....	179
5.4. Мережі відносних стохастичних та детермінованих потоків.....	180
5.4.1. Умовна ймовірність та формула повної ймовірності	180
5.4.2. Визначення формули Байєса.....	181
5.4.3. Використання формули Байєса для розпізнавання хвороб	184
5.4.4. Використання формули Байєса для прогнозування дощу	186
5.4.5. Зв'язок відносних стохастичних та детермінованих потоків... ..	187
5.4.6. Електрична схема заміщення Байєсової мережі з потоками умовних ймовірностей.....	190
5.4.7. Теоретична основа мереж стохастичних та детермінованих потоків	195
5.5. Статистичні способи обробки експериментальних даних	197
5.5.1. Методи первинної статистичної обробки результатів експерименту	197
5.5.2. Початкове обчислення елементарних математичних статистик	198
5.5.3. Методи вторинної статистичної обробки результатів експерименту	201
5.6. Генетичні алгоритми та еволюційне програмування.....	205
5.7. Метод групового урахування аргументів (МГУА)	208
6. Вирішальні функції та вирішальні правила	210
6.1. Особливості вирішальних функцій та вирішальних правил	210
6.2. Вирішальні функції та вирішальні правила для складного розміщення класів.....	212
6.3. Апаратна реалізація вирішальних функцій та вирішальних правил... ..	213
7. Загальні відомості про нейрони	215
7.1. Історія нейронних мереж.....	215

7.2. Недоліки та переваги нейронних мереж	217
7.3. Математична нерівність вирішальної функції – основа роботи штучного нейрону.....	220
7.4. Функції моделей нейронів	223
7.5. Нейронна мережа для керування судновою електроенергетичною системою.....	227
7.6. Класифікація нейронних мереж	229
8. Адаптивний лінійний нейрон Adaline	232
8.1. Схема нейрону Adaline	232
8.2. Логічна функція AND	233
8.3. Логічні функції OR і XOR.....	234
9. Навчання нейронних мереж.....	236
9.1. Загальні відомості про навчання нейронів	236
9.2. Призначення шарів та нелінійних виходів нейронів у перцептроні ...	238
9.3. Розділ шарами перцептрону простору ознак	241
9.4. Дельта-правило навчання нейрона	244
9.5. Алгоритм зворотного розповсюдження похибки у процесі навчання.....	246
9.6. Деякі напрямки навчання нейронних мереж	251
10. Реалізація простих функцій нейронами	258
10.1. Нейрони «Троянда»	258
10.1.1. Порівняння нейронів «Троянда» з нейронами Adaline	258
10.1.2. Типи нейронів «Троянда»	262
10.1.3. Моделювання логічної функції XOR одним нейроном «Троянда».....	266
10.2. Логічні функції T_AND та T_OR_T з багатьма різнополярними входами.....	268
10.3. Вхідна та вихідна зірки Гросберга.....	270
10.4. Нейронна мережа Івахненко «Альфа».....	272
10.5. Нейронна мережа «MAXNET» для кодування елементів вхідного вектора	272
10.6. Нейрон для введення формул.....	274
10.7. Нейронна мережа для стиснення інформації «КОД n:m:n»	275
10.8. Нейронні мережі для виведення екстремального елемента з вхідного вектора	278
10.9. Нейрони « T_MAX » та « T_MIN » для двоїстого кодування елементів вхідного вектора з виділенням серед них екстремального значення.....	282
10.10. Стохастичний нейрон	284
10.11. Динамічна модель нейрона Хопфілда.....	285
11. Приклади нейронних мереж	287
11.1. Топологія нейронних мереж	287

11.2. Асоціативна пам'ять	289
11.2.1. Нейронна мережа Хопфілда.....	289
11.2.2. Нейронна мережа Андерсона.....	293
11.2.3. Асоціативна нейронна мережа з кількох вхідних зірок.....	294
11.2.4. Двоспрямована асоціативна пам'ять.....	297
11.2.5. Нейронна мережа Хеммінга.....	299
11.2.6. Нейронні мережі «Миколаїв»	302
11.3. Багатошарові нейронні мережі з одним шаром навчання.....	305
11.3.1. Загальні особливості багатошарових нейронних мереж з одним шаром навчання.....	305
11.3.2. Однозначні нейро-нечіткі системи з нейронами Троянда.....	306
11.3.3. Однозначні нечіткі нейронні мережі	308
11.4. Самонавчальні нейронні мережі	312
11.4.1. Самонавчальна мережа векторного квантування Кохонена.....	312
11.4.2. Самоорганізувальна мапа Кохонена	315
11.4.3. Самонавчальна нейронна мережа з мапою	318
11.4.4. Мережі зустрічного розповсюдження	322
11.5. Стохастичні нейронні мережі.....	324
11.5.1. Напрямки аналізу стохастичних нейронних мереж	324
11.5.2. Архітектура стохастичної нейронної мережі Спехта.....	326
11.5.3. Стохастична кластерна нейронна мережа	327
11.5.4. Машина Больцмана.....	338
11.5.5. Нейрон Байеса для класифікації об'єктів n-вимірного простору	341
11.6. Апроксимація функцій нейронною мережею	343
11.6.1. Нейронна мережа з радіальними базовими функціями	343
11.6.2. Мережа регресійного аналізу.....	347
11.7. Динамічна рекурсивна нейронна мережа	348
11.8. Нейрорегулятори.....	351
11.9. Імпульсні нейронні мережі.....	352
11.10. Адаптивна резонансна теорія нейронної мережі.....	357
11.11. Згорнута нейронна мережа	358
11.12. Деякі напрямки розвитку нейронних мереж	359
12. Метод потенціалів	364
12.1. Потенціали у просторі цифрових ознак.....	364
12.2. Двоїсті значення потенціалів рецепторів у просторі зображення	365
12.3. Дробові значення потенціалів рецепторів у просторі зображення	366
13. Лінгвістичні методи розпізнавання образів	370
13.1. Структурний метод розпізнавання образів	370
13.1.1. Загальні відомості	370

13.1.2. Формальні граматики структурного методу	371
13.1.3. Виведення граматик.....	374
13.1.4. Приклади використання структурного методу.....	377
<i>13.2. Роль СЛОВА у розпізнаванні образів</i>	<i>383</i>
13.2.1. СЛОВО як матеріальна та ідеальна основа інформатики.....	383
13.2.2. Онтологія	386
13.2.3. Тезаурус – словник представлення знань.....	388
<i>13.3. Кластеризація на основі теорії грубих множин</i>	<i>389</i>
14. Деякі напрямки систем розпізнавання образів.....	391
<i>14.1. Систематика.....</i>	<i>391</i>
14.1.1. Визначення простих, великих і складних систем.....	391
14.1.2. Специфікація і створення чотирьох словників (понять, відносин, процесів і дій).....	391
14.1.3. Штучна мова і загальний опис складних систем.....	392
14.1.4. Методологічні особливості опису складної системи	395
14.1.5. Моделювання динаміки складних систем	397
<i>14.2. Маскування та демаскування образів.....</i>	<i>398</i>
<i>14.3. Сукупність алгоритмів та формул.....</i>	<i>403</i>
<i>14.4. Фрактальна геометрія.....</i>	<i>405</i>
<i>14.5. Теорія інформаційної грануляції.....</i>	<i>405</i>
<i>14.6. Метод «найближчого сусіда»</i>	<i>406</i>
Список використаної літератури	407

1. ЗАГАЛЬНІ ВІДОМОСТІ ПРО РОЗПІЗНАВАННЯ ОБРАЗІВ

1.1. Розпізнавання образів, ознаки, дії, історія

Образ – це об’єкт, процес або явище реального та абстрактного світу, який розпізнається за даними (ознаками), що збираються та оброблюються індивідуально і у сукупності. Розпізнавання образу не є безцільною грою і завжди супроводжується **дією**. «**Адекватна дія**» на розпізнаний конкретний об’єкт є основою життєдіяльності будь-якого живого організму. В штучних інтелектуальних системах дія має вигляд порівняння з еталоном, увімкнення в роботу виконавчого пристрою, вилучення з розгляду образу, запису або видачі інформації тощо.

Приклади образів: реальні та абстрактні об’єкти; люди (їх голоси, почерки, пульс, хвороби тощо); вироблення товарів; економічні явища та процеси; економічний стан; виробничі, соціальні, суспільні процеси; небезпека у роботі фірми; літери; автомашини тощо.

Примітка. **Абстрактний світ** – це уявний (не існуючий в природі) продукт розумової діяльності людини, матеріалізований **СЛОВОМ**: математика, хімія, вокал, закони тощо.

Неможливо охопити всі напрямки діяльності людини, в кожному з яких потрібно розпізнавати свої власні образи. Наприклад, науки про хвороби, риби, технічні пристрої можуть бути достатньо об’ємними ділянками знань, які не може охопити одна людина. Інформаційні технології є різноманітними: їх можна розділити на технології спілкування на природній мові, перекладу на різні мови «з голосу», виробничі технології, доступу до даних, Інтернет, захисту інформації, розвідки корисних копалин з космосу тощо. Тому дисципліна розпізнавання образів (РО) завжди вимушено обмежується описом лише загальних найбільш розповсюджених методів.

Образи розпізнають (класифікують, виділяють, описують) по ознаках, станах, зв’язках; по сукупності власних наявних субобразів. Образи у вигляді вектора $X=(x_1, x_2, \dots, x_j, \dots, x_n)$ не є чіткою азбукою Морзе; вони частіше є сукупністю чітких та нечітких, повних та неповних, прямих та непрямих, постійних та змінних, точних та перекручених даних із загальною масою неістотних, взаємно суперечливих та зв’язаних між собою деталей. У військовій справі перекручений образ може мати й вигляд моделі (макета) з наявними головними ознаками (в цьому випадку потрібно розглядати й додаткові ознаки). У найбільш поширеному випадку ознаки мають вигляд числових значень змінних вектора $X=(x_1, x_2, \dots, x_j, \dots, x_n)$.

Ознаки є описом образу і розділяються на числові, лінгвістичні, якісні, реальні, абстрактні, детерміновані, стохастичні, часові, просторові,

причинно-наслідкові характеристики об'єкта та ін. В якості ознак також можуть виступати самі об'єкти РО, події, процеси, явища, які, у свою чергу, стають ознаками ієрархічно вищих образів.

У загальному випадку ознаки об'єктів можуть мати різний вигляд:

– *опис людини* охоплює **числові** ознаки (зріст, вагу), **якісні** дані (порядність, ерудицію, психічні дані), **описові** дані (лінгвістичні опис рис обличчя та одяжі, колір очей, кваліфікацію, хобі) тощо;

– *товар має комбіновані* ознаки, які діляться на **числові** (вага, ціна, кількість), **якісні** (якість виготовлення і збереження, конкурентоспроможність, привабливість) та **описові** (виробник товару та його адреса; максимальний термін зберігання; місце і умови зберігання товару; «залежала товарна група»; «товарний сектор підвищеного попиту»; «товар підвищеної якості»).

Розглядати історичний розвиток у часі і становлення напрямку «Розпізнавання образів» складно, бо воно притаманне з дня народження будь-якій живій істоті, а також черв'яку, рослині та машинній системі. Черв'як і рослина розпізнають пори року, розмножуються, захищаються.

«*Машинне розпізнавання образів*» народилось разом зі створенням нашими предками кібернетичних машин у вигляді пастки для ловлі риби (лабіринти із заглиблень на морському узбережжі, які під час припливу заповнювались водою і рибою), або капканів та замаскованих ям для звірів. Пастка – це автомат одноразової дії, налаштований на розпізнавання ознак образу.

Сам курс «розпізнавання образів» почав формуватись з 1950-х років для задоволення потреб численних машинних автоматичних і автоматизованих систем штучного інтелекту.

1.2. Ієрархія інтелектів

У будь-якому інтелектуальному напрямку існує **ієрархія інтелектів**. Можна вважати, що система штучного інтелекту створена тоді, коли вона виконує хоча б частку **цілеспрямованої діяльності людини** (пастка, підтримка частоти обертів валу машини, підтримка температури приміщення, розшифровка тексту).

Відомий тест Тьюрінга на створення штучного інтелекту («штучний інтелект створений тоді, коли ми розмовляємо по телефону і не можемо розпізнати, з ким розмовляємо, з людиною, чи із штучним інтелектом») насправді є **тестом лише на ієрархію інтелектів** (студент у розмові з дитиною може дійти хибного висновку, що вона не має інтелекту). Загальний ієрархічний рівень інтелекту може оцінити експерт по комплексним показникам [49]. Норберт Вінер, математик США, «батько кібернетики» порівнював *ЕОМ та мозок людини*, казав про «розум мурашки» і назвав свою книжку «Кібернетика, або керування й зв'язок у тварини й машини» (1948).

Ситуації, коли «1. Людина штовхає стовбур дерева», чи «2. Пастка звільняє стовбур», і цей стовбур у падінні вбиває тварину багато хто схильні розглядати як «1. Інтелектуальну роботу людини» та «2. Неінтелектуальну роботу пастки». Але насправді ця робота – *однакового інтелектуального рівня*. І невідомо ще, хто ліпше її зробить: пастка, чи недосвідчена людина або дитина. Аналогічно: природний регулятор серцебиття людини керується мозком і може вважатись «інтелектуальним»; його може замінити ліпше працюючий механічний регулятор серцебиття, який отримує ярлик «неінтелектуальності». Відомо, що створені наразі численні системи автоматичного керування можуть працювати *ліпше за людину*, особливо при дефіциті часу.

В деяких напрямках людина поступається інтелектом перед:

– *штучним (машинним) інтелектом*, який: виконує роботу значно швидше і ліпше; запам'ятовує величезні об'єми інформації; розпізнає серед шумів моря шум підводного човна; по вигляду поверхні Землі з космосу визначає ділянки, підозрілі на наявність корисних копалин; вчасно регулює за складною математичною формулою кількість поданого палива у залежності від подачі повітря (окислювача), величини температури та тиску газу в реактивному двигуні з метою запобігання помпажу (помпаж – зрив вихорів з лопаток турбіни, сприймається як вибух); розпізнає і вимірює магнітне поле тощо;

– *інтелектом живих істот*, які мають ліпший нюх та слух. Саме тому людина на полюванні об'єднується з собакою;

– *інтелектом рослин*, які восени передбачають температуру повітря майбутньої зими. Саме тому людина «звертається за інформацією» до рослин стосовно погоди.

Найбільш пристосованим до навколишнього середовища є *об'єднаний інтелект*: «людина + людина», «людина + ЕОМ», «людина + ЕОМ + машина + рослина», «сукупність людей». Тобто існує ієрархія інтелектів, які або знають алгоритм прийняття рішень (алгоритм дій) або створюють його.

Саме тому люди радяться між собою при прийнятті важливих рішень і існують колегіальні органи для прийняття колективного (більш зваженого) рішення. Але при реалізації прийнятого рішення життя часто вводить зміни в умови реалізації і тому реалізацію рішення потрібно невинно контролювати і корегувати.

1.3. Класи задач розпізнавання образів

Сучасні інформаційні системи РО є *складовою часткою будь-яких штучних інтелектуальних систем* і використовуються в експертних системах, базах даних та базах знань, в інформаційних, прогнозних та інших системах. Прикладами можуть бути комплексні інтелектуальні системи наприклад, «Здобуття знань» («DataMining») для прийняття рішень, які використовують ряд методів РО – рис. 1.3.1).

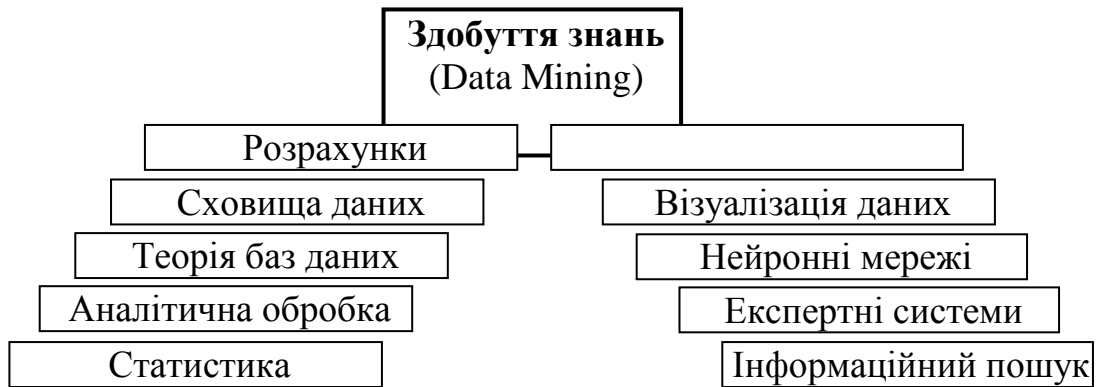


Рис. 1.3.1. Система «Здобуття знань» («DataMining»)

Стосовно всіх класів задач і перелічених напрямків тяжко сказати, до якого курсу вони відносяться:

- до «Розпізнавання образів?»;
- до «Штучного інтелекту?»;
- до «Теорії прийняття рішень?»;
- до «Інтелектуальних довідкових/експертних/прогнозних/інформаційних/керуючих систем?».

Виявляється, що неможливо *строго визначити клас задач*, який відноситься виключно до «Розпізнавання образів», так само як і *дати вичерпне визначення самого предмета вивчення, строго відмінного від зазначених вище напрямків розвитку інтелектуальних систем*.

Будь-які існуючі або майбутні автоматичні чи автоматизовані інтелектуальні системи включно з РО повинні мати однакові риси («мету», «об'єкт зовнішнього середовища», «розпізнавання об'єкта по ознаках») та повинні виконувати відповідні «оптимальні інтелектуальні дії».

Але різниця між «Розпізнаванням образів» та згаданими вище напрямками все ж є: дисципліна «Розпізнавання образів» узагальнює, об'єднує, досліджує і удосконалює основні оптимальні методи, алгоритми, моделі і засоби інтелектуальних систем при обробці даних реального чи абстрактного середовища, в той час як інші згадані інтелектуальні системи повинні використовувати і поглиблювати ці загальні методи дисципліни РО з урахуванням особливостей процесів у конкретних напрямках і ситуаціях.

1.4. Проектування системи РО

Предметом дослідження курсу РО є математичні методи, алгоритми, моделі, технології та засоби класифікації об'єктів шляхом визначення, обробки, аналізу, зберігання та використання їх ознак. Метод класифікації охоплює сукупність правил створення класифікаційних груп.

Методи РО охоплюють широкий клас наукових та евристичних методів: статистики, матричної алгебри та диференційного і інтегрального числення, аналізу даних та множин, теорії нечітких множин, дослідженні операцій, нейронних мереж, теорії графів, математичного моделювання, онтології і тезаурусу.

Універсумом зветься виділена для класифікації сукупність образів з однаковим вмістом схожих суттєвих ознак $X = (x_1, x_2, \dots, x_j, \dots, x_n)$, $j = 1, 2, \dots, n$; n – кількість ознак об'єкта. Приклади універсумів однорідних об'єктів: люди, економічні стани та процеси, товари, дерева, ситуації, явища, тексти, голоси, слони, комахи, літаки, машини тощо з власними ознаками, властивостями, характеристиками X .

Алфавіт образів розділяє універсум на однотипні класи. Наприклад, однотипні образи універсуму «Автомобілі» розділяються на класи товарів «Волга», «Лада», «Запорожець» і т. д. Клас товару «Волга» може розділятися на групи (кластери, таксони) по цінах, прибутку, попиту, роках випуску машини; по машинах, що попали в аварію, або у яких відмовив якійсь пристрій. Часто до існуючих m класів додається $(m+1)$ й «нерозпізнаний образ».

При проектуванні системи РО складаються початковий та робочий словники ознак.

Початковий словник ознак – це найбільш повний (у розумних межах, тобто забезпечений існуючими відповідними датчиками, або реальною можливістю їх створення та використання) перелік ознак образів універсуму, які отримуються за допомогою датчиків, лінгвістичних визначень та абстрактних понять.

Робочий словник ознак – це частка початкового словника ознак $X = \{x_j\}$, $j = 1, \dots, n$, яка практично використовується і обробляється у системі РО і складається з найбільш суттєвих прямих або дотичних ознак. Наявність відповідних датчиків може впливати на вибір ознаки у робочий словник.

Групою (кластером, таксоном, класом) універсуму зветься підмножини образів з найближчими значеннями схожих суттєвих ознак між собою, які описується робочим словником ознак.

Метою РО є класифікація об'єкта по визначеним ознакам $X = (x_1, x_2, \dots, x_j, \dots, x_n)$, $j = 1, 2, \dots, n$, що дає можливість з виконати адекватні оптимальні дії системою РО стосовно розпізнаного об'єкта.

Систему РО можна уявити як абстрактну сукупність ряду елементів $R = \{X, P, A, D\}$, де $X = \{x_j\}$, $j = 1, \dots, n$ – множина ознак образу; $P = \{P_l\}$, $l = 1, \dots, L$ – множина правил прийняття рішення; $A = \{A_k\}$, $k = 1, \dots, K$ – алфавіт класів (груп, кластерів, таксонів, категорій, альтернатив); $D = \{D_\lambda\}$, $\lambda = 1, \dots, A$ – перелік дій.

Систему «Розпізнавання образів» (РО) можна уявити у вигляді рис. 1.4.1.

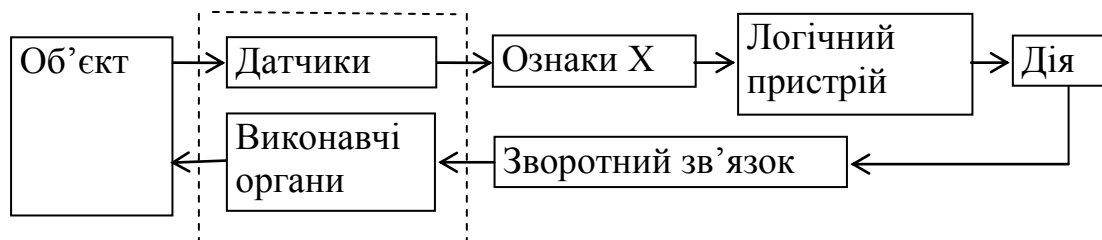


Рис. 1.4.1. Взаємодія системи РО з об'єктом розпізнавання

Загальний алгоритм системи РО:

– *сприйняття «Об'єкта»* за допомогою «Датчиків» у вигляді деякого вектора оброблених кодованих ознак – змінних $X=(x_1, x_2, \dots, x_j, \dots, x_n)$. При цьому прийнятті ознаки можуть додатково оброблятися для отримання вектора X ;

– *розпізнання (класифікація) об'єкта* у «Логічному пристрої» на основі значень «Ознак X »; *еталонні вектори кодованих змінних X^{Ee}* (тут $e = 1, 2, \dots, E_e$ – порядковий номер класу) зберігаються у пам'яті системи РО і за правилом «переможець отримує все» «конкурують» між собою за «привласнення» вхідного кодованого вектора об'єкта $X=(x_1, x_2, \dots, x_j, \dots, x_n)$ шляхом порівняння відстані, або близькості вектора X з даними еталонів X^{Ee} .

– *прийняття рішення про виконання відповідних «Дій»* у «Логічному пристрої», яка може бути спрямована або на зміну параметрів об'єкта за допомогою «Зворотного зв'язку» та «Виконавчих органів», або на передачу інформації в іншу інтелектуальну систему для запам'ятовування отриманих результатів чи подальшої їх обробки.

1.5. Визначення ознак образів

1.5.1. Технічні датчики ознак

Метою огляду не є повний перелік всіх існуючих датчиків, а лише ознайомлення з основними напрямками вимірювання ознак об'єктів технічними засобами.

Датчиком зветься прилад, призначений для сприйняття, вимірювання, переробки і зберігання однієї або кількох ознак образу.

Технічні датчики мають наступні властивості: точність; надійність; стабільність роботи у часі; швидкодію; лінійність між вхідними та вихідними даними; гістерезис; старіння; безконтактність (такі датчики не мають елементів з механічним контактом між ними і взаємним пересуванням); залежність від зовнішніх параметрів (вібрації, температури, освітлення, опромінювання, вологості повітря); наявність шуму при вимірюванні.

Технічний датчик, звичайно, сприймає досить *вузьку кількість* ознак образу: датчик відеосигналу (телевізійна камера) сприймає колір, освітлення та напрямок переміщення об'єкта, алене може визначати погоду; тензодатчик визначає деформацію образу і не сприймає його освітлення.

Вихідні характеристики датчиків у принципі можуть мати різну фізичну природу, але частіше це – аналогові або цифрові *електричні сигнали*.

Технічні датчики для вимірювання відстаней та розмірів реальних об'єктів мають вигляд контактів замикання – розмикання (кінцевих вимикачів); лазерних та електромагнітних локаторів; гідролокаторів; луно лотів; ємнісних чи індуктивностей датчиків, параметри яких змінюються при переміщенні рухомого елемента, або появи стороннього об'єкту; резистивно-

го датчика-потенціометра, вихідна напруга якогозалежить від величини переміщення повзунка.

Датчик деформації об'єкта (тензодатчик) має вигляд проволоки, наклеєної на поверхні об'єкта. Електричний опір проволоки збільшується при збільшенні механічної деформації. Датчик деформації може мати вигляд пів провідникового покриття, а також вимірювати розміри чи переміщення (у вигляді п'єзокристалу чи ємнісного датчика).

Датчики температури мають вигляд:

Термістора – резистора з нелінійною стрибкоподібною залежністю опору від температури. Діоди та транзистори можуть використовуватись як чутливі елементи з нелінійною залежністю опору від температури на $-50...+150^{\circ}\text{C}$.

Термометра опору (мідні, платинові у вигляді намотаної на каркас проволоки), які приблизно лінійно збільшують свій електричний опір із зростанням температури.

Термопари (металеві і напівпровідникові), що генерують ЕРС постійного струму, яка створюється при нагріванні місця з'єднання двох різнорідних металів або напівпровідникових матеріалів.

Термодетектора, який може використовувати розширення газу при нагріванні з переміщенням мембрани, що діє або на мікровимикач.

Джерелом теплового випромінювання може бути будь-який об'єкт з підвищеною температурою по відношенню для навколишнього середовища: вогонь, жива істота (людина), працюючий двигун, літак, електричні лампи (наприклад, з вольфрамовою ниткою у балоні з інертним газом (ксенон, кріптон)) і т. д. Балон електричної лампи з кварцевого скла є джерелом інфрачервоного випромінювання.

Датчики зображення і випромінювання:

Як датчики зображення використовують: вакуумні телевізійні передаючі трубки; твердотільні матричні фотодіодні та фототранзисторні перетворювачі зображення.

– **Фотодіод** на додаток до функцій звичайного діоду зменшує свій опір у прямому напрямі при збільшенні освітлення.

– **Фототранзистор** має перехід «коллектор-база», опір якого залежить від освітлення. Має підвищену чутливість до світла у порівнянні з діодом.

– **Фотоелектронний перемножувач** – найбільш чутливий прилад, який уловлює фотони. Виникає лавинне збільшення струму електродів, пропорційний яскравості освітлення.

– **Сітківка** (у розпізнаванні образів) – це набір світлочутливих елементів у вигляді матриці – пласкої поверхні (з фотодіодів, фотоелементів), на яку проектується оптичне зображення для перетворення у електричні сигнали.

– **Термопара** використовується для отримання ЕРС під впливом нагрівання від *оптичного випромінювання*.

Термодетектори реагують на підвищення температури під дією *оптичного випромінювання* зміною тиску газу у балоні і відповідно – зміною положення мембрани.

Розрізняють наступні класи твердотільних світлочутливих приладів:

– *квантові* (при поглинанні фотона змінюють електричні властивості півпровідника – до них відносяться фотодіоди, фоторезистори, прилади із зарядовим зв'язком);

– *теплові* (при поглинанні фотона змінюється ємність, величина опору внаслідок нагріву);

– *піроелектричні фотоприймачі*, які використовують тонкі шари кристалічних діелектриків зі зміною при нагріванні ємності, яка при постійній напрузі приводить до зміни заряду та струму. Перевагою піроелектричних приймачів є низька вартість та майже постійна спектральна чутливість в широкому діапазоні (800 нм ... 25 мкм);

– *термопари*, що складаються з металу та півпровідників.

Джерелом світла може бути сигнальна електрична лампа накалювання. Одним з джерел випромінювання є лазер та рентгенівська трубка.

Тактильні (контактні) датчики («штучна шкіра») потребують фізичного контакту між датчиком та об'єктом і використовуються у вузлах безпосереднього фізичного захвату об'єкта робочими захватними вузлами робота. Розпізнавальна спроможність тактильних датчиків досягає сотень точок на 1 см². Для двомірних тактильних датчиків використовують, наприклад, плівки з кремнієвих сполук, які виготовлені за тією ж технологією, що і інтегральні схеми [117].

Акустичні датчики поділяються на гучномовці, динаміки, ультразвукові датчики, мікрофони. Вони перетворюють звук у електричний сигнал і навпаки – електричний сигнал у звукові коливання.

Датчики хімічного складу. Звичайно до них відносять газоаналізатори: аміаку (при наявності відповідних холодильних установок), метану (у шахтах) тощо.

Поліметричні вимірювання фізичних сигналів полягають у тому, що за допомогою одного датчика вимірюють кілька (до 4-х) фізичних даних. *Поліметричне вимірювання* було створене в Національному університеті кораблебудування імені адмірала Макарова [25; 31]. Наприклад, при вимірюванні рівня рідини у паливних цистернах висотою у 15 м посилається зондуючий радіоімпульс порядку 0,5 нс (що відповідає частоті 2 ГГц) по хвильоводу зверху униз. Відзеркалений сигнал вказує рівень рідини, а його подальша обробка дозволяє встановити ще додаткові два–три параметри. Для характеристик рідких середовищ потрібно контролювати ряд різних параметрів, які залежать від частоти, температури та самого середовища і потребують ряду вимірювальних перетворювачів різних принципів дії. Для калібрування датчиків використовується порівнянням отриманого сигналу з взірцями від різних еталонів [25].

Наноскопічні сенсори відрізняються надзвичайною чутливістю, надійністю при мінімальних (молекулярних) розмірах. Створені датчики виявлення молекул або біологічних структур. «Електронний ніс» звичайно складається із сукупності датчиків і працює в режимі трьох етапів: 1) проба засмоктується відділення з лінійкою або матрицею сенсорів і вступає з ними у взаємодію; 2) реакція сенсорів аналізується й передається в процесорний блок; 3) проба виштовхується парами спирту. Наносенсори настільки чутливі, що можуть визначити присутність до однієї молекули.

Випробувані США під час останньої військової кампанії в Афганістані мікроскопічні датчики міліметрових розмірів під назвою «SmartDust» («розумний пил») показали свою ефективність (сигнали з великої кількості цих дешевих датчиків централізовано приймаються та аналізуються).

Інтелектуальні датчики – це нове покоління датчиків на базі нанотехнологій для вимірювання кількох параметрів паралельними датчиками, які вимірюють одну й ту ж величину. Датчик має власний комп'ютер, який контролює і коригує нелінійні характеристики та гістерезис, замінює несправні датчики на справні, забезпечує роботу в умовах перешкод, збоїв та виникнення дефектів без доступу до них персоналу. Інтелектуальні датчики призначені для космонавтики, ядерної енергетики та військових потреб.

1.5.2. Людина як датчик реальних та абстрактних ознак

Дані людини є суб'єктивними, залежать від людини, її стану та навичок, стану навколишнього середовища, числового значення вимірюваної величини. Людина має **датчики** – чуттєві органи, які пов'язують її з навколишнім середовищем:

1. *Зору* – для визначення часу (по навколишньому середовищу), розмірів, об'єму, кольорів, інтенсивності кольорів, нерухомості, швидкості і напрямку переміщення, контрастності, яскравості, структури.

2. *Слуху* – для визначення інтенсивності звуку, однотонності, гамм, музики, висоти звуку, шуму, безперервності, мовлення, співу.

3. *Дотику* – для визначення гладкості, липучості, грубозності, жорсткості, м'якості, твердості, шороховатості, пружності, тиску, сили, текстури, границі, легкості переміщення, розмірів, об'єму.

4. *Смаку* – для визначення приємності, неприємності, інтенсивності, специфічності, побічних ефектів.

5. *Нюху* – для визначення приємності, неприємності, інтенсивності, специфічності, побічних ефектів.

Потрібно ураховувати, що технічні датчики можуть мати *переваги перед людиною-датчиком* по швидкодії, точності, надійності тощо. Людина не може вимірювати деякі технічні дані (наприклад, магнітне поле, радіацію, рентгенівські промені тощо). Технічні датчики випереджають людину

по вимірюванню сили, швидкості, ЕРС, струму, температури, освітлення.

Людина є універсальним інтелектуальним датчиком. Для підтримки зв'язку з навколишнім середовищем людина застосовує:

– **розум**, який переробляє, об'єднує, фільтрує, виробляє і створює додаткову інформацію;

– **слово**, яке є класифікатором об'єктів та процесів навколишнього середовища.

У світі науки та при спілкуванні з ЕОМ потрібно мати мову однозначного тлумачення. З цією метою за допомогою методів **онтології та тезаурусу** створюють окремі чисельні формати природної мови, які стосуються окремих напрямків знань і разом з тим є несуперечливими, однозначними за термінологією.

Онтологія в інформатиці використовується **для припинення хаосу в термінології**, створення для людей та ЕОМ однозначної спільної мови. Онтології містять *концепти* (поняття, класи, сутності, категорії), *властивості концептів* (слоти, атрибути, ролі), *метаописи відносин* між концептами, *метаописи структури*, *метадані* та *взаємні зв'язки* термінів і відображення сенсу об'єкта в семантичних метаописах.

Тезаурус є поняттям, вужчим за онтологію: це *предметно або проблемно-орієнтований словник*, що містить в собі термінологію предметної ділянки, визначає відносини між поняттями і містить типізовані зв'язки. При цьому природну мову можна поділити на окремі компоненти (об'єкти, процеси, ознаки та ситуації). Тезаурус є формально-мовною системою для побудови довідкових систем (індексування), або для наступного ефективного доступу до інформації. *Тезаурус* вичерпним чином покриває деяку специфічну галузь знань і відображає смислові відносини як ієрархічного типу (наприклад, «родо-видові»), так і неієрархічного типу. Зафіксований набір *мовних одиниць* інформаційно-пошукового тезаурусу може бути прирівняний до деякої *інформаційно-пошукової мови, до формально-мовної системи* [36].

Лінгвістичне конструювання мови. Однією з фундаментальних робіт з систематизації та класифікації словників є робота Ю. Н. Караулова [36]. У цій роботі запропоновано систему лексикографічних параметрів словників у вигляді **«матриці покриття»**, в якій виділені такі особливості:

1. *Перелік параметрів* – стовпці матриці (виділяється 67 лексикографічних параметрів).

2. *«Поле можливих рішень»* є переліком рядків.

Сукупність заповнених комірок на перетині рядів і стовпців «матриці покриття» дає матеріал для комбінування параметрів при конструюванні нових лексикографічних об'єктів.

Така система параметрів є достатньою *для опису всіх існуючих сьогодні словників*. При розпізнаванні та класифікації образів отриманий СИГНАЛ чи ІНФОМАЦІЮ можна вважати СЛОВОМ. У цьому випадку можна, у числі інших напрямків, використати методи онтології та тезаурусу.

1.5.3. Ознаки маркетингового аналізу

Для оптимального реагування на зміну ринкових умов, підприємцю потрібна точна інформація про ринкові проблеми і про способи їх подолання (*власних знань та інтуїції тут недостатньо*).

Економічна інформація використовується для оптимізації власних дій на ринку, а не для «перевиховування покупців» (вважається, що це або неможливо або занадто дорого, хоча реклама та мода саме на це й спрямовані). Маркетинг охоплює взаємно корисну ринкову діяльність при переміщенні послуг від продавця до покупця (рис. 1.5.1).

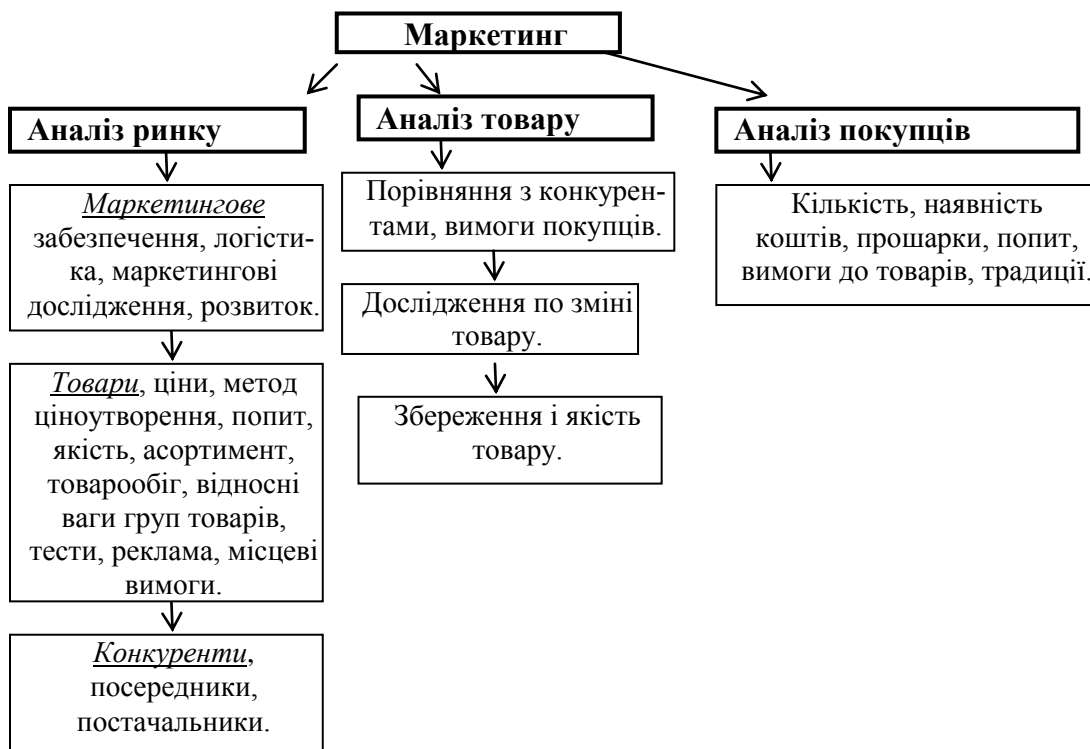


Рис. 1.5.1. Ознаки маркетингового аналізу

Маркетинговий аналіз оперує групами товарів, тому потрібно знати їх торговельну класифікацію.

Товари за ознакою призначення розділяються на групи: споживчі товари, товари промислового призначення та ін.

Під асортиментом товарів розуміють перелік, сорти, різновиди та групи товарів. Розрізняють вузький виробничий асортимент і значно ширший торговий асортимент. Товари поділяється на галузі, групи, підгрупи, види.

Чим ширший асортимент, тим більший прибуток. Виникає необхідність *перетворення виробничого асортименту в торговий*, що здійснюється в роздрібній торговельній мережі, але основна маса товарів підлягає сортуванню переважно на підприємствах оптової торгівлі.

Асортиментна політика підприємств відіграє важливу роль у конку-

рентній боротьбі між торговельними підприємствами за покупця. Чим ширший асортимент, тим міцніші конкурентні позиції магазину на ринку. Торговий асортимент формується на основі попиту.

Додатково до розглянутих «офіційних» ознак, товари можуть групуватися за поточними ознаками, прийнятими для магазину, або введеними дослідником:

- за умов збереження товарів, їх якості та сезонності;
- по наближенню критичних термінів продажу;
- по взаємних зв'язках при продажі;
- по попиту населення тощо.

1.5.4. Ознаки ризиків фінансових операцій

Менеджер відповідає за неповернені гроші.

Перед прийняттям рішення по кредиту банк класифікує позичальників за групами ризику та визначає: чи видавати кредит, обсяг кредитування і відсотки для клієнта. Від цього залежить прибутковість банку. Намагаються побудувати в банку таку технологію автоматизованої перевірки і оцінки кредитоспроможності фізичних осіб та прийняття рішень, яка забезпечить *мінімальний відсоток неповернення кредитів* позичальниками.

Надавати кредит можна підприємству або приватній особі. Аналіз ризиків при цьому відрізняється: у першому разі потрібно збирати ознаки по підприємству, у іншому випадку – про особу.

Найбільш значущим і важко керованим є кредитний ризик. Невиправданий ризик може призвести до банкрутства і тому в фінансових операціях ризик необхідно передбачити. У той же час вважається, що не потрібно відхиляти вигідну пропозицію через великий ризик.

Ми розглянемо ознаки ризику при наданні позики під визначений відсоток особі (рис. 1.5.2).



Рис. 1.5.2. Система розпізнавання ризиків надання кредитів банком
Зазвичай, оцінка кредитного ризику виконується за автоматизованою

спеціальною програмою по введеним менеджером даним, які охоплюють 10–12 базових параметрів, наприклад, *сімейний стан, наявність власності, частота зміни місця роботи та ін.* Менеджер по кредитах повинен бути психологом, щоб розпізнати, чи можна людині довіряти.

Додатково до даних співбесіди у програмі використовують *технології детектора брехні*: впевненість позичальника, тембр голосу, паузи, відведення очей при питаннях, напруженість, вираз обличчя та очей.

Ця система аналізує відповіді на питання, присвоює позичальникові деяку кількість балів і, залежно від їх загальної суми, визначає свою поведінку по відношенню до клієнта: *давати кредит, чи не давати; величину кредиту; відсотки; термін повернення.*

Тестування займає 5–7 хвилин. Дослідження показали, що застосування автоматизованої системи, дає можливість одному диспетчеру обробляти до 10 заявок на годину при неповерненні кредитів до 2 %.

1.5.5. Визначення інтелекту, психічних даних та характеру людини

Найм персоналу, кадрові рішення. Потрібно розуміти, що ліпші працівники не шукають працю, вони її вже мають. Щоб підприємство було успішним, потрібно найняти людей з оцінкою 100 по 100-бальній шкалі. Розрахунки підтвердили, що менеджер по персоналу *приносить фірмі прибуток, рівний 20–25 %* від зарплати працюючого. Тому витрати на тестування нових робітників є доцільними.

Відомі різні напрямки розпізнавання образу людини, які спираються на різну систему ознак. Психологи, лікарі, військові, роботодавці, науковці, педагоги, тренери з фізкультури з різних боків оцінюють і класифікують людину. Для аналізу однієї особи звичайно призначають деякий час і використовують сучасне програмне забезпечення, наприклад, комплекс комп'ютерних психодіагностичних методик («EffectonStudio: Управління персоналом. Психологія»).

При наймі персоналу намагаються вивчити психологічні та професійні якості, фізичний стан, потенціал, ділові якості, акуратність, чіткість. Найм складається з ряду етапів: співбесіда для попереднього відсіювання, заповнення заяв і анкет; перевірка, хторекомендує претендента; перевірка паперових даних; тестування, аналіз даних, складання прогнозів стосовно нового робітника; медичний огляд, прийняття рішення.

Людина може проходити випробування (тобто вона класифікується, розпізнається) на індекс інтелекту, на розумові здібності (тестування – як в школі), на кваліфікацію, на працелюбність, на порядність, на брехливість, на самооцінку, на доброзичливість, на детекторі брехні тощо. Кожний з цих напрямків має *власну, іноді достатньо складну методологію та си-*

стему ознак. Усі існуючі напрямки розглянути неможливо.

Інформація про характертапсихічний стан людини отримується при спілкуванні. Об'єм отриманих розподіляється наступним чином:

- 7 % – отримується при вербальному спілкуванні (у розмові);
- 35 % – за рахунок аналізу модуляції голосу;
- 55 % – по виразу обличчя. Тобто обличчя дає найбільш важливу інформацію стосовно стану людини, його фізичного та психічного здоров'я [177].

У претендентів шукають наступні якості:

1. *Історію успіху.* Намагаються виявити успіх в будь-якому напрямку: в роботі, в житті, в хобі, в лідерстві. Людина, що добре працювала у минулому, буде добре працювати й у майбутньому.

2. *Інтелект* виявляють тестуванням. При інших рівних умовах ліпше найняти розумну людину.

3. *Енергія.* Часто її характеризує хобі, спорт, бо вони означають і наявність надлишкової енергії.

4. *Характер* визначається по кредитній історії, рекомендаціях, медичному обстеженні (включаючи перевірку на наркотики та алкоголь, бо серед робітників не повинно бути наркоманів та п'яниць). Але тут бажано знати більше: чи достатньо претендент наполегливий, чи пишається власними досягненнями і в чому вони полягають, чи допомагає комусь з близьких.

5. *Тестування.* Якщо потрібен механік, то бажано знати, чи любить він працювати у гаражі, які технічні журнали читає, чим займаються батьки. Тест не завжди працює, але все ж поліпшує результати: до тестування правильний відбір звичайно складає 25 %, а після тестування – 50 %. Але можна вважати, що тести дають близько 30 % відомостей про претендента. Значно більше дає бесіда. Тому бесіду проводять кілька разів. На завершення можна організувати сніданок з парою співробітників для перевірки, наскільки їм подобається бути разом з ним.

У залежності від посади встановлюється *термін випробування претендента*. Приклади оцінки психічному образу людини за допомогою *тестування*:

1. *Тестування об'єму оперативної пам'яті.* Пропонується 5 однозначних чисел (9, 3, 7, 1, 2). Їх слід запам'ятати, потім в умі скласти і ввести в комп'ютер (всього 4 цифри): перше з другим; друге з третім; третє з четвертим; четверте з п'ятим. Відповіді переводяться у бальні оцінки.

2. *Тестування логічної пам'яті.* Показується або зачитується коротка розповідь. Розповідь розділена на частки з виділенням ключових слів (до 16-ті ключових слів). Розповідь треба відновити із застосуванням всіх виділених логічних одиниць. Відповіді переводяться у бальні оцінки.

3. *Тестування механічного запам'ятовування.* Послідовно пропонується 10 двозначних чисел (29, 53, ...). Їх треба запам'ятати і потім ввести в комп'ютер. Відповіді переводяться у бальні оцінки на основі двох проб.

4. *Тестування об'єму образної пам'яті.* Одночасно показують 9 геометричних фігур на протязі 40 сек. Потім показують таблицю з 28 фігур,

серед яких є й показані фігури. Треба помітити показані фігури. Відповіді переводяться у бальні оцінки.

5. **Тестування** (іспити) на фахові знання.

У разі потреби, проводять тестування на знання англійської мови, комп'ютера, на вміння ведення перемовин тощо.

Створений поліграф (детектор брехні), який застосовується при розкритті злочинів; для контролю правдивості свідків; при виконанні судової експертизи; при кадровій перевірці робітників і кандидатів на вакантні посади та навчання (наприклад, у підрозділи служби національної безпеки). Кожний поліграф має власну програму навчання. Поліграф може фіксувати ознаки **психосемантики**: кардіограму; зміну тиску крові; зміну шкірогальванічної реакції (появу «гусячої шкіри» та ін.); зміну дихання (грудини та брюшини); збудження вегетативної системи (фізіологічні зміни в організмі); ресстрацію тремору; рух м'язів; зміну кольору обличчя; зміну напрямку погляду, діаметру зіниць тощо. На базі нанотехнологій створюються датчики емоцій (виявлення страху, радості).

Психометрія виміряє знання, здібності, погляди та моральні якості людини шляхом застосування теорії, методів та алгоритмів психологічних вимірів за рахунок таких «**вимірювальних інструментів**», як *переліки питань, тести, інтерв'ю і методик оцінки особи* в основному по отриманих дотичних відомостях. **Вважається, що немає різниці між психометричними вимірювальними інструментами та фізичними приладами.**

Дані психометрії використовуються при проведенні соціологічних досліджень, при прийнятті на роботу, створенні колективів спортсменів, космонавтів тощо.

«Батьком психометрії» вважається Френсіс Гальтон, який ставив за мету вимірювати інтелект людини (зараз «*Індекс інтелекту*» широко використовується для тестування). Психометрист Луїс Тьорстоун розробив теоретичний підхід до психометричних вимірів, який зветься «законом порівняльних суджень». Тести проходять «**нормування**», яке забезпечує приведення всіх форм тестування до загальної шкали. Психометрична теорія розробила методи аналізу відповідей, окремих тестів та наборів тестів. Психометричний аналіз забезпечує надійність отримуваних висновків та оцінок і відповідність поставленій меті. В Ізраїлі, наприклад, введений обов'язковий психометричний вступний іспит у ВНЗ, який проводиться окремим «Центром екзаменів» і складається з тестів на словесне (вербальне) мислення, на математичні здібності та на англійську мову.

1.5.6. Застосування класифікаторів

Класифікатор – це державний, галузевий чи локальний документ, що визначає поділ об'єктів на підмножини у вигляді класу, підкласу, групи, підгрупи, виду, підвиду, типа.

При проектуванні системи РО потрібно використовувати існуючі класифікатори. Це не означає «заборону» на введення в систему РО власної

класифікації. Але нові позначення не повинні вносити хаос у систему загальновідомих визначень.

Найбільш поширені такі методи класифікації об'єктів:

– **Ієрархічний метод** класифікації характеризується тим, що універсум об'єктів спочатку поділяється на класи першого рівня ієрархії, потім на угруповання наступного рівня та ін. Утворюється граф у вигляді ієрархічної деревоподібної структури. При групуванні об'єктів додержуються двох правил:

– починають розподіл об'єктів на групи за найбільш загальними важливими ознаками;

– розділ об'єктів на групи на одній стадії виконується тільки за однією ознакою.

Перевагами ієрархічної системи є простота побудови та використання, а недоліками є жорстка структура, яка ускладнює внесення змін, та неможливість групувати об'єкти за непередбаченим поєднанням ознак.

Фасетний метод класифікації, на відміну від ієрархічного, не ураховує семантику об'єкта. Для кожного об'єкта задається множина фасет. Кожна фасета – це список припустимих значень однієї ознаки.

Перевагами фасетної системи класифікації є можливість використання великої кількості ознак класифікації та можливість простої модифікації всієї системи класифікації, а недоліком є складність її будови.

Дескрипторна (описова) система класифікації використовує, наприклад, мову *тезаурисів*, яка схожа на природну мову. Суть цієї системи класифікації полягає в тому, що:

– створюється сукупність **дескрипторів** *ключових слів або словосполучень*, що описують універсум і серед них вибирається найбільш вживаний (це зветься *нормалізацією*);

– між дескрипторами встановлюються зв'язки, які дають змогу розширяти межі пошуку інформації. Зв'язки можуть бути трьох типів:

▪ **синонімічні** – визначають сукупності близьких за значеннями ключових слів (*студент <-> учень <-> відмінник <-> школяр*);

▪ **родові** – відображають включення певного класу об'єктів до іншого як підмножини (*ВНЗ <-> університет <-> факультет <-> кафедра*);

▪ **асоціативні** – мають спільні властивості (*студент <-> іспит <-> професор <-> аудиторія <-> ВНЗ*).

Класифікація пов'язується з метою:

– *Для підприємств та оптових складів лікарських товарів метою є вихідний товар і матеріал для його випуску: лікарські засоби; тара, закупорювальні засоби і пакувальні матеріали; перев'язувальні засоби і матеріали; предмети санітарії, гігієни і догляду за хворими; хімічні реактиви; дезінфекційні засоби; товари обмеженого аптечного асортименту (мінеральні води, медичні п'явки, миючі засоби й ін.).*

– *Для аптек метою є зв'язок ліків з хворобою (ліки від головного бо-*

лю, шлункові, протизапальні, або перев'язувальні засоби, предмети догляду за хворими та ін.), що забезпечує зручність розміщення, замовлення, контролю комплектації ліків за їх призначенням.

– Для торгівлі метою є раціональність організації торгового процесу. Головною ознакою для них є призначення товару на основі галузевих класифікацій продукції та стандартів (відділи виробів: хлібо-булочних, м'ясних, рибних, молочних тощо).

– Для економіко-статистичних класифікацій (по класифікаторах ЦСУ України) метою є зручність при плануванні виробництва, постачанні, для обліку продукції. Як правило, для них застосовується сукупність ознак, яка задовольняє вимоги виробництва, планування, постачання і статистичного обліку.

1.5.7. Ознаки соціологічного стану

Соціологічні дослідження суспільства спрямовані на розв'язання соціальних, економічних і політичних проблем та запобігання соціальної напруженості.

Соціологічних дослідження суспільства вимагають знань про (рис. 1.5.3):

1. *Ресурси*, які має держава (прошарки населення, розподіл майна серед прошарків населення – землі, власності, заощаджень).

2. *Думки* різних прошарків населення про напрямок дій, або про обрану тему обговорення.

3. *Напруженість у суспільстві*, яка оцінюється по рівню безробіття, зарплати, податків, поділу прибутків, наповненню «споживчого кошику». Вважається, що у суспільстві не виникає соціальної напруги, якщо заробіток 10 % робітників з високою оплатою в 4–7 разів перевищує заробіток 10 % робітників з низькою оплатою.

Конфліктні ситуації у виробничих колективах зростають із зростанням напруженості у суспільстві і спостерігаються між окремими персонами та соціальними групами. *Причини конфліктних ситуацій* у колективах можна поділити на частки:

1. Наявність різних цілей, різної влади, різного сприйняття ситуації, різного бачення виробничого процесу.

2. Умови праці; оцінки завантаження; мотивація; кар'єра.

3. Неспівпадіння власної самооцінки у порівнянні з існуючою оцінкою окремих осіб або груп людей у вигляді різниці в заробітній платі, в оцінці вкладу в роботу, в посадах, в результатах роботи, в переміщенні по службі, в преміях, у службових відносинах тощо.

4. Персональні: отримана образа, людська заздрість, несумісність характерів, особисті якості. Ці конфлікти розв'язуються найбільш тяжко, і в них потрібно чітко дотримуватись Закону.

5. Використання керівником службового становища для особистої ви-

годи. Наприклад: керівники намагаються мати навколо себе менш значущих заступників; ставити на значущі посади родичів чи знайомих; ставити перед підлеглими тяжкі задачі, які перетворюють роботу на важкий тягар: при Радянській владі такі керівники першими доповідали «про виконання та перевиконання плану хлібозаготівель», залишаючи працівників без хліба.

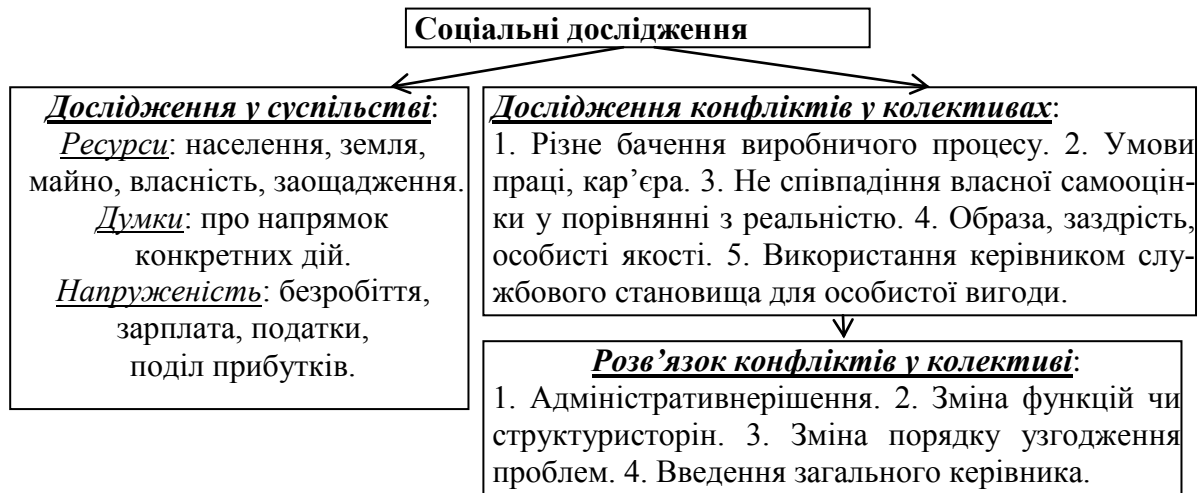


Рис. 1.5.3. Напрямки соціальних досліджень і ознаки

Конфлікти призводять до зниження ефективності роботи, до зростання витрат ресурсів та ризиків у виробництві послуг. В результаті роботи наноситься не завжди помітний збиток, який іноді є вирішальним в умовах конкуренції.

Конфлікти заважають нормальному виробничому процесу, бо ефективна сумісна робота припиняється і замінюється «ворожими» діями. Часто одна із сторін *обирає тактику*: «роботи за правилами»; формального (не по суті) виконання рішень; «затягування часу» у виконанні роботи; пошуку і використанню засобів «захисту» від «нападів» (ненадання своєчасної інформації; обмеження «зворотного зв'язку» і сумісної роботи в одному напрямку; неприйняття або не своєчасне прийняття рішень; погіршення якості роботи; навмисного пропуску ситуацій підвищення ефективності роботи).

Конфлікти мають схильність до поширення і охоплення все більшої кількості людей. Виникає проблема виявлення наявності конфліктів і їх ліквідації.

Для виявлення наявності внутрішніх конфліктів можуть бути досліджені ряд дотичних показників (ознак):

1. Рівень заробітної плати робітників.
2. Завантаження, вимоги та мотивація робітників.
3. Загальна плинність кадрів.
4. Збільшення кількості звільнених працівників «за власним бажанням» чи «за згодою сторін».
5. Кількість нарад стосовно конфліктів та їх рішення.
6. Дисципліна керівників та робітників.

7. Конкретні нові управлінські рішення з новими вимогами.

Розв'язують конфлікти наступними методами: адміністративним рішенням; зміною і перерозподілом функцій чи структури сторін; зміною порядку узгодження проблем; введенням загального керівника тощо.

1.5.8. Ознаки лінгвістичного аналізу текстової інформації та мовлення

Слово як ідеальна та матеріальна основа інформатики.

Всі наші знання отримуються через слово, яке має дві форми: форму матеріального відбитку (у вигляді людської та електронної пам'яті, письменності, книжок, Інтернету, преси тощо) та ідеальну понятійну форму, яка створюється системою навчання і зберігається у пам'яті людей як матеріальна сукупність навчених нейронів. Діями окремої людини, фірми та держави керують ідеальні у природі не існуючі ЗНАННЯ, основою якого є СЕНС СЛОВА. Нами керує ДУХОВНА ФОРМА СЛОВА, яка хоча і «підтверджена практикою» та «має матеріальну форму навчених нейронів», але реально не існує у природі, є моделлю, має ідеальний вміст як і все «зовнішнє середовище» у свідомості людини. СЛОВО є ідеальним найбільш поширеним класифікатором, яким керуються люди і інші живі істоти.

Текстова інформація.

В автоматизованих інформаційних системах по обробці тексту в якості «датчиків» лінгвістичних ознак тексту чи речень виступають підпрограми, які визначають міри близькості та відстані між окремими словами та реченнями. Для цього створюється сукупність *дескрипторів* (ключових слів або словосполучень), які описують універсум і серед них вибирається найбільш вживаний (це зветься *нормалізацією*). Між дескрипторами встановлюються *зв'язки: синонімічні* (визначають сукупності близьких за значеннями ключових слів: *студент <-> учень <-> відмінник <-> школяр*); *родові* (відображають включення певного класу об'єктів до іншого як підмножини: *ВНЗ <-> університет <-> факультет <-> кафедра*); *асоціативні* (мають спільні властивості (*студент <-> іспит <-> професор <-> аудиторія <-> ВНЗ*)).

Для обробки текстової інформації використовують спеціальну мову програмування (наприклад, Scala). Міра семантичної близькості для пари слів природної мови (це кількісна величина, яка показує наскільки два поняття близькі, тобто пов'язані або схожі між собою) використовується при виконанні семантичного аналізу текстів, розв'язанні смислових неоднозначностей [4].

Семантична схожість подібна до *синонімії*, наприклад: шофер і водій.

Обчислення *семантичної близькості* для пари слів природної мови виконується за двома методами:

– «методами, заснованими на відстані між концептами»: у ієрархічній семантичній мережі відстань вимірюється кількістю вершин у найко-

ротшому шляху між парою слів; урахуванням для пари слів в ієрархії спільного предка з найменшою глибиною;

– *«методи, засновані на лінгвістичному оточенні (лексичному перетині)»*, який обчислюється як кількість спільних слів двох означень [4; 174; 188; 205]. Лексичний перетин отримується у припущенні, що пов'язані поняття повинні пояснюватись однаковими словами (контекст цільового слова визначається по словам супроводу в межах заданої кількості позицій). Далі обирається значення слова з найбільшим сумарним перетином, і вважається, що це слово найбільше відповідає контексту.

– Ноам Хомський доводив, що для розуміння мови потрібно урахувати такі суттєві властивості, як *структуру (відношення між частинами речення), правила та граматику*. З плином часу виявили, що ряд лінгвістичних понять не потрібні. Тепер для обробки текстової інформації все більше схиляються до семантичних ознак та зв'язків між словами (це слабка сторона теорії Хомського).

– *Тексти* відрізняються наборами *ключових слів* і є свідченням *грамотності* висловлення думок [123].

Інформація мовлення.

Мовлення має *акустичні характеристики* та текст з відповідними *ознаками текстової інформації* [96]. Тому лінгвістична інформація по сприйняттю мови поділяється на ієрархічні одиниці *фонем (звуку), складу, слова, речення та тексту*. Одиниці нижчої ієрархії є елементами більших одиниць сприйняття, але кожна з них не має властивості цих більших одиниць [123].

Фонемі відповідає набір акустичних ознак, взаємозв'язок фонем з іншими фонемами, розпізнавальних ознак звуку. Звукові характеристики охоплюють тональність, розділ речення на слова, наголос на окремих словах та фонемах, розклад речення та окремих фонем на сукупність гармонік або рівнів інтенсивності та терміну луни звуку.

Для слова лінгвістичними ознаками виступають його граматичні категорії (частина мови, відмінок); довжина слова в складах, фонемах; ударна фонема; ударна голосна; морфеми (морфема – це найменша одиниця мови, що має деякий сенс); ритмічність структур; ударний голосний; початковий звук; частотність слова; довжина слова в складах.

Речення відрізняються порядком слів, заставою, довжиною речення, твердженням, запереченням, поширеністю, нерозповсюдженням, частотністю, *емоційним значенням*. Банківські автоматизовані системи особливу увагу приділяють саме емоційному вмісту речень при перемовинах по видачі кредиту та по прийому на роботу.

Тексти відрізняються наборами *ключових слів* і є свідченням *грамотності* висловлення думок [123].

1.6. Обробка сигналів

1.6.1. Початковий вибір числових змінних образу

Образи звичайно розглядають як деяку вихідну функцію $y(X)$, де $X=(x_1, x_2, \dots, x_j, \dots, x_n)$ – вектор вхідних змінних.

З метою зменшення вартості системи РО при її проектуванні намагаються обмежити загальну кількість змінних. У ряді випадків зменшення вартості системи РО має вирішальне значення при створенні системи РО. Ознаки можуть позначатись цифрою, переліком, лінгвістичною назвою, діапазоном, порівнянням у відносних одиницях між «еталоном» і «браком». Але завжди потрібно розв'язати проблему використання ознак в арифметичних операціях.

Необхідність переробки ознак об'єкта пояснюється намаганням отримати ознаку з більшою інформативністю. Для спрощення процедури розпізнавання літака, ознаку можуть отримати, наприклад, складанням/множенням довжини корпусу літака і довжини його крил, або шляхом сумісної математичної обробки інших ознак. Іноді розраховують взаємну кореляцію між двома функціями з метою встановлення їх ідентичності до еталонних образів.

Ознаки X намагаються замінити більш простими з визначенням об'єднуючих змінних. Іноді даних не вистачає, або їх не можуть безпосередньо виміряти. Тоді їх намагаються визначити за допомогою другорядних побічних ознак.

Обробка числових, нечислових, недостовірних та нечітких даних

Вхідні числові дані масштабуються по лінійній шкалі у придатний для мережі діапазон. Більш важкою задачею є робота з даними нечислового, недостовірного та нечіткого характеру.

Для переведення нечислової лінгвістичної змінної у числове значення можна виконати наступні дії:

1. Якісні ознаки перетворюють у число в діапазоні 0...1 або 0...100 (наприклад, ознака «привабливий» в діапазоні 0...1 може мати надану експертом оцінку 0,65 для конкретного об'єкта).

2. Двозначні лінгвістичні твердження перетворюють в числові змінні. Наприклад, *Стать = (Чоловік, Жінка)* перетворюють на (*Чоловік = 0, Жінка = 1*).

3. Застосовують поняття «наявності/відсутності ознаки» у вигляді цифр «+1 /-1» або «+1 / 0» (наприклад, у яблука «колір червоний» може мати числове значення 1 або 0).

4. Якщо потрібно визначити вартість об'єктів нерухомості у залежності від районів міста, то експерт ранжує райони міста з позначенням вздовж лінгвістичної осі числових рангів районів міста.

5. Використання *метода заміни лінгвістичної змінної* (текстового поля), наприклад, «Прізвище працівника», на кілька відповідних змінних, які

дорівнюють елементам списку лінгвістичної змінної із застосуванням щодо цих елементів числового поняття «наявності/відсутності ознаки» (1\0).

6. Нечисловими даними (текстовими полями, лінгвістичними змінними) можна нехтувати при їх незначному впливі на вихідне значення.

7. Значення дати і часу можна розділити на дві незалежні ортогональні змінні (або скоротити одну з них).

8. Нечіткі лінгвістичні змінні з мірою впевненості можна перевести у числові значення або за методом Л. Заде [33], або за методом М. В. Турти [115].

Факторний аналіз

При умові наявності множини навчальних об'єктів з наявним та відсутнім фактором x_j можна отримати оцінку впливу вхідної змінної x_j на середнє значення вихідної величини по дисперсії, викликаній відсутністю фактора $a_j x_j$ (тут a_j – деякий невідомий коефіцієнт)

$$S_j^2 = \frac{1}{N} \sum_{j=1}^N (y_j^* - y_j)^2,$$

де y_j^* , y_j – значення середньої величини відповідно при відсутності та наявності фактора $a_j x_j$.

Чим менша дисперсія S_j , тим менше впливає складова $a_j x_j$ на вихід. При малих значеннях дисперсії відповідний фактор вилучається з аналізу.

Кореляційний аналіз

При наявності N реалізацій $y_i(x_{1i}, x_{2i}, \dots, x_{ji}, \dots, x_{ni})$, $i = 1, 2, \dots, N$, можна визначити взаємну залежність між кожною з вхідних змінних $X=(x_1, x_2, \dots, x_j, \dots, x_n)$ і вихідною величиною y , *по коефіцієнту парної кореляції* (приклад наведений для вхідної змінної x_1 та вихідного значення y)

$$k_1 = \frac{\sum_{i=1}^N (x_{1i} - \bar{x}_{1i})(y_i - \bar{y}_i)}{\sqrt{\sum_{i=1}^N (x_{1i} - \bar{x}_{1i})^2 \sum_{i=1}^N (y_i - \bar{y}_i)^2}},$$

де $j = 1$ – порядковий номер змінної x_j ; $\bar{x}_{1i} = \frac{1}{N} \sum_{i=1}^N x_{1i}$; $\bar{y}_i = \frac{1}{N} \sum_{i=1}^N y_i$; $i = 1, 2, \dots, N$ – порядковий номер реалізації; N – число реалізацій.

Значення $k_j < 0,6$, $j = 1, 2, \dots, n$, вважають за порогове. Наприклад, по коефіцієнтам кореляції між виходом та входами k_j виконують ранжування входів з вилученням несуттєвих вхідних змінних.

Коефіцієнти зважування

При розпізнаванні образів суттєвим є врахування реального впливу кожного параметра входу $X=(x_1, \dots, x_n)$ на вихідний вектор y . Урахувати цей вплив можна за допомогою коефіцієнтів зважування. Для цього початково обчислюються коефіцієнти парної кореляції k_j між виходом y та кожним з входів $(x_1, \dots, x_j, \dots, x_n)$. Для найбільш впливової змінної x_j (з найбільшим коефіцієнтом парної кореляції k_j) експерт призначає коефіцієнт зважування $K_{ff} = 1$, а для інших змінних $K_{ff} = 0 \dots 1$. У розрахунку вихід y^* розраховується по змінним $(K_{f1} x_1, \dots, K_{ff} x_j, \dots, K_{fn} x_n)$. Тобто всі змінні, крім однієї, зменшують свої значення. Такий підхід вимагає проведення попереднього кореляційного аналізу, але значно підвищує ефективність роботи системи РО.

1.6.2. Попередня обробка числових сигналів: нормалізація, квантування, фільтрація, диференціювання, фрагментизація, об'єднання, згладжування ознак

Отримані системою РО через датчики ознаки образу, не завжди придатні або не завжди зручні для аналізу. Тому ці дані іноді перетворюють для подальшого використання у системі РО. Типові методи обробки сигналів умовно назвемо нормалізацією, квантуванням, фільтрацією, фрагментизацією, диференціюванням, об'єднанням ознак. **Нормалізація ознак.** Припустимо, що з аеропорту спостерігають за літаком, який наближається до нього. При такому наближенні розміри літака на екрані локатору збільшуються. Але система РО повинна розглядати його дійсні розміри. Тому отримані розміри потрібно *нормалізувати* – привести їх до дійсних розмірів, що надає можливість порівнювати їх з еталоном.

Аналогічним чином:

– у процесі автоматизованої перевірки купу банкнот, висота цієї купу зменшується, купюра сприймається оптичною системою «зменшеною» у розмірах – потрібна нормалізація для порівняння з еталоном;

– при отриманні кардіограми потрібно виконати нормалізацію – привести амплітуду кардіограми та час скорочення – розширення серця до еталонних значень і лише потім порівнювати кардіограму з еталоном;

– при розпізнаванні автомашин за допомогою індукційної котушки, вкладеної у шлях. величина ЕРС, що наводиться у котушці, залежить від швидкості руху транспорту, і тому осцилограма ЕРС потребує нормалізації з метою розгляду образів в однакових умовах.

Квантування безперервних ознак. Квантування (дискретизація) сигналів дозволяє отримати зображення у вигляді матриці цілих чисел. Відновлення безперервної функції по дискретним значенням яскравості елементів зображення виконується інтерполяційними методами.

Припустимо, що в сучасній лікарні ЕОМ стежить за температурою хворого і дає поради медсестрі за такими *рівнями квантування* по температурі:

1. При температурі вище $38,5^{\circ}\text{C}$ ситуація аварійна, і медсестра повинна постійно знаходитись біля хворого.
2. При температурі вище $37,5^{\circ}\text{C}$ потрібно перевірити стан хворого.
3. При температурі нижче $35,5^{\circ}\text{C}$ потрібно перевірити стан хворого.

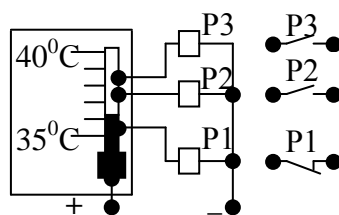


Рис. 1.6.1. Ртутний термометр з електричною сигналізацією

В цьому випадку кажуть, що сигнали потрібно квантувати. Таким чином, ЕОМ має справу не з нескінченною кількістю сигналів, а лише з трьома строго визначеними сигналами, які до того ж потребують заздалегідь визначеної дії. Знати точну температуру хворого немає потреби: достатньо знати, чи не перейшла температура тіла у небажаний діапазон. Такою є і властивість людини: на нескінченну сукупність зовнішніх образів вона відповідає кінцевою кількістю реакцій. Приклад такого квантування показаний на рис. 1.6.1 у вигляді ртутного термометра з електричною сигналізацією за допомогою реле Р1, Р2, Р2. Ртуть є електропровідним матеріалом.

Коли температура підвищиться від нормальної до $37,5^{\circ}\text{C}$, то реле Р2 отримає живлення, його контакти Р2 замкнуться і передадуть сигнал про підвищену температуру. Якщо температура підвищиться більше за $38,5^{\circ}\text{C}$, то отримає живлення реле Р3, його контакти Р3 замкнуться і передадуть сигнал про аварійну ситуацію. Якщо ж температура хворого стане нижче за $35,5^{\circ}\text{C}$, то реле Р1 втрачає живлення, його контакти замкнуться і передадуть сигнал про необхідність перевірки стану хворого.

Інші приклади квантування ознак образу:

– напрямки переміщення роботів квантуються у просторі площини дискретними кутами;

– якщо яблука сортуються по діаметру, то це робиться з урахуванням квантування;

– захисні електроприлади при перевантаженні у 5–10 % передають сигнал через 5–30 сек, а при перевантаженні у 10–15 разів спрацьовують миттєво.

Оптимальне значення рівня квантування відповідає середині інтервалу квантування: у цьому випадку максимальна похибка квантування всередині інтервалу не перевищує половини інтервалу квантування.

Фільтрацію даних виконують шляхом вилучення *шумів (перешкод)* з метою отримання достовірного зображення.

Диференціювання ознак виконують для «загострення» величини ознаки (це відноситься до похідних по отриманих кривих яскравості, зміни контрастності зображення, збільшення/зменшення числових ознак і т. д.).

Фрагментизація та об'єднання фрагментів ознак. Перетворення зображення у бінарну форму звичайно супроводжується розкладом зображення на окремі *фрагменти* (сегменти та непохідні елементи) з наступним *об'єднанням фрагментів* у більш складні форми. Універсального методу фрагментизації не існує. Звичайно використовується порогова інформація, наприклад, по яскравості; товщині, кривизні, напрямку лінії; розмірам і т. д.

Статистична обробка отриманих сигналів з наступним розкладом у ряд Фур'є допомагає виділити реальний сигнал без перешкод і збільшити надійність роботи системи РО.

Викиди, тобто значення, які лежать далеко від ділянки нормальних значень змінної і можуть суттєво спотворити образ. Викиди замінюють апроксимуючими значеннями з урахуванням сусідніх значень, або за середніми величинами.

Пропущені значення відновлюються спеціальними методами і можуть бути замінені на середнє значення цієї змінної чи на інші її статистики.

Згладжування звичайно полягає в обробці сусідніх ознак, які не повинні суттєво відрізнятись.

1.6.3. Контрольні карти процесів

Контрольні карти використовуються для візуального виявлення порушень технологічного процесу при зіставленні вимірних даних з дозволеними (контрольними) межами (рис. 1.6.2).

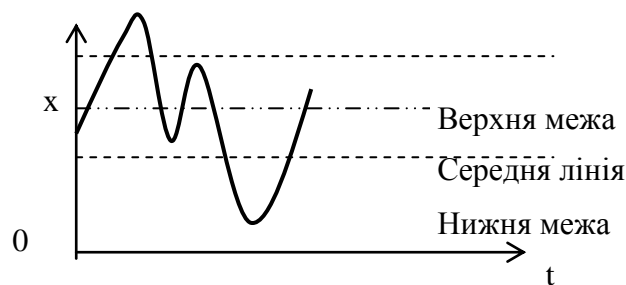


Рис. 1.6.2. Контрольна карта кількості випущеної продукції x у залежності від часу t

Контрольні карти можуть вимірювати будь-які поточні та статистичні змінні: випуск продукції, контроль бракованої продукції, споживання електричної потужності, тиск, температуру, вібрацію і т. д. На практиці поширені контрольні карти середніх значень; середніх геометричних, накопичених сум; індивідуальних значень; комбінованих вимірів.

Зміни вихідної змінної у об'єкта можуть бути викликані випадковими зовнішніми і внутрішніми збуреннями; різного роду порушеннями в роботі систем (аварія) і помилковими діями персоналу.

Якщо значення контрольної карти виходить за контрольні межі, то це свідчить про необхідність перевірки та втручання у технологічний процес.

1.7. Приклад системи розпізнавання образів

Система розпізнавання образів – це класифікатор, який за допомогою відповідних методів відносить вхідні дані до певного класу об'єктів, процесів, понять на основі аналізу істотних сенсорних та понятійних ознак, що характеризують об'єкт.

Засоби, які використовуються при РО розділяються на групи:

1. *Датчики* (реальних, абстрактних, кількісних, якісних і т. п. ознак), які дозволяють виділити і «зважити» суттєві *первинні ознаки образу*.

2. *Технічні засоби інтелектуальної обробки* (переробки, порівняння, об'єднання, переоформлення, накопичення) інформації, в результаті чого отримують *вторинні ознаки образу*, у просторі яких здійснюється, власне, класифікація.

3. Математичне та програмне забезпечення у складі методів, вирішальних правил, обмежень, алгоритмів, програм, які використовуються для визначення класу образу.

4. Апаратне забезпечення: інтелектуальний банк даних (ІБД) у складі бази цілей (БЦ), бази знань (закономірності середовища, виведення нових знань, прогноз наслідків), бази даних (опис відомих фактів, кількісні дані); блок інтерпретації (забезпечує розуміння вхідної інформації); блок прийняття рішень (класифікатор).

5. Виконавчий пристрій.

Але треба визначити, що технічні засоби не мають такої універсальності, яка притаманна живій істоті: якщо технічна система РО класифікує літаки, то вона не може розпізнавати текст, або голос людини. Тобто технічні засоби працюють лише за вузьким вказаним призначенням.

Приклад системи РО літаків.

1. Складаємо *алфавіт класів об'єктів*: 1-ий клас – 4-моторні літаки; 2-ий клас – 2-моторні літаки; 3-ий клас – 1-моторні літаки.

2. Складаємо *первинний словник ознак*: швидкість і висота льоту; габарити; кількість і тип двигунів; конфігурація (конструкція); вага літака та ін. При цьому враховується можливість використання як *ап'юріорної* (до безпосереднього використання системи РО) інформації так *апостеріорної* інформації, яка отримується при безпосередньому використанні системи РО.

3. На базі *первинного словника ознак* будуємо *робочий словник ознак*, у який входять лише ті ознаки з первинного словника ознак, які реально будуть використовуватись у системі РО. Перевіряють забезпеченість *датчиками*.

3. Переходять до проектування з визначенням математичного, програмного та апаратного забезпечення.

4. «Навчають», налагоджують і випробують систему РО у дії з визначенням класу невідомого літака на основі порівняння *ап'юріорних та апостеріорних ознак*.

В дійсності етапи створення систем РО більш складні. Наприклад, якщо датчики не забезпечують отримання потрібної якості та кількості ознак, то замість 3-х класів літаків система РО буде розпізнавати лише 2 класи. Основний напрямок проектування системи РО такий: система РО уточнюється та удосконалюється у процесі іспиту з урахуванням її економічних показників.

1.8. Класифікація систем розпізнавання образів

Класифікація систем розпізнавання ґрунтується на характеристиках ознак, які використовуються у процесі розпізнавання. Можлива класифікація систем розпізнавання показана на рис. 1.8.1.

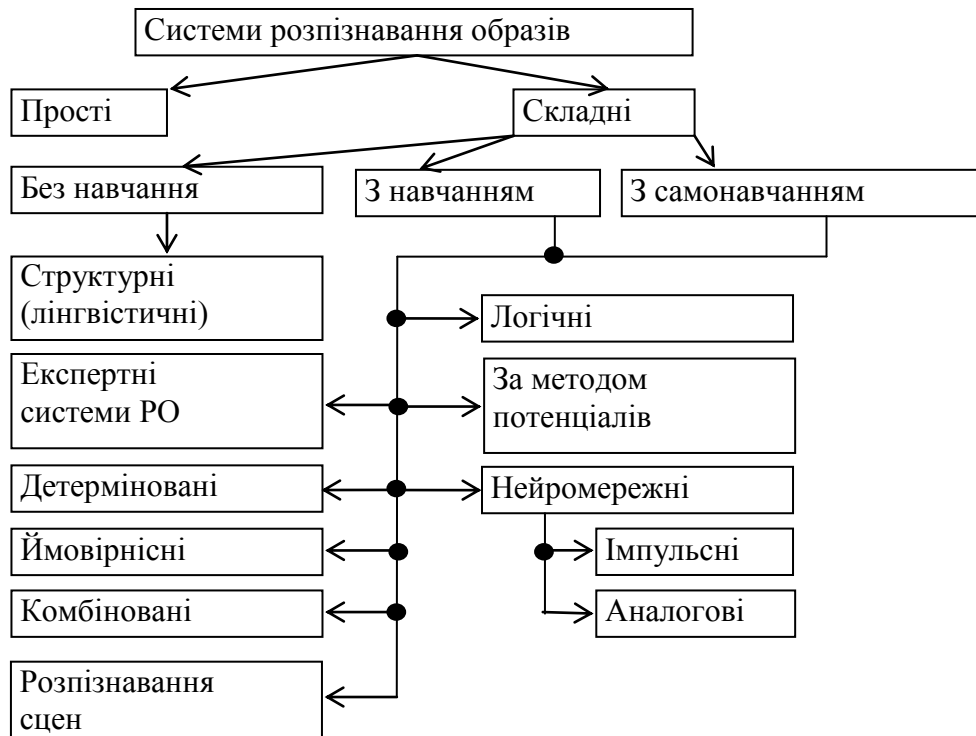


Рис. 1.8.1. Класифікація систем розпізнавання

Прості системи розпізнавання використовують лише одну ознаку з робочого словника (маса монети у автоматі в метрополітені для розміну монет; лінійні розміри для бракування деталей і т. д.).

Складні системи розпізнавання використовують кілька ознак *робочого словника* (системи медичної діагностики, в яких використовуватися дані аналізу крові, кардіограма, температура, дані кров'яного тиску і тощо). Сюди ж відносяться ієрархічні системи розпізнавання (попередні шари дають варіанти та елементи розпізнавання, вихідний шар класифікує об'єкт) та системи *попереднього групування об'єктів* по сукупностям ознак. Ці групи також можуть складатись з об'єктів різних класів: прикладом є групування товарів з малим, середнім та великим терміном зберігання; товарів по рівням попиту тощо.

Однорівневі системи РО мають один рівень датчиків, що вимірюють ознаки об'єктів і передають їх у блок алгоритмів розпізнавання. *Багаторівневі системи РО* визначається отриманням ознак першого рівня та, на додаток, отриманням переробленої інформації другого і наступного рівнів.

Системи без навчання мають достатньо апріорної інформації для розпізнавання образів.

Системи з навчанням працюють з «вчителем», який багато разів надає системі еталонні пари «вхід-вихід», що дозволяє визначити роздільні функції. Після навчання «вчитель» тестуванням перевіряє систему РО на працездатність.

Самонавчальна система використовує набір правил, за якими сама виробляє класифікацію, яка може відрізнятись від природної.

Структурні (лінгвістичні) системи: для побудови алгоритмів розпізнавання використовується речення, кожне з яких описує структуру (будову) об'єкту з непохідних («атомарних») елементів. Ці речення складають спеціальну мову з власною граматикою [147]. Класифікація об'єкту виконується шляхом порівняння речення невідомого об'єкту з еталонними реченнями класів.

Експертні системи застосовують там, де існують евристичні або інтуїтивні методи рішень і немає точних алгоритмів або розрахунків. Експертна система містить накопичені знання багатьох експертів у певній предметній області і здатна в рамках цієї області класифікувати об'єкти, давати відповіді, рекомендації, поради, запитуючи при необхідності додаткову інформацію. Експертна система найчастіше використовує правила, які мають форму: «ЯКЩО ТОДІ ІНАКШЕ». До експертних систем можна віднести медичні системи (по ознаках розпізнають хвороби), ситуаційні системи (розпізнають ситуацію і приймають рішення стосовно наступних дій [89]), прогнозуючі системи та ін.

Детерміновані системи. У цих системах для побудови алгоритмів розпізнавання використовують методи, засновані на вимірюванні відстаней між розпізнаваним об'єктом і еталонами класів.

Ймовірнісні системивикористовують ймовірнісні метод розпізнавання, засновані на теорії статистичних рішень.

Логічні системи використовують логічні методи розпізнавання, засновані на наявності логічних зв'язків, в якій змінні – логічні ознаки розпізнаваних об'єктів, а невідомі величини – класи, до яких ці об'єкти відносяться.

Системи РО з використанням метода потенціалів, за яким ознака об'єкту розглядається як його електричний потенціал, який зменшується із зростанням відстані до об'єкту.

Нейромережні системивикористовують моделі нейронів для розпізнавання образів і розділяються на імпульсні та аналогові системи: імпульсні складаються з нейронів, які отримують на входи короткі імпульси (спайків) та генерують на виходах імпульси; аналогові складаються з нейронів, які аналізують безперервний сигнал.

Комбіновані системи РО використовують комбінації методів та систем. Приклади комбінованих систем – інтелектуальний аналізатор даних DataMining [42].

1.9. Необхідність зовнішніх та внутрішніх знань

Якщо попросити лікаря (лікаря-людину ми вважаємо найкращою інтелектуальною «системою») пізнати *зовнішній об'єкт* – хворобу конкретної людини, але не надавати йому жодних даних (лікар не має знань про

зовнішнє середовище: ні зорового зображення хворого, ні аналізів, буквально – нічого), то цю інтелектуальну проблему він не розв'яже і мабуть ще й образиться на такі умови. Теж саме відбудеться, якщо **не навчати людину лікарській справі** (тоді він не має бази знань – еталонних **внутрішніх знань** про хвороби людини) і вимагати, щоб він пізнав якусь складну до визначення хворобу. Лікар назве хворобу, але звичайна людина – ні.

Таким чином, щоб прийняти рішення та виконати *адекватні оптимальні цільоспрямовані дії* при взаємодії із зовнішнім середовищем, **навіть найкраща інтелектуальна система (людська)** повинна мати внутрішні знання у вигляді еталонної моделі образу та відомості про поточний стан у вигляді його моделі.

Система штучного інтелекту (ШІ) в цьому відношенні не відрізняється від людини: вона теж повинна мати інформацію про ознаки реального образу та внутрішні знання стосовно його можливих станів. **Зовнішню інформацію** людина отримує через чуттєві органи (зору, слуху, дотику, смаку, нюху).

Ця зовнішня інформація внутрішньо переробляється з участю мозку. Про важливість процесу такої обробки інформації свідчить те, що досвідчена людина визначить хворобу людини, а недосвідчена – ні; досвідчена людина при визначенні відстані до об'єкта помилиться на 5–10 %, а недосвідчена – на 20–30 % і т. д. Фінансист по сукупності окремих ознак визнає, що фінансове становище є загрозливим або не загрозливим. При цьому фахівець та нефахівець мають однакові початкові дані.

Усьому цьому можна навчити людину. Але основна проблема полягає в тому, щоб цьому навчити машину. Системи ШІ є технічними об'єктами, які моделюють розумову діяльність людини.

Розумна поведінка системи штучного інтелекту може бути представлена у вигляді послідовності наступних дій:

1. Отримання інформації із зовнішнього середовища.
2. Виділення з навколишнього середовища, що спостерігається, зовнішньої події або об'єкта. Отримання (вимірювання) ознак події X_{20} .
3. Інтелектуальна обробка ознак X_{20} із отриманням до розгляду ознак (X_2). Порівняння з еталонами (з внутрішнім знанням). Розуміння ситуації. Прийняття рішення.
4. Адекватна цільоспрямована оптимальна дія, як реакція на зовнішню подію.

Внутрішні еталонні знання (модель), спрямовані на отримання адекватного і цільоспрямованого рішення.

Розглянемо приклади роботи інтелектуальних систем.

На людину раптово напала агресивна собака. Людина сприймає ознаки агресії у вигляді X_{20} , переробляє їх у X_2 й мусить приймати одне з можливих рішень $EX_j = \{j=1 - \text{не звертати уваги, бо собака маленька; } j=2 - \text{тікати; } j=3 - \text{захищатись}\}$. Яке рішення прийме людина? Якщо людина прий-

няла рішення захищатись, то вона виконує для цього адекватні дії: підіймає камінь, або палку; сама нападає на собаку. Але якщо людина не має часу на більш тверезе обмірковування становища (не бачить, що собака маленька і небезпеки немає, тобто ознаки X_{20} сприймаються не усі, або лише їх частка переходить у X_2), або небезпека занадто велика, або у людини слабка нервова система, то вона прийме рішення – захистити свою дорогоцінну інтелектуальну систему ганебною втечею.

Розглянемо гру у шахи. У відповідь на хід суперника й отриману позицію фігур на шаховій дошці у вигляді моделі X_{20} , система РО переробляє її на модель X_2 , яка оцінює існуючу загрозу, рухливість фігур, контрольовані поля та ін. Далі система РО починає **створювати скінчену кількість можливих проміжних еталонних рішень EX_{1J}** . Для цього ставиться машинний експеримент і перебираються можливі варіанти наступних проміжних еталонних рішень-ходів з оцінкою позицій-станів; серед створених перебраних варіантів проміжних еталонних рішень-ходів EX_{1J} обирають найліпший варіант, який має переваги перед іншими і веде до перемоги.

В системі розпізнавання літаків модель X_{20} у вигляді отриманих розмірів літака (найпростіший випадок) порівнюється з різними внутрішніми еталонними моделями EX_J , які мають відомі значення розмірів різних класів літаків. Це є класифікацією образу літака (розпізнаванням типу літака). А визначення типу літака одночасно означає й указівку на еталонні рішення та відповідні дії.

Спрощений алгоритм РО може розглядатись як ряд послідовних кроків, виконання яких оптимальним чином веде до незворотного досягнення мети. Прикладом такого алгоритму може бути автомат, який виконує алгоритм варкі рису. Згідно цьому алгоритму вказується наступні ознаки: кількість рису, кількість води та зміна температури у часі.

2. Теорія аналітичної геометрії n-вимірних тілесних кутів

2.1. Функції аналітичної геометрії n-вимірного простору

Ми аналізуємо n-вимірний простір взаємно незалежних числових змінних з охопленням часу ($x_1, x_2, \dots, x_j, \dots, x_n$) та з відповідними осями координат, на яких визначаються цифрові значення незалежних змінних. Взаємна незалежність числових змінних ($x_1, x_2, \dots, x_j, \dots, x_n$) в математиці визначається через взаємну перпендикулярність відповідних осей координат.

У випадку розгляду в n-вимірному просторі двовимірної площини чи трьохвимірного об'єму взаємна незалежність змінних ($x_1, x_2, \dots, x_j, \dots, x_n$) означає:

– *для площини в n-вимірному просторі* будь-яка з двох осей координат (x_1 =ширина, x_2 =глибина) може бути замінена віссю координат будь-якої іншої змінної з її одиницями вимірювання. Припустимо, що ми в трьохвимірній кімнаті розглядаємо лише площу стіни і міряємо її висоту x_1 , переміщуючись у часі t вздовж довжини стінки x_2 . Тоді замість системи координат (x_1, x_2) отримуємо (x_1, t) – висоту стінки у залежності від часу. Якщо площа має, наприклад, різнокольорові ділянки, інтенсивність кольору яких вимірюється іншими змінними, то можна отримати інші функціональні та геометричні залежності;

– *для трьохвимірної кімнати n-вимірного простору* будь-яка з трьох осей координат (x_1, x_2, x_3) теж може бути замінена віссю координат будь-якої іншої змінної x_4 з її одиницями вимірювання, наприклад: освітлення, температури, запилення. В результаті отримуємо систему координат (x_4, x_2, x_3), де x_4 – будь-яка змінна, включаючи час.

У практичній діяльності всі люди вільно і часто виконують подібні заміни осей в одновимірному, двовимірному та трьохвимірному просторах, не замислюючись про те, що використовують математичні особливості n-вимірного простору. Але з математичної точки зору це означає, що людина знаходиться в n-вимірному просторі і вільно використовує його особливості. У дійсності кожна точка простору трьохвимірної кімнати є n-вимірною, бо, крім трьох вимірів (x_1 =ширина, x_2 =глибина, x_3 = висота), **точка трьохвимірної кімнати характеризується багатьма іншими незалежними змінними**: температурою (x_4), часом (x_5), освітленням (x_6), запахом (x_7), пилом (x_8), мікробами (x_9), рухом повітря (x_{10}) тощо. Окреме використання будь-яких двох чи трьох змінних з наведеного переліку дозволяє отримати геометрію древніх греків.

Для аналізу n -вимірного простору ми будемо використовувати відомі з аналітичної геометрії поняття та формули гіперкулі, гіперсфери та їх радіусів, тілесних гіперкутів (**Т-кутів**), гіперплощин, гіперпрямих ліній та їх відрізків, гіперфігур [27, с. 281–310].

Гіперкуля n -вимірного простору – це геометричне *тіло*, всі точки якого знаходяться від центру на відстані, не більшої за вказаний **гіперрадіус** R (рис. 2.1.1) [27, с. 281–310]

$$R = \sqrt{\sum_{j=1}^n x_j^2}, \quad (2.1.1)$$

де x_j – проекція гіперрадіусу R на відповідну ось координат (дорівнює числовому значенню точки осі координат, у яку попадає кінець вектора R). В n -вимірному просторі *вектор, радіус вектора та модуль вектора* нами помічаються однаковою літерою R .

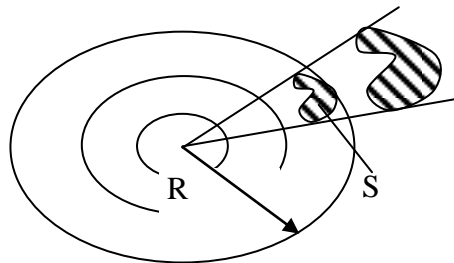


Рис. 2.1.1. Гіперкуля із зовнішнім радіусом R з вкладеними одна в одну порожнистими гіперкулями та тілесним гіперкутом (Т-кутом), що спирається на гіперплощу S гіперсфери радіусом R

Гіперкулю можна розділити на кілька вкладених одна в одну порожнистих гіперкуль заданої товщини з позначенням більшого радіусу.

Гіперсфера – це гіперповерхня гіперкулі радіусом R у n -вимірному просторі, створена рівновіддаленими точками від її центру.

Довжину кола C_{2n} , площу круга S_{2n} площу сфери S_{3n} і об'єм кулі V_{3n} , при розгляді їх геометричних форм в n -вимірному просторі, розраховуємо по формулам, аналогічним відповідним формулам двовимірного та трьохвимірного простору

$$C_{2n} = 2\pi R; S_{2n} = \pi R^2; S_{3n} = 4\pi R^2; V_{3n} = \frac{4}{3} \pi R^3. \quad (2.1.2)$$

Подібні залежності не означають, що n -вимірний простір можна розглядати як двохвимірний чи трьохвимірний. Це означає, що в просторі вищого виміру *можна виділити знайомі геометричні фігури простору нижчого виміру* (точки, прямі лінії, кола, площі тощо), які допомагають створювати і виділяти відомі геометричні фігури у просторі вищих вимірювань при обмеженні деяких значень змінних.

Тілесним кутом (Т-кутом) n -мірної гіперкулі зветься частина її простору, яка об'єднує всі вектори об'єктів, промені яких (напрямки векторів) виходять з центру гіперкулі і перетинають деяку поверхню з площею S на

гіперсфері. Тілесний кут з вершиною у центрі гіперкулі вимірюється в *стерадіанах* і дорівнює $\Omega = S/R^2$ – тобто дорівнює відношенню площі S на гіперсфері гіперкулі, на яку спирається тілесний кут, до квадрата радіусу гіперсфери R (рис. 2.1.1).

Кількість однакових T -кутів, виділених осями координат $(x_1, x_2, \dots, x_j, \dots, x_n)$ у n -вимірному просторі дорівнює:

– « n » при позитивних осях координат з порядковими значеннями номерів T -кутів в десятинній системі числення $g = 0, 1, 2, \dots, (n-1)$ – якщо нумерація починається з нуля;

– « 2^n » при позитивних та негативних осях координат з порядковими значеннями номерів T -кутів в десятинній системі числення $g = 0, 1, 2, \dots, (2^n - 1)$ – якщо нумерація починається з нуля.

Далі ми розглядаємо лише випадок 2^n , коли кожна ось x_j вміщує позитивні і негативні числові значення. Вважаємо всі T -кути взаємно незалежними з можливістю довільної зміни їх положення у просторі та відносно інших T -кутів «при виконанні ними всіх без винятків обов’язків» системи з ортогональними осями координат.

Вважаємо, що всі T -кути, які виділяються осями координат $(x_1, x_2, \dots, x_j, \dots, x_n)$ із T -кулі, є рівними, і вони розділяють поверхню гіперсфери та тіло гіперкулі на відповідні рівні частки, які у загальному випадку можуть взаємно перекриватись. Знаючи кількість T -кутів, можна визначити положення серединних ліній T -кутів і положення точки їх перетину з гіперсферою, вважаючи, що всі осі координат «взаємно перпендикулярні» і «знаходяться на однаковій відстані від серединної лінії кожного T -кута». Можна також розрахувати числові значення відповідних «часток поверхні гіперсфери та об’єму тіла гіперкулі, які приходяться на один T -кут». Але визначені таким чином геометричні образи можуть геометрично наочно відображуватись лише для $n \leq 2$. При $n \geq 3$ для отримання візуального зображення необхідно зменшити масштаб площі S , на яку спирається T -кут, і розглядати кожний T -кут окремо.

Деякі з поширених формул аналітичної геометрії, які можуть бути використані при аналізі n -вимірного простору, наведені нижче [27, с. 281–310]:

– **пряма лінія** описується параметричними рівнянням $x_j = a_j + b_j t$, де $j = 1, 2, \dots, n$ – порядковий номер змінної; t – параметр зі значеннями $-\infty \leq t \leq +\infty$. Якщо дві прямі лінії перетинаються, то при деяких значеннях t^* та t^{**} справедливі рівності $a_i^* + b_i^* = a_i^{**} + b_i^{**}$;

– канонічне рівняння прямої лінії $\frac{x_1 - a_1}{b_1} = \frac{x_2 - a_2}{b_2} = \dots = \frac{x_n - a_n}{b_n}$. Для площини використовують рівняння $\frac{x_1}{b_1} + \dots + \frac{x_j}{b_j} + \dots + \frac{x_n}{b_n} = 1$.

– **відстань між двома n -вимірними точками** $M(x_1, x_2, \dots, x_n)$ та $M^*(x_1^*, x_2^*, \dots, x_n^*)$

$$MM^* = \sqrt{\sum_{j=1}^n (x_j - x_j^*)^2};$$

– **вектор між точками** $M(x_1, x_2, \dots, x_n)$ та $M^*(x_1^*, x_2^*, \dots, x_n^*)$ має координати $(x_1 - x_1^*; x_2 - x_2^*; \dots; x_n - x_n^*)$. Якщо M є початком координат, то вектор $\overrightarrow{MM^*}$ зветься радіусом-вектором;

– **кут φ між двома прямими лініями** $x_i = a_i + b_i t$ та $x_i^* = a_i^* + b_i^* t$ вимірюється у межах $0 \leq \varphi \leq \pi/2$ і дорівнює

$$\cos \varphi = \frac{|\sum_{j=1}^n b_j b_j^*|}{\sqrt{\sum_{j=1}^n (b_j)^2} \cdot \sqrt{\sum_{j=1}^n (b_j^*)^2}} \quad (\text{для паралельних прямих } \cos \varphi = 1, \text{ тобто } \varphi = 0);$$

– **рівняння гіперплощини** $\sum_{j=0}^n \left(\frac{x_j}{a_j}\right) = 1, j = 0, 1, 2, \dots, n$, де $x_0 = 1$.

Деякі особливості n-вимірного простору:

1. Гіперпряма лінія $\frac{x_1}{b_1} = \dots = \frac{x_j}{b_j} = \dots = \frac{x_n}{b_n}$ має властивість **одновимірності**,

бо з гіперпрямої лінії можна отримати одновимірну залежність $\frac{x_j}{b_j} = \left(\frac{x_1 b_j}{b_1}\right) \frac{1}{b_j}$.

Тоді можна вважати, що гіперплощина $\frac{x_1}{a_1} + \dots + \frac{x_j}{a_j} + \dots + \frac{x_n}{a_n} = 1$ визначається перетином двох гіперпрямих ліній і тому має **властивість двовимірності**, а гіпероб'єм визначається перетином трьох гіперпрямих ліній і тому має **властивість трьохвимірності**.

2. Розглянемо радіус R за формулою (2.1.1) в трьохвимірній системі координат (x_1, x_2, x_3) . Якщо в ній замінити дві осі (x_2, x_3) однією віссю $y_{2,3} = \sqrt{x_2^2 + x_3^2}$, то отримаємо **ортогональну** еквівалентну двохвимірну систему координат $(x_1, y_{2,3})$, до якої можна застосувати формулу отримання гіпотенузи $y_{1,2,3} = \sqrt{x_1^2 + y_{2,3}^2} = R$, а взагалі в площині $(x_1, y_{2,3})$ можна отримати всі геометричні образи двохвимірного простору. В результаті замість трьохвимірної системи координат **ми отримали цілком еквівалентну одновимірну систему** з використанням модуля – радіуса-вектора R , який додатково характеризується кутом нахилу до відповідних осей. По кутах нахилу, або з формули для прямокутного трикутника можна відновити проєкції на осях, наприклад $x_1 = \sqrt{R^2 - (x_2^2 + x_3^2)}$.

Якщо розповсюдити подібні розмірковування на n-вимірний простір, то виходить, що **будь-яку n-вимірну систему можна розглядати в скороченій системі координат**.

3. Поверхню гіперсфери, на якій розміщується кінець вектора R , може розглядатись як **своєрідну** «мапу Землі». Тому для n-вимірної Т-кулі при відомих координатах (x_1, x_2, \dots, x_n) вектора R можна визначити

на гіперсфері функцію мети $Q(R)$ у вигляді ізоліній; визначити за методами грануляції даних ділянки дозволеного, нормального та аварійного станів для положення вектора R ; вказати в T -кулі годограф вектора R і положення на цьому годографі його поточної точки.

При урахуванні грануляції змінних [139] і використанні «чорного ящика» для опису стану об'єкта, можна:

– виділити за допомогою експерта небезпечні ділянки T -кулі з наочною демонстрацією наближення поточного стану об'єкта у вигляді кінця його вектора до небезпечних чи небажаних станів всередині T -кулі;

– відобразити сам годограф об'єкта всередині T -кулі у вигляді ділянок з нормальними чи загрозливими станами.

В результаті «чорний ящик» об'єкта можна перетворити для користувачів на «прозорий» «скляний» ящик, що є корисним у різних напрямках дослідження.

2.2. Візуальна аналітична геометрія об'єктів n -вимірному простору

Твердження про «нездатність уявлення людиною геометричних об'єктів вищих вимірів» є застарілим, тому що наразі аналітична геометрія їх візуалізує.

Багатовимірність і ускладнення систем прийняття рішень при аналізі n -вимірних функцій мети $Q(R)$ у залежності від вхідного вектора $X = (x_1, x_2, \dots, x_j, \dots, x_n)$ з цифровими змінними x_j вимагає ширшого застосування візуальної інформації, яка ліпше сприймається людиною у порівнянні з текстом.

Протягом двох минулих тисячоліть будь-який математик, який би наважувався розглянути четвертий вимір, ризикував би репутацією. Дискусія про наявність четвертого просторового виміру є *найдовшою в історії людства*, бо четвертий вимір на перший погляд суперечить здоровому глузду.

Аристотель (384–322 до н. е.) в трактаті «Про небо» писав: «Величина, яка ділиться в одному напрямку, є лінією, в двох – площина, в трьох – тіло, і, крім них, немає іншої величини, тому що три виміри є суттю всіх вимірів».

В 1685 р. математик Джон Уолліс назвав четвертий вимір «чудовиськом у природі, можливим не більше, ніж химера або кентавр».

В 1746 р. И. Кант висловив думку про багатовимірний простір.

В 1764 р. Ж. Д'Аламбер писав про чотирьохвимірність простору доданням до нього часу в якості 4-ої координати.

А. Келі (1843), Г. Грасман (1844) та Л. Шлефлі (1852) запропонували n -вимірну геометрію.

«Король математики» Карл Гаус (1777–1855 рр.) теж проводив експерименти по виявленню 4-го виміру. На одному з експериментів він розмістив трьох людей на вершинах трьох пагорбів. Кожна людина мала ліхтар, а сумісно вони створювали у просторі величезний трикутник. Сам Гаус ретельно виміряв кути цього трикутника і підтвердив, що їх сума дійсно дорівнює 180° . З цього вчений зробив висновок, що якщо відступ від грецької геометрії існує, то він такий малий, що його неможливо виявити.

В 1854 р. *основи математики вищих криволінійних вимірів* заклав Ріман Г. Б., учень Гауса. Відкриття Рімана привело у захоплення всю Європу, а четвертий вимір викликав сенсацію серед інтелігенції – письменників, філософів, художників. Але фізики розглядали 4-й вимір лише як забавку. Жодних свідоцтв існування 4-го виміру не було.

В 1908 р. *Германом Мінковським був запропонований чотиривимірний псевдоевклідовий простір Мінковського* як геометрична інтерпретація простору-часу спеціальної *теорії відносності*: три координати являють собою Декартові координати тривимірного евклідового простору, а четверта – координату ct , де c – швидкість світла, t – час події. Зв'язок між просторовими відстанями та проміжками часу, що розділяють події, характеризується квадратом інтервалу $s^2 = c^2(t_1 - t_0)^2 - (x_1 - x_0)^2 - (y_1 - y_0)^2 - (z_1 - z_0)^2$.

Сучасна візуалізація образів вищих вимірів має кілька напрямків:

1. *Заперечення можливості візуалізації методами аналітичної геометрії* (найбільш поширений напрямок). Його можна демонструвати цитатою: «Точковий n -вимірний простір називають арифметичним простором. Із запровадженими поняттями ми не пов'язуємо ніяких наочних уявлень, тому що своїми органами чуття людина нездатна сприймати більше ніж три взаємно перпендикулярні напрямки» [27, с. 282].

2. *Замовчування*: у підручниках, довідниках по аналітичній геометрії сучасні теоретичні основи візуалізації образів вищих вимірів не аналізуються і не коментуються.

3. *Заміна складної функції $Q(x_1, x_2, \dots, x_j, \dots, x_n)$ на спрощену*:

– у вигляді залежності виходу від однієї «головної» вхідної змінної $q_{C1}(x_j)$, або залежності від двох «головних» вхідних змінних $q_{C2}(x_j, x_i)$, $j \neq i$. Сучасні комп'ютерні технології дозволяють автоматизовано виконувати подібні перерахунки, що зменшує їх вартість. Прикладом може бути підтримка прийняття рішень через аналіз даних (Online Analytical Processing (OLAP));

– у вигляді розміщення у двовимірному просторі по координатах $X_2=(x_1, x_2)$ або у трьохвимірному просторі по координатах $X_3=(x_1, x_2, x_3)$ складних багатовимірних об'єктів з ознаками $q_y(y_1, y_2, \dots, y_i, \dots, y_m)$ і з графічним відображенням лише деяких з цих ознак y_i за допомогою форми, розмірів, кольору, символічних позначень.

4. *«Маскування»* під різні напрямки досліджень. Можна вважати, що складні технічні розрахунки вихідних даних у вигляді, наприклад, ваги чи

розмірів [парових, дизельних, електричних, газових двигунів; мостів; будинків; суден; ракет], або логічні висновки системи прийняття рішень по суті визначають положення змінної виходу на числовій осі у залежності від багатьох вхідних змінних проектування і тому відносяться до методів візуалізації результатів досліджень в багатовимірній аналітичній геометрії. Сюди ж відносяться й результати аналізу виходів будь-яких складних методів групування в n-вимірному просторі змінних: метода статистичного аналізу, метода дерев, метода ієрархічних потоків, методу аналізу експериментів, тощо.

5. Методи точної геометричної візуалізації багатовимірних образів мають одну особливість: вони здатні відобразити одну точку в n-вимірному просторі змінних, що означає спроможність візуалізувати будь-який геометричний об'єкт. Наразі ці методи складаються з трьох напрямків:

5.1. Аналіз в Паралельних Координатах, розглянутий Альфредом Інсельбергом в 1981–1990 рр. і спрямований на візуалізацію традиційних фігур аналітичної геометрії – точки, прямої лінії, площини та ін.

Візуалізація точки в багатовимірному просторі показана на рис. 2.2.1, на якій в евклідовій площині на взаємно перпендикулярних осях (y,x) відображується:

- на осі y вказується масштаб для всіх осей координат (x_1, x_2, x_3, x_4, x_5);
- вздовж осі x на рівних відстанях паралельно розміщені лінії осей координат (x_1, x_2, x_3, x_4, x_5). На кожній з цих ліній відмічають відповідне цифрове значення координати точки $C=(C_1, C_2, C_3, C_4, C_5)$. Помічені точки сусідніх координат з'єднують штриховими лініями.

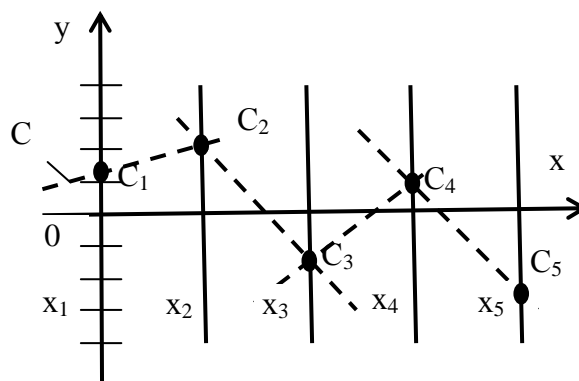


Рис. 2.2.1. Візуалізація в Паралельних Координатах $X=(x_1, x_2, x_3, x_4, x_5)$ положення точки $C=(C_1, C_2, C_3, C_4, C_5)$ за допомогою ламаної штрихової лінії C

В результаті ламана штрихова лінія $C = (C_1 - C_2 - C_3 - C_4 - C_5)$ візуалізує положення точки $C=(C_1, C_2, C_3, C_4, C_5)$ в паралельних координатах $X=(x_1, x_2, x_3, x_4, x_5)$. Методологія теорії А. Інсельберга була застосована для візуалізації геометричних образів, а також процесів в середовищі «DataMining», в медицині, фінансах, торгівлі, в алгоритмі уникнення зіткнення літаків, в статистиці та ін.

5.2. *Геометрична теорія багатовидів* (багатовиди розглядаються як геометричні моделі багатовимірних геометричних фігур, кривих ліній і поверхонь), розроблена в Україні О. М. Гумен (2011 р.) Вона дозволяє конструювати раціональні візуалізовані n -вимірні багатовиди для моделювання багатопараметричних процесів, об'єктів та систем; дослідити властивості тензорів нелінійного відображення n -вимірного простору; виконати аналіз фазових траєкторій багатопараметричних систем з їх комп'ютерною візуалізацією та ін. В роботі Лясковської С. Є. (2010), виконаній під керівництвом О. М. Гумен, геометрична модель розглядається стосовно розміщених у площині осей координат, наприклад, у вигляді осей координат $(xyzt)$ рис. 2.2.2, а, кожна з яких має і позитивні і негативні числові значення. Припустимо, що площина $x - y - z - t = A$ у чотирьохвимірному просторі $(Oxyzt)$ пересікає осі координат у точках $x = A, y = -A, z = -A, t = -A$ (рис. 2.2.2, б). Положення цієї площини на рис. 2.2.2, б показано штриховими лініями. На рис. 2.2.2, б також показано положення точки В при заданих координатах (x_1, y_1, z_1, t_1) . Математична модель тривимірної площини у просторі $(xyzt)$ при фіксованому значенні t набуває вигляду

$$x - y - z = (A+t).$$

Проекція гіперплощини чотиривимірного простору $(Oxyzt)$ при $t=0$ наведена на рис. 2.2.2, в.

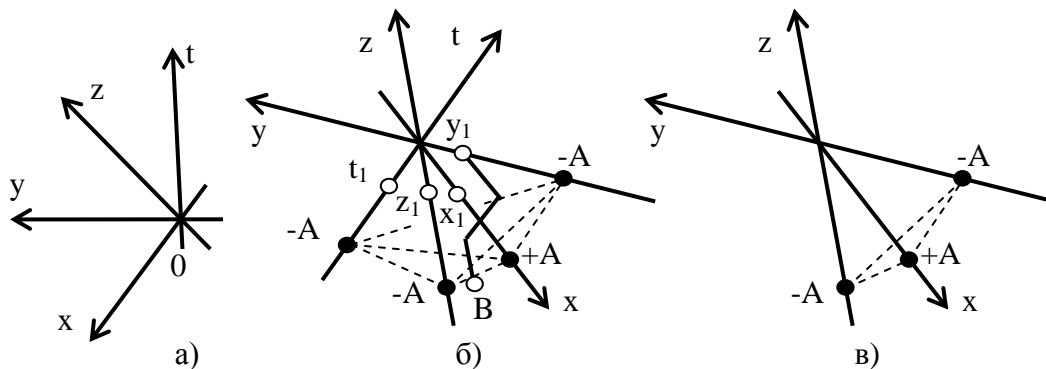


Рис. 2.2.2. Узагальнена чотиривимірна декартова система координат $Oxyzt$: а) розміщення осей координат; б) гіперплощина у чотиривимірному просторі $Oxyzt$; в) проекція гіперплощини у чотиривимірному просторі $Oxyzt$ при $t = 0$

С. Є. Лясковська (під керівництвом О. М. Гумен) на основі аналізу моделей фазового n -простору розглянула візуалізацію геометричних образів багатовидів з метою аналізу перехідних процесів у багатопараметричних системах, одержала наочні зображення $(n-1)$ – циліндрів як гіперповерхонь n -вимірних просторів, геометричну інтерпретацію частинної паралельності багатовимірних лінійних підпросторів, розробила алгоритм побудови гіперповерхонь фазових траєкторій при можливості зміни їх параметрів та ін.

5.2. *Геометрична геометрія n -вимірних тілесних кутів* [54], яка розглядається нижче.

2.3. Аналітичні та геометричні особливості n-вимірних тілесних кутів

2.3.1. Контекстні метрики та числові значення змінних по осях координат

Контекстні метрики аналітичної геометрії арифметичного точкового простору

В аналітичній геометрії n-го вимірного простору $(x_1, x_2, \dots, x_j, \dots, x_n)$, $j = 1, 2, \dots, n$, кожна змінна має власну ось координат з показаними на них *цифрами вимірювання змінних*. Метрикою осі (значення одиниці вимірювання осі x_j , $j = 1, 2, \dots, n$) є *контекстна величина* у вигляді довільно заданої дослідником реальної чи відносної величини (рис. 2.3.1).

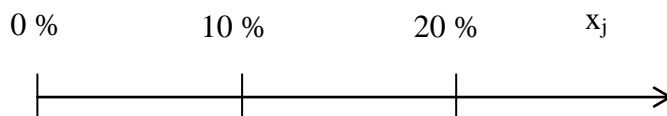


Рис. 2.3.1. Контекстне визначення дослідником для j-ої осі числових значень змінної x_j (діапазону і метрики у вигляді відсотків)

*Якщо взяти ось 1-вимірного простору x_1 з вказаними на осі цифрами вимірювання змінної x_1 і спитати, з якою метрикою і яка величина на цій осі вимірюється, то виявиться, що числові значення змінної і значення метрики **контекстно** визначаються дослідником: вони можуть відноситись і до виміру росту рослини і до виміру концентрації речовини в розчині – взагалі до всього що завгодно. Тобто і діапазон числових значень змінної і метрика довільно і контекстно (лінгвістично) пояснюються дослідником.*

*Якщо стосовно осей координат (x_1, x_2, x_3) трьохвимірного простору стародавніх греків з числами для вимірювання змінних спитати, де тут довжина, ширина та висота, то теж приходимо до висновку, що арифметичні **цифри є контекстними** (ім потрібно дати назву, вказати їх призначення, діапазони та метрику). Будь-яку з цих осей можна замінити, наприклад, на температуру, і старогрецький трьохвимірний геометричний аналіз від цього не постраждає.*

І древні греки і навіть великий математик Гаус стверджували про відсутність вимірів вищих за три, посилаючись на виміри трьохвимірної кімнати. Це твердження збереглося до наших часів, хоча й похитнете. Адже у дійсності будь-яка кімната має n-вимірний простір. Користуючись взаємною перпендикулярністю осей координат n-вимірного простору та можливістю тлумачення контексту, можна висоту кімнати замінити на координату, наприклад запаху, щоб у повній мірі реально відчувати і побачити 4-й фізичний геометричний вимір у кімнаті. А по черзі геометрично (наочно) можна відобразити всі n змінні, що описують простір кімнати.

Ми будемо вважати, що довжина x_1 , ширина x_2 та висота x_3 – це геометрично **різні фізичні поняття**, які можуть мати **різні метрики розміру**: x_1 – в см, x_2 – в км, x_3 – в мкм. Розрахований радіус кулі в трьохвимірному просторі і сама куля у цьому випадку мають контекстне тлумачення та контекстну метрику, рівно як і у випадку, коли будь-яка ось координат з метрикою довжини замінюється на змінну x_4 , що вимірює температуру. Виходить, що **розрахований радіус кулі трьохвимірного простору не завжди вимірюється за допомогою метрики розміру, а його числове значення має контекстне пояснення.**

Точка з однаковими числовими значеннями координат має **однакове геометричне положення** в системах координат з однаковими числовими осями незалежно від метрик осей. Тому геометрична фігура **в будь-якій з таких систем координат буде мати однаковий геометричний вигляд** незалежно від метрик відповідних осей.

Виникає питання: чи можна по **однаковим** числовим значенням координат точки (x_1, x_2) , але **при різних** контекстних метриках осей, отримати **однакове** числове і візуальне значення гіпотенузи $g = \sqrt{x_1^2 + x_2^2}$? Відповідь на питання: «так, можна, але отримані однакові числові значення гіпотенуз та їх візуальні зображення мають різні контекстні тлумачення», наприклад: «гіпотенуза отримана для метрики по осі x_1 ... (далі – пояснення метрики x_1) та метрики по осі x_2 ... (далі пояснення метрики x_2)».

Виходить, що в n -вимірному просторі будь-який відрізок чи фігура, визначені за логічним виразом чи за формулою, повинні в обов'язковому порядку супроводжуватись **контекстним поясненням** стосовно не лише метрик всіх осей але й стосовно отриманого відрізка чи фігури. Наприклад, контекстно можна вважати, що всі осі n -вимірного простору вимірюються в сантиметрах – від цього геометричний візуальний результат аналізу не змінюється при довільних реальних вимірах з використанням будь-яких інших фізичних метрик. Контекстність підтверджується й практикою: в трьохвимірному просторі ми можемо вдарити ногою футбольний м'яч і не можемо вдарити, якщо цей «м'яч» отриманий в координатах температури, тиску та освітлення.

В подальшому для спрощення аналізу, ми будемо використовувати в аналітичній геометрії для цифр всіх осей n -вимірного простору єдиний контекст – вважати, що **всі змінні виміряні в однакових відносних одиницях** (відповідно, для рис. 2.3.1 в аналітичній геометрії ураховуються лише цифри, без відсотків). Це має перевагу: при поверненні до реальних застосованих метрик осей без зміни чисел на осях координат, числовий і візуальний результат отриманого геометричного аналізу не змінюється.

Визначений таким чином арифметичний простір з контекстом метрик вимірів по всіх осях координат у «однакових відносних одиницях» не відрізняється від **відомого арифметичного точкового простору**, для якого у фаховій літературі спеціально оговорено, що з цим простором **«ми не**

пов'язуємо ніяких наочних уявлень, тому що своїми органами чуття людина нездатна сприймати більше аніж три взаємно перпендикулярні напрямки» [27].

Древні греки стосовно трьохвимірного простору (x_1, x_2, x_3) скористались своїм правом контекстного присвоєння всім осям *однакової метрики розміру* (хоча значенням x_1, x_2, x_3 можна надавати й різні метрики розміру – м, см, км).

На основі аналізу древніх греків, подальший пошук вищих вимірів продовжувався з *помилковою вимогою обов'язкового використання метрики наступного виміру в сантиметрах* (без урахування того, що: всі цифри в аналітичній геометрії – контекстні; осі координат та радіус шару в трьохвимірному просторі можуть не вимірюватись обов'язково в сантиметрах; всі вищі виміри знаходяться тут же, прямо перед ними: достатньо, наприклад, висоту x_3 в сантиметрах замінити на температуру x_4 в градусах вздовж цієї висоти x_3 , що замінюється).

Образно кажучи, аналітична геометрія за рахунок використання лише чисел по осях координат без їх метрик «без зусиль і легко» виконує заміну «сантиметрів» по координатній осі або на «градуси Цельсія», або на «швидкість руху», або на «тиск в трубопроводі» тощо, **що є недозволеним і неможливим з точки зору науковців**. Саме тому пошук метрик гіперрадіусу чи відрізка в аналітичній геометрії заходить у безвихідь. Переведення числових значень осей у *відносні одиниці* нічого не дає, бо проблема об'єднання й пояснення метрик вимірів залишається.

В аналітичній геометрії метрики не пов'язані між собою, як до цього звикли науковці в системі СІ. Навпаки – **метрики осей координат є окремими, рівноправними, незалежними** і теоретично кожна метрика може використовуватись як для вимірювання власної змінної, так і для вимірювання будь-якої іншої змінної. Цифри в одній і тій же системі координат з числовим виміром осей можна контекстно тлумачити довільно. А метрика гіперрадіусу може контекстно розглядатись і вимірюватись за метрикою будь-якої однієї осі координат даного n -вимірного простору, або за заданою сукупністю метрик різних осей координат цього ж n -вимірного простору, розглядаючи сукупність як «гіперрадіус» або «гіпергіпотенузу».

Радіус шару в трьохвимірному просторі не обов'язково має вимір в сантиметрах. Образно кажучи, «м'яч» в трьохвимірному геометричному просторі не завжди можна «пхнути ногою», бо він може знаходитись цілком або частково «в іншому просторі».

В *інтелектуальних, фізичних, та технічних* системах ми розглядаємо деяку складну функціональну залежність $Q(R)$, де $R = (x_1, x_2, \dots, x_j, \dots, x_n)$ – радіус гіперкулі, в межах якої проводиться дослідження. Тому всі процеси будь-якої інтелектуальної системи вміщуються всередині деякої гіперкулі радіусом R .

Метрика вимірювання радіусу залежить від метрик осей координат, але *контекстно ми маємо право* вважати метрики всіх осей вимірними в сантиметрах, у відносних одиницях чи інших метриках, бо від цього не змінюються ні геометричний образ ні результати геометричного і математичного аналізів.

При цьому вважаємо, що кожний з отриманих геометричних об'єктів (*навіть при рівній кількості змінних і незмінності їх метрик*) може мати незлічену кількість «*функціональних контекстних тлумачень*» із загальним аналітичним та геометричним оформленням.

Нижче наводяться деякі початкові дані, які дозволяють стверджувати, що людина здатна **наочно уявляти геометричні фігури n -вимірного простору** [54]. При цьому «технологія розпізнавання» геометричних n -вимірних фігур по наочності нічим не відрізняються від фігур в трьох-вимірному просторі в системі координат (x_1, x_2, x_3) . Це спростовує наведене вище твердження про те, що «людина нездатна сприймати більше ніж три взаємно перпендикулярні напрямки» [27].

Особливості сприйняття n -вимірного арифметичного простору.

Геометричний аналіз в n -вимірних тілесних кутах виконується на основі наступних даних:

1. Людина живе *не в 3-вимірному, а в n -вимірному просторі*. Про це свідчить кількість природних датчиків – чуттєвих органів людини (слух, зір, нюх, смак, дотик) та розум, який дозволяє вимірювати час та багато інших додаткових змінних. Людина якщо не в теорії, то на практиці чудово знає про незалежність змінних (математично незалежність відображується взаємною перпендикулярністю осей координат) та багатовимірність навколишнього середовища: знаходячись в *в 3-вимірній* кімнаті $(x_1$ – ширина, x_2 – глибина, x_3 – висота) людина легко замінює, наприклад, ось x_3 на ось « x_4 – температуру/освітлення/наявність запаху/радіоактивність/пил тощо» і *використовує це на практиці*. При цьому в новій системі координат (x_1, x_2, x_4) можна отримати всі геометричні фігури древніх греків. В теоретичному 3-вимірному просторі (x_1, x_2, x_3) подібна заміна неможлива, бо в ньому змінна x_4 просто відсутня.

Природні датчики людини, практика його дій («критерії істини») свідчать про те, що людина сприймає, розуміє і використовує на практиці n -вимірний простір, хоча іноді й інтуїтивно та без твердих математичних знань, *але зате – поголовно*. Але разом з тим із цього не завжди робиться цілком природний висновок, що людина, яка на практиці використовує змінні $x_j, j = 1, 2, \dots, n$, *сама живе в n -вимірному просторі змінних*.

Багатовимірний простір вищого порядку вміщує в собі всі геометричні фігури (а внаслідок цього – формули та властивості) простору нижчого порядку при введених відповідних обмеженнях координат. Саме за вимогою практичних потреб отримано *багатовимірну евклідову геометрію* шляхом заміни аксіоми Евкліда, за якою простір має три виміри, на аксіому, за якою простір має n вимірів.

2. *Будь-який вищий n -вимірний простір має здатність використовувати геометричні візуальні фігури нижчих вимірів.* Якщо прийняти, що деякі змінні замінені константами, або пов'язані формулами чи нерівностями, то ми можемо в n -вимірному просторі отримати знайомі елементи нижчих вимірів: відрізок прямої лінії, площину, трикутник та ін., які дозволяють нам розглядати обмеження об'єктів вищих вимірів у більш нам зрозумілій наочній формі.

3. *В n -вимірному геометричному просторі використовуються метрики, які є децю незвичними для інженерів та наукових працівників.* Інженери і науковці звикли, наприклад, до системи СІ, за якою будь-які метрики жорстко взаємно пов'язані між собою. З точки зору науковця радіус гіперкулі *має довжину* і тому виникає питання, в яких одиницях вимірюється довжина радіусу гіперкулі, якщо відповідні осі координат мають взаємно несумісні числові та якісні метрики?

Тут слід підкреслити, що в аналітичній геометрії подібні проблеми виникають при будь-якій кількості вимірів, починаючи з $n=1$. Наприклад, при $n=1$ «відрізок прямої» на осі x_1 , в просторі (x_1) , може *контекстно* відображувати: діапазон існування (чи можливості зростання) деякої рослини у вказаних границях зміни температури/воло-гості/наявності добрив; діапазон опорів резисторів; діапазон температур взимку тощо. При збільшенні кількості змінних зростає число і без того необмежених по кількості варіантів *контекстних тлумачень*. Тобто «радіус» *не обов'язково має метрику довжини, а «відрізок прямої лінії» не обов'язково має метрику температури.*

Будь-які відрізки прямих, функціональні залежності, фігури та формули n -вимірного простору *мають контекстний характер* і у загальному випадку можуть мати несумісні, контекстні, різні метрики вимірювання в одній і тій же системі координат при однакових цифрових вимірюваннях. Формули, отримані в n -вимірному просторі є контекстними тому, що вони вірні при будь-яких метриках осей.

Їх об'єднує лише арифметичні, числові метрики осей. *Лише арифметичні метрики з контекстним тлумаченням й використовуються аналітичною геометрією в n -вимірному просторі.* Якщо контекстні метрики мають однакоє арифметичне відображення на числових осях, то й фігури з однаковими відповідними координатами будуть мати однаковий вигляд.

Для пояснення цих тверджень розглянемо радіус гіперкулі R , для визначення якого використані несумісні якісні та числові ознаки осей координат. Цей радіус R має числові проєкції на кожен вісь (в тому числі й на якісну вісь, для якої звичайно використовується числова метрика), і ці проєкції радіусу R виокремлюють на кожній осі x_j відповідні відрізки « $a_j b_j$ », в рамках яких змінюється змінна x_j . *Відрізок « $a_j b_j$ » та радіус гіперкулі R взаємно пов'язані:* можна вважати, що *по черзі* радіус R може вимірюва-

тись в окремих метриках всіх осей x_j , гарантуючи заодно дотримання принципу рівноправності та незалежності всіх змінних і числових значень їх осей.

Тому в геометрії n -вимірному арифметичного простору для всіх без винятку осей цифри осей *не потрібно переводити у відносні одиниці*: достатньо лише їх фізичним метрикам надати контекст вимірювання у відносних одиницях. Отриманий візуальний геометричний чи математичний результат досліджень не буде відрізнятися від даних аналізу простору з довільними фізичними метриками і лише може вимагати контекстного пояснення.

2.3.2. Порядкові номери тілесних кутів та їх позначення

Якщо ми маємо деякий *довільний вектор з координатами* $R_1 = (x_1 = -10, x_2 = -7, x_3 = +3, x_4 = +6)$, то порядковий двоїстий номер його T -кута визначаємо шляхом надання кодів «1» та «0» відповідним позитивним та негативним осям [54]. В результаті для вектора $R_1 = (-10, -7, 3, 6)$, отримуємо відповідний *двоїстий порядковий номер T -кута « $d = 0011$ »*.

Потім отриманий двоїстий порядковий номер T -кута « $d = 0011$ » *переводимо у десятинний порядковий номер « $g = 3$ »*. У десятинному численні порядкові номери мають вигляд $g = 0, 1, 2, \dots, G$, де $G = (2^n - 1)$.

Позначення T_g^n -кута означає: n – загальна кількість змінних; g – десятинний порядковий номер T -кута. Таким чином, вектор R_1 знаходиться в тілесному куті $T_{g=3}^{n=4} = T_3^4$.

Іншу порядкову двоїсту та десятинну нумерацію отримуємо, якщо представимо вектор із зворотним розміщенням змінних ($R_1 = (x_4 = +6, x_3 = +3, x_2 = -7, x_1 = -10)$; $d = 1100$; $g = 12$; T_{12}^4), але порядок нумерації тілесних кутів особливого значення не має.

Тілесні кути для двох змінних (x_1, x_2) зветься *квадрантами* (рис. 2.3.2, а), для трьох змінних (x_1, x_2, x_3) – *октантами* (рис. 2.3.2, б).

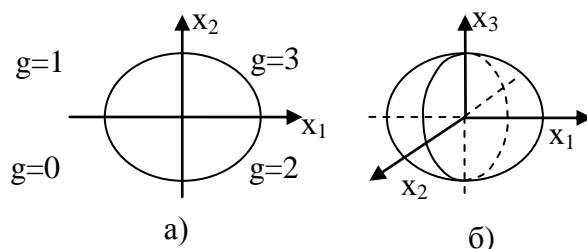


Рис. 2.3.2. Тілесні кути простору двох (а) і трьох (б) змінних

На рис. 2.3.2, а для $n=2$ наведена десятинна нумерація квадрантів. Як бачимо, вона не співпадає із звичною нам нумерацією.

Гіперкуля розділяється позитивними і негативними осями координат $(x_1, x_2, \dots, x_j, \dots, x_n)$ на *однакові за формою* тілесні T^n_g -кути, де n – кількість змінних; $g = 0, 1, 2, \dots, G$ – порядковий десятинний номер T -кута; G – максимальне значення порядкового номера T -кута.

При невід’ємних значеннях координат загальна кількість T^n_g -кутів дорівнює $(G+1) = n$, а якщо осі вміщують позитивні та негативні числові змінні, то *кількість T -кутів* дорівнює $(G+1) = 2^n$. Далі ми розглядаємо лише випадок $(G+1) = 2^n$, коли кожна ось x_j вміщує позитивні і негативні числові значення.

У загальному випадку « T^n_g -кути» будемо скорочено звати « T -кутами». T -кут виділяється в гіперкулі сукупністю площин, кожна з яких проходить через початок координат та через дві сусідні осі координат, які можуть бути і позитивними і негативними. Кожний T -кут має *масштабний коефіцієнт тілесного кута $k_m=1$, що пропорційно змінює площу, на яку спирається T -кут на гіперсфері*, та має середню лінію (лінію тяжіння), яка проходить через початок координат та через середину тіла T -кута.

В двовимірній системі координат осі координат розділяють коло на $2^n = 2^2 = 4$ нумерованих $T^{n=2}_g$ – кутів при $g=0, 1, 2, 3$; в трьохвимірній системі координат осі координат розділяють кулю на $2^n = 2^3 = 8$ нумерованих $T^{n=3}_g$ -кутів при $g=0, 1, 2, \dots, 7$. При $n = 1$ T -кут та T -куля (T -сфера) змінної x_1 мають вироджений вигляд прямої лінії з двома ($2^n = 2^1$) нумерованими T -кутами. При $n = 2$ T -куля (та T -сфера) вироджується у тілесний круг (у коло).

Окремість і незалежність T -кута означає, що «розкидані» по всьому простору, повернені на довільні кути, взаємно накладені один на одний і «перемішані» T -кути, незалежно від масштабів власних тілесних кутів і розміщення, продовжують точно і неухильно виконувати свої «математичні, аналітичні і геометричні обов’язки» стосовно єдиної прямокутної системи координат з єдиним початком координат.

Надалі порядкові номери T -кута будемо позначати як T^n_g , де n – загальна кількість змінних, $g = 0, 1, 2, \dots, G$ – порядковий десятинний номер T -кута, $G = 2^n - 1$.

$T^n_{g;r}$ -комірки. Координати (порядкові номери) T -комірки тілесного T -кута будемо помічати як $T^n_{g;r}$. Позначення $T^n_{g;r} = T^3_{5;4}$ означає: $n=3$ – загальна кількість змінних; $g=5$ – десятинний порядковий номер відповідного T -кута; $r = 4$ – порядковий номер радіусу гіперкулі, в середині якої (і на поверхні її гіперсфери) знаходяться кінці векторів даної $T^n_{g;r}$ -комірки; $r = 1, 2, \dots, E$ – визначені дослідником порядкові номери радіусів (R_1, R_2, \dots, R_E) вставлених одна в одну порожнистих гіперкуль. Всі гіперкулі розглядаються як порожнисті, вставлені одна в одну. Гіперкуля найменшого радіусу розглядається теж як порожниста з нульовим внутрішнім радіусом. Якщо порожниста гіперкуля радіусом R_r має сусідню вкладену меншу гіперкулю з меншим радіусом $R_{r-1} < R_r$, то T -комірка $T^n_{g;r}$ вміщує ті вектори, у яких їх модуль R відповідає логічній умові $R_{r-1} < R \leq R_r$ (рис. 2.3.3).

Вважаємо, що T -комірки мають пірамідальну форму « T -многогранника» з вершиною в центрі координат і однаковим (симетричним і по колу) розміщенням точок перетину своїми осями координат поверхні відповідної T -сфери. Ці точки перетину T -сфери у своїй площині є вершинами правильного T -багатокутника. При $n \rightarrow \infty$ цей правильний T -багатокутник по формі наближується до кола

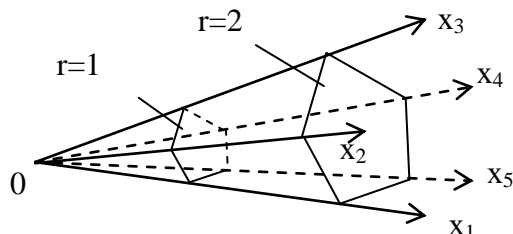


Рис. 2.3.3. Дві комірки 5-вимірного простору ($T_{g;r=1}^n$ та $T_{g;r=2}^n$)

Кожна $T_{g;r}^n$ -комірка може мати власну вихідну функцію $Q_{g;r}^n(\mathbb{R})$, підпорядковану у загальному випадку загальній функції $Q(\mathbb{R})$. Ця особливість може використовуватись в багатоагентних системах.

Очевидно, що будь-який універсум (сукупність векторів, що розглядається) цілком розподіляється по $T_{g;r}^n$ -комірках, які складають гіперкулю, причому будь-який один вектор знаходиться лише в одному T -куті (вектори, які знаходяться на межах між T -кутами, повинні відповідати заданим експертом логічним умовам належності до визначеного T -кута). Порожні T -комірки, які не вміщують вхідних векторів, не аналізуються.

2.3.3. Сусідство тілесних кутів та об'єктів

Рівень сусідства між T -кутами або об'єктами може визначатись за різними напрямками:

1. Сусідство T -кутів *відносно обраних T -кутів* [54].
2. Сусідство T -кутів *відносно обраних осей* з незмінними знаками (в тому числі: розділ T -кутів між «північною» та «південною» частками гіперкулі; сусідство T -кутів по поверхні гіперсфери в точці перетину гіперсфери однією з осей $x_1, x_2, \dots, x_j, \dots, x_n$ або стосовно перетину гіперсфери кількома однаковими осями).
3. Сусідство по найменшій *евклідовій відстані* між координатами об'єктів чи між координатами визначених однакових характерних точок T -кутів в n -вимірному просторі $X = (x_1, x_2, \dots, x_j, \dots, x_n)$.
4. Визначення положення одно-, дво- чи трьох-вимірного об'єкта в n -вимірному просторі: задаються відповідними фіксованими *числовими значеннями координат* $(x_2, \dots, x_n), (x_3, \dots, x_n), (x_4, \dots, x_n)$ при змінних об'єкта $(x_1), (x_1, x_2), (x_1, x_2, x_3)$ в n -вимірному просторі.

5. Визначення n -вимірному об'єкта в одно-, дво- чи трьох-вимірному просторі: в перелічених просторах помічають положення n -вимірному об'єкта по його відповідним координатам (x_1) , (x_1, x_2) , (x_1, x_2, x_3) , а вплив всіх інших змінних (x_2, \dots, x_n) , (x_3, \dots, x_n) , (x_4, \dots, x_n) на характеристики об'єкта помічають гранульовано (геометричною формою, кольором, цифрами, кодом, лінгвістично).

Нижче розглядається перші два напрямки визначення сусідства для T -кутів.

1. Сусідство T -кутів відносно одного обраного T -кута.

В табл. 2.3.1 наведені номери T -кутів двовимірному простору рис. 2.3.2,а в двоїстий d та десятинній g системах числення, а також показані позначення T_g^n та числові значення десятинного порядкового номеру T -кутів з визначенням їх «сусідів».

Таблиця 2.3.1.

Нумерація порядкових номерів T_g^2 -кутів у двовимірному просторі $n=2$

Номер $T_g^{n=2}$ -кута			Позначення $T_g^{n=2}$ -кута	Номер g Сусідів 1 (мають один збіжний двоїстий біт)	Номер g Сусіда 0 (всі біти незбіжні)
Десятинний номер, g	Двоїстий номер d				
	x_1	x_2			
0	0	0	T_0^2	1, 2 (T_1^2, T_2^2)	3 (T_3^2)
1	0	1	T_1^2	0, 3 (T_0^2, T_3^2)	2 (T_2^2)
2	1	0	T_2^2	3, 0 (T_3^2, T_0^2)	1 (T_1^2)
3	1	1	T_3^2	1, 2 (T_1^2, T_2^2)	0 (T_0^2)

Вважаємо, що у двовимірній системі координат сусідній тілесний «гіперкут» «Сусід 1» відрізняється від даного тілесного кута лише одним бітом двоїстого номеру (однією віссю, яка змінює свій знак на протилежний). Наприклад, для тілесного кута $g = (0,0)$ (у двоїстому численні) замінюємо $x_1 = 0$ на $x_1 = 1$ і отримуємо квадрант сусіда $g = (1,0) = 2$; для тілесного кута $g = (0,0)$ номер протилежного кута «Сусіда 0» отримується заміною всіх бітів на протилежні з отриманням значення $g = (1,1) = 3$. Аналогічним чином отримуються сусіди для всіх квадрантів $g = (0, 1, 2, 3)$.

В табл. 2.3.2 наведене визначення нумерації T -кутів трьохвимірному простору рис. 2.3.2,б.

Таблиця 2.3.2.

Десятинна нумерація T^3_g -кутів у трьохвимірному просторі $n=3$

Номер $T^{n=3}_g$ -кута				Сусіди 2 (мають два збіжні біти)	Сусіди 1 (мають один збіжний біт)	Сусід 0 (не має жодного збіжного біта)
Десятинний номер g	Двоїстий номер d					
	x_1	x_2	x_3			
0	0	0	0	1, 2, 4	3, 5, 6	7
1	0	0	1	0, 3, 5	2, 4, 7	6
2	0	1	0	0, 3, 6	1, 4, 7	5
3	0	1	1	1, 2, 7	0, 5, 6	4
4	1	0	0	0, 5, 6	1, 2, 7	3
5	1	0	1	1, 4, 7	0, 3, 6	2
6	1	1	0	2, 4, 7	0, 3, 5	1
7	1	1	1	3, 5, 6	1, 2, 4	0

У трьохвимірному просторі маємо $2^n = 2^3 = 8$ початкових T -кутів № 1 $g = 0, 1, \dots, 7$ (кут № 1 має найбільший об'єм; за рахунок введення нової системи координат всередині T -кута № 1 отримуємо менші T -кути № 2; при введенні нової системи координат всередині T -кута № 2 отримуємо ще менші T -кути № 3 і т. д.). Кожний T -кут № 1 має: 3-х «Сусідів 2», у яких два біти збіжні; 3-х «Сусідів 1», у яких один біт збіжний; одного «Сусіда 0», у якого всі три біти незбіжні (кути № 2 та № 3 мають аналогічних сусідів).

У чотирьохвимірному просторі (табл. 2.3.3) маємо $2^n = 2^4 = 16$ T -кутів № 1 $g = 0, 1, \dots, 15$. Кожна T -комірка чотирьохвимірного простору (кожний тілесний T -кут) має: 4-х «Сусідів 3», у яких три біти збіжні; 6-х «Сусідів 2», у яких два біти збіжні; 4-х «Сусідів 1», у яких один біт збіжний; одного «Сусіда 0», у якого всі чотири біти незбіжні.

Таблиця 2.3.3.

Нумерація T^4_g -комірок (T -кутів)
у чотирьохвимірному просторі $n=4$

Номер $T^{n=4}_g$ -кута					Сусід 3	Сусід 2	Сусід 1	Сусід 0
g	Двоїстий номер d							
	x_1	x_2	x_3	x_4				
0	0	0	0	0	1, 2, 4, 8	3, 5, 6, 9, 10, 12	7, 11, 13, 14	15
1	0	0	0	1	0, 3, 5, 9	2, 4, 7, 8, 11, 13	6, 10, 12, 15	14
...
14	1	1	1	0	6, 10, 12, 15	2, 4, 7, 8, 11, 13	0, 3, 5, 9	1
15	1	1	1	1	7, 11, 13, 14	3, 5, 6, 9, 10, 12	1, 2, 4, 8	0

У п'ятивимірному просторі (табл. 2.3.4) маємо $2^n = 2^5 = 32$ T -кутів № 1 $g = 0, 1, \dots, 31$, кожний з яких має: 5-х «Сусідів 4», у яких збіжні 4 біти; 4-х «Сусідів 3», у яких три біти збіжні; 3-х «Сусідів 2», у яких два біти збіжні; 2-х «Сусідів 1», у яких один біт збіжний; одного «Сусіда 0», у якого всі п'ять бітів незбіжні (в табл. 2.3.4 для її скорочення перелічені лише «Сусіди 4» та «Сусіди 0»).

Таблиця 2.3.4.

Нумерація Т-кутів у п'ятивимірному просторі $n=5$

Номер T^{n-5}_g -кута						Сусіди 4 (мають 4 збіжні біти)	...	Сусід 0 (всі біти незбіжні)
Деся- тичний номер g	Двоїстий номер d							
	x_1	x_2	x_3	x_4	x_5			
0	0	0	0	0	0	1, 2, 4, 8, 16	...	31
1	0	0	0	0	1	0, 3, 5, 9, 17	...	30
2	0	0	0	1	0	0, 3, 6, 10, 18	...	29
...
30	1	1	1	1	0	14, 22, 26, 28, 31	...	1
31	1	1	1	1	1	15, 23, 27, 29, 30	...	0

Отримана десятинна нумерація T^n_g -кутів $g = 0, 1, 2, \dots, G$ не співпадає зі звичною нам нумерацією квадрантів (при $n = 2$) і октантів (при $n = 3$). Як приклад, на рис. 2.3.2,а наведена десятинна нумерація T^2_g -кутів $g = 0, 1, 2, 3$.

З рис. 2.3.4 також випливає, що якщо ми маємо деяку фігуру (наприклад, трикутник, виділений жирними лініями), яка охоплює всі T^n_g -кути, то зміною положення центра координат (новий центр координат показаний пересіченням двох штрихових ліній) ми можемо перевести будь-яку фігуру цілком в один тілесний кут $g = 3$.

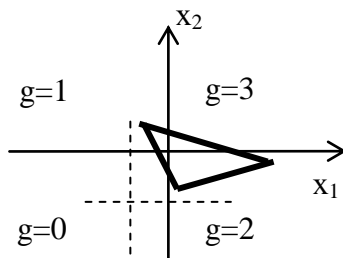


Рис. 2.3.4. Десятинна нумерація T^2_g -кутів

Вище ми розглянули «кутове» сусідство, коли вимір сусідства визначається по відношенню до одного обраного Т-кута, який стає центральною фігурою. Нижче розглянемо «осьове» сусідство, коли вимір сусідства визначається по Т-кутам, які мають осі з однаковими знаками.

2. Сусідство Т-кутів відносно однієї обраної осі з незмінним знаком.

Розглянемо позначення рівня сусідства Т-кутів відносно однієї обраної осі з незмінним знаком на прикладі даних трьохвимірного простору (x_1, x_2, x_3) табл. 2.3.2:

– G_{XXX} – початкова не розділена на частки група Т-кутів 1-го ієрархічного рівня у кількості $2^{n-3} = 2^3 = 8$, які складають у сукупності всю гіперкулю і де нижні індекси «XXX» відповідають позначенням координат (x_1, x_2, x_3), а кожний окремий індекс X може приймати значення або 0 або 1.

– G_{0XX} та G_{1XX} – дві рівні підгрупи 2-го ієрархічного рівня у кількості $2^{n-1=3-1}=2^2=4$ кожна, отримані з початкової групи G_{XXX} . G_{0XX} можна умовно назвати «північною» групою Т-кутів гіперкулі при $x_1=0$, а G_{1XX} можна назвати «південною» групою Т-кутів гіперкулі при $x_1=1$. Кожний залишений окремий нижній індекс «X» може приймати значення або 0 або 1. Нижче будемо розглядати розділ на дві рівні частки лише однієї отриманої групи G_{0XX} , бо інша група G_{1XX} розділяється на дві рівні частки аналогічно. Тут в позначеннях індексів перша цифра визначає ієрархію груп Т-кутів гіперкулі: група G_{XXX} є батьківською по відношенню до нащадкової групи G_{0XX} та G_{1XX} . В свою чергу, група G_{0XX} є батьківською для отримуваних з неї підгруп G_{01X} та G_{00X} .

– G_{01X} та G_{00X} – дві рівні підгрупи 3-го ієрархічного рівня у кількості $2^{n-2=3-2}=2^1=2$ кожна, отримані з групи G_{0XX} 2-го ієрархічного рівня. Відносно осі x_2 вони розділили між собою батьківську групу Т-кутів G_{0XX} .

– G_{011} та G_{010} – дві рівні частки групи G_{01X} у кількості $2^{n-3=3-3}=2^0=1$ кожна. Відносно осі x_3 вони розділили між собою батьківську групу Т-кутів G_{01X} .

2.3.4. Початкові дані аналітичної геометрії n-вимірних тілесних кутів

Аналітична геометрія в n-вимірних тілесних кутах (Т-кутах) визначає особливості геометричного візуального вигляду, будови і структури елементів, фігур та об'єктів з функціональною залежністю $Q(R)$, обмежених в Т-кутах сукупністю взаємно перпендикулярних осей координат, з використанням числових значень та знаків змінних $R=(\pm x_1, \pm x_2, \dots, \pm x_j, \dots, \pm x_n)$ будь-якої метрики та будь-якого фізичного змісту.

Аналітична геометрія в Т-кутах охоплює:

1. **Арифметичний n-вимірний простір** [27, с. 281–310], для якого метрики відсутні;

2. **Метричний векторний простір**, що використовує функціональні залежності $Q(R)$ n-вимірних векторів, які мають розклад по осях координат у вигляді $v = \sum_{j=1}^n v_j e_j$, де v_j – числові координати арифметичного вектора; e_j – орти, для яких можна знайти загальну метрику; $j = 1, 2, \dots, n$ – порядковий номер осі координат.

3. **Афінний векторний простір**, що характеризується n-вимірними векторами, для яких «неможливо знайти загальну метрику для ортів e_1, e_2, \dots, e_n ». Афінний векторний простір найчастіше використовується при аналізі соціологічних, хімічних, біологічних, географічних об'єктів. Але вище було показано, що контекстний характер визначення метрик осей координат робить афінний векторний простір тотожним до арифметичного. Тому вважаємо, що конкретний образ афінного простору (наприклад, по-

сада особи у залежності від її інтелектуального рівня, авторитету та зв'язків) має всі особливості вектора з числовими елементами в *арифметичному метричному векторному просторі* [27, с. 281–310], якщо якісні ознаки переведені у числові.

При аналізі Т-кутів ми будемо посилались на їх відповідність арифметичному n -вимірному простору. Арифметичний n -вимірний простір нами розглядається з точки зору ефективного розпізнавання образів. Тому при арифметичному вимірюванні змінних (ознак образу) може використовуватись:

1. *Нормалізація* – розгляд числових значень у відносних одиницях, як відношення змінної до деякої номінальної або межової (максимальної) величини. Нормалізація запобігає впливу великих чисел на розпізнавання образів, бо у загальному випадку одна змінна з найменшим числовим значенням може мати більший вплив на розпізнавання образу, ніж сукупність всіх інших разом взятих ознак.

2. *Множення* вагових коефіцієнтів впливу змінних $k_j = 0 \dots 1$ на відповідну змінну x_j , щоб урахувати практично різний вплив змінних на розпізнавання образу.

У цьому випадку ознаки розглядаються з точки зору потреб розпізнавання образів, які контекстно вимагають, щоб великі й малі числові значення ознак (змінних) розглядалися як *рівноцінні рівноправні цифри*, а сам вплив визначався якістю та ефективністю практичного розпізнавання об'єкту у заданих діапазонах координат при використанні конкретної формули залежності ефективності розпізнавання від змінних.

На рис. 2.3.5 при $n = 1, \dots, 6$ нами розглядаються Т-кути, кожний з яких *обмежений в гіперкулі заданою сукупністю осей координат x_j з урахуванням їх від'ємних значень*.

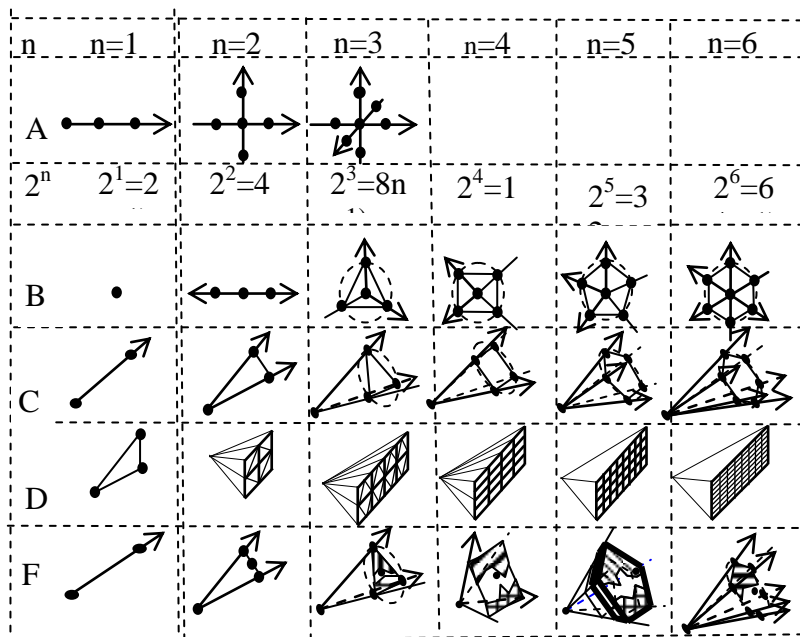


Рис. 2.3.5. Вигляд Т-кутів n -вимірного простору з вказівкою по рядках:

n – кількість осей координат ($n = 1, 2, 3, 4, 5, 6$).

A – взаємне розміщення осей координат з точками перетину поверхні «гіперсфери».

2^n – кількість T -кутів.

B – вигляд T -кута при його розгляді з поверхні гіперсфери у напрямку до центра координат з вказівкою точок перетину гіперсфери осями координат.

C – загальний геометричний вигляд окремого T -кута з вказівкою точок перетину гіперсфери осями координат.

D – загальний вигляд розміщення на площині всіх T -кутів у кількості 2^n , які у сукупності розглядаються як відображення гіперсфери n -вимірного простору.

F – використання поверхні на гіперсфері для геометричного відображення функціональних особливостей та встановлених експертом обмежень (для $n=5$ підкреслено, що відображені обмеження експерта стосуються об'єктів, які знаходяться в тілі деякої порожнистої гіперкулі, яке виділене жирними лініями).

В одновимірному просторі ознак «гіперкуля» спрощується до відрізка прямої лінії з двома ($2^n = 2^1$) нумерованими T -кутами, а в двохвимірному просторі – до круга, який розділяється осями (x_1, x_2) на чотири нумеровані T -кути ($2^n = 2^2 = 4$). При $n = 3$ осі координат (x_1, x_2, x_3) розділяють T -кулю та T -сферу на вісім нумерованих T -кутів ($2^n = 2^3 = 8$).

Для T -кутів приймаємо наступні умови їх геометричного відображення [54]:

1. Всі T -кути є геометрично та аналітично рівними між собою та незалежними. Незалежність означає, що кожний T -кут може розглядатись окремо і може бути розміщеним довільним чином у просторі змінних та по відношенню до інших T -кутів: від цього його математичний аналіз не змінюється.

2. Методом переміщення центра координат довільного T -кута всі негативні значення координат його об'єктів можна перетворити на позитивні значення. Це означає, що у принципі можна обмежитись будівництвом геометричних фігур в одному T -куті лише з позитивними (чи позитивними та негативними) значеннями координат.

3. Осі координат ($x_1, x_2, \dots, x_j, \dots, x_n$), якими виділяється T -кут в n -вимірному просторі, є рівноправними і умовно розміщуються на однаковій відстані навколо лінії тяжіння (лінії симетрії, серединні лінії) T -кута. Також умовно вважаємо, що точки перетину осей координат одного T -кута зі сферою розміщуються на одному колі з центром на серединній лінії тяжіння T -кута на рівних відстанях між сусідніми осями координат і на рівних відстанях від центру кола.

Один з варіантів можливого геометричного аналізу n -вимірного простору наведений на рис. 2.3.6.

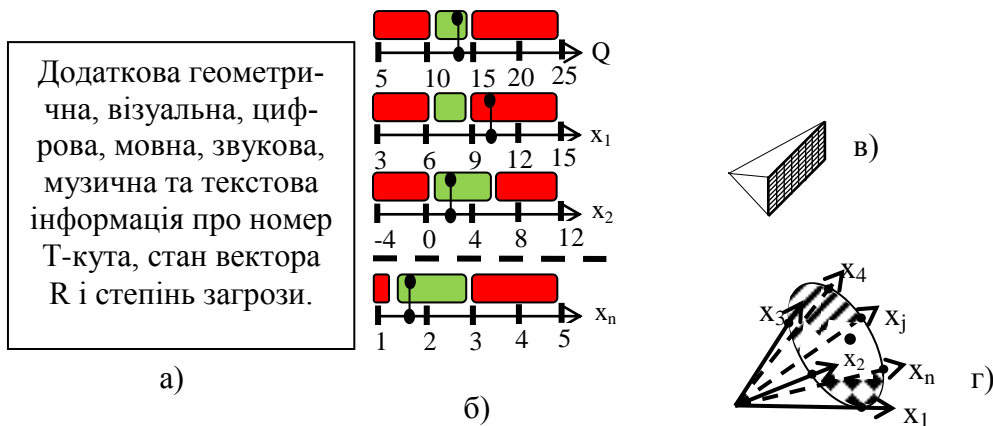


Рис. 2.3.6. Варіант можливого геометричного аналізу n -вимірного простору:

- а) додаткова інформація;
- б) контроль зміни координат з урахуванням гранулювання;
- в) контроль порядкового номера активованого Т-кута;
- г) контроль стану поточного вектора в Т-куті.

2.3.5. Точки, лінії, площини, вектори та геометричні фігури n -вимірних тілесних кутів

В n -вимірному тілесному куті між двома точками можна провести лише одну пряму; між трьома точками можна провести лише одну площину; пересічення двох площин створює пряму лінію; справедливі всі математичні залежності евклідової геометрії в n -вимірній системі координат; будь-які дві осі координат є взаємно перпендикулярними.

В n -вимірній системі координат можна геометрично побудувати точку, лінію, вектор, площину, геометричну фігуру [54].

Якщо косокутні осі системи координат є взаємно незалежними (взаємно перпендикулярними), то в системах осей координат (x_1, x_2) , (x_1, x_2, x_3) , (x_1, x_2, x_3, x_4) можна побудувати точку згідно рис. 2.3.7. На рис. 2.3.7, в показані лише складові вектора точки по осях x_1, x_2, x_3, x_4 .

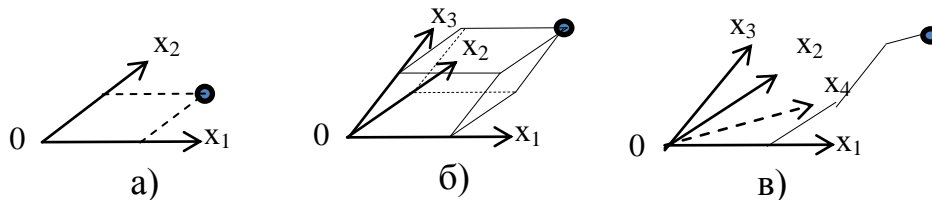


Рис. 2.3.7. Будівництво точки в системах координат:
а) (x_1, x_2) ; б) (x_1, x_2, x_3) ; в) (x_1, x_2, x_3, x_4)

Але якщо ми можемо намальовати точку в заданій системі координат, то це означає можливість намальовати в ній будь-яку фігуру.

Прикладом може бути намальовані фігури рис. 2.3.8 в системі координат $(x_1, x_2, x_3, x_4, x_5)$, які виконані у припущенні, що кожний з вказаних

п'ятикутників знаходиться у одній площині, а осі координат проходять через кутові точки багатокутника (незважаючи на зовнішній вигляд, всі осі – взаємно перпендикулярні і незалежні!).

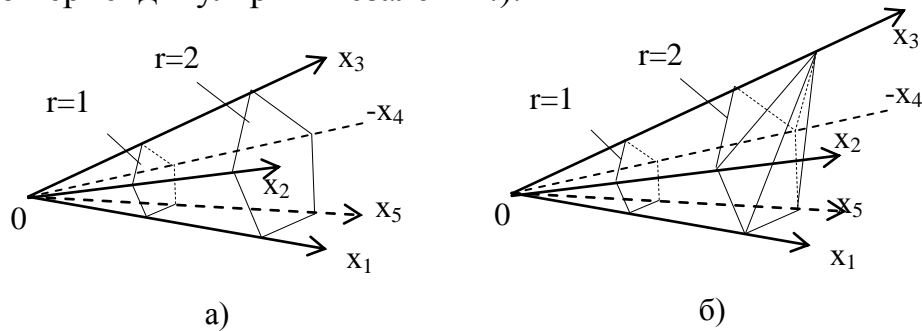


Рис. 2.3.8. Геометричні фігури у 5-вимірному просторі

У Т-комірок (із-за властивостей їх Т-кутів) відсутня вимога точного геометричного взаємного співпадіння осей координат з однаковими позначеннями у різних Т-комірок. Всі лінії тяжіння (лінії симетрії, серединні лінії) Т-комірок симетрично розміщуються в тілі Т-кулі і мають симетрично розміщені точки перетину її Т-сфери відповідними осями координат. Навколо точки перетину лінією тяжіння Т-сфери симетрично (по колу) розміщуються відповідні n вершин правильного Т-багатокутника з можливістю не співпадіння осей координат різних Т-кутів.

При наявності порожнистих куль Т-кути розрізаються кількома сферами різних радіусів ($r=1$; $r=2$) на Т-комірки; Т-комірки з більшими радіусами обмежуються з двох протилежних сторін зовнішньою та внутрішньою гіперсферами. В нижньому індексі $T_{g,r}^n$ -комірки наводиться радіус r зовнішньої гіперсфери. В геометрії найчастіше використовують системи координат: декартові прямокутні, сферичні та циліндричні. В кожній з них можна сформувати Т-кути, які цілком заповнюють заданий гіпероб'єм.

На рис. 2.3.9 наведені деякі геометричні образи чотирьохвимірному простору в декартовій прямокутній системі координат ($n = 4$). Для всіх них при зменшеному масштабі площини, на яку спирається Т-кут на гіперсфері, із центра гіперкулі 0_1 для Т-кута проведені його осі координат (x_1, x_2, x_3, x_4). Нижче описується порядок будовання геометричних образів. Рис. 2.3.9, а. По осях координат (x_1, x_2, x_3) відкладені *позитивні* координати деякої точки 4-вимірному простору, а по осі « $-x_4$ » відкладена *негативна* координата (при позитивних значеннях координати точка не належала б до даного Т-кута). Починаючи з осі x_1 по черзі додаються координати інших осей до отримання положення кінцевої точки. В цьому випадку і позитивні і негативні значення координат відкладені у однаковому напрямку вздовж осі координат.

Рис. 2.3.9, б з початку координат Т-кута 0_1 по осях координат (x_1, x_2, x_3, x_4) відкладені відповідні ребра п'ятигранника з вершиною в точці 0_1 (ребра показані жирними лініями). Для спрощення рисунка прийнято, що

чотири точки кінців цих ребер знаходяться в одній площині показаного пунктиром круга. Тому кінцеві точки ребер з'єднані між собою ребрами основи і створюють у сукупності з початком координат п'ятигранник в 4-вимірному просторі ознак. Другий п'ятигранник має за основу чотири отримані точки першого п'ятигранника, а його вершину O_2 отримуємо на середній лінії Т-кута на площині, яка визначена за допомогою другого показаного пунктиром круга. Очевидно, що вершину O_2 можна було б визначити на одній з осей координат, або визначити її положення у просторі ознак за допомогою значень її координат. Рис. 2.3.9, в. Наведена фігура 4-вимірному простору отримана при умові заміни вершини O_2 другого п'ятигранника рис. 2.6.9, б на зміщені вздовж середньої лінії чотири точки його основи.

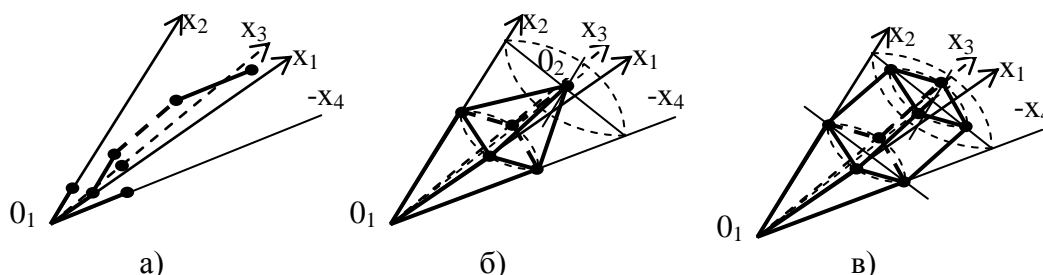


Рис. 2.3.9. Будування візуальних геометричних фігур у n -вимірному просторі: а) будування точки по заданим координатам; б) будування двох п'ятигранників; в) будування складної фігури

2.3.6. Система координат тілесних кутів на двовимірній площині

Порядок розміщення осей n -вимірного простору на площині можна визначити, якщо розглядати розміщення осей координат тілесного кута простору нижчого порядку та відповідні «площі» S , на які спираються «Т-кути» (рис. 2.3.10) [54].

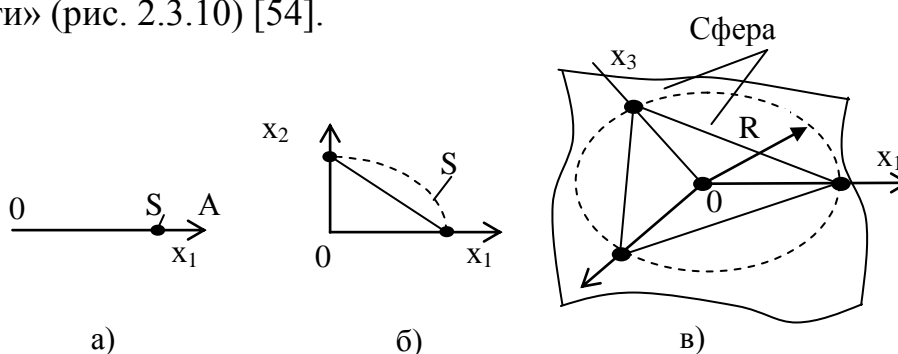


Рис. 2.3.10. Т-кути для: а) 1-вимірного простору («площа» S є точкою); б) 2-вимірного («площа» S є довжиною частки кола); в) 3-вимірного простору з центром координат 0 і перетином «гіперсфери» трьома осями координат («площа» S є часткою площі сфери)

Кожний Т-кут має середню лінію (лінію тяжіння) яка проходить через початок координат та середину тіла Т-кута і має загальний масштаб

T-кутів $k_m = 0 \dots 1$ (для поверхні їх гіперсфери), що дозволяє пропорційно змінювати площі S , на які спираються T-кути на гіперсфері радіусом R .

Результати і порядок аналізу не змінюються, якщо збільшувати чи зменшувати масштабні коефіцієнти площі ($k_m = 0 \dots 1$), на яку спираються T-кути, бо це впливає лише на їх геометричний образ.

Вплив масштабів T-кутів на геометричні зображення при $n = 2$ показаний на рис. 2.3.11.

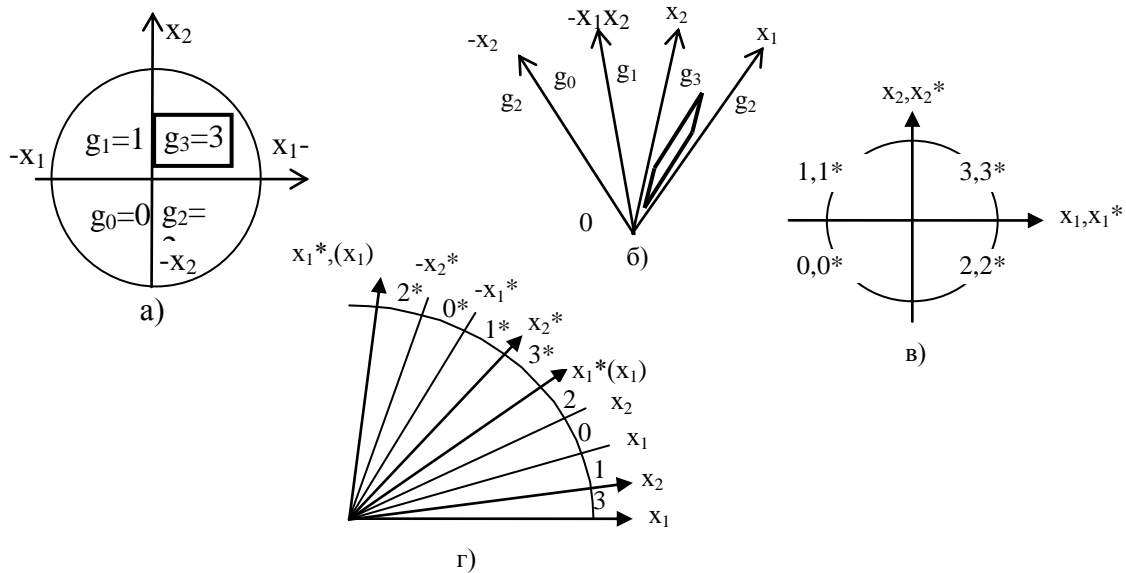


Рис. 2.3.11. Вигляд T-кутів при $n = 2$:

а, б) для одного круга з введеним прямокутником при масштабних коефіцієнтах $k_m = 1$ та $k_m < 1$; в, г) для двох кругів при масштабних коефіцієнтах T-кутів $k_m = 1$ та $k_m < 1$

На рис. 2.3.11, а для тілесного кута $g_3=3$ наведена геометрична фігура – прямокутник. Якщо зменшити масштабний коефіцієнт T-кутів $k_m < 1$, то прямокутник набуває вигляду рис. 2.3.11, б при збереженні властивості прямокутності осей в отриманій геометрично косокутній (насправді – прямокутній!) системі координат.

На рис. 2.3.11, в, г для двовимірної системи координат (x_1, x_2) показані десятинні порядкові номери T-кутів для двох кругів: («T-круг № 1» $\rightarrow g = 3, 1, 0, 2$) та («T-круг № 2» $\rightarrow g^* = 3^*, 1^*, 0^*, 2^*$) при масштабних коефіцієнтах T-кутів відповідно $k_m = 1$ та $k_m < 1$.

Для $n = 3$ при зменшеному масштабному коефіцієнті $k_m < 1$ $T^{n=3}$ -сфера та $T^{n=3}$ -куля перетворюються у частки $T^{n=3}$ -сфери та $T^{n=3}$ -кулі і набувають вигляд сукупності однакових опуклих пірамідальних T-многогранників з рівномірно розміщеними в них T-кутами і зі збереженням початку координат 0, величини радіусу R , масштабів осей координат та властивості їх прямокутності (рис. 2.3.12, а).

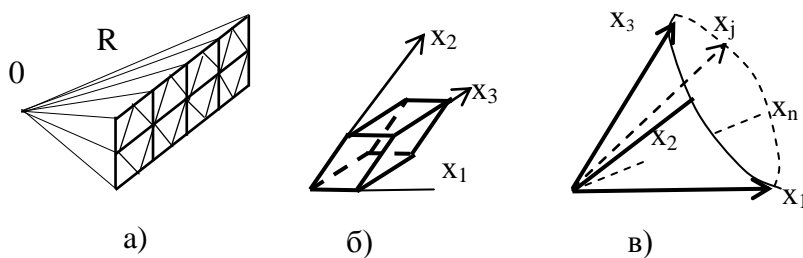


Рис. 2.3.12. Вигляд Т-кутів при зменшеному масштабному коефіцієнті $k_m < 1$: а) розгортання поверхні гіперсфери всіх кутів для $n=3$ у прямокутну площу (осі координат не наведені); б) вигляд 6-гранника в Т-куті ($x_1 < 0$; $x_2 > 0$; $x_3 > 0$); в) рекомендоване розміщення осей координат тілесного кута n -вимірному простору (осі $j = 2$ та $j = n$ -від'ємні)

Кожний Т-кут рис. 2.3.12, а в системі координат (x_1, x_2, x_3) у дійсності має «трикутну» форму виділення на «гіперсфері» (бо при $n=3$ Т-кут виділяється із «гіперсфери» *трьома площинами*), але для його графічного зображення може використовуватись й більш зручна і найбільш розповсюджена *прямокутна форма*.

Внаслідок незалежності аналізу Т-кутів, їх розміщення на двовимірній мапі при аналізі вихідної функції *може бути довільним*.

При будівництві геометричних фігур у n -вимірному просторі для зручності їх створення рекомендується розміщувати осі координат згідно рис. 2.3.12, в (вважаємо, що точки перетину осей координат одного Т-кута зі сферою розміщуються на одному колі з центром на серединній лінії тяжіння Т-кута на рівних відстанях між сусідніми осями координат і на рівних відстанях від центру кола).

Таким чином, у загальному випадку мапа на двовимірній площині для деякої складної залежності $Q(R)$, де $R = (x_1, x_2, \dots, x_j, \dots, x_n)$, має вигляд сукупності тілесних кутів розміщених довільним чином у вигляді таблиці, а кожний Т-кут має вигляд рис. 2.3.12, в, в якому й слід розглядати конкретні геометричні образи n -вимірному простору.

2.3.7. Годографи, ізолінії та геометрична, текстова, математична, акустична і мовна інформація n -вимірних тілесних кутів

Припустимо, що у спортсмена вимірюється лише один параметр – температура t (рис. 2.3.13). У цьому випадку інформаційна система видає графічний образ розміщення параметра людини у вигляді чорної точки на осі температури t з розмежуванням значень температури по степені їх небезпеки (заштрихована ділянка вказує на небезпеку, а не заштрихована ділянка є безпечною).

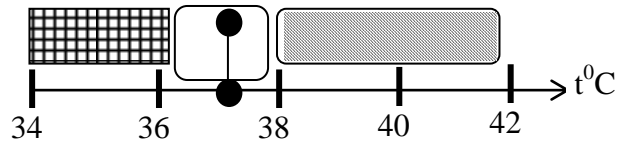


Рис. 2.3.13. Стан здоров'я пацієнта в одновимірному просторі

Якщо вимірюються два параметри (температура t та тиск p), то медичний стан людини у вигляді геометричного образу в двовимірному просторі має вигляд рис. 2.3.14, а (при роздільному аналізі параметрів) та рис. 2.3.14, б (при урахуванні їх сумісного впливу на здоров'я пацієнта). Вважаємо, що вказані границі є умовними (навчальними), а їх дійсне положення визначає лікар для кожної конкретної людини.

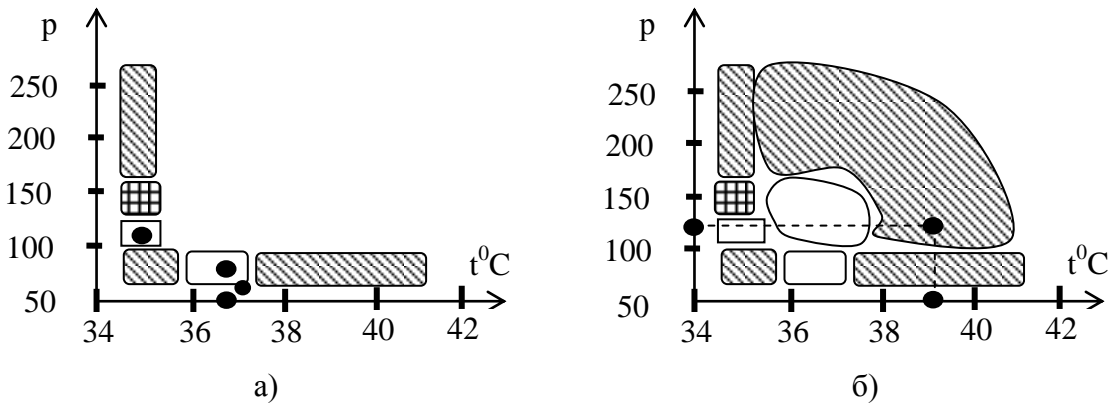


Рис. 2.3.14. Стан здоров'я людини у двовимірному просторі

Припустимо, що у людини досліджується деякий умовний фізіологічний стан в трьохвимірному просторі з вимірюванням трьох параметрів (температури t , тиску p та серцебиття f). Тоді поверхня гіперсфери геометричного образу стану людини в трьохвимірному просторі може бути відображена згідно аналітичної геометрії в n -вимірних тілесних кутах [54] у вигляді 8-ми октантів рис. 2.3.15, а для $n=3$. При будь-яких параметрах, стан людини може поточно визначатись лише в одному активованому тілесному куті на поверхні гіперсфери заданого радіусу (як це показано на рис. 2.3.15, а, б). Тілесні кути активуються по одному, по черзі, а всі інші тілесні кути не активовані. Методом зміни початку координат можна перевести всі процеси та стани лише в один тілесний кут у вигляді рис. 2.3.15, б, в якому показані дві «гіперповерхні», як частки двох гіперсфер для різних радіусів порожнистих «гіперкуль» [54].

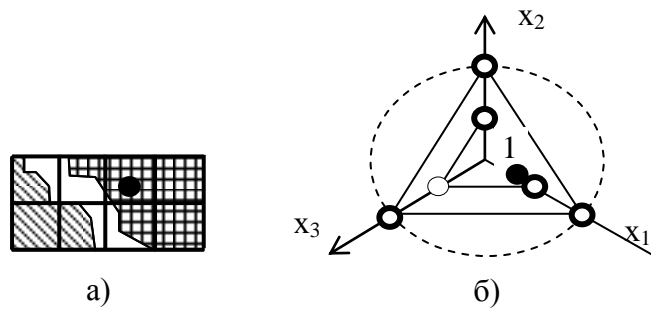


Рис. 2.3.15. Умовний досліджуваний фізіологічний стан людини в трьохвимірному просторі (різним фоном та різними геометричними фігурами показаний рівень загрози; чорна куля – стан людини): а) наведені всі тілесні кути при $n=3$ і одному радіусі гіперсфери; б) наведений один тілесний кут при двох радіусах «гіперсфери» (цифрою «1» вказаний поточний номер радіуса «гіперсфери»)

В T^5 -комірці на рис. 2.3.16 у п'ятивимірному просторі $n=5$ показаний виділений на мапі жирною штриховою лінією годограф деякого п'ятивимірного вхідного вектора з розділом T -кута сферами різних радіусів на три частки [54]. Лінії п'ятикутника утворюються перетином бокових граней T -кута з площиною, на якій розміщені точки перетину гіперсфери осями координат $Tng;g$ -комірки. Хоча на рис. 2.3.16, а осі (x_2, x_4) – додатні, а осі (x_1, x_3, x_5) – від'ємні, але якщо абсолютні значення координат у всіх інших T -кутах є чисельно однаковими, то відповідні годографи і ізолінії у всіх інших T -кутах будуть мати аналогічний геометричний вигляд (при інших рівних умовах).

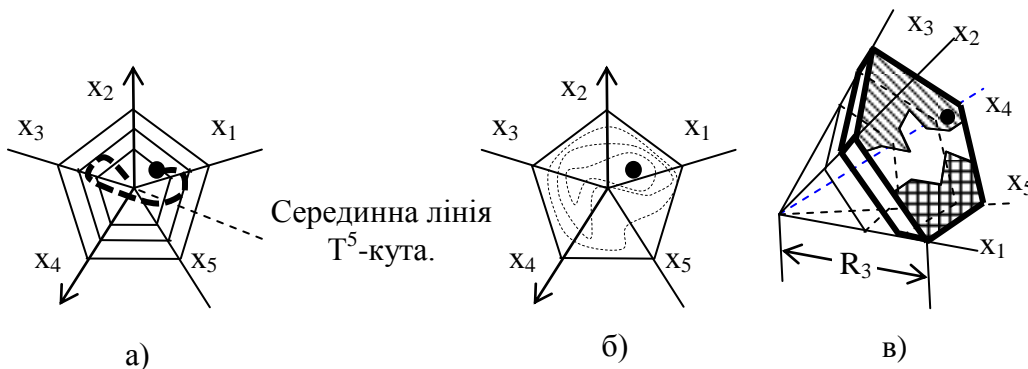


Рис. 2.3.16. Годограф (а), ізолінії (б) та визначені експертом граничні лінії станів об'єктів (в) на двовимірній мапі п'ятивимірного вхідного вектора R_3 (в) в одному T^5 -куті з його розділом сферами різних радіусів на три $T^5_{g;r}$ -комірки при $r=1, 2, 3$

Годографи звичайно проводять для поточного значення функції виходу $Q(R)$. Очевидно, що для кожного радіуса T^5 -кута на гіперсфері можна визначити ізолінії деякої складної залежності $Q(R)$, де $R = (x_1, x_2, \dots, x_j, \dots,$

хп), з доданням довільної математичної та текстової інформації.

На поверхні гіперсфери тілесного кута $T_n=5g=0$ рис. 2.3.16, в показаний у вигляді чорної точки стан об'єкта, а ступінь загрози для об'єкта визначений експертом і позначений ділянками з різними способами штрихування.

В окремих T-кутах можна ввести деякі геометричні показники математичної залежності $Q(R)$ (наприклад, точку в n-вимірному просторі ознак, максимальне, мінімальне та середнє значення $Q(R)$, положення годографа вектора R , ізолінії функції $Q(R)$).

В n-вимірному просторі ознак візуальну геометричну та іншу інформацію можна надавати у різному вигляді [54]:

- ізолініями (гранулами) функції $Q(R)$ на поверхні гіперсфери;
- годографом вектора поточного стану вектора R ;
- розміщенням паралельно осей координат з вказівкою реальних параметрів і стану вектора R з доданням текстової, мовної, звукової, музичної інформації;
- визначенням експертами різних границь можливих станів вектора R (тобто відображення кількох вихідних функцій: $Q_1(R)$, $Q_2(R)$, ...) на двовимірній мапі тієї частки сфери, яка приходить на долю конкретного T-кута;
- введенням постійних та блискаючих візуальних геометричних образів у супроводі мовної та звукової інформації;
- положенням окремих векторів та центроїда їх кластерів;
- положенням стану вектора R на кількох мапах при різних радіусах гіперкулі.

2.3.8. Двовимірна мапа тілесних n-вимірних кутів

Гіперсферу n-вимірної гіперкулі як її поверхню кулі в 3-вимірному просторі у принципі можна розглядати як мапу Землі з використанням однієї будь-якої осі координат як «осі обертання Землі». В результаті при використанні «осі обертання» x_4 , наприклад, для $n=4$ ми можемо:

- розділити T-кути між «північною» ($x_4=0$) та «південною» ($x_4=1$) ділянками гіперкулі, в яких знаходяться T-кути з координатою $x_4=0$ та $x_4=1$;
- визначити «північний» (по осі $x_4=0$) та «південний» (по осі $x_4=1$) географічні полюси гіперкулі, навколо яких на полюсах знаходяться відповідні T-кути з координатою $x_4=0$ та $x_4=1$ (виходить, що всі «північні» T-кути мають одну загальну вісь $x_4=0$ і «скупчені» навколо осі $x_4=0$; теж саме відноситься й до T-кутів «південної» гіперкулі зі знаками координат $x_4=1$);
- ввести «лінії меридіан», що з'єднують полюси, та «лінії широт», які проходять паралельно лінії екватору.

Але n-вимірний простір гіперсфери та гіперкулі у нашому випадку розділені на частки тілесними кутами у кількості 2^n . Тому бажано в цьому просторі переміщуватись не по координатах меридіан та широт, а по но-

мерам групованих Т-кутів.

Для цього розглянемо гіперкулю $n = 4$, у якій кількість тілесних кутів дорівнює $2^n = 2^4 = 16$. Нумерація тілесних кутів у чотирьохмірному просторі в двоїстому та десятинному численні наведена у табл. 2.3.5 з вказівкою лише найближчих Сусідів 3, які мають з вказаним Т-кутом три однакові осі координат (три однакові біти в двоїстому номері Т-кута) [54].

Таблиця 2.3.5.

Нумерація тілесних кутів у чотирьохмірному просторі $n = 4$

Десятинний номер g	Номер тілесного кута				Сусіди 3 (мають три однакових біти)	Сусід К (проти-лежний тілесний кут)
	Двоїстий номер d					
	x_4	x_3	x_2	x_1		
0	0	0	0	0	1, 2, 4, 8	7
1	0	0	0	1	0, 3, 5, 9	6
2	0	0	1	0	0, 3, 6, 10	5
3	0	0	1	1	1, 2, 7, 11	4
4	0	1	0	0	0, 5, 6, 12	3
5	0	1	0	1	1, 4, 7, 13	2
6	0	1	1	0	2, 4, 7, 14	1
7	0	1	1	1	3, 5, 6, 15	0
8	1	0	0	0	0, 9, 10, 12	7
9	1	0	0	1	1, 5, 11, 13	6
10	1	0	1	0	2, 8, 11, 13	5
11	1	0	1	1	3, 9, 10, 15	4
12	1	1	0	0	4, 8, 13, 14	3
13	1	1	0	1	5, 9, 12, 15	2
14	1	1	1	0	6, 10, 12, 15	2
15	1	1	1	1	7, 11, 13, 14	1

Якщо розглядати гіперкулю з даними табл. 2.3.5, як Землю, а вісь x_4 як «вісь обертання Землі», то всі Т-кути гіперкулі розділяються стосовно осі x_4 на дві частки (рис. 2.3.17, а):

– G_{0xxx} – «північну групу», яка по табл. 2.3.5 має значення $x_4 = 0$. Цю групу Т-кутів ($g = 0 - 7$) можна позначити як G_{0xxx} , що відповідає позначенню двоїстих номерів адрес $G(x_4 = 0; x_3 = x; x_2 = x; x_1 = x)$, де « x » можуть набувати комбінаторних значень або 0 або 1. Група G_{0xxx} охоплює Т-кути за десятинними номерами $g = 0 - 7$ при $x_4 = 0$;

– G_{1xxx} – «південну групу», яка по табл. 2.3.5 має значення $x_4 = 1$. Цю групу Т-кутів можна позначити як G_{1xxx} , що відповідає умовному позначенню $G(x_4 = 1; x_3 = x; x_2 = x; x_1 = x)$.

Згідно даним табл. 2.3.5 група G_{0xxx} може аналогічним чином поділитись на «східну» та «західну» частини у вигляді підгруп G_{00xx} та G_{01xx} .

Далі «східна» частина G_{00xx} теж поділяється на дві підгрупи G_{000x} та G_{001x} , по два Т-кути кожна. Можна помітити, що найбільш тісно зв'язані Т-кути, з однаковими осями G_{000x} та G_{001x} не мають однакових за номером найближчих Сусідів 3 (бо вони самі по відношенню один до одного є Сусідами 3).

З цієї точки зору для $n=4$ десятинні номери Т-кутів представлені у вигляді мапи рис. 2.3.17, а з поділом їх на відповідні групи та підгрупи.

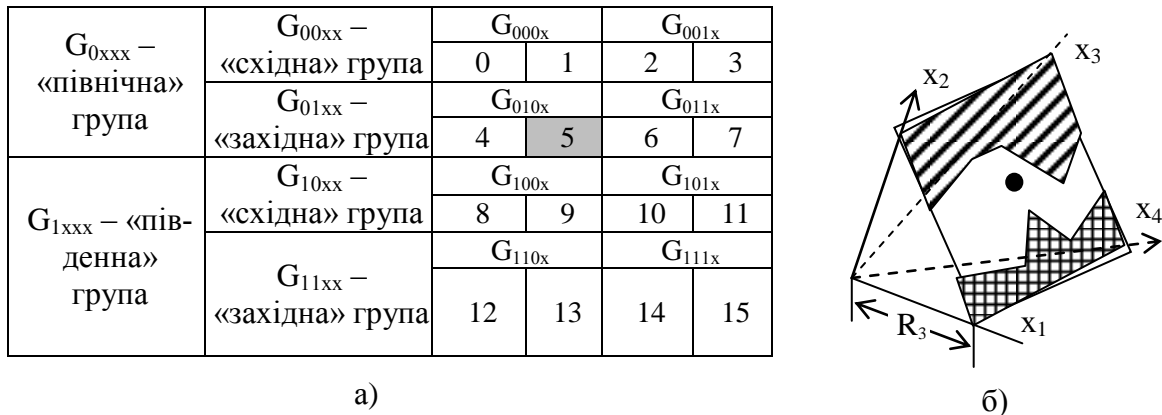


Рис 2.3.17. Представлення для $n=4$ груп Т-кутів з десятинними номерами у вигляді: а) мапи з виділеним поточним Т-кутом $g = 5$, $d = 0101$, в якому знаходиться об’єкт; б) візуального геометричного зображення гіперсфери поточним радіусом R_3 для Т-кута $g = 5$ з введеними експертом заштрихованих ділянок обмежень і з розміщеним на ній поточним положенням об’єкта, показано чорною точкою

Нумерація квадрантів та тілесних кутів у p -ятивимірному просторі для $n=5$ наведена у табл. 2.3.6.

Таблиця 2.3.6.

Нумерація квадрантів та тілесних кутів у p -ятивимірному просторі $n=5$

Десятичний номер K_i	Номер тілесного кута					Сусіди 4 (мають чотири однакових біти)	Сусід К (проти-лежний тілесний кут)
	Двоїстий номер						
	x_5	x_4	x_3	x_2	x_1		
0	0	0	0	0	0	1, 2, 4, 8, 16	31
1	0	0	0	0	1	0, 3, 5, 9, 17	30
2	0	0	0	1	0	0, 3, 6, 10, 18	29
3	0	0	0	1	1	1, 2, 7, 11, 19	28
4	0	0	1	0	0	0, 5, 6, 12, 20	27
5	0	0	1	0	1	1, 4, 7, 13, 21	26
6	0	0	1	1	0	2, 4, 7, 14, 22	25
7	0	0	1	1	1	3, 5, 6, 15, 23	24
8	0	1	0	0	0	0, 9, 10, 12, 24	23
9	0	1	0	0	1	1, 8, 11, 13, 25	22
10	0	1	0	1	0	2, 8, 11, 14, 26	21
11	0	1	0	1	1	3, 9, 10, 15, 27	20
12	0	1	1	0	0	4, 8, 13, 14, 28	19
13	0	1	1	0	1	5, 9, 12, 15, 29	18
14	0	1	1	1	0	6, 10, 12, 15, 30	17
15	0	1	1	1	1	7, 11, 13, 14, 31	16
16	1	0	0	0	0	0, 17, 18, 20, 24	15
17	1	0	0	0	1	1, 16, 19, 21, 25	14

18	1	0	0	1	0	2, 16, 19, 22, 26	13
19	1	0	0	1	1	3, 17, 18, 23, 27	12
20	1	0	1	0	0	4, 16, 21, 22, 28	11
21	1	0	1	0	1	5, 17, 20, 23, 29	10
22	1	0	1	1	0	6, 18, 20, 23, 30	9
23	1	0	1	1	1	7, 19, 21, 22, 31	8
24	1	1	0	0	0	8, 16, 25, 26, 28	7
25	1	1	0	0	1	9, 17, 24, 27, 29	6
26	1	1	0	1	0	10, 18, 24, 27, 30	5
27	1	1	0	1	1	11, 19, 25, 26, 31	4
29	1	1	1	0	0	12, 20, 24, 29, 30	3
29	1	1	1	0	1	13, 21, 25, 28, 31	2
30	1	1	1	1	0	14, 22, 26, 28, 31	1
31	1	1	1	1	1	15, 23, 27, 29, 30	0

Відповідна мапа кластерів п'ятивимірного простору $n=5$ отри-мується у вигляді розміщених по порядку номерів тілесних кутів, розділених на групи (рис. 2.3.18,а).

G_{0xxxx} – «північна» група	G_{00xxx} – «східна» група	G_{000xx}		G_{001xx}	
		0,1	2,3	4,5	6,7
G_{0xxxx} – «північна» група	G_{01xxx} – «західна» група	G_{010xx}		G_{011xx}	
		8,9	10,11	12,13	14,15
G_{1xxxx} – «південна» група	G_{10xxx} – «східна» група	G_{100xx}		G_{101xx}	
		16,17	18,19	20,21	22,23
G_{1xxxx} – «південна» група	G_{11xxx} – «західна» група	G_{110xx}		G_{111xx}	
		24,25	26,27	28,29	30,31

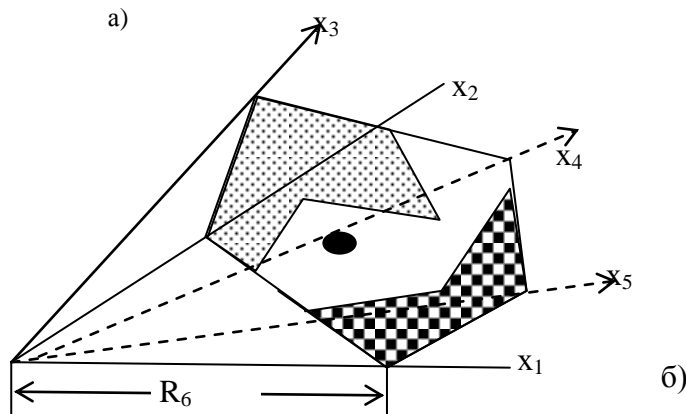


Рис 2.3.18. Представлення для $n=5$ груп Т-кутів з десятинними номерами у вигляді: а) мапи з виділеним поточним Т-кутом $g = 7, d = 00111$, в якому знаходиться об'єкт; б) візуального геометричного зображення гіперсфери з поточним радіусом R_6 для Т-кута $g = 7$ з введеними експертом заштрихованих ділянок обмежень і з розміщеним на гіперсфері поточним положенням об'єкта, показано чорною точкою

У цьому випадку на мапі рис. 2.3.18, а групування Т-кутів не доведено до найдрібнішої групи у два Т-кути, бо це не завжди й потрібно.

n-вимірною геометричною інформацією у вигляді даних рис. 2.3.17 та рис. 2.3.18 у залежності від поставленої задачі візуального спостереження, об'єму та виду потрібної інформації, може бути надана:

1. Засобами малювання в текстовому редакторі.
2. В графічному режимі будь-якої алгоритмічної мови.
3. У вигляді додаткової інформації супроводу основної геометричної візуальної інформації:
 - звуковими сигналами, включаючи голос та музику;
 - текстовими повідомленнями, включаючи анімацію;
 - зоровою інформацією, включаючи довільні зображення з анімацією, фотографіями, фільмами.

Групування Т-кутів може виконуватись наступним чином:

1. Кожний окремий Т-кут, виділений основними осями координат, що пересікаються між собою в центрі гіперкулі, може бути розділений на частки, наприклад:

- заданими радіусами гіперкулі, яка уявляється у вигляді кількох порожнистих гіперкуль, вставлених одна в одну;
- розділом на довільну кількість упорядкованих часток площі гіперсфери Т-кута, на які спираються додаткові тілесні кути, що у сукупності складають батьківський Т-кут;
- введенням нової системи координат всередині Т-кута;
- розділом експертом об'єктів Т-кута на довільну кількість об'єктних кластерів із заданим їх розміщенням на гіперсфері Т-кута (у вигляді прямої лінії, трикутника, прямокутника, шестикутника) з використанням, наприклад, метода будівництва мапи Кохонена [35; 44; 103; 168]. При цьому слід ураховувати, що в мапі Кохонена інформація може псуватись із-за: змішування інформації однієї (північної) половини гіперкулі з її протилежною (південною) половиною; змішування інформації різних радіусів гіперкулі; відсутності попередження, що на мапі зображується зовнішня поверхня (гіперсфера), а ураховання на ній глибинних об'єктів близьких до поверхні гіперсфери, збільшує похибку аналізу.

2. Т-кути та їх частки можуть об'єднуватись у довільні кластери.

2.4. Напрямки використання аналітичної геометрії *n*-вимірних тілесних кутів

Інтелектуальну систему людина звичайно розглядає як деяку функцію $Q(R) = Q(x_1, x_2, \dots, x_j, \dots, x_n)$. Тому при обмеженому значенні радіуса R всі процеси в *n*-вимірному просторі *відбуваються в гіперкулі*.

*Аналітична геометрія в *n*-вимірних тілесних кутах* може застосовуватись у наступних напрямках аналізу складної *n*-вимірної числової залежності $Q(R) = Q(x_1, x_2, \dots, x_j, \dots, x_n)$:

– для отримання на двовимірній площі *геометричних образів* n -вимірного простору у вигляді: *ізоліній*; *годографу* вектора $R = (x_1, x_2, \dots, x_j, \dots, x_n)$ з визначеними геометричними ознаками (мінімального, максимального, поточного значення); *геометричних об'єктів* (точки, лінії, площини, геометричної фігури, визначених експертом гранульованих ділянок); *двовимірної мапи* тілесних кутів з метою дослідження зміни стану підприємства по даним дискримінантного аналізу; для аналізу складних систем керування та ін. [54];

– для перетворення «чорного ящика» у «прозорий скляний ящик» за допомогою визначеної експертом мапи T -кутів на двовимірній площині та для спостереження за наближенням координат об'єкта «чорного ящика» до визначеної експертом небезпечної границі тощо;

– як *теоретична основа кластерного аналізу*, який дозволяє групувати та класифікувати стохастичні або детерміновані об'єкти з числовими ознаками $X(x_1, x_2, \dots, x_j, \dots, x_n)$;

– як *теоретична основа аналізу нейронних мереж*, що дозволяє отримати «самоорганізувальну мапу» нейронної мережі, яка має переваги перед аналогічною «мапою Кохонена» [35; 44; 103; 168] із-за того, що навчання вагових коефіцієнтів нейронів замінене реєстрацією навчальних даних в тілесних кластерах з відомими адресами з отриманням реальної кількості об'єктних кластерів при відсутності псування навчальної інформації додатковими латеральними зв'язками та «впливом сусідів»;

– як *теоретична основа багатоагентних систем*, в яких кожний агент у вигляді T -комірки виконує свою частку задачі по розпізнаванню образів і може мати власну вихідну функцію, підпорядковану загальній меті;

– для *систем гранульованих координат*, теорія яких запропонована Л. Заде [206; 207], і далі розширена, наприклад, в моделях, що використовують гранули однакових розмірів з однаковим математичним представленням у вигляді матриць під назвою базових елементів Грасмана [16].

– для *прийняття рішення* $F_{f1}(R_x)$, яке залежить від стану ієрархічно вищої системи $F_{f0}(R_{z0})$ та станів нижчих систем $F_{f2}(R_{z2})$;

для *аналізу перехідних процесів складної нелінійної системи n -го порядку*, яка описується диференціальними рівняннями першого порядку

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \dots, x_j, \dots, x_n)$$
 у залежності від часу з урахуванням початкових умов. В результаті числового інтегрування отримуються нові координати стану технічної системи в n -вимірному просторі, що визначають *годограф*, який дозволяє візуально контролювати сталість системи у часі та залежність перехідних процесів від параметрів та збурень. Для переходу до методу аналізу перехідних процесів у фазовій площині необхідно n змінних $X=(x_1, x_2, \dots, x_j, \dots, x_n)$ замінити двома змінними, наприклад, у вигляді

$$u_1(u_2), R(t), Q(R,t),$$

де $u_1 = \sqrt{x_1^2 + x_2^2 + \dots + x_i^2}$, $u_2 = \sqrt{x_{i+1}^2 + x_{i+2}^2 + \dots + x_n^2}$; t – час.

– як теоретична основа створення «загальної теорії всіх знань людства», якщо в одній n -вимірній системі осей координат описати всі знання у вигляді цифрових складних залежностей при наявності різних або частково співпадаючих сукупностей осей. При спрощенні системи координат (вилученні осей окремих знань), ця модель в спрощеній системі координат буде давати вірні і підтверджені практикою дані. Подібні моделі є **шкідливими**, якщо вони не додають нових знань, бо вони лише ускладнюють аналіз, викликають даремні витрати ресурсів та часу на їх створення, освоєння, застосування і спростовування. З цієї точки зору треба обережно ставитись до подібних пропонованих об'єднуючих моделей знань: вони правдоподібні і навіть вірні, але наносять шкоду і не мають жодної **нової** наукової чи практичної цінності.

2.5. Теорія тілесних кластерів n -вимірного простору

Кластерний аналіз об'єктів (від англійського слова cluster – гроно, скупчення) виконує збір даних про об'єкти, що характеризують порівняно однорідні групи універсуму. Великі кластери можуть бути складеними з менших кластерів. Для заданого універсуму кількість кластерів може визначатись людиною, але є і автоматизовані методи визначення кількості кластерів. Існує приблизно 100 методів кластеризації об'єктів [35; 169; 170].

Нижче розглядається **теорія тілесних кластерів $T_{g,r}^n$ на основі даних аналітичної геометрії в n -вимірних тілесних кутах** [54].

Тілесний $T_{g,r}^n$ -кластер є $T_{g,r}^n$ -коміркою і має тотожне позначення, де n – кількість змінних, g – десятинний порядковий номер T -кута в гіперсфері, а r – зовнішній радіус гіперсфери кластера. Будь-які вхідні вектори обов'язково попадають в один з кластерів № 1 різних зовнішніх радіусів r (кластер № 1 – найбільший тілесний кластер). На відміну від *кластера об'єктів*, **тілесний кластер** може вміщувати нескінченну кількість точок – об'єктів.

При кластерному аналізі вхідні координати бажано нормалізувати. Вважаємо, що кластери є векторами об'єктів з однаковими знаками при числових значеннях їх координат. Тому всі об'єкти універсуму без винятку розподіляється по початковим кластерам (T -кутам), які потім можна поєднувати, або подрібнювати з наступним об'єднанням.

Розглянемо кластерний аналіз в двовимірній системі координат на прикладі рис. 2.5.1, на якому показані 15 векторів трьох класів ($i = 1, 2, 3$). Кінцеві точки векторів для різних класів мають різну форму і позначені цифрами $e = 1, 2, \dots, 15$. Порядкові номери T -кутів в десятинній системі числення $g = 0, g=1, g=2, g=3$ для осей координат (x_1, x_2) виділені сірим фонном. Кластер $T_{g,r}^n$ може супроводжуватись окремою математичною і лінгвістичною інформацією.

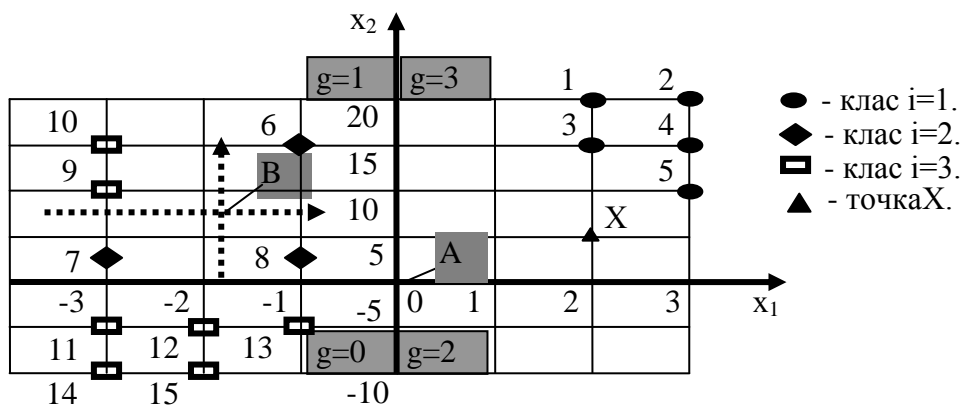


Рис. 2.5.1. Об'єкти класифікації

Тілесні кути, виділені у просторі змінних осями координат, будемо звати «*тілесними кластерами*». На відміну від звичайних *об'єктних кластерів*, які складаються з «груна» (множини) об'єктів з розрахованим «центроїдом» (умовним вектором з осередненими координатами), «тілесний кластер» вміщує тілесний n -вимірний простір T -кута, в якому розміщуються об'єкти. «Центроїдом» (умовним вектором з осередненими координатами) «*тілесного кластера*» є координати заданої експертом точки всередині T -кута. Дані векторів, які входять в тілесний кластер, можуть бути використані для переходу до звичайних об'єктних кластерів.

Згідно початкової системи осей координат з початком координат для першого ієрархічного рівня $\lambda = 1$ в точці $A(x_1^{\lambda=1} = 0; x_2^{\lambda=1} = 0)$, всі навчальні вектори $e = 1, \dots, 15$ експерт вводить у табл. 2.5.1, в якій вказує для кожного вектора: порядковий номер $e = 1, \dots, 15$; клас; координати; двоїстий d та десятинний g порядкові номери їх T -кутів.

Таблиця 2.5.1.

Перший ієрархічний рівень $\lambda = 1$ з точкою початку координат $A(x_1^{\lambda=1} = 0; x_2^{\lambda=1} = 0)$

Дані навчального вектора $X^{k,e}$				Порядкові номери T -кутів	
Номер e	Контроль k	Клас i	Координати $x_{k,e,\lambda=1}^1; x_{k,e,\lambda=1}^2$	Двоїстий номер T -кута $d = (x_1^C; x_2^C)$	Десятинний номер T -кута g
1	0	1	2;20	1;1	3
2	0	1	3;20	1;1	3
3	0	1	2;15	1;1	3
4	0	1	3;15	1;1	3
5	0	1	3;10	1;1	3
6	1	2	-1;15	0;1	1
7	1	2	-3;2,5	0;1	1
8	1	2	-1;2,5	0;1	1
9	1	3	-3;10	0;1	1

10	1	3	-3;15	0;1	1
11	0	3	-3;-5	0;0	0
12	0	3	-2;-5	0;0	0
13	0	3	-1;-5	0;0	0
14	0	3	-3;-10	0;0	0
15	0	3	-2;-10	0;0	0

Колонка «Контроль k» означає:

– при $k=0$ всі вектори е даного кута мають однаковий клас, і Т-кут не слід ділити на окремі додаткові тілесні кластери;

– при $k=1$ вектори е даного кута мають різний клас, і Т-кут потрібно ділити на окремі додаткові тілесні кластери.

В табл. 2.5.1 у колонці «Контроль k», керуючись даними колонки класу відповідного Т-кута, вказують значення типу $k=0$ (об'єкти Т-кута не потрібно розділяти) або $k=1$ (об'єкти Т-кута належить розділити, бо вони відносяться до різних класів). В табл. 2.5.1 сірим фоном виділені комірочки для розділу об'єктів Т-кута ($k=1$).

Нижче для скорочення будемо посилатись лише на десятинні номери Т-кута g.

З даних табл. 2.5.1 та рис. 2.5.1 на основі системи координат, центр якої помічений літерою А, для першого ієрархічного рівня $\lambda = 1$ впливає:

– В Т-куті $g = 0$ присутні навчальні вектори е одного класу $i = 3$. Тому Т-кут $g = 0$ відноситься до типу $k = 0$. Внаслідок цього вважаємо, що якщо будь-який вхідний невідомий вектор, який належить класифікувати, входить у Т-кут $g = 0$, то цей вектор має клас $i = 3$.

– В Т-куті $g = 1$ присутні вектори двох класів ($i=2$ та $i = 3$), і тому Т-кут відноситься до типу $k = 1$. Нову скорочену множину нижчої ієрархії $\lambda=2$, яка складається з векторів $e = 6, 7, 8, 9, 10$, що попали у даний Т-кут № 1 типу $k = 1$, потрібно розділити на дрібніші тілесні кластери № 2, аналогічно описаному раніше з метою отримання Т-кутів типу $k = 0$.

– В Т-куті $g = 2$ навчальні вектори відсутні і тому в табл. 2.5.1 цей Т-кут відсутній.

– В Т-куті $g = 3$ присутні навчальні вектори одного класу $i = 1$. Тому Т-кут $g = 3$ відносять до типу $k = 0$. Внаслідок цього вважаємо, що якщо будь-який вхідний невідомий вектор X (рис. 2.5.1), який належить класифікувати, входить у Т-кут $g = 3$, то цей вектор X має клас $i = 1$.

Таким чином, на першому ієрархічному рівні $\lambda = 1$ згідно даних табл. 2.5.1 та рис. 2.5.1 отриманий лише один Т-кут $g = 1$ типу $k = 1$ з навчальними векторами $e = 6, 7, 8, 9, 10$, сукупність яких експерт розглядає як ієрархічно нижчу скорочену навчальну множину, яку потрібно далі розділяти на наступні менші тілесні кластери з метою отримання тілесних кластерів лише типу $k = 0$.

Для нижчого ієрархічного рівня $\lambda = 2$ експерт позначив на рис. 2.5.1 пунктирними лініями осі нової системи координат з центром в точці В ($\Delta x_1^{\lambda=2} = -1,8$; $\Delta x_2^{\lambda=2} = +7,5$). Для всіх навчальних векторів $e = 6, 7, 8, 9, 10$, які знаходяться в Т-куті $g = 1$ типу $k = 1$ ієрархічного рівня $\lambda = 1$, експерт перераховує значення їх нових координат за формулою

$x_j^{e,\lambda=2} = (x_j^{e,\lambda=1} - \Delta x_j^{\lambda=2})$ і вводить їх у табл. 2.5.2 з виконанням у ній розрахунків, аналогічних табл. 2.5.1.

Таблиця 2.5.2.

Другий ієрархічний рівень $\lambda = 2$ з новою точкою початку координат В ($\Delta x_1^{\lambda=2} = -1,8$; $\Delta x_2^{\lambda=2} = +7,5$)

Дані навчального вектора				Порядкові номери Т-кутів	
Номер е	Контроль к	Клас і	Координати $x_1^{k,e,\lambda=2}$; $x_2^{k,e,\lambda=2}$	Двоїстий номер Т-кута $d = (x_2^C; x_1^C)$	Десятинний номер Т-кута g
6	0	2	+0,8;+7,5	1;1	3
7	0	2	-1,2;-5	0;0	0
8	0	2	+0,8;-5	1;0	2
9	0	3	-1,2;+2,5	0;1	1
10	0	3	-1,2;+7,5	0;1	1

В табл. 2.5.2 у колонці «Контроль к» вказують значення типу к відповідного Т-кута.

З даних табл. 2.5.2 та 2.5.1 для другого ієрархічного рівня $\lambda = 2$ з системою координат, поміченою літерою В, випливає, що усі Т-кути $g = 0, 1, 2, 3$ відносяться до типу $k = 0$.

Отримані для класифікації тілесні кластери мають наступні позначення $T_{g=0}^{n=2}$ -кутів:

- Клас $i=1$: $T_{g=3}^{n=2}$.
- Клас $i=2$: $T_{g=1}^{n=2}$: ($T_{g=3}^{n=2}, T_{g=2}^{n=2}, T_{g=0}^{n=2}$). Нова точка початку координат В ($\Delta x_1^{\lambda=2} = -1,8$; $\Delta x_2^{\lambda=2} = +7,5$).
- Клас $i=3$: $T_{g=0}^{n=2}$; $T_{g=1}^{n=2}$: ($T_{g=1}^{n=2}$). Нова точка початку координат В ($\Delta x_1^{\lambda=2} = -1,8$; $\Delta x_2^{\lambda=2} = +7,5$).
- Непізнаний клас: $T_{g=1}^{n=2}$.

Об'єкт з координатами точки Х ($x_1=2$; $x_2=5$) (рис. 2.5.1) знаходиться в Т-куті $T_{g=3}^{n=2}$ і тому має клас 1.

Отримані дані по тілесним кластерам можна аналізувати далі по об'єктним кластерам. Об'єкти отриманих тілесних кластерів перевіряють на щільність всередині кластерів і можливість їх об'єднання або розділу на окремі кластери (тілесні або об'єктні).

Стосовно об'єктів в отриманих тілесних кластерах виконуються наступні дії:

- не аналізуються і не розглядаються тілесні кластери без векторів (без об'єктів) в них;

– для заповнених векторами тілесних кластерів $T_{g;r}^n$ визначаються координати векторів-центроїдів кластерів (осереднених значень координат їх векторів, що попали у кластер) у вигляді

$$X_{g;r}^C = (x_{gr;1}^C, x_{gr;2}^C, \dots, x_{gr;j}^C, \dots, x_{gr;n}^C), \quad (2.5.1)$$

де $x_{gr;j}^C = \frac{1}{A_{g;r}} \sum_{gr=1}^{A_{g;r}} x_{gr;j}$,

$gr = 1, 2, \dots, A_{g;r}$ – порядковий номер вектора, який попав в кластер $T_{g;r}^n$; $A_{g;r}$ – загальне число векторів, яке попало в кластер $T_{g;r}^n$.

Значення $X_{g;r}^C$ є центроїдом-еталоном, призначеним для розпізнавання векторів стосовно їх належності до кластера № 1 $T_{g;r}^n$ -кута;

– з метою виявлення необхідності подальшого розділу об’єктних кластерів № 1 на менші частки, *всередині об’єктних кластерів $T_{g;r}^n$* для всіх пар векторів з порядковими номерами $gr = k$ та $gr = \delta$ при $\delta \neq k$ розраховуються квадрати евклідової відстані

$$X_{gr;k\delta}^E = \sum_{j=1}^n (x_{gr=k,j} - x_{gr=\delta,j})^2. \quad (2.5.2)$$

Евклідові відстані (2.5.2) не повинні перевищувати квадрат заданого дослідником максимального дозволеного значення відстані між всіма парами векторів всередині кластерів;

– з метою виявлення можливості об’єднання об’єктних кластерів № 1 визначаються квадрати евклідових відстаней між всіма парами векторів-центроїдів $X_{g;r}^C$ з порядковими номерами $gr = \alpha$ та $gr = \beta$

$$(\Delta X_{gr;\alpha\beta}^C)^2 = \sum_{j=1}^n (x_{gr=\alpha,j}^C - x_{gr=\beta,j}^C)^2; \quad (2.5.3)$$

якщо значення $(\Delta X_{gr;\alpha\beta}^C)^2$ перевищує квадрат заданого експертом максимального дозволеного значення відстані між центроїдами, то дана пара центроїдів не об’єднується, а у іншому разі вони об’єднуються у складний кластер.

Ми розглядаємо автоматизовану систему. Остаточне рішення про об’єднання кластерів у складні кластери та про роз’єднання кластера на менші кластери звичайно приймає експерт. Але ці дії можуть відбуватись також і автоматично.

Складний кластер при збільшенні в ньому кількості об’єднаних кластерів у просторі змінних може набувати будь-яку форму: випуклої чи не випуклої фігури; «черв’яка», що охоплює своїм «тілом» весь простір змінних і вміщує центроїди елементарних кластерів з дуже великими відстанями між ними. Відомо, що складний кластер у загальному випадку описується не одним вектором-центроїдом, а сукупністю векторів-центроїдів відповідних елементарних кластерів, що входять у складний кластер [35]. Відповідно вважаємо, що вектор входить у складний кластер, якщо він має

відстань до будь-якого з центроїдів складного кластера, меншу за дозволену експертом величину відстані між двома центроїдами.

Якщо евклідові відстані між векторами всередині кластера мають більші за дозволені значення, то це означає необхідність розділу такого кластера № 1 на менші кластери № 2. Подібний розділ великих кластерів № 1 на менші кластери № 2 (а після цього на ще менші кластери № 3, № 4...) потрібно виконувати неодноразово до задоволення вимоги стосовно заборони перевищення дозволених значень евклідових відстаней між векторами всередині великих і малих кластерів.

Слід урахувати, що розрахунки відстаней між векторами, розрахунки координат векторів центроїдів та відстаней між центроїдами виконуються по початковим значенням координат векторів, які не змінюються на протязі всього аналізу.

Після виконання вказаних дій, навчання є завершеним, і отримані кластери можуть використовуватись для класифікації вхідних векторів.

Кластери можна зменшити по об'єму збільшенням кількості вкладених порожнистих гіперкуль (збільшенням кількості внутрішніх радіусів), чи розділом (за різними способами) на частки тілесних кутів. У цьому відношенні універсальним є розглянутий нами вище «метод перенесення початку координат» [54], бо він зменшує одразу дві координати кластера: вздовж осі радіусу гіперсфери та вздовж осі зміни тілесного кута (наприклад, для рис. 2.5.1 кластер № 5 менше за кластер № 1 в $(2^n)^{5-1} = (2^2)^{5-1} = (4)^4 = 256$ разів).

Завдання

1. **Визначення тілесних та об'єктних кластерів.** В площині 4-х квадрантів, отриманих за допомогою осей координат $X1$, $X2$ при максимальних значеннях змінних $X1_{max} = N$, $X2_{max} = N$, де N – порядковий номер студента у групі, вказати точками 15 довільно розміщених навчальних об'єктів 3-х класів (точки об'єктів різних класів мають різний графічний вигляд, кількість об'єктів в класах – довільна). В чотирьох квадрантах 15 навчальних об'єктів розмістити довільно за правилами: один квадрант – порожній; один квадрант вміщує об'єкти 1-го класу; один квадрант вміщує об'єкти 2-го класу; один квадрант вміщує об'єкти класів 1–2.

В квадранті з об'єктами класів 1–3 ввести дві додаткові системи осей координат, стосовно яких розмістити 15 навчальних об'єктів таким чином, щоб при визначенні тілесних кластерів отримати три ієрархічних рівня $\lambda = 1, 2, 3$ тілесних кластерів для розділу об'єктів по групах.

На рисунку привести положення 15-ти точок для навчальних об'єктів та положення в трьох квадрантах трьох додаткових довільних векторів, які належить класифікувати.

Визначити:

- адреси тілесних кластерів для класифікації невідомих об'єктів у вигляді послідовності пов'язаних між собою T-кутів та приблизні (по рисунку) координати центроїдів тілесних кластерів у вигляді «центра тяжіння» їх тіла;
- центроїди об'єктних кластерів;
- положення нульових точок систем координат трьох ієрархічних рівнів $\lambda = 1, 2, 3$;
- класи трьох додаткових довільних векторів по адресах тілесних комірок і по центроїдах об'єктних кластерів.

3. Обробка ознак та групування образів

3.1. Прості методи класифікації образів

1. Метод зондів

Метод зондів був запропонований британським вченим Д. Даймондом (1958). Він використовується для розпізнавання літер та цифр, навіть написаних від руки та з деякими відхиленнями у розмірах та стилі написання [18; 100]. Розглянемо принцип використання зондів у припущенні, що наша система РО повинна розпізнавати образи лише чотирьох великих друкованих літер «ЛЕНА», написаних від руки.

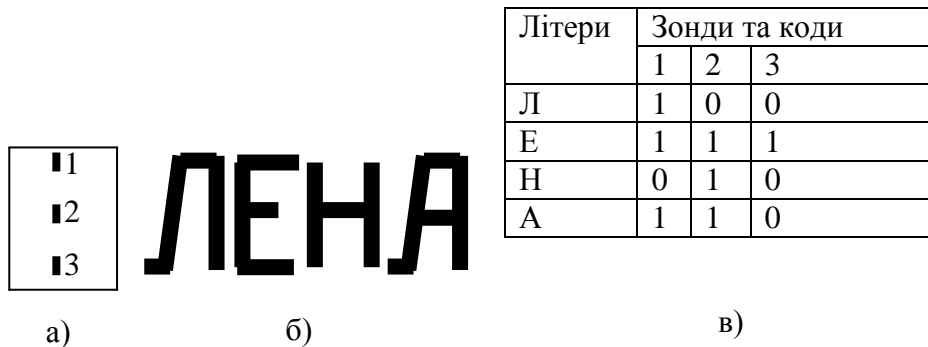


Рис. 3.1.1. Використання зондів: а) вигляд зондів для літер «ЛЕНА»; б) слово «ЛЕНА»; в) коди літер «ЛЕНА»

Ці літери пишуться у заданих розмірах *електропровідними чорнилами* на звичайному папері. Можна вважати, що *кожний зонд 1, 2, 3 (рис. 3.1.1, а) складається з двох електродів*, розділених ізоляційним проміжком. Дотик до електропровідних чорнил на папері електрично з'єднує ці два електроди, по них протікає струм, який і фіксує наявність лінії символу на даній ділянці під зондом. Сигнали від усіх електродів, що дотикаються до літери, передаються на дешифратор, який і виконує класифікацію літери (рис. 3.1.1, в). Кожному зображенню одного класу (одній літері) відповідає одна комбінація збуджених зондів. Зонд вважається збудженим, якщо він пересікає лінії літери. Реакцією зонду є «1» у разі збудження і «0» в іншому випадку. Таким чином, зонди видають код, який складається з нулів та одиниць.

В дійсності зонди для визначення літер усього алфавіту мають складніший вигляд у порівнянні з показаними на рис. 3.1.1, бо розглядається більша кількість літер і одночасно враховується можливе відхилення форми літери від еталона. Можуть також використовуватись зонди, які окреслю-

ють літеру з боків і центрують зонди для точного визначення місця знаходження літери. Зонди можуть також бути *оптичними* і виглядати як тонкостінні металеві рупори, які прикладаються до літери; із протилежного кінця рупор має вигляд невеликого квадрата з фотомножником на ньому. В цьому випадку чорнила можуть бути звичайними.

2. Квазітопологічний метод

Топологія – це наука, яка розглядає властивості графів. Граф складається з точок (вершин) та з'єднуючих їх ребер (ліній). Кожному ребру відповідають дві вершини. Математичний опис графу не залежить від його креслення. Тому за допомогою графів можна розпізнавати деформований текст (рис. 3.1.2, а).

Хоча ми будемо розглядати не топологічний, а квазітопологічний метод, все ж деякі властивості графів використовуються. При цьому методі встановлюються правила, згідно з якими кожній літері алфавіту призначається якесь число. Один із таких алгоритмів розглянемо стосовно літери «А» (рис. 3.1.2, б):

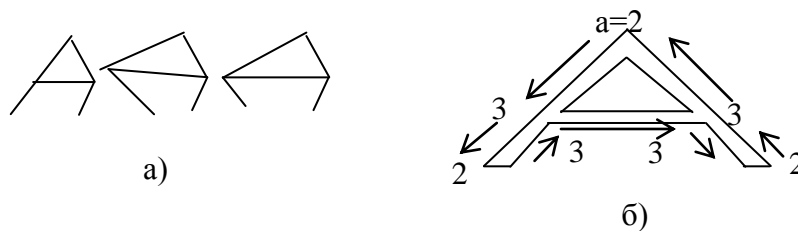


Рис. 3.1.2. Квазітопологічний метод: а) графи літери «А»; б) отримання коду літери «А»

1. Обхід літери виконується проти руху стрілки годинника по замкненому зовнішньому контуру, починаючи з верхньої правої точки, яка позначається кодом « $a=2$ ».

2. При обході літери позначається цифра коду, яка вказує загальну кількість ребер, що належить вершині графу.

Згідно цих правил кодова комбінація літери «А» має вигляд «2,3,2,3,3,2,3».

Недоліки цього методу: бруд, переміщення лінії затрудняє якісне розпізнавання образу; потрібні додаткові ознаки для розпізнавання, наприклад, літер «І», «Ї», «И», «Й», «О».

3. Метод розпізнавання автомобілів, який використовується в інтелектуальних системах для регулювання руху автотранспорту [117]. Під полотно дороги закладають електричну індукційну котушку і вимірюють наведену в ній електрорушійну силу (ЕРС). Коли над котушкою проїзжає автотранспорт, то в ній наводиться ЕРС, яка викликається зміною магнітного поля Землі у даному місці. Цю ЕРС нормалізують по швидкості руху транспорту і по амплітуді ЕРС і подають у логічну частку розпізнавальної системи.

4. Метод центральних моментів для розпізнавання рухомих об'єктів

Вважаємо, що на площині ознак за допомогою локатора отримано зображення літака на площині ознак X, Y [19; 117]. Для кожного літака апріорно визначаються координати заданих характерних точок (наприклад, початок та кінець корпусу літака, координати кінцівок крил тощо).

Ці координати беруться відносно середніх значень всій множини точок, що розглядаються.

Центральний момент розраховується за формулою

$$M_{pq} = \frac{1}{N} \sum_{i=1}^N (X_i - X_{cp})^p (Y_i - Y_{cp})^q,$$

де X_i, Y_i – координати обраних точок на об'єкті; $i = 1, 2, \dots, N$ – порядковий номер точки; X_{cp}, Y_{cp} – середні значення координат на множині з N точок; p, q – довільний ступінь (звичайно $p = q = 2$).

Для розпізнавання образу використовуються також додаткові ознаки у вигляді моментів:

$$M_1 = M_{20} + M_{02}; M_2 = (M_{20} + M_{02})^2 + 4M_{11}; M_3 = (M_{30} - 3M_{12})^2 + (3M_{21} - M_{03})^2.$$

Тут подвійні нижні індекси означають відповідні значення ступенів p та q .

Отримані значення моментів складають багатомірний вектор, який використовується для визначення відстані до еталонних значень. Цей метод не є достатньо надійним.

5. Метод суміщення з еталоном

Розпізнавальні системи розділяються по способу зберігання еталонних наборів: на системи з фотомаскою, системи з електричними моделями еталонів (у вигляді схем резисторів; феритових осердь із котушками і т. п.) та математичні системи [18].

В системах РО з фотомаскою використовується метод суміщення об'єкту з еталоном. Об'єкт, що розпізнається (наприклад, літери), проектується оптичною системою на еталонні маски. Наприклад, маска – це трафаретний отвір у непрозорому матеріалі. Промінь світла освітлює об'єкт (який звичайно має темний колір на білому фоні) і через маску направляється оптичною системою на фотомножник, розміщений за еталонною маскою. Чим більше темний об'єкт співпадає з еталоном, тим менше освітлюється фотомножник. В ідеалі при повному співпадінні зображення об'єкту з еталоном, освітлення фотомножника мінімальне, і вихідний сигнал дорівнює нулю, що є ознакою розпізнавання образу. Еталонні маски подаються до порівняння по порядку, і тому знають, з якою літерою співпав об'єкт.

Недоліки такої системи РО: якщо розпізнається літера, то з цією літерою порівнюються по черзі всі маски; забруднення об'єкту приведе до невірної класифікації; об'єкт повинний бути строго визначених розмірів і на визначеній відстані від оптичної системи; потрібна дуже точна центровка схеми.

У системах РО з електричними моделями розпізнавання порівнюється «зважена» сума сигналів від рецепторного поля об'єкту з еталонами у вигляді матриць опорів, або феритових осердь із катушками.

Математична модель системи РО, яка теж використовує принцип суміщення з еталонами, поділяє зображення на багато елементарних прямокутних комірок, які у сукупності створюють загальний прямокутник зображення. У кожній комірці вимірюється яскравість, яка характеризує освітлення, і виробляється цифра коду: «+1» відповідає освітленій комірці, а «-1» – неосвітленій. Результуючий код складається з всіх рядків рецепторного поля і виглядає як рядок, складений «+1» та «-1».

Набір результатів таких вимірювань рецепторного поля створює вектор, який у двоїстій системі числення має, наприклад, вигляд $P = \{-1-1-1+1+1+1\}$. Відповідний математичний еталон теж має вигляд вектора $M_j = \{+1-1-1+1-1+1\}$, який може відрізнитися від зображення рецепторного поля Р. Скалярний добуток характеризує їхню схожість: чим більший добуток, тим ближчі об'єкти між собою. Задача полягає у знаходженні еталону, який має найбільший коефіцієнт кореляції із зображенням.

6. Метод опорного словника

Метод опорного словника використовує каталог, у якому спочатку перелічуються слова з мінімальною кількістю літер, а потім кількість літер поступово збільшується. Кожна літера алфавіту має власну вирішальну функцію.

Припустимо, що ми розпізнаємо слово «ЛИСТ» із 4-х літер. Система РО перебирає всі слова з 4-х літер, які є у словнику, в кожне з цих слів підставляє літери «ЛИСТ» і для кожного словарозраховують суму з 4-х вирішальних функцій. Для всіх цих слів чотири літери, які є у словнику у вигляді (АБАТ, ..., ЛИСТ, ..., ЯЩУР), ми отримуємо функції у вигляді:

$$S_1(АБАТ) = g_A(X_L) + g_B(X_H) + g_A(X_C) + g_T(X_T);$$

$$S_1(ЛИСТ) = g_L(X_L) + g_H(X_H) + g_C(X_C) + g_T(X_T);$$

$$S_1(ЯЩУР) = g_Y(X_L) + g_{Щ}(X_H) + g_Y(X_C) + g_P(X_T),$$

де g_A, g_B, g_A, g_T – розпізнавальні функції для літер АБАТ;...; g_L, g_H, g_C, g_T – розпізнавальні функції для літер ЛИСТ;...; $g_Y, g_{Щ}, g_Y, g_P$ – розпізнавальні функції для літер ЯЩУР.

В результаті виділяється те слово у словнику, для якого отримана сума $S_1(ЛИСТ)$ має найбільше значення.

Недоліком розглянутого алгоритму є швидке зростання об'єму розрахунків із зростанням кількості літер та слів. В результаті словник збільшується й зростає час класифікації об'єкту.

7. Використання магнітних чорнил

Можуть використовуватись також магнітні чорнила. Напис намагнічується постійним струмом, а читання виконується за допомогою магнітних голівок, які використовуються у техніці магнітного запису інформації.

Забруднення, помітки олівцем не псують роботу. Сам текст може читатись людиною [18].

Кожний символ також може «малюватись» за допомогою кількох вертикальних ліній з проміжками між ними змінної ширини (рис. 3.1.3). Зчитується запис за допомогою світлодіодів або магнітних голівок(можливе використання й світлодіодів): отримується код з окремих імпульсів із змінними інтервалами між ними. Магнітні чорнила дозволяють робити якісну класифікацію при поганій якості паперу та друку.

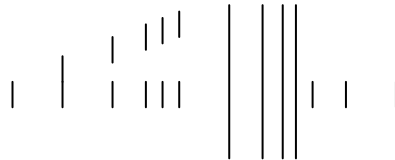


Рис. 3.1.3. Малювання символів вертикальними лініями

8. Метод еталонів, що «дробляться»

Для кожного з образів визначається сфера *мінімального радіусу*, або мінімальна прямокутна комірка, яка охоплює всі координати одного класу навчальної послідовності (рис. 3.1.4).

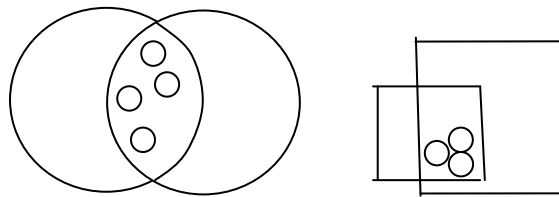


Рис. 3.1.4. Еталони, що «дробляться»

Після цього досліджуються пересічення отриманих сфер між собою. Ці ділянки пересічення великих сфер (великих прямокутників) знову заповнюються меншими сферами чи прямокутниками, які охоплюють об'єкти одного класу навчальної послідовності в рамках указаної ділянки. Такий підхід дозволяє у спрощеній формі отримати умови визначення ділянок ознак для визначення класів об'єктів.

9. Маркування об'єктів

Метод кодування (маркування) символу представлений на рис. 3.1.5: у цьому випадку класифікують не символ, а код. Код нагадує азбуку Морзе. Він наводиться окремо, і по ньому визначається образ.

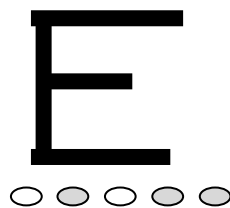


Рис. 3.1.5. Маркування символу

Внаслідок простоти та зручності подібне кодування широко використовується для визначення товарів та виробів. При цьому код може мати вигляд *спрощеного малюнка, штихів, літер та цифр*.

Штрих-код (рис. 3.1.6) кодує вертикальними лініями символи та літери і розшифровується сканером на відстані до 6–20 м. Штрих-код не призначений для декодування людиною. *Лінійні штрих-коди* (рис. 3.1.6, а) читаються водному напрямку (по горизонталі) і є найбільш розповсюдженими. Вони також зветься *полосковими кодами*.

Штрих-коди (в тому числі – кольорові) збільшують швидкість обслуговування клієнтів у торгівлі в кілька разів; прискорюють на складі процес ідентифікації товарів при його прийомі, здачі, комплектації; дозволяють використовувати автоматизований облік об'єктів при їх купівлі, прийомі, здачі, зберіганні; дозволяють паралельно обробляти інформацію по базі даних магазину і передавати інформацію в комп'ютер. На штрих-коди існують *багато стандартів* різних держав. Багатопроменеві сканери пом'якшують загальний недолік штрих-кодів – жорстке позиціонування сканера.



Рис. 3.1.6. Приклади штрих-кодів: а) лінійний; б) матричні двомірні (у тому числі кольорові)

Двомірні штрих-коди (рис. 3.1.6, б) охоплюють розшифровку у двох-вимірах (по горизонталі і по вертикалі) і призначені для кодування великого обсягу інформації. Двомірні коди діляться на *багаторівневі* (являють собою кілька рядків звичайних лінійних кодів) і *матричні*. Матричні коди більш щільно упаковують інформаційну.

Сканери штрих – коду розшифровують штрих-коди. Їх поділяють за типом виконання (стаціонарні, ручні, комбіновані) і за типом підсвічування (сканери штрих-кодів, які не потребують підсвічування; світлодіодні; лазерні).

Голограми використовують для захисту:

1. *Цінних паперів та документів* (як захисні елементи державних в'їзних віз, паспортів, сертифікатів, пластикових платіжних засобів, посвідчень, дипломів, проїзних документів тощо).

2. *Товарів* (як захист від підробки або як голографічні пломби при пакуванні спиртних напоїв, косметики, чаю, кави та цілої низки промислових товарів).

Найбільш поширені два види голограм: у вигляді фольги для гарячого тиснення та липкої голографічної наклейки.

В голограмі разом з описаними оптичними рівнями захисту можуть бути приховані або закодовані зображення, а також елементи магнітного,

люмінесцентного, термохромного, термореактивного та інших видів захисту. Підробка голограм може бути виявлена у процесі експертного дослідження.

10. Розкладання образу на двовимірній площині у ряд Фурь'є дозволяє класифікувати об'єкт по *амплітудам гармонік та фазовим зсувам між ними*. Наприклад, літак має деяку нерівномірно розподілену площу вздовж його довжини. Цю площу розкладають в ряд Фурь'є. Тип літака визначають по базі даних з розкладами силуетів літаків в ряд Фурь'є з урахуванням відстані, висоти та напрямку руху.

Крім рядів Фурь'є використовують і інші ряди. Наприклад, у порівнянні з розкладом в ряд Фурь'є більш ефективним є *дискретне косінусне перетворення*, у якого із зростанням частоти коефіцієнти зменшуються швидше, ніж при перетворенні Фурь'є.

3.2. Перетворення ознак образів у коди

Образи, які класифікує ЕОМ можуть бути дуже різними: літери, музикальні звуки, фотокартка, людина, фінансова діяльність і т. д. Ознаки цих об'єктів теж суттєво відрізняються. Дослідників цікавлять не частні методи кодування, якими б оригінальними вони не були б, а найбільш узагальнені, які дозволяють виконати розпізнавання найбільшої кількості об'єктів.

Системою кодування зветься сукупність правил кодового позначення об'єктів. У обчислювальній техніці використовується двоїсте кодування, яке застосовує лише два значення («1» та «0»). Ці знаки представляються одним бітом, який може прийняти значення «1» та «0». Кодування повинне забезпечити надійне декодування та захист інформації. Захист звичайно відбувається введенням додаткової інформації у коди.

ЕОМ працюють з *цифрами, або з символами*. Тому всі ознаки образів повинні вводитись у ЕОМ у вигляді кодів – *деякої комбінації символів, якою може оперувати машина*.

Загального способу кодування ознак немає, бо що добре для зорового зображення не може бути використане для розпізнавання музики. Під кодом ознаки образу розуміють не дещо стабільне, незмінне, як, наприклад, азбука Морзе, а сукупність не завжди чітко визначених ознак-цифр (символів), яка дозволяє класифікувати об'єкт. Ця сукупність ознак для одного і того ж об'єкту може бути нечіткою, розмитою, іноді суперечливою, неповною, замаскованою.

1. Бінарне кодування наявності ознаки (якісної чи кількісної) відбувається заміною ознаки кодом «1» (є ознака) або «-1» (немає ознаки). В результаті вектор ознак об'єкта складається з елементів «1» та «-1». Можливе використання елементів «1» та «0». Бінарне кодування часто використовується як *детектор відхилення від норми*, наприклад, якщо у спортсмена нормальна температура та тиск крові, ці ознаки отримують у сукупності значення «1», а в іншому випадку – «0».

2. **Кодування зорових образів** (рис. 3.2.1.), за яким зображення проєктується на екран, що розбитий на нумеровані комірки. Кожна комірка розглядається як *рецептор*, який у збудженому стані має на виході сигнал «1», у незбудженому стані – «0». Такий спосіб кодування можна виконати, наприклад, за допомогою фотоелементу у кожній комірці: якщо фотодіод освітлений, то на його виході з'являється сигнал «1», а якщо не освітлений – «0». Такий спосіб обробки дуже зручний у випадку розпізнавання зорових образів.

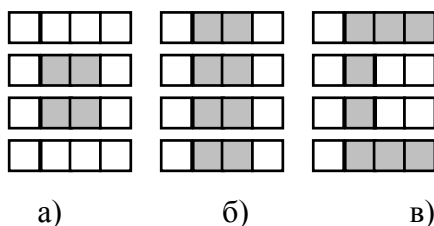


Рис. 3.2.1. Кодування зображень

Тоді згідно рис. 3.2.1 отримуємо коди:

Рис. 3.2.1, а:	Рис. 3.2.1, б:	Рис. 3.2.1, в:
0000	0110	0111
0110	0110	0100
0110	0110	0100
0000	0110	0111

Звичайно ці цифри розміщуються у один ряд, і тоді вони мають вигляд:

Рис. 3.2.1, а: 0000 0110 0110 0000,

Рис. 3.2.1, б: 0110 0110 0110 0110,

Рис. 3.2.1, в: 0111 0100 0100 0111.

Якщо сигнали не дискретні, а безперервні, то реакцією кожного рецептора може бути величина у межах $0 \dots 1$. В цьому випадку система РО буде реагувати на напівтони.

Кольорове зображення в кольоровій системі RGB (R – red; G – green; B – blue) представляється трьома матрицями (одна матриця під кожний окремий колір) зі зміною яскравості кожного пікселя в межах 256 градацій.

Грануляція (заміна універсуму на середі величини виділених діапазонів числових значень, або на окремі позначення діапазонів) та *фрактали* (заміна одного пікселя на їх сукупність з визначенням їх розміщення за формулами) дозволяють стиснути інформацію і використати її у динамічних системах зі зміною зображення у часі.

3. **Кодування граничних значень.** Для безперервних сигналів часто використовується квантування у тому числі типу «більше – менше за деяку границю»: «ЯКЩО $x_j \geq 0$ (тобто «границя» дорівнює 0), ТО подається вихідний сигнал».

Якщо вхідний сигнал $x_j = 0 \dots 1000$ потрібно розділити на ряд дискретних значень 0, 1, 2, 3 з рівномірним проміжком між ними, то граничні значення сигналу розраховуються у вигляді:

$$x_j^0 = 1000 \cdot 0/3 = 0; \quad x_j^1 = 1000 \cdot 1/3 = 333; \\ x_j^2 = 1000 \cdot 2/3 = 666; \quad x_j^3 = 1000 \cdot 3/3 = 1000.$$

4. **Кодування по заданих рівнях.** Припустимо, щовхідний сигнал $x_j = 0 \dots 750$ потрібно квантувати у межах 5 рівнів ($K=5$, якому відповідає ряд із 5-ти розрядів: «XXXXX») з рівномірним проміжком між ними і з використанням двоїстих сигналів 0 та 1. Тоді вхідний сигнал $x_j = 0 \dots 750$ ділимо на $K=5$ часток $750:5 = 150$. Відповідні сигнали мають вигляд:

ЯКЩО ($0 \leq x_j^1 < 150$) ТО $x_j^1 = 10000$; ЯКЩО ($150 \leq x_j^1 < 300$) ТО $x_j^2 = 11000$;
ЯКЩО ($300 \leq x_j^1 < 450$) ТО $x_j^3 = 11100$; ЯКЩО ($450 \leq x_j^1 < 600$) ТО $x_j^4 = 11110$.

5. **Кодування цілих дійсних чисел.** Цілі числа ділять навпіл до тих пір, доки у залишку не залишиться нуль чи одиниця. Сукупність залишків від кожного ділення, записана при переміщенні ліворуч, створює двоїстий аналог десятинного числа. Приклад для числа «6»: ($6:2=3$, залишок «0»); ($3:2=1$, залишок «1»); ($1:2=0$, залишок «1»). При переміщенні ліворуч отримуємо код 110. Для кодування цілих чисел від 0 до 255 потрібно мати 8 розрядів двоїстого коду (8 бітів).

Для кодування дійсних чисел використовують більш складне 80-розрядне кодування з використанням більшої частки з 80 біт під мантису і меншої частки під характеристику.

6. **Кодування текстової інформації** відбувається призначенням кожному символу алфавіту деякого цілого числа у двоїстій формі. Для цього при кодуванні тексту створюємо алфавіт і розміщуємо його символи у порядку зменшення їх ймовірностей $p_i = w_i / w_0$, де w_0 – загальна кількість символів в тексті; w_i – кількість i -го символу в тексті; $i = 1, \dots, N$ – порядковий номер символу в тексті. Якщо текст файлу складається з 51 символу (включаючи літери, цифри, пробіли тощо), то ми можемо застосувати 6 бітів (бо $2^6=64$)).

Початкова частка бітового коду складається з нулів і тому не застосовується. В результаті інформація стискається приблизно вдвічі. Але у дійсності інформація стискається більше ніж вдвічі, якщо урахувати те, що найбільш скорочені коди надаються символам, які мають найбільшу ймовірність зустрічі в тексті.

Кодування повинне задовольняти наступним вимогам: використовуються коди змінної довжини (бо нулі перед цифрою відкидаються); найкоротші коди призначають символам, які зустрічаються частіше; коди повинні задовольняти умовам префіксу (жодний код з меншою ймовірністю символу не повинен мати в початку послідовності попереднього коду); потрібно мати алгоритм, який завжди породжує множину найкоротших кодів; надійність у розпізнаванні забезпечує додаткова інформація, яка супроводжує основний код.

7. **Кодування графічних даних.** Зображення складається з маленьких точок (пікселів), які складають візерунок, що зветься растром. Загально прийнятим є представлення чорно-білих ілюстрацій у вигляді комбінацій точок з 256 градаціями сірого кольору (застосовується 8-розрядне двоїсте число). До кольорових зображень застосовують принцип декомпозиції до-вільного кольору на основні складові (червоний, зелений, голубий кольори) з відповідною градацією яскравості кожної складової аналогічно градаціям сірого кольору.

8. **Кодування звукової інформації.** Використовують розклад аналого-вого звукового сигналу на ряд послідовних дискретних значень незмінної амплітуди або на ряд гармонік, які кодуються. Числові коди можуть вказувати тип інструмента, висоту тону, час та інтенсивність звучання та інші фактори.

9. **Кодування знань** з інформацією довільного об'єму у вигляді сукупності тексту, формул, таблиць, рисунків, графіків, фотографій. Інформація розділяється на довільні частки, які отримують *цифрові позначення (коди)* експерта, які можуть відноситись також до типу знань у виділеній частці [46]. В деякі частки інформації *експерт навмисно вводить помилки*. Відповіддю учня має вигляд *підсумку цифр кодів* експерта, у яких у відповідних частках інформації, на думку учня, експертом навмисно введені помилки. Ця відповідь порівнюється з еталонною цифровою відповіддю експерта. Якщо вся введена експертом інформація є вірною, то вірною відповіддю є цифра «0». Якщо інформація поділена на 5 часток, то згідно подібного підходу, вся вона може бути вірною, або в одній її частці можуть бути помилкові відомості.

3.3. Перетворення аналогового сигналу у код

Часто у системах РО отримується аналоговий сигнал u_C (рис. 3.3.1), який потрібно переробити у коди (цифрову форму) для подальшої обробки за допомогою ЕОМ.

З цією метою створюється трикутний опорний сигнал u_0 з амплітудою, яка більше за амплітуду сигналу u_C . Додатковий аналоговий сигнал з незмінною амплітудою u_K починається у точці трикутника зубчастого опорного сигналу u_0 , де напруга $u_0=0$ і закінчується точкою рівності напруг $u_C = u_0$. Таким чином, величина аналогового сигналу u_C перетворюється на час, який визначає ширину імпульсу u_K .

Імпульс напруги u_K вмикає вихід високочастотного генератора імпульсів на лічильник імпульсів N_U (вони не показані). Зафіксована лічильником імпульсів N_U кількість імпульсів високочастотного генератора імпульсів і є поточною величиною сигналу u_C .

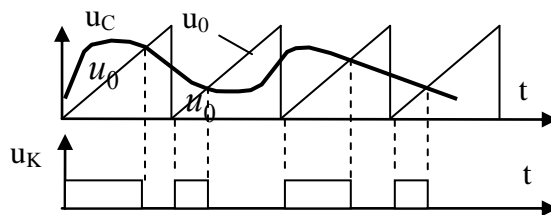


Рис. 3.3.1. Перетворення ординати у час

Завдання

1. Визначити коди за *методом зондів* для 4-х перших великих друкованих літер власного прізвища.

3. Визначити коди за *квазітопологічним методом* для 4-х перших великих друкованих літер власного прізвища.

3. За *методом центральних моментів* для розпізнавання 2-х літаків, розміри яких розрізняються приблизно в 2 рази, на площині ознак $x_1, x_2 = 0 \dots N$, де N – порядковий номер студента у групі. Використати для порівняння такі моменти:

$$M_1 = M_{20} + M_{02}; M_2 = (M_{20} + M_{02})^2 + 4M_{11}; M_3 = (M_{30} - 3M_{12})^2 + (3M_{21} - M_{03})^2.$$

3.4. Відстань Евклідова, Манхеттена, Чебишева, Махаланобіса, Хеммінга. Близькість поХеммінгу та за скалярним добутком

Нижче наведені найбільш уживані міри відстані та близькості між двома об'єктами.

1. *Відстані між об'єктами* у вигляді двох векторів з числовими ознаками описується:

– як *квадрат евклідової відстані*, що дорівнює сумі квадратів різниць числових координат векторів (використовується найчастіше),

$$d_1^2(\omega_k^K; \omega_e^E) = \sum_{j=1}^n (x_{kj}^K - x_{ej}^E)^2,$$

або як *евклідова відстань*, що дорівнює кореню квадратному від суми квадратів різниць числових координат векторів

$$d_2(\omega_k^K; \omega_e^E) = \sqrt{\sum_{j=1}^n (x_{kj}^K - x_{ej}^E)^2},$$

де $k = 1, 2, \dots, K$ – порядкові номери векторів класу K ; $e = 1, 2, \dots,$

E – порядкові номери векторів класу E ; $j = 1, 2, \dots, n$ – порядкові номери ознак векторів обох класів; x_{kj}^K – j -ий елемент k -го порівнюваного вектора ω_k^K класу K ; x_{ej}^E – j -ий елемент e -го порівнюваного вектора ω_e^E класу E .

– як *відстань по Манхеттену*, що дорівнює сумі абсолютних значень різниць числових координат

$$d_3(\omega_k^K; \omega_e^E) = \sum_{j=1}^n |x_{kj}^K - x_{ej}^E|,$$

– як *відстань Чебишева*, що дорівнює абсолютному значенню максимальної різниці по одній j -ої координаті

$$d_4(\omega_k^K; \omega_e^E) = \max_j |x_{kj}^K - x_{ej}^E|$$

– як *відстань Махаланобіса*, що є мірою відстані між двома точками в просторі, при наявності кореляції між змінними. Наприклад, для двох чи більше некоррелірованих змінних відстань Махаланобіса між точками дорівнює *відстані Евкліда*. Але у випадку, **коли змінні корельовані**, осі на графіку можуть розглядатися як *не ортогональні* (осі вже не спрямовані під прямими кутами одна по відношенню до іншої). У цьому випадку просте визначення відстані Евкліда не підходить, тоді як відстань Махаланобіса є адекватно визначеною.

– як *відстань Хеммінга*, що для двох векторів однакової довжини дорівнює числу позицій з різними елементами векторів, які можуть мати лінгвістичне або числове значення. Це відображується умовною формулою

$$d(x_j^{A1}; x_j^{A2}) = \sum_{k=1}^K |x_k^{A1} - x_k^{A2}|,$$

де $k = 1, 2, \dots, K$ – порядкові номери елементів вектора; x_k^{A1}, x_k^{A2} – елементи першого та парного вектора.

Приклади *відстані Хеммінга* між двома векторами: $d(1101; 1110) = 2$; $d(3,7,1,4; 5,7,1,4) = 1$; $d(\text{нейрон}; \text{таксон}) = 3$. Очевидно, що такий підхід можна застосувати не лише до символічних елементів, але й до елементів вектора у вигляді слів та речень.

Є багато інших формул по розрахунку відстаней між векторами.

3. *Найбільш уживані міри близькості між двома об'єктами:*

– *близькість по Хеммінгу*, що дорівнює числу позицій з однаковими елементами векторів. Близькість між об'єктами визначається тим більшим числом, чим менша відстань між ними. Приклади *близькості за Хеммінгом* між двома векторами:

$d(11011; 11101) = 3$, бо однакові три розряди;

$d(3,7,1,4; 5,7,1,4) = 3$, бо однакові три розряди;

$d(\text{нейрон}; \text{таксон}) = 2$, бо однакові два символи.

Очевидно, що такий підхід можна застосувати не лише до символічних елементів, але й до елементів вектора у вигляді слів та речень.

Близькість за скалярним добутком векторів:

– два вектори V_1 та V_2 однакового n -го порядку з елементами « ± 1 » можуть мати скалярний добуток $S = V_1 \cdot V_2^T$ від мінімального значення

« $S=-n$ » (повне неспівпадіння) до максимального значення « $S=+n$ » (повне співпадіння). Скалярний добуток $S = V_1 \cdot V_2^T$ є мірою близькості між векторами, бо чим більше значення S , тим ближчими є між собою вектори.

– Скалярний добуток $S = V_1 \cdot V_2^T$ може використовуватись як міра близькості між векторами і при складанні їх з елементів «1» та «0»: у цьому випадку інформаційно значущими вважають співпадіння елементів «1», а скалярний добуток змінюється у межах « $S=0 \dots n$ ».

Класифікація об'єктів з числовими координатами по відстані. Для кожної деякої окремої групи об'єктів у вибірці можна визначити середні значення координат. Отримані середні координати характеризують точку, яка називається *центроїдом групи*. Якщо вибірка розглядається у просторі ознак $x_j, j = 1, 2, \dots, n$, то для кожного вхідного невідомого об'єкта можна обчислити відстань Махаланобіса від нього до кожного центроїда групи. Заданий об'єкт належить до тієї групи об'єктів, для якої його відстань Махаланобіса є мінімальною.

Для підвищення надійності розпізнавання образів, всі ознаки (змінні) нормалізують, щоб великі числа не набували більшого впливу на образ. Якщо деякі ознаки мають менший вплив на розпізнавання образів, ніж інші, то вплив цих ознак експерт зменшує додатковим множенням на відповідний зменшувальний коефіцієнт $k_{xj} < 1$.

3.5. Групування об'єктів з цифровими ознаками по середньоквадратичному розкиду

1. Оцінка щільності об'єктів в середині класу Ω^A (рис. 3.5.1, а).

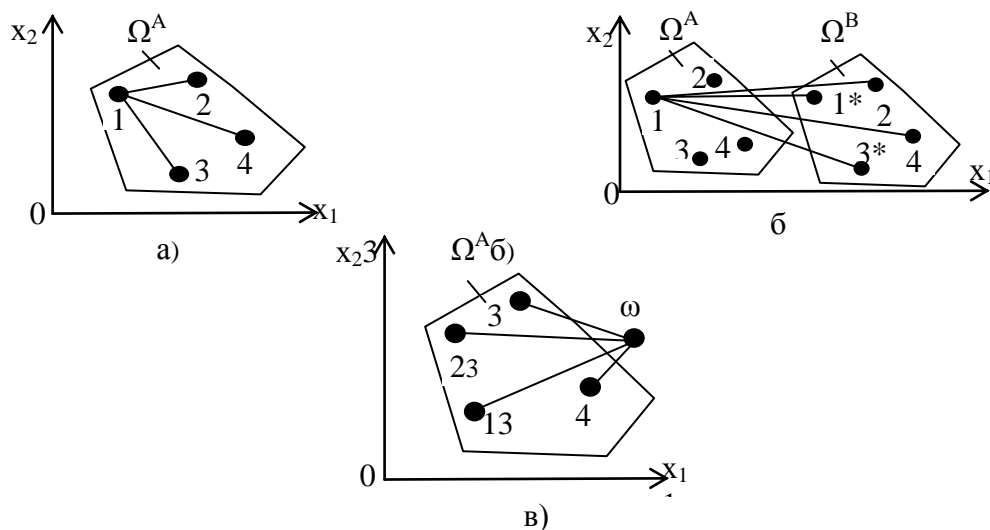


Рис. 3.5.1. Визначення середньоквадратичної відстані:
 а) в середині класу; б) між класами; в) між об'єктом та класом

Для оцінки відстані між об'єктами класу Ω^A по числових ознаках використовується розрахунок середньоквадратичного розкиду між об'єктами у середині цього класу [18]

$$S(\Omega^A) = \sqrt{\frac{2}{M^A(M^A - 1)} \sum_{k=1}^{M^A} \sum_{e=1}^{M^A} \sum_{j=1}^n (x_{kj}^A - x_{ej}^A)^2},$$

де M^A – загальна кількість об'єктів класу Ω^A ; $k, e = 1, 2, \dots, M^A, k \neq e$ – порядкові номери об'єктів класу Ω^A .

Розглянемо деякі пояснення до середньоквадратичного розкиду між об'єктами у середині класу Ω^A . Наведена нижче матриця перебирає всі можливі комбінації відстаней між всіма об'єктами класу. В результаті отримуємо таку нумерацію відстаней (рис. 3.5.1, а):

Для $M^A=3$. Для $M^A=3$. Для $M^A=3$.

$$\begin{bmatrix} 11 & 12 \\ 21 & 22 \end{bmatrix} \cdot \begin{bmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 31 & 32 & 33 \end{bmatrix} \cdot \begin{bmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & 44 \end{bmatrix}.$$

Розгляд наведених матриць (індексів відстаней) указує:

1. Діагональні елементи «11», «22», «33», «44» дорівнюють нулю і можуть не враховуватись.

2. Отримані матриці є симетричними, бо, наприклад, відстані «12»=«21». Тому можна враховувати відстані лише під або над діагоналлю, що скорочує розрахунки у 2 рази.

3. Наявність в формулі $S(\Omega^A)$ середньоквадратичного розкиду між об'єктами цифри 2 не змінює висновки, бо отримані варіанти розрахунків відрізняються один від другого лише через постійний множник.

2. Оцінка відстані між об'єктами двох класів (Ω^A та Ω^B) (рис. 3.5.1, б). Оцінка використовується для об'єднання або роз'єднання об'єктів. В якості міри близькості між даною парою класів Ω^A та Ω^B використовують середньоквадратичний розкид між класами

$$R(\Omega^A, \Omega^B) = \sqrt{\frac{2}{M^A M^B} \sum_{k=1}^{M^A} \sum_{e=1}^{M^B} \sum_{j=1}^n (x_{kj}^A - x_{ej}^B)^2},$$

де M^A, M^B – загальна кількість об'єктів у класах Ω^A та Ω^B ; $k = 1, 2, \dots, K$ – порядкові номери векторів класу A ; $e = 1, 2, \dots, E$ – порядкові номери векторів класу B ; $j = 1, 2, \dots, n$ – порядкові номери ознак векторів обох класів.

3. Оцінка відстані між об'єктом ω та об'єктами одного класу Ω^A (рис. 3.5.1, в). Оцінка використовується для віднесення об'єкта ω до того з кількох класів, до якого він має найменшу відстань. Середньоквадратична відстань між об'єктом ω та класом Ω^A розраховується по формулі

$$L(\omega, \Omega^A) = \sqrt{\frac{1}{M^A} \sum_{k=1}^{M^A} \sum_{j=1}^n (x_j^\omega - x_{kj}^A)^2}.$$

Завдання

У площині 1-го квадранту з осей $X1, X2$ (рис. 3.5.2), на яких позначаються вимірні ознаки (при максимальних значеннях $X1_{max} = N, X2_{max} = N$, де N – порядковий номер студента у групі), виділити дві довільні ділянки (класи), одна з яких по площі приблизно в три рази менша за іншу. У кожній з цих ділянок позначити по чотири довільні точки.

Приблизно посередині між цими ділянками (класами) помітити додаткову точку-об'єкт. Координати всіх позначених точок вважати цифровими ознаками об'єктів-точок, отриманими від датчиків системи РО.

Визначити: евклідову відстань між об'єктами у середині кожного класу; середньоквадратичний розкид між об'єктами класів; середньоквадратичний розкид між класами; до якого класу належить додаткова точка.

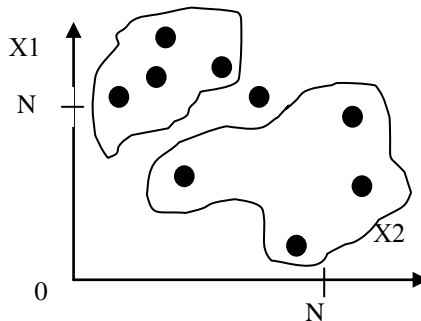


Рис. 3.5.2. Розміщення об'єктів у класах

3.6. Напрямки групування об'єктів по аналізу даних в магазині

1. Метод еталонних векторів груп об'єктів (товарних сегментів) є багатовимірним математико-статистичним методом групування об'єктів, розвинений в роботах вчених України Н. М. Шмиголя, Л. А. Белозерського та ін. [10; 122].

Нижче наводиться розрахунок відстані між векторами *двоїстого кодування якісних показників* (наявність ознаки у об'єкта позначається 1, відсутність – 0) по наведеному в роботі Н. М. Шмиголя прикладу [122].

1. *Визначення вектора конкретного об'єкта по кодах якісних еталонних ознак сегментів товарів.* Експерт розділяє універсум товарів магазину на еталонні сегменти (наприклад: «повсякденного попиту T_{601} », «попереднього вибору T_{602} » та «особливого попиту T_{603} »), обирає для них сукупність однакових якісних «характеристик/ознак» і призначає кожному елементу базових (еталонних) векторів коди «1» (наявність ознаки) та «0» (відсутність ознаки) – табл. 3.6.1.

Після визначення елементів еталону, експерт визначає значення кодіваних ознак для конкретного товару $T_{6i}, i=1, 2, \dots, m$ – порядковий номер това-

ру. У даному випадку конкретний товар T_{ei} – це «морозиво у вафельному стаканчику», а його коди наведені у колонці з сірим фоном (табл. 3.6.1).

Таблиця 3.6.1.

Значення кодів якісних ознак елементів еталонних векторів сегментів товарів магазину та вектора конкретного об'єкта («морозива у вафельному стаканчику»)

Якісні характеристики, ознаки	Базові еталонні вектори сегментів товарів з якісними ознаками			«Морозиво у вафельному стаканчику» T_{ei}
	1. «Товари повсякденного попиту T_{e01} »	3. «Товари попереднього вибору T_{e02} »	3. «Товари особливого попиту T_{e03} »	
1. Продукція стандартизована	1	0	0	1
3. Продукція диференційована	0	1	0	0
3. Продукція має унікальні характеристики	0	0	1	0
3. Товар купується споживачем часто	1	0	0	1
5. Процес придбання обмежений мінімальними зусиллями при виборі	1	0	0	1
6. Процес придбання триває деякий час, споживач порівнює різні товари за якістю, ціною, оформленням	0	1	0	1
7. Щоб придбати товар, споживачі готові докласти додаткових зусиль	0	0	1	0
8. Місце розташування торгівельної точки не має для споживача значення	0	0,5	1	0

Після цього експерт розраховує коефіцієнти степені близькості (c_1, c_2, c_3, c_4) вектора морозива T_{ei} до еталонного вектора сегменту «товарів повсякденного попиту T_{e01} ».

Нижче наведений приклад такого порівняння.

Подібність товарів по наявності ознаки «1»: коефіцієнт урахування кількості однакових ознак «1» у еталонних товарах T_{e0j} та у морозива T_{ei}

$$c_1 = \sum_{k=1}^K (m_{e_{0jk}} \cdot m_{e_{ik}}) = 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 + 1 \cdot 1 + 0 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 = 3, \quad (3.6.1)$$

де $k = 1, 2, \dots, K$ – порядковий номер якісної ознаки сегмента товарів; $m_{e_{0jk}}, m_{e_{ik}}$ – кодовані двійкові компоненти векторів, що описують якісні характеристики товарів еталону T_{e0ji} та морозива T_{ei} .

– *Подібність товарів по наявності ознаки «0»*: коефіцієнт урахування кількості однакових ознак «0» у товарів $T_{\theta 0j}$ та $T_{\theta i}$

$$c_2 = \sum_{k=1}^K (1 - m_{\theta 0jk}) (1 - m_{\theta ik}) = 0 \cdot 0 + 1 \cdot 1 + 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 0 + 1 \cdot 1 + 1 \cdot 1 = 3 \quad (3.6.2).$$

– *Різниця між товарами наявності ознак «1» та «0»*: коефіцієнти урахування кількості ознак «1» у товарів $T_{\theta 0j}$ при наявності ознаки «0» для товарів $T_{\theta i}$ та урахування кількості ознак «0» у товарів $T_{\theta 0j}$ при наявності ознаки «1» для товарів $T_{\theta i}$:

$$c_3 = \sum_{k=1}^K m_{\theta 0jk} (1 - m_{\theta ik}) = 1 \cdot 0 + 0 \cdot 1 + 0 \cdot 1 + 1 \cdot 0 + 1 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 0 \cdot 1 = 0,$$

$$c_4 = \sum_{k=1}^K (1 - m_{\theta 0jk}) m_{\theta ik} = 0 \cdot 1 + 1 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + 0 \cdot 1 + 1 \cdot 1 + 1 \cdot 0 + 1 \cdot 0 = 1.$$

(3.6.3)

Далі визначається числове значення міри близькості між товаром та кожним з базових еталонних векторів ознак сегментів товарів $T_{\theta 0j}$ у вигляді *функції подібності*.

При $c_1=3$, $c_2=4$, $c_3=0$, $c_4=1$, порівняння морозива $T_{\theta i}$ з еталонними векторами товарів $T_{\theta 0j}$ виконується по *функціям подібності* ΦP_e , де $e = 1, 2, \dots, 7$ – порядковий номер функції подібності:

1. Функція Расела $\Phi P_1(T_{\theta i}, T_{\theta 0j}) = \frac{c_1}{c_1 + c_2 + c_3 + c_4} = \frac{c_1}{n} = 3: (3+4+0+1) = 0,38$;
(тут $n = c_1 + c_2 + c_3 + c_4 = 3+4+0+1 = 8$).

2. Функція Жокара та Нідмена $\Phi P_2(T_{\theta i}, T_{\theta 0j}) = \frac{c_1}{n - c_2} = 3: (8-4) = 0,75$.

3. Функція Дайса $\Phi P_3(T_{\theta i}, T_{\theta 0j}) = \frac{c_1}{2c_1 + c_3 + c_4} = 3: (2 \cdot 3 + 0 + 1) = 0,43$.

4. Функція Сніфа $\Phi P_4(T_{\theta i}, T_{\theta 0j}) = \frac{c_1}{c_1 + 2(c_3 + c_4)} = 3: [3 + 2(0 + 1)] = 0,60$.

5. Функція Сокаля та Мішнера $\Phi P_5(T_{\theta i}, T_{\theta 0j}) = \frac{c_1 + c_2}{n} = (3+4): 8 = 0,88$.

6. Функція Кульжинського $\Phi P_6(T_{\theta i}, T_{\theta 0j}) = \frac{c_1}{c_3 + c_4} = 3: (0+1) = 3,00$.

7. Функція Юла $\Phi P_7(T_{\theta i}, T_{\theta 0j}) = \frac{c_1 c_2 - c_3 c_4}{c_1 c_2 + c_3 c_4} = (3 \cdot 4 - 0 \cdot 1): (3 \cdot 4 + 0 \cdot 1) = 1,00$.

На практиці рекомендується застосовувати *всі наведені вище функції подібності*. Остаточний висновок про належність досліджуваного об'єкта до еталону товарів робиться на підставі *сукупної більшості тверджень* всіх функцій подібності ΦP_e [122].

За аналогічними розрахунками перевіряється належність «морозива $T_{\theta i}$ » до товарів «попереднього вибору $T_{\theta 02}$ » та «особливого попиту $T_{\theta 03}$ ». Зведені результати розрахунків наведено в табл. 3.6.2, з якої видно, що найбільше значення функції подібності ΦP_e (всі без винятку) мають для

випадку товарів «повсякденного попиту $T_{\epsilon 01}$ ». Це означає, що досліджувана товарна група (морозиво) за своїми характеристиками ліпше відповідає ознакам товару «повсякденного попиту».

Таблиця 3.6.2.

Порівняння значень функцій подібності морозива для магазину

Функція подібності	Товари «повсякденного попиту $T_{\epsilon 01}$ »	Товари «попереднього вибору $T_{\epsilon 02}$ »	Товари «особливого попиту $T_{\epsilon 03}$ »
Расела $\Phi\Pi_1$	0,38	0,10	0,00
Жокара та Нідмена $\Phi\Pi_2$	0,75	0,20	0,00
Дайса $\Phi\Pi_3$	0,43	0,20	0,00
Сніфа $\Phi\Pi_4$	0,60	0,10	0,00
Сокаля та Мішнера $\Phi\Pi_5$	0,88	0,40	0,10
Кульжинського $\Phi\Pi_6$	3,00	0,30	0,00
Юла $\Phi\Pi_7$	1,00	0,00	-1,00

Аналогічні розрахунки виконуються для всіх товарів $T_{\epsilon i}$ магазину. В результаті всі товари магазину $T_{\epsilon i}$ розділяються по товарних сегментах $T_{\epsilon 01}$, $T_{\epsilon 02}$, $T_{\epsilon 03}$. Подібна класифікація полегшує подальший аналіз виробничих процесів.

Але на цьому не зупиняються. Весь універсум товарної маси магазину поділяють на нові товарні сегменти з новими значеннями якісних ознак.

При подальшому дослідженні можна розглядати зміну у часі структури товарів; розробляти *стратегії розвитку*, спрямовані на максимізацію прибутків за рахунок виділення найбільш перспективних товарних сегментів.

2. Обробка кошику покупця [15; 129].

Асоціативні розрахунки виконуються по вмісту ринкового кошику покупця (під «кошиком» розуміють транзакцію, платіж, покупку), по якому визначається: типовий набір і розміщення товарів в магазині; прийняття рішення по проведенню рекламної компанії; регулювання запасів тощо. При знаходженні послідовностей ураховується час, що дозволяє працювати із серією подій для знаходження послідовних асоціацій протягом деякого періоду часу. Деякі алгоритми пошуку асоціацій перед аналізом спочатку *сортують дані*.

Якщо $E = \{ e_1, e_2, \dots, e_j, \dots, e_n \}$ – це загальна сукупність товарів магазину, то кошик покупця можна розглядати як вибірку $C_p = \{ e_j \mid e_j \in E; j = 1, \dots, n \}$, $p = 1, 2, \dots, P$, P – загальна кількість кошиків.

Асоціація «Поширеність» появи кожного окремого товару e_j або групи товарів (вимірюється у відсотках), є частотою їх появи і визначається діленням кількості появи цього товару у всіх подіях (покупках) на загальну кількість подій. Асоціація вважається не істотною, якщо значення поширеності дорівнюють *менше однієї тисячної відсотка*.

Асоціація «Послідовність» «Довірчість А до Б» (Взаємозв'язок А і Б) показує як часто з появою А з'являється Б и розраховується як відношення частоти появи (поширеності) А і Б разом до поширеності А. Якщо довірчість А до Б дорівнює 20 %, то це означає, що при покупці товару А в кожному п'ятому випадку придбають і товар Б. *Послідовність* також може мати вигляд правила $A \rightarrow B$ з окремим коефіцієнтом підтримки правила).

Якщо поширеність А не дорівнює поширеності Б, то і довірчість А до Б не дорівнює довірчості Б к А: покупка комп'ютера частіше веде до покупки дискет, ніж покупка дискети до покупки комп'ютера.

Асоціація «Достовірність правила $A \rightarrow B$ » є співвідношення транзакцій, які містять А та Б і транзакцій, до яких входить вибірка А.

Потужність асоціації розраховується по формулі: «довірчість А до Б»/«поширеність Б». Чим більше потужність, тим сильніше вплив, який поява А робить на появу Б.

Створення баз даних по поширеності, довірчості і потужності асоціацій дозволяє звертатись до них при запиті. Наприклад: «Знайти всі асоціації, в яких для товару Х довірчість більша за 50 % і поширеність не менша за 2,5 %».

3. Аналіз часових рядів [13] ураховує події по часових рядах у минулому («тиждень – місяць – квартал – рік»), сезонність, свята й ін. *Існують дві основні цілі* аналізу часових рядів: 1) визначення природи ряду і 2) прогнозування-передбачення майбутніх значень часового ряду.

Найпоширеніші методи аналізу часових рядів:

– *спектральний аналіз* (дозволяє знаходити періодичні складові часового ряду);

– *кореляційний аналіз* [дозволяє знаходити суттєві періодичні залежності і відповідні їм затримки (лаги) як всередині одного ряду (автокореляція), так і між кількома рядами (кроскореляція)];

– *моделі авторегресії і ковзного середнього* орієнтовані на опис процесів; виявляють однорідні коливання, порушувані випадковими впливами; дозволяють передбачати майбутні значення ряду тощо;

– *метод рухомого (ковзного) середнього*.

При аналізі часових рядів проходять такі етапи: графічне зображення; виділення закономірних складових часового ряду (тренда, сезонних і циклічних складових); вилучення низько або високочастотних складових процесу; підбір математичної моделі і перевірка її адекватності; прогнозування даних.

Метод рухомого (ковзного) середнього

Припустимо, що нам відома щоденна кількість проданого товару F_i , де $i = 1, 2, \dots, 7$ – порядковий номер дня у тижні. Залежність F_i , $i = 1, 2, \dots, 7$, наведена в табл. 3.6.3 та на графіку рис. 3.6.1. На основі цих даних *метод рухомого (ковзного) середнього* дозволяє визначити прогнозне числове значення кількості проданого товару на наступний $(n+1)$ -й день за допомогою формули

$$F_{C,j=(n+1)} = \frac{1}{n} \sum_{j=1}^n F_j,$$

де $j= 1, 2, \dots, n$ – порядковий номер n попередніх вимірів числового значення F_j ; $j= n + 1$ – порядковий номер розрахованого прогнозного числового значення *простого ковзного середнього* $F_{C,j=(n+1)}$ на майбутній $(n + 1)$ –й день; n – період КС (звичайно $n = 3$) – кількість чисел, що входять у розрахунок ковзного середнього; F_j – контрольований цифровий показник у часі j -го порядкового номеру (кількість проданого товару, ціна однієї акції тощо).

Таблиця 3.6.3.

Щоденна кількість проданого товару F_i за тиждень

j	1, Пн	2, Вт	3, Ср	4, Чт	5, Пт	6, Сб	7, Нд	8, -
F_j	5	4	14	10	19	14	16	-
F_C	-	-	-	$F_{C4} = 7,67$	$F_{C5} = 9,33$	$F_{C6} = 14,3$	$F_{C7} = 13,67$	$F_{C8} = 16,3$

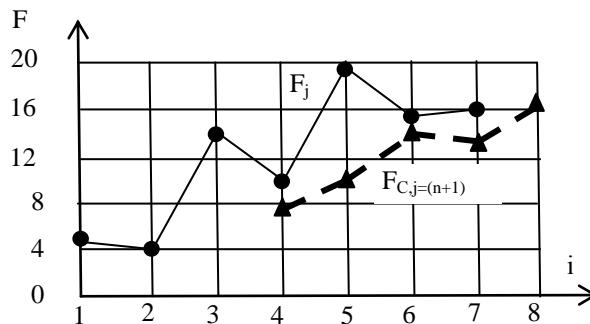


Рис. 3.6.1. Кількість проданого товару F_i у залежності від дня тижня $j=i= 1, 2, \dots, 7$

Приклади розрахунків по даним табл. 3.6.4 ($n=3$):

$$F_{C4}=(5+4+14)/3=7,67; F_{C5}=(4+14+10)/3=9,33; \dots; F_{C8}=(19+14+16)/3=16,3.$$

Ковзне середнє (КС) показує середнє значення часового ряду за певний період часу як тенденцію розвитку процесу і відображують середні ціни або середній об’єм продажу товарів, акцій, сировини і та ін. При цьому дні можуть бути замінені на тижні, місяці, квартали або роки.

В більшості випадків динамічний ряд, окрім тренда і випадкових відхилень від нього, характеризується ще сезонними і циклічними складовими. Циклічні складові відрізняються від сезонних більшою тривалістю і непостійністю амплітуди. Звичайна тривалість сезонної компоненти вимірюється днями, тижнями або місяцями, а циклічної – роками або десятками років.

До ковзного середнього потрібно ставитись з деякою обережністю із-за того, що майбутнє оцінюється по даним відрізків минулого часу, які мають однакову вагу, хоча логічніше було б припустити, що числові значення мають змінний ваговий множник у вигляді $(w_i F_i)$, де вагові коефіцієнти задаються експертом (наприклад: $w_1=1/3, w_2=2/3, w_3=1$). Часовий ряд

відрізняється від звичайної випадкової вибірки тим, що майбутні значення залежать від попередніх, а математичні сподівання та дисперсії можуть залежати від часу. Аналіз часових рядів використовують для короткострокових прогнозів, бо він не ураховує можливе втручання ззовні у процес, і, значить, не може передбачити зміни тенденції.

Методів ковзних середніх існує багато: просте КС; експоненційне КС; зважене КС; трикутне КС; адаптивне КС; синус-зважене КС; КС кінцевої точки та інші, менш поширені.

Виходячи з цілей дослідження, кожний часовий ряд звичайно розглядають як суміш таких компонент: тренд або довгострокова тенденція в розвитку ряду (для аналізу використовують апарат регресійного аналізу і ковзних середніх); сезонна компонента, яка повторюється через певний період (для аналізу використовують спеціальні моделі сезонного згладжування: спектрального аналізу, гармонійні моделі, метод ковзного середнього).

4. Генетичні асоціації. Припустимо, що треба обрати сукупність фіксованого числа параметрів ринку, що впливає на його динаміку (це буде набір імен деяких параметрів). Про цей набір можна казати як про сукупність хромосом, що визначають якість рішення поставленої задачі. Значення параметрів, що визначають рішення, будуть тоді називатися *генами*. Пошук оптимального рішення при цьому схожий на еволюцію популяції індивідів, представлених їхніми наборами хромосом. У цій еволюції діють три механізми: відбір найсильніших *наборів хромосом*; *схрещування* – виробництво нових індивідів за допомогою змішування хромосомних наборів; *мутації* – випадкові зміни генів у деяких індивідів популяції. У результаті зміни поколінь виробляється оптимальне рішення поставленої задачі.

Генетичні алгоритми мають два недоліки:

- постановка задачі в їхніх термінах не дає можливості проаналізувати статистичну значимість одержуваного за їх допомогою рішення;
- ефективно сформулювати задачу, визначити критерій відбору хромосом під силу лише фахівцю.

5. Метод дерев рішень по суті є структурним способом *представлення правил*. У кожному вузлі дерева задається питання, на яке потрібно відповісти «Так» чи «Ні». Корінь дерева – це початкове питання, а наступні ієрархічно нижчі вузли є перевіркою деякої умови. Такі бінарні дерева є найбільш простими, частними випадками дерев рішень (в інших випадках відповідей на питання може бути більше за дві).

На етапі будування дерева переміщуються від кореня до кінцевого вузла дерева («вузла рішення», «листа»).

На етапі використання дерева шлях від кореня до ієрархічно нижчої вершини є набором правил у вигляді «Якщо..., то...» для відповіді на поставлене питання.

Внутрішні вузли дерева є атрибутами бази даних. Кожна гілка від ієрархічно нижчого вузла помічена предикатом розщеплення, який описує унікальний шлях від кореня дерева лише до одного конкретного «листа» – рішення. Якість побудованого дерева залежить від вірності предиката розщеплення.

Дерева рішень надають можливість отримувати правила з бази даних на природній мові. Користувач подає на вхід алгоритму всі відомі йому атрибути, а алгоритм сам обирає найбільш значущі серед них, які і використовуються для будування дерева.

3.7. Метод ієрархічних потоків

При визначенні рівня та об'єму рекламних заходів, різні товари бажано ділити на окремі групи внаслідок дискретності рекламних засобів та втрат прибутків.

Розрахунок розділу коштів на рекламу між трьома товарами Т1 – Т3 з якісними та кількісними ознаками виконаємо за методом ієрархічних потоків (рис. 3.7.1) [53], який є подальшим розвитком методу аналізу ієрархій, описаному в роботі [106; 107], але скорочує розрахунки.



Рис. 3.7.1. Ієрархічні потоки витрат по трьох критеріях К1 – К3 на рекламу трьох товарів Т1 – Т3

Нумерація вершин мережі 1–7 проставлена на сірому фоні. «Батьківський потік» виділених коштів на рекламу $W_m=1$ (у відносних одиницях) входить в корінь дерева «1. Кошти на рекламу» і потім розділяється по трьох критеріях К1 – К3 на значення потоків ($W_{12}=0,4$; $W_{13}=0,3$; $W_{14}=0,3$) за суб'єктивно визначеними експертом «відносними потоками локальної ієрархії» ($\lambda_{12}=0,4$; $\lambda_{13}=0,3$; $\lambda_{14}=0,3$), підсумок яких дорівнює 1 (табл. 3.7.1).

Таблиця 3.7.1.

Визначення відносних потоків локальної ієрархії

«№. Критерій»	Відносні потоки локальної ієрархії $\lambda_{12} + \lambda_{13} + \lambda_{14} = 1$	Вхідний потік нащадків – критеріїв К1-К3
3. К1. Прибуток	$\lambda_{12} = 0,4$	$W_{12} = \lambda_{12} W_m = 0,4 \cdot 1 = 0,4;$
3. К3. Кількість діб, що залишилась до завершення терміну зберігання	$\lambda_{13} = 0,3$	$W_{13} = \lambda_{13} W_m = 0,3 \cdot 1 = 0,3$
3. К3. Незручність зберігання	$\lambda_{14} = 0,3$	$W_{14} = \lambda_{14} W_m = 0,3 \cdot 1 = 0,3$

Реальні вхідні потоки «вузлів-нащадків» К1 – К3 дорівнюють вхідному потокові «батьківського вузла» $W_m = 1$, помноженому на «відносний потік локальної ієрархії $\lambda_{12} - \lambda_{14}$ ». У цьому випадку вхідні потоки $W_{12} - W_{14}$ «вершин – критеріїв – нащадків К1 – К3» по значенню співпадають з відносними потоками локальної ієрархії $\lambda_{12} - \lambda_{14}$.

В свою чергу кожний критерій К1 – К3 є «батьківським вузлом» з власним вхідним потоком $W_{12} - W_{14}$, який розділяється на три частки по трьох ієрархічно нижчих потоках $W_{T1} - W_{T3}$ для «Альтернатив Т1 – Т3 = Товарів Т1 – Т3 = Нащадків Т1 – Т3» (див. рис. 3.7.1).

Нижче для кожного критерію К1 – К3 окремо розглядається розподіл батьківських потоків $W_{12} - W_{14}$ по «Нащадках Т1 – Т3 = Товарах Т1 – Т3 = Альтернативах Т1-Т3».

Критерій «2. К1. Прибуток». К1 має вхідний потік $W_{12} = 0,4$, який розділяється по нащадках – товарах Т1 – Т3 по «оцінках знизу (від нащадків)» у вигляді відповідних прибутків $a_{25} = 1000; a_{26} = 1500; a_{27} = 750$ (рис. 3.7.1). Чим більший прибуток (a_{25}, a_{26}, a_{27}) має товар Т1 – Т3, тим більшу частку батьківського потоку $W_{12} = 0,4$ на рекламу має отримати товар, бо реклама має підвищувати прибуток.

Значення відносних потоків локальної ієрархії λ_{2j} ($j = 5, 6, 7$) та відповідні значення реальних потоків W_{2j} як результат розподілу батьківського потоку $W_{12} = 0,4$ по нащадках – альтернативах Т1 – Т3 розраховуються за формулами $\lambda_{2j} = a_{2j} / (a_{25} + a_{26} + a_{27})$, $W_{2j} = \lambda_{2j} W_{12}$, звідки:

$$\lambda_{25} = 1000 / (1000 + 1500 + 750) = 0,308; W_{25} = \lambda_{25} W_{12} = 0,4 \cdot 0,308 = 0,123;$$

$$\lambda_{26} = 1500 / (1000 + 1500 + 750) = 0,461; W_{26} = \lambda_{26} W_{12} = 0,4 \cdot 0,461 = 0,184;$$

$$\lambda_{27} = 750 / (1000 + 1500 + 750) = 0,231; W_{27} = \lambda_{27} W_{12} = 0,4 \cdot 0,231 = 0,093.$$

Перевірка розрахованих потоків ($W_{25} + W_{26} + W_{27}$) показує, що їх підсумок не дорівнює вхідному батьківському потоку $W_{12} = 0,4$ із-за похибок округлення. Тому експерт суб'єктивно підвищує значення потоку, наприклад, $W_{27} = 0,093$ з тим, щоб підсумок потоків строго відповідав батьківському потоку, якщо не оговорена інша умова [53].

Критерій «3. К3. Кількість діб, яка залишилась до завершення терміну зберігання». Критерій К2 має вхідний потік $W_{13} = 0,3$, який розділяється по «нащадках – товарах Т1 – Т3» по «оцінках знизу (від нащадків)» у вигляді кількості діб, яка залишилась до завершення терміну зберігання $c_{35} = 25$ діб; $c_{36} = 8$ діб; $c_{37} = 15$ діб (рис. 3.7.1). Чим більший термін (c_{35}, c_{36}, c_{37}) має товар Т1 – Т3, тим меншу частку батьківського потоку $W_{13} = 0,3$ має отримати товар, бо реклама має більше сприяти продажу товару з найменшим терміном зберігання, щоб не нести збитки. Тому що оцінка c_{3j} діє у зворотному напрямку у порівнянні з a_{ij} , у розрахунках використовуємо $a_{3j} = 1/c_{3j}$: $a_{35} = 1/c_{35} = 1/25 = 0,04$; $a_{36} = 1/c_{36} = 1/8 = 0,125$; $a_{37} = 1/c_{37} = 1/15 = 0,0667$, звідки значення потоків локальної ієрархії λ_{3j} та відповідні значення реальних потоків W_{3j} , як результат розподілу батьківського потоку $W_{13} = 0,3$ по нащадках – альтернативах Т1 – Т3 розраховують за формулами $\lambda_{3j} = a_{3j} / (a_{35} + a_{36} + a_{37})$, $W_{3j} = \lambda_{3j} W_{13}$:

$$\lambda_{35} = 0,04 / (0,04 + 0,125 + 0,0667) = 0,173; W_{35} = 0,3 \cdot 0,173 = 0,0519;$$

$$\lambda_{36} = 0,125 / (0,04 + 0,125 + 0,0667) = 0,539; W_{36} = 0,3 \cdot 0,539 = 0,1617;$$

$$\lambda_{37} = 0,0667 / (0,04 + 0,125 + 0,0667) = 0,288; W_{37} = 0,3 \cdot 0,288 = 0,0863.$$

Підсумок потоків ($W_{35} + W_{36} + W_{37}$) = 0,3 = W_{13} показує, що він дорівнює вхідному батьківському потоку $W_{13} = 0,3$, внаслідок чого його не вправляють.

Критерій «4. К3. Незручність зберігання» – це якісний показник і для кожного з товарів Т1 – Т3 він вказується у відсотках. Із-за незручності у зберіганні товари Т1 – Т3 змушені швидше продати за допомогою реклами. К3 має вхідний потік $W_{14} = 0,3$, який розділяється по нащадках – товарах Т1 – Т3 по «оцінках знизу (від нащадків)» у вигляді $a_{45} = 50$ %; $a_{46} = 20$ %; $a_{47} = 70$ % (рис. 3.7.1). Чим більшу незручність (a_{45}, a_{46}, a_{47}) має товар Т1 – Т3, тим більшу частку батьківського потоку $W_{14} = 0,3$ на рекламу він має отримати.

Значення відносних потоків локальної ієрархії λ_{4j} та відповідні значення реальних потоків W_{4j} , як результат розподілу батьківського потоку $W_{14} = 0,3$ по «нащадках – альтернативах Т1 – Т3» розраховуються за формулами $\lambda_{4j} = a_{4j} / (a_{45} + a_{46} + a_{47})$, $W_{4j} = \lambda_{4j} W_{14}$:

$$\lambda_{45} = 50 / (50 + 20 + 70) = 0,357; W_{45} = \lambda_{45} W_{14} = 0,3 \cdot 0,357 = 0,107;$$

$$\lambda_{46} = 20 / (50 + 20 + 70) = 0,143; W_{46} = \lambda_{46} W_{14} = 0,3 \cdot 0,143 = 0,0429;$$

$$\lambda_{47} = 70 / (50 + 20 + 70) = 0,5; W_{47} = \lambda_{47} W_{14} = 0,3 \cdot 0,5 = 0,15.$$

Перевірка потоків ($W_{45} + W_{46} + W_{47}$) показує, що їх підсумок не дорівнює вхідному батьківському потоку $W_{14} = 0,3$ із-за похибок округлення. Тому експерт суб'єктивно підвищує значення потоку $W_{46} = 0,043$ з тим, щоб підсумок потоків строго відповідав батьківському потоку.

Підсумкові ієрархічні потоки для «товарів – альтернатив – нащадків Т1 – Т3» дорівнюють:

$$- \text{Для Т1: } W_{T1} = W_{25} + W_{35} + W_{45} = 0,123 + 0,0519 + 0,107 = 0,2819.$$

– Для T2: $W_{T2} = W_{26} + W_{36} + W_{46} = 0,184 + 0,1617 + 0,0429 = 0,3886$.

– Для T3: $W_{T3} = W_{27} + W_{37} + W_{47} = 0,092 + 0,0864 + 0,15 = 0,3283$.

Підсумок потоків виходу не дорівнює W_m : ($W_{T1} + W_{T2} + W_{T3} = 0,2819 + 0,3886 + 0,3284 = 0,9989$) \neq ($W_m = 1$). Тому експерт суб'єктивно виправляє цю похибку збільшенням, наприклад, значення потоку $W_{T2} = 0,3897$.

В результаті розрахунків отримуємо наступні рівні вартості рекламних заходів:

– на товар T1 витрачається частка $W_{T1} = 0,2819$ рекламних коштів;

– на товари (T2, T3) витрачається загальна частка ($W_{T2} + W_{T3}$) = ($0,3897 + 0,3284$) = $0,7181$ рекламних коштів.

3.8. Ринкова ціна на товар

Нижче наведені деякі методи розрахунку ринкової ціни, які урахують різні ознаки [99].

1. *За величиною основного параметру* ціна нового товару дорівнює [99]

$$C_{H1} = \frac{C_B}{B_B} B_H,$$

де C_B – ціна базового виробу; B_B – величина параметру базового виробу; B_H – величина параметру нового виробу.

Недолік методу – ураховується один фактор (наприклад, величина одного яблука).

2. *Метод регресійного аналізу* [99]. По ряду конкуруючих товарів у кількості m складається таблиця спостережень вихідної ціни продажу $C_K(c_1, c_2, \dots, c_i, \dots, c_m)$ у залежності від матриці кількісних та якісних показників $X_K(x_1, x_2, \dots, x_j, \dots, x_n)$ у кількості n . Складається матричне рівняння

$$C_K = X_K B_K, \tag{3.8.1}$$

де $C_K = \begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_i \\ \dots \\ c_m \end{bmatrix}; X_K = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1j} & \dots & x_{1n} \\ x_{21} & x_{22} & & x_{2j} & & x_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{i1} & x_{i2} & \dots & x_{ij} & \dots & x_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mj} & \dots & x_{mn} \end{bmatrix}; B_K = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_j \\ \dots \\ b_n \end{bmatrix};$

C_K – вектор цін на m товарів; X_K – матриця якостей у кількості n для m товарів; B_K – невідомі коефіцієнти пропорційності, які потрібно визначити.

Коефіцієнти пропорційності B_K визначаються з формули (3.8.1) за методом найменших квадратів

$$X_K^T C_K = X_K^T X_K B_K; (X_K^T X_K)^{-1} X_K^T C_K = (X_K^T X_K)^{-1} (X_K^T X_K) B_K; B_K = (X_K^T X_K)^{-1} X_K^T C_K.$$

Ціна нового товару залежить від змінних нового товару $X_H^T(x_{H1}, x_{H2}, \dots, x_{Hj}, \dots, x_{Hn})$ і дорівнює $C_H = B_K^T X_H$.

При визначенні ціни урахують наступні особливості та ознаки ринку:

- урахування політики ціноутворення, тобто залежність ціни від різних обставин: від місця торгівельної точки (у центрі міста ціни вищі); від часу (сезону, дня тижня, години продажу); від об'єму купленого товару; від покупців (для молоді і старих ціни нижчі; відношення покупців до товару та його якостей); від мети утвердження на ринку (співпраця, захоплення ринку) тощо;
- визначення попиту;
- порівняння товарів по усіх властивостях (прибуток, ціна, якість, надійність, зручність, привабливість тощо) з отриманням умовної ціни з точки зору покупця;
- верхня ціна ринку (обмежена проблемою реалізації – платіжоспроможним попитом),
- нижня ціна ринку (обмежена собівартістю товару; собівартість повинна урахувувати перевезення на ринок та додаткові витрати продажу на даному ринку);
- обмеження по цінах (заборона демпінгових цін) та об'єму продажу.

Далі виконується практична перевірка ціни на товар та часу його продажу по кількох малих партіях товару за різними цінами. Товари групуються по прибутковості, збагаченню у часі та іншим показникам.

Для визначення ціни на товар потрібно ретельно вивчити ринок шляхом опитування покупців, спостереження за об'ємами продаж, реєстрації цін, обробки всієї можливої інформації стосовно уподобань покупців та їх критичних зауважень. Наразі конкуренція стала більш жорсткою: можуть помітити навіть контрольну спробу вийти на ринок з власним товаром [99].

3.9. Групування об'єктів розділом та об'єднанням множин

Будь-яка інтелектуальна діяльність людини потребує поділу об'єктів на групи, які мають власні особливості та ознаки.

Від угруповань слід відрізнити **класифікацію**, яка має ієрархічно вищий рівень і охоплює систематизований розділ явищ і об'єктів на певні взаємопов'язані класи, групи, розряди на підставі їх подібності та відмінності.

Групування об'єктів відбувається розділом множини на малі групи з наступним об'єднанням малих груп у більші групи.

Вибір групуючої ознаки (по зменшенню чи збільшенню груп) залежить від заданого типу групування:

- *Структурнегрупування* – поділ однорідної сукупності на групи, що характеризують *структуру досліджуваної системи за деякою ознакою* (аналіз оплати співробітників по відділам банку). Аналіз структурних угруповань, взятих за ряд періодів показує структурні зрушення.

– *Типологічне групування по одній головній ознаці* – поділ упорядкованої ранжованої сукупності на однорідні групи – освіти, віку, зарплаті. За кількістю робітників або доходів групують підприємств на малі, середні та великі.

– *Інтелектуальне групування по кількох зважених ознаках* по розділу множини об'єктів на групи: обрання альтернатив; пошук рішень у просторі ознак; експертні системи; поділ людей медичною експертною системою по хворобах; представлення знань; спілкування на природній мові; комп'ютерний зір тощо.

– *Кластерне групування по всіх ознаках* упорядковує об'єкти у порівняно однорідні групи – кластери за великою кількістю *методів кластерного аналізу* [нейронна мережа Кохонена, К-середніх (K-means) ієрархічна кластеризація або *таксономія*, метод просіювання тощо].

Серед усіх типів угруповань окремо виділяється широким застосуванням *статистичне групування*, яке розділяє об'єкти на класи, групи та підгрупи за суттєвими ознаками. *Мета групування* статистичних даних: виявлення *закономірностей* у даних вибірки; *стиснення даних* без суттєвої втрати інформативності та *полегшення подальшої роботи* з даними.

При розділі множини на малі групи число груп визначається наступним чином:

- По окремим ознакам, їх сукупностям та результатам обробки ознак.
- По кількості значень якісної ознаки.
- По числовій ознаці – за *формулою Стерджеса*:
 $n = 1 + 3.322 \cdot \lg N = \log_2 N + 1$, де n – число груп; N – число одиниць вибірки.

Дані за формулою Стерджеса наведені в табл. 3.9.1.

Таблиця 3.9.1.

Число груп за формулою Стерджеса

<i>N</i>	15-24	25-44	45-89	90-179	180-359	360-719
<i>n</i>	5	6	7	8	9	10

Інтервал угруповання розглядається як нижня та верхня межа інтервалу ($X_{\max} \dots X_{\min}$). *Рівні інтервали* застосовуються в випадках, коли значення кількісної ознаки всередині сукупності змінюється рівномірно. Величина рівних інтервалів визначається за формулою $i = (X_{\max} - X_{\min})/n$, де i – величина інтервалу; X_{\max} – максимальне значення ознаки в сукупності; X_{\min} – мінімальне значення ознаки в сукупності; n – число груп.

При об'єднанні малих груп у більші групи використовується:

– *Аналітичне (функціональне) угруповання* – це об'єднання об'єктів навколо узагальнюючого показника (інші показники розглядаються як фактори, що впливають на узагальнюючий показник).

– *Вторинне угруповання* – це об'єднання об'єктів при використанні вже наявних угруповань, які можуть бути неоднаковими і тому непорівнян-

ними (зарплата робітників банку та підприємства). Для об'єднання таких угруповань застосовують наступні два способи утворення нових узагальнюючих груп:

1. Укрупнюють інтервали з охопленням деяких існуючих груп.
2. Після виконання п. 1 кожна група може бути додатково розділена на ряд нових груп з введенням нових інтервалів (це зветься методом часткового перегрупування).

3.10. Дискримінантний аналіз

Багатофакторний дискримінантний аналіз розв'язує проблему визначення стану об'єкта шляхом порівняння з деякими наданими експертом діапазонами значень результатів розрахунків евристичної багатофакторної дискримінантної функції у формі $Z = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + \dots + a_nx_n$, де $a_0, a_1, a_2, \dots, a_n$ – підібрані експертом коефіцієнти дискримінантної функції; $x_1, x_2, x_3, \dots, x_n$ – підібрані експертом показники (змінні) дискримінантної функції.

Дискримінантний аналіз є близьким до регресійного аналізу, який виражає одну із залежних змінних у вигляді лінійної залежності від інших змінних. Але у дискримінантному аналізі залежна змінна Z є *категоріальною змінною* (наприклад, при перевищенні деякого значення робиться категоріальний вербальний висновок про банкрутство) в той час, як у регресійному аналізі залежна змінна є *числовою величиною*. Існує багато моделей прогнозування банкрутства на основі багатофакторного дискримінантного аналізу.

Дискримінантний аналіз дозволяє по оцінках сукупностей змінних:

1. *При створенні нового сорту рослин*: віднести групу рослин до нового сорту (група 1); віднести групу рослин до перспективних для подальшої їх обробки (група 2); віднести групу рослин до браку (група 3).

2. *При створенні нових ліків прогнозувати*: пацієнт видужав (група 1), пацієнт одужав частково (група 2), пацієнт не одужав (група 3).

3. *Дискримінантом багатих та бідних людей* є певний рівень прибутку, який їх розділяє.

4. *Прикладом дискримінантного аналізу* є отримання логічно-аналітичної залежності віднесення випускника середньої школи до однієї з трьох груп: а) група випускників, яка поступає у ВНЗ; б) група випускників, яка поступає у професійні школи; в) група випускників, яка стає робітниками.

Для отримання функції дискримінантного аналізу можна використати: якість освіти в школі; середня оцінка випускника; оцінки по окремим дисциплінам; заможність батьків тощо.

5. *Визначити загрози банкрутства*, сегментацію ринку, оцінку ступеня новизни товарів тощо. Згідно закордонному досвіду прогнозувати банкрутство *можна за 1,5–2 роки* до появи його очевидних ознак. У світі існує велика кількість алгоритмів здійснення такого прогнозування, але найчас-

тіше на практиці застосовується дискримінантний аналіз.

Загальний висновок про якість фінансового стану підприємства робиться на основі порівняння з граничними значеннями ряду змінних, які включені в систему аналізу. Найвідомішими моделями однофакторного дискримінантного аналізу є *системи В. Бівера та П. Вайбеля*, які складаються з 6-ти показників [134; 202]. Показниками є:

- відношення наявних грошей до позичкового капіталу;
- відношення чистого прибутку до суми балансу;
- відношення позичкового капіталу до суми балансу;
- відношення оборотних активів до поточних зобов'язань;
- відношення робочого капіталу до суми балансу;
- відношення різниці між очікуваними грошовими надходженнями і поточними зобов'язаннями до витрат підприємства (без амортизації).

У разі, якщо значення *всіх шести показників* підприємства перебувають у безпечній зоні, то цьому суб'єкту господарювання не загрожує банкрутство. Якщо ж *більше трьох показників* наближаються до «хворобних» значень, то аналізованому підприємству загрожує банкрутство.

Об'єктивне прогнозування банкрутства при однофакторному дискримінантному аналізі є затрудненим, тому що одні значення показників можуть свідчити про позитивний розвиток підприємства, а інші – про незадовільний розвиток.

3.11. Системи автоматичного регулювання та захисту

Розпізнавання образу завжди пов'язане з дією. Системи автоматичного регулювання та захисту органічно об'єднують обробку ознак об'єкта, який захищається, визначення його стану та виконання адекватної дії.

Будь-які системи управління та захисту (для керування підприємством та окремими його частками; для управління космічними комплексами, ракетами чи електричними приводами тощо) можна класифікувати як кібернетичні, бо всі вони розглядають об'єкт керування у вигляді *деякої сукупності станів* $K=(k_1, k_2, \dots, k_i, \dots, k_m)$, $i=1, 2, \dots, m$, які залежать від початкових ознак і зворотних зв'язків. Тому контролюються *ознаки і стани керованої системи*.

Кожний стан характеризується *вектором ознак* $X=(x_1, x_2, \dots, x_j, \dots, x_n)$, $j=1, 2, \dots, n$, обробка яких дозволяє *класифікувати стан об'єкту* з метою виконання наступних адекватних керуючих *оптимальних дій*.

Тобто загальний об'єкт керування можна розглядати як *універсум (сукупність, множина) станів*, який потрібно розпізнати (класифікувати) і виконати відповідні дії.

Наприклад, класифікація об'єкта системою захисту фреонового комп-

ресора кондиціонування повітря може характеризуватись за наступними ознаками нормальної роботи чи аварії, які змінюються релейно і визначають відповідні стани:

- тиск всмоктування: «нормальний/занижений» («стан1»/«стан 2»);
- тиск нагнітання: «нормальний/підвищений» («стан1»/«стан 3»);
- змащення деталей тертя: «є/відсутнє» («стан1»/«стан 4»);
- нагрів приводного електричного двигуна: «нормальний/підвищений» («стан1»/«стан 5»).

На «стан 1» система захисту не реагує; при наявності «станів 2–5» система захисту вимикає з роботи компресор з відповідною сигналізацією.

Не лише система захисту, алей система керування може використовувати ознаки, що змінюються безперервно. Тому система керування *теж відноситься до систем розпізнавання образів-станів*. Наприклад, щоб не було помпажу у реактивному двигуні, система автоматичного регулювання повинна за відповідними ознаками (*температурою, тиском, об'єму окислювача*) подавати відповідну кількість палива. Прилади автоматичного керування забезпечують безпечний взаємний зв'язок цих чотирьох ознак.

Примітка. Помпаж – це зрив газового потоку з лопаток турбіни при перевищенні швидкості звуку; це є аварією і сприймається зовні як вибух. Після помпажу турбіну звичайно далі не використовують.

Систему безперервного регулювання, яка реагує на подібні ознаки об'єкта можна віднести до розпізнавання станів образу керованого об'єкта: вона розпізнає, що об'єкт, який захищається, набуває ненормальну, або небезпечну ознаку, яку треба або виправити, або вимкнути керований об'єкт з роботи, або подати відповідний попереджувальний сигнал.

3.12. Особливості комп'ютерного керування

Комп'ютерне керування відноситься до більш високого рівня керування і насамперед означає підвищення «інтелекту» та надійності роботи за рахунок:

– *розпізнавання* ознак і подій технологічного процесу, їх відхилень від запланованих даних. Комп'ютер дозволяє урахувати окремі ознаки, їх сукупності та складні функціональні, логічні і лінгвістичні обмеження та правила;

– *контролю* часу, своєчасності рішення, логістики та взаємного зв'язку між подіями;

– *ієрархічного диспетчерського керування*;

– *підвищення швидкодії* за рахунок розпаралелювання процесів. Під розпаралелюванням процесів розуміється заміна послідовного виконання операцій у звичайному комп'ютері на їх паралельне виконання;

– *прийняття оптимального рішення* у будь-яких умовах та станах;

– *документалізації* станів та даних;

– використання контролю більшої кількості параметрів; багатокритеріальності (з обранням оптимального критерія для існуючого стану), дублювання функцій;

– вироблення послуг у «*потрібній кількості, потрібної якості з доставленням збережених послуг у потрібне місце і у потрібний час*»;

– охоплення керуванням взаємопов'язаних, розподілених у часі і просторі конвеєрних процесів виробництва послуг, які охороняються.

Перелічені особливості вимагають вимірювання, отримання і обробки значної кількості ознак.

Ієрархічні рівні автоматизації керування підприємством наведені на рис. 3.12.1.

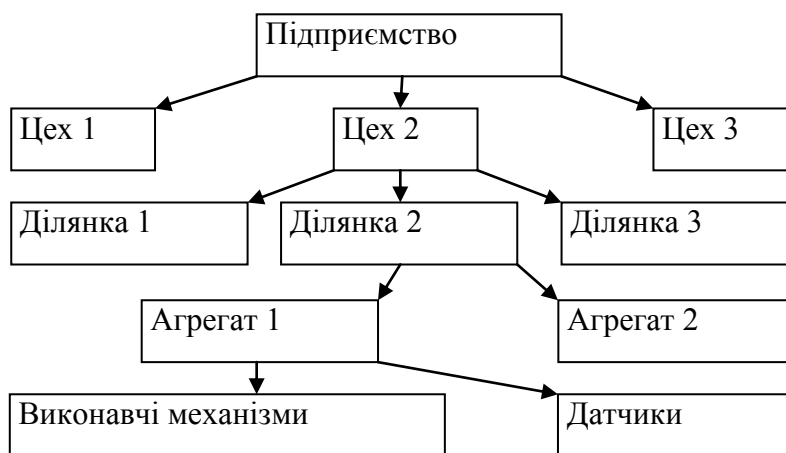


Рис. 3.12.1. Ієрархічні рівні автоматизації з використанням комп'ютерного керування

Важливого значення набуває контроль такої ознаки як **час**. Швидкодія сучасних комп'ютерів досить висока і, у принципі, може задовольняти вимоги керування багатьох технологічних процесів.

Але це не означає створення систем «реального часу». На звичайних ЕОМ можна створити такий «нескінченний» цикл, що звичайна ЕОМ «зависне», або не дасть рішення у потрібний термін.

Поняття «реального часу» означає насамперед *гарантоване вчасне* отримання оптимального рішення. Тому операційна система (ОС) замінюється на операційну систему реального часу (ОСРЧ).

ОСРЧ на обробку кожної Події 1 надає гарантований термін її обробки $T_{рч}^1$ (рис. 3.12.2), в тому числі при створенні циклів або при реалізації будь-яких алгоритмів. Цим ОСРЧ відрізняється від звичайних ОС. Подібний режим гарантованого забезпечення реального часу звичайні комп'ютери не підтримують. Тому вимушені були створювати ОС та ЕОМ реального часу.

На рис. 3.12.2 наведено реагування у часі програмного забезпечення інформаційно-керуючої системи у реальному часі (РЧ) на Подію 1 та Подію 2.

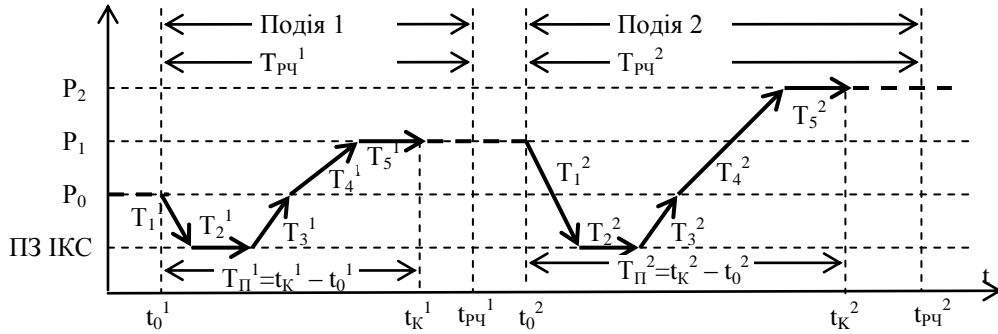


Рис. 3.12.2. Реагування у часі на події в об'єкті програмного забезпечення РЧ

- де P_0, P_1, P_2 – рівні отриманих рішень для процесів у керованому об'єкті;
 ПЗ ІКС – рівень програмного забезпечення (ПЗ) інформаційно-керуючої системи (ІКС) у реальному часі (РЧ);
 t_0^1 – момент початку Події 1 у процесах керованого об'єкту;
 t_k^1 – час завершення виконання рішення по Події 1 з урахуванням перехідних процесів в об'єкті керування;
 t_{rc}^1 – припустимий час для завершення виконання рішення по Події 1;
 T_{rc}^1 – заданий припустимий відрізок часу для завершення виконання рішення ПЗ ІКС по Події 1 з урахуванням перехідних процесів в об'єкті керування;
 T_1^1 – час передачі інформації у ПЗ ІКС про виникнення Події 1 в об'єкті керування;
 T_2^1 – час на прийняття рішення у ПЗ ІКС по Події 1;
 T_3^1 – час передачі рішення по Події 1 із ПЗ ІКС на виконавчі пристрої об'єкту;
 T_4^1 – час реалізації рішення по Події 1 виконавчими пристроями;
 T_5^1 – час перехідних процесів по Події 1 в об'єкті керування.

З рис. 3.12.2 випливає, що «оптимальне» рішення, отримане для Події 1 у час $t_k^1 > t_0^1$ *реалізується у час t_k^1 завжди із запізненням, яке дорівнює $T_{II}^1 = t_k^1 - t_0^1$* , бо воно є оптимальним лише на момент часу t_0^1 .

Між тим за час обробки інформації та виконання рішення зовнішні умови можуть змінюватись, і отримане *запізнене рішення* може бути неприпустимо далеким від реально існуючих умов. Потрібно дотримуватись *не лише оптимальності, але й вчасності видачі рішення*.

Перед зверненням до функцій ОСРЧ потрібно скласти часові діаграми по виникненню події на керованому об'єкті, надходженню сигналу про подію на апаратно-програмний комплекс (АПК), видачі рішення, зникнення події на керованому об'єкті (рис. 3.12.3).

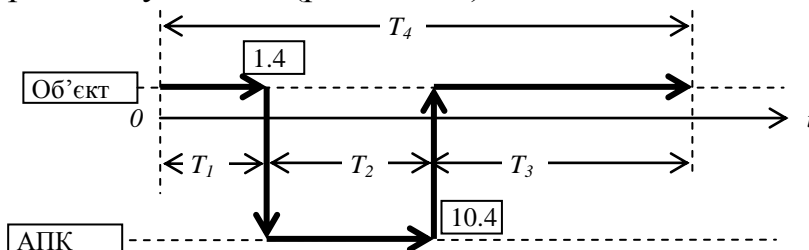


Рис. 3.12.3. Часова діаграма виникнення та ліквідації події на керованому об'єкті

Подія, яка обробляється на рис. 3.12.3 показана жирною лінією. На цій лінії визначають час: T_1 – час з моменту виникнення події до моменту видачі її датчиком, T_2 – час з моменту отримання АПК сигналу до моменту видачі рішення, T_3 – час з моменту видачі рішення до моменту реалізації рішення на об'єкті, T_4 – час з моменту виникнення події до моменту реалізації рішення на об'єкті (з урахуванням швидкості обміну даними між об'єктом та АПК). Вимоги до швидкодії АПК потрібно ставити у залежності від T_4 – часу з моменту виникнення події до моменту реалізації рішення на об'єкті. У прямокутнику біля цих подій вказані зв'язки даних: перша цифра – номер пристрою об'єкту, друга цифра – номер алгоритму АПК. При циклічній часовій діаграмі обирають термін циклу – час, на протязі якого виконуються розрахунки по різних частних алгоритмах чи їх модулях (відповідна частота вважається рівною $f = 1/T$ Гц) [119].

Звичайно швидкість вибірки для вхідних сигналів промислових вимірів знаходяться у межах 0.1сек. ... 1 день.

Розрізняють системи «**жорсткого часу**» (відхилення від визначеного часу неприпустимі: аварійний захист; системи керування літаком) та системи «**м'якого часу**» (відхилення від визначеного часу припустимі, хоча це може погіршити якість керування: відхилення у припустимих межах напруги на комп'ютері).

Вимоги до швидкодії АПК потрібно ставити у залежності від часу з моменту виникнення події до моменту реалізації рішення на об'єкті. Тобто розробник повинен розрізняти «реакцію системи **управління** на подію» від «реакції **системи на подію**», яка ураховує час приведення процесу в об'єкті до норми. Системи реального часу повинні ураховувати швидкість всіх елементів системи, пов'язаних з комп'ютером: **виконавчих органів, параметрів об'єкта, проміжних пристроїв, датчиків, операторів.**

4. Моделі представлення знань

4.1. Знання і їх представлення в системах розпізнавання образів

Представлення знань звичайно відносять до курсу штучного інтелекту. Але разом з тим системи представлення знань *мають унікальні можливості* і є потужним засобом для класифікації образів (їх розпізнавання): вони дозволяють використовувати для розпізнавання образів об'єднану текстову, числову, візуальну, логічну, структурну, семантичну, причинно-наслідкову та алгоритмічну інформацію. Тому з цієї точки зору нижче скорочено розглянемо деякі особливості представлення знань.

Знання можна поділити на:

- *екстенціональну частку* – факти;
- *інтенціональну частку* – правила і процедури отримання логічного висновку або нових знань.

Факти (*екстенціональна частка знань*) охоплюють перевірену практикою сукупність даних у вигляді моделей пізнання навколишньої дійсності [14].

Знання в системах штучного інтелекту зберігаються у вигляді організованих даних, які мають:

- *внутрішню інтерпретацію* (для цього кожна одиниця пам'яті повинна мати унікальне ім'я, а поряд з даними зберігається інформація, яка пояснює дані);

- *структурованість* (принцип «матрьошки», вкладання одного поняття в інше), під якою розуміється встановлення класифікаційних відношень і ієрархічних зв'язків: елемент – підклас – клас ціле – частка, тип – субтип, клас і т. д.;

- *зв'язність* (між інформаційними одиницями передбачаються зв'язки: логічні, функціональні, ситуаційні, просторові, часові, дедуктивні; асоціативні та семантичні; причинно-наслідкові; та ін.);

- *семантична близькість* (між інформаційними одиницями передбачаються семантичні зв'язки, для оцінки яких вводять семантичні метрики);

- *активність* (при виконанні деяких дій в системі РО, активується лише ті частки бази знань, які пов'язані з появою нових даних системи РО);

- *ефективність*, під якою розуміють швидкодію, спроможність використати лише необхідні знання та достатність інформації бази знань;

- *можливість навчання, самонавчання, зміни, корегування та доповнення.*

Знання допомагають вижити у конкурентній боротьбі.

Найбільш розповсюдженими є наступні моделі представлення знань (у відсотках указана уживаність моделі):

1. Логічні моделі (формальні системи, числення висловлень, логіка предикатів, нечітка логіка) – 4 %.
2. Продукційні моделі – 70 %.
3. Семантичні мережі – 8 %.
4. Фрейми – 18 %.

На сучасному етапі розвитку інформатики традиційна (реляційна, ієрархічна, мережна) *модель бази даних* (БД) *змінюється у напрямі використання бази знань*.

4.2. Формальні системи

У стародавні часи грецькі вчені розробили науку «геометрію». Вона побудована *методом, який зветься аксіоматичним*. Він складається з початкових висловлень, які зветься *постулатами або аксіомами* і приймаються «без доведення», із-за їх «зрозумілості та наочності», а далі весь вміст теорії отримується дедуктивним шляхом, на основі доведення теорем. Наприклад, одним із цих постулатів був постулат Евкліда: «через точку, яка не належить даній прямій, можна провести лише одну пряму, паралельну даній лінії». Усі інші висловлення даної науки виводяться шляхом розмірковувань, які зветься доведеннями [14].

Формальну логічну математичну систему (ФС) можна розглядати як теорію, яка задається четвіркою

$$\text{ФС} = \langle \text{Алф}, \text{Пс}, \text{Акс}, \text{Пв} \rangle,$$

де *Алф* – алфавіт (нескінченна або кінцева множина базових символів); *Пс* – правила синтаксису, на основі яких перевіряється правильність будівництва формули; *Акс* – множина базових аксіом, тобто **правильно побудованих формул (ППФ)**, які приймаються за істинні. **Аксіоми** мають вигляд тверджень про задані об'єкти й зв'язки між ними. В формальних системах прагнуть використати лише синтаксичні аспекти мови і не розглядають питання семантики та прагматики; *Пв* = { R_1, R_2, \dots, R_K } – множина правил виведення нових ППФ – рядків з існуючих ППФ. Кожне правило виведення R_i є предикатом $R_i(\beta_1, \beta_2, \dots, \beta_K)$ з аргументами $(\beta_1, \beta_2, \dots, \beta_K)$ у вигляді або деяких базових аксіом системи $(\beta_1, \beta_2, \dots, \beta_n)$ або інших нових ППФ. За обраним правилом R_i можна послідовно створювати нові рядки ППФ W_1, \dots, W_e з використанням базових аксіом системи та (або) аргументів обраного правила. Нові рядки ППФ у вигляді ланцюжка W_1, \dots, W_e створюються за правилом $R_i(\beta_1, \beta_2, \dots, \beta_K, W_1, \dots, W_e)$, є **аксіомами формальної системи**; кінцева ППФ – аксіома W_e , якщо вона далі не може бути перетвореною, – зветься **теоремою**. За правилами можна отримати ряд теорем. Усі проміжні ППФ, які використовувались при доведенні теореми, входять у сукупність аксіом даної теорії. Користувач може задати в формальній системі деяку власну формулу і може перевірити, чи є вона теоремою *ФС*. В *ФС* ППФ можуть охоплювати: символи алфавіту, базові аксіоми, аргументи правил R_i , отримані за правилами нові ППФ.

Процес створення нової *ППФ* виконується *вилученням з існуючої ППФ або доданням до неї теорем та аксіом (базової або нової)*.

Формальну систему можна реалізувати машинним способом (наприклад, за допомогою ЕОМ).

ФС можна розглядати як генератор нових знань, тому що з аксіом Акс (початкових знань) за правилами виведення *Пв* виводяться нові знання – *ППФ*. Механізм автоматичного доведення теорем покладено в основу *логічного програмування*: програма розглядається як набір логічних формул разом з теоремою, що має бути доведена. Крім того, теорія формальних систем розглядає процес *створення алгоритму* у формалізованих умовах.

ФС є *розв'язуваною*, якщо вона має процедуру, що дозволяє отримати виведення заданої користувачем *ППФ* в *ФС*; якщо такої процедури немає, то *ФС* є *не розв'язуваною*.

Приклад формальної математичної системи. Задані алфавіт $A = \{a, b, c, d, e, f\}$ та множина аксіом

$$abffe, \quad (1)$$

$$eedca, \quad (2)$$

$$ffb. \quad (3)$$

Правила виведення: $R_1(ff, eedca, abffe) = I$ (істина), $R_2(abca, ff, ffb, eedca) = I$ (істина).

Знайдемо можливе виведення в цій формальній системі:

1. $eedca$ – аксіома (2);
2. $abffe$ – аксіома (1);
3. ff – виводиться з (1), (3), R_1 ;
4. ffb – аксіома (3);
5. $abca$ – виводиться з (2), (3), *ППФ* ff та R_2 .

Для будь-якого заданого рядка нелегко перевірити, чи є він теоремою конкретної формальної системи. *Загалом немає алгоритму такої перевірки*. Якщо цей рядок не є теоремою, то намагання виконання всіх можливих варіантів виведення цього рядка перетворюється у нескінченний процес (тобто рядок не доводиться і не спростовується). Якщо ж задається послідовність у вигляді рядків з аксіом виведення, то цю послідовність легко перевірити.

Формальна система повинна алгоритмічно реалізуватись, тобто повинен існувати алгоритм виведення на ЕОМ.

Формалізовані логічні математичні моделі можна розбити на ряд напрямків: **формальна система, числення висловлень, числення предикатів, логіки високого порядку та псевдо фізичні логіки**. До логічних моделей можна віднести також й *машини Поста, машини Тюрінга, алгоритми автоматизації дедуктивних побудов* (від загальних правил до часткових наслідків): *Маркова*, які знайшли практичне застосування при створенні реальних логічних систем. Логічні моделі виконують дві функції на основі

– розв’язується задача виведення наслідків із заданих фактів (з аксіом на основі правил виведення отримуються усі правильно побудовані формули – ППФ);

– розв’язується задача перевірки істинності даного твердження (твердження вважається теоремою, і її істинність доводиться з аксіом даної теорії).

Перевагою логічних моделей є можливість чіткого теоретичного обґрунтування й реалізації формально точних і однозначних логічних побудов.

Недоліком логічних моделей є формальний процедурний стиль мислення і недостатня пристосованість до практичних потреб використання інтелектуальних систем із нечіткою структурою та розмитими суперечливими даними. Треба відмітити обмеженість формальних процедур по класах розв’язуваних задач, тому що звичайно реальна задача не має вигляд сукупності обмежень з однорідного набору аксіом та однаковими логічними правилами. В задачі можуть існувати різні числові обмеження, і людина у цьому випадку керується не лише класичною логікою для розв’язання логічних проблем. Тому в реальних системах часто відбувається об’єднання переваг логічних моделей з іншими системами.

4.3. Твердження та логічні формули

Логічне висловлення (речення, твердження) – це будь-яке оповідальне речення, про яке можна однозначно сказати, що воно «Істинне» або «Хибне»:

- речення «Київ – столиця України» є «Істинним»,
- речення «6 плюс 2 дорівнює 10» є «Хибним».

Логічна змінна($a-z$, $A-Z$) може приймати лише два значення: (1; Істина; I; True; T) або (0; Хиба; X; False; F).

Висновок – це форма розмірковування, за допомогою якого з одного чи кількох суджень можна отримати нове судження: «Всі дерева мають корені. Береза – дерево. Отже, береза має корені».

Логічна функція (логічний вираз) – це складне висловлювання у вигляді формули, яка складається з логічних змінних ($a-z$, $A-Z$) та логічних операцій над ними. Результатом розрахунку логічної формули є «Істина» (1, I, True), або «Хиба» (0, X, False).

Базові (основні) логічні операції мають вигляд:

- **кон’юнкції** (позначення: \wedge , $\&$, \bullet , логічне I, AND);
- **диз’юнкції** (позначення: \vee , логічне додавання, логічне «АБО», OR),
- **інверсії** (позначення: \neg , надкреслення літери, логічне НЕ, NOT).

Пріоритет логічних операцій: (дії у дужках), \neg , \wedge , \vee , \Rightarrow .

Звичайно аналіз і перетворення логічних формул виконують за допомогою Булевих функцій.

Приклад 1. Для простих висловлювань «A = Лив дощ» та «B = Трава була мокра» складне висловлювання може мати вигляд формул

$$F(A,B)=\{\underbrace{\text{Лив дощ, і трава була мокра}}_{A=1}\} = A \wedge B=1,$$

$$F(A,B)=\{\underbrace{\text{Лив дощ, і трава була мокра}}_{A=1}\} = A \wedge B=0,$$

Тут у першому реченні обидва висловлювання А та В є вірними, а у другому – лише одне.

Приклад 2. Для розпізнавання двох класів хвороб (Ω_1 – звичайна простуда та Ω_2 – ангіна) використовуємо логічні ознаки, які розглядаються як речення:

- А – висока температура (A=0 – «Ні», A=1 – «Так»);
- В – нежить (B=0 – «Ні», B=1 – «Так»);
- С – налив у горлі (C=0 – «Ні», C=1 – «Так»).

Припустимо, що для конкретного пацієнта нам відомі такі логічні ознаки: *висока температура A=1, нежить B=0, налив у горлі C=1*. Ці ознаки підставляються у логічні співвідношення і отримується правдивість або неправдивість опису класів. Без особливих претензій на точність діагнозу такі співвідношення можна уявити як:

$\Omega_1 = ABC + \overline{A}BC + A\overline{B}C = 0+0+0=0$ – звичайної простуди немає (тут ABC – є висока температура, є нежить, немає нариву у горлі; $\overline{A}BC$ – є висока температура, немає нежиті, немає нариву у горлі; $A\overline{B}C$ – немає високої температури, є нежить, немає нариву у горлі);

$\Omega_2 = \overline{A}BC + A\overline{B}C = 1+0=1$ – спостерігається ангіна.

Тут *множення* відповідає логічному «AND» (кон'юнкція «&»), а *складання* відповідає логічному «OR» (диз'юнкція «V»).

4.4. Таблиця істинності для відомої логічної формули

Таблиця істинності – таблиця, яка показує, яке значення приймає складне висловлювання при всіх варіантах (наборах) значень простих висловлень, які в нього входять.

Таблиці істинності (рішень) забезпечують ясність, однозначність, послідовність, несуперечливість, зрозумілість змісту інформації за формулою та виявляють можливі альтернативи розв'язку [9].

В таблиці істинності використовується основні булеві функції (табл. 4.4.1), у якій перші два стовпчики відведено аргументам (А, В), наступні – запереченням («¬А» та «¬В»), кон'юнкції «А ∧ В», диз'юнкції «А ∨ В», імплікації «А ⇒ В», еквівалентності «А ⇔ В».

Серед великої кількості булевих функцій особливу увагу *слід звернути на три з них*, бо за їх допомогою можна побудувати всі інші:

кон'юнкцію « \wedge » (логічне «і», логічне множення), диз'юнкцію « \vee » (логічне «або», логічне додавання) і заперечення « \neg ». Перші дві функції мають по два булевих аргументи, остання – лише один.

Таблиця 4.4.1.

Таблиця істинності булевих функцій

A	B	$\neg A$	$\neg B$	$A \wedge B$	$A \vee B$	$A \Rightarrow B$	$A \Leftrightarrow B$
0	0	1	1	0	0	1	1
0	1	1	0	0	1	1	0
1	0	0	1	0	1	0	0
1	1	0	0	1	1	1	1

Для визначення значення складного висловлювання при всіх наборах значень простих його висловлень, у таблиці істинності наводяться логічні значення: усіх змінних, окремих часток формули та цілком всієї формули.

Приклад 1. Будування таблиці істинності для заданої формули $Z = (\neg X \wedge Y) \vee (\neg(X \vee \neg Y))$.

Алгоритм має наступні кроки (табл. 4.4.2):

- визначають кількість змінних $n = 2$ у формулі (бо розглядаються змінні X та Y);
- розраховують кількість логічних операцій $p = 6$ в формулі ($\neg, \wedge, \vee, \neg, \vee, \neg$);
- визначають кількість рядків таблиці разом із заголовком по формулі $m = 2^n + 1 = 5$, де $n=2$ – кількість змінних (X, Y);
- встановлюють послідовність виконання логічних операцій з урахуванням пріоритетів та дужок;
- розраховують кількість колонок $s = n + p = 2 + 6 = 8$, яка дорівнює підсумку змінних « n » та кількості логічних операцій $p=6$;
- будують таблицю розміром $m \times s = 5 \times 8$.

Таблиця 4.4.2.

Таблиця істинності булевої функції

$Z = (\neg X \wedge Y) \vee (\neg(X \vee \neg Y))$ для двох змінних

X	Y	$\neg X$	$\neg Y$	$\neg X \wedge Y$	$X \vee \neg Y$	$\neg(X \vee \neg Y)$	Z
0	0	1	1	0	1	0	0
0	1	1	0	1	0	1	1
1	0	0	1	0	1	0	0
1	1	0	0	0	1	0	0

Заповнення таблиці:

- у заголовках колонок зліва направо вводять вхідні змінні, заперечення змінних, набори змінних з логічними операціями у порядку їх виконання, у кінцевій колонці – логічну формулу;
- розділяють колонку значень першої змінної X навпіл і заповнюють верхню частку колонки логічними значеннями 0, а нижню 1;

– розділяють колонку значень другої змінної Y на 4 частки і заповнюють непарні частки колонки логічними значеннями 0, а парні 1, починаючи нумерацію часток зверху;

– продовжують ділення наступних колонок змінних (якщо вони є) на 8, 16, 32 і т. д. із заповненням часток цих колонок логічними значеннями 0 та 1 до тих пір, поки ці частки не будуть складатись з одного символу;

– в результаті вказаних дій всі колонки змінних заповнені логічними значеннями змінних. Вводять наступні колонки із запереченнями логічних змінних;

– виконують заповнення таблиці істинності по колонках зліва направо у відповідності з послідовністю виконання логічних операцій.

Примітка. Від спокуси створення таблиці істинності без обчислення проміжних результатів ліше утриматись, бо інакше виникають труднощі пошуку можливих помилок.

У кінцевій колонці вказані розв'язки логічної формули Z , які дозволяють сконцентрувати увагу на окремих альтернативних варіантах розв'язку.

Приклад 2. Використання середовища табличного процесора *LibreOfficeCalc*. Усі функції табличного процесора *LibreOfficeCalc* поділено на категорії: математичні, статистичні, фінансові, текстові, інформаційні, логічні. Тобто табличні процесори дозволяють розв'язувати задачі Булевої алгебри. Логічний аналіз базується на поняттях Булевої алгебри (алгебри логіки) з позначенням істинності 1 та хибності 0.

З метою аналізу вводять таблицю, встановлюють курсор миші у потрібну комірку і звертаються до *«Помічника з функцій»*, наприклад, натиснувши клавіші *Ctrl + F2*. У вікні діалогу *«Function Wizard»* на закладці *«Functions»* вибрати категорію *«Логічні функції»*, а в ній – потрібну функцію. Ввести у поля аргументів потрібні параметри і натиснути кнопку *«Гаразд»*.

Алгебра логіки символічно зображує логічні складені висловлення у вигляді *логічних формул* з подальшим їх спрощенням за рахунок тотожних перетворень. Для отримання значення (*True* чи *False*) логічних формул потрібно виконати одночасне приписування змінним конкретних значень в логічній формі (*True* чи *False*), тобто виконати *конкретизацією* формули алгебри логіки.

Завдання 1. Побудувати власну логічну формулу на 7 логічних операцій з використанням двох змінних (A, B), дужок та зв'язків (\wedge, \vee, \neg). Визначити істинність власної формули.

Нижче у табл. 4.4.3 наведений приклад визначення істинності логічного виразу $F(A, B) = (A \vee B) \wedge (\neg A \vee B)$.

1. У виразі дві змінні A та B ($n=2$).

2. У формулі використано 4 логічних операцій ($p=4$).

3. Таблиця істинності має кількість рядків разом із заголовком $m=2n+1=2\cdot 2+1=5$.

4. Визначаємо порядок дій

1) $A \vee B$; 2) $\neg A$; 3) $\neg B$; 4) $\neg A \vee \neg B$; 5) $F = (A \vee B) \wedge (\neg A \vee \neg B)$ для введення їх у заголовки колонок таблиці істинності.

5. Таблиця істинності має кількість колонок $K = n + p = 2 + 4 = 6$.

Таблиця 4.4.3.

Таблиця істинності формули $F = (A \vee B) \wedge (\neg A \vee \neg B)$

A	B	$A \vee B$	$\neg A$	$\neg A \vee \neg B$	F
0	0	0	1	1	0
0	1	1	1	1	1
1	0	1	0	0	0
1	1	1	0	1	1

Висновок: логічний вираз має значення «Істина» при наборах $F(0,1)=1$ та $F(1,1)=1$.

Завдання 2. Побудувати таблицю істинності для власної логічної формули на не менше ніж 7 логічних операцій з використанням трьох змінних (A, B, C), дужок та зв'язків (\wedge , \vee , \neg).

4.5. Таблиця істинності для сукупності тверджень

Таблиці рішень є зручним інструментом аналізу ситуацій з організаційно-технічним та соціально-економічним аспектами. Вони є різноманітними і надають можливість вносити додаткові та вилучати зайві чи несуттєві умови.

Приклад 1. Скористаємось сукупністю складних тверджень та алгоритмом, наведених в [23]:

«1. Якщо замовлення від замовника буде на суму 90–100 тис. грн і замовник матиме довіру, то замовлення буде виконане у першу чергу та вчасно при умові надходження достатньої кількості сировини.

2. Замовлення на суму 90–100 тис. грн не виконується, якщо замовник не має довіри.

3. Сума замовлення не має значення, але якщо виникає сумнів у довірі до замовника, то навіть при наявності сировини його замовлення буде відкладене.

4. Якщо замовлення від замовника буде на суму понад 100 тис. грн, то замовлення не буде виконане вчасно».

У наведеному тексті виділяються **умови та дії** (у текстах дії вимагаються після слова «то», або перед словом «бо», «якщо» та ін.):

1. Експерт виділяє з наведеного тексту **перелік умов**, які можуть мати одне з двох логічних значень (I/X). При цьому першими перелічуються взаємно не пов'язані між собою умови:

Y_1 – «Замовлення буде на суму 90–100 тис. грн».

Y_2 – «Замовник має довіру».

- $\neg Y_2$ – «Замовник не має довіри».
- Y_3 – «Вчасно надійде достатньо сировини».
- Y_4 – «Замовлення буде на суму понад 100 тис. грн».
- $Y_1 \vee Y_4$ – «Замовлення на будь-яку з вказаних сум».
- Y_5 – «Виникає сумнів у довірі до замовника».

2. Експерт виділяє з наведеного тексту *перелік дій (перелік виходів логічної системи)*, які є результатом перелічених умов і можуть мати одне з двох логічних значень (I/X):

- D_1 – «Замовлення буде виконане у першу чергу».
- D_2 – «Замовлення буде виконане вчасно».
- $\neg D_2$ – «Замовлення не буде виконане вчасно».
- D_3 – «Замовлення не буде виконане».
- D_4 – «Замовлення буде відкладене».

Складання логічної формули.

Відображуємо зв'язок умов та вихідних дій логічної системи у вигляді логічних формул:

1. *Якщо замовлення від замовника буде на суму 90–100 тис. грн і замовник матиме довіру, то замовлення буде виконане у першу чергу та вчасно при умові надходження достатньої кількості сировини:* $\{Y_1 - \text{«Замовлення буде на суму 90–100 тис. грн»} \wedge Y_2 - \text{«Замовник має довіру»} \wedge Y_3 - \text{«Вчасно надійде достатньо сировини»}\} \Rightarrow \{D_1 - \text{«Замовлення буде виконане у першу чергу»} \wedge D_2 - \text{«Замовлення буде виконане вчасно»}\}$

$$Y_1 \wedge Y_2 \wedge Y_3 \Rightarrow D_1 \wedge D_2; \quad (4.5.1)$$

Якщо замовлення від замовника буде на суму 90–100 тис. грн і замовник матиме довіру, то замовлення буде виконане у першу чергу та вчасно при умові надходження достатньої кількості сировини: $\{Y_1 - \text{«Замовлення буде на суму 90–100 тис. грн»} \wedge Y_2 - \text{«Замовник має довіру»} \wedge \neg Y_3 - \text{«Не надійде вчасно достатньо сировини»}\} \Rightarrow \{D_1 - \text{«Замовлення буде виконане у першу чергу»} \wedge \neg D_2 - \text{«Замовлення не буде виконане вчасно»}\}$

$$Y_1 \wedge Y_2 \wedge \neg Y_3 \Rightarrow D_1 \wedge \neg D_2; \quad (4.5.2)$$

2. *Замовлення на суму 90–100 тис. грн не виконується, якщо замовник не має довіри:* $\{Y_1 - \text{«Замовлення буде на суму 90–100 тис. грн»} \wedge \neg Y_2 - \text{«Замовник не має довіри»}\} \Rightarrow \{D_3 - \text{«Замовлення не буде виконане»}\}$

$$Y_1 \wedge \neg Y_2 \Rightarrow D_3; \quad (4.5.3)$$

3. *Сума замовлення не має значення, але якщо виникає сумнів у довірі до замовника, то навіть при наявності сировини його замовлення буде відкладене:* $\{[Y_1 - \text{«Замовлення буде на суму 90–100 тис. грн»} \vee Y_4 - \text{«Замовлення буде на суму понад 100 тис. грн»}] \wedge Y_5 - \text{«Виникає сумнів у довірі до замовника»} \wedge Y_3 - \text{«Вчасно надійде достатньо сировини»}\} \Rightarrow \{D_4 - \text{«Замовлення буде відкладене»}\}$

$$(Y_1 \vee Y_4) \wedge Y_5 \wedge Y_3 \Rightarrow D_4. \quad (4.5.4)$$

4. Якщо замовлення від замовника буде на суму понад 100 тис. грн, то замовлення не буде виконане вчасно»: $\{ [Y_4 - \text{«Замовлення буде на суму понад 100 тис. грн»} \Rightarrow \{ \neg D_2 - \text{«Замовлення не буде виконане вчасно»} \}$

$$Y_4 \Rightarrow \neg D_2 \quad (4.5.5)$$

Отримані взаємно пов'язані та суперечливі зв'язки серед змінних $Y_1 - Y_5$ повинні знайти відображення у значеннях змінних в таблиці рішень 4.5.1.

Якщо виконується умова $\neg Y_2$ («Не має замовник довіри»), то вона тягне за собою й умову Y_5 («Виникає сумнів у довірі до замовника»). Умова $Y_2 = 1$ («Замовник має довіру») ще не означає, що існує умова $\neg Y_5$ («Не існують сумніви у довірі до замовника»): про це звичайно кажуть «Довіряй, але перевіряй»).

Всього отримуємо $2^5 = 32$ варіанти наявних сукупностей умов $Y_1 - Y_5$. Позначимо символами «1/0» «виконання/не виконання» умов $Y_1 - Y_5$, а в окремих колонках – виконання їх сукупностей згідно лівим часткам залежностей (4.5.1)–(4.5.5) – в табл. 4.5.1 це не показано.

Експертом в табл. 4.5.1 показана лише невелика частка вказаних розрахунків з наведенням значень правих часток залежностей (4.5.1)–(4.5.5) з використанням позначень $D_1 - D_4$. Зміст табл. 4.5.1 скорочується експертом з урахуванням логічних операторів (4.5.1)–(4.5.5) і вилученням суперечностей.

Таблиця 4.5.1.

Таблиця рішень для сукупності тверджень ($2^5=32$)

Ситуація	Y_1	Y_2	Y_3	Y_4	Y_5	$D_1 \wedge D_2$	$D_1 \wedge \neg D_2$	$\neg D_2$	D_3	D_4
1(4.5.4)	0	0	1	1	1	0	0	0	0	1
2(4.5.5)	0	1	0	1	0	0	0	1	0	0
3(4.5.3)	1	0	0	0	0	0	0	0	1	0
4(4.5.3)	1	0	1	1	1	0	0	0	1	0
5(4.5.2)	1	1	0	0	0	0	0	1	0	0
6(4.5.1)	1	1	1	0	0	1	0	0	0	0

З табл. 4.5.1 випливає, що є 6 ситуацій-рядків-кортежів, які можуть викликати 5 різних дій, що відображені в колонках праворуч від колонок змінних $Y_1 - Y_5$. Експертом оцінюється ризик ситуацій і обирається найменш ризикозна дія.

Алгебра логіки може не давати однозначної відповіді на поставлене питання, але надає можливість скоротити кількість варіантів, що розглядаються, і зменшити кількість факторів при мінімальній трудомісткості наступного аналізу.

4.6. Логіка предикатів

Предикатом зветься двозначна логічна функція $p(x_1, x_2, \dots, x_n)$, яка приймає значення 1 або 0 і яка означає довільно вкладену програмістом контекстну властивість або контекстне відношення у вигляді речення

стосовно довільного числа *предикатних змінних* (x_1, x_2, \dots, x_n) . Тобто предикат $p(x_1, x_2, \dots, x_n)$ задає відношення між елементами x_1, x_2, \dots, x_n , яке означає: «Змінні (x_1, x_2, \dots, x_n) знаходяться між собою у відношенні «р». Цей контекст не змінюється на протязі всієї програми. Предикат $p(x_1, x_2, \dots, x_n)$ помічається літерою алфавіту (у даному випадку помічений літерою «р», але може помічатись словом і навіть складним лінгвістичним визначенням) і у залежності від числа n предикатних змінних зветься *n-місним (або n-арним) предикатом*. Наприклад:

– предикат «високий(x)» має *властивість* оцінки росту стосовно предикатної змінної x ; при заміні змінної x на *предметну постійну* «Вова» змінна «вова» пишеться з маленької літери: «високий(вова)»;

– предикат «father(x_1, x_2)» означає, що змінні (x_1, x_2) знаходяться між собою у родинних відносинах: x_1 є батьком (або матір'ю) x_2 . Відповідно, x_2 є нащадком по відношенню до x_1 .

Контекст предиката довільно задається програмістом і діє від початку і до кінця програми. Наприклад, предикат «дорогий(x)» може мати два контексти: «дорогий у сенсі ринкової ціни» і «дорогий у сенсі пам'яті для людини». Щоб легше складати, використовувати, перевіряти і розуміти програму, потрібно знати контекст, який вкладається програмістом у предикати. Тому предикат бажано задавати відповідним значущим словом, хоча це не є обов'язковим: при складанні програми програміст може обрати будь-який з предикатів «високий (вова)», «в(вова)», «f(вова)» як тотожні за контекстом речення, кожне з яких може вважатись предикатом з контекстом «високий(вова)».

Числення предикатів або логіка першого порядку – це формальна система математичної логіки, в якій визначають висловлювання відносно змінних, фіксованих функцій і предикатів.

Приклади програм на мові VisualProlog.

Для загального пояснення задач, які розв'язуються за логікою предикатів, використаємо приклади програм на мові VisualProlog, яка «бере на себе» у захованому вигляді інтерпретацію отриманої системи логічних формул а також перетворення її у відповідь. Мова Visual Prolog є простою до розуміння і дуже близька до природної мови.

Алфавіт логіки предикатів VisualProlog:

- предикатні літери P, Q, R, S, T, V, N;
- предикатні (предметні) змінні або терми (x, y, z, \dots) з індексами або без них;
- предикатні (предметні) постійні або термами (a, b, c, \dots) з індексами або без них;
- функціональні символи f, g, h, \dots ;
- знаки пунктуації – ліва дужка та права дужка $()$;
- знаки логічних операцій $\neg, \wedge, \vee, \rightarrow$;
- знаки кванторів \exists, \forall .

Програма набирається по шаблону, який надає VisualProlog. Наведена нижче програма призначена для обрання або звуження об'єму пошуків автомобіля, коли покупець вводить обмеження: автомобіль повинен коштувати менше 25000, бути сірого кольору. Програма має вигляд (знак % означає коментар; слово *nondeterm* – кількаразове використання предиката у програмі):

```

INCLUDE «туех1.inc»    % директива з назвою програми «туех1.inc»
PREDICATES
іх змінних
nondeterm car          (symbol, long, integer, symbol, long)
nondeterm run          % предикат правила і включення програми
у роботу (без вказівки % типів змінних)
CLAUSES                % набір фактів та правил. Їх допускається
писати у рядок.
car (ford, 90000, 4, gray, 24000). % факт: автомобіль марки Ford,
пройшов шлях 90000 км, 4 років
% експлуатації, сірого кольору, коштує 24000 – цей контекст на-
дається програмістом.
car (volga, 120000, 3, red, 23000).
car (volga, 8000, 1, black, 50000).
car (zaporojec, 80000, 6, blue, 8000).
car (lada, 50000, 6, white, 18000).
car (toyota, 25000, 2, black, 35000).
run: – car (Make, Odometer, Years_on_road, ColorBody, Cost), % пра-
вило: автомобіль Cost<25000, % коштує менше 25000 «AND»
ColorBody=gray,          % колір – сірий «AND»
Write (Make, '\n').      % друк «Make» з нового рядку:
'\n' означає new line.
GOAL % використовується завжди лише один предикат run.
    
```

Відповідь VisualProlog на команду *Write(Make, '\n')* має вигляд (всі відповіді супроводжуються у кінці словом «yes»):

```
ford
yes
```

Правило в мові VisualProlog має структуру «Голова:-Тіло.», або «Голова:-<СубЦіль1>, <СубЦіль2>, ..., <СубЦільК>.»), де позначення мають наступне тлумачення:

– команда «:-» виконує логічну операцію «якщо, <СубЦіль1>, <СубЦіль2>, ..., <СубЦільК> є вірними, то предикат «Голова» є істиною» (тобто відому логічну операцію «якщо..., тоді...» записують у зворотному порядку «тоді ..., якщо ...»);

– кома «,» означає логічний зв'язок «AND» (<СубЦіль1> ^<СубЦіль2> ^...^ <СубЦільК>). Щоб замінити «AND» на «OR», потрібно замість коми «,» використовувати точку з комою «;».

Пролог є діалоговою мовою і складається з введених користувачем початкових даних (доменів (типів даних), предикатів, фактів, правил отримання нових властивостей і відносин з фактів), введеного користувачем питання до Прологу (GOAL) і відповіді Пролога.

Ще приклад програми на мові VisualProlog:

```
INCLUDE «myex2.inc»
PREDICATES likes (symbol, symbol)
CLAUSES likes (tom, apple). likes (tom, milk). likes (tom, sport).
% Факти: Том любить яблуко, молоко, % спорт
GOAL likes (tom, milk), likes (tom, sport).
% Чи любить Том молоко і спорт?
```

Відповідь Пролога: yes.

Якщо задати питання GOAL likes (tom, milk), likes (tom, tennis) (Чи любить Том молоко і тенніс?), то відповідю Пролога буде «no», бо не існує факту, що Том любить теніс. Якщо задати питання GOAL likes (tom, milk); likes (tom, tennis) (Чи любить Том молоко або тенніс?), то відповідю Пролога буде «yes»,

Якщо в предикаті $p(x_1, x_2, \dots, x_n)$ замість « x_n » задатись « a_n » і отримати $p(x_1, x_2, \dots, a_n)$, то $(n - 1)$ – арний предикат $p(x_1, x_2, \dots, a_n)$ вже не залежить від « x_n ». Якщо всі змінні предикату замінити на постійні, то отримуємо **0-арний предикат, або речення** [14].

У загальному випадку кожна змінна (x_1, x_2, \dots, x_n) визначена на множині об'єктів предметної ділянки (наприклад, змінна $x_2 =$ ольга предиката «father (x_1, x_2) » розшукується на множині всіх нащадків стосовно предикату та змінної x_1).

Квантори – це знаки (\exists, \forall) , які у сполучені зі змінними дають можливість стверджувати, що для всіх або деяких об'єктів предметної ділянки виконується деяка властивість або відношення. Якщо $p(x)$ – предикат, для якого змінна x визначена на множині X (це записують як $x \in X$), то твердження що розглянуті $x \in X$ мають властивість «р», можуть записуватись за допомогою двох кванторів [14]:

– $\forall x p(x)$ (читається «для всіх x р від x ») – твердження, що усі $x \in X$ мають властивість р. Квантор узагальнення \forall походить від англійського слова All (всі) з переверненою першою літерою. Знак квантора загальності \forall можна розуміти як узагальнення кон'юнкції: висловлювання $\forall x p(x)$, якщо множина $X=(a,b,c)$, є еквівалентним кон'юнкції $p(a) \wedge p(b) \wedge p(c)$.

– $\exists x p(x)$ (читається «існує таке x , що р від x ») – твердження, що існує хоча б один об'єкт $x \in X$, який має властивість р. Квантор існування \exists походить від англійського слова Exist (існує) з поверненою першою літерою. Знак квантора існування \exists можна розуміти як узагальнення диз'юнкції: висловлювання $\exists x p(x)$, якщо множина $X=(a,b,c)$, є еквівалентним диз'юнкції $p(a) \vee p(b) \vee p(c)$.

Якщо $p(x)$ означає « x – парне число m », то при наявності в множині X парних і непарних чисел, твердження $\exists x p(x)$ буде вірним, а твердження $\forall x p(x)$ – хибним.

Якщо ділянка дії квантора охоплює кілька предикатів, які зв'язані логічними операціями, то їх об'єднують дужками: $\forall x \forall y (P(x,y) \rightarrow R(x))$. Порядок слідування однойменних кванторів можна змінювати, але порядок слідування різнойменних кванторів не можна змінювати, бо це будуть різні висловлювання.

Для перетворення n -арного предиката $p(x_1, x_2, \dots, x_n)$ у висловлювання, потрібно зв'язати квантором кожну змінну. Квантори $\forall x$ та $\exists x$ пов'язують змінну x в ділянці своєї дії. Тому змінна x в формулах $\forall x p(x)$ та $\exists x p(x)$ є зв'язаною. Якщо змінна не пов'язана з квантором, то вона зветься *вільною змінною*. Наявність вільної змінної вказує на те, що дана формула не є висловлюванням, а є логічною функцією. Формула без вільних змінних є *замкнутою формулою*.

Предикати є елементарними формулами (атомами). Їх сукупність, пов'язана знаками логічних операцій ($\wedge, \vee, \rightarrow, \leftarrow, \leftrightarrow, \neg$), створюють формули.

Інтерпретація отриманої системи логічних формул призначена для перевірки, чи має ця система узагальнений сенс. Вважається, що отримані формули числення предикатів мають сенс, якщо існує хоча б одна **інтерпретація символів**, з яких складаються формули.

Один з способів *інтерпретації* полягає у тому, що задається не порожня множина ділянки *інтерпретації* M у вигляді об'єктів предметної ділянки і визначається функція, яка співставляє між собою формули та відношення між об'єктами з ділянки інтерпретації M . На основі інтерпретації визначається наявність сенсу, тобто можливість отримання загального рішення із заданої системи логічних формул.

Фрази Хорна.

Основою мови Пролог та метода резолюцій є фрази Хорна.

Фраза Хорна має вигляд логічного виразу

$$\boxed{\sim A \vee \sim B \vee \sim C \vee \sim D \vee E}, \sim \boxed{(ABCD) \vee E}, \text{ або } \boxed{\sim A \& \sim B \& \sim C \& \sim D \& E}, \quad (4.6.1)$$

у якому усі літерали, крім E , є негативними (тут знак « \sim » означає заперечення). Цей вираз може бути записаний і в іншій формі («якщо істинні A і B і C і D , то є істинним E »)

$$\boxed{E \leftarrow A, B, C, D} \quad (4.6.2)$$

Речення Хорна (4.6.2) записується у в мові Пролог у вигляді правила (із точкою у кінці):

$$\boxed{E: \sim A, B, C, D.} \quad (4.6.3)$$

або у більш загальному вигляді

$$\boxed{p: \sim q_1, q_2, \dots, q_n.} \quad (4.6.4)$$

Таке речення інтерпретується наступним чином: «Для усіх значень змінних у реченні (4.6.4) предикат p є істинним, якщо істинні q_1 « I » q_2 « I »... « I » q_n , де « I » – логічна функція. В реченні (4.6.4) пара символів « $: \rightarrow$ »

читаються «ЯКЩО», а коми «;» читаються як логічне «/». Якщо бажано логічне «/» замінити на «АБО», то в Пролозі використовується знак «;», але це вже є відступом від фраз Хорна, і цей відступ програмно обробляється окремо. Цей відступ та інші відступи спеціально вводяться в мову Пролог для наближення мови до реальних умов програмування та для більшої зручності в користуванні мовою.

Таким чином, усі правила Прологу (4.6.4), які вводяться користувачем, можна перетворити на фрази Хорна (4.6.1). Користувач мови Пролог по суті змушений використовувати фрази Хорна у вигляді правил мови Пролог.

Тому що фрази Хорна (4.6.1) мають не більше за одного позитивного літерала, існують два типи фраз Хорна: повна Хорнівська фраза – з одним позитивним літералом та неповна Хорнівська фраза – без жодного позитивного літерала. Наприклад:

інженер(A): – вища_освіта(B), спеціальність(C), знання_англ_мови(D). % у розділі CLAUSES: – інженер(A). % в розділі GOAL. Результатом співставлення двох повних хорнівських фраз завжди є повна фраза. Але, з іншого боку, в результаті аналізу методом резолюцій ми повинні отримати негативну відповідь у вигляді порожньої фрази «:-». Тому в програмі всі фрази Хорна, крім однієї (фрази мети в розділі GOAL), повинні бути повними, бо будь-яка теорема, виражена через хорнівські фрази може бути доведена лише при такій умові.

Нагадаємо деякі положення числення предикатів 1-го порядку стосовно мови Пролог.

Предикатом $P(x_1, x_2, \dots, x_n)$ зветься деяка логічна функція відношення P від довільного числа аргументів-об'єктів (x_1, x_2, \dots, x_n) , яка, в залежності від вкладеного користувачем довільного контексту відношень P між аргументами та конкретизації об'єктів, набуває значення «істина» або «хибність». В численні предикатів предикат (або його заперечення) вважається атомарною формулою (літералом).

Термами звуться аргументи предиката, які можуть бути змінними або константами. Термами також звуться функції від термів.

Літерою зветься або атом, або заперечення атома. Одиначним диз'юнктом зветься однолітерний диз'юнкт. Якщо диз'юнкт не має жодної літери, то він зветься пустим диз'юнктом і позначається 0. Літери L та $\neg L$ звуться контрарними.

Метод резолюцій Дж. Робінсона.

При використанні числення предикатів розв'язок можна отримати за допомогою правила *modus ponens* $A \Rightarrow B$ («Якщо A то B »). Але в дійсності автоматизоване доведення теореми (отримання розв'язку проблеми) виконується на основі більш ефективного запропонованого Дж. Робінсоном у 1965 р. *метода резолюцій*, який дозволяє автоматично доводити теореми на основі аксіом та правил виведення. *На основі метода резолюцій француз Альбер Кольмерое в 1973 р. створив мову Пролог.*

Головна ідея метода резолюцій полягає у перевірці, чи вміщує множина диз'юнктивів S *пустий диз'юнкт*, який не вміщує жодних літер. Якщо множина диз'юнктивів S вміщує *пустий диз'юнкт*, то S є суперечливим (не може бути виконаним).

Метод резолюцій полягає у тому, що ми навмисно «псуємо» нашу модель заміною існуючої цілі « p » на заперечення цілі « $\sim p$ » (створивши таким чином нову суперечливу теорію CT); далі доводиться, що нова створена теорія CT є суперечливою. Теорія є невірною, якщо в ній можна вивести протиріччя такого вигляду: $A \& \sim A$. Наявність цих двох літералів (A та $\sim A$) в одній теорії є протиріччям. Якщо вони резольвують між собою, то для правильної теорії отримаємо порожню фразу « $:-$ », яка означає, що теорія є суперечливою. Щоб із теорії CT отримати порожню фразу « $:-$ », потрібно по черзі перебрати усі необхідні фрази Хорна.

Для формул, які не є теоремами, метод резолюцій може працювати нескінченно довго і не отримується порожня фраза « $:-$ ».

Таким чином, доводять суперечливість фраз Хорна програми, введеної користувачем.

По суті метод резолюцій є методом «доведення від супротивного», бо ми доводимо, що теорія є суперечливою, коли ми прагнемо іншої мети.

В основному Пролог використовує стратегію пошуку «у глибину»: **він розглядає лише одну з альтернатив, вважаючи цей вибір вірним, доки не буде доведено зворотне**. Пошук «у глибину» дозволяє перевіряти варіанти до кінця, є простішим в оперуванні і займає менше пам'яті комп'ютера. Пошук «у ширину» Пролог використовує лише при спеціальному програмуванні.

Метод резолюцій є **переборною процедурою**, який може навіть викликати «експоненційний інформаційний вибух». Недоліком методу резолюцій є те, що більшість резольвент не потрібні. Тому на практиці використовуються модифікації принципу резолюцій, спрямовані на підвищення його ефективності (**семантична резолюція, лок-резолюція, лінійна резолюція**), які скорочують перебір за рахунок використання критеріїв відбору диз'юнкт, що беруть участь у черговій резолюції. Це робить метод резолюцій цілком придатним для практичного застосування.

При доведенні теореми (отриманні розв'язання введеної логічної задачі-гіпотези) фрази Хорна вставляються одна в іншу, і в кінцевому результаті отримується лише одна кінцева формула, яка може бути або хибною або істинною. Якщо отримана кінцева формула є хибною, то вважається, що гіпотеза-мета розділу *GOAL* є істинною (тому що Пролог використовує метод доведення «від зворотного»).

Спочатку відбувається процес політерного **співставлення** мети з головами правил або фактом. Якщо назва предиката мети в результаті співставлення політерно співпала, то починають зіставлятись аргументи предиката мети та голови правила або факт. Якщо ім'я аргумента-об'єкта

починається з маленької літери, то воно вважається відомим (константним). Для констант співставлення є успішним лише при їх політерній рівності. Вільні змінні (вони починаються з великої літери) зв'язуються або зі змінною або з константою, яка знаходиться у відповідному місці предиката, який порівнюється. Вважається, що усі змінні, які записані однако-вими літерами в правилі, є однією змінною. Але тотожно записані змінні в інших правилах або клаузах – це інші змінні. Таким чином, співставлення виконує таку ж роль, що й передача параметрів функції у процедурних мо-вах програмування (Фортран, С++, Паскаль, Бейсик і др.) [4]. Якщо **співставлення** мети з головою правила «р» є успішним, то вважається, що хвіст правила « q_1, q_2, \dots, q_n » складеться з суб-цілей, які теж належить зіста-вити та уніфікувати.

4.7. Однозначні нечіткі системи

4.7.1. Теорія однозначних нечітких систем

Чітка логіка маніпулює даними, що можуть належати до однієї з двох межових значень: бути або «не істиною 0», або – «істиною 1». *Нечітка логіка* застосовується для таких наборів даних, де ймовірність належності даних до будь-якої множини може мати значення в інтервалі «0...1».

Класична теорія нечітких систем заснована на працях вченого США *Лотфі Аскер Заде* [33]. Саме Л. А. Заде ввів лінгвістичну змінну та функцію належності – міру упевненості у логічно-лінгвістичному твер-дженні. Дякуючи роботам Л. А. Заде та іншим вченим, нечіткі системи знайшли широке застосування для розв'язання соціально-економічних, го-сподарчих, фінансових, технічних та інформаційних проблем; для оптимі-зації систем автоматизованого керування; для апроксимації складних фор-мул, графіків, табличних даних чи результатів складних розрахунків і ло-гічних висновків.

Нечіткі системи апроксимують деяку оптимальну функцію мети (оп-тимальний вихід системи). *Ця апроксимація при введенні значень вхідних числових чи логічно-лінгвістичних змінних дозволяє отримати на виході оптимальне числове чи еквівалентне йому логічно-лінгвістичне рішення.*

Теорія однозначних нечітких систем розроблена в Україні *М. В. Турти* (2007) в Національному університеті кораблебудування і є подальшим розвитком робіт Л. Заде [115]. Перевагою теорії М. В. Турти перед теорією Л. Заде є значне спрощення процесу дефазифікації (в теорії М. В. Турти дефазифікація виконується розв'язком матричного рівняння за методом найменших квадратів (МНК).

Загальні відомості про однозначні нечіткі системи [115].

Термін «*однозначність*» найбільш точно передається у перекладі англійським словом «*synonymous*» – «синоніміальність», бо в однозначній не-чіткій системі «число» та «лінгвістична змінна з функцією належності» є «синонімами» – однозначними однако-вими тлумаченнями і з взаємним од-нозначним переходом від одного поняття до іншого, чого не завжди спо-

стерігається в класичній нечіткій системі стосовно переходів від термів з функцією належності до числа.

Метою аналізу в нечіткій системі є отримання математичного опису (апроксимації) невідомої (або відомої, але занадто складної) **оптимальної функції цілі** $F(X)$, $X = \{x_1, x_2, \dots, x_j, \dots, x_n\}$, $j = 1, 2, \dots, n$.

Для цієї невідомої функції вважаються відомими **універсуми** – чіткі числові діапазони входів x_j та виходу $F(x)$. Універсуми входів та виходу помічаються назвами (у даному випадку – літерами $U_{x_1}, U_{x_2}, \dots, U_{x_j}, \dots, U_{x_n}, U_F$) і мають **позитивний напрямок осей у бік зростання чисел**. В нечіткій логіці якісний універсум має числове підґрунтя. Наприклад ‘Технічна досконалість’, ‘Якість’, ‘Надійність’, ‘Акуратність’ можуть змінюватись у межах 0...100 %. Звичайно розглядають один універсум виходу, але не забороняється розгляд кількох виходів.

Терми-числа на універсумі $U_{ВАГА} = Вага = 10 \dots 40$ (рис. 4.7.1) мають **лінгвістичні назви з мірою упевненості 100 %** («Мала», «Середня», «Велика») та відповідні однозначні числові значення: «Мала»=10, «Середня»=25, «Велика»=40.

Два терми («Мала»=10 та «Велика»=40) завжди розміщені по краях універсуму, а кількість та розміщення інших «середніх» термів довільно визначає експерт.

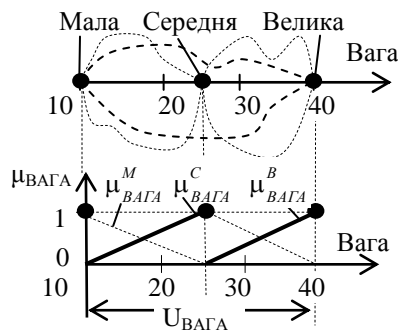


Рис. 4.7.1. Нечіткі множини, терми лінгвістичної змінної, функції упевненості

Нечітка множина належить кожному терму і складається з числових значень вздовж універсуму до сусідніх термів. Нечіткі множини взаємно перекриваються. На універсумі $U_{ВАГА} = 10 \dots 40$ ці нечіткі множини, які належать термам $Мала=10$, $Середня=25$, $Велика=40$, обведені пунктирними лініями, а самі числа-терми помічені чорними точками на функціях упевненості $\mu_{ВАГА}^M$, $\mu_{ВАГА}^C$, $\mu_{ВАГА}^B$.

Лінгвістична змінна («Вага») може набувати значень одного з термів-чисел ($Мала=10$, $Середня=25$, $Велика=40$) з деякою мірою упевненості, яка визначається по логічній функції упевненості $\mu_{ВАГА}^M$, $\mu_{ВАГА}^C$, $\mu_{ВАГА}^B$, що змінюються у межах 0...1. Таким чином, лінгвістична змінна з вказаною величиною функції упевненості є **однозначним (сінонімальним) екви-**

валентом числа універсуму. Коли логічна функція упевненості лінгвістичної змінної дорівнює «1», то змінна точно дорівнює числу-терму. Згідно рис. 4.7.1 одне числове значення змінної має два лінгвістичних еквіваленти з мірами упевненості, підсумок яких дорівнює 1.

Логічні функції упевненості $\mu_{ВАГА}^M$, $\mu_{ВАГА}^C$, $\mu_{ВАГА}^B$ для термів *Мала*=10, *Середня*=25, *Велика*=40 охоплюють відповідні нечіткі множини, мають у даному випадку трикутну форму і змінюються по прямих лініях від 1 на координаті власного терму до 0 на координаті сусіднього терму (можливе застосування і нелінійної функції упевненості).

Підсумок функцій впевненості сусідніх термів дорівнює «1» для будь-якого числового входу (наприклад, $\mu_{ВАГА}^M + \mu_{ВАГА}^C = 1$). Наведені на рис. 4.7.1 функція $\mu_{ВАГА}^M$ та частка функції $\mu_{ВАГА}^C$ для «Ваги»=10...25 дають дубльовану інформацію, і тому одна з них може не використовуватись у розрахунках. Щоб збутися дубльованої інформації урахуються лише ті частки функцій належності, які зростають у напрямку зростання універсуму (показані жирними лініями на рис. 4.7.1). В результаті зміна числового значення змінної на будь-якому універсумі супроводжується однозначною зміною його логічно-лінгвістичного еквіваленту і навпаки: досягається повна тотожність (сінонімальна однозначна взаємозалежність) між числовим та логічно-лінгвістичним значеннями змінних універсуму.

Два найближчі сусідні терми конкретного числа x^j універсуму мають назви **молодший терм** x_j^{MT} (менше сусіднє число-терм) та **старший терм** x_j^{CT} (більше сусіднє число-терм). Функція упевненості для чисел між ними розраховується за формулою

$$\mu_{x_j^A} = (x_j^A - x_j^{MT}) / (x_j^{CT} - x_j^{MT}) = 0...1, \quad (4.7.1)$$

де x_j^A – реальне значення змінної для вхідної точки A вздовж осі x_j ; x_j^{MT} , x_j^{CT} – числові значення «молодшого» та «старшого» терму числа x_j^A .

Згідно формули (4.7.1) функція впевненості вказує логічну міру впевненості віднесення заданого числа x_j^A до множини «старшого» терму x_j^{CT} .

При введених термах-числах функції упевненості можуть визначатись *автоматично* (без участі експерта, який може ввести суб'єктивну помилку), а при використанні метода найменших квадратів – **можна обійтись взагалі без поняття функції упевненості** [115; 116].

Опорними точками для деякої функції $F(X)$ зводяться вхідні точки з координатами вхідних термів-чисел [115; 116]. Для **реалізованих опорних точок** експерт вказує значення виходу, які у загальному випадку не співпадають з термами виходу. **Нереалізовані опорні точки** не мають значень виходу.

Інформаційний модуль. Для двох вхідних змінних $\{x_1, x_2\}$ на рис. 4.7.2 наведений виділений сірим фоном **інформаційний модуль** з ви-

ходами $F^0, F_{x_1}, F_{x_2}, F^m$ (виходи $F^0, F_{x_1}, F_{x_2}, F^m$ можуть не дорівнювати термам виходу) для опорних точок, заданих вхідними числами-термами x_1^{MT}, x_1^{CT} та x_2^{MT}, x_2^{CT} .

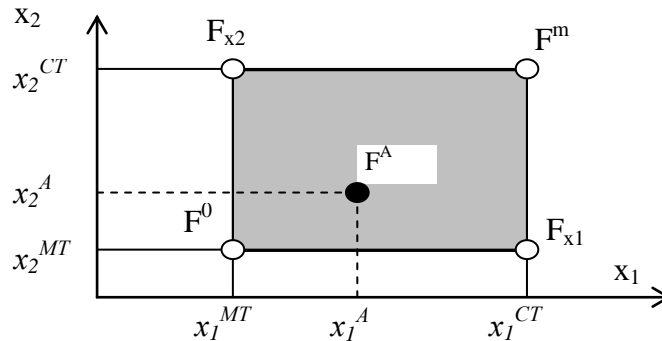


Рис. 4.7.2. Інформаційний модуль

В багатокординатній системі будь-яка точка A з вхідними координатами $\{x_1^A, x_2^A, \dots, x_n^A\}$ теж попадає між деякими значеннями «молодшого» та «старшого» термів по кожній осі універсумів. Отримані значення «молодшого» та «старшого» термів по осях універсумів створюють виокремлений **інформаційний модуль**, в межах якого знаходиться точка A .

Розрахунок виходу однозначної нечіткої системи. Розв'язок задачі за теорією однозначних нечітких систем значно спрощується у порівнянні з класичною теорією, бо можна *не урахувувати функції належності та значно спростити процес дефаззифікації*. Для цього використовується матриця спостережень табл. 4.7.1 з даними входів та виходів інформаційного модуля рис. 4.7.2 та **метод найменших квадратів** (МНК) [115; 116].

Примітка: МНК можна застосувати до довільної кількості інформаційних модулів та для всієї системи в цілому. В цьому разі отримана аналітична залежність не проходить через точки виходу F_t .

З інформаційного модуля рис. 4.7.2 ми маємо матрицю спостережень з відомими даними (табл. 4.7.1).

Таблиця 4.7.1.

Матриця спостережень інформаційного модуля

t	x_{0t}	x_{1t}	x_{2t}	F_t
1	1	x_1^{MT}	x_2^{MT}	F^0
2	1	x_1^{CT}	x_2^{MT}	F_{x_1}
3	1	x_1^{MT}	x_2^{CT}	F_{x_2}
4	1	x_1^{CT}	x_2^{CT}	F^m

Лінійна апроксимаційна формула (не інтерполяційна) має вигляд

$$F_t = \beta_0 x_{0t} + \beta_1 x_{1t} + \beta_2 x_{2t} + u_t, \quad (4.7.2)$$

де $\beta_0, \beta_1, \beta_2$ – невідомі коефіцієнти; $x_{ot} = 1 = const$ – штучно введена змінна (для зручності розрахунків); u_t – випадкова величина; t – порядковий номер спостереження.

В матричній формі математична модель (4.7.2) має вигляд

$$F = XB + U, \quad (4.7.3),$$

де

$$F = \begin{bmatrix} F^0 \\ F_{x_1} \\ F_{x_2} \\ F^m \end{bmatrix}; \quad X = \begin{bmatrix} 1 & x_1^{MT} & x_2^{MT} \\ 1 & x_1^{CT} & x_2^{MT} \\ 1 & x_1^{MT} & x_2^{CT} \\ 1 & x_1^{CT} & x_2^{CT} \end{bmatrix}; \quad B = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}; \quad U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}.$$

Для розв'язання задачі ми розглядаємо рівняння (4.7.3) як його математичне очікування. Із статистичного співвідношення (тут E – математичне очікування; $E(U) = 0$) $E(F) = E(XB + U) = E(XB)$, отримуємо

$$F = XB; \quad X^T F = (X^T X)B;$$

$$(X^T X)^{-1} X^T F = (X^T X)^{-1} (X^T X)B; \quad B = (X^T X)^{-1} X^T F, \quad (4.7.4)$$

де $(X^T X)$ – квадратна матриця, симетрична відносно головної діагоналі; $(X^T X)^{-1} (X^T X)$ – одинична матриця (усі елементи дорівнюють нулю, а діагональні елементи дорівнюють одиниці), яку можна опустити.

Як бачимо, у даному випадку не використовується процес дефазифікації у традиційному його вигляді і не використовуються функції належності [115].

Для визначення виходу точки A в середині багатомірного простору інформаційного модуля можна також застосувати інтерполяційну апроксимуючу формулу [115]

$$y^A = y^0 + \sum_{j=1}^n (y^j - y^0) \mu_{x_j} + [y^m + (n-1)y^0 + \sum_{j=1}^n y^j] \prod_{j=1}^n \mu_{x_j}, \quad (4.7.5)$$

де $j = 1, 2, \dots, n$ – порядковий номер вхідних змінних; $X^A = \{x_1^A, x_2^A, \dots, x_n^A\}$ – вектор вхідних координат досліджуваної точки A в інформаційному модулі; $y^0 \{x_1^{MT}, x_2^{MT}, \dots, x_n^{MT}\}$ – значення виходу y^0 в опорній точці, у якій кожна координата входу $\{x_1, x_2, \dots, x_n\}$ дорівнює найменшому числовому значенню відповідного вхідного терму інформаційного модуля (**молодшому терму**); $y^m \{x_1^{CT}, x_2^{CT}, \dots, x_n^{CT}\}$ – значення виходу y^m в опорній точці, у якій кожна координата входу $\{x_1, x_2, \dots, x_n\}$ дорівнює найбільшому числовому значенню відповідного вхідного терму інформаційного модуля (**старшому терму**); $y^1 = f\{x_1^{CT}, x_2^{MT}, \dots, x_j^{MT}, \dots, x_n^{MT}\}$ – числовий вихід для осьової опорної точки-терму по осі x_1 ; $y^2 = f\{x_1^{MT}, x_2^{CT}, \dots, x_j^{MT}, \dots, x_n^{MT}\}$ – числовий вихід для осьової опорної точки-терму по осі x_2 ; $y^j = f\{x_1^{MT}, x_2^{MT}, \dots, x_j^{CT}, \dots, x_n^{MT}\}$ – числовий вихід для осьової опорної точки-терму по осі x_j ; $y^n = f\{x_1^{MT}, x_2^{MT}, \dots, x_j^{MT}, \dots, x_n^{CT}\}$ – числовий вихід для осьової опорної точки-терму по осі x_n ;

$\mu_{x_j} = (x_j^A - x_j^{MT}) / (x_j^{CT} - x_j^{MT}) = 0 \dots 1$ – логічна функція упевненості щодо віднесення вхідного значення змінної x_j^A до найближчого старшого терму x_j^{CT} по даній вхідній осі x_j .

Для двохмірного простору вихід для заданої точки A в середині інформаційного модуля рис. 4.7.4 з вхідними координатами x_1^A, x_2^A може бути розрахований за апроксимуючою інтерполяційною формулою

$$F^A = F^0 + (F_{x_1} - F^0)\mu_{x_1}^A + (F_{x_2} - F^0)\mu_{x_2}^A + [F^m + F^0 - (F_{x_1} + F_{x_2})]\mu_{x_1}^A\mu_{x_2}^A \quad (4.7.6).$$

4.7.2. Переваги однозначних нечітких систем

Однозначна нечітка система дозволяє розв'язати ті ж самі задачі, що й класична теорія Л. А. Заде, але має ряд переваг:

- значно спрощуються розрахунки;
- підвищується швидкодія системи за рахунок відмови від обробки кожного сигналу входу усіма правилами бази знань та спрощення дефазифікації (аж до практичної відмови від дефазифікації);
- створена можливість визначення числового виходу без використання функцій належності – шляхом застосування метода найменших квадратів;
- обмежується суб'єктивний вплив експерта на створення математичної моделі, тому що форму функції належності можна навіть не згадувати;
- усуваються логічні вагові коефіцієнти для правил бази знань;
- виправляються логічні похибки; збільшується точність розрахунків внаслідок однозначного точного переходу від лінгвістичного твердження з мірою впевненості до числа;
- замінюються на одну рівноцінну модель дві моделі класичної теорії нечітких систем: Мамдані (з логічно-лінгвістичним визначенням виходу) та Такагі-Сугено (з аналітичним визначенням виходу) [88; 101; 104; 124; 196; 197].

Деякі приклади логічних неузгодженостей класичної теорії нечіткої логіки, які позначаються на точності розрахунків і які усунені в теорії однозначних нечітких систем наведені нижче [115]:

1. З рис. 4.7.3, взятого з роботи [33, с. 16, рис. 1.5, а], можна зробити висновок, що *підсумкова* впевненість в істинності функції упевненості термів «Малий» та «Дуже малий» складають приблизно 2 (у той час, як «Істина» змінюються в межах «0...1»); теж саме відноситься і до термів «Великий» та «Дуже великий».

Аналогічні висновки можна зробити стосовно даних з книжки [104, с. 54, рис. 3.10].

Слова «нечітка логіка» ще не означають, що в ній можна вводити логічні помилки без їх пояснення та обґрунтування необхідності.

2. В класичній теорії може не існувати чіткого однозначного переходу від лінгвістичного значення змінної (з функцією належності) до її числової

величини. Наприклад, можна бути «Молодим» з мірою упевненості 100 % у межах 0...10 років, в той час як відомо, що людина починає старіти з дня народження. Такі дані для технічних пристрійів означають, що лінгвістичне твердження «Малий струм збудження» з мірою упевненості у 100 % може дати похибку числового виходу у межах 0–10 % від номінального значення струму.

3. З рис. 4.7.4, взятого з роботи [33, с. 42, рис. 4.1, формула (4.2)] (є аналогічні публікації [98, с. 288 рис. 4.3]) випливає, що на деякій ділянці числове значення може не мати лінгвістичного тлумачення. Замість **обов'язковості застосування функції упевненості** для всіх числових субмножин універсуму, пропущена функція упевненості ураховується штучним застосуванням декартового добутку у вигляді графічної залежності « \wedge », хоча декартовий добуток маже мати і іншу графічну залежність, наприклад, « \vee » або « \times ». Це лише ускладнює аналіз.

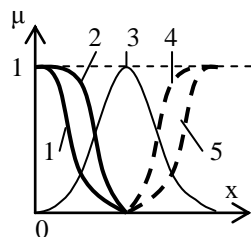


Рис. 4.7.3. Функції впевненості для термів:
1 – Дуже малий; 2 – Малий;
3 – Не дуже малий і не дуже великий;
4 – Великий; 5 – Дуже великий.

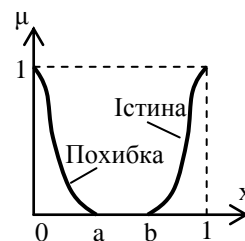


Рис. 4.7.4. Функції впевненості для термів «Істина» та «Похибка».

4. Нечіткість у визначенні коефіцієнтів упевненості, їх взаємна нелінійна залежність, існуюча велика кількість методів дефазифікації (результати виходів яких не співпадають) вносять невпевненість в опис роботи класичної нечіткої системи щодо точності розрахунків.

Внаслідок складності аналізу (особливо процесу дефазифікації) в Matlab створене спеціальне середовище, яке засноване на теорії Л. Заде [2].

4.7.3. Чайові в ресторані

На основі теорії однозначних нечітких систем М. В. Турти [115] розглянемо приклад, який наводиться в *Matlab* для пояснення роботи класичної нечіткої системи Л. Заде. Вважаємо, що **чайові F в ресторані** (універсум $U_F=0...100$ %, лінгвістична змінна «Чайові», терми-числа $F^M=0$ %; $F^C=50$ %; $F^B=100$ %) залежать від (рис. 4.7.5):

– **якості страв** x_1 (універсум $U_{x1}=0...100$ %; лінгвістична змінна «Страви»; терми-числа $x_1^M=0$ %, $x_1^C=50$ %, $x_1^B=100$ %);

– **якості обслуговування** x_2 (універсум $U_{x2}=0...100$ %; лінгвістична змінна «Обсл»; терми-числа $x_2^M=0$ %, $x_2^C=50$ %, $x_2^B=100$ %).

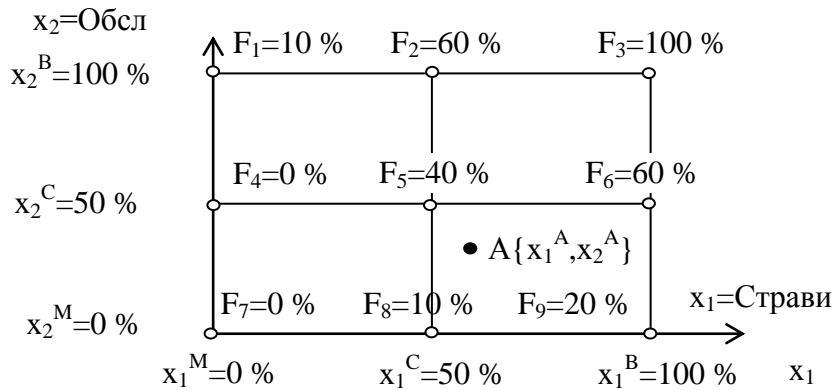


Рис. 4.7.5. Реалізовані опорні точки нечіткої системи визначення чайових

Експерт визначає по двох осях координат (x_1, x_2) терми [«Страви (якість страв)» = (x_1^M, x_1^C, x_1^B) ; «Обсл (якість обслуговування)» = (x_2^M, x_2^C, x_2^B)], які ділять весь площину ознак (x_1, x_2) на чотири інформаційних модуля прямокутної форми. Точки пересічення термів зветься **опорними точками**: вони позначені на рис. 4.7.5 білими колами. Для всіх опорних точок експерт записує значення чайових $F_1 - F_9$, керуючись наступними розмірковуваннями:

– $F_7 = 0\%$ (тому що якість страв $x_1^M=0\%$ та якість обслуговування $x_2^M=0\%$);

– $F_3 = 100\%$ (тому що якість страв $x_1^B=100\%$ та якість обслуговування $x_2^B=100\%$);

– $F_5 = 40\%$ (тому що точка F_5 знаходиться «посередині» між точками F_7 та F_3). Ці дані суб'єктивно визначаються експертом;

– $F_9 = 20\%$ (тому що так суб'єктивно вважає експерт для обслуговування $x_1^B=100\%$ та $x_2^M=0\%$);

– $F_8 = 10\%$ (тому що F_8 знаходиться посередині між точками F_7 та F_9);

– $F_6 = 60\%$ (згідно розрахунку, при лінійній зміні значення F від $F_9=20\%$ до $F_3=100\%$).

– $F_1 = 10\%$ (тому що так суб'єктивно вважає експерт для обслуговування $x_1^M=0\%$ та $x_2^B=100\%$);

– $F_4 = 10\%$ (тому що так суб'єктивно вважає експерт для обслуговування $x_1^M=0\%$ та $x_2^C=50\%$);

– $F_2 = 60\%$ (тому що так суб'єктивно вважає експерт при зміні значення F від $F_1=10\%$ до $F_3=100\%$).

Сукупність отриманих точок складають базу незалежних даних нечіткої системи. Цю базу даних можна відобразити як сукупність незалежних продукційних правил

if $x_1 = 0\%$ and $x_2 = 0\%$ then $F = 0\%$.

if $x_1 = 50\%$ and $x_2 = 0\%$ then $F = 10\%$.

if $x_1 = 100\%$ and $x_2 = 100\%$ then $F = 100\%$.

З інформаційного модуля рис. 4.7.5, обмеженого значеннями вихідної функції ($F_5=40\%$; $F_6=60\%$; $F_8=10\%$; $F_9=20\%$) ми маємо матрицю спостережень, яка використовується в методі найменших квадратів (МНК) з відомими даними (табл. 4.7.2).

Таблиця 4.7.2.

Матриця спостережень одного інформаційного модуля для використання МНК (тут t – порядковий номер спостереження)

t	x_0	x_1	x_2	F
1	1	$x_1=50\%$	$x_2=50\%$	$F_5=40\%$
2	1	$x_1=100\%$	$x_2=50\%$	$F_6=60\%$
3	1	$x_1=50\%$	$x_2=0\%$	$F_8=10\%$
4	1	$x_1=100\%$	$x_2=0\%$	$F_9=20\%$

Лінійна апроксимаційна формула «інформаційного модуля з координатами ($x_1=50\dots 100\%$; $x_2=0\dots 50\%$)» має вигляд $F = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2$, де $\beta_0, \beta_1, \beta_2$ – невідомі коефіцієнти; $x_0 = 1 = const$ – штучно введена змінна (для зручності розрахунків).

Розв’язок матричного рівняння для «інформаційного модуля з координатами ($x_1 = 50\dots 100\%$; $x_2 = 0\dots 50\%$)» $F = XB$,

$$F = \begin{bmatrix} F_5 = 40 \\ F_6 = 60 \\ F_8 = 10 \\ F_9 = 20 \end{bmatrix}; \quad X = \begin{bmatrix} 1 & x_1 = 50 & x_2 = 50 \\ 1 & x_1 = 100 & x_2 = 50 \\ 1 & x_1 = 50 & x_2 = 0 \\ 1 & x_1 = 100 & x_2 = 0 \end{bmatrix}; \quad B = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix};$$

де F, X – вектор вихідних значень та матриця змінних (визначаються по матриці спостережень); B – вектор коефіцієнтів, отримується в MathCAD (або MatLab) за методом найменших квадратів (МНК) у вигляді

$$B = (X^T X)^{-1} X^T F = \begin{bmatrix} -7,5 \\ 0,3 \\ 0,7 \end{bmatrix}.$$

звідки для точки $A\{x_1^A = 60\%; x_2^A = 30\%\}$ розрахунок дає значення $F = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 = -7,5 + 0,3x_1 + 0,7x_2 = -7,5 + 0,3 \cdot 60 + 0,7 \cdot 30 = 31,5\%$.

Як бачимо, у даному випадку не використовується процес дефазифікації у традиційному його вигляді. а функції належності навіть не згадуються [115].

Однозначні нечіткі системи мають наступні переваги:

- немає потреби у разі заміни одного продукційного правила переглядати усі правила бази знань сумісно з їх функціями належності;
- по числовим значенням входів розраховується числове значення виходу. Перехід від числових до лінгвістичних позначень чисел виконується на універсумах;

– при визначення виходу немає потреби розглядати усю сукупність правил бази (розглядаються лише правила інформаційного модуля), що збільшує швидкодію нечіткої системи і спрощує розрахунки;

– немає потреби вводити в базу даних для кожного правила додатковий логічний ваговий коефіцієнт, значення якого береться у межах $0 \dots 1$ і який ураховує вплив даного правила на кінцевий результат (це зменшує суб'єктивний вплив експерта на результати дослідження), бо самі правила бази даних чітко і однозначно відображують знання експерта відносно системи, яка моделюється.

Будь яка вхідна точка $A_{\{x_1^A = 60\%; x_2^A = 30\%\}}$ попадає у відповідний **інформаційний модуль** ІМ (рис. 4.7.5). Вихід однозначної нечіткої системи розраховується за інтерполяційною формулою (4.7.5)

$$F^A = F^0 + (F_{x_1} - F^0)\mu_{x_1}^A + (F_{x_2} - F^0)\mu_{x_2}^A + [F^m + F^0 - (F_{x_1} + F_{x_2})]\mu_{x_1}^A\mu_{x_2}^A =$$

$$= 10 \% + (20 \% - 10 \%)\cdot 0,2 + (40 \% - 10 \%)\cdot 0,6 + [60 \% + 10 \% - (20 \% + 40 \%)] 0,2\cdot 0,6 = 30 \%,$$

де $\mu_{x_1}^A = (x_1^A - x_1^{MT}) / (x_1^{CT} - x_1^{MT}) = (60 \% - 50 \%) / (100 \% - 50 \%) = 0,2$ – функція упевненості віднесення координати x_1^A до нечіткої субмножини старшого терму універсуму U_{x_1} по осі x_1 ;

$\mu_{x_2}^A = (x_2^A - x_2^{MT}) / (x_2^{CT} - x_2^{MT}) = (30 \% - 0 \%) / (50 \% - 0 \%) = 0,6$ – функція упевненості віднесення координати x_2^A до нечіткої множини старшого терму універсуму U_{x_2} по осі x_2 ; $x_1^{MT}, x_1^{CT}, x_2^{MT}, x_2^{CT}$ – входи ІМ; $F^0, F_{x_1}, F_{x_2}, F^m$ – виходи ІМ.

Таким чином, при якості страв $x_1^A = 60 \%$ та якості обслуговування $x_2^A = 30 \%$, ми повинні дати чайові у розмірі $F^A = 30 \%$, значення яких знаходиться у межах, визначених інформаційним модулем.

Завдання

Розробити власну однозначну нечітку систему (можна використовувати дані з розміщеного нижче розділу «Приклади завдань») з кількісними, або якісними ознаками з універсумами ($x_1=0 \dots N$; $x_2=0 \dots N$), де N – порядковий номер студента у групі. Універсум виходу $F = 0 \dots N$ (але вихід можна задавати і у вигляді індивідуальної довільної формули $F(x_1, x_2)$). Кожний універсум (F, x_1, x_2) розбити мінімум на дві частки.

Для всіх інформаційних модулів отримати розв'язки за МНК.

Для одного інформаційного модуля:

– за інтерполяційною формулою отримати додатково значення вихідної функції в опорних точках та для однієї точки всередині інформаційного модуля;

– порівняти між собою у відсотках точність розрахованих даних по формулі МНК та по інтерполяційній формулі (у вигляді відношення) для усіх опорних точок та для однієї точки всередині даного інформаційного модуля;

– по розрахованим за МНК даним визначити лінгвістичні еквіваленти з відповідними значеннями функції належності для всіх числових значень входів та виходів в опорних точках та в точці всередині даного інформаційного модуля.

Приклади завдань.

1. Створити однозначну нечітку систему з визначення оцінки студента ($y=2..5$) у залежності від терміну праці студента над матеріалами лекцій ($x_1 = 0...(20+0,5N)$ годин). N – порядковий номер студента у групі.

2. Розрахувати споживання пива у залежності від коливання температури повітря в червні місяці та середнього заробітку робітника у порівнянні з вартістю прожиткового мінімуму.

3. Розрахувати залежність швидкості судна від швидкості вітру (x_1) та стану моря (x_2).

4. Отримати (по наданій базі знань у вигляді таблиці) аналітичну залежність однозначної нечіткої системи регулятора збудження синхронного генератора $u_{ЗБ} = f(x_1, x_2)$, де x_1 – струм, який віддається генератором в мережу; x_2 – коефіцієнт потужності $\cos \varphi_1$; N – порядковий номер студента у групі.

$x_2 \backslash x_1$	0	50 N	100 N
0	30	60	100
0,5	30	50	80
1	30	40	60

5. Розрахувати струм виходу сонячної батареї у залежності від якості сонця та площі батареї.

6. Розрахувати ціни продукції у залежності від ціни на сировину та ціни на паливо.

7. Розрахувати залежність ефективності роботи від заробітної плати та кваліфікації.

8. Розрахувати залежність витікання рідини від рівня рідини в ємності та положення заслонки.

9. Розрахувати швидкість переміщення вантажу від стану машини та професійності водія.

10. Розрахувати залежність величини переміщення заслінки подачі пару від температури у приміщенні та зовнішньої температури.

11. Розрахувати зріст рослин у залежності від інтенсивності поливу та температури.

12. Скласти програму визначення траєкторії покрокового переміщення рухомого об'єкту (літаючого апарату, судна, підводного човна, автономного робота тощо) за допомогою застосування однозначних нечітких

множин у залежності від координат початкової та цільової точки, координат перешкод та даних «радарного датчика», який вказує кути обходу найближчої перешкоди з використанням двох вхідних нечітких множин: напрямку на ціль (x_1); кути обходу перешкод (x_2).

4.7.4. Оплата працівника

Однозначна нечітка система [115] повинна апроксимувати функцію $y=Y(x_1, x_2)$, де y – оплата робітника, x_1 – час виконання роботи, x_2 – акуратність виконання роботи. Оптимальну оплату робітника повинен визначити експерт: чим швидше та акуратніше виконана робота, тим вище оплата робітника. Експерт визначає універсуми, лінгвістичні змінні та терми згідно даних табл. 4.7.3.

Таблиця 4.7.3.

Універсуми, лінгвістичні змінні та терми однозначної нечіткої системи для визначення оплати робітника

Змінна	Назва	Універсум	Лінгвістична змінна	Терми – числа (верхні індекси: М – мале; С – середнє; В – велике)
y	Оплата праці	$U_y=0\dots 1$	Плата	$y^M = 0,1; y^C = 0,5; y^B = 1,0$
x_1	Час виконання роботи	$U_{x_1}=1\dots 5$ діб	Час	$x_1^M = 1; x_1^C = 3; x_1^B = 5$
x_2	Акуратність виконання роботи	$U_{x_2}=0\dots 100\%$	Ак	$x_2^M = 0\%; x_2^C = 50\%; x_2^B = 100\%$

Експерт суб'єктивно визначає виходи $y_1 - y_9$ в позначених колами опорних точках (рис. 4.7.6, функції належності не показані).

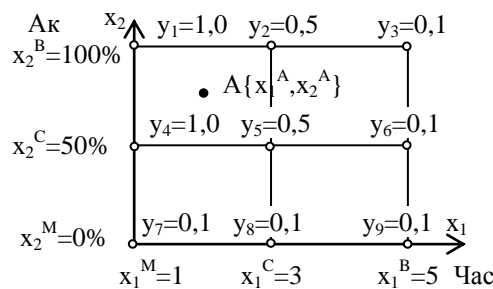


Рис. 4.7.6. Реалізовані опорні точки нечіткої системи для визначення оплати робітника

База даних може бути представлена сукупністю продукційних правил

- if $x_1 = 1$ and $x_2 = 0\%$ then $y = 0,1$.*
- if $x_1 = 3$ and $x_2 = 0\%$ then $y = 0,1$.*
- if $x_1 = 5$ and $x_2 = 100\%$ then $y = 0,1$.*

Лінійна апроксимаційна формула «інформаційного модуля з координатами, наприклад, ($x_1=1\dots3$; $x_2=50\dots100\%$)» має вигляд

$$Y = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2,$$

де $\beta_0, \beta_1, \beta_2$ – невідомі коефіцієнти; $x_0 = 1 = const$ – штучно введена змінна (для зручності розрахунків); e – порядковий номер «інформаційного модуля. Розв’язок матричного рівняння для «інформаційного модуля з координатами ($x_1=1\dots3$; $x_2=50\dots100\%$)» для точки $A\{x_1^A = 2 \text{ доби}; x_2^A = 75\%\}$ $Y = XB$,

$$\text{де } Y = \begin{bmatrix} y_1 = 1 \\ y_2 = 0,5 \\ y_4 = 1 \\ y_5 = 0,5 \end{bmatrix}; \quad X = \begin{bmatrix} 1 & x_1 = 1 & x_2 = 100 \\ 1 & x_1 = 3 & x_2 = 100 \\ 1 & x_1 = 1 & x_2 = 50 \\ 1 & x_1 = 3 & x_2 = 50 \end{bmatrix}; \quad B = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix};$$

Y, X – вектор вихідних значень та матриця змінних (визначаються по матриці спостережень); B – вектор коефіцієнтів, отримується в MathCAD (або MatLab) за методом найменших квадратів (МНК) у вигляді

$$B = (X^T X)^{-1} X^T F = \begin{bmatrix} 1,25 \\ -0,25 \\ 0 \end{bmatrix},$$

звідки $Y = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 = 1,25 \cdot 1 - 0,25 \cdot 2 + 0 \cdot 75 = 0,75$.

Для заданої вхідної точки $A\{x_1^A = 2 \text{ доби}; x_2^A = 75\%\}$ вихід повинен знаходитись у межах $y^A = 0,5 \dots 1,0$: за МНК вихід дорівнює $y^A = 0,75$ і за інтерполяційною формулою (4.7.5) теж дорівнює

$y^A = y^0 + (y_{x1} - y^0)\mu_{x1}^A + (y_{x2} - y^0)\mu_{x2}^A + [y^m + y^0 - (y_{x1} + y_{x2})]\mu_{x1}^A\mu_{x2}^A = 1 + (0,5 - 1,0) \cdot 0,5 + (1,0 - 1,0) \cdot 0,5 + [0,5 + 1,0 - (0,5 + 1,0)] \cdot 0,5 \cdot 0,5 = 0,75$, де $\mu_{x1}^A = (x_1^A - x_1^{MT}) / (x_1^{CT} - x_1^{MT}) = (2 - 1) / (3 - 1) = 0,5$ – функція упевненості відношення координати x_1^A до нечіткої множини універсуму U_{x1} по осі x_1 ; $\mu_{x2}^A = (x_2^A - x_2^{MT}) / (x_2^{CT} - x_2^{MT}) = (75\% - 50\%) / (100\% - 50\%) = 0,5$ – функція упевненості відношення координати x_2^A до нечіткої множини універсуму U_{x2} по осі x_2 .

4.7.5. Апроксимація «невідомої» функції $F = x_1 \sin x_2$

Введені в табл. 4.7.4 дані визначає експерт. Змінна x_2 наведена в радіанах.

Таблиця 4.7.4.

Визначені експертом дані однозначної нечіткої системи

Змінна	Універсум	Лінгвістична змінна	Терми
F	$U_F = 0...10$	ВихідF	$F^M = 0; F^C = 5; F^B = 10$
x_1	$U_{x_1} = 0...10$	ВхідX1	$x_1^M = 0; x_1^C = 5; x_1^B = 10$
x_2	$U_{x_2} = 0..6,28$	ВхідX2	$x_2^M = 0; x_2^{C1} = 1,57; x_2^{C2} = 3,14;$ $x_2^{C3} = 4,71; x_2^B = 6,28$

Для *реалізованих опорних точок* (на рис. 4.7.7 вони показані білими колами) експерт згідно формули $F = x_1 \sin x_2$ розраховує значення виходу $F_1 - F_{15}$ [115].

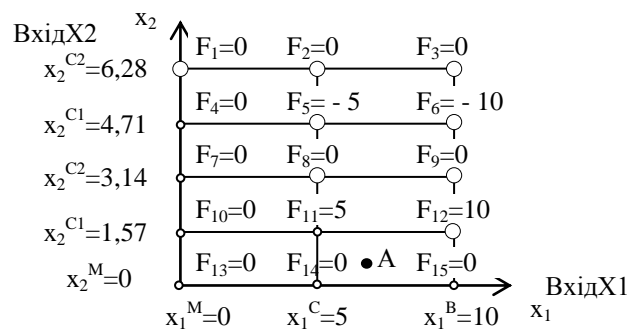


Рис. 4.7.7. Реалізовані опорні точки нечіткої системи визначення «невідомої» функції $F = x_1 \sin x_2$; $A\{x_1^A = 7; x_2^A = 0,7\}$

З рис. 4.7.7 випливає, що ми можемо обрати такі опорні точки, що всі вихідні значення F будуть дорівнювати нулю і тоді вихідні функції будуть невірні розраховані – всі будуть рівними нулю (цей недолік відноситься й до продукційних правил класичної теорії Л. Заде). Тому очевидно, що експерт повинен обирати для опорних точок такі дані, щоб вони охоплювали як максимуми так і мінімуми функції виходу.

Правила бази даних однозначної нечіткої системи складаються на основі даних реалізованих опорних точок рис. 4.7.7 і можуть бути представлені вигляді сукупності окремих не пов'язаних між собою продукційних правил.

$$\text{if } x_1 = 0 \text{ and } x_2 = 6,28 \text{ then } F_1 = 0;$$

$$\text{if } x_1 = 5 \text{ and } x_2 = 6,28 \text{ then } F_2 = 0;$$

$$\text{if } x_1 = 10 \text{ and } x_2 = 0 \text{ then } F_{15} = 0.$$

Лінійна апроксимаційна формула «інформаційного модуля з координатами, наприклад, $(x_1=5...10; x_2=0...1,57)$ » має вигляд

$$F = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2,$$

де $\beta_0, \beta_1, \beta_2$ – невідомі коефіцієнти; $x_0 = 1 = const$ – штучно введена змінна (для зручності розрахунків).

Розв’язок матричного рівняння для «інформаційного модуля з координатами ($x_1=5\dots 10$; $x_2=0\dots 1,57$)» $F = XB$,

$$F = \begin{bmatrix} F_{11} = 5 \\ F_{12} = 10 \\ F_{14} = 0 \\ F_{15} = 0 \end{bmatrix}; \quad X = \begin{bmatrix} 1 & x_1 = 5 & x_2 = 1,57 \\ 1 & x_1 = 10 & x_2 = 1,57 \\ 1 & x_1 = 5 & x_2 = 0 \\ 1 & x_1 = 10 & x_2 = 0 \end{bmatrix}; \quad B = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix};$$

де F, X – вектор вихідних значень та матриця змінних (визначаються по матриці спостережень); B – вектор коефіцієнтів,

отримується в MathCAD (або MatLab) за методом найменших квадратів (МНК) у вигляді

$$B = (X^T X)^{-1} X^T F = \begin{bmatrix} -3,75 \\ 0,5 \\ 4,777 \end{bmatrix},$$

звідки $F = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 = -3,75 \cdot 1 + 0,5 \cdot 7 + 4,777 \cdot 0,7 = 3,09$.

Задана вхідна точка $A\{x_1^A = 7; x_2^A = 0,7\}$ попадає між сусідніми реалізованими опорними точками, які у сукупності складають для даної точки **інформаційний модуль**. Вихід однозначної нечіткої системи за МНК дорівнює $F^A = 3,09$, а за інтерполяційною формулою

$$F^A = F^0 + (F_{x_1} - F^0)\mu_{x_1}^A + (F_{x_2} - F^0)\mu_{x_2}^A + [F^m + F^0 - (F_{x_1} + F_{x_2})]\mu_{x_1}^A \mu_{x_2}^A = 0 + (0 - 0) \cdot 0,4 + (5 - 0) \cdot 0,445 + [10 + 0 - (0 + 5)] \cdot 0,4 \cdot 0,445 = 3,04,$$

де $\mu_{x_1}^A = (x_1^A - x_1^{MT}) / (x_1^{CT} - x_1^{MT}) = (7 - 5) / (10 - 5) = 0,4$ – функція упевненості координати x_1^A до нечіткої множини універсуму U_{x_1} по осі x_1 ;

$\mu_{x_2}^A = (x_2^A - x_2^{MT}) / (x_2^{CT} - x_2^{MT}) = (0,7 - 0) / (1,57 - 0) = 0,445$ – функція упевненості координати x_2^A до нечіткої множини універсуму U_{x_2} по осі x_2 ;

$x_1^{MT}, x_1^{CT}, x_2^{MT}, x_2^{CT}$ – позначення входів ІМ; $F^0, F_{x_1}, F_{x_2}, F^m$ – позначення виходів ІМ.

Експерт перевіряє точність апроксимації (дійсне значення $F = x_1 \sin x_2 = 7 \cdot \sin(0,7) = 4,5$, а отримане значення $F = 3,04$) і, якщо отримана точність розрахунків його не задовольняє, то він обирає більшу кількість реалізованих опорних точок. У даному випадку отримана достатньо велика похибка. Але якщо обрати точку $B\{x_1^B = 10; x_2^B = 1,57\}$, то отримуємо точну відповідь – $F^B = 10$: це пояснюється тим, що обрана інтерполяційна форма апроксимації.

Завдання

Розробити власну однозначну нечітку систему з числовими ознаками для формули $y = N \sin(x_1) \sin(x_2)$, де N – порядковий номер студента у групі;

$x_1 = x_2 = 0^0, 90^0, 180^0, 270^0, 360^0$, яка грає роль експерта по визначенню даних опорних точок та при перевірці результатів розрахунків. Для всіх числових значень входів та виходу точки всередині одного з інформаційних модулів визначити лінгвістичні еквіваленти з відповідними значеннями функції належності.

4.7.6. Соціальний стан на підприємстві

Вважаємо, що *соціальний стан F на підприємстві* (універсум $U_F = 0 \dots 100\%$, лінгвістична змінна «СоцСтан», терми-числа $F^M = 0\%$; $F^C = 50\%$; $F^B = 100\%$) залежать від:

– *заробітної плати x_1* (універсум $U_{x_1} = 0 \dots 100\%$, лінгвістична змінна «Зарплата», терми-числа $x_1^M = 0\%$; $x_1^C = 50\%$; $x_1^B = 100\%$);

– *цін на ринку x_2* (універсум $U_{x_2} = 0 \dots 100\%$, лінгвістична змінна «Ціни», терми-числа $x_2^M = 0\%$; $x_2^C = 50\%$; $x_2^B = 100\%$).

Експерт визначає базу даних у вигляді реалізованих опорних точок рис. 4.7.8 [115]. Ця база даних має вигляд сукупності продукційних правил

if $x_1 = 0\%$ and $x_2 = 0\%$ then $F_7 = 0\%$.

if $x_1 = 50\%$ and $x_2 = 0\%$ then $F_8 = 50\%$.

if $x_1 = 100\%$ and $x_2 = 100\%$ then $F_3 = 40\%$.

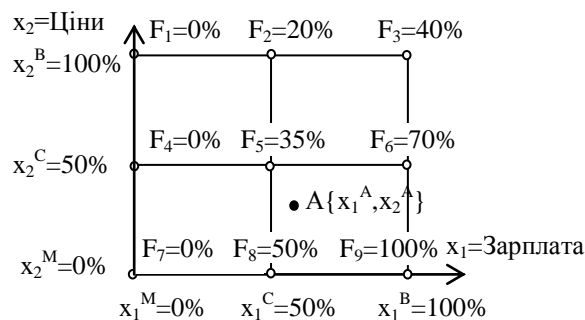


Рис. 4.7.8. Реалізовані опорні точки системи визначення соціального стану $A\{x_1^A = 60\%; x_2^A = 30\%\}$

Лінійна апроксимаційна формула «інформаційного модуля з координатами, наприклад, $(x_1 = 50 \dots 100\%; x_2 = 0 \dots 50\%)$ » має вигляд

$$F = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2,$$

де $\beta_0, \beta_1, \beta_2$ – невідомі коефіцієнти; $x_0 = 1 = const$ – штучно введена змінна (для зручності розрахунків).

Розв'язок матричного рівняння для «інформаційного модуля з координатами $A\{x_1^A = 60\%; x_2^A = 30\%\}$ » $F = XB$,

$$F = \begin{bmatrix} F_5 = 35 \\ F_6 = 70 \\ F_8 = 50 \\ F_9 = 100 \end{bmatrix}; \quad X = \begin{bmatrix} 1 & x_1 = 50 & x_2 = 50 \\ 1 & x_1 = 100 & x_2 = 50 \\ 1 & x_1 = 50 & x_2 = 0 \\ 1 & x_1 = 100 & x_2 = 0 \end{bmatrix}; \quad B = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix};$$

де F , X – вектор вихідних значень та матриця змінних (визначаються по матриці спостережень); B – вектор коефіцієнтів, отримується в MathCAD (або MatLab) за методом найменших квадратів (МНК) у вигляді

$$B = (X^T X)^{-1} X^T F = \begin{bmatrix} 11,25 \\ 0,85 \\ -0,45 \end{bmatrix}.$$

звідки $F = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 = 11,25 \cdot 1 + 0,85 \cdot 60 - 0,45 \cdot 30 = 48,75\%$.

Вхідна точка $A\{x_1^A = 60\%; x_2^A = 30\%\}$ попадає між сусідніми реалізованими опорними точками, які у сукупності складають для даної точки **інформаційний модуль** ІМ (рис. 4.7.8). Вихід однозначної нечіткої системи за МНК дорівнює $F^A = 48,75$, а за інтерполяційною формулою

$$\begin{aligned} F^A &= F^0 + (F_{x_1} - F^0)\mu_{x_1}^A + (F_{x_2} - F^0)\mu_{x_2}^A + [F^m + F^0 - (F_{x_1} + F_{x_2})]\mu_{x_1}^A \mu_{x_2}^A = \\ &= 50 \% + (100 \% - 50 \%)\cdot 0,2 + (35 \% - 50 \%)\cdot 0,6 + \\ &+ [70 \% + 50 \% - (100 \% + 35 \%)] \cdot 0,2 \cdot 0,6 = 49,2 \%, \end{aligned}$$

де $\mu_{x_1}^A = (x_1^A - x_1^{MT}) / (x_1^{CT} - x_1^{MT}) = (60 \% - 50 \%) / (100 \% - 50 \%) = 0,2$ – функція упевненості віднесення координати x_1^A до нечіткої множини універсуму U_{x_1} по осі x_1 ; $\mu_{x_2}^A = (x_2^A - x_2^{MT}) / (x_2^{CT} - x_2^{MT}) = (30 \% - 0 \%) / (50 \% - 0 \%) = 0,6$ – функція упевненості віднесення координати x_2^A до нечіткої множини універсуму U_{x_2} по осі x_2 ; $x_1^{MT}, x_1^{CT}, x_2^{MT}, x_2^{CT}$ – позначення входів ІМ; $F^0, F_{x_1}, F_{x_2}, F^m$ – позначення виходів ІМ.

Таким чином, при зарплаті $x_1^A = 60\%$ та цінах на ринку $x_2^A = 30\%$ соціальної стан дорівнює $F^A = 49,2\%$.

4.7.7. Математична апроксимація графіка

Припустимо, що фірма – виробник пива має графічні залежності об'єму попиту пива за місяць Q від температури повітря T при загальних значеннях цін на ринку S (рис. 4.7.9). Потрібно визначити об'єм виробництва пива Q у залежності від прогнозу температури T та прогнозу цін на ринку S згідно теорії [115].

По даних рис. 4.7.9 визначаємо універсуми, лінгвістичні змінні та терми (табл. 4.7.5).

Таблиця 4.7.5.

Універсуми, лінгвістичні змінні та терми нечіткої аналітичної функції

Змінна	Назва	Універсум	Лінгвістична змінна	Терми-числа
Q	Попит	$U_Q=0\dots 2$	Попит	$Q^M=0; Q^C=1; Q^B=2$
T	Температура	$U_T=0\dots 30^{\circ}\text{C}$	Темп	$T^M=0; T^C=15; T^B=30$
S	Ціни на ринку	$U_S=1\dots 2$	Ціни	$S^M=1; S^C=1,5; S^B=2$

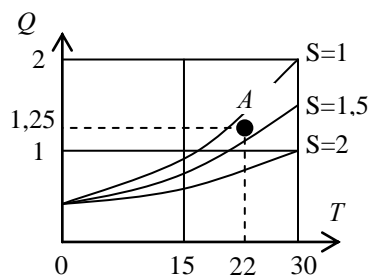


Рис. 4.7.9. Залежність попиту на пиво Q від температури повітря T та цін на ринку S

Експерт визначає реалізовані опорні точки (рис. 4.7.10).

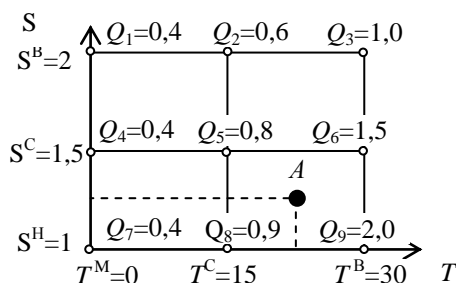


Рис. 4.7.10. Реалізовані опорні точки нечіткої системи визначення попиту на пиво $A\{T = 22^{\circ}\text{C}; S = 1,4\}$

База даних може бути представлена сукупністю продукційних правил

if $T = 0$ and $S = 1$ then $Q = 0,4$.

if $T = 15$ and $S = 1$ then $Q = 0,9$.

if $T = 30$ and $S = 2,0$ then $Q = 1,0$.

Лінійна апроксимаційна формула «інформаційного модуля з координатами, наприклад, $(T=15\dots 30^{\circ}\text{C}; S=1\dots 1,5)$ » має вигляд

$$Q = \beta_0 x_0 + \beta_1 T + \beta_2 S,$$

де $\beta_0, \beta_1, \beta_2$ – невідомі коефіцієнти; $x_0 = 1 = \text{const}$ – штучно введена змінна (для зручності розрахунків).

Розв'язок матричного рівняння для «інформаційного модуля з координатами ($T=15\dots 30^0C$; $S=1\dots 1,5$)» $Q = XB$,

$$\text{де } Q = \begin{bmatrix} Q_5 = 0,8 \\ Q_6 = 1,5 \\ Q_8 = 0,9 \\ Q_9 = 2,0 \end{bmatrix}; \quad X = \begin{bmatrix} 1 & T = 15 & S = 1,5 \\ 1 & T = 30 & S = 1,5 \\ 1 & T = 15 & S = 1 \\ 1 & T = 30 & S = 1 \end{bmatrix}; \quad B = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix};$$

Q, X – вектор вихідних значень та матриця змінних (визначаються по матриці спостережень); B – вектор коефіцієнтів, отримується в MathCAD (або MatLab) за методом найменших квадратів (МНК) у вигляді

$$B = (X^T X)^{-1} X^T Q = \begin{bmatrix} 0,7 \\ 0,06 \\ -0,6 \end{bmatrix},$$

звідки $Q = \beta_0 x_0 + \beta_1 T + \beta_2 S = 0,7 \cdot 1 + 0,06 \cdot 22 - 0,6 \cdot 1,4 = 1,18$.

Для заданої вхідної точки $A\{T = 22^0 C; S = 1,4\}$ згідно інформаційного модуля вихід Q повинен знаходитись у межах $Q = 0,8\dots 2,0$ і дорівнює за МНК $Q^A = 1,18$, а за інтерполяційною формулою

$$Q^A = y^A = y^0 + (y_{x_1} - y^0)\mu_{x_1}^A + (y_{x_2} - y^0)\mu_{x_2}^A + [y^m + y^0 - (y_{x_1} + y_{x_2})]\mu_{x_1}^A \mu_{x_2}^A =$$

$$= 0,9 + (2 - 0,9) \cdot 0,467 + (0,4 - 0,4) \cdot 0,8 + [1,5 + 0,9 - (2,0 + 0,8)] \cdot 0,467 \cdot 0,8 = 1,27,$$

де $\mu_T^A = (T^A - T^{MT}) / (T^{CT} - T^{MT}) = (22 - 15) / (30 - 15) = 0,467$ – функція упевненості у належності координати T^A до нечіткої множини універсуму U_T по осі T ; $\mu_S^A = (S^A - S^{MT}) / (S^{CT} - S^{MT}) = (1,4 - 1) / (1,5 - 1) = 0,8$ – функція упевненості координати S^A до нечіткої множини універсуму U_S по осі S .

Завдання

1. Створити однозначну нечітку систему з визначення оцінки студента ($y=2\dots 5$) у залежності від терміну праці студента над матеріалами лекцій ($x_1 = 0\dots (20+0,5N)$ годин). N – порядковий номер студента у групі.
2. Розрахувати споживання пива у залежності від коливання температури повітря в червні місяці та середнього заробітку робітника у порівнянні з вартістю прожиткового мінімуму.
3. Розрахувати залежність швидкості судна від швидкості вітру (x_1) та стану моря (x_2).

4. Отримати (по наданій базі знань у вигляді таблиці) аналітичну залежність однозначної нечіткої системи регулятора збудження синхронного генератора $u_{зБ} = f(x_1, x_2)$, де x_1 – струм, який віддається генератором в мережу; x_2 – коефіцієнт потужності $\cos \varphi_1$; N – порядковий номер студента у групі.

$x_2 \backslash x_1$	0	50 N	100 N
0	30	60	100
0,5	30	50	80
1	30	40	60

5. Розрахувати струм виходу сонячної батареї у залежності від яскравості сонця та площі батареї.

6. Розрахувати ціни продукції у залежності від ціни на сировину та ціни на паливо.

7. Розрахувати залежність ефективності роботи від заробітної плати та кваліфікації.

8. Розрахувати залежність витікання рідини від рівня рідини в ємності та положення заслонки.

9. Розрахувати швидкість переміщення вантажу від стану машини та професійності водія.

10. Розрахувати залежність величини переміщення заслонки подачі пару від температури у приміщенні та зовнішньої температури.

11. Розрахувати зріст рослин у залежності від інтенсивності поливу та температури.

12. Скласти програму визначення траєкторії покрокового переміщення рухомого об'єкту (літаючого апарату, судна, підводного човна, автономного робота тощо) за допомогою застосування однозначних нечітких множин у залежності від координат початкової та цільової точки, координат перешкод та даних «радарного датчика», який вказує кути обходу найближчої перешкоди з використанням двох вхідних нечітких множин: напрямку на ціль (x_1); кути обходу перешкод (x_2).

4.8. Продукційна база знань

4.8.1. Форми продукційних правил в базах знань

Продукційні системи були запропоновані американським математиком Постом Е. (1943) [14]. *При цьому правило-продукція трактувалося як модель заміни одного ланцюжка літер у деякому слові на інший ланцюжок.*

Переваги *продукційної моделі*:

– *універсальність* (практично будь-яка область знань може бути представлена в продукційній формі);

- *модульність* (кожен елемент знання, «чанк», є незалежним; видалення одних і додавання інших продукцій виконується незалежно);
- *декларативність* (продукція визначає знання про предметну область, але не механізм керування логічним висновком для виведення експертної консультації);
- *наочність і простота* логічного висновку, що багато в чому аналогічний процесові міркувань експерта;
- *можливість використання* інших моделей представлення знань;
- *легкість внесення змін*.

Недоліки продукційної моделі:

1. Низька ефективність процесу висновку, тому що значна частина часу непродуктивно витрачається на перевірку умов застосування правил.
2. Складність вибору продукції з конфліктної множини.

Системи продукцій часто зображують у вигляді графів, які мають назву *мереж виведення*. Наприклад, для продукційної системи (тут G – мета)

$$A, B \rightarrow C; B \rightarrow E; D \rightarrow E; F \rightarrow Q; F \rightarrow P; P \rightarrow R; C, E, R \rightarrow G$$

мережа виведення має вигляд, близький до *I-АБО графу* (рис. 4.8.1), у якому дужкою помічені дуги підзадач, що повинні бути розв'язані у сукупності (використовується логічна залежність «I»); дуги, які не мають дужок, відповідають використанню логічної залежності «АБО».

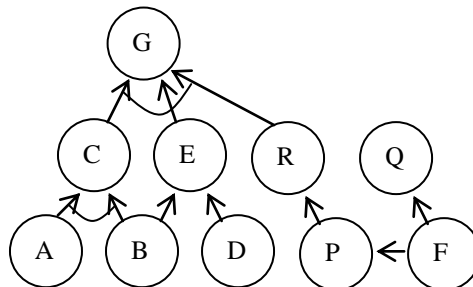


Рис. 4.8.1. Мережа виведення продукційної системи

Існує кілька форм запису правила продукції.

1. Найпростіший вигляд має форма

$A \Rightarrow B$, що означає «Якщо A , то B », («якщо справедлива умова A , то виконати дію B »), де A і B можуть розумітися як «ситуація – дія», «умова – висновок» і т. д.

Практично правило є парою «передпосилка – дія», таке правило традиційно звать *продукцією*. Представлення знань за допомогою *правил – продукцій* має подібність із правилами висновку логічних моделей і наочно відображують знання. Однак *не треба ототожнювати правило-продукцію і логічну операцію*, тому що інтерпретація продукції залежить від того, що ми маємо на увазі під A і B . Поняття продукції ширше логічного твердження.

Приклад. Побудувати формальну систему з правилами виведення у формі продукції, теоремами якої були б непарні додатні числа в унарній системі числення: в цій системі для подання чисел використовується єдиний символ – 1. Рядки, які є теоремами формальної системи, – це 1, 111, 11111, Якщо взяти за аксіому «1» і щоразу додавати парне число «11», то знову дістанемо непарне число. У цьому разі утворення рядка з попереднього полягає у приєднанні до нього фіксованих символів «11». Якщо попередній рядок позначити a , то можна скласти продукцію: $a \rightarrow a11$.

Тоді формальна система матиме вигляд

– алфавіт $A = \{1\}$; (1)

– аксіома 1; (2)

– продукція $a \rightarrow a11$. (3)

Покажемо, що 11111 є теоремою цієї формальної системи. Виведення

1) 1 – отримуємо з аксіоми (2); (4)

2) 111 – отримуємо з аксіоми (4) та продукції (3); (5)

3) 111111 – отримуємо з аксіоми (5) та продукції (3).

Тут для утворення нового рядка зі старого не потрібні операції розчленування і переміщення підрядків у старому рядку, бо достатньо приписування фіксованого числа символів.

2. Систему продукцій можна розглядати як формальну систему з алфавітом $A = \{a_1, a_2, \dots, a_n\}$, кінцевою множиною аксіом $Акс$ і кінцевою множиною правил висновку P_i у вигляді:

$P_i : \alpha_i s \Rightarrow s\beta_i$, де $i=1, 2, \dots, m$; α_i, β_i, s – слова алфавіту A ; засто-

сування правила $\alpha_i s \Rightarrow s\beta_i$ до ланцюжка літер зі слів означає викреслювання слова α_i і приписування слова β_i з аксіом $Акс$. Така заміна відповідає підстановці, що є основним оператором алгоритму Маркова.

Розглянута форма запису правила-продукції є спрощеною і являє собою лише ядро продукції. У дійсності вона може бути ускладнена введенням пріоритетів правил продукції; передумовою застосовності ядра продукції; постумовами, які треба виконати після виконання ядра продукції.

3. В узагальненому вигляді правило-продукція може бути записане як $R_{nj} : (Pr, P\partial, A \Rightarrow B, Ac)$, де R_{nj} – ідентифікатор j -ої продукції; $j=1, \dots, n$ – порядковий номер продукції;

Pr – пріоритет правила продукції;

$P\partial$ – предикат, при виконанні якого активізується ядро продукції (передумова застосовності ядра продукції);

$A \Rightarrow B$ – оснвний зміст правила-продукції;

Ac – дії і процедури, які треба виконувати після виконання ядра продукції (постумови B продукції).

Приклади продукцій:

1. Якщо A – фрейм, то B – деяка дія по перетворенню фрейму.

2. **Імплікація** (з істинності A випливає істинність B): «Якщо тварина є хижаком, має жовто-коричневий колір і темні смуги, те це – тигр».

3. **Умовна операція** (якщо справедлива умова A , то виконати дію B): «Якщо струм перевищує значення 200 А, то вимкнути електродвигун».

4. **Правила підстановки** (заміни A на B), наприклад, правила підстановки **формальних граматик**.

Під «умовою A » розуміється якесь твердження-еталон, по якому виконується пошук у базі знань, а під «дією B » – дії, які виконуються у випадку успішності такого пошуку. Найчастіше висновок у такий БЗ є прямим (від даних до пошуку цілей), але може бути і зворотним (від мети для її підтвердження – до даних). Тут під даними розуміються початкові факти в базі фактів, на основі яких запускається машина висновку або інтерпретатор правил, що перебирає правила продукційної бази знань.

4.8.2. Структурна схема продукційної бази знань

Структурна схема продукційної бази знань наведена на рис. 4.8.2 [14].

Робоча пам'ять ініціюється **фактами**, що описують задачу (уміст робочої пам'яті, як правило, не зберігається після рішення задачі). У результаті в робочій пам'яті збуджується ряд **образів**. Образи через **інтерпретатор**, що реалізує логічний висновок, надходять у **базу правил**. У базі продуктивних правил обираються правила, у яких **умова правила** (A_i , $i=1,2,\dots,n$) порівнюється з образами у робочій пам'яті. Як результат цього процесу утвориться **конфліктна множина правил**. Але за допомогою алгоритму розв'язання конфлікту активізується тільки одне з правил. Виконання дії (**наслідку правила**) приводить до зміни стану **робочої пам'яті** (до зміни кількості фактів). Далі цикл керування пошуком повторюється для наступного образу. Процес висновку є циклічним і завершується, коли не залишиться правил, умови яких дозволяється порівнювати з **образами** робочої пам'яті.

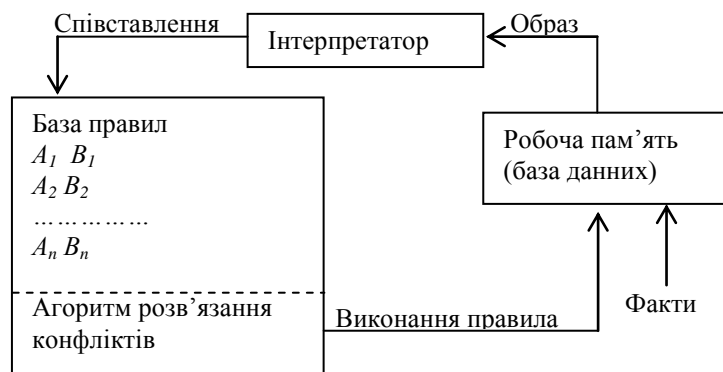


Рис. 4.8.2. Структурна схема продукційної бази знань

Таким чином, процес висновку, заснований на пошуку за зразком, складається з 4-х кроків:

1. Вибір образу й передача його через інтерпретатор у базу правил.
2. Порівняння образу зі зразком і формування конфліктної множини правил.
3. Розв'язання конфліктів: вибір одного правила.
4. Виконання правила.

Керування розв'язанням конфлікту. Порядок активації правил конфліктної множини визначається стратегією розв'язання конфліктів. Продукція може виконуватись негайно, а можуть створюватись конфліктні набори продукцій (фронт готових продукцій). У приведених раніше прикладах конфліктні правила дописувалися в кінець списку. Якщо вибрати стратегію розв'язання конфліктів за рахунок виділення першого правила, то одержимо пошук у ширину, а вибір останнього правила відповідає пошукові в глибину. Найчастіше застосовують пошук у глибину. Іноді в постумовах продукції вказується ім'я наступної продукції, яка повинна виконуватись; це перетворює систему продукцій у звичайний алгоритм.

Іншими принципами, що використовуються при розв'язанні конфліктів, є:

1. Принцип *«стопки книг»*. Список упорядковується відповідно до частоти використання продукцій у минулому. У першу чергу вибирається продукція, що використовується частіше усіх.

2. Принцип *найбільш довгої умови* віддає перевагу продукції з найбільш довгою умовою, тому що вважається, що при цьому враховується більше інформації про поточну ситуацію, і тому цей напрямок повинний швидше приводити до успіху.

3. Принцип *метaproдукцій* заснований на введенні в базу знань метарулів із заданими пріоритетами розв'язання конфліктів.

4. Принцип *пріоритетного вибору* заснований на «прив'язці» до кожної продукції статичного або динамічного пріоритету, що визначає порядок активації.

5. Принцип *«класної дошки»*, на якій знаходять інформацію, що може ініціювати запуск різних продукцій.

2. Принцип *ранжирування правил по пріоритету*: найбільший пріоритет має правило, що було занесено в базу останнім.

4.9. Семантичні мережі

4.9.1. Базові компоненти семантичної мережі та їх призначення

Термін *семантика* означає «зміст», а сама семантика – це наука, що визначає відношення між об'єктами. Уперше семантичні мережі були введені Куліаном М. Р. (Quillian, 1967) для тлумачного словника, який пояснював сенс англійських слів (звідси виникла й назва мережі): в цій мережі кожне слово визначається через множину інших слів, що дозволяє виявити

сенс слова. Елемент бази знань має вигляд сторінки (площини), на якій представлений граф одного слова) [14].

Семантичні мережі (СМ) мають вигляд **направленого орієнтованого графу** (рис. 4.9.1), вузли (вершини) якого відповідають абстрактним або конкретним об'єктам (сутностям, поняттям, концептам), а дуги – відношенням (зв'язкам) між ними. Вузли та дуги звичайно мають мітки (імена). Зв'язки в системі практично завжди є орієнтованими, тому що відношення, вказані зваженими зв'язками, не повинні бути симетричними.

Обмежень на типи об'єктів та типи відношень у семантичній мережі не існує.

Зображення семантичної мережі у вигляді рис. 4.9.1 із повідомленнями про оцінки студента Ковалья не є єдино можливим. Існують різні описи СМ, у тому числі з переведенням у вершини не об'єктів, а відношень (концептуальні графи Дж.Соува); або у вигляді блочних структур Г. Хендрикса.

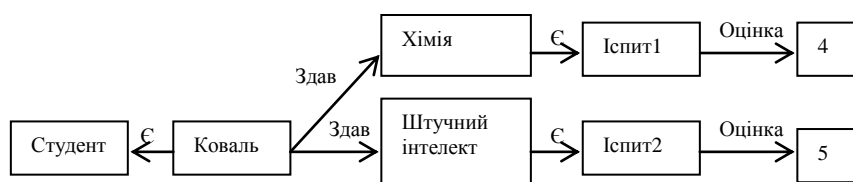


Рис. 4.9.1. Приклад семантичної мережі

Частіше за все в семантичних мережах використовуються такі **відношення**:

– «Є представником (екземпляром)» – Коваль К. Н. «є представником (екземпляром)» студентів із напрямком стрілки від елемента до множини. Екземпляр може бути представником кількох узагальнених об'єктів. Зв'язки типу «частка – ціле», «є представником (множини)», «елемент – множина» у ряду випадків можуть використовуватись з протилежним напрямком: «множина – підмножина», «ціле – частка», «загальне – частка», «клас – підклас». Наприклад, «Рід-вид»: «рід – тварина»; «вид – птиця/кішка/слон». Іноді це викликає непорозуміння і тому такі зв'язки формалізують відношеннями: *ako* (від англійського *a kind of* – різновид множини з напрямком стрілки від елемента до множини) та *is_a* (від англійського *is a member of the class* – бути представником класу-множини з напрямком стрілки від елемента до множини)

– **Функціональні зв'язки** (звичайно з дієсловами «виготовлює», «впливає»).

– **Кількісні зв'язки** (більше, менше, рівне).

– **Просторові зв'язки** (далеко від, близько від, за, під, над, праворуч, ліворуч, зверху, вище, нижче).

– **Часові зв'язки** (раніше, пізніше, на протязі 5 хвилин).

– **Атрибутивні зв'язки** (мати властивість, мати значення).

– *Логічні* відношення (кон'юнкція, диз'юнкція, заперечення, імплікація). Логічні зв'язки («І», «АБО», «НІ»).

– *Квантіфіковані* відношення, які розділяються на *нечіткі* квантори (багато, кілька, часто, декілька, не часто) та *логічні* квантори загальності та існування (для усіх, для частки, для п'яти)

– *Лінгвістичні* відношення (відмінки, дієслово та ін.).

Відношення СМ можуть бути *однорідними* (з одним типом відношень) та *неоднорідними* (із різними типами відношень), а самі типи відношень можуть бути *бінарними* (відношення пов'язує два об'єкти) та *N-арними* (відношення пов'язує більше за два об'єкти). Як бачимо, СМ мають різні типи зв'язків-відношень. У часткових випадках у СМ використовуються спеціальні типи зв'язків:

1. *Класифікуючі мережі* задають відношення ієрархії між інформаційними одиницями.

2. *Функціональні мережі (обчислювальні моделі)* характеризуються наявністю функціональних відношень, які описують процедури обчислень одних інформаційних одиниць через інші.

Важливою особливістю семантичних мереж є можливість забезпечення *транзитивності* відношень $(\alpha R \beta \wedge \beta R \gamma) \Rightarrow \alpha R \gamma$, що означає: «для будь-яких об'єктів α, β, γ », які мають відношення R між собою у вигляді $(\alpha R \beta)$ та $(\beta R \gamma)$, справедливе твердження « α знаходиться у відношенні R з γ ».

Транзитивність дозволяє, наприклад, забезпечити наслідування властивостей від виду до суб-виду, що дозволяє отримати найпростіше дедуктивне твердження типу «Усі люди – смертні. Сократ – людина. Таким чином, Сократ – смертний».

Об'єкти предметної області з точки зору узагальнення й структури семантичної мережі можна умовно розділити на три групи:

1. *Узагальнений об'єкт* – деяка узагальнена абстракція, об'єднуючий образ. Наприклад, «співробітник», «автомобіль», «учень», «виріб» (фактично вони є класами предметної області).

2. *Індивідуальний об'єкт* – виділений екземпляр класу. Наприклад – «співробітник Ковальов П. К.».

3. *Агрегатний об'єкт* – це складний об'єкт, який створений з інших об'єктів. Окремі об'єкти розглядаються як складові частини складного об'єкта. Наприклад, кімната складається з підлоги, стін, стелі, вікон, дверей (див. далі фрейми).

Об'єкти мережі з точки зору семантики можна у більшості випадків розділити на такі умовні типи:

1. *Об'єкти-поняття* – відомості про фізичні та абстрактні об'єкти предметної області.

2. *Об'єкти-події* – абстрактні або конкретні дії, які можуть змінити стан предметної області.

3. Об'єкти-властивості, які уточнюють *поняття* (колір, форму, розміри) та *події* (місце, час, термін виконання), вказуючи їх параметри.

4.9.2. Апарат успадкування властивостей об'єктів семантичної мережі

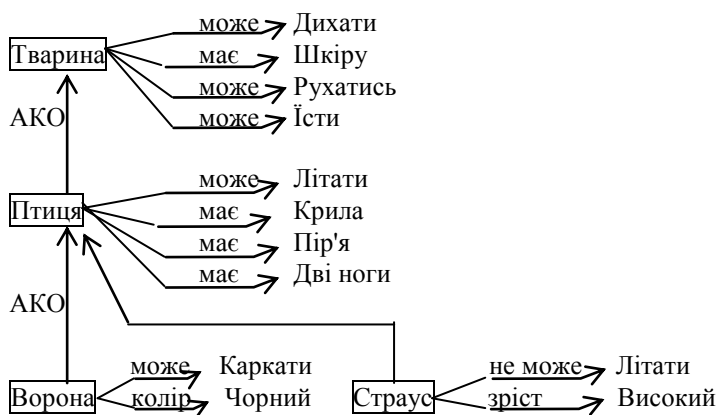


Рис. 4.9.2. Тварини у формі семантичної мережі

Прикладом семантичної мережі є рис. 4.9.2. Мережа рис. 4.9.2 була розроблена А. М. Коллінзом та М. Р. Куїлліаном (1969) для моделювання механізму пам'яті людини. За рахунок відношень АКО (від англ. *a-kind-of*, «різновид») та *is_a* (*is a member of class* «є представником класу») мережа набуває ієрархічні властивості, які дозволяють відповісти на питання: «Чи є ворона птицею?», «Чи має ворона шкіру?», «Чи може ворона їсти?». Як видно з рис. 4.9.2, властивості тварин (дихати, мати шкіру, рухатись, їсти) д'якуючі *наслідуванню* властивостей, *не дублюються* безпосередньо для птиці у зв'язку з АКО – ієрархією. Наприклад, можливість рухатись та мати пір'я властиві для птиці і, отже, властиві й вороні) [14].

Успадкування властивостей зберігає пам'ять, але знижує продуктивність і викликає ряд проблем. Зв'язок семантики з деталями реалізації ускладнює використання СМ. Наприклад, не зрозуміло, чи слово «ЕОМ» є конкретним комп'ютером, або являє собою клас усіх комп'ютерів; чи слово «м'яч» є конкретним м'ячем, або являє собою клас усіх м'ячів. Виникає проблема комбінаторного вибуху особливо при пошуку зв'язків між вершинами графу.

4.10. Фреймові моделі

4.10.1. Загальні положення теорії фреймів

Поняття фреймів ввів Марвін Мінський (1975 р.). *Фрейм* – це модель для відображення деякого складного збірного об'єкта, що складається з декількох різнотипних елементів, які його описують.

Фрейм можна розглядати як «кошик» із *слотів*, призначений для збирання в слоти найбільш важливих фрагментів опису типових складних об'єктів, ситуацій, подій; кожен окремий слот «кошика» призначений для опису окремого елемента складного об'єкта. Ці «кошки» мають ієрархічні зв'язки. У якості слотів можуть виступати імена інших фреймів [14].

Наприклад, фрейм «кімната» складається зі слотів: стін, вікон, фрейму «двері», підлоги, стелі, протягу, кішки в кімнаті, мешканців. Фрейм «двері» складається зі слотів: дошок, ручки, замка. До зазначених слотів додаються допоміжні слоти: службових процедур «кошика» і його зв'язків, взаємних зв'язків між слотами й управління процесом логічного висновку.

Існують: фрейми-структури, що використовуються для позначення об'єктів і понять (позиція, застава, вексель); фрейми-посади (менеджер, касир, клієнт); фрейми-образи; фрейми-події; фрейми-сценарії (банкрутство, збори акціонерів, святкування іменин); фрейми-ситуації (тривога, аварія, робочий режим).

В експертній системі фрейм «кімната» створюється як абстрактне узагальнене поняття (клас або шаблон). На основі цього абстрактного поняття створюються конкретні екземпляри фреймів: «кімната1», «кімната2» ... з конкретною вказівкою: висоти, довжини, фарбування стін; типу, площі, розміщення вікон і т. д. Наповнення цього фрейму змістом (окремими елементами) виконується користувачем. Якщо в кімнаті немає кішки, то відповідні поняття в конкретному екземплярі класу не заповнюються й утворюється «дірка». Після заповнення слотів конкретними даними, фрейм-екземпляр буде представляти ту або іншу ситуацію, процес, об'єкт, явище предметної області.

Якщо при описі загального фрейму деякі слоти вже заповнені конкретними значеннями, ці значення передаються всім екземплярам цього фрейму, і вони можуть успадковуватись фреймами-нащадками. Указівник наслідування фіксує інформацію, яка передається зі слотів фрейму верхнього рівня до слотів фрейму нижчого рівня. Таким чином, фрейми створюють ієрархічні мережі з наслідуванням властивостей.

Фрейми можуть бути статичними (інформація не міняється) і динамічними (інформація міняється). Перевагою фреймів є можливість об'єднання великої кількості різнотипної інформації, а недоліком – ускладнення обробки цієї інформації.

У загальному вигляді фрейм можна представити структурою [2]:

$\{ \text{ім'я_фрейму, загальні_службові_процедури, (Ім'я_Слота1, Значення_Слота1, Процедури_Слота1), \dots, (Ім'я_СлотаN, Значення_СлотаN, Процедури_СлотаN)} \}$

Можна вважати, що у типовому фреймі зберігається така інформація:

- ім'я і загальний опис фрейму;
- інформація про батьківський фрейм;

– інформація про окремі слоти: імена та значення слотів, а також їх властивості (наприклад, тип слота, умови коректності інформації, яка зберігається в слоті, тощо). Важливою властивістю слота є, зокрема, **режим успадкування**: можна задати обов'язкове успадкування, успадкування за певної умови, відмову від успадкування тощо.

4.10.2. Типові процедури роботи із фреймами

У фреймових системах використовується три способи керування висновком: за допомогою механізму спадкування, за допомогою процедур-демонів, за допомогою приєднаних службових процедур [14].

Механізм спадкування є основним засобом висновку, яким оснащуються фреймові системи: він забезпечує значну економію пам'яті й автоматичне визначення значень для слотів фреймів нижніх рівнів.

У фреймах звичайно використовується такий набір процедур:

1. **Процедури конструювання класу**, що формують фрейми-прототипи (шаблони для заповнення) із необхідним набором слотів і відповідними посиланнями на суперкласи.

2. **Процедури конструювання екземпляра класу**, що дозволяють конкретизувати клас і одержати тим самим *екземпляри фреймів* (фрейми-прикладі). Ця процедура автоматично встановлює зв'язок екземпляра фрейму з відповідним класом за допомогою «is_a» слота.

3. **Процедури запису й читання значень слотів** (процедури-демони й процедури-слуги) здійснюють доступ до слотів відповідних фреймів і дозволяють користувачеві ввести або визначити значення слота.

4. **Приєднані службові процедури**: ім'я приєднаної процедури виступає як значення слота. Вона запускається за повідомленням, переданому з іншого фрейму. У цьому повідомленні з іншого фрейму вказується: *ім'я фрейму з даним слотом, ім'я слота, параметри*. Приєднана процедура одержує необхідні параметри і, у свою чергу, може передати повідомлення в інший фрейм і викликати іншу приєднану процедуру і т. д. Виникає ланцюжок передач від одного фрейму до іншого і відповідне повернення відповідей. У підсумку відповідь повертається в точку вихідного первісного виклику функції і повідомляється користувачеві. У такий спосіб на кожний із фреймів покладаються визначені обов'язки і забезпечується їхнє погоджене виконання.

Використання процедур підвищує гнучкість фреймових систем, але ускладнює керування.

Важливою особливістю фрейму є можливість включення в слоти приєднаних процедур. Серед них виділяють два види процедур: процедури-демони й процедури-слуги.

Процедури-демони підтримують БЗ у робочому стані й активізуються автоматично при кожній спробі додавання або вилучення даних із слота. Найчастіше використовується три типи процедур-демонів:

- «якщо_додано» використовується, коли в слот містяться нові дані;
- «якщо_вилучено» використовується, коли дані вилучаються зі слота;
- «якщо_потрібно» використовується, коли запитується інформація зі слота, а зі слотом не зв'язані ніякі значення.

Процедура-слуга обслуговує конкретний слот, активується тільки по запиту до цього слоту і показує його чисельне заповнення.

Якщо у фреймових системах не використовувати процедури-демони і приєднані процедури, то фреймова модель представлення знань буде відповідати семантичній мережі.

Спадкування.

Фрейм може містити слот, що задає відношення даного фрейму до інших фреймів мережі. Дуже часто на практиці використовуються мережі фреймів ієрархічної структури. У цьому випадку спеціальний «*is_a*» слот установлює стосунки «екземпляр класу», а слот «*ako*» указує, що даний фрейм є прямим підкласом вищого фрейму, що називають *батьківським* або *суперкласом*. До складу слотів може бути включена й інформація зі спадкування властивостей: покажчик спадкування (розрізняють їхні типи: *unique*, *same*, *range*, *override* й ін.) фіксує ту інформацію, яка зі слотів фрейму верхнього рівня передається до відповідних слотів нижнього рівня:

unique (унікальний) – слоти з однаковими іменами у фреймах, які знаходяться на різних рівнях ієрархії, можуть мати різні значення;

same (такий самий) – показує, що слоти мають однакові значення;

range (діапазон) – установлює границі зміни значень слота фрейму нижнього рівня;

override – комбінація можливостей властивостей покажчиків *unique* і *same*.

4.10.3. Ієрархічність фреймової моделі

Як уже вказувалося, важливою особливістю фрейму є можливість включення в слоти ***приєднаних процедур слотів*** (процедури-демони, процедури-слуги, приєднані службові процедури, які були описані вище). При цьому варто враховувати, що хоча до слотів окремих фреймів і можуть бути підключені спеціальні процедури, але ці локальні модулі не в змозі взяти на себе всі турботи по організації обчислювального процесу в системі. Необхідно мати ***спеціальний інтерпретатор***, який буде формувати й обробляти запити і приймати рішення [14].

Тому фрейми найчастіше використовуються з іншими засобами представлення знань:

- з *породжувальними правилами*;
- з *об'єктно-орієнтованим програмуванням*, яке організує пересилання повідомлень між фреймами і ініціалізує більш складні обчислення.

Існує багато об'єктно-орієнтованих мов програмування, тобто мов, які безпосередньо підтримують об'єктну модель та об'єктно-орієнтоване програмування. Історично першою була мова **SmallTalk**. Зараз популярними є **Delphi** (візуальна реалізація Паскалю), **C++** (стандарт де-факто у сфері об'єктно-орієнтованого програмування), **Java**. Також розвиваються технології об'єктно-орієнтованих баз даних [1; 4].

Об'єктом називається абстракція, що характеризується станом, поведінкою та ідентифікованістю; сукупності схожих об'єктів утворюють клас; терміни «екземпляр класу» та «об'єкт» рівноправні.

Ідентифікованість – це така властивість об'єкта, яка відрізняє його від усіх інших об'єктів.

При цьому для екземплярів класу **спільним є перелік усіх характеристик**, а розрізняються екземпляри одного класу між собою за рахунок різних значень характеристик.

Об'єкти та класи мають такі найважливіші властивості:

1. **Абстрагування** – виокремлення в описі класу найважливіших характеристик деякої сутності, що відрізняють її від усіх інших сутностей.

2. **Інкапсуляція** – клас повідомляє про свої функціональні можливості, але приховує від зовнішнього світу особливості реалізації цих можливостей і свою внутрішню будову.

Слід відмітити, що в програмуванні термін «інкапсуляція» часто застосовується у вузькому значенні, а саме: у класі можуть бути описані процедури (**методи**), що визначають його поведінку.

3. **Модульність** – об'єкт складається з окремих відносно самостійних модулів.

4. **Ієрархія** – об'єкти та класи є ієрархічно упорядкованими. Зокрема, вони утворюють ієрархію класів та ієрархію об'єктів. Успадкування (наслідування) пов'язане саме з цими ієрархіями, в першу чергу, з ієрархією класів.

Кожен підклас (або екземпляр класу) успадковує слоти свого суперкласа. Якщо підклас і суперклас мають слоти зі співпадаючими іменами, то використовуються значення слотів підкласу) [14].

У загальному випадку порядок спадкування визначається **списком попередників**, що формується відповідно до напрямку зв'язків «*is_a*» і «*ako*». Більш складний випадок виникає, якщо фрейм має кілька зв'язків «*is_a*» і «*ako*». У цьому випадку говорять про множинне спадкування.

Звичайно, будь-який клас повинний з'явитися в списку попередників *раніш, ніж його безпосередній суперклас*.

Семантичні мережі та фрейми мають значно більше спільних рис, ніж відмінностей, і тому на сучасному етапі фреймові моделі та семантичні мережі розглядають, як правило, сумісно. З одного боку, ніщо не заважає розглядати вузли семантичної мережі як фрейми з власною внутрішньою структурою. З іншого боку, можна вводити різноманітні зв'язки між слотами фреймів. Тоді фрейм набуває рис семантичної мережі.

4.11. Експертні системи

4.11.1. Призначення експертних систем

Експертна система – це програмний засіб, що використовує експертні знання у певній предметній області з метою ефективного вирішення задач на рівні середнього професіонала (експерта).

Експертні системи використовуються для вирішення так званих *неформалізованих задач*:

1. Які недостатньо добре розуміються або вивчені.
2. Не можуть бути задані в числовій формі.
3. Цілі не можна виразити в термінах точно визначеної цільової функції.
4. Не існує відомого алгоритмічного рішення задачі.
5. Якщо алгоритмічне рішення є, то його не можна використовувати через обмеженість ресурсів (час, пам'ять).

Експертні системи застосовують там, де існують евристичні або інтуїтивні методи рішень і немає точних алгоритмів або розрахунків. ЕС містить накопичені знання багатьох експертів у певній предметній області й здатна в рамках цієї області класифікувати об'єкти, давати відповіді, пояснення, рекомендації, поради, запитуючи при необхідності додаткову інформацію [14].

Існують ЕС, які перевершують в *медичній діагностиці* досвідчених фахівців; теж саме відноситься до пошуку корисних копалин, передбачення захворювання зернових культур тощо. Знання – це дефіцитний ресурс, який дорого обходиться суспільству і який дорого оплачується. Традиційно знання експерта передаються новачку через багато років навчання та практичне стажування. І цей процес можна прискорити і здешевити за рахунок використання ЕОМ.

У більшості ЕС знання організовані як деяка сукупність фактів та правил їх використання. Правила дозволяють зробити висновок на основі початкових даних та припущень.

До особливостей ЕС можна віднести:

1. ЕС обмежена визначеною областю експертизи. Не можна задавати медичній ЕС питання про погоду.
2. ЕС спроможна розмірковувати при сумнівних даних.
3. ЕС може пояснити свої дії і, отже, може використовуватись для навчання.
4. Факти та механізм висновку чітко розділені.
5. Вона видає на виході пораду.
6. Вона економічно вигідна і має можливість поступового нарощування.

Найчастіше ЕС використовує правила, які мають форму: «ЯКЩО. ТОДІ. ІНАКШЕ.». Твердження, як правило, мають ймовірнісну оцінку, або ваговий коефіцієнт.

Вважається, що програму в експертних системах треба виконувати на мові Пролог або LISP.

Завдяки ЕС сформувався новий розділ інформатики – *інженерія знань (knowledge engineering)*, спрямований на «здобуття», формалізацію, зберігання та розподіл знань у заданій області (репутація фірм, з якими маємо справу; наукові знання; ринкова інформація і т. д.). Поява експертних систем ознаменувала перехід від суто теоретичної сфери штучного інтелекту до прикладної.

4.11.2. Створення експертної системи

– У розробці й експлуатації експертної системи беруть участь такі фахівці.

– *Експерт* – висококваліфікований фахівець у проблемній області, забезпечує повноту і правильність введених в систему знань.

– *Інженер зі знань* (когнітолог) – фахівець з розробки експертних систем, який повинен знання експерта перевести у програму бази знань. Інженер виявляє знання у експерта через форми спілкування (лекції, діалог, ігри, перелік питань), спостереження, інформативні джерела.

– *Програміст* – фахівець з розробки інструментальних засобів.

– *Користувач* – людина, що використовує вже побудовану експертну систему.

Створення ЕС проходить етапи:

1. *Ідентифікації*: визначаються задачі, що підлягають вирішенню.

2. *Концептуалізації*: експерт та інженер зі знань виявляють вихідні і вихідні дані, підзадачі, зв'язки, визначають методи вирішення задач.

3. *Формалізації*: визначаються способи подання й інтерпретації усіх видів знань, моделюється робота системи, вибираються програмні засоби розробки, оцінюється адекватність цілям і повнота системи зафіксованих понять, методів рішення, засобів подання і маніпулювання знаннями.

4. *Виконання* (реалізація): створення одного або декількох прототипів, що вирішують необхідні задачі.

5. *Тестування*: експерт (та інженер зі знань) перевіряє компетентність експертної системи на великій кількості репрезентативних задач.

6. *Дослідної експлуатації*: перевіряється придатність експертної системи для кінцевих користувачів, яка визначається в основному зручністю роботи із системою та її корисністю. За результатами цього етапу може знадобитися істотна модифікація експертної системи. Після успішного завершення етапу дослідної експлуатації і використання різними користувачами експертна система може класифікуватися як комерційна.

Модифікація експертної системи здійснюється майже постійно в ході її створення.

Ідеальна статична експертна система містить такі основні компоненти, як: машина логічного виведення (вирішувач, інтерпретатор

правил), база знань, підсистема набуття знань, підсистема пояснення рішень, інтерфейсна підсистема (діалоговий компонент), робоча пам'ять (база даних).

Найпростішим є *прямий ланцюжок* вирішення проблеми (*від фактів до мети*): він дає змогу переглянути все дерево можливих рішень і знайти всі проміжні вершини.

Зворотний висновок використовує механізм перевірки гіпотез і тому завжди приводить до рішення про істинність або помилковість гіпотези. Якщо простір ймовірних гіпотез великий і відсутні обмеження, то зворотний висновок не ефективніший від прямого. На практиці іноді доцільно поєднувати на різних етапах роботи прямий та зворотний висновок МЛВ.

Використовують пошук «у глибину» і «вшир». При пошуку у глибину почергово розглядаються шляхи по дереву рішення до кінцевих вершин. Якщо виявлені вершини не будуть рішенням, то досліджується наступний шлях, і процес продовжується в режимі «відкату із забуванням».

Режим управління «вшир» зводиться до дослідження всіх гілок дерева рішень, що виходять із початкового пункту завдовжки в один маршрут.

Найдоцільнішим є поєднання пошуку «вшир» із пошуком «у глибину» з відсіканням гілок за результатом пошуку, що зводиться до перегляду кожної гілки дерева рішення просуванням не на один крок, а на число кроків, встановлене самою ЕС.

Логічний висновок поділяється на монотонний і не-монотонний.

Підсистема пояснення в ЕС реалізується різними способами. Це може бути:

- набір інформаційних довідок про стан системи на момент припинення її роботи;
- повний або частковий опис подоланого системою шляху по дереву рішень;
- список гіпотез, що перевіряються (основи для їх формування і результати перевірки);
- список цілей, які керують роботою системи, і шляхів їх досягнення.

Режими роботи експертної системи виділяють такі: *режим набуття знань, режим консультації* (рішення).

Перевагами експертних систем є: підвищена доступність, можливість одержання й об'єднання експертних знань з багатьох джерел, сталість (знання експертної системи зберігаються протягом невизначено довгого часу і нікуди не зникають, низька вартість, підвищена надійність, можливість пояснення рішень, можливість застосування в якості навчальної програми, можливість застосування у якості інтелектуальної бази даних, формалізація і перевірка знань на правильність, несуперечність і повноту.

Виготовлення експертних систем залучають значні грошові інвестиції і людські зусилля і є тривалим процесом витягу знань з експерта, їхньої формалізації, перевірки на несуперечність і усунення протиріч.

4.11.3. Звичайні та спеціальні мови програмування експертних систем

На ранньому етапі проектування проектувальники для подання знань та управління їх застосуванням використовували звичайні мови програмування. Але виявилось, що для ЕС набагато економічніше використовувати спеціальні мови програмування.

Мова представлення знань (МПЗ). Знання, які виставляють у СШ, зберігаються в базі знань. Є чотири найбільш поширені основні моделі подання знань:

- логічні моделі;
- продукційні моделі;
- моделі, засновані на правилах;
- моделі семантичної мережі; фреймові моделі.

Оболонки ЕС. Мимоволі напрошується думка, що компоненти ЕС можуть бути використані повторно для створення експертних систем в різних предметних областях. Використання цих програм в якості базових компонентів конкретних експертних систем дозволило називати їх оболонкою системи.

Мови програмування ЕС високого рівня. Мови високого рівня є прекрасним засобом швидкого створення прототипу експертної системи. Але інтерфейс користувача такого середовища поступається *інтерфейсу оболонок* по «дружності».

Мови програмування ЕС високого рівня надають проектувальнику експертних систем значно більшу свободу дій, ніж оболонки ЕС. Але, природно, можливості мов високого рівня також не безмежні – кожна мова має свої обмеження.

У ранніх моделях систем, заснованих на породжуючих правилах, до 90 % часу роботи йшло на виконання операцій зіставлення умов. Алгоритм, запропонований Фордж [Forgy, 1982] і реалізований в *мовах опису* породжуючих правил, дозволяє значно знизити кількість внутрішніх ітерацій за рахунок використання сортування.

Подолання недоліків програмування породжуючих правил лежить на шляху об'єднання їх з іншими напрямками програмування. Прикладом такого об'єднання може служити *комбінування породжуючих правил і фреймів*, що дозволяє зіставляти умови, специфіковані в правилах, з вмістом слотів фреймів.

Об'єктно-орієнтовані мови (ООМ). Формат правил об'єктно-орієнтованих мов добре узгоджується з поданням знань у формі «при виконанні умов S_1, \dots, S_n виконати дію A », але менш підходить для опису складних об'єктів і відносин між ними. Мови об'єктно-орієнтованого програмування надають у розпорядження програміста програмне середовище для організації знань в термінах декларативного представлення об'єктів предметної області. Все що пов'язане з процедурною стороною

рішення проблем, розподіляється між об'єктами, які в такому випадку мають власні процедури і можуть спілкуватися один з одним за допомогою протоколів передачі повідомлень.

Мови логічного програмування експертних систем. На основі першого досвіду застосування інструментальних засобів типу оболонок при проектуванні експертних систем, багато дослідників вважали, що більш перспективним є підхід, заснований на логічному програмуванні.

Наприклад, передбачалося, що *експертні системи* можуть бути досить просто реалізовані на мові PROLOG. Правила можна представити у вигляді фраз Хорна, в яких головний (позитивний) літерал відповідає висновку, а інші (негативні) літерали будуть відповідати умовам. Але досвід застосування логічного програмування, продемонстрував ряд явних відхилень від синтаксису обчислення предикатів першого порядку і його процедурної інтерпретації в стандартній версії PROLOG.

Багатофункціональні програмні середовища. Оскільки не існує єдиної універсальної мови подання знань для довільної експертної системи, то у розробників виникло бажання об'єднати декілька різних схем подання, особливо на етапі створення прототипу.

Раціональне зерно в перших спробах звести разом стилі, засновані на правилах і фреймах, полягало в тому, щоб об'єднати здатність представляти об'єкти, характерні для фреймів, з можливостями пов'язувати умови і дії за допомогою породжуючих правил.

У рамках об'єктно-орієнтованого програмування модулі середовища, які підтримують різні стилі програмування, можна зв'язувати зі слотами структурованих об'єктів, а правила модифікують значення цих слотів.

Спілкування з комп'ютером на природній мові передбачається у майбутньому. Недоліки природної мови (багатозначність, контекстність сенсу, необхідність врахування граматики та синтаксису) призвело до того, що перші програми машинного перекладу виявилися неефективними. Тому наразі спілкування користувача з комп'ютером виконується *на проблемно-орієнтованих підмножинах природної мови* із застосуванням онтологій та тезаурусу.

Останнім часом в задачах обробки природної мови все частіше застосовують емпіричний підхід, який базується на використанні методів статистики і машинного навчання. У цьому випадку підсистема розуміння висловлювання навчається на великих обсягах реальних мовних даних. Це дозволяє автоматизувати процес формування необхідних баз знань.

4.12. Бази даних

4.12.1. Загальні відомості про бази даних

База даних (БД) використовується багатьма користувачами (прикладними програмами) і є єдиним централізованим джерелом даних певної предметної області.

З точки зору розпізнавання образів БД має унікальні можливості, бо дозволяє класифікувати ознаки об'єкта у вигляді рядка таблиці з урахуванням у відповідних колонках цього рядка цифр, тексту, зображення, фотографій, формул, креслень, музики та зв'язків з іншими об'єктами (теж у формі таблиці). БД у даному випадку розглядаються у спрощеному вигляді – лише з точки зору використання їх особливостей в системах розпізнавання образів. Задача проектування БД не ставиться.

Загальний вигляд БД наведений на рис. 4.12.1, де позначено:

- БД – база даних у вигляді програмами з багатьма таблицями, в яких зберігаються дані.
- СУБД – система управління БД.
- Програма 1, ..., Програма 3 – прикладна програма, за допомогою якої користувач отримує потрібні йому дані з банку даних.
- Банк даних – це СУБД плюс БД.

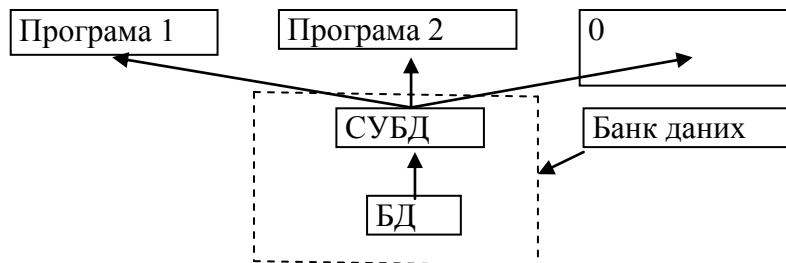


Рис. 4.12.1. Загальний вигляд БД

Найбільш розповсюджені ієрархічні, мережні, реляційні моделі БД.

Структурні особливості ієрархічної моделі відображені на рис. 4.12.2:

1. Вони допускають лише зв'язки 1:1, 1:М і не допускають М:1, М:М (розшифровка зв'язків: 1:1 – «один-до-одного»; 1:М – «один-до-багатьох»; М:1 – «багато-до-одного»; М:М – «багато-до-багатьох»).
2. Вони мають вигляд дерев, що складаються з вузлів та дуг (гілок).

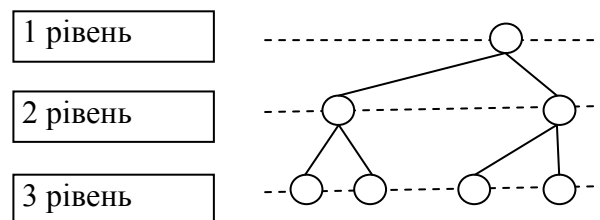


Рис. 4.12.2. Ієрархічна модель БД

Кожний вузол ієрархічної моделі БД є конкретним об'єктом. Зв'язки між вузлами підпорядковуються таким правилам:

- На верхньому рівні ієрархії знаходиться один кореневий вузол.
- Нижче знаходяться залежні від нього вузли. Взаємні зв'язки між вузлами будуються по принципу «батько – нащадок».

– Кожний вузол – «батько» може мати кілька вузлів – «нащадків», зв'язаних з ним дугами. Доступ до кожного вузла отримується через його батьківську вузол.

– Вузол – «батько» та вузли – «нащадки» сумісно створюють один логічний запис. Тому з вилученням батьківського вузла вилучаються й вузли – «нащадки».

Недоліки ієрархічних моделей: відсутність зв'язків М:1, М:М та значні витрати часу на пошук даних.

Мережева модель бази даних є орієнтованим графом з вершинами – об'єктами та дугами – зв'язками, які підтримують усі можливі зв'язки між об'єктами.

Реляційні бази даних складаються з пов'язаних між собою таблиць. Загальне уявлення про таблиці реляційної бази даних дає табл. 4.12.1 та її структурна форма (табл.4.12.2).

Таблиця 4.12.1.

«ТбКлієнти» – таблиця реляційної БД

КлКОД	КлПрізвище	КлАдреса	КлТелефон
1	Буков	Мира, 25	37-80-21
2	Маслаев	Соболья, 87	22-64-26
3	Усков	Масляна, 145	33-40-57

Стосовно реляційної БД ми будемо використовувати визначення, які є загальними при розгляді таблиць і не відповідають термінології реляційних БД:

1. **Таблиці** – це таблиці БД (в теорії реляційної БД таблиці зветься *відношеннями, relation*).

2. **Колонки (поля)** – це колонки в таблиці (в теорії реляційної БД колонки зветься *атрибутами, attributes*).

3. **Рядки (записи)** – це рядки в таблиці, які вміщують комірки для усіх колонок з відповідними інформаційними записами у них (в теорії реляційної БД рядки зветься *кортежами*). Унікальність рядків забезпечує **первинний ключ** – одна колонка або група колонок, які мають унікальне для даного рядка значення.

Кількість колонок таблиці може досягати 255, а кількість рядків – наскільки вистачає пам'яті.

Таблиці є основним компонентом реляційної БД. З таблиць за рахунок фільтрації, сортування та комбінаторного групування отримується відповідна інформація, яка зветься в програмному середовищі БД **запитами, формами, звітами**.

Дані об'єкта в таблиці записують у її рядку. Для ідентифікації цих записів використовується **ключова колонка «КлКОД»**, яке забезпечує унікальність запису у рядках (відсутність тотожних записів) і можливість зв'язку між таблицями. **Складний первинний ключ** складається з кількох

колонок (наприклад, може складатись з трьох колонок: прізвище, ім'я та по-батькові).

У скороченому вигляді табл. 4.12.1 може бути представленою структурною формою табл. 4.12.2, в якій даються лише назва таблиці ТбКлієнти, а нижче розміщуються заголовки її колонок.

Таблиця 4.12.2.

Структурна форма таблиці «ТбКлієнти»

«ТбКлієнти»
КлКОД
КлПрізвище
КлАдреса
КлТелефон

Звичайно БД вміщує ряд таблиць з однаковими назвами, тому назви таблиць бажано починати з префіксу: ТбКлієнти (для таблиці), ЗпКлієнти (для запиту), ФКлієнти (для форми), ЗвКлієнти (для звіту).

По тій же причині назви колонок бажано помічати *префіксом*, взятим з назви таблиці. Приклад назв колонок для таблиці ТбКлієнти: КлКОД, КлПрізвище, КлАдреса, КлТелефон. Це робиться тому, що назви колонок «Прізвище» для таблиць (ТбКлієнтів та ТбПостачальників) можуть співпадати у запитах, формах та звітах.

Таблиці реляційних БД можна поділити на два види:

1. Базова (первинна) таблиця, яка обов'язково вміщує колонку (колонки) первинного ключа, колонку (колонки) зовнішніх ключів (це дублюючі назви первинних ключів сторонніх таблиць) та одну колонку або кілька колонок властивостей об'єкту.

2. Таблиця зв'язку, яка не має первинного ключа і вміщує набір зовнішніх ключів.

Зв'язки між таблицями мають назву 1:1 («один-до-одного», рис. 4.12.3); 1:М («один-до-багатьох», рис. 4.12.4); М:1 («багато-до-одного»; на рис. 4.12.4, читається навпаки «один-до-багатьох», 1:М), М:М («багато-до-багатьох», рис. 4.12.5).

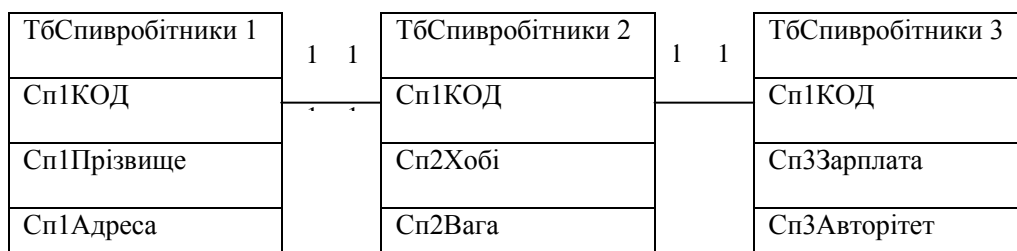


Рис. 4.12.3. Зв'язок 1:1 («один-до-одного»)

З рис. 4.12.3 бачимо, що зв'язок 1:1 означає, що одна таблиця розділена на три різні таблиці з однаковою кількістю рядків: ТбСпівробітники 1, ТбСпівробітники 2, ТбСпівробітники 3. В цих трьох таблицях один й той же рядок (з однаковим числовим значенням Сп1КОД) відноситься до даних одного співробітника, розділених на три окремі частки в трьох різних таблицях. Звичайно це робиться для зручності обробки даних.

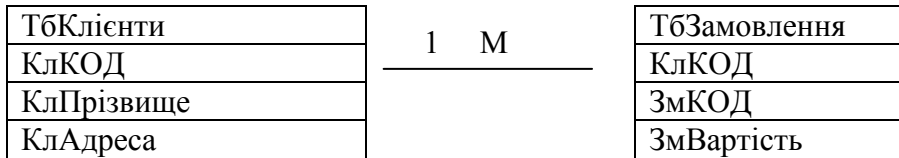


Рис. 4.12.4. Зв'язок «один-до-багатьох»

На рис. 4.12.4 відображений зв'язок 1:М між двома таблицями ТбКлієнти та ТбЗамовлення, яке означає, що один клієнт з первинним ключем КлКОД може зробити багато замовлень. Складеним унікальним ключем замовлення є код клієнта КлКОД (порядковий номер клієнта) плюс код замовлення ЗмКОД (порядковий номер замовлення). Таблиця ТбКлієнти має додаткові власні колонки: КлПрізвище, КлАдреса, а таблиця ТбЗамовлення – колонку ЗмВартість. До таблиці ТбЗамовлення при необхідності можна було б додати назви колонок ЗмКількість, ЗмВага, ЗмДатаЗамовлення, ЗмДатаДоставки у склад, ЗмВиконавець, ЗмДатаВиконання тощо.

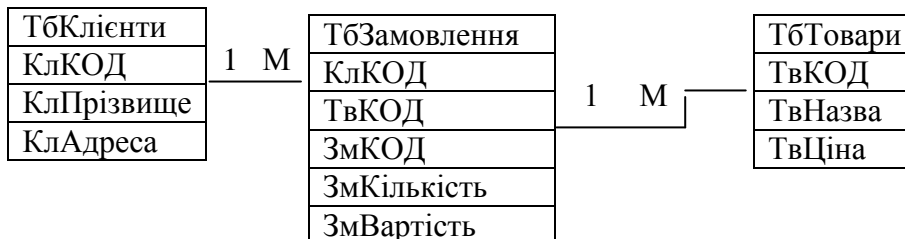


Рис. 4.12.5. Зв'язок М:М («багато-до-багатьох») таблиці ТбЗамовлення

На рис. 4.12.5 зв'язок М:М («багато-до-багатьох») таблиці ТбЗамовлення відображує, що на одного клієнта з кодом КлКОД може приходиться кілька замовлень з кодом ЗмКОД, а на один товар з кодом ТвКОД теж може приходиться кілька замовлень з кодом ЗмКОД.

З пов'язаних між собою таблиць ми можемо отримати звіти – таблиці, які складаються з колонок різних таблиць. Отримані дані можна фільтрувати, упорядковувати, використовувати у розрахунках.

4.12.2. Концептуальна модель реляційної бази даних

Принципи реляційної моделі були сформульовані у 1969–1970 рр. Е. Ф. Коддом (E. F. Codd).

База даних(БД) забезпечує пошук інформації, захист даних від випадкової втрати або псування і економно використовує людські і технічні ресурси.

Будування концептуальної моделі є процесом моделювання смислового наповнення бази даних. У даному випадку вважаємо, що Службовці оформлюють кілька Проектів шляхом створення Замовлень, що складаються з Деталей, які вироблюють різні Постачальники.

У реляційній базі даних всі дані зберігаються в таблицях. Тому концептуальна модель переводиться до табличної структурної форми, а розміщені нижче під назвою таблиці рядки структурної форми визначають назви колонок таблиці. Цілісність даних в реляційній базі даних ґрунтується **на концепції ключів**.

Первинний ключ – це заголовок колонки таблиці, в якій числові значення елементів ідентифікують записи (дані) у рядках таблиці. Так, у таблиці «ТбПостачальники» – ключем стане ідентифікаційний номер постачальника з позначенням «ПсКОД»; у таблиці «ТбДеталі» – номер деталі з позначенням «ДтКОД», у таблиці «ТбПроект» – номер проекту з позначенням «ПрКОД», у таблиці «ТбСлужбовці» – номер службовця з позначенням «СлКОД», а в таблиці «ТбЗамовлення» – номер замовлення з позначенням «ЗмКОД».

Зовнішній ключ – це заголовок колонки таблиці, який існує в кількох таблицях і є первинним ключем однієї з цих таблиць. Зв'язок проводимо від первинного ключа однієї таблиці до зовнішнього ключа іншої таблиці.

Концептуальна модель складається з наступних **трьох основних компонентів**: таблиць, колонок для таблиць та зв'язків між таблицями (рис. 4.12.6).

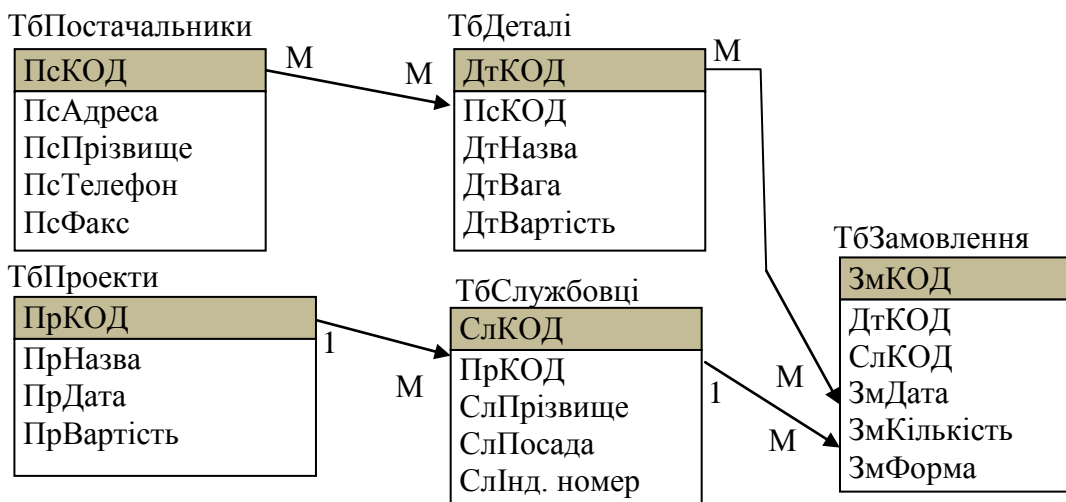


Рис. 4.12.6. Концептуальна модель бази даних

1. **Таблиці** відображують сутності реального світу, які можуть існувати незалежно. У даному випадку таблицями є: ТбПроект, ТбПостачальники, ТбДеталі, ТбСлужбовці, ТбЗамовлення. Таблиці представляється у концептуальній моделі прямокутниками, в яких зазначені їх ім'я.

2. **Колонки** описують таблицю, є своєрідними ознаками таблиці і відповідних об'єктів. У даному випадку:

- «ТбПроектам» відповідають колонки: ПрКОД, ПрДата, ПрВартість.
- «ТбПостачальникам» відповідають колонки: ПсКОД, ПсАдреса, ПсТелефон, ПсФакс.
- «ТбДеталям» відповідають колонки: ДтКОД, ПсКОД, ДтНазва, ДтВага, ДтВартість.
- «ТбСлужбовцям» відповідають колонки: СлКОД, СлПрізвище, СлПосада, СлШнд. номер.
- «ТбЗамовленням» відповідають колонки: ЗмКОД, ДтКОД, СлКОД, ЗмДата, ЗмКількість, ЗмФорма.

3. **Зв'язки**. Зв'язок відображує взаємодію між таблицями. На діаграмі зв'язки зображені *прямими лініями зі стрілками*, які з'єднують таблиці.

Зв'язок між *ТбПроекти і ТбСлужбовці* позначений як «один до багатьох (1:М)», тому що один проект можуть розробляти кілька службовців. Згідно даному зв'язку один службовець не може прийняти участь у оформленні *Замовлень* для кількох *Проектів*.

Зв'язок між *ТбСлужбовці і ТбЗамовлення* позначений як «один до багатьох (1:М)», тому що один службовець може оформлювати багато замовлень. Згідно даному зв'язку одне замовлення не може оформлюватись кількома службовцями.

Зв'язок між *ТбЗамовлення і ТбДеталі* позначений як «багато до багатьох (М:М)», тому що багато замовлень можуть бути на багато деталей (кожне замовлення може охоплювати багато деталей, а кожна деталь може входити у багато замовлень).

Зв'язок між *ТбПостачальники і ТбДеталі* позначений як «багато до багатьох (М:М)», тому що кілька постачальників можуть поставляти кілька деталей (один постачальник може постачати багато деталей, і одна деталь може постачатись багатьма постачальниками).

З рис. 4.12.6 випливає, що «розпізнавання образу» конкретного замовлення може супроводжуватись аналізом ознак різних типів – цифрових, логічних, текстових та інших. Можливість сортування, фільтрації та комбінаторного групування об'єктів надають додаткові переваги для розпізнавання образів за допомогою БД.

Нормалізація таблиць БД відбувається послідовно, від першої нормальної форми 1НФ до останньої 5НФ при умові, що кожна наступна нормальна форма в деякому сенсі краще попередньої, а всі попередні властивості нормальних форм зберігаються (рис. 4.12.7):

- 1НФ – перша нормальна форма;

- 2НФ – друга нормальна форма;
- 3НФ – третя нормальна форма;
- БКНФ – нормальна форма Бойса-Кодда;
- 4НФ – четверта нормальна форма;
- (5НФ чи ПЗНФ) – п'ята нормальна форма, чи нормальна форма проєкції – з'єднання.

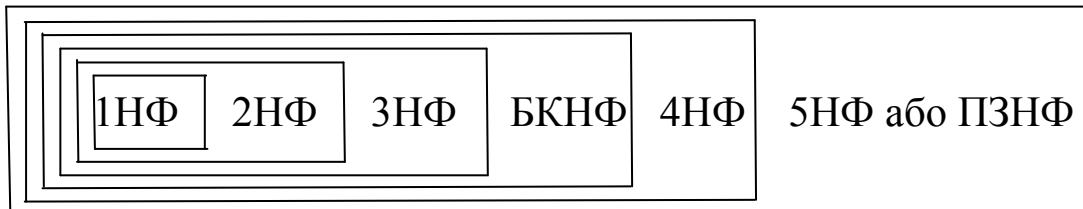


Рис. 4.12.7. Послідовність виконання нормальних форм

Метод нормалізації полягає у декомпозиції таблиці, що знаходиться в попередній нормальній формі, в дві або більшу кількість таблиць, які задовольняють вимогам наступної нормальної форми.

При нормалізації дотримуються фундаментального в теорії реляційних баз даних поняття **функціональної залежності**.

Вважається, що в **таблиці $R(X, Y)$** колонка Y **функціонально залежить** від колонки X (X і Y можуть складатись з кількох атомарних колонок) в тому випадку, якщо кожному значенню X відповідає в точності одне значення Y : $R.X \rightarrow R.Y$. Функціональна залежність $R.X \rightarrow R.Y$ називається **повною**, якщо колонка Y не залежить функціонально від будь-якої підмножини колонки X . Функціональна залежність $R.X \rightarrow R.Y$ називається **транзитивною**, якщо існує така колонка Z , що існують функціональні залежності $R.X \rightarrow R.Z$ і $R.Z \rightarrow R.Y$ і відсутні функціональні залежності $R.Y \rightarrow R.Z$. Дві або більше колонки **взаємно незалежні**, якщо жодна з цих колонок не є функціонально залежною від інших.

Неключовою колонкою зветься будь-яка колонка таблиці, яка не входить до складу ключа (зокрема, первинного).

Можливим ключем таблиці зветься його атомарна або складена колонка, значення яких цілком функціонально визначають значення всіх інших колонок таблиці.

Дві або більше колонок таблиці є **взаємно незалежними**, якщо жодна з цих колонок не залежить від інших колонок таблиці.

Таблиця R знаходиться у першій нормальній формі (1НФ), якщо:

- значення колонок є атомарними;
- немає колонок, які повторюються;
- немає рядків, які повторюються;
- рядки не впорядковані;
- колонки не впорядковані (можна переставляти стовпці).

Таблиця R знаходиться у другій нормальній формі (2НФ), якщо вона перебуває в 1НФ, і кожна неключова колонка повністю функціонально залежить від первинного ключа.

Якщо припустити наявність декількох ключів, то визначення прийме наступний вигляд.

Таблиця знаходиться у другій нормальній формі (2НФ), якщо вона перебуває в 1НФ, і кожна неключова колонка повністю функціонально залежить від кожного ключа.

Тут і далі ми не будемо наводити приклади для відношень з декількома ключами. Вони занадто громіздкі і відносяться до ситуацій, які не часто зустрічаються на практиці.

Таблиця R знаходиться в третій нормальній формі (3НФ), якщо вона перебуває в 2НФ і кожна з неключових колонок нетранзитивно залежить від первинного ключа (визначення дійсне при існуванні єдиного ключа).

Якщо припустити наявність декількох ключів, то визначення прийме наступний форму.

Таблиця R знаходиться в третій нормальній формі (3НФ), якщо вона перебуває в 2НФ, і кожна з неключових колонок є взаємно незалежною і цілком залежить від первинного ключа R.

У більшості випадків на практиці *третя нормальна форма таблиць є достатною, і на цьому процес проектування реляційної бази даних зазвичай закінчується.*

Однак іноді корисно продовжити процес нормалізації.

Таблиця R знаходиться в нормальній формі Бойса-Кодда (БКНФ), якщо вона перебуває в 3НФ і кожний *детермінант* (колонка, від якої повністю функціонально залежить деяка інша колонка) є можливим ключем. В результаті в формі БКНФ: кожний детермінант є можливим ключем; таблиця має два або більше альтернативних (потенційних) ключів; альтернативні ключі є складеними; складені ключі перекриваються (мають один чи кілька спільних колонок).

Наприклад, в ТбСпівробітники особистість співробітника цілком визначається як його номером, так і ПІБ (прізвище, ім'я, по батькові). В результаті той факт, що існують функціональні залежності колонок таблиці від колонки, яка є частиною первинного ключа, призводить до наступної аномалії: щоб узгодженим чином змінити ПІБ співробітника з даним номером, нам потрібно модифікувати всі кортежі, що включають його номер.

Четверта нормальна форма.

В таблиці R(A, B, C) існує *багатозначна залежність*, яку будемо позначати як $R.A \twoheadrightarrow R.B$, у випадку, якщо множина значень B, яка відповідає парі значень A і C, залежить тільки від A і не залежить від C.

Р. Фейджин довів, що якщо існує *багатозначна залежність* $R.A \twoheadrightarrow R.B$, то одночасно існує й *багатозначна залежність* $R.A \twoheadrightarrow R.C$. Цю

зв'язану між собою пару $R.A \rightarrow \rightarrow R.B$ та $R.A \rightarrow \rightarrow R.C$ позначають виразом $A \rightarrow \rightarrow B|C$. Наявність виразу $A \rightarrow \rightarrow B|C$ дозволяє таблицю $R(A, B, C)$ віднести до *четвертої нормальної форми* і розділити її на дві таблиці $R(A, B)$ та $R(A, C)$.

П'ята нормальна форма.

Таблиця $R(X, Y, \dots, Z)$ знаходиться в п'ятій нормальній формі (5НФ) чи нормальній формі проєкції – з'єднання (ПЗНФ) лише у випадку, коли будь-яка залежність сполучення в R створюється з проєкцій за ключовими атрибутами, які у сукупності є можливим ключем відношення.

На практиці 5НФ майже не зустрічається.

У цьому випадку не розглянуті на прикладах особливості обробки табличних даних (їх нормалізації, створення запитів, форм, звітів), які детально описані у відповідних реляційних БД.

4.13. Бази знань

База знань (БЗ) – це особливого роду база даних (БД), яка розроблена для управління знаннями (для збору, зберігання, пошуку і видачі знань) і яка відрізняється від БД наступними властивостями [14]:

1. **Активність**, під якою розуміють ініціалізацію дій базою знань під введені дані і програми (команди). Звичайно вважають, що дані пасивні, а команди активні, бо дані надаються за вимогою. У БЗ знання можуть ініціювати появу нових фактів та зв'язків з найближчих за сенсом розділів БЗ.

2. **Формалізація представлення знань і здійснення над ними операцій** за допомогою *моделей знань*: формалізують механізми логічного виведення нових знань; знання описують і маніпулюють ними на спеціальних мовах; створюють процедури порівняння нових знань з уже існуючими.

3. **Рекурсивна структурованість знань** дозволяє використовувати принцип *наслідування* і дає можливість робити агрегування і декомпозицію інформаційних одиниць. Для знань виконується *«принцип матрьошки»*, тобто рекурсивне вкладення одних інформаційних одиниць в інші.

4. **Зв'язаність**. У БЗ між інформаційними одиницями встановлюються зв'язки різного типу, що характеризують відносини між об'єктами, явищами і т. п. Наприклад, між певними одиницями знань можна установити зв'язки: причинно-наслідкові, просторові, семантичні, часові; зв'язків типу род – вид, клас – представник класу, тип – підтип, частка – ціле, причина – наслідок, одночасно, бути рядом, «елемент – множина», «ситуація – підситуація». Зв'язки дозволяють аналізувати знання на сумісність, несуперечність і т. п., що важко реалізувати при збереженні традиційних масивів даних.

5. **Інтерпретованість**. Знання однакової природи, призначення та однакової дії (однакових наслідків) об'єднуються у окремі програмні блоки. Кожна інформаційна одиниця забезпечується іменем, пояснювальним описом та послідовністю імен тих множин чи класів, до яких ця одиниця входить.

6. **Семантичну близькість знань**, схожість та семантичні відношення між об'єктами визначають за рахунок відповідних шкал.

7. **Кількісна оцінка якісних факторів**: якісні фактори типу «подобається» та «не подобається», «має/не має авторитету», тощо, бажано щоб мали суб'єктивну кількісну оцінку у межах 0...100 %, яка встановлюється експертом або користувачем на базі одночасного порівняння з браком (0 %) та взірцем (100 %). **Використання квантованих сигналів** замість аналогових: замість числового позначення температури використовують поняття «холодно», «тепло», «гаряче».

Перелічені особливості інформаційних одиниць визначають ту грань, за якою дані перетворюються у знання, а бази даних (БД) перетворюються у бази знань (БЗ).

Сукупність засобів, які забезпечують роботу зі знаннями, створюють систему управління базою знань (СУБЗ), яка дозволяє вибрати знання із БЗ та доповнювати їх.

Розділ штучного інтелекту, що вивчає бази знань і методи роботи зі знаннями, називається **інженерією знань**.

Найбільш важливий параметр БЗ – якість знань. Кращі БЗ містять достовірну і свіжу інформацію, мають довершені системи пошуку інформації і ретельно продуману структуру і формат знань.

Залежно від рівня складності систем, в яких застосовуються бази знань, розрізняють:

- 1) БЗ всесвітнього масштабу – наприклад, Інтернет чи Вікіпедія;
- 2) БЗ галузеві наприклад, автомобільної енциклопедії;
- 3) БЗ організацій;
- 4) БЗ експертних систем.

Прості бази знань можуть використовуватися для створення експертних систем і зберігання даних організації: документації, інструкцій, статей технічного забезпечення. Головна мета створення таких БЗ – допомогти менш досвідченішим людям знайти існуючий опис способу вирішення якої-небудь проблеми предметної області.

4.14. Сценарії

Сценарії були запропоновані у 1975р. в СРСР д-р техн. наук, професором Поспеловим Д. А. [97].

Сценарієм зветься *фреймоподібна структура знань, яка описує стандартну послідовність подій, характерну для певного процесу чи для певної ситуації, або причинно-наслідкові зв'язки між подіями.*

Сценарій включає: початкові умови; ролі – окремих учасників; результати дій, що виконують окремі учасники (ролі); контекст сценарію; послідовність сцен, які описують розвиток сценарію у часі.

Сценарії описують способи досягнення цілей дійових осіб сценарію (наприклад, обід в ресторані, відрядження, політ літака, вступ до вузу

тощо). В системах розпізнавання образів сценарій дозволяє прослідкувати зміну особливостей і ознак образу на протязі розвитку сценарію.

Сценарій «Купівля мороженого» розглядається у вигляді ряду сцен:

Сцена 1 (перевірка грошей): витягнути гаманець; перелічити гроші (припустимо, що грошей на морозиво вистачає);

Сцена 2 (черга): спитати, за ким стати у чергу; стати у чергу; дочекатись можливості замовлення морозива;

Сцена 3 (купівля): замовити назву морозива; віддати гроші; отримати морозиво; отримати здачу; вийти з черги;

Сцена 4 (споживання морозива): розгорнути морозиво; з'їсти морозиво; обгортку кинути в урну.

У сценаріях використовуються відношення типу «причина – наслідок», «дія», «засіб дії» та ін.

Типовий сценарій можна розглядати як семантичну мережу, всі зв'язки якої мають тип «причина – наслідок» або «бути раніше – бути пізніше». Це ще раз підкреслює тісний зв'язок між семантичними мережами та фреймами.

Як і інші типи фреймів, сценарії можна використовувати для поповнення опису ситуацій. Так, якщо інтелектуальна система сприймає текст «Іванов відвідав ресторан», вона зіставляє його з відомими їй сценаріями і на основі цього може відповідати на запитання, пов'язані з тим, що саме робив Іванов у ресторані.

5. Стохастичний аналіз

5.1. Початкові відомості з теорії стохастичних рішень

Алгоритми розпізнавання може бути заснованим на статистичних даних [3]. При цьому безпосередньо використовується повна апріорна інформація, а якщо її немає, то використовується процедура навчання та самонавчання. Використовують також припущення про однакову апріорну ймовірність появи класів, якщо не існує іншої інформації.

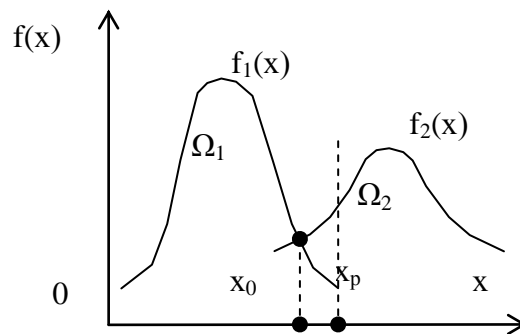


Рис. 5.1.1. Густина розподілу ймовірностей $f_1(x)$ та $f_2(x)$ значень ознаки « x » для класів Ω_1 та Ω_2

Для ймовірнісних систем РО потрібно визначити (експериментально, теоретично або моделюванням) характеристики цих розподілів: $f_i(x_1, x_2, \dots, x_j, \dots, x_n)$ — густину розподілу випадкових подій, де $j=1, 2, \dots, n$ — порядковий номер події x_j , $i=1, 2, \dots, m$ — порядковий номер класу; $P(\Omega_i)$ — апріорну ймовірність того, що випадково обраний із загальної сукупності об'єкт належить i -му класу Ω_i .

Розглянемо сукупність об'єктів з ознакою « x », розділених на класи Ω_1, Ω_2 . Вважаємо, що нам відомі умовні густини розподілу ймовірностей $f_1(x), f_2(x)$ значень ознаки « x » еталонних об'єктів класу Ω_1, Ω_2 , а також апріорні ймовірності $P(\Omega_1), P(\Omega_2)$ появи об'єктів. Для нормального закону розподілу (Гауса) умовні густини розподілу ймовірностей у залежності від ознаки « x » описуються рівняннями (рис. 5.1.1):

$$f_1(x) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x-m_1)^2}{2\sigma_1^2}}; \quad f_2(x) = \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{(x-m_2)^2}{2\sigma_2^2}},$$

де x — ознака об'єкта; m_1, m_2 — математичне очікування об'єктів класів Ω_1, Ω_2 ; σ_1, σ_2 — середнеквадратичне відхилення ознаки « x » об'єктів класів Ω_1, Ω_2 .

Якщо вважати, що, наприклад, для невідомого літака система РО своїми датчиками визначила (із природною випадковою похибкою) розмах крил x_p , то з визначеною ймовірністю цей вимір можна віднести до

кожного з класів, але якщо величина x_p розміщена ближче до відповідного математичного очікування (одного з центрів розсіяння), то ймовірність віднесення його до цього класу більша.

Здавалося б, що для класифікації природним і найбільш вірним рішенням було б отримати при навчанні системи значення x_0 – точки пересічення еталонних кривих $f_1(x), f_2(x)$ із рівняння

$$\lambda_0 = \frac{f_2(x_0)}{f_1(x_0)} = 1 = \frac{\sigma_1 e^{-(x_0 - m_2)^2 / (2\sigma_2^2)}}{\sigma_2 e^{-(x_0 - m_1)^2 / (2\sigma_1^2)}} = \frac{\sigma_1}{\sigma_2} e^{[(x_0 - m_1)^2 / (2\sigma_1^2) - (x_0 - m_2)^2 / (2\sigma_2^2)]}$$

де $\lambda_0 = 1$ – коефіцієнт правдоподібності.

Тоді вимірjana ознака x_p об'єкта при умові $x_p > x_0$ вказує на належність об'єкта до класу Ω_2 , а якщо $x_p < x_0$, то об'єкт відноситься до класу Ω_1 .

Таке вирішальне правило в дійсності і використовується для розрахунку x_0 , але з невеликим і в той же час, дуже важливим уточненням: коефіцієнт правдоподібності λ_0 визначають з врахуванням ризику невірної рішення. Тоді коефіцієнт правдоподібності λ_0 відрізняється від «1», а уточнена величина x_0 знаходиться поблизу точки пересічення еталонних кривих $f_1(x), f_2(x)$.

Пояснюється цей підхід тим, що ми маємо справу не з детермінованими, а з випадковими подіями. Тому завжди існує небезпека виникнення помилки у визначенні класу об'єкта. Відповідно виникає проблема оцінки впливу цієї помилки на подальший розвиток подій. Наприклад, якщо система РО призначена для виявлення й класифікації літаючих ракет (клас Ω_1), а транспортні літаки та літаючих гусаків відносить до класу Ω_2 , то ризик, коли дійсні ракети прийняли за гусей значно більший, ніж у протилежному випадку – коли гусей прийняли за ракети. Під ризиком розуміють втрати, пов'язані з вірними та невірними рішеннями.

Для уточнення коефіцієнта правдоподібності λ_0 треба визначити ймовірність помилки у класифікації та ризик прийняття рішення.

5.2. Ймовірності помилок класифікації одновимірних образів двох класів

При класифікації з урахуванням ймовірності виникнення подій можливі два роди помилок, які можна визначити за допомогою таблиць:

1. Помилка 1-го роду, коли об'єкт класу Ω_1 помилково відносять до класу Ω_2 . Умовна ймовірність помилки 1-го роду

$$\Delta P_{\Omega_1} = \int_{x_0}^{\infty} f_1(x) dx .$$

Якщо використовувати термінологію теорії статистичних рішень, то кажуть, що «помилково вибрали гіпотезу H_2 при справедливій гіпотезі H_1 ».

2. Помилка 2-го роду, коли об'єкт класу Ω_2 помилково відносять до класу Ω_1 . Умовна ймовірність помилки 2-го роду:

$$\Delta P_{\Omega_2} = \int_{-\infty}^{x_0} f_2(x) dx.$$

У термінах статистичних рішень кажуть, що «помилково вибрали гіпотезу H_1 при справедливій гіпотезі H_2 ».

5.3. Визначення коефіцієнта правдоподібності з врахуванням ризику рішення для одновимірних образів двох класів

При розділі простору ознак x на два півпростори (два класи) враховують втрати, пов'язані з вірними та помилковими рішеннями.

Функція втрат, яка характеризує втрати від помилок 1-го та 2-го родів, а також втрати від вірних рішень, складають платіжну матрицю у вигляді

$$C = \begin{vmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{vmatrix},$$

де c_{11} , c_{22} – втрати, пов'язані з вірними рішеннями (звичайно приймають $c_{ii} = 0$).

c_{12} , c_{21} – втрати, пов'язані з помилками 1-го та 2-го роду (об'єкт класу Ω_1 помилково відносять до класу Ω_2 ; об'єкт класу Ω_2 помилково відносять до класу Ω_1).

Математичне очікування ризику або середній ризик при багаторазовому розпізнаванні невідомих об'єктів дорівнює сумі втрат, пов'язаних з невірними та вірними рішеннями, з урахуванням ймовірностей їх появи і апіорних ймовірностей надходження на вхід системи РО об'єктів класів Ω_1 , Ω_2 :

$$R = P(\Omega_1) \left[c_{11} \int_{-\infty}^{x_0} f_1(x) dx + c_{12} \int_{x_0}^{\infty} f_1(x) dx \right] + P(\Omega_2) \left[c_{22} \int_{x_0}^{\infty} f_2(x) dx + c_{21} \int_{-\infty}^{x_0} f_2(x) dx \right].$$

Шляхом прирівнювання похідної dR/dx нулю, визначаємо коефіцієнт правдоподібності (відношення функцій) $\lambda_0 = \frac{f_2(x_0)}{f_1(x_0)}$, при якому середній ризик буде мінімальним.

При цьому вважаємо, що $x = x_0$, тобто це відношення відповідає мінімізації ризику

$$\begin{aligned}
 |dR/dx|_{x=x_0} &= P(\Omega_1) \left[c_{11} f_1(x) \Big|_{-\infty}^{x_0} + c_{12} f_1(x) \Big|_{x_0}^{\infty} \right] + P(\Omega_2) \left[c_{22} f_2(x) \Big|_{x_0}^{\infty} + c_{21} f_2(x) \Big|_{-\infty}^{x_0} \right] = \\
 &= P(\Omega_1) \left[c_{11} f_1(x_0) - c_{12} f_1(x_0) \right] + P(\Omega_2) \left[-c_{22} f_2(x_0) + c_{21} f_2(x_0) \right] = \\
 &= P(\Omega_1) f_1(x_0) (c_{11} - c_{12}) + P(\Omega_2) f_2(x_0) (c_{22} - c_{21}) = 0.
 \end{aligned}$$

Тут врахували, що $f_1(\pm\infty) = f_2(\pm\infty) = 0$. В результаті отримуємо значення λ_0 , яке зветься пороговим (критичним) коефіцієнтом правдоподібності

$$P(\Omega_1) f_1(x_0) (c_{11} - c_{12}) = P(\Omega_2) f_2(x_0) (c_{22} - c_{21});$$

$$\lambda_0 = \frac{f_2(x_0)}{f_1(x_0)} = \frac{P(\Omega_1)(c_{11} - c_{12})}{P(\Omega_2)(c_{22} - c_{21})}.$$

Значення порогового (критичного) коефіцієнта правдоподібності λ_0 використовується для визначення величини x_0 згідно виразу $\lambda_0 = \frac{f_2(x_0)}{f_1(x_0)}$.

5.4. Мережі відносних стохастичних та детермінованих потоків

5.4.1. Умовна ймовірність та формула повної ймовірності

Якщо вибірка випадкових подій n_0 за несумісними гіпотезами $H = (h_1, h_2, \dots, h_\lambda, \dots, h_\Lambda)$, де $\lambda = 1, 2, \dots, \Lambda$ – порядковий номер гіпотези, розділена на відповідні окремі групи nh_λ , то з неї можна отримати формулу для повної групи ймовірностей, підсумок яких дорівнює 1

$$n_0 = \sum_{\lambda=1}^{\Lambda} n_{h\lambda}; \quad 1 = \sum_{\lambda=1}^{\Lambda} n_{h\lambda} / n_0; \quad 1 = \sum_{\lambda=1}^{\Lambda} p_{h\lambda}, \quad (5.4.1)$$

де $p_{h\lambda} = n_{h\lambda} / n_0$ – ймовірність гіпотези h_λ , $\lambda = 1, 2, \dots, \Lambda$ – порядковий номер гіпотези.

Якщо кожна група nh_λ має деяку власну підгрупу nA_λ з випадковими подіями A_λ , то отримуємо аналогічні вирази для повної групи умовних ймовірностей

$$n_A = \sum_{\lambda=1}^{\Lambda} n_{A\lambda}; \quad 1 = \sum_{\lambda=1}^{\Lambda} n_{A\lambda} / n_A; \quad 1 = \sum_{\lambda=1}^{\Lambda} p(A_\lambda | A) \quad (5.4.2)$$

де $p(A_\lambda | A) = n_{A\lambda} / n_A = p_{A\lambda}$ представляє умовну ймовірність: «ймовірність спостереження подій A_λ , якщо всі події A відбулись». Вертикальна риска у позначенні $p(A_\lambda | A)$ відноситься до подій A і тлумачиться словом «якщо».

З формул (5.4.2) отримуємо формули повної ймовірності:

$$n_A = \sum_{\lambda=1}^{\Lambda} n_{A\lambda}; \quad p(A) = \frac{n_A}{n_0} = \sum_{\lambda=1}^{\Lambda} \frac{n_{A\lambda}}{n_0} = \sum_{\lambda=1}^{\Lambda} \left(\frac{n_{A\lambda}}{n_A} \cdot \frac{n_A}{n_0} \right) = \sum_{\lambda=1}^{\Lambda} [p(A_\lambda | A) p(A)]; \quad (5.4.3)$$

$$p(A) = \frac{n_A}{n_0} = \sum_{\lambda=1}^{\Lambda} \frac{n_{A\lambda}}{n_0} = \sum_{\lambda=1}^{\Lambda} \left(\frac{n_{A\lambda}}{n_{h\lambda}} \cdot \frac{n_{h\lambda}}{n_0} \right) = \sum_{h=1}^H p(A_\lambda | h_\lambda) p_{h\lambda} \quad (5.4.4)$$

5.4.2. Визначення формули Байєса

Припустимо, що у вибірці спостережених вогнищ n_0 ймовірності спостереження окремих подій А (полум'я), В (дим) та їх сумісного спостереження АВ (полум'я та дим) дорівнюють

$$p(A) = \frac{n_A}{n_0}; \quad p(B) = \frac{n_B}{n_0}; \quad p(AB) = \frac{n_{AB}}{n_0}, \quad (5.4.5)$$

де $n_A = n_{A1} + n_{AB}$ – кількість спостережених подій А (полум'я); $n_B = n_{B1} + n_{AB}$ – кількість спостережених подій В (дим); n_{AB} – кількість одночасно спостережених подій (полум'я та дим), яка одночасно входить у множини n_A та n_B .

Умовні ймовірності спостереження події А (якщо події В вже відбулись) та події В (якщо події А вже відбулись) дорівнюють

$$p(A | B) = \frac{n_{AB}}{n_B}, \quad p(B | A) = \frac{n_{AB}}{n_A}. \quad (5.4.6)$$

Ймовірність одночасного спостереження двох подій дорівнює

$$p(AB) = \frac{n_{AB}}{n_0} = \frac{n_{AB}}{n_B} \cdot \frac{n_B}{n_0} = p(A | B) p(B); \quad (5.4.7)$$

$$p(AB) = \frac{n_{AB}}{n_0} = \frac{n_{AB}}{n_A} \cdot \frac{n_A}{n_0} = p(B | A) p(A). \quad (5.4.8)$$

Якщо прирівняти (5.4.7) та (5.4.8), то отримуємо правило Байєса

$$p(A | B) = \frac{p(B | A) p(A)}{p(B)}, \quad p(B | A) = \frac{p(A | B) p(B)}{p(A)}. \quad (5.4.9)$$

Напрямки аналізу ймовірнісних мереж за методом Байєса. Для розпізнавання кожної групи – гіпотези вектора $H = (h_1, h_2, \dots, h_\lambda, \dots, h_\Lambda)$, де $\lambda = 1, 2, \dots, \Lambda$ – порядковий номер гіпотези початкової вибірки об'єктів n_0 , використовують ознаки – змінні об'єктів вибірки $X = (x_1, x_2, \dots, x_j, \dots, x_n)$, кожна з яких для окремого об'єкта приймає одне з двох значень: «0 / + 1», «-1/+1», «Орел / Решка» тощо.

У залежності від кількості ознак-змінних задачі Байєса можна розділити на дві групи:

- вибірка з однією ознакою x_1 ;
- вибірка з кількома ознаками $x_j, j = 1, \dots, n$.

Кожна з цих вибірок може мати задану кількість взаємно несумісних гіпотез $H = (h_1, h_2, \dots, h_\lambda, \dots, h_\Lambda)$, де $\lambda = 1, 2, \dots, \Lambda$ – порядковий номер гіпотези, ймовірності яких $p_{h\lambda}$ складають повну групу.

1. Вибірка з об'єктами, що описуються однією ознакою x_1 .

Одна ознака – змінна x_1 може приймати одне з двох значень: або x_{11} ; $h_{\lambda}=1$, або x_{10} ; $h_{\lambda}=0$. Тому кожна група гіпотези розділяється на дві підгрупи у вигляді $[(nh_{11}; x_{11}, nh_{10}; x_{10}), \dots, (nh_{\lambda 1}; x_{11}, nh_{\lambda 0}; x_{10}), \dots, (nh_{\Lambda 1}; x_{11}, nh_{\Lambda 0}; x_{10})]$ з дотриманням рівності $nh_{\lambda} = nh_{\lambda 1} + nh_{\lambda 0}$.

Ми можемо:

– Визначити апіорну ймовірність спостереження гіпотези h_{λ} , $\lambda = 1, 2, \dots, \Lambda$, за формулою

$$P_{h_{\lambda}} = n_{h_{\lambda}} / n_0, \quad (5.4.10)$$

де $n_0 = (nh_{11} + nh_{10} + \dots + nh_{\lambda 1} + \dots + nh_{\lambda 0})$.

– Визначити апіорну ймовірність спостереження ознаки $x_{11} = +1$ гіпотези h_{λ} за формулою

$$P_{h_{\lambda}, x_{11}} = \frac{n_{h_{\lambda}, x_{11}}}{n_0} = \frac{n_{h_{\lambda}}}{n_0} \cdot \frac{n_{h_{\lambda}, x_{11}}}{n_{h_{\lambda}}} = P_{h_{\lambda}} P(x_{11}; h_{\lambda} | h_{\lambda}), \quad (5.4.11)$$

де $p_{h_{\lambda}} = n_{h_{\lambda}} / n_0$ – апіорна ймовірність гіпотези h_{λ} ; $p(x_{11}; h_{\lambda} | h_{\lambda}) = n_{h_{\lambda}, x_{11}} / n_{h_{\lambda}}$ – умовна ймовірність спостереження ознаки $x_{11}; h_{\lambda}=1$ в групі $n_{h_{\lambda}}$ (гіпотези h_{λ}); $p(x_{10}; h_{\lambda} | h_{\lambda}) = 1 - p(x_{11}; h_{\lambda} | h_{\lambda})$ – умовна ймовірність спостереження ознаки $x_{10}; h_{\lambda}=0$ (або $x_{10}; h_{\lambda}=-1$) в групі $n_{h_{\lambda}}$.

– Розрахувати ймовірність спостереження об'єктів з заданим значенням ознаки – змінної $x_{11}; h_{\lambda}=+1$ для всієї вибірки

$$P_{x_{11}} = \sum_{\lambda=1}^{\Lambda} \frac{n_{h_{\lambda}, x_{11}}}{n_0} = \sum_{\lambda=1}^{\Lambda} \left(\frac{n_{h_{\lambda}}}{n_0} \cdot \frac{n_{h_{\lambda}, x_{11}}}{n_{h_{\lambda}}} \right) = \sum_{\lambda=1}^{\Lambda} P_{h_{\lambda}} P(x_{11}; h_{\lambda} | h_{\lambda}). \quad (5.4.12)$$

З формул (5.4.11) та (5.4.12) визначаємо формулу Байєса для однієї змінної при наявності кількох гіпотез, за якою ймовірність спостереження гіпотези h_{λ} при умові появи ознаки $x_{11}; h_{\lambda}=+1$ дорівнює

$$p(h_{\lambda} | x_{11}) = \frac{n_{h_{\lambda}, x_{11}}}{\sum_{\lambda=1}^{\Lambda} n_{h_{\lambda}, x_{11}}} = \frac{n_{h_{\lambda}, x_{11}}}{n_0 \frac{1}{n_0} \sum_{\lambda=1}^{\Lambda} n_{h_{\lambda}, x_{11}}} = \frac{P_{h_{\lambda}, x_{11}}}{P_{x_{11}}} = \frac{P_{h_{\lambda}} P(x_{11}; h_{\lambda} | h_{\lambda})}{\sum_{\lambda=1}^{\Lambda} P_{h_{\lambda}} P(x_{11}; h_{\lambda} | h_{\lambda})}. \quad (5.4.13)$$

2. У вибірці з об'єктами, що описуються багатьма змінними $X = (x_1, x_2, \dots, x_j, \dots, x_n)$, при умові незалежності змінних, елементи вектора X для однієї гіпотези спостерігаються одночасно. Тому в формулі (5.4.13) ймовірність спостереження однієї гіпотези h_{λ} замінюється добутком ймовірностей спостереження кількох ознак – змінних вектора X . В результаті отримуємо відому формулу наївного Байєсового класифікатора при спостереженні подій – гіпотез (об'єктів) багатомірного простору, за якою ймовірність спостереження гіпотези h_{λ} при класифікації невідомого вектора X

$$P_{h_{\lambda}}^D = \frac{P_{h_{\lambda}} \prod_{j=1}^n p(x_j | h_{\lambda})}{\sum_{\lambda=1}^{\Lambda} \left[P_{h_{\lambda}} \prod_{j=1}^n p(x_j | h_{\lambda}) \right]}, \quad (5.4.14)$$

де $p_{h\lambda} = nh_{\lambda}/n_0$ – апіорна ймовірність гіпотези h_{λ} ; $p(x_j|h_{\lambda})$ – ймовірність наявності ознаки x_j в гіпотезі h_{λ} .

Тому що в навчальній групі nh_{λ} гіпотези h_{λ} існують навчальні об'єкти з ознаками $x_j = +1$ та $x_j = -1$, то, внаслідок їх рівнозначності при розпізнаванні образів, при навчанні класифікатора Байєса в окремій таблиці визначають обидва значення ймовірностей у вигляді $p(x_j|h_{\lambda})$ та $p^*(x_j|h_{\lambda})$.

Для конкретного невідомого об'єкта X , який класифікується, використовують з цієї таблиці одну з умовних ймовірностей у залежності від знаку x_j об'єкта класифікації X : або $p(x_j|h_{\lambda})$ або $p^*(x_j|h_{\lambda})$. Якщо при навчанні отримано $p(x_j|h_{\lambda}) = 0$ або $p^*(x_j|h_{\lambda})=0$, то такі значення перетворюються, наприклад, за методом згладжування Лапласа на «приблизний нуль» (див. далі), тому що ймовірність спостереження існуючої гіпотези $p_{h\lambda}D$ не може дорівнювати нулю.

Формула (5.4.14) широко застосовується і підтверджується практикою, але вона має наступний недолік, який збільшує похибку класифікації гіпотези: якщо $p(x_j|h_{\lambda})$ дорівнює нулю, або близька до нуля, то порушується основний принцип статистики: одноосібне «рішення однієї змінної x_j » у виразі $\prod_{j=1}^n p(x_j | h_{\lambda})$ «забороняє» «рішення» всіх інших змінних, навіть якщо всі їх умовні ймовірності $p(x_j|h_{\lambda})$ дорівнюють 1 (тобто якщо всі інші змінні свідчать на користь прийняття гіпотези).

Цього недоліку позбавлена формули Байєсового класифікатора, які рекомендуються до використання [81]:

$$p_{h\lambda}^{\Sigma} = \frac{p_{h\lambda} \sum_{j=1}^n p(x_j | h_{\lambda})}{\sum_{\lambda=1}^{\Lambda} [p_{h\lambda} \sum_{j=1}^n p(x_j | h_{\lambda})]}, \quad (5.4.15)$$

$$p_{h\lambda}^{p\Sigma} = p_{h\lambda} \sum_{j=1}^n p(x_j | h_{\lambda}), \quad (5.4.16)$$

Формули Байєсового класифікатора (5.4.15) та(5.4.16), на відміну від формули (5.4.14), отримані на основі принципу математичного очікування ознак гіпотези, що замінює добуток у формулі (5.4.14) на підсумок. В результаті усуваються відповідні стохастичні похибки класифікації формули (5.4.14): відпадає ніколи не дотримувана і невірна вимога взаємної незалежності ознак x_j (ця незалежність математично означає неможливість розпізнати окремий об'єкт, який повинен мати індивідуальні залежні між собою ознаки); жодна змінна x_j не має переваг перед іншими змінними; цілком природна і нормальна нульова вхідна інформація не псується шляхом згладжування (надання їй не спостереженого числового значення); ліквідується порушення основного принципу статистики, коли одна ознака з малим значенням ймовірності «одноосібно забороняє» обрання гіпотези всупереч протилежному «колективному рішення» всіх інших ознак.

5.4.3. Використання формули Байєса для розпізнавання хвороб

Розглянемо використання формули Байєса для розпізнавання трьох класів (гіпотез, хвороб) $\Omega_1, \Omega_2, \Omega_3$ по чотирьох узагальнених ознаках $X(x_1, x_2, x_3, x_4)$. Навчальна вибірка для розрахунку еталонів (умовних ймовірностей $p(x_j | \Omega_\lambda)$ та $p^*(x_j | \Omega_\lambda)$), задана у табл. 5.4.1 для 15-ті навчальних образів (15 пацієнтів – по 5 пацієнтів на кожну хворобу) при $\lambda = 1, 2, 3; j = 1, 2, 3, 4$.

Навчальні *апріорні ймовірності* спостереження хвороб повинні бути задані на основі попереднього досвіду. Ці дані беруться з «досвіду лікаря», або з статистичних даних. Припустимо, що у нас немає ніяких додаткових відомостей, крім навчальної вибірки, а хвороби, що класифікуються, по своїй небезпечності однакові. Тоді врахуємо, що формула Байєса має справу з повною групою ймовірнісних подій; в цій формулі спостереження класів (гіпотез) вважаємо рівноймовірнісними і тому приймаємо: $p_{\Omega_1}=1/3; p_{\Omega_2}=1/3; p_{\Omega_3}=1/3$.

Таблиця 5.4.1.

Початкові навчальні дані

Хвороби $\Omega_\lambda, \lambda = 1, 2, 3$														
Клас $\Omega_1; p_{\Omega_1}=1/3$					Клас $\Omega_2; p_{\Omega_2}=1/3$					Клас $\Omega_3; p_{\Omega_3}=1/3$				
№ пацієнта	Ознаки хвороби Ω_1				№ пацієнта	Ознаки хвороби Ω_2				№ пацієнта	Ознаки хвороби Ω_3			
	x_1	x_2	x_3	x_4		x_1	x_2	x_3	x_4		x_1	x_2	x_3	x_4
1	-1	1	-1	-1	6	-1	1	1	-1	11	-1	1	1	-1
2	1	-1	-1	1	7	1	1	-1	-1	12	1	-1	1	-1
3	-1	1	-1	1	8	1	-1	1	-1	13	-1	1	1	1
4	1	-1	-1	1	9	1	1	-1	1	5	-1	-1	1	1
5	1	-1	1	-1	10	-1	1	-1	1	15	-1	-1	1	1

У режимі навчання з даних табл. 5.4.1 визначимо умовні ймовірності того, що ознака x_j присутня або відсутня у образі Ω_i (табл. 5.4.2). Розглянемо розрахунок умовних ймовірностей (насправді – відносних частот по апріорним даним) на прикладі для пацієнта № 1 (клас Ω_1 , ознака x_1). Як бачимо з табл. 5.4.1, ознака x_1 спостерігалась у 3-х пацієнтів (для них « $x_1=1$ ») і була відсутні у 2-х пацієнтів (для них « $x_1=-1$ »). Тому умовна ймовірність (відносна частота) наявності ознаки x_1 у образі Ω_1 дорівнює $p(x_1 | \Omega_1) = 3/5 = 0,6$; умовна ймовірність відсутності ознаки x_1 у образі Ω_1 дорівнює $p^*(x_1 | \Omega_1) = 1 - p(x_1 | \Omega_1) = 1 - 3/5 = 0,4$. Розрахунки умовних ймовірностей для інших пацієнтів виконується аналогічно.

Таблиця 5.4.2.

Навчальні умовні ймовірності того, що ознаки x_j присутні/відсутні у образі Ω_λ

Клас Ω_λ	Ймовірність $p(x_j \Omega_\lambda)$ (наявності ознаки x_j у образі Ω_λ)	Ймовірність $p^*(x_j \Omega_\lambda)$ (відсутності ознаки x_j у образі Ω_λ)
Ω_1 $p_{\Omega_1}=1/3$	$p(x_1 \Omega_1) = 3/5=0,6$ $p(x_2 \Omega_1) = 2/5=0,4$ $p(x_3 \Omega_1) = 1/5=0,2$ $p(x_4 \Omega_1) = 5/5=1$	$p^*(x_1 \Omega_1) = 0,4$ $p^*(x_2 \Omega_1) = 0,6$ $p^*(x_3 \Omega_1) = 0,8$ $p^*(x_4 \Omega_1) = 0$
Ω_2 $p_{\Omega_2}=1/3$	$p(x_1 \Omega_2) = 3/5=0,6$ $p(x_2 \Omega_2) = 4/5=0,8$ $p(x_3 \Omega_2) = 2/5=0,4$ $p(x_4 \Omega_2) = 2/5=0,4$	$p^*(x_1 \Omega_2) = 0,4$ $p^*(x_2 \Omega_2) = 0,2$ $p^*(x_3 \Omega_2) = 0,6$ $p^*(x_4 \Omega_2) = 0,6$
Ω_3 $p_{\Omega_3}=1/3$	$p(x_1 \Omega_3) = 1/5=0,2$ $p(x_2 \Omega_3) = 2/5=0,4$ $p(x_3 \Omega_3) = 4/5=0,8$ $p(x_4 \Omega_3) = 4/5=0,8$	$p^*(x_1 \Omega_3) = 0,8$ $p^*(x_2 \Omega_3) = 0,6$ $p^*(x_3 \Omega_3) = 0,2$ $p^*(x_4 \Omega_3) = 0,2$

Тепер наша система РО навчена і готова до діагностики хвороб.

Припустимо, що у режимі класифікації новий хворий, якому треба поставити діагноз, має такі апостеріорні ознаки $X(x_1=-1; x_2=-1; x_3=1; x_4=1)$. Ці ознаки вектора $X(x_1, x_2, x_3, x_4)$ перевіряють на схожість з ознаками хвороб по класах $\Omega_1, \Omega_2, \Omega_3$ з використанням наступного правила для даних табл. 5.4.2: перші дві відсутні ознаки ($x_1=-1; x_2=-1$) означають, що в табл. 5.4.2 використовується колонка праворуч з ймовірностями $p^*(x_j | \Omega_\lambda)$, а наступні дві ознаки ($x_3=1; x_4=1$) означають, що в табл. 5.4.2 використовується колонка ліворуч з ймовірностями $p(x_j | \Omega_\lambda)$.

Оцінка апостеріорної ймовірності спостереження ознак x_j ($j=1,2,3,4$) в класі (хворобі, гіпотезі, образі) Ω_λ визначається по чисельнику класичної формули Байєсового класифікатора (5.4.14) $p_{\Omega_\lambda}^{pD} = p_{\Omega_\lambda} \prod_{j=1}^n p(x_j | \Omega_\lambda)$, звідки:

– для класу Ω_1 оцінка імовірності

$$p_{\Omega_1}^{pD} = p_{\Omega_1} \prod_{j=1}^n p(x_j | \Omega_1) = (1/3) * 0,4 * 0,6 * 0,2 * 1 = 0,016;$$

– для класу Ω_2 оцінка імовірності

$$p_{\Omega_2}^{pD} = p_{\Omega_2} \prod_{j=1}^n p(x_j | \Omega_2) = (1/3) * 0,4 * 0,2 * 0,4 * 0,4 = 0,00426;$$

– для класу Ω_3 оцінка імовірності

$$p_{\Omega_3}^{pD} = p_{\Omega_3} \prod_{j=1}^n p(x_j | \Omega_3) = (1/3) * 0,8 * 0,6 * 0,8 * 0,8 = 0,102.$$

У цьому випадку розраховані лише чисельники у формулі наївного класифікатора Байєса (5.4.14). Знаменник у формулі Байєса однаковий для всіх класів (він дорівнює підсумку розрахованих значень), і тому можна обмежитись лише розрахунком значень чисельників $p_{\Omega_1}^{pD}, p_{\Omega_2}^{pD}, p_{\Omega_3}^{pD}$.

Компаратор системи РО порівнює величини оцінок формули Байєса ($p_{\Omega_1}^{pD}, p_{\Omega_2}^{pD}, p_{\Omega_3}^{pD}$) і вкаже на хворобу Ω_3 , тому що ймовірність її появи більше за ймовірності появи інших хвороб ($p_{\Omega_2}^{pD} < p_{\Omega_1}^{pD} < p_{\Omega_3}^{pD}$).

Оцінка апостеріорної ймовірності спостереження ознак x_j ($j = 1, 2, 3, 4$) в класі (хворобі, гіпотезі, образі) Ω_λ згідно формули Байєсового класифікатора (5.4.16) дорівнює $p_{h\lambda}^{p\Sigma} = p_{h\lambda} \sum_{j=1}^n p(x_j | h_\lambda)$, звідки:

– для класу Ω_1 оцінка ймовірності

$$p_{\Omega_1}^{p\Sigma} = p_{\Omega_1} \sum_{j=1}^n p(x_j | \Omega_1) = (1/3) * 0,4 * 0,6 * 0,2 * 1 = 0,016;$$

– для класу Ω_2 оцінка ймовірності

$$p_{\Omega_2}^{p\Sigma} = p_{\Omega_2} \sum_{j=1}^n p(x_j | \Omega_2) = (1/3) * 0,4 * 0,2 * 0,4 * 0,4 = 0,00426;$$

– для класу Ω_3 оцінка ймовірності

$$p_{\Omega_3}^{p\Sigma} = p_{\Omega_3} \sum_{j=1}^n p(x_j | \Omega_3) = (1/3) * 0,8 * 0,6 * 0,8 * 0,8 = 0,102.$$

Компаратор системи РО порівнює величини оцінок формули Байєса ($p_{\Omega_1}^{p\Sigma}$, $p_{\Omega_2}^{p\Sigma}$, $p_{\Omega_3}^{p\Sigma}$) і вкаже на хворобу Ω_3 , тому що ймовірність її появи більше за ймовірності появи інших хвороб ($p_{\Omega_2}^{p\Sigma} < p_{\Omega_1}^{p\Sigma} < p_{\Omega_3}^{p\Sigma}$).

Відома послідовна процедура розпізнавання образу, яка полягає у тому, що у об'єкта з ознаками $X = \{x_1, x_2, \dots, x_j, \dots, x_n\}$ спочатку визначають одну ознаку x_1 . Якщо ймовірність розпізнавання образу по одній ознаці x_1 є достатньою, то образ відносять до визначеного класу, а якщо недостатньо, то розпізнають наступні ознаки. Метод використовується, якщо вимірювання ознак потребує великих витрат матеріальних чи часових ресурсів.

5.4.4. Використання формули Байєса для прогнозування дощу

Припустимо, що у нас є навчена експертна система передбачення дощу (Ω_1 – завтра буде дощ; Ω_2 – завтра не буде дощу) при апріорній ймовірності наявності дощу $p_{\Omega_1} = 0,2$ та відсутності дощу $p_{\Omega_2} = 0,8$ [92]. Для отримання прогнозу про погоду на завтра користувач вводить дані про погоду сьогодні: для кожного рядка табл. 5.4.3 користувач вводить одну з можливих відповідей «ТАК / НІ».

Таблиця 5.4.3.

Факти про погоду сьогоднішню погоду, які вводяться користувачем для отримання прогнозу на завтра

Ознаки погоди, які вводяться користувачем	Введені відповіді користувача	Ω_1 ($p_{\Omega_1} = 0,2$)		Ω_2 ($p_{\Omega_2} = 0,8$)	
		$p(x_j \Omega_1)$	$p^*(x_j \Omega_1)$	$p(x_j \Omega_2)$	$p^*(x_j \Omega_2)$
x_1 = Сьогодні є дощ?	ТАК; $x_1 = 1$	0,65	0,35	0,30	0,7
x_2 = Сьогодні холодно?	НІ; $x_2 = -1$	0,6	0,4	0,35	0,65
x_3 = Сьогодні туман?	ТАК; $x_3 = 1$	0,3	0,7	0,65	0,35
x_4 = Сьогодні сиро?	ТАК; $x_4 = 1$	0,55	0,45	0,40	0,6
x_5 = Сьогодні сонячно?	НІ; $x_5 = -1$	0,2	0,8	0,75	0,25
x_6 = Сьогодні сухо?	НІ; $x_6 = -1$	0,1	0,9	0,8	0,2

Виконуємо для кожного класу окремі розрахунки у вигляді чисельників класичної формули Байєсового класифікатора (5.4.14), використовуючи введені користувачем дані згідно табл. 5.4.3:

– для класу Ω_1 (завтра буде дощ)

$$p_{\Omega_1}^{pD} = p_{\Omega_1} \prod_{j=1}^6 p(x_j | \Omega_1) = 0,2 \cdot 0,65 \cdot 0,4 \cdot 0,3 \cdot 0,55 \cdot 0,8 \cdot 0,9 = 0,00617; \quad (5.4.17)$$

– для класу Ω_2 (завтра не буде дощу)

$$p_{\Omega_2}^{pD} = p_{\Omega_2} \prod_{j=1}^6 p(x_j | \Omega_2) = 0,8 \cdot 0,3 \cdot 0,65 \cdot 0,65 \cdot 0,4 \cdot 0,25 \cdot 0,2 = 0,00203. \quad (5.4.18)$$

Таким чином, за класичною формулою Байєсового класифікатора (5.4.14) завтра буде дощ, тому що $p_{\Omega_1}^{pD} > p_{\Omega_2}^{pD}$. На це рішення вплинули малі значення двох множників $(0,25 \cdot 0,2)$ в формулі (5.4.18).

За формулою Байєсового класифікатора (5.4.16) визначаємо, що завтра дощу не буде, бо $p_{\Omega_2}^{p\Sigma} > p_{\Omega_1}^{p\Sigma}$, де

$$p_{\Omega_1}^{p\Sigma} = p_{\Omega_1} \sum_{j=1}^n p(x_j | \Omega_1) = 0,2 \cdot (0,6+0,4+0,3+0,55+0,8+0,9) = 0,71; \quad (5.4.19)$$

$$p_{\Omega_2}^{p\Sigma} = p_{\Omega_2} \sum_{j=1}^n p(x_j | \Omega_2) = 0,8 \cdot (0,3+0,65+0,65+0,4+0,25+0,2) = 1,96. \quad (5.4.20)$$

З розрахунків за формулами (5.4.19) і (5.4.20) випливає, що підсумки ймовірностей ознак [оцінки математичного очікування ознак $[(0,6+0,4+0,3+0,55+0,8+0,9) = 3,55$ та $(0,3+0,65+0,65+0,4+0,25+0,2) = 2,45]$ надають перевагу гіпотезі Ω_1 (завтра буде дощ), але апріорні ймовірності $p_{\Omega_1} = 0,2$ та $p_{\Omega_2} = 0,8$ змінюють це рішення на протилежне.

5.4.5. Зв'язок відносних стохастичних та детермінованих потоків

Звичайно стохастичні й детерміновані задачі розглядають окремо, і їх не зв'язують між собою. Не існує також поняття «повної групи ймовірнісних і детермінованих подій».

Але у житті ймовірнісні і детерміновані події часто існують поряд, зв'язані між собою і взаємно перемішані. Життя не складається з «чисто стохастичних» або «чисто детермінованих» подій. Існують випадки, коли їх належить аналізувати разом, бо саме їх сукупність складає повну групу і дає більш точні відповіді на поставлені питання.

Для доведення зв'язку ймовірнісних і детермінованих подій, розглянемо універсум, рівний по об'єму умовній 1, якій відображує відносний час, що може витратиться людиною на виконання сукупності ймовірнісних і детермінованих подій. З універсуму витікає стохастичний та детермінований відносні потоки подій, які у сукупності складають повну групу.

Зв'язок відносних стохастичних та детермінованих потоків розглянемо на прикладі покупки книги робітником та пенсіонером [80]. В магазині 2 книжка буває з ймовірністю $p_{2,4}=0,6$, а в магазині 3 – з ймовірністю $p_{3,4}=0,8$; відповідні ймовірності відсутності книжок $p_{2,6}=0,4$, $p_{3,6}=0,2$. Ро-

бітник і пенсіонер рівноймовірно і кожний окремо відвідують магазини 2 та 3 з наміром купити книгу (рис. 5.4.1, рис. 5.4.2).

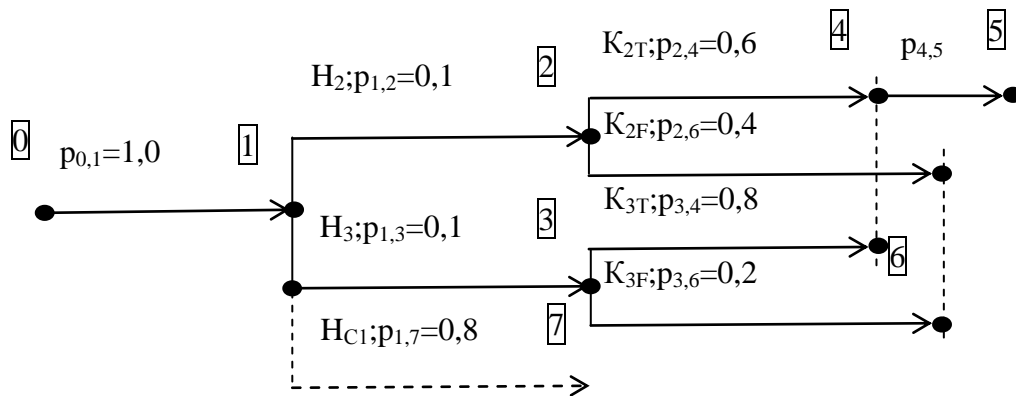


Рис. 5.4.1. Схема заміщення мережі для робітника з відносними потоками купівлі книжки

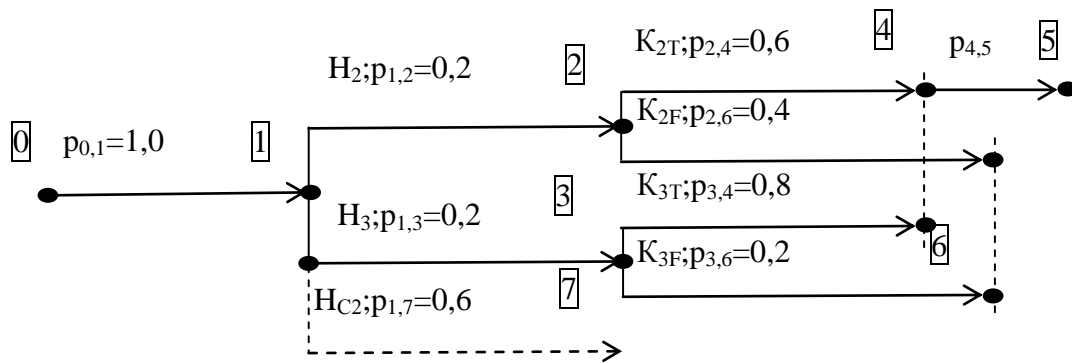


Рис. 5.4.2. Схема заміщення мережі для пенсіонера з відносними потоками по купівлі книжки

На рис. 5.4.1 для робітника помічені потоки-гіпотези:

- H_2, H_3 – відвідини магазинів 2 та 3 (магазини мають вигляд вузлових точок 2 та 3);
- K_{2T}, K_{3T} – купівля книги в магазинах 2 та 3;
- K_{2F}, K_{3F} – некупівля книги в магазинах 2 та 3;
- H_{C1} – детермінований час, який вилучається у робітника на роботу, сон та інші справи (розглядається як окрема «гіпотеза», яка у дійсності є детермінованою подією).

Потрібно визначити: «З якою ймовірністю робітник та пенсіонер куплять книжку?».

Для робітника з рис. 5.4.1 впливає:

- в гілці $\boxed{0} - \boxed{1}$ повний потік ймовірнісних подій у вигляді відносного часу $p_{0,1}=1,0$ витікає з вузла $\boxed{0}$ та втікає у вузол $\boxed{1}$ (тут потік подій $p_{0,1}=1,0$ одночасно розглядається: як потік стохастичних подій, як ймовірність спостереження подій, як відносний час спостереження подій);

– у вузлі $\boxed{1}$ батьківській потік подій $p_{0,1}=1,0$ (якщо потік подій $p_{0,1}=1,0$ цілком відбувся) розгалужується на три нащадкові потоки, які у сукупності складають «повну локальну групу»:

- $p_{1,2}=0,1$ – потік ймовірнісних подій у вигляді відносного часу відвідин магазину – вузла 2;
- $p_{1,3}=0,1$ – потік ймовірнісних подій у вигляді відносного часу відвідин магазину – вузла 3;
- $p_{1,7}=0,8$ – потік детермінованих подій у вигляді відносного часу, виділеного на роботу, сон та інші справи (природно, що у робітника порівняно з пенсіонером цей відносний час більший);
 - у вузлі-магазині 2 батьківській потік $p_{1,2}=0,1$ (якщо потік подій $p_{1,2}=0,1$ цілком відбувся) розділяється на два нащадкові потоки, які у сукупності складають «повну локальну групу»:
 - $p_{2,4}=0,6$ – локальний потік ймовірнісних подій в гілці 2–4 у вигляді відносного часу наявності (купівлі) книжки в магазині – вузлі 2;
 - $p_{2,6}=0,4$ – потік ймовірнісних подій в гілці 2–6 у вигляді відносного часу відсутності (не купівлі) книжки в магазині – вузлі 2.

Нащадкові потоки схеми рис. 5.4.1 можна розглядати як локальні потоки метода ієрархічних потоків [53], а рис. 5.4.1 можна розглядати як схему заміщення для аналізу умовних ймовірностей.

В результаті ймовірність купівлі книжки робітником дорівнює
 $p_{4,5} = p_{2,4}p_{1,2} + p_{3,4}p_{1,3} = 0,6 \cdot 0,1 + 0,8 \cdot 0,1 = 0,14$.

Для пенсіонера з рис. 5.4.2 отримуємо наступні дані:

- в гілці 0–1 повний потік ймовірнісних подій у вигляді відносного часу $p_{0,1}=1,0$ витікає з вузла 0 та втікає у вузол 1;
- у вузлі 1 батьківській потік $p_{0,1}=1,0$ (якщо потік подій $p_{0,1}=1,0$ цілком відбувся) розділяється на три нащадкові потоки, які у сукупності складають «повну локальну групу»:
- $p_{1,2}=0,2$ – потік ймовірнісних подій у вигляді відносного часу відвідин магазину – вузла 2;
- $p_{1,3}=0,2$ – потік ймовірнісних подій у вигляді відносного часу відвідин магазину – вузла 3;
- $p_{1,7}=0,6$ – потік детермінованих подій у вигляді відносного часу на побутову роботу, сон та інші справи (у пенсіонера порівняно робітником з цей відносний час менший);
 - у вузлі-магазині 2 батьківській потік $p_{1,2}=0,2$ (якщо потік подій $p_{1,2}=0,2$ цілком відбувся) розділяється на два нащадкові потоки, які у сукупності складають «повну локальну групу»:
 - $p_{2,4}=0,6$ – локальний потік ймовірнісних подій в гілці 2–4 у вигляді відносного часу наявності (купівлі) книжки в магазині – вузлі 2;
 - $p_{2,6}=0,4$ – потік ймовірнісних подій в гілці 2–6 у вигляді відносного часу відсутності (не купівлі) книжки в магазині – вузлі 2.

В результаті ймовірність купівлі книжки пенсіонером вдвічі більша порівняно з робітником і дорівнює
 $p_{4,5} = p_{2,4}p_{1,2} + p_{3,4}p_{1,3} = 0,6 \cdot 0,2 + 0,8 \cdot 0,2 = 0,28$.

Але якщо не враховувати *детермінований час*, який відбирається у покупців за «гіпотезою» H_{C1} та H_{C2} на роботу, сон та інші справи, то гіпотези відвідин магазинів (H_2, H_3) у них мають однакові потоки ймовірностей ($p_{1,2}=0,5$; $p_{1,3}=0,5$). Тому ймовірність купівлі книжки кожним з покупців є однаковою і дорівнює $p_{4,5} = p_{2,4}p_{1,2} + p_{3,4}p_{1,3} = 0,6 \cdot 0,5 + 0,8 \cdot 0,5 = 0,7$.

Таким чином, якщо не враховувати, що ймовірнісні та відносні детерміновані потоки подій одночасно сумісно складають *«повну групу відносних потоків»*, то можна помилитись у відповіді на питання «Хто з покупців раніше придбає книгу?».

Разом з тим виникає ще одне питання: за який час спостерігаються ці отримані нами ймовірності? Тобто потрібно у статистичних розрахунках (щоб вони відбулись) «набрати вибірку» відвідин магазинів, а це не виконується миттєво. В результаті за експериментальними даними можна визначити для робітника і пенсіонера поступове збільшення у часі значення ймовірностей купівлі книги.

Цей приклад демонструє практичну значущість урахування взаємної залежності потоків відносних стохастичних і детермінованих подій та необхідність контролю у часі зміни статистичних і детермінованих показників. З розглянутого аналізу випливає:

– у загальному випадку *підсумок «повної групи ймовірнісних подій» може дорівнювати не «одиниці», а деякому фіксованому числу, меншому за 1*;

– потрібно враховувати час, за який набираються статистичні дані з моменту початку розв'язку задачі;

– потоки в мережах слід розглядати як потоки *умовних ймовірностей*, бо для рис. 5.4.1 локальний потік $p_{2,4}=0,6$ аналізується, *якщо* відбувся батьківський локальний потік $p_{1,2}=0,1$ і *якщо* відбувся батьківський локальний потік $p_{0,1}=1,0$. В результаті потік ймовірності $P_{2,4}$ в гілці 2 – 4 дорівнює добутку локальних потоків $P_{2,4} = p_{0,1} p_{1,2} p_{2,4} = 1,0 \cdot 0,1 \cdot 0,6 = 0,06$.

5.4.6. Електрична схема заміщення Байєсової мережі з потоками умовних ймовірностей

Схеми заміщення мереж рис. 5.4.1 та рис. 5.4.2 можна розглядати як «електричні схеми» для вхідного «джерела струму (потіку)» $p_{0,1}=1,0 = \text{const}$ з розподілом цього потоку по гілкам мережі у залежності від їх «електричних параметрів» (активного опору R , індуктивності L , ємності C , ЕРС E), значення яких можна розрахувати.

Це означає, що на аналіз схем заміщення рис. 5.4.1 та рис. 5.4.2 можна розповсюдити добре розвинені методи аналізу електричних схем на основі фізичних законів Кірхгофа [6] з урахуванням того, що стосовно ймовірнісних мереж вказані «струми» дорівнюють відповідним ймовірностям (потікам стохастичних подій у відносних одиницях).

Байєсова мережа – це орієнтований ациклічний граф, у якого вершини – це змінні, а ребра – ієрархічні зв’язки між вершинами [7; 11]. При цьому звичайно залежність між змінними – вершинами відображується таблицями (рис. 5.4.3 [7]). В результаті комбінаторний характер залежності між змінними зменшує наочність розв’язку задачі і не дозволяє розглядати стохастичні процеси в динаміці.

При розгляді Байєсових ймовірнісних мереж ймовірність початкової вибірки будемо розглядати як рівний «1» ймовірнісний потік подій, який, без зміни своєї підсумкової величини, розгалужується по ієрархічно нижчим вершинам.

Припустимо, що відома Байєсова мережа (рис. 5.4.3, [7]) призначена для визначення ймовірності спостереження мокрої Трави (G) у залежності від ймовірності Дощу (R) та ймовірності увімкнення Розбризувача (S).

В таблицях рис. 5.4.3 використані відповідні позначення з нижніми індексами «Т (TRUE)»: «R_T – Дощ пройшов» «S_T – Розбризувач увімкнений», «G_T – Трава мокра». Нижні індекси «F (FALSE)» означають протилежні події: «R_F – Дощ не пройшов» і т. д. Мережу рис. 5.4.3 представимо у вигляді електричної схеми рис. 5.4.4 [80].

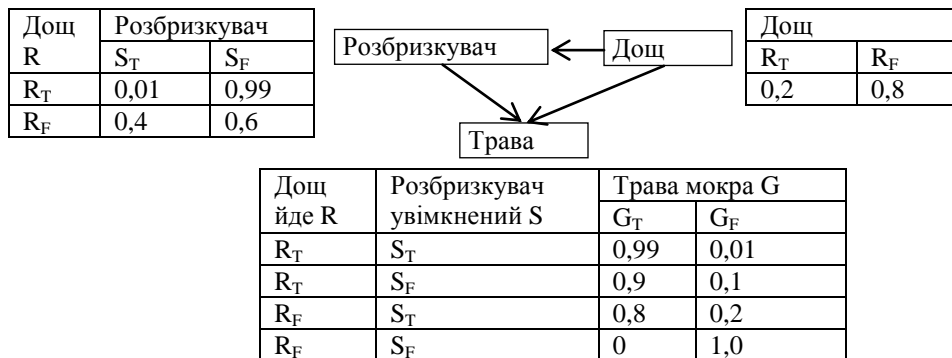


Рис. 5.4.3. Байєсова мережа з вершинами – змінними [7]

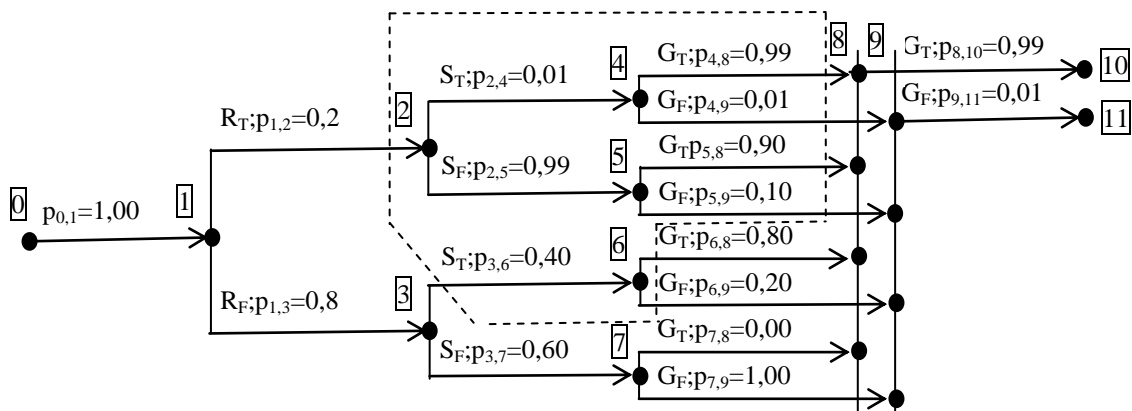


Рис. 5.4.4. Електрична схема заміщення з потоками умовних ймовірностей рис. 5.4.3

Електрична схема заміщення з потоками умовних ймовірностей рис. 5.4.4 відображує наступні особливості Байєсової мережі рис. 5.4.3:

– $p_{0,1}=1$ – вхідний батьківській повний потік подій стохастичної вибірки розгалужується у вузлі [1] на два вихідні нащадкові потоки вузла, які у сукупності складають «повну локальну групу»:

- $p_{1,2}=0,2$ – локальний потік ймовірнісних подій у вигляді відносного часу спостереження наявності Дощу – вузла [2], якщо всі події потоку $p_{0,1}=1$ відбулись;

- $p_{1,3}=0,8$ – локальний потік ймовірнісних подій у вигляді відносного часу спостереження відсутності Дощу – вузла [3], якщо всі події потоку $p_{0,1}=1$ відбулись.

Щоб не повторювати по суті однаковий опис розгалуження потоків $p_{1,2}=0,2$ та $p_{1,3}=0,8$, нижче розглянемо лише гілки, по яких протікає потік $p_{1,2}=0,2$.

– Вузол [2]: батьківській потік ймовірності наявності Дощу $p_{1,2}=0,2$ розгалужується на два нащадкових потоки, які у сукупності складають «повну локальну групу»:

- $p_{2,4}=0,01$ – нащадковий потік ймовірнісних подій у вигляді відносного часу спостереження увімкнення Розбризувача S при наявності Дощу;

- $p_{2,5}=0,99$ – нащадковий потік ймовірнісних подій у вигляді відносного часу спостереження вимкненого Розбризувача S при наявності Дощу.

– Вузол [4]: батьківській потік $p_{2,4}=0,01$ ймовірності увімкнення Розбризувача S_T при наявності Дощу R_T розгалужується на два нащадкових потоки, які у сукупності складають «повну локальну групу»:

- $p_{4,8}=0,99$ – нащадковий потік ймовірнісних подій у вигляді відносного часу спостереження мокрої Трави G_T при увімкненні Розбризувача S_T та при наявності Дощу R_T ;

- $p_{4,9}=0,01$ – нащадковий потік ймовірнісних подій у вигляді відносного часу спостереження не мокрої Трави G_F при увімкненні Розбризувача S_T та при наявності Дощу R_T .

Якщо аналогічним чином розглянути розгалуження всіх вузлів схеми заміщення рис. 5.4.4, то вузол [8] збере всі потоки ймовірностей спостереження мокрої Трави G_T , а вузол [9] – всі потоки ймовірностей спостереження не мокрої Трави G_F . Підсумок цих двох потоків ймовірностей складають повну групу.

Ймовірнісна мережа (рис. 5.4.4) задовольняє всім вимогам аналізу мереж за методом ієрархічних потоків [53] і має наступні особливості:

– локальні нащадкові потоки умовних ймовірностей $p_{i,j}$, на які розгалужується вхідний батьківський потік вузла, за методом ієрархічних потоків складають повну групу ймовірностей (вони вказують, яка частка будь-якого вхідного батьківського потоку належить вихідному потоку j-го нащадка);

– батьківські потоки ймовірностей при необхідності теж можна перевести у локальні значення стосовно попередніх вузлів; кожний вхідний батьківській потік одного вузла розгалужується між нащадковими потоками пропорційно локальним потокам їх умовних ймовірностей;

– рівність «1» основного вхідного батьківського потоку та підсумку всіх потоків поперечного січення в електричній схемі заміщення означає, що «електрична» схема живиться від «джерела струму» $p_{0,1} = 1 = \text{const}$, а кожен гілку можна розглядати як електричну гілку з RLCE – параметрами з можливістю аналізу схеми згідно теоретичним основам електротехніки [6]. Струми схеми заміщення (потоки електронів) моделюють відповідні потоки стохастичних подій.

На рис. 5.4.4 помічено:

– $\boxed{1}, \boxed{2}, \dots, \boxed{11}$ – вузли електричної схеми заміщення (показані в прямокутниках). У вузол $\boxed{1}$ втікає загальний батьківський потік $p_{0,1} = 1 = \text{const}$. У вузол $\boxed{10}$ входить потік спостереження мокрої Трави $p_{8,10} = 0,99$, а у вузол $\boxed{11}$ входить потік спостереження не мокрої Трави $p_{8,10} = 0,01$, які складають повну групу $p_{8,10} + p_{9,11} = 1$;

– $p_{0,1}, p_{1,2}, \dots, p_{9,11}$ – потоки умовних ймовірностей між відповідними вузлами схеми, де нижні індекси у позначенні *потіку умовної ймовірності* $p_{\alpha,\beta}$ означають:

- «потік умовної ймовірності витікає з вузла $\boxed{\alpha}$ та впадає у вузол $\boxed{\beta}$ »;
 - «потоки вузла $\boxed{\beta}$ реалізуються, якщо реалізовані потоки вузла $\boxed{\alpha}$ »;
- R_T, S_T, G_T – одиничні функції відповідних гілок електричної схеми, які можуть мати два значення (1 або 0): 1 – апостеріорно спостережено, що Дощ R йде, Розбризкувач S увімкнений і Трава G мокра; 0 – у протилежних випадках;

– R_F, S_F, G_F – одиничні функції відповідних гілок електричної схеми, які можуть мати два значення (1 або 0): 1 – апостеріорно спостережено, що Дощ R не йде, Розбризкувач S не увімкнений і Трава G не мокра; 0 – у протилежних випадках.

Апріорні значення всіх одиничних функцій ($R_T, R_F; S_T, S_F; G_T, G_F$) вважаємо рівними нулю; при цьому спостерігаються потоки апріорних ймовірностей навченої мережі. Подібні значення одиничних функцій прийняті з урахуванням того, що одна й та ж схема заміщення повинна відображувати три стани: апріорні дані; апостеріорні дані при спостереженні наявності чи відсутності деякої події.

При наявності апостеріорного спостереження одна з одиничних функцій в парах (R_T, R_F), (S_T, S_F) та (G_T, G_F) може приймати значення 1.

Вхідний батьківський потік ймовірності $p_{0,1} = n_0/n_0 = 1$ із загальною кількістю подій вибірки n_0 на 1-му ієрархічному рівні у вузлі $\boxed{1}$ розділяється на частки ($n_0 = n_{1,2} + n_{1,3}$) між показаними у прямокутних рамках вузлом $\boxed{1}$ та вузлами $\boxed{2}$ і $\boxed{3}$ 2-го ієрархічного рівня, для яких відповідні локальні потоки ймовірності складають повну групу ($p_{1,2} + p_{1,3} = n_{1,2}/n_0 + n_{1,3}/n_0 = 1$). Дали

отримані локальні потоки $p_{1,2} + p_{1,3} = 1$ знову розділяються на *локальні потоки умовних ймовірностей нижчих ієрархічних рівнів* $p_{\alpha,\beta}$, де α – порядковий номер *батьківської вершини*, звідки витікає потік $p_{\alpha,\beta}$; β – порядковий номери *нащадкової вершини*, куди втікає потік $p_{\alpha,\beta}$. Потік ймовірності $p_{\alpha,\beta}$ кожної гілки схеми заміщення характеризується відповідною часткою вхідного батьківського потоку стохастичної вибірки p_0 , але загальний потік перерізу мережі потоків у будь-якому місці не змінюється і дорівнює p_0 .

На рис. 5.4.4 пунктирною лінією помічений відомий в електротехніці *складний вузол*, який охоплює кілька вузлів схеми і стосовно якого також можна визначити вхідні батьківські та вихідні нащадкові локальні потоки.

Для вузлів електричної схеми заміщення рис. 5.4.4 при використанні одиничних функцій ($R_T, R_F, S_T, S_F, G_T, G_F$) можна скласти математичну модель (рис. 5.4.5), рішення для якої отримане в MathCAD [80]:

$RT:=0$	$RF:=0$	$ST:=1$	$SF:=0$	$GT:=0$	$GF:=0$
$p12:=(1- RF)(0.2 + RT\cdot 0.8)$		$p13:=(1- RT)(0.8 + RF\cdot 0.2)$			
$p24:=(1- SF)(0.01 + ST\cdot 0.99)$	$p12$	$p25:=(1- ST)$			
$(0.99 + SF\cdot 0.01)p12$					
$p36:=(1- SF)(0.4 + ST\cdot 0.6)\cdot p13$		$p37:=(1- ST)(0.6 + SF\cdot 0.4)\cdot p13$			
$p48:=(1- GF)(0.99 + GT\cdot 0.01)\cdot p24$		$p49:=(1- GT)$			
$(0.01 + GF\cdot 0.99)\cdot p24$					
$p58:=(1- GF)(0.9 + GT\cdot 0.1)\cdot p25$		$p59:=(1- GT)(0.1 + GF\cdot 0.9)\cdot p25$			
$p68:=(1- GF)(0.8 + GT\cdot 0.2)\cdot p36$		$p69:=(1- GT)(0.2 + GF\cdot 0.8)\cdot p36$			
$p78:=(1- GF)(0 + GT\cdot 1)\cdot p37$		$p79:=(1- GT)(1 + GF\cdot 0)\cdot p37$			
$p810:= p48+p58+p68+p78$		$p911:= p49+p59+p69+p79$			
$p12 = 0.2$	$p24 = 0.2$	$p25 = 0$	$p48 = 0.198$	$p49 = 0.002$	$p58 = 0$
$p59 = 0$					
$p13 = 0.8$	$p36 = 0.8$	$p37 = 0$	$p68 = 0.64$	$p69 = 0.16$	$p78 = 0$
$p79 = 0$					
$p810 = 0.838$		$p911 = 0.162$			

Рис. 5.4.5. Математична модель MathCAD для аналізу схеми рис. 5.4.4:

$p810 = 0.838$ – потік ймовірності, що Трава мокра;

$p911 = 0.162$ – потік ймовірності, що Трава суха

Модель рис. 5.4.5 розраховує *не локальні потоки умовних ймовірностей, а потоки ймовірностей інформації*, які дорівнюють добутку локальної ймовірності даної гілки на локальні ймовірності всіх ієрархічно вищих гілок. Тому позначення цих величин ймовірностей змінені на потоки без використання нижніх індексів.

Для моделі рис. 5.4.5 використовуються значення одиничних функцій ($ST:=1$; $SF:=0$), внаслідок чого модель відображує відповідь на питання: «Якими будуть потоки ймовірностей у випадку, якщо апостеріорно помічено, що Розбризкувач увімкнений (тобто $ST:=1$; $SF:=0$)?». В результаті отримана відповідь: $p12 = 0.2$; $p24 = 0.2$; $p25 = 0$; $p48 = 0.198$; $p49 = 0.002$; $p58 = 0$; $p59 = 0$; $p13 = 0.8$; $p36 = 0.8$; $p37 = 0$; $p68 = 0.64$; $p69 = 0.16$; $p78 = 0$;

$p_{79} = 0$; $p_{810} = 0.838$; $p_{911} = 0.162$. Як бачимо, у даному прикладі формула Байеса не застосовується для отримання відповіді.

Якщо всі одиничні функції прирівняти нулю ($RT:=0$, $RF:=0$, $ST:=0$, $SF:=0$, $GT:=0$, $GF:=0$), то потоки моделі рис. 5.4.5 відображують стан навченої мережі: $p_{12} = 0.2$; $p_{24} = 0.002$; $p_{25} = 0.198$; $p_{48} = 0.00198$; $p_{49} = 0.00002$; $p_{58} = 0.1$; $p_{59} = 0.02$; $p_{13} = 0.8$; $p_{36} = 0.32$; $p_{37} = 0.48$; $p_{68} = 0.256$; $p_{69} = 0.064$; $p_{78} = 0$; $p_{79} = 0.48$; $p_{810} = 0.436$; $p_{911} = 0.564$.

Якщо ввести дані апостеріорного спостереження «йде Дощ ($RT:=1$, $RF:=0$)»; «увімкнений Розбризувач ($ST:=1$, $SF:=0$)»; «чи мокра Трава – не спостережено ($GT:=0$, $GF:=0$)», то модель видає значення потоків ймовірностей: $p_{12} = 1$; $p_{24} = 1$; $p_{25} = 0$; $p_{48} = 0.99$; $p_{49} = 0.01$; $p_{58} = 0$; $p_{59} = 0$; $p_{13} = 0.8$; $p_{36} = 0$; $p_{37} = 0$; $p_{68} = 0$; $p_{69} = 0$; $p_{78} = 0$; $p_{79} = 0$; $p_{810} = 0.99$; $p_{911} = 0.01$.

З отриманих рівнянь можна бачити, що ймовірність потоку гілки дорівнює добуткам локальних ймовірностей даного конкретного потоку та потоків локальних ймовірностей всіх вищих ієрархічних рівнів при наявності лише одного батьківського потоку у всіх вершин. У випадку наявності кількох батьківських потоків, потрібно ураховувати відповідні еквівалентні параметри гілок з потоками ймовірностей.

5.4.7. Теоретична основа мереж стохастичних та детермінованих потоків

До мереж з відносними потоками можна застосувати перефразовані закони Кірхгофа, які використовуються для аналізу роботи вентильних перетворювачів [47; 50]:

1. «Для з'єднаних у вершині простих та складних дуг алгебраїчна сума добутків потоків дуг p_{ij} (чи похідних від потоків p_{ij} по часу) на їх перемикаючі функції дорівнює нулю (ці рівняння складаються для $(n-1)$ вершин)»

$$\sum f_{ij} p_{ij} = 0 \quad , \quad \sum f_{ij} dp_{ij} / dt = 0 \quad , \quad (5.4.15)$$

де t – час; f_{ij} – перемикаюча функція, яка має значення 1, якщо дуга увімкнена, та 0, якщо вимкнена [47]; $i = 1, 2, \dots, m$ – порядковий номер вершини; $j = 1, 2, \dots, n$ – порядкові номери сусідніх вершин при $i \neq j$; n – кількість вершин; p_{ij} – потік дуги між вершинами i та j .

2. «Алгебраїчна сума падінь тисків потоків вздовж замкнутого простого чи складного контуру, помножена на перемикаючу функцію цього контуру, дорівнює нулю (згідно цьому закону складаються $[m-(n-1)]$ рівнянь)»

$$f_K \sum u_{ij} = 0 \quad , \quad (5.4.16)$$

де f_K – перемикаюча функція контуру, яка приймає значення 1, якщо контур замкнений та 0, якщо розімкнений; u_{ij} – падіння тиску потоку на гілці між вершинами i та j .

У формулах «повної ймовірності», «повної групи ймовірностей», у «формулі Байєса» частка гіпотез може визначатись детермінованими потоками (лапки у назвах перелічених формул використані для підкреслення їх відмінності від оригіналу). Тому потоки одного вузла можна розглядати сумісно, без розділу їх на стохастичну і детерміновану складові.

Наприклад, вважаємо, що електрична схема заміщення рис. 5.4.4 має вхідний електричний струм (потік ймовірнісних подій) $p_{0,1}=1 = \text{const}$ від «джерела постійного струму». По відомим даним розподілу апріорних струмів в гілках мережі, наведених на рис. 5.4.4, визначаємо відповідні значення електричних активних опорів $R_{\alpha,\beta}$ гілок, увімкнених між вузлами α та β [50; 144; 145]. Якщо ми спостерігаємо, що Дощу немає, то за допомогою перемикаючих функцій f_{ij} та f_K в формулах (5.4.15) та (5.4.16) значення відповідного батьківського струму $p_{1,2} = 0$; внаслідок цього одночасно також стають рівними нулю всі залежні від величини $p_{1,2}$ нащадкові потоки $p_{2,4} = p_{2,5} = p_{4,8} = p_{4,9} = p_{5,8} = p_{5,9} = 0$. Тому змінюється величина струму $p_{1,3} = p_{0,1} = 1$, який далі розподіляється по гілках мережі згідно активних опорів, розрахованих за апріорними даними. В результаті розглянутий режим схеми заміщення Байєсової мережі рис. 5.4.4 аналізує мережу з детермінованими струмами (Дощу немає) та ймовірнісними струмами у залежності від апріорних (початкових) спостережень включення Розбризкувача та спостережень мокрої і не мокрої Трави.

Визначення параметрів схеми та її аналіз може відбуватись в статичній та динаміці на основі методів курсу «Теоретичних основ електротехніки» [6] та методів [47; 50; 52; 144; 145]. При аналізі динаміки можна використати рівняння *розривних контурів* півпровідникових ключових приладів з введенням в схему електричних опорів [47].

Аналіз мереж з відносними ймовірнісними та детермінованими потоками має наступні особливості:

1. Мережі можуть розглядатись з урахуванням сумісності їх стохастичних та детермінованих складових, коли частка «гіпотез» визначається детермінованими потоками. По аналогії з «Теоретичними основами електротехніки» фізичні закони Кірхгофа та еквівалентні схеми зі змінною структурою є основою для аналізу статичних, динамічних, детермінованих і стохастичних потоків з урахуванням дій системи керування. Потік гілки мережі може мати стохастичну і детерміновану складові.

2. *Об'єднаний аналіз стохастичних і детермінованих процесів* може бути використаний у будь-якій сфері, включаючи економічну.

3. Мережа з потоками може бути представлена у вигляді еквівалентної електричної схеми заміщення зі змінною структурою та з джерелами пото-

ків. Спостереження деяких стохастичних потоків у вузлі перетворює їх на детерміновані потоки.

4. Формула Байєса в мережі замінюється на відношення між потоками. Локальні потоки одного вузла за «формулою Байєса», стохастичні формули «повної ймовірності» та «повної групи ймовірностей» у принципі можуть урахувувати поряд з стохастичними потоками також й детерміновані потоки, що спрощує деякі висновки. Саме Байєс зв'язав єдиним аналізом стохастичні і апостеріорно визначені детерміновані процеси. Якщо прийняти потік універсуму рівним «1», то відповідний підсумок потоків ймовірностей може бути меншим за «1».

5. Формули Байєса дають значення батьківських чи нащадкових *локальних потоків будь-якого одного вузла*, включаючи й *складний вузол*, створений січенням, що охоплює кілька вузлів.

5.5. Статистичні способи обробки експериментальних даних

5.5.1. Методи первинної статистичної обробки результатів експерименту

Статистично осереднені групи з «образів» товарів та «покупців» займають важливе місце в аналізі образів і часто використовуються при маркетинговому аналізі. Тому нижче наводяться основні відомості стосовно визначення статистичних ознак.

Слово «статистика» походить від *латинського «status» – стан*. Звідси утворилися слова: «stato» – держава; «statista» – знавець держави; «statistica» – *сума знань про державу*. Стохастична статистика виникла *відгрецького «стохастика» – припускати*. Основні складові частини статистики: теорія статистики; економічна статистика; галузеві статистики; соціальна статистика [45; 93; 95; 105; 110; 111; 112].

Життя майже завжди надає факти, приклади, *які або підтверджують, або спростовують одне і теж твердження*. Перехід від окремих фактів до масових виявляє загальну закономірність, позбавлену впливу випадкових причин. Статистика досліджує *не поодинокі, а масові явища*.

Виявлені закономірності проявляють себе *лише у великій масі явищ*. Таку масу називають *статистичною сукупністю*, яка є об'єктом статистичного дослідження і складається з множини існуючих елементів, поєднаних спільними властивостями, умовами, причинами існування і розвитку. Обсяг статистичної сукупності у малій вибірці може складати 5–10 одиниць.

Будь-яке статистичне дослідження послідовно проходить 3 етапи:

1. Визначають мету спостереження і збирають первинний статистичний матеріал.

2. *Обчислюють елементарні математичні статистики* (абсолютні, відносні чи середні величини, групування).

3. *Аналізують отримані дані* для одержання висновків про стан явищ, що вивчаються, та закономірностей їх розвитку (методи вивчення взаємозв'язків, варіації, тенденцій розвитку, прогнозування, диференціації та сталості тощо).

5.5.2. Початкове обчислення елементарних математичних статистик

Ряд розподілу – це групування одиниць сукупності за однією певною ознакою, яке характеризується двома величинами – варіантом і частотою:

- *варіант* – окреме значення групувальної ознаки;

- *частота* – число, що показує як часто варіант зустрічається в ряді.

Замість частот іноді вживають відносну частоту, що виражена в долях одиниці.

Перша група методів статистичної обробки охоплює обчислення елементарних математичних статистик:

1. *Вибіркового середнього* або середнього арифметичного значення по вибірці

$$x_{CP} = \frac{1}{n} \sum_{j=1}^n x_j,$$

де n – кількість випробуваних у вибірці показників; x_j – приватні значення показників у j -м досліді.

2. *Моди* – так зветься кількісне значення досліджуваної ознаки, яке найчастіше зустрічається у вибірці. Мода має практичне значення у соціологічних, маркетингових та інших дослідженнях. *Наявність декількох мод* свідчить про неоднорідність сукупності – про об'єднання в одній сукупності різноякісних одиниць.

3. *Медіани* – так зветься значення ознаки, яке ділить навпіл вибірку, впорядковану за величиною даної ознаки. Середня і медіана для нормального розподілу зазвичай збігаються. Сума відхилень варіант від медіани є *найменшою величиною*, що використовується, наприклад, у маркетингу, будівництві – при плануванні розміщення торговельного закладу, зупинок міського транспорту тощо.

4. *Розкид у вибірці* (іноді цю величину звать *розмахом*)

$$R = x_{\max} - x_{\min}.$$

5. *Дисперсії* – це середнє арифметичне квадратів відхилень значень змінної від її середнього значення x_{CP}

$$D = \frac{1}{n} \sum_{j=1}^n (x_j - x_{CP})^2.$$

В умовах нормального розподілу, якщо обсяг сукупності досить великий, встановлена залежність між стандартним відхиленням та кількістю

спостережень:

- в межах $x_{\text{сер}} \pm 1\sigma$ розташовано 68,3 % кількості спостережень (членів ряду);
- в межах $x_{\text{сер}} \pm 2\sigma$ – 95,4 % спостережень;
- в межах $x_{\text{сер}} \pm 3\sigma$ – 99,7 % спостережень. На практиці майже не зустрічаються відхилення, що перевищують $\pm 3\sigma$. Відхилення 3σ вважається максимально можливим, і його називають «правилом 3-х сигм».

Групування поділяються на три види:

1. *Типологічне* – розділ населення на однорідні групи за соціальним складом, формою власності, рівнем освіти тощо.

2. *Структурне* – розділ населення на однорідні групи з урахуванням відношень між окремими частинами цілого. Наприклад, для підприємства – розділ по цехам; для ряду підприємств – розділ за кількістю працюючих, кількістю продукції тощо.

3. *Аналітичне групування* служить для дослідження взаємного зв'язку між факторами (незалежними змінними) та результатом (залежною змінною).

Якщо групування проводять за атрибутивною (лінгвістичною, якісною) ознакою, то кількість груп, на які поділяється сукупність, визначається кількістю різновидів (варіант) цих ознак.

Групування з великою кількістю ознак стає ненаочним. Тому складні групування використовують не більш ніж три ознаки. Ознаки поділяються на атрибутивні (лінгвістичні: описові, якісні), або кількісні (варіаційні); на абсолютні, відносні та середні величини.

Форма зображення статистичних даних

Статистичну таблицю тлумачать як форму логічного статистичного речення, що містить підмет і присудок. Підметом таблиці є об'єкт дослідження (те, про що йдеться в таблиці): перелік елементів сукупності, їх групи, окремі територіальні одиниці або часові інтервали тощо. Звичайно підмет розміщують зліва, у назві рядків. Присудок таблиці – це система показників, що характеризують підмет в кількісній формі. Присудок формує верхні заголовки і складає зміст графу з логічно послідовним розміщенням показників зліва направо. Розміщення підмета і присудка може мінятися місцями. Макет статистичної таблиці наведений на рис. 5.5.1.

Підмет (зміст рядків)	Присудок (назви граф – верхніх заголовків)			
А	1	2	...	n
Пілсумковий рядок				Пілсумкова графа

Рис. 5.5.1. Макет статистичної таблиці

В статистичній таблиці не повинно бути пустих комірок: відсутність ознаки позначають *знаком тире, відсутність відомостей – трьома крапками*.

Побудов графіків статистичних даних можна виконати у пакетах STATISTICA, SPSS, STATGRAPHICS, EXCEL та інших. Багатовершинність графіків свідчить про *неоднорідність досліджуваної сукупності* і вимагає перегрупування даних для виділення більш однорідних груп. *Коефіцієнти асиметрії та ексцесу* визначають ступінь скошеності і рівень гостроти чи плосковершинності розподілу даних.

Індекси – відносні величини ознак (у вигляді коефіцієнтів, відсотків), які характеризують зміну *в часі* чи просторіскладних суспільних явищ, ступіні відхилення значення показника від певного стандарту та ін. Індекси належать до найбільш поширених статистичних динамічних показників. Для подолання непорівняльностіознак складного явища, застосовують *сумірник* (звичайно, вартість).

Основною формою індексу є агрегатна. Суми добутків показників, що індексуються (наприклад, q – кількість та p – ціна) *створюють агрегат* $\sum q_p$.

Динамічний (часовий) ряд – це таблична чи графічна сукупність значень статистичних показників, які розташовані у хронологічному порядку і подається у вигляді таблиці чи графіка.

Для динамічного (часового) ряду:

1. Визначають середні характеристики з урахуванням коливання (зміни) сезонних рівнів часового ряду.

2. Для вирівнювання ряду використовується метод збільшення інтервалів часу (замість щоденних обчислюються щомісячні рівні).

3. Для вирівнювання ряду використовується *метод плинної (ковзної) середньої* – спочатку знаходять середнє арифметичне значення з перших m рівнів ряду, потім знаходять нову середню з того ж числа рівнів, але зі зміщенням дати на одиницю. Знайдене значення середньої *стосується того періоду, який знаходиться в середині інтервалу*.

4. Виконують *інтерполяцію* (знаходження показника в середині ряду на основі закономірності розвитку явища за період) та *екстраполяцію* (знаходження *прогнозованих показників динаміки* наприкінці чи на початку динамічного ряду за межами наявних фактичних даних).

5. Для деяких часових рядів *виявляють сезонні коливання*, які вимагають використання даних *кількох років* (частіше це місячні або квартальні дані за 3 роки), оскільки окремі коливання можуть бути зумовлені випадковими факторами.

Факторний аналіз – статистичний метод, який використовується при обробці великих масивів експериментальних даних. Завданнями факторного аналізу є: скорочення числа змінних (редукція даних) та визначення структури взаємозв'язків між змінними, тобто класифікація змінних, тому факторний аналіз використовується як метод скорочення даних або як метод структурної класифікації.

5.5.3. Методи вторинної статистичної обробки результатів експерименту

Друга група методів математичної статистики дозволяє судити про *функціональну залежність і точність* статистик вибірки: *метод найменших квадратів, дисперсійний аналіз, регресійний аналіз.*

Третя група методів дозволяє судити про *статистичні зв'язки*, що існують між змінними величинами: *кореляційний аналіз, методи порівняння вибіркових даних.*

Методи вторинної статистичної обробки результатів експерименту стосуються:

- *регресійного числення* за методом найменших квадратів (МНК);
- *методів порівняння* між собою двох або декількох елементарних статистик (середніх, дисперсій і т. п.), що відносяться до різних вибірок;
- *методів оцінки статистичних взаємозв'язків між змінними*, наприклад їх кореляції один з одним;
- *методів виявлення внутрішньої статистичної структури* емпіричних даних (наприклад, *факторний аналіз*).

Метод найменших квадратів (МНК) дозволяє знайти лінійну апроксимаційну формулу для змінних у вигляді (для двох змінних)

$$F_t = \beta_0 x_{0t} + \beta_1 x_{1t} + \beta_2 x_{2t} + u_t, \quad (5.5.1)$$

де $\beta_0, \beta_1, \beta_2$ – невідомі коефіцієнти; $x_{0t}=1=\text{const}$ – штучно введена змінна (для зручності розрахунків); u_t – випадкова величина; t – порядковий номер спостереження.

Для визначення невідомих коефіцієнтів $\beta_0, \beta_1, \beta_2$ використовують матрицю спостережень з відомими даними (табл. 5.5.1).

Таблиця 5.5.1.

Матриця спостережень

t	x_{0t}	x_{1t}	x_{2t}	F_t
1	1	x_{11}	x_{21}	F_1
2	1	x_{12}	x_{22}	F_2
3	1	x_{13}	x_{23}	F_3
4	1	x_{14}	x_{24}	F_4

Регресійний аналіз полягає у визначенні *аналітичного виразу кореляційного зв'язку* – опису виду і параметрів функції зв'язку (регресійної моделі). Термін «регресія» (від лат. *regredior* – повертаюсь) *означає повернення до середньої.*

В матричній формі математична модель (5.5.1) має вигляд

$$F = XB + U, \quad (5.5.2)$$

$$F = \begin{bmatrix} F^0 \\ F_{x_1} \\ F_{x_2} \\ F^m \end{bmatrix}; \quad X = \begin{bmatrix} 1 & x_1^{MT} & x_2^{MT} \\ 1 & x_1^{CT} & x_2^{MT} \\ 1 & x_1^{MT} & x_2^{CT} \\ 1 & x_1^{CT} & x_2^{CT} \end{bmatrix}; \quad B = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}; \quad U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}.$$

де

Для розв'язання задачі ми розглядаємо рівняння (5.5.2) як його математичне очікування. Із статистичного співвідношення (тут E – математичне очікування)

$$E(F) = E(XB + U) = E(XB)$$

після перетворень

$$F = XB; \quad X^T F = (X^T X)B; \quad (X^T X)^{-1} X^T F = (X^T X)^{-1} (X^T X)B;$$

отримуємо

$$B = (X^T X)^{-1} X^T F, \quad (5.5.3)$$

де $(X^T X)$ – квадратна матриця, симетрична відносно головної діагоналі; $(X^T X)^{-1} (X^T X)$ – одинична матриця (усі елементи дорівнюють нулю, а діагональні елементи дорівнюють одиниці), яку можна опустити.

Метод кореляцій показує, яким чином одна змінна впливає на іншу. Якщо підвищення рівня однієї змінної супроводжується підвищенням рівня інший, то кажуть про позитивну кореляцію, а в іншому разі кажуть пронегативну кореляцію. Привід сутності зв'язку кажуть пронульову кореляцію.

Є кілька різновидів цього методу кореляції: лінійний, ранговий, парний і множинний.

Лінійний кореляційний аналіз дозволяє встановлювати прямі зв'язки між змінними величинами з їхнім абсолютним значенням. Ці зв'язки графічно виражаються прямою лінією, звідси назва «лінійний».

Рангова кореляція визначає залежність не між абсолютними значеннями змінних, а між порядковими місцями, або рангами, зайнятими ними в упорядкованому за величиною ряду.

Парний кореляційний аналіз вивчає кореляційні залежності між парами змінних.

Множинний, або багатовимірний вивчає кореляційні залежності між багатьма змінними одночасно.

Поширеною у прикладній статистиці формою багатовимірного кореляційного аналізу є *факторний аналіз*.

Коефіцієнт лінійної кореляції визначається за формулою

$$r_{xy} = \frac{\sum_{j=1}^n [(x_j - x_{CP})(y_j - y_{CP})]}{n \sqrt{S_x^2 S_y^2}},$$

де r_{xy} – коефіцієнт лінійної кореляції; $j = 1, 2, \dots, n$ – порядковий номер змінної; x_j, y_j – значення змінних у вибірці; x_{CP}, y_{CP} – середні вибіркові значення порівнюваних величин; n – загальна кількість змінних; S^2_x, S^2_y – дисперсії, квадрати відхилень порівнюваних величин від середніх значень.

Більшість показників, які отримують у психолого-педагогічних дослідженнях, належать до порядкових, а не до інтервальних шкал (наприклад, оцінки типу «так», «ні», «скоріше ні, ніж так» та інші, які можна переводити у бали), тому коефіцієнт лінійної кореляції до них непридатний. У цьому випадку звертаються до використання *коефіцієнта рангової кореляції*, формула якого має вигляд:

$$R_s = 1 - \frac{6 \sum_{\alpha=1}^N d_{\alpha}^2}{N^3 - N},$$

де R_s – коефіцієнт рангової кореляції за Спирмену; d_{α} – різниця між рангами показників в одних і тих же упорядкованих рядах; N – число досліджуваних або цифрових даних (рангів) у корельованих рядах.

Метод множинних кореляцій (навідміну від методу парних кореляцій) дозволяє виявити і представити у вигляді деякої системи загальну структуру кореляційних залежностей всередині багатовимірного експериментального матеріалу, що включає більше двох змінних.

Отже на відміну від функціональних детермінованих, стохастичні зв'язки *неоднозначні*. Такі зв'язки виявляються як узгодженість варіації двох чи більше ознак.

На відміну від функціональної залежності, *кореляційний зв'язок є неповним* (відзначають лише певне співвідношення між причиною і наслідком), оскільки залежність між функцією (\hat{y}) і аргументом (x) у кожній ситуації перебуває під впливом інших факторів.

Кореляційна залежність може встановлюватися для пари показників (*парна кореляція*) або для декількох показників (*множинна*) кореляція.

Отже у тих випадках, коли зростання величини факторної ознаки тягне за собою зростання величини результативної ознаки, говорять про можливість наявності *прямого кореляційного зв'язку*. Якщо ж із збільшенням факторної ознаки величина результативної ознаки має тенденцію до зменшення, то можна припустити, що існує *зворотній кореляційний зв'язок* між ознаками.

Визначити внесок кожного з факторів, а також тісноту зв'язку дозволяють *методи кореляційно-регресійного та дисперсійного аналізу*.

Ступінь відповідності між фактичним і теоретичним розподілами оцінюється за допомогою показників (критеріїв) узгодженості. *Критерій згоди* – це статистичний показник, що використовується для з'ясування розбіжностей між прийнятою статистичною моделлю і спостережуваними даними ознаки, які має описати ця модель.

Серед інших критеріїв, одним з найпоширеніших є критерій Пірсона (хі-квадрат) – χ^2 . Чим більшою буде різниця між емпіричними та теоретичними частотами, тим *більше* буде значення критерію Пірсона. Розраховане фактичне значення критерію порівнюють з табличним (критичним) значенням:

– якщо $\chi^2_{\text{ф}} > \chi^2_{\text{табл}}$, тобто χ^2 , то попадає у критичну область: емпіричний розподіл є принципово відмінним від теоретичного;

– якщо $\chi^2_{\text{ф}} < \chi^2_{\text{табл}}$, то емпіричний розподіл відповідає теоретичному.

Необхідною умовою використання цих критеріїв є достатньо велике число спостережень – не менше 100 (20–30 за критерієм Пірсона).

Адекватність регресійної моделі, тобто можливість *надійного прогнозування середніх значень* за даними вибірки перевіряють за *F – критерієм Фішера і методом перевірки гіпотез*. Якщо $F_{\text{розн.}} > F_{\text{табл.}}$, то гіпотеза про значущість рівняння приймається.

Слід звернути увагу на те, що *моделям, які побудовані на основі рівнянь регресії, притаманні слабкі екстраполяційні властивості* і тому вони використовуються для побудови *короткочасних прогнозів*.

Інтерпретація моделей регресії дозволяє виявити *лише резерви розвитку* досліджуваних явищ.

Для кількісної оцінки сили зв'язку взаємозв'язаних ознак статистика використовує низку коефіцієнтів.

Найпоширенішим є лінійний **коефіцієнт кореляції Пірсона** (r) – характеризує тісноту і напрям зв'язку між двома ознаками, що корелюють у випадку наявності між ними лінійної залежності. Змінюється в межах: $-1 \leq r \leq 1$. При прямому зв'язку r – величина додатна, при зворотному – від'ємна. Знаки коефіцієнтів кореляції і регресії співпадають.

Існують непараметричні методи дослідження перевірки адекватності, які не вимагають числового вираження значень ознак, обчислення параметрів розподілу, інформації про розподіл ознак в сукупності. *Але вони забезпечують лише оцінку щільності зв'язку та перевірку його істотності і не дають змогу представити зв'язок аналітично.*

Методи рангової кореляції використовуються для обробки даних соціологічних досліджень (анкет), медичних обстежень, рейтингів, експертних оцінок тощо, тобто там, де ознаки вимірюються за допомогою номінальної та порядкової шкали (наприклад, «стать», «соціоекономічний статус сім'ї», «діагноз» і т. ін.).

При ранговій кореляції кожній одиниці сукупності надається ранг (порядковий номер у вигляді 1, 2, 3, ...). *на основі віддавання переваг* для кожної ознаки окремо. Зі збільшенням кількості рангів ступінь «розпізнаваності» елементів зменшується, тому *рангові оцінки щільності зв'язку доцільно використовувати для сукупностей невеликого обсягу.*

До рангових оцінок щільності належать коефіцієнт кореляції рангів Спірмена (базується на основі різниці рангів) та Кендала. Ці коефіцієнти

мають ті самі властивості, що і лінійний коефіцієнт кореляції – змінюються в тих же межах, оцінюють щільність зв'язку та вказують його напрям.

Зв'язок між ознаками можна визнати статистично істотним, якщо значення коефіцієнтів рангової кореляції Спірмена і Кендала більше 0,5.

5.6. Генетичні алгоритми еволюційне програмування

Генетичні алгоритми (ГА) є комбінаторно-статистичними методами відбору і спадкування, які застосовують для збереження «життя» ліпшої хромосоми X_j . Вони запропоновані італійсько-норвежським математиком Баричеллі Н. А. (1954), який використав ідеї Ч. Дарвіна стосовно природного (еволюційного) розвитку на основі спадковості, зміни властивостей та природного добору. Значний внесок в генетичні алгоритми внесли і інші дослідники, наприклад, Фрастер А. С. та Холенд Д. Г. [150; 160].

Еволюцію розглядають як перехід від попереднього гіршого об'єкта-хромосоми X_j до наступного ліпшого об'єкта-хромосоми X_j .

Особливості ГА:

- при створенні хромосоми та її зміні використовують ймовірнісні правила;
- обробляють не значення параметрів задач, а закодовані форми хромосом X_j ;
- виконують пошук рішення виходячи не з однієї точки, а з популяції $P=\{X_1, X_2, \dots, X_j, \dots, X_n\}$;
- використовують лише цільову функцію F , а не її похідні чи іншу додаткову інформацію.

Генетичні алгоритми увійшли зараз у стандартний інструментарій методів інтелектуальних обчислень. Найліпшою вважається хромосома з максимальним значенням функції мети.

Структурна схема ГА наведена на рис. 5.6.1.

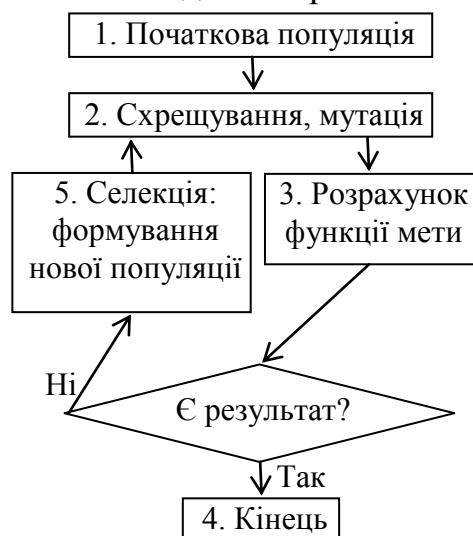


Рис. 5.6.1. Структурна схема ГА

Припустимо, що ми розглядаємо розв'язок рівняння

$f_j = a + 2b + 3c + 4d = 24$. Нашою метою є отримання коренів хромосоми $X_j = (a, b, c, d)$, які б дали значення вільного члена. Генетичний алгоритм пояснюється на прикладі табл. 5.6.1 та табл. 5.6.2.

Початкова популяція об'єктів $P = \{X_1, X_2, \dots, X_j, \dots, X_n\}$ утворюється з $n=5$ хромосом $X_j = (a, b, c, d)$, які є деякими розв'язками задачі і кожна з яких складається з 4-х генів. У цьому випадку для рівняння $f_j = a + 2b + 3c + 4d = 24$ вважаємо, що всі розв'язки мають цілочислові значення і повинні знаходитись у межах відрізка $0 \dots 24$. Спочатку за допомогою генератора випадкових чисел отримуємо початкову популяцію хромосом $X_j = (a, b, c, d)$ (табл. 5.6.1).

Таблиця 5.6.1.

**Перевірка початкової популяції
на рівність $f_j = a + 2b + 3c + 4d = 24$**

№	$X_j = (a, b, c, d)$	$f_j = a + 2b + 3c + 4d$	$\Delta_j = f_j - 24 $	$1/\Delta_j$	Δp_j	Відрізок
1	(2, 15, 10, 21)	$2 + 30 + 30 + 84 = 146$	122	0,0082	0,14	$0 \dots 0,14$
2	(15, 1, 17, 8)	$15 + 2 + 51 + 32 = 100$	76	0,0132	0,22	$0,14 \dots 0,36$
3	(11, 4, 19, 12)	$11 + 8 + 57 + 48 = 124$	100	0,0100	0,16	$0,36 \dots 0,52$
4	(22, 18, 3, 1)	$22 + 36 + 9 + 4 = 71$	47	0,0213	0,35	$0,52 \dots 0,87$
5	(4, 9, 23, 14)	$4 + 18 + 69 + 56 = 147$	123	0,0081	0,13	$0,87 \dots 1,00$

Далі для кожної хромосоми X_j розраховуємо відповідну функцію мети $f_j = a + 2b + 3c + 4d = 24$ та різницю

$\Delta_j = |f_j - 24| \rightarrow \min$, яка оцінює оптимальність пристосування хромосоми X_j до «наколишнього середовища».

Хромосома $X_j = (a, b, c, d)$ є набором генів, де кожний ген (a, b, c, d) описує один розв'язок рівняння $f_j = a + 2b + 3c + 4d = 24$. У більш складному випадку **ген** $G_{ji} = \{A_{jil}, \dots, A_{jie}\}$ може складатись з послідовності **E аллелів** A_{jie} (**цілих чисел**), які є подальшим дробленням використаних складових хромосоми X_j .

З табл. 5.6.1 видно, що більше наближеними до розв'язку є хромосоми 4 та 2 (вони повинні мати більшу ймовірність розмноження), а найбільш віддалені від розв'язку є хромосоми 5 та 1, які повинні мати меншу ймовірність розмноження.

Для схрещування повинні відбиратись пари батьківських хромосом з найбільшою «життєздатністю» (які дають найліпший розв'язок для рівняння $f_j = a + 2b + 3c + 4d = 24$). Разом з тим потрібно урахувати, що батьки обираються випадковим чином серед хромосом всієї початкової популяції без винятку. Цих вимог можна дотриматись, якщо обрання пар батьків для схрещування відбувається випадковим чином за методом «рулетки» (існують і інші методи вибору батьків):

– пошук адрес батьків відбувається стохастично, для чого на відрізку $0 \dots 1$ розміщуються всі 5 хромосом X_j початкової популяції без винятку з розміщенням кожної хромосоми на своєму відрізку, який розраховується

за формулою $\Delta p_j = \frac{1/\Delta_j}{\sum_{j=1}^5 (1/\Delta_j)}$. З формули бачимо, що найбільш «життєздатні» хромосоми займають більшу частку відрізка 0...1 і мають більшу ймовірність попасти у батьки (табл. 5.6.1, колонка «Відрізок»);

– кожна хромосома має свою частку на відріжку 0...1; по нумерації відріжку, у який попало число, генероване генератором рівномірно розподілених випадкових чисел (яке попадає на відповідний відрізок), обираємо хромосому першого батька, а по генерованому наступному випадковому числу обираємо хромосому другого батька. Отримана пара батьків повинна схрещуватись (табл. 5.6.2).

Таблиця 5.6.2.

Операція схрещування початкової популяції

№	Номери пар батьків	Схрещувані пари		Нащадок	Функція нащадка f_j	$\Delta_j= f_j-24 $
		Батько	Мати			
1	(2;4)	(15, 1, 17, 8)	(22, 18, 13, 1)	(15, 1, 3, 1)	30	6
2	(3;4)	(11, 4, 19, 12)	(22, 18, 3, 1)	(11, 18, 3, 1)	60	36
3	(4;3)	(22, 18, 3, 11)	(11, 4, 19, 12)	(22, 18, 3, 12)	115	91
4	(1;2)	(2, 15, 10, 21)	(15, 1, 17, 8)	(2, 15, 17, 8)	115	91
5	(4;2)	(22, 18, 3, 1)	(15, 1, 17, 8)	(22, 1, 17, 8)	107	83

Для застосування схрещування для кожної пари батьків за методом «рулетки» обираємо випадковим чином місце розриву хромосом (воно помічене в табл. 5.6.2 жирною вертикальною рисою). **Операція схрещування** полягає у обміні фрагментами ланцюгів між двома батьківськими хромосомами (від двох батьків можна отримати двох нащадків рис. 5.6.2).

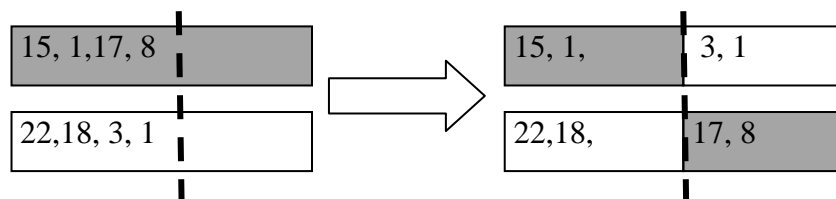


Рис. 5.6.2. Операція схрещування

З табл. 5.6.2 видно, що в результаті схрещування нові значення хромосом X_j мають ліпші показники наближення до функції $f_j = a+2b+3c+4d = 24$.

Мутація – це випадкова зміна генів у деяких індивідів популяції. В ГА **мутація може** виконуватись на хромосомах перед схрещуванням або після схрещування. Під мутацією розуміють зміну числа випадкового обраної позиції на випадкове число у межах 1...24.

Після виконання схрещування та мутації розраховується функція мети і визначається найліпший розв’язок, в результаті чого алгоритм або зупиняється (переходить на «4. Кінець»), або переходить до нового кроку – селекції нової популяції.

В результаті селекції вся попередня популяція замінюється популяцією нащадків з *тією ж кількістю хромосом*, але при ліпшому значенні їх генів (ліпшому наближенню функції мети до оптимального розв'язку).

Процес схрещування і мутації також називається *еволюцією*, на якому відбувається рекомбінація генів у хромосомах. У цій еволюції діють три механізми:

- *відбір найсильніших* – наборів хромосом, яким відповідають оптимальні рішення;
- *схрещування* – виробництво нових індивідів за допомогою змішування хромосомних наборів відібраних індивідів;
- *мутації* – випадкові зміни генів у деяких індивідів популяції.

Еволюційне програмування полягає у тому, що на будь якій мові складається програма, по якій визначається функція мети будь-якої інтелектуальної системи. Процес удосконалення функції мети розглядається як еволюція у світі програм. Тому метод еволюційного програмування схожий з генетичними алгоритмами. Програма по замовленню користувача формулює гіпотези про додаткові залежності функції мети від змінних та їх комбінацій.

Коли система знаходить програму, що досить точно відображує шукану функцію мети, вона починає вносити в неї невеликі модифікації і відбирає серед побудованих таким чином дочірніх програм ті, які підвищують точність.

Тобто система наче «вирощує» кілька генетичних ліній програм, що конкурують між собою поточності відображення функції мети. Транслюючий модуль переводить знайдені залежності на зрозумілі користувачу математичні формули, таблиці та ін.

Пошук залежності функція мети від змінних проводиться у формі функцій якогось визначеного виду. Наприклад, в одному з найбільш розповсюджених алгоритмів цього типу (методі групового урахування аргументів – МГУА) функцію мети шукають у формі поліномів. При цьому намагаються складні поліноми замінювати кількома більш простими.

5.7. Метод групового врахування аргументів (МГУА)

Метод групового врахування аргументів (МГВА) використовує алгоритм переробки інформації у природі (О. Г. Івахненко, СРСР, 1971): висівається насіння; вирощують рослини – складні успадковані комбінації, і лише з оптимальної частини рослин (з потрібними властивостями) обирається оптимальна кількість насіння; далі знову висівають це обране насіння – і етап повторюється. Через кілька поколінь селекція зупиняється на оптимальному результаті, бо якщо і далі продовжувати селекцію, то наступає «інцухт» – виродження рослин.

Алгоритми МГУА використовують цей підхід. Згідно з цим методом виконуються такі етапи [34]:

1. Отримується початковий «повний» математичний опис об'єкту у вигляді елементарної функції (наприклад, у вигляді степеневого поліному)

$$\varphi = f(X_1, X_2, \dots, X_m).$$

2. Функція f замінюється кількома наступними послідовними рядами, які складаються з окремих виразів

$$1\text{-ряд селекції: } Y_1 = f(X_1, X_2), Y_2 = f(X_1, X_3), \dots, Y_s = f(X_{m-1}, X_m);$$

$$2\text{-ряд селекції: } Z_1 = f(Y_1, Y_2), Z_2 = f(Y_1, Y_2), \dots, Z_p = f(Y_{s-1}, Y_s) \text{ і т. д.}$$

Вхідні аргументи и проміжні змінні використовуються попарно, складність комбінацій на кожному ряду обробки інформації зростає (як при масовій селекції), доки не буде отримана одна модель оптимальної складності. Деякі проміжні змінні вилучаються, якщо це можливо. Із ряду у ряд селекції переходять *лише деяка кількість регулярних змінних*. Ступень регулярності оцінюється або по середньоквадратичній похибці (середньої у даному ряді селекції) або по коефіцієнту кореляції.

3. Ряди селекції нарощуються, а оптимальний математичний опис отримується при мінімумі похибки. Тоді селекцію зупиняють, щоб не мати «інцухт». Практично рекомендують зупинити селекцію дещо раніше повного мінімуму похибки, бо це дозволяє отримати більш прості і більш вірогідні рівняння.

Можливе використання послідовних рядів у вигляді

$$Y_i = a_0 + a_1 X_i + a_2 X_j + a_3 X_i X_j; \quad Y_k = a_0 + a_1 X_i + a_2 X_j + a_3 X_i X_j + a_4 X_i^2 + a_5 X_j^2.$$

Складність моделі збільшується при переміщенні по рядах селекції від входу до виходу. Складність зростає як по кількості аргументів, що враховуються, так і по степені аргументів. У першому рядку використовуються квадратичні описи, у другому – четвертої степені, у третьому – восьмої степені і т. д. Нарощування складності припиняється, якщо у селекції збільшується критерій середньоквадратичної похибки.

На основі МГВА виконують також нейронні мережі [103].

6. Вирішальні функції та вирішальні правила

6.1. Особливості вирішальних функцій та вирішальних правил

Вирішальні функції [18]. Припустимо, що нам відомі області існування трьох класів $\Omega_1, \Omega_2, \Omega_3$. Звичайно при проектуванні НМ кількість класів, що розпізнається збільшується на одиницю: додатковий клас Ω_4 має назву «Не розпізнаний об'єкт» і призначений для об'єктів, які не розпізнаються. Для пояснення такого принципу розпізнавання розглянемо класи $\Omega_1, \Omega_2, \Omega_3$ на площині ознак $\{x_1, x_2\}$ (рис. 6.1.1).

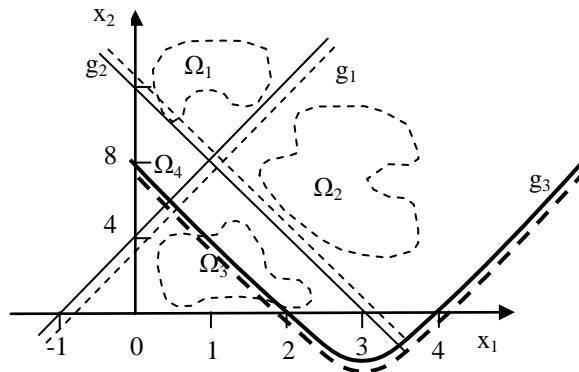


Рис. 6.1.1. Простір ознак класів $\Omega_1 - \Omega_4$

Вказані на рис. 6.1.1 ділянки класів можуть бути розділені двома довільними безперервними прямими лініями g_1 та g_2 , які і є вирішальними (розділюючими) функціями чотирьох класів (з врахуванням додаткового класу Ω_4 «Не розпізнаний об'єкт»). Після накреслення цих двох довільних прямих ліній, що розділяють класи, визначають вирішальні (розділюючі) функції цих прямих ліній по відрізках, які вони відсікають на осях координат.

Для функції g_1 отримуємо рівності $\frac{x_1}{-1} + \frac{x_2}{4} = 1$; $4x_1 - x_2 = -4$;

$g_1^0 = 4x_1 - x_2 + 4 = 0$, останню з яких перетворюємо у нерівність (за традицією знак рівності « $=$ » замінюють на знак нерівності « \geq »); але з рівним успіхом можна використовувати й знак « \leq »)

$$g_1 = 4x_1 - x_2 + 4 \geq 0 \quad (6.1.1)$$

Аналогічно для функції g_2 отримуємо $\frac{x_1}{3} + \frac{x_2}{12} = 1$; $4x_1 + x_2 = 12$, звідки

$$g_2 = 4x_1 + x_2 - 12 \geq 0. \quad (6.1.2)$$

Кожна з двох прямих (6.1.1) та (6.1.2) розділяє площину ознак $\{x_1, x_2\}$ на дві напівплощини: в одній напівплощині нерівність виконується, а в іншій – ні.

Наприклад, для точки $\{x_1 = x_2 = 0\}$ згідно (6.1.1) величина $g_1 = 4 > 0$. Ця нерівність виконується для всій напівплощини, у якій знаходиться точка $\{x_1 = x_2 = 0\}$. У цих твердженнях можна пересвідчитись розрахунками для будь-яких інших довільних точок всієї площини ознак $\{x_1, x_2\}$. Напівплощина, у якій виконується нерівність $g_1 > 0$, позначається пунктирною лінією. Тоді відповідно для іншої напівплощини, не позначеною пунктирною лінією, нерівність $g_1 > 0$ не виконується. Таким чином, достатньо отримати значення функції для будь-якої довільної точки, щоб визначити, у якій напівплощині виконується нерівність $g_1 > 0$. Звичайно така перевірка виконується для точки $\{x_1 = x_2 = 0\}$.

Таким же самим чином позначається пунктирною лінією напівплощина, у якій виконується нерівність $g_2 > 0$.

Вирішальні правила використовують вирішальні (розділяючі) функції для визначення класів (див. рис. 6.1.1):

1. Клас Ω_1 : якщо $g_1 < 0$ та $g_2 \geq 0$;
 2. Клас Ω_2 : якщо $g_1 \geq 0$ та $g_2 \geq 0$;
 7. Клас Ω_3 : якщо $g_1 \geq 0$ та $g_2 < 0$;
 4. Клас Ω_4 : якщо $g_1 < 0$ та $g_2 < 0$.
- (6.1.3)

У цьому випадку для визначення класів ми використали нерівності, які, як відомо з математики, виконують логічні функції. Можливо отримати й апаратну реалізацію виразів (6.1.1) – (6.1.3).

У принципі вирішальні функції можуть бути не лише прямими лініями, але й *нелінійними*, наприклад, мати вигляд *параболи* $g_3 = x_1^2 - x_2 - bx_1 + 8 \geq 0$ (рис. 6.1.1), при цьому пунктирна парабола частку півплощини, в якій нерівність виконується.

Максимізація мінімальної відстані вирішальної функції від двох сусідніх класів. По принципу дії апарат вирішальних функцій та вирішальних правил мало відрізняється від нейронних мереж (НМ), які виконують аналогічний логічний аналіз по розділу класів з аналогічним математичним та графічним поясненнями. Для обох напрямків аналізу **розміщення вирішальних функцій є невизначеним**: з рис. 6.1.1 видно, що для розділу класів можна використати безліч вирішальних функцій. Виникає проблема визначення «оптимальних» вирішальних функцій. Метод вирішальних правил розглядається окремо від НМ, бо він у порівнянні з середовищем НМ з її більш жорсткими обмеженнями дозволяє зручніше виділити (шляхом математичних перетворень) «оптимальні» вирішальні функції. З цією метою використовується, наприклад, *метод «опорних векторів»*, за яким у багатомірному гіперпросторі у двох сусідніх класах виділяють найближчі сусідні «опорні вектори» двох різних класів, між якими й визначають положення розділюючої їх «оптимальної» гіперплощини (цей метод використовується також й для розділу багатьох класів).

6.2. Вирішальні функції та вирішальні правила для складного розміщення класів

Вирішальні функції та вирішальні правила використовуються у n -вимірному просторі кількісних ознак $\{x_1, x_2, \dots, x_n\}$. Але далеко не завжди у таких НМ класи розміщуються таким чином, щоб між ними було зручно визначати вирішальні функції. Припустимо, що нам відомі області існування трьох класів $\Omega_1, \Omega_2, \Omega_3$ на площині ознак $\{x_1, x_2\}$ (рис. 6.2.1), але класи $\Omega_1, \Omega_2, \Omega_3$ «переплутані» між собою [5; 11].

У цьому випадку ми можемо довільними вирішальними функціями « $g_1 > 0$ », « $g_2 > 0$ », « $g_3 > 0$ » розділити між собою класи. На базі отриманих вирішальних функцій та відомого розміщення класів згідно рис. 6.2.1 отримуємо вирішальні правила для визначення класів:

1. **Клас Ω_1** , якщо ($(g_1 > 0$ та $g_3 < 0$) або $(g_1 < 0$ та $g_3 < 0$) або $((g_1 < 0$ та $g_2 < 0$) та $g_3 > 0$) або $(g_1 < 0$ та $g_2 > 0)$).
2. **Клас Ω_2** , якщо $(g_1 > 0$ та $g_2 > 0)$.
3. **Клас Ω_3** , якщо $((g_1 > 0$ та $g_2 < 0$) та $g_3 > 0)$.

Можна впевнитись, що вирішальні функції однозначно визначають потрібні класи.

Таким чином, для класифікації об'єктів не є завадою те, що вирішальна функція пересікає область існування класу. У зв'язку з цим всю область існування реальних ознак всіх класів можна розділити на однакові комірки, для кожної з яких можна визначити клас за вирішальними правилами.

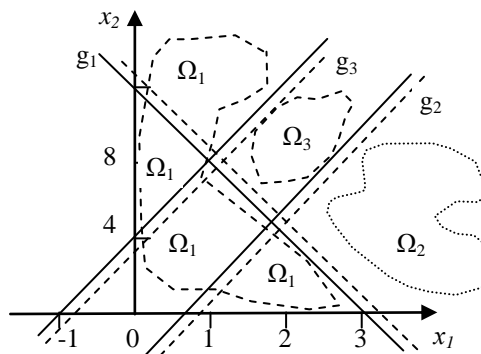


Рис. 6.2.1. Простір ознак класів $\Omega_1, \Omega_2, \Omega_3$.

З рис. 6.2.1 видно, що *розміщення вирішальних функцій є невизначеним*: для розділу класів можна використати безліч вирішальних функцій.

Завдання

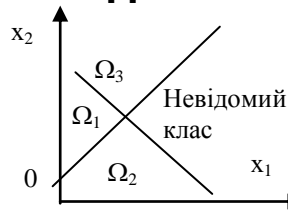


Рис. 6.2.2. Розміщення трьох класів

У площині 1-го квадранту з осей x_1 , x_2 (рис. 6.2.2), на яких позначаються вимірні ознаки при максимальних значеннях $x_{1max} = N$, $x_{2max} = N$ де N – порядковий номер студента у групі, виділити двома довільними прямими лініями три довільні ділянки (класи): Ω_1 , Ω_2 , Ω_3 . Четверту ділянку розділу вважати невідомим класом.

Отримати:

- вирішальні функції та вирішальні правила для визначення класів;
- схему апаратної реалізації системи РО;
- налагоджену програму (на будь-якій мові) із використанням вирішальних функцій та вирішальних правил.

Примітка. Для отримання вирішальної функції використовують рівняння $x_1/a_1 + x_2/b_2 = 1$ (тут a_1 , b_2 – координати точок пересічення прямою вирішальної функції відповідно осей x_1 , x_2) з наступним переведенням його у нерівність вигляду $u = b_2x_1 + a_1x_2 \geq a_1 b_2$.

У налагодженій програмі використати генератор випадкових чисел для генерації ознак x_1, x_2 об'єктів у визначених межах. На екрані монітору у графічному режимі показати: осі x_1, x_2 ; положення вирішальних функцій; точки, які генеруються генератором випадкових чисел; показати у відсотках кількість точок, яка попала у кожний клас та кількість точок, яка попала до невідомого класу.

6.3. Апаратна реалізація вирішальних функцій та вирішальних правил

Розглянемо апаратну реалізацію системи РО згідно співвідношень (6.1.1) – (6.1.2), які на рис. 6.3.1 реалізуються суматорами (операційними підсилювачами) Σ_1 та Σ_2 .

При множенні вхідних для суматорів величин (для суматора Σ_1 це «40 В», «+ x_1 », «- x_2 ») на відповідні коефіцієнти передачі (для суматора Σ_1 це 0,1; 4; 1) отримуємо значення вирішальних (розподіляючих) функцій «- g_1 » та «- g_2 » зі зворотнім знаком згідно властивостям аналогових суматорів.

Потрібно відмітити, що підсумки $(4+4x_1-x_2)$ та $(-12+4x_1+x_2)$ отримуються на резистивних входах операційних підсилювачів Σ_1 та Σ_2 . Роль операційних підсилювачів Σ_1 та Σ_2 полягає лише у підсилюванні потужності

сигналу виходу з тим, щоб їх виходи могли бути джерелами входів для багатьох стандартних аналогових елементів.

Інвертори Інв.1 та Інв.2 знаки вхідних функцій « $-g_1$ » та « $-g_2$ » підсилювачів Σ_1 та Σ_2 змінюють на протилежні: « $+g_1$ » та « $+g_2$ ».

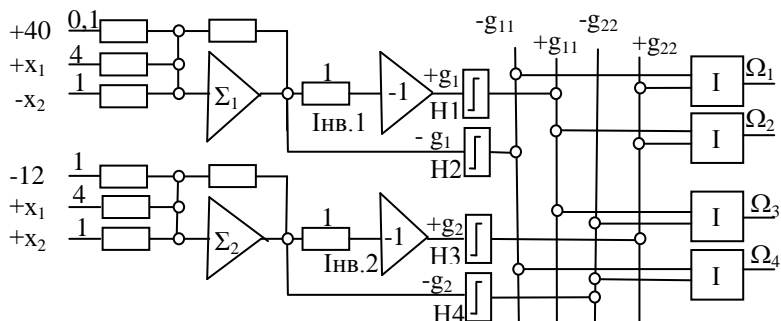


Рис. 6.3.1. Апаратна реалізація системи РО з вирішальними функціями та вирішальними правилами

Нелінійні елементи Н1-Н4 виконують роль активаційних функцій нейронів і забезпечують нелінійність вихідного сигналу. Функції (g_1, g_2) перетворюються на (g_{11}, g_{22}) після проходження через нелінійні елементи Н1-Н4, які перетворюють від'ємні значення на «0», а позитивні значення на «+1».

Логічні функції «I» виконують роль вирішальних правил (6.1.3) при розпізнаванні образів. Наприклад, для класу Ω_1 згідно вирішальних правил потрібно, щоб вирішальні функції мали значення $g_1 < 0$ та $g_2 \geq 0$.

В результаті ми отримали апаратну реалізацію на операційних підсилювачах системи РО з вирішальними функціями та вирішальними правилами, яка практично не відрізняється від нейронної мережі.

При апаратній реалізації, ми повинні використати:

1. Кількість суматорів Σ_j , яка дорівнює кількості вирішальних функцій g_j .
2. Кількість інверторів «Інв_j», яка дорівнює кількості вирішальних функцій g_j .
3. Кількість нелінійних елементів «Н_j», яка дорівнює подвійній кількості вирішальних функцій g_j .
4. Кількість логічних елементів «I» (визначається по кількості слів «I» в логічних умовах вирішальних правил).
5. Кількість логічних елементів «АБО1-АБО3» (визначається по кількості слів «АБО» в логічних умовах вирішальних правил).

7. Загальні відомості про нейрони

7.1. Історія нейронних мереж

Основною функціональною одиницею мозку людини є нервова клітина – нейрон, яких налічується приблизно 100 типів загальною кількістю приблизно 10^{12} нейронів (нервова система мурашки потребує менше 1000 нейронів).

Наразі існують два напрямки дослідження штучних нейронів, які відрізняються апаратною та програмною реалізацією нейронів та метою роботи:

1. *Моделювання реальних нейронів* (як біологічних об'єктів з їх фізичними, хімічними та біологічними процесами та властивостями).

2. *Використання штучних нейронів в інформаційних та керуючих системах штучного інтелекту.*

Ми будемо розглядати другий напрямок.

Штучні нейронні мережі моделюють роботу біологічних нейронів і демонструють велику кількість властивостей, притаманних лише мозку людини. Вони можуть розпізнавати (класифікувати) образи; розв'язувати задачі адаптивної оптимізації; навчатись на основі досвіду; можуть прогнозувати події; обирати суттєву інформацію при надлишкових даних; можуть англійський текст перетворити на англійську мову або виконати переклад на іншу мову, можуть розпізнати рукописний текст і т. д.

В лексиконі розробників та користувачів нейромереж з'явилися нові слова, які нагадують опис поведінки людини: «прийняти рішення», «давати поради», «навчати», «забувати», «реагувати», «узагальнювати», «самооргані-зовуватись».

Нейрони можуть використовуватись в будь-якій ділянці діяльності людини. але більш доцільно їх застосовувати там, де *не відомі принципи рішення* поставленої задачі, але накопичено достатньо прикладів їх розв'язання, або існують неповні та зашумлені дані. Штучні нейронні мережі можуть виконувати нудні, монотонні або небезпечні завдання. А все це складає великий клас задач.

У 1943 р. У. Маккалох і У. Піттс першими опублікували дослідження про роботу штучних нейронів з пороговим виходом «1\0» у вигляді електричної схеми. Мак-Каллок і Піттс довели, що обчислення будь-якої функції за Тюрінгом можна реалізувати за допомогою спеціально організованої мережі штучних нейронів.

В 1949 р. канадський фізіолог Д. Хебб запропонував перше формальне правило навчання нейрону, за яким вага входу нейрону $w_j^{СТАРА}$ зміню-

ється пропорційно добутку вхідного x_j та вихідного u сигналів $w_j^{НОВА} = w_j^{СТАРА} + \eta u x_j$, де $\eta = 0...1$ – коефіцієнт навчання.

В 1957 р. нейробіолог США Ф. Розенблатт виконав моделювання *персептрон*. Персептрон – це нейронна мережа (НМ), яка використовується для розв’язання складних задач і вміщує *однин або кілька шарів нейронів*. В кожному шарі розміщується власна кількість нейронів, які обробляють сигнали входу власного шару і передають ці оброблені сигнали на наступний шар (від вхідного шару до вихідного).

Персептрони мають прямі зв’язки та випадково застосовані з’єднання між шарами. У 1960 р. *персептрон* був використаний для розпізнавання літер алфавіту.

Нейрони звичайно розглядаються як нейронні мережі (НМ) у вигляді з’єднаних між собою шарів з нейронів, причому можливості НМ зростають із збільшенням в них шарів, кількості нейронів та зв’язків між ними.

У 1959 р. Б. Відров та М. Хофф запропонували найбільш поширений наразі *нейрон ADALINE (ADaptive Llinear NEuron – Адаптивні Лінійні Нейрони)* та мережі з ADALINE у вигляді *MADALINE (Multiple ADaptive Llinear NEurons – Множинні Адаптивні Лінійні Нейрони)*, який має вхідні сигнали та різні види вихідних активаційних функцій.

В 1965–1980 рр. в СРСР народжується новий напрямок – ситуаційне управління (проф. Поспелов Д. А.) і були розроблені моделі представлення знань.

У 1969 р. М. Мінські та С. Пейперт вказали недоліки одношарового персептрон, що викликало обмеженість фінансування досліджень НМ, яке продовжувалось до 80-х років.

У 1974 р. П. Д. Вербос та А. И. Галушкин незалежно знайшли алгоритм зворотного розповсюдження похибки, який раніше був відкритий *Брайсоном та Хо* у 1969 р. [22; 203].

У 1982 р. американський біофізик Д. Хопфілд представив свою НМ Хопфілда, в якій створив докорінно нові підходи до моделювання [161]. В результаті інтерес до НМ був відновлений.

У 1982–2003 р.р. Т. Кохонен запропонував НМ по розділу універсуму образів на окремі кластери і отримав «мапу Кохонена» по розміщенню кластерів з використанням принципу «переможець отримує все» [35; 44].

У 1983 р. Я. А. Андерсон за рахунок збільшення зворотних зв’язків нейронів поліпшив НМ Хопфілда: збільшив кількість еталонних векторів, яку запам’ятовує НМ [130].

У 1987 р. Б. Коско запропонував НМ двоспрямованої асоціативної пам’яті, яка складається з двох взаємно пов’язаних між собою шарів, *кожний з яких є і вхідним і вихідним* [172].

У 1987 р. Р. П. Липман запропонував схему трьохшарової НМ Хеммінга зі збільшеною у порівнянні з НМ Хопфілда кількістю еталонних векторів, що запам’ятовується [175].

У 2007 р. М. В. Турти (Україна, Миколаїв) на основі розробленої нової теорії однозначних нечітких систем [115] запропонувала:

– *нейрон «Троянда»*, який може замінити кілька нейронів ADALINE [55; 56; 57; 58]. Це дозволило спростити задачу розпізнавання складних образів, а також дало можливість розв'язати *одним нейроном «Троянда» логічну функцію XOR* (що М. Мінські вважав *теоретично неможливим*) [59];

– *багат шарову НМ нового покоління* – з навчанням лише одного останнього шару за методом найменших квадратів, без використання метода зворотного розповсюдження похибок, при відсутності традиційного процесу дефазифікації і при підвищенні швидкодії інтелектуальної системи. Далі ця робота продовжувалась у напрямку удосконалення [62]. Внаслідок лінійності характеристик, НМ у принципі *не може попасти у локальний мінімум* і тому завжди отримує оптимальний розв'язок;

– *новий метод навчання* нечітких багат шарових нейронних мереж – за методом найменших квадратів *без використання метода зворотного розповсюдження похибок* (back propagation), що значно спростило процес дефазифікації нечітких систем [115].

У 2016 р. в Україні (Миколаїв) створена *«самонавчальна нейронна мережа з двомірною мапою n-вимірних об'єктів»* для розв'язання детермінованих і стохастичних задач [78; 79], яка у порівнянні «самонавчальною мережею векторного квантування Кохонена» спрощує навчання, автоматично визначає кількість вихідних кластерів і не псує інформацію за даними «міри близькості по латеральним зв'язкам» та по урахуванню «сусідства». Підґрунтям цієї НМ стала розроблена в Україні аналітична геометрія в n-вимірних тілесних кутах [54].

Об'єднання нейронних мереж, генетичних алгоритмів та нечітких систем дозволяють об'єднати типові переваги цих трьох напрямків досліджень [104]: розглянути процеси, які ще не мають чіткого осмислення; обрати рішення, які є «найбільш пристосовані» для розв'язання проблеми; описати складні процеси з ослабленими вимогами до точності даних.

7.2. Недоліки та переваги нейронних мереж

НМ вигідно застосувати, якщо отримано великий обсяг вхідної інформації, але: відсутній алгоритм; не відомі принципи вирішення завдань; дані є неповними, або надлишковими чи частково суперечливими [125].

Наразі НМ розв'язують наступні класи задач.

Класифікація образів: вхідний образ має вигляд вектора ознак, для якого розпізнається відповідний еталонний вектор.

Кластеризація (категоризація), при якій близькі образи розміщуються в один кластер (одну групу).

Апроксимація функцій, при якій по даних вхід-вихід для невідомої функції $F(x)$ знаходиться її оцінка.

Прогнозування, по якому передбачається деяке значення у деякий майбутній момент часу. Фахівці інвестиційного банку за допомогою програмного нейропакета роблять короткострокові прогнози коливань курсів валют [17].

Оптимізація, завданням якої є знаходження такого рішення, яке задовільняє обмеження системи і максимізує чи мінімізує цільову функцію. *Задача комівояжера* є класичним прикладом задачі оптимізації.

Відомо, що у моделі обчислень фон Неймана при наявності помилки в адресі отримується помилкова інформація. НМ дозволяє виправити цей недолік за рахунок використання в особливо надійних інформаційних баз даних *асоціативних адрес*, в яких помилкові адреси асоціативно виправляються на точні адреси.

Проведені теоретичні та практичні дослідження, по *створенню нейронних мереж для вирішення завдань лінійної, квадратичної, нелінійної, комбінаторної оптимізації* [108; 113]. Для вирішення складних неформалізованих комбінаторних задач оптимізації теж застосовуються *штучні НМ* [28; 31; 155].

Але не треба занадто захоплюватись можливостями нейронних мереж, бо реальний «інтелект» найскладнішої нейронної мережі не перевищує рівня дощового черв'яка, а традиційні комп'ютери та обчислювальні методи є ідеальними для багатьох застосувань. Звичайно, НМ є *спеціалізованою*, тобто пристосованою для розв'язання вузької задачі: виконати її універсальною поки що не дозволяють апаратні ресурси. В цьому плані великі надії покладають на нанотехнології: передбачають, що нанотехнології нададуть можливість усі знання людини ввести в один комп'ютер.

Основні недоліки НМ:

1. *НМ можуть «помилятись»*, тобто не мають достатнього рівня надійності. Наприклад, не треба нейронні мережі використовувати для нарахування заробітної плати, або при введенні у дію бойових балістичних ракет (у таких випадках кажуть про недостатню надійність нейронних мереж).

2. *Не можна гарантувати*, що розроблена мережа є оптимальною мережею.

3. У загальному випадку невідома кількість потрібних внутрішніх шарів та кількість нейронів в цих шарах.

4. Невідома *кількість еталонних пар «вхід-вихід»* та кількість кроків, які потрібні для навчання.

5. Основні успіхи нейронних мереж отримані як результат використання алгоритму зворотного розповсюдження похибок, який має ряд проблем: немає гарантії, що мережа є оптимальною і може бути

навчена *за виправданий час*; немає впевненості, що мережа *не попаде у локальний мінімум* і тому може отримати гірше навчання, у порівнянні з оптимумом.

6. Навчена НМ не може відповісти на питання: «А як вона працює?». Хоча і існують методи, які розв'язують проблему подолання логічної непрозорості НМ (вони використовуються в експертних системах), але за це приходиться «платити»: усі надлишкові зв'язки вилучаються й залишаються лише найбільш важливі, без яких задача не може бути розв'язана.

7. Не доведено, що можна виділити окремі образи з навчальної еталонної множини.

8. Не доведено, що персептронний алгоритм навчання є більш швидким у порівнянні з перебором усіх можливих значень ваг; в деяких випадках цей примітивний захід може виявитись ліпшим.

Будь-яку задачу, яку розв'язує нейрокомп'ютер, можна розв'язати за допомогою звичайної спеціалізованої ЕОМ, яка іноді розв'яже її навіть ліпше. *Тому застосування нейромереж треба обґрунтовувати* [1–4].

Нейронні мережі мають такі переваги:

1. НМ можуть керувати процесом, який *ще не має математичного опису або іншого обґрунтування*. НМ застосовують там, де тяжко скласти явний алгоритм дій для класифікації образу: читання рукописних текстів, перекладу з однієї мови на іншу, узнавання людини і т. п.

2. НМ *адаптується* до зовнішніх умов керування процесом і постійно переналагоджується під умови зовнішнього середовища, які змінюються. Якщо спеціалізована машина розв'яже одну стандартну задачу, то нейрокомп'ютер – одну, потім – другу, третю і т. п., бо він *адаптується* до зовнішніх умов, навчається й переналагоджується у процесі роботи згідно введеної програми. Тому не потрібно кожного разу проектувати нову спеціалізовану схему керування.

4. Досягається економія часу за рахунок гнучкості переналагодження та використання паралельних процесів у ЕОМ.

5. НМ мають здатність до *узагальнення, навчання та самонавчання у реальному часі*. Штучна нейронна мережа працює автоматично за рахунок власній структурі, а не внаслідок використання інтелекту людини у формі комп'ютерних програм.

6. НМ *толерантні до помилок* за рахунок розподілу інформації у зв'язках та вагах нейронних мереж. Стан окремого нейрону визначається станом багатьох інших нейронів, пов'язаних з ним. Тому втрата одного або кількох зв'язків, або перекручення інформації і відмова окремих елементів не впливає кардинально на результат роботи у цілому (хоча якість виконання роботи і погіршується).

7. Досягається економія часу за рахунок виконання *паралельних обчислень* [120].

7.3. Математична нерівність вирішальної функції – основа роботи штучного нейрону

Відомо, що будь-яка *математична нерівність виконує логічну функцію*. Цю особливість нерівності можна використати для розділу на «худих» та «товстих» множини людей з характеристиками (x_1, x_2) , де $x_1=0..150$ кг – вага людини, а $x_2=0..2,0$ м – її зріст (рис. 7.3.1, а). Координати найбільш «худой» та найбільш «товстої» людини помічені чорними точками.

Тоді всю множину людей, що розміщена у прямокутнику при $x_1=0..150$ кг, $x_2=0..2$ м можна розділити на дві частки («худих» та «товстих») за допомогою прямої лінії (*вхідної вирішальної функції нейрону*), яка знаходиться між вказаними чорними точками у вигляді нерівності

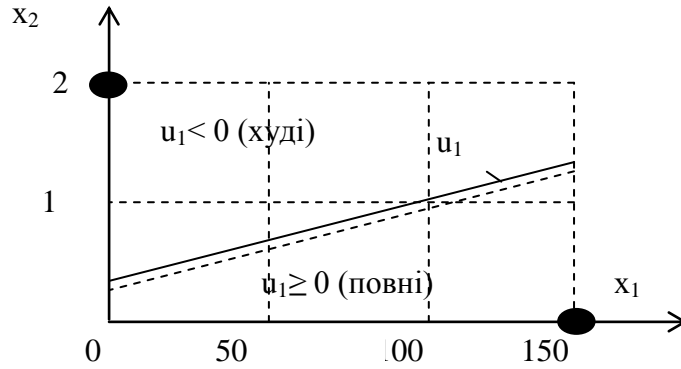
$$u_1 = x_1 - 150x_2 + 45 \geq 0. \quad (7.3.1)$$

Нерівність (7.3.1) розділяє площу ознак (x_1, x_2) на дві на півплощини: в одній півплощині нерівність виконується, а в іншій – ні. У цьому можна пересвідчитись, якщо підставити в нерівність (7.3.1) координати будь-якої довільно обраної точки. На практиці для перевірки нерівності найчастіше використовують точку $(x_1=0, x_2=0)$. В цій точці ми отримуємо відповідь на питання: «Чи виконується нерівність u_1 в півплощині, в якій знаходиться точка $(x_1=0, x_2=0)$?». У нашому випадку ці координати початку осей координат дають значення $u_1 = 45 \geq 0$, тобто нерівність виконується на тій півплощині. Ця півплощина на прямій u_1 помічається пунктирною лінією. На іншій півплощині нерівність не виконується для всіх її точок.

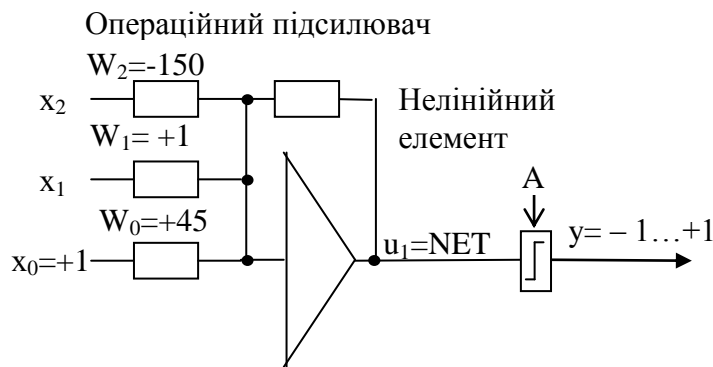
Таким чином, *вхідна функція нейрону* є вирішальною функцією і виконує логічну умову: якщо $u_1 \geq 0$ (нейрон видає сигнал «1=Істина»), то це – людина «повна», якщо ні, то «худя» (нейрон видає сигнал «0=Хибність»).

З рис. 7.3.1, а видно, що *розміщення вирішальної функції (7.3.1) є невизначеним*: для розділу класів можна використати безліч подібних вирішальних функцій у залежності від індивідуального для даного експерта тлумачення поняття «повноти».

Відповідний нейрон Adaline згідно вирішальної функції (7.3.1) наведений на рис. 7.3.1, б. Нелінійний вихід нейрону рис. 7.3.1, б призначений для обмеження числових величин NET значеннями $y = -1...+1$, які виконують логічний висновок нейрона і одночасно запобігають отриманню надто великих вихідних цифр, які можуть призвести до аварійної зупинки ЕОМ по переповненню.



а)



б)

Рис. 7.3.1. Розділ людей на два класи по вазі x_1 та зросту x_2 :
а) графічна залежність; б) відповідний нейрон Adaline

Таким чином, для об'єктів з кількісними ознаками за допомогою математичних нерівностей ми можемо створювати логічні вирази для їх класифікації.

Цю задачу можна розв'язати двома шляхами:

1. *За рахунок створення програми* на будь-якій мові (найчастіше так й роблять). Тоді можна казати про *програмну реалізацію* нейронної мережі.

2. *За рахунок створення спеціального пристрою*, якій зветься штучним нейроном (наприклад, за рахунок використання операційного підсилювача та нелінійного перетворювача виходу – рис. 7.3.1, б). Тоді можна казати про *апаратну реалізацію* нейронної мережі, маючи на увазі що мережа складається з окремих пристроїв – штучних нейронів.

Операційний підсилювач перемножує вхідні постійні величини та змінні сигнали x_j на коефіцієнти передачі W_j (вагові коефіцієнти) і потім отримує їх підсумок.

Нас у даному випадку цікавить лише знак функції u , а не її величина. Тому додатні значення перетворюють на «+1» (маленький зріст), а від'ємні значення – на «-1» або «0» (високий зріст). Це змушує мати на виході операційного підсилювача нелінійний елемент, який дає

логічні відповіді (ІСТИНА/TRUE/+1) та (ХИБНІСТЬ/FALSE/-1, «0»). Практично ми перетворюємо математичну нерівність в логічну умову.

Функція u іноді зветься *NET* (у матричній формі це має вигляд рівняння $U = NET = W \cdot X$), а вихідна нелінійна функція $y = f(NET)$ – зветься *OUT*. Замість вихідних сигналів « $y = +1/-1$ » можуть застосовуватись сигнали « $y = +1/0$ ».

Функція перетворення значення *NET* у вихідну функцію $y = f(OUT)$ зветься *активаційною (вихідною) функцією нейрона*.

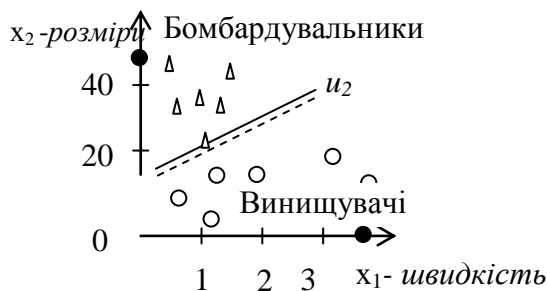
Розглянемо іншу задачу: розділ літаків на бомбардувальники та винищувачі по значеннях кількісних ознак (x_1, x_2), як показано на рис. 7.3.2, а (тут $x_1 = 0 \dots 3$ – швидкість (М); $x_2 = 0 \dots 40$ – розміри (м)).

Ми можемо провести на рис. 7.3.2,а пряму лінію

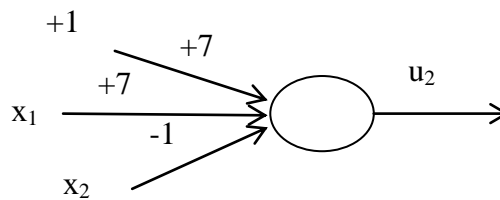
$$u_2 = 7x_1 - x_2 + 7 \geq 0, \quad (7.3.2)$$

яка є границею для двох класів об'єктів і знаходиться між граничними значеннями характеристик бомбардувальників та винищувачів.

Нерівність $u_2 \geq 0$ використовується для розділу області ознак (x_1, x_2) на дві півплощини: в одній півплощині з точкою ($x_1 = 0, x_2 = 0$) нерівність (7.3.2) дотримується (визначаються винищувачі при $u_2 \geq 0$), а в іншій – ні (визначаються бомбардувальники при $u_2 < 0$).



а)



б)

Рис. 7.3.2. До класифікації літаків:

- а) розміщення експериментальних даних навколо нерівності $u_2 = 7x_1 - x_2 + 7 \geq 0$; б) відповідний нейрон

Знак нерівності («більше або дорівнює» \geq , або «менше або дорівнює» \leq) не має значення для аналізу і довільно обирається експертом (звичайно використовується « \geq »). На рис. 7.3.2 нерівність (7.3.2) виконується на півплощині, яка помічена пунктирною лінією; якщо ми змінимо знак нерівності, то пунктирна лінія переміститься на іншу півплощину, але від цього принцип класифікації не змінюється. Функція u_2 – це лінія у просторі ознак, яка розділяє між собою об'єкти двох класів. Очевидно, що таких функцій можна провести багато і вони можуть мати різний вигляд.

Звичайно нейрон Adaline позначається у спрощеному вигляді (рис. 7.3.2, б). Якщо відомі вхідні функції, то це означає відомість вагових

коефіцієнтів; тоді для класифікації об'єктів *можна не застосовувати процес навчання*.

Функція активації $y=F(\text{NET})=F(u)=\text{OUT}$ може мати різний заданий програмістом вигляд. На рис. 7.3.3 вони показані у порядку, який приблизно відображує їх уживаність. Усі вони можуть бути зсунуті вздовж осі u .

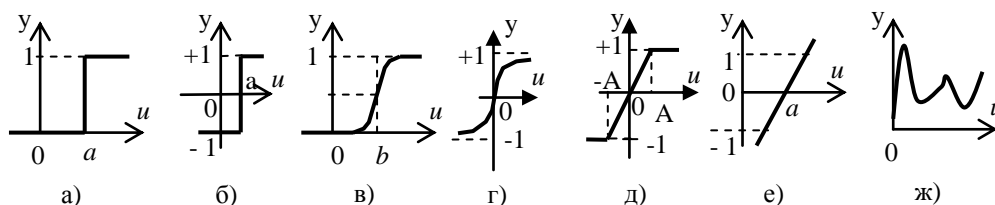


Рис. 7.3.3. Функції активації штучного нейрону

Іноді після активаційної функції на виході нейрона використовується додаткова вихідна функція $f_{\text{вих}}(y)$, яка призначена для перетворення та уточнення значення виходу нейрону.

Найбільш розповсюдженими функціями активації є: уніполярна стрибкоподібна функція (рис. 7.3.3, а); біполярна стрибкоподібна функція (рис. 7.3.3, б); сігмоїда $y = 1/[1 + e^{-a(u-b)}]$, де a, b – постійні коефіцієнти (рис. 7.3.3, в); тангенс гіперболічний (рис. 7.3.3, г); складена з відрізків прямих ліній (рис. 7.3.3, д); лінійна залежність (рис. 7.3.3, е); довільна задана програмістом функція (рис. 7.3.3, ж). Тобто активаційна функція *здається користувачем* і може мати будь-який вигляд, наприклад, при бажанні вона може мати вигляд або прямої лінії, або складатись з 10-ти відрізків прямих ліній, або мати задану користувачем будь-яку нелінійну залежність. Вигляд активаційної функції обирається користувачем у залежності від вимог конкретної задачі.

Звичайно для складної НМ, яка розв'язує конкретну задачу, *всі нейрони мають однакову активаційну функцію*.

Головне призначення активаційної функції – обмеження максимального сигналу виходу *значенням модуля «1»*: це обмежує розрахункові числові значення НМ і не дозволяє їм наблизитись до неприпустимо великих величин, які викликають зупинку роботи ЕОМ. Слід сказати, що такої переваги не має активаційна функція рис. 7.3.3, е, яка є лінійною функцією $\text{OUT} = Ku$, де K – константа.

7.4. Функції моделей нейронів

У 1943 р. У. Маккалох і У. Піттс першими опублікували дослідження про роботу штучних нейронів з пороговим виходом «1\0» у вигляді електричної схеми рис. 7.4.1, а.

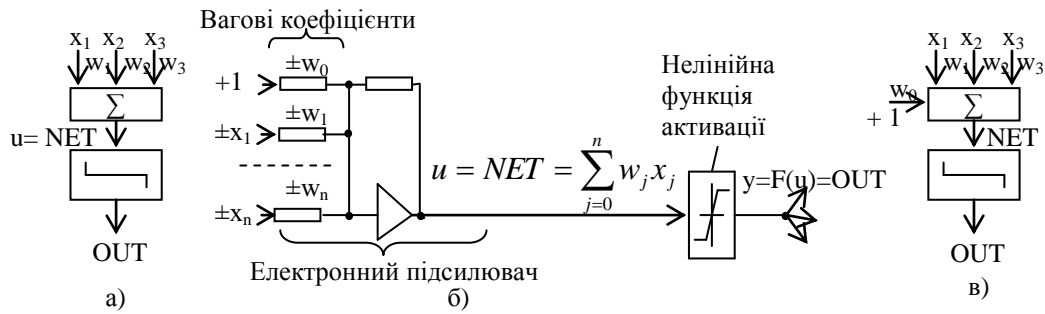


Рис. 7.4.1. Схеми штучного нейрону:

- а) штучний нейрон У. Маккалоха і У. Піттса (1943);
- б) штучний нейрон *ADALINE* Б. Відрова та М. Хоффа (1959);
- в) перцептрон Розенблата (1960)

В 1959 р. Б. Відров та М. Хофф розробили моделі *ADALINE* (*ADaptive LInear NEuron* – Адаптивні Лінійні Нейрони) та мережі з *ADALINE*. Нейрон *ADALINE* показаний на рис. 7.4.1, б. Як бачимо, на вхід нейрону була додана постійна складова. Зараз *ADALINE* є найбільш поширеним елементом нейронних мереж.

Нейрон ADALINE має наступні особливості:

- вхідні сигнали x_j множаться на вагові коефіцієнти w_j , підсумовуються і підсилюється по потужності електронним підсилювачем, в результаті чого на його виході отримуємо значення **вирішальної функції** $u = NET$, яке, у залежності від вхідних координат x_j , може мати додатне або від’ємне значення

$$u = \sum_{j=0}^n w_j x_j = NET \geq 0; \tag{7.4.1}$$

- значення $u = NET$ проходить через нелінійну *функцію активації* у вигляді

$$y = F(u) = F(NET) = OUT = \begin{cases} 1 & \text{– при } NET \geq 0, \\ 0 & \text{– при } NET < 0. \end{cases}$$

і подаються на входи інших нейронів у вигляді логічних сигналів OUT, рівних 1 або 0.

Сигнал *NET* може бути розрахованим у матричній формі

$$NET[1,1] = X^T[1,n+1] W[n+1,1],$$

де $X(x_0=1=const, x_1, \dots, x_n)$ – вектор вхідних сигналів (значення $x_0=1=const$ вектора X задається програмістом); $W(w_0, w_1, \dots, w_n)$ – вектор вагових коефіцієнтів.

Більшість мереж використовують для знаходження зваженої суми входів нейрона формулу (7.4.1), але в деяких випадках, якщо цього вимагає поставлена задача, можуть використовуватись і інші формули, наприклад,

$$u = \sum_{j=0}^n w_j x_j^2, \quad \sum_{j=0}^n w_j (a_1 x_j + a_2 x_j^2).$$

Введення подібних нелінійностей може збільшити обчислювальну потужність НМ, тобто дозволяє розв'язати поставлену проблему з меншою кількістю нейронів.

Математично доведено, що будь-яка нерівність, в тому числі і у вигляді (7.4.1), виконує логічні функції: якщо отримана сума «NET = u» більше заданого порогового значення, то вихід дорівнює 1, інакше – 0.

Структура штучного нейрону наведена на рис. 7.4.2. Наразі існують різні моделі нейронів. Нейрони можуть реалізуватись апаратно та програмно. Найбільше поширення отримала програмна реалізація НМ – з використанням ЕОМ. Тому для вирішальної функції $u = NET \geq 0$, для функцій активації $z = f_a(u)$ чи виходу $Y = f_{вих}(z) = OUT$ можуть бути використані довільні **функції будь-якої складності**. Обмеження тут накладаються лише з точки зору зменшення складності аналізу та навчання. При цьому кількість активаційних та вихідних функцій не обмежується.

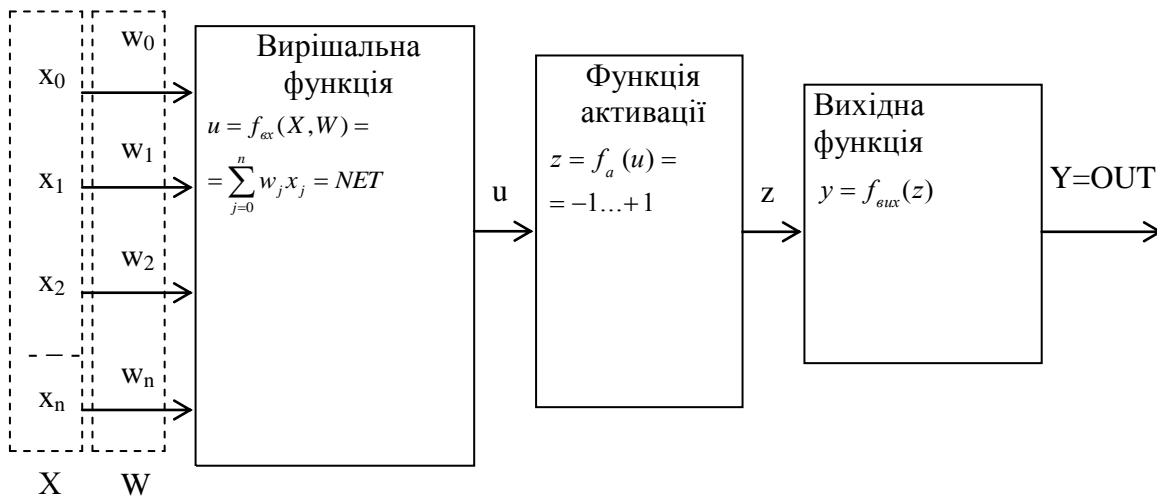


Рис. 7.4.2. Структура штучного нейрона

Наприклад, існують моделі нейронів:

- з введенням конденсатора та джерела струму у входи електронного підсилювача, що дозволяє аналізувати динамічні процеси [161];
- з добутком вхідних змінних при визначенні нерівності $NET \geq 0$;
- з використанням нелінійної залежності при визначенні нерівності $NET \geq 0$ (в тому числі, функції еліпса);
- з урахуванням взаємного відношення підсумку збуджуючих вхідних сигналів до підсумку гальмуючих вхідних сигналів тощо.

Слід нагадати, що розміщення вирішальних функцій у просторі ознак є невизначеним: для розділу класів можна використати безліч вирішальних функцій. Найліпшим вважається її розміщення при рівній відстані від об'єктів різних класів.

Прості функції нейронів є базисом створення потужних мереж.

В 1957р. нейробіолог США Ф. Розенблатт виконав моделювання перцептрону – багатоварової НМ з ваговими коефіцієнтами прямих

зв'язків між шарами. Кожний шар має довільну кількість нейронів. У 1960 р. перцептрон був реалізований на ЕОМ «Марк-1» для розпізнавання літер алфавіту.

Перцептрони мають наступні особливості:

1. Перцептрони складаються з одного або кількох шарів штучних нейронів з прямим розповсюдженням дискретних сигналів. В кожному шарі розміщується довільна кількість нейронів. Зі збільшенням кількості нейронів НМ може розв'язати більш складні задачі.

2. Входи та виходи нейронів мають значення або «1» або «0».

3. Перцептрони не мають зворотних зв'язків.

4. Перцептрони можуть виконати класифікацію лише для лінійно розділювальної сукупності векторів (якщо пряма лінія або площина може розділити введені вектори на класи, то вважається, що вектори є лінійно розділювальними).

Наразі НМ розглядаються як з'єднані між собою шари, кожен з яких вміщує деяку кількість нейронів, які обмінюються інформацією з використанням прямих та зворотних зв'язків (рис. 7.4.3).

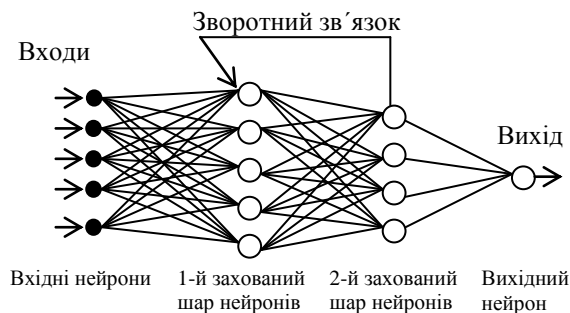


Рис. 7.4.3. Нейронна мережа для визначення технічного стану літака

Теоретично може бути довільною кількість прошарків НМ та кількість нейронів в кожному прошарку. Чим складніша НМ, тим складніші проблеми вона може розв'язувати. У більшості випадків оптимальний варіант отримується на базі вивчених властивостей існуючих мереж та інтуїтивного підбору.

Наразі розроблено багато нейронів з власними схемами заміщення та навчання. Наприклад, модель японського вченого Фукушімі має роздільні входи для збуджувальних (зі знаком «+») та гальмуючих (зі знаком «-») сигналів; нейрон Хопфілда має на вході операційного підсилювача конденсатор і тому має здатність моделювати процеси у часі; Σ - Π – нейрон (читається «сігма – пі» нейрони) застосовує добуток вхідних сигналів, або їх кореляцію; в стохастичній моделі нейрона вихідний сигнал приймає значення « $y = \pm 1$ » з деякою ймовірністю, яка залежить від NET; застосовуються нейрони з нелінійним NET (у вигляді багаточлена, квадратичної залежності, еліпса тощо) [103]. Існує також багато біологічних моделей нейронів.

Нейрон може бути реалізованим *апаратно* (аналогічно розв'язку за методом вирішальних функцій) та *програмно* – на будь-якій алгоритмічній мові.

Найчастіше використовується *програмне виконання нейронних мереж*. Хоча нейронна мережа зовні й виглядає як звичайна програма, насправді вона у принципі відрізняється від програмного забезпечення, бо більшість реалізацій на основі нейронних мереж навчається у процесі своєї роботи, а не складає однозначний фіксований алгоритм роботи.

Звичайно НМ є *спеціалізованою*, тобто пристосованою для розв'язання вузької задачі: виконувати її універсальною поки що не дозволяють апаратні ресурси. В цьому плані великі надії покладають на нанотехнології: передбачають, що засвоєння нанотехнологій надасть можливість усі знання мозку людини вводити в комп'ютер. НМ створюються у вигляді програмних продуктів, апаратно, а також у вигляді комбінації програм та апаратів.

7.5. Нейронна мережа для керування судновою електроенергетичною системою

Припустимо, що нам потрібно розробити нейронну мережу для керування судновою електроенергетичною системою (СЕЕС).

СЕЕС складається з кількох генераторів (звичайно – з трьох, але їх кількість може доходити до 10 і більше), головного розподільного щита (рис. 7.5.1) та місцевих розподільних щитів, із яких отримують живлення численні споживачі електроенергії. Прикладами споживачів електроенергії є допоміжні окремі потужні двигуни (рефрижераторної установки, підіймального крану, бурової установки, компресора пускового повітря тощо); допоміжні електроприводи головного двигуна, який забезпечує рух судна, у вигляді охолоджуючих насосів прісної та з або забортної води, паливних насосів, насосів змащення тощо; освітлення; компресори провізійних камер та кондиціонування повітря; вентилятори трюмів; електрообладнання камбуза; палубні механізми.

Усі споживачі електроенергії у номінальному режимі звичайно отримують електричну енергію з постійною фазною напругою U_1 змінного струму та постійною частотою f_1 . Якість електроенергії (наявність вищих гармонік напруги та струму в мережі) при цьому повинна відповідати встановленим стандартним значенням.

Відхід від номінального (нормального) режиму СЕЕС ми можемо контролювати за допомогою датчиків навантаження генераторів, датчиків стабільності напруги та частоти та датчиків виходу якості електроенергії за визначену межу.

Для розробки нейронної мережі ми повинні спочатку визначити кількість вхідних сигналів, кількість класів (вихідних сигналів) та участь людини у визначенні передбачуваних подій.

Очевидно, що кількість нейронів у вхідному шарі дорівнює довжині вхідного вектора. Вхідний шар може отримувати, наприклад, такі дані у i -ій точці СЕЕС: U_1 – фазна напруга змінного струму; f_1 – частота СЕЕС; $U_{mj\gamma}$ – амплітуда γ -ої гармоніки фазної напруги j -ої фази; $\varphi_{j\gamma}$ – зсув по фазі γ -ої гармоніки фазної напруги j -ої фази; P_{ja} – активна потужність j -го генератора; P_{jr} – реактивна потужність j -го генератора; W_j – рівень шуму j -го генератора і т. д.

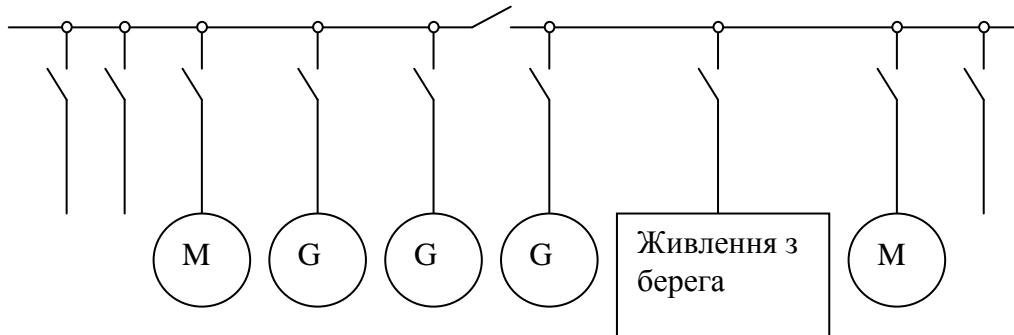


Рис. 7.5.1. Однолінійна схема головного розподільного щита суднової електроенергетичної мережі:
G – генератори; М – потужні двигуни

Кількість нейронів у вихідному шарі дорівнює кількості станів СЕЕС, яка контролюється.

Навчання НМ полягає у налагодженні зв'язків між шарами нейронної мережі СЕЕС таким чином, щоб вхідному векторові параметрів відповідав вірний вихідний вектор прогнозованого режиму. НМ повинна виконувати дії, які передбачаються в наведеній табл. 7.5.1.

Таблиця 7.5.1.

Планування дій у зножності від класу розпізнаного образу СЕЕС

Клас образу режиму СЕЕС	Дії
Клас 1. Номінальний режим.	Клас 1. Жодних захисних дій не виконують.
Клас 2. Аварія.	Клас 2. Аварія будь-якого елемента СЕЕС супроводжується його миттєвим вимкненням з метою обмеження завданої шкоди. У випадку тяжкої аварії у вигляді втрати напруги СЕЕС передбачається послідовне увімкнення в роботу дизель-генераторів та споживачів електроенергії.
Клас 3. Перевантаження генераторів СЕЕС.	Клас 3. Вимкнення другорядних споживачів електроенергії (системи кондиціонування повітря, вентиляторів трюмів, камбузного обладнання), увімкнення додаткових генераторів.
Клас 4. Недовантаження генераторів СЕЕС.	Клас 4. Вимкнення частки генераторів, якщо не передбачена можливість увімкнення потужних споживачів.
Клас 3. Перевантаження генераторів СЕЕС.	Клас 3. Вимкнення другорядних споживачів електроенергії (системи кондиціонування повітря, вентиляторів трюмів, камбузного обладнання), увімкнення додаткових генераторів.

Клас 4. Недовантаження генераторів СЕЕС.	Клас 4. Вимкнення частки генераторів, якщо не передбачена можливість увімкнення потужних споживачів.
Клас 5. Пуск потужного двигуна.	Клас 5. Якщо не вистачає потужності СЕЕС – увімкнення додаткових генераторів перед пуском такого двигуна.
Клас 7. Різка зміна величини навантаження генераторів СЕЕС (наприклад, зміна у часі навантаження у межах 30–100 % від номінального значення).	Клас 7. Передбачити неможливість вимкнення частки генераторів при недовантаженні.

7.6. Класифікація нейронних мереж

У НМ нейрони розташовані шарами і мають односпрямовані зв'язки між шарами. Чітка класифікація затрудняється тим, що часто НМ різного спрямування мають однакові ознаки (наприклад, стохастична НМ одночасно може бути НМ ієрархічної структури).

Розділ НМ по типам нейронів, типам мереж та ін. показаний на рис. 7.6.1.

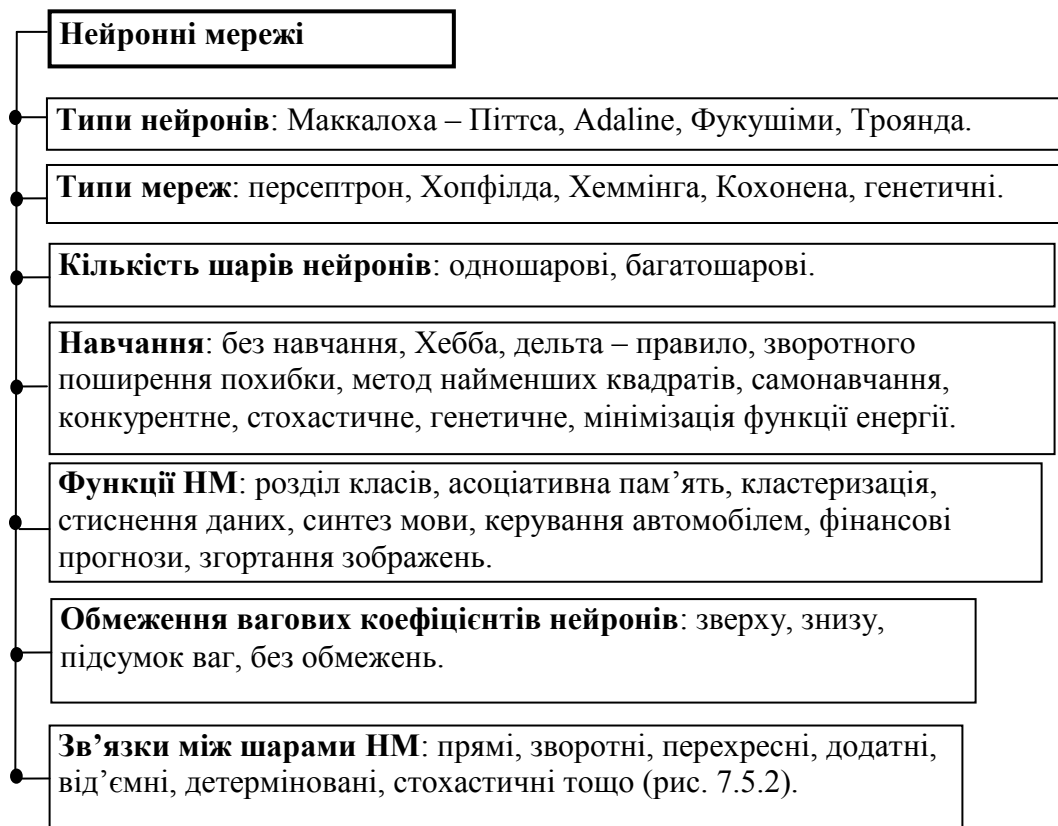


Рис. 7.6.1. Класифікація нейронних мереж

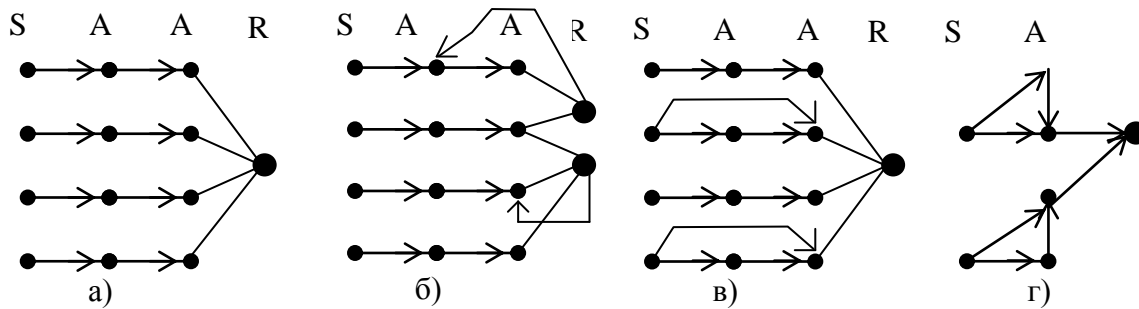


Рис. 7.6.2. Зв'язки між шарами нейронів: а) прямі; б) зворотні; в), г) перехресні

НМ по домінантним ознакам можна розділити на:

1. *НМ ієрархічної структури*: – одношаровий перцептрон; – багатошаровий перцептрон; – мережа радіальних базисних функцій.

2. *НМ з конкуруючими об'єктами та структурами*: – мережа Кохонена; – НМ зустрічного поширення; – мережа MAXNET. Вони працюють за принципом: «переможець забирає все».

3. *Рекурентні НМ із зворотними зв'язками*: – мережа Хопфілда; дво-направлена асоціативна НМ. *Рекурентні мережі є динамічними*, тому що в силу зворотних зв'язків у них модифікуються входи нейронів, що призводить до зміни стану мережі. Інші НМ є *статичними* в тому сенсі, що на вхід вони поставляють одну сукупність вихідних значень, що не залежать від попереднього стану мережі.

4. *Стохастичні НМ*.

5. *НМ адаптивного резонансу*.

6. *Генетичні НМ*.

7. *Класифікація по часу передачі сигналу*: активаційна функція залежить як від вагових коефіцієнтів входів нейронів у часі $w_{ij}(t)$ так і від часу передачі сигналу (імпульсу). Тобто сигнал виходу $y=f[w_{ij}(t), u_i^*(t-T_{ij})]$. При $T_{ij}=0$, $T_{ij}=\text{const}$ НМ зветься синхронною, а при фіксованому різному значенні T_{ij} – асинхронною.

8. *НМ згортання* запропонована Я. Лекуном, призначена для *розпізнавання зображення* і входить у склад НМ глибокого навчання (англ. *deep learning*). Структура мережі – односпрямована без зворотних зв'язків, багатошарова. Назву отримала за операцію згортання, за якою кожний фрагмент зображення множить на матрицю (ядро) згортання поелементно, а результат підсумовується і записується у позицію вихідного зображення. Шари згортання (англ. *convolution layers*) передуються з шарами субдискретизації (англ. *subsampling layers*, шари передвибірки). В НМ згортання використовується *матриця ваг* невеликого розміру 5×5 , яка кодує лише одну графічну ознаку (наприклад, лінію під деяким кутом) і яку при кодуванні «переміщують» по шару на невелику відстань – на один чи два нейрони (піксели). Наступний шар отримує сигнал від матриці ваг, показує

наявність вказаної лінії та її координати і формує тим самим *карту ознак*. У дійсності використовується *багато матриць ваг*, які кодують різні лінії та дуги під різними кутами і створюють багато *карт ознак* в одному шарі. Операція субдискретизації (англ. *subsampling*) зменшує розмірність *карт ознак* (з кількох сусідніх нейронів обирається і далі використовується один – найбільш характерний, осереднений), що прискорює наступні розрахунки.

Шари згортки та субдискретизації повторюються кілька разів, що дозволяє розпізнавати складні зображення. Якщо на першому шарі ядро згортання проходить по одному початковому зображенню, то на внутрішніх шарах ядра проходять паралельно по всіх картах ознак даного шару, а результат підсумовується, формуючі одну карту ознак наступного шару, яка відповідає даному ядру. Перевагою такої обробки є розпаралелювання розрахунків та сталість до повернення і зсуву зображення. Недоліком є велика кількість параметрів, які обираються емпірично (кількість ядер та їх розмір, крок переміщення, степе́нь зменшення розмірності тощо).

8. Адаптивний лінійний нейрон *Adaline*

8.1. Схема нейрона *Adaline*

Б. Відров та М. Хофф розробили нейрон *Adaline* (Adaptive Linear Neuron, 1959 р.), який добре зарекомендував себе на розпізнаванні числових векторів у n -мірному просторі ознак.

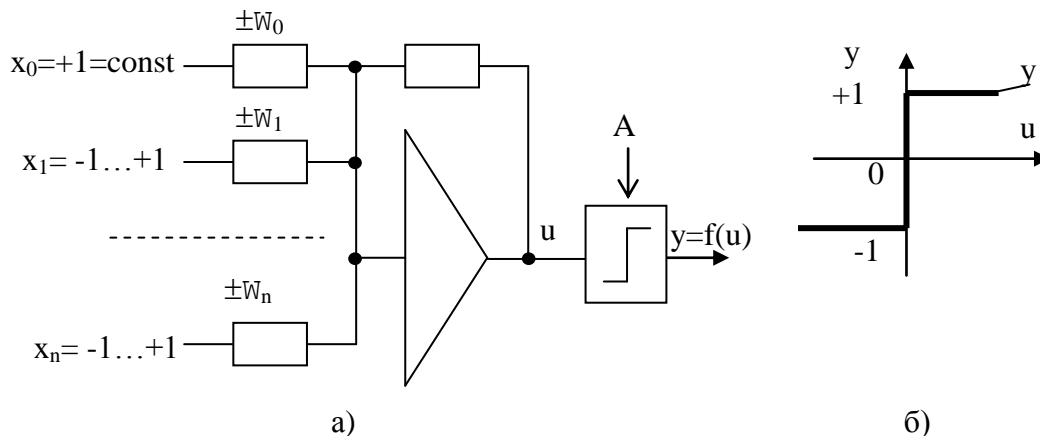


Рис. 8.1.1. Структура штучного нейрона *Adaline* для n входів

Нейрон *Adaline* складається з операційного підсилювача рис. 8.1.1 [204], у якій на один з входів подається постійний сигнал $x_0 = +1 \text{ В} = \text{const}$ при ваговому коефіцієнті « $\pm w_0$ », а на інші n входи подаються сигнали $x_1, x_2, \dots, x_j, \dots, x_n$, які можуть мати значення $\pm 1 \text{ В}$ (« $+1 \text{ В}$ » – наявність ознаки; « -1 В » – немає ознаки) з ваговими коефіцієнтами $w_1, w_2, \dots, w_j, \dots, w_n$. Вважаємо, що вихідна активаційна функція нейрону *Adaline* має вигляд рис. 8.1.1, б.

У загальному плані *Adaline* виконує сумування сигналів з врахуванням вагових коефіцієнтів (коефіцієнтів передачі)

$$u = \sum_{i=0}^n \pm w_i x_i = w_0 x_0 + \sum_{j=1}^n w_j x_j$$

а потім отримана у суматорі функція $u = \text{NET}$ подається на нелінійний елемент, який видає на виході сигнал (рис. 8.1.1, б)

$$y = \begin{cases} f(u) = +1, & \text{якщо } u > 0; \\ f(u) = -1, & \text{якщо } u \leq 0. \end{cases}$$

Таким чином, нейрон використовує вирішальну функцію для розділу простору ознак на дві частки: в одній з них нерівність виконується, а в іншій – ні. При цьому *розміщення вирішальних функцій є невизначеним*,

бо для розділу класів можна використати нескінченну множину вирішальних функцій.

N-Adaline (*Non linear Adaline*) є різновидом *Adaline*, на вхід якої крім звичайних вхідних сигналів x_j подаються їх нелінійні комбінації (у вигляді, наприклад, x_1^2 , x_2^2 , $x_1 x_2$, $x_1^2 x_2^2$, тощо), які множаться на відповідні вагові коефіцієнти та складаються. Цими нейронами реалізується розроблений О. Г. Івахненком (Україна) **метод групового урахування аргументів (МГУА)** з виходом, наприклад, у вигляді

$$u = NET = w_0 x_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 + w_5 x_1 x_2 + \dots$$

Madaline (*Many Adalines*) є багатошаровою мережею, кожний шар якої складається з нейронів *Adaline*.

8.2. Логічна функція AND

В табл. 8.2.1 наведені вимоги для входів та виходів логічних функцій AND, OR, XOR при наявності входів до нейронів у вигляді значень «±1», які мають виходи нейронів після активаційних функцій.

Таблиця 8.2.1.

Логічні функції нейрону Adaline для рівнів сигналів «±1»

x ₁	x ₂	y		
		y _{AND}	y _{OR}	y _{XOR}
-1	-1	-1	-1	-1
-1	+1	-1	+1	+1
+1	-1	-1	+1	+1
+1	+1	+1	+1	-1

Для логічної функції AND вихід нейрону Adaline може бути описаним нерівністю

$$u_1 = w_1 x_1 + w_2 x_2 + w_0 x_0 = x_1 + x_2 - 1,5 > 0, \quad (8.2.1)$$

де $w_1 = w_2 = 1$; $w_0 = -1,5$; $x_0 = 1$.

Схема нейрону, який відповідає нерівності (8.2.1), показана на рис. 8.2.1.

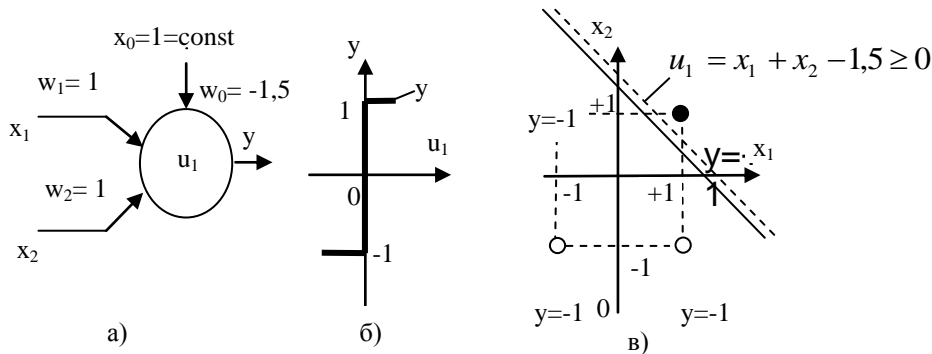


Рис. 8.2.1. Функція AND нейрона Adaline: а) схема; б) вихідна функція; в) геометричне пояснення

Нейрон з геометричної точки зору розділяє площу ознак x_1, x_2 на дві півплощини: в одній з них нерівність (8.2.1) виконується (вона позначена пунктирною лінією), а в іншій – ні. Вихід нейрона дорівнює « $y=+1$ » лише в одній точці – згідно табл. 8.2.1.

Разом з тим з рис. 8.2.1, в видно, що *розміщення вирішальної функції є невизначеним*: для розділу класів можна використати нескінченну кількість подібних вирішальних функцій.

8.3. Логічні функції OR і XOR

Для логічної функції OR нейрона Adaline вихід NET описується нерівністю

$$u_2 = w_1 x_1 + w_2 x_2 + w_0 x_0 = x_1 + x_2 + 1,5 \geq 0, \quad (8.3.1)$$

де $w_1 = w_2 = 1; w_0 = +1,5; x_0 = 1$.

Схема нейрону, який відповідає нерівності (8.3.1), показана на рис. 8.3.1.

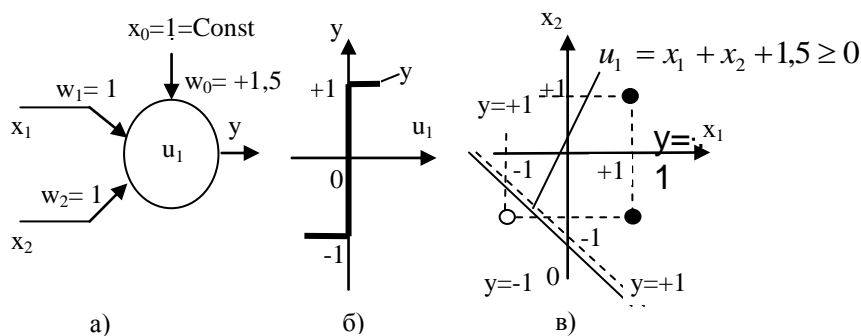


Рис. 8.3.1. Функція OR нейрону Adaline: схема (а); вихідна функція (б); геометричне пояснення (в)

Нейрон рис. 8.3.1 виконує логічну функцію OR табл. 8.2.1 і розділяє площу ознак x_1, x_2 на дві півплощини: в одній з них вихідна функція $y=+1$ (у цій півплощині нерівність (8.3.1) виконується, і вона позначена пунктирною лінією), а в іншій $y=-1$.

Для отримання логічної функції XOR (рис. 8.3.2), використані два нейрони Adaline, параметри яких відповідають логічним нерівностям

$$u_3 = x_1 - x_2 - 0,5 \geq 0; u_4 = -x_1 + x_2 - 0,5 \geq 0,$$

а для вирішального правила використана логічна функція OR згідно нерівності

$$u_5 = y_3 + y_4 + 1,5 \geq 0.$$

Очевидно, що замість двох розділювальних прямих ліній u_3, u_4 , показаних на рис. 8.3.2, б, можна провести *безліч інших прямих ліній*.

У процесі навчання у нейронній мережі намагаються змінювати вагові коефіцієнти таким чином, щоб мережа вірно класифікувала об'єкти. Але у цьому випадку ми чітко знали вимоги до системи і тому призначили потрібні значення вагових коефіцієнтів. Тобто навчання нейронів можливе шляхом призначення їх входам вагових коефіцієнтів, включаючи вільний член.

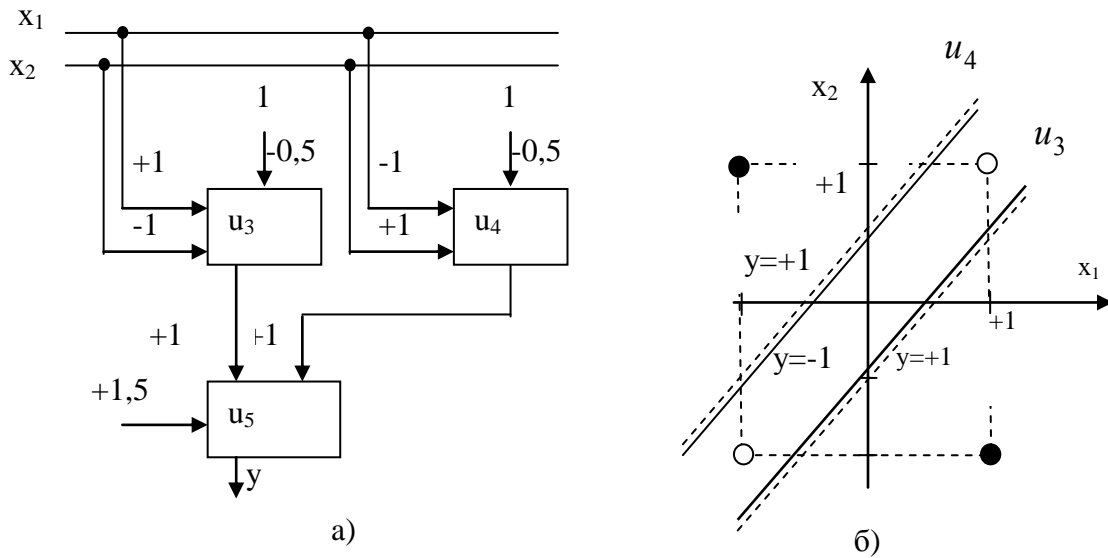


Рис. 8.3.2. Нейронна мережа Madaline на трьох Adaline для логічної функції XOR на площині ознак x_1, x_2

9. Навчання нейронних мереж

9.1. Загальні відомості про навчання нейронів

НМ складається із зв'язаних між собою шарів нейронів. Сукупність датчиків ознак образу надає на входи нейронів першого шару вектор вхідних ознак, які обробляються у проміжних («захованих») шарах НМ. Вихідний шар НМ видає сигнал класифікації по вхідному образу.

Ми розглянемо загальні напрямки навчання НМ, бо НМ охоплюють занадто великий обсяг людської діяльності з різних спеціальностей: переклад тексту, виділення статистичних закономірностей зміни вартості акцій на біржі, розпізнавання наявності у заданому регіоні покладів нафти по фотографіям з космосу, автоматичне керування складною системою, розпізнавання автора музики сумісно з датчиками реальних та абстрактних ознак об'єкта розпізнавання тощо. Кожна НМ, включаючи такі класичні, як мережа Хопфілда, Хеммінга, Кохонена тощо, має власні *особливості навчання*.

НМ розв'язує задачі, *по яких людина не може скласти алгоритму її розв'язку*, не за рахунок хитромудрості програміста, а на основі властивостей НМ, які імітують роботу нейронів людського мозку. За це розплачуємось: людина звичайно не здатна зрозуміти, яким чином НМ розв'язала задачу (хоча у цьому напрямку і існують програми, які частково дають на це основні пояснення).

Навчання *обов'язково* повинне супроводжуватись *зміною параметрів* НМ і охоплює:

1. *Зміну вагових коефіцієнтів входів нейронів*, їх вирішальних, активаційних та вихідних функцій (наприклад, вирішальна функція може мати вигляд прямої лінії, паралельних прямих ліній, параболи, еліпса, кола, гіперплощини тощо).

2. *Зміну структури нейронів* (кількості входів та виходів, активаційних та вихідних функцій, скорочення чи збільшення кількості блоків нейрона).

3. *Зміну структури (конфігурації) НМ* шляхом зміни кількості шарів та кількості нейронів у шарах; кількості прямих та зворотних зв'язків між шарами та нейронами.

4. *Способи навчання у вигляді*: призначення для всіх функцій нейрону математичної моделі та її коефіцієнтів; навчання з учителем та без учителя (самонавчання); навчання з підкріпленням; конкурентне навчання; стохастичне навчання; градієнтні методи навчання тощо.

Кожний з перелічених напрямків або їх сукупність і складають *сутність навчання*, яка впливає на спроможність НМ розв'язувати поставлені задачі із заданою точністю. НМ є вузько спеціалізованими, і ми розглянемо

лише основні найбільш розповсюджені напрямки навчання. Наприклад, навчання розпізнавання тексту відрізняється від навчання музиці.

При навчанні ураховують, що інтелектуальні можливості НМ зростають із:

- збільшенням кількості нейронів НМ;
- збільшенням кількості зв'язків між нейронами;
- введенням зворотних зв'язків (але одночасно виникає проблема динамічної сталості мережі, і тому більше за 80 % НМ не використовують зворотні зв'язки);
- ускладненням алгоритмів функціонування мережі (наприклад, введенням гальмуючих та збуджуючих зв'язків, введенням затримання сигналів у часі, ускладненням алгоритму передачі сигналів між прошарками нейронів, запам'ятовуванням сигналів прошарку з наступним їх використанням тощо);
- зміною активаційної функції, яка обробляє підсумовуючий сигнал нейрону;
- зміною операцій процесора, наприклад, за рахунок введення додаткових математично-логічних операцій, або навпаки – вилученням процесора.

Теоретично кількість шарів НМ та кількість нейронів в кожному шарі може бути довільною: чим складніша НМ, тим складніші проблеми вона може розв'язувати.

Звичайно, при навчанні спочатку намагаються навчати складну НМ з наступною спробою її спрощення зменшенням кількості шарів та кількості нейронів в шарах, щоразу перевіряючи точність роботи НМ.

При цьому можуть вилучити елементи, які дають мінімальну складову у розв'язок проблеми:

- вхід нейрона, який дає мінімальну складову у значення NET;
- вилучити нейрон після обчислення значущості його вкладу в аналіз;
- вилучити шар нейронів.

Після кожної подібної зміни, перевіряють НМ на точність роботи. Для оцінки можливості вилучення ваг та нейронів використовуються спеціальні вартісні функції [103; 179].

Широке розповсюдження у навчанні наразі отримало *призначення математичних операцій* (тобто призначення математичних моделей, вирішальних та активаційних функцій), які у сукупності виконує нейрон, та призначення або розрахунок зміни вагових коефіцієнтів у прийнятій математичній моделі нейрону. При навчанні змінюють (навчають) як вагові коефіцієнти окремих змінних, так і вагові коефіцієнти вільних членів нейронів.

Призначення вагових коефіцієнтів та вільних членів моделі нейрона відноситься у першу чергу до логічних функцій: їх входи та виходи відомі заздалегідь і тому у деяких випадках немає потреби поступово навчати ці нейрони.

Тому що виходи нейронів мають значення «-1/+1» чи «0/+1», то після першого захованого асоціативного шару можна умовно вважати, що значна кількість нейронів багат шарової НМ починає виконувати логічні функції (типу NOT, AND та OR, але для багатьох входів) з метою виділення класів.

По історії розвитку нейронів можна помітити, що саме за рахунок попереднього призначення характеристик нейронів, шарів і зв'язків у минулому були отримані нові перспективні НМ (наприклад, логічні функції, класичні НМ Хопфілда, Хеммінга, Кохонена тощо), бо ми починаємо вивчати ці НМ саме з вивчення попередньо призначених їм властивостей.

Тому одним з перспективних напрямків навчання та створення нових НМ є саме призначення окремим часткам НМ потрібних функцій, призначення входам нейронів відповідних вагових коефіцієнтів та призначення методів навчання. В теорії НМ таке призначення звичайно не розглядається як навчання (про подібні НМ звичайно кажуть, що це НМ «без навчання»). «Навчанням» НМ звичайно вважається поступове розрахункове уточнення введених програмістом випадкових невеликих величин вагових коефіцієнтів нейронів. Якщо ж в результаті достатньо складних попередніх розрахунків введені в модель нейрона коефіцієнти далі не змінюються, то таку НМ можуть вважати «НМ без навчання».

Але у більшості випадків нейрони навчають за відомими методами, які розглядаються нижче.

9.2. Призначення шарів та нелінійних виходів нейронів у персептроні

Один нейрон здатний виконати лише найпростіші процедури розпізнавання. Тому для розв'язання складних задач нейрони з'єднуються між собою і створюють багат шарову нейронну мережу прямого розповсюдження сигналів, яка за Розенблатом зветься *персептроном* (іноді персептроном зветься один нейрон). Кожний із нейронних шарів сприймає й переробляє інформацію і потім передає її наступному нейронному шару (*Розенблат розглядав три шари: сенсорний S, асоціативний A, реагуючий R* – рис. 9.2.1). Збільшення загальної кількості нейронів та шарів збільшує інтелект НМ. Кількість нейронів у кожному шарі (*S, A, R*) – довільна. Якщо не обумовлене протилежне, то кожний вихідний сигнал *i*-го шару подається на входи усіх нейронів наступного (*i+1*)-го шару.

Вирішальні функції нейронів НМ **повинні мати вільні члени:**

– для нейронів першого захованого асоціативного шару вільний член є необхідним, бо його зміна переміщує паралельно вирішальні функції у просторі ознак;

– кожна раніше розглянута логічна функція має вільний член. Тому що входи і виходи усіх нейронів наступних шарів звичайно мають значен-

ня ± 1 (що притаманно логічним функціям), то умовно можна вважати, що нейрони проміжних шарів і вихідного шару виконують деякі логічні функції по об'єднанню часток простору ознак у кластери та класи.

Тому при аналізі НМ до елементів *векторів сигналів входу всіх шарів НМ штучно додається нульовий елемент*, рівний «+1», який є необхідним для корегування ваг вільного члену і визначення $u = \text{NET}$, по якому розраховується вихід нейрону. Таким чином, *розмірність вхідного вектора шару НМ визначається кількістю ознак об'єкта плюс один нульовий елемент*. *Нульовий елемент* не ураховується лише у вихідному векторі НМ. Нульовий елемент штучно вводиться у вхідний вектор шару НМ для зручності розрахунків, і він *не збільшує розмірність простору ознак*, бо призначений лише для зміни положення вирішальної функції у цьому заданому просторі ознак.

Різні елементи персептрону мають свої функціональні призначення:

1. Нелінійна обробка активаційною функцією вихідного сигналу з *трьома ділянками вихідної нелінійної функції* забезпечують наступні особливості нейрона: нижчий рівень із малим коефіцієнтом підсилювання відсікає шуми в сигналах; середній рівень забезпечує високе значення коефіцієнта підсилювання слабкого сигналу; верхній рівень із малим коефіцієнтом підсилювання захищає мережу від занадто великого сигналу, який веде до «паралічу НМ».

2. *Кількість ознак n_S вектора $X = \{x_0, x_1, x_2, \dots, x_i, \dots, x_{n_S}\}$, $i = 0 \dots n_S$* в сенсорному S -шарі вказує кількість нейронів n_S в сенсорному шарі НМ та розмірність геометричної області, в якій розглядається образ. Значення « $x_0 = +1$ » не є ознакою розпізнаваного об'єкту, і у вектор вводиться штучно програмістом для ураховання вільного члена – деякої постійної величини. З геометричної точки зору при $n_S = 1$ образ – одномірний; при $n_S = 2$ образ можна розглядати у площині (x_1, x_2) ; при $n_S = 3$ образ є об'ємним в трьохвимірній системі координат (x_1, x_2, x_3) ; при $n_S > 3$ образ є гіпероб'ємним в системі координат $(x_1, x_2, \dots, x_{n_S})$.

3. *Кількість нейронів в першому асоціативному A_1 -шарі* вказує, на скільки частин ми розділяємо геометричну ділянку існування образів з метою поділення їх на класи.

4. *Кількість нейронів в наступних «захованих» шарах та кількість наступних шарів* визначається складністю логічного узагальнення попередньої інформації.

5. *Кількість нейронів в останньому вихідному шарі* вказує кількість класів, на яку розрахована НМ.

Наявність зворотних зв'язків між шарами нейронів свідчить про застосування елементів пам'яті про попередні стани.

Примітка. У випадку аналізу частки НМ з двох шарів A та B з кількістю нейронів A та B і вихідними векторами шарів $A_B[(A+1),1]$ та $B_B[(B+1),1]$ при використанні матриць потрібно звернути увагу на те, що:

1. До векторів $A_B[(A+1),1]$ та $B_B[(B+1),1]$ штучно додається один рядок для вільного члена у вигляді додаткового вхідного незмінного сигналу «+1»; саме тому тут для векторів штучно вказується кількість рядків $(A+1)$ та $(B+1)$. Якщо шар B є вихідним для НМ, то до вихідного вектора B нульовий елемент не додається.

2. Вагова матриця $W[B,(A+1)]$ між шарами A та B має кількість рядків B (по кількості нейронів та відповідних вирішальних функцій $u_B = NET[B,1]$ для B нейронів у наступному шарі). **Кількість колонок** у ваговій матриці $W[B,A+1]$ штучно збільшена до $(A+1)$, тому що до входів нейронів шару B приєднаний вхідний вектор $A_B[(A+1),1]$ з штучно доданим нульовим членом. Таким чином, по матричному рівнянню $u_B [B,1] = NET[B,1] = W[B,(A+1)] \cdot A_B[(A+1),1]$ з урахуванням активаційної функції ми визначаємо лише частку з B елементами вихідного вектора $B_B[(B+1),1]$, а далі штучно формуємо вихідний вектор у вигляді $B_B[(B+1),1]$ з додатковим нульовим елементом, який не змінюється і завжди дорівнює «+1».

3. Ця особливість ураховується також при зворотному переміщенні похибки з кінця у початок, при якому ми корегуємо міжшарову вагову матрицю $W[B,(A+1)]$. У матричному дельта-правилі ми повинні використовувати міжшарову вагову матрицю з урахуванням додатково введених вільних членів вирішальних функцій $W^{HUBE}[B,(A+1)] = W^{CTAPE}[B,(A+1)] + \eta \delta_B[B,1] A_B[1,(A+1)]$, де похибку $\delta_B[B,1]$ ми визначаємо лише по нейронах B (штучно доданий вільний член вихідного вектора $B_B[(B+1),1]$ не ураховується).

Якщо нехтувати інформацією щодо вільних членів та нелінійних перетворень виходів нейронів за допомогою активаційних функцій, то процеси в НМ рис. 9.2.1 можна описати такими матричними перетвореннями:

$$Z[n_R,1] = W_{RA}[n_R,n_A] * \{Y[n_A,1]\} = W_{RA}[n_R,n_A] * \{W_{AS}[n_A,n_S] * X[n_S,1]\} = \{W_{RA}[n_R,n_A] * W_{AS}[n_A,n_S]\} * X[n_S+1,1] = \{W_P[n_R, n_S]\} * X[n_S,1] \quad (9.2.1)$$

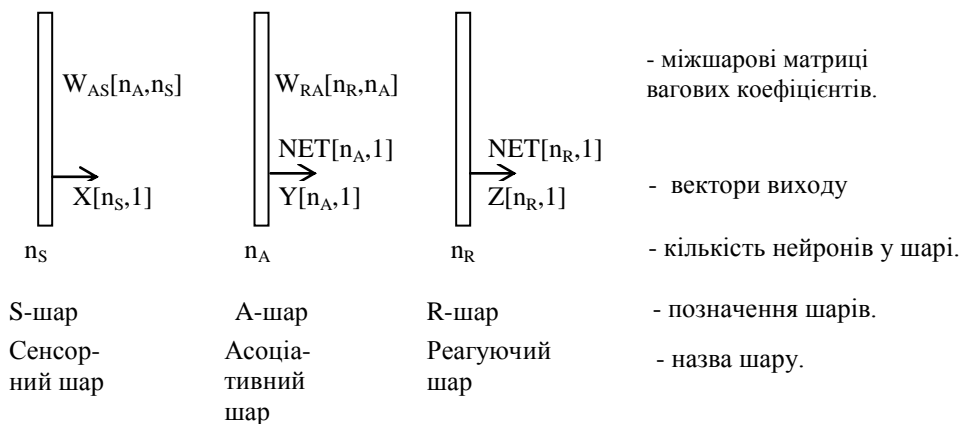


Рис. 9.2.1. Трьохшарова нейронна мережа

Тут для векторів та матриць у квадратних дужках вказано: [кількість рядків, кількість колонок].

З матричного рівняння (9.2.1) у фаховій літературі робиться наступний **невірний висновок**: нелінійна багатошарова НМ (з нелінійною активаційною функцією) має збільшені інтелектуальні можливості у порівнянні з аналогічною лінійною багатошаровою НМ без активаційної функції, бо за рівнянням (9.2.1) лінійну багатошарову НМ можна перетворити на еквіваленту одношарову НМ з меншою кількістю нейронів.

Наведені матричні перетворення (9.2.1) є точними, а **невірність висновку пояснюється двома не урахованими особливостями нейронів**:

1. Вказаний висновок невірний, тому що *один нелінійний нейрон має гірші інтелектуальні можливості у порівнянні з одним лінійним нейроном*. Будь-який нейрон (нелінійний чи лінійний) описує математичну нерівність у вигляді деякої вирішальної функції. При цьому нелінійний нейрон вказує значення цієї нерівності не у вигляді її розрахованої величини « \pm NET», а у вигляді повідомлень «+1/-1» (тобто передає лише знак « \pm NET» напрямку відступу від NET). Лінійний нейрон без вирішальної функції передає більший об'єм інформації, бо відображує і знак і число « \pm NET». Саме цією особливістю пояснюється той факт, що запропонований Турти М. В. в Україні у 2007 р. вітчизняний лінійний нейрон *Троянда* (без нелінійної активаційної функції) замінює три, десять, а теоретично нескінченну кількість нелінійних нейронів *ADALINE* з нелінійною активаційною функцією [с. 55–58, 115].

2. Наведене матричне рівняння є невірним з точки зору збереження і перетворення логічної інформації (хоча математичні перетворення є вірними). В основі логіки роботи будь-якого нейрону знаходиться *математична нерівність, яка розділяє простір ознак на дві частки*. Але наведені матричні перетворення (9.2.1) використовують **рівності**, тобто вони взагалі **не ураховують і втрачають** задану логічну інформацію нейронів у вигляді нерівностей. Логічна нееквівалентність наведених матричних перетворень доводиться також тим, що початкова велика кількість лінійних нейронів у багатьох початкових шарах лінійної НМ чомусь (після перетворень за формулою (9.2.1) вважається еквівалентною у інформаційному сенсі *одному нейрону* n_R в еквівалентному шарі n_R (в еквівалентному шарі у принципі може бути лише один нейрон $n_R = 1$)).

9.3. Розділ шарами персептрону простору ознак

Коли Розенблат довів теорему про навчання персептронів, то це викликало величезний інтерес. Але згодом Мінський показав, що існують жорсткі обмеження на те, що можуть виконати одношарові персептрони. Наприклад, Мінський довів, що одношаровий персептрон теоретично не може виконати функцію ВИКЛЮЧНЕ АБО. Існують багато простих машин, які не можуть бути представленими персептроном незалежно від величин його ваг. Про їх функції кажуть, що вони є лінійно нероздільними.

Мінський також навів патологічні приклади, у яких для представлення ваг потрібна кількість бітів пам'яті непомірно зростає і не може бути виконана. Усе це викликало розчарування (бо властивості багат шарових мереж та їх методи навчання ще не були відомі), і тому дослідження нейронних мереж зменшились й відновились лише згодом.

Обмеження, яке накладається на одношарові мережі, можна подолати за рахунок послідовного з'єднання двох та більше одношарових мереж [35; 44; 103; 191; 204].

Розглянемо розділ перцептроном простору ознак більш детально на прикладі вказаних на рис. 9.3.1 шарів нейронів:

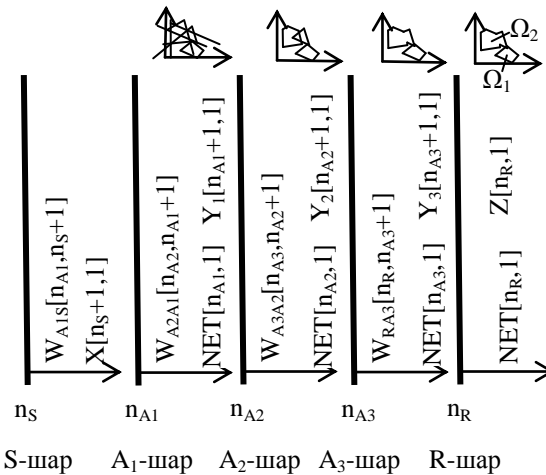


Рис. 9.3.1. П'ятишарова нейрона мережа [48; 49]

1. Сенсорний S-шар призначений для видачі нормалізованих ознак $X[n_S, 1]$ у n_S -мірному просторі. Кількість нейронів n_S дорівнює кількості ознак.

2. Асоціативний A₁-шар кожним своїм нейроном розділяє простір ознак на дві частини: в одній частині вихід нейрона дорівнює «+1», а в іншій – «-1»; кількість таких розділів залежить від кількості нейронів n_A . Розмірність n_S -ознак визначає геометричний вигляд області, в якій розглядається об'єкт:

- для однієї ознаки ($n_S=1$) одномірна ознака x_1 (пряма лінія) розділяється точкою нейрона на дві частини;

- для двох ознак ($n_S=2$) площа ознак x_1, x_2 розділяється прямою лінією нейрона у вигляді $y = a_0 + a_1x_1 + a_2x_2 > 0$ на дві півплощини: в одній півплощині нерівність $y > 0$ виконується, а в іншій – не виконується (рис. 9.3.2);

- для трьох ознак ($n_S=3$) об'єм ознак x_1, x_2, x_3 розділяється площею нейрона $y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 > 0$ на два пів об'єми: в одному пів об'ємі нерівність $y > 0$ виконується, а в іншому – не виконується;

- для кількості ознак $n_S > 3$ гіпероб'єм ознак x_1, x_2, \dots, x_N розділяється гіперплощею нейрона $y = a_0 + a_1x_1 + a_2x_2 + \dots + a_nx_n > 0$ на дві півгіпероб'єми: в одному півгіпероб'ємі нерівність $y > 0$ виконується, а в іншому – не виконується.

3. Асоціативний A₂-шар призначений для створення випуклих фігур із геометричних образів A₁-шару.

Створення випуклих фігур залежить від розмірності n_S -ознак:

- для однієї ознаки x_1 ($n_S=1$) із прямої лінії можна виділити відрізок;
- для двох ознак ($n_S=2$) із площі ознак x_1, x_2 можна виділити випуклі трикутники, чотирикутники, багатокутники (рис. 9.3.3);
- для трьох ознак x_1, x_2, x_3 ($n_S=3$) з об'єму ознак можна виділити випуклі об'ємні фігури – багатогранники: призми, піраміди, паралелепіпед та ін. (рис. 9.3.4);
- для кількості ознак $n_S > 3$ із гіпероб'єму ознак x_1, x_2, \dots, x_N можна виділити випуклі гіпероб'ємні фігури.

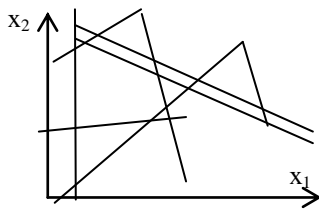


Рис. 9.3.2. Розділ кожним нейроном A_1 -шару простору ознак на дві частки.

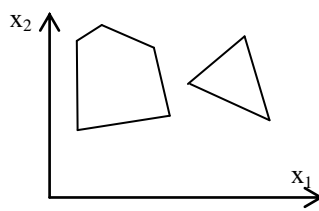


Рис. 9.3.3. Створення випуклих фігур A_2 -шаром для ознак x_1, x_2 .

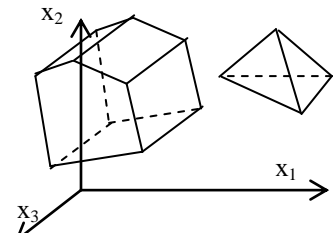


Рис. 9.3.4. Створення випуклих фігур A_2 -шаром для ознак x_1, x_2, x_3 .

Випуклі фігури створюються шляхом об'єднання вирішальних функцій окремих нейронів при використанні логічної функції «AND». Але нейрони A_2 -шару не обов'язково повинні обмежуватись лише логічною функцією «AND». Використання, наприклад, логічної функції «OR» дозволяє об'єднувати в один клас окремо розташовані області і створювати тим самим не випуклі фігури.

4. Асоціативний A_3 -шар призначений для створення складних областей існування класів за рахунок складання та віднімання фігур, створених попереднім A_2 -шаром: розділених простором, складних випуклих, складних не випуклих областей (рис. 9.3.5, рис. 9.3.6). Виконується це об'єднанням випуклих фігур, створених у попередньому A_2 -шар, за допомогою логічних функцій.

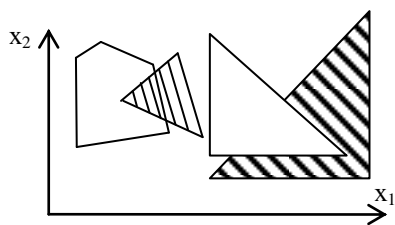


Рис.9.3.5. Створення не випуклих фігур A_3 -шаром для ознак x_1, x_2 .

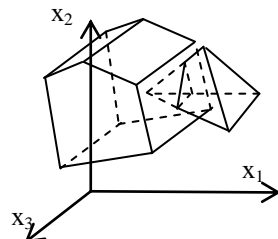


Рис. 9.3.6. Створення не випуклих фігур A_3 -шаром для ознак x_1, x_2, x_3 .

Нейрон A_3 -шару приймає набір випуклих багатокутників і за допомогою логічної комбінації може визначити не випуклу ділянку. Можна також

об'єднувати у один клас різні області (випуклі і не випуклі, розміщені у різних місцях таким чином, що вони не пересікаються).

Шар A_3 виконує узагальнюючі логічні дії, збираючи у загальну сукупність у вигляді випуклих і не випуклих фігур ділянки, отримані в шарі A_2 .

5. Проміжні асоціативні шари A_4, A_5, \dots (їх може бути кілька, на рис. 9.3.1 вони не показані) можуть виконувати наступні логічні дії по групуванню отриманих образів.

6. Реагуючий R-шар призначений для об'єднання створених ділянок асоціативного A_3 -шару у класи: його вихідні сигнали відповідають класу об'єкта, який розпізнається.

Трьохшаровий перцептрон (S–A₁–R) має такі недоліки:

- Для навчання вимагає занадто велику кількість еталонів, із-за чого збільшується час навчання.

- Невелика спроможність до виділення узагальнюючих елементів у складному сенсорному полі.

Для зменшення цих недоліків можуть використовуватися багатошарові перцептрони та перцептрони з перехресними зв'язками.

З рис. 9.3.2 – рис. 9.3.6 впливає математична неоднозначність вирішальних функцій, які розділяють простір ознак на частки (кластери) з метою визначення класів, бо можна провести багато вирішальних функцій, які розв'язують однакову задачу.

9.4. Дельта-правило навчання нейрона

На початку навчання значення вагових коефіцієнтів встановлюються рівними малим випадковим значенням, наприклад, у діапазоні $\pm(0,1 \dots 0,3)$; розподіл знаків « \pm » вагових коефіцієнтів теж виконується стохастично. Це захищає входи нейронів від насичення великими вагами, внаслідок чого настає «параліч НМ», і ЕОМ зупиняється. Також не рекомендується надавати вагам нульові чи однакові початкові значення. За дельта – правилом навчаються *всі вагові коефіцієнти нейрона, включаючи коефіцієнти вільного члена.*

Алгоритм, який зветься *дельта-правилом, або правилом Відроу-Хоффа* [204], використовується як для бінарних, так і для безперервних входів та виходів. Розрахунки вагових коефіцієнтів i -го нейрону виконуються за формулою

$$w_{ji}^{HOBE} = w_{ji}^{СТАРЕ} + \eta(y_{Ei} - y_i) \cdot x_j = w_{ji}^{СТАРЕ} + \eta \delta_i \cdot x_j, \quad (9.4.1)$$

де y_{Ei} , y_i – еталонне та реальне (розраховане або виміряне) значення виходу i -го нейрону; $\delta_i = (y_{Ei} - y_i)$ – похибка i -го нейрона; $w_{ji}^{СТАРЕ}$, w_{ji}^{HOBE} – старе та нове значення вагових коефіцієнтів j -го входу i -го нейрону; $\eta = 0 \dots 1$ – коефіцієнт навчання (звичайно приймається однаковим для всіх нейронів і рівним $\eta = 0,05 \dots 0,3$); $i = 1, 2, \dots, m$ – порядковий номер нейрона; $j = 1, 2, \dots, n$ – порядковий номер входу та вагового коефіцієнту i -го нейрону; x_j – змінна на j -му вході i -го нейрону.

За формулою (9.4.1) для корегування вагових коефіцієнтів $w_{ji}^{СТАРЕ}$ отримується не точна величина поправки (ми знаємо, що *положення вирішальної функції є математично невизначеним*), а лише **вимога до напрямку їх зміни**. Зміна вагових коефіцієнтів відбувається поступово за ітераційними розрахунками з неодноразовою перевіркою дотримання вимог формули (9.4.1).

Перевірити вірність напрямку зміни ваги за формулою (9.4.1) можна наступним чином:

1. Зміна ваги відбувається, якщо похибка виходу та вхідна змінна не дорівнюють нулю ($\delta = (y_{IE} - y_I) \neq 0; x_j \neq 0$).

2. Модуль зміни старого значення вагового коефіцієнту $w_{ij}^{СТАРЕ}$ визначається з добутку похибки $\delta = (y_{IE} - y_I)$ та вхідної змінної x_j : чим більшою є кожна з цих величин, тим на більшу величину змінюється відповідна вага.

3. Знак зміни ваги похибки $\eta\delta \cdot x_j = \eta(y_{IE} - y_I) x_j$ визначається наступним чином:

– **Якщо похибка виходу має позитивне значення** $(y_{Ei} - y_i) > 0$, то потрібно збільшувати y_i , тобто збільшувати NET_i . В цьому разі для позитивної вхідної змінної x_j потрібно збільшувати вагу, а для від'ємної вхідної змінної x_j потрібно зменшувати вагу: ці дії збільшують $NET_i = \sum_{j=0}^n w_{ij} x_j$ і збільшують y_i .

– **Якщо похибка виходу має негативне значення** $(y_{Ei} - y_i) < 0$, то потрібно зменшувати y_i , тобто зменшувати NET_i . В цьому разі для позитивної вхідної змінної x_j потрібно зменшувати вагу, а для від'ємної вхідної змінної x_j потрібно збільшувати вагу: ці дії зменшують $NET_i = \sum_{j=0}^n w_{ij} x_j$ і зменшують y_i .

В процесі навчання на вхід системи подається одна навчальна еталонна пара за іншою, в результаті чого вагові коефіцієнти нейронів корегуються до тих пір, доки для усіх еталонів похибка у розпізнаванні об'єктів не стане менше деякого прийнятого значення.

Завдання. 1. Розділ простору ознак прямою лінією. На площі ознак з координатами $x_1 = 0 \dots N$, $x_2 = 0 \dots N$ провести довільну ідеальну вирішальну функцію за формулою $u = x_1/A + x_2/B \geq 1$, де одне із значень (А або В – обираються студентом) є від'ємним, а N – порядковий номер студента у групі. Навести графік ідеальної вирішальної функції, і на графіку показати по 10 навчальних еталонних точок для кожного з двох класів. Навести еталонні навчальні пари «вхід-вихід» у таблиці. Навести програму навчання нейрону за дельта – правилом на будь-якій мові і за її допомогою розділити 50 об'єктів, координати (x_1, x_2) для яких отримати генератором випад-

кових чисел. Навести графіки ідеальної та отриманої вирішальної функції. Навести три схеми нейрона: з ідеальними ваговими коефіцієнтами, до навчання та після навчання.

Використати графічний режим ЕОМ. У графічному режимі вивести на екран ідеальний та отриманий графіки вирішальних функцій, точки координат 50-ти об'єктів і цифрові результати їх розділу по 2-х класах.

2. Розділ простору ознак гіперболою. На площі ознак у межах значень координат $x_1 = 0 \dots N$, $x_2 = 0 \dots N$ використати ідеальну вирішальну функцію за формулою $u = x_1 x_2 - N \geq 0$, де N – порядковий номер студента у групі (для $N = 1$ використати $x_1 = 1 \dots 1,5$; $x_2 = 1 \dots 1,5$). Навести схему нейрона. Навести графік ідеальної вирішальної функції, і на графіку показати по 10 навчальних еталонних точок для кожного з двох класів. Навести еталонні навчальні пари «вхід-вихід» у таблиці. Навести програму навчання нейрону за дельта – правилом на будь-якій мові і за її допомогою розділити 50 об'єктів, координати (x_1, x_2) для яких отримати генератором випадкових чисел. Навести формулу отриманої вирішальної функції.

Використати графічний режим ЕОМ. У графічному режимі вивести на екран ідеальний та отриманий графіки вирішальних функцій, точки координат 50-ти об'єктів і цифрові результати їх розділу по 2-х класах.

3. Розділ простору ознак колом. На площі ознак у межах значень координат $x_1 = 0 \dots (2N+10)$, $x_2 = 0 \dots (2N+10)$ використати ідеальну вирішальну функцію за формулою $u = (x_1 - x_1^0)^2 - (x_2 - x_2^0)^2 - N^2 \geq 0$, де $x_1^0 = N+5$; $x_2^0 = N+5$; N – порядковий номер студента у групі. Навести схему нейрона. Навести графік ідеальної вирішальної функції, і на графіку показати по 10 навчальних еталонних точок для кожного з двох класів. Навести еталонні навчальні пари «вхід-вихід» у таблиці. Навести програму навчання нейрону за дельта – правилом на будь-якій мові і за її допомогою розділити 50 об'єктів, координати (x_1, x_2) для яких отримати генератором випадкових чисел. Навести ідеальну та отриману вирішальну функцію.

Використати графічний режим ЕОМ. У графічному режимі вивести на екран ідеальний та отриманий графіки вирішальних функцій, точки координат 50-ти об'єктів і цифрові результати їх розділу по 2-х класах.

9.5. Алгоритм зворотного розповсюдження похибки у процесі навчання

Розглянемо у багатошаровій НМ три шари A_z , A_x та A_y з кількістю нейронів p , n та m (рис. 9.5.1).

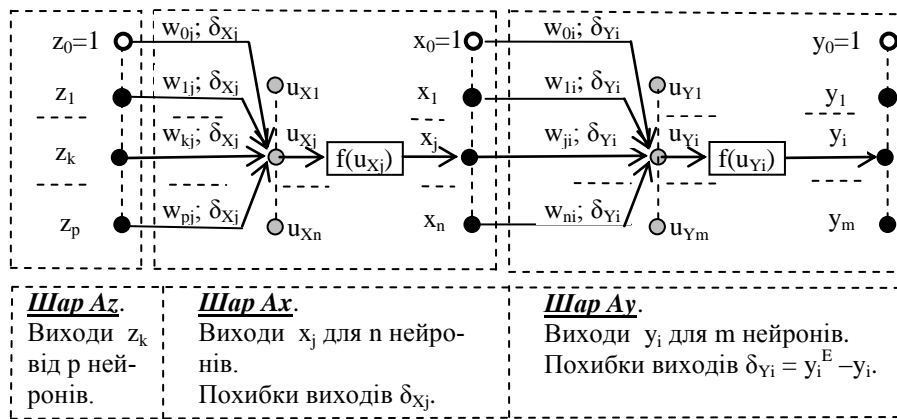


Рис. 9.5.1. Фрагмент схеми заміщення НМ (три шари багат шарової НМ) для розрахунку прямого потоку сигналів від початкового вхідного шару до кінцевого шару на першому етапі

На рис. 9.5.1 помічено:

1. Для трьох шарів (Az , Ax та Ay) виходи нейронів кожного шару позначені чорними точками з вказівкою значень їх виходів ($z_1, \dots, z_k, \dots, z_p$); ($x_1, \dots, x_j, \dots, x_n$); ($y_1, \dots, y_i, \dots, y_m$). Входи нейронів для урахування вільних членів вирішальної функції нейрона ($z_0=1$, $x_0=1$, $y_0=1$) помічені білим колом.

2. Для шарів Ax та Ay :

- вагові коефіцієнти входів нейронів x_j та y_i позначені ($w_{0j}, \dots, w_{kj}, \dots, w_{pj}$), ($w_{0i}, \dots, w_{ji}, \dots, w_{ni}$);
- похибки нейронів шарів Ax , Ay показані на вагах лише одного нейрона для кожного шару у вигляді δ_{xj} та δ_{yi} . Повний перелік похибок на виходах нейронів шарів Ax , Ay : ($\delta_{x1}, \dots, \delta_{xj}, \dots, \delta_{xn}$); ($\delta_{y1}, \dots, \delta_{yi}, \dots, \delta_{ym}$).
- похибки виходів нейронів x_j та y_i позначені δ_{xj} та δ_{yi} ;
- значення виходів нейронів NET позначені у вигляді ($u_{x1}, \dots, u_{xj}, \dots, u_{xn}$); ($u_{y1}, \dots, u_{yi}, \dots, u_{ym}$);
- активаційні функції нейронів x_j та y_i позначені $f(u_{xj})$ та $f(u_{yi})$.

Перед початком навчання всім вагам ($w_{0j}, \dots, w_{kj}, \dots, w_{pj}$) та ($w_{0i}, \dots, w_{ji}, \dots, w_{ni}$) надаються невеликі випадкові значення у межах $-0,25 \dots +0,25$. Це запобігає насиченню входів нейронів великими вагами. Не рекомендується надавати вагам нульові чи однакові початкові значення. Збільшення рівня навчання η зменшує час навчання, але значення рівня навчання η повинне бути достатньо малим (звичайно $\eta = 0,1 \dots 0,3$), щоб уникнути «вибухового» збільшення розрахованих значень вагових коефіцієнтів («розходження»), із-за чого ЕОМ припиняє свою роботу, бо наступає «параліч НМ».

У 1974 р. П. Д. Вербос та А. И. Галушкин незалежно знайшли алгоритм зворотного розповсюдження похибки, який раніше був відкритий Брайсоном та Хо у 1969 р. Алгоритм зворотного розповсюдження похибки (backpropagation), запропонований Руммелхазтом [191] та Вербосом [203], є узагальненням дельта-правила для багат шарових НМ з будь-якою кількістю шарів у тому числі і для мереж із зворотними зв'язками.

Згідно алгоритму *зворотного розповсюдження похибки* виконуються наступні два етапи:

1. *На першому етапі* розраховується прямий потік сигналів від початкового вхідного шару до кінцевого шару. В результаті для кожного окремого нейрону кожного шару визначається значення вирішальної функції ($u_{x_j} = \text{NET}_{x_j}$, $u_{y_i} = \text{NET}_{y_i}$) та нелінійний вихід нейрону (x_j , y_i) з урахуванням активаційної функції [$x_j = f(u_{x_j})$, $y_i = f(u_{y_i})$].

2. *На другому етапі* виконується розрахунок зворотного потоку похибок від кінцевого шару до початкового вхідного шару з корегуванням вагових коефіцієнтів всіх нейронів *включаючи ваги вільних членів*.

Якщо вважати шар A_y вихідним, то спочатку для кожного i -го, $i = 1, 2, \dots, m$, окремого нейрону y_i кінцевого вихідного шару A_y визначається похибка $\delta_{y_i} = (y_i^E - y_i)$ між еталонним y_i^E та розрахованими значеннями виходу y_i і розраховуються нові вагові коефіцієнти входів нейронів шару A_y , включаючи ваги вільних членів w_{0i} , за формулою дельта-правила

$$w_{ji}^{HOBE} = w_{ji}^{CTAPE} + \eta(y_{Ei} - y_i) \cdot x_j = w_{ji}^{CTAPE} + \eta \delta_{y_i} \cdot x_j, \quad (9.5.1)$$

де $i = 1, 2, \dots, m$ – порядковий номер нейрону y_i шару A_y ; $j = 0, 1, 2, \dots, n$ – порядковий номер входу кожного i -го нейрону шару A_y ; $x_0 = 1$, w_{0i}^{HOBE} , w_{0i}^{CTAPE} – параметри входу вільного члена нейрону y_i шару A_y .

Але далі (при переміщенні з кінця у початок) *ми не знаємо еталонні сигнали* виходів нейронів x_{Ej} і тому не можемо визначити всі похибки $\delta_{x_j} = (x_{Ej} - x_j)$, як різниці між еталонними x_{Ej} і розрахованими сигналами x_j .

Якщо далі формально застосувати дельта-правило до наступного нейрона (у ближчому до виходу шарі A_x) з корегуванням вагових коефіцієнтів ($w_{0j}, \dots, w_{kj}, \dots, w_{pj}$) входів нейрона x_j , то розрахунки потрібно виконувати за аналогічною формулою

$$w_{kj}^{HOBE} = w_{kj}^{CTAPE} + \eta(x_{Ej} - x_j) \cdot z_k = w_{kj}^{CTAPE} + \eta \delta_{x_j} \cdot z_k, \quad (9.5.2)$$

де $\delta_{x_j} = (x_{Ej} - x_j)$ – невідома величина із-за незнання x_{Ej} .

Для якісної оцінки значення δ_{x_j} можна уявити, що вихідні похибки ($\delta_{y_0}, \dots, \delta_{y_i}, \dots, \delta_{y_m}$) проходять у зворотному напрямку по всім шарам (з виходу НМ до її входу) у відповідності з наявними ваговими коефіцієнтами w_{ji}^{CTAPE} та w_{kj}^{CTAPE} .

Для зв'язків між нейронами шарів (A_y, A_x) та (A_x, A_z) проходження похибок у зворотному напрямку (з виходу НМ на її вхід) показане стрілками на рис. 9.5.2 [35]. Тобто у дійсності ми отримуємо дві схеми заміщення нейронів: одну – для прямого переміщення сигналів (з початку у кінець, рис. 9.5.1) і другу – для переміщення сигналу у зворотному напрямку (з кінця у початок, рис. 9.5.2). У зворотному напрямку значення похибок проходять по зваженим вагам у напрямку від кінця до початку.

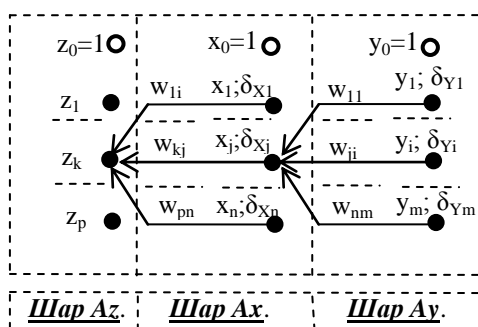


Рис. 9.5.2. Фрагмент схеми заміщення НМ (три шари багатошарової НМ) для розрахунку проходження похибок у зворотному напрямку від кінцевого шару до початкового вхідного шару на другому етапі

Вважаємо, що в НМ з одним захованим шаром Ax при розповсюдженні похибок у зворотному напрямку відома розрахована нами похибка кожного вихідного елемента ($\delta_{y_0}, \dots, \delta_{y_i}, \dots, \delta_{y_m}$) вносить свій «вклад» в похибку ($\delta_{x_0}, \dots, \delta_{x_j}, \dots, \delta_{x_n}$) нейрона захованого шару Ax. Для нейрона x_j захованого шару Ax цей вклад залежить від похибки δ_{y_i} та вагового коефіцієнта w_{ji} кожного нейрону y_i шару Ay.

У сукупності похибку на виході нейрону x_j можна вважати рівною

$$\delta_{x_j} = \sum_{i=1}^m w_{ji} \delta_{y_i} \quad (9.5.3)$$

Ця залежність підтверджена й аналітичними розрахунками при використанні сігмоїди для активаційної функції. Але отриманий аналітичний вираз має дещо складніший вигляд

$$\delta_{x_j} = \sum_{i=1}^m \{w_{ji} \delta_{y_i} [x_j(1-x_j)][y_i(1-y_i)]\} \quad (9.5.4)$$

В формулі (9.5.4) добутки у вигляді $[x_j(1-x_j)]$ та $[y_i(1-y_i)]$ не дозволяють змінювати вагові коефіцієнти при наближенні x_j (або y_i) до 1 або до 0. Це обмежує можливість неконтрольованого зростання значень вагових коефіцієнтів, яке може призвести до «паралічу НМ». Тому ці коефіцієнти $[x_j(1-x_j)]$ та $[y_i(1-y_i)]$ рекомендується використовувати при навчанні всіх НМ, навіть якщо для їх нейронів застостовуються інші активаційні функції.

З урахуванням коефіцієнтів $[x_j(1-x_j)]$ та $[y_i(1-y_i)]$ дельта-правило має вигляд

$$w_{kj}^{HOBE} = w_{kj}^{CTAPE} + \eta z_k \sum_{i=1}^m \{w_{ji} \delta_{y_i} [x_j(1-x_j)][y_i(1-y_i)]\} = w_{kj}^{CTAPE} + \eta z_k \delta_{x_j} \quad (9.5.5)$$

де $\delta_{x_j} = \sum_{i=1}^m \{w_{ji} \delta_{y_i} [x_j(1-x_j)][y_i(1-y_i)]\}$.

Завдання 1. Програма роботи навченої НМ. НМ складається з двох шарів: сенсорного шару з кількістю нейронів $n_S = 2$ та реагуючого шару з кількістю нейронів $n_R = 2$. Дві ознаки сенсорного шару є числовими

($x_1=0\dots N$, $x_2=0\dots N$, де N – порядковий номер студента у групі). Розділити на чотири частки простір ознак ($x_1=0\dots N$, $x_2=0\dots N$) двома довільними вирішальними функціями з їх «X-подібним розміщенням» для об'єктів 4-х класів (K1 – K4). Площа одного з обраного студентом класів повинна складати приблизно 10 % від загальної площі. Ділянки інших класів повинні бути приблизно однаковими.

Визначити нерівності двох вирішальних функцій і по них – вагові коефіцієнти нейронів реагуючого шару з кількістю нейронів $n_R = 2$. Скласти вирішальні правила. За допомогою генератора випадкових чисел отримати координати 100 об'єктів і програмно визначити, скільки об'єктів попало у кожний клас (K1 – K4).

У графічному режимі вивести на екран: 1-й квадрант з вирішальними функціями; помічені координати всіх об'єктів, що розпізнавались; кількість розпізнаних об'єктів для кожного класу у текстовому вигляді.

У звіті привести: завдання; вирішальні функції та вирішальні правила; ідентифікатори; програму; вигляд екрану (з масштабуванням осей, вирішальними функціями, координатами розглянутих об'єктів, текстовим значенням розв'язку).

Завдання 2. Програма навчання НМ. НМ складається з двох шарів: сенсорного шару з кількістю нейронів $n_S = 2$ та реагуючого шару з кількістю нейронів $n_R = 1$. Дві ознаки сенсорного шару є числовими ($x_1=0\dots N$, $x_2=0\dots N$, де N – порядковий номер студента у групі). Розділити на дві частки простір ознак ($x_1=0\dots N$, $x_2=0\dots N$) однією довільною вирішальною функцією для об'єктів 2-х класів (K1, K2). Площа одного з обраного студентом класів повинна складати приблизно 30 % від загальної площі. Вагові коефіцієнти сенсорного шару не корегуються.

За допомогою вирішальної функції скласти таблицю для 30-ти еталонних навчальних пар у вигляді координат об'єктів та їх належності до класу. Ще додаткові 6 додаткових еталонних пар двох класів призначити для перевірки вірності навчання НМ.

Всім вагам вихідного нейрону НМ $n_R = 1$ надати випадкові значення у межах 0,05...0,25 з випадково розміщеними знаками «+» та «-». Рівень навчання $\eta = 0,1$. На будь-якій мові виконати програму навчання НМ за дельта-правилом. У кінці програми отримати сигнали виходу навченої НМ для 6-х додаткових передбачених еталонних пар.

У графічному режимі вивести на екран: 1-й квадрант з масштабованими осями, теоретичною та навченою вирішальною функцією вихідного нейрону; помічені координати для 6-ти еталонних пар, що розпізнавались.

У звіті привести: завдання; програму; вигляд екрану із значеннями осей ($x_1=0\dots N$, $x_2=0\dots N$), з початковою вирішальною функцією та функцією вихідного навченого нейрону. На екран вивести координати 6-ми об'єктів з їх класами, що розпізнавались навченою НМ, та текст з вказівкою класів за навченим нейроном.

9.6. Деякі напрямки навчання нейронних мереж

Методи навчання шляхом спрощення структури НМ розділяють на ряд напрямків:

1. Пряме визначення структури. Приклад – визначення логічних функцій.
2. Вилучення ваг, які найменше впливають на похибку (наприклад внаслідок малого числового значення добутку вагового коефіцієнта на вхідну змінну).
3. Вилучення нейронів із вхідного чи прихованого шарів на основі:
 - малої різниці між похибками двох мереж, які отримані шляхом введення в мережу та вилучення з мережі нейрону, що перевіряється;
 - рішення експерта по зміні НМ з наступною перевіркою точності роботи НМ;
 - використання вартісної функції (наприклад, за виразом Д. Румельхарта $C = \sum_{j=1}^n \sum_{i=1}^m \frac{w_{ji}^2}{1 + w_{ji}^2}$, яка для малих ваг w_{ji} зменшується до нуля;
4. Вилучення експертом шарів нейронів з наступною перевіркою точності роботи НМ.

У повнопов'язаних мережах кожний нейрон отримує всі вхідні сигнали та передає власний сигнал усім іншим нейронам, включаючи самого себе. У цьому випадку суматор нейрона практично розпадається на дві частини: одна розраховує суму вхідних сигналів, а інша – суму зворотних вихідних сигналів нейронів. Виходом є або всі виходи нейронів, або виходи деяких нейронів після кількох тактів роботи.

У загальному випадку мережа може мати зворотні зв'язки, тобто виходи деякого шару з'єднуються з входами цього ж шару або попередніх шарів. Подібні НМ мають пам'ять.

Нечіткі нейронні мережі (fuzzy-neural networks) ANFIS та TSK є *адаптивними нечіткими системами* (adaptive fuzzy systems). Алгоритми навчання адаптивних нечітких систем відносно трудомісткі і складні у порівнянні з алгоритмами навчання НМ; вони, як правило, складаються з двох стадій:

1. Генерація лінгвістичних правил у вигляді задач переборного типу.
2. Коректування функцій належності, яка застосовує оптимізацію в безперервних середовищах.

При цьому виникає суперечливість, бо для генерації нечітких правил потрібно знати функції належності. Крім того, при автоматичній генерації нечітких правил необхідно забезпечити їх повноту та несуперечливість.

Стохастичні методи навчання: для зменшення функції мети випадковим чином обирають ваги і коригують їх на невелике випадкове значення. Якщо корекція зменшує функцію мети, то зберігають її, а якщо ні, то корекцію відкидають і повертаються до початкового значення ваги. Кроки повторюють до мінімізації функції мети.

Цей процес може завести розрахунки у локальний мінімум без змоги вийти з нього. Тому корисною є стратегія великих початкових кроків (зі збереженням тільки тих змін ваг, які зменшують функцію мети) при поступовому зменшенні величини середнього випадкового кроку. Це дозволяє мережі пройти локальні мінімуми і забезпечує кінцеву стабілізацію мережі у глобальному мінімумі. Це суттєва перевага стохастичних методів навчання перед іншими, включаючи і метод зворотного розповсюдження помилок.

Алгоритми навчання за Хебом (Hebb D.). За алгоритмом Хеба вага зростає, якщо активовані обидва нейрони (джерело і приймач):

$$w_{ij}(t+1) = w_{ij}(t) + \eta y_i y_j,$$

де $w_{ij}(t+1)$ – нове значення величини ваги від нейрона i до нейрону j ; $w_{ij}(t)$ – старе значення величини ваги від нейрона-джерела i до нейрона-приймача j ; η – коефіцієнт швидкості навчання; y_i – вихід нейрона-джерела i ; y_j – вихід нейрона-приймача j ; $(t+1)$, (t) – номер ітерації.

Використовується це правило у асоціативних зв'язках та при самонавчанні з метою надання спроможності до самоорганізації, самоадаптації та поліпшення якості функціонування НМ. Зв'язки, які часто використовуються, підсилюються (це пояснює феномен навчання людини шляхом повторення та звикання).

Існує також диференційний метод навчання Хебба:

$$w_{ij}(t+1) = w_{ij}(t) + \eta [y_i(t+1) - y_i(t)][y_j(t+1) - y_j(t)],$$

де $y_i(t+1)$ і $y_i(t)$ – вихідні значення нейрона i на ітераціях $(t+1)$ та (t) ; $y_j(t+1)$ і $y_j(t)$ – теж саме для нейрона j . З формули видно, що навчаються ваги нейронів, виходи яких змінилися.

При навчанні з підкріпленням використовуються вектори вхідних сигналів $x(k)$ та вказівки вчителя $r(k)$: значення $r(k) = +1$ свідчить про вірний напрямок зміни ваг і про необхідність збільшення їх числового значення; $r(k) = -1$ вказує на необхідність протилежних дій. Учителем є середовище або його модель. НМ діє на середовище, а середовище – на НМ, тобто використовується зворотній зв'язок. Цей вид навчання не є популярним. Найчастіше використовується навчання з вчителем та самонавчання.

Навчання «з вчителем». НМ має у своєму розпорядженні еталонні навчальні пари: кожна еталонна пара вміщує еталонний вектор входу та еталонний вектор виходу. Навчання виконується подачею множини еталонних образів по одному на вхід НМ і налаштуванням ваг до отримання необхідного виходу.

Самонавчання нейронної мережі виконується шляхом послідовного розгляду множини лише вхідних навчальних векторів. Навчальний алгоритм сам встановлює такі вагові коефіцієнти, щоб достатньо близькі вхідні вектори давали б однакові виходи, тобто належали одному класу. При цьому звичайно змінюються вагові коефіцієнти, хоча деякі алгоритми змінюють й структуру НМ.

Самонавчання може виконуватись також шляхом розділу простору ознак на ряд елементарних об'ємів (елементарних кластерів), куди попадають кінцеві точки векторів вхідних сигналів з наступним об'єднанням елементарних кластерів чи їх подальшим зменшенням.

Мережа ART [23] навчається без учителя і реалізує алгоритм кластеризації, дуже схожий на алгоритм *«послідовного лідера»*.

Навчання без «вихователя» можливе, наприклад, у мережах адаптивного резонансу, при якому відбувається порівняння вхідного образу з еталонними образами в пам'яті мережі. Якщо немає відповідного еталону, то створюється новий еталон, що містить в собі цей вхідний образ.

Конкурендне навчання полягає у тому, що всі нейрони одного шару вважаються конкурентами, а перевагу надають лише одному нейрону (за принципом «переможець забирає все»).

Гرادієнтні методи навчання засновані на мінімізації деякої цільової функції $F(W)$. Корекція ваг відбувається за правилом

$$\Delta w_{ij} = -\eta \frac{\partial F(W)}{\partial w_{ij}}$$

Мінімізація функції енергії НМ, Хопфілд і Танк в своїй класичній роботі показали, як конструювати функцію енергії для конкретної оптимізаційної задачі [198]. Цей підхід отримав розвиток і для вирішення інших комбінаторних оптимізаційних задач: задачі комівояжера, задачі оптимального покриття тощо. Розвитком моделі Хопфілда для вирішення комбінаторних оптимізаційних задач і задач штучного інтелекту є машина Больцмана, запропонована і досліджена Джефері [153]. Для мережі Хопфілда характерним є те, що досягнення сталого стану ще не гарантує правильний відгук мережі через те, що мережа може зійтись до помилкового розв'язку, іноді званим *«химерою»*. Як правило, химери представляють собою *фрагменти різних образів*.

1. Вхідний сигнал вважається зразком першого кластера.

2. Наступний вхідний сигнал належить першому кластеру, якщо відстань до зразка першого кластера менше порога. У іншому разі другий вхідний сигнал – зразок другого кластера.

Цей процес повторюється для всіх наступних вхідних сигналів. Таким чином, число кластерів росте з часом і залежить, як від значення порога, так і від метрики відстані, що використовується для порівняння вхідних сигналів і зразків класів.

Основна частина мережі *ART* схожа з мережею Хемінга.

За допомогою послідовних зв'язків обчислюється відповідність вхідних сигналів і зразків кластерів. Максимальне значення відповідності підсилюється за допомогою латеральних зв'язків вихідних нейронів.

Розпізнавання спотвореного образу стосується НМ, для якої на вхід подається спотворений образ $X = X^E + \Delta X$, а вихід НМ видає спотворену

відповідь $Y = Y^E + \Delta Y$. Найбільш схожий еталонний образ НМ визначається, наприклад, за скалярним добутком. Можна також визначити кореляцію вхідного сигналу з еталонними сигналами з викликом того еталонного образу, для якого кореляція є максимальною. Правильне розв'язання можна отримати, якщо для сигналу завади ($\Delta X, \Delta Y$) поставити деякий «поріг», який дозволяє лише найбільш вагомим компонентам з'являтися на виході НМ.

Наявність у нейронів впливу окремих гальмуючих входів $v_i, i=1, 2, \dots, m$, на дію окремих збуджуючих входів $x_j, j=1, 2, \dots, n$ [151].

Забування інформації (у НМ з власною «системою підтримки») моделює спроможність людини забувати непотрібну вивчену раніше інформацію і звичайно полягає у поступовому зменшенні модуля (впливу) отриманих раніше у процесі навчання вагових коефіцієнтів.

Генетичні алгоритми (ГА) оптимізують НМ при використанні ідей адаптації до навколишнього середовища, які застосовують при виживанні популяції об'єктів живої природи. Для НМ в заданих умовах розшуковуються оптимальні вагові коефіцієнти, або визначається оптимальна структура НМ. Оптимізація структури є важливим напрямком аналізу, бо «сьогодні не існує єдиної або хоча б достатньо ефективної методології проектування структури мереж, тому вибір топології мереж є емпіричним, заснованим на досвіді проектувальника» [103, с. 364]. Корегування вагових коефіцієнтів за ГА нагадує стохастичне програмування, але у ньому закладені інші принципи оптимізації, які забезпечують досягнення глобального екстремуму, використання паралельних розрахунків, нечутливість до диференційованості активаційних функцій.

Стиснення даних виконується часто за методом кодування. Наприклад, можна не передавати весь образ художньої картини, а передати лише її номер в каталозі. У чорно-білому малюнку можна не передавати окремо кожний білий «б» або чорний «ч» піксели, а використовувати для кожного рядка пікселів код у вигляді «1065ч3462ч», що означає «10 білих + 5 чорних + 34 білих + 2 чорних (піксели)». Самоорганізувальна мережа Кохона розбиває зображення на кластери, об'єкти яких незначно розрізняються між собою. При умові, що невеликі втрати інформації не істотні, усі об'єкти одного кластера можуть бути описаними одним осередненим еталонним вектором у вигляді скороченої інформації, яка приблизно відповідає реальному образу. Таким чином, практично без впливу на якість, вдається скоротити інформацію в десятки разів.

Регресійний аналіз. Звичайно математична модель технологічних процесів будується на основі накопиченої інформації. Під моделлю розуміють оператор $F: X \rightarrow Y$, який перетворює вхідний вектор $X = (x_1, x_2, \dots, x_n)$ у вихідний вектор $Y = (y_1, y_2, \dots, y_n)$ [87]. Структура і параметри моделі визначаються по навчальній вибірці, яка характеризується парами векторів $(x_k, y_k), k=1, 2, \dots, K$.

Найчастіше використовуються *регресійні моделі* [91]. Але у [91; 138] показано, що апроксимація функціональної залежності $F: X \rightarrow Y$ степеневими багаточленами регресійного аналізу може використовуватись для малої розмірності n вхідного вектора X , бо із зростанням n різко збільшується число параметрів многочлена, які корегуються – це діє так зване «*прокляття розмірностей*» [136]. Наприклад, для апроксимації функції на 30 вхідних змінних потрібен степеневий багаточлен, який має приблизно 46 000 параметрів, що корегуються [138]. Ефективним математичним апаратом для обробки вибірки є *нейронні мережі*, які розв’язують проблему *нелінійної* апроксимації залежності $F: X \rightarrow Y$ з використанням байєсового підходу [87].

В роботі [138] доведено, що від числа вагових коефіцієнтів нейронів залежить точність апроксимації експериментальних даних. НМ з малою кількістю нейронів не може достатньо точно апроксимувати залежність $F: X \rightarrow Y$, а надлишок нейронів веде до підвищення точності в експериментальних точках і одночасно – до суттєвого підвищення похибки в проміжних точках (в цілому похибка апроксимації збільшується) [138; 176]. Оптимізація по дискретним параметрам (числу нейронів) є менш ефективною у відношенні розрахунків, ніж по безперервним [8; 87; 138].

При навчанні можна ураховувати деякі нові напрямки навчання, до яких можна віднести:

1. «Спосіб навчання нейронної мережі» [60], який не використовує дельта-правило чи *backpropagation*. Цей метод нагадує розпізнавання образів за методом еталонів, що «дробляться».

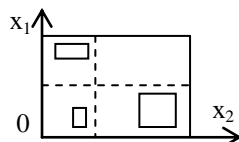


Рис. 9.6.1. Спосіб навчання нейронної мережі

Кожну j -ту ось ознаки x_j , де $j = 1, 2, \dots, n$, ділимо на фрагменти (див. штрихові лінії рис. 9.6.1), і на кожний фрагмент виділяємо одну активаційну функцію нейрона осі, яка змінює значення з $P_{ji}=0$ на $P_{ji}=1$, де $i = 1, 2, \dots, m_j$, якщо вхідний сигнал попав на фрагмент. Може трапитись три випадки для ділянки простору ознак, для якої всі $P_{ji}=1$:

1. В ній немає жодного об’єкта, і тому цей кластер вилучається з подальшого аналізу, як порожній.
2. В ній є об’єкти лише одного класу, і тому ділянка вилучається з подальшого аналізу як кластер цього класу.
3. В ній є об’єкти різних класів, і тому її фрагменти діляться на дрібніші фрагменти з наступним аналогічним аналізом.

Виділені кластери однотипних класів використовуються для розпізнавання вхідного образу.

2. Багатошарова НМ з навчанням лише передостаннього шару та призначенням всім іншим вагам нейронів детермінованого значення «+1» [62]. Виділяються ділянки окремих класів аналогічно рис. 9.6.1. Кожна окрема така ділянка характеризується навчальними векторами об'єктів з відповідними координатами. Вагові коефіцієнти вихідних нейронів передостаннього шару визначаються або за дельта правилом або за методом найменших квадратів. Нейрони останнього шару призначені для об'єднання різних кластерів в один клас у разі такої необхідності.

3. «Кластерний спосіб навчання нейронної мережі» [61] не використовує дельта-правило чи *backpropagation*. Навчання відбувається попереднім вилученням з аналізу тих навчальних об'єктів, які мають сусідами лише об'єкти однакового з ним класу (рис. 9.6.2). По суті виділяються лише пари сусідніх об'єктів на границях протилежних класів. Для таких сукупностей за методом найменших квадратів визначається вирішальна функція нейрону (тобто вагові коефіцієнти його входів). Виділені таким чином кластери об'єднуються у класи.

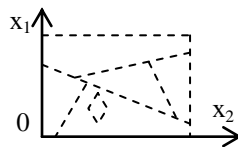


Рис. 9.6.2. Спосіб навчання нейронної мережі

При змішаному навчанні частина ваг визначається за допомогою навчання зі вчителем, у той час, як інша визначається за допомогою самонавчання чи за призначенням.

Завдання

1. Нейрон *Adaline* із входами x_1 , x_2 має початкові дані $W_0=+0,12$; $W_1=-0, N$; $W_2=+0,01 \cdot N$; $x_0 = +1$, де N – порядковий номер студента у групі. Скласти на будь-якій мові програму навчання мережі по виконанню AND – функції, або OR – функції. Навести програму і три значення вагових коефіцієнтів: початкові, проміжні, кінцеві.

2. Отримати систему РО для двох класів на базі нейронної мережі з використанням нейронів *Adaline* у площі ознак x_1 , x_2 (максимальні значення координат: $x_{1\max}=N$, $x_{2\max}=N$, де N – порядковий номер студента у групі):

– Двома паралельними лініями (які паралельні або « $12x_1(-1)^N - Nx_2=0$ » або « $5x_1 + 0,3N(-1)^N x_2=12N$ »). На площі ознак x_1 , x_2 виділити два класи.

– Отримати математичну модель-систему нерівностей для вирішальних функцій нейронної мережі та вирішальні правила. Показати графіки для вирішальних функцій.

- Створити електричну схему системи РО на базі нейронів Adaline та отриманої математичної моделі.
- Описати алгоритм навчання нейронної мережі.
- На будь-якій алгоритмічній мові розробити програму для навчання і перевірки якості навчання нейронної мережі.

Для навчання нейронної мережі використати вектори ознак не менше як 20-ти об'єктів. Ознаки x_1 , x_2 отримати від генератора випадкових чисел у межах $0..N$ (тут N – порядковий номер студента у групі). Інші 20 векторів з випадковими даними, отриманими від генератора випадкових чисел використати для перевірки якості навчання нейронної мережі.

3. Отримати нейронну мережу з використанням нейронів Adaline згідно п. 2 для випадку, коли площа ознак x_1 , x_2 розділяється двома довільними перехресними лініями (що по формі приблизно утворюють літеру «X») на чотири класи з координатами точки пересічення ($X_1=N/2$, $X_2=N/2$).

4. У площі двох ознак виділити у вигляді трикутника трьома довільними лініями довільну область класу. Отримати математичну модель для визначення об'єктів трикутника. Показати графіки для вирішальних функцій. Створити електричну схему системи РО. Описати алгоритм навчання нейронної мережі. На будь-якій алгоритмічній мові розробити програму для навчання й перевірки якості навчання нейронної мережі.

5. Скласти програму за методом зворотного розповсюдження помилки для навчання нейронної мережі, отриманої згідно п. 1.

10. Реалізація простих функцій нейронами

10.1. Нейрони «Троянда»

10.1.1. Порівняння нейронів «Троянда» з нейронами Adaline

Відомо, що математична нерівність нейрону $NET \geq 0$ виконує логічну функцію, бо *розділяє простір ознак на дві частки*: з одного боку поверхні функції NET у просторі ознак нерівність виконується, а з іншого – ні. Наприклад, лінійний вихід NETнейрону Adaline визначає нерівність гіперплощини з урахуванням деякого вільного члена a_0 , яка *розділяє гіпероб'єм ознак на дві частки*

$$y = a_0 + \sum_{j=1}^n a_j x_j \geq 0, \quad (10.1.1)$$

де $j = 1, 2, \dots, n$ – порядковий номер вхідних змінних; x_j – вхідні змінні; a_0 – вільний член; a_j – постійний коефіцієнт.

При зміні вільного члена a_0 будь-яка гіперплощина переміщується паралельно сама собі. Саме цей принцип і закладений у роботу нейрона «Троянда», для якого вираз (10.1.1) набуває вигляду [115]

$$y_i = \sum_{j=1}^n a_{ji} x_j \geq a_{0i}, \quad (10.1.2)$$

a_{0i} – заздалегідь задані вільні члени i -ої вирішальної функції; $i = 1, 2, \dots, m$ – порядковий номер i -ої вирішальної функції.

Вхідний образ НМ має вигляд вектора $X^A = (x_1^A, x_2^A, \dots, x_j^A, \dots, x_n^A)$. Ці значення координат нейрона підставляють у ліву частину нерівності (10.1.2) і розраховують вільний член a_0^A , який знаходиться між заздалегідь заданими фіксованими величинами a_{0i} .

Таким чином, **однією** математичною нерівністю (10.1.2) ми розділяємо простір ознак не на дві частки, як це виконується нейроном Adaline, а на $(m + 1)$ частку, тобто **один нейрон «Троянда» замінює m нейронів Adaline** і у принципі здатний розділити цей простір на довільну («нескінченну») кількість часток заданими граничними значеннями a_{0i} , тобто один нейрон «Троянда» теоретично може замінити «нескінченну» кількість нейронів Adaline.

Як приклад, на рис. 10.1.1 для двох вхідних змінних (x_1, x_2) показане розміщення трьох паралельних прямих ліній $x_1 + x_2 \geq a_i$ при трьох заздалегідь заданих вільних членах $a_{01} = +0,5$; $a_{02} = 0$; $a_{03} = -0,5$. Пунктирні лінії на рис. 10.1.1 вказують, для якої півплощини нерівність виконується. Для по-

дібного розділу треба використати три нейрони Adaline (по кількості існуючих прямих ліній). Ці три нейрони Adaline замінюються одним нейроном «Троянда», який на додаток у принципі може розділити весь простір ознак на будь-яку задану кількість ділянок m .

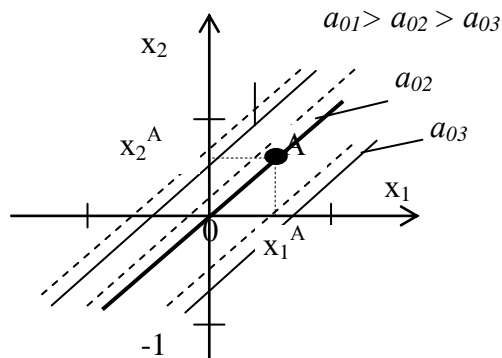


Рис. 10.1.1. Один нейрон «Троянда» замінює роботу 3-х нейронів Adaline

З цього випливає основні принципи, закладені в роботу нейрону «Троянда» [115]:

1. Нейрон «Троянда» не має активаційної функції.
2. По координатах вхідної точки $A(x_j^A)$ визначається вільний член a_{0i}^A площини, на якій знаходиться точка $A(x_j)$.
3. По значенню вільного члена a_{0i}^A площини точки $A(x_j)$ визначаються дві сусідні паралельні граничні площини, які заздалегідь задані їх вільними членами a_{0i} .
4. Таким чином, нейрон «Троянда» своїм вільним членом a_{0i}^A та двома сусідніми значеннями a_{0i} виділяє із всієї множини об'єктів універсуму обмежену їх кількість (разом з об'єктом входу A). Сама точка A має два обмеження – зверху та знизу сусідніми площинами.

Якщо для простору ознак $(x_1, x_2, \dots, x_j, \dots, x_n)$ на кожну координату x_j передбачити один нейрон «Троянда», то в цьому просторі виділяється мікроб'єм і виникає можливість визначення складного розміщення класів при скороченій кількості нейронів у порівнянні з використанням Adaline.

Для двох вхідних змінних на рис. 10.1.2 показано застосування двох нейронів «Троянда» для «розрізу» всього простору ознак «масштабною сіткою» на окремі інформаційні модулі, які можуть об'єднуватись у класи за допомогою логічних виразів. Для подібного розділу треба використати 8 нейронів Adaline (по кількості існуючих прямих ліній), які замінюються двома нейронами «Троянда» (по одному на кожну ось) і які на додаток можуть розділити весь простір ознак на значно більшу задану кількість ділянок.

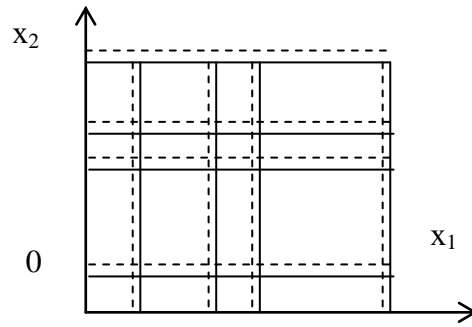


Рис. 10.1.2. Розділ двома нейронами «Троянда» універсуму на прямокутні інформаційні модулі (замінюють 4+4=8 нейронів Adaline)

Між тим в фаховій літературі підкреслюються лише недоліки нейронів без нелінійних активаційних функцій.

За допомогою нейронів «Троянда» можна виділити будь-яку окрему ділянку в просторі ознак [115].

Припустимо, що ми маємо виділену сірим кольором складну ділянку існування рішень рис. 10.1.3.

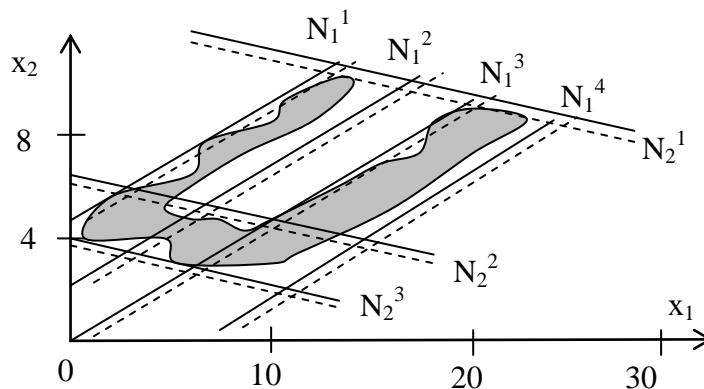


Рис. 10.1.3. Виділення довільної ділянки двома нейронами «Троянда» (замінюють 7 шт. нейронів Adaline)

Для виділення розглянутої ділянки ми повинні використати стільки нейронів Adaline, скільки ми маємо прямих ліній (7 шт.: $N_1^1 - N_1^4$, $N_2^1 - N_2^3$) з відповідним їх об'єднанням за допомогою логічних функцій.

Нейрони «Троянда» скорочують розрахунки, а при апаратній реалізації спрощують модель і зменшують її вартість.

Завдання.1. Порівняння нейронів Троянда з нейронами Adaline.

Частина 1. На площі ознак з координатами $x_1 = 0 \dots N$, $x_2 = 0 \dots N$ (тут N – порядковий номер студента у групі) довільно провести як показано на рис. 10.1.4 для 2-х нейронів («Троянда1» та «Троянда2») і 12-ти нейронів

Adaline вирішальні функції за формулами $u_{ij} = \frac{x_1}{a_{x1ij}} + \frac{x_2}{a_{x2ij}} - 1 \geq 0$, де $i=1,2$ – порядковий номер нейрона «Троянда1» та «Троянда2»; $j=1,2, \dots, 6$ – порядковий номер вирішальної функції двох нейронів «Троянда1» та «Троян-

да2»; a_{x1ij} , a_{x2ij} – числове значення координат точок, через які проходять вирішальні функції двох нейронів «Троянда1» та «Троянда2» і 12-ти нейронів Adaline.

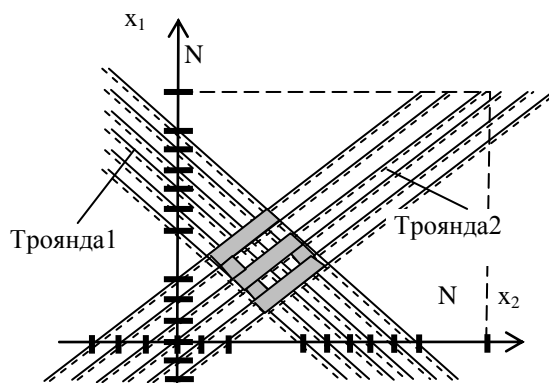


Рис. 10.1.4. Вирішальні функції нейронів «Троянда1» та «Троянда2» і 12-ти нейронів Adaline

В отриманій сітківці навести першу велику літеру власного прізвища (на рис. 10.1.4 наведена велика літера «Б»). Об'єкти, які попали своїми координатами на літеру, відносяться до класу Ω_1 , а всі інші об'єкти – до класу Ω_2 . Як бачимо, два нейрони «Троянда1» та «Троянда2» замінюють собою 12 нейронів Adaline.

Привести дві структурні схеми програм: одну для «Троянда1» та «Троянда2»; другу – для 12-ти нейронів Adaline.

В ділянці введеної літери ввести 5 точок, з яких три знаходяться у введеної літері (тобто відносяться до класу Ω_1). Вручну, без складання програми, за допомогою вирішальних функцій, згідно двох структурних схем програм, розділити ці 5 точок між класами Ω_1 та Ω_2 .

Навести: рисунок типу рис. 10.1.4 для власного варіанту рішень; аналітичні вирази вирішальних функцій; вирішальні правила; дві структурні схеми програм (для 2-х нейронів «Троянда1» та «Троянда2» і 12-ти нейронів Adaline); розрахунки по віднесенню 5-ти точок до відповідного класу.

Частина 2. Навести на будь-якій мові *дві програми* для розпізнавання складного розміщення двох класів (перший клас – це площа ознак літери, другий клас – все інше): одну з використанням 12-ти нейронів Adaline, а іншу з використанням 2-х нейронів Троянда.

Порівняти результати роботи двох програм по розділу 100 об'єктів, координати (x_1, x_2) для яких отримати за допомогою генератора випадкових чисел.

У графічному режимі вивести на екран два розв'язки (за нейронами Adaline та Троянда): вирішальні функції, точки координат 100 об'єктів і цифрові результати їх розділу по 2-х класах.

Завдання 2. Використання метода однозначних нечітких систем для розпізнавання образу. У даному випадку образ описується невідомою

досліднику функцією $y(x_1, x_2)$, яку потрібно визначити. На площі ознак $x_1, x_2 = 0 \dots N$ (N – порядковий номер студента у групі) виконати апроксимацію вихідної функції $y = Nx_1 x_2$, яку ми «не знаємо» і яка виконує роль експерта, що контролює отриманий результат. Розділити простір ознак на 4 частки аналогічно рис. 10.1.1 (на 4 інформаційних модуля). За формулою $y = Nx_1 x_2$ визначити виходи нечіткої нейронної мережі в опорних точках (можна додати й додаткові точки всередині інформаційних модулів). За методом найменших квадратів (МНК) визначити власну апроксимуючу функцію $y_A = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ для кожного з 4-х інформаційних модулів.

У звіті навести:

1. Для 4-х довільних точок (по одній всередині кожного інформаційного модуля) вказати: числові та нечіткі однозначні лінгвістичні значення координат x_1, x_2 ; числові та нечіткі лінгвістичні значення виходів y та y_A ; числові похибки апроксимації.

2. Визначити числові похибки апроксимації для всіх опорних точок по всіх сусідніх функціях інформаційних модулів.

10.1.2. Типи нейронів «Троянда»

Нижче розглядаються різні типи нейронів «Троянда».

Нейрон «Троянда1» складається з активаційної функції, яка розділяє ось вхідної змінної x_1 на множину ділянок незмінної величини $\Delta x_1^{\alpha_1} = \Delta x_1 = const$, порядкові номери яких $\alpha_1 = 1, 2, 3, \dots$ зростають у бік зростання величини x_1 [55] (рис. 10.1.5,а).

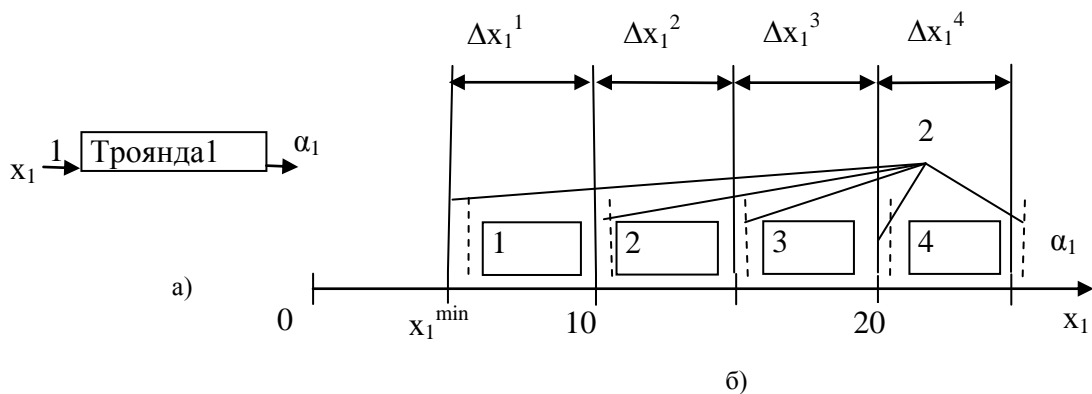


Рис. 10.1.5. Нейрон «Троянда1»

Ділянка $\Delta x_1^{\alpha_1}$, у яку попадає вхідне значення x_1 , має цілочисловий порядковий номер, що визначається активаційною функцією нейрону

$$\alpha_1 = 1 + \text{Trunc} \left(\frac{x_1 - x_1^{\min}}{\Delta x_1} \right),$$

де $\text{Trunc}(R)$ – функція усічення числа реального типу R до цілочислового значення у бік зменшення R шляхом вилучення його дробової частини; x_1^{\min} – мінімальне значення x_1 , з якого починається розділ x_1 на ділянки.

З рис. 10.1.5 випливає, що один нейрон «Троянда1» замінює 5 нейронів Adaline.

Для мережі з багатьма змінними, на кожену ось (x_1, x_2, \dots, x_n) можна використати власний нейрон «Троянда1», які у сукупності розділяють весь універсум на окремі інформаційні модулі аналогічні рис. 10.1.2.

Нейрон «Троянда2» виконує такі ж функції, що й «Троянда1», але він розділяє ось змінної x_1 на ділянки $\Delta x_1^{\alpha_1}$ не однакової довжини [56]. Вхідна змінна x_1 порівнюється з елементами вектора нижніх границь $x_1^{\alpha_1}$ ділянок $\Delta x_1^{\alpha_1}$ різної довжини, розміщених вздовж осі x_1 . Вектор границь зберігається у базі даних. Довжина ділянок може мати наперед задану довільну довжину

$$\Delta x_1^{\alpha_1} = x_1^{\alpha_1 + 1} - x_1^{\alpha_1} > 0,$$

де $\alpha_1 = 1, 2, \dots, K$ – порядкові номери ділянок $\Delta x_1^{\alpha_1}$ та їх нижніх границь $x_1^{\alpha_1}$; K – загальна кількість ділянок $\Delta x_1^{\alpha_1}$.

Вихід нейрона визначається на основі нерівності

$$y(x_1) = 0; \text{ якщо } x_1 \geq x_1^{\alpha_1}, \text{ то } y(x_1) = \alpha_1,$$

у вигляді порядкового номеру α_1 ділянки $\Delta x_1^{\alpha_1}$, на яку припадає значення вхідної змінної x_1 .

Перевірка нерівності $x_1 \geq x_1^{\alpha_1}$ починається з номера кінцевої ділянки $\alpha_1 = K$.

Нейрон «Троянда3» складається з двох блоків (рис. 10.1.6) [57]:

1. На вході першого блоку 1 з ваговими коефіцієнтами $w_1, w_2, \dots, w_j, \dots, w_n$ поступають вхідні змінні $x_1, x_2, \dots, x_j, \dots, x_n$, а на виході першого блоку 1 отримуються значення вільного члена $u_1 = \sum_{j=1}^n w_j x_j$ для площини, до якої належить вхідна точка.

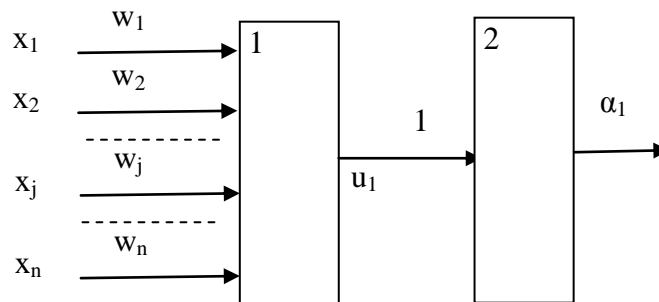


Рис. 10.1.6. Нейрон «Троянда3» та «Троянда4»: 1 – перший блок; 2 – другий блок

Координати будь-якої точки B , підставлені у вираз вигляду (10.1.2), дозволяють визначити величину вільного члена A_1^B для площини, на якій ця точка знаходиться

$$u_1^B = \sum_{j=1}^n w_j x_j^B = -A_1^B \quad (10.1.3)$$

2. Якщо заздалегідь задати сукупністю площин з фіксованими значеннями вільних членів A_i по рівнянню (10.1.2), то по величині $u_1^B = -A_1^B$ можна визначити, між якими з них знаходиться точка B .

Ця задача розв'язується у блоці 2, який складається з активаційної функції, що вказує вихід блоку 2-порядковий номер $\alpha_1=1, 2, 3, \dots$ однакових ділянок $\Delta u_1^{\alpha_1}$, упорядковано розміщених вздовж осі u_1 у бік зростання величини u_1 , на який припало вхідне значення u_1^B ,

$$\alpha_1 = 1 + \text{Trunc} \left(\frac{u_1^B - u_1^{\min}}{\Delta u_1} \right) = 1, 2, 3, \dots, \quad (10.1.4)$$

де $\text{Trunc}(\mathbb{R})$ – функція усічення числа реального типу \mathbb{R} до цілочислового значення у бік зменшення \mathbb{R} шляхом видалення його дробової частини;

$\Delta u_1^{\alpha_1} = \Delta u_1 = \text{const}$ – ділянки незмінної величини, на які ділиться ось u_1 ; u_1^B – вхідна величина блока 1; u_1^{\min} – мінімальне значення, з якого починається розділ u_1 на ділянки; $\alpha_1=1, 2, 3, \dots$, – вихідне значення нейрону, яке отримується з виходу блока 2.

На рис. 10.1.7 показаний графік розділу двома нейронами «Троянда3» простору змінних (x_1, x_2) та (u_1, u_2) на окремі однакові ділянки.

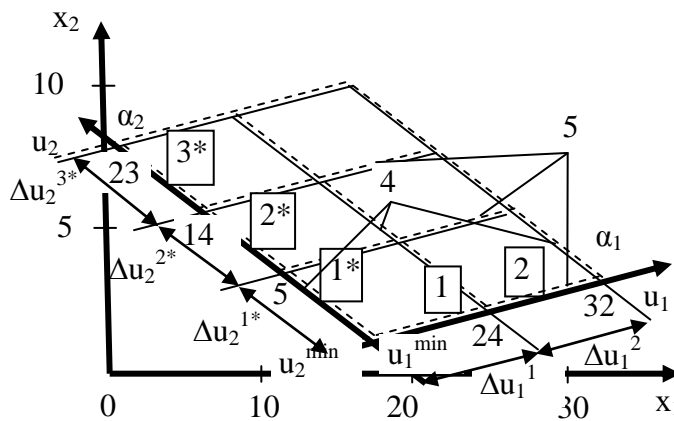


Рис. 10.1.7. Графік розділу двома нейронами «Троянда3» простору змінних (x_1, x_2) та (u_1, u_2) на окремі однакові ділянки

З рис. 10.1.7 випливає, що два нейрони «Троянда1» замінюють 7 нейронів Adaline.

Нейрон «Троянда4» працює аналогічно до нейрону «Троянда3» і так само складається з двох блоків [58], але розділяє двома нейронами «Троян-

да4» простір змінних (x_1, x_2) та (u_1, u_2) на окремі *не однакові* ділянки (рис. 10.1.8).

Тут два нейрони «Троянда4» замінюють $3+4=7$ нейронів Adaline. Таким чином, складність нейронної мережі з використанням нейронів «Троянда4», її вартість та час виконання операцій зменшуються приблизно у $7/2=3,5$ рази у порівнянні з використанням нейронів Adaline.

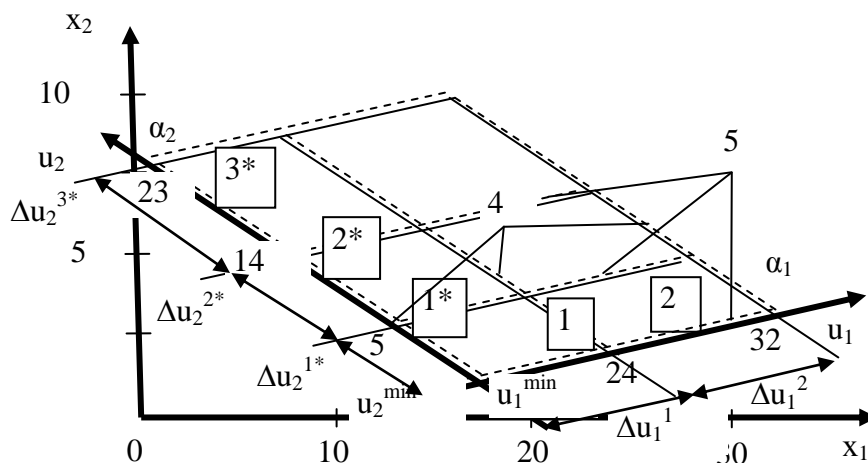


Рис. 10.1.8. Графік розділу двома нейронами «Троянда4» простору змінних (x_1, x_2) та (u_1, u_2) на окремі неоднакові ділянки

Нейрон «Троянда4» працює наступним чином:

1. На вході першого блоку 1 з ваговими коефіцієнтами $w_1, w_2, \dots, w_j, \dots, w_n$ поступають вхідні змінні $x_1, x_2, \dots, x_j, \dots, x_n$ для отримання на виході першого блоку значення проміжної функції $u_1^B = \sum_{j=1}^n w_j x_j^B = -A_1^B$, яке дорівнює вільному члену для площини, на якій знаходиться вхідна точка.

2. Блок даних 2 має вектор нижніх границь $u_1^{\alpha_1}$ ділянок з неоднаковою довжиною, на які ділиться ось u_1

$$\Delta u_1^{\alpha_1} = u_1^{\alpha_1 + 1} - u_1^{\alpha_1} > 0,$$

де $u_1^{\alpha_1}$ – нижні границі ділянок з неоднаковою довжиною $\Delta u_1^{\alpha_1}$, упорядковано розташованих за порядковими номерами $\alpha_1 = 1, 2, 3 \dots K$ вздовж осі значень проміжної функції u_1 у напрямку її збільшення.

Блок 2 нейрону складається з активаційної функції у вигляді логічної умови з використанням нерівності

$$y(u_1) = 0; y(u_1) = \alpha_1, \text{ якщо } u_1 \geq u_1^{\alpha_1}.$$

Перевірка нерівності $u_1 \geq u_1^{\alpha_1}$ по вектору бази даних з визначенням порядковому номеру α_1 починається з більшого номера ділянки $\alpha_1 = K$ з ве-

ликою величиною $u_1^{\alpha_1}$ у напрямку зменшення α_1 . При виконанні умови $u_1 \geq u_1^{\alpha_1}$ визначається порядковий номер α_1 ділянки $\Delta u_1^{\alpha_1}$, на яку припадає вхідне значення u_1 блоку 2.

10.1.3. Моделювання логічної функції XOR одним нейроном «Троянда»

В публікації [59] запропоновано використання одного нейрона «Троянда» під назвою «T_XOR» для виконання логічної функції XOR (рис. 10.1.9, а), яка звичайно моделюється трьома нейронами Adaline [35, с. 24, 44–48; 103, с. 84].

Для цього до двох входів нейрону «T_XOR» з ваговими коефіцієнтами 1 увімкнені відповідні дві вхідні змінні (x_1, x_2), які приймають значення «0» або «1». Основний блок нейрону є суматором, призначеним для розрахунку проміжної функції

$$u = x_1 - x_2,$$

а активаційний блок нейрону призначений визначати значення виходу логічної функції нейрону y за логічною умовою $y:=0; \text{ if } ((u > -0,5) \text{ and } (u < 0,5)) \text{ then } y:=1$.

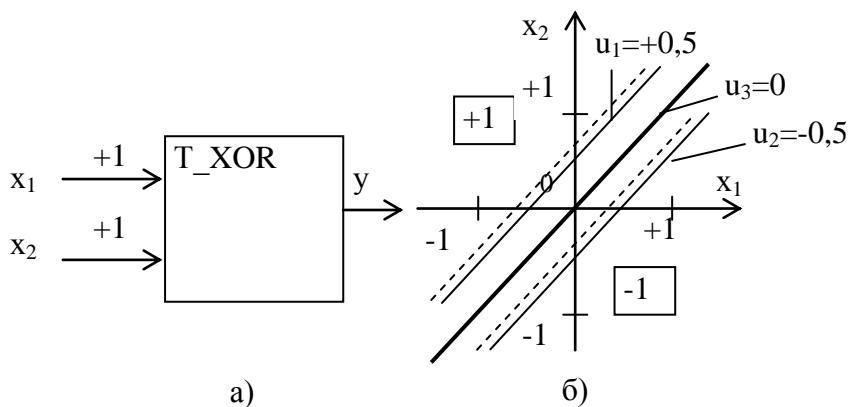


Рис. 10.1.9. Нейрон T_XOR

Тим самим спростовується відоме твердження М. Мінського та С. Пейперта (1969 р.) про теоретичну неможливість отримання логічної функції XOR на одному нейроні. Аналогічні твердження стосовно функції XOR наведені також в [35, с. 44–51].

Нижче описані інші три можливі варіанти отримання логічної функції XOR на одному нейроні.

1. В [115, Ч. II, с. 62] запропоновано використання одного нейрона «Троянда» для виконання логічної функції XOR. Теоретична можливість реалізації функції XOR підтверджується рис. 10.1.1, на якому один нейрон

«Троянда» виділяє саме ті ділянки, які виділяються двома нейронами Adaline на основі нерівностей для вирішальних функцій

$$u_1 = -x_1 + x_2 \geq 0,5; \quad (10.1.5)$$

$$u_2 = -x_1 + x_2 \geq -0,5. \quad (10.1.6)$$

Ці дві нерівності виконуються для півплощин, помічених на рис. 10.1.1 штриховими лініями.

Для виконання логічної функції XOR один нейрон «Троянда» використовує рівняння для середньої лінії між нерівностями (10.1.5) та (10.1.6)

$$u_3 = 0,5(u_2 - u_1) = x_1 - x_2 = 0, \quad (10.1.7)$$

до якої додається дія вихідної активаційної функції яка при програмній реалізації нейрона має вигляд логічної умови

$$y_{\text{XOR}} := +1; \text{ if } (u_1 \leq 0 \text{ AND } u_2 \geq 0) \text{ then } y_{\text{XOR}} := 0. \quad (10.1.8)$$

2. Для виконання логічної функції XOR замість логічної умови (10.1.8) також можуть бути використані активаційні «чашеподібні» функції, які показані на рис. 10.1.6, а, б та описуються формулами:

$$y_1(u) = \frac{1 - e^{-u}}{1 + e^{-u}}; \quad y_2(u) = \frac{\left(\frac{x-b}{c}\right)^2}{1 + \left(\frac{x-b}{c}\right)^2}. \quad (10.1.9)$$

де c – коефіцієнт концентрації – розтягування функції; b – початкова координата максимуму функції.

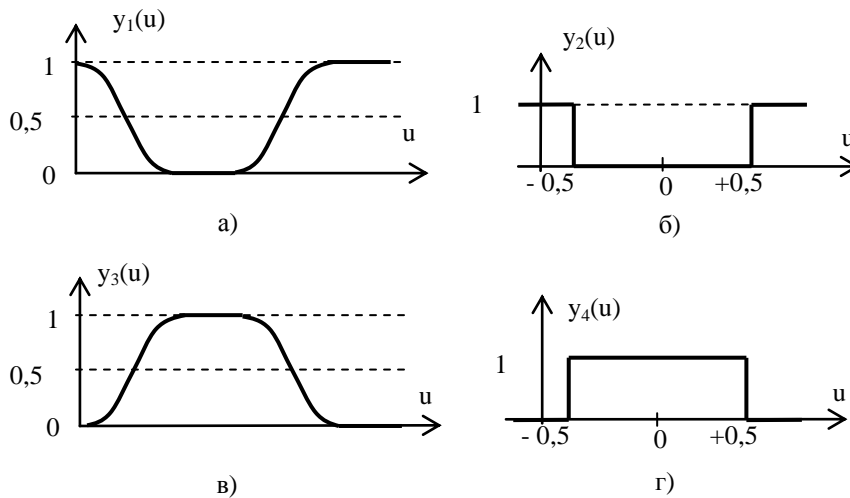


Рис. 10.1.10. Активаційні функції нейрона «Троянда»:

а), б) активаційні чашоподібні сігмоїдна та жорстка функції;
в), г) активаційні дзвоноподібні сігмоїдна та жорстка функції

3. Замість чашоподібних активаційних функцій (рис. 10.1.2, а, б) можуть бути використані «дзвоноподібні» функції – сігмоїдна (рис. 10.1.2, в) та жорстка (рис. 10.1.2, г), формули яких мають вигляд:

$$y_3(u) = \frac{2e^{-u}}{1 + e^{-u}}; \quad y_4(u) = \frac{1}{1 + \left(\frac{x-b}{c}\right)^2}, \quad (10.1.10)$$

Можливі структурні схеми нейрону «Троянда» з використанням «чашеподібних» (10.1.10) та «дзвоноподібних» (10.1.11) активаційних функцій наведені на рис. 10.1.9, а, б [115, Ч. II, с. 62].

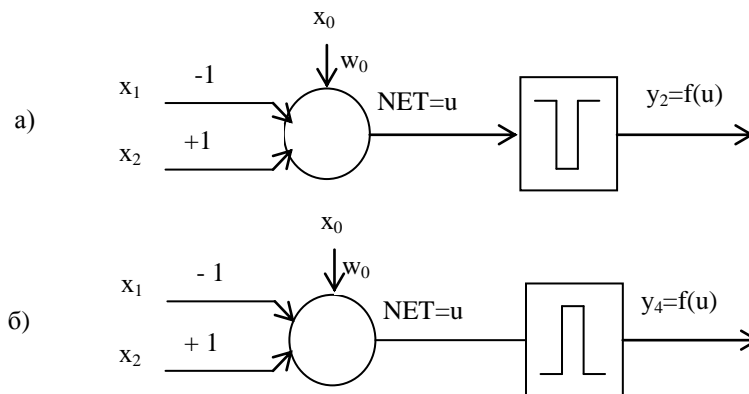


Рис. 10.1.11. Нейрон «Троянда»: а) з чашеподібною активаційною функцією (10.1.9); б) з дзвоноподібною активаційною функцією (10.1.9)

10.2. Логічні функції T_AND та T_OR з багатьма різнополярними входами

Недоліки використання лише стандартних булевих функцій «AND», «OR», «XOR» (y_{2AND} , y_{8OR} , y_{10XOR}) при наявності двох вхідних змінних можна визначити на прикладі розпізнавання класів $\Omega_1 - \Omega_3$, які на рис. 10.2.1 розділені вирішальними функціями $u_1 = 4x_1 - x_2 + 4 \geq 0$ та $u_2 = 12x_1 + 3x_2 - 36 \geq 0$ у вигляді двох рядом розмішених безперервної та штрихової паралельних ліній. Звичайно вважають, що з боку штрихової лінії нерівність $u_1 \geq 0$ виконується, а з протилежного боку – ні (тобто $u_1 < 0$). По рис. 10.2.1 видно, що у загальному випадку виділення одного класу вимагає сумісного використання вирішальних функцій зі значенням $u_i \geq 0$ та $u_i < 0$.

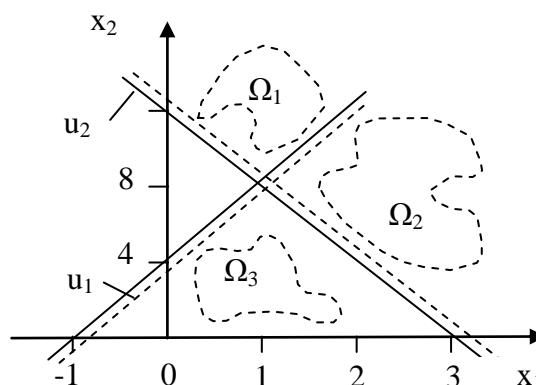


Рис. 10.2.1. Розділ двома нейронами ADALINE трьох класів $\Omega_1 - \Omega_3$

Наприклад, за рис. 10.2.1 для розпізнавання класу Ω_1 вимагається дотримання одночасних нерівностей $u_1 < 0$ та $u_2 \geq 0$, які можна об'єднати за

допомогою логічної функції «AND». Але Булева функція «AND» для отримання виходу «+1» вимагає від всіх вхідних змінних значень ($u_1 > 0$, $u_2 > 0$). Тому для розпізнавання класу Ω_1 треба інвертувати відповідне вхідне значення u_1 для використання Булевої функції «AND».

З цього видно, що при використанні Булевих функцій *навіть для двох вхідних змінних* не завжди отримується оптимальна конфігурація нейронної мережі. Подібна обмеженість ускладнює апаратну та програмну реалізацію нейронних мереж, особливо при багатьох змінних.

Створені в Україні, призначені для роботи в логічній системі з базовими змінними 1/0 нейрони T_AND_0, T_OR_0 [63; 73] спрощують нейронну мережу та розширюють її логічні можливості:

– Нейрони T_AND_0 [63] видають на виході 1, якщо вся перша частка заздалегідь вказаних входів мають значення 1, а вся друга частка заздалегідь вказаних входів мають значення 0. Кількість цих заздалегідь вказаних входів – довільна.

– Нейрони T_OR_0 [73] видають на виході 1, якщо будь-який хоча б один вхід серед першої частки заздалегідь вказаних входів має значення 1, або якщо будь-який хоча б один вхід серед другої частки заздалегідь вказаних входів має значення 0. Кількість цих заздалегідь вказаних входів – довільна.

Нейрони T_AND_1, T_OR_1 виконують ті ж самі функції, але призначені для роботи в логічній системі з базовими змінними 1/-1 [72; 74].

Структурна схема нейронів T_AND_0, T_OR_0, T_AND_1, T_OR_1 наведена на рис. 10.2.2.

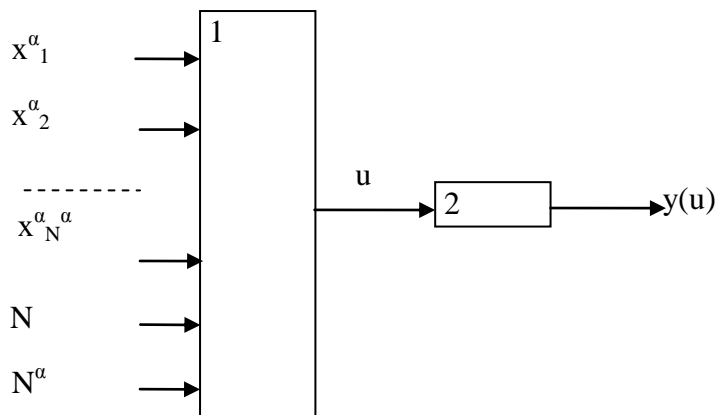


Рис. 10.2.2. Структурна схема нейронів T_AND_0, T_OR_0, T_AND_1, T_OR_1

Позначення на рис. 10.2.2 означають: 1 – блок для визначення вирішальної функції; 2 – блок активаційної функції; N – загальна кількість входів; $(x_1^\alpha, x_2^\alpha, \dots, x_{N^\alpha}^\alpha)$ – порядкові номери першої групи входів; N^α – загальна кількість першої групи входів; $(x_{N^\alpha+1}^\alpha, x_{N^\alpha+2}^\alpha, \dots, x_N^\alpha)$ – порядкові номе-

ри другої групи входів; $(N-N^\alpha)$ – загальна кількість другої групи входів; u – вирішальна функція; $y(u)$ – вихід.

Дані нейронів T_AND_0 , T_OR_0 , T_AND_1 , T_OR_1 наведені в табл. 10.2.1.

Таблиця 10.2.1.

Дані нейронів T_AND_0 , T_OR_0 , T_AND_1 , T_OR_1

Назва нейрона	Вирішальна функція	Функція активації
T_AND_0	$u = \sum_{i=1}^{N^\alpha} x_i^\alpha + \sum_{i=N^\alpha+1}^N (1-x_i^\beta) + (1-N)$	$y(u) = \begin{cases} 1, & u \geq 0,5, \\ 0, & u < 0,5. \end{cases}$
T_AND_1	$u = \sum_{i=1}^{N^\alpha} x_i^\alpha - \sum_{i=N^\alpha+1}^N x_i^\beta + (1-N)$	$y(u) = \begin{cases} 1, & u \geq 0,5, \\ -1, & u < 0,5. \end{cases}$
T_OR_0	$u = \sum_{i=1}^{N^\alpha} x_i^\alpha + \sum_{i=N^\alpha+1}^N (1-x_i^\beta)$	$y(u) = \begin{cases} 1, & u \geq 0,5, \\ 0, & u < 0,5. \end{cases}$
T_OR_1	$u = \sum_{i=1}^{N^\alpha} (1+x_i^\alpha) + \sum_{i=N^\alpha+1}^N (1-x_i^\beta)$	$y(u) = \begin{cases} 1, & u \geq 0,5, \\ -1, & u < 0,5. \end{cases}$

10.3. Вхідна і вихідна зірки Гросберга

Існують *вхідна і вихідна зірки С. Гросберга* (1969) – з входом в один нейрон та виходом з одного нейрону (рис. 10.3.1).

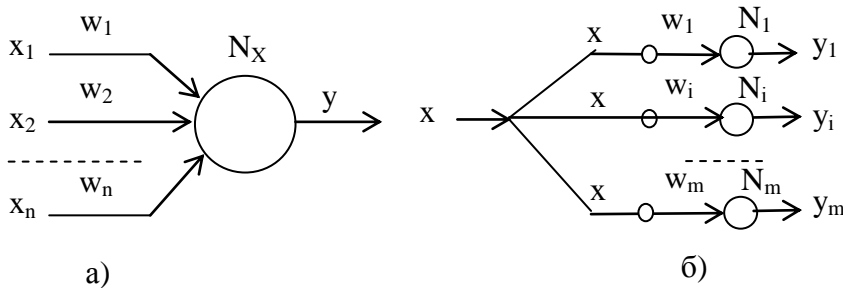


Рис. 10.3.1. Вхідна зірка (а) та вихідна (б) зірки Гросберга

Вхідна зірка Гросберга реалізується на одному нейроні N_X (рис. 10.3.1, а), який має n входів з вагами $\{w_1, w_2, \dots, w_n\}$, один вихід y з пороговою активаційною функцією 0/1 [103, с. 75]. На входи подається вхідний вектор $X = \{x_1, x_2, \dots, x_n\}$ і отримується підсумок добутків сигналів на

вагові коефіцієнти $u = \sum_{j=0}^n \pm w_j x_j$, а потім за допомогою *порогової активаційної функції* отримується вихід $y=f(u)=1$ при $u \geq 0$ та $y=f(u)=0$ у іншому випадку. Зовнішньо вхідна зірка Гросберга не відрізняється від *Adaline*. Різниця полягає у навчанні вагових коефіцієнтів, яка дозволяє видавати **вихідний сигнал 1 лише на визначену групу сигналів входу**, яка є близькою до центруду, координати якого записані у вагах w_j рис. 10.3.1, а.

При цьому вагові коефіцієнти W та вхідний сигнал X розглядаються як координати двох окремих векторів в заданому просторі ознак w_j, x_j при $j=1, 2, \dots, n$.

В математиці квадрат модуля вхідного вектора $X = \{x_1, x_2, \dots, x_n\}$ дорівнює квадрату радіусу $R^2 = x_1^2 + x_2^2 + \dots, x_n^2$ гіперкулі, всередині якої знаходиться вектор. Отримане у вхідній зірці Гросберга значення $NET=u=w_1 \cdot x_1 + w_2 \cdot x_2 + \dots, w_n \cdot x_n$ можна уявити як скалярний добуток двох векторів

$$NET=u= w_1 \cdot x_1 + w_2 \cdot x_2 + \dots, w_n \cdot x_n,$$

де вектор $W=\{w_1, w_2, \dots, w_n\}$ записаний у вагах нейрона і є еталоном, до якого повинен наближуватись вхідний сигнал $X = \{x_1, x_2, \dots, x_n\}$.

Саме з точки зору взаємодії цих двох векторів і навчається зірка Гросберга. Якщо вектори близькі один до одного у просторі ознак, то скалярний добуток двох спрямованих приблизно однаково векторів $u= \mathbf{w} \cdot \mathbf{x} = \mathbf{w} \cdot \mathbf{x} \cdot \cos\varphi \geq 0$ має максимальне значення, а якщо вони спрямовані у протилежні сторони, то добуток стає від'ємним $u < 0$ (тут $\mathbf{w} \cdot \mathbf{x}$ – добуток модулів векторів \mathbf{w} та \mathbf{x} ; φ – кут у просторі ознак між двома векторами). Для отримання $u=+1$ у вхідній зірці Гросберга при $n=3$ необхідно отримати значення скалярного добутку двох близько розміщених векторів (сигналу \mathbf{x} та записаного \mathbf{w}) при $j = 1, 2, 3$:

$$w_1 x_1 + w_2 x_2 + w_3 x_3 \geq w_0 x_0$$

Якщо задати занадто велике значення $w_0 x_0$, то вхідна зірка Гросберга при обмеженості величин реальних даних двох векторів ніколи не буде мати на виході $u=f(u)=1$. Якщо поступово зменшувати $w_0 x_0$, то у вхідній зірці Гросберга сигнал $u=f(u)=1$ з'явиться лише якщо два вектори максимального модуля (\mathbf{w} та \mathbf{x}) співпадуть по напрямку (якщо $\cos\varphi=1$). Якщо далі зменшувати значення $w_0 x_0$, та сигнал виходу $u=f(u)=1$ з'являється для зростаючої кількості векторів з приблизно однаковим напрямком. Тобто, чим менше $w_0 x_0$, тим більше можливих образів може виділити нейрон. Якщо ще далі зменшувати значення $w_0 x_0$, то можна отримати сигнал виходу $u=f(u)=1$ для будь яких реальних досліджуваних векторів \mathbf{w} та \mathbf{x} .

Вихідна зірка Гросберга (рис. 10.3.1, б) розглядається як один нейрон [103, с. 77]. Вона має скалярний вхід (він може розглядатись, як розмножувач вхідного сигналу) та m виходів з нейронами (N_1, N_2, \dots, N_m) з окремими активаційними функціями; якщо вхідний сигнал перевищує заданий однаковий для всіх виходів поріг, то на виході можна отримати вектор довільних цифр $Y=(y_1, y_2, \dots, y_m)$, визначений експертом. У загальному випадку вихідна зірка Гросберга має розширені можливості за рахунок зсуву вздовж осі x сусідніх значень $u_i = NET_i$ та використання різних функцій активації y_i для кожного виходу

$$y_i = f(w_i x), \quad i = 1, 2, \dots, m$$

з функцією активації

$$y_i = \begin{cases} u_i, & \text{якщо } -1 \leq u_i \leq 1; \\ 1, & \text{якщо } u_i > 1; \\ -1, & \text{якщо } u_i < -1. \end{cases}$$

У векторній формі навчання вихідної зірки виконується за правилом [103]

$$w_{ji}(t+1) = w_{ji}(t) + \eta \cdot x_i(t)[y_j(t) - w_{ji}(t)], \quad (10.3.1)$$

де $i = 1, 2, \dots, m$ – порядковий номер виходу нейрону; $w_{ji}(t+1)$, $w_{ji}(t)$ – нове і старе значення вагового коефіцієнту вхідного зв'язку нейрону; η – коефіцієнт навчання.

При $y_j(t) = w_{ji}(t)$ навчання припиняється. Зсув u_i вздовж осі x задається експертом.

10.4. Нейронна мережа Івахненко «Альфа»

Система «Альфа» (1971) розроблена під керівництвом українського вченого Івахненко А. Г. [18; 34]. Її перевагою є простота та здатність до самонавчання (цим вона нагадує персептрон). На вході до «Альфа» встановлені детерміновані датчики кількох ознак. Діє додатній зворотній зв'язок: він по черзі обходить всі вхідні сигнали і змінює стан системи таким чином, щоб досягнути найбільшого значення якогось показника якості (за рахунок зворотного додатного зв'язку система самонавчається). Якщо вхідні сигнали суттєво відрізняються від еталонів, то система РО запам'ятовує їх. Робота системи полягає у порівнянні отриманого сигналу у вигляді сукупності ознак $X=+1-1-1+1+1$ з еталонами, які мають вектор такого ж типу, наприклад, $E_1=+1-1-1-1+1$; $E_2=+1+1+1+1+1$.

Обробка сигналів виконується за скалярним добутком і полягає у сумуванні попарних добутків елементів вхідного вектора на відповідні елементи еталону $C_1=X \cdot E_1^T = (+1)(+1)+(-1)(-1)+(-1)(-1)+(+1)(-1)+(+1)(+1)=2$ (більша схожість).

$C_2=X \cdot E_2^T = (+1)(+1)+(-1)(+1)+(-1)(+1)+(+1)(+1)+(+1)(+1)=1$ (менша схожість).

Більше значення скалярного добутку означає більшу схожість отриманого сигналу з еталоном. Подібна мережа може використовуватись для розв'язку ряду задач. Вона має переваги у прискоренні розрахунків і збереженні оперативної пам'яті.

10.5. Нейронна мережа MAXNET для кодування елементів вхідного вектора

НМ MAXNET (рис. 10.5.1) описана у складі НМ Хеммінга [175] і призначена для кодування елементів вхідного вектора $X(x_1, x_2, \dots, x_n)$ у вигляді коду «1» для елемента – переможця (з максимальним позитивним значенням виходу) та коду «0» для всіх інших елементів. Але якщо усі елементи

вхідного вектора є від'ємними, то мережа буде мати на виході лише нульові значення, бо нейрон – переможець зведе власний вихід до 0. Якщо ввести значення вхідного вектора з протилежними знаками то MAXNET може використовуватись й для виділення найменшого негативного елемента сигналу.

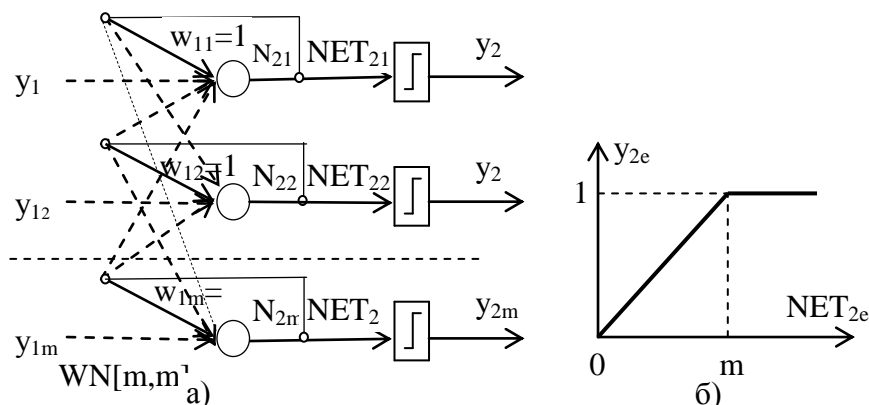


Рис. 10.5.1. Мережа MAXNET: а) структурна схема, б) активаційна функція нейронів

Початкові дані у вигляді $Y1\{y_{11}, y_{12}, \dots, y_{1m}\}$ вводять в НМ MAXNET **лише один раз** – на початок рекурентних розрахунків (далі вхідні дані викликаються і не використовуються). Тому на рис. 10.5.1, а введення вектора вхідних даних показане умовною пунктирною лінією, бо насправді вхідний вектор активізує своїми значеннями вихідний вектор $Y2(y_{21}, y_{22}, \dots, y_{2m}) = Y1(y_{11}, y_{12}, \dots, y_{1m})$, після чого початковий сигнал виходу $Y2(y_{21}, y_{22}, \dots, y_{2m}) = Y1(y_{11}, y_{12}, \dots, y_{1m})$ перераховується в процесі рекурентних розрахунків кілька разів до виділення з них лише одного елемента – переможця з найбільшим позитивним значенням (усі інші елементи набувають від'ємного значення NET і після проходження активаційних функцій на виході MAXNET отримують нульові значення).

НМ MAXNET має m входів для кожного з m нейронів $N_{21} \dots N_{2m}$ та m виходів (але можна вважати, що окремих входів $Y1$ не існує, оскільки НМ ініціюється початковими значеннями виходу $Y2$). Нейрони не мають постійного сигналу вільного члена $\{y_{10}=+1, w_{10}\}$ і пов'язані між собою **від'ємними зворотними зв'язками** через свої входи з однаковими вагами « $-\xi$ » (ці від'ємні зв'язки показані пунктирними лініями). Разом з тим кожний нейрон має **один вхід із власним додатнім зворотнім зв'язком** з вагою « $+1$ »: на рис. 10.5.1, а вони всі показані жирними лініями. Таким чином, усі вагові коефіцієнти входів з від'ємними зворотними зв'язками дорівнюють постійній величині $|\xi| = const$, де $0 < |\xi| < 1/m$, а вага зв'язку, показаного жирною лінією і пов'язаного з входом власного нейрона, дорівнює « $+1$ ».

MAXNET подавляє усі сигнали вектору виходу, крім найбільш потуж-

ного (використовується принцип «переможець забирає все»). В результаті з усіх нейронів $N_{21} \dots N_{2m}$ буде активований лише один нейрон, з максимальним додатним значенням виходу NET . Якщо усі елементи вхідного вектора є від'ємними, то мережа буде мати на виході лише нульові значення.

Активаційна функція усіх нейронів є однаковою і наведена на рис. 10.5.1, б.

Якщо ми маємо вхідні сигнали нейронів $N_{21} - N_{2m}$, серед яких потрібно виділити максимальний елемент, то *починається циклічний рекурентний ітераційний процес перерахунків виходів нейронів* MAXNET. Він завданий початковим станом нейронів $N_{21} - N_{2m}$ і продовжується до стабілізації виходів $y_{21} - y_{2m}$, коли всі нейрони, крім одного переможця, не дають на виході значення, рівні нулю.

Розрахунки повторюється до отримання лише одного переможця. Далі перевіряється, чи змінились виходи нейронів за останню ітерацію. Якщо змінились – продовжують розрахунки, а у іншому випадку – їх завершують.

10.6. Нейрон для введення формул

Нейрон «Формула» (Україна, 2016, ЧДУ [77]) має входи з ваговими коефіцієнтами 1, до яких приєднані вхідні змінні $x_1, x_2, \dots, x_j, \dots, x_n$. Операторний блок N вміщує довільну сукупність заданих без навчання вихідних лінійних чи нелінійних функцій

$$y_i = F_i(x_1, x_2, \dots, x_j, \dots, x_n) + A_i,$$

де A_i – вільний член функції y_i , $i = 1, 2, \dots, m$ – порядковий номер функції y_i (рис. 10.6.1).

Подальше збільшення функціональних можливостей нейрону «Формула» досягається тим, що всі або частка функцій y_i мають вихідні активаційні функції (на рис. 10.6.1 не показані).

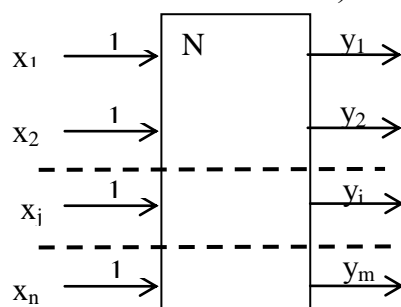


Рис. 10.6.1. Нейрон «Формула»

Нейрон «Формула»: не потребує навчання; використовує точні задані експертом формули виходу, що усуває похибки апроксимації; збільшує функціональні можливості через введення в операторний блок заданої сукупності функцій виходу. Подальше збільшення функціональних можливостей нейрона «Формула» досягається можливістю введення для всіх ви-

ходів (або їх частки) вихідних активаційних функцій.

10.7. Нейронна мережа для стиснення інформації «КОД n:m:n»

НМ складається з трьох шарів: А – сенсорного шару з n нейронами, В – прихованого шару каналу зв'язку з кількістю нейронів $m = \log_2 n$, С – вихідного шару з n нейронами. На вхід НМ мережі подається n – мірний вектор, який містить лише один одиничний елемент при всіх інших нульових елементах. НМ навчають кодувати n – розрядний вхідний сигнал у $m = \log_2 n$ – розрядний, а потім знову його декодувати у n – розрядний вихідний сигнал. Таким чином, розглянутий тришаровий персептрон повинен вирішувати проблему 8-3-8 [182]. Ця НМ описана також в роботі [103, с. 86–87]. Недоліками даної НМ є:

- наявність помилок, які склалися для прототипу 8 % після завершення навчання;
- необхідність навчання (для навчання прототипу при стисненні 8 вхідних розрядів у 3 розряди каналу зв'язку було застосовано 3150 подаєних векторів);
- збільшення на один нейрон кількості нейронів у вхідному сенсорному шарі.

Аналогічна НМ, призначена для стиснення інформації у каналі зв'язку В шляхом кодування та декодування інформації на вході А і виході С наведена на рис. 10.7.1 [75].

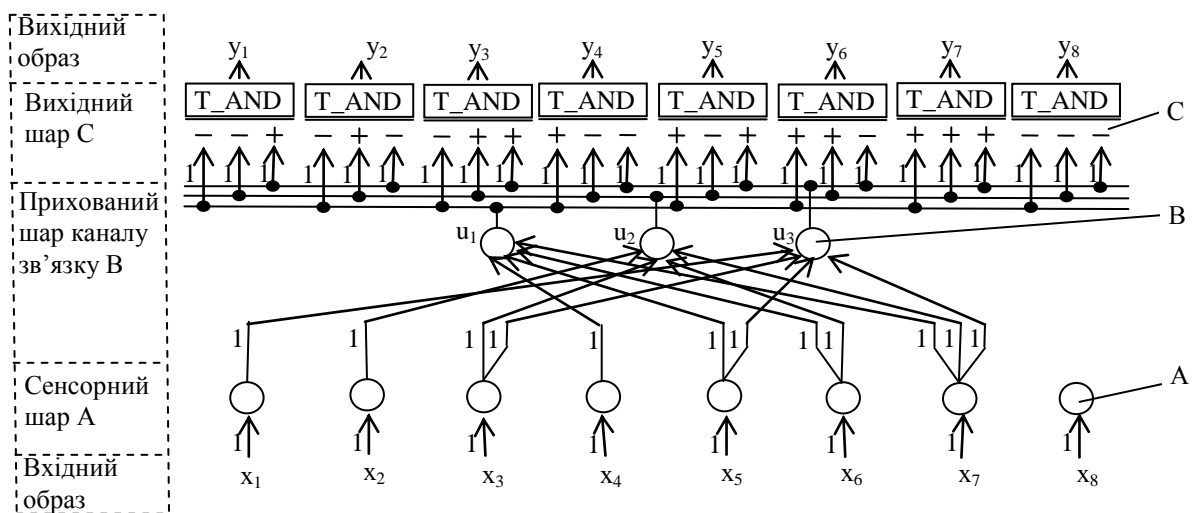


Рис. 10.7.1. Нейронна мережа для стиснення інформації «КОД n:m:n»

На рис. 10.7.1 використані позначення:

- $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$ – вхідний вектор з $n = 8$ елементів, який вводиться у сенсорний шар з n нейронів і складається з «0» та містить лише одну «1», що зміщується вздовж $n = 8$ елементів від початку до кінця у

залежності від порядкового номера n вхідних векторів. З фігури видно, що 8-й нейрон сенсорного шару не використовується і тому цей нейрон може не застосовуватись в реальній мережі.

– А – вхідний сенсорний шар НМ, у який треба вводити $n = 8$ елементів вхідного вектору $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$, що підлягають стисненню. У дійсності вводиться лише 7 елементів вектора (скорочення елементів вхідного вектору пояснюється тим, що відсутність сигналу «1» у всіх перших 7-ми елементах вхідного вектору $(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ означає наявність елементу $x_8=1$). Тому вхідний сенсорний шар складається з 7-ми діючих нейронів.

– В – захований шар НМ, який складається з $m = \log_2 n = \log_2 8 = 3$ нейронів (u_1, u_2, u_3) з ваговими коефіцієнтами входів, рівними «1».

– С – вихідний шар НМ, який складається з $n = 8$ нейронів «Т_AND» типу «Т_AND_0», що видає на виході «1», якщо задана частка вхідних сигналів «+» дорівнює «1», а інша задана частка вхідних сигналів «-» дорівнює «0» [63].

– «+», «-» – знаки, які вказують для нейронів типу «Т_AND_0» задану кількість вхідних сигналів, рівних «1» (на це вказує кількість знаків «+»), та задану кількість вхідних сигналів, рівних «0» (на це вказує кількість знаків «-»).

– $(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)$ – вихідний декодований вектор у вигляді $n = 8$ елементів, які повинні співпасти з елементами вхідного вектора $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$.

– «1» – вагові коефіцієнти зв'язків між нейронами.

До сенсорного шару А нейронної мережі увімкнені $n = 8$ елементів вхідного вектору $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$, які складаються з «0» та містять лише одну «1».

Всі вагові коефіцієнти входів нейронів НМ мають значення «1» і не змінюються.

У табл. 10.7.1 наведені значення елементів вхідного та вихідного вектору і m елементів виходу нейронів прихованого шару.

З табл. 10.7.1 видно, що m виходів прихованого шару утворюють двоїсту нумерацію положення елемента «1» вхідних векторів (або порядкових номерів векторів), яку повинні розшифрувати нейрони «Т_AND_0» вихідного шару.

Розглянемо роботу НМ при передачі сигналів $j = 1, 3, 8$.

Для вхідного вектора $j = 1$ у вигляді $n = 8$ елементів $(x_1=1, x_2=0, x_3=0, x_4=0, x_5=0, x_6=0, x_7=0, x_8=0)$, які складаються з «0» та містять лише одну «1», на виходах нейронів прихованого шару (u_1, u_2, u_3) при $m = \log_2 n = \log_2 8 = 3$ повинні отримати «001». Тому вихід відповідного першого нейрону сенсорного шару з'єднується з входом нейрону u_3 прихованого шару відповідно номеру «001», а два перших нульових зв'язка з номера «001» на схемі НМ не відображуються.

Таблиця 10.7.1.

**Значення $n=8$ елементів вхідного вектору ($x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$),
 $m = \log_2 n = 3$ елементів виходу нейронів прихованого шару (u_1, u_2, u_3)
та $n=8$ елементів вихідного вектору ($y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8$)**

Порядковий номер вхідного вектору $j = 1, 2, \dots, n;$ $n = 8.$	Значення входу елементів вхідного вектору ($x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$)	Виходи прихованого шару у кількості $m = \log_2 n = 3$ (u_1, u_2, u_3)			Значення виходу елементів вихідного вектору ($y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8$)
		u_1	u_2	u_3	
1	10000000	0	0	1	10000000
2	01000000	0	1	0	01000000
3	00100000	0	1	1	00100000
4	00010000	1	0	0	00010000
5	00001000	1	0	1	00001000
6	00000100	1	1	0	00000100
7	00000010	1	1	1	00000010
8	00000001	0	0	0	00000001

В результаті на виходах нейронів прихованого шару (u_1, u_2, u_3) утворюється сигнал «001», на який спрацьовує лише один нейрон вихідного шару «T_AND_0» і видає на виході $y_1=1$. Усі інші виходи нейронів вихідного шару «T_AND_0» дорівнюють нулю: $y_2=0, y_3=0, y_4=0, y_5=0, y_6=0, y_7=0, y_8=0$. Тобто на виході НМ ми отримуємо вектор($y_1=1, y_2=0, y_3=0, y_4=0, y_5=0, y_6=0, y_7=0, y_8=0$), який точно відповідає вхідному вектору ($x_1=1, x_2=0, x_3=0, x_4=0, x_5=0, x_6=0, x_7=0, x_8=0$).

Для вхідного вектора $j = 3$ у вигляді ($x_1=0, x_2=0, x_4=0, x_5=0, x_6=0, x_7=0, x_8=0$) на виходах нейронів прихованого шару (u_1, u_2, u_3) ми повинні отримати значення «011». Тому вихід відповідного третього нейрону сенсорного шару $x_3=1$ з'єднується з входами нейронів u_2 та u_3 прихованого шару. Перший зв'язок з номера «011» зі значенням «0» на схемі НМ не відображуються.

В результаті на виходах нейронів прихованого шару (u_1, u_2, u_3) утворюється сигнал «011», на який спрацьовує нейрон вихідного шару «T_AND_0» і видає на виході $y_3=1$. Усі інші виходи нейронів вихідного шару «T_AND_0» дорівнюють нулю. Тобто на виході НМ ми отримуємо вектор($y_1=0, y_2=0, y_3=1, y_4=0, y_5=0, y_6=0, y_7=0, y_8=0$), який точно відповідає вхідному вектору ($x_1=0, x_2=0, x_3=1, x_4=0, x_5=0, x_6=0, x_7=0, x_8=0$).

Для вхідного вектора $j = 8$ на виходах нейронів прихованого шару (u_1, u_2, u_3) ми повинні отримати значення «000» (з одиницею переходу у старший розряд). В результаті отримані три нульові виходи $u_1=0, u_2=0, u_3=0$ для одиниці переходу у старший розряд. Тому 8-й нейрон сенсорного шару не використовується і показаний умовно з метою пояснення роботи НМ.

В результаті на виходах нейронів прихованого шару (u_1, u_2, u_3) утворюється сигнал «000», на який спрацьовує нейрон вихідного шару «T_AND_0» і видає на виході $y_8=1$. Усі інші виходи нейронів вихідного

шару «T_AND_0» дорівнюють нулю. Тобто на виході НМ ми отримуємо вектор($y_1=0, y_2=0, y_3=0, y_4=0, y_5=0, y_6=0, y_7=0, y_8=1$), який точно відповідає вхідному вектору ($x_1=0, x_2=0, x_3=0, x_4=0, x_5=0, x_6=0, x_7=0, x_8=1$).

Аналогічним чином нейронна мережа буде працювати при подачі інших вхідних векторів.

10.8. Нейронні мережі для виведення екстремального елемента з вхідного вектора

В математичному аналізі поняття максимуму та мінімуму об'єднуються одним словом «екстремум» (крайній). Недоліком мережі MAXNET є складність і збільшений час рекурсивних розрахунків та виділення не самого числа – переможця, а лише позначення (маркування) переможця.

Відома також НМ – компаратор, яка виділяє вже число – переможця, але у неї збільшена на 30 % кількість нейронів [103, с. 102–106; 207].

Нижче на конкретних прикладах розглядається робота розроблених в Україні схем нейронних мереж «MAX_0/1», «M_MAX_0/1», «MIN_0/1», «M_MIN_0/1», «MAX_2» та «MIN_2» [64–69], які усувають недоліки згаданих мереж. Останні цифри «_0/1» в назвах НМ визначають базові величини, прийняті у розрахунках активаційних функцій.

НМ MAX_0 наведена на рис. 10.8.1.

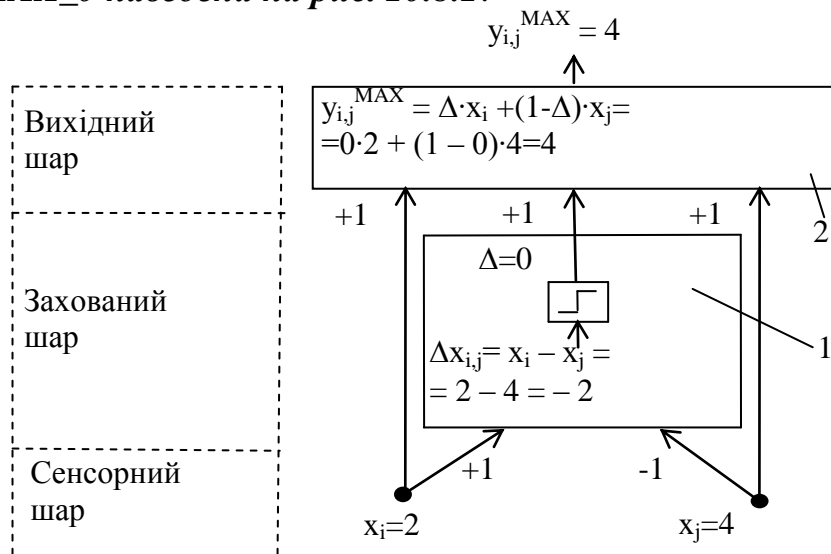


Рис. 10.8.1. Схема нейронної мережі MAX_0

На входи НМ MAX_0 подані значення двох вхідних змінних ($x_i=2, x_j=4$).

Якщо вирішальна функція нейрону 1 у вигляді різниці між вхідними змінними $\Delta x_{i,j} = x_i - x_j = 2 - 4 = -2$ має від'ємне значення, то це вказує що більше значення має x_j , яке й потрібно перевести у «переможця» при пороговій функції активації $\Delta = f_1(\Delta x_{i,j}) = f(-2) = 0$ нейрону 1.

Нейрон 2 не має активаційної функції і видає на власний вихід значення найбільшої вхідної змінної $x_j=4$, розрахованої за вирішальною функцією $y_{i,j}^{MAX} = \Delta \cdot x_i + (1-\Delta) \cdot x_j = 0 \cdot 2 + (1 - 0) \cdot 4 = 4$.

Нейронна мережа виконує свої функції при розгляді як позитивних так і негативних вхідних змінних.

На рис. 10.8.2 показано спрощене зображення нейронної мережі MAX_0 без вказівки внутрішньої структури та значень вагових коефіцієнтів.

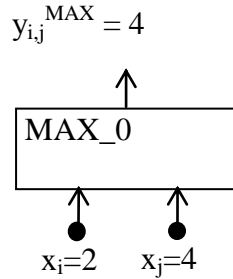


Рис. 10.8.2. Спрощене зображення нейронної мережі MAX_0

На рис. 10.8.3 показана складна нейронна мережа M_MAX_0, призначена для фільтрації максимального числа серед ряду вхідних чисел ($x_1=2, x_2=4, x_3=1, x_4=7, x_5=9, x_6=3, x_7=5$), у якій використовується спрощене зображення нейронної мережі MAX_0 за рис. 10.8.2. З рис. 10.8.3 випливає, що загальна кількість нейронних мереж MAX_0 дорівнює $(n-1)$, де n – кількість вхідних змінних сенсорного шару.

При подачі на сенсорний шар ряду вхідних чисел ($x_1=2, x_2=4, x_3=1, x_4=7, x_5=9, x_6=3, x_7=5$) мережа M_MAX_0 попарно порівнює вхідні змінні і обирає з них найбільше $y_{1-7}^{MAX} = x_5 = 9$.

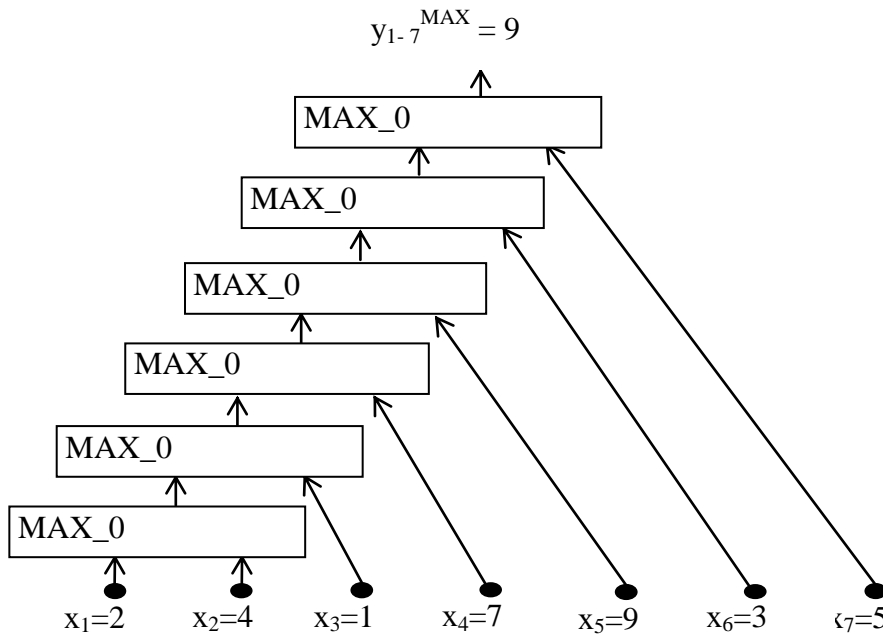


Рис. 10.8.3. Складна нейронна мережа M_MAX_0 по фільтрації максимального числа серед ряду вхідних чисел

На входи НМ *MIN_0* (рис. 10.8.4) подані значення двох вхідних змінних ($x_i=2$, $x_j=4$).

Якщо вирішальна функція нейрону 1 у вигляді різниці між вхідними змінними $\Delta x_{i,j} = x_i - x_j = 2 - 4 = -2$ має від'ємне значення, то це вказує що менше значення має x_i , яке й потрібно перевести у «переможця» при порогової функції активації $\Delta = f_1(\Delta x_{i,j}) = f_1(-2) = 0$ нейрону 1.

Нейрон 2 не має активаційної функції і видає на власний вихід значення найменшої вхідної змінної $x_j=4$, розрахованої за вирішальною функцією $y_{i,j}^{MIN} = (1-\Delta) \cdot x_i + \Delta \cdot x_j = (1-0) \cdot 2 + 0 \cdot 4 = 2$.

Нейронна мережа виконує свої функції при розгляді як позитивних так і негативних вхідних змінних.

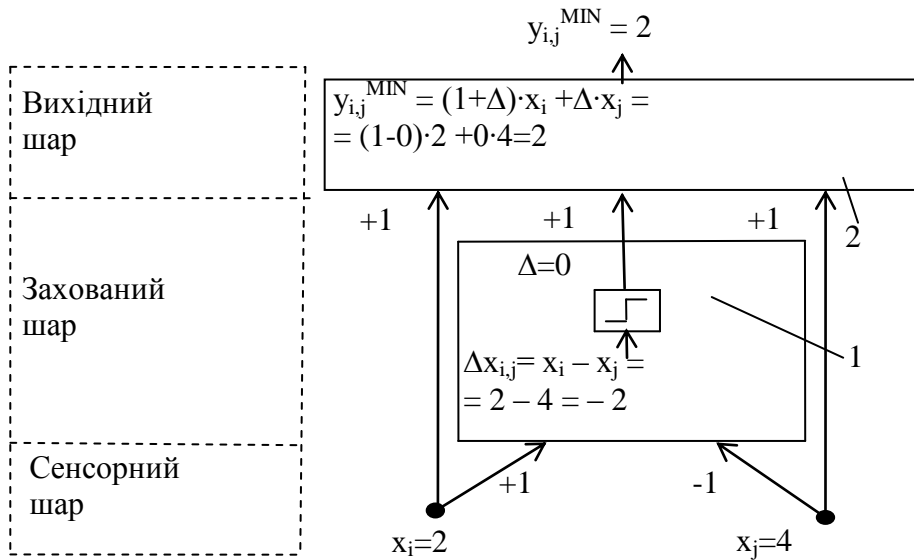


Рис. 10.8.4. Схема нейронної мережі *MIN_0*

На рис. 10.8.5 показане спрощене зображення нейронної мережі *MIN_0* без вказівки внутрішньої структури та значень вагових коефіцієнтів.

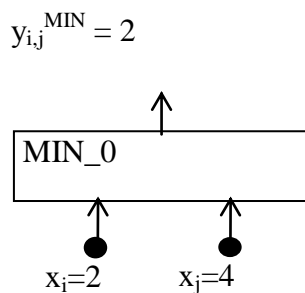


Рис. 10.8.5. Спрощене зображення нейронної мережі *MIN_0*

На рис. 10.8.6 показана складна нейронна мережа *M_MIN_0*, призначена для фільтрації мінімального числа серед ряду вхідних чисел ($x_1=2$, $x_2=4$, $x_3=1$, $x_4=7$, $x_5=9$, $x_6=3$, $x_7=5$), у якій використовується спрощене зображення нейронної мережі *MIN_0* за рис. 10.8.5. З рис. 10.8.6 випливає,

що загальна кількість нейронних мереж MIN_0 дорівнює $(n-1)$, де n – кількість вхідних змінних сенсорного шару.

При подачі на сенсорний шар ряду вхідних чисел ($x_1=2, x_2=4, x_3=1, x_4=7, x_5=9, x_6=3, x_7=5$) мережа M_MIN_0 попарно порівнює вхідні змінні і обирає з них найменше $y_{1-7}^{\text{MIN}} = x_3 = 1$.

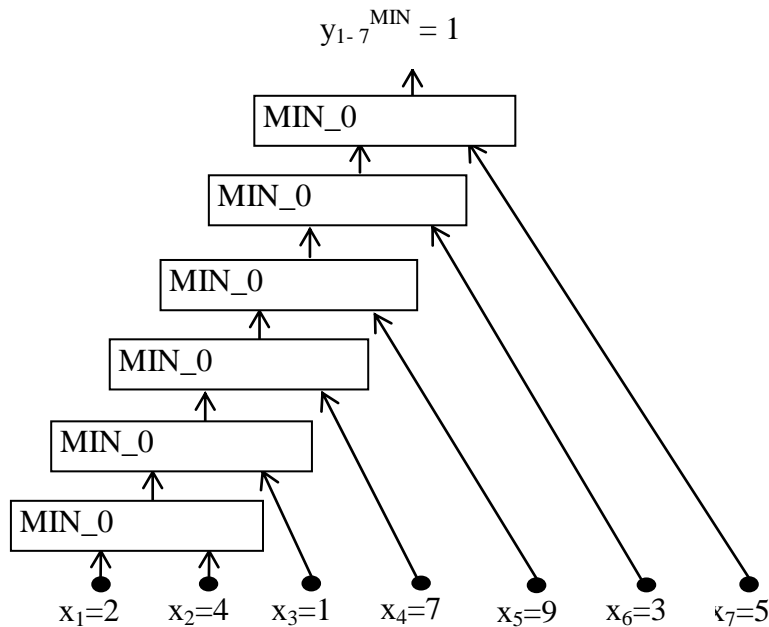


Рис. 10.8.6. Складна нейронна мережа M_MIN_0 по фільтрації мінімального числа серед ряду вхідних чисел

Ми розглянули роботу нейронних мереж «MAX_0/1», «M_MAX_0», «MIN_0», «M_MIN_0».

Нейронні мережі «MAX_1», «M_MAX_1», «MIN_1», «M_MIN_1» працюють аналогічним чином і відрізняються лише виглядом логічної активаційної функції.

Наприклад, нейронна мережа «MAX_1» при даних схеми рис. 10.8.1 (вхідних змінних ($x_i=2, x_j=4$); різниці між ними $\Delta x_{i,j} = x_i - x_j = 2 - 4 = -2$) має іншу порогову функцію активації $\Delta = f_1(\Delta x_{i,j}) = f(-2) = -1$ нейрону 1 та відрізняється вирішальною функцією для розрахунку виходу $y_{i,j}^{\text{MAX}} = 0,5(1+\Delta) \cdot x_i + 0,5(1-\Delta) \cdot x_j = 0,5(1-1) \cdot 2 + 0,5(1+1) \cdot 4 = 4$. Інші НМ «M_MAX_1», «MIN_1», «M_MIN_1» відрізняються аналогічним чином.

Нижче розглядаються нейрони «MAX_2» та «MIN_2» для виведення з вхідного вектора екстремального елемента.

Нейрони «MAX_2» та «MIN_2» аналогічні за дією нейронам «T_MAX» та «T_MIN», але мають у якості виходу не вектор, а один екстремальний елемент вхідного вектора [68; 69].

Структурна схема нейронів «MAX_2» та «MIN_2» ілюструється рис. 10.8.7 при використанні наступних позначень:

- $X = (x_1, x_2, \dots, x_j, \dots, x_n)$ – вхідний вектор нейрону;

- Y – вихід нейрону;
- N – нейрон («MAX_2» або «MIN_2»), який складається лише з блоку активаційної функції.

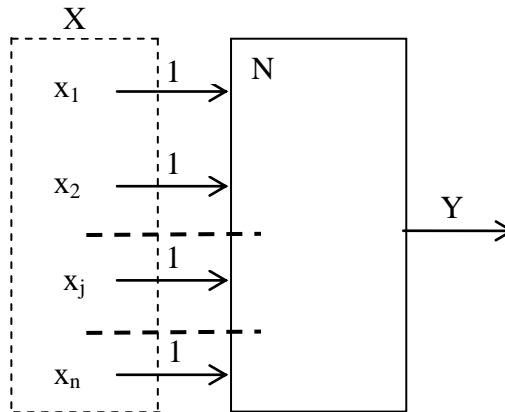


Рис. 10.8.7. Структурна схема нейрона «MAX_2» або «MIN_2»

При подачі на входи нейрону MAX_2 вхідного вектора $X = (x_1, x_2, \dots, x_j, \dots, x_n)$ нейрон використовує блок активаційної функції, який призначений для виконання операцій:

1. Для виведення елемента з максимальним значенням
 $XM := x(1)$; for $j := 2$ to n do if $x(j) > XM$ then $XM := x(j)$; $Y := XM$;
2. Для виведення елемента з мінімальним значенням
 $XM := x(1)$; for $j := 2$ to n do if $x(j) < XM$ then $XM := x(j)$; $Y := XM$;

10.9. Нейрони «T_MAX» та «T_MIN» для двоїстого кодування елементів вхідного вектора з виділенням серед них екстремального значення

Нейрони «T_MAX» та «T_MIN» виконують функції, аналогічні нейронній мережі MAXNET (рис. 10.3.1). Вони не потребують навчання, дозволяють зменшити кількість нейронів та скоротити час на обслуговування і налагодження нейронної мережі внаслідок спрощення розрахунків [70; 71].

Структурна схема нейронів «T_MAX» або «T_MIN» наведена на рис. 10.9.1 при використанні наступних позначень:

- $X = (x_1, x_2, \dots, x_j, \dots, x_n)$ – вхідний вектор нейрону;
- $Y = (y_1, y_2, \dots, y_j, \dots, y_n)$ – вихідний вектор нейрону;
- N – нейрон («T_MAX» або «T_MIN»), який складається лише з блоку активаційної функції.

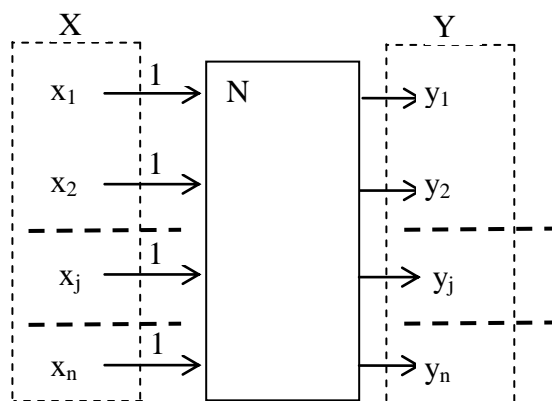


Рис. 10.9.1. Структурна схема нейрона T_MAX або T_MIN

При подачі на входи нейрону T_MAX вхідного вектора $X = (x_1, x_2, \dots, x_j, \dots, x_n)$ нейрон використовує блок активаційної функції, який призначений для виконання операцій

```
for j:=1 to n do y(j):=0;
XM:= x(1); for j:=2 to n do if x(j) > XM then XM:= x(j);
for j:=1 to n do if x(j) = XM then y(j):= 1;
```

Останній рядок логічної операції можна замінити на «for j:=1 to n do if x(j) = XM then y(j):= XM;».

В результаті на виході нейрону отримуємо вихідний вектор $Y = (y_1, y_2, \dots, y_j, \dots, y_n)$, кожний елемент y_j якого дорівнює «0», якщо відповідне значення елемента x_j вхідного вектора менше максимального значення XM елементів вхідного вектора X та дорівнює «1» (або XM), якщо значення елемента x_j вхідного вектора дорівнює максимальному значенню XM елементів вхідного вектора X.

Нейрон T_MIN для обробки даних вхідного вектора $X = (x_1, x_2, \dots, x_j, \dots, x_n)$ використовує блок активаційної функції, який виконує операції

```
for j:=1 to n do y(j):=0;
XM:= x(1); for j:=2 to n do if x(j) < XM then XM:= x(j);
for j:=1 to n do if x(j) = XM then y(j):= 1;
```

Останній рядок логічної операції можна замінити на «for j:=1 to n do if x(j) = XM then y(j):= XM;».

В результаті на виході нейрону отримуємо вихідний вектор $Y = (y_1, y_2, \dots, y_j, \dots, y_n)$, кожний елемент y_j якого дорівнює «0», якщо відповідне значення елемента x_j вхідного вектора більше мінімального значення XM елементів вхідного вектора X та дорівнює «1» (або XM), якщо значення елемента x_j вхідного вектора дорівнює мінімальному значенню XM елементів вхідного вектора X.

Завдання

Завдання 1. Програмна реалізація простих функцій нейронами. Отримати еталонний вектор з 5-ти елементів ± 1 , якій відображує у двійко-

вій системі числення порядковий номер N студента у групі (при цьому «-1» вважати нулем). Наприклад, для студента з $N=1=\{00001\}$ величина $X^{E1} = \{-1,-1,-1,-1,+1\}$; для $N=10=\{01010\}$ величина $X^{E1} = \{-1,+1,-1,+1,-1\}$.

По відношенню до еталонного вектору X^{E1} скласти три додаткові «зіпсовані» вектори тієї ж розмірності: перший з них відрізняється від еталонного одним довільним елементом, другий – двома елементами, а третій має прямо протилежні елементи.

За допомогою нейронної мережі знайти для цих трьох векторів:

– Міру близькості і вигляді скалярного добутку векторів з еталонним вектором.

– Відстань Хеммінга трьох векторів від еталонного вектора.

Скласти програмну реалізацію для розрахунку скалярного добутку та відстані Хеммінга.

Завдання 2. Програмна реалізація простих функцій нейронами.

В отриманому еталонному векторі серед усіх значень $+1$ залишити лише одне рівним $+0,8$, а абсолютні значення усіх інших елементів без зміни їх знаків прийняти рівними $\|\pm 0,2\|$. За допомогою нейронної мережі *MAXNET* на будь-якій мові програмування виділити максимальний елемент.

Завдання 3. Програмна реалізація простих функцій нейронами.

Задати довільну криву $f(x)$ з амплітудою, рівною N (N – порядковий номер студента у групі) або у вигляді аналітичної залежності, або у вигляді таблиці. Навчити нейронну мережу з радіальними базовими функціями у програмній реалізації давати на виході криву $f(x)$ у припущенні, що ми апроксимуємо 10 точок. При цьому використовувати дельта-правило навчання (закон Відроу-Хоффа). Навести схему НМ, ідеальний початковий графік цієї функції та відповідний графік НМ.

10.10. Стохастичний нейрон

Стохастичний нейрон (рис. 10.10.1), розглянутий в роботі [157], є основою машини Больцмана та машини Коші а також використовується при навчанні з підкріпленням [103].

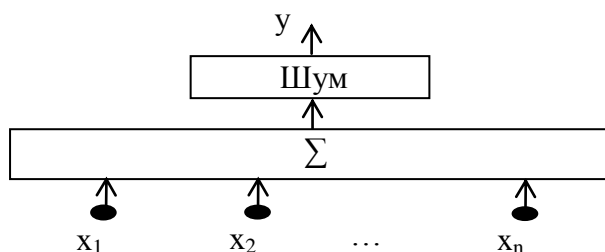


Рис. 10.10.1. Стохастичний нейрон

На відміну від детермінованого нейрона з однозначним виходом, вихідний сигнал стохастичного нейрона може бути або $+1$ або -1 і визнача-

ється одночасно з ймовірністю його появи, яка розраховується активаційною функцією

$$y(u) = \begin{cases} +1 & \text{з ймовірністю } P(u); \\ -1 & \text{з ймовірністю } 1 - P(u), \end{cases}$$

де u – сигнал на виході суматора нейрону; $P(u)$ – ймовірність отримання на виході нейрона $+1$, яка визначається активаційною функцією.

Активаційна функція нейрону може мати вигляд сигмоїди

$$P(u) = \frac{1}{1 + e^{-u/T}}$$

де T – аналог температури (насправді T довільно задається програмістом і характеризує хаотичність виходу нейрону у залежності від випадкового значення T), який використовується для керування степені невизначеності між отримуваними значеннями $+1$ та -1 на виході нейрона.

Якщо $T \rightarrow 0$, то стохастичний нейрон приймає детерміновану форму нейрону із ступінчастою функцією активації

$$y(u) = \begin{cases} +1 & \text{якщо } u \geq 0; \\ -1 & \text{якщо } u < 0. \end{cases}$$

10.11. Динамічна модель нейрона Хопфілда

Нижче розглянутий нейрон Хопфілда, який є динамічною моделлю біологічного нейрону [161].

На рис. 10.11.1 показана електрична схема нейрона Хопфілда [161], яка має: вектор вхідних змінних $X = (x_1, x_2, \dots, x_j, \dots, x_n)$, які вважаються рівними електричним напругам; резистори ($R_1, R_2, \dots, R_j, \dots, R_n$), R_0 ; ємність C_0 ; джерело струму I_0 ; нелінійний підсилювач з сигмоїдальною вихідною функцією $f(u)$. Вважається, що всі резистори мають позитивне значення, а позитивний чи негативний вплив входу отримується за рахунок знаку вхідних змінних $X = (x_1, x_2, \dots, x_j, \dots, x_n)$. Стрілками показані напрямки струмів. Операційний підсилювач має два асиметричних виходи.

Згідно закону Кірхгофа про підсумок струмів у вузлі отримуємо рівняння для вузла входу у підсилювач (вхідний струм підсилювача вважаємо рівним нулю внаслідок великого коефіцієнта підсилювання операційного підсилювача)

$$-C \frac{du}{dt} + I_0 - \frac{u}{R_0} + \sum_{j=1}^n \frac{x_j}{R_j} = 0 \tag{10.11.1}$$

Тоді з рівняння (10.11.1) отримуємо більш загальну форму

$$-\frac{du}{dt} - w_0 u + \sum_{j=0}^n w_j x_j = 0; \tag{10.11.2}$$

$$y = f(u), \quad (10.11.3)$$

де $w_U = 1/(CR_0)$; $w_0 = 1/C$; $x_0 = I_0$; $w_j = 1/(CR_j)$ при $j = 1, 2, \dots, n$.

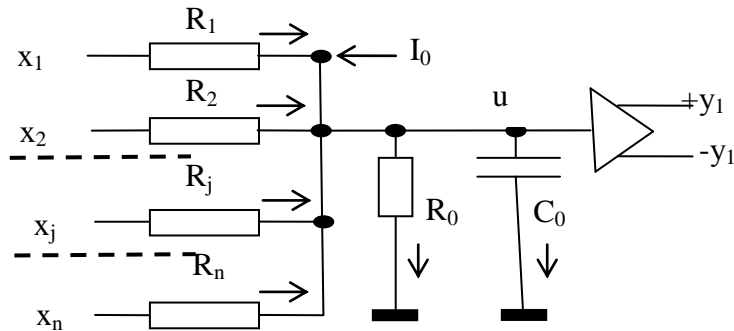


Рис. 10.11.1. Електрична схема нейрона Хопфілда

З рівнянь (10.11.2) та (10.11.3) отримуємо динамічний нейрон Хопфілда рис. 10.11.2, який аналізує безперервні дані, де N_1 – нейрон-суматор; N_2 – нейрон – інтегратор; $f(u)$ – вихідний нелінійний елемент з сигмоїдальною активаційною функцією.

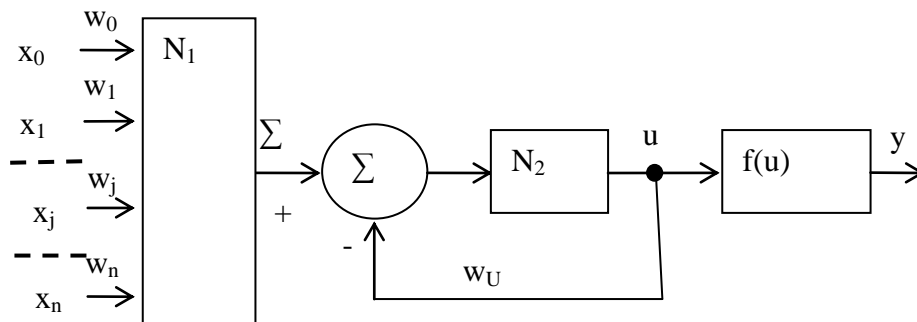


Рис. 10.11.2. Динамічний нейрон Хопфілда, який аналізує безперервні дані

Існує також дискретний динамічний нейрон Хопфілда, який замість нейрону N_2 виконує «інтегрування» підсумовуванням вхідних даних.

11. Приклади нейронних мереж

11.1. Топологія нейронних мереж

Структура НМ задається у вигляді графа, у якому кожний нейрон з довільною кількістю входів та виходів є вершиною графа, а входи і виходи нейрона являють собою ребра (зв'язки). Зв'язки можуть бути одно- та дво-спрямованими.

Звичайно НМ складається з кількох шарів, кожний з яких може мати довільну кількість нейронів. Серед цих шарів розрізняють:

– *Вхідний шар* (сенсорний): його нейрони отримують на свої входи сигнали ознак об'єктів чи процесів і передають їх своїми виходами на нейрони інших шарів.

– *Вихідний шар* (реагуючий): його нейрони відображують реакцію НМ на вхідні ознаки. Звичайно це є інформація щодо класифікації об'єкта по ознаках входу, тобто є рішенням про виконання деякої дії.

– *Приховані шари* (асоціативні, в багатошарових НМ)у довільній кількості: вони розміщуються між вхідним та вихідним шарами і виконують функцію проміжного логічного аналізу вхідних ознак. Вважається, що чим більше шарів має нейрон і чим більше є нейронів в шарах, тим складніші проблеми може розв'язувати НМ.

На рис. 11.1.1 показаний *перцептрон* – НМ з прямим розповсюдженням сигналів від вхідного до вихідного шару, який уперше був описаний Розенблатом. Зв'язки між нейронами задаються у вигляді векторів і матриць.

Вектори вхідних та вихідних сигналів нейронів позначені ($X[n_S:1]$, $X^*[n_S:1]$, $U[n_A:1]$, $Z[n_R:1]$, $Y[n_R:1]$), а вагові матриці як ($W_{UX}[n_A:n_S]$, $W_{ZU}[n_R:n_A]$): у квадратних дужках перед двокрапкою «:» вказана кількість рядків, а після двокрапки – кількість колонок. У вагових матрицях відсутній елемент зв'язку помічається як «0».

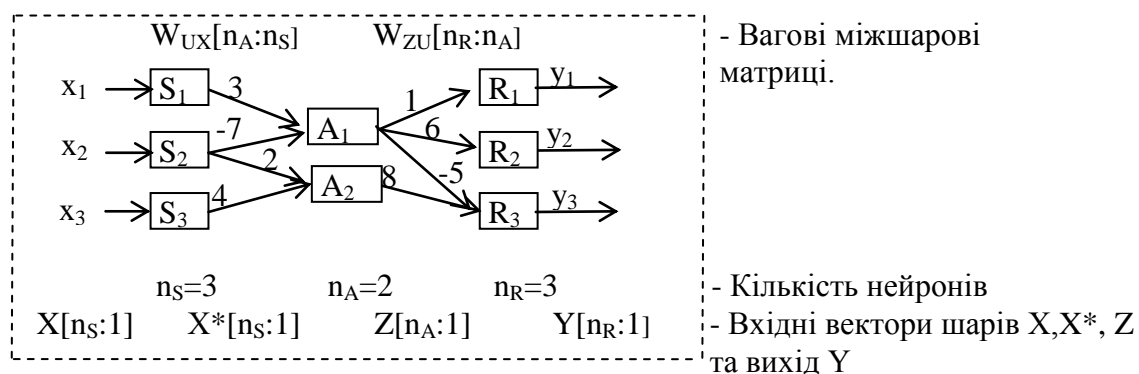


Рис. 11.1.1. Структура перцептрона

Кількість рядків n_S вхідного вектора $X[n_S:1]$ дорівнює кількості ознак об'єкта, який розпізнається. Сенсорний шар S з нейронів $S_1 - S_3$ призначений для нормалізації і розмноження вхідного сигналу $X[n_S:1]$. Вважаємо, що вхідний вектор $X[n_S:1]$ перетворюється сенсорним шаром S на нормалізований вектор $X^*[n_S:1]$, який і поступає в НМ. Нормалізація вхідного вектора $X[n_S:1]$ означає приведення його елементів до нормального вигляду, коли їх числові значення у відносних одиницях змінюються в однакових межах $0 \dots 1$: це запобігає збільшенню впливу на рішення НМ ознак з великими числовими значеннями. Внаслідок цієї особливості роботи сенсорний шар S іноді не ураховується як шар НМ (відносять до датчиків).

Кількість рядків векторів X^*, Z, Y дорівнює кількості нейронів, з яких вектор виходить:

$$X[n_S,1] = X[3,1] = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}; \quad X^*[n_S,1] = X^*[3,1] = \begin{bmatrix} x_1^* \\ x_2^* \\ x_3^* \end{bmatrix}; \quad Z[n_A,1] = Z[2,1] = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}; \quad Y[n_R,1] = Y[3,1] = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}.$$

Елементи вагових матриць $W_{UX}[n_A:n_S]$ та $W_{ZU}[n_R:n_A]$ згідно рис. 11.1.1 дорівнюють (тут цифри наведені як приклади):

$$W_{UX}[n_A, n_S] = W[2,3] = \begin{bmatrix} 3 & -7 & 0 \\ 0 & 2 & 4 \end{bmatrix}; \quad W_{ZU}[n_R, n_A] = W[3,2] = \begin{bmatrix} 1 & 0 \\ 6 & 0 \\ -5 & 8 \end{bmatrix},$$

де нульові значення елементів вказують на відсутність зв'язку між нейронами НМ.

Для шарів A та R значення виходів нейронів розглядаються як вектори $NET_A[n_A:1]$ та $NET_R[n_R:1]$, які розраховуються за матричними виразами

$$NET_A[n_A:1] = W_{UX}[n_A, n_S] \cdot X[n_S,1]; \quad NET_R[n_R:1] = W_{ZU}[n_R, n_A] \cdot Z[n_A,1].$$

За допомогою нелінійних активаційних функцій на основі значень елементів $NET_A[n_A:1]$ та $NET_R[n_R:1]$ для нейронів шарів A та R розраховуються їх вихідні вектори $Z[n_A:1]$ та $Y[n_R:1]$.

Крім НМ типу персептрону (рис. 11.1.1) існують інші НМ:

- **повнозв'язані НМ**, в яких кожний нейрон має зв'язки з усіма іншими нейронами;

- **НМ із зворотними зв'язками**, в яких виходи нейронів даного шару використовуються як входи нейронів даного шару та входи до нейронів попередніх шарів;

- **НМ з латеральними зв'язками**, які зв'язують нейрони одного шару із сусідніми нейронами цього шару та нейронами сусідніх шарів; для конкретного нейрону зв'язки можуть бути збуджуючими та гальмуючими.

11.2. Асоціативна пам'ять

11.2.1. Нейронна мережа Хопфілда

Асоціативним зветься такий умовивід, за яким по схожості ознак об'єктів різної фізичної природи (по зовнішньому вигляду, структурі, математичному опису, по схемі заміщення тощо) робиться висновок про асоціативну схожість самих об'єктів.

Пам'ять людини є асоціативною: наприклад, хмари можуть асоціюватися з будівлею чи іншим зображенням. НМ теж має асоціативну пам'ять і у відповідь на зіпсований вхідний сигнал на виході може дати виправлений сигнал.

У 1982 р. біофізик США Хопфілд вперше представив свою асоціативну мережу у Національній Академії Наук, і вона отримала його прізвище [161].

Мережа Хопфілда працює з n -мірними вхідними векторами X з елементами « ± 1 » (хоча існують мережі Хопфілда з елементами вектора входу «0» та «1»). Вона використовується для відновлення асоціативної пам'яті даних зображень, звукутощо. Мережа по частковій і зіпсованій інформації зашумленого вхідного сигналу визначає найближчий еталонний зразок і видає його значення на виході.

Структурна схема мережі Хопфілда наведена на рис. 11.2.1.

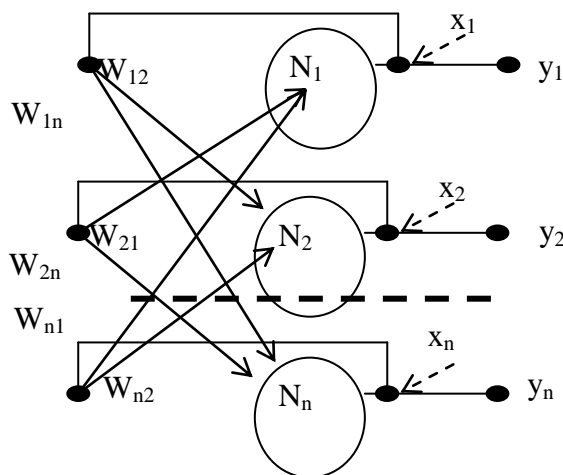


Рис. 11.2.1. Мережа Хопфілда

Мережа Хопфілда складається з одного шару нейронів N_1, N_2, \dots, N_n , кількість яких дорівнює кількості ознак x_1, x_2, \dots, x_n об'єктів. Кожний нейрон має порогову функцію активації. Вектори не мають окремого елемента для вільного члену. Вихід кожного одного нейрону з'єднується з входами усіх інших нейронів, але не з'єднується з входом власного нейрона.

На вхід нейронної мережі Хопфілда подають зіпсований n -мірний вхідний вектор X з елементами « ± 1 ». Цей вхідний сигнал запам'ятовують, після чого вхід X вимикають за рахунок виконання циклічних рекурсив-

них розрахунків (які змінюють X), на виході НМ отримують найближчий «найбільш схожий» до X еталонний n -мірний сигнал X^{Ei} , який запам'ятовується у вагах мережі. В вагах мережі Хопфілда можна запам'ятати не більше за $m \approx 0,15n$ еталонів, інакше сигнал виходу мережі може збігатись до нового не існуючого образу, або може взагалі не збігатись. Мережа може стати нестабільною, якщо навчальні приклади є занадто схожими. Ця проблема може бути вирішена вибором навчальних прикладів, які є більш ортогональними між собою.

Перед початком роботи експертпоеталонних сигналах входу тільки раз розраховує вагові коефіцієнти нейронів, і потім ваги не змінюються. Тому кажуть, що еталонні сигнали «записані в її вагах». А дехто стверджує, що ця мережа функціонує «без навчання» (бовони під навчанням розуміють не одноразове встановлення ваг людиною, а поступове уточнення їх величин).

Значення вагових коефіцієнтів розраховуються у вигляді квадратної матриці $W[n:n]$ по підсумку добутків n -вимірних еталонних векторів- $X_i^E (i=1, 2, \dots, m)$, які запам'ятовуються мережею Хопфілда

$$W[n:n] = \sum_{E=1}^m [X^E (X^E)^T] \quad (11.2.1)$$

де X^E – E -й еталонний вектор; $E=1, 2, \dots, m$ – порядковий номер еталонного вектора X^E ; m – загальна кількість еталонних векторів, які запам'ятовується (звичайно $m < 0,15n$, n – кількість елементів еталонних векторів X^E).

Елементи квадратної матриці $W[n:n]$ мають вигляд вагових коефіцієнтів зворотних зв'язків між нейронами $w_{ij} (i=1, 2, \dots, n$ – індекс нейрону, від якого отримується зворотній зв'язок; $j=1, 2, \dots, n$ – індекс нейрону, який отримує зворотній зв'язок; n – загальна кількість елементів еталонних векторів X^E) і визначаються наступним чином:

- Порядковий номер колонки симетричної квадратної матриці $W[n:n]$ дорівнює порядковому номеру $i=1, 2, \dots, n$ нейрону, від якого отримується зворотній зв'язок.

- Порядковий номер рядка симетричної квадратної матриці $W[n:n]$ дорівнює порядковому номеру $j=1, 2, \dots, n$ нейрону, який отримує зворотній зв'язок ($i \neq j$).

- Всі діагональні елементи w_{ii} матриці $W[n:n]$ прирівнюють нулю ($w_{ii}=0$).

- Всі рівні нулю діагональні елементи $w_{ii}=0$ та рівні нулю недіагональні елементи $w_{ij} = 0$ в мережі Хопфілда не вказуються і не використовуються.

Ваги нейронів є симетричними, тобто $w_{ij} = w_{ji}$.

Розрізняють три стадії (фази) функціонування мережі:

1. Ініціалізація. Встановлюються вагові коефіцієнти за формулою (11.2.1).

2. Подання і запам'ятовування вхідного образу X. Вимкнення вхідного образу X.

3. Обчислення станів нейронів шляхом циклічних перерахунків з визначенням елементів виходу Y до отримання їх сталого значення.

Розглянемо приклад розрахунків вагових коефіцієнтів мережі Хопфілда.

Припустимо, що ми маємо два еталони X^{E1} та X^{E2} , по яких виконується розпізнавання X:

$$X^{E1} = \begin{bmatrix} +1 \\ -1 \\ -1 \\ +1 \end{bmatrix}; \quad X^{E2} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}. \quad (11.2.2)$$

Тоді вагові коефіцієнти можуть бути уявленими у вигляді вагової квадратної матриці розміром $[n*n]=[4*4]$

$$\begin{aligned} W &= X_i^{E1} [4:1] \cdot (X_i^{E1})^T [1:4] + X_i^{E2} [4:1] \cdot (X_i^{E2})^T [1:4] \\ &= \begin{bmatrix} +1 \\ -1 \\ -1 \\ +1 \end{bmatrix} * [+1 \ -1 \ -1 \ +1] + \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix} * [-1 \ -1 \ +1 \ +1] = \\ &= \begin{bmatrix} 0 & -1 & -1 & +1 \\ -1 & 0 & +1 & -1 \\ -1 & +1 & 0 & -1 \\ +1 & -1 & -1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & +1 & -1 & -1 \\ +1 & 0 & -1 & -1 \\ -1 & -1 & 0 & +1 \\ -1 & -1 & +1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -2 \\ -2 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \end{bmatrix}. \quad (11.2.3) \end{aligned}$$

Тут згідно формули (11.2.1) діагональні елементи у квадратних матрицях дорівнюють нулю (для них $i=j$). Вважаємо, що номер рядка означає номер нейрону, а номер колонки вказує номер нейрону, від якого отриманий зв'язок. Значення елемента вагової матриці $W_{ij}=0$ означає відсутність зв'язку, і цей зв'язок у структурній схемі мережі Хопфілда **можна не показувати**.

В результаті за матрицею (11.2.3) отримуємо навчену мережу Хопфілда рис. 11.2.2, яка може безпосередньо використовуватись для асоціативного розпізнавання образів шляхом циклічних розрахунків.

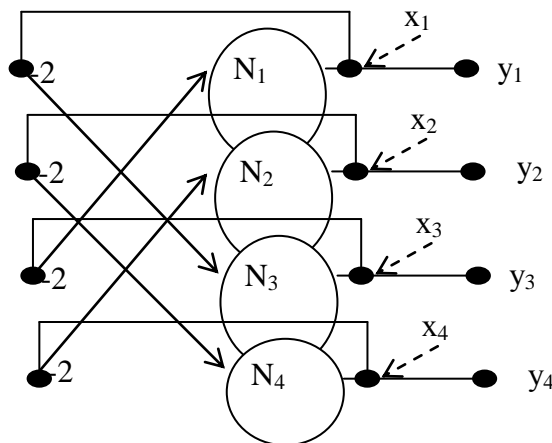


Рис. 11.2.2. Мережа Хопфілда

Припустимо, що на вхід навченої мережі Хопфілда ми подали на короткий час сигнал X_C від об'єкта, який потрібно віднести до найближчого еталона (X^{E1} або X^{E2}), де

$$X_C = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ -1 \end{bmatrix}.$$

Навчена мережа Хопфілда запам'ятовує цей введений сигнал X_C , вмикає входи і після кількаразової циклічної процедури перерахунків елементів $Y^T=(y_1, y_2, y_3, y_4)$, дає на виходах стабільні елементи $Y^T=(y_1, y_2, y_3, y_4)$ еталону, який є найближчим до введеного сигналу X_C .

Існує *асинхронний та синхронний алгоритм* роботи НМ Хопфілда:

1. *В синхронному режимі* всі нейрони одночасно змінюють свій стан момент часу «t+1» відносно стану на момент «t».

2. *В асинхронному режимі роботи:*

– в момент часу «t+1» змінює свій стан лише один нейрон відносно стану на момент «t»;

– потім в момент часу «t+2» змінює свій стан інший нейрон відносно стану мережі на момент «t+1» і т. д.

Досягнутий стаціонарний стан мережі є однаковим і не залежить від синхронності алгоритму роботи нейронів.

Елементи вихідного сигналу $Y^T=(y_1, y_2, y_3, y_4)$ циклічно замінюються на нові перераховані значення. Для цього розраховуються значення *NET* нейронів і у виходи $Y^T=(y_1, y_2, y_3, y_4)$ записуються нові значення з врахуванням дії нелінійнихактиваційних функцій нейронів.

Далі виконується перевірка, чи змінились (порівняно з попереднім станом) вихідні значення нейронів $Y=(y_1, y_2, y_3, y_4)$ за останню ітерацію. Якщо $Y^T=(y_1, y_2, y_3, y_4)$ змінились, то перерахунки повторюються, а в іншому випадку (якщо отримуємо стабільні виходи) – зупиняються. У цьому випадку вихідний вектор являє собою еталон, який найліпшим чином відповідає вхідним даним. Тобто ми повинні переконатися у тому, що мережа Хопфілда через кілька ітерацій не змінює отриманий еталон, бо це є ознакою завершення розпізнавання.

В результаті на 4-му кроці в асинхронному режимі був отриманий вектор виходу

$$Y_{K1} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix},$$

який відповідає еталону X^{E2} .

Наведений приклад є навчальним(в ньому запам'ятовувались $0,5 \cdot n$ еталонних векторів замість рекомендованої кількості $0,15 \cdot n$) і призначений лише для демонстрації порядку розрахунків.

Якщо мережа Хопфілда не розпізнає образ, то це може бути пов'язаним з такими причинами:

1. Кількість образів-еталонів m , що запам'ятовуються, не повинна перевищувати значення величини, приблизно рівної $0,15n$, де n – кількість нейронів N_1, N_2, \dots, N_n , яка дорівнює кількості ознак об'єктів.

2. Із-за великої схожості образів, мережа їх плутає.

У деяких публікаціях навчання мережі Хопфілда відносять до навчання без учителя [24].

Великою заслугою Хопфілда та Танка є використання **функції енергії** для налаштування ваг (навчання) НМ. Хопфілд та Танк сформулювали основні поняття використання функції енергії для розв'язання задач комбінаторної оптимізації. В першу чергу до таких задач відноситься *задача комівояжера, задача розподілу ресурсів, задача про призначення та інші*. Вирішення цих задач методами математичного програмування пов'язані з великими витратами часу, що зменшує ефективність їх застосування.

Хопфілд і Танк запропонували підхід до **наближеного** рішення комбінаторних задач на основі мереж Хопфілда [162; 198]. **Похибка у розв'язку комбінаторних задач досягає 12 %**, але при цьому значно зменшується час його отримання.

11.2.2. Нейронна мережа Андерсона

Одношарова асоціативна мережа Андерсона [130] наведена на рис. 11.2.3. Структура та принцип роботи НМ є аналогічними з НМ Хопфілда. Відмінність мережі Андерсона від НМ Хопфілда полягає у тому, що допускається наявність власних зворотних зв'язків нейронів(тобто не виконується обмеження $w_{ii} = 0$) та є можливість запам'ятовування в 8...10 разів більшої кількості еталонів (n еталонів замість $(0,1 \dots 0,12) n$ за НМ Хопфілда).

Для вектора ознак $X = (x_1, x_2, \dots, x_n)$ на k -му кроці (крок розглядається як момент часу) мережа забезпечує існування n стійких станів, таких, що

$$Y_k = W \cdot X_k,$$

де $k = 1, 2, \dots, n$ – порядковий номер еталону; Y_k – вихідний вектор; W – вагова матриця, яка визначається аналогічно матриці мережі Хопфілда (11.2.1); X_k – вхідний вектор.

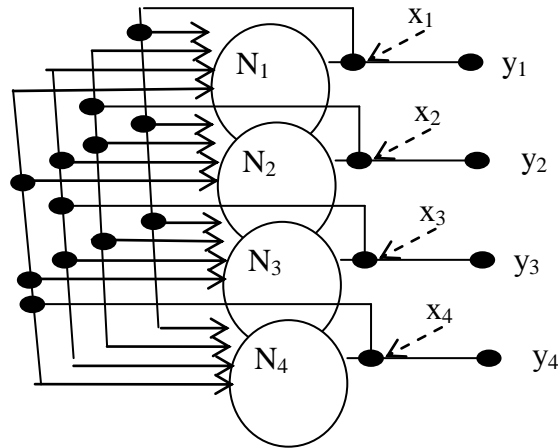


Рис. 11.2.3. Асоціативна НМ Андерсона

На $(k+1)$ -й момент часу вихідне значення j -го сигналу мережі визначаються за правилами:

$$y_j(k+1) = \begin{cases} +1, & \text{якщо } u_j(k) \geq +1; \\ y_j(k), & \text{якщо } |u_j(k)| \leq 1; \\ -1, & \text{якщо } u_j(k) \leq -1, \end{cases}$$

де $u_j(k) = y_j(k) + \beta \sum_{i=1}^n w_{ij} x_i(k)$ – сигнал NET_j нейрона у момент часу k ; β – мала додатна величина, яка зветься параметром зворотного зв'язку.

Введені додатні зворотні зв'язки переводять виходи всіх нейронів у бінарні форми режиму насичення ± 1 (це спостерігається також й для аналогових сигналів).

У початковому стані мережа перебуває всередині одиничного гіперкуба, обмеженого по осях біполярними вхідними сигналами ± 1 . По суті асоціативна НМ Андерсона розглядає вершини гіперкуба як центроїди кластерів для заданого масиву даних. НМ рекурсивно змінює свій стан у напрямку бінарних значень гіперкуба ± 1 , які відповідають збереженим образам. При навчанні використовують малі почіткові значення вагової матриці нейронів W та дельта – правило (або правило Хебба). Після подання вхідного вектора, розпізнавання відповідного еталону продовжується до отримання сталого стану НМ.

Якщо центроїди кластерів відомі заздалегідь, то їх координати можна використати як вершини гіперкуба, що полегшує отримання відповіді від НМ.

11.2.3. Асоціативна нейронна мережа з кількох вхідних зірок

Раніше ми виявили (розд. 10.3), що вхідна зірка Гросберга та Adaline мають на виході скалярний добуток двох векторів однакової розмірності

$w \cdot x = u = w_1 \cdot x_1 + \dots + w_n \cdot x_n = w \cdot x \cdot \cos \varphi \geq 0$: один з них – вхідний вектор $X = (x_1, \dots, x_j, \dots, x_n)$, а інший – вектор вагових коефіцієнтів $W = (w_1, \dots, w_j, \dots, w_n)$. Величина *скалярного добутку двох векторів при постійних модулях векторів* збільшується при співпадінні їх напрямків (якщо кут $\varphi \rightarrow 0$). Тому скалярний добуток векторів можна використати як *міру близькості вхідного сигналу X з еталоним X^{Ee}* , записаним в вагах вектора W : чим більший скалярний добуток $u = w_1 \cdot x_1 + \dots + w_n \cdot x_n = w \cdot x \cdot \cos \varphi \geq 0$, тим більш схожі між собою вхідний вектор X та еталон X^{Ee} .

Найпростіша асоціативна лінійна НМ наведена на рис. 11.2.4, а (перший шар – сенсорний з кількістю нейронів n , другий шар – асоціативний з кількістю нейронів m). Кожний нейрон другого шару має вигляд зірки Гросберга або вигляд нейрону Adaline (рис. 11.2.4, б).

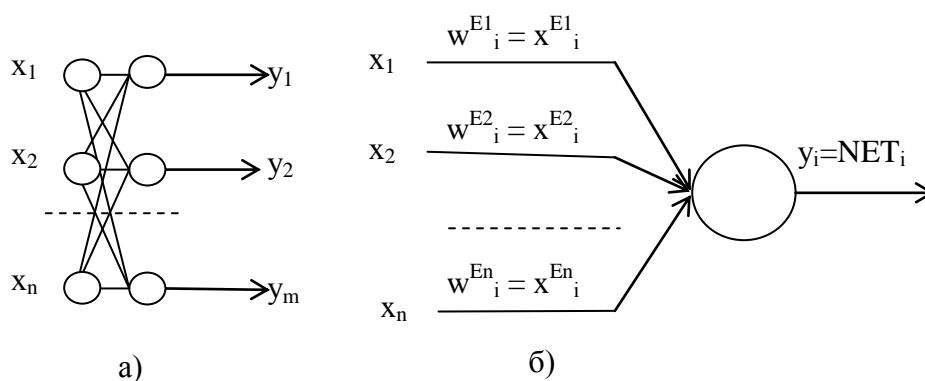


Рис. 11.2.4. НМ для визначення скалярних добутків вхідного та ряду еталонних векторів

Мережа Гросберга [103, с. 75 Руденко О. Г., Бодянский С. В. Штучні нейронні мережі. – Харків : ТОВ «Компанія СМІТ», 2006. – 404 с.] отримує скалярний добуток двох векторів у вигляді

$$u_1 = \sum_{j=1}^n w_j x_j = NET_1$$

А її активаційна функція класифікує вхідний вектор

$$f_1(u_1) = \begin{cases} 1, & \text{якщо } u_1 \geq A; \\ 0 & \text{в іншому випадку,} \end{cases}$$

де w_j – ваговий коефіцієнт нейрону зірки Гросберга; x_j – відповідний елемент вхідного вектора.

Недоліком зірки Гросберга є те, що нейрон активізується у залежності від трьох параметрів: w_j , x_j та їх знаків. Це може вести до помилок: якщо вагові коефіцієнти w_j отримані від надзвичайно великого вектора, а x_j – від маленького, то їх скалярний добуток не буде характеризувати різницю між векторами, що призведе до помилки у розпізнаванні образу. Наприклад, при $w_j=100$ та $x_j=1$ добуток $w_j x_j = 100 \cdot 1 = 100$ (розглядається лише один елемент вектора). І таке ж значення отримується для векторів $w_j^*=10$ та $x_j^*=10$: $w_j^* x_j^* = 10 \cdot 10 = 100$.

Тому замість зірки Гросберга для цієї мети ліпше використовувати нейрон, який по мінімальній евклідовій відстані

$$u_i = \sum_{j=1}^n (w_j - x_j)^2 = NET_i$$

або по відстані по Манхеттену, що дорівнює сумі абсолютних значень різниць числових координат

$$u_i = \sum_{j=1}^n |w_j - x_j| = NET_i$$

Застосовує активаційну функцію

$$f_i(u_i) = \begin{cases} 1, & \text{якщо } u_i \leq A_i; \\ 0 & \text{в іншому випадку,} \end{cases}$$

де A_i – задана експертом величина, яка обмежує класифікацію вхідних векторів X простором у вигляді деякого шару навколо кінця вектора W (для евклідової відстані), або діапазонами значень по окремих осях навколо кінця вектора W (для відстані по Манхеттену); $i = 2$ для евклідової відстані; $i = 3$ для відстані по Манхеттену, і виділяє вхідні вектори X , які є найближчими до вектора W , записаного у вагах нейрона.

Чим більше значення A_i , тим більше можливих образів може активізувати нейрон, що стає при цьому все менш «розбірливим». Значення u_i описують навколо кінця вектора W деякий гіперпростір, і вхідна зірка виділяє сигналом 1 лише ті вхідні вектори X , кінцівки яких попадають в цей гіперпростір.

Крім того, якщо у вхідній зірці не використовувати активаційну функцію, то величини u_1 – u_3 кількох вхідних зірок можна застосувати для визначення належності вхідного вектора X до вектора W , записаного у вагах відповідної зірки.

Завдання

Навести для вхідної зірки вхідний сигнал X_1 , який має 6 елементів, що складаються з «+1» та «-1». Якщо вважати «-1» за «0», то елементи «+1» повинні створювати порядковий номер студента у групі у двоїстому численні. Для вектора X_1 навести довільні еталонні вектори X^{E1} , X^{E2} , X^{E3} , які відрізняються від вектора X_1 та між собою. На основі наведеного нижче прикладу визначити найближчий еталон до вектора X_1 .

Припустимо, що ми маємо вхідний сигнал X_1 , який потрібно порівняти з трьома еталонами X^{E1} , X^{E2} , X^{E3} :

$$X_1 = \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix}; \quad X^{E1} = \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}; \quad X^{E2} = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}; \quad X^{E3} = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix}.$$

Очевидно, що сигнал X_1 є найближчим до навчального еталону X^{E1} , бо відрізняється від нього в одному пункті, в той час, як від інших еталонів X_1

відрізняється в двох пунктах. Але цей висновок підтверджується розрахунком скалярних добутків $X_1 X^{E1} = 2$; $X_1 X^{E2} = 0$; $X_1 X^{E3} = 0$. Отримані скалярні добутки показують, що найближчим до вектора X є еталонний вектор X^{E1} . Отримати відстані евклідові та по Манхеттену між векторами.

11.2.4. Двоспрямована асоціативна пам'ять

Запропонована Б. Коско [172] двоспрямована асоціативна пам'ять (ДАП, рис. 11.2.5), розширює модель Хопфілда. В ній подача зіпсованого вектора $X=(x_1, \dots, x_n)$ на вхід НМ викликає відповідний еталонний вектор $Y^E=(y_1^E, \dots, y_m^E)$ на виході. І навпаки: подача зіпсованого вектора $Y=(y_1, \dots, y_m)$ на виході НМ викликає відповідний еталонний вектор на вході $X^E=(x_1^E, \dots, x_n^E)$.

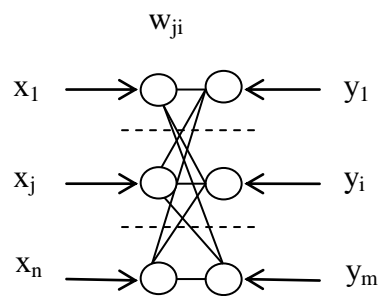


Рис. 11.2.5. Двоспрямована асоціативна пам'ять

Зв'язки між двома шарами є двоспрямованими (сигнали можуть проходити в двох напрямках): звідси й назва мережі. Кількість еталонних векторів, яку запам'ятовує ДАП (її «ємність») *не може перевищувати найменшого з двох чисел (n та m), а у дійсності може бути ще меншою, бо визначається кількістю незалежних рядків та колонок вагової матриці W* . Якщо n – кількість нейронів у вхідному шарі, то число еталонних векторів, які запам'ятовуються в мережі не перевищує $L=n/2\log_2 n$. Наприклад, якщо $n=1024$, то мережа здатна запам'ятати не більше за 25 образів (і то якщо матриця W не має залежні рядки чи колонки).

На відміну від мережі Хопфілда, двоскерована асоціативна пам'ять дає можливість визначати асоціації *між векторами $X=(x_1, \dots, x_n)$ та $Y=(y_1, \dots, y_m)$ різних розмірностей*, що збільшує масштаб їх застосування.

Елемент вагової матриці w_{ji} ($j = 1, 2, \dots, n$; $i = 1, 2, \dots, m$), що зв'язує i -й нейрон образу виходу y_i з j -м нейроном образу входу x_j , використовується під час аналізів сигналів, переданих в обох напрямках. Вагова матриця обчислюється як сума добутків всіх навчальних векторних пар [103, с. 126–130]. Процедура визначення ваг така сама, як в мережі Хопфілда, але у даному випадку [35]:

- матриця не обов'язково є квадратною;
- діагональні елементи не обнуляються.

Сигнали можуть бути дискретними (для них бажано обирати релейні активаційні функції) та безперервними (для них бажано обирати релейні сигмоїдальні активаційні функції).

Приклад. Розглянемо ДСП рис. 11.2.6, по якій розпізнаються пара векторів X_C, Y_C , що спотворена відносно еталонних векторів (X^{E1}, Y^{E1}) .

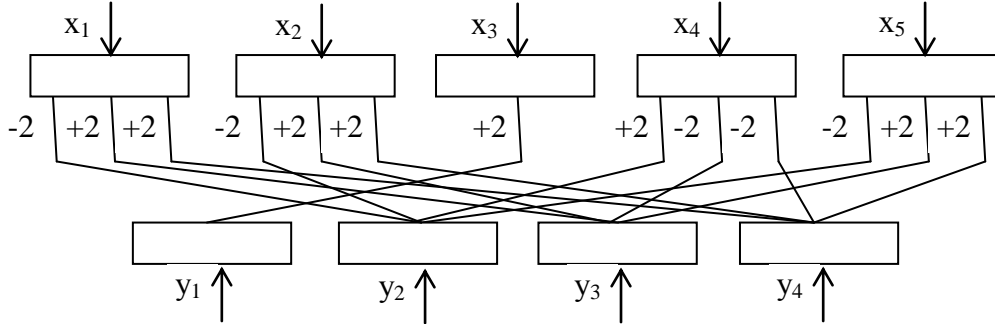


Рис. 11.2.6. Розрахована двоспрямована асоціативна пам'ять

Припустимо, що ДСП запам'ятала дві пари еталонних векторів (X^{E1}, Y^{E1}) та (X^{E2}, Y^{E2}) :

$$X^{E1} = \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \\ +1 \end{bmatrix}; \quad Y^{E1} = \begin{bmatrix} +1 \\ -1 \\ +1 \\ +1 \end{bmatrix}; \quad X^{E2} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ +1 \\ -1 \end{bmatrix}; \quad Y^{E2} = \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \end{bmatrix}; \quad X_C = \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \\ -1 \end{bmatrix}; \quad Y_C = \begin{bmatrix} +1 \\ +1 \\ +1 \\ +1 \end{bmatrix}. \quad (11.2.4)$$

Згідно двох пар еталонних векторів прямокутна матриця вагових коефіцієнтів $W[i:j]=W[4:5]$ може бути отримана у вигляді

$$W[4:5] = Y^{E1}[4:1] \cdot (X^{E1})^T [1:5] + Y^{E2}[4:1] \cdot (X^{E2})^T [1:5] = \begin{bmatrix} +1 \\ -1 \\ +1 \\ +1 \end{bmatrix} \begin{bmatrix} +1 & +1 & -1 & -1 & +1 \end{bmatrix} + \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 & +1 & -1 \end{bmatrix} = \begin{bmatrix} +1 & +1 & -1 & -1 & +1 \\ -1 & -1 & +1 & +1 & -1 \\ +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & -1 & -1 & +1 \end{bmatrix} + \begin{bmatrix} -1 & -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 & -1 \\ +1 & +1 & +1 & -1 & +1 \\ +1 & +1 & +1 & -1 & +1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -2 & 0 & 0 \\ -2 & -2 & 0 & +2 & -2 \\ +2 & +2 & 0 & -2 & +2 \\ +2 & +2 & 0 & -2 & +2 \end{bmatrix}. \quad (11.2.5)$$

Вважаємо, що у матриці $W[i:j]$ номер колонки j вказує на номер нейрону для елемента вектора X ; номер рядка i означає номер нейрону Y . Значення елемента вагової матриці $W_{ij}=0$ означає відсутність зв'язку, і цей зв'язок **можна не показувати**. В результаті за матрицею (11.2.5) отримуємо навчену двоспрямовану НМ рис. 11.2.6, яка може безпосередньо використовуватись для асоціативного розпізнавання образів.

Перевіряємо двічі отриману матрицю W по зіпсованій парі векторів X_C та Y_C :

– по вектору X_C : спочатку перевіряємо вихід еталону Y^{E1} , як результат входу еталону X^{E1} , а потім – вихід еталону Y^{E1} , як результат входу зіпсованого вектора X_C ;

– по вектору Y_C : спочатку перевіряємо вихід еталону X^{E1} , як результат входу еталону Y^{E1} , а потім – вихід еталону X^{E1} , як результат входу зіпсованого вектора Y_C .

$$W[4:5] \cdot X^{E1}[5:1] = \begin{bmatrix} 0 & 0 & -2 & 0 & 0 \\ -2 & -2 & 0 & +2 & +2 \\ +2 & +2 & 0 & -2 & +2 \\ +2 & +2 & 0 & -2 & +2 \end{bmatrix} \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \\ +1 \end{bmatrix} = \begin{bmatrix} +2 \\ -4 \\ +8 \\ +8 \end{bmatrix} \Rightarrow \begin{bmatrix} +1 \\ -1 \\ +1 \\ +1 \end{bmatrix} = Y^{E1}[4:1];$$

$$W[4:5] \cdot X_C[5:1] = \begin{bmatrix} 0 & 0 & -2 & 0 & 0 \\ -2 & -2 & 0 & +2 & +2 \\ +2 & +2 & 0 & -2 & +2 \\ +2 & +2 & 0 & -2 & +2 \end{bmatrix} \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} +2 \\ -8 \\ +4 \\ +4 \end{bmatrix} \Rightarrow \begin{bmatrix} +1 \\ -1 \\ +1 \\ +1 \end{bmatrix} = Y^{E1}[4:1];$$

$$W[5:4]^T \cdot Y^{E1}[4:1] = \begin{bmatrix} 0 & -2 & +2 & +2 \\ 0 & -2 & +2 & +2 \\ -2 & 0 & 0 & 0 \\ 0 & +2 & -2 & -2 \\ 0 & +2 & +2 & +2 \end{bmatrix} \begin{bmatrix} +1 \\ +1 \\ -1 \\ +1 \end{bmatrix} = \begin{bmatrix} +6 \\ +6 \\ -2 \\ -6 \\ +2 \end{bmatrix} \Rightarrow \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \\ +1 \end{bmatrix} = X^{E1};$$

$$W[5:4]^T \cdot Y_C[4:1] = \begin{bmatrix} 0 & -2 & +2 & +2 \\ 0 & -2 & +2 & +2 \\ -2 & 0 & 0 & 0 \\ 0 & +2 & -2 & -2 \\ 0 & +2 & +2 & +2 \end{bmatrix} \begin{bmatrix} +1 \\ +1 \\ +1 \\ +1 \end{bmatrix} = \begin{bmatrix} +2 \\ +2 \\ -2 \\ -2 \\ +2 \end{bmatrix} \Rightarrow \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \\ +1 \end{bmatrix} = X^{E1}.$$

11.2.5. Нейронна мережа Хеммінга

Схема нейронної мережі Хеммінга запропонована Ліппманом [175]. Назву мережі Ліппман надав на честь Хеммінга. Нейронна мережа Хеммінга може розглядатись як послідовне з'єднання двох мереж: НМ з кількох вхідних зірок Гроссберга плюс НМ «MAXNET».

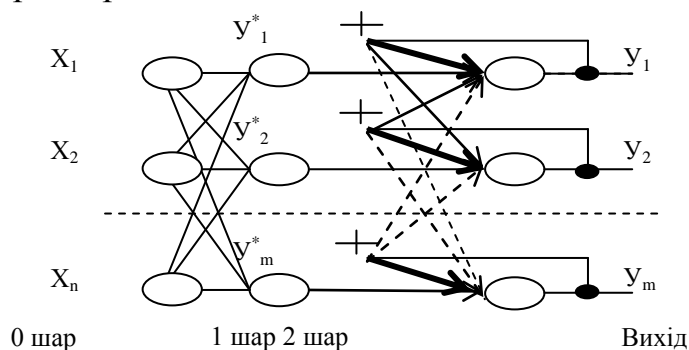


Рис. 11.2.7. Мережа Хеммінга (запропонована Ліппманом)

Якщо мережа Хопфілда на виході дає вектор найближчого еталону, якому відповідає вхідний вектор, то мережа Хеммінга активізує нейрон, у вагах якого записаний відповідний найближчий еталон і вказує порядковий номер найближчого еталону.

Переваги мережі Хеммінга перед мережею Хопфільда:

- Виведення більшої кількості класів.
- На порядок менша кількість зв'язків між нейронами.
- Сталість до випадкових перешкод і помилок у вхідній інформації (перекручення можуть досягати до 43 % бінарних елементів вхідного вектора).
- Простота метода навчання: вхідні ваги у ній задаються один раз.
- Висока швидкодія і стабільність досягнення стану стабілізації.

Схема нейронної мережі складається з наступних шарів (рис. 11.2.7).

1. **Вхідний 0 шар** з n нейронами (*по кількості ознак об'єкта*), який розмножує нормалізований невідомий вхідний вектор ознак X_C .

2. **Проміжний асоціативний 1 шар** з m нейронами (*по кількості класів*). Ваги входів окремого нейрону $U^*_1 \dots U^*_m$ асоціативного шару 1 мають значення елементів відповідного еталонного вектора X^{Ei} , $i = 1, 2, \dots, m$: кожний один нейрон $U^*_1 \dots U^*_m$ призначений для отримання на виході скалярного добутку вектора сигналу X_C та вектора еталону X^{Ei} . Найбільший скалярний добуток векторів $(X_C X^{Ei})$ на виході нейронів $U^*_1 \dots U^*_m$ вказує на нейрона-переможця, бо саме записаний в його вагах еталон є найбільш схожим з вхідним вектором X_C .

Нейрони асоціативного шару 1 не мають активаційних функцій і використовуються лише один раз – для отримання їх виходів у вигляді відповідних скалярних добутків $(X_C X^{Ei})$ на входах шару «MAXNET». Далі асоціативний шар 1 вимикається і вступає в роботу лише шар «MAXNET», який виконує циклічні розрахунки з метою виділення найбільшого скалярного добутку векторів $(X_C X^{Ei})$. Шар «MAXNET» зупиняє циклічні розрахунки, якщо перевірка показує, що вихід не змінюється.

Припустимо, що ми маємо вхідний сигнал сенсорного шару

$$X_C = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ -1 \end{bmatrix}$$

та чотири еталони $X^{E1} - X^{E4}$, для яких виконується розпізнавання:

$$X^{E1} = \begin{bmatrix} +1 \\ -1 \\ -1 \\ +1 \end{bmatrix}; \quad X^{E2} = \begin{bmatrix} -1 \\ +1 \\ +1 \\ +1 \end{bmatrix}; \quad X^{E3} = \begin{bmatrix} +1 \\ -1 \\ +1 \\ +1 \end{bmatrix}; \quad X^{E4} = \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix}. \quad (11.2.6)$$

Виходи нейронів «1-го шару», як відповідні скалярні добутки «сигналу X_C » і ваг входів нейронів (як відповідних еталонів) дорівнюють наступним скалярним добуткам:

$$X_C (X^{E1})^T = (-1)(+1) + (-1)(-1) + (+1)(-1) + (-1)(+1) = -1 + 1 - 1 - 1 = -2;$$

$$X_C (X^{E2})^T = +1 - 1 + 1 - 1 = 0;$$

$$X_C (X^{E3})^T = -1+1+1-1=0;$$

$$X_C (X^{E4})^T = -1+1+1+1=+2.$$

Найбільший скалярний добуток $X_C (X^{E4})^T = +2$ вказує на те, що *вхідний вектор* X_C має найбільшу *схожість* з еталонним вектором X^{E4} . Чим більший скалярний добуток, тим більша *схожість* спостерігається між векторами.

Тобто у мережі Хеммінга, яку запропонував Ліппман, використовується *не міра близькості по Хеммінгу, а міра близькості по скалярному добутку «+2»*.

Міра близькості по Хеммінгу між векторами X_C та X^{E4} дорівнює «3», тобто є завжди невід'ємним числом, бо дорівнює кількості позицій, у яких біти входу співпадають з бітами класу-еталону. Перехід від скалярного добутку «+2» до схожості векторів по Хеммінгу «3» *можливий і використовується, але є зайвою математичною операцією*, яка лише ускладнює розрахунки без впливу на їх кінцевий результат.

3. Вихідний 2-й шар є стандартним шаром MAXNET, який призначений для виділення шляхом повторних розрахунків нейрона – переможця 1-го шару з максимальним значенням скалярного добутку, тобто з *найбільшою близькістю до вхідного вектора* X_C . MAXNET подавляє усі сигнали, крім найбільш потужного в 1-му шарі. В результаті буде активований тільки один вихід мережі, який відповідає цьому еталону X^{E4} .

Робота мережі Хеммінга складається з таких кроків:

1. На входи мережі подається невідомий вектор $\mathbf{X} = \{x_j; j=1\dots n\}$. На базі цього вектора розраховуються виходи нейронів 1-го шару у вигляді міри близькості, яка дорівнює скалярному добутку векторів входу та записаного у вагах нейрону еталону.

2. Отриманими величинами ініціалізуються значення U_i^* виходів 2-го шару (MAXNET). Величини U_i^* запам'ятовуються мережею MAXNET, і вхідний вектор X сумісно з відповідними значеннями U_i^* шарів 0 та 1 вилучаються з розгляду. Далі використовується *лише мережа MAXNET*, яка виділяє найбільший елемент скалярного добутку двох векторів U_i^* за правилом «Переможець забирає все».

Довідка. Як міра близькості в мережі Хеммінга використовується скалярний добуток двох векторів, а не відстань Хеммінга.

Блиькість по Хеммінгу дорівнює числу позицій з однаковими елементами векторів у вигляді цифр, символів, слів, речень. Наприклад, близькість між двома векторами за Хеммінгом дорівнює невід'ємному цілому числу: $d(11011; 11101) = 3$; $d(3,7,1,4; 5,7,1,4) = 3$; $d(\text{нейрон}; \text{таксон}) = 2$.

Скалярний добуток $S = V_1 \cdot V_2^T$ двох векторів V_1 та V_2 однакового n -го порядку з елементами « ± 1 » може змінюватись від « $S = -n$ » (повне неспівпадіння) до « $S = +n$ » (повне співпадіння). Скалярний добуток теж є мірою близькості між векторами, але він може мати від'ємне значення на відміну від близькості за Хеммінгом.

Скалярний добуток можна визначити через відстань Хеммінга $R(X^C, X^{Ee})$ за формулою

$$D = (X^C)^T \cdot X^{Ee} = n - 2R(X^C, X^{Ee}), \quad (11.2.7)$$

де n – порядок векторів X^C та X^{Ee} ;

$R(X^C, X^{Ee})$ – відстань Хеммінга між векторами X^C та X^{Ee} ;

$e = 1, 2, \dots, t$ – порядковий номер еталону.

З формули (11.1.7) отримуємо значення відстані Хеммінга для одного e -го навчального еталону, яка дорівнює

$$NET_{1e} = R(X^C, X^{Ee}) = 0,5n - 0,5((X^C)^T \cdot X^{Ee}) = (+1) \cdot [0,5n] + (X^C)^T [-0,5 \cdot X^{Ee}], \quad (11.2.8)$$

де показані в квадратних дужках величини дорівнюють вагам вхідних зв'язків нейрону. Таким чином, нейрон повинен мати постійний вхідний сигнал $\{x_0=+1, w_{10}=0,5n\}$.

11.2.6. Нейронні мережі «Миколаїв»

Нижче наведені розроблені в Україні НМ «Миколаїв», НМ «Миколаїв 2», НМ «Миколаїв 3» (рис. 11.2.8), які замінюють мережі Хопфілда, Андерсена та Хеммінга. Вони відрізняються наступними особливостями:

1. Вилучені циклічні перерахунки в даних вхідного вектора. Ці перерахунки замінені на простішу операцію: виділення за допомогою активаційної функції максимуму скалярного добутку вхідного та еталонних векторів. В мережі Хеммінга вилучається шар MAXNET з його циклічними перерахунками та налаштуванням.

2. Вилучені операції вимкнення вхідного сигналу та перевірки завершення операцій циклічних перерахунків.

3. Загальна кількість еталонів може не обмежуватись і може дорівнювати максимальній можливій величині 2^n , n – загальна кількість ознак вхідного вектора X .

НМ «Миколаїв», по роботі перших двох шарів нічим не відрізняються від НМ Хеммінга: на виходах нейронів 2-го шару отримуються скалярні добутки двох векторів (вхідного вектора X та записаних в вагах кожного нейрону 2-го шару еталонних векторів) у вигляді вектора $Y=(y_1, y_2, \dots, y_i, \dots, y_m)$.

Їх відмінністю є заміна вихідної мережі MAXNET мережі Хеммінга з m нейронами на активаційну функцію, яка виконує логічні операції. Ці логічні операції теж виділяють в 3-му вихідному шарі з вектора скалярних добутків $Y=(y_1, y_2, \dots, y_i, \dots, y_m)$ максимальний скалярний добуток нейрона-переможця 2-го шару $y_i=ZM$, інформація про якого передається на вихід. Але ця передана інформація має три варіанти:

1. **НМ «Миколаїв» має такий же вихід, як і НМ Хеммінга (рис. 11.2.8, а) [76]:**

for $i:=1$ to m do $z_i:=0$;

$ZM:= y_1$; for $i:=2$ to m do if $y_i > ZM$ then $ZM:= y_i$;

for $i:=1$ to m do if $y_i = ZM$ then $z_i:= 1$;

В результаті на виходах нейрону Z реагуючого шару утворюється вихідний вектор $Z = (z_1, z_2, \dots, z_i, \dots, z_m)$, у якого всі елементи дорівнюють нулю, за вилученням найближчого до еталону скалярного добутку z_i .

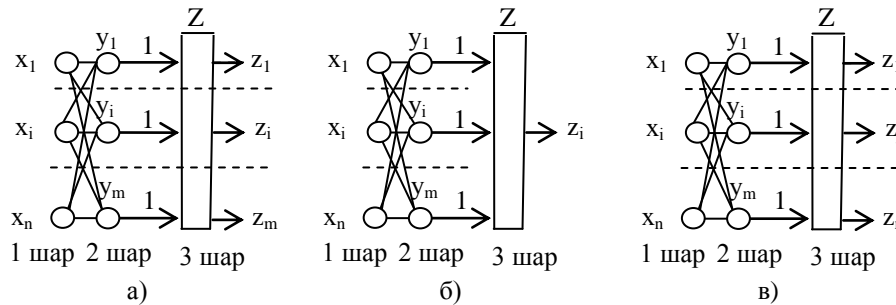


Рис. 11.2.8. Нейронні мережі:

а) «Миколаїв»; б) «Миколаїв 2»; в) «Миколаїв 3»

2. *НМ «Миколаїв 2» відрізняється від НМ Хеммінга тим, що має один вихід (рис. 11.2.8, а), на якому отримує порядковий номер максимального скалярного добутку за рахунок виконання логічних операцій:*

```
z:=1; y1:=0; Forj:=1 tondoy1:= y1 + (x1j·xj);
For e:=2 to E do Begin ye:=0;For j:=1 to n do ye:= ye + (xej·xj);
If ye>y1then Begin z:= e; y1:= ye; end; end;
```

де $z = e$ – вихід нейрона реагуючого шару та нейронної мережі у вигляді порядкового номера e еталонного вектора X_e , який має найбільший скалярний добуток з вхідним вектором X ;

x_{ej}, x_j – елементи з j -м порядковим номером векторів X_e та X ;

y_e – скалярний добуток векторів X_e та X ;

$e = 1, 2, \dots, E$ – порядковий номер еталонного вектора X_e ;

$j = 1, 2, \dots, n$ – порядковий номер елементів x_j та x_{ej} векторів X та X_e ;

Інша активаційна функція виводить мінімальне значення скалярного добутку. Різновидом НМ «Миколаїв 2» є видача кодованого повідомлення $z=g_e$, яке відповідає порядковому номеру e еталонного вектора X_e з найменшим скалярним добутком з вхідним вектором X згідно активаційної функції. В активаційній функції може використовуватись близькість по Хеммінгу між векторами.

Асоціативний шар має E нейронів, де $e = 1, 2, \dots, E$ – порядковий номер нейрона асоціативного шару та відповідного йому еталонного вектора $X_e = (x_{e,1}, x_{e,2}, \dots, x_{e,j}, \dots, x_{e,n})$. Кожний нейрон асоціативного шару має n входів з ваговими коефіцієнтами входів, рівними значенням елементів відповідного еталонного вектора $X_e = (x_{e,1}, x_{e,2}, \dots, x_{e,j}, \dots, x_{e,n})$, та має один вихід, на якому видається скалярний добуток двох векторів у вигляді значення

$$y_e = \sum_{j=1}^n x_{e,j} x_j$$

Сигнали асоціативного шару поступають на реагуючий шар з одним нейроном, який має E входів з ваговим коефіцієнтом 1, до яких увімкнені

відповідні виходи y_e нейронів асоціативного шару. Нейрон реагуючого шару має один вихід, призначений для виведення порядкового номеру $z = e$ еталонного вектора X_e , який має найбільший скалярний добуток з вхідним вектором X .

3. **НМ «Миколаїв 3»** відрізняється від НМ Хеммінга тим (рис. 11.2.8, в), що має n виходів, на яких отримує, як і мережа Хопфілда, елементи найближчого еталонного вектора $X_e = (x_{e,1}, x_{e,2}, \dots, x_{e,j}, \dots, x_{e,n})$, $e = 1, 2, \dots, E$, згідно активаційної функції

```

 $\alpha := 1; y_1 := 0; \text{For } j := 1 \text{ tondo } y_1 := y_1 + (x_{1,j} \cdot x_j);$ 
 $\text{For } e := 2 \text{ to } E \text{ do } \text{Begin } y_e := 0; \text{For } j := 1 \text{ tondo } y_e := y_e + (x_{e,j} \cdot x_j);$ 
 $\text{If } y_e > y_1 \text{ then } \text{Begin } \alpha := e; y_1 := y_e; \text{end; end;}$ 
 $\text{For } j := 1 \text{ tondo } z_j := x_{\alpha,j};$ 

```

де $\alpha = e$ – порядковий номер e еталонного вектора X_e , який має найбільший скалярний добуток з вхідним вектором X ;

$x_{e,j}, x_j$ – елементи з j -м порядковим номером векторів X_e та X ;

$x_{\alpha,j}$ – елементи з j -м порядковим номером векторів X_e при $e = \alpha$;

z_j – елементи з j -м порядковим номером вихідного вектора $Z = (z_1, z_2, \dots, z_j, \dots, z_n)$;

y_e – скалярний добуток векторів X_e та X ;

$e = 1, 2, \dots, E$ – порядковий номер еталонного вектора X_e , скалярного добутку y_e та входу нейрону реагуючого шару;

$j = 1, 2, \dots, n$ – порядковий номер елементів $(x_j, x_{e,j}, z_j)$ векторів (X, X_e, Z) .

Сингеричні НМ. Засновником сингеретики є Х.Хакен та В. Банзхатт [133], а основним принципом сингеретики є аналіз стаціонарних станів динамічної системи, які залежать від початкових умов і ділянок «притягування» стаціонарних точок системи [103, с. 149].

Сингерична НМ відноситься до асоціаторів і має однакову мету з НМ Хопфілда: на її вхід подається спотворений n -вимірний вектор $X_1[n:1]$, а на виході отримується відповідний вихідний n -вимірний еталонний вектор $X_2[n:1]$. У порівнянні з НМ Хопфілда сингерична НМ має **наступні переваги**: сингерична НМ не має нестійких станів; кількість еталонів, яка зберігається, дорівнює n (порядку вхідного вектора).

В сингеричній НМ вхідний вектор $X_1[n:1]$ за допомогою вагової матриці $W_1[n:m]$ перетворюється у проміжний вектор $Y^*_2[1:m] = X_1^T[1:n] \cdot W_1[n:m]$. Вектор $Y^*_1[m:1]$ подається на нелінійний динамічний шар, який описується системою диференційних рівнянь. За кілька ітерацій системи диференційних рівнянь на виході нелінійного динамічного шару замість вектора Y^*_2 отримується вектор $Y_2[m:1]$ з новими елементами. Вектор $Y_2[m:1]$ подається на наступний шар з матрицею вагових коефіцієнтів $W_2[m:n]$, в результаті чого на виході сингеричної НМ отримується найближчий збережений еталон $X_2[1:n] = Y_2^T[1:m] \cdot W_2[m:n]$.

НМ «Миколаїв» може замінити сингеричний трьохшаровий комп'ютер без використання вказаних складних обчислень.

11.3. Багат шарові нейронні мережі з одним шаром навчання

11.3.1. Загальні особливості багат шарових нейронних мереж з одним шаром навчання

Нечіткі НМ з одним шаром навчання розроблені в Україні на базі «теорії однозначних нечітких систем» і призначені для розв'язку чітких та однозначно нечітких задач [62; 115]. Тут слово «однозначність» означає однозначно (сінонімальний) перехід всередині нечіткої множини від числа до лінгвістичного твердження з мірою впевненості і навпаки – зворотний однозначно перехід від лінгвістичного твердження до числа. Внаслідок лінійності характеристик NET, НМ з одним шаром навчання у принципі *не може попасти у локальний мінімум* і тому завжди отримує оптимальний розв'язок.

У порівнянні з НМ, побудованими на класичній теорії Л. Заде [33], НМ по теорії М. В. Турти [115] мають більш простий і більш швидкодіючий алгоритм з наскрізними числовими розрахунками і з використанням лінгвістичних визначень чисел лише для пояснень і у випадку потреби, з відмовою від використання великої кількості методів дефазифікації (переведення лінгвістичної форми інформації в числову форму).

Згідно теорії однозначних нечітких систем фазифікація та дефазифікація виконується [115]:

а) для входу – дефазифікація лінгвістичних даних експерта (переведення лінгвістичного твердження експерта в число) для подальших числових розрахунків в ЕОМ;

б) для числового виходу може бути застосована фазифікація лише у випадку необхідності переведення вихідної числової інформації у зручну для експерта лінгвістичну форму.

За теорією М. В. Турти [115] використання чисел на вході, всередині та на виході НМ не заважає їх однозначному переведенню у лінгвістичну форму з мірою впевненості згідно введеному Л. Заде поняттю лінгвістичної змінної з деякою мірою впевненості.

НМ з одним шаром навчання отримані за теорією М. В. Турти [115] і мають наступні переваги:

– спрощуються розрахунки, пов'язані з класифікацією об'єктів із-за відсутності алгоритму зворотного розповсюдження похибок (back propagation);

– збільшується швидкодія за рахунок того, що при навчанні вагові коефіцієнти визначаються або за дельта – правилом або шляхом розв'язку матричного рівняння скороченого порядку за методом найменших квадратів (МНК). Наприклад, за МНК розв'язуються лише 4 рівняння з бази даних на 9 правил, які задаються експертом; один раз отримане рішення у вигляді розв'язку за МНК у подальшому безпосередньо використовується

для розрахунку числового виходу на введені числові ознаки без потреби у фазифікації та дефазифікації);

– в нечітких НМ зміна одного чи кількох з правил бази даних не викликає необхідності перегляду всієї бази даних та її функцій належності. Одночасно значно *спрощується процедура дефазифікації* нейронної мережі, бо у відповідь на *вхідні цифрові дані* $X=(x_1, x_2, \dots, x_j, \dots, x_n)$ отримується *вихідне цифрове рішення* $Y(X)$, яке лише при необхідності переводиться у лінгвістичний еквівалент з функцією належності.

11.3.2. Однозначні нейро-нечіткі системи з нейронами Троянда

Однозначні нейро-нечіткі системи з нейронами Троянда розроблені М. В. Турти в Україні (2007) і представлені в роботі [115].

НМ з дзвоноподібними активаційними функціями наведена на рис. 11.3.1.

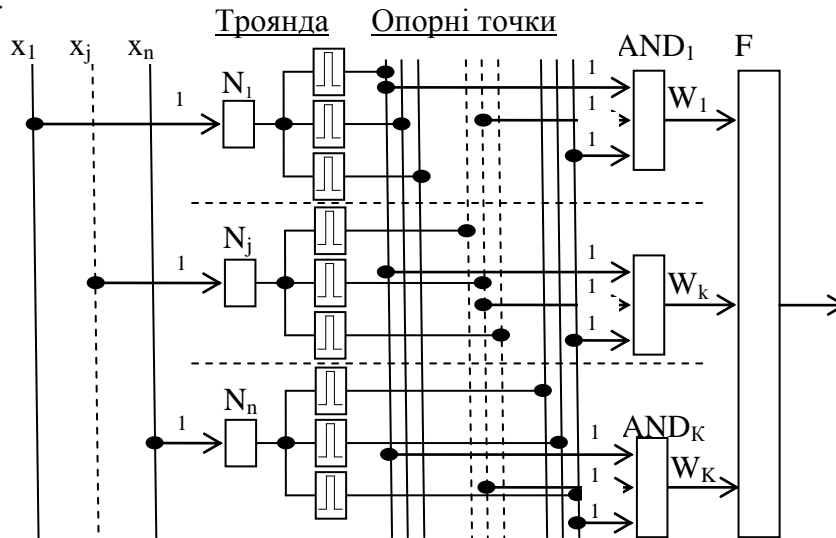


Рис. 11.3.1. Однозначна нечітка НМ з нейронами Троянда, які мають дзвоноподібні активаційні функції

Вона видає на виході осереднені вихідні дані по інформаційним модулям для деякої функції $F(X)$, де $X=(x_1, x_2, \dots, x_j, \dots, x_n)$ – вектор вхідних змінних. НМ не потребує навчання, бо всі її вагові коефіцієнти входів або дорівнюють 1, або призначаються. Кожна вхідна змінна увімкнена до входу з ваговим коефіцієнтом 1 власного нейрону Троянда, які позначені N_1, N_2, \dots, N_n . Кожний нейрон Троянда N_1, N_2, \dots, N_n має задану кількість виходів, кожний з яких охоплює власний діапазон значень відповідної вхідної змінної x_j (у даному випадку застосовано по три виходи; тому універсум кожної змінної поділений на три піддіапазони) [115].

Сигнал входу x_j через нелінійні жорсткі дзвоноподібні активаційні функції надходять у вигляді або 1 або 0 на шини опорних точок. У сукупності три виходи одного нейрону Троянда охоплюють весь універсум від-

повідної змінної x_j : кожний вихід нейрону Троянда дорівнює 1 лише на відповідній власній множині типу «Велике значення», «Середнє значення» та «Мале значення» даної змінної x_j , а в усіх інших точках змінної x_j дорівнює 0.

Нейрони логічних блоків $AND_1, \dots, AND_k, \dots, AND_K$ видають на виході сигнал 1, якщо всі вхідні сигнали блоку одночасно дорівнюють 1. Тобто конкретний логічний блок AND_k сигналізує по власному інформаційному модулю про те, що вхідна точка з координатами $X=(x_1, x_2, \dots, x_j, \dots, x_n)$ знаходиться (вихідний сигнал 1), або не знаходиться (вихідний сигнал 0) в даному інформаційному модулі. В кожний даний момент з усіх логічних блоків лише один логічний блок $AND_1, \dots, AND_k, \dots, AND_K$ має на виході сигнал 1, а всі інші логічні блоки мають на виходах сигнал 0.

Вагові коефіцієнти $w_1, \dots, w_k, \dots, w_K$ є середніми значеннями (чи іншими заданими фіксованими даними) функції $F(X)$ в даному інформаційному модулі, які видає на виході НМ, якщо координати X вхідної точки знаходяться у відповідному інформаційному модулі. Вихідний шар складається з одного нейрону F , який виконує операцію підсумовування виходів логічних блоків $AND_1, \dots, AND_k, \dots, AND_K$. Таким чином, НМ рис. 11.3.1 видає на виході ступінчасті значення функції $F(X)$ окремо для кожного інформаційного модуля.

Недоліком схеми рис. 11.3.1 є відображення на виході лише фіксованих даних в кожному інформаційному модулі.

НМ з трикутними активаційними функціями позбавлена цього недоліку (рис. 11.3.2) [115].

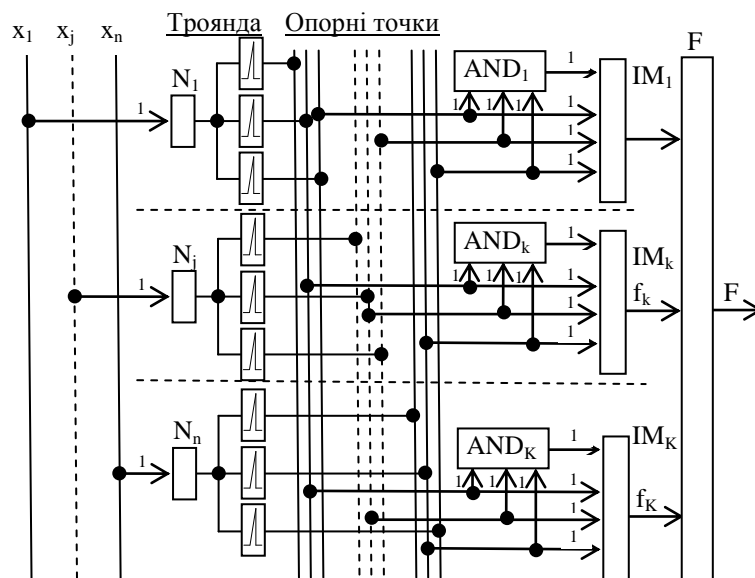


Рис. 11.3.2. Однозначна нечітка НМ з нейронами Троянда, які мають трикутні активаційні функції

Трикутні активаційні функції в межах заданого діапазону вхідної змінної («Велике значення», «Середнє значення» та «Мале значення») змінюються від 0 до 1 в сторону зростання даної змінної x_j і поступають на

входи нейронів інформаційних модулів $IM_1, \dots, IM_k, \dots, IM_K$ та на входи логічних блоків $AND_1, \dots, AND_k, \dots, AND_K$ з ваговими коефіцієнтами 1.

Якщо на заданий нейрон інформаційного модуля $IM_1, \dots, IM_k, \dots, IM_K$ поступає повна призначена йому сукупність вхідних сигналів, то відповідний логічний блок $AND_1, \dots, AND_k, \dots, AND_K$ дає на своєму виході сигнал 1, який дозволяє отримати вихід з відповідного нейрону інформаційного модуля. В кожний момент часу на виході має сигнал лише один логічний блок $AND_1, \dots, AND_k, \dots, AND_K$ та відповідний йому нейрон інформаційного модуля $IM_1, \dots, IM_k, \dots, IM_K$. Всередині нейронів інформаційних модулів $IM_1, \dots, IM_k, \dots, IM_K$ отримані змінні x_j у відносних одиницях можуть перетворюватись у значення реальних координат вхідного вектора X . Можливий варіант подачі змінних вектора X безпосередньо на входи нейронів $IM_1, \dots, IM_k, \dots, IM_K$ при використанні дзвоноподібних активаційних функцій для нейронів N_1, \dots, N_n .

Вихідний шар складається з одного нейрону F , який виконує операцію підсумовування виходів нейронів інформаційних модулів $IM_1, \dots, IM_k, \dots, IM_K$. Таким чином, НМ рис. 11.3.2 видає на виході значення функції $F(X)$ окремо для кожного інформаційного модуля.

Вагові коефіцієнти вихідних функцій $f_1, \dots, f_k, \dots, f_K$ нейронів інформаційних модулів $IM_1, \dots, IM_k, \dots, IM_K$ корегуються у процесі навчання або за дельта – правилом або за матричним рівнянням МНК.

11.3.3. Однозначні нечіткі нейронні мережі

На рис. 11.3.3 наведений зв'язок вхідних даних (x_1, x_2) з основними елементами структурної схеми НМ (рис. 11.3.4) [62; 115].

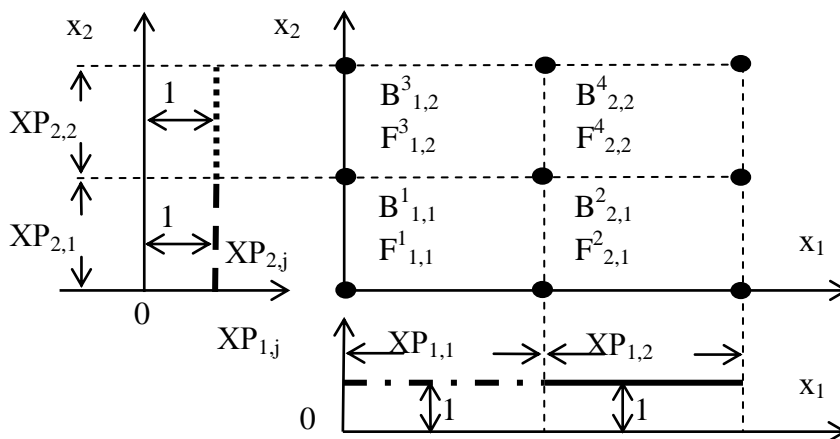


Рис. 11.3.3. Зв'язок вхідних даних (x_1, x_2) з основними елементами структурної схеми НМ

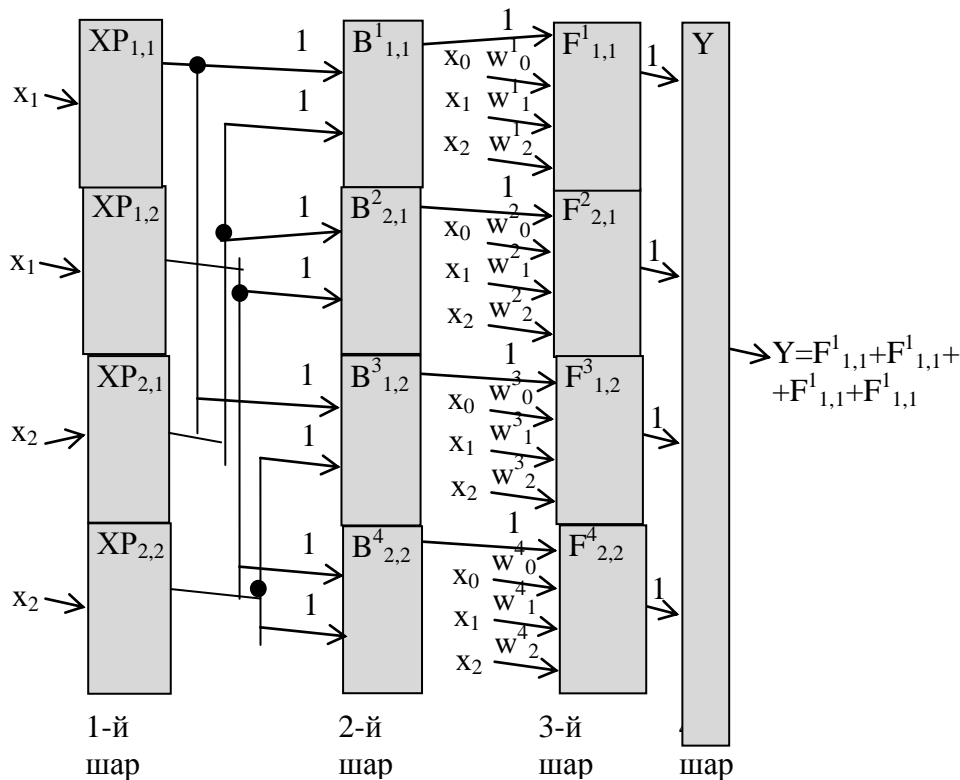


Рис. 11.3.4. Структурна схема нечіткої НМ при двох вхідних змінних

На рис. 11.3.3, рис. 11.3.4 однаково позначені нейрони та їх вихідні сигнали.

На рис. 11.3.3 показано, що ось x_1 контролюється двома нейронами ($XP_{1,1}$, $XP_{1,2}$). Кожний з цих нейронів має вихід 1, якщо координата x_1 має значення, яке входить у показаний діапазон значень для кожного з двох нейронів, та значення 0 у іншому разі. Таку ж саму функцію виконують нейрони ($XP_{2,1}$, $XP_{2,2}$) по відношенню до осі x_2 . В результаті будь-який вхідний вектор $X = (x_1, x_2)$ збуджує два нейрони: один з них з пари ($XP_{1,1}$, $XP_{1,2}$), а інший – з пари ($XP_{2,1}$, $XP_{2,2}$).

На рис. 11.3.3 показано, що будь-яка пара нейронів [$(XP_{1,1}$, $XP_{1,2})$, ($XP_{2,1}$, $XP_{2,2}$)] з різних осей може збудити один з 4-х нейронів $B^1 - B^4$, кожний з яких виконує у даному разі логічну функцію AND (рис. 11.3.4) і, у свою чергу, «вмикають в роботу» один з функціональних блоків $F^1 - F^4$. У випадку більшої кількості змінних можна застосувати логічні нейрони T_AND з багатьма входами [63; 72].

Кожний з нейронів $B^1 - B^4$ при збудженні змінює вихідний сигнал з 0 на 1 і тим самим дає сигнал дозволу виходу нейронів $F^1 - F^4$ (рис. 11.3.3, рис. 11.3.4). Кожний з нейронів $F^1 - F^4$ має на виході значення функції, яка описує потрібну ділянку поверхні виходу. Ці виходи для всіх значень $F^1 - F^4$ об'єднуються нейроном $Y = F_{1,1}^1 + F_{1,1}^1 + F_{1,1}^1 + F_{1,1}^1$, який дає підсумкову функцію виходу НМ. Кожна окрема ділянка поверхні виходу описується власним матричним рівнянням $F = XB$, для якого за методом найменших квадратів визначаються коефіцієнти регресії:

$X^T F = (X^T X)B$; $(X^T X)^{-1} X^T F = (X^T X)^{-1} (X^T X)B$; $B = (X^T X)^{-1} X^T F$,
де X, F – вектори входу та виходу, які задаються експертом; B – коефіцієнти регресії (вони ж – вагові коефіцієнти входів нейронів $F^1 - F^4$); $(X^T X)$ – квадратна матриця, симетрична відносно головної діагоналі; $(X^T X)^{-1} (X^T X)$ – одинична матриця, яку можна опустити, бо усі її елементи дорівнюють нулю, а діагональні елементи дорівнюють одиниці.

На рис. 11.3.5 наведена структурна схема нечіткої НМ при n вхідних змінних.

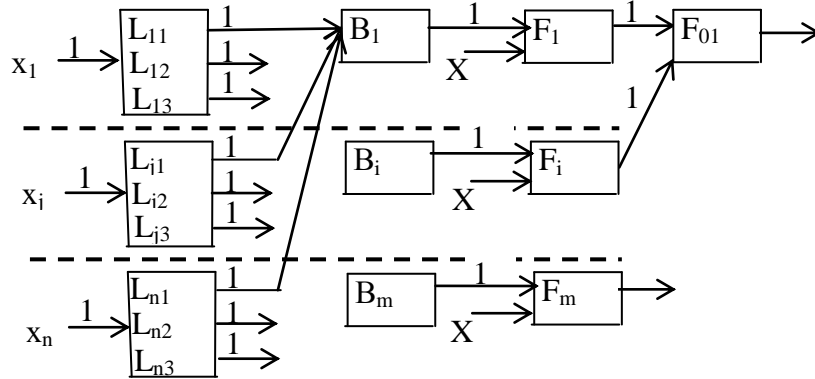


Рис. 11.3.5. Структурна схема нечіткої НМ при n вхідних змінних (тут кожна ось x_j поділена на три ділянки: L_{j3}, L_{j3}, L_{j3})

Навчання вагових коефіцієнтів нейронів $F^1 - F^4$ також може виконуватись по навчальних еталонах за дельта – правилом [115].

Недоліком розглянутої вище НМ є розрахунок виходу всіх інформаційних модулів, в той час, коли виводяться з НМ дані виходу лише одного інформаційного модуля. На рис. 11.3.6 наведена структурна схема нечіткої НМ, яка позбавлена цього недоліку.

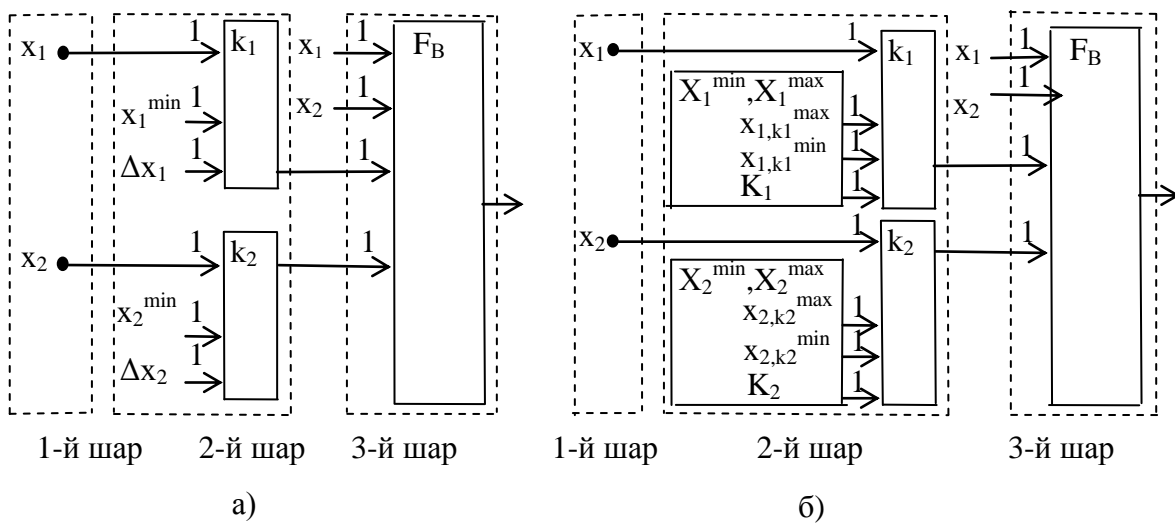


Рис. 11.3.6. Структурна схема нечіткої НМ при навчанні одного вихідного шару: а) для однакових кроків Δx_j по кожній осі x_j ; б) для різних кроків Δx_j по кожній осі x_j .

На рис. 11.3.6, а застосовані позначення:

– 1-й шар, 2-й шар, 3-й шар – позначення шарів нейронів нейронної мережі;

– x_1, x_2 – елементи вхідного вектора $X = (x_1, x_2)$, який потрібно класифікувати;

– 1 – вагові коефіцієнти входів нейронів (вони не змінюються у процесі навчання);

– $\Delta x_1, \Delta x_2$ – числові значення відрізків, на які діляться осі x_1 та x_2 ;

– x_1^{\min}, x_2^{\min} – найменші значення змінних (x_1, x_2), з яких починається розділ на відповідні рівні частки ($\Delta x_1, \Delta x_2$) кожної осі (x_1, x_2);

– k_1, k_2 – позначення двох нейронів 2-го шару, які є однаковими з позначенням виходів цих нейронів. Виходи нейронів k_1, k_2 дорівнюють порядковому номеру відповідних відрізків ($\Delta x_1, \Delta x_2$), на які попала відповідна координата вхідного вектора;

– $k_j = 1, 2, \dots, K_j$ – порядкові номери відрізків, на які ділиться кожна ось змінних вхідного вектора $X = (x_1, x_2, \dots, x_n)$, $j = 1, 2, \dots, n$;

– F_B – позначення одного нейрону 3-го шару, яке є однаковим з позначенням його виходу, що одночасно є виходом всієї нейронної мережі.

У загальному випадку нейрон F_B має на входах з ваговими коефіцієнтами 1:

– значення змінних вхідного вектора $X = (x_1, x_2, \dots, x_n)$, $j = 1, 2, \dots, n$;

– адресу інформаційного модуля у вигляді переліку порядкових номерів відрізків, на які попали координати вхідного вектора $e = (k_1, k_2, \dots, k_j, \dots, k_n)$, $j = 1, 2, \dots, n$.

У загальному випадку для вхідного вектора $X = (x_1, x_2, \dots, x_n)$ за адресою $e = (k_1, k_2, \dots, k_j, \dots, k_n) = \text{const}$, $j = 1, 2, \dots, n$, навчений нейрон F_B по внутрішній навченій матриці вагових коефіцієнтів $W[0^*..n^*, 1..K_1, 1..K_2, \dots, 1..K_j, \dots, 1..K_n]$ визначає всі вагові коефіцієнти вихідного рівняння $w_{e;j^*} = W[j^*, e] = W[j^*, k_1, k_2, \dots, k_j, \dots, k_n] = (w_{e;0^*}, w_{e;1^*}, w_{e;2^*}, \dots, w_{e;j^*}, \dots, w_{e;n^*})$, $j^* = 0^*, 1^*, 2^*, \dots, n^*$, з урахуванням штучно введеної змінної $x_{0^*} = 1 = \text{const}$, та коефіцієнта $w_{e;0^*}$ для визначення вільного члена, в результаті чого на виході нейрона F_B отримуємо поточне значення вихідної функції e -го нейрона

$$F_{e,j^*=0}^n = \sum_{j^*=0^*}^{n^*} w_{e,j^*} x_{j^*}, \quad (11.3.1)$$

де e – адреса відповідного інформаційного модуля у вигляді переліку порядкових номерів ділянок по всіх осях ($k_1, k_2, \dots, k_j, \dots, k_n$), у які попали координати вхідного вектора X ;

$j = 1, 2, \dots, n$ – порядкові номери осей змінних x_j ;

$j^* = 0^*, 1^*, 2^*, \dots, n^*$ – порядкові номери складових рівняння (11.3.1) з урахуванням штучно введеної змінної $x_{0^*} = 1 = \text{const}$, та коефіцієнта $w_{e;0^*}$ для визначення вільного члена вихідної функції.

На рис. 11.3.6, б застосовані позначення:

- 1-й шар, 2-й шар, 3-й шар – позначення шарів нейронів нейронної мережі;
- x_1, x_2 – елементи вхідного вектора;
- 1 – вагові коефіцієнти входів нейронів (вони не змінюються у процесі навчання);
- $x_{1,k1}^{\max}, x_{1,k1}^{\min}$ – записані в векторах X_1^{\max} та X_1^{\min} найбільше та найменше граничні значення координати x_1 для k_1 -го за порядковим номером відрізка $\Delta x_{1,k1}$ серед неоднакових за розмірами відрізків, розміщених вздовж осі x_1 ;
- $x_{2,k2}^{\max}, x_{2,k2}^{\min}$ – записані в векторах X_2^{\max} та X_2^{\min} найбільше та найменше граничні значення координати x_2 для β_2 -го за порядковим номером відрізка $\Delta x_{2,k2}$ серед неоднакових за розмірами відрізків, розміщених вздовж осі x_2 ;
- $k_1 = 1, 2$ – порядкові номери не однакових відрізків $\Delta x_{1,k1}$ вздовж осі x_1 ;
- $k_2 = 1, 2$ – порядкові номери не однакових відрізків $\Delta x_{2,k2}$ вздовж осі x_2 ;
- m_2 – загальна кількість не однакових відрізків вздовж осі x_2 ;
- K_1, K_2 – позначення двох нейронів 2-го шару, виходи (k_1, k_2) яких дорівнюють порядковому номеру відповідних відрізків $(\Delta x_{1,k1}, \Delta x_{2,k2})$, на які попала відповідна координата вхідного вектора;
- F_B – позначення одного нейрону 3-го шару, який має вихід аналогічний розглянутому рівнянню (11.3.1) за винятком визначення порядкових номерів відрізків по осях k_j .

Розглянемо алгоритм роботи НМ згідно рис. 11.3.3 – рис. 11.3.6. Задане *числове* значення координати x_j , або вихід F_e переводиться в однозначне (синонімальне) *лінгвістичне* значення з функцією належності згідно введеного L . Заде поняття лінгвістичної змної. І навпаки – будь-яке лінгвістичне значення з функцією належності однозначно (синонімально) переводиться у числове значення. Подібні однозначні (синонімальні) перетворення чисел та лінгвістичних тверджень можна використати також і для чітких систем. Це означає, що нейронні мережі, створені на основі теорії однозначних нечітких систем [115] можна використати також для аналізу як чітких так і нечітких систем.

11.4. Самонавчальні нейронні мережі

11.4.1. Самонавчальні мережі векторного квантування Кохонена

НМ Кохонена має *шар Кохонена*, що складається з ряду «лінійних формальних нейронів-центроїдів», які є лінійними суматорами. У вагах входів кожного «нейрона-центроїда» записані координати центроїдів – осереднених координат об'єктів, що складають окремий кластер. Кількість кластерів (тобто нейронів-центроїдів) задає експерт.

НМ Кохонена виконує дві функції:

– вона на основі скалярного добутку вхідного вектора і записаних у вагах координат центроїду визначає клас вхідного об'єкту (його належність до кластеру);

– додатково вона має функцію навчання записаних у вагах координат центроїдів. У випадку навчання вихідний сигнал переможця нейрона-центроїда використовується для корегування записаних ваг нейрона-центроїда за правилом «Переможець отримує все» (інші нейрони-центроїди своїх ваг не корегують). По способам навчання ваг центроїдів і по задачах існує багато різновидів НМ Кохонена.

Найбільш розповсюджені з НМ Кохонена:

– самонавчальні мережі *векторного квантування* (пов'язані з кластерним аналізом; методом «динамічних ядер» або К-середніх; навчання з учителем);

– *самоорганізуючі мапи Кохонена.*

Мережа векторного квантування призначена для автоматизованого розділу деякого універсуму образів на окремі кластери по мінімальній евклідовій відстані з використанням принципу «переможець отримує все» та на основі метода «найближчого сусіда». Уперше такі мережі розглянув Т. Кохонен [35; 44; 103; 168].

Одним з найпростіших підходів до розділу універсуму на кластери полягає у тому, що програміст у початку навчання вводить у програму довільну кількість кластерів і довільно (чи стохастично) обирає для них координати центроїдів кластерів, керуючись значеннями координат об'єктів універсуму [35].

Вектори центроїдів кластерів $y_1 \dots y_m$ показані на рис. 11.4.1, а: центроїд – помічена хрестиком точка «тяжіння» окремого i -го кластера (групи, класу) об'єктів універсуму з порядковими номерами об'єктів в середині i -го кластеру $k_i = 1_i, 2_i, \dots, K_i$, $i = 1, 2, \dots, m$ – порядковий номер кластера (класу). Центроїд має осереднені координати всіх об'єктів кластера (класу) і розглядається як опорний представник класу.

В розглянутій нижче НМ кожний нейрон $y_1 \dots y_m$ (рис. 11.4.1, б) описує у вагах своїх входів W_{ji} числові значення координат відповідного центроїда (тобто еталонного опорного представника кластера), але вони призначені для *отримання евклідової відстані* (не для отримання скалярного добутку вхідного сигналу та вектора центроїду). Нейрони $y_1 \dots y_m$ шару векторного квантування Y є *конкуруючими нейронами Кохонена*, які можуть мати зв'язки з усіма іншими нейронами, власні зворотні зв'язки для підсилення власного сигналу та латеральні зв'язки для придушення впливу сигналів, які надходять від інших нейронів (на рис. 11.4.1, б ці додаткові зв'язки не показані).

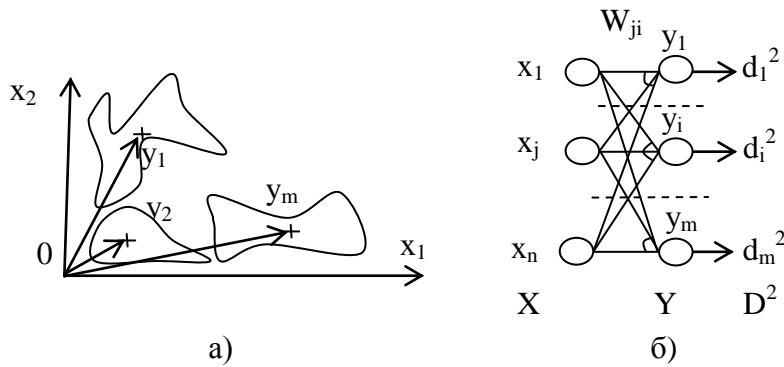


Рис. 11.4.1. Мережа автоматизованої кластеризації об'єктів (векторного квантування)

На входи кожного нейрону $y_1 \dots y_m$ шару векторного квантування Y (рис. 11.4.1) подаються ознаки конкретного об'єкта універсуму у вигляді вектора X . Для кожного нейрону $y_1 \dots y_m$ визначається на виході *евклідова відстань* d_i^2 між вхідним вектором X та вектором ознак центроїду без урахування дії нелінійної активаційної функції (щоб не зіпсувати значення евклідової відстані d_i^2)

$$d_i^2 = \sum_{j=1}^n (x_j - w_{ji})^2 = NET_i$$

де $i = 1, 2, \dots, m$ – порядковий номер кластера та нейрону $y_1 \dots y_m$.

Вважається, що центроїди вступають у конкурентну боротьбу за вхідний вектор X , і переможцем є нейрон $y_1 \dots y_m$ з найбільшою евклідовою відстанню d_i^2 на виході.

Параметри вхідного об'єкта може використовувати для уточнення власних координат лише центроїд – переможець, який «отримує все» [35; 44; 103].

Вказані дії нейронів $y_1 \dots y_m$ відрізняються, наприклад, від дій нейрону Adaline. В результаті розрахунків евклідових відстаней d_i^2 множина векторів універсуму розбивається на кінцеву кількість кластерів, де кожний кластер представлений вектором центроїда.

Додатково кластери нейронів $y_1 \dots y_m$ шару векторного квантування Y набувають упорядкування по класах за рахунок *латеральних зв'язків по впливу сусідства*. Справа у тому, що вихідні нейрони можуть розміщуватись на вихідній площині у вигляді розділених між собою границями довільних множин (рис. 11.4.1, а). Тоді (при корегуванні ваг нейрона-переможця) нейрони, які є найближчими до переможця теж отримують підтримку у вигляді деякого корегування ваг по даним входу (але корегування є менш значущим, ніж у переможця). Більш віддалені сусіди можуть й відштовхуватись від переможця. Ці латеральні зв'язки між сусідами дозволяють швидше визначити кінцевий вигляд кластерів.

При кластеризації можуть виникнути наступні ситуації [35; 103]:

1. Деякі з нейронів ніколи не стають переможцями.
 2. Кластери мають різні розміри; можуть створюватись кластери з одним об'єктом.
 3. Один кластер може охоплювати занадто великий об'єм простору ознак.
 4. Виникає потреба в укрупненні кластерів, або у їх роз'єднанні.
- Існує багато евристичних методів класифікації без учителя, які використовують міру близькості між об'єктами. Загальним їх недоліком є відсутність чіткої формалізації задачі [103].

11.4.2. Самоорганізувальна мапа Кохонена

Структура НМ Кохонена (*самоорганізувальної мапи*) для формування мапи ознак теж може розглядатись на основі кластерів рис. 11.4.1, а. Різниця полягає у тому, що початкові стохастичні кластери задаються автоматично без участі людини. Але кількість кластерів обирає людина. Людина також розміщує вихідні нейрони на площині у лінійку, трикутник, прямокутник, шестикутник тощо. Наступний розділ об'єктів на кластери з корегуванням ваг переможців відбувається аналогічно векторному квантуванню.

Мережа Кохонена призначена для самостійного розділу у процесі навчання на кластери всіх вхідних навчальних еталонних векторів $X_{Ee} = (x_{Ee1}, x_{Ee2}, \dots, x_{Eej}, \dots, x_{Een})$, де $e = 1, 2, \dots, E$ – порядковий номер навчального вектора, без жодної інформації щодо їх належності до класів. Навчена НМ Кохонена здатна класифікувати нові об'єкти $X = (x_1, x_2, \dots, x_j, \dots, x_n)$ шляхом порівняння їх координат з центроїдами кластерів [169; 170; 171].

Мережа Кохонена має *два шари нейронів* (рис. 11.4.2, а):

1. **Перший сенсорний шар** призначений для нормалізації елементів вхідних векторів $X = (x_1, x_2, \dots, x_j, \dots, x_n)$ [а при навчанні – для нормалізації навчальних векторів $X_{Ee} = (x_{Ee1}, x_{Ee2}, \dots, x_{Eej}, \dots, x_{Een})$, де $Ee = 1, 2, \dots, E$ – порядковий номер навчального вектора], а **другий кластерний шар (КШ)** передає X до найближчого кластеру на виході НМ. На рис. 11.4.2,а показані зв'язки лише для однієї ознаки x_1 з нейронами N_λ другого двомірного КШ. Хоча зв'язки інших ознак x_j не наведено, але вони є аналогічними. Крім того, нейрони мають латеральні зв'язки (зв'язки сусідства), які теж не показані.

Нормалізацію вхідних векторів виконують за формулою

$$x_j = \frac{x_j^*}{x_m^*}, \quad (11.4.1)$$

де $x_m^* = \sqrt{x_1^{*2} + x_2^{*2} + \dots + x_j^{*2} + \dots + x_n^{*2}}$ – модуль вхідного вектора; x_j^* – реальне значення j -го елемента вхідних векторів X .

2. **Другий кластерний шар (КШ)** має N_λ кластерних нейронів з n входами, кожний з яких призначений для запису у своїх вагах (після навчання) координат центроїду відповідно отриманому кластеру. Крім того, нейрони N_λ виконують додаткову допоміжну роль – групування навколо себе кластерів з близькими («сусідніми») ознаками центроїдів за рахунок впливу на центроїди сусідніх кластерних нейронів через латеральні зв'язки у вигляді або формул або алгоритмів.

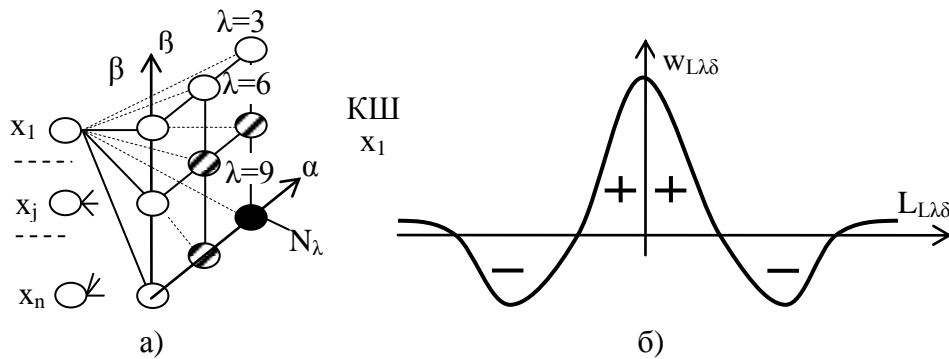


Рис. 11.4.2. Самоорганізувальна мапа мережі Кохонена:

а) структурна схема моделі мережі Кохонена з кластерним шаром (КШ) у площині осей координат α і β ;

б) вплив збудженого нейрону N_λ на сусідні нейрони шару площини (α, β) через латеральну вагу зв'язків $w_{L\lambda\delta} = f_L(L_{L\lambda\delta})$ між нейронами з послідовними номерами λ та δ кластерного шару у залежності від латеральної відстані $L_{L\lambda\delta}$ між ними.

Наприклад, функція латеральних ваг нейронів λ та δ у площині шару з цілочисловими координатами (α, β) звичайно має вигляд «мексиканського капелюха» (рис. 1.4.2,б), який розраховується за формулою

$$w_{L\lambda\delta}(L_{\lambda\delta}, \sigma) = \left[1 - \left(\frac{L_{\lambda\delta}}{\sigma} \right)^2 \right] e^{-(L_{\lambda\delta}/\sigma)^2}, \quad (11.4.2)$$

де σ – ширина «мексиканського капелюха»; $L_{\lambda\delta}$ – латеральна відстань між нейронами (N_λ, N_δ) .

Формулу (11.4.2) слід сприймати як напрямок дій: використовуються і інші формули замість (11.4.2) [103], або замість неї у програмі можуть використати будь-який алгоритм аналогічної дії [35].

Алгоритм навчання. Кластерні нейрони N_λ звичайно розміщуються у вигляді одно- чи двомірних масивів у кластерному шарі КШ. У початку навчання вагові коефіцієнти входів кластерних нейронів N_λ дорівнюють малим випадково визначеним величинам з випадковими знаками «+» та «-».

На рис. 11.4.2, а кластерні нейрони N_λ розміщені у «прямокутнику». Але у принципі вони можуть бути розміщені по лінійці, по трикутниках або шестикутниках, хоча найчастіше все ж використовується розміщення лінійне та по квадратах. Від форми їх групування залежить латеральна дія (вплив «сусідства»).

Алгоритм навчання за латеральними зв'язками демонструється на рис. 11.4.3.

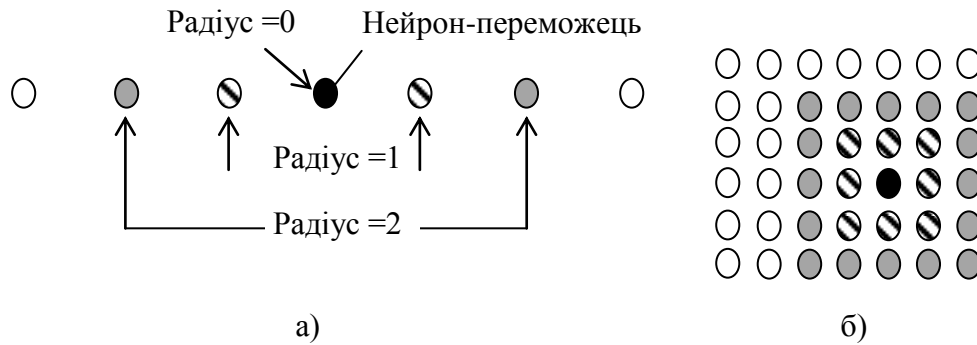


Рис. 11.4.3. Урахування дії нейрона-переможця N_λ (●) по латеральних зв'язках: а) при лінійному розміщенні нейронів КШ; б) при квадратному розміщенні нейронів КШ

У початку навчання радіус латеральних дій обирається максимальним – щоб охоплювати всі елементи. У процесі навчання радіус латеральних дій зменшується і у кінці навчання може охоплювати лише нейрон-переможець, для якого «Радіус =0» (рис. 11.4.3, а). Аналогічні дії виконуються для урахування впливу латеральних зв'язків при квадратному розміщенні нейронів КШ (рис. 11.4.3, б).

Навчальні вектори обираються із навчального універсуму випадковим чином. Алгоритм навчання має наступний вигляд [35]:

1. Для навчального еталонного вектора X_{Ee} на вході НМ розраховується квадратична евклідова відстань між ним і всіма центроїдами всіх кластерних нейронів N_λ

$$d_\lambda^2 = \sum_{j=1}^n (w_{\lambda j} - x_{Eej})^2$$

2. Нейрон N_λ з найменшою квадратичною евклідовою відстанню $d_{\lambda e}^2$ є переможцем. У переможця корегуються ваги центроїда за даними навчального еталонного вектора X_{Ee} на вході по формулі

$$w_{\lambda j}^{HOBE} = w_{\lambda j}^{CTAPE} + \eta^{CTAPE} (x_j - w_{\lambda j}^{CTAPE}).$$

Але за алгоритмом урахування латеральних зв'язків **по цій формулі** виконується оновлення не лише нейрона-переможця, але й ваг центроїдів найближчих кластерних нейронів, якщо вони знаходяться всередині кола заданого радіусу з центром в місці нейрона-переможця. Норма навчання з плином часу зменшуються: наприклад, з початкового значення $\eta=0,9$ зменшуються до деякої величини $\eta=0,01$, яке далі не зменшується. Радіус кола, яке охоплює зону дії латеральних зв'язків теж зменшується (див. рис. 11.4.3). Тому за алгоритмом далі перевіряють, чи потрібно оновлювати норми навчання η чи радіус кола.

Усі дії при навчанні, які пов'язані з обробкою даних нейрона-переможця, зветься «*епохою*». З кожною новою епохою зменшується коефіцієнт навчання і звужується радіус охоплення сусідні нейронів.

Припустимо, що при двох ознаках x_1, x_2 ми маємо чотири вхідних навчальних вектори $X_{E1}, X_{E2}, X_{E3}, X_{E4}$ та два «нейрони-кластери» на виході W_{E1}, W_{E2} . Ми можемо представити їх взаємне розміщення у вигляді рис. 11.4.4.

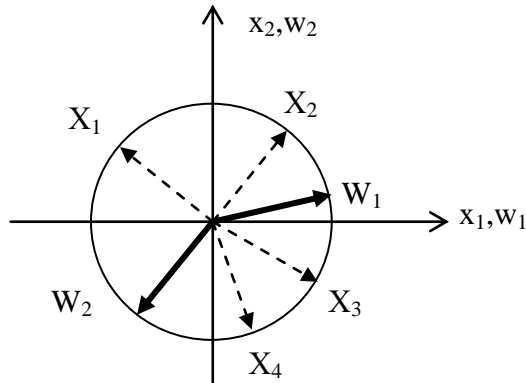


Рис. 11.4.4. Взаємне розміщення вхідних навчальних векторів X_1, X_2, X_3, X_4 та початкових вагових векторів W_1, W_2 нейронів шару КШ

Тому що значення *нормованих* ознак x_j та відповідних ваг $w_{\lambda j}$ мають однаковий діапазон значень ($0 \dots 1$), то відповідні осі координат (x_1, x_2) та (w_1, w_2) *можна вказати разом*. По переміщенню «векторів-кластерів» W_1, W_2 у процесі навчання можна прослідкувати, як об'єднуються навчальні вектори X_1, X_2, X_3, X_4 у кластери. Мережа Кохонена починає навчання з випадково обраного розміщення центроїдів W_1, W_2 і поступово наближує їх до центроїдів кластерів навчальних даних.

Навчання КШ полягає у тому, що кожний навчальний вектор X_1, X_2, X_3, X_4 намагається *так змінити ваги векторів нейронів W_1, W_2 , щоб вони співпали з вхідним вектором X_1, X_2, X_3, X_4* . Таким чином, у даному випадку головною ознакою векторів входу є не величина модуля вектора, а його напрямок. По рис. 11.4.4 зручно прослідкувати, як змінюється напрямок векторів W_1, W_2 при складанні з кожним з векторів X_1, X_2, X_3, X_4 . Можна також запропонувати власний розділ навчальних векторів X_1, X_2, X_3, X_4 на два кластери з призначенням їм позначень W_1, W_2 .

Мапу Кохонена можна відобразити як мапу даних з інформаційним фарбуванням, написами, позначками тощо.

11.4.3. Самонавчальна нейронна мережа з мапою

Існуючий метод Кохонена має погіршені інтелектуальні можливості по наступним причинам:

- застосовується командний метод призначення кількості кластерів;
- командно задається взаємне розташування кластерів у вихідному шарі (у «лінійку», «прямокутник», «трикутник», «шестикутник» – з впливом такого розташування на навчання нейронів [35, с. 85]);
- псується початкова інформація із-за «відштовхування» «чужих» об'єктів від «своїх» кластерів-переможців при використанні латеральних зв'язків;
- складність розрахунків при поступовому наближенню до рішення [103, с. 194, рис. 11.15];
- вилучення з розгляду порожніх кластерів, які не мають навчальних векторів, але несуть інформацію стосовно взаємного їх розміщення.

Розроблена в Україні [78; 79] самонавчальна НМ прямого розповсюдження вхідного сигналу з мапою на основі теорії тілесних кластерів n -вимірного простору (див. розділ 2.4) не має вказаних особливостей.

У порівнянні із самонавчальними НМ Кохонена [35; 103; 168; 169; 170] дослідник не задає кількість кластерів, не визначає їх розміщення, а використовує дані їх реального розміщення та сусідства. По суті **проводиться не навчання нейронної мережі, а групування** всієї сукупності навчальних векторів по кластерах, які упорядковано розміщені в гіперсфері і у які попадають кінцеві точки навчальних векторів.

НМ, що розглядається, може складатись мінімум з двох шарів нейронів, а максимум – з трьох шарів нейронів:

- *вхідний сенсорний шар X* , призначений для отримання, розмноження і нормалізації вхідного сигналу у вигляді вектора з цифровими елементами (кількість нейронів дорівнює кількості ознак вхідного вектора);

- *вихідний кластерний шар Y* , що складається з нейронів-центроїдів, на входах яких у вигляді вагових коефіцієнтів записані координати векторів центроїдів відповідних кластерів (кількість нейронів визначається у процесі навчання кластерів);

- *вихідний шар складних кластерів Y^** , що складається з нейронів, які об'єднують в один клас сигнали кластерів кластерного шару Y . Кількість нейронів цього шару визначається у процесі навчання. Якщо об'єднані кластери відсутні, то вихідний шар складних кластерів Y^* теж відсутній.

Розглянемо деякі дані щодо характеристик класифікації, які використовуються при самонавчанні класифікуючої нейронної мережі.

Згідно теорії тілесних кластерів n -вимірного простору [54] розділяють вектори навчальної сукупності по тілесним кластерам, які у найпростішому вигляді мають вигляд тілесного кута (рис. 11.4.5, а). Для кожного з отриманих «*тілесних кластерів*» можна визначити «*об'єктний кластер*» – перелік належних йому векторів, які у двовимірному можна відобразити у вигляді «*хмари*» векторів, які займають частку тілесних кластерів. Для кожного з об'єктних кластерів можна визначити центроїд – координати

центра тяжіння «хмари» векторів об'єктного кластера (на рис. 11.4.5, б центроїд помічений хрестиком).

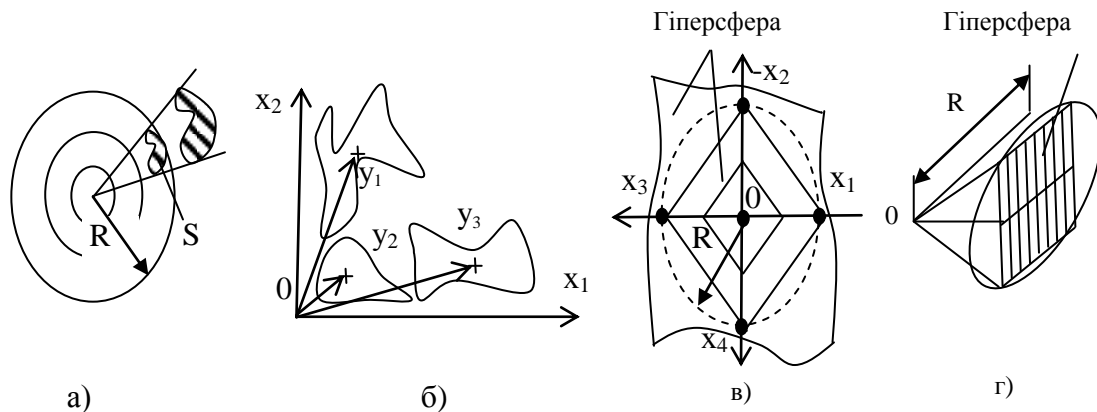


Рис. 11.4.5. Тілесні та об'єктні кластери n -вимірного простору:

- а) гіперкуля із зовнішнім радіусом R з вкладеними одна в одну порожнистими гіперкулями та тілесним гіперкутом, що спирається на площу S гіперсфери радіусом R ;
- б) вигляд «хмар» об'єктних кластерів в двовимірному просторі з їх центроїдами (y_1, y_2, y_3) ;
- в) тілесний кут радіусом R з осями координат $(-x_1, +x_2, +x_3, +x_4)$ і порядковим номером в двоїстому численні $d=(0111)$ та у десятинному численні $g=7$, розрізаний навпіл гіперсферою меншого радіусу (розріз показаний меншим прямокутником);
- г) двовимірна «мапа» (таблиця) тілесних кутів n -вимірного простору (при $n = 4$) у кількості $2^n = 2^4 = 32$ на поверхні гіперсфери.

Тілесний кут у чотирьохвимірному просторі гіперкулі радіусом R з центром в точці «0» і з осями координат $(-x_1, +x_2, +x_3, +x_4)$ має порядковий номер в двоїстому численні $d=(0111)$ та у десятинному численні $g=7$ і (при зменшенні площі поверхні гіперсфери за допомогою масштабу) має вигляд рис. 11.4.5, в, якщо дивитись з поверхні гіперсфери у центр «0» гіперкулі. Даний тілесний гіперкут розрізаний навпіл гіперсферою меншого радіусу.

Для $n=4$ сукупність подібних тілесних кутів у кількості $2^n = 16$ можна представити на гіперсфері у вигляді двовимірної мапи n -вимірного простору рис. 11.4.5, г. На поверхні такої «мапи» в тілесних кутах можна відобразити геометричні образи, які характеризують тіло гіперкулі. Дану гіперкулю у вигляді «піраміди» рис. 11.4.5, г можна розрізати на кілька поставлених одна на одну «пірамід»; на кожній отриманій таким чином гіперсфері в тілесних кутах можна візуально відобразити кінцеві точки об'єктів, ізолінії залежної від n -вимірного простору складної функції тощо. Потрібно лише урахувувати, що за аналітичною геометрією в n -вимірних тілесних кутах розміщення T -кутів може бути довільним без урахування вимог сусідства.

При класифікації детермінованих даних також може бути застосованим наведений нижче алгоритм аналізу стохастичної кластерної нейронної мережі (розділ 11.5.3).

Вхідний сенсорний шар такої НМ складається з n нейронів з одним входом з ваговим коефіцієнтом 1 та з одним виходом і призначений для нормалізації та розмноження елементів вектора вхідного сигналу $X = (x_1, x_2, \dots, x_j, \dots, x_n)$, де $j = 1, 2, \dots, n$ – порядковий номер елемента вхідного сигналу.

Нормалізовані елементи вектора X передаються на кластерний шар, що складається з m нейронів-центроїдів, які мають позначення $Y = (y_1, y_2, \dots, y_i, \dots, y_m)$, де $i = 1, 2, \dots, m$ – порядковий номер нейрона-центроїда, кожний з яких має записаний на його входах у вигляді вагових коефіцієнтів вектор центроїда i -го кластера $W_i = (w_{i1}, w_{i2}, \dots, w_{ij}, \dots, w_{in})$. На виході нейрона центроїда отримується скалярний добуток вхідного вектора та еталонного вектора-центроїда (або відстань між ними). Найбільше значення скалярного добутку (чи найменша відстань між ними) визначає переможця y_i серед нейронів-центроїдів шару Y .

Блок навчання шару Y по принципу «переможець отримує все» у процесі навчання уточнює значення вагових коефіцієнтів $W_i = (w_{i1}, w_{i2}, \dots, w_{ij}, \dots, w_{in})$ нейрона-переможця y_i шару Y з урахуванням додаткових внутрішніх даних (наприклад, може ураховуватись оцінка впливу змінної на розпізнавання образу).

У процесі навчання блок навчання через n входів з ваговими коефіцієнтами 1 отримує з вхідного сенсорного шару всі навчальні нормалізовані вектори вхідних сигналів $X_e = (x_{e1}, x_{e2}, \dots, x_{ej}, \dots, x_{en})$, де $e = 1, 2, \dots, E$ – порядковий номер навчального вектора, визначає зовнішній радіус гіпершару R_m , в середині якого розміщуються всі отримані навчальні вектори і на основі зовнішньої експертної інформації встановлює три характеристики для системи навчання:

- порядкові номери та радіуси вкладених один в одній порожнистих гіпершарів, радіуси яких повинні бути меншими за зовнішній радіус гіпершару;

- максимальне дозволене значення відстані між навчальними векторами, яке не повинне перевищуватись всередині будь-якого кластера;

- максимальне значення відстані, яке не повинне перевищуватись між центроїдами двох кластерів, що об'єднують у складний кластер.

Далі для кожного навчального вектора X_e визначають двомірні координати кластера у вигляді порядкового номеру радіусу гіпершару та порядкового номеру тілесного кута, у межах яких розміщені кінцеві точки навчальних векторів, виділяють кластери, всередині яких відстані між навчальними векторами не перевищують дозволені значення, визначають кількість нейронів у шарі Y , яка дорівнює загальній кількості отриманих великих і малих кластерів, визначають вектори центроїдів всіх отриманих кластерів по координатах їх навчальних векторів, визначають відстані між всіма парами векторів-центроїдів отриманих кластерів з метою об'єднання пар кластерів у складний кластер.

За кінцевими отриманими кластерами закріплюють їх центроїди, а за складними кластерами – сукупності центроїдів кластерів, що утворюють складний кластер.

Самонавчання завершується тим, що при наявності складних кластерів утворюють додатковий шар вихідних нейронів складних кластерів Y^* , кожний нейрон якого має входи з ваговими коефіцієнтами 1, що з'єднані з виходами тих нейронів вихідного шару Y , які у сукупності складають один складний кластер. Складний кластер дає на виході сигнал «1» при наявності на його входах хоча б одного вхідного сигналу «1» від належного йому простого кластера.

11.4.4. Мережи зустрічного розповсюдження

Мережа зустрічного розповсюдження запропонована Р. Хехт-Нілсеном (R.Hecht-Nielsen, 1987) [103, с. 197; 159]. НМ об'єднує два відомі алгоритми: *шару Кохонена та зірки Гросберга*. На рис. 11.4.6 показана повна мережа зустрічного розповсюдження. Вектори X та Y можуть бути двоїчними, безперервними, неповними і навіть невірними. Кожний нейрон шару вектора X пов'язаний з кожним нейроном шару Кохонена, а кожний нейрон шару Кохонена пов'язаний з кожним нейроном шару Гросберга. При подачі на вхід лише одного вектора X , на виході з'являються копії обох векторів X^* та Y^* . НМ має можливість *одержувати вихідні сигнали по вхідним і навпаки*.

Спрощена версія прямої дії мережі використовує один вхід X і один вихід Y^* .

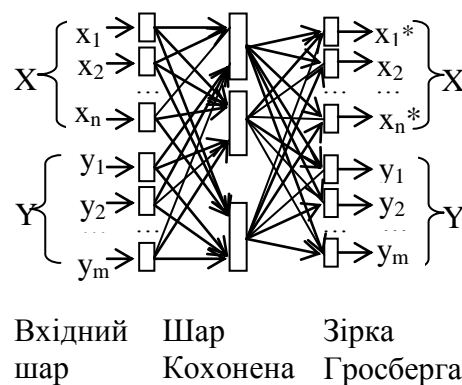


Рис. 11.4.6. Повна НМ зустрічного розповсюдження

При навчанні на входи НМ подають еталонні нормалізовані навчальні вектори X та Y , а на виходах намагаються отримати копії цих же векторів.

Нейрони шару Кохонена функціонують по принципу «переможець забирає все»: з усієї сукупності нейронів Кохонена лише один має на виході сигнал «1», а всі інші мають «0».

НМ зустрічного розповсюдження реалізує переваги шару Кохонена та мережі Гросберга і підтверджує можливість успішного блокового будівництва НМ для отримання логічних висновків. Вона використовується в задачах апроксимації функцій, розпізнаванні образів, класифікації, стиснення даних, обробки і відновлення зображень [103].

Об'єднання двох мереж надає нові можливості, яких немає у кожного з указаних алгоритмів окремо. Мережі зустрічного розповсюдження мають наступні переваги у порівнянні з одношаровими НМ:

- час навчання у них зменшується приблизно в **100 разів** (ми вводимо значення ваг входу та виходу);
- надається можливість **«стиснення зображень»** – передачі даних при меншому обсязі пам'яті (передаємо лише номер переможця);
- отримуються **статистичні дані** із множини вхідних даних (бо ми їх групуємо по класах);
- **відновлюються образи** (на виході ми можемо вказати еталонні сигнали).

При навчанні мережі зустрічного розповсюдження спочатку розглядають шар Кохонена, а потім – шар Гросберга. Як звичайно при роботі з усіма нейронними мережами, всім вагам привласнюються невеликі випадкові значення з випадковим знаком «±». При навчанні шару Кохонена випадково обрані вагові вектори треба нормалізувати. Кожна вага нейронів шару Гросберга корегується лише у тому випадку, якщо вона з'єднана з нейроном-переможцем Кохонена, який має ненульовий вихід. Шар Кохонена навчається сам, без учителя. *При навчанні може використовуватись:*

- *додання «шуму» до векторів виходу* (це підвищує ефективність навчання);
- надання нейрону *«почуття справедливості»*: якщо нейрон стає переможцем частіше, ніж $1/k$, де k – число нейронів Кохонена, то він не вмикається, даючи можливість навчатися й іншим нейронам;
- використовують *«метод інтерполяції»*, за яким ціла група нейронів Кохонена, що мають найбільші виходи, може передавати свої вихідні сигнали в шар Гросберга (це підвищує точність рішень).

З метою виграшу часу **при передачі зображень або мови**, мережа зустрічного розповсюдження може бути використана для стиснення даних передачі (**стиснення кількості бітів від 10 до 100 разів**). Для цього прямокутне зображення розбивають на ряд підобразів $S_{11}, \dots, S_{ij}, \dots, S_{mn}$, кожний з яких складається з сукупності пікселів. Замість передачі всіх пікселів, кожному підобразу (сукупності пікселів) привласнюють номери, які і передаються (як більш короткі) замість реального зображення. Це й скорочує час передачі.

11.5. Стохастичні нейронні мережі

11.5.1. Напрямки аналізу стохастичних нейронних мереж

На відміну від машини Больцмана, функціонування стохастичних НМ є *детермінованим*. Нижче перелічені основні напрямки аналізу.

1. **Метод кластерів** є найліпшим для використання у НМ, призначених для розпізнавання стохастичних об'єктів, але при цьому об'єкти різних класів повинні розміщуватись компактними окремими групами у просторі ознак [30; 35; 38; 84; 103; 199]. Кластер (група об'єктів) замінюється одним абстрактним об'єктом (центроїдом), стосовно якого виконуються всі розрахунки.

2. «**Метод найближчого сусіда**», коли вхідний вектор набуває класу ближчого сусіда у вибірці, дає ліпші результати, якщо стохастичні об'єкти перемішані між собою. Але такий підхід теж дає похибки, бо не ураховує розподіл густини ймовірностей класів.

3. **Метод байєсового класифікатора**, введеного у 1988р. Д. Ф. Спехтом, є більш результативним при визначенні класу невідомого вектора X [35; 193; 194], якщо «хмари» об'єктів вибірок різних класів перемішані у просторі ознак між собою. По максимальній величині байєсового класифікатора

$$p^k c^k f^k(X), \quad (11.5.1)$$

де p^k – апіорна ймовірність появи події класу k ; c^k – ціна похибки класифікації при невірному визначенні класу вхідного вектора X , який насправді належить до класу k (тобто c^k – це підсумкова ціна витрат при появі об'єкта X^k , якого визначили за іншим класом); $X = (x_1, x_2, \dots, x_j, \dots, x_n)^T$ – транспонований вхідний вектор невідомого класу; $j = 1, 2, \dots, n$ – порядковий номер змінної x_j вхідного вектора X невідомого класу; $f^k(X)$ – значення функції розподілу густини ймовірностей класу k , яка відповідає введеному вхідному вектору X невідомого класу, визначають серед усіх класів клас переможця, до якого належить вхідний вектор X .

4. **Метод «вікон Парзена»** дозволяє замінити у байєсовому класифікаторі (11.5.1) функцію розподілу густини ймовірностей $f^k(X)$ класу k *на її оцінку*, яку отримують найчастіше за допомогою функцій Гауса [118; 183]. При цьому іноді приймають добуток $p^k c^k$ однаковим для всіх класів.

Вікна Парзена визначаються по навчальній статистичній вибірці. Для кожного навчального екземпляра $X^{e_k} = (x_1^{e_k}, x_2^{e_k}, \dots, x_j^{e_k}, \dots, x_n^{e_k})^T$, де $e_k = 1, 2, \dots, E_k$ – порядковий номер навчального екземпляра k -го класу, у вибірці застосовують *функцію потенціалу* (найчастіше у вигляді наведеної нижче функції Гауса), яка використовується *для оцінки* функції розподілу густини ймовірностей відповідного класу $f^k(X)$

$$g^k(X) = \sum_{e_k=1}^{E_k} \sum_{j=1}^n \left[\exp - \left(\frac{x_j - w_j^{e_k}}{\sigma} \right)^2 \right],$$

де x_j – поточна координата j -ої ознаки вхідного вектора $X = (x_1, x_2, \dots, x_j, \dots, x_n)^T$ невідомого класу; $j = 1, 2, \dots, n$ – порядковий номер ознаки x_j ; $k = 1, 2, \dots, K$ – порядковий номер класу; K – загальна кількість класів; $w_j^{e_k} = x_j^{e_k}$ – координата навчального екземпляра k -го класу з порядковим номером $e_k = 1, 2, \dots, E_k$; σ – параметр, який впливає на «гладкість» функції $g^k(X)$.

Оцінка функції густини ймовірностей $g^k(X)$ належності вхідного об'єкту до визначеного класу виконується на основі наступних тверджень. Координати **одного** навчального об'єкта вказують у просторі ознак точку, в якій можуть знаходитись об'єкти його класу. Поблизу цієї навчальної точки ймовірність належності сусідніх точок простору до даного класу є більшою, а у віддалених точках ця ймовірність зменшується, як у функції Гауса.

Чим більше точок даного класу знаходиться в околі деякої точки (рис. 11.5.1), тим більшим буде підсумок гаусових функцій, який приймається за **оцінку** густини ймовірностей одного класу.

Внесок в загальну оцінку густини ймовірностей (рис. 11.5.1), який вносить **один нейрон** (одного деякого класу k шару A з порядковим номером e_k у вибірці) **вздовж однієї осі x_j** дорівнює

$$g_j^k(x_j) = \exp - \left(\frac{x_j - w_j^{e_k}}{\sigma} \right)^2$$

і має вигляд «капелюха», наведеного на рис. 11.5.1 для однієї точки (одного навчального об'єкта) безперервною лінією. Малі величини σ відповідають «гостроверхим капелюхам»; значення параметра σ обирає програміст у процесі дослідження. Для трьох рядом розміщених навчальних об'єктів однакового класу відповідні функції наведені штриховими лініями. Підсумок ординат функції, наведених штриховими лініями, буде показувати підвищену оцінку значення функції $f^k(X)$.

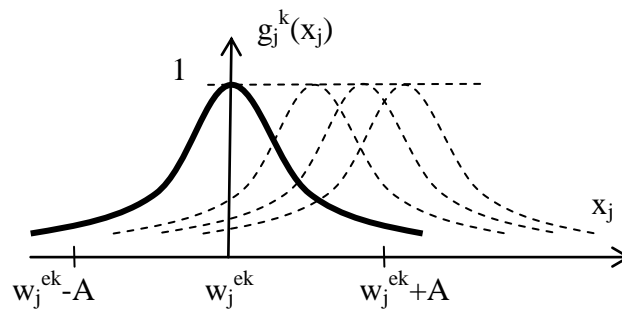


Рис. 11.5.1. Функція Гауса для однієї змінної x_j (A – постійна величина)

11.5.2. Архітектура стохастичної нейронної мережі Спехта

Архітектура стохастичної нейронної мережі Спехта (Probabilistic Neural Network – PNN) наведена на рис. 11.5.2 [193].

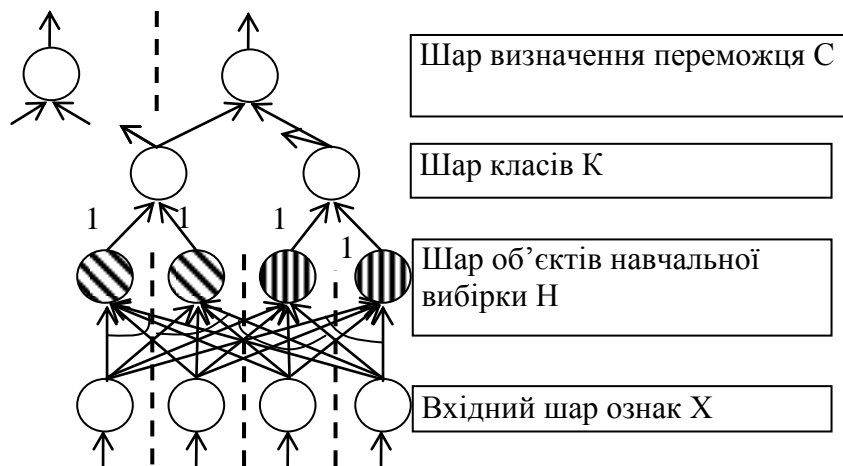


Рис. 11.5.2. Архітектура стохастичної нейронної мережі Спехта з використанням «вікон Парзена»

У вхідному шарі ознак X кількість нейронів дорівнює кількості елементів вхідного вектора $X=(x_1, \dots, x_j, \dots, x_n)$ невідомого класу, який повинен бути нормалізованим.

У шарі об'єктів навчальної вибірки H кількість нейронів дорівнює $E = \sum_{k=1}^K E_k$ – загальній кількості векторів навчальної вибірки, і вони згруповані по класах k навчальної вибірки: на рис. 11.5.2. нейрони однакового класу k виділені однаковим узором. Необхідність запам'ятовувати велику кількість об'єктів вибірки є недоліком НМ Спехта, бо перевантажує пам'ять НМ.

Параметри кожного еталонного навчального вектора вибірки $X^{e_k} = (x_1^{e_k}, x_2^{e_k}, \dots, x_j^{e_k}, \dots, x_n^{e_k})^T$, де $e_k = 1, 2, \dots, E_k$ – порядковий номер навчального екземпляра k -го класу; $x_j^{e_k}$ – координати еталонного навчального вектора у вибірці, відображені у відповідному одному нейроні шару H , який має на входах вагові коефіцієнти

$(w_1^{e_k}, w_2^{e_k}, \dots, w_j^{e_k}, \dots, w_n^{e_k})^T = (x_1^{e_k}, x_2^{e_k}, \dots, x_j^{e_k}, \dots, x_n^{e_k})^T$. Вихід такого нейрона звичайно визначається по функції Гауса

$$g^{e_k}(X) = \sum_{j=1}^n \left[\exp - \left(\frac{x_j - w_j^{e_k}}{\sigma} \right)^2 \right].$$

В шарі класів K по сукупності виходів з шару N такої «групи нейронів класу k » одним підсумовуючим нейроном вихідних функцій Гауса з вагами входів «1» отримується $f^k(X)$ – оцінка значення функції розподілу густини ймовірностей класу k . В результаті кожний один нейрон шару класів K на виході видає **оцінку густини ймовірностей по власному класу** [35] за формулою

$$g^k(X) = \sum_{e_k=1}^{E_k} g^{e_k}(X) = \sum_{e_k=1}^{E_k} \sum_{j=1}^n \left[\exp - \left(\frac{x_j - w_j^{e_k}}{\sigma} \right)^2 \right]. \quad (11.5.2)$$

Вихідний шар C визначає переможця і виконує роль селектора, який серед нейронів шару класів K виділяє переможця з найбільшим вихідним сигналом $g^k(X)$, який є оцінкою функції $f^k(X)$.

Примітка. Хоча ми використовуємо формулу Гауса, але з рівним успіхом для **оцінки густини ймовірностей** може бути використаним й розподіл Коші за формулою ймовірності

$$p_{ej}^{\alpha}(x_j, \theta) = \frac{1}{\pi [1 + (x_j - \theta)^2]},$$

де θ – центр симетричного розподілу.

Недоліками розглянутої стохастичної НМ є: великий об'єм потрібної пам'яті, в яку записується вся навчальна вибірка; збільшення часу отримання рішення внаслідок ускладнення розрахунків [35, с. 163].

Нижче розглядається стохастична кластерна НМ, яка дозволяє створювати стохастичні кластери довільної форми і розміщення.

11.5.3. Стохастична кластерна нейронна мережа

В основу наведеної нижче нейронної мережі покладена розглянута вище теорія тілесних кластерів в n -вимірних тілесних кутах [54; 78; 79].

Хоча наведені нижче алгоритми відносяться до стохастичних кластерних нейронних мереж, але вони можуть застосовуватись для класифікації будь-яких множин детермінованих чи стохастичних об'єктів, в тому числі й при виконанні статистичного аналізу з розділом множин на групи.

Спочатку розглянемо деякі основні визначення, які використовуються нижче при описі стохастичної нейронної мережі.

Порівняння тілесного кута (T -кута) і кластера. Нижче у n -вимірному гіпероб'ємі, поряд з поняттям **об'єктного кластера**, як підмножини найбільш схожих навчальних векторів з числовими змінними та з визначенням для кластера типового представника (**центроїда**) як вектора з середніми значеннями координат навчальних векторів, що складають клас-

тер), також використовується більш широке поняття *тілесного кута (Т-кута)*.

Т-кутом зветься частина n -вимірної гіпероб'єму, яка об'єднує усі вектори, промені яких (напрямки векторів) виходять з центру координат і перетинають деяку гіперповерхню S на гіперсфері радіусом R (рис. 11.5.3).

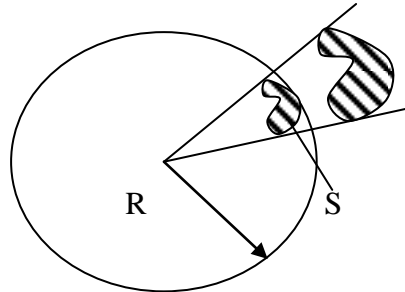


Рис. 11.5.3. Гіпершар радіусом R_m з тілесним кутом

Т-кут відрізняється від кластера тим, що охоплює не лише точки координат одного або кількох векторів X^{e_k} кластера, але й нескінченну кількість точок всього простору гіперкулі між точками з векторами X^{e_k} .

Разом з тим для Т-кута, який охоплює один кластер, при аналізі можна використовувати всі характеристики цього кластера: позначення кластера, його центроїд, кількість векторів кластера X^{e_k} , параметри конкретних векторів X^{e_k} .

Нумерація Т-кутів. Розділ Т-кутів на частки. Як і при ієрархічній кластеризації великі Т-кути також можуть бути розділені на ієрархічно нижчі дрібніші Т-кути (розглядається нижче).

Вважаємо, що початкова навчальна вибірка стохастичних векторів $X^{e_k} = (x_1^{e_k}, x_2^{e_k}, \dots, x_j^{e_k}, \dots, x_n^{e_k})^T$, де $e_k = 1, 2, \dots, E_k$ – порядковий номер навчального екземпляра k -го класу, чи будь-яка група з векторів вибірки можуть бути:

– **«типу 0»**, якщо серед векторів X^{e_k} навчальної вибірки чи окремих груп вибірки відсутні вектори різних класів чи вектори з іншими *розділювальними обмеженнями*, під якими розуміють заборону мати у навчальній вибірці чи у її групі вектори X^{e_k} із забороненими відстанями між векторами або із забороненими стохастичними показниками вибірки чи її групи та ін.;

– **«типу 1»**, якщо серед векторів X^{e_k} навчальної вибірки чи окремих груп вибірки присутні розділювальні обмеження.

Розглянемо порядкову нумерацію Т-кутів, яка використовується як адреса Т-кута. Далі вважаємо, що кожний Т-кут, якщо не оговорено інше,

виділяється взаємно перпендикулярними n осями координат. В n -вимірному просторі прямокутної системи координат позначення осей координат $(x_1, x_2, \dots, x_j, \dots, x_n)$ по знаках «+» чи «-» їх числових значень замінюють на коди «1» чи «0». Сукупність отриманих кодованих значень осей координат $(x_1^C, x_2^C, \dots, x_j^C, \dots, x_n^C)$ утворює порядковий номер відповідного Т-кута у двоїстій системі числення d . Цей двоїстий порядковий номер Т-кута d можна перевести у порядковий номер десятинної системи числення g . Т-кут виділяється в гіперкулі сукупністю площин, кожна з яких охоплює початок координат та дві сусідні осі координат.

На рис. 11.5.4 як приклад показані порядкові номери Т-кутів в десятинній системі числення $g = 0, g=1, g=2, g=3$ для осей координат (x_1, x_2) .

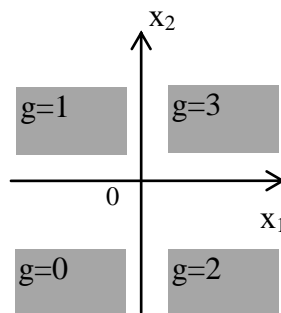


Рис. 11.5.4. Порядкові номери з тілесних кутів у десятинній системі числення

На рис. 11.5.4 позначено: x_1, x_2 – осі координат двовимірного простору; $g = 0, g=1, g=2, g=3$ – порядкові номери Т-кутів в десятинній системі числення. Порядкові номери Т-кутів отримуються спочатку у двоїстій системі числення, а потім переводяться у десятинну систему числення:

- Перший квадрант: $(+x_1, +x_2) \rightarrow d = (1, 1) \rightarrow g = 3.$
- Другий квадрант: $(-x_1, +x_2) \rightarrow d = (0, 1) \rightarrow g = 1.$
- Третій квадрант: $(-x_1, -x_2) \rightarrow d = (0, 0) \rightarrow g = 0.$
- Четвертий квадрант: $(+x_1, -x_2) \rightarrow d = (1, 0) \rightarrow g = 2.$

Аналогічним чином можна отримати порядкові номери Т-кутів для системи координат з великою кількістю осей координат. Отримані порядкові номери g не співпадають зі звичною загальноприйнятою нумерацією квадрантів на площині, а також не співпадають з відомою нумерацією октантів в трьохвимірній системі координат.

Таким чином, для стохастичного вектора невідомого класу $X = (x_1, x_2, \dots, x_j, \dots, x_n)$ з числовими змінними можна отримати відповідну двоїсту сукупність $d = (x_1^C, x_2^C, \dots, x_j^C, \dots, x_n^C)$, елементи якої складаються з 0 та 1 та яка є порядковим номером Т-кута вектора X у двоїстій системі числення d . Двоїстий порядковий номер $d = (x_1^C, x_2^C, \dots, x_j^C, \dots, x_n^C)$ можна перевести у десятинний порядковий номер g .

Логічні нейрони «T_AND_0». Для можливості виділення окремих Т-кутів по їх порядковим двоїстим номерам $d = (x_1^C, x_2^C, \dots, x_j^C, \dots, x_n^C)$,

використовують нейрони «T_AND_0» [63], які виконують складну логічну функцію (рис. 11.5.5).

На рис. 11.5.5 позначені вихідна змінна та входні змінні нейрона «T_AND_0», який виконує складну логічну функцію:

– $x_j^\alpha, j = 1, 2, \dots, n^\alpha$ – входні змінні, які всі повинні мати призначені їм значення «1» для отримання виходу $y=1$ нейрона «T_AND_0».

– $x_j^\beta, j = n^\alpha + 1, n^\alpha + 2, \dots, n$ – входні змінні, які всі повинні мати призначені їм значення «0» для отримання виходу $y=1$ нейрона «T_AND_0».

y – вихід нейрона «T_AND_0», який дорівнює 1, якщо значення всіх входів « $x_j^\alpha = 1$ » та всіх входів « $x_j^\beta = 0$ » дотримуються сумісно; в іншому разі вихід $y=0$.

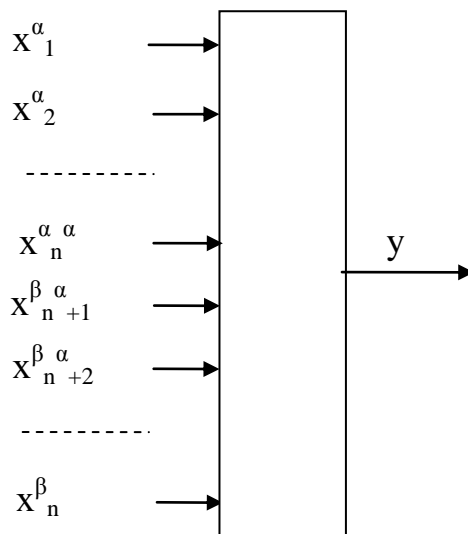


Рис. 11.5.5. Структурна схема нейрона T_AND_0

На відміну від звичайної логічної функції AND з двома входними змінними (x_1, x_2) зі значеннями 0 або 1, нейрон «T_AND_0» виконує аналогічну логічну операцію з n входними значеннями змінних ($x_1, x_2, \dots, x_j, \dots, x_n$), причому для отримання на виході сигнала 1 частка з входних змінних (на рис. 11.5.5 – змінні $x_j^\alpha, j = 1, 2, \dots, n^\alpha$) повинна мати значення 1, а інша частка з входних змінних (на рис. 11.5.5 – змінні $x_j^\beta, j = n^\alpha + 1, n^\alpha + 2, \dots, n$) повинна мати значення 0.

Будування нейронної мережі по даним стохастичної навчальної вибірки. Архітектура нейронної мережі є індивідуальною для кожної стохастичної навчальної вибірки і визначається по даним її векторів.

Загальним для всіх отриманих таким чином нейронних мереж є лише кількість і призначення шарів (два або три шари), а кількість нейронів та їх зв'язки визначаються у процесі обробки вибірки. Побудована стохастична НМ не потребує навчання, оскільки вона відображує особливості конкретної навчальної вибірки.

Припустимо, що кінцеві точки векторів $e = (1 \dots 14)$ стохастичної навчальної вибірки для трьох класів відображені на рис. 11.5.6.

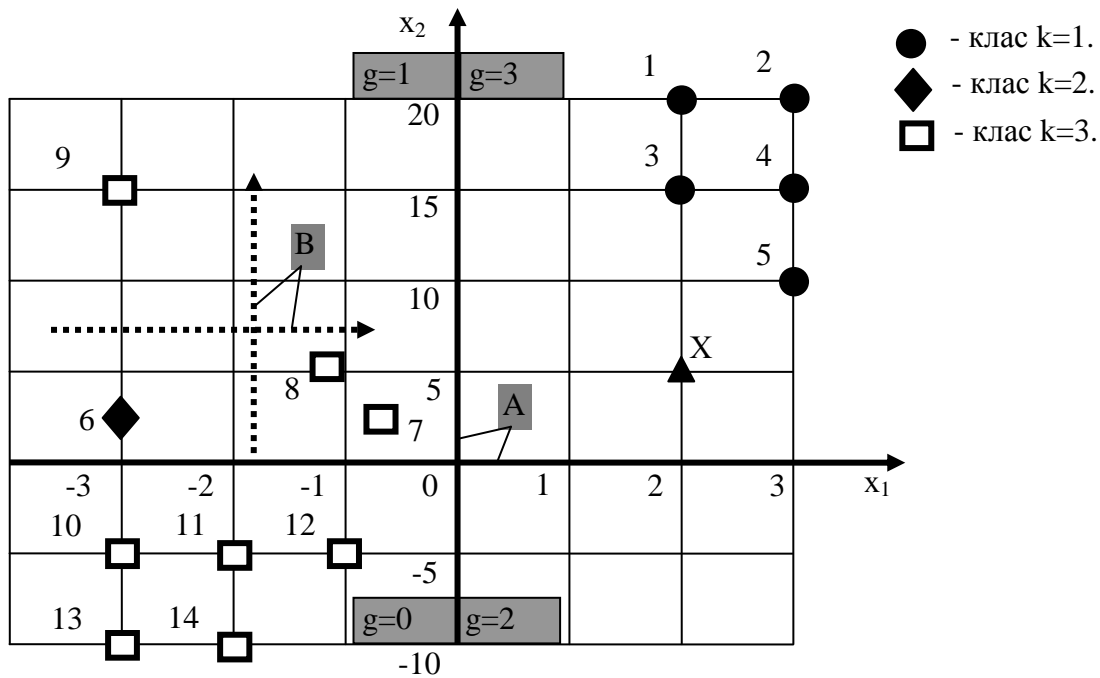


Рис. 11.5.6. Дані навчальних векторів стохастичної навчальної вибірки

На рис. 11.5.6 позначено:

– x_1, x_2 – позначення осей початкової системи координат, яка помічена літерою А. Пунктиром наведена позначена літерою В нова система координат для ієрархічно нижчих і менших за розмірами Т-кутів навчальних векторів без позначення її осей, бо вони співпадають з позначеннями і напрямками наведених осей (x_1, x_2), позначеними літерою А.

– $g = 0, g=1, g=2, g =3$ – порядкові номери Т-кутів в десятинній системі числення (виділені у прямокутниках з сірим фоном) для початкової системи координат, яка позначена літерою А. Ці порядкові номери є вірними й для системи координат, поміченої літерою В.

– $e = (1, 2, 3, 4, 5); e = (6); e = (7, 8, 9, 10, 11, 12, 13, 14)$ – кінцеві точки навчальних векторів X^{e_k} класів $k=1, k=2, k=3$.

– X – точка, яка відображує координати (x_1, x_2) вхідного вектора X невідомого класу.

Стохастична нейронна мережа, яка побудована по даним вибірки рис. 11.5.6, представлена на рис. 11.5.7. На рис. 11.5.7 позначено:

– 1 – вагові коефіцієнти входів нейронів, які не змінюються у процесі навчання нейронної мережі.

– $(x_1^{C,\lambda=1}, x_2^{C,\lambda=1}); (x_1^{C,\lambda=2}, x_2^{C,\lambda=2})$ – нейрони «1-го шару кодування Т-кута» нейронної мережі, які складають ієрархічні рівні $\lambda=1, \lambda=2$.

– AND_A0, AND_A1, AND_A3 – нейрони типу «T_AND_0» [63]. Кінцеві частки позначень у вигляді «A0», «A1», «A3» означають: літера

«А» – позначення системи осей координат на рис. 11.5.6; кінцеві цифри 0, 1, 3 – порядкові номери Т-кутів в десятинній системі числення g_u вказаній системі координат А на рис. 11.5.6. У даному випадку замість нейронів типу «Т_AND_0» можна було б використати звичайні нейрони «AND» з двома вхідними змінними (x_1, x_2), але застосування нейронів «Т_AND_0» призначене для визначення архітектури мережі, коли кількість змінних є довільною.

– AND_B0, AND_B1, AND_B2 – нейрони типу «Т_AND_0». Кінцеві частки позначень «B0», «B1», «B2» означають: літера «В» – позначення системи осей координат на рис. 11.5.6; кінцеві цифри 0, 1, 2 – порядкові номери Т-кутів в десятинній системі числення g_u у вказаній системі координат В на рис. 11.5.6.

– k3 – нейрон – суматор 3-го шару нейронів, на виході якого отримують вихідний сигнал для класу k3 нейронної мережі.

Вихідні сигнали всіх нейронів помічаються по їх позначенню.

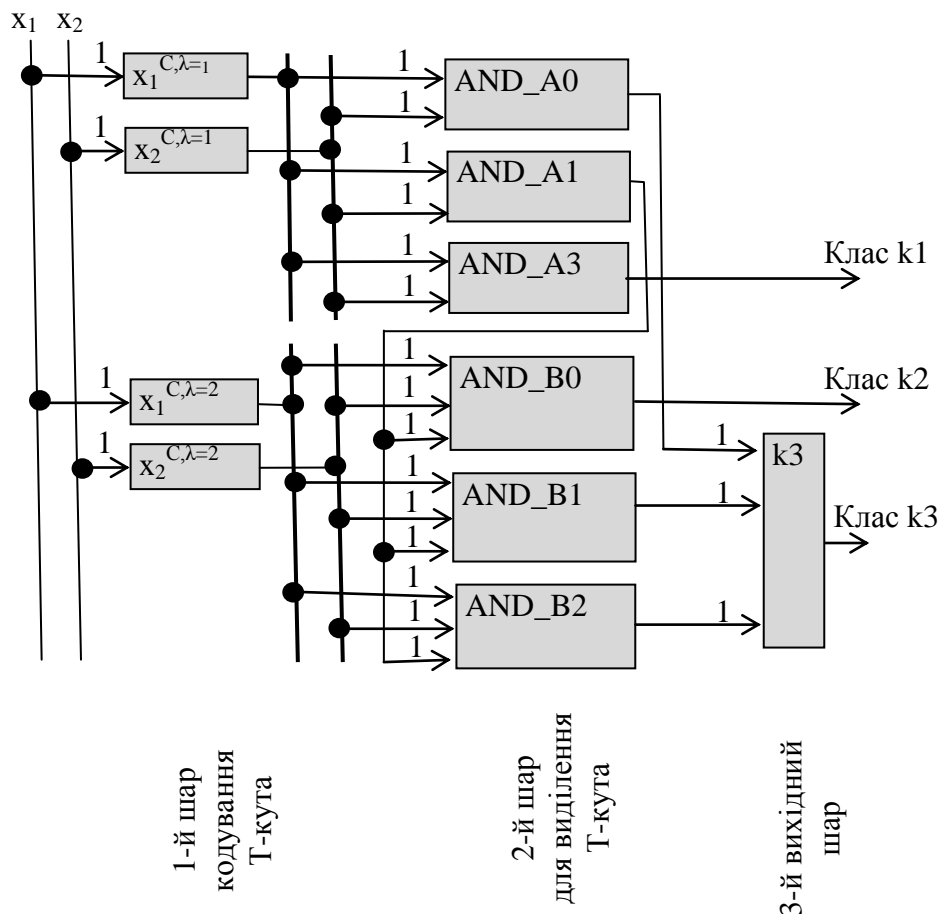


Рис. 11.5.7. Стохастична нейронна мережа

Стохастична нейронна мережа рис. 11.5.7 має три шари:

– 1-й вхідний шар кодування Т-кута призначений для отримання порядкового номера Т-кута, у якому знаходиться вхідний вектор невідомого класу $X = (x_1, x_2, \dots, x_j, \dots, x_n)$ з цифровими змінними x_j .

– 2-й шар складається з нейронів «Т_AND_0», кожний з яких має призначений йому екпертом для розпізнавання індивідуальний двоїстий номер Т-кута або «типа 0» або «типа 1». Якщо вектор входу $X^{C,\lambda}$ нейрона «Т_AND_0» співпадає з призначеним йому екпертом індивідуальним двоїстим номером Т-кута, то на виході цього нейрону «Т_AND_0» 2-го шару сигнал 0 замінюється на сигнал 1;

– 3-й вихідний шар складається з нейрона – суматора, на входи якого підключають виходи нейронів «Т_AND_0» 2-го шару, призначених для розпізнавання Т-кутів «типа 0» з навчальними векторами однакового класу. 3-й вихідний шар відсутній у нейронній мережі у випадку, коли у 2-му шарі відсутні два чи більше нейронів «Т_AND_0», призначених для розпізнавання Т-кутів «типа 0» з навчальними векторами однакового класу.

Для першого ієрархічного рівня $\lambda = 1$ експерт вводить у табл. 11.5.1 всі навчальні вектори X^{e_k} згідно системі осей координат А з початком координат ($x_1 = 0$; $x_2 = 0$). В табл. 11.5.1 експерт вказує номер e кожного вектора X^{e_k} , типи кутів (0 або 1), клас вектора (або іншу розділювальну характеристику), координати вектора з їх переведенням у двоїстий d та десятичний g порядкові номери їх Т-кутів. Початок координат першого ієрархічного рівня $\lambda = 1$ приймається ($x_1^{\lambda=1} = 0$; $x_2^{\lambda=1} = 0$), бо навчальні вектори вибірки векторів $X^{e_k} = (x_1^{e_k}, x_2^{e_k}, \dots, x_j^{e_k}, \dots, x_n^{e_k})^T$, $e_k = 1, 2, \dots, E_k$, вміщують змінні $x_j^{e_k}$ з різними знаками при однакових нижніх індексах j .

Таблиця 11.5.1.

Перший ієрархічний рівень $\lambda = 1$ з точкою початку координат ($x_1^{\lambda=1} = 0$; $x_2^{\lambda=1} = 0$). Порядкові номери Т-кутів стохастичних векторів X^{e_k} початкової навчальної вибірки

Дані навчальних векторів X^{e_k}				Порядкові номери Т-кутів векторів X^{e_k}	
Номер e	Тип векторів Т-кутів	Клас k	Координати $x_1^{e_k, \lambda=1}$ $x_2^{e_k, \lambda=1}$	Двоїстий номер Т-кута $d = (x_1^C; x_2^C)$	Десятичний номер Т-кута g
1	0	1	2;20	1;1	3
2	0	1	3;20	1;1	3
3	0	1	2;15	1;1	3
4	0	1	3;15	1;1	3
5	0	1	3;10	1;1	3
6	1	2	-3;2,5	0;1	1
7	1	3	-0,8;2,5	0;1	1
8	1	3	-1,2;5,6	0;1	1

9	1	3	-3;15	0;1	1
10	0	3	-3;-5	0;0	0
11	0	3	-2;-5	0;0	0
12	0	3	-1;-5	0;0	0
13	0	3	-3;-10	0;0	0
14	0	3	-2;-10	0;0	0

В табл. 11.5.1 для вектора X^{e_k} , керуючись даними колонки класу, вказують значення «типу 0» або «типу 1» Т-кута відповідного порядкового номера. Нижче для скорочення будемо посилатись лише на десятинні номери Т-кутаг.

З даних табл. 11.5.1 та рис. 11.5.6 на основі системи координат, поміченої літерою А, для першого ієрархічного рівня $\lambda = 1$ впливає:

– В Т-куті $g = 0$ присутні навчальні вектори X^{e_k} одного класу $k = 3$ при $e = (10...14)$. Тому Т-кут $g = 0$ відноситься до «типу 0» (цей кут не має розділювальних обмежень стосовно належних йому векторів класу $k = 3$). Внаслідок цього вважаємо, що якщо у 1-му шарі нейронної мережі для першого ієрархічного рівня $\lambda = 1$ будь-який вхідний вектор X буде віднесений до Т-кута $g = 0$, то цей вектор X має клас $k = 3$, а відповідний цьому Т-куту $g = 0$ нейрон AND_A0 типу «Т_AND_0» у 2-му шарі нейронної мережі (рис. 11.5.7) має призначений йому для запам'ятовування індивідуальний номер Т-кута $g = 0$ (у двоїстому численні $d=00$). Кінцева частка позначення нейрона AND_A0 у вигляді «A0» означає: літера «А» – позначення осей координат на рис. 11.5.6; цифра «0» – порядковий номер Т-кута $g = 0$ у вказаній системі координат А на рис. 11.5.6. Нейрон AND_A0 повинен мати вихідний сигнал розпізнавання класу $k = 3$. Але у даному випадку Т-кут $g = 0$ охоплює лише частку навчальних векторів класу $k = 3$. Тому вихід нейрона AND_A0 з'єднують з входом нейрона-суматора $k3$ у 4-му вихідному шарі нейронів (рис. 11.5.7).

– В Т-куті $g = 1$ присутні вектори X^{e_k} двох класів ($k=2$ та $k = 3$) при $e = (6, 7, 8, 9)$, і тому Т-кут відноситься до «типу 1» (цей кут має розділювальні обмеження, бо належні йому вектори X^{e_k} відносяться до різних класів). Тому відповідний нейрон AND_A1 типу «Т_AND_0»(рис. 11.5.7), який експерт призначає для запам'ятовування індивідуального номера $g = 1$ Т-кута (у двоїстому численні $d=01$) не може мати вихідного сигналу. Належні Т-куту $g = 1$ «типу 1» навчальні вектори мають різні класи, і тому ця скорочена сукупність навчальних векторів розглядається як окрема скорочена стохастична навчальна вибірка векторів другого ієрархічного рівня $\lambda=2$ з номерами навчальних векторів $e = (6)$ класу « $k = 2$ » та $e = (7, 8, 9)$

класу « $k = 3$ ». Відповідний нейрон AND_A1 Т-кута $g = 1$ «типу 1» у 2-му шарі нейронної мережі розпізнає наданий йому індивідуальний номер $g = 1$ і видає на виході сигнал 1 про виділення з розглянутої початкової навчальної вибірки з векторами $e = 1, 2, \dots, 14$ нової скороченої вибірки нижчої ієрархії $\lambda=2$ у вигляді сукупності відповідних навчальних векторів $e = 6, 7, 8, 9$, які попали у даний Т-кут «типу 1» і які *потребують подальшого дроблення* з використанням алгоритму, аналогічного описаному вище.

– В Т-куті $g = 2$ навчальні вектори X^{e_k} відсутні, і його не аналізують.

– В Т-куті $g = 3$ присутні навчальні вектори X^{e_k} одного класу $k = 1$ з порядковими номерами $e = (1, 2, 3, 4, 5)$. Тому Т-кут $g = 3$ відносять до «типу 0» (цей кут не має розділювальних обмежень стосовно належних йому векторів класу $k = 1$). Внаслідок цього вважають, що якщо у 1-му шарі нейронної мережі для першого ієрархічного рівня $\lambda = 1$ будь-який вхідний стохастичний вектор невідомого класу $X = (x_1, x_2, \dots, x_j, \dots, x_n)$ відносять до Т-кута $g = 3$, то цей вектор X має клас $k = 1$. Тому відповідному нейрону AND_A3 у 2-му шарі нейронної мережі (рис. 11.5.7) призначають для запам'ятовування індивідуальний номер Т-кута $g = 3$ (у двоїстому численні $\text{id}=11$). Кінцева частка позначення нейрона AND_A3 у вигляді «A3» означає: літера «A» – позначення осей координат на рис. 11.5.6; цифра «3» – порядковий номер Т-кута $g = 3$ у вказаній системі координат А на рис. 11.5.6. Нейрону AND_A3 надають вихідний сигнал розпізнавання класу $k = 1$ (тому що згідно табл. 11.5.1 відповідний йому Т-кут $g = 3$ (рис. 11.5.7) охоплює всі навчальні вектори класу $k = 1$ за порядковими номерами $e = (1, 2, 3, 4, 5)$ початкової навчальної вибірки).

Таким чином, на першому ієрархічному рівні $\lambda = 1$ отриманий лише один Т-кут $g = 1$ «типу 1» з навчальними векторами $e = (6, 7, 8, 9)$, сукупність яких експерт розглядає як ієрархічно нижчу скорочену навчальну вибірку, яку потрібно далі розділяти на наступні менші частки з метою отримання ієрархічно нижчих Т-кутів «типу 0».

Для нижчого ієрархічного рівня $\lambda = 2$ точку центра нової системи координат $(x_1^{\lambda=2}, x_2^{\lambda=2})$ для Т-кута $g = 1$ «типу 1» вищого ієрархічного рівня $\lambda = 1$ експерт може визначати по-різному: може довільно задати координати всередині Т-кута $g = 1$ «типу 1»; може розрахувати координати, як кінцеву точку центроїда векторів $e = (6, 7, 8, 9)$, які належать цьому куту; може визначити координати, як деяку «точку центра тяжіння» площі («гіпероб'єму») Т-кута з осями координат $(-x_1; +x_2)$.

У даному випадку вважаємо, що для Т-кута $g = 1$ «типу 1» ієрархічного рівня $\lambda = 1$ на рис. 11.5.6 експерт визначив центр нової системи координат $(x_1^{\lambda=2} = -1,5; x_2^{\lambda=2} = +10)$ з осями, позначеними літерою В. Далі експерт перераховує у всіх навчальних векторів $e = (6, 7, 8, 9)$ значення їх нових ко-

ординат за формулою $x_j^{k,e,\lambda=2,H} = (x_j^{k,e,\lambda=1} - x_j^{\lambda=2})$ і вводить їх у табл. 11.5.2 з виконанням у ній розрахунків, аналогічних табл. 11.5.1.

Таблиця 11.5.2.

**Другий ієрархічний рівень $\lambda = 2$ з точкою початку координат
($x_1^{\lambda=2} = -1,5$; $x_2^{\lambda=2} = +10$)**

Дані навчальних векторів X^{e_k}				Порядкові номери Т-кутів векторів X^{e_k}	
Номер e	Тип векторів Т-кутів	Клас k	Координати $x_1^{e_k, \lambda=2}$ $x_2^{e_k, \lambda=2}$	Двоїстий номер Т-кута $d = (x_1^C; x_2^C)$	Десятинний номер Т-кута g
6	0	2	-1,5;-7,5	0;0	0
7	0	3	+0,7;-7,5	1;0	2
8	0	3	+0,3;-4,4	1;0	2
9	0	3	-1,5;+5	0;1	1

В табл.11.5.2 всі вектори X^{e_k} розміщуються в Т-кутах «типу 0».

З даних табл. 11.5.2 та рис. 11.5.6 для другого ієрархічного рівня $\lambda = 2$ з системою координат, поміченою літерою В, впливає:

– В Т-куті $g = 0$ присутній лише один навчальний вектор $e = 6$ класу $k = 2$. Тому Т-кут $g = 0$ відноситься до «типу 0». Відповідний цьому Т-куту $g = 0$ нейрон AND_V0 типу «Т_AND_0» у 2-му шарі нейронної мережі (рис. 11.5.7) має призначений йому для запам'ятовування індивідуальний номер Т-кута $g = 0$ (у двоїстому численні $d = 00$). У даному випадку Т-кут $g = 0$ охоплює всі навчальні векторикласу $i = 2$. Тому вихід нейрона AND_V0 при наявності дозвольного сигналу з нейрона AND_A1 є виходом НМ по розпізнаванню векторів класу $k = 2$ (рис. 11.5.7).

– В Т-куті $g = 1$ теж присутній лише один навчальний вектор $e = 9$ класу $k = 3$. Тому Т-кут $g = 1$ відноситься до «типу 0». Відповідний цьому Т-куту $g = 1$ нейрон AND_V1 у 2-му шарі нейронної мережі (рис. 11.5.7) має призначений йому для запам'ятовування індивідуальний номер Т-кута $g = 1$ (у двоїстому численні $d = 01$). Але у даному випадку Т-кут $g = 1$ охоплює лише частку навчальних векторів класу $k = 3$. Тому при наявності дозвольного сигналу з нейрона AND_A1 вихід нейрона AND_V1 з'єднують з входом нейрона-суматора $k3$ у 3-му вихідному шарі (рис. 11.5.7).

– В Т-куті $g = 2$ присутні два вектори $e = (7, 8)$ одного класу $k = 3$, і тому Т-кут відноситься до «типу 0». Експерт призначає нейрону AND_V2 (рис. 11.5.7) для запам'ятовування індивідуальний номер Т-кута $g = 2$ (у двоїстому численні $d = 10$). При наявності дозвольного сигналу з нейрона AND_A1 вихід нейрона AND_V1 з'єднують з входом нейрона-суматора $k3$ у 3-му вихідному шарі (рис. 11.5.7).

Таким чином, будовання НМ продовжується до охоплення всіх векторів X^{e_k} вищого ієрархічного рівня $\lambda = 1$ всіма вихідними сигналами

T-кутів «типу 0» з 2-го та 3-го шарів, по яких визначають клас вхідного вектора X.

Отримана НМ рис. 11.5.7 не потребує навчання.

Застосування отриманих даних аналізу для визначення класів може використовуватись в 3-х напрямках:

1. **Класифікація векторів X по T-кутах** згідно НМ рис. 11.5.7.

2. **Класифікація векторів X по кластерах з використанням центроїдів кластерів.** Згідно з описаним вище алгоритмом, з кожним вихідним нейроном 2-го шару рис. 11.5.7, які охоплюють T-кути «типу 0», пов'язана визначена сукупність навчальних векторів, наведена в табл. 11.5.3, тобто пов'язаний відповідний об'єктний кластер. В табл. 11.5.3 наведений взаємний зв'язок отриманих вище груп навчальних векторів «типу 0» з позначеннями відповідних нейронів AND_A0, AND_A3, AND_B0, AND_B1, AND_B2. Клас вхідного вектора X визначається по класу відповідного центроїда кластера, до якого вектор X має найбільшу близькість.

Таблиця 11.5.3.

Взаємний зв'язок отриманих на першому етапі навчання груп навчальних векторів (кластерів) з позначеннями відповідних нейронів-центроїдів

Дані першого етапу навчання	
Позначення нейрона на рис. 11.5.7	Дані кластера
AND_A0	Вектори e = 10...14; клас k=3
AND_A3	Вектори e = 1...5; клас k=1
AND_B0	Вектор e = 6; клас k=2
AND_B1	Вектор e = 9; клас k=3
AND_B2	Вектори e = 7, 8; клас k=3

Тому визначають всі T-кути «типу 0», які охоплюють всі вектори навчальної вибірки, і дають їм порядкові номери $\delta = 1, 2, \dots, q$. Всі T-кути «типу 0» замінюють відповідними центроїдами кластерів.

1-й шар НМ складають з n нейронів з одним входом та одними виходом, у якому нормалізують і розмножують змінні вхідного стохастичного вектора невідомого класу $X = (x_1, x_2, \dots, x_j, \dots, x_n)$.

2-й шар складають з q нейронів з порядковими номерами $\delta = 1, 2, \dots, q$, кожному з яких надають один вихід та n входів з ваговими коефіцієнтами, рівними координатам відповідних центроїдів кластерів. На виході кожного з q нейронів 2-го шару отримують скалярний добуток M^δ вхідного вектора X та відповідного вектора центроїда, який має вигляд вагових коефіцієнтів входів.

3-й шар складають з q нейронів, у якому за принципом «переможець отримує все» по найбільшому значенню виходу серед q нейронів 2-го шару призначають вихідний сигнал 1 для кластера-переможця при нульових сигналах виходу для всіх інших кластерів.

4-й шар додають, якщо у 3-му шарі існують кластери однакового класу і потрібно класифікувати класи. У цьому випадку 4-й шар вміщує нейрони-суматори, які об'єднують вихідні сигнали кластерів однакового класу у 3-му шарі.

3. **Класифікація векторів X по кластерах з використанням байєсового класифікатора.** У цьому випадку використовують НМ по п. 2, але кожний з q нейронів 2-го шару замість скалярного добутку M^δ видає на виході значення байєсового класифікатора

$$p^k c^k \Gamma M^\delta,$$

де p^k – апіорна ймовірність появи k -го класу даного кластеру, яку або визначають по опублікованим даним, або приймають рівною відношенню кількості векторів k -го класу в початковій навчальній вибірці до загальної кількості навчальних векторів E в початковій вибірці;

c^k – витрати внаслідок помилкового заперечення наявності відповідного k -го класу даного кластеру при його реальній присутності;

f^δ – оцінка частотної характеристики даного δ -го кластера T -кута «типу 0», яку приймають рівною відношенню кількості векторів k -го класу в δ -му кластері до загальної кількості векторів k -го класу в початковій вибірці.

У всьому іншому робота нейронної мережі з використанням кластерів не змінюється по відношенню до п. 2.

Завдання. Задатись стохастичною вибіркою з 15-ти навчальних векторів 3-х класів, розміщених в 4-х квадрантах осей ознак $x_1 = -0,5N \dots + N$; $x_2 = -N \dots + N$, де N – порядковий номер студента у групі.

– Побудувати НМ для трьох ієрархічних рівнів при класифікації векторів X невідомого класу по T -кутах.

– Побудувати НМ на основі попереднього аналізу при класифікації векторів X невідомого класу по центроїдах кластерів.

– Побудувати НМ на основі попереднього аналізу при класифікації векторів X невідомого класу з використанням байєсового класифікатора.

– Розробити програму на будь-якій мові для НМ з байєсовим класифікатором. Привести спочатку результати класифікації по заданій стохастичній вибірці з 15-ти навчальних векторів 3-х класів, а потім отримати дані по 100 векторам X невідомого класу, координати для якого визначити за допомогою генератора випадкових чисел.

11.5.4. Машина Больцмана

Машина Больцмана отримала назву на честь австрійського фізика Л. Больцмана [126].

Машина Больцмана може розглядатись як стохастичний варіант мережі Хопфілда: стосовно вхідного вектора X вона призначена для виділення найближчого асоціативного еталонного вектора; вона підкоряється поняттю «енергії» (як і мережа Хопфілда); кожний нейрон не має зв'язку з самим собою; вагова матриця НМ є симетричною (як і мережа Хопфілда). Машина Больцмана виправляє недолік мережі Хопфілда, яка має тенденцію до стабілізації в локальному мінімумі замість глобального мінімуму. Для цього при навчанні машину Больцмана змушують частіше проходити через глобальні мінімуми.

Пошук мінімуму функції має схожість з процесом «відпалу» металу і, крім того, з фізичного процесу «відпалу» металу використовуються деякі формули, терміни і пояснення:

1. Для відпалу метал спочатку нагрівається до температури плавлення T , при якій атоми рухаються хаотично з великою амплітудою. Це є аналогією великих початкових кроків пошуку глобального оптимуму (розшукується мінімум функції).

2. Якщо метал з самого початку одразу раптом охолодити водою, то атоми застигнуть у випадкових положеннях (ми нехтуємо уточненнями місця знаходження мінімуму).

3. Якщо метал охолоджувати поступово, то атоми намагаються вишикуватись у регулярну форму (відбувається поступове зменшення кроків пошуку і уточнення місця знаходження мінімуму). Тому основою всього процесу «відпалу» металу є керування графіком зниження температури T . Чим повільніше ми будемо знижувати температуру, тим напевніше будемо наближуватись до глобального мінімуму (але тим більше будемо втрачати машинного часу).

Машина Больцмана на високій температурі більш схожа з випадковою моделлю, а на низьких температурах вона веде себе як детермінована модель.

Енергія мережі. У своїй роботі [161] Хопфілд довів, що *мережа Хопфілда* повинна сходиться до сталого стану значень елементів векторів при мінімізації «енергії мережі», яка може розраховуватись за формулою [35]

$$E = -\frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n s_j s_i w_{ji}, \quad (11.5.3)$$

де $i=1, 2, \dots, n$ – індекс нейрону, від якого надається зв'язок; $j=1, 2, \dots, n$ – індекс нейрону, до якого надходить зв'язок; w_{ji} – ваговий коефіцієнт зв'язку між j -м та i -м нейронами; s_j, s_i – стани елементів системи (у даному випадку – значення елементів вектора, які розраховуються за активаційною функцією сігмоїди).

Якщо елемент s_j змінює свій стан на Δs_j , то зміна енергії мережі дорівнює

$$\Delta E_j = -\Delta s_j \sum_{i=1}^n s_i w_{ji}. \quad (11.5.4)$$

Хопфілд довів, що Δs_j та $\sum s_i w_{ji}$ в виразі (11.5.4) мають однакові знаки. Тому знак зміни енергії мережі ΔE_j завжди від'ємний, і енергія мережі Хопфілда зменшується.

Загальні кроки навчання.

У початку навчання всім вагам мережі привласнюються невеликі (0,01...0,03) випадкові значення. Обирається достатньо велика початкова температура T так, щоб для всіх змін енергії виконувалась нерівність $\exp(-\Delta E/T) \geq 0,9999$. В машині Больцмана спочатку обробляються нейрони зовнішнього шару, а потім – нейрони вихідного шару.

На введений в машину Больцмана вхідний вектор X з елементами ± 1 випадково подаються по черзі навчальні еталонні пари. Випадково обирається і *перемикається* один елемент двійкового вектора X (тобто біт 0 замінює значення на 1, чи навпаки). Якщо при цьому енергія $f(X)$ зменшується, то зміна приймається, інакше розраховується ймовірність, за якою приймається зміна.

Зменшується температура T зворотно-пропорційно логарифму часу. Якщо було виконане задане число M змін вектора X , для яких величина енергії $f(X)$ зменшилась, або якщо було виконано задану кількість N зменшень X з моменту останнього зменшення температури, то встановлюється нова знижена температура $T_{\text{НОВА}} = \alpha T_{\text{СТАРА}}$, де $\alpha = 0,8 \dots 0,9999$.

Якщо мінімальне значення енергії $f(X)$ не змінюється на протязі заданого числа ітерацій, то процес зупиняється. Інакше знову повторюють процес.

11.5.5. Нейрон Байєса для класифікації об'єктів n -вимірного простору

Вхідний вектор n -вимірного простору з стохастичними ознаками (змінними) $X = (x_1, x_2, \dots, x_j, \dots, x_n)$ при $x_j = \pm 1$, який увімкнений до відповідних n входів з ваговими коефіцієнтами 1, нейрон Байєса (рис. 11.5.10) відносить до деякої гіпотези вихідного вектора $H = (h_1, h_2, \dots, h_i, \dots, h_m)$.

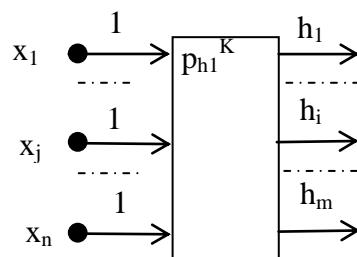


Рис. 11.5.10. Нейрон Байєса для віднесення об'єкта

$X = (x_1, x_2, \dots, x_j, \dots, x_n)$ до гіпотези $H = (h_1, h_2, \dots, h_i, \dots, h_m)$

Всі значення елементів вихідного вектора гіпотез $H = (h_1, h_2, \dots, h_i, \dots, h_m)$ дорівнюють нулю, за вилученням однієї гіпотези-переможниці, яка має значення $h_i = 1$. Обрання гіпотези-переможниці $h_i = 1$ серед інших гіпотез виконується по максимальному значенню Байєсового класифікатора, по якому розраховують ймовірність спостереження кожної i -ої гіпотези h_i [81]

$$P_i = \frac{p_i^0 \sum_{j=1}^n p(x_j | h_i)}{\sum_{i=1}^m \left[p_i^0 \sum_{j=1}^n p(x_j | h_i) \right]}, \quad (11.5.4)$$

де $i = 1, 2, \dots, m$ – порядковий номер гіпотези; $j = 1, 2, \dots, n$ – порядковий номер стохастичних змінних (ознак) об'єкта, по яким розпізнають, до якої гіпотези h_i відноситься об'єкт з ознаками вхідного вектора $X = (x_1, x_2, \dots,$

x_j, \dots, x_n); x_j – ознака (змінна) з порядковим номером j , яка може приймати значення ± 1 (наявність ознаки/її відсутність); h_i , $i = 1, 2, \dots, m$ – гіпотеза з порядковим номером i ; p_i^0 , $i = 1, 2, \dots, m$ – апріорна ймовірність гіпотези з порядковим номером i ; $p(x_j|h_i)$ – умовна ймовірність спостереження ознаки (змінної) x_j стосовно гіпотези h_i з порядковим номером $i = 1, 2, \dots, m$.

Для спрощення нейрону, складна формула (11.5.4) замінюється на її чисельник

$$p_i^K = p_i^0 \sum_{j=1}^n p(x_j | h_i), \quad (11.5.5)$$

тому що максимальне значення чисельника визначає найбільшу величину за формулою (11.5.4).

Розрахунок максимального значення чисельника згідно формули (11.5.5), а, отже, визначення гіпотези-переможниці $h_i = 1$, виконується в активіційній функції нейрона:

for $j:=1$ to n do if $x_j>0$ then $p(x_j|h_i):= p(+x_j|h_i)$ else $p(x_j|h_i):= p(-x_j|h_i)$;
 {формування елементів вектора P_i з елементів векторів ($P+i$ та $P-i$)}

for $i:=1$ to m do $p_i^K:=0$;

for $i:=1$ to m do for $j:=1$ to n do $p_i^K:= p_i^K + p(x_j|h_i)$; $p_i^K:= p_i^0 \cdot p_i^K$; {розрахунок p_i^K за формулою (11.5.5)}

$NM:= p_1^K$; {визначення гіпотези-переможниці нейрона Байєса h_i }

for $i:=2$ to m do if $p_i^K > NM$ then $NM:= p_i^K$;

for $i:=1$ to m do Begin $h_i:=0$; if $p_i^K = NM$ then $h_i:=1$; End;

де $p(+x_j|h_i)$, $i = 1, 2, \dots, m$, i -й елемент першого вектора бази даних нейрона Байєса $P+i = (p(+x_1|h_i); p(+x_2|h_i); \dots p(+x_j|h_i); \dots p(+x_n|h_i))$, який ураховує умовні ймовірності спостереження позитивних значень змінних ($+x_j$);

$p(-x_j|h_i)$, $i = 1, 2, \dots, m$, i -й елемент другого вектора бази даних нейрона Байєса, $P-i = (p(-x_1|h_i); p(-x_2|h_i); \dots p(-x_j|h_i); \dots p(-x_n|h_i))$ який ураховує умовні ймовірності спостереження негативних значень змінних ($-x_j$). По першому $P+i$ і другому $P-i$ векторам бази даних підсумок ймовірностей однієї ознаки складають повну групу: $p(+x_j|h_i) + p(-x_j|h_i) = 1$;

$p(x_j|h_i)$, $i = 1, 2, \dots, m$, i -й елемент вектора умовної ймовірності $P_i = (p(x_1|h_i); p(x_2|h_i); \dots p(x_j|h_i); \dots p(x_n|h_i))$, який призначений для використання в формулі (11.5.5) і визначається для вхідного вектора $X = (x_1, x_2, \dots, x_j, \dots, x_n)$ по двом векторам бази даних ($P+i$, $P-i$) за правилом: якщо деяка змінна вхідного вектора X дорівнює $x_j = +1$, то відповідна умовна ймовірність вектора P_i визначається по вектору $P+i$, а в іншому разі – по вектору $P-i$. В результаті використання векторів (X , $P+i$, $P-i$) з двох векторів бази даних ($P+i$, $P-i$) отримуємо вектор $P_i = (p(x_1|h_i); p(x_2|h_i); \dots p(x_j|h_i); \dots p(x_n|h_i))$, який використовується для розрахунку величини p_i^K згідно формули (11.5.5) з урахуванням апріорної ймовірності гіпотези p_i^0 ;

p_i^0 , $i = 1, 2, \dots, m$ – i -й елемент визначеного експертом вектора апріорних ймовірностей $P^0 = (p_1^0, p_2^0, \dots, p_i^0, \dots, p_m^0)$ для гіпотез h_i ;

$p_i^K, i = 1, 2, \dots, m$ – i -й елемент розрахованого за формулою (11.5.5) вектора чисельників, $Ph = (p_{h1}^K, p_{h2}^K, \dots, p_{hi}^K, \dots, p_{hm}^K)$;

$h_i, i = 1, 2, \dots, m$ – i -й елемент вектора значень виходів нейрона Байєса $H = (h_1, h_2, \dots, h_i, \dots, h_m)$;

НМ – максимальне значення елемента гіпотези – переможці серед елементів вектора $H = (h_1, h_2, \dots, h_i, \dots, h_m)$.

11.6. Апроксимація функцій нейронною мережею

11.6.1. Нейронна мережа з радіальними базовими функціями

Основні дані по нейронним мережам з радіальними базовими функціями. Математичною основою радіально-базисних нейронних мереж є метод потенціальних функцій, який розроблений М. А. Айзерманом, Е. М. Браверманом і Л. І. Розонером (1970) і дозволяє представити будь-яку функцію у вигляді суперпозиції вектора однакових зміщених між собою у просторі ознак потенціальних функцій [1].

Радіально-базисна нейронна мережа вперше розглянута в роботі [187].

За її допомогою можна як завгодно точно апроксимувати задану лінійну чи нелінійну функцію багатьох змінних [35; 188].

Відображення від вхідного шару до захованого шару є нелінійним, а відображення захованого шару до вихідного є лінійним [35].

В радіально-базисній нейронній мережі вибираються базисні $f_B(|x-c|)$, які є невід’ємними і зростаючими при зменшенні абсолютного значення модуля $|x-c|$. Прикладами потенційних функцій є функція Гауса f_1^B (використовується найчастіше), зворотна мультікватратична функція f_2^B та функція Коші f_3^B :

$$f_1^B(x) = \exp\left[-\frac{(x-c)^2}{2\sigma^2}\right], \quad f_2^B(x) = \left[\frac{(x-c)^2}{\sigma^2} + a^2\right]^{-\frac{1}{2}}, \quad f_3^B(x) = (1 + |x|). \quad (11.6.1)$$

де x – довільна змінна; a, c – константи, які зміщують базисну функцію у просторі; σ – константа, яка змінює «гостроту» амплітуди базисної функції.

Внесок в загальну оцінку значення вихідної функції (рис. 11.6.1), який вносить один нейрон з порядковим номером k_j вздовж однієї осі x_j може мати різні модифікації [103]; наприклад, при використанні функції Гауса цей внесок може дорівнювати

$$f_{kj}^B(x_j) \Rightarrow r_k \cdot \exp\left[-\lambda_k^2 [d_1^2(x_1 - c_{k1})^2 + d_2^2(x_2 - c_{k2})^2]\right],$$

де r_k – коефіцієнт, який змінює амплітуду базисної потенційної функції $f_{kj}^B(|x_j - c_{kj}|)$; $\lambda_k, d_1, d_2, c_{k1}, c_{k2}$ – константи.

Функція Гауса f_1^B (11.6.1) має вигляд «капелюха», наведеного жирною безперервною лінією на рис. 11.6.1. Малі величини σ відповідають «гост-

роверхим капелюхам» (значення σ обирає програміст у процесі дослідження). Для чотирьох рядом розміщених навчальних об'єктів однакового класу відповідні функції Гауса наведені штриховими лініями. Підсумок ординат всіх наведених функції буде показувати оцінку значення шуканої функції $f^B(X)$ (на рис. 11.6.1 підсумкова крива не наведена; іноді замість підсумку функцій Гауса використовується *одна функція Гауса* [35]).

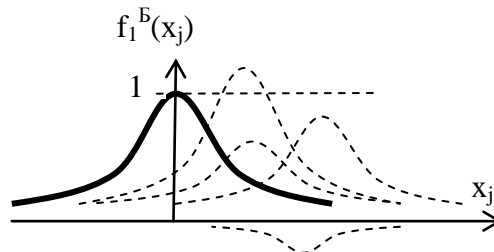


Рис. 11.6.1. Функція Гауса для однієї змінної x_j

Одну й ту ж проблему можна розв'язати як за допомогою радіально-базисної НМ так і за допомогою багатозарової НМ [35, с. 69].

Приклад 1. Нейронна мережа з радіальними базовими функціями, яка по вхідній змінній x дає на виході криву $f(x) = 2x^2$, наведена на рис. 11.6.2. Вона складається з трьох шарів:

1. **Вхідний 1-й сенсорний шар** має один нейрон N_1 , який призначений для розмноження вхідного значення змінної x .
2. **Захований 2-й асоціативний шар** складається з 10 нейронів $N_{2,1} \dots N_{2,10}$, у яких ваги входів дорівнюють 1, а вихідними активаційними функціями є *десять функцій Гауса* (в нейронах 2-го захованого шару), рівномірно зміщених по відношенню одна до одної вздовж осі x .

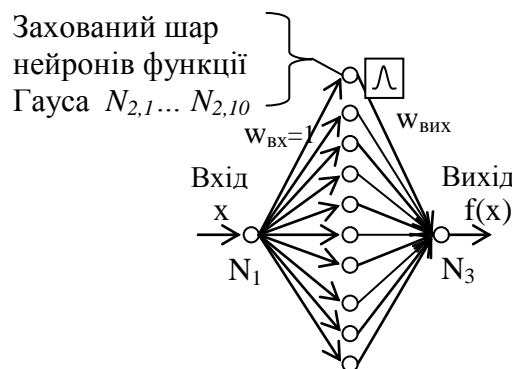


Рис. 11.6.2. Нейронна мережа з радіальними базовими функціями, навчена давати на виході криву $f(x)$

Вважаємо, що значення вхідної змінної у рівнянні $f(x) = 2x^2$ змінюються в межах $x = -0,7 \dots +0,7$. Розділимо цей відрізок $[-0,7 \dots +0,7]$ на 10 окремих проміжків – по одному на кожний нейрон 2-го шару. Припустимо, що в якості потенційної функції ми обрали функцію Гауса (11.6.1). Кожний з нейронів моделює свою потенційну функцію (11.6.1) з однаковою шириною $=0,5$, але з різними іншими параметрами та центрами в точках

$C = X\{c_1, c_2, \dots, \dots, c_K\} = \{-0.6, -0.4, -0.3, -0.1, 0.0, 0.1, 0.3, 0.4, 0.6, 0.7\}$,
де c_k – центр функції Гауса, $k=1, 2, \dots, K$ – порядковий номер функції Гауса (у даному випадку $K=10$).

Цим самим ми приблизно рівномірно розподілили центри c_k десяти потенційних функцій Гауса вздовж осі x на проміжку $x = -0,7 \dots +0,7$. Звичайно потенційна функція має однакову загальну форму для всіх нейронів захованого 2-го шару, а їх кількість та параметри (c_k, σ) визначаються експертом. Значення можуть також задаватись випадковим чином.

Перший шар ваг $w_{вх}$ від нейрону входу до нейронів захованого 2-го шару визначається за навчальною вибіркою і вважається рівним вектору центрів c_k функції Гауса [103, с. 231; 38, с. 71, 74].

кожний нейрон захованого 2-го шару має привласнену йому функцію Гауса; початкові значення ваг $w_{вих}$ для входів одного вихідного нейрона мають малі значення і визначається випадковим чином (як і їх знаки), а потім вони уточнюються за дельта-правилом Відроу-Хоффа

$$\Delta w = \eta \delta \cdot u_i,$$

де $\delta = t - f(x)$ – похибка; $t = 2x^2$ – еталонне значення функції; $f(x)$ – розрахований вихід мережі; η – коефіцієнт навчання; u_i – розрахований вихід нейронів захованого 2-го шару.

Вихідний нейрон підсумовує виходи нейронів захованого шару. Хоча теоретично у даній мережі часто достатньо мати один прихований шар, але іноді кілька захованих шарів дають ліпші результати [35].

В [35] для навчання подібної мережі використали 21 навчальну пару для кожної з яких вхідне значення обиралось у проміжку $x = -0,7 \dots +0,7$, а цільовий вихід розраховувався за формулою $t = 2x^2$. Коефіцієнт навчання дорівнював $\eta = 0,1$.

Приклад 2. Припустимо, що нам потрібно у залежності від x (часу) прогнозувати ціну акцій на наступний день торгів за інформацією про їхній курс за попередні 10 днів. Ця інформація змінюється у часі і тому використання «точного» (незмінного, фіксованого) алгоритму гри на біржі веде до банкрутства.

Найбільш придатні для розв'язання цієї проблеми НМ з їх здатністю до навчання. Для цього використовуємо НМ з архітектурою 10:15:1, яка повинна вчитися на курсах вартості акцій за попередні 10 днів та переналагоджуватись з плином часу. На вхід цієї НМ подається вектор розмірності 10, кожний елемент якого дорівнює відомому курсу акцій за один відповідний робочий день біржі з терміну у 10 днів перед 11-м прогнозованим днем. При 5-денному робочому тижні біржі вхідний вектор на 10 нейронів охоплює своєю інформацією два попередніх тижня перед 11-м днем прогнозу. Захований шар з 15-ти нейронів обробляє інформацію, а один нейрон виходу дає прогноз про курс акцій на 11-й робочий день біржі. Таким чином, можна скласти 11-денний графік курсу акцій, для якого 11-й день є прогнозом. Кількість епох навчання НМ – 9.

Використовується і інша архітектура НМ (15:10:1), яка надає прогноз на 16-й робочий день за 15-денною інформацією про курс паперів на попередніх 3-х тижнях. Обробляє цю інформацію 10 нейронів захovanого шару. Для виконання графіку курсу акцій ми маємо значення ціни акцій у часі за попередні 15 днів (15 входів НМ) плюс 16-й прогнозний день на виході НМ.

Примітка. Приклад 2 наведений для пояснення принципу розпізнавання образу. У дійсності курс акцій визначається з урахуванням значно більшої кількості критеріїв і за значно складнішою сукупністю алгоритмів, ніж описується. І треба пам'ятати, що для гри необхідно мати значні кошти (маржу) на неминучі програші. Звичайно програють 95 % біржових гравців, які не дуже добре обізнані з фінансовою ситуацією та політичними новинами. Крім того, великі гравці (банки) можуть штучно створювати вигідні для себе ситуації, які приблизно і охоплюють 5 % успішних біржових операцій.

Нейронні мережі з радіальними базовими функціями у багатовимірному просторі.

Для апроксимації багатьох вихідних функцій $y_1 - y_E$, які залежать від багатьох змінних $x_1 - x_n$, використовується НМ рис. 11.6.3 [103].

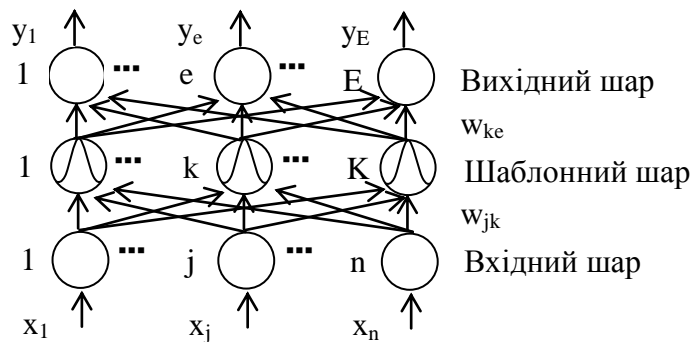


Рис. 11.6.3. Структурна схема радіально-базисної НМ при багатьох змінних

Інформація з вхідного сенсорного шару у вигляді вектора змінних $X=(x_1, x_2, \dots, x_j, \dots, x_n)$ передається в наступний прихований (шаблонний) шар. Кожен нейрон шаблонного шару обчислює функцію [103, с. 230–231]

$$f_k(X) = f((X - C_j)^T R^{-1}(X - C_j)),$$

де X – вектор вхідних сигналів порядку n ; C_j – вектор центрів порядку n ; W – вагова матриця порядку $(n \cdot n)$.

В шаблонному шарі визначається відстань між вхідним вектором X та центром C_j . Вектор центрів C_j визначається по навчальній вибірці і зберігається в просторі ваг від вхідного шару до шару шаблонів.

Зворотна коваріаційна матриця R^{-1} має вигляд

$$R^{-1} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1k} & \dots & r_{1K} \\ r_{21} & r_{22} & \dots & r_{2k} & \dots & r_{2K} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ r_{k1} & r_{k2} & \dots & r_{kk} & \dots & r_{kK} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ r_{K1} & r_{K2} & \dots & r_{Kk} & \dots & r_{KK} \end{bmatrix},$$

де $r_{jk} = \sigma_{jk}^{-2} = \sigma_{kj}^{-2}$ – деякі керовані параметри при $j, k = 1, 2, \dots, K$.

Часто матриця R^{-1} вибирається діагональною (коли для $j \neq k$ значення $r_{jk} = 0$) і приймають, що $r_{jj} = \sigma_{jj}^{-2} = \sigma^{-2} = \text{const}$. У цьому випадку для функції Гауса σ є стандартним відхиленням. Значення σ_{jj} визначається експертом у процесі налаштування НМ [103, с. 230–231].

Вихідна функція НМ рис. 11.6.4 дорівнює

$$y_e = \sum_{k=1}^K (f_k(X) w_{ke}).$$

Отримані сигнали y_e іноді нормалізують за формулою

$$y_e^* = \frac{y_e}{\sum_{k=1}^K f_k(X)}.$$

Завдання

Задатись довільною нелінійною функцією $t = f(x_1)$ при значеннях $x_1 = -N \dots +N$. Використати шар шаблонів на 10 нейронів. На будь-якій мові написати програму навчання НМ. Операцію нормалізації не виконувати.

11.6.2. Мережа регресійного аналізу

Мережа регресійного аналізу призначена для отримання на виході з НМ *оцінки $F^0(X)$ значення складної функції $F(X)$* , яка задана, наприклад, у формі таблиці, отриманої в результаті експериментів, складних розрахунків чи спостережень. Можна отримати математичну модель такої функції на основі використання метода найменших квадратів (МНК), але, на відміну від МНК, мережа регресійного аналізу дозволяє більш адекватно визначити рельєф поверхні функції [194].

Точки мережі регресійного аналізу *не поділяються на класи* і тому її архітектура дещо відрізняється від архітектури стохастичної нейронної мережі.

Навчальна множина має координати відповідних навчальних точок X та числові значення функції $F(X)$ цих точках (рис. 11.6.4, б) і записується у шар взірців (рис. 11.6.4, б) як один клас порівняно із стохастичною мережею. Кожна точка навчальної множини записується в одному нейроні *ша-*

ру аналізу взірців (A) у вигляді добутку цієї функції на функцію Гауса $[F(X)g(X)]$. Отриманий добуток $[F(X)g(X)]$ дорівнює (ϵ оцінкою) функції $F(X)$ у заданій точці симетрії функції Гауса, в якій $g(X)=1$, і поступово зменшує своє значення у міру відхилення значень координат від цієї точки симетрії.

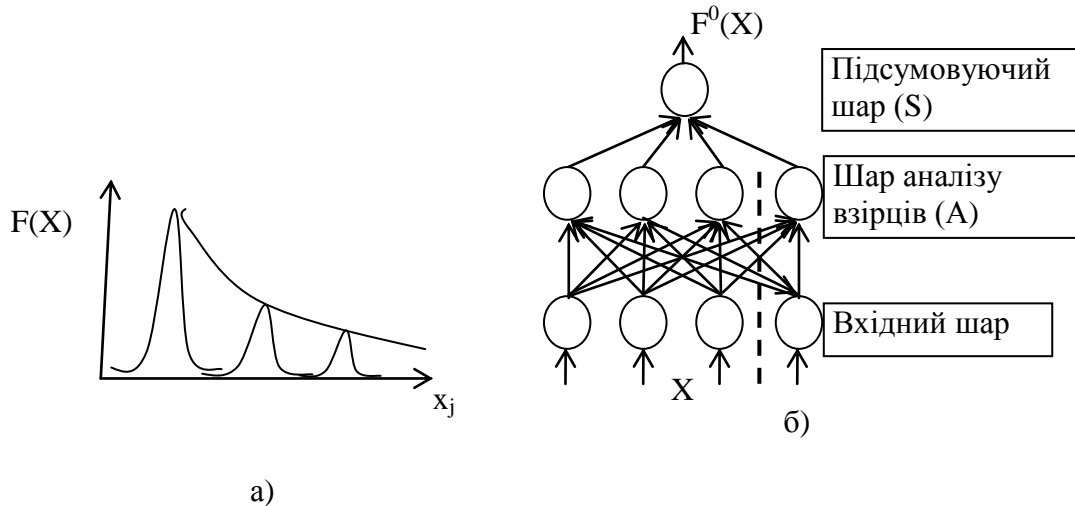


Рис. 11.6.4. Архітектура мережі регресійного аналізу:
а) принцип утворення оцінки функції; б) структура мережі

У цьому випадку розділу навчальних точок на класи звичайно не існує, і тому підсумовуючий шар (S) складається з одного підсумовуючого нейрону, на виході якого отримуємо *оцінку* вихідного значення регресійної функції $F(X)$.

11.7. Динамічна рекурсивна нейронна мережа

Рекурсивні НМ відрізняються тим, що вони мають зворотні зв'язки від наступних до попередніх шарів. Припускається *будь-який варіант зв'язку*: зв'язок виходу одного нейрону з власними входами; зв'язки між нейронами одного чи кількох шарів [128; 148; 165; 185; 186].

Рекурсивні нейромережі є динамічними системами, що оперують з послідовностями вхідних даних і перетворюють їх на послідовності вихідних векторів (реакцій).

Поведінка рекурентних нейромереж віддзеркалює набуті при навчанні стереотипи, що робить їх близькими до *цілеспрямованих адаптивних динамічних систем*, запрограмованих на досягнення заздалегідь визначених цілей. Вони здатні ефективно керувати об'єктом, який розглядається як «чорний ящик», в умовах невизначеності, зокрема, вирішувати задачі адаптивного керування поведінкою складних систем у нестационарному оточенні, приймати оперативні рішення в системах ситуаційного управління.

Програмування поведінки рекурентних нейромереж здійснюється шляхом навчання на прикладах, що не потребує формального визначення цілей [5; 114].

Нейрони різних шарів можуть бути однаковими або відрізнятися активаційними функціями, характером навчання та призначення нейронів. Із-за затриманих зворотних зв'язків **рекурентні мережі є динамічними системами**: їх поведінка має зовнішню складову, яка відповідає спостереженим даним входу та виходу, та приховану складову, яка характеризує внутрішній стан нейромережі. Ці дві складові поведінки описують двома рівняннями, перше з яких є рівнянням вхід-вихід, а друге – рівнянням стану [90]. Стан рекурентної НМ визначається сукупністю поточних та затриманих значень реакцій нейронів, які наразі діють на входи нейронів.

Навчання рекурентних нейромереж здійснюється на прикладах.

На рис. 11.7.1, а показана рекурсивна НМ із зворотним поширенням сигналів.

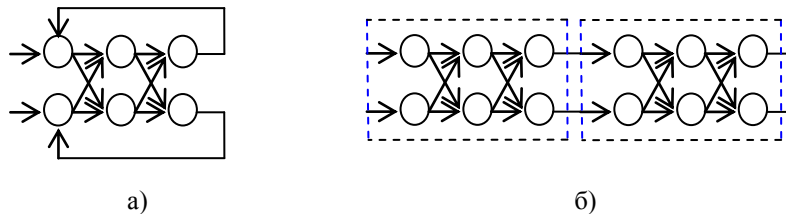


Рис. 11.7.1. Рекурсивна НМ зі зворотним поширенням сигналів

Особливість рекурсивної НМ полягає у тому, що вона обробляє деякі послідовності у вигляді ланцюга взірців, що мають відношення до одного об'єкта [35].

Вважається, що будь-яка рекурсивна НМ має НМ прямого поширення сигналів аналогічної поведінки, але без зворотних зв'язків. Рис. 11.7.1, б демонструє два кроки **роботи у часі НМ** рис. 11.7.1, а.

Тобто, для НМ рис. 11.7.1, а ми повинні зробити у часі деяку кількість повторних розрахунків до отримання кінцевого стану на виході: розрахувати вихід при першому проході НМ; вимкнути входи і подати на них розраховані виходи; повторити розрахунки до отримання на виході сталого результату (але він отримується не завжди). У цьому сенсі рекурсивна НМ нагадує НМ Хопфілда, яка є окремим випадком динамічних НМ.

Існує також мережа Елмана (один з видів рекурентної мережі), яка має зворотні зв'язки у багатошаровій НМ, але йдуть вони не з виходу, а з внутрішніх шарів. Це урахує передісторію процесу, сприяє запам'ятовуванню послідовностей і дозволяє використати інформацію для виробки стратегії керування рухомими об'єктами. На основі подвійної мережі Елмана виду $2N-N-2N$, яку використовують для стиснення і шифрування інформації, будується нейронна мережа класу RAAM (Рекурсивна авто-асоціативна пам'ять). Звичайно мережа має кількість бітів у шарах

20–10–20; вхідні і вихідні 20 бітів ділять на групи по 10 бітів: перші 10 бітів зветься «лівими», другі 10 бітів – «правими». У початку навчання на «ліву матрицю» входу подається нульовий вектор бітів «0000000000», а на «праву матрицю» входу – код символу або речення (наприклад, 0010000000= «А»). Те ж саме подається на вихідну матрицю. *НМ навчається методом зворотного поширення похибки*. Потім 10 бітів з захованого шару передаються на «ліву» вхідну матрицю, а на праву подається черговий символ. У процесі такого рекурсивного навчання інформація стискається і шифрується.

У загальному випадку, наявність зворотних зв'язків може привести до наступних кінцевих станів рекурсивних НМ [103]:

1. Досягнення стійкого стану.
2. Осцилювання (періодичне повторення вихідних сигналів). Нейрофізіологі використовують осциляторні НМ (повнозв'язані, з локальними зв'язками, із затримками у зв'язках) для вивчення впливу коливань у *природних НМ* на розумову діяльність людини.
3. Хаотична зміна станів.

За архітектурою рекурентні нейромережі можна поділити на три типи:

1. Відкриті рекурентні НМ (вони не мають прихованих нейронів: прикладом може бути НМ Хопфілда).
2. Рекурентні перцептрони з прихованими шарами.
3. Ядерні рекурентні НМ (мережа Ельмана та RMLP є базовими моделями, що стали основою для розробки ядерних рекурентних нейромереж).

Основною проблемою, яка стримує практичне впровадження *рекурентних НМ*, є складність і висока трудомісткість процесу їх навчання. Виключенням є відкриті рекурентні нейромережі, але їх застосування обмежено задачами, в яких множина динамічних образів є фіксованою.

Більша ефективність рекурентних НМ пояснюється їх кращим пристосуванням до вирішуваної задачі. Але при збільшенні розмірності задачі або зростанні ступеня невизначеності вхідних даних переваги рекурентних НМ втрачаються: вони є ефективними лише для відносно простих задач.

Обчислювальна складність навчання за методом зворотного поширення в часі становить від $O(N^3)$ до $O(N^4)$, що значно перевищує оцінку $O(N^2)$ для статичних нейромереж [195].

Найбільші витрати ресурсів припадають на *коригування зворотних зв'язків нейромережі*, тому вирішення цієї проблеми потребує суттєвого удосконалення архітектури та методів коригування зворотних зв'язків рекурентної нейромережі або взагалі відмови від їх модифікації у процесі навчання. Дослідження з удосконалення архітектури привели до створення гібридних рекурентних нейромереж із застосуванням нейропарадигм радіально-базисних функцій [132; 143; 166; 180]. Деякі з цих моделей показали вражаючу ефективність: тривалість навчання складала лише кілька десятків епох [143].

Більша ефективність рекурентних НМ пояснюється їх кращим пристосуванням до вирішуваної задачі. Але при збільшенні розмірності задачі або зростанні ступеня невизначеності вхідних даних переваги рекурентних НМ втрачаються: *вони є ефективними лише для відносно простих задач.*

Для радикального вирішення цієї проблеми в рекурентній нейромережі ESN (Echo State Network) [142; 164], була запропонована повна відмова від коригування зворотних зв'язків при навчанні рекурентної НМ. При її навчанні модифікуються лише прямі зв'язки нейронів прихованого шару. Нейромережу ESN часто називають рідинною (Liquid State Machine), бо прихований шар мережі ESN як деякий резервуар, містить фрагменти динамічних образів, з яких формуються реакції нейромережі.

До мережі ESN застосовують звичайні методи навчання, як для мереж прямого поширення, наприклад, перцептронів. При цьому проблема швидкого зростання відстані між мінімальними та максимальними значеннями ваги зв'язків [164] усувається методами, які базуються на припущенні, що контекстний шар змінює свої властивості набагато повільніше, ніж реакція НМ на зміни зовнішніх даних.

11.8. Нейрорегулятори

Нейрорегулятори (використання НМ для управління динамічними об'єктами) мають здатність до навчання, до адаптації при зміні умов керування, стійкість до пошкоджень, швидкодію внаслідок нейромережевого паралелізму і придатні для синтезу нелінійних регуляторів. Відомі численні приклади практичного застосування нейронних мереж для керування літаком, вертольотом, автомобілем-роботом, швидкістю обертання валу двигуна тощо.

Динаміка поведінки об'єкта описується формулами у дискретному вигляді

$$S(k+1) = \Phi(S(k), u(k));$$

{(k+1)-й стан об'єкта управління є функцією k-го стану};

$$y(k+1) = \Psi(S(k)),$$

{(k+1)-й вихід об'єкта управління є функцією k-го стану},

де $S(k) \in R^N$ – стан об'єкта управління порядку N на такті k; $u(k) \in R^P$ – значення P – мірного вектора управління на такті k; $y(k+1) \in R^V$ – значення V – мірного виходу об'єкта управління на такті (k+1).

Нейроуправління динамічними об'єктами є напрямом, що знаходиться на стику таких різних дисциплін, як *автоматичне управління, штучний інтелект, нейрофізіологія.*

Нейронні мережі мають ряд унікальних властивостей, які роблять їх потужним інструментом для створення систем управління і дозволяють

реалізувати *будь-який необхідний для процесу нелінійний алгоритм управління* при неповному, неточному і суперечливому описі об'єкта управління [31; 94].

НМ може безпосередньо генерувати керуючі впливи на об'єкт, або може виконувати допоміжні функції (придушення шуму, оперативна зміна коефіцієнтів ПД-контролера).

В задачах нейроуправління об'єкт управління розглядають як *чорний ящик*, в якому контролюються поточні значення *входу і виходу*. Стан об'єкта вважається недоступним для зовнішнього спостереження, хоча розмірність вектора станів зазвичай вважається відомою.

Навчальні пари для навчання НМ можуть набиратись, наприклад, або в результаті запису дій людини-оператора, або автоматичного регулятора; чи в режимі он-лайн. Для стабілізації перехідних процесів може використовуватись еталонна модель [121; 181; 201].

11.9. Імпульсні нейронні мережі

Подальший розвиток НМ привів до створення частотних імпульсних моделей нейрона.

Виконані дослідження довели, що реальний біологічний нейрон отримує на входах багатомірний потік імпульсів напруги – спайків, і сам теж генерує на виході спайки [86; 109]. Спайк – це сукупність імпульсних сигналів, кортеж імпульсів входу та виходу нейрона. З'явилися свідчення, що деякі біологічні нейрони для кодування сигналу використовують точний час приходу спайков [135; 141]. В роботі [178] показано, що в деяких структурах мозку частотна інформація перекодується у часовий код.

Виникає припущення, що у нейронів живих організмів обробка послідовностей спайків є однією з основних операцій, і реальний нейрон здатний розв'язати задачі аналізу по даних багатомірного потоку імпульсних вхідних сигналів [109].

Нейрони Маккаллока-Питса, Adaline та Тоянда, персептрон можна використовувати у дискретному режимі, і з цієї точки зору *їх теж можна віднести до імпульсних нейронів*.

Спайки надають можливість обробки одним нейроном сигналів у часі і є корисними в задачах прогнозування; розпізнавання образів, які швидко рухаються, та управління [41; 192].

Імпульсний нейрон виконує зважене підсумовування (інтегрування) вхідних імпульсних сигналів до досягнення цією сумою певного заданого порогу, після чого на виході нейрона формується вихідний імпульс, а зважена сума обнулюється.

У спайковому нейроні інтегрується як просторова так і часова інформація: вхідний спайк, який прийшов у час t_1 , може змусити нейрон до генерації вихідного спайка у час $(t_1 + \Delta t_1)$. Це вказує на можливість адаптив-

ного керування об'єктами (віртуальними агентами чи роботами). Прикладом використання спайкових НМ є розпізнавання облич багат шаровими НМ, у яких використання часового кодування дозволило багаторазово зменшити об'єм пам'яті обчислювальних ресурсів.

Звичайно вважається, що імпульсний нейрон підсумовує вхідні імпульси з деякими вагами, і, якщо підсумок перевищує деякий поріг, то нейрон на кожному кроці моделювання видає вихідний сигнал: 1 – вихідний спайк, 0 – немає спайка. Найпростішою моделлю спайкового нейрона є модель «Integrate-and-Fire» [149], що описується диференціальним рівнянням першого порядку. Якщо накопичена (інтегрована) напруга у деякий момент часу переходить обумовлений поріг, то нейрон генерує на виході спайк, а значення накопиченої напруги скидається до деякої величини. Але, на відміну від бінарних та частотних моделей нейронів, в спайковому нейроні ведеться контроль не лише частоти, але й часової інформації: вхідний спайк, який прийшов у деякий момент часу, може змусити нейрон генерувати вихідний спайк в інший момент часу. Це дозволяє зробити припущення, що спайкові НМ дозволяють природними чином урахувати час. Тому науковці все більше уваги приділяють НМ з імпульсними сигналами.

Наразі виявлені наступні позитивні особливості імпульсної передачі сигналів:

1. Економія енергії, бо скорочується час проходження сигналу.
2. Підвищується поміхосталість щодо шумових сигналів.
3. Спайки реалізують високопаралельні процеси і не вимагають синхронізації і сигналів на кожний крок обчислень ЕОМ. Ці особливості спрощують архітектуру, зменшують розміри і енергоспоживання нейрокомп'ютерів. В результаті задача розв'язується більш простими НМ з меншою кількістю зв'язків.
4. Спайкі здатні обробляти і безперервні сигнали поряд з частотними (приклад – бінарні моделі Маккаллока-Питса). При цьому безперервна величина кодується довжиною інтервала часу між спайками.
5. Спайковий нейрон реагує на історію сенсорних входів 6. У цілому імпульсний нейрон може фіксувати події у часі, нести більше інтелектуальне навантаження, а імпульсний характер процесів забезпечує зменшене споживання енергії при апаратній реалізації нейронів. Підвищена інформативність частотного сигналу із-за можливості зміни частоти, фази, амплітуди, показників широтно-імпульсної модуляції. В результаті з'явилися НМ із затримками, рекурсивні мережі з використанням процедури зворотного розповсюдження похибки у часі, НМ з використанням часової структури сигналу при застосуванні диференціальних рівнянь.
6. У 90-х роках відкрили явище зміни ваги входу у залежності від точного часу прибуття вхідного спайку і часу генерації вихідного спайку

(Spike Timing Dependent Plasticity, STDP) [137]. Явище STDP є узагальненням правила Хебба:

– якщо подія А відбулась раніше події В, то зв'язок між А та В треба підсилити;

– якщо подія А відбулась пізніше події В, то зв'язок між А та В треба зменшити.

У цьому випадку створюється причинно-наслідкова сенсорна асоціація з урахуванням порядку виникнення подій у часі (класичне правило Хебба цього не ураховувало).

7. У деяких нейронів має значення лише частота імпульсів, яка лежить у деяких інтервалах [127; 167; 190]. Нейрон нелінійно підсумовує ці сигнали і видає на виході власний частотний сигнал. При навчанні бінарних і частотних НМ використовують метод навчання Д. Хебба [158]: якщо дві події А та В відбуваються одночасно, то їх зв'язок потрібно підсилити, бо це створює сенсорну асоціацію між ними.

8. Звичайно зміною амплітуди спайків та зміною їх продовження у часі можна нехтувати [86].

9. Спайковий нейрон оперує з безперервним вхідним багатомірним потоком спайків як з потоком точкових подій.

Недоліки спайків:

1. Недостатня практичність математичних методів їх навчання (в основному використовуються евристичні правила, які не гарантують розв'язок задачі).

2. Спайкові моделі ураховують зміну у часі внутрішнього стану та історією сигналів при їх обробці, що ускладнює використання математичних методів навчання. В деяких випадках спайкові НМ використовуються без їх навчання.

3. Розрив процесу роботи у моменти генерації спайків затрудняє будівництво безперервних функцій при навчанні НМ.

4. Складність автоматизації формування структури їх зв'язків.

Найпростіша модель спайкового нейрона [149] оперує з вхідним багатомірним потоком спайків (як з потоком точкових подій) і описується диференціальним рівнянням першого порядку; якщо в деякий момент часу накопичена напруга перевищує встановлений поріг, то значення напруги зменшується до деякого *заданого значення*, а нейрон генерує спайк на виході.

На рис. 11.9.1 показана модель апаратної реалізації імпульсного нейрона, яка містить джерело живлення U , два фотодіоди ($VD1$, $VD2$) та ланцюг з послідовно увімкнених стабілітрона $VD3$ (пороговий елемент), резистора R і світло випромінювача $VD4$. З ланцюгом паралельно з'єднаний конденсатор C .

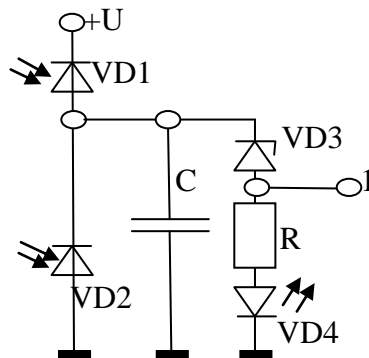


Рис. 11.9.1. Модель нейрона

Вивід 1 стабілітрона VD3 є електричним виходом моделі; оптична апертура фотодіода VD1 є входом для збуджувальних сигналів моделі нейрона; оптична апертура другого фотодіода VD2 є входом для гальмівних сигналів нейрона; оптична апертура світловипромінювача VD4 є оптичним виходом моделі нейрона [40].

Збуджувальні вхідні оптичні сигнали на фотодіоді VD1 викликають струм, що заряджає конденсатор C, а гальмівні вхідні оптичні сигнали на фотодіоді VD2 викликають розряд конденсатора C. Вихідний електричний імпульс формується на резисторі R (знімається з виходу 1), а оптичний імпульс – на світловипромінювачі VD4 в момент, коли напруга заряду конденсатора C під впливом вхідних сигналів досягає порогового значення, і через стабілітрон VD3 та резистор R протікає імпульс струму розряда конденсатора C.

Розрядження конденсатора C викликає запирання стабілітрона VD3, і процес повторюється знов. Частота вихідних імпульсів моделі прямо пропорційна різниці між фотострумом першого фотодіода 4 та другого фотодіода 5.

Навчання і моделювання бінарних і частотних нейронів ведеться у дискретному часі на основі даних сигналів на поточному такті часу. Тому частотні нейрони для їх навчання повинні мати необхідну навчальну інформацію у кожний момент часу [109]. Існують рекурсивні частотні НМ, які використовують процедуру зворотного розповсюдження похибки у часі [179].

Основним постулатом при навчанні бінарних та частотних НМ є правило Д. Хебба [158]: якщо події A та B відбуваються одночасно, то зв'язок між ними потрібно підсилити, що призведе до створення сенсової асоціації між подіями. При навчанні частотних НМ застосовують також метод зворотного розповсюдження похибки.

З'явилися НМ з часовою затримкою інформації [17], а також рекурсивні, але вони поступаються по рівню інтелектуальності перед спайковими НМ.

Розглянуто задачу побудови нейроном ідеалізованої спайк-послідовності з використанням *ідеалізованих імпульсів з нульовою тривалістю* і одиничною амплітудою [100].

Запропонована багатошарова самонавчальна спайк-нейронна мережа: навчання спайк-нейронної мережі було виконано за 4 епохи, у порівнянні з 50 епохами навчання мапи Кохонена [12]. При цьому розглядалась трьохшарова НМ, призначена для класифікації. 1-й захований шар призначений для кодування аналогових вхідних сигналів $0 \leq x_j \leq 1$ у кортежі імпульсів-спайків і перетворення амплітудно-імпульсного сигналу у часоімпульсну форму. При цьому один вхідний сигнал обробляється групою (пулом) нейронів з гаусовими активаційними функціями. Пул нейронів генерує кортежі імпульсів. Одночасно у 1-му шарі використовуються додаткове кодування активаційними функціями, які забезпечують отримання «вузьких» та «широких» імпульсів. Таким чином, на кожний елемент вхідного вектора $0 \leq x_j \leq 1$ на виході 1-го шару створюється кортеж спайка.

2-й захований шар НМ створений з m спайк-нейронів, кожний з яких має множину входів з різним часом затримки та власними вагами, які можуть налагоджуватись, та має вихідний вектор потенціалів з n -елементів.

3-й вихідний шар створений m стандартними радіально-базисними нейронами, при цьому центр кожної радіально-базисної багатомірної функції співпадає з вектором центроїдом відповідного кластера. У 3-му шарі нейрони пов'язані між собою латеральними зв'язками, які забезпечують віднесення вхідного образу до конкретного класу по принципу «переможець отримує все», аналогічно самоорганізуючим мапам Кохонена. Латеральні зв'язки не використовуються у процесі самонавчання. Їх роль визначається у процесі використання навченої НМ, якщо на вхід попадає образ, що не належить до навчальної вибірки. Навчання виконується за правилом Хебба. Якщо вибірка достатньо велика, то навчання виконується за одну епоху. Центри радіально-базисних функцій вихідного 3-го шару визначаються після визначення ваг нейронів 2-го шару.

З метою *логічної універсалізації і інтелектуалізації (розширення набору виконуваних функцій)* процесорів та нейронних модулів НМ розглянута обробка **двох вхідних векторів** (мінімум). Кожний вхідний безперервний сигнал перетворюється за допомогою ШІМ. Частоти з виходу ШІМ порівнюють і кодують в логічному пристрої. Нормалізатори та ШІМ з часовим та просторовочасовим нелінійним інтегруванням описані в [173], а деякі варіанти реалізовані як універсальні логічні елементи двозначної логіки [173].

Існують нейрони, які підсумовують частоти вхідних сигналів і обробляють зважену суму активаційною функцією; частота сигналів змінюється у деякому діапазоні; вихід нейрона має власну частоту [109].

Існують пропозиції використання ПЛІС для побудови кластерних нейронних мереж і запропонована архітектурна схема використання [43].

Запропоновані функціональні схеми нейронів із частотно-, широтно- та фазо-імпульсним представленням інформації. Згідно одного із способів здійснення синапсного множення, будь-який із параметрів імпульсу – амплі-

літуда, частота, тривалість, зсув у часі – відповідає значенням вхідної інформації, а інший – вазі синапсу, два останні параметри є константами [189].

У праці [37], показано, що порівнянні з відео-імпульсним, амплітудно-імпульсним і фазо-імпульсним методами представлення інформації, **частотно-імпульсний метод є найбільш завадостійким і відмовостійким, та забезпечує найбільшу швидкодією.**

11.10. Адаптивна резонансна теорія нейронної мережі

Теорія адаптивного резонансу (АРТ) запропонована Г. А. Карпентером та С. Гросбергом (1988) [140]. НМ АРТ набуває нову інформацію при зберіганні набутих знань. Подібні задачі не може розв'язувати багатошаровий перцептрон, мережі Кохонена та Ліпмана-Хеммінга. НМ АРТ має високий ступінь паралелізму роботи і підґрунтя у біологічних та поведінкових даних.

Свою назву АРТ-мережа отримала із-за того, що в ній безперервно змінюваний вхідний вектор X передається у прямому і зворотному напрямках (резонує) між двома шарами F_1 (шар порівняння) та F_2 (шар розпізнавання), з яких складається НМ (рис. 11.10.1). Навчання НМ відбувається лише у процесі резонансу [103].

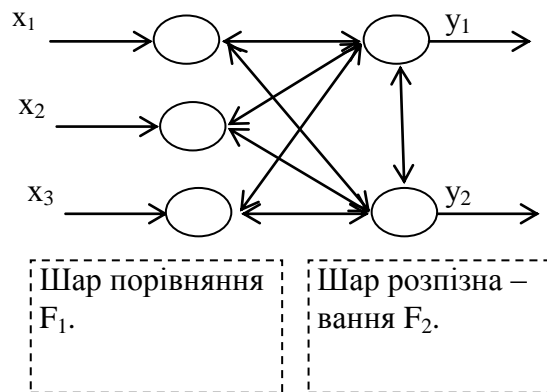


Рис. 11.10.1. Основні компоненти АРТ – класифікатора Карпентера-Гросберга

У процесі функціонування НМ АРТ розрізняють наступні фази:

1. *Ініціалізація.* Обирають елементи матриці двійкових ваг U (звичайно, рівними 1), елементи матриці безперервних ваг T та параметр подібності p . Матриця безперервних ваг T використовується для визначення подібності вхідного вектора X наявному еталонному образу у фазі розпізнавання. Матриця двійкових ваг U використовується для визначення коректності віднесення даного вхідного вектора X до наявного еталонного вектора у фазі розпізнавання. Вагові коефіцієнти даних матриць повільно змі-

нюються. Окремий прилад Приймач_2 реалізує логічну функцію «або» вхідного вектора X у вигляді $G_2 = x_1 \vee x_2 \vee \dots \vee x_j \vee \dots \vee x_n$ і передає дані у шар розпізнавання F_2 . Приймач_1 зоставляє вхідний сигнал X з сигналом від шару розпізнавання F_2 .

2. *Розпізнавання*. Визначається переможець матриці безперервних ваг T .

3. *Порівняння* відбувається покомпонентно між елементами вектора-переможцем X та вихідним сигналом U шару розпізнавання F_2 з обчисленням подібності S вектора-переможцем.

4. *Пошук*. Якщо подібність $S > p$, то пошук не потрібний. У іншому разі запам'ятовані образи переглядаються для рошуку образу найбільш відповідного вхідному вектору X . Якщо жодний з наявних еталонів не відповідає вхідному сигналу X , то вхідний сигнал перетворюють на новий еталон.

5. *Навчання* виникає тоді, коли виявляється відповідність між вхідним сигналом X та образом еталону у пам'яті. Навчання відбувається без учителя.

Перевагою НМ є навчання без учителя, а недоліками є складний процес навчання та необмежене збільшення числа нейронів у процесі навчання, особливо у присутності шуму. Описана мережа АРТ працює достатньо швидко у аналоговому варіанті і *повільно у цифровому*. Тому для цифрового варіанта бажано використовувати паралельні процесори.

11.11. Згорнута нейронна мережа

Згорнута НМ запропонована Я. Лекуном і є найліпшою з точки зору розпізнавання зображень. Вона входить у склад технологій глибинного навчання і має спеціальну архітектуру, призначену для ефективного розпізнавання зображень. Вона спрямована не на попіксельне запам'ятовування зображення, а на узагальнення вхідної інформації з отриманням відносної сталості роботи при поверненні та при зсуву зображення.

Назву НМ отримала по операції згортки, за якою кожний фрагмент зображення попереднього шару множиться на матрицю (ядро) згортки поелементно, а результат підсумовується і записується у аналогічну позицію наступного зображення, яке у свою чергу обробляється аналогічно. Ці дві операції повторюються при переміщенні від входу до виходу: перший з таких шарів зветься шаром згортки, а другий – шаром субдискретизації. *Кожна наступна пара шарів використовує інформацію попередньої пари шарів*. Це створює спроможність розпізнавання складних образів.

В операції згортки використовується «вікно» (ще зветься «ядро» згортки) – обмежена матриця ваг невеликого розміру у 5×5 пікселів (можлива інша кількість пікселів), яке пересувають по зображенню одного шару (спочатку, по вхідному шару) і яке несе в собі деякий елементарний образ (наприклад відрізок прямої під визначеним кутом). Обмежена матриця ваг «вікна» множиться на відповідну матрицю зображення, а отримане число (згортка вхідного зображення по «зображенню вікна») записується у вигляді числа в аналогічну позицію вихідного зображення. Зсув «вікна» що-

разу виконують на один чи два нейрони (піксели) замість п'яти, щоб не пропустити шукану ознаку. Спочатку розглядають схожість, наприклад, з елементами штрихових ліній під одним кутом, потім – під іншим кутом; потім змінюють образ елементів «вікна» з ліній під різними кутами на дуги чи інші образи. Використовується *багато таких «вікон» (наборів ознак)*, причому їх не закладає дослідник: вони автоматично створюються НМ у процесі окремого навчання. Результати кожного подібного аналізу зображення за допомогою різних «вікон» записується у наступному шарі у вигляді *карти ознак*. Сукупність карт різних ознак дозволяє розпізнавати складні складові частки зображень.

Наступна «операція об'єднання» зменшує розмірність сформованих *карт ознак*. При цьому вважається, що інформація про наявність розшукованої ознаки є важливішою за інформацію про її точні координати. Тому з *кількох сусідніх нейронів карти ознак* обирається максимальний і приймається *за один нейрон карти зменшеної розмірності* (іноді використовують операцію осереднення сигналів сусідніх нейронів). Наслідком цієї операції є зменшення об'єму інформації (прискорення подальших розрахунків) та менша залежність від масштабу вхідного зображення.

Після проходження кількох шарів наступна карта ознак звичайно вироджується у вектор і навіть скаляр, але отримують сотні таких карт ознак. На виході НМ часто додатково використовують кілька шарів перцептронів, на вхід якому подають кінцеві карти ознак.

Навчання НМ відбувається за методом зворотного розповсюдження похибки.

До недоліків НМ відносять невизначеність кількості шарів, розмірності «вікна» згортки, кількості «вікон», кроку зсуву «вікна», ступеня зменшення зображень при згортанні карт ознак, наявності і параметри вихідного перцептронів. Хоча існують конфігурації дуже добре працюючих мереж, але не існує алгоритма її визначення для розв'язку нових задач.

11.12. Деякі напрямки розвитку нейронних мереж

НМ моделюють лише частку процесів, що відбуваються в мозку. Існує багато проблем, які поки що не підвладні розв'язуванню наявними комп'ютерами.

Подальший розвиток НМ пов'язують з поширенням застосування апаратної реалізації, медичної діагностики, паралельних нейрообчислень, побутової техніки («розумних» помешкань з адаптивним керуванням, інтелектуальних помічників, інтелектуальних агентів по обробці і користуванню інформації), керування та захисту складних систем з контролем рівня їх безпеки, розпізнаванню мови, забезпечення безпеки фінансових операцій.

Клітинні нейронні мережі – це система чисельно фіксованих, локально з'єднаних нелінійних обчислювальних одиниць [146].

Кожна одиниця має багато входів і один вихід. Інформація закодована через початковий стан обчислювальної одиниці, входи та змінні, які визначають її поведінку. Взаємодія між обчислювальними одиницями відбувається лише у межах охоплення сусідніх одиниць певним радіусом. Обмін інформацією між віддаленими ділянками нейронної мережі може відбуватися шляхом дифузії. КНМ застосовують для обробки зображень, аналізу 3D поверхонь, обчислення диференційних рівнянь в часткових похідних, приведення негеометричних завдань до геометричних карт, моделювання зору та інших функцій. Поведінка КНМ більш складна, ніж поведінка окремої клітини (нейрона). Це вказує на те, що КНМ мають емерджентні властивості.

Моделі нейрокомп'ютерів. Термін «нейрокомп'ютер» вживається для позначення всього спектру робіт в рамках підходу до побудови систем штучного інтелекту, заснованого на моделюванні елементів, структур, взаємодій і функцій нервової системи. Так як наразі дослідження в цій ділянці ведуться в основному на рівні моделей нейронних мереж, то розуміння терміну «нейрокомп'ютери» звужують, ставлячи знак рівності між ним і нейронними мережами.

Залежно від способу реалізації моделей нейронних мереж виділяють 4 рівні нейрокомп'ютерів.

– *Теоретичний.* Роботи, в яких в тій чи іншій формі (математичної, алгоритмічної, словесної і т. д.) представлено опис моделей нейронних мереж.

– *Програмний.* Моделі нейронних мереж, програмно реалізовані на звичайних послідовних комп'ютерах.

– *Програмно-апаратний.* Застосовують співпроцесор для прискорення моделювання нейронних мереж.

– *Апаратний.* Використовують фізично реалізовані моделі нейронних мереж.

Специфічність нейромережових операцій, а також зверхпаралельність структури і функціонування моделей нейронних мереж надзвичайно уповільнюють їх реалізацію на звичайних послідовних комп'ютерах. Це призвело до появи спеціалізованих обчислювальних пристроїв для ефективного моделювання нейронних мереж – нейрокомп'ютерів у вузькому сенсі слова.

Модулярна нейронна мережа складається з групи модулів (окремих НМ), які керуються певним посередником. Кожний модуль має власні вхідні сигнали для розв'язку певного підзавдання з групи завдань, які повинна виконати вся модулярна нейронна мережа [131; 156; 163]. Модулі не взаємодіють між собою. Посередник приймає вихідні сигнали кожного модуля НМ, обробляє їх і створює вихідний сигнал усієї модулярної мережі. Посередник не отримує інших вхідних сигналів, окрім сигналів з виходів модулів.

Перевагою модулярних НМ є здатність розбивки великих НМ на частки, якими легше керувати. Це також зменшує обчислювальний час, бо розрахунки виконуються одночасно (паралельно) усіма модулями. Збільшується і надійність НМ та ефективність її навчання внаслідок розділу ділянки діяльності всієї НМ на кілька частин.

Мережа Ворда має три шари: вхідний, прихований і вихідний. Внутрішній (прихований) шар нейронів розбитий на блоки. Виходи кожного блоку з'єднуються з входами нейронів третього вихідного шару. Крім того, входи вихідного шару можуть бути з'єднані з виходами вхідного шару. В результаті можна використовувати різні функції для паралельної обробки вхідних сигналів у різних блоках прихованого шару, тобто НМ аналізує різні аспекти даних. НМ дає гарні результати при прогнозуванні і розпізнаванні образів.

Об'єднання нейронних мереж, генетичних алгоритмів та нечітких систем.

Проблеми нейронних мереж, генетичних алгоритмів та нечітких систем викликають велику цікавість внаслідок своїх типових переваг (рис. 11.12.1):

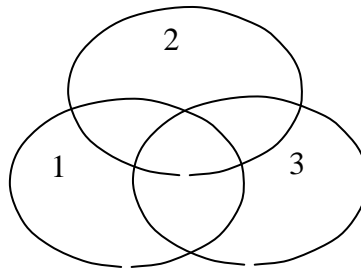


Рис. 11.12.1. Взаємні зв'язки між нейронними мережами (1), генетичними алгоритмами (2) та нечіткими системами (3)

Аналіз нейронних мереж може охоплювати кілька напрямків досліджень [104]:

1. *Нейронні мережі (НМ)* дозволяють робити висновки по процесах, які не мають поки що чіткого осмислення, математичного та логічного обґрунтування. Можливість самонавчання дозволяє їм безперервно самоудосконалюватись й перенавчатись у процесі роботи по отриманим результатам експериментів.

2. *Генетичні алгоритми* за рахунок виконання умов природного вибору, який панує в природі, з множини можливих рішень обирають рішення, які є найбільш конкурентоздатними, «найбільш пристосовані» для розв'язання проблеми. Доведено, що генетичні алгоритми *можуть на порядок скоротити розрахунки* порівняно з методами стохастичного обрання ваг НМ. Генетичні алгоритми можна використовувати для підбору ваг зв'язків і топології НМ, для формування бази правил та функцій належності нечіткої системи.

3. *Нечіткі множини та нечіткі висновки* дозволяють описувати складні процеси з ослабленими вимогами до наявності та точності даних, бо використовуються відомості на рівні лінгвістичних змінних та функцій належності. Методи нечітких множин дозволяють визначити *параметри схрещування та мутації генетичних алгоритмів, архітектуру та коефіцієнти швидкості навчання НМ*.

Кожний з указаних напрямків може використовуватись окремо один від одного, але вони можуть мати й пересічення. У зв'язку з цим доцільно розглядати роздільні описи вказаних систем, а також результати, отримані при їх об'єднанні.

Об'єднання цих трьох наукових напрямків, кожний з яких охоплює різні області одного й того ж процесу, дозволяє отримати безперервні або дискретні рішення в результаті навчання.

Когнітрон та неокогнітрон є багат шаровими мережами прямого поширення, що навчаються без учителя (використовується принцип конкурентного навчання, коли підтримку отримує більш збуджений нейрон, і інші подавляються) [103].

Спочатку був створений *когнітрон* [152]. Кожний шар когнітрону реагує на власний підобраз вектора X (частку загального вхідного образу X), який ускладнюється за рахунок об'єднання підобразів у міру переміщення від входу когнітрону до його виходу; останній шар реагує на *весь образ незалежно від його положення, розміру й орієнтації у просторі*. У нейронів передбачений функціональний вплив зваженого підсумку окремих гальмуючих входів v_i , $i=1, 2, \dots, m$, на зважений підсумок окремих збуджуючих входів x_j , $j=1, 2, \dots, n$ з виходом результуючого сигналу на лінійну (без обмежень) активаційну функцію. У процесі навчання ваги зв'язків у когнітроні можуть зростати без обмежень. Це не призводить до занадто сильного зростання ваг із-за особливостей формули виходу когнітрону.

Навчання виконується звичайно без учителя. При цьому відсутні попередньо визначені вихідні образи; мережа навчається сама їх розпізнавати. В заданій області шару навчається лише сильно збуджений нейрон (це порівнюється з «елітним навчанням», або навчанням «найбільш розумних» елементів). Добре навчанні нейрони отримують підтримку з метою збільшення їх збудження, але велике збудження одного вузла гальмує збудження сусідніх вузлів.

Схожий з еталоном вектор збуджує відповідний нейрон, а не схожий подавляється конкуренцією. Збудження нейрону визначається *зваженою сумою його збуджуючих та гальмуючих входів*. Орієнтовно вихід збудженого нейрону у когнітроні визначається відношенням його збуджуючих входів до гальмуючих.

Вихід нейрона може розраховуватись згідно логічних умов:

$$\begin{aligned} \text{OUT} &= \text{NET}, & \text{якщо } \text{NET} \geq 0; \\ \text{OUT} &= 0, & \text{якщо } \text{NET} < 0, \end{aligned}$$

де $OUT = \frac{E}{I} - 1$; $E = \sum_{j=0}^n w_j x_j \gg 1$; $I = \sum_{i=0}^m w_{ii} v_i \gg 1$; E – зважена сума збуджуючих сигналів; I – зважена сума гальмуючих сигналів; w_j – вага j -го збуджуючого зв'язка (вага може бути тільки додатною); x_j – вхід j -го збуджуючого сигналу, w_{ii} – вага i -го гальмуючого зв'язка, v_i – вхід i -го гальмуючого сигналу.

Неокогнітрон має схожість із когнітроном: він є багат шаровою ієрархічною системою, у якій на більш високих рівнях вузли реагують на більш складні образи [152]. У порівнянні з когнітроном неокогнітрон більш потужний у розпізнаванні зорових образів незалежно від викривлень, зміни масштабу, обертання та перетворень.

Кожний *внутрішній шар* неокогнітрону розглядається як один модуль, який складається з двох шарів (S_q та C_q). Перший (сенсорний) шар S_q , де $q = 1, 2, \dots, Q$ – порядковий номер внутрішньої пари шарів, складається з $(8 \times 8) \dots (128 \times 128)$ клітин. Найчастіше використовуються поля у (16×16) та (32×32) клітини. Нейрон шару S_q активізується лише тоді, коли знайдений певний образ (фрагмент вхідного образу) у попередньому C_{q-1} – шарі. Шар S_q виділяє з них відповідні підобрази і потім передає це на другий шар пари C_q , який корегує спотворені зображення.

Пара шарів (S_q та C_q) утворюють незмінний модуль неокогнітрону, бо зв'язки між шарами S_q та C_q не змінюються у процесі навчання (змінюються лише ваги зв'язків між модулями).

Ця система використовує самоорганізацію у процесі навчання і має властивість самовідновлення: якщо один із вузлів вийде з ладу, то буде найдений інший вузол, який реагує сильніше, і цей вузол буде виконувати функції зіпсованого вузла. В ідеальному випадку лише один нейрон вихідного шару повинен збуджуватись. Але в дійсності збуджуються кілька нейронів; тому образ визначається з врахуванням співвідношення їх виходів.

12. Метод потенціалів

12.1. Потенціали у просторі цифрових ознак

Точковий електричний заряд в однорідному середовищі створює електричне поле (рис. 12.1.1) [26]. Для виду зверху радіальні лінії на рис. 12.1.1 – це силові лінії електричного поля, а пунктирні концентричні кола – це лінії однакового (рівного) потенціалу.

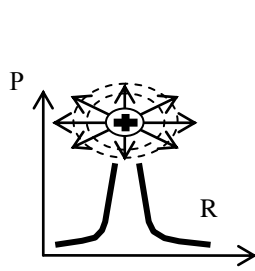


Рис. 12.1.1. Зміна потенціалу P у залежності від відстані R до точкового заряду

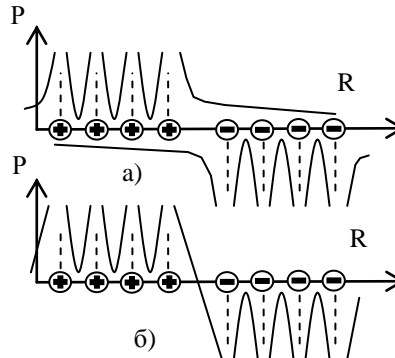


Рис. 12.1.2. Розподіл потенціалів

Потенціал для точкового електричного заряду дорівнює

$$P = kq / R^2$$

де k – постійний коефіцієнт; q – величина заряду; R – відстань від даної точки до заряду.

Зміна потенціалу в залежності від відстані до заряду зображується жирними лініями на рис. 12.1.1.

Коли електричне поле створене кількома зарядами, то потенціал в будь-якій точці поля дорівнює сумі потенціалів, створених у цій точці кожним зарядом. Якщо заряди, що створюють поле, розміщені компактною групою, то потенціал буде мати найбільше значення у середині групи зарядів і зменшуватись по мірі віддалення від групи, тобто потенціал будь-якої точки простору буде характеризувати віддалення цієї точки від усієї групи зарядів у цілому.

На рис. 12.1.2, а показаний розподіл потенціалів окремо для додатних і для від'ємних зарядів, а на рис. 12.1.2, б – результуюче поле.

Ці основні особливості електричних потенціалів використовуються для розпізнавання образів за методом потенціалів: потенціал в будь-якій точці простору залежить від зарядів різних класів і «переможець забирає все».

Недоліком наведеної функції для розрахунку потенціалів є те, що вона збільшується до нескінченності при $R=0$. Тому цю формулу замінюють на інші, які не мають такого недоліку, але схожі за властивостями. Суттєвих

обмежень на вид потенціальних функцій не накладається. Для цього використовують будь-яку функцію, форма якої нагадує криву електричного потенціалу: вона повинна досягати максимуму у кожній точці із зарядом і повинна зменшуватись по всім напрямкам збільшення відстані від цієї точки (точка таким чином вважається як би джерелом потенціалу – звідси назва методу).

Одна з можливих формул для визначення потенціалів в трьохвимірному просторі в точці x, y, z може мати вигляд

$$\varphi = \frac{q}{1 + \alpha[(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2]},$$

де (x, y, z) – координати об'єкта, який розпізнається; (x_0, y_0, z_0) – координати розміщення заряду q ; $R^2 = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2$ – квадрат «відстані» між зарядом q з координатами (x^0, y^0, z^0) та об'єкта з координатами (x^0, y^0, z^0) .

Припустимо, що ми маємо деякі класи A та B з власними точками позитивних та негативних зарядів (рис. 12.1.3). Тоді будь-яка точка C буде належати позитивному класу (A) або негативному класу (B) у залежності від знаку потенціалу у цій точці. Тобто в заданій точці C ми повинні розрахувати два сумарних потенціали від усіх точок-зарядів кожного класу A та B окремо і найбільший серед них указує, до якого класу належить точка C .

Якщо образи розміщені в трьохмірному просторі (мають три ознаки), то поверхня потенціалів буде мати образ, який нагадує горні хребти, а долини є розділяючою поверхнею (рис. 12.1.4).

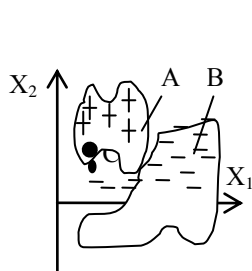


Рис. 12.1.3. Розпізнавання образів для двох класів.

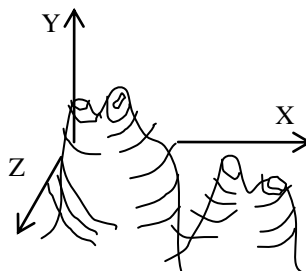


Рис. 12.1.4. Вид потенціалів у трьохмірному просторі

12.2. Двоїсті значення потенціалів рецепторів у просторі зображення

У найпростішому випадку у полі рецепторів вектор образу складається з кодів, кожний з яких може мати значення або «1» або «0» (збуджений рецептор має потенціал «1», а не збуджений – «0») [26]. Відстані між цими векторами дорівнюють сумі абсолютних значень

різниць кодів векторів рецепторних зображень. Якщо у попередньому випадку ми розраховували потенціал, для визначення якого використовували, наприклад, евклідові відстані, то у полі рецепторів ми безпосередньо використовуємо відстані між потенціалами рецепторів, які дорівнюють або кількості розрядів, якими розрізняються вектори відповідних об'єктів, або евклідовій відстані між векторами зображень.

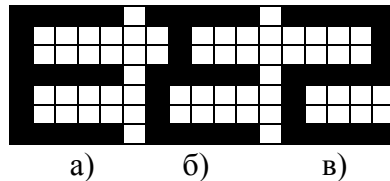


Рис. 12.2.1. До визначення відстані відносно еталону (а) спотвореної літери «Е» (б) та цифри «2» (в)

Для об'єктів рис. 12.2.1 на площі 5x7 рецепторів визначаємо такі коди рецепторів для отриманих символів (вони розбиті по рядках зображень):

- для рис. 12.2.1, а: 11111 10000 10000 11111 10000 10000 11111;
- для рис. 12.2.1, б: 11111 01000 01000 11111 10000 10000 11111;
- для рис. 12.2.1, в: 11111 00001 00001 11111 10000 10000 11111.

Звідси отримуємо відстані між зображеннями як абсолютне значення різниць кодів векторів:

- різниця у кодах між рис. 12.2.1, а та рис. 12.2.1, б 00000 11000 11000 00000 00000 00000 00000;
- різниця у кодах між рис. 12.2.1, а та рис. 12.2.1, в 00000 10001 10001 00000 00000 00000 00000.

З наведеного прикладу випливає, що між еталоном «Е» та зіпсованим символом «Е» існує така ж сама відстань, як і між еталоном «Е» та цифрою «2», бо кількість розрядів, що розрізняє ці об'єкти однакова.

Цей приклад показує, що прийнятий нами метод кодування у вигляді потенціалів «1» та «0» недостатньо добре передає різницю об'єктів. «Значне зміщення» елементів фігур по полю рецепторів (що як правило суттєво псує об'єкти) у даному випадку призвело до таких само змін відстаней у просторі рецепторів, що і «незначне зміщення» (яке фактично не змінює об'єкт). В результаті знижується надійність розпізнавання. Виникла потреба оцінки «величини зміщення».

12.3. Дробові значення потенціалів рецепторів у просторі зображення

Надійність розпізнавання може бути підвищена, якщо враховувати при кодуванні величину зміни потенціалу при зміщенні елементів фігур по полю рецепторів. Це можна виконати шляхом ступінчастої апроксимації

функції потенціалів у рецепторному полі й врахуванням взаємного впливу зарядів [26].

На рис. 12.3.1, а показана функція потенціалу, який виникає під дією точкового заряду і розрахований по формулі (12.1.1). Цю зміну потенціалу ми апроксимували ступінчастою функцією рис. 12.3.1, б, коли рецептори приймають значення або «1» або «0». Але криву рис. 12.3.1, а можна апроксимувати і дробовими значеннями потенціалів рецепторів, як це показано на рис. 12.3.1, в, г.

Для цього можна, наприклад, використати такі правила розрахунку потенціалів (кодів векторів зображення):

1. Збуджений рецептор має потенціал «1» (на рис. 12.9 він показаний чорною коміркою).
2. Рецептор, розміщений рядом із збудженим рецептором має потенціал «0,5» (на рис. 12.3.2 він показаний сірою коміркою).
3. Усі інші рецептори мають потенціал «0» (на рис. 12.3.2 вони показані білою коміркою).

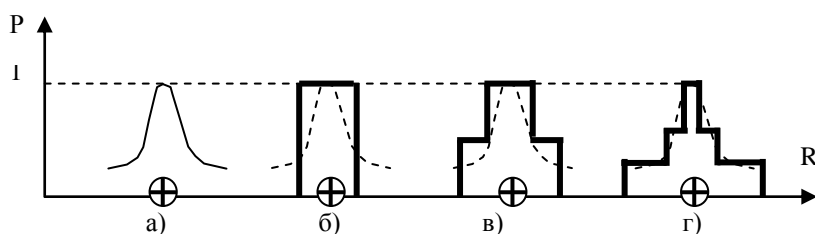


Рис. 12.3.1. Апроксимація функції потенціалів заряду дробовими значеннями



Рис. 12.3.2. Переміщення об'єкту у рецепторному полі

Тоді коди об'єктів у рецепторному полі 6x3 для отриманих зображень мають значення (вони розбиті по рядках зображень):

1. Рис. 12.3.2, а: $\{ \frac{1}{2} \frac{1}{2} \frac{1}{2} 0 0 0 \frac{1}{2} 1 \frac{1}{2} 0 0 0 \frac{1}{2} \frac{1}{2} \frac{1}{2} 0 0 0 \}$.

2. Рис. 12.3.2, б: $\{ 0 \frac{1}{2} \frac{1}{2} \frac{1}{2} 0 0 0 \frac{1}{2} 1 \frac{1}{2} 0 0 0 \frac{1}{2} \frac{1}{2} \frac{1}{2} 0 0 \}$.

12. Рис. 12.3.2, в: $\{ 0 0 0 \frac{1}{2} \frac{1}{2} \frac{1}{2} 0 0 0 \frac{1}{2} 1 \frac{1}{2} 0 0 0 \frac{1}{2} \frac{1}{2} \frac{1}{2} \}$.

Відповідні елементи відстаней між отриманими образами об'єктів:

1. Між рис. 12.3.2, а та рис. 12.3.2, б: $\{ \frac{1}{2} 0 0 \frac{1}{2} 0 0 \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} 0 0 \frac{1}{2} 0 0 \frac{1}{2} 0 0 \}$.

2. Між рис. 12.3.2, а та рис. 12.3.2, в: $\{ \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} 1 \frac{1}{2} \frac{1}{2} 1 \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \}$.

З цих даних отримуємо евклідові відстані:

$$d_{a\bar{b}} = \sqrt{8\left(\frac{1}{2}\right)^2 + 10 * (0)^2} = \sqrt{2} ; d_{\bar{a}a} = \sqrt{16\left(\frac{1}{2}\right)^2 + 2 * (1)^2} = \sqrt{6}$$

Таким чином, $d_{a\bar{b}} < d_{a\bar{v}}$, у той час, як при старому кодуванні без врахування величини зміщення отримали б $d_{a\bar{b}} = d_{a\bar{v}} = \sqrt{2}$.

Розглянемо подібний метод кодування для більш складних об'єктів на базі правил:

1. Збуджений рецептор має «власний потенціал», рівний «1».

2. Усі рецептори (у тому числі і збуджені) отримують від сусідніх збуджених рецепторів додатковий потенціал «0,5». Сусіднім вважається збуджений рецептор, який має сумісну грань або кутову точку (зверху, знизу, праворуч, ліворуч) із рецептором, для якого визначається потенціал.

3. Усі інші рецептори мають потенціал «0».

Таке кодування можна використати, наприклад, для визначення відстані між образами спотвореної літери «Е» (рис. 12.3.3). Отримані коди, як приклад, показані на полях 5x7 рецепторів.

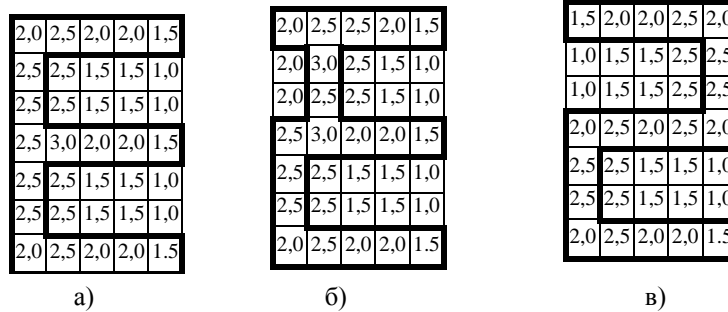


Рис. 12.3.3. До визначення відстані відносно еталону (а) спотвореної літери «Е» (б) та цифри «2» (в).

З цих даних отримуємо евклідові відстані:

$$d_{a\bar{b}} = \sqrt{29(0)^2 + 4\left(\frac{1}{2}\right)^2 + 2 * (1)^2} = \sqrt{3} ; d_{a\bar{v}} = \sqrt{19(0)^2 + 8\left(\frac{1}{2}\right)^2 + 4 * (1)^2 + 4\left(\frac{3}{2}\right)^2} = \sqrt{15}$$

Таким чином, $d_{a\bar{b}} < d_{a\bar{v}}$.

Завдання

1. На площині ознак $X_1=0...N$ та $X_2=0...N$ (N – порядковий номер студента у групі) визначити дві довільні області існування двох класів (Ω_1, Ω_2). В кожній області довільно розмістити по 5 об'єктів. Координати цих об'єктів визначають їх ознаки (у реальній системі РО ці дані отримуються від датчиків).

Методом потенціалів визначити належність до класів трьох довільно розміщених об'єктів (А, В, С): об'єкт А розмістити в області класу Ω_1 ,

об'єкт В – в області класу Ω_2 , об'єкт С – між цими класами.

Привести графік розміщення об'єктів і межі класів. Визначити вектори ознак (властивостей) об'єктів. Потенціали розраховувати за формулою $\varphi = \frac{1}{1+2R}$. Належність об'єкту А, В, С до класу визначається шляхом обрання найбільшого значення серед підсумкових потенціалів, які створюються сукупностями об'єктів окремих класів у точці з координатами об'єкта, що класифікується. При розрахунках використовувати евклідову відстань.

2. У полі рецепторів 9x12 зробити з першої літери власного прізвища два зміщення трьох рецепторів:

- Мале зміщення на один рецептор.
- Велике зміщення на два або більше рецепторів. Методом потенціалів знайти дві відстані між трьома отриманими таким чином об'єктами. Еталон – незіпсована літера.

Використати метод потенціалів із такими правилами: збуджений рецептор має власний потенціал «+1» і додає «+0,5» потенціалу до кожного сусіднього рецептора – зверху, знизу, праворуч, ліворуч, по діагоналі, включаючи збуджений рецептор.

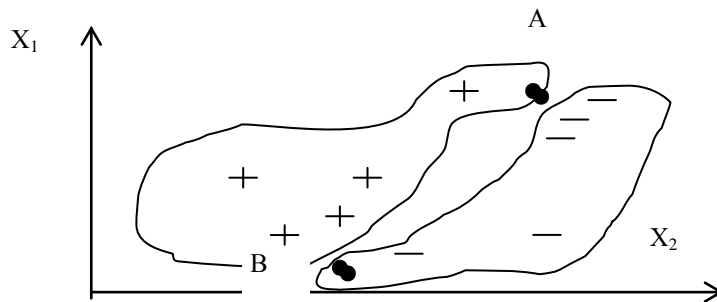


Рис. 12.3.4. Приклад розміщення

3. На площині ознак $X_1=0\dots N$ та $X_2=0\dots N$ (N – порядковий номер студента у групі) визначити дві довільні області існування двох класів (Ω_1 , Ω_2). В кожній області довільно й нерівномірно розмістити по 5 об'єктів (рис. 12.3.4).

Координати цих об'єктів визначають їх ознаки (у реальній системі РО ці дані отримуються від датчиків). За методом потенціалів розробити програму навчання нейронної мережі для двох класів, яка б вірно визначала класи об'єктів А та В, незважаючи на нерівномірність розміщення об'єктів.

13. Лінгвістичні методи визначення образів

13.1. Структурний метод розпізнавання образів

13.1.1. Загальні відомості

Живі істоти передають інформацію через природну мову. Наприклад, людина звичайною мовою може описати об'єкт (класифікувати його) і за описом – розпізнати об'єкт.

Структурні (лінгвістичні) методи розпізнавання описують структуру об'єкта [36]. Для цього об'єкт розкладається на атомарні елементи. Загального методу визначення непохідних атомарних елементів не існує.

Кожний атомарний елемент позначається символом. Сукупність символів створює алфавіт мови. Символи об'єднуються у ланцюги, утворюючи слова та речення мови, *які визначають як самі елементи об'єкта так і їх розміщення відносно один одного*. Тому у структурних методах «міра близькості» специфічна: визначається схожість речення невідомого об'єкта (воно описує структуру невідомого об'єкта) з еталонним реченням класу.

У принципі структурні (лінгвістичні) методи можна використовувати для розпізнавання об'єктів будь-якої природи, але найчастіше вони застосовуються для аналізу сигналів та зображень: аерофотографування; сучасних медичних систем обслідування та діагностики (рентгенографії, машинної томографії, електрокардіограми, енцефалограми і т. п.); «органів зору» роботів; результатів реалізації методів неруйнівних іспитів у промисловості; дистанційного зондування шляхом фотографування із супутників; об'ємних зображень об'єктів, отриманих за допомогою голографічних установок; акустичних сигналів працюючих машин і даних радіолокації; для аналізу сцен.

Типова система структурного розпізнавання має *блок попередньої обробки сигналів* (призначений для переведення об'єкта у цифрову форму, зменшення шумів, нормалізації та квантування сигналів, розклад їх у ряди), кодування, сегментації, *блок формування опису* (виділення атомарних елементів та зв'язків між ними; формування речення, що описує об'єкт), *блок синтаксичного аналізу* (здійснює граматичний аналіз речення і визначає, якою граматикою воно породжується; приймає рішення щодо віднесення об'єкта до відповідного класу).

13.1.2. Формальні граматики структурного методу

Основою структурного методу є формальна граMATика з чотирьох елементів [36]

$$G = \{V_A, V_D, P, S\},$$

де $V_A = \{a, b, c, d\}$ – скінчений алфавіт символів для позначення окремих атомарних (основних, термінальних, непохідних) елементів об’єкта, що розпізнається. При цьому використовуються маленькі літери алфавіту;

$V_D = \{S, A, B, C, D\}$ – скінчений допоміжний алфавіт – метазмінні, які використовуються у правилах для виведення правильних слів: для природних мов такими змінними є граматичні класи – іменники, дієслова, займенники тощо. При цьому використовуються великі літери алфавіту. Аксиома S є складовою допоміжного алфавіту V_D ;

S – літера допоміжного алфавіту V_D , якою позначається *аксіома* – граматичне правило, *із якого починається створення* всіх можливих правильних слів мови шляхом послідовного використання правил підстановок. Наприклад, в аксіомі $S \rightarrow aB$ вказується початковий елемент « a », із якого починається структурне будівництво образу, та символ допоміжного алфавіту « B », який використовується для підстановки замість нього інших символів згідно правила $B \rightarrow bC$. Далі для заміни « C » на інший елемент або їх групу використовується наступне граматичне правило і т. д. ГраMATика (граматичні правила) може вміщувати кілька аксіом $S: S \rightarrow aB$ з використанням граматичних правил $B \rightarrow bC, \dots; S \rightarrow AB, A \rightarrow aC, \dots$;

$P = \{\alpha_1 \rightarrow \beta_1, \alpha_2 \rightarrow \beta_2, \dots, \alpha_m \rightarrow \beta_m\}$ – множина граматичних (породжувальних) правил, або правил підстановки, де α_i позначається символом, який береться з V_D , або вміщує хоча б один символ із V_D , а β_j вміщує вказівки по будівництву об’єкта. Правило вигляду $\alpha_i \rightarrow \beta_i$ можна розуміти як « α_i замінюється на β_i ». У β_i можуть вміщуватись будь-які задані конкретні вказівки щодо будівництва образу. Наприклад, якщо ми маємо аксіому $S \rightarrow aB$ та граматичне правило $B \rightarrow bC$, то можемо отримати abC ; якщо ми маємо ланцюг XYZ із правилом підстановки $Y \rightarrow abcdK$, то ми отримуємо $XabcdKZ$; якщо додати правило $X \rightarrow bbc$, то ми отримуємо $bbcabcdKZ$.

Об’єднання V_A та V_D складає повний алфавіт V граматики G .

Уявимо собі, що ми маємо дитячий конструктор. Тоді алфавіт атомарних елементів V_A позначає окремі початкові деталі конструктора, які знаходяться в коробці; допоміжний алфавіт V_D позначає метазмінні (або атомарні елементи, або зібрану сукупність кількох атомарних деталей); граMATика вміщує аксіому S (вона вказує, з якої деталі потрібно почати з’єднання, та наступну метазмінну) та наступні граматичні правила, які вказують наступні деталі, що з’єднуються; місця з’єднання атомарних деталей треба обумовити у правилах, або окремо.

Приклад. Припустимо, що нам надаються чотири еталонних образи (рис. 13.1.1) і потрібно скласти відповідну граматику по їх класифікації. Атомарними елементами цих образів є вектори рис. 13.1.2.

Складемо граматику для чотирьох зорових образів рис. 13.1.1:

$$G = \{V_A, V_D, P, S\},$$

де $V_A = \{a, b, c, d\}$ – скінчений алфавіт символів для позначення **атомарних (основних, термінальних, непохідних)** елементів рис. 13.1.2. При цьому використовуються маленькі літери алфавіту;

$V_D = \{S, A, B, C, D\}$ – скінчений допоміжний алфавіт – метазмінні, які використовуються для виведення правильних слів: ab, bc, cd, da . Аксиома S є складовою допоміжного алфавіту V_D ;

S – аксіома – початкове правило, із якого шляхом послідовного використання правил підстановок виводяться всі можливі слова мови;

$P = \{S \rightarrow aA, S \rightarrow bB, S \rightarrow cC, S \rightarrow dD, A \rightarrow b, B \rightarrow c, C \rightarrow d, D \rightarrow a\}$ – множина граматичних (породжувальних) правил, або правил підстановки. Правило вигляду $S \rightarrow aA$ можна розуміти як « S замінюється на aA ». Далі замість « A » використовується правило підстановки « $A \rightarrow b$ ». У даному випадку аксіома S складається з чотирьох можливих компонентів: aA, bB, cC, dD .

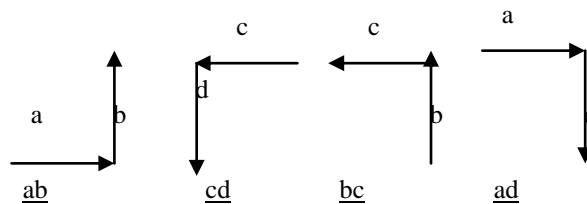


Рис. 13.1.1. Чотири еталонних образи

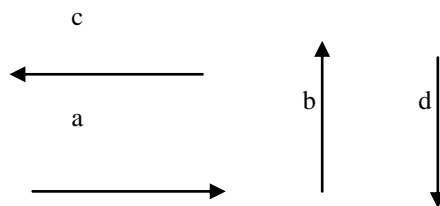


Рис. 13.1.2. Атомарні елементи a, b, c, d

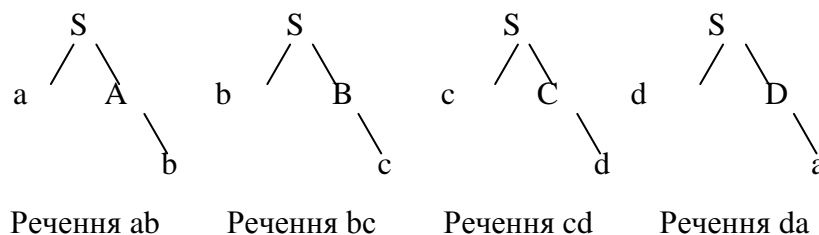


Рис. 13.1.3. Виведення речення у вигляді дерева

Отримана граматики надає можливість описати зображення рис. 13.1.1 словами: ab, bc, cd, da . Якщо мати змогу виділяти атомарні елементи зображення рис. 13.1.1, то кожне з них може бути описаним на створеній мові одним із правильних слів (ab, bc, cd, da) і може бути порівняне з відповідним еталоном.

Речення вважається сформованим, якщо його ланцюг уміщує тільки атомарні елементи.

У даному випадку використаний захід, типовий для ланцюгової граматики: елементи двомірних об'єктів дозволяється з'єднувати тільки у особливих точках (виділені точки інтерпретуються як «голова» та «хвіст» атомарного елемента).

На рис. 13.1.3 показаний процес виведення речень у вигляді дерев, які підкреслюють ієрархічну структуру процесу розпізнавання образу. Кожне одне дерево задає одне речення. Початковий символ S виконує роль кореня дерева, а кінцеві вузли, які читаються зліва направо, створюють виведене речення (тобто результат класифікації визначається на нижньому ярусі дерева). Звичайно з вузлів дерев відходять дві гілки, але у деяких випадках можуть відходити кілька гілок. Подібні дерева дають ієрархічне уявлення про те, які атомарні елементи, підобрази та відношення характеризують об'єкт.

Використання дерев – не складна й зручна процедура, але для них відсутні оптимальні процедури навчання.

Більшість граматик дозволяє отримати значну кількість речень, а при використанні ймовірних правил виведення – і нескінченну кількість речень.

За класифікацією С. Хомського усі відомі граматики поділяються на чотири класи у залежності від правил підстановки:

1. *Граматики типу 0* (необмежена граматики). Правила підстановки для таких граматик можуть бути будь-якими, бо на них не накладається обмежень.

2. *Граматики безпосередньо складових* мають правила підстановки вигляду

$$\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2,$$

де $A \in V_D$; $\alpha_1, \alpha_2 \in V_A$; $\beta \in V$; $\beta \neq \varepsilon$; ε – один з символів мови.

Ці правила можна трактувати таким чином: символ допоміжного алфавіту A можна замінити на непустий ланцюжок β у контексті α_1, α_2 . Тому граматики безпосередньо складових також звать *контекстно-залежними*.

3. *Граматики контекстно-вільна (безконтекстна)* використовує правила виду

$$A \rightarrow \beta,$$

де $A \in V_D$; $\beta \in V$.

Це трактується як заміна символу A допоміжного алфавіту на непустий ланцюжок β без врахування контексту.

4. *Автоматна (регулярна) граматики* використовує правила виду

$$A \rightarrow aB \text{ або } A \rightarrow b,$$

де $A, B \in V_D$; $b \in V_A$.

Наприклад, мова $L(G) = \{a^n b^m c^n d^m, n=1, 2, \dots, m=1, 2, \dots\}$ породжується за допомогою правил автоматної граматики $S \rightarrow aA, A \rightarrow aA, A \rightarrow bB; B \rightarrow bB, B \rightarrow cC; C \rightarrow cC, C \rightarrow dD; D \rightarrow dD$.

13.1.3. Виведення граматики

Розглянемо більш детально процес створення лінгвістичного методу для опису зорового зображення, який по суті є процесом створення формальної граматики мови [36].

1. Об'єкт класу розділяється на **атомарні** (непохідні, термінальні, основні) **елементи**, які є ознакою об'єкта. Атомарний елемент означає якусь суттєву ознаку образу. Наприклад, атомарні елементи автомашини – це колесо, кабіна водія, кузов, мотор. Універсального методу виділення атомарних елементів не існує. Для шрифту атомарними елементами можуть бути штрихи для створення літер. Для виділення ознак зображення використовується розклад у математичні ряди Фур'є; поліномів Ерміта, Лежандра, Чебишева.

Недоліком лінгвістичного методу й є якраз необхідність точного безпомилкового розпізнавання атомарних елементів.

2. **Кожний з атомарних елементів позначається своїм окремим символом алфавіту мови V_A** . Усі ці символи у сукупності складають **алфавіт** мови V_A . В алфавіт V_A можуть входити літери, цифри, знаки. Приклади двох алфавітів із символів: алфавіт $V_{A1} = \{a, b, c, \dots, z, 0, 1, 2, \dots, 9\}$; алфавіт $V_{A2} = \{\blacktriangleleft, \blacksquare, =, \ominus, +, \bullet, \blacktriangle, \odot, x, k, m\}$. З одного набору символів можна побудувати кілька мов (як наприклад, із символів української мови можна побудувати англійську, французьку і інші мови).

3. Починається **створення граматики**. Для синтаксичного розпізнавання образів вводять допоміжний алфавіт (із великих літер) V_D . Визначаються характеристики взаємного розміщення атомарних елементів («Зверху», «Ліворуч», «Праворуч», «У середині», «Норма», «Відхилення від норми», правила з'єднання векторів або відрізків і т. п.) і вводяться відповідні предикати.

Центральне місце в граматиці займають граматичні правила – правила підстановки. Наприклад, аксіома $S \rightarrow \ast B \text{ середині } (A, a)$ означає, що в середині образу голови сніговика «а» (це – намальоване коло) ми повинні вставити символ допоміжного алфавіту «А», який потім замінюється поступово на атомарні елементи – «очі», «рот», «ніс» з указівкою їх взаємного розміщення. Виявляється, що за рахунок комбінування обмеженої кількості атомарних (непохідних) елементів, можна отримати опис різних об'єктів, як це виконується за рахунок звичайної мови.

Таким чином, граматики – це множина правил підстановки (наприклад, $A \rightarrow aB$ або $A \rightarrow b$, де A, B береться із V_D ; b береться із V_A), за якими з символів алфавіту V_A будуються *правильні слова-ланцюги*, що відображують структуру об'єкта. Ліві та праві частини правил підстановки можуть вміщувати символи як алфавіту V_A так і алфавіту V_D . Граматичне правило (правило підстановки) може використовуватись будь-яку кількість разів. Порядок використання правил у граматиці є довільним, а дві різні граматики можуть породжувати однакову мову (тоді вони зветься еквівалентними граматики).

Слова виокремлюються за допомогою символу алфавіту – *пропуску*, а символи допоміжного алфавіту V_D дають змогу поступово будувати опис об'єкта з атомарних елементів. Слово може складатись з одного або кількох символів. Крім того можуть використовуватись предикати (наприклад у вигляді слів «зверху», «у середині», «ліворуч») для визначення взаємного розміщення атомарних елементів. Часто замість поняття «слово» використовується поняття «ланцюг».

Ланцюг – це кінцева послідовність символів, які беруться з алфавіту. Довжина ланцюга може бути будь-якою, включаючи порожній ланцюг. Ланцюги можуть розглядатись як мовні об'єкти різних рівнів: словоформи, слова, словосполучення, речення та ін. Усі можливі ланцюги алфавіту V створюють універсальну множину рядків (замикання V). В універсальну множину ланцюгів може входити і порожній ланцюг. Речення складаються з ланцюгів *атомарних елементів*.

Структурний підхід не має ще строгої математичної теорії, і тому розглядається як сукупність евристичних засобів. Граматика створюється в основному дослідником, хоча є й автоматизовані системи створення граматик.

Задача полягає у тому, щоб отримати правила опису об'єктів з елементів заданого алфавіту символів і щоб із багатьох можливих речень відібрати найбільш суттєві. Якщо використовувати породжуючі правила з ймовірністю переходів, то отримуємо клас стохастичних граматик, які передають як структуру так і статистичну інформацію про образи. Допоміжні символи, що входять у перетворений ланцюг, можуть знову перетворюватись згідно правил граматики і т. п. *Речення вважається сформованим, якщо його ланцюг складається з атомарних елементів.*

4. Розпізнавання *образу* найбільш простим способом полягає у порівнянні вхідного ланцюга образу з еталонним у термінах атомарних елементів. Можливе також використання порівняння графів, що описують образи.

Під кожний клас розробляється власна граматика. У випадку m класів розглядається m граматик і пов'язаних із ними мов. Об'єкт, що розпізнається, відноситься до класу Ω_i тільки у тому випадку, якщо він є граматично вірним реченням мови.

Етап класифікації невідомого образу настає після створення граматики. Він визначається прийняттям рішення, наскільки невідомий об'єкт

відповідає еталону класу. Значення граматики полягає також у тому, що вона дозволяє говорити про синтаксичну правильність, або неправильність представленого образу.

Виконується це за допомогою граматичного розбору, який поділяється на два види:

1. Граматичний розбір зверху вниз. Починаючи з початкового символу S намагаються отримати задане речення шляхом послідовного використання правил.

2. Граматичний розбір знизу угору. У цьому випадку відновлюється дерево виведення, починаючи з елементів основного словника і використовують інвертовані правила підстановки. Ця процедура починається з конкретного речення і закінчується при отриманні початкового символу S .

Схеми граматичного розбору неефективні, якщо потребують повного перебору правил підстановки; цього часто можна уникнути, якщо перевіряти припустимість проміжних результатів. Підвищення ефективності граматичного розбору можна забезпечити за допомогою врахування синтаксичних правил, які встановлюють або забороняють специфічні відношення між об'єктами.

Фактично, для кожного вхідного об'єкту будується одна або кілька граматик, які відповідають різним структурним описам вхідного об'єкта. Можна використовувати такі загальні правила:

1. Виконується сегментація і виявляються атомарні елементи кожного зображення.

2. Уведенням правил підстановки будуються підобрази, які відображують лише деякі властивості образу. Будування підобразів із використанням атомарних елементів повторюється до тих пір, доки неможливо буде побудувати щось нове. Перевіряються правила підстановки для всіх об'єктів. Усі описи, що не пройшли перевірку, із подальшого розгляду виключаються.

3. Перевага надається коротким граматакам. Будується граматика шляхом вилучення всіх надлишкових правил. Відшукується така пара правил скінченого допоміжного алфавіту V_D , що заміна одного іншим призводить до появи надлишкових правил із наступним їх вилученням. Упорядковуються, спрощуються, об'єднуються отримані граматики. З кількох отриманих граматик обирається найкоротша.

Важливою особливістю синтаксичних методів РО є те, що кожній граматиці можна привести у відповідність деяку логічну машину, наприклад, машину Тюрінга. Тоді створення лінгвістичної системи РО складається з розробки граматики, конструювання автомата й використання автомата для обробки масивів даних.

13.1.4. Приклади використання структурного методу

1. *Структурний метод розпізнавання кривих ліній по Фрімену* (H. Freeman) показаний на рис. 13.1.4. Атомарними елементами є одиничні вектори 1...8 із різними напрямками, які використовують для апроксимації кривої лінії, переміщуючись із початку кривої лінії в кінець [36].

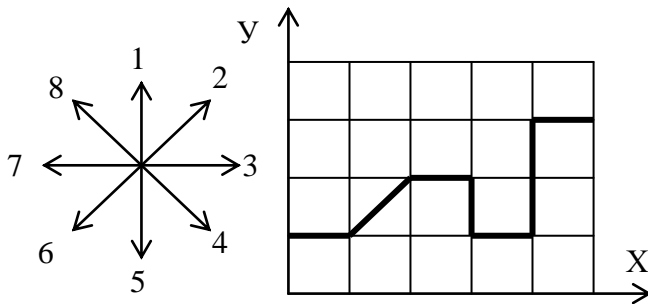


Рис. 13.1.4. Розпізнавання кривих по Фрімену (код кривої: 32353113)

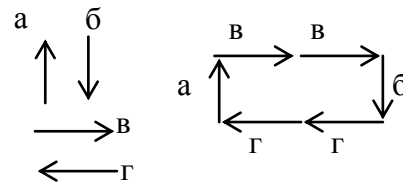


Рис. 13.1.5. Розпізнавання прямокутників (код «аввбгг»)

Кожний з атомарних елементів (кожний з векторів) є символом мови. Ці символи ми позначили цифрами 1...8. Символи складають алфавіт мови. За допомогою символів можна апроксимувати будь-яку криву при умові, що початок наступного одиничного вектора з'єднується з кінцем попереднього вектора. Ця апроксимація має вигляд ланцюга-слова «32353113» для кривої рис. 13.1.4. Якщо йти з кінця кривої в її початок, то слово змінюється на «75571767». Таким чином, ми створили мову для опису кривих. Для перевірки належності невідомої кривої до класу, достатньо перевірити ступень тотожності слова невідомого об'єкту зі словом-еталоном класу, наприклад, «32353113». Звичайно подібні граматики будуються на базі апріорних відомостей та евристичних міркувань.

2. *Структурний метод розпізнавання прямокутників* показаний на рис. 13.1.5. Атомарними елементами зображення («атомами») є одиничні вектори «а, б, в, г». Символи «а, б, в, г» складають алфавіт мови. За допомогою символів можна апроксимувати будь-який прямокутник при умові, що кінець будь-якого одиничного вектора з'єднується з початком наступного вектора. Для прямокутника рис. 13.1.5 ця апроксимація має вигляд ланцюга-слова «аввбгг» або «ав²бг²». Будь-який прямокутник у рамках указаних одиничних векторів має вигляд «аⁿв^mбⁿг^m», де $n, m = 1, 2, 3, \dots$ – степені одиничних векторів. Таким чином, ми створили мову $L = \{a^n, v^m, b^n, z^m; n, m = 1, 2, \dots\}$ для опису будь-яких прямокутників.

3. *Структурний метод розпізнавання областей* (рис. 13.1.6) був запропонований Павлідисом (Pavlidis T.) у 1968 році. Непохідні елементи для областей – це півплощини у просторі ознак образів (у полі спостере-

жень). На рис. 13.1.6 півплощини області, у яких виконуються нерівності, позначені великими літерами $\{A, B, \Gamma, Д\}$, а інші півплощини позначені маленькими літерами $\{a, б, в, г, д\}$. Можна вважати, що символи визначення півплощин – це «літери», отримані багатокутники – «слова», а всі випуклі «багатокутники» – це «речення». Багатокутник рис. 13.1.6 описується словом «АБВГД». Це нагадує виділення ділянки рішень у задачах лінійного програмування.

Очевидно, що за допомогою одиничних векторів-півплощин «1...8» рис. 13.1.6 також можна побудувати будь-які області.

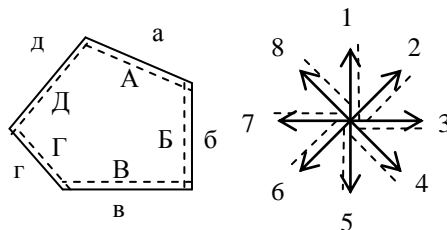


Рис. 13.1.6. Розпізнавання областей (код АБВГД)

4. **Розпізнавання електрокардіограми** показано на рис. 13.1.7 [154]. Граматика розпізнавання має вигляд

$$G_4 = \{V_A, V_D, P, S\},$$

де $V_A = \{p, r, t, b\}$ – алфавіт атомарних елементів {вони позначені відрізками на рис. 13.1.7}; $V_D = \{S, A, B, D, E, H\}$ – допоміжний алфавіт правил будівництва образу; S – початкове правило; $P = \{S \rightarrow pA; A \rightarrow rbB; B \rightarrow b; B \rightarrow tD; D \rightarrow b; D \rightarrow bE; E \rightarrow b; E \rightarrow bH; H \rightarrow b; H \rightarrow bS; H \rightarrow pA\}$ – правила підстановки.

Тут розглядаються принципи будівництва граматики розпізнавання електрокардіограми.

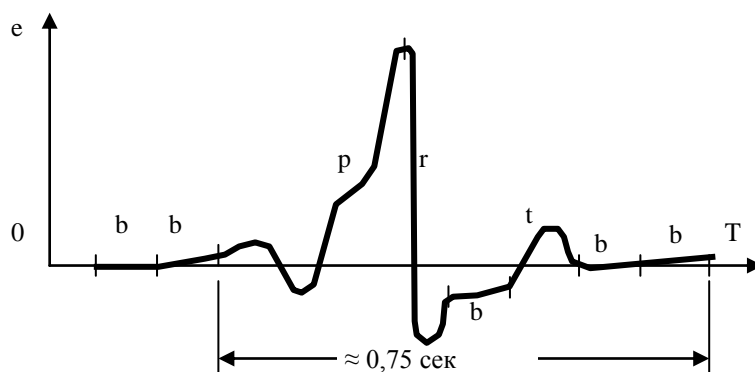


Рис. 13.1.7. Електрокардіограма

Якщо за початок підрахунку прийняти хвилю «р», то у вказаних позначеннях електрокардіограм може бути описана послідовностями символів $prbtb$, $prbtbb$, $prbtbbb$ і т. п.


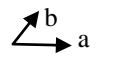
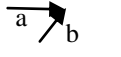
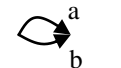
Кардіограма – це реєстрований у вигляді осцилограми електричний сигнал між електродами, розміщеними на долоні правої руки та стопі лівої ноги. Сигнал (із різницею потенціалів порядку мілівольт) підсилюється, виконується його нормування та реєстрація. Під нормуванням по амплітуді і по часу розуміється приведення осцилограми до стандартного вигляду. Справа у тому, що із-за стану здоров'я людини, фізичних зусиль, віку та інших причин період розширення та скорочення серця (для еталона це порядку 0,75 сек.) і амплітуда можуть суттєво відрізнятись від еталонних значень. Тому кардіограму перед порівнянням з еталоном обов'язково нормалізують.

В результаті отримується безперервна крива, яку треба описати. Атомарними елементами зображення («атомами») кардіограми є окремі ділянки кривої p , r , t , b . Кожний з атомарних елементів є символом мови. Символи складають алфавіт мови. За допомогою символів можна апроксимувати будь-яку кардіограму при умові, що кінець будь-якої ділянки з'єднується з початком наступної ділянки і створені методи безпомилкового визначення символів. Для кардіограми ця апроксимація має вигляд ланцюга-слова « $prbtbb$ ». Таким чином, ми створили мову для опису кардіограм. Для перевірки належності невідомої кардіограми до класу, достатньо перевірити ступень тотожності її слова зі словом-еталоном класу.

5. **Мова опису малюнка (PDL (Picture Description Language))** дозволяє описати графічні об'єкти. Приклад операцій з двома векторами, які мають початок та кінець, показаний у табл. 13.1.1.

Таблиця 13.1.1.

Приклади операцій з двома векторами на мові PDL

Позначення	Операція над векторами a та b	Графічний вигляд
$a+b$	1. Голова « a » примикає до хвоста « b »	
axb	2. Хвіст « a » примикає до хвоста « b »	
$a-b$	3. Голова « a » примикає до голови « b »	
$a*b$	4. Голови « a » та « b » і їх хвости з'єднуються	

На основі наведених в табл. 13.1.1 операцій, на рис. 13.1.8 наведені цифри «3» та «9» для використаних векторів (a , b , c , d) згідно речень $((a + a + d + d) - (a+a)) + ((d + d) + (c + c))$ та $((c + c) + (d + d) + (a + a) + (b + b)) + ((b + b) + (c + c))$.

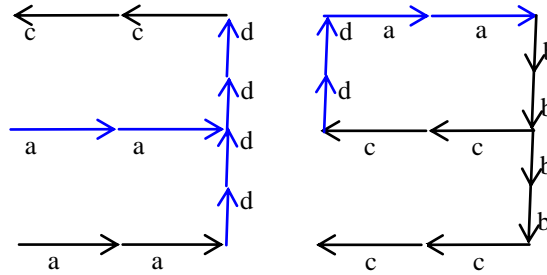


Рис. 13.1.8. Цифри «3» та «9» на мові PDL згідно речень $((a + a + d + d) - (a+a)) + ((d + d) + (c + c))$ та $((c + c) + (d + d) + (a + a) + (b + b)) + ((b + b) + (c + c))$

б. Створення граматики для опису двох рівносторонніх трикутників з трьох одиничних векторів (рис. 13.1.9,а), помічених символами $\{a, b, \blacksquare\}$, показано на рис. 13.1.9,б.

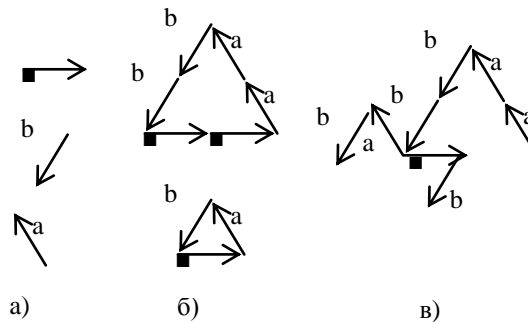


Рис. 13.1.9. Створення граматики опису рівносторонніх трикутників

Вважаємо, що у трикутниках початок кожного наступного вектора з'єднується з кінцем попереднього вектора.

Складемо граматику для двох трикутників: $G_2 = \{V_A, V_D, P, S\}$, де $V_A = \{a, b, \blacksquare\}$ – алфавіт атомарних елементів $\{a, b, \blacksquare\}$; $V_D = \{S, A, B, C, D, E, F, K, L, N\}$ – алфавіт правил будування образу; із якого шляхом послідовного використання правил підстановок виводяться всі можливі термінальні слова мови; $P = \{S \rightarrow \blacksquare A; A \rightarrow \blacksquare B; B \rightarrow a\}; \{S \rightarrow DC; D \rightarrow aK; K \rightarrow a; C \rightarrow EF; E \rightarrow bL; L \rightarrow b; F \rightarrow \blacksquare N; N \rightarrow \blacksquare\}$ – множина граматичних (породжувальних) правил, або правил підстановки.

За допомогою граматики G_2 можна отримати речення для будування образів рис. 13.1.9,б:

$S \Rightarrow A \Rightarrow b \blacksquare B \Rightarrow b \blacksquare a$; $S \Rightarrow DC \Rightarrow aKC \Rightarrow aaC \Rightarrow aaEF \Rightarrow aabLF \Rightarrow aabbF \Rightarrow aabb \blacksquare N \Rightarrow aabb \blacksquare \blacksquare$, де \Rightarrow символ виведення речення.

Для опису трикутників можна також використовувати речення $\{a^N, b^N, \blacksquare^N\}$, де N – кількість одиничних векторів, що будують одну сторону трикутника.

На рис. 13.1.9, в показано будування дерева. В дійсності можна отримати кілька подібних дерев та відповідних речень. Для будування дерева $P = \{A \rightarrow aB; B \rightarrow bC; C \rightarrow bD; D \rightarrow EF; \{E \rightarrow aG; G \rightarrow b\}; \{F \rightarrow \blacksquare H; H \rightarrow b\}\}$ потріб-

но показати початкове правило $S \rightarrow aA$ з наступним нарощуванням дерева за допомогою правил.

7. **Структурний метод розпізнавання опису англійського скоропису** розробили Иден та Хелл. Атомарні елементи у цієї моделі позначені на рис. 13.1.10 літерами «a, b, c, d» («Риска», «Крючок», «Дуга», «Петля»); ці чотири елементи додатково описуються трійкою

$$\sigma_j = [(x_{j1}, y_{j1}), (x_{j2}, y_{j2}), \theta_j]$$

де $(x_{j1}, y_{j1}), (x_{j2}, y_{j2})$ – приблизне положення кінцевих точок сегмента; θ_j – напрямок повороту від першої кінцевої точки до другої. Величина $\theta_j > 0$ якщо поворот виконується по часовій стрілці і $\theta_j < 0$ – у протилежному випадку.

Вони можуть бути перетворені зміною знака θ або віддзеркаленням відносно вертикальної або горизонтальної осі, дякуючи чому *отримуємо 28 штрихів*. В англійському скоропису з цієї кількості *використовується лише дев'ять штрихів*. На рис. 13.1.10 наведений приклад написання слова «globe».

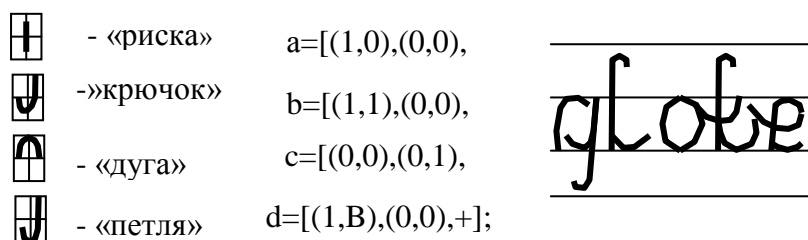


Рис. 13.1.10. Розпізнавання англійського скоропису

При розпізнаванні літер атомарними елементами є найрізноманітніші прямолінійні відрізки та дуги, а граматика складається з набору правил, за якими будується перший відрізок, а потім – усі наступні до отримання певної літери. Аналіз полягає у тому, що для певного зображення літери евристичним методом виявляють всі відрізки та дуги. Потім виділяють відрізок, який може відігравати роль першого при побудові даної літери за даними правилами. Також перевіряється, чи приєднується належним чином до відрізка наступний відрізок даної літери і т. д.

8. **Складання граматики для розпізнавання образу сніговика** рис. 13.1.11:

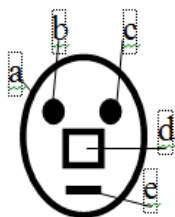


Рис. 13.1.11. Елементи сніговика

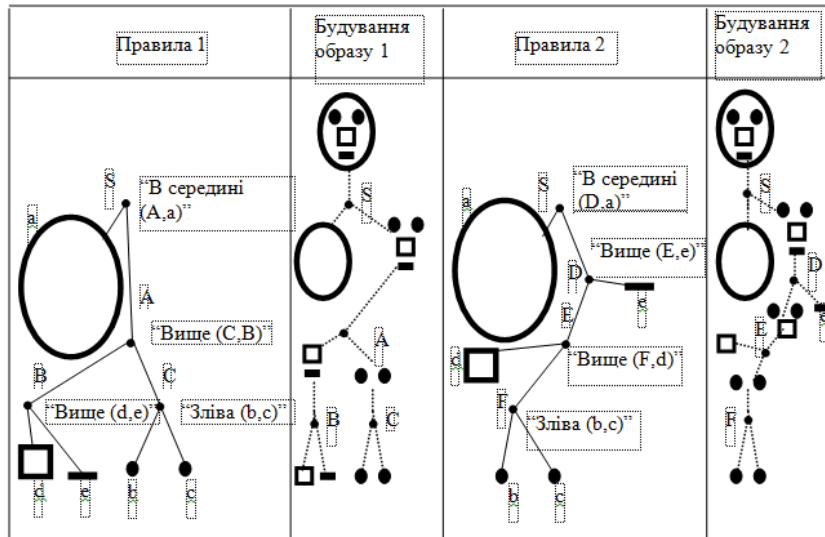


Рис.13.1.12. Виведення граматики сніговика

$$G3 = \{V_A, V_D, P, S\},$$

де $V_A = \{a, b, c, d, e\}$ – алфавіт атомарних елементів $\{a$ – коло; b – ліве око; c – праве око; d – ніс; e – рот}; $V_D = \{S, A, B, C, D, E, F\}$ – допоміжний алфавіт правил будування образу; S – початкове правило; $P = \{S \rightarrow \text{«В середині (A,a)»}; A \rightarrow \text{«Вище (C,B)»}; B \rightarrow \text{«Вище (d,e)»}; C \rightarrow \text{«Ліворуч (b,c)»}; S \rightarrow \text{«У_середині (D,a)»}; D \rightarrow \text{«Вище (E,e)»}; E \rightarrow \text{«Вище (F,d)»}; F \rightarrow \text{«Ліворуч (b,c)»}\}$ – правила підстановки, які зветься предикатами. Кожний виконаний предикат створює новий структурний тип (новий елемент). Сукупність предикатів складає основу граматики мови.

Тут вважаємо, що задані такі предикати:

- «Ліворуч (x,y) » – елемент « x » розмістити зліва від елемента « y »;
- «Вище (x,y) » – елемент « x » розмістити вище елемента « y »;
- «У_середині (x,y) » – елемент « x » розмістити в середині елемента « y ».

Послідовність виведення граматики мови з використанням указаних предикатів показана на рис. 13.1.12.

В дійсності процес виведення граматики складніший: потрібно багаторазово використовувати предикати для будування різних підобразів і серед отриманої сукупності предикатів обрати «найліпші».

Завдання

1. Скласти за допомогою одиничних векторів лінгвістичний опис 4-х перших літер власного прізвища (шрифт – друковані великі літери): алфавіт атомарних елементів V_A ; допоміжний алфавіт правил будування образу V_D ; початкове правило (аксіому) S ; правила підстановки P .

2. Отримати граматику довільного малюнка (дім, машина, каструля і т. д.) з кількістю елементів не менше чотирьох: алфавіт атомарних елементів V_A ; допоміжний алфавіт правил будування образу V_D ; початкове пра-

вило S ; правила підстановки P .

3. Отримати граматику для трьох довільно намальованих осцилограм з однаковими атомарними елементами: алфавіт атомарних елементів V_A ; допоміжний алфавіт правил будовання образу V_D ; початкове правило S ; правила підстановки P . Додатково отримати їх опис за допомогою одиничних векторів.

13.2. Роль СЛОВА у розпізнаванні образів

13.2.1. СЛОВО як матеріальна та ідеальна основа інформатики

Спочатку було СЛОВО. Так сказано в Біблії. За допомогою СЛІВ створив Бог небо, землю, світло, твердь посередині води, сушу, моря, рослини, зірки на небі, тварин та людей.

Але можливість перетворення ідеального поняття СЛОВА (ідеї) у матерію заперечують матеріалісти у своїх суперечках з ідеалістами. Розглянемо це питання детальніше [51].

Головне спрямування СЛОВА – це передача *здобутих знань*. СЛОВО за рахунок переданої інформації примножує добробут людства, дозволяє *генерувати нові знання і збільшує тривалість життя* людини. Підраховано, що 75 % національного продукту США дає кваліфікація робітників, *їх знання*. І лише 25 % національного продукту приходить на матеріальне забезпечення процесу праці. Саме тому держави звертають увагу на підтримку системи навчання.

З точки зору штучного інтелекту СЛОВО є джерелом інформації, *класифікатором* та датчиком семантичних зв'язків між словами.

СЛОВО має вигляд двох форм, розрив між якими призводить до втрати розуміння СЛОВА для людства:

1. *Форму матеріального відбитку* (включаючи звукову та знакову форми) з *власними системами знань*: даним людської та електронної пам'яті, письменності, друкування книжок, Інтернету, преси, пошти, електричного телеграфу, електричного телефону, кіно, радіо, телебачення, ЕОМ, археології.

2. *Форму ідеального, духовного, сенсового, понятійного вмісту СЛОВА*, яка створюється *системою навчання* і зберігається у пам'яті людей теж у вигляді матеріального відбитку, як навчена сукупність нейронів мозку.

Розпізнавання образів вміщує процес вимірювання кількісних та якісних ознак.

Саме тому система розпізнавання образів має *датчики* визначення матеріальних ознак образу і «*понятійні блоки*» (наприклад, *блок логічного висновку*), за допомогою яких вкладається сенс у виміряні

матеріальні ознаки об'єкта, що розпізнається. Образ розпізнається, якщо у вимірянні кількісні чи якісні ознаки вкладається «понятійне розпізнавання».

Без *ідеальної* форми (без навчання, без «понятійного розпізнавання») матеріальний відбиток СЛОВА перетворюється на шум – нічного не значущий малюнок, чи звук. «Поняття слова» є ідеальним, бо у реальному світі воно матеріально не існує: якщо матеріально зникають навчені люди, то зникає «система навчання», зникає саме СЛОВО (від СЛОВА можуть залишитись лише форми його матеріального відбитку).

Дві форми слова є різними і можуть існувати окремо: слова на незнайомій мові мають вигляд матеріального відбитку без понятійної форми; однакова понятійна форма слова може мати різні матеріальні відбитки.

Ідеальний, духовний світ окремої людини *стає матеріальним* для інших людей, якщо ця людина вступає у взаємодію з оточенням і стає для нього учителем через матеріальну форму СЛОВА та вкладеного у СЛОВО ідеального сенсу. *Кожна людина може матеріалізувати власні ідеї* і може створити власний матеріальний світ, який через СЛОВО у двох формах стає зрозумілим оточуючим його людям. Недарма про кожну знайому нам людину ми маємо власне уявлення (добре або погане).

Не існуючий у природі ідеальний абстрактний об'єкт «СЛОВО» відображує (*моделює*) об'єкти, явища та процеси матеріального реального навколишнього середовища і уявного ідеального внутрішнього середовища абстрактних знань людини.

Модель – це зручне для практичних чи абстрактних цілей відтворення у визначеному масштабі з деякою точністю усіх або деяких сторін (форми, будови, процесів, властивостей тощо) реального чи уявного (ідеального, абстрактного) об'єкта, який зветься *оригіналом*.

Моделювання – це дослідження реального та абстрактного середовища за допомогою моделей для використання у практичних чи абстрактних цілях.

Відтворення процесів в оригіналі у вигляді математичного опису дає нам *математичну модель*. Якщо модель та оригінал фізично подібні і відрізняються лише масштабами форми та внутрішніх процесів, то розглядається *фізична модель*. Якщо модель описується словами, то маємо *описову модель*.

Для людини все «зовнішнє середовище» має вигляд сукупності ідеальних моделей. «*Зовнішнє середовище*» людини можна розглядати у як два оточення: матеріальне (реальне; приклад – ландшафт) та ідеальне (у природі не існуюче, створене людською уявою, абстрактне; приклад – математика). Ці два оточення людина у своїй свідомості розглядає як єдину сукупність різних моделей, які мають матеріальну форму (ідеальна «математика» відображується СЛОВОМ у двох його формах).

За допомогою СЛІВ кожна людина у своїй свідомості створює моделі двох ідеальних взаємопов'язаних світів, розрив між якими призводить до втрати знань для людства:

– моделі структури («устрою», «конструкції») «зовнішнього середовища» стосовно об'єктів та явищ реального та абстрактного світу (людина бачить ландшафт, окремі його елементи, але не розуміє сенсу у цьому «малюнку»: взаємодії об'єктів, процесів, загроз. Цьому потрібно навчитись);

– моделі ідеального поняття та вкладеного сенсу у значенні, розвитку процесів та взаємодії між елементами моделі структури «зовнішнього середовища» (створюється системою навчання).

Теорія – це упорядковане та узагальнене знання, спрямоване на пояснення якого-небудь явища об'єктивної чи абстрактної дійсності і представлене у вигляді учення про властивості, алгоритми, правила протікання та розвитку процесів в моделі цього явища.

Метод – теоретично пояснений загальний шлях, підхід, алгоритм, послідовність дій при практичному чи теоретичному застосуванні теорії, система регуляторних принципів в будь-якій ділянці діяльності (матеріальній чи абстрактній, практичній чи теоретичній), визначеним чином упорядкована матеріальна чи абстрактна загальна діяльність. Фактично метод відрізняється від теорії лише функціонально: метод зв'язує теорію з практикою.

Методика – більш вузьке застосування методу для розв'язання часткової задачі; частковий захід упорядкованої діяльності.

Методи, методики – це теоретично пояснені алгоритми практичного використання теорії. Вони з'єднують теорію з практикою.

Знання людини засновані на «словах – моделях» обох світів людини: матеріального та духовного (абстрактного).

Тому усі знання людини є ідеальними абстрактними моделями – не існуючими в матеріальному світі формами. Знання з фізики, хімії, математики, фахових дисциплін у вигляді ідей за допомогою СЛОВА матеріалізуються і стають матеріальною рушійною силою людства.

Поведінкою окремої людини, фірми та держави **керують ЗНАННЯ, основою яких є СЛОВО.** Але тоді виходить, що всім людством в його практичній та духовній діяльності «керує» ідеальна, духовна, сенсова, понятійна форма СЛОВА, як першоджерело усіх знань. Нами керує **ДУХОВНА ФОРМА СЛОВА**, яка хоча і «підтверджена практикою» та «має матеріальну форму навчених нейронів», але реально не існує у природі, а є моделлю, має ідеальний вміст.

Як розв'язати це протиріччя між матеріальним та ідеальним? Протиріччя тут немає, це – «єдність протилежностей», яку потрібно спокійно усвідомити. Але людина часто використовувала цю суперечливість на власну матеріальну користь. Тут бажано керуватись

мудрістю предків, для яких відповідь на це протиріччя лежить дещо у іншій площині: якщо ти людина, то не потрібно з невиправданою жорстокістю карати людей за їх гріхи та помилки; потрібно керуватись терпимістю, толерантністю, доброзичливістю до людини. Бо усі вади людини притаманні й тобі. Бо ти – «сам такий» і по справедливості повинен з такою ж жорстокістю карати сам себе.

Системи штучного інтелекту автоматизують розумову діяльність людини і розглядають в основному об'єкт у вигляді найбільш примитивної «моделі» з вимогою розпізнавати лише обраний тип образу у навколишньому середовищі та з виміром лише деяких обраних ознак образу.

Але при створенні програм для складних систем штучного інтелекту часто використовують СЛОВО, яке є класифікатором, що займає центральне місце у розпізнаванні образів людиною. Тут під СЛОВОМ потрібно розуміти також і будь-який КОД та штучну мову. У випадку використання слів велика увага приділяється їх однозначності для людини і ЕОМ та несуперечливості використаних термінів. Тому для розпізнавання складних образів використовуються *методи онтології та тезаурусу*, які займаються лінгвістичною обробкою СЛОВА.

13.2.2. Онтологія

При створенні і поширенні знань виникає *ряд проблем*:

1. Необхідно, щоб нові знання *розуміла не лише людина, але й ЕОМ*.
2. На планеті існує більше 2700 мов, що ускладнює обмін знаннями між людьми. Необхідно впорядкувати існуючий *хаос* і створити однозначні визначення для *різних людських мов та ЕОМ*.
3. Вимагається *систематичне поповнення знань* із-за зростаючого рівня діяльності.
4. *Матеріальні втрати людства*, пов'язані проблемами однозначності мов зростають із-за швидкого зростання чисельності населення.
5. Знання потрібно концентрувати *в одному місці*.

Історично ці проблеми частково вирішувались переходом в науковій сфері *на одну мову* (англійську, французьку, німецьку, іспанську, латинську, грецьку). Зараз найпоширенішою є англійська мова.

Онтологія – розділ *філософії* зі спрямуванням до всеохоплюючої та детальної формалізації деякої *ділянки* знань за допомогою концептуальної схеми (концептів, класів). Але не можна охопити неохватне. Тому *існує багато онтологій на одній мові*, які відносяться до різних ділянок знань.

Зараз *виділяють три типи онтологій*:

1. *Метаонтологію*, яка розглядає загальні питання організації онтології.

2. *Онтології предметної області* які описують знання окремої предметної області (інформатики, математики, хімії і т. д.).

3. *Онтології додатків*, які описують поняття, залежні від розв'язуваної задачі.

Мета онтології: спільна однозначна мова між людьми та ЕОМ (розуміння термінів, обмін інформацією, упорядкування знань, полегшення пошуку знань) з напрямками «не вигадувати колесо», припинити хаос в термінології.

Вимоги до знань: знання повинні бути адекватними, однозначними, несуперечливими, представлятись в машинній формі, давати можливість міркувати і породжувати нові знання як людині так і машині.

Онтологія складається з термінів, організованих у *таксономію* (ієрархію) їх визначень, атрибутів, а також пов'язаних з ними аксіом і правил висновку. Важливе місце в онтології посідає взаємні зв'язки термінів і *відображення сенсу об'єкта* в семантичних метаописах.

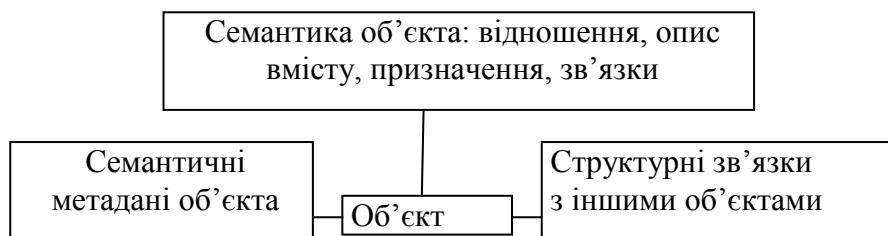


Рис. 13.2.1. Відображення сенсу об'єкта семантикою.

Під семантичними метаданими об'єкта розуміється множина понять онтологій, які мають відношення до i -го об'єкта. З кожним поняттям пов'язаний свій *ваговий коефіцієнт* K_i .

Всі моделі онтологій містять:

- *концепти* (поняття, класи, сутності, категорії), їх *властивості* (слоти, атрибути, ролі) та *відносини*;
- *метаописи структури*;
- системними метаданими (рис. 13.2.2).
- Термін *екземпляр* використовуються для представлення елемента даного концепту. *Онтологія разом з множиною окремих екземплярів складає базу знань*.

Сьогодні онтології широко застосовуються в інформатиці.

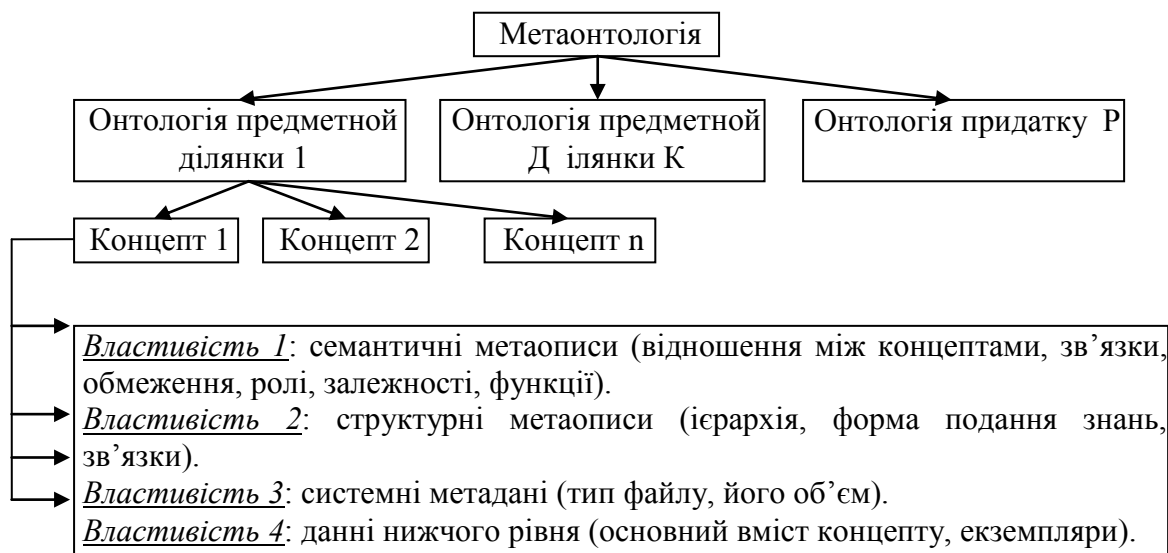


Рис. 13.2.2. Структура онтології

13.2.3. Тезаурус – словник представлення знань

Тезаурус: (1) *словник*, у якому максимально повно представлені слова мови з прикладами їх вживання в тексті (у повному обсязі здійснимо лише для мертвих мов); (2) *словник*, в якому слова, які стосуються якої-небудь області знання, розташовані за *тематичним принципом* і показані *семантичні відносини* (родовидові, синонімічні та ін.) між лексичними одиницями. В інформаційно-пошукових тезаурусах лексичні одиниці тексту замінюються дескрипторами.

Довідка. Семіотика під *текстом* розуміє послідовність будь-яких знаків, що задовольняє вимогам зовнішньої зв'язності, внутрішньої осмисленості, можливості своєчасного сприйняття, здійснення необхідних умов комунікації та ін

Лексикографія займається теорією і практикою складання словників. Теорія лексикографії класифікує словники на мовні та енциклопедичні.

Безпосередніми попередниками словникового опису знань були *глоси*, тобто пояснення значень окремих слів на полях древніх текстів. *Збірники глосів (гларії)* були першими лексикографічними описами.

Незалежно від того, яку мову програмування вибере програміст, перед ним постане необхідність використання *природної мови (ПМ)* для *опису програмного забезпечення (ПЗ)*.

Таку можливість надає *тезаурус* у вигляді *предметно-* або *проблемно-орієнтованого словника*, який містить в собі *термінологію предметної області*, *визначає відносини між поняттями і містить типізовані зв'язки*.

Однією з фундаментальних робіт з систематизації та класифікації словників є робота Караулова Ю. Н. [36].

У цій роботі запропоновано *систему лексикографічних параметрів словників* у вигляді так званої «*матриці покриття*», в якій виділено такі елементи:

1. *Перелік параметрів* – стовпці матриці (виділяється 67 лексикографічних параметрів), що включають варіювання значень параметрів від мови до мови і спосіб їх завдання і глибини розкриття.

2. *Значення параметрів* – ряди матриці, що об'єднують певні параметри – слова і показують їх глибину в конкретній мові.

3. «*Поле можливих рішень*» є сукупністю заповнених комірок на перетині рядів і стовпців «матриці покриття» і дає матеріал для комбінування, екстраполяції і виявлення інших комбінацій параметрів при конструюванні нових лексикографічних об'єктів.

Така система параметрів, є *необхідною і достатньою для опису всіх існуючих сьогодні словників*. Більше того, комбінування відповідних параметрів дає можливість конструювати нові типи словників.

Якщо в *тезаурус* в якості *семантичних відносин* обрати *логічні операції*, тоді тезаурус буде являти собою деяку логічну модель певної розумової діяльності. Він складається з контрольованого, але змінюваного словника термінів, між якими вказані смислові зв'язки. Такий словник вичерпним чином покриває деяку специфічну галузь знань, логічно впорядкований по систематичному і алфавітному принципам і відображає смислові відносини як ієрархічного типу (наприклад, «родо-видові»), так і неієрархічного типу. Якщо з тезаурусу зняти термінологічні та галузеві обмеження, то принципової різниці між загальномовним та *інформаційно-пошуковим тезаурусом* немає [36]. Зафіксований набір *мовних одиниць* інформаційно-пошукового тезаурусу і правила їх співвіднесення (*обчислення відносин*) може бути прирівняний до деякої *інформаційно-пошукової мови, до формально-мовної системи*.

Тезаурус є формально-мовною системою:

1. Для побудови довідкових систем (індексування).

2. Для наступного ефективного доступу до інформації.

13.3. Кластеризація на основі теорії грубих множин

Теорія грубих множин (ТГМ) запропонована З. Паулаком (1991) [184; 200], Теорія грубих множин дозволяє обробляти неявні масиви неупорядкованих даних і на основі такої обробки отримувати нові знання [20; 21; 39].

Ознаки та знання отримуються із різних джерел та представляються у вигляді *таблиці*, у яких рядки відповідають об'єктам грубої множини, а колонки – ознакам об'єктів. У комірці таблиці відображується значення ознаки об'єкта.

Теорія грубих множин, за рахунок підсумку, різниці, добутку, розширення та ін, дозволяє виконати звичайну *структуризацію образів* початкової множини X і формувати групи підмножин об'єктів $X = \{X_1, X_2, \dots, X_j, \dots, X_n\}$, де $j = 1, \dots, n$ – порядковий номер підмножини; X_j – підмножина (сімейство) об'єктів з *власними ознаками* при $X_j \subset X$; $X_j \neq \emptyset$; $X_i \cap X_j$.

Але додатково теорія також дозволяє виділити *семантично представлені нові знання*, які взаємно пов'язують групи об'єктів, для чого використовує дані науки «*психосемантики*» з метою виявлення глибинних семантичних зв'язків і захованих категорій структури знань.

Алгоритми «*психосемантики*», як правило, охоплюють процедури, які засновані на *парному порівнянні схожості ознак об'єктів* на деякій шкалі «0...9» чи «-2...+2», внаслідок чого формуються концепти більш високого рівня абстракцій.

14. Деякі напрямки систем розпізнавання образів

14.1. Систематика

14.1.1. Визначення простих, великих і складних систем

Систематика – це наука про структуру, статику, динаміку і проектування складних систем. Основний внесок у створення нового наукового напрямку (систематики) вніс Ю. І. Шемакін [82].

Система – це сукупність взаємопов'язаних компонент (об'єктів), яка характеризується зв'язками і взаємодією її підсистем і об'єктів, єдністю сутності, як єдиного об'єкту із загальною поведінкою та стійкістю у часі [82].

Системи діляться на:

- *прості* (незначна кількість компонент),
- *великі* (велика кількість компонент з *однозначною* реакцією на сигнали),

– *складні* (велика кількість компонент з *неоднозначною*, *непередбачуваною*, *не розкритою* реакцією на сигнали). Передісторія складних систем не визначає їх реакцію. Складна система характеризується наявністю біологічних, соціально-економічних, природних та статистичних компонентів; великою кількістю ступенів свободи; нелінійністю і різноманітністю способів взаємодії між підсистемами і об'єктами.

Система в кожний момент часу може характеризуватися *станом*, який описується сукупністю таких сутностей, як *об'єкти, класи, ситуації, дії*.

14.1.2. Специфікація і створення чотирьох словників (понять, відносин, процесів і дій)

Специфікація програми – це опис задачі, яку повинна вирішувати програма. Тому виникає *проблема формування лексики абстрактної системи*, треба підібрати точні поняття, які є основним інструментарієм опису складних систем.

З цією метою *створюються чотири основні словника: **понять, відносин, процесів і дій (операцій)***.

Суттєва відмінність організації понятійних засобів при розробці специфікації від завдань програмування полягає у тому, що опис складної системи вимагає від понять виразності і гнучкості, але, на відміну від завдань програмування, не є обов'язковою їх безпосередня машинна реалізація.

Опис складної системи може бути дуже великим за обсягом. Тому

сучасні методи опису складних систем практично повністю орієнтовані на використання інструментальних засобів у вигляді пакетів прикладних програм, що функціонують на ЕОМ.

14.1.3. Штучна мова і загальний опис складних систем

Опис складних систем вимагає застосування адекватного математичного і алгоритмічного апарата.

1. *Формальна мова (наприклад, UML – Unified Modeling Language)* є одним із засобів, що забезпечує можливість формулювання і передачі великого інформаційного змісту. Нижче використовуються інструменти мови UML (стандарт з 1997 р.) з метою демонстрації особливостей системантики при моделюванні поведінки складної системи.

2. *Використовуються два рівня розгляду проблеми подання знань з використанням логіко-алгебраїчних методів* аналізу складних систем:

– *Теорія системи на зовнішньому, синтаксичному рівні* будується як абстрактний опис системи. *Моделі верхнього рівня (моделі прецедентів)* дозволяють відобразити взаємодію систем із зовнішнім оточенням (середовищем). Основним принципом опису складних систем є принцип абстрагування, за яким всі другорядні деталі не ураховуються. Поведінку складної системи можна представити як зовнішню траєкторію переходів з одного стану в інший в просторі обмеженої кількості станів системи. Основну увагу зосереджено на зовнішніх взаємних зв'язках між станами системи, без явного визначення внутрішніх операцій чи елементів структур станів.

– *Моделі сценаріїв внутрішніх функціональних процесів систем.* На внутрішньому, більш конкретному рівні, сценарії описують динаміку (поведінку) системи у вигляді виконання окремих задач незалежно від ієрархії між елементами системи і ураховують внутрішні і зовнішні зв'язки системи. У сценарії процесу використовують послідовне і паралельне виконання задач. Задачі миттєво переходять із одного стану в інший.

3. *Методи аналізу* використовують універсальні штучні мови моделювання систем типу UML, теорію категорій, універсальні алгебри, математичну логіку. Моделювання системи здійснюється за *об'єктно-орієнтованою методологією* (при створенні моделей стану, поведінки, взаємозв'язку, повідомлень, подій, об'єктів, процесів, станів і цілей). Модель системи має *ієрархічну (пірамідальну) структуру управління* з внутрішніми зв'язками між підсистемами і об'єктами. Розглядаються ситуації не лише з *детермінованою цільовою поведінкою*, але і ситуації, коли поведінка системи не визначається чіткою метою і задається у вигляді системи загальних цінностей.

4. *Для визначення станів* систем використовується *реляційна алгебра*

RelAl. Стани представляють у вигляді **сукупності поійменованих відносин (таблиць)**, а **функціонування системи – у вигляді виконання набору операцій над цими відносинами**. Рядками таблиці є елементи відносин, а атрибутам відповідають стовпці таблиці. Реляційна алгебра дозволяє моделювати динаміку складних систем, використовуючи операції над відносинами (над таблицями): «1:1», «1: M», «M: 1», «N: M». Операції над відносинами розуміються як алгебраїчні операції.

Використовуються і **інші залежності між вхідними аргументами і виходом**. Наприклад, застосовуються **диференційні рівняння**. При цьому вважається, що отримана локальна інформація дозволить зрозуміти глобальну поведінку системи в часі і в просторі. Але для складних систем локальна інформація може не відповідати зовнішній поведінці системи, яка залежить від рішень ієрархічно вищого логістичного рівня (ринку, планів накопичення, політичних рішень). Складна система розглядається також у вигляді «чорного ящика», для якого внутрішній механізм перетворення входу у вихід залишається невідомим. Такий підхід достатній для аналізу багатьох фізичних і технічних систем, але його використання для опису кібернетичних, біологічних чи соціально-економічних систем часто є малоефективним, а іноді і неможливим внаслідок залежності виходу від комбінаторного складу входу, а також від зміни соціальних і ринкових умов.

На відміну від традиційних математичних моделей, які спираються на безперервні структури і відповідні методи аналізу, в нових моделях основну роль відіграють дискретні структури сучасної алгебри, що дозволяє представити опис системи Σ у вигляді п'ятірки (при цьому передбачається, що множини X , Y і Q кінцеві):

$$\Sigma = \langle X, Y, Q, \mu, \lambda \rangle,$$

де $\bullet X$ – множина допустимих входів,

- $\bullet Y$ – множина спостережуваних виходів, $\bullet Q$ – множина станів системи,
- $\bullet \mu: Q \times X \rightarrow Q$ – функція переходу, $\bullet \lambda: Q \times X \rightarrow Q$ – функція виходу.

Наразі в математиці сформувався **теорія категорій і функторів**, яка дозволяє оперувати структурованими множинами і встановлювати відповідності між класами об'єктів з різною аксіоматикою і розглядати стан складної системи у вигляді універсуму понять і переходів між ними. Існує стан системи і стан об'єктів. Зміна стану (подія) відбувається миттєво.

Система визначається структурою і поведінкою. Під **структурою** розуміється інваріантна у часі фіксація зв'язків між елементами системи (статичний опис системи), а під **поведінкою** системи розуміється її функціонування у часі (динамічний опис системи).

Поняття представлення знань про системи *shmP* можна представити у вигляді трійки ознак

$$shmP = \langle D, H, V \rangle,$$

де $D = \{D_i\}$, $i=1,2,\dots, n$ – множина імен **диференційних ознак**, які

використовуються як характеристики змісту поняття; $H = \{H_j\}$, $j=1,2,\dots, m$ – множина імен *характеристичних ознак*, які дозволяють відрізнити сутності, що відносяться до одного й того самого поняття; $V = \{V_k\}$, $k=1,2,\dots, L$ – множина імен *валентних ознак* (звичайно вони бінарні), які забезпечують зв'язок між різними поняттями шляхом виконання ролі *якісних властивостей* по відношенню до інших ознак.

Ознаки понять – це стійкі риси, які внутрішньо властиві поняттю і характеризуються *ім'ям і значенням* певного типу (наприклад, ознака «ВІК» разом зі значенням «20 років» утворює найменування ознаки «ВІК 20 років»). *Ім'я ознаки разом з його значенням утворює повне найменування відповідної ознаки*. Можна виділити кілька типів значень ознак: *логічні, числові, символічні* та ін.

Концепт поняття це правила, за якими дана сутність може бути віднесена до деякої *множини однотипних сутностей*.

Екстенціонал поняття (позначається $extP$) – це сукупність усіх його припустимих *денотатів*. Так, щоб описати екстенціонал *поняття АВТОМОБІЛЬ*, слід розглянути клас всіх автомобілів.

Інтенціонал поняття (позначається $intP$) – це семантика або зміст чи концепт поняття.

Зв'язки між різними категоріями, які використовуються для опису понять в логіці, семіотиці та інформатики, представлені на рис. 14.1.1.

Поняття у систему вводять у вигляді трійки (рис. 14.1.1)

$$P = \langle intP, shmP, extP \rangle.$$

Поняття стає надзвичайно зручним засобом, який дозволяє шляхом використання інтенціоналу висловити семантичні відносини для деякого фрагмента системи, а також за допомогою схеми забезпечити перехід до більш детального опису та подання інформації в моделі її реалізації.

Складна система розглядається як сукупність *метакласів, класів, їх екземплярів та об'єктів*.

Стани задач не допускають переходів під час свого виконання. При об'єктно-орієнтованому підході поведінка системи задається не у вигляді процесів, що розвиваються у часі, а у вигляді посилки *повідомлень, викликаних станом самих об'єктів*, які, у свою чергу, викликають в інших об'єктах наступні події.



Рис. 14.1.1. Опис поняття

Об'єкти приймають повідомлення, інтерпретують їх, активізують відповідні методи, змінюють свій стан, посиляють власні повідомлення і т. д.

Якщо зміна відбувається у часі, то її можна розглядати у вигляді пари подій, одна з яких відбувається до зміни стану, а друга – після зміни стану. Якщо потрібно урахувати умову, то використовують пару подій з передумовою та з результатом завершення умови. Одночасні події (аналогічно поняттям) можуть бути об'єднані в класи подій, між якими можуть бути встановлені відносини узагальнення і агрегації.

14.1.4. Методологічні особливості опису складної системи

Основними методологічними прийомами системного аналізу є:

1. **Мультимоделність**: ніяка одна модель складної системи не є достатньою для її адекватного відображення [82].

Приклад типів моделей, які використовуються в мові UML: найбільш важливі прецеденти; сценарій функціонального процесу; взаємодія об'єктів; логічні моделі; класи; об'єкти; переходи станів; сценарії, динамічної поведінки системи; реалізації; розміщення.

2. **Асоціація**, яка використовує механізми типізації, узагальнення, агрегації та асоціації. При цьому створення абстрактних понять асоціацій здійснюється в процесі об'єднання понять більш низького рівня на основі їх подібності (включаючи математичну подібність).

Тому що для кожного абстрагування можливе як підвищення так і зниження рівня абстракції, ми маємо справу з парами: типізація – конкретизація, узагальнення – спеціалізація, агрегація – декомпозиція, асоціація – індивідуалізація.

Асоціація типу 1:1 є взаємно однозначне відображенням сутностей. Прикладом служить зв'язок між поняттями ОСОБИСТІТЬ і ПАСПОРТ. Ці поняття пов'язані між собою бінарною асоціацією з ім'ям *Має*.

Асоціація 1: M виникає в тих випадках, коли одна з сутностей дозволяє ідентифікувати кілька інших сутностей. Наприклад, між поняттями ВІДДІЛ й СЛУЖБОВЕЦЬ може бути визначена асоціація типу 1: M. Вони пов'язані між собою бінарною асоціацією *Працює*.

N-арная асоціація містить окремий випадок бінарної асоціації, коли кожен екземпляр асоціації являє собою упорядкований набір (кортеж), що містить n екземплярів відповідних понять.

N-арная асоціація пов'язує ставленням кілька понять, при цьому одне поняття може брати участь в асоціації більш ніж один раз.

Асоціація виду N: M виникає в тих випадках, коли в обох напрямках асоціації можна вказати більше однієї сутності, з якої можливий зв'язок, тобто зв'язки в асоціації в обох напрямках не унікальні. Прикладом такої асоціації служить зв'язок між ВИРОБАМИ і ПОСТАЧАЛЬНИКАМИ.

3. **Абстракція**, яка виконується заміною реального об'єкта *його ім'ям* з описом співвідношень між операціями *через імена*. При такому підході тип конструюється шляхом виконання *деякого набору операцій над вихідними множинами, званими основами*.

Абстракція ігнорує другорядні внутрішні подробиці і ураховує тільки ті знання, які дозволяють охопити проблему цілком. Це дозволяє розбити систему на підсистеми, кожна з яких, як правило, простіше початкової системи.

Абстракція може здійснювати перехід від загальних ознак до окремих індивідуальних понять (такий процес називають також *індивідуалізацією*).

Типізація – це процес групування об'єктів у певні групи (типи). Тип об'єкта визначає деяка множина (клас) однорідних сутностей при ігноруванні їхніх індивідуальних відмінностей. Абстракція типізації – лише окремий випадок абстракції узагальнення.

Фундаментальними видами абстракцій є *агрегація, узагальнення, модульність*.

4. **Модульність**. Кожному такому *модулю* повинна відповідати смислова одиниця у вигляді *поняття*.

Процес виділення багаторівневої сукупності понять, яка абстрагується від конкретної реалізації типів – це процес узагальнення понять, в результаті якого породжується ієрархія типів, що описують дану систему.

5. **Узагальнення понять** – це така форма породження нового поняття R на основі одного або декількох подібних понять P і Q, коли породжене поняття R зберігає загальні ознаки й операції вихідних понять P і Q, але ігнорує їх більш тонкі розпізнавальні ознаки. Так, із сукупності понять СТИЛ, КРИСЛО, ШКАФ можна утворити родові поняття МЕБЛІ, якщо не розглядати їх індивідуальні ознаки. Процес узагальнення допускає можливість багаторазового його застосування, в результаті чого виникає ієрархія узагальнень.

6. **Спеціалізація** (обмеження понять) – процес, коли початкове поняття може ділитися на декілька вузких понять. При спеціалізації до ознак початкового родового поняття додаються індивідуальні ознаки і операції: до родового ЛЮДИНА додаються ознаки ВОДІЙ ВАНТАЖНОЇ МАШИНИ, ВОДІЙ ТОЙОТИ, ВОДІЙ ТРАКТОРА і т. д. Абстракцію узагальнення часто пов'язують з відношенням «є-деякий (is-a)» між видовим і родовим поняттями.

7. **Узагальнення класів** – це відношення між класом (батьком) і підкласом (нащадком). Підклас володіє всіма властивостями і поведінкою класу, а також має власні властивості і поведінку, які можуть бути відсутніми у суперкласа. Тому нащадок може бути підставлений замість батька, але не навпаки. Часто, хоча і не завжди, у нащадків є власні властивості та операції, крім тих, що існують у батька.

Клас може мати одного або декількох батьків або не мати їх зовсім.

8. «*Агрегація*» – це абстракція, яка описується відношенням «є-частина»; вона має місце між декількома класами в тому випадку, якщо один з класів являє собою сутність, яка включає в себе в якості складових частин інші сутності. Відношення агрегації, що відрізняє її від інших видів абстракцій, описує функціональний зв'язок між компонентами і об'єктом типу *N*: *1*. Наприклад, об'єкт АВТОМОБІЛЬ включає в якості своїх складових частин ДВИГУН, ШАСІ, КАБІНУ, КУЗОВ.

«*Поняття*» *теж є агрегатом*, що складається зі своїх компонентів.

Можна також розглядати й *агрегат властивостей*. Семантична відміна *агрегації властивостей* від *агрегації понять* полягає в тому, що властивості не визначаються як незалежні поняття і, отже, можуть бути представлені в описі системи тільки в тому випадку, коли визначені відповідні базові поняття.

Поряд з агрегацією використовується:

– *Декомпозиція* – процес, протилежний агрегації. При декомпозиції початкове поняття розчленовується на ряд незалежних компонент, кожна з яких має лише частку ознак початкового поняття.

– *Узагальнення*, на відміну від «агрегації» навпаки, *ігнорує* ті властивості і операції власних компонент, які відрізняються між собою, і залишають тільки такі ознаки і поведінку, які є загальними для всіх цих компонент. Якщо для агрегації характерно спадкування властивостей і операцій понять «знизу вгору», то для узагальнення, навпаки, «зверху вниз».

– «*Композиція*» – окремий випадок відносини «агрегації».

14.1.5. Моделювання динаміки складних систем

Для моделювання поведінки *в мові UML* можуть використовуватися одразу кілька канонічних моделей (прецедентів, сценаріїв діяльності, переходів станів і послідовності взаємодій об'єктів), кожна з яких фіксує увагу на окремому аспекті *динаміки системи*.

Кожний прецедент визначає деякий *сценарій (набір дій)*, що чиниться системою при діалозі з дійовою особою. Наприклад, *діюча особа Клієнт* в рамках заданої моделі ініціює різні прецеденти: Зняти гроші з рахунку, Переказати гроші, Покласти гроші на рахунок, Показати баланс, Змінити ідентифікатор, Здійснити оплату. *Прецедент* – це типова дія системи у відповідь на події, ініційовані ззовні з використанням *ролей*. *Дійова особа «Банківський службовець»* може ініціювати прецедент «Змінити ідентифікатор клієнта». *Кожна роль* може вступати у взаємодію з одним або декількома прецедентами.

У процесі опису системи *кожен прецедент моделюється незалежно від інших*. Якщо для реалізації прецеденту використовують об'єкти, які *взаємодіють з іншими прецедентами*, та між прецедентами виникають

неявні залежності.

Модель прецедентів показує, що дійові особи мають власні цілі і ініціюють видимі ззовні функції системи. Модель дозволяє візуалізувати прецеденти окремо від їх реалізації.

Зазвичай складна система містить кілька десятків прецедентів, кожен з яких може розгортатися в *кількох десятках сценаріїв*. Будь-який прецедент має *основні і допоміжні сценарії*.

14.2. Маскування та демаскування образів

В природі а також у людській діяльності основні ознаки об'єктів іноді маскують, змінюють, заховують, викривлюють. Для замаскованих і незамаскованих реальних об'єктів та їх моделей (макетів) виникає необхідність створення:

- інформаційних баз розпізнавання і дешифрування ознак маскованих і немаскованих об'єктів;
- систем розпізнавання реальних об'єктів та їх макетів (моделей).
- Нижче описуються деякі напрямки маскування та демаскування образів.

Визначення образу по дотичних ознаках. Іноді образ неможливо розпізнати із-за його ретельного маскування і заховування. Прикладом може бути: розвідка покладів корисних копалин, води під Землею при неможливості безпосереднього спостереження об'єкта; складність процесу класифікації (визначення хвороби людини, діагностики технічного об'єкта); велика загальна кількість ознак і їх комбінаторний характер спостереження; переклад з однієї мови на іншу внаслідок багатозначності; заховування негативних рис людиною (заховування схильності до брехні; жорстокості; можливості двозначної поведінки тощо); нечіткість даних.

У таких випадках визначення образу намагаються отримати по дотичним ознакам, за аналізом по збільшеній кількості ознак, по ознакам різної фізичної природи.

В військових операціях засоби маскування образів є найбільш розповсюдженими і охоплюють:

- *обмеження доступу* до текстової інформації та до реальних об'єктів;
- *маскування* найбільш характерних ознак реальних об'єктів. Наприклад, одяг солдат та окраска техніки не повинні їх демаскувати; використовують маскувальні властивості місцевості та маскувальні сітки; створюють замасковані укриття для людей і техніки;
- *моделювання* хибних споруд (моделей будинків, шляхів, тунелів, мостів, хибних позицій, окопів) та імітація елементів бойової діяльності. Моделі роблять на фабриках та з підручних матеріалів (землі, снігу, прогумованої тканини, металевої сітки, дощок тощо), піноутворювачами, з

імітацією та без імітації функціонування. Під час бойових дій у Перській затоці (1991 рік) армія Іраку застосувала *надувні макети техніки*, по яких й були нанесені основні удари авіації США, Великобританії, Франції. Тому потрібно знати маскуючі і демаскуючі ознаки реальних об'єктів та їх моделей;

– *дезінформацію* стосовно реального стану речей;

– *розділ уваги* між багатьма об'єктами, серед яких у дійсності використовуються лише декілька об'єктів (наприклад, пускових шахт для ракет виконують значно більше, ніж самих ракет).

Текстову інформацію додатково оберігають використанням методів *криптографії*, яка маскує (шифрує) вміст інформації, та використанням методів *стеганографії*, яка маскує сам факт передачі інформації.

Шифрування використовується для приховування змісту інформації методами *криптографії*. Для шифрування застосовують криптографічний ключ у вигляді певної кількості символів, довільно заданих програмістом.

При симетричному шифруванні використовують один секретний ключ, який служить і для шифрування і для розшифрування потоку або блоків інформації фіксованого розміру. Перевагами симетричного шифрування є велика пропускна здатність, а до недоліків відноситься необхідність використання великої кількості ключів з частою їх заміною (і з можливістю їх перехоплення).

При асиметричному шифруванні використовують два відмінних ключа (*відкритий та секретний*), які пов'язані математичними залежностями. *Відкритий* ключ відомий всім, хто шифрує інформацію, а *секретний* – відомий лише тому, хто має право розшифрувати інформацію. Ключі використовують у парі: інформація шифрується відкритим ключем, а розшифровується по математичним залежностям секретним ключем та навпаки. Інформацію, зашифровану за допомогою секретного ключа, розшифровують лише за допомогою відповідного відкритого ключа. Це означає, що немає потреби передавати ключ, бо відкритий ключ є публічним (відкритим). Але асиметричне шифрування є значно повільнішим, тому воно часто застосовується разом з симетричним шифруванням, яке працює на 3-4 порядки швидше.

Стеганографія відрізняється від криптографії тим, що вона приховує сам факт передачі інформації і не викликає підозр, в той час як криптографія приховує вміст повідомлення. Звичайно стеганографію використовують разом з криптографією.

При використанні стеганографії повідомлення має вигляд чогось безвинного: зображення, музики, тексту статті. *Демокрит* першим в 440 р. до н.е. використав стеганографію: він записав повідомлення про підготовку нападу на Грецію на дерев'яній дошці, яка потім була покрита воском (такі дошки використовувались для письма).

До стеганографії відноситься:

- використання *симпатических чорнил*, які є невидимими, але проявляються при нагріві, освітленні чи під дією хімічних реактивів. Наприклад, текст, написаний молоком, стає видимим при нагріві над полум'ям свічки;
- *мікроточки* – мікроскопічні фотознімки, які вклеювались у текст листа;
- *трафарети*, які при накладанні на текст виділяють потрібні слова або літери;
- *запис на поверхні яйця*, який при варці яйця зникає з поверхні, зате зберігається з внутрішньої сторони шкарлупи;
- «*жаргонні слова*», які мають інший сенс;
- наразі частіше використовують *заховування інформації в текстових, графічних та аудіо файлах* з використанням спеціального програмного забезпечення.

Приклади комп'ютерної стеганографії:

- частка полів розширень файлів не використовується і по замовчанню заповнюється нулями. Ця «нульова частка» використовується для запису власних даних;
 - при використанні гнучких дисків інформація записується в не використані частки диска;
 - використання особливостей фону (наприклад, напис чорним шрифтом на чорному фоні);
 - використання в файлі вільних кластерів, які не зайняти файлом;
 - в зображенні заховують інше зображення шляхом використання усіх бітів кольорового компонента дійсного зображення, крім двох останніх бітів;
 - заховування або введення в цифрові об'єкти додаткової інформації.
- Отримані викривлення людина не здатна фіксувати на фоні існуючих шумів;
- використання особливостей протоколів передачі даних (заховування інформації в полях заголовків, які не використовуються; затримка, або псування пакетів інформації, які ігноруються приймачем (наприклад, пакет голосового повідомлення вилучається, замінюється секретною інформацією і затримується. Затриманий пакет по замовчанню відкидається приймачем, але якщо приймач знає про захований зв'язок, то відкидання пакета замінюється вилученням захованої інформації);
 - в контейнері повідомлення два останніх біти використовуються для захованого повідомлення (такий алгоритм використовують при відсутності шуму);
 - передача інформації за рахунок правки коефіцієнтів дискретного косінусного перетворення. Для цього все зображення розділяється на блоки по 8x8 пікселів, і кожний з блоків використовується для шифрування одного пікселя захованого повідомлення.

При розслідуванні злочинів (фінансових, матеріальних, технічних,

військових, проти особи) часто зустрічаються факти маскувannya, знищення та псування ознак образу злочину та злочинника. Деякі методи злочинності вимагають від злочинника маскувannya під невинну, беззахисну, порядну людину. Кредитні картки, електронні гроші і можливість їх витрат через Інтернет зробили інтернет-шахрайство одним з найбільш поширених злочинів.

Особливості відмивання грошей на стадії їх розміщення в банках.

Здобуті злочинним шляхом великі суми грошей злочинці намагаються «відмити» шляхом розміщення їх у банки.

Для цього їх маскують:

1. *Спосіб реструктуризації.* Виконують у великій кількості різні операції під малі суми, які не вимагають реєстрації та ідентифікації вкладника: вкладення депозитів; переказів чеків на пред'явника; багаторазові заміни банкнот дрібних номіналів на більші в сумі.

2. *Змішування «брудних» грошей з легальним доходом, особливо коли власник легального доходом є також власником «брудних» грошей.* Досягається це також шляхом підкупу банківських службовців і навіть членів правління банків. З цією метою використовують також іммігрантів з азіатських і африканських країн, а також держав Східної Європи, якікладають невеликі суми грошей на один рахунок, а ці гроші згодом пересилаються до країн походження іммігрантів.

3. *Купівля і продаж валюти, заміна ушкоджених грошей.*

4. *Договір про обмін валют, наприклад, значної суми доларів США на швейцарські франки.*

5. *Приховування різних предметів і грошей у банківських сейфах.*

6. *Використання грошових переказів, «як плоти за послуги».*

Загрози Інтернет. Зловмисники користуються тим, що Інтернет не має територіальних меж і не є чієюсь власністю. Вони розміщують віруси і інші шкідливі програми на веб-ресурсах, «маскують» їх під корисне і безкоштовне програмне забезпечення, які у дійсності спрямовані:

– на розкрадання (отримання номерів банківських рахунків, кредитних карт, паролей);

– на виведення обчислювальних систем з ладу або підлив ділової репутації, порушення працездатності окремих компаній, державних установ та фірм (розповсюдження вірусів та троянських програм; розголошення конфіденційної інформації).

Віруси розповсюджуються через Інтернет, дискети, CD-диски, флеш-карти, електронну пошту. Вони заражають інші файли, додаючи в них свій код з метою отримати управління при запуску зараженого файлу, можуть пошкодити дані комп'ютера і здатні саморозповсюджуватись на інші носії інформації або комп'ютери мережі. Віруси діляться на кілька типів.

Черви (Worms) є найбільш шкідливими і найпоширенішими типами вірусів, які використовують для розповсюдження недоліки («діри») опера-

ційної системи. Вони обчислюють мережеві адреси інших комп'ютерів або адреси поштових клієнтів і розсилають по ним свої копії.

Троянські програми (Trojans) знищують інформацію на дисках, виводять систему до «зависання», крадуть конфіденційну інформацію тощо. Але вони не здатні *самостійно* розмножуватись і розповсюджуються під виглядом «корисного» безкоштовного програмного забезпечення.

Деякі віруси мають ознаки двох-трьох вірусів.

Програми-реклами (Adware) – без відома користувача включаються у програмне забезпечення з метою реклами. Звичайно розповсюджуються через «безкоштовне» програмне забезпечення.

Програми-шпигуни (Spyware) збирають відомості про окремо взятого користувача або організацію без їх відома (відстежування дій користувача, копіювання даних).

Програми-жарту (Jokes) часто виводять повідомлення про форматування диска (насправді форматування не відбувається), виявляють віруси в незаражених файлах тощо.

Програми-маскувальники (Rootkit) – утиліти, які модифікують операційну систему комп'ютера і замінюють основні її функції, щоб приховати свою власну присутність і дії зловмисника.

Потенційно небезпечні додатки (Riskware) не мають *власної* шкідливої функції, але ними можуть скористатися зловмисники як допоміжними складовими шкідливої програми. До таких програм відносять деякі утиліти видаленого адміністрування, програми перемикування розкладки клавіатури, утиліти зупинки процесів тощо.

До *шкідливих* відносяться вбудовані в браузер програми, які перенаправляють трафік: при запиті одного сайту вони відкривають зовсім інший.

Кажуть також: *файлово-бутові, пакетні, мережеві, WinWord-віруси, Windows-віруси, OS/2-віруси, Novell NetWare-віруси, BIOS-віруси, CD-ROM-віруси* тощо.

Перелічити всі способи маскування і демаскування неможливо, тому що їх тисячі.

Ознаки зараження комп'ютера: поява непередбачених повідомлень, зображення або звукових сигналів; псування або зникнення деяких команд; довільний запуск програм; часті зависання і збої в роботі комп'ютера; уповільнення роботи комп'ютера; неможливість завантаження операційної системи; зникнення файлів і каталогів або спотворення їх вмісту; часте звернення до жорсткого диска (часто мигає лампочка на системному блоці); веб-браузер зависає або поводить ся несподіваним чином (наприклад, вікно програми є закритим). У багатьох випадків наявність непрямих симптомів викликана збоєм у апаратному або програмному забезпеченні.

До основних методів боротьби з вірусами відносяться:

– використання антивірусних програм (однієї або кількох);

- копіювання і окреме збереження основних даних;
- виявлення демаскуючих ознак об'єктів/моделей;
- використання розпізнавальних систем з більшою кількістю ознак та з фізично різними системами датчиків ознак;
- метод перебору при виконанні розшифровки.

Ознаки, які частково відрізняються від еталонів, можуть бути відновлені за допомогою асоціативних методів, які використовуються в нейронних мережах.

Спам (небажана пошта рекламного характеру, яка розповсюджується по Інтернет) не є джерелом прямої загрози, але наносить значні фінансові втрати із-за втрати робочого часу отримувачем інформації. Ця набридлива небажана кореспонденція збільшує навантаження на поштові сервери і засмічує поштову скриньку користувача. Спам може змусити користувача перейти на якийсь інтернет-ресурс, що є вже елементом небезпеки. *Боротьба зі спамом* спрямована проти інтернет-шахрайства та розповсюдження шкідливих програм і ведеться окремими програмами.

Відоме маскування цінних повідомлень (графічних робіт, рисунків, тексту) при пересиланні їх по мережі Інтернет. При розмаскуванні файлу електричної схеми (по окремо одержаним ключам по Інтернет – пошті) спочатку визначають координати розташування елементів схеми, а потім лінії їх електричних з'єднань.

14.3. Сукупність алгоритмів та формул

Бустінг (англ. boosting – поліпшення) – це процедура послідовного будування алгоритмів, кожний з яких *поступово компенсує недоліки попередніх алгоритмів*. Бустінг став популярним із-за простоти, універсальності і гнучкості (можливості модифікації під задану задачу). Бустінг має властивість функції Фурьє: чим більше алгоритмів, тим точніше результат апроксимації функції.

Колектив вирішальних правил [34] – це обрання однієї альтернативи із їх сукупності. Для підвищення ефективності роботи системи РО можна об'єднати у «колективи» різні за характером алгоритми $R_i(X)$, ($i = 1, m$), що приймають рішення $S_i = R_i(X)$ у ситуації X , і які класифікують об'єкти на основі вирішальних правил, прийнятих в теорії колективних рішень.

Множина алгоритмів $\{R\} = \{R_1, R_2, \dots, R_1, \dots, R_m\}$ зветься **колективом алгоритмів рішення задачі (колективом вирішальних правил)**, яке визначене на множині рішень

$$S = F(S_1, S_2, \dots, S_m, X),$$

де R_i звать **членами колектива**, S_i – рішенням i -го члена колектива у ситуації X , S – **колективним рішенням**. F – функція, що узагальнює усі інди-

відуальні рішення.

Коллективне рішення допомагає розв'язувати різні задачі:

1. В задачах розпізнавання образів X є зображенням, а рішення S – клас, до якого належить це зображення.

2. В задачі управління – об'єднання ситуації навколишнього середовища та мети управління (наприклад – самоуправління, яке приводить об'єкт до визначеного стану).

3. В задачах прогнозування X є початковим станом об'єкта, а S – прогнозованим станом.

Одним з напрямків методу колектива вирішальних правил є використання вагових коефіцієнтів для кожного вирішального правила R_i у залежності від зображення, що розпізнається. Ваговий коефіцієнт для вирішального правила R_i дорівнює

$$\mu_i(X) = \begin{cases} 1, & \text{якщо } X \in B_i \\ 0, & \text{якщо } X \notin B_i \end{cases},$$

де B_i – ділянка, на яку розповсюджується вплив вирішального правила R_i .

Вагові коефіцієнти вирішальних правил обираються таким чином, щоб для всіх можливих значень X

$$\sum_{i=1}^L \mu_i(X) = 1$$

Рішення колективу визначається рішенням того вирішального правила R_i , до якого належить ділянка зображення об'єкта X . Рішення колективу визначається у два етапи: на першому етапі визначається до якої ділянки належить об'єкт, а на другому етапі використовується вирішальне правило, компетентність якого найбільша у знайдений ділянці. Рішення цього правила є «диктаторським», тобто ототожнюється з рішенням всього «колективу». Головним у цьому питанні є навчання системи РО розпізнаванню областей компетентності, які можна визначати на основі ймовірносних властивостей правил колективу; при самонавчанні вважати, що однаковим компактним ділянкам відповідають однакові правила.

У нейронній мережі кожний вхід підсумовуючого вихідного нейрону має свій ваговий коефіцієнт і може розглядатись як член колективу, який набуває вагові коефіцієнти з урахуванням особливостей колективного рішення. Рішення на виході перцептрону розглядається як середньозважене рішення колективу.

Розглянутий нами раніше метод *групового врахування аргументів (МГВА)* використовує алгоритм переробки інформації у природі: висівається насіння (початкові елементи формул); вирощують рослини (складні успадковані комбінації формул), і лише з оптимальної частини рослин (з формул з потрібними властивостями) обирається оптимальна кількість насіння; далі знову висівають це обране насіння – і етап повторюється. Через

кілька поколінь селекція зупиняється на оптимальному результаті, бо якщо і далі продовжувати селекцію, то настає «інцухт» – виродження рослин [34].

14.4. Фрактальна геометрія

Фрактальна геометрія [85] – це один з напрямків *комп'ютерної графіки*, який математичною формулою задає незмінну форму малюнка-фрактала і «малює» не пікселями, а фракталами, а зміною заданих даних впливає на вигляд зображення і кольори.

Фрактали – це однотипні геометричні фігури, які використовуються при створенні графічних образів, малюнків, картин фрактальної анімації та картин фрактального супроводу музики. Наразі створені алгоритми синтезу коефіцієнтів фракталів, які дозволяють *відтворити копію будь-якої картини із заданою точністю*.

Перші фрактали вивчались у XIX сторіччі (з'явилися у 70-х роках): функція Больцмана, функція Вейерштраса, множина Кантора. Комп'ютери дозволили візуалізувати ці фігури.

Прикладом базового елемента фрактальної графіки є *рівносторонній трикутник*: по середнім точкам сторін будуються трикутники – нащадки, які повторюють властивості всього об'єкта. Наслідування можна продовжувати до нескінченності.

Приклад рекурсивного алгоритму будівництва фрактальної кривої: задають на площині довільну ламану лінію з кінцевим числом відрізків, яка зветься *генератором*. Далі заміняють кожний відрізок *генератором* зі змінним масштабом, а в отриманій ламаній знову кожний відрізок замінюємо *генератором* і т. д.

Фрактальну графіку використовують для відображення живої природи (кровоносної системи та бронхів людини; квітів, рослин, дерев, кущів, коралів тощо) та неживої природи (блискавки, сніжинок, кристалів, хмар, гір, поверхні моря тощо), в радіотехніці, інформатиці (стиснення зображень, комп'ютерна графіка, децентралізовані мережі), в економіці, в рекламі, в науці (за допомогою фракталів моделюють турбулентний рух рідини, полум'я, хмари) тощо.

14.5. Теорія інформаційної грануляції

Під гранулами розуміємо багатовимірні об'єкти, позначені у вигляді геометричних фігур різних форм з різним кольором, які розглядають в одно-, дво- чи трьох-вимірному просторі. Цим грануляція відрізняється від кластеризації, яка «не покидає» n -вимірний простір. По теорії грануляції, заснованою Л. Заде, грануляційні розрахунки описуються засобами обмеженої природної мови на основі узагальнених обмежень і в десятки разів скорочують отриману інформацію [206; 207].

Грануляція призначена для поліпшення розуміння і *наочного уявлен-*

ня багатовимірних образів, з неупорядкованими, неформалізованими неметричними даними та даними з великим об'ємом інформації.

Гранули – це частки цілого, кластер, множини (лінгвістичні, нечіткі, наближені, вкладені), підзадачі, одиниця знання. Гранули можуть об'єднуватись у більші гранули та роз'єднуватись на дрібніші гранули. Гранули – це окремі об'єкти n -вимірному простору, які розглядаються в одно-, двох- чи трьох-вимірному просторі як об'єкти з різною геометричною формою та з різним кольором. Внаслідок грануляції обов'язково втрачається частка початкової інформації із-за переходу у простір меншого виміру. Тому розроблений ряд алгоритмів, які дозволяють будувати оптимальні покриття гранулами, включаючи використання гранул розміром в один елемент, при якому початкова інформація зберігається цілком.

В роботі [16] розглядається *екологічний моніторинг поверхні Землі* на основі теорії інформаційної грануляції Л. Заде з використанням багатовимірних даних. При цьому невизначені об'єкти, наприклад, планктон, частіше описуються власними границями, які можуть бути невідомої форми.

Даний аналіз дозволяє оцінити площу або об'єм, зайняті отриманими гранулами. Визначені в результаті аналізу уніфіковані алгебро-геометричні моделі можуть бути застосовані для будування більш складних інтелектуальних систем аналізу *багатовимірних даних*. Вони перетворюють гранульовані моделі «чорного ящика» у «прозорий» для користувача «скляний ящик», бо замінюють багатовимірні дані гранул в одно-, двох-, трьох-вимірний об'єкт із заданим кольором та геометричною формою з чіткими границями розповсюдження.

14.6. Метод «найближчого сусіда»

Системи міркування на основі аналогічних випадків. Ідея алгоритму вкрай проста. Для того щоб зробити прогноз на майбутнє або вибрати правильне рішення, ці системи знаходять у минулому близькі аналоги наявної ситуації і вибирають ту ж відповідь, що була для них правильною. Тому цей метод ще називають методом *«найближчого сусіда»*. Системи міркування на основі аналогічних випадків показують добрі результати в найрізноманітніших задачах.

Їх теоретичною основою може вважатись метод кластеризації: якщо вхідний невідомий вектор ознак має майже співпадаючі ознаки з ознаками найближчого сусіда, то з великою долею ймовірності невідомий вектор та його найближчий сусід належать до одного кластера (до одного класу). Тому невідомому вектору привласнюють клас його найближчого сусіда.

Список використаної літератури

1. Айзерман М. А., Браверман Э. М., Розоноэр Л. И. Метод потенциалных функций в теории обучения машин. – М. : Наука, 1970. – 384 с.
2. Акіменко В. В., Загородній Ю. В. Проектування СППР на основі нечіткої логіки : навчально-методичний посібник. – К. : Вид-во КНУ, 2007. – 94 с.
3. Акулич И. Л. Математическое программирование в примерах и задачах. – СПб.: Изд-во «Лань», 2009. – 352 с.
4. Анісімов А. В., Глибовець М. М., Марченко О. О., Кисенко В. К. Метод обчислення семантичної близькості для слів природної мови // Наукові записки Наукма. – Том 125. Комп'ютерні науки. – Київ : НАУКМА, 2011. – С. 8–13.
5. Архангельский В. И., Богаенко И. Н., Грабовский Г.Г., Рюмшин Н. А. Нейронные сети в системах автоматизации. – К. : Техніка, 1999. – 234 с.
6. Атабеков Г. И. Теоретические основы электротехники. В 3-х томах. – Ч. 1. Линейные электрические цепи. – М. : Энергия, 1978. – 592 с.
7. Байєсова мережа [Електронний ресурс]. – Режим доступу : https://uk.wikipedia.org/wiki/Байєсова_мережа (14.03.2016).
8. Балакирев В. С, Володин В. М., Цирлин А. М. Оптимальное управление процессами химической технологии (экстремальные задачи в АСУ). – М. : Химия, 1978. – 383 с.
9. Беляцкий Н. П. Интеллектуальная техника менеджмента. – Минск, 2001. – 320 с.
10. Белозерський Л. А. Розпізнавання образів / Л. А. Белозерський [Електронний ресурс] – Режим доступу : <http://dodge.thebest-on.com/rozpznannja-obrazv>.
11. Бідюк П. І., Коршевніюк Л. О. Методика побудови ймовірнісних мережних моделей. // Наукові праці, 2011. – Вип. 148. – Том 160. – »Комп'ютерні технології». – С. 6–14.
12. Болянский Е. В., Долотов А. И. Многослойная самообучающаяся спайк-нейронная сеть // Бионика интеллекта. – 2007. – № 2(67). – С. 21–26.
13. Бокс Дж., Дженкинс Г. Анализ временных рядов, прогноз и управление : Пер. с англ. // Под ред. В. Ф. Писаренко. – М. : Мир, 1974, кн. 1. – 406 с.
14. Бондарев Б. Н., Аде Ф. Г. Искусственный интеллект, Учебное пособие для вузов. – Севастополь : Изд-во СевНТУ, 2002. – 615 с.
15. Бурлаченко І. С., Єндзевич О. В. Дослідження алгоритмів пошуку асоціативних правил мультиагентною інформаційною системою на основі VLE-пристроїв // Наукові праці : наук.-метод. журн. – Вип. 254. – Т. 266. Комп'ютерні технології. – Миколаїв : Вид-во ЧДУ ім. Петра Могили, 2015. – С. 88–96.

16. Бутенков С. А., Бутенков Д. С., Кривша В. В. Гранулированные вычисления в системах интеллектуального анализа пространственных данных // Интеллектуальный анализ информации. Труды V Международной конференции, 2005. – С. 108–117.
17. Бэстенс Д. Э., Ван Ден Берг В. М. Нейронные сети и финансовые рынки. Принятие решений в торговых операциях. – М. : ТВП Научное издательство, 1997. – 235 с.
18. Васильев В. И. Распознающие системы. Справочник. – Киев : Наукова думка, 1983. – 422 с.
19. Восприятие и распознавание образов / Пер. С фр. А. В. Серединского ; под ред. Г. П. Катуса. – М. : Машиностроение, 1989. – 272 с.
20. Гаврилова Т. А. Извлечение и структурирование знаний для экспертных систем. – М. : Радио и связь, 1992. – 186 с.
21. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. – СПб. : Питер, 2001. – 384 с.
22. Галушкин А. И. Синтез многослойных систем распознавания образов. – М. : Энергия, 1974. – 368 с.
23. Гладунський В. Н. Логічні таблиці для виявлення альтернатив у процесі вироблення управлінських рішень. – Львів : Національний університет «Львівська політехніка». 04.09.2015. [Електронний ресурс]. – Режим доступу : http://vlp.com.ua/files/05_27.pdf.
24. Головкин В. А. Нейронные сети: обучение, организация и применение / В. А. Головкин. – М. : ИПРЖР, 2001. – 122 с.
25. Гордеев Б. Н. Математическая модель полиметрических измерений // Наукові праці : наук.-метод. журн. – Миколаїв : Вид-ва ЧДУ ім. П. Могили.– 2009. – Т. 111, Вип. 98. Техногенна безпека. – С. 176–182.
26. Горелик А. Л., Скрипкин В. А. Методы распознавания / Учебное пособие для студентов вузов. – М. : Высшая школа, 1989. – 232 с.
27. Гриньов Б. В., Кириченко І. К. Аналітична геометрія. – Харків : Гімназія, 2008. – 340 с.
28. Гужва А. Г. Нейроинформатика / И. Г. Персианцев, Ю. С. Шугай // Сборник научных трудов, ч. 2. – М. : МИФИ, 2008. – С. 216–225.
29. Гумен О. М. Моделювання проєктивних n-просторів багатопараметричних технічних систем. Автореферат дисертації на здобуття наукового ступеня доктора технічних наук. Спеціальність 05.01.01 – Прикладна геометрія, інженерна графіка. – Мелітополь : Таврійський державний агротехнологічний університет, 2011. – 39 с.
30. Дюрбан Б., Оделл П. Кластерный анализ. – М. : Статистика, 1977. – 128 с.
31. Ежов А. А., Шумский С. А. Нейрокомпьютеринг и его применение в экономике и бизнесе / А. А. Ежов, С. А. Шумский. – М., 1998. – 222 с.
32. Жуков Ю. Д. Теория полиметрических измерений // Инновации в судостроении и океанотехнике : материалы Первой Междунар. науч.-техн. конф. – Николаев : НУК, 2010. – С. 387–389.

33. Заде Л. Понятие лингвистической переменной и ее применение к принятию решений. – М. : Мир, 1976. – 165 с.
34. Ивахненко А. Г. Системы эвристической самоорганизации в технической кибернетике. – К. : Техніка, 1971. – 372 с.
35. Калан Роберт. Основные концепции нейронных сетей : пер. с англ. – М. : Издательский дом «Вильямс», 2003. – 288 с.
36. Караулов Ю. Н. Лингвистическое конструирование и тезаурус литературного языка. – М. : Наука, 1981. – 367 с.
37. Кичак В. М. Синтез частотно-імпульсних елементів цифрової техніки : монографія / В. М. Кичак. – Вінниця : УНІВЕРСУМ-Вінниця, 2005. – 266 с.
38. Классификация и кластер. Под ред. Дж. Вэн Райзина. – М. : Мир, 1980 – 390 с.
39. Коваленко И. И., Пономаренко Т. В., Швед А. В. Представление знаний на основе теории грубых множеств. – Николаев : «Илион», 2012. – 52 с.
40. Колесницький О. К., Левченко Ю. В., Колесницька Г. О. Модель нейрона. Патент на корисну модель UA 76924, G06G 7/60. Бюл. № 2, 25.01.2013. – 2 с.
41. Комарцова Л. Г., Максимов А. В. Нейрокомпьютеры : учеб. пособие для вузов. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2004. – 399 с.
42. Кононюк А. Ю. Нейроні мережі і генетичні алгоритми. – К. : «Корнійчук», 2008. – 446 с.
43. Кременецький Г. М. Використання ПЛІС для побудови кластерних нейронних мереж // Проблеми інформатизації та управління, 3(27)'2009. – С. 79–84.
44. Кохонен Т. Ассоциативные запоминающие устройства / Т. Кохонен. – М. : Мир, 1982.
45. Кулинич О. І. Теорія статистики. – К. : Вища школа, 1992. – 135 с.
46. Кутковецкий В. Я. О повышении объёма полезной информации в технических средствах обучения // Труды НКИ, Основы научной организации учебного процесса. – Николаев : НКИ, 1977. – Вып. 119. – С. 21–22.
47. Кутковецкий В. Я. Обобщенные методы переключающих функций и их применение для расчета электромагнитных процессов в вентильных цепях. Специальности: 05.09.12 – полупроводниковые преобразователи электроэнергии; 05.09.5 – теоретическая электротехника. Автореферат диссертации на соискание ученой степени д.т.н. Киев : Институт электродинамики АН Украины, 1992. – 30 с.
48. Кутковецкий В. Я. Розпізнавання образів / В. Я. Кутковецкий. – Миколаїв : Вид-во МДГУ ім. П. Могили, 2003. – 196 с.
49. Кутковецкий В. Я. Штучний інтелект / В. Я. Кутковецкий. – Миколаїв : Вид-во Європейський університет, Миколаївська філія, 2005. – Ч. I. – 158 с. – Ч. II. – 106 с. – Ч. III. – 172 с.

50. Кутковецький В. Я. Теоретичні основи мереж потоків // Наукові праці : наук.-метод. журн. – Миколаїв : Вид-во ЧДУ ім. Петра Могили. – Вип. 148, Том 160. – Комп'ютерні технології. – 2011. – С. 173–183.
51. Кутковецький В. Я. Слово як ідеальна та матеріальна основа інформатики // Наукові праці : наук.-метод. журн. – Миколаїв : Вид-во ЧДУ ім. Петра Могили, 2013. – Вип. 203, Т. 215. – Педагогіка. – С. 58–62.
52. Кутковецький В. Я. Закони потоків мереж при максимальному збагаченні у часі // Наукові праці : наук.-метод. журн. – Миколаїв : Вид-во ЧДУ ім. Петра Могили, 2013. – Вип. 201, Том 213. – Комп'ютерні технології. – 2013. – С. 73–77.
53. Кутковецький В. Я. Метод ієрархічних потоків // Наукові праці : наук.-метод. журн. – Миколаїв : Вид-во ЧДУ ім. Петра Могили, 2014. – Вип. 225, Т. 237. – Комп'ютерні технології. – С. 49–57.
54. Кутковецький В. Я. Аналітична геометрія в n-вимірних тілесних кутах // Наукові праці : наук.-метод. журн. – Миколаїв : Вид-во ЧДУ ім. Петра Могили, 2015. – Вип. 254, Т. 266. – Комп'ютерні технології. – С. 30–41.
55. Кутковецький В. Я., Турти М. В. Нейрон «Троянда1». Патент України на корисну модель G06N 3/00, № 96451, 10.02.2015, Бюл. № 3. – 5 с.
56. Кутковецький В. Я., Турти М. В. Нейрон «Троянда2». Патент України на корисну модель G06N 3/00, № 96452, 10.02.2015, Бюл. № 3. – 5 с.
57. Кутковецький В. Я., Турти М. В. Нейрон «Троянда3». Патент України на корисну модель G06N 3/00, № 96453, 10.02.2015, Бюл. № 3. – 6 с.
58. Кутковецький В. Я., Турти М. В. Нейрон «Троянда4». Патент України на корисну модель G06N 3/00, № 96454, 10.02.2015, Бюл. № 3. – 6 с.
59. Кутковецький В. Я., Турти М. В. Нейрон «Т_XOR». Патент України на корисну модель G06N 3/00, № 96450, 10.02.2015, Бюл. № 3. – 4 с.
60. Кутковецький В. Я., Турти М. В. Спосіб навчання нейронної мережі. Патент України на корисну модель G06N 7/00, № 96456, 10.02.2015, Бюл. № 3. – 7 с.
61. Кутковецький В. Я., Турти М. В. Кластерний спосіб навчання нейронної мережі. Патент України на корисну модель G06N 7/00, № 96457, 10.02.2015, Бюл. № 3. – 8 с.
62. Кутковецький В. Я., Турти М. В. Нечітка нейрона мережа. Патент України на корисну модель G06N 3/02, G06N 7/02, № 97763, 10.04.2015, Бюл. № 7. – 3 с.
63. Кутковецький В. Я., Турти М. В. Нейрон «Т_AND_0». Патент України на корисну модель G06N 3/00, № 97526, 10.04.2015, Бюл. № 7. – 3 с.
64. Кутковецький В. Я., Турти М. В. Нейрона мережа «MAX_0». Патент України на корисну модель G06N 3/02, G06N 7/02, № 97940, 10.04.2015, Бюл. № 7. – 3 с.
65. Кутковецький В. Я., Турти М. В. Нейрона мережа «MAX_1». Патент України на корисну модель G06N 3/00, № 97941, 10.04.2015, Бюл. № 7. – 3 с.

66. Кутковецький В. Я., Турти М. В. Нейрона мережа «MIN_0». Патент України на корисну модель G06N 3/02, G06N 7/02, № 98308, 27.04.2015, Бюл. № 8. – 5 с.
67. Кутковецький В. Я., Турти М. В. Нейрона мережа «MIN_1». Патент України на корисну модель G06N 3/02, G06N 7/02, № 98309, 27.04.2015, Бюл. № 8. – 5 с.
68. Кутковецький В. Я., Турти М. В. Нейрон «MAX_2». Патент України на корисну модель G11C 11/54 (2006.01), № 104873, 25.02.2016. – Бюл. № 4. – 2 с.
69. Кутковецький В. Я., Турти М. В. Нейрон «MIN_2». Патент України на корисну модель G11C 11/54 (2006.01), № 104874, 25.02.2016. – Бюл. № 4. – 2 с.
70. Кутковецький В. Я., Турти М. В. Нейрон «T_MAX». Патент України на корисну модель G11C 11/54 (2006.01), G06N 3/00, № 104875, 25.02.2016. – Бюл. № 4. – 2 с.
71. Кутковецький В. Я., Турти М. В. Нейрон «T_MIN». Патент України на корисну модель G06N 3/00, № 104876, 25.02.2016. – Бюл. № 4. – 3 с.
72. Кутковецький В. Я., Турти М. В. Нейрон «T_AND_1». Патент України на корисну модель G06N 3/00, № 99112, 25.05.2015. – Бюл. № 10. – 4 с.
73. Кутковецький В. Я., Турти М. В. Нейрон «T_OR_0». Патент України на корисну модель G06N 3/00, № 99113, 25.05.2015. – Бюл. № 10. – 5 с.
74. Кутковецький В. Я., Турти М. В. Нейрон «T_OR_1». Патент України на корисну модель G06N 3/00, № 102141, 26.10.2015. – Бюл. № 20. – 5 с.
75. Кутковецький В. Я., Турти М. В. Нейронна мережа для стиснення інформації «Код n:m:n» Патент України на корисну модель G06N 3/00, № 104871, 25.02.2016. – Бюл. № 4. – 4 с.
76. Кутковецький В. Я., Турти М. В. Нейронна мережа «Миколаїв». Патент України на корисну модель G06N 3/00, № 104872, 25.02.2016. – Бюл. № 4. – 4 с.
77. Кутковецький В. Я., Турти М. В. Нейрон. Патент України на корисну модель G06N 3/00, № 106289, 25.04.2016. – Бюл. № 8. – 3 с.
78. Кутковецький В. Я., Турти М. В. Спосіб самонавчання класифікуючої нейронної мережі Патент України на корисну модель G06N 3/00, № 108187, 11.07.2016. – Бюл. № 13. – 14 с.
79. Кутковецький В. Я., Турти М. В. Спосіб навчання стохастичної нейронної мережі. Патент України на корисну модель G06N 3/00 7/02, № 109297, 25.08.2016. – Бюл. № 16. – 15 с.
80. Кутковецький В. Я. Статика і динаміка мереж відносних ймовірнісних та детермінованих потоків // Наукові праці : наук.-метод. журн. – Миколаїв : Вид-во ЧДУ ім. Петра Могили, 2016. – Вип. 257. Т. 269. – Педагогіка. – С. 66–73.
81. Кутковецький В. Я. Удосконалення класичного наївного байєсового класифікатора багатовимірних об'єктів // Всеукраїнська науково-

- методична конференція «Могилянські читання 2016». Збірник тез. Том 5. Технічні науки. Комп'ютерні науки. 14–18 листопада 2016. – Миколаїв, 2016. – С. 74–75.
82. Ломако Е. И. Математические и понятийные средства системантики. – М. : «Системная Энциклопедия», 2008. – 110 с.
83. Ляковська С. Є. Геометричне моделювання багатопараметричних систем способом епюра n-простору / Автореферат дис. Спеціальність 05.01.01 – Прикладна геометрія, інженерна графіка. – Мелітополь : Міністерство аграрної політики України, Таврійський державний агротехнологічний університет, 2010. – 25 с.
84. Мандель И. Д. Кластерный анализ. – М. : Финансы и статистика, 1988. – 176 с.
85. Мандельброт Б. Б. Фрактальная геометрия природы. – М. : Институт компьютерных исследований, 2002. – 656 с.
86. Мартин Р., Николлс Дж., Валлас Б., Фукс П. От нейрона к мозгу. – М. : УРСС, 2003. – 676 с.
87. Матвейкин В. Г., Фролов С. В. Использование байесовского подхода в обучении нейронных сетей – М. : ФГБОУ ВПО «МГТУ им. Н. Э. Баумана», 2005. – 52 с.
88. Матвійчик А. В. Аналіз та прогнозування розвитку фінансово-економічних систем із використанням теорії нечіткої логіки : монографія. –К. : Центр навчальної літератури, 2005. – 206 с.
89. Мелихов А. Н., Бернштейн Л. С., Коровин С. Я. Ситуационные советующие системы с нечеткой логикой. – М. : Наука, 1990. – 272 с.
90. Месарович М., Такахара Я. Общая теория систем. Математические основы. – М. : Мир, 1978. – 310 с.
91. Налимов В. В., Чернова Н. А. Статистические методы планирования экстремальных экспериментов. – М. : Наука, 1965. – 340 с.
92. Нейлор К. Как построить свою экспертную систему. – М. : Энергоатомиздат, 1991. – 280 с.
93. Общая теория статистики : учебник / А. Я. Боярский, Л. Л. Викторова, А. М. Гольдберг и др. ; Под ред. А. М. Гольдберга, В. С. Козлова. – М. : Финансы и статистика, 1985. – 367 с.
94. Омату С. Нейроуправление и его приложения / Пер. с англ. // М. : ИПРЖР, 2000. – 272 с.
95. Плошко Б. Г., Елисеева И. И. История статистики : учеб. пособие. – М. : Финансы и статистика, 1990. – 295 с.
96. Порческу Г. В. Оперативные единицы восприятия речи на разных лингвистических уровнях // Филологические науки в России и за рубежом : материалы междунар. науч. конф. (г. Санкт-Петербург, февраль 2012 г.). – СПб.: Реноме, 2012. – С. 165–168.
97. Поспелов Д. А. Большие системы. Ситуационное управление / Математика. Кибернетика. – М. : Знание, 1975. – № 01. – 66 с.

98. Рогальский Ф. Б., Курилович Я. Е., Цокурено А. А. Математические методы анализа экономических систем. Книга 1. Теоретические основы. – Киев : Наукова думка, 2001. – 436 с.
99. Родионова Л. Н., Руднева Ю. Р., Пашин С. Т. Методы расчета цены на новый товар // Маркетинг в России и за рубежом, 2001. – № 2. – С. 34–45.
100. Романишин Ю. М., Петрицька С. Р. Побудова заданої послідовності імпульсів на основі багатотходового спайк-нейрона. – Львів : Національний університет «Львівська політехніка», 2013. – С. 66–72 // [Електронний ресурс]. – Режим доступу : <http://ena.lp.edu.ua:8080/bitstream/ntb/23914/1/10-66-72.pdf>.
101. Ротштейн А. П. Интеллектуальные технологии идентификации: нечеткие множества, генетические алгоритмы, нейронные сети. – Винница : УНІВЕРСУМ-Вінниця, 1999. – 320 с.
102. Рощина А. Н., Трохимчук Р. Н. Распознавание конкретных изолированных изображений с помощью модифицированного метода зондов // Проблемы управления и информатики, 2000. – № 1. – С. 78–82.
103. Руденко О. Г., Бодяньский С. В. Штучні нейронні мережі. – Харків : ТОВ «Компанія СМІТ», 2006. – 404 с.
104. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. – М. : Горячая линия – Телеком, 2004. – 452 с.
105. Рязузов Н. Н. Общая теория статистики. – М. : Финансы и статистика, 1984. – 4 изд. – 290 с.
106. Саати Т., Кернс К. Аналитическое планирование. Организация систем. – М. : Радио и связь, 1991. – 224 с.
107. Саати Т. Л. Принятие решений при зависимостях и обратных связях: Аналитические сети. – М. : Либроком, 2011. – 360 с.
108. Сергиенко И. В. Математические модели и методы решения задач дискретной оптимизации. – К. : Наукова думка, 1988. – 65 с.
109. Синявский О. Ю. Обучение спайковых нейронных сетей на основе минимизации их энтропийных характеристик в задачах анализа, запоминания и адаптивной обработки пространственно-временной информации тема / Дисс. на ученую степень канд. техн. наук. Специальность 05.13.17: Теоретические основы информатики. – М. : 2011. – 149 с.
110. Статистика : підручник / А. В. Головач, А. М. Єріна, О. В. Козирев та ін. : За ред. А. В. Головача, А. М. Єріної, О. В. Козирева. – К. : Вища школа, 1993. – 623 с.
111. Статистика підприємництва : навч. посібник / П. Г. Вашків, П. І. Пастер, В. П. Сторожук, Є. І. Ткач. – К. : Слобожанщина, 1999. – 600 с.
112. Статистический словарь / Гл. ред. М. А. Королёва. – М. : Финансы и статистика, 1989. – 623 с.
113. Струченков В. И. Методы оптимизации. Основы теории, задачи, обучающие компьютерные программы : Учебное пособие В. И. Струченков. – М. : Издательство «Экзамен», 2005. – 256 с.

114. Стюарт Р. Искусственный интеллект: современный подход / Пер. с англ. – М. : Издательский дом «Вильямс», 2006. – 2 изд – 548 с.
115. Турти М. В. Теорія однозначних нечітких систем та нейронні мережі : монографія. – Миколаїв : Вид-во Європейський університет, Миколаївська філія, 2007. – Ч. I. – 140 с. – Ч. II. – 114 с.
116. Турти М. В. Теорія однозначних нечітких систем та нейронні мережі // Проблеми автоматики та електрообладнання транспортних засобів : Матеріали Всеукраїнської науково-технічної конференції з міжнародною участю. – Миколаїв : НУК, 2007. – С. 229–237.
117. Фор А. Восприятие и распознавание образов. – М. : Машиностроение, 1989. – 272 с.
118. Фукунага К. Введение в статистическую теорию распознавания образов / Пер. с англ. – М. : Наука, 1979. – 367 с.
119. Хетагуров Я. А. Проектирование автоматизированных систем обработки информации и управления. – М. : Высш. шк., 2006. – 223 с.
120. Царегородцев В. Г. Перспективы распараллеливания программ нейросетевого анализа и обработки данных / В. Г. Царегородцев // Материалы III Всерос. конф. «Математика, информатика, управление – 2004». – Иркутск, 2004. – С. 11–117.
121. Чернодуб А. Н. Обучение нейроэмуляторов с использованием псевдо-регуляризации для метода нейроуправления с эталонной моделью // Искусственный Интеллект, 2012. – № 4. – С. 602–614.
122. Шмиголь Н. М. Застосування теорії розпізнавання для аналізу та прогнозування доходів і фінансових результатів підприємств // Ефективна економіка. – Дніпропетровськ : Дніпропетровський державний аграрно-економічний університет, 2014. – № 9. [Електронний ресурс]. – Режим доступу : <http://www.economy.nauka.com.ua/?op=1&z=3313>.
123. Штерн А. С. Перцептивный аспект речевой деятельности: (экспериментальное исследование). – СПб.: Издательство С.-Петербургского университета, 1992. – 236 с.
124. Штовба С. Д. Идентификация нелинейных зависимостей с помощью нечеткого логического вывода в системе MATLAB // Exponenta Pro, 2, 2003. – С. 9–15.
125. Юрчак И. Ю. Синтез и использование нейронных сетей с проективными и упорядоченными латеральными связями / И. Ю. Юрчак. – Львов : ПТК, 1999. – 16 с.
126. Ackley D. H., Hinton G. E. and Sejnowski T. J. A learning algorithm for Boltzmann machine // Cognitive Science, 1985 – № 9. – P. 147–169.
127. Adrian E. D. The impulses produced by sensory nerve endings // J. Physiology Lond, 1926. – Vol. 61. – P. 49–72.
128. Almeida L. B. Learning Rule for Asynchronous Perceptron with Feedback in a Combinatorial Environment // Proc. of the First IEEE Int. Conf. on Neural Networks, USA, San Diego, 1987. – Vol. 2. – P. 609–618.

129. AL-Zawaidah F. H., Jbara Y. H., Marwan A. A. An improved algorithm for mining association rules in large databases. // *World of Computer Science and Information Technology Journal*, 2011. – № 1(7). – P. 311–316.
130. Anderson J. A. Cognitive and Psychological Computation with Neural Models // *IEEE Trans. on Syst., Man, and Cybern*, 1983. – 13. –P. 799–815.
131. Azam, Farooq. Biologically Inspired Modular Neural Networks. PhD Dissertation, Virginia Tech. May, 2000 scholar.lib.vt.edu/theses/available/etd-06092000-12150028/unrestricted/etd.pdf. – 138 p.
132. Baier V. Motion Perception with Recurrent Self-Organizing Maps Based Models // *Proc. of IJCNN'05*. – Monreal, Canada, 2005. – July 31 – Aug. 4. – P. 1182–1186.
133. Banzhatt W., Haken H. A Network for Recognition and Classification of Continues Pattern // *Neural Networks*, 1988. – № 1. – P. 6.
134. Beaver W. Financial Ratios as Predictors of Failure // *Journal of Accounting Research*, 1966. – 4. – P. 71–111.
135. Bell C. C., Han V., Sugawara Y., Grant K. Synaptic plasticity in a cerebellum-like structure depends on temporal order// *Nature*, 1997. – Vol. 387 – P. 278–281.
136. Bellman R. Adaptive Control Processes : A Guided Tour. New Jersey: Princeton University Press, 1961. – 255 p.
137. Bi G. Q., Poo M. M. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type // *Journal of Neuroscience*, 1998. – Vol. 18. – № 24. – P. 10464–10472.
138. Bishop C. M. Neural Networks for Pattern Recognition. Oxford : Oxford University Press, 1995. – 504 p.
139. Butenkov S. A., Krivsha V. V., Dhouyaru S. Al. Granular Computing in Computer Image Perceptron: basic issues and Glass Box models // *Artificial Intelligence and applications «AIA-2006»*: [Proc. Of IASTED Conf]. – Innsbruk (Austria), 2006. – P. 462–467.
140. Carpenter G. A., Grossberg S. The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network // *Computer*, 1988. – Match. – P. 77–88.
141. Carr C. E., Konishi M. A circuit for detection of interaural time differences in the brain stem of the barn owl // *J. Neuroscience*, 1990. – Vol. 70. – № 10. – P. 3227–3246.
142. Cernansky M., Macula M. Feed-forward Echo State Networks // *Proc. of IJCNN'05*. – Monreal, Canada, 2005. – July 31– Aug. 4. – P. 1479–1482.
143. Cheung Y. A new recurrent radial basis function network // *Neural Information Proceeding, ICONIP'02*, 2002. – Vol. 2. – P. 1032–1036.
144. Christiano P., Kelner J. A., Madry A., Shang-Hua Teng, Spielman D. Electrical Flows, Laplacian Systems, an Faster Approximation of Maximum Flow in Undirected Graph. (November 4, 2010) [www document]. document]. [Електронний ресурс]. – Режим доступу :

- <http://people.csail.mit.edu/madry/docs/maxflow.pdf> (18.10.2011).
145. Christiano P., Kelner J. A., Madry A., Shang-Hua Teng, Spielman D. (STOC 2011) Approximate Maximum Flow in Undirected Network. (September 28, 2011) [www document]. [Электронный ресурс]. – Режим доступа : <http://www.mpi-inf.mpg.de/~mehlhorn/ftp/ApproxMaxFlow.pdf> (18.10.2011).
 146. Chua L. and Yang L. Cellular Neural Networks: Theory // IEEE Trans. on Circuits and Systems, 1988. – № 35(10). – P. 1257–1272.
 147. Cook C., Rosenfeld A., Aronson A. Grammatical inference by hill-climbing // Information Science, 1976. – № 10. – P. 59–80.
 148. Elman J. L. Finding Structure in Time // Cognitive Science, 1990. – 14. – P. 179–211.
 149. Feng J. Is the integrate-and-fire model good enough a review // Neural Networks, 2001. – Vol. 14. – № 6. – P. 955–975.
 150. Fraser A. S. Simulation of genetic systems by automatic digital computers. // I. Introduction. Part I,II. Aust. J. Biol. Sci., 1957. – Vol. 10. – P. 484–499.
 151. Fukushima K. Cognitron : A Self-organizing Multiplayered Neural Network // Biol. Cybern, 1975. – 20. – P. 121–136.
 152. Fukushima K. A. A Neural Network for Visual Pattern Recognition // IEEE Computer, 1988. – 21. – № 3. – P. 65–75.
 153. Garey M. R. and Johnson D. S., Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979. – 340 p.
 154. Gonzales R. C., Thomason M. G. Syntactic pattern recognition. // Edit. Addison Wesley. – London, 1978. – P. 118–120.
 155. Grossberg S. Nonlinear Neural Networks: Principles, Mechanisms and Architectures // Neural Networks, 1988. – 1. – № 1. – P. 17 – 61.
 156. Happel, Bart and Murre, Jacob. The Design and Evolution of Modular Neural Network Architectures // Neural Networks, 7. – 1994. – P. 985–1004.
 157. Haykin S. Neural Networks. A Comprehensive Foundation // Upper Saddle River. – N. Y. : Prentice Hall, 1999.
 158. Hebb D. O. The Organization of Behavior New York : John Wiley & Sons, 1949. – 378 p.
 159. Hecht-Neilsen R. Kolmogorov's Mapping Neural Network Existence Theorem // Proc. of IEEE First Ann. Int. Conf. on Neural Networks. – San-Diego, 1987. – 3. – P. 11–13.
 160. Holland J. H. Nonlinear Environments Permitting Efficient Adaptation // Computer and Information Science IIs. – New York. – NY : Academic Press, 1967.
 161. Hopfield J. J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities // Proc. of the National Academy of Science, 1982. – 79. – P. 2554–2558.
 162. Hopfield J. J., Tank D. W. Neural computation of decisions in optimization problems // Biol Cybern, 1985. – Vol. 52. – № 3. – P. 141–152.

163. Hubel D. H. and Livingstone M. S. Color and contrast sensitivity in the lateral geniculate body and primary visual cortex of the macaque monkey // *Journal of Neuroscience*. 10: – 1990. – P. 2223–2237 ; [Електронний ресурс]. – Режим доступу : <http://www.jneurosci.org/cgi/content/abstract/10/7/2223>.
164. Jaeger H. Reservoir Riddles: Suggestion for Echo State Network Research // *Proc. of IJCNN'05*. – Monreal, Canada, 2005. – July 31 – Aug. 4. – P. 1460–1462.
165. Jordan M. I. Attractor Dynamics and Parallelism in a Connectionist Sequential Machine // *Proc. of the Eight Annual Conf. of the Cognitive Science Society*, Erlbaum, Hillsdale NJ, 1986. – P. 531–546.
166. Juang C. F., Ku K. C., Chen S. K. Temporal Hand Gesture Recognition by Fuzzified TSK-Type Recurrent Fuzzi Network // *Proc. of IJCNN'05*. – Monreal, Canada, 2005. – July 31 – Aug. 4. – P.1848–1853.
167. Kandel E. C.; Schwartz J. H. *Principles of Neural Science* New York : Elsevier, 3rd edition. – P. 1991–1992.
168. Kohonen T. *Associative Memory: A System Theoretic Approach*. – Berlin : Springer, 1977.
169. Kohonen T. The Self-Organizing Formation of Topologically Correct Future Maps // *Biological Cybernetics*. – 1982. – 43. – P. 59–69.
170. Kohonen T. The Self-Organizing Map // *Proc. of IEEE*. – 1990. – 78. – № 9. – P. 1464–1480.
171. Kohonen T. *Self-Organization and Associative Memory*. – Berlin : Springer, 2012, 312 с.
172. Kosko B. Adaptive Bidirectional Associative Memories // *Appl. Optics*. – 26. – № 33. – 1987. – P. 4947–4960.
173. Krasilenko V. G. et al., Design of time-pulse coded optoelectronic neuronal elements for nonlinear transformation and integration // *Proc. SPIE*, Vol. 6974. – 2008. – P. 286–297.
174. Lesk M. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. // *In SIGDOC'86: Proceedings of the 5th annual international conference on Systems documentation*. – New York : NY, USA. ACM. – 1986. – P. 24–26.
175. Lippman R. P. An Introduction to Computing with Neural Nets // *IEEE ASSP Magazine*, 1987. – № 4. – P. 4–22.
176. MacKay D. J. C Bayesian interpolation // *Neural Computation*, 1992. – Vol. 4. – № 3. – P. 415–447.
177. Mehrabian A. Communication without words // *Psychology Today*, 1968. – Vol. 2. – № 4. – P. 53–56.
178. Mehta M. R., Lee A. K., Wilson M. A. Role of experience and oscillations in transforming a rate code into a temporal code // *Nature*, 2002. – Vol. 417. – P. 741–746.
179. Mozer M. C., Hillsdale N. J. A focused backpropagation algorithm for

- temporal pattern recognition. // Backpropagation, Lawrence Erlbaum Associates, 1995. – P. 137–169.
180. Muezzinoglu M. K., Zurada J. M. A Recurrent RBF Network Model for Near Neighbour Classification // Proc. of IJCNN'05. – Monreal, Canada, 2005. – July 31 – Aug. 4. – P. 343–348.
181. Narendra K. S., Parthasarathy K. K. Identification and control of dynamical systems using neural networks // IEEE Transactions on Neural Networks. – 1990. – N 1. – P. 4–27.
182. Nauck D. B., Klawonn F., Kruse R. Neuronale Netze und Fuzzy – Systeme. // Fuzzy Modelling: Paradigms and Practice/ Kluwer, Boston, 1996. – P. 203–228.
183. Parzen E. Estimation of a Probability Density Function and Mode // Fnn. Math. Stat. – 1962. – 33. – P. 1065–1076.
184. Pawlak Z. RoughSetsTheoreticalAspects of Reasoning about Data. – Boston : London : Academic Publishers, 1991. – 229 p.
185. Pineda F.J. Dynamic and Architectures for Neural Computation // J. of Complexity. – 1988. – 4. – P. 216 – 245.
186. Pineda F. J. Recurrent Backpropagation and the Dynamic Approach to Adaptive Neural Computation // Neural Computation, 1989. – 1. – P. 161–172.
187. Powell M. J. D. Radial Basis Functions for Multyvariable Interpolation : A review // Proc. Of IMA Conf. on Algorithms for the Approximation of Functions and Data, Shrivenham, UK. – 1985. – P. 143–167.
188. Resnik P. Using information content to evaluate semantic similarity in a taxonomy. In International Joint Conference for Artificial Intelligence (IJCAI-95). – 1995.– P. 448–453.
189. Reyneri L. M. A Comparison between Analog and Pulse Stream VLSI Hardware for Neural Networks and Fuzzy Systems / L. M. Reyneri, H. C. A. M. Withagen, J. A. Hegt, M. Chiaberge // Proc. of MICRONEURO 94, Int'l Conf. on Microelectronics for Neural Networks and Fuzzy Systems, Torino (I), September 1994. – IEEE Computer Society Press. – P. 77–86.
190. Rieke F., Warland D., Ruyter R. van Steveninck, Bialek W. Spikes: Exploring the Neural Code // Computational Neurosciences series. – MIT Press, 1997. – 395 p.
191. Rumelhart D. E., Hinton G. E., Williams R.J. Learning Internal Representation by Error Propagation // Parallel Distributed Processing. – Cambridge, MA, MIT Press, 1986. – Vol. 1. – Chapt. 8. – P. 318–364.
192. Saggie K., Keinan A., Ruppin E. Solving a delayed response task with spiking and McCulloch-Pitts agents // Advances in Artificial Life: 7th European Conference, ECAL 2003 Dortmund, Germany? 2003. – P. 199–208.
193. Specht D.F. Probabilistic Neural Network for Classification, Mapping or Associative Memory // Proc. of the IEEE Int. Conf. on Neural Networks, SanDiego, 1988. – Vol. 1. – P. 525–532.

194. Specht D. F. A General Regression Neural Network // Proc. of the IEEE Int. Conf. on Neural Networks, SanDiego, 1991. – 2. – № 6. – P. 568–576.
195. Steil J. Backpropagation – Decorelation : online recurrent learning with O(N) complexity // Proc. of IJCNN'04. – Budapest, 2004. – P. 843–849.
196. Takagi H., Hayashi L. NN-criven fuzzy reasoning // International Journal of Approximate Reasoning, 1991. – Vol. 5. – № 3. – P. 192–212.
197. Takagi T., Sugeno M. Fuzzy Identification of Systems and its Applications to Modelling and Control // IEEE Transactions on Systems, Man and Cybernetics. – 1985. – Vol. 15. – P. 116–132.
198. Tank D. W. and Hopfield J. J. Simple neural optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit // IEEE Trans. Circuits Syst. – 1986. – Vol. CAS-33. – P. 533–541.
199. Tryon R. C. Cluster analysis / R. C. Tryon. – London : Ann Arbor Edwards Bros, 1939. – 139 p.
200. Uzga-Rebrovs O. Nenoteiktibuparvaldisana 2. – Rezekne : R.A. Izdevnieciba, 2010. – 560 p.
201. Venelinov Topalov, A. Kaynak. Online learning in adaptive neurocontrol schemes with a sliding mode algorithm // IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics. – 2001. – Vol. 31. – I. 3. – P. 445–450.
202. Weibel P. F. Die Aussagefähigkeit von Kriterien zur Bonitätsbeurteilung im Kreditgeschäft der Banken. – Bern, 1973. – S. 235.
203. Werbos P. J. Beyond regression: New Tools for Prediction and Analysis in Behavioral Sciences. Ph. D. thesis. – Cambridge : Harvard University, MA, 1974.
204. Widrow B., Hoff M. E. Jr. Adaptive switching circuits // Western Conference Rec., IRE. – 1960. – № 4. –P. 96–104.
205. Wu Z. and Palmer M. Verb semantics and lexical selection // 32nd Annual Meeting of the Association for Computational Linguistics. – New Mexico State University, Las Cruces, New Mexico. – 1994. – P. 133–138.
206. Zadeh L. A. Fuzzy Sets and Information Granularity // Advances in Fuzzy Set Theory and Applications, M. Gupta, R. Ragade, and R. Yager, Eds. Amsterdam, The Netherlands: North-Holland. – 1979. – P. 3–18.
207. Zadeh L. A. Toward a Theory of Fuzzy Information Granulation and its centrality in human reasoning and fuzzy logic // Fuzzy Sets Syst. – Vol. 90. – 1997. – P. 111–127.

Навчальне видання

Валентин Якович Кутковецький

Розпізнавання образів

Навчальний посібник

Редактор, технічний редактор, комп'ютерна верстка, дизайн обкладинки *Л. Бернацька*.
Друк *С. Волинець*. Фальцювально-палітурні роботи *О. Кутова*.

Підп. до друку 26.10.2017.
Формат 60 × 84 ¹/₁₆. Папір офсет.
Гарнітура «Times New Roman». Друк ризограф.
Ум. друк. арк. 48,8. Обл.-вид. арк. 21.
Тираж 300 пр. Зам. № 5301.

54003, м. Миколаїв, вул. 68 Десантників, 10.
Тел.: 8 (0512) 50-03-32, 8 (0512) 76-55-81, e-mail: rector@chmnu.edu.ua.
Свідоцтво суб'єкта видавничої справи ДК № 3460 від 10.04.2009.