Міністерство освіти і науки України
Сумський державний університет

В. В. Москаленко

МОДЕЛІ І МЕТОДИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ БАГАТОВИМІРНИХ ДАНИХ ЗА УМОВ АПРІОРНОЇ НЕВИЗНАЧЕНОСТІ

Монографія

Рекомендовано вченою радою Сумського державного університету

Суми
Сумський державний університет
2020

УДК 004.75
М82

Рецензенти:
Є. В. Бодянський – доктор технічних наук, професор (Харківський національний університет радіоелектроніки);
С. О. Субботін – доктор технічних наук, професор (Національний університет «Запорізька політехніка»)

Рекомендовано до видання
вченою радою Сумського державного університету
(протокол № 4 від 14 листопада 2019 року)

Москаленко В. В.

М82 Моделі і методи інтелектуального аналізу багатовимірних даних за умов апріорної невизначеності : монографія / В. В. Москаленко. – Суми : Сумський державний університет, 2020. – 184 с.

ISBN 978-966-657-815-3

Монографія присвячена викладенню сучасних ідей і методів синтезу та оптимізації моделей аналізу даних. Значну увагу приділено принципам інтелектуальної інформаційно-екстремальної технології аналізу та синтезу здатних навчатися систем прийняття рішень, розробленій науковим колективом лабораторії інтелектуальних систем Сумського державного університету. Викладений у монографії матеріал може бути корисним під час створення сучасних інтелектуальних систем різного призначення та підготовки наукових працівників, викладачів, аспірантів і магістрантів за спеціальністю «Комп'ютерні науки».

УДК 004.75

ISBN 978-966-657-815-3

ЗМІСТ

С.

СПИСОК СКОРОЧЕНЬ……5
ПЕРЕДМОВА…….. 6
ВСТУП…….. 8
РОЗДІЛ 1. ВСТУП ДО ПРОБЛЕМИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ БАГАТОВИМІРНИХ ДАНИХ……..……...12
1.1. Сучасний стан та тенденції розвитку технологій інтелектуального аналізу даних…….....12
1.2. Аналітичний огляд моделей і методів побудови ознакового опису спостережень…….……..26
1.3. Аналітичний огляд моделей і методів побудови вирішувальних правил……..……48
Розділ 2. ІНФОРМАЦІЙНИЙ СИНТЕЗ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ АНАЛІЗУ ДАНИХ……...75
2.1. Формалізована постановка задачі інформаційного синтезу систем аналізу даних……..75
2.2. Модель і метод навчання екстрактора ознакового опису спостережень……..……78
2.3. Модель і метод синтезу класифікаційних вирішувальних правил…….....98
2.4. Модель і метод синтезу регресійних вирішувальних правил……...119
2.5. Методи точного налаштування моделі аналізу даних……127
2.6. Критерії та методи оптимізації параметрів функціонування системи аналізу даних…….....134

Розділ 3. ПРИКЛАДИ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЙ АНАЛІЗУ ДАНИХ……..…146

3.1. Інтелектуальна система детектування об’єктів інтересу на аерозображенні……...….146

3.2. Інтелектуальна система візуальної навігації…...**.**156

3.3. Інтелектуальна система розпізнавання шкідливого мережевого трафіку……..…168

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ……...….176

СПИСОК СКОРОЧЕНЬ

СКД – система полів контрольних допусків.

КФЕ – критерій функціональної ефективності.

ІЕІ-технологія – інформаційно-екстремальна інтелектуальна технологія.

ПЕРЕДМОВА

У монографії викладено аналітичний огляд сучасних принципів і підходів до синтезу систем інтелектуального аналізу багатовимірних даних за умов структурної, параметричної та ймовірнісно-статистичної невизначеностей, а також власний досвід створення гібридних інтелектуальних систем. У науковій праці розглянуто особливості побудови екстрактора ознакового опису спостережень і синтезу вирішувальних правил за умов обмежених обсягів вибірки розмічених навчальних даних та обчислювальних ресурсів. Запропоновано методи, алгоритми та критерії оптимізації параметрів функціонування системи аналізу даних в інформаційному та вартісному сенсах.

Монографію підготовлено в межах виконання держбюджетної науково-дослідної роботи «Інтелектуальна автономна бортова система безпілотного літального апарату для ідентифікації об’єктів на місцевості» (ДР № 0117U003934). У монографії наведено приклади застосування запропонованих моделей і методів інтелектуального аналізу даних для вирішення завдань ідентифікації об’єктів інтересу на місцевості, візуальної навігації та детектування шкідливого мережевого трафіку. Дослідницькі роботи виконували на базі лабораторії інтелектуальних систем Сумського державного університету, тому в монографії крім власних напрацювань уміщено узагальнений досвід колективу лабораторії.

Викладений у монографії матеріал може бути корисним під час створення сучасних інтелектуальних систем

різного призначення та під час підготовки наукових працівників, викладачів, аспірантів та магістрантів зі спеціальності «Комп'ютерні науки».

ВСТУП

Сучасні досягнення в галузях сенсорної техніки та інфокомунікаційних технологій сприяють накопиченню даних про навколишній світ. Існує потреба перетворення накопичених даних на інформацію та знання для прогнозування функціонального стану і оптимізації процесів, що становлять інтерес для господарської діяльності людини. На сьогодні основним підходом до вирішення подібних завдань є використання методів інтелектуального аналізу даних. У межах даного підходу розроблено багато алгоритмів, що моделюють когнітивні процеси, притаманні людині під час прийняття рішень. Однак різного роду невизначеності є негативними факторами, що знижують ефективність аналізу даних. Джерелами невизначеності є неточність, неповнота та суперечливість даних. Крім того, залежно від постановки завдання аналізу вхідні дані можуть бути нерепрезентативними, неструктурованими, незбалансованими, з неповною чи неточною розміткою. При цьому багатовимірність спостережень висуває додаткові вимоги до обсягу вибіркових даних та обчислювальних ресурсів. Ця проблема відома як «прокляття розмірності».

Дослідження, пов'язані з розробленням систем інтелектуального аналізу даних різного призначення в науково-технічній літературі знайшли широке висвітлення завдяки ідеям і науковим здобуткам, насамперед, О. Г. Івахненка, В. І. Васильєва, Є. В. Бодянського, А. С. Довбиша, С. О. Субботіна, О. Ю. Соколова, М. І. Шлезінгера,

В. С. Степашко, В. Н. Вапніка, А. Я. Червоненкіса, Н. Г. Загоруйка, Й. Бенджіо, Д. Хінтона, Я. Лі Куна та інших учених. При цьому питання підвищення ефективності систем інтелектуального аналізу даних за умов ресурсних та інформаційних обмежень усе ще залишаються не повністю дослідженими через науково-методологічні ускладнення. Ці ускладнення полягають у наявності структурної, параметричної та ймовірнісно-статистичної невизначеностей під час синтезу оптимальної моделі аналізу даних. Однак існують підходи, в межах яких розглядається ефективне використання, крім розміченої навчальної вибірки інших джерел інформації, що мають статистичний зв'язок із вирішуваним завданням. Також останнім часом розробляється багато універсальних стратегій і методів пошукової оптимізації та композиції моделей, спрямованих на зниження вимог до обчислювальних ресурсів або вхідних даних. Усі ці розробки мають свої недоліки та переваги залежно від умов їх застосування. Тому перспективним напрямком підвищення ефективності систем аналізу даних є гібридизація та комплексне використання різних підходів з метою одержання оптимального в інформаційному та вартісному сенсі рішення.

Монографія присвячена викладенню сучасних ідей і методів синтезу та оптимізації моделей аналізу даних. Значну увагу приділено принципам інтелектуальної інформаційно-екстремальної технології аналізу та синтезу здатних навчатися систем прийняття рішень, розробленій науковим колективом лабораторії інтелектуальних систем Сумського державного університету. Перевагою цієї технології є ви-

користання обчислювально ефективних операцій для транформації простору ознак та побудови оптимальних в інформаційному сенсі класифікаційних вирішувальних правил за умов апріорної невизначеності та ресурсних обмежень.

Монографія складається з трьох розділів. У першому розділі здійснено аналітичний огляд сучасних моделей та методів навчання екстракції інформативного ознакового опису спостережень, побудови вирішувальних правил та методів оптимізації їх параметрів. У другому розділі запропоновано моделі та методи інтелектуального аналізу даних за умов ресурсних та інформаційних обмежень. При цьому розглядаються особливості архітектури екстрактора ознакового опису для багатовимірних даних різної топології, алгоритми його попереднього навчання та точного налаштування. Запропоновано алгоритми побудови регресійних і класифікаційних вирішувальних правил, а також алгоритми та комплексні критерії оптимізації параметрів функціонування системи аналізу даних. У третьому розділі наведено приклади застосування запропонованих моделей і методів інтелектуального аналізу даних для розв'язання задач ідентифікації об'єктів інтересу на місцевості, візуальної навігації та детектування шкідливого мережевого трафіку. При цьому приклади супроводжуються результатами фізичного моделювання та їх аналізом.

Автор висловлює щиру подяку колективу лабораторії інтелектуальних систем канд. техн. наук А. С. Москаленко, А. Г. Коробову та М. О. Зарецькому за їх участь у розробленні та програмній реалізації окремих алгоритмів. Глибо-

ка вдячність та шана керівнику лабораторії інтелектуальних систем д-ру техн. наук, професору А. С. Довбишу за консультування та підтримку.

РОЗДІЛ 1
ВСТУП ДО ПРОБЛЕМИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ БАГАТОВИМІРНИХ ДАНИХ

1.1. Сучасний стан та тенденції розвитку технологій інтелектуального аналізу даних

Під інтелектуальним аналізом даних розуміють процес пошуку у вхідних даних раніше невідомих, нетривіальних, практично корисних знань, що доступні для інтерпретації та необхідних для прийняття рішень у різноманітних сферах діяльності. Сучасне інфокомунікаційне середовище накопичує значні обсяги даних про процеси навколишнього середовища, зокрема й даних про функціональний стан елементів самої інфокомунікаційної системи. На практиці дані накопичені в інфокомунікаційному середовищі з метою їх подальшого аналізу, можуть бути неточними, неповними, суперечливими, різнорідними, непрямими та характеризуватися високою розмірністю. Процеси перетворення вхідних даних на інформацію та знання потребують значних інтелектуальних зусиль і ресурсів, тому існує потреба в їх автоматизації та оптимізації.

На процес синтезу та верифікації моделей інтелектуального аналізу даних впливає ряд негативних факторів, що погіршують результат. Ці фактори називають невизначеностями, поява яких зумовлена неповнотою та неточністю наших знань стосовно досліджуваного процесу чи об’єкта. Невизначеності можуть бути зумовлені неповнотою даних чи похибками (шумами) вимірювань, негативним впливом

випадкових збурень на функціонування досліджуваного об’єкта, неправильним оцінюванням структури моделі чи ймовірнісного розподілу спостережень. У загальному випадку невизначеності можна поділити на структурні, параметричні та ймовірнісно-статистичні.

Структурна невизначеність означає, що структура моделі, що описує дані, точно невідома. Можливість пояснення одних і тих самих процесів із різних точок зору призводить до мультиструктурності опису. Одну й ту саму залежність (функцію) можна апроксимувати моделями різної структури, однак обчислювальна складність моделей та ефективність налаштування їх параметрів буде різною. При цьому неповнота апріорних і фактичних даних вимагає спрощення структури моделей, щоб не нав’язувати даним невластиву їм структуру. Тобто складність моделі не повинна перевищувати складність описуваного процесу. При неоптимальному виборі структури моделі можливі проблеми недонавчання (underfitting) та перенавчання (overfitting). У разі недонавчання, модель недостатньо складна для опису даних із необхідною точністю. У разі перенавчання, що виникає при надлишковій складності моделі, середня помилка на тестовій вибірці суттєво вища ніж на навчальній вибірці. Розроблено багато різноманітних методів оцінювання складності, використовуваній під час вибору структури моделі аналізу даних. Одним із них є використання критерію Акаіке (AIC), що ґрунтується на принципі бритви Оккама, а також тісно пов’язаний із ним Баєсів інформаційний критерій (BIC) [1]. У теорії Вапніка-Червоненкіса одним із ключових понять є розмірність, що

також є характеристикою складності сімейства алгоритмів і дозволяє узгодити кількість вільних параметрів моделі з довжиною вибірки. Проте, оскільки задача опису даних формально еквівалентна кодуванню, то складність моделі можна оцінити також як довжину коду, необхідного для її опису. На цьому ґрунтується принцип мінімальної довжини опису (MDL, minimum description length), згідно з яким необхідно мінімізувати загальну довжину опису даних за допомогою моделі та довжину опису самої моделі [2]. Крім того, на вибір структури моделі накладають обмеження вимоги алгоритмів оптимізації параметрів. Деякі алгоритми навчання вимагають диференційованості функцій, що є структурними елементами моделі, інші вимагають існування оберненої функції для структурних елементів моделі, а деякі мають менше вимог до самої моделі, проте характеризуються високими ресурсними вимогами до обчислювального середовища.

У задачі зняття структурної невизначеності важливу роль відіграють методи регуляризації та оптимізації гіперпараметрів. Під регуляризацією розуміють введення додаткових обмежень до завдання навчання з метою уникнення перенавчання, що полягає у втраті узагальнювальної здатності. Під гіперпараметрами розуміють невелику кількість параметрів, що регулюють складність та ємність моделі. Оптимізацію гіперпараметрів здійснюють із метою отримання моделі, оптимальної в інформаційному та вартісному сенсах. При цьому розрізняють гіперпараметри, що впливають на ємність моделі, та гіперпараметри, що впливають на поведінку алгоритму навчання. Регуляризацію

використовують у процесі навчання, а гіперпараметри задають до початку навчання. До гіперпараметрів відносять і параметри регуляризації, що налаштовують до початку навчання. Можна виділити чотири сучасні підходи до зняття структурної невизначеності:

– навчання структурно надлишкової моделі з використанням методів регуляризації з подальшим видаленням елементів, що істотно не впливають на ефективність моделі, яку оцінюють за результатами валідації (наприклад, Structured Probabilistic Pruning);

– навчання простої моделі з інкрементальним ускладненням її структури до моменту припинення покращання її ефективності, що оцінюють за результатами валідації (наприклад, Group method of data handling, Boosting, Growing Neural Gas);

– використання пошукового алгоритму для вибору оптимальної структури, що забезпечує максимальну ефективність навчання за результатами валідації (наприклад, NeuroEvolution of Augmenting Topologies, Weight Agnostic Neural Networks);

– побудову мультимоделі методом навчання декількох диверсних моделей, спрямованих на розв'язання однієї і тієї самої задачі, та об'єднання їх прогнозів із метою взаємної компенсації помилок (наприклад, Bagging, Boosting);

– комбінацію перелічених підходів.

Параметрична невизначеність означає, що оптимальні параметри моделі з відомою структурою невідомі. Задача їх визначення ускладнюється багатоекстремальністю функціоналу якості моделі. Параметрична невизначеність хара-

ктерна для складної моделі з великою кількістю параметрів, що описують слабоформалізований процес із нелінійними залежностями. Існують різноманітні методи локальної та глобальної оптимізації параметрів моделі аналізу даних. Найбільш популярним підходом до оптимізації параметрів є використання алгоритму зворотного поширення помилки, що ґрунтується на алгоритмі градієнтного спуску та його модифікаціях. Команда Йошуа Бенгіо експериментально встановила, що під час оптимізації параметрів великих нейронних мереж фактично немає проблем із застряванням у локальному мінімумі [3]. Але існують сідлові точки, які є локальними мінімумами в деяких вимірах, і навчання в цих точках може значно сповільнюватися. Особливо затягнутим у часі може бути процес оптимізації, якщо алгоритм наштовхується на локальні мінімуми в багатьох вимірах одночасно. Проте якщо почекати досить довго, то після коливань в межах сідлової точки алгоритм покине її в пошуках глобального оптимуму. При цьому застосування методів градієнтного спуску зумовлює вимогу диференційованість для функції структурних елементів та функціоналу якості моделі.

Алгоритм зворотного поширення помилки реалізує навчання складної моделі за схемою із кінця в кінець (end-to-end learning), тобто цілісно, згідно з єдиним принципом, що дозволяє підвищити рівень автоматизації процесів машинного навчання. При цьому кожен крок навчання спрямований на кінцеву мету, закодовану загальною цільовою функцією, що усуває необхідність у навчанні окремих модулів для виконання допоміжної (проміжної)

мети, не пов'язаної з кінцевим завданням. Однак під час навчання з кінця в кінець нівелюється цінна архітектурна інформація, сформована в результаті декомпозиції загальної задачі на підзадачі у вигляді модулів, що може призводити до розмивання функцій окремих дуже різних модулів (окремі модулі можуть брати на себе допоміжну невластиву їм функцію), а також до перешкоджання навчання окремих модулів один одному внаслідок нетривіальних взаємодій. Крім того, часто для вирішення окремої підзадачі доступна більша кількість даних і вона може бути ефективніше вирішена з окремими налаштуваннями алгоритму.

Розпаралелене виконання процесу навчання окремих шарів багатошарової моделі за схемою з кінця в кінець на основі алгоритму зворотного поширення помилки ускладнене необхідністю дочекатися поширення помилки від високорівневих шарів для оновлення вихідних значень та параметрів нижніх шарів. Навчання з використанням відкладених та нелокальних помилок унеможливлює узгодження зворотного поширення з механізмами навчання, спостережуваних у біологічних нейронних мережах, оскільки це вимагає, щоб нейрони зберігали пам'ять входу досить довго, доки не з'являються помилки більш високого рівня.

До ймовірнісно-статистичних невизначеностей можна віднести невизначеність закону розподілу даних та випадкових збурень, що впливають на досліджуваний процес, наявність пропусків у даних та імпульсних значеннях (викидів), наявність шумів (похибок) вимірювань, наявність прихованих (невимірюваних змінних), вплив коротких ви-

бірок та недостатньої інформативності даних для побудови та оцінювання ефективності моделей. Для зниження впливу ймовірнісно-статичних невизначеностей використовують різноманітні методи заповнення пропусків, нормалізації та зниження розмірності вхідних даних, видалення дублювальних та напівдублювальних даних, розширення (аугментації) та балансування вибіркових даних [4].

Сучасна концепція інтелектуального аналізу даних полягає у використанні ідей і методів машинного навчання та розпізнавання образів, спрямованих на автоматизацію синтезу моделей, що узагальнюють вхідні дані. Залежно від наявності та типу навчального сигналу (зворотного зв'язку) виділяють такі види машинного навчання: навчання з учителем, навчання без учителя, навчання з незначним залученням вчителя (weakly-supervised learning) та навчання з підкріпленням (reinforcement learning).

Навчання з учителем (supervised learning) є одним зі способів машинного навчання, під час якого модель навчається на вибірці пар <вектор ознак, мітка> («стимул-реакція») з метою визначення залежності між ними. На даний момент навчання з учителем є найбільш розвиненим підходом із великою кількістю практичних застосувань. Однак цей підхід досі критикують за його біологічну неправдоподібність. У мозку досі не знайдено механізму, який би формував бажані виходи й порівнював бажані та дійсні значення виходів із метою здійснення корекції за допомогою зворотного зв'язку.

Навчання без вчителя (unsupervised learning) полягає в спонтанному навчанні без втручання з боку експеримента-

тора, тому його вважають більш біологічно правдоподібним підходом. Навчання без учителя використовують для виявлення внутрішніх взаємозв'язків, залежностей і закономірностей між об'єктами вибірки. Навчання без учителя може вирішувати такі завдання як побудова генеративних моделей, кластерний аналіз даних, зниження розмірності даних або пошук асоціативних правил.

Синтез генеративної моделі має на меті допомогти зрозуміти, як влаштований процес, що вивчається, для його відтворення. З математичної точки зору синтез генеративної моделі полягає в оцінюванні щільності ймовірності навчальних даних для генерації нових даних із цього розподілу, тобто подібних до навчальних даних. До методів побудови генеративних моделей відносять ймовірнісний метод головних компонент (Probabilistic Principal Component Analysis), мережі змагального навчання (Generative Adveserial Network), мережі варіаційних автоенкодерів (Variational Autoencoder) та авторегресійні генеративні моделі (Autoregressive Generative Model). Основним обмеженням до широкого практичного використання генеративних моделей є потреба в дуже великих обсягах навчальних даних, що можуть бути недоступними для багатьох прикладних сфер.

Задача кластерного аналізу даних полягає в розбитті заданої навчальної вибірки спостережень на підмножини, що називаються кластерами, так, щоб спостереження одного кластера були подібними, а спостереження різних кластерів істотно відрізнялися. Методів кластер-аналізу багато [5]. Водночас одні методи утворюють кластери, ґрунтую-

чись на відстані між спостереженнями, водночас інші методи здійснюють розбиття даних на кластери залежно від щільності ділянок у просторі даних. Також деякі методи кластер-аналізу здійснюють поділ на кластери залежно від інтервалів, у яких знаходяться спостереження, або ґрунтуються на належності спостережень до конкретних статистичних розподілів. Одним із найбільш популярних та обчислювально ефективних методів кластер-аналізу є метод k-середніх, та його модифікації, що полягає в мінімізації відхилення спостережень кластера від його центру (центр мас спостережень кластера).

Використання навчання без учителя для задачі зниження розмірності вхідних даних полягає в трансформації вхідного багатовимірного простору ознак під час навчання в простір меншої розмірності з мінімальними втратами інформації. До класичних методів зниження розмірності простору ознак відносять метод головних компонент (Principal Component Analysis) і його ядерні модифікації, метод незалежних компонент (Independent Component Analysis) та метод факторизації невід'ємних матриць (Nonnegative Matrix Factorization) [6]. У методі головних компонент завдання навчання формулюється як задача апроксимації даних лінійними багатовидами меншої розмірності, в ортогональній проекції, на які дисперсія даних буде максимальною. У методі незалежних компонент завдання навчання формулюється як задача розділення багатовимірного сигналу на адитивні підкомпоненти за допомогою максимізації їх статистичної незалежності. У методі факторизації невід'ємних матриць завдання навчання фор-

мулюється як задача розкладання невід'ємної матриці на добуток двох невід'ємних матриць без необхідності бути ортогональними. При цьому як цільову функцію використовують норму Фробеніуса [6]. Однак у випадку великого обсягу навчальних даних класичні алгоритми можуть бути достатньо грубими та ресурсомісткими. Тому більш популярними є нейромережеві підходи на основі багатошарових автоенкодерів, які можна навчати за схемою з «кінця-в-кінець» на основі алгоритму зворотного поширення помилки. Завдання навчання автоенкодера формулюється як мінімізація помилки реконструкції вхідних даних на виході мережі. При цьому на проміжний шар автокодувальника накладають обмеження: проміжний шар повинен бути меншої розмірності, ніж вхідний і вихідний шари.

На практиці розмічених навчальних даних набагато менше, ніж доступних нерозмічених, даних (incomplete supervision). Крім того, якість розмітки може бути грубою (inexact supervision) або неточною (inaccurate supervision). Тому все більшого розвитку набуває так зване навчання з незначним залученням учителя (weakly-supervised learning). В межах цього підходу виділяють такі види, як активне навчання (Active learning) та навчання з частковим залученням учителя (Semi-supervised learning). Активне навчання передбачає наявність «оракула», наприклад, людини-експерта, до якого можна направити запит на одержання точної мітки для вибраних нерозмічених спостережень. Зазвичай, експерту на розмітку відправляють спостереження, на яких алгоритм навчання має найбільшу невизначеність. На противагу цьому під час навчання з

частковим залученням учителя для поліпшення результатів, окрім розмічених даних, намагаються максимально ефективно використати великі обсяги нерозмічених даних.

Ще одним важливим видом машинного навчання є навчання з підкріпленням (reinforcement learning), де тренувальні дані (у вигляді винагород та покарань) надходять як зворотний зв'язок на дії програми в динамічному середовищі. Навчання з підкріпленням відрізняється від стандартного навчання з учителем тим, що пари правильних входів/виходів ніколи не представлені, а недостатньо оптимальні дії явно не виправляються в процесі функціонування. Метод навчання з підкріпленням повинен забезпечувати баланс між дослідженням (незвіданої території, англ. exploration) та використанням (поточного знання, англ. exploitation).

Для більшості процесів людської діяльності не існує функціонально повних математичних моделей, тобто процеси є слабоформалізованими. При цьому автоматизація багатьох слабоформалізованих процесів дозволяє знизити вплив суб'єктивних факторів і зменшити накладні витрати. Тому серед моделей інтелектуального аналізу даних найбільше практичне значення мають моделі класифікаційного та регресійного аналізу, оскільки вони дозволяють знімати невизначеність щодо поточного та майбутнього стану слабоформалізованих процесів. При цьому в моделях як класифікаційного, так і регресійного аналізу даних можна умовно виділити дві частини – екстрактор ознак та вирішувальні правила (рис. 1.1). Основне завдання екстрактора ознак – формування інформативного та компактного ознакового подання, зручного для класифікаційного чи регре-

сійного аналізу. Основне завдання вирішувальних правил є відображення ознакового подання в рішення, що приймається моделлю.

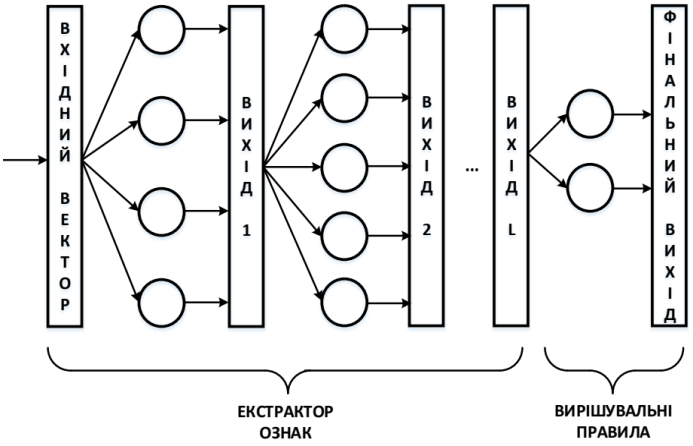


Рисунок 1.1 – Структура інтелектуальної моделі аналізу даних

На рисунку 1.1 показано, що кожна частина може мати багато шарів нелінійної трансформації даних, оскільки одного шару може бути недостатньо для апроксимації складних залежностей. Крім того, було доведено, що одношарові та багатошарові (ієрархічні) моделі можуть апроксимувати функціональні залежності з однаковою точністю, проте багатошарові моделі, які мають меншу обчислювальну складність, є більш компактними [7]. При цьому кожен шар може складатися зі скінченної кількості паралельних елементів нелінійного перетворення (хоча допускається наявність одного елемента лінійного перетво-

рення). Ці елементи зображені колами. Вихід кожного елемента конкатенується в загальний вихідний вектор даного шару. На рисунку цього не показано, але між шарами можуть використовуватися різні операції нормалізації вихідного вектора шару для підвищення завадозахищеності моделі.

Останнім часом багато досліджень і розробок пов'язано з реалізацією концепції неперервного навчання (Continual Learning), в межах якої передбачена постійна адаптація моделі до змін середовища й задач, повторне використання вже накопичених знань, набуття нових знань та розширення функціональних можливостей. Базовими для реалізації цієї концепції є техніки перенесення знань (Transfer learning) (рис. 1.2 а) та метанавчання (Meta-learning) (рис. 1.2 б), що полягає в навчанні навчатися (Learning-to-learn).

Техніка перенесення знань полягає в перенесенні та адаптації знань із моделі, навченої на великому обсязі даних, для вирішення нового завдання. Наприклад, вагові коефіцієнти нейромережевої моделі, навченої на датасеті ImageNet для розпізнавання тисячі класів, можуть бути запозичені для їх точного налаштування на нову задачу замість навчання моделі з «нуля». При цьому, чим сильніше відрізняється нова цільова область використання мережі від образів ImageNet, тим менше шарів навчених мереж можна повторно використати, або тим довше необхідно здійснювати їх точне налаштування.

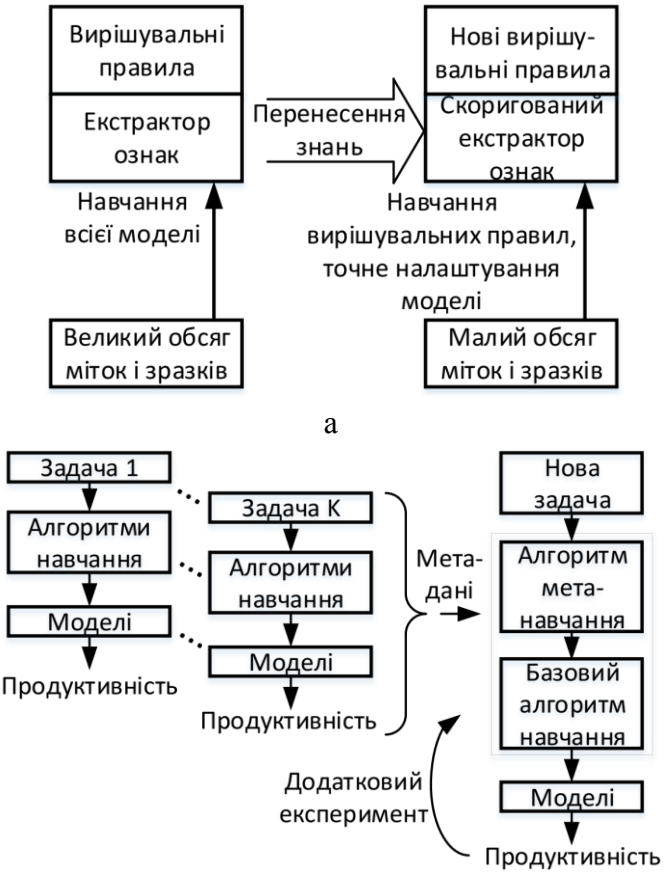


Рисунок 1.2 – Ілюстрація базових технік у концепції безперервного навчання: а – перенесення знань; б – навчання навчатися

Техніка мета-навчання полягає в аналізуванні процесу навчання моделі для розв’язання різних задач із метою навчання навчатися на нових задачах швидше та ефективніше. Найпростіша реалізація метанавчання дозволяє адап-

тивно змінювати гіперпараметри моделі в процесі її навчання. Більш складні реалізації можуть здійснювати корекцію кожного параметра моделі, повністю замінюючи базовий алгоритм навчання. Останні дослідження, пов'язані з мета-навчанням, спрямовані на пошук незалежного від моделі рішення (model agnostic solutions) [8].

Отже, роз'язання задачі синтезу ефективної моделі інтелектуального аналізу даних може бути ускладнено наявністю структурної, параметричної або ймовірнісно-статистичної невизначеностей. Крім того, наявні нейромережеві рішення не є біологічно правдоподібними й тому не дозволяють повною мірою моделювати когнітивні процеси, притаманні людині під час прийняття рішень. Однак забезпечення ієрархічності, модульності, обчислювальної ефективності та здатності до неперервного навчання є ключовими напрямками розвитку сучасних моделей аналізу даних.

1.2. Аналітичний огляд моделей і методів побудови ознакового опису спостережень

В умовах апріорної невизначеності інформативність окремих ознак у навчальних даних невідома. Кожне спостереження може мати велику кількість статистично залежних, неінформативних та дезінформуючих ознак. При цьому можна зіштовхнутися з проблемою «прокляття розмірності» (curse of dimensionality), що полягає у збільшенні розміру простору, доступного для «розсіювання» точок даних, унаслідок збільшення кількості вимірів – ознак

(рис. 1.3). Це призводить до експоненційного росту необхідного розміру навчальної вибірки для щільнішого заповнення важливої для аналізу області простору з метою виявлення будь-яких залежностей.

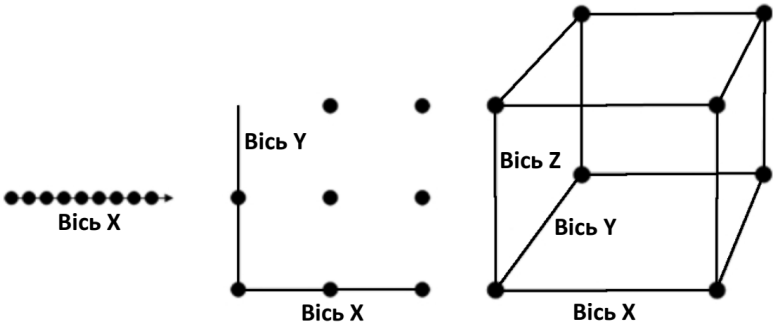


Рисунок 1.3 – Ілюстрація до проблеми
«прокляття розмірності»

На боротьбу з проблемою «прокляття розмірності» спрямовано багато класичних методів зниження розмірності, серед яких найбільш популярним є метод головних компонент. Проте ефективність цих методів є невисокою в умовах складної нелінійної залежності факторів [9].

Для подолання проблеми «прокляття розмірності» за умов складних нелінійних залежностей між факторами було розроблено багато локальних непараметричних алгоритмів навчання. Серед них найбільш популярними стали алгоритми, що ґрунуються на ядрах (kernel machines). У більшості подібних алгоритмів використано принцип локального узагальнення (local generalization). Ці алгоритми

ґрунтувалися на припущеннях про достатню гладкість цільової функції f, якій треба навчитися, тобто $f(x) \approx f(y)$ за умов $x \approx y$. Однак для ефективності ядерних алгоритмів навчальна вибірка повинна містити такі навчальні зразки, за допомогою яких можна явно вивчити всі піки та впадини цільової функції. У такому разі узагальнення досягають за рахунок локальної інтерполяції між сусідніми зразками навчальної вибірки. Проте кількість піків і впадин цільової функції може зростати експоненційно з кількістю взаємодіючих факторів, якщо алгоритм працює з «сирими» даними на вході. Тому алгоритми, в основу яких покладені властивість гладкості цільової функції та лінійна залежність факторів, краще використовувати на попередньо підготованому ознаковому просторі. Зручне подання даних спрощує завдання синтезу вирішувальних правил.

Галузь машинного навчання, що займається питаннями формування ознакового опису, називають наукою про подання даних (Representation Learning) або наукою про навчання ознак (Feature learning). Дослідниками цієї галузі було сформульовано ряд положень щодо ознакового опису спостережень, які є спільними для всіх задач інтелектуального аналізу даних. До таких положень належать [10]:

– ієрархічна організація пояснювальних факторів (першопричин);

– множинність пояснювальних факторів (першопричин);

– навчання з частковим залученням вчителя;

– спільність факторів під час розв’язання різних задач;

– гіпотеза багатовидів;

– гіпотеза про природну кластеризацію;

– просторово-часова зв’язаність;

– розрідженість ознакового подання;

– простота залежності високорівневого подання факторів.

Поняття, використовувані для опису середовища, можуть бути визначені в термінах інших більш абстрактних понять, тобто ієрархічно. У термінах аналізу даних це означає, що високорівневі ознаки формуються методом композиції низькорівневих (рис. 1.4). Тобто результуюча функція f для екстракції ознак може бути описана у вигляді композиції n шарів трансформації простору ознак $f = f1 \circ f2 \circ ::: \circ fn$.

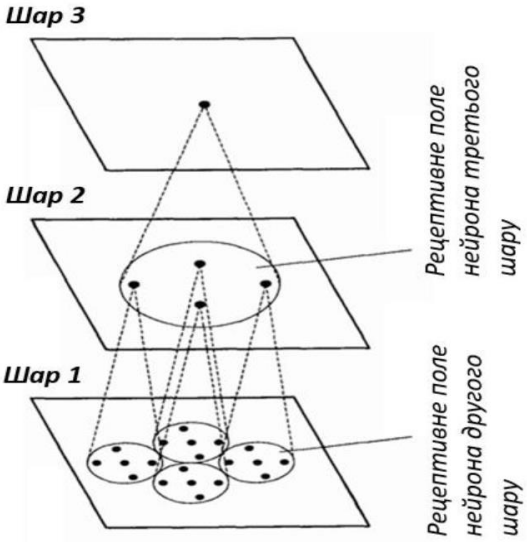


Рисунок 1.4 – Приклад ієрархічної композиції ознак

Варто відзначити, що неієрархічними та ієрархічними моделями можна апроксимувати будь-яку функцію з однаковою точністю, проте ієрархічні моделі потребують меншу кількість параметрів для цього [11]. Ієрархічні моделі мають більшу ємність і потенційно можуть бути навчені значно більшим абстракціям ознак на верхніх рівнях.

За допомогою ієрархічного подання можна ефективно вирішувати два взаємопов'язані але різні завдання : навчання інваріантним ознакам (invariant features) і навчання розділенню пояснювальних факторів. Метою навчання інваріантних ознак є формування високорівневих ознак, що є нечутливими до таких змін у вхідних даних, що є неінформативними і нецікавими для подальшого оброблення. Метою навчання розділенню пояснювальних факторів є формування якомога більшої кількості незалежних ознак (факторів) різного рівня абстрактності, оскільки інформативність ознак для подальшого оброблення може бути невідомою. При цьому сформований ознаковий опис може повторно використовуватися для вирішення різних завдань.

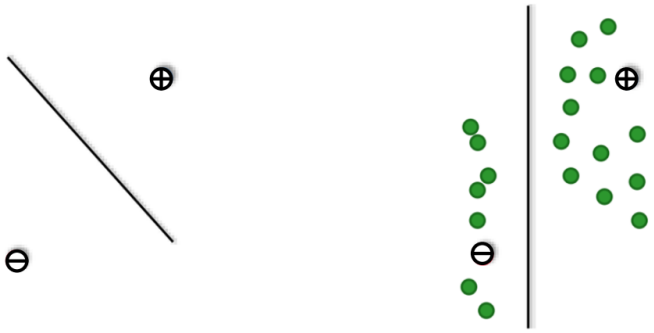
Вхідні дані є результатом взаємодії багатьох пояснювальних факторів. Тому навчання моделі новому фактору приводить до його узагальнення в конфігураціях інших факторів. Саме ця ідея лежить в основі розподіленого подання даних. Кожен параметр може повторно бути задіяним для кодування різних вхідних спостережень чи частин вхідного спостереження. При цьому ці спостереження можуть навіть не бути близькими сусідами. У розподіленому поданні даних експоненційно більша кількість ознак або

прихованих змінних можуть бути активовані вхідним сигналом, у той час як в алгоритмах із локальним узагальненням різні частини вхідного простору асоціюються лише зі своїм персональним набором параметрів [12]. Отже, розподілене подання даних може кодувати більшу кількість різноманітних вхідних конфігурацій ніж подання з використанням локального узагальнення.

Під час синтезу моделей аналізу даних багато дослідників використовують гіпотезу природньої кластеризації. Вона полягає в тому, що люди виділяють категорію (клас) і дають їй назву на основі статистичної подібності спостережень, віднесених до відповідної категорії. Локальні зміни в багатовидах мають тенденцію до збереження категорії, а лінійна інтерполяція між зразками різних класів проходить у загальному випадку через області з низькою щільністю ймовірності. Тобто для ознакового подання даних X щільність імовірності $P(X|Y = k)$ для різних категорій k прагне розділятися, а не накладатися. Іншими словами категорії асоціюється з різними багатовидами.

Важливе практичне значення має встановлення статистичного взаємозв'язку між навчанням без учителя та навчанням з учителем. Це дозволяє ефективно використати нерозмічені навчальні дані, які одержати простіше й дешевше, для формування інформативного ознакового опису. Було доведено, що ознакове подання даних X, зручне для обчислення ймовірнісного розподілу $P(X)$, є зручним і для навчання $P(Y|X)$, де Y – цільова змінна вирішувальних правил [13]. Саме в цьому й полягає ідея навчання з частковим залученням учителя. На рисунку 1.5 показано, що

використання навчання з частковим залученням учителя дозволяє уточнити межі категорій, беручи до уваги, що роздільна гіперповерхня повинна проходити через області простору з низькою щільністю ймовірності.



а б

Рисунок 1.5 – Ілюстрація до навчання з частковим залученням учителя: а – розподільна гіперповерхня для розмічених навчальних зразків; б – розподільна гіперповерхня для розмічених навчальних зразків з урахуванням щільності розподілу нерозмічених даних

У багатозадачних моделях аналізу даних може досягатися вищий рівень узагальнення в навчанні внаслідок підсилення статистичного взаємозв'язку між задачами, у яких використовуються спільні пояснювальні фактори. Такі моделі є дуже цінними особливо з точки зору передачі знань з метою їх адаптації до нових задач.

Необхідність зниження розмірності часто ґрунтується на гіпотезі багатовидів, згідно з якою основна щільність ймовірності даних зосереджена біля регіонів, що мають

набагато меншу розмірність, ніж оригінальний простір вхідних даних.

Під час синтезу моделей аналізу даних варто враховувати обмеження на зміну «крізь час і простір». Тобто спостереження, сформовані в сусідніх областях простору чи одержані послідовно в часі, повинні бути асоційованими з однаковими значеннями відповідної категорії понять, чи, приводити до невеликого руху по поверхні багатовиду високої щільності. Деякі дослідники вводили додаткову регуляризуючу складову до функціоналу якості, що враховує різницю значень ознак в різні моменти часу. Інші використовують апріорні знання про топологічну структуру даних для використання локальних рецептивних полів нейронів та операторів агрегації відгуку нейронів на сусідні ділянки вхідного простору в одне компактне подання. В обох випадках було досягнуто підвищення ефективності вирішувальних правил, побудованих із використанням одержаного ознакового подання. Крім цього, просторово-часова зв'язаність спостережень обґрунтовує спосіб розширення навчальних даних, основою якого є застосування невеликих випадкових деформацій образів в існуючих навчальних даних. При цьому деформувальні зміни повинні бути обмежені, щоб зберегти відношення згенерованих зразків до тієї самої категорії, що й оригінальний зразок.

Для будь-якого даного спостереження *x* лише мала частина з усіх можливих факторів є значимою. Більшість виділених ознак повинні бути нечутливими до малих змін спостереження *x*. Тобто більшість детектованих ознак повинна бути нульовою. Цю властивість називають розрі-

дженістю (sparsity). Вона може бути досягнута за рахунок різноманітних технік, що ґрунтуються на ефекті редукції причини (explaining away). Редукція причини полягає у зв'язуванні двох апріорно не зв'язаних причин події, якщо з'являється спостереження цієї події. У цих техніках можуть використовувати спеціальної форми приховані змінні h, більшість із яких прямують до нуля, або спеціальна нелінійність, значення якої знаходяться переважно близько нуля, або обмеження матриці Якобіана (або похідних функції) перетворення вхідних даних в обране подання [14]. При цьому знаходження апостеріорної ймовірності розподілу для активації прихованих факторів (причин) h, $p(h \mid x)$, часто використовуване як базис для екстракції ознак, виявляється складною задачею. У разі дискретного h задача взагалі може не мати розв'язку.

В ефективних високорівневих поданнях даних фактори зв'язані один з одним через прості залежності. Тому синтез вирішувальних правил варто здійснювати в межах підходів, що характеризуються найбільшою ефективністю з точки зору обчислень. Якщо ознакове подання здійснювали з самого початку з метою забезпечення максимальної інформативності для конкретних вирішувальних правил, то вирішувальні правила можуть бути максимально прості, наприклад, у вигляді лінійної моделі чи системи порогів. Проте у разі навчання інваріантного ознакового подання для ефективної адаптації до однієї чи декількох задач вирішувальні правила можуть містити додатковий шар прихованих змінних і нелінійні перетворення.

Навчання ієрархічної моделі з нуля за принципом із кі-

нця в кінець без попереднього навчання екстрактора є ресурсовитратним і часто неефективним через невдало обрану стартову точку пошуку параметрів моделі. Тому загалом навчання моделей ознакового опису здійснюють у два етапи. Перший етап полягає в попередньому навчанні моделі замість ініціалізації параметрів моделі випадковими значеннями із заданого інтервалу рівномірного розподілу. Метою цього етапу є синтез субоптимального екстрактора ознак, адаптація якого для розв'язання конкретних задач потребуватиме мінімальних ресурсів. При цьому попереднє навчання моделі ознакового подання може здійснюватися з учителем або без учителя. Другий етап полягає в точному налаштуванні екстрактора ознак і вирішувальних правил як єдиної моделі з метою максимального наближення до глабального оптимуму функції ефективності моделі.

Найпростішим прикладом навчання екстрактора ознак з учителем є запозичення нижніх шарів попередньо навченої моделі на подібній задачі згідно з принципом перенесення знань (Transfer Learning). Однак варто розглянути ще три підходи попереднього навчання ієрархічного екстрактора ознак з учителем:

a) багатошаровий стекінг;
b) «жадібне» пошарове навчання з учителем;
c) навчання сіамської моделі.

Основна ідея багатошарового стекінгу полягає в навчанні ансамблю моделей для формування на виході закодованої версії оригінальних навчальних даних (рис. 1.6 a).

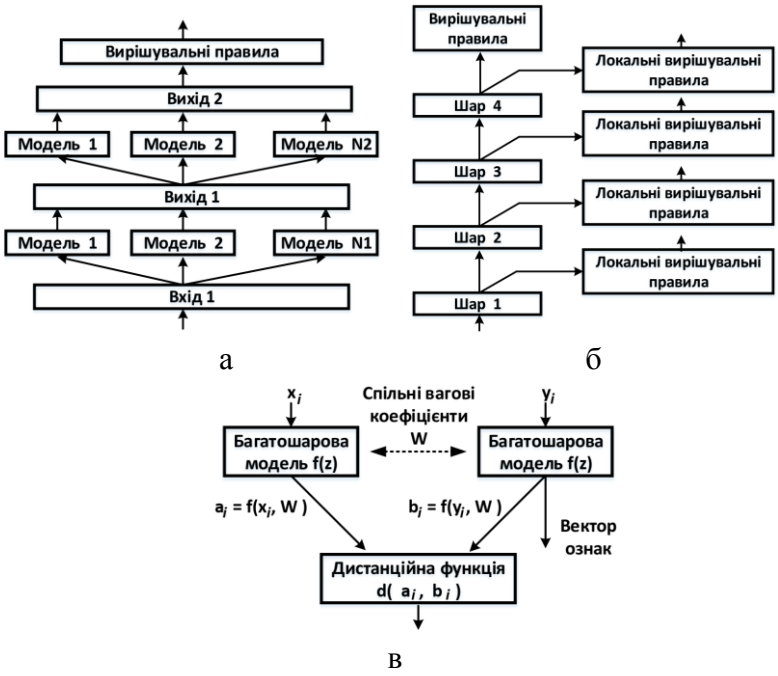


Рисунок 1.6 – Структура моделей навчання ієрархічного екстрактора ознак з учителем: а – багатошаровий стекінг; б – «жадібне» пошарове навчання; в – навчання сіамської моделі

Виходи моделей першого шару розглядають як ознаковий опис для моделей другого шару (метамоделі). При цьому розмітка даних для навчання моделей другого шару залишається такою самою як і для першого. Аналогічно і для наступних шарів. Як було показано в працях [15] для уникнення ефекту перенавчання в багатошарових стекованих моделях необхідно уникати затухання диверсності (рі-

зноманіття) моделей кожного нового шару. Під диверсністю, зазвичай, розуміють кількісне оцінювання різниці допущення однакових помилок серед моделей в ансамблі. Тому диверсність інколи оцінюють на основі виразу

$$\overline{D} = E - \bar{E},$$

де $\overline{D}$ – усереднена оцінка диверсності моделей ансамблю;

E – загальна помилка рішень, ухвалених за результатами голосування моделей ансамблю;

$\bar{E}$ – середня помилка рішень, ухвалених індивідуальними моделями ансамблю.

Існує багато визначень диверсності та способів її оцінювання. Один із підходів до оцінювання диверсності полягає в аналізі ймовірностей одночасної незгоди моделей ансамблю під час прийняття рішень. В інших підходах диверсність моделей ансамблю розглядають як оцінку структурної різноманітності в ансамблі, що може бути виміряна ентропійними та інформаційними мірами [15].

Одним із методів забезпечення диверсності є використання в ансамблі моделей, що ґрунтуються на різних принципах. Однак також популярним підходом до забезпечення диверсності є введення елемента рандомізації [16]:

– маніпуляції з навчальними вибірками моделей (формування випадкових підвибірок, ініціалізація вагових коефіцієнтів навчальних зразків);

– вибір різних підпросторів (випадкова вибірка ознак);

– маніпуляції з параметрами навчання (ініціалізація

нейронної мережі випадковими числами, введення регуляризуючого штрафу за кореляцію результатів мережі з результатами інших мереж ансамблю);

– маніпуляції з поданням вихідних значень моделей (подання класів кодами, що виправляють помилки, або випадкові зміни міток класів у деяких навчальних зразках).

Методи стекінгу мають потенціал для розвитку, однак досі існують проблеми науково-методологічного характеру, пов'язані з невизначеністю під час вибору методів оцінювання та способів забезпечення диверсності моделей ансамблю. Проте в працях [17] було показано, що методи стекінгу краще піддаються теоретичному аналізу ніж традиційні методи глибоких нейронних мереж.

«Жадібне» пошарове навчання з учителем ієрархічної моделі даних потребує додавання допоміжної моделі – локальних вирішувальних правил (рис. 1.6 б). Перший шар мережі ініціалізують випадковими числами, а до його виходу додають вирішувальні правила та здійснюють навчання одержаної моделі. Потім нижній шар «заморожують», тобто залишають усі параметри фіксованими й незмінними. До виходів першого шару під'єднують входи наступного шару, до якого так само додають вирішувальні правила та здійснюють їх сумісне навчання. Процес повторюється для наступних шарів. У праці [18] було запропоновано додати до функціоналу якості вирішувальних правил складову помилки реконструкції даних. Тобто процес навчання кожного шару може бути з частковим залученням учителя, що покращує узагальнювальну здатність моделі. Після закінчення пошарового навчання всі локальні

(міжшарові) вирішувальні правила видаляються. При цьому одержана модель може бути покращена методом точного налаштування за схемою «з кінця в кінець».

У працях [19] було запропоновано архітектуру так званих сіамських нейронних мереж для побудови інваріантного подання вхідних даних та зниження розмірності. Сіамська мережа складається з двох екстракторів ознак зі спільними параметрами (рис. 1.6 в). Параметри екстрактора ознак вважають оптимальними, якщо для пари семантично подібних зразків вихід мережі має близьке до нуля значення, а для пари семантично відмінних зразків – значення виходу, близьке до одиниці. Тобто навчання сіамських мереж можна вважати навчанням метрики подібності (metric learning). Навчена сіамська мережа може відігравати роль дистанційної метрики для «лінивих» алгоритмів побудови вирішувальних правил, таких як метод k-найближчих сусідів. На цьому принципі ґрунтуються алгоритми навчання з одного погляду (one-shot learning). Також дані мережі можуть розглядатися як адаптивне ядро для ядерних алгоритмів побудови вирішувальних правил. Водночас задача побудови функції подібності (similarity function) є еквівалентною задачі побудови простору інваріантних ознак. Тому навчений екстрактор можна також використовувати окремо, без близнюка.

Навчальна вибірка сіамської нейронної мережі складається з семантично подібних пар зразків S та семантично відмінних пар зразків D, тобто

$S = \{(x_1^{(i)}, x_2^{(i)}) : x_1^{(i)}$ *та* $x_1^{(i)}$

є семантично подібними з маркуванням $y^{(i)} = 0\}$,

$D = \{(x_1^{(i)}, x_2^{(i)}) : x_1^{(i)}$ *та* $x_1^{(i)}$

є семантично відмінними з маркуванням $y^{(i)} = 1\}$.

Функція втрат для сіамської мережі може мати вигляд

$$L(x_1^{(i)}, x_2^{(i)}) = y(x_1^{(i)}, x_2^{(i)}) \log\, p(x_1^{(i)}, x_2^{(i)}) +$$
$$(1 - p(x_1^{(i)}, x_2^{(i)})) \log(1 - p(x_1^{(i)}, x_2^{(i)})) + \lambda^T |w|^2,$$

де $(x_1^{(i)}, x_2^{(i)})$ – пара зразків міні-пакета навчальної вибірки;

$y(x_1^{(i)}, x_2^{(i)})$ – вектор маркування міні-пакета;

λ^T – регуляризаційні ваги.

Також у сіамських нейромережах набула поширення триплетна функція втрат із різноманітними стратегіями формування триплетів – трійок вхідних зразків [19].

Для попереднього навчання ознакового подання без учителя найбільшого поширення набули такі підходи:

– пошарове навчання на основі автоенкодерів;

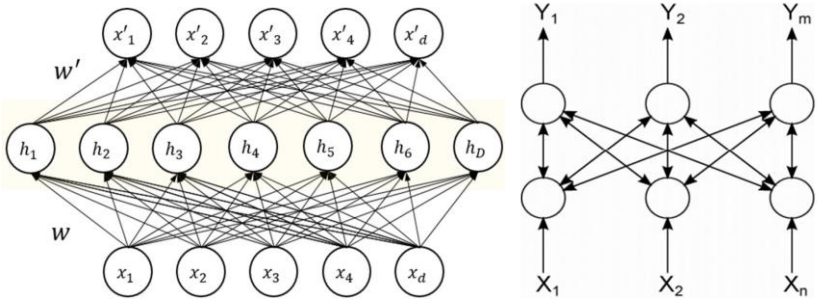
– пошарове навчання на основі обмежених машин Больцмана;

– використання методів пошарового розрідженого кодування;

Автоенкодер (автоасоціатор) – спеціальна архітектура нейронних мереж, що дозволяє застосовувати навчання без

учителя з використанням методу зворотного поширення помилки. Найпростіша архітектура автоенкодера наведена на рис. 1.7 а – мережа прямого поширення без зворотних зв'язків містить вхідний шар, проміжний шар із K нейронів та вихідний шар. Головна мета навчання автоенкодера – досягти того, щоб вхідний вектор ознак викликав відгук мережі, що дорівнює вхідному вектору.

Обмежена машина Больцмана є породжувальною стохастичною нейронною мережею, що навчається формувати деякий імовірнісний розподіл даних на своїх входах [20]. Обмежена машина Больцмана складається з видимого та прихованого шару. Кожен нейрон має двоспрямований зв'язок з іншими нейронами в сусідньому шарі (рис. 1. 7 б).



а б

Рисунок 1.7 – Нейромережеві моделі для навчання без учителя: а – автоенкодер; б – обмежена машина Больцмана

Провідна ідея навчання полягає в максимально точному відновленні розподілу вхідних даних на основі станів

нейронів прихованого шару. Це еквівалентно максимізації функції логарифмічної правдоподібності розподілу вхідних даних методом модифікації синаптичних зв'язків нейронної мережі. Хінтон запропонував використовувати метод контрастної дивергенції (Contrastive divergence, CD), що ґрунтується на семплюванні Гіббса, для навчання такої мережі [20].

Стекування автоенкодерів або обмежених машин Больцмана, відбувається за допомогою з'єднання виходу навченого прихованого шару із входом автоенкодера нового шару, що навчається. Основним недоліком автоенкодерів та обмежених машин Больцмана є потреба великого обсягу навчальних зразків та обчислювальних ресурсів для успішного навчання. При цьому є необхідність значного нарощування глибини моделі за наявності семантично подібних зразків вибірки зі значними відмінностями у векторі ознак.

У працях [14] запропоновано багато методів розрідженого кодування, які вхідному вектору x ставлять у відповідність приховане подання даних h (вектор випадкових змінних або вектор ознак) через лінійний взаємозв'язок W, що називається словником. При цьому в методах розрідженого кодування використовують спеціальне обмеження для забезпечення саме розрідженої активації h під час кодування x. Розріджене кодування можна розглядати як задачу відновлення вектора ознак, що асоційований зі значенням входу x згідно з виразом

$$h^* = f(x) = \arg\min_h \|x - Wh\|_2^2 + \lambda \|h\|_1. \quad (1.2.1)$$

Навчання словника W може бути виконано в процесі мінімізації середньоквадратичної помилки реконструкції

$$W^* = \arg\min_W \sum_t \left\| x^{(t)} - W h^{*(t)} \right\|_2^2, \qquad (1.2.2)$$

де $x^{(t)}$ – t-й навчальний зразок;

$h^{*(t)}$ – відповідний розріджений код.

Розріджене кодування також має ймовірнісну інтерпретацію, згідно з якою кодування полягає у відновленні максимуму апостеріорної ймовірності h, тобто

$$h^* = \arg\max_h p(h \mid x). \qquad (1.2.3)$$

Процес навчання словника W в межах ймовірнісної інтерпретації може здійснюватися як максимізація правдоподібності даних, одержаних шляхом використання методу максимізації апостеріорної ймовірності h^*, тобто

$$W^* = \arg\max_W \prod_t p(x^{(t)} \mid h^*(t)). \qquad (1.2.4)$$

У працях [21, 14] було запропоновано здійснювати навчання словника W (dictionary learning) окремо на основі автоенкодерів, обмежених машин Больцмана, методів кластер-аналізу та векторного квантування. Водночас у разі використання автоенкодерів або обмежених машин Боль-

цмана необхідно до функції втрат уводити штраф за нерозрідженість. За умови використання методів векторного квантування необхідно забезпечити оптимальну взаємну узгодженість (ступінь неортогональності) елементів словника W для ефективного кодування даних.

Після навчання словника W кодування зразка даних x у результуючий вектор ознак h здійснюють на основі функції розрідженого кодувальника $f(x;W)$. Популярними алгоритмами реалізації цієї функції є алгоритми узгодженого переслідування (matching pursuit), ортогонального узгодженого переслідування (orthogonal matching pursuit), мішка переслідувачів (bag of pursuits) та ін.

Розріджене кодування навіть із надлишковим словником дозволяє здійснювати ймовірнісне виведення під час пошуку h^* шляхом вибору найбільш оптимальних базисів та обнулення решти, незважаючи на те, що вони мають високий ступінь кореляції зі входом. Цієї властивості не має ні в автоенкодерів, ні в обмежених машинах Больцмана. На відміну від автокодувальників та обмежених машин Больцмана вектор ознак (код) в алгоритмі розрідженого кодування вільно змінюється для кожного зразка даних. Платою за це є наявність додаткового циклу для оптимізації h^*, що збільшує обчислювальні витрати під час екстракції розрідженого ознакового опису.

Для підвищення ефективності розрідженого кодування було запропоновано метод прогнозувальної розрідженої декомпозиції (Predictive Sparse Decomposition), де ресурсно-затратний нелінійний ітераційний процес пошуку h (кодування) замінюється на швидку неітеративну апроксима-

цію під час розпізнавання [14, 22]. Основна ідея цього методу полягає у використанні такого критерію навчання

$$J_{PSD} = \sum_t \lambda \left\| h^{(t)} \right\|_1 + \left\| x^{(t)} - W h^{(t)} \right\|_2^2 + \left\| h^{(t)} - f_\alpha(x^{(t)}) \right\|_2^2,$$

де $x^{(t)}$ – вхідний вектор t-го навчального зразка;

$h^{(t)}$ – оптимізований прихований вектор (подання) даних для t-го навчального зразка;

$f_\alpha(\cdot)$ – функція кодувальника, яка у найпростішому варіанті має вигляд

$$f_\alpha(x^{(t)}) = \tanh(b + W^T x^{(t)})$$

У праці [22] було показано, що побудову швидкого апроксимувального кодувальника $f_\alpha(\cdot)$ можна здійснити на основі операції усадження (shrinkage), ансамблю дерев рішень та інших обчислювально ефективних моделей. В інших працях [14, 21] було показано, що критерій розрідженого кодування може бути досить негладкою та недиференційованою функцією, що ускладнює повну оптимізацію моделі подання даних. Тому багато методів спрямовані на згладжування результуючих кодів розрідженого кодування, щоб забезпечити сумісне навчання етапу розрідженого кодування з іншими етапами обчислення глибокої архітектури.

Для ознакового опису послідовностей (тобто процесів,

розгорнутих у часі) необхідно детектувати вхід і вихід із контекстів для коректного подання поточних даних. Рекурентні та згорткові нейронні мережі є лідерами у сфері аналізу послідовностей. Рекурентні мережі рекурсивно аналізують вхідний потік, тобто здійснюють його сканування в часі, з метою запам'ятовування, забування чи передавання для наступного аналізу розпізнаних контекстів. Однак рекурентні мережі мають ряд таких істотних проблем, як чутливість до перших зразків даних на вході, складний характер динаміки навчання з можливим переходом до хаотичної поведінки та складність розпаралелювання внаслідок послідовної організації функціонування моделі. Згорткові мережі використовують локальні рецептивні поля для сканування сигналів з 1D-, 2D- чи 3D-топологіями. Останні дослідження показали, що аналіз послідовностей згортковими мережами в межах 1D-топології з модифікованими рецептивними полями, що називають дірявими (dialated), перевершують рекурентні мережі як в оперативності, так і в ємності [23]. А застосування механізму уваги (attention) у вигляді додаткового модуля, що обчислює маску для фокусування на важливих просторових та канальних ознаках і придушення неважливих ознак, дозволяє додатково підвищити потужність ознакового подання. При цьому згорткові моделі подання даних є зручними для застосування різноманітних технік і протоколів навчання.

Точне налаштування попередньо навчених моделей зазвичай здійснюють методами зворотного поширення помилки за умов диференційованості та гладкості функцій,

що описують компоненти й функціонал якості цих моделей. Водночас інтенсивність оновлення параметрів під час точного налаштування, зазвичай, на порядок нижча, ніж у режимі попереднього навчання з метою уникнення втрати накопиченого моделлю досвіду. Однак за умов недиференційованості чи негладкості функції, що описує модель, більш ефективними у використанні є метаевристичні популяційні та траєкторні методи пошукової оптимізації.

Популяційні алгоритми дозволяють забезпечити оптимальне співвідношення між збіжністю та дослідженням простору пошуку. Проте популяційні алгоритми є надто ресурсозатратними для оптимізації великих моделей, оскільки оперують популяцією модифікацій цієї моделі, кожен агент якої повинен обробити пакет навчальних та валідаційних даних. Більш обчислювально ефективними можна вважати траєкторні алгоритми, що на кожній ітерації оперують одним новим рішенням, подібно до методу градієнтного спуску. Серед траєкторних методів найбільш популярним є використання алгоритму симуляції відпалу (Simulted Annealing), алгоритму сходження на пагорб (Hill Climbing) або їх модифікацій. Траєкторні мета-евристичні алгоритми є більш придатними для точного налаштування, оскільки для обмеженого кола пошуку достатньо одного агета замість їх популяції.

Отже, сучасні підходи до побудови ознакового опису даних ґрунтуються на принципах розрідженого, розподіленого та ієрархічного кодування інформації. При цьому навчання ознакового опису, переважно, здійснюють у два етапи: попереднє навчання з учителем або без учителя та

точне налаштування. Етап попереднього навчання спрямований на максимально ефективне використання всієї доступної інформації з метою забезпечення достатньої інформативності чи інваріантності ознакового опису для його повторного використання та побудови простих вирішувальних правил. Етап точного налаштування спрямований для адаптації ознакового опису до конкретної задачі й полягає у використанні градієнтних чи мета-евристичних алгоритмів оптимізації параметрів в обмеженій області пошуку.

1.3. Аналітичний огляд моделей і методів побудови вирішувальних правил

Вирішувальне правило (вирішувальна функція) f для задач класифікаційного чи регресійного аналізу використовують для відображення сформованого ознакового опису X в конкретні рішення Y на виході моделі, тобто $f: X \rightarrow Y$. Вирішувальне правило може бути як однорівневим, так і ієрархічним. В однозадачних моделях основні перетворення даних відбуваються в екстракторі ознак, тому вирішувальне правило має найпростішу, часто однорівневу, структуру. У моделях, де ознаковий опис передбачено використовувати для багатьох задач, вирішувальні правила можуть мати складнішу структуру, що містить один чи більше прихованих шарів.

Загалом як критерій ефективності навчання вирішувальних правил використовують емпіричний ризик Q, що характеризує середню помилку рішень на вибірці даних

$$Q = \frac{1}{n}\sum_{i=1}^{n} L(o_i, y_i),$$

де n – кількість зразків навчальної вибірки;

$L(\cdot)$ – функція втрат, що характеризує помилку для i-го зразка x_i;

o_i – фактичний вихід моделі аналізу даних для i-го зразка x_i;

y_i – очікуваний вихід моделі аналізу даних для i-го зразка x_i.

У задачі регресійного аналізу функція втрат може бути розширена регуляризаційною складовою $R(w)$ для збереження працездатності моделі й на тестовій вибірці

$$L(o_i, y_i) = J(o_i, y_i) + R(w),$$

де $J_i(o_i, y_i)$ – помилка моделі для i-го зразка x_i, що може бути обчислена за однією з формул

$J(o_i, y_i) = (o_i - y_i)^2$ або

$J(o_i, y_i) = |o_i - y_i|$;

$R(w)$ – функція регуляризації (функція вагових коефіцієнтів нейронів w), що може бути обчислена за однією з таких формул

$R(w)_{LASSO} = \lambda_1 |w|,$

$$R(w)_{RIDGE} = \lambda_2 |w|^2,$$

$$R(w)_{ELASTICNET} = \lambda_1 |w| + \lambda_2 |w|^2,$$

де λ_1, λ_2 – коефіцієнти впливу регуляризованої складової.

Прийнято розрізняти задачі двокласової (бінарної) класифікації (binary classification), коли на виході моделі лише два варіанти, та багатокласової класифікації (multi-class classification), коли категорій розпізнавання більше двох. При цьому задачу двокласової класифікації, де неможливо зібрати достатньо велику кількість об'єктів одного з класів або неможливо здійснити однозначне подання одного з класів, часто називають задачею однокласової класифікації (one-class classification). Однокласова класифікація актуальна під час вирішення задач детектування аномалій (anomaly detection), виявлення викидів у даних (outlier detection), а також під час розпізнавання новизни в даних (novelty detection). Крім того, багатокласову класифікацію, де зразок із навчальної вибірки може одночасно належати декільком категоріям, називають багатозначною (політематичною) класифікацією (multi-label classification). Багатозначну класифікацію зазвичай будують на основі серії бінарних класифікаторів.

У задачах класифікації вихід моделі часто інтерпретують як імовірність належності до класів розпізнавання. Для цього вихід моделі нормують до діапазону [0, 1] за допомогою softmax-функції, що для випадку двокласової та багатозначної класифікації вироджується до сигмоїдної функції. Найбільшого поширення в задачах класифікації

набули такі функції втрат:

$$L_H(o_i, y_i) = \max(0, 1 - y_i o_i) + R(w),$$

$$L_L(o_i, y_i) = \log(1 + \exp(-y_i o_i)) + R(w),$$

$$L_{CE}(o_i, y_i) = \sum_{c=1}^{M} y_{i,c} \log(o_{i,c}) + R(w),$$

$$L_{KL}(o_i, y_i) = \sum_{c=1}^{M} y_{i,c} \log\left(\frac{y_{i,c}}{o_{i,c}}\right) + R(w),$$

де $L_H(\cdot)$, $L_L(\cdot)$, $L_{CE}(\cdot)$, $L_{KL}(\cdot)$ – кусково-лінійна (hinge loss) для бінарної класифікації за принципом максимального відступу (maximum margin), логістична (logistic loss) для бінарної класифікації, крос-ентропійна (cross-entropy loss) та Кульбака-Лейблера (Kullback-Leibler divergence loss) функції втрат відповідно;

M – кількість класів розпізнавання.

Багатокласову задачу завжди можна звести до серії двокласових. Методи розв'язання задач двокласової класифікації добре розроблені. При цьому часто обчислювально ефективніше працювати не з монолітним багатокласовим класифікатором, а з еквівалентною множиною двокласових класифікаторів. До методів зведення багатокласової класифікації до серії двохкласових належать двійкове кодування кодами, що коригують помилки (Error Correcting Output Codes, ECOC), «кожен проти всіх» (one-

against-all), «кожен проти кожного» (one-against-one), «турнір на вибування» (survivor), дихотомія (dichotomy), «кожен сам за себе» (everyone for himself) [24].

Під час використання кодів, що коригують помилки, номер класу записують у вигляді *k*-значного двійкового числа. Для цього здійснюють навчання *k*-класифікаторів, кожен із яких розпізнає один із *k* розрядів номера класу. За результатами розпізнавання вхідного вектора кожним із k класифікаторів однозначно відновлюють номер класу, до якого він належить. Якщо окремі класифікатори помиляються, то номер класу відновлюють методом заміни одержаного номера номером класу, що найближчий до одержаного за метрикою Хеммінга (рис. 1.8) [24].

На рисунку 1.8 показано кодову матрицю, де чорні й білі комірки позначають мітки 0 або 1 для зразків відповідних класів на етапі навчання бінарних класифікаторів. У режимі екзамену цю матрицю використовують для порівняння прогнозованого коду з кодами класів. Ефективна кодова матриця повинна забезпечувати найбільшу відстань Хеммінга як між рядками кодової матриці, так і між стовбцями. Відстань між рядками забезпечує можливість самовиправлення помилок, а відстань між стовбцями забезпечує некорельованість результатів кожного з бінарних класифікаторів, що навчаються. При цьому існує багато способів формування номера класу для кодової матриці, однак саме коди, що виправляють помилки (самокоригувальні коди) забезпечують кодування з необхідними властивостями. Поширеним способом формування самокоригувальних кодів є використання матриць і кодів Адамара [24].

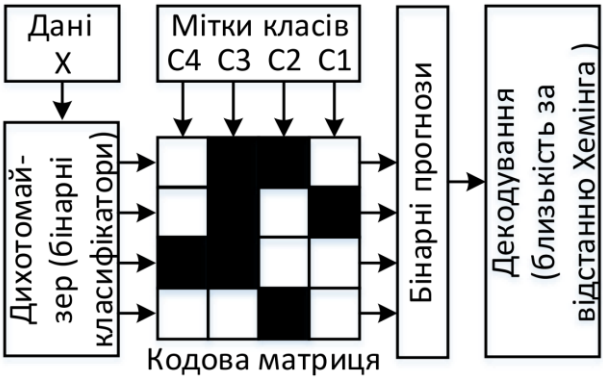


Рисунок 1.8 – Ілюстрація до схеми багатокласової класифікація з використанням двійкового кодування номера класу кодами, що виправляють помилки

Основним недоліком методу ECOC є ігнорування структури класів під час побудови кодової матриці без можливості оптимізації коду кожного класу в процесі навчання. Крім того, відсутність кодового радіуса для кожного номера класу, що вказує на кратність помилок, які можуть бути виправлені, ускладнює виявлення викидів або новизни в даних.

Побудова бінарних класифікаторів згідно з принципом «кожен проти всіх» полягає в навчанні для кожного окремого класу моделі, яка вважає зразки відповідного класу позитивними, а зразки решти класів – негативними. Тому під час розпізнавання віднесення вхідного зразка даних до одного з класів відбувається за максимумом вихідного значення бінарних класифікаторів.

Принцип «кожен проти кожного» дозволяє замість навчання *M* складних класифікаторів, що відрізняють кожен клас від решти класів, здійснювати побудову *M(M-1)/2* простіших класифікаторів, які розрізняють лише пари класів між собою. Для кожного вхідного зразка *x* обчислюють *M(M-1)/2* значень *fij (x) = –fji(x),*
де *fij (*)* – функція моделі бінарного класифікатора, що розрізняє класи *i* та *j*. Розпізнавання відбувається за максимальним значенням обчислених M сум

$$f_i(q, x) = \sum_{j \neq i} q(f_{i,j}(x)), i = \overline{1, M},\ j = \overline{1, M},$$

де $q(\cdot)$ – деяка монотонна неспадна функція, наприклад, логістична функція або функція Хевісайда.

Для зменшення кількості обчислень під час розпізнавання бінарні класифікатори, навчені за принципом «кожен проти кожного», можуть використовуватися за принципом «турнір на вибування». У межах цього підходу для вхідного вектора *x* влаштовується змагання серед *M* класів. Турнір між двома класами полягає в застосуванні до вхідного вектора *x* класифікатора, що розрізняє ці два класи бінарного класифікатора. Клас до якого класифікатор відніс вектор *x* вважають переможцем і він продовжує брати участь у змаганні, а альтернативний клас вибуває. Після *M-1* змагань, тобто роботи *M-1* бінарних класифікаторів, всі класи крім «чемпіона» вибувають, і саме йому приписують вектор *x*.

Принцип дихотомії часто використовують для побудо-

ви ієрархічної структури класифікатора. Дихотомічний підхід полягає в розбитті множини всіх *M* класів на дві підмножини – «надкласи» – і в навчанні класифікатора, що визначає належність вхідного вектора *x* до надкласу. Ця процедура повторюється для кожного надкласу. Урешті-решт буде одержано двійкове дерево, де кожна із *M-1* вершин розгалуження відповідає класифікатору, а кожен з *M* листків відповідає окремому класу розпізнавання. У межах цього підходу можна отримати результат розпізнавання не більше ніж за *M-1* кроків (а для збалансованого дерева за *log2M* кроків). Однак ефективність класифікатора на тестовій вибірці залежатиме від обраної ієрархічної структури класів, тому вона потребує додаткової оптимізації.

Реалізація принципу «кожен сам за себе» передбачає побудову для кожного класу однокласового класифікатора. При цьому розпізнавання належності вхідного вектора *x* відбувається за максимальним значенням на виході побудованих однокласових класифікаторів. Однак на практиці цей підхід рідко забезпечує побудову високодостовірних вирішувальних правил.

Під час аналізування образів часто користуються гіпотезою компактності, яка полягає в тому, що подібним в просторі ознак об'єктам вибірки відповідають подібні мітки. Зокрема в задачах класифікації ця гіпотеза полягає в тому, що класи утворюють компактно локалізовані підмножини у просторі об'єктів вибірки. При цьому для формалізації поняття «подібності» вводять функцію (метрику) відстані *d(x1, x2)* в *N*-вимірному просторі об`єктів вибірки. Методи, що ґрунтуються на аналізуванні подібності

об’єктів, часто називають метричними, навіть у тих випадках, коли функція d не задовольняє всіх аксіом метрики (наприклад, аксіому трикутника). До метричних методів побудови вирішувальних правил відносять метод найближчого сусіда (nearest neighbor, NN) та метод k-найближчих сусідів (k-nearest neighbors, kNN). Метричні алгоритми здійснюють локальну апроксимацію вибірки, при якій обчислення відкладають до моменту, доки не стане відомим вхідний об’єкт. Тому більшість метричних алгоритмів відносять до методів лінивого навчання (lazy learning).

Метод найближчого сусіда є найбільш простим метричним методом, згідно з яким вхідному об’єкту присвоюють мітку найближчого до нього об’єкта навчальної вибірки. Навчання в межах цього методу полягає в елементарному запам’ятовуванні навчальної вибірки

$$\{< x_i^{(j)}, y^{(j)} > | \quad j = \overline{1,n}; i = \overline{1,N}\},$$

де n – обсяг навчальної вибірки;

N – кількість ознак розпізнавання.

Єдиною перевагою цього методу є простота реалізації. Однак недоліків набагато більше. По-перше, наявності викидів у навчальній вибірці призводить до нестійкості та похибок. По-друге відсутні параметри, які можна було налаштувати за навчальною вибіркою. При цьому алгоритм повністю залежить від успішності вибору дистанційної міри $d(x_1, x_2)$, що є єдиним гіперпараметром алгоритму, який можна налаштувати [25].

У межах методу *k*-найближчих сусідів із метою згладжування шумового впливу викидів здійснюють аналіз вхідного об'єкта методом голосування за *k* найближчими сусідами навчальної вибірки $\{x^{(j)} \mid j = \overline{1,k}\}$. На рисунку 1.9 показано узагальнену схему методу k-найближчих сусідів, де перший етап аналізу полягає в обчисленні відстані $d(x, x_j)$ вхідного вектора *x* до навчальних векторів $\{x^{(j)}\}$. До того ж указані відстані можуть використовувати як для відбору сусідів так і для визначення ваги їхнього голосу залежно від їх віддаленості від вхідного вектора, $w_j = 1/d(x, x^{(j)})$.

Метод k-найближчих сусідів придатний для розв'язання як задач регресійного, так і класифікаційного аналізу. При цьому прийняття результуючого рішення для регресійного аналізу y_{reg} та класифікаційного аналізу y_{cls} відбувається відповідно до формул:

$$y_{reg} = \frac{\sum_{j=1}^{k} w_j y^{(j)}}{\sum_{j=1}^{k} w_j}, \quad y_{cls} = \arg\max_{m} \sum_{j=1}^{k} w_j \cdot \delta(y_m, y^{(j)}),$$

де $\delta(\cdot)$ – дельта-функція, що дорівнює 1, якщо аргументи однакові між собою, і дорівнює 0 – в протилежному випадку;

m – індекс класу розпізнавання.

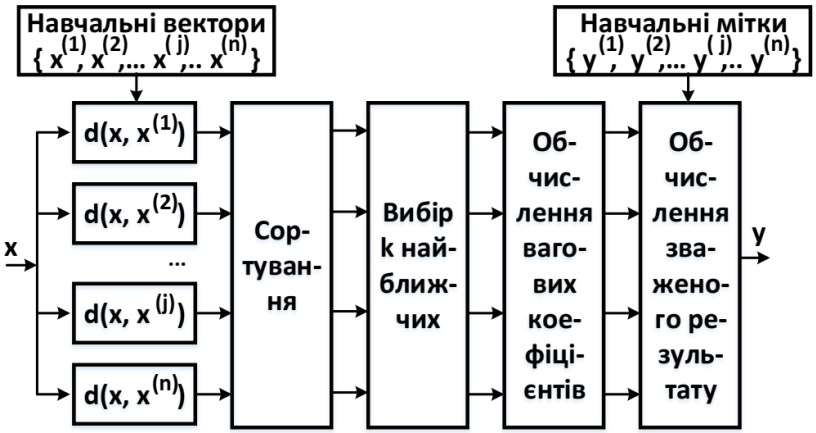


Рисунок 1.9 – Узагальнена схема алгоритму
k-найближчих сусідів

Оптимальне значення параметра k визначають за критерієм ковзного контролю з виключенням об'єктів по одному (leave-one-out, LOO). Однак пошук найближчих сусідів методом обчислення відстані до всіх об'єктів навчальної множини може бути надто довгим, якщо розмір вибірки має великий обсяг. Тому набули поширення алгоритми пошуку приблизних найближчих сусідів (approximate nearest neighbour), що використовують ймовірнісні структури даних. До таких алгоритмів відносяться MinHash, R-дерево, BallTree, KDTree, FAISS, Locality-sensitive hashing (LSH), Annoy та LSH Forest [25].

Вибір міри подібності для побудови вирішувальних правил у межах метричних методів аналізу даних досі залишається неоднозначним. Найбільшого поширення набу-

ли такі метрики, як Евклідова, Манхеттенська, Степенева та Журавльова [26]. До того ж для вибору чи конструювання міри подібності (метрики) необхідно враховувати властивості об'єктів навчальної вибірки. Деякі ознаки можуть бути категорійними, що потребує їх перекодування в числовий формат на основі таких методів як Dummy-кодування, ортогональне поліноміальне кодування (Orthogonal Polynomial Coding), Helmert-кодування, пряме та зворотне різницеве кодування (Forward/Backward Difference Coding) та інші. Водночас більшість метрик мають адитивний характер, що потребує шкалування діапазону значень ознак. Проте адитивний характер метрик робить їх непридатними для випадку високої розмірності вхідного простору ознак унаслідок проблеми «прокляття розмірності».

Одним із найбільш ефективних методів побудови метрики відстані для метричних методів синтезу вирішувальних правил є використання моделей сіамських нейронних мереж (Siamese Network), структура яких показана на рисунку 1.6 в [19]. Навчання сіамських нейронних мереж дозволяє сформувати таку метрику відстані, що відстань між об'єктами вибірки із подібними мітками буде меншою, а відстань між об'єктами вибірки з неподібними мітками – більшою. При цьому кожний навчальний зразок сіамської мережі є парою об'єктів початкової навчальної множини. Тому за рахунок комбінування пар штучно розширюється навчальна множина, а модель акумулюватиме більше інформації про взаємне відношення об'єктів початкової навчальної множини. Отже, в процесі машинного навчання

сформована метрика відстані максимально враховуватиме особливості об'єктів навчальної множини та дозволятиме використання меншої кількості сусідніх зразків. До того ж у задачі класифікації як сусідні зразки можливо використовувати випадкову підвибірку кожного класу розпізнавання.

Найпростішими вирішальними правилами є лінійні, що ставлять у відповідність вхідному N-вимірному спостереженню x вихідну змінну через лінійну комбінацію його ознак. Одночасно вирішувальні правила для задачі регресійного та класифікаційного аналізу відповідно мають такий вигляд:

$$y_{reg}(x) = \sum_{i=1}^{N} w_i x_i + w_0 = w^T x,$$

$$y_{cls}(x) = \begin{cases} +1, \textit{ якщо } 1/(1+\exp(-w^T x)) > 0,5; \\ -1, \textit{ в протилежному разі.} \end{cases}$$

Навчання лінійної регресійної моделі здійснюють, зазвичай, на основі методу найменших квадратів (Least Mean Square) за алгоритмом градієнтного спуску, псевдо-інверсії Мура – Пенроуза або іншими методами. Основними припущеннями під час побудови лінійних вирішувальних правил є лінійність залежності, нормальність і/або незмінність дисперсії. Навчання класифікаційних вирішувальних правил, які ще називають логістичною регресією, здійснюють на основі градієнтного спуску з крос-ентропійною функцією втрат. Сигмоїдна функція в класифікаційному вирішу-

льному правилі необхідна для відображення вхідного вектора у відповідне значення ймовірності належності до класу розпізнавання. Основними припущеннями під час синтезу лінійних класифікаційних вирішувальних правил є можливість розділення об'єктів різних класів у просторі ознак однією лінійною межею (гіперплощиною), яку ще називають лінійним дискримінантом. У багатьох ситуаціях підготований ознаковий опис спостережень задовольняє вказаним припущенням, що забезпечує придатність для практичного використання синтезованих лінійних вирішувальних правил.

Для підвищення ефективності вирішувальних правил можна використовувати додатковий прихований шар з r нейронів, лінійна комбінація виходів яких формуватиме один з виходів моделі. На рисунку 1.10 показано узагальнену модель мережі прямого поширення з одним прихованим шаром, де функція $K(\cdot)$ загалом є нелінійною функцією проекції вхідного зразка x на вектор вагових коефіцієнтів штучного нейрона w_j^1.

До мереж прямого поширення з одним прихованим шаром відносять машину опорних векторів (support vector machine, SVM), машину екстремального навчання (extreme learning machine, ELM), мережу радіально базисних функцій (Radial basis function network, RBF-network), машину релевантних векторів (Relevance Vector Machine, RVM) та звичайний перцептрон з одним прихованим шаром.

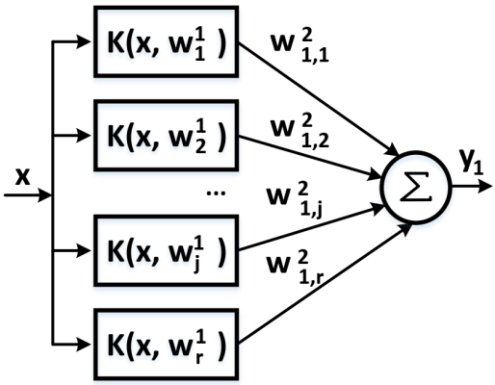


Рисунок 1.10 – Узагальнена структура мережі прямого поширення з одним прихованим шаром

У методі опорних векторів як вектори вагових коефіцієнтів прихованого шару w_j^1 використовують підмножину навчальних зразків (опорні вектори), що робить його частково лінивим методом навчання. Однак як коефіцієнти $w_{1,j}^2$ лінійної комбінації виходів прихованого шару в методі опорних векторів використовуються множники Лагранжа, пошук яких здійснюють за допомогою методів розв'язання задач квадратичної оптимізації, що забезпечують пошук глобального оптимуму. Цільовою функцією є структурний ризик, мінімізація якого призводить до максимізації відступу між мітками. При цьому функцію $K(\cdot)$ називають функцією ядра, яку інтерпретують як скалярний добуток векторів у деякому розширеному просторі ознак

$$K(x, x') = \varphi(x)^T \varphi(x'),$$

де $\varphi(x)$ – функція відображення вектора x у розширений простір, у якому забезпечується лінійна розділюваність класів (для класифікації) чи лінійна залежність із цільовою змінною (для регресії).

Вибір функції ядра відбувається відповідно до теореми Мерсера (Мерсер, 1909): функція $K(x, x')$ є ядром лише тоді, коли вона симетрична, $K(x, x') = K(x', x)$ і невід'ємно визначена [27]. Поширеними функціями ядра є лінійне ядро (Linear), сигмоїдне ядро (Sigmoid), поліноміальне (Polynomial) та радіально базисне (RBF). Однак вибір або конструювання ядра для конкретних вхідних даних досі залишається нетривіальною задачею. При цьому SVM модель характеризується значними ресурсними потребами в разі великого обсягу навчальних даних, що ускладнює повний перебір усіх можливих ядер.

У машинах екстремального навчання як вектори вагових коефіцієнтів прихованого шару w_j^1 використовуються випадкові числа згенеровані з рівномірного розподілу даних, або розподілів даних, оцінених за навчальною множиною. При цьому для пошуку коефіцієнтів вихідного шару моделі $w_{1,j}^2$ використовують псевдоінверсію Мура-Пенроуза або процес ортогоналізації Грамма-Шмідта [28]. В обох випадках оптимізацію здійснюють згідно з методом найменших квадратів (Least Mean Square), а процес навчання характеризується високою швидкістю. Однак ре-

зультат навчання значною мірою залежить від вибору кількості нейронів прихованого шару. Тому останні вдосконалення цього методу полягають в інкрементальному додаванні нових нейронів до досягнення необхідного результату. За цих умов функцію $K(\cdot)$ називають функцією активації, що є нелінійною функцією скалярного добутку вхідного вектора та вектора вагових коефіцієнтів нейрона. Ця функція, зазвичай, є диференційована, тому результат навчання завжди можна покращити методом точного налаштування на основі традиційного алгоритму градієнтного спуску.

У мережах радіально-базисних функцій як вектори вагових коефіцієнтів прихованого шару w_j^1 використовують центри кластерів, одержаних у результаті кластер-аналізу навчальної множини. Функція $K(\cdot)$ здійснює нелінійне перетворення відстані вхідного вектора x до центру відповідного кластера w_j^1. Як відстань використовують Евклідову метрику або функцію щільності розподілу. Нелінійне перетворення, зазвичай, відбувається на основі функції Гауса, але можна використовувати й інші симетричні функції, такі як мультиквадратична чи обернена квадратична функції [29]. Вихідний шар нейронної мережі радіально-базисних функцій можна навчати згідно з методом найменших квадратів за алгоритмом градієнтного спуску.

У практиці машинного навчання значного поширення набули вирішувальні правила у вигляді дерев рішень чи їх композицій. Дерево рішень є способом подання правил в

ієрархічній, послідовній структурі, де кожному вхідному вектору x відповідає єдиний вузол, що дає рішення. Навчання дерева рішень полягає в рекурсивному поділі (розщепленні) навчальної множини на підмножини (зазвичай на дві), таким, щоб зменшити деяку цільову функцію, яку ще називають критерієм «забрудненості» (impurity) утворених підмножин. У класичних деревах рішень розщеплення відбувається порогом на значення однією з обраних ознак розпізнавання, проте в нейронних деревах рішень поділ може здійснюватися нейроном чи навіть нейромережею. У результаті навчання вузлам розщеплення приписують оптимальне (оптимальні) значення ознаки (чи функції ознак), з яким (якими) здійснюють порівняння для поділу вибірки чи прийняття рішення. При цьому в задачах регресійного аналізу цільовою функцією є середньоквадратичне відхилення, а в задачах класифікації – помилка класифікації, етропійний критерій або індекс Джині [30]. На рисунку 1.11 та рисунку 1.12 показано приклади регресійного та класифікаційного дерев рішень.

Існує багато алгоритмів побудови дерева рішень, серед яких найбільш популярними є ID3, C4.5, CART, CHAID та MARS [30]. Ці алгоритми відрізняються за критерієм розщеплення, здатністю до обробки пропущенних та різнотипних даних.

У більшості алгоритмів побудови дерева рішень використовують наївний підхід (naive approach), що полягає у використанні гіпотези про статистичну незалежність ознак розпізнавання. Тобто, зазвичай, під час вибору ознак, за якими здійснюють розщеплення, використовують «жа-

дібний» алгоритм. Таке рішення забезпечує локальну оптимальність, однак не гарантує побудови оптимального дерева. Це часто призводить до нестабільності й невисокої точності рішень. Однак дерева рішень часто є ефективними будівельними блоками складних моделей аналізу даних, оскільки мають ряд переваг:

– інтерпретованість моделі;

– здатність до оброблення ознак будь-якого типу;

– здатність до оброблення пропущених даних;

– автоматична побудова структури моделі;

– здатність до розв’язання як задач класифікаційного аналізу, так і задач регресійного аналізу;

– порівняно швидка побудова.

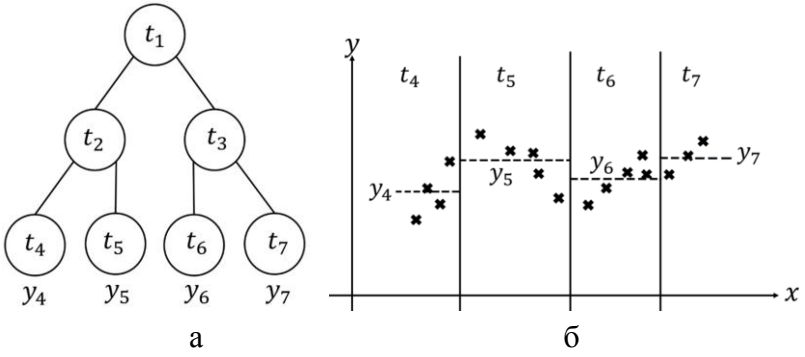
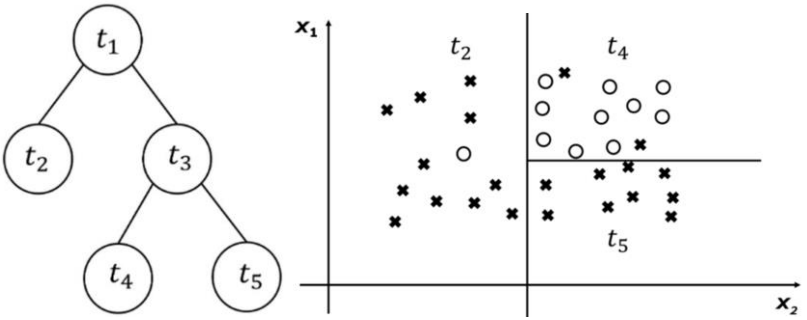


Рисунок 1.11 – Ілюстрація до регресійного дерева рішень: а – приклад дерева; б – приклад результату його роботи в разі однієї ознаки розпізнавання



а б

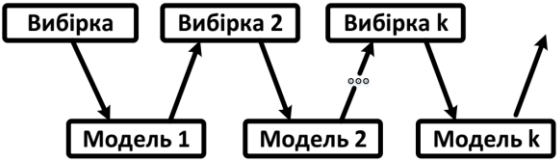
Рисунок 1.12 – Ілюстрація до класифікаційного дерева рішень: а – приклад дерева; б – приклад результату його роботи в разі двох ознак розпізнавання

У кінці 80-х років було опубліковано праці, що пов’язані з проблемами слабкої та сильної здатності алгоритмів до навчання [31]. Під слабкою здатністю до навчання розуміють здатність за поліноміальний час побудувати алгоритм розпізнавання, точність якого дещо вища ніж 50 %. Під сильною здатністю до навчання розуміється здатність побудувати алгоритм, який би міг досягати результатів будь-якої точності. При цьому дослідження показали, що сильна здатність до навчання еквівалентна слабкій, оскільки будь-яку слабку модель можна підсилити за допомогою побудови правильної композиції. Тому для побудови більш складних та ефективних моделей вирішувальних правил використовують метаалгоритми паралельної та послідовної композиції простіших моделей (рис. 1.13).

До алгоритмів паралельної композиції відносять алгоритм багінгу, що полягає у формуванні бутстреп-вибірок (Bootstrap sample) і навчанні на них простих моделей. Водночас у режимі екзамену рішення ухвалюють за результатами голосування ансамблю сформованих моделей. До послідовної композиції відносять алгоритми бустінгу та стекінгу [32]. При цьому під час бустінгу за результатами валідації побудованої моделі змінюються вагові коефіцієнти навчальних зразків так, щоб побудова додаткової моделі компенсувала помилки попередньої моделі. Під час стекінгу навчальна вибірка доповнюється новими ознаками (метаознаками), що є результатом прогнозування побудованої моделі з ансамблю простих моделей. Доповнену вибірку використовують для побудови нової моделі (ансамблю моделей) – моделі (ансамблю моделей) вищого рівня. Розширення ознакового простору за рахунок стекінгу збільшує ймовірність лінійної розділюваності класів чи лінійної залежності з цільовою змінною.

Високої популярності набув алгоритм паралельної композиції дерев рішень, що називають випадковим лісом (Random Forest). Диверсність ансамблю забезпечено методом подвійної ін’єкції випадковості в індуктивний алгоритм – за рахунок багінгу (бутстреп-вибірка зразків) і використання випадковості підпросторів. Однак для виявлення складних закономірностей може знадобитися надто велика кількість дерев рішень, що ускладнює аналіз в умовах обмеженого обсягу вибірки та обчислювальних ресурсів.

а



б

Рисунок 1.13 – Схема навчання композиції моделей:
а – паралельна композиція; б – послідовна композиція

Одним зі способів підвищення ефективності паралельної композиції дерев рішень є застосування різноманітних технік зниження їх надлишковості або прунінгу (pruning) шляхом видалення неінформативних дерев чи їх гілок. Крім цього в алгоритмі повернутого лісу (Rotation Forest) було запропоновано попередню трансформацію підвибірок на основі методу головних компонент (principal component analysis), що дозволяє формувати більш компактну модель. Цього досягають унаслідок зниження корельованості як ознак бутстреп-вибірки, так і самих дерев рішень без утрат інформації, притаманних методам випадкових підпросторів [33].

Серед алгоритмів послідовної композиції найбільш ефективним вважають алгоритм градієнтного бустингу

(Gradient Boosting). Особливої популярності він набув у поєднанні з базовим алгоритмом дерева рішень. У даному алгоритмі результуюча модель F_m на m-й ітерації формується методом додавання до моделі F_{m-1} з (m-1)-ї ітерації складової $-b_m h(x, a_m)$, де $h(x, a)$ є базовою моделлю, a_m – вектор параметрів базової моделі, що навчається так, щоб її вихід був максимально подібний на градієнт функціонала помилки ∇Q для навчальної множини $\{x^{(j)}, y^{(j)} \mid j = \overline{1,n}\}$. При цьому як функцію втрат $L(y^{(j)}, F_{m-1}(x^{(j)}))$ функціонала помилки Q можуть розглядати квадратичну помилку (Least square), модуль відхилення, функцію Хубера, функцію відступу (margin), логістичну, експоненційну та крос-ентропійну функції втрат. Формули обчислення градієнта для відповідних функцій утрат відомі й добре досліджені. Коефіцієнт важливості кожної базової моделі в композиції b_m оптимізується так, щоб мінімізувати функціонал помилки

$$b_m = \arg\min_{b \in R} Q =$$

$$\arg\min_{b \in R} \sum_{j=1}^{n} L(y^{(j)}, F_{m-1}(x^{(j)}) - bh(x^{(j)}, a_m)).$$

Градієнтний бустинг можуть застосовувати до широкого кола базових алгоритмів із різноманітними функціями втрат. Проте бустинг малопридатний для побудови композиції зі складних і потужних алгоритмів, оскільки цей про-

цес повільний і ресурсозатратний, але не приводить до помітного покращання. При цьому алгоритми бустингу характеризуються схильністю до перенавчання, якщо не вдаватися до спеціальних регуляризувальних мір.

Теоретичне обґрунтування методів класифікаційного аналізу переважно відбувалося в межах статистичних методів навчання. Статистичні методи дозволяють побудувати вирішувальні правила у разі перетину класів розпізнавання, що має місце в практичних задачах контролю та керуванні слабко формалізованими процесами. Одним із класичних статистичних методів класифікації є метод Байєса [34], відповідно до якого прийняття класифікаційних рішень здійснюють методом знаходження максимальної апостеріорної умовної ймовірності $p(X_m^o / x)$, обчисленої для заданого алфавіту класів розпізнавання $\{X_m^o \mid m = \overline{1, M}\}$ за формулою

$$p(X_m^o / x) = \frac{P(X_m^o) p(x / X_m^o)}{\sum_{k=1}^{M} P(X_k^o) p(x / X_k^o)},$$

де $P(X_m^o)$ – безумовна ймовірність появи класу X_m^o;

$p(x / X_m^o)$ – значення функції правдоподібності (щільності розподілу ймовірностей) класу X_m^o для вхідної реалізації x.

Безумовну ймовірність появи класу X_m^o визначають як відношення числа реалізацій, що належать класу X_m^o, до загальної кількості реалізацій

$$P(X_m^o) = \frac{count(x^{(j)} \in X_m^o)}{n},$$

де $count(x^{(j)} \in X_m^o)$ – число реалізацій навчальної вибірки, що належать класу X_m^o;

n – загальна кількість реалізацій образів у навчальній вибірці.

Значення функції правдоподібності класу X_m^o для реалізації x при статистичній незалежності ознак розпізнавання обчислюють за формулою

$$p(x / X_m^o) = \prod_{i=1}^{N} p(x_i / X_m^o),$$

де $p(x_i / X_m^o)$ – значення щільності розподілу ймовірностей i-ї ознаки в класі X_m^o для вхідної реалізації x.

Щільності розподілу ймовірностей $p(x_i / X_m^o)$ можуть бути оцінені в межах припущення про тип розподілу. Наприклад, може використовуватися гіпотеза про нормальний закон розподілу

$$p(x_i / X_m^o) = \frac{1}{\sigma_{m,i}\sqrt{2\pi}} \exp\left(-\frac{\left(x_i - \bar{x}_{m,i}\right)^2}{2\sigma_{m,i}^2} \right),$$

де $\sigma_{m,i}^2$ – дисперсія i-ї ознаки в класі X_m^o;

$\bar{x}_{m,i}$ – математичне очікування i-ї ознаки в класі X_m^o.

Методи розпізнавання, ґрунтуються на використанні формули Байєса й гіпотези про незалежність ознак, зазвичай, називають наївними байєсовськими класифікаторами. Перевагою байєсовських класифікаторів є простота реалізації алгоритмів класифікації, а недоліком – низька достовірність класифікації реалізацій образів у разі обмеженого обсягу вибірок та перетину в просторі ознак класів розпізнавання. При цьому під час побудови байєсовського класифікатора під етапом навчання розуміють набір статистичних даних.

Основними недоліками статистичних методів, що обмежують їх використання на практиці, є необхідність великих обсягів статистики для апроксимації функції щільності розподілу імовірностей, виконання жорстких умов для забезпечення статистичних стійкості та однорідності, висока чутливість до репрезентативності навчальних вибірок.

Отже, існує багато підходів до синтезу вирішувальних правил, серед яких лідерами є композиції простих моделей. Водночас залишається проблема компромісного вибору

моделі й методу навчання вирішувальних правил, оскільки існує деяке протиріччя між вимогами до обсягу навчальних даних та продуктивністю в режимі екзамену.

РОЗДІЛ 2
ІНФОРМАЦІЙНИЙ СИНТЕЗ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ АНАЛІЗУ ДАНИХ

2.1. Формалізована постановка задачі інформаційного синтезу системи аналізу даних

Розглянемо формалізовану постановку задачі інформаційного синтезу здатної навчатися системи аналізу даних. Нехай $D_{train}^{cls} = \{ x^{(j)}, y^{(j)} \mid j = \overline{1, n_1} \}$ та $D_{test}^{cls} = \{ x^{(j)}, y^{(j)} \mid j = \overline{1, n_2} \}$ є наборами навчальних та тестових даних, де $x^{(j)}$ – j-те N-вимірне спостереження, $y^{(j)}$ – мітка класу j-го спостереження, n_1, n_2 – обсяги навчального та тестового наборів даних відповідно. Водночас $y^{(j)} \in \{ X_z^o \mid z = \overline{1, Z} \}$, де X_z^o – z-й клас із заданого алфавіту класів розпізнавання $\{ X_z^o \mid z = \overline{1, Z} \}$. Нехай $D_{train}^{reg} = \{ x^{(j)}, y^{(j)} \mid j = \overline{1, n_1} \}$ та $D_{test}^{reg} = \{ x^{(j)}, y^{(j)} \mid j = \overline{1, n_2} \}$ є наборами навчальних та тестових даних, де $x^{(j)}$ – j-те N-вимірне спостереження, $y^{(j)}$ – мітка j-го спостереження, n_1, n_2 – обсяги навчального та тестового наборів даних відповідно. При цьому $y^{(j)} \in \mathbb{R}$. Нехай $D_{train}^{un} = \{ x^{(j)} \mid j = \overline{1, n_3} \}$ є набором нерозмічених навчальних зразків із відповідної доменної області використання системи.

Дано структурований вектор просторово-часових параметрів функціонування інтелектуальної системи аналізу даних, який у загалом має структуру

$$g =< e_1,...,e_{\xi_1},...,e_{\Xi_1},\ f_1,...,\ f_{\xi_2},...,\ f_{\Xi_2} >,\ \ \Xi_1 + \Xi_2 = \Xi, \quad (2.1)$$

де $e_1,...,e_{\xi_1},...,e_{\Xi_1}$ – генотипні параметри функціонування, що прямо впливають на параметри алгоритмів екстракції ознакового опису спостережень;

$f_1,...,\ f_{\xi_2},...,\ f_{\Xi_2}$ – фенотипні параметри функціонування, що прямо впливають на ефективність вирішувальних правил моделі аналізу даних.

Водночас відомі обмеження на параметри функціонування:

$$R_{\xi_1}(e_1,...,e_{\xi_1},...,e_{\Xi_1}) \le 0,\ R_{\xi_2}(f_1,...,\ f_{\xi_2},...,\ f_{\Xi_2}) \le 0.$$

Необхідно:

1. На етапі машинного навчання інтелектуальної моделі аналізу даних визначити оптимальний вектор параметрів g (2.2), що забезпечує на етапі екзамену максимум комплексного критерію ефективності (2.3)

$$g^* = \arg\max_G \{J(g)\} \quad (2.2)$$

$$J = F(J_{Cls},\ J_{\mathrm{Re}g},\ J_{Complexity}) \quad (2.3)$$

де $F(...)$ – функція-агрегатор, що здійснює згортання часткових критеріїв в один комплексний критерій (може мати адитивну, мультиплікативну та мультиплікативно-адитивну природу);

J_{Cls} – нормований інформаційний критерій функціональної ефективності класифікаційних вирішувальних правил;

$J_{\mathrm{Re}g}$ – нормований критерій точності регресійних вирішувальних правил;

$J_{Complexity}$ – нормований критерій трудомісткості моделі аналізу даних;

G – допустима область значень параметрів, які впливають на екстракцію ознак та прийняття рішень.

2. На етапі екзамену, тобто безпосередньо в робочому режимі, необхідно прийняти рішення про належність вхідного спостереження x до одного з класів алфавіту $\{X_z^o \mid z = \overline{1,Z}\}$ чи спрогнозувати значення сигналу на одному з виходів моделі.

Отже, задача інформаційного синтезу здатної навчатися системи аналізу даних полягає в оптимізації за комплексним критерієм ефективності параметрів її функціонування. При цьому комплексний критерій ефективності враховує точнісні та вартісні характеристики інтелектуальної системи.

2.2. Модель і метод навчання екстрактора ознакового опису спостережень

Формування ознакового опису спостережень за умов ресурсних та інформаційних обмежень варто здійснювати на основі принципів, що дозволяють утилізувати всю доступну інформацію як із розміченої й нерозміченої навчальних вибірок, так і з зовнішніх джерел. Вибіркові дані і без розмітки містять багато інформації про їх структуру, яку можна виділити на основі моделей розділення пояснювальних факторів (disentangle explanatory factors). Моделі розділення пояснювальних факторів зазвичай будують на основі автокодувальників, машин Больцмана, алгоритмів розрідженого кодування та генеративних моделей. Крім того, навчені моделі для аналізу образів у подібних задачах є зовнішніми джерелами, що акумулюють у собі досвід, який можна використати для поточної задачі. Саме в цьому полягає принцип перенесення знань (Transfer Learning), згідно з яким вагові коефіцієнти навчених мереж можна запозичити як квазіоптимальну стартову точку пошуку для нової задачі. Узагальнену структуру екстрактора ознак, побудованого на таких принципах, показано на рисунку 2.1.

Серед моделей розділення пояснювальних факторів заслуговують на увагу ті, що ґрунтуються на ідеях і методах розрідженого кодування. Покладений у їх основу ефект редукції причини (explaining away) дозволяє виявити приховані фактори (першопричини) й забезпечити інформативне та завадозахищене ознакове подання вибіркових спостере-

жень навіть за умов обмеженого обсягу даних.

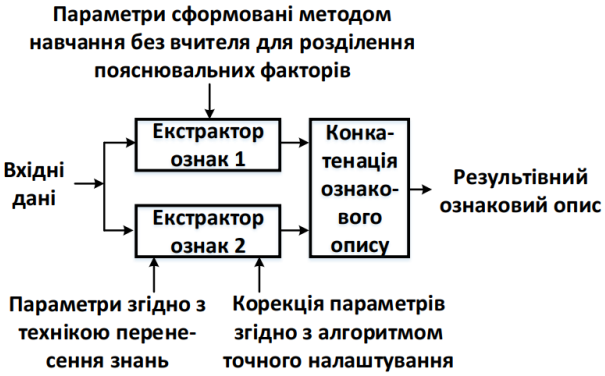


Рисунок 2.1 – Узагальнена структура екстрактора ознак

Для підвищення рівня абстрактності пояснювальних факторів екстрактор ознак може містити декілька шарів розрідженого кодування. Реалізація кодера може бути виконана на основі алгоритмів узгодженого переслідування (matching pursuit), ортогонального узгодженого переслідування (orthogonal matching pursuit), мішка переслідувачів (bag of pursuits) чи інших. Точне налаштування екстрактора ознак за необхідності можна виконати за допомогою алгоритму зворотного розповсюдження з тимчасовим або постійним нейромережевим класифікатором на виході моделі [40]. Проте в умовах нестаціонарності точне налаштування екстрактора ознак не використовується, оскільки інформативність ознак заздалегідь не відома. Для прискорення моделі в режимі екзамену згідно з принципами дистиляції знань (knowledge distillation) обчислювально-складний етап оптимізації для пошуку розріджених коефі-

цієнтів можна замінити апроксимувальним кодером [35, 36]. Навчальна вибірка апроксимувального кодера формуватиметься як із вхідних даних, що кодуються, так і з результувального розрідженого коду. На рисунку 2.2 показано схему синтезу екстрактора ознак із використанням ідей і методів розрідженого кодування.

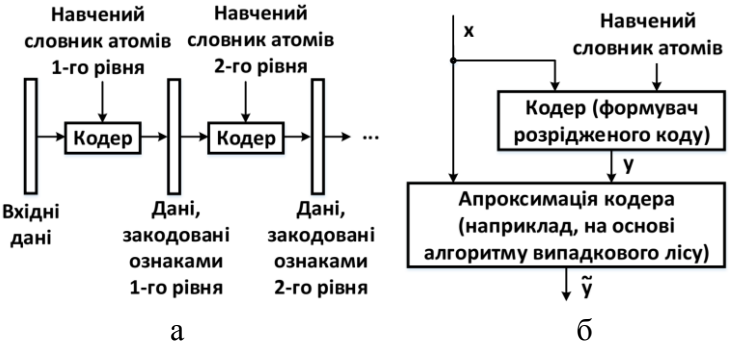


Рисунок 2.2 – Схема побудови екстрактора ознак на основі розрідженого кодування: а – послідовність оброблення даних у багатошаровому екстракторі; б – схема дистиляції знань з розрідженого кодера

Для розрідженого кодування використовують навчений словник атомів. Атоми мають розмірність таку саму як і будь-яке вхідне спостереження x і їх у найпростішому випадку, можна сформувати методом кластер-аналізу чи векторного квантування спостережень. У більш складному випадку навчання можуть здійснювати за схемою з кінця в кінець і навіть із частковим залученням учителя.

Перспективним підходом до реалізації навчання слов-

ника атомів без учителя є використання принципів нейронного газу та правила Ойа, що реалізовано в так званому алгоритмі розріджено кодувального нейронного газу. Алгоритм нейронного газу характеризується м'якою конкурувальною схемою навчання, що приводить до більш надійної збіжності алгоритму й оптимального розподілу кластерів на вибірці вхідних даних. При цьому використання правила Ойа дозволяє сформувати словник, що забезпечує мінімальну корельованість ознак і завадозахищеність кодування спостережень.

Вхідними даними для алгоритму розріджено кодувального нейронного газу є потужність словника базисних векторів M, розмірність простору ознак N, $\lambda_0, \lambda_{final}$ – початкове та кінцеве значення коефіцієнта розміру околу сусідів, η_0, η_{final} – початкове та кінцеве значення коефіцієнта швидкості навчання. Розглянемо основні кроки алгоритму:

1) ініціалізація словника базисних векторів $D = (d_1, ..., d_M)$ випадковими числами з рівномірного розподілу;

2) ініціалізація лічильника навчальних векторів $t := 1$.

3) вибір випадкового вектора x з множини навчальних векторів X.

4) L2-нормалізація векторів зі словника $D = (d_1, ..., d_M)$ методом приведення до одиничної довжини;

5) обчислення поточних значень коефіцієнта розміру околу сусідів λ_t та швидкості навчання η_t:

$$\lambda_t := \lambda_0(\lambda_{final} / \lambda_0)^{t/t_{\max}}, \quad \eta_t := \eta_0(\eta_{final} / \eta_0)^{t/t_{\max}};$$

6) обчислення міри подібності вхідного вектора x до базисних векторів $d_{l_k} \in D$ для їх сортування

$$-(d_{l_0}^T x)^2 \le ... \le -(d_{l_k}^T x)^2 \le ... \le -(d_{l_{M-1}}^T x)^2;$$

7) оновлення координат базисних векторів $d_{l_k} \in D$ за правилом Ойа [37]:

$$d_{l_k} := d_{l_k} + \eta_t \exp(-k / \lambda_t) y(x - y d_{l_k}),$$
$$y := d_{l_k}^T x, \; k = \overline{0, M-1};$$

8) якщо $t < t_{\max}$, то інкремент лічильника $t := t + 1$ та перехід до кроку 3, інакше – припинення виконання алгоритму.

Основним недоліком розріджено кодувального нейронного газу є те, що кількість кластерів наперед невідома й задається на розсуд розробника, або оптимізується, що призводить до збільшення кількості ітерацій навчання. Необхідну кількість кластерів наперед оцінити важко, тому перспективним підходом є використання принципів зростаючого нейронного газу, що дозволяє автоматично визначити необхідну кількість нейронів [38]. Однак механізм вставки нових нейронів в алгоритмі зростаючого нейрон-

ного газу на основі задавання періоду вставки часто призводить до викривлення утворених структур і нестабільності процесу навчання. Проте забезпечити стабільність навчання можна методом задавання «радіусу досяжності» нейронів, що передбачає заміну періоду вставки нейронів на поріг максимального віддалення нейрона від кожної з віднесених до нього точок навчальної множини. Водночас з метою адаптації процесу навчання до процедури розрідженого кодування спостережень потрібно змінити механізми оновлення кластерів та оцінення віддаленості точок вхідного простору до кластерів. Таку зміну можна виконати так, як це зроблено в алгоритмі розріджено-кодувального нейронного газу. Побудований таким чином алгоритм назвемо зростаючим розріджено кодувальним нейронним газом. Розглянемо основні кроки алгоритму зростаючого розріджено кодувального нейронного газу [39]:

1) ініціалізація лічильника навчальних векторів $t := 0$;

2) два початкових вузли (нейрони) w_a і w_b ініціалізуються методом випадкового вибору векторів із навчальних даних. Вузли w_a і w_b з'єднують ребром, вік якого беруть за нульовий. Ці вузли вважають нефіксованими;

3) обирають наступний вектор x, що нормалізується методом приведення до одиничної довжини (L2-нормування);

4) нормалізують кожний базисний вектор, $w_k, k = \overline{1, M}$, методом приведення до одиничної довжини (L2-нормалізація);

5) розраховують міру подібності вхідного вектора x до базисних векторів $w_{s_k} \in W$ для сортування

$$-(w_{s_0}^T x)^2 \le ... \le -(w_{s_k}^T x)^2 \le ... \le -(w_{s_{M-1}}^T x)^2;$$

6) визначають найближчий вузол w_{s0} та другий за подібністю вузол w_{s1};

7) збільшують на одиницю вік усіх вузлів, інцидентних до w_{s0};

8) якщо вузол w_{s0} фіксований, то переходять до кроку 9, інакше – до кроку 10;

9) якщо $(w_{s0}^T x)^2 \ge v,$ то переходять до кроку 12. В іншому разі додають новий нефіксований нейрон $w_r = x$ та нове ребро, що з'єднує w_r і w_{s0}, потім переходять до кроку 13;

10) вузол w_{s0} та його топологічні сусіди (вузли, зв'язані з ним ребрами) зміщуються у напрямку вхідного вектора x відповідно до правила Ойа [29] за формулами

$$\triangle w_{s0} = \varepsilon_b \eta_t w_{s0}^T x (x - w_{s0}^T x w_{s0}),$$

$$\triangle w_{sn} = \varepsilon_n \eta_t w_{sn}^T x (x - w_{sn}^T x w_{sn}),$$

$$0 < \varepsilon_b \ll 1, \;\; 0 < \varepsilon_n \ll \varepsilon_b, \quad \eta_t := \eta_0 (\eta_{final} / \eta_0)^{t/t_{\max}},$$

де $\triangle w_{s0}$, $\triangle w_{sn}$ – вектори корекції вагових коефіцієнтів нейрона-переможця та його топологічних сусідів відповідно; ε_b, ε_n – константи міри оновлення вагових коефіцієн-

тів нейрона-переможця та його топологічних сусідів відповідно; η_0, η_t, η_{final} – початкове, поточне та кінцеве значення швидкості навчання відповідно;

11) якщо $(w_{s_0}^T x)^2 \geq v$, то нейрон w_{s0} позначають як фіксований;

12) якщо w_{s0} і w_{s1} з'єднані ребром, їхній вік обнуляють, в іншому разі – між w_{s0} і w_{s1} формують нове ребро з нульовим віком;

13) усі ребра графа з віком більше ніж $a_{\max}$ видаляються. У тому випадку, коли в деяких вузлах немає ребер (вузли стають ізольованими), ці вузли також видаляють;

14) якщо $t < t_{\max}$, то переходять до кроку 15, інакше – збільшують лічильник $t := t+1$ і переходять до кроку 3;

15) якщо всі нейрони фіксовані, то виконання алгоритму припиняють, інакше переходять до кроку 3 і починається нова епоха навчання (повторення набору даних).

На ефективність аналізування образів значною мірою впливає врахування апріорної інформації про топологію вхідних даних. Для 1D-, 2D- та 3D-топологій, що широко використовуються в задачах оброблення зображення та послідовностей, аналіз відбувається в межах локальних рецептивних полів. Тобто вхідні дані скануються вікном (рецептивним полем) у межах якого відбувається розріджене кодування або згортка зі згортковими фільтрами. На рисунку 2.3 показано габарити вхідних даних із відомою топологією та умовні розміри рецептивного поля для формування карти ознак.

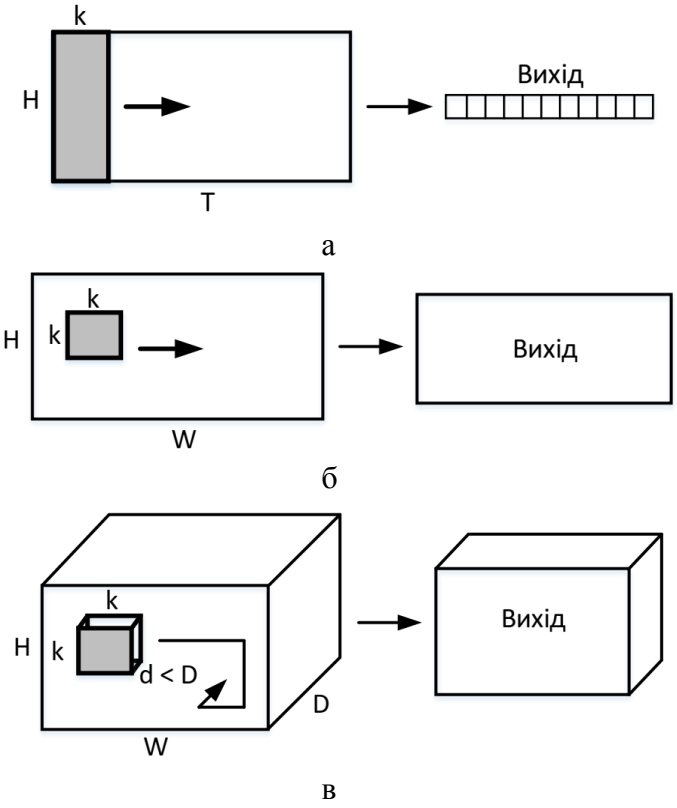


Рисунок 2.3 – Схема формування карти ознак кодувальним вікном: а – 1D-топологія даних; б – 2D-топологія даних; в – 3D-топологія даних

Функцію відображення даних у межах сканувального рецептивного поля в піксель карти ознак назвемо скануючим кодером. На рисунку 2.3а показано дані з 1D-топологією у вигляді багатовимірної послідовності даних, що складається з H сигналів, які аналізуються у межах буфера довжиною в T відліків.

На рисунку 2.3 б показано дані з 2D-топологією у вигляді одноканального зображення шириною W та висотою H, що сканується кодером за вертикаллю і горизонталлю рецептивним полем k x k пікселів. На рисунку 2.2 в показано дані з 3D-топологією у вигляді D-канального зображення з шириною W та висотою H, що сканується кодером за вертикаллю, горизонталі і вглиб рецептивним полем k x k x d пікселів.

У разі зображення процес його розрідженого кодування полягає в декомпозиції на патчі, що перетинаються, кожен із яких кодується кодером у багатоканальний піксель карти ознак (рис. 2.4).

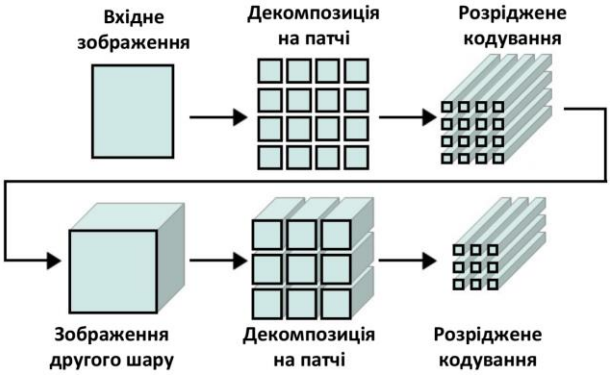


Рисунок 2.4 – Схема аналізу зображення двошаровою моделлю на основі розрідженого кодування

Оскільки кодер оснований на розрідженому кодуванні, то піксель карти ознак матиме розріджену активацію каналів. Одержану карту ознак можна сприймати як нове багатоканальне зображення, для якого можна повторити про-

цедуру розрідженого кодування. При цьому кількість атомів (кластерів) кодера може бути як більшою за розмірність вхідного патчу, що відповідає надповному базису (overcomplete dictionary), так і меншою за розмірність вхідного патчу, що відповідає неповному базису (undercomplete dictionary). Дослідження показують, що для задачі класифікаційного аналізу можуть бути ефективними як надповні, так і неповні базиси екстрактора ознак [40].

У задачах аналізу часових послідовностей набула популярності каузальна архітектура моделей для роботи з даними, що мають 1D-топологію. У цій архітектурі рецептивне поле кодера охоплює поточні й минулі дані. Тобто результат аналізу кожного кодера залежить лише від значень сигналу в минулих відліках часу (рис. 2.5). При цьому для збільшення рецептивного поля моделі її необхідно стекувати в глибину.

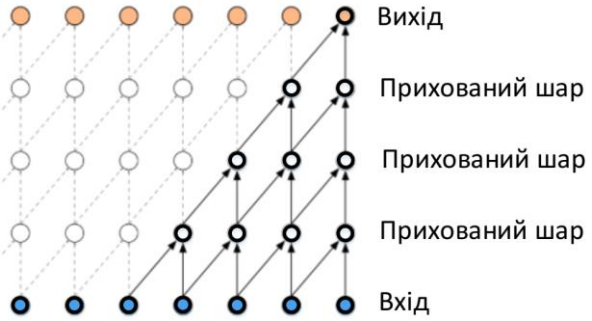


Рисунок 2.5 – Архітектура каузальної моделі
для аналізу послідовностей

Для аналізу даних високої розмірності з відомою топо-

логією можна використовувати діряві (dilated) рецептивні поля, що застосовують до локальних ділянок даних, розмір яких перевищує кількість входів кодера. Тобто частина даних, що накривається рецептивним полем, ігнорують, а аналізу піддають дані з певним просторовим або часовим кроком (рис. 2.6).

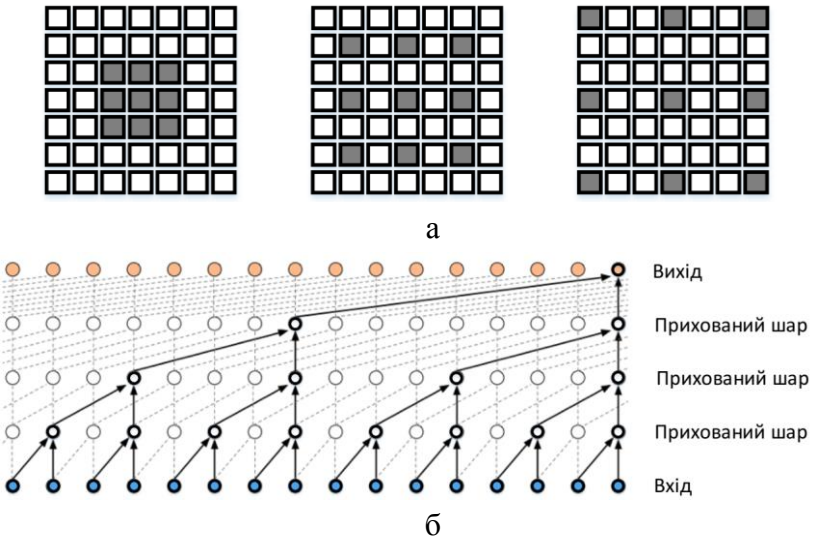


Рисунок 2.6 – Діряві рецептивні поля моделі аналізу даних: а – 2D-топологія; б – 1D-топологія каузальної моделі

Стекування кодерів із дірявими рецептивними полями дозволяє досягти великого рецептивного поля моделі з використанням невеликої кількості шарів і параметрів. Разом із тим не відбувається втрат інформації, оскільки кодер здійснює сканування вхідних даних у просторі або

часі. Крім того, наявність пропусків дозволяє знизити чутливість до високочастотних складових шуму і, в наслідок цього, підвищити завадозахищеність моделі.

Для дослідження залежностей на різних рівнях деталізації в межах окремого шару моделі використовують поєднання кодерів із рецептивними полями різного розміру (рис. 2.7).

Рисунок 2.7 – Приклад використання кодерів із рецептивними полями різного розміру для формування ознакового опису різного рівня деталізації

Малі рецептивні поля кодерів дозволяють здійснювати екстракцію детальної (fine-grained) інформації, а великі рецептивні поля кодерів забезпечують екстракцію грубої (coarse-grained), переважно, контекстної інформації. Наприклад у популярних Inception-модулях згорткових мереж використовують згорткові фільтри з ядрами 1x1, 3x3 та 5x5

для сприйняття різномасштабної просторової інформації (від детального до загрубленого рівнів).

Одним із шляхів ефективної регуляризації глибоких моделей аналізу даних є застосування методу Dropout, що зменшує ймовірність ефекту перенавчання й забезпечує навчання за схемою «з кінця в кінець» (end-to-end). Його суть полягає у випадковому вимиканні частини нейронів нейронного шару під час навчання та усередненні результатів у режимі екзамену. Шар нейронів на кожному окремому кроці навчання розглядають як ансамбль експериментів Бернулі. Множина вимкнених нейронів на кожній ітерації навчання є випадковою величиною, розподіленою за біноміальним законом.

Отже, метод Dropout реалізує псевдоансамблювання підмереж однієї нейронної мережі. При цьому основного ефекту його застосування досягають за рахунок усування прояву взаємоадаптації (co-adaptation) нейронів. Взаємоадаптація нейронів полягає в адаптації одних нейронів для компенсації помилок інших нейронів, однак результати цього ефекту не узагальнюються на дані, які не брали участі у навчанні. Крім очевидних переваг застосування методу Dropout, існують і недоліки. Наприклад, застосування методу Dropout призводить до збільшення у 2–3 рази часу необхідного для навчання внаслідок досить зашумленого сигналу оновлення параметрів.

У разі апріорної невизначеності виправданим є протокол інкрементального ускладнення моделі до моменту перенавчання. Тому важливе місце поряд зі схемою навчання «з-кінця-в-кінець» (end-to-end) посідає і схема навчання

«шар-за-шаром» (layer-by-layer). Цю схему можна використовувати як для навчання без учителя моделі розділення пояснювальних факторів, так і при додаванні чи точному налаштуванні запозичених шарів у межах техніки переносу знань (Transfer Learning). Водночас за умов ресурсних та інформаційних обмежень альтернативою до методу Dropout можуть бути техніки, що ґрунтуються на звичайному ансамблюванні та *L1* або *L2* регуляризації.

Серед методів ансамблювання в задачі виділення компактного ознакового подання заслуговує на увагу алгоритм випадкового лісу, побудований із використанням методу предиктивних кластеризувальних дерев (Predictive Clustering Trees). Синтез дерев рішень в рамках методу предиктивних кластеризувальних дерев полягає в рекурсивному зменшенні внутрішньокластерної дисперсії з кожним розщепленням вузлів дерева. У результаті шлях вхідного вектора x від вершини до листка містить цінну інформацію про структуру даних. При цьому ансамбль таких дерев рішень дозволяє сформувати ознакове подання, стійке до шуму. Екстракція компактного ознакового опису в цьому разі може здійснюватися методом конкатенації шляху рішення у деревах рішень, закодованого двійковим кодом (рис. 2.8).

Аналіз рисунку 2.8 показує, що вузли кожного дерева пронумеровані, а кількість вузлів у дереві рішень рівна довжині двійкового коду. Тоді кодування вхідного спостереження полягає в присвоєнні одиничних значень тим бітам, що відповідають вузлам, через які проходить шлях прийняття рішення від кореня до листка відповідного де-

рева [41].

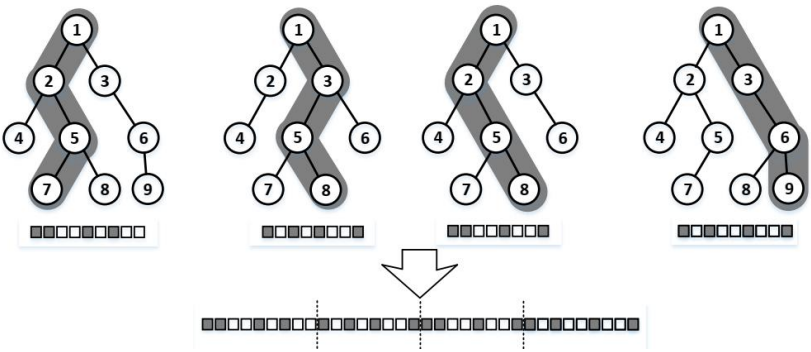


Рисунок 2.8 – Індукція ознакового опису ансамблем випадкових дерев рішень

У загальному випадку модель будь-якої інтелектуальної моделі аналізу даних можна подати у вигляді діаграми відображення множин, задіяних у процесах трансформації даних та прийняття рішень (рис. 2.9).

На рисунку 2.9 прийнято такі позначення:

T – множина моментів часу зняття інформації;

G – простір вхідних сигналів (факторів), що діють на систему;

Ω – простір ознак розпізнавання;

Z – простір можливих функціональних станів, у яких перебуває система;

Y – вибіркова множина (вхідна навчальна матриця);

Φ: $G \times T \times \Omega \times Z \times V \rightarrow Y$ – оператор формування вибіркової навчальної множини Y;

θ – оператор відображення вибіркової множини Y в

простір вторинних ознак, у якому формуються вирішувальні правила $\tilde{\Re}$ ($\tilde{\Re} \subset \Omega$);

ψ – оператор прийняття рішень;

I – множина результатів перевірки статистичних тестів;

ξ_1 – оператор зворотного зв'язку для корекції параметрів вирішувальних правил;

ξ_2 – оператор зворотного зв'язку для корекції параметрів формування вторинного ознакового опису;

$U : I \to G \times T \times \Omega \times Z$ – оператор регламентації процесу функціонування інтелектуальної системи аналізу вхідних даних.

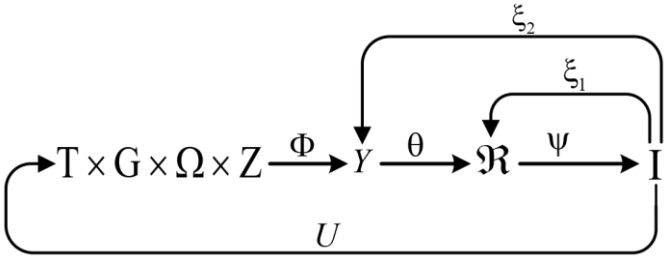


Рисунок 2.9 – Узагальнена модель інтелектуальної моделі аналізу даних у вигляді відображення множин

Для деталізації процесів оптимізації екстрактора ознак контури, утворені операторами ξ_1 та ξ_1, розщеплюються на композицію нових контурів. На рисунку 2.10 показано модель процесу навчання ієрархічного екстрактора ознак для інтелектуального аналізу спостережень у вигляді діаг-

рами відображення множин.

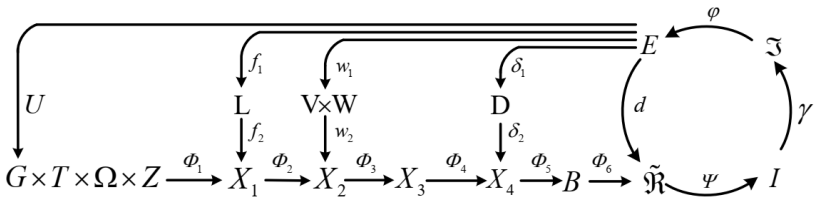


Рисунок 2.10 – Модель навчання ієрархічного екстрактора ознак для аналізу вхідних даних у вигляді діаграми відображення множин

Діаграма, показана на рисунку 2.10, містить вхідний математичний опис системи аналізу вхідних даних у вигляді структури

$$< G, T, \Omega, Z, X_1, X_2, X_3, X_4, B; \Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5, \Phi_6 >,$$

де X_1 – вибірка необроблених спостережень;

X_2 – спостереження, доповнені розрідженим кодом пояснювальних факторів;

X_3 – спостереження, додатково доповнені ознаковим описом, сформованим запозиченими шарами моделі згідно з технікою перенесення знань (Transfer Learning);

X_4 – спостереження, описані лише індукованими ознаками (внаслідок відкидання початкового оригінального ознакового опису);

B – спостереження, закодовані ознаковим описом най-

вищого рівня, в межах якого відбувається побудова вирішувальних правил;

Φ_1 – оператор формування вибірки даних;

Φ_2 – оператор екстракції пояснювальних факторів для доповнення вибірки;

Φ_3 – оператор екстракції ознакового опису на основі запозичених шарів відповідно до принципу перенесення знань (Transfer Learning) для додаткового доповнення вибірки;

Φ_4 – оператор видалення початкового оригінального ознакового опису, що повертає лише індукований ознаковий опис;

Φ_5 – оператор формування ознакового опису найвищого рівня.

На рисунку 2.10 оператор $\Phi_6 : B \to \tilde{\Re}$ відображає вибіркові дані, закодовані ознаковим описом найвищого рівня, в множину параметрів, що описує конфігурацію вирішувальних правил. Оператор прийняття рішень ψ: $\tilde{\Re} \to$I перевіряє статистичні гіпотези. Оператор γ: I|l| $\to \Im$|q| методом оцінювання статистичних гіпотез формує множину точнісних характеристик $\Im$. Оператор $\phi : \Im \to$E обчислює множину значень критерію функціональної ефективності, який є функціоналом точнісних характеристик. Контур оптимізації геометричних параметрів вирішувальних правил $\tilde{\Re}$ методом пошуку максимуму критерію навчання замикається оператором d : E$\to \tilde{\Re}^{|M|}$.

У процесі оптимізації параметрів і гіперпараметрів моделі беруть участь такі множини: L – множина гіперпараметрів моделі для виділення пояснювальних факторів; V – множина масок на верхні шари запозиченої нейронної мережі в межах техніки перенесення знань; W – множина значень корекції параметрів запозиченої мережі для її точного налаштування; D – множина значень параметрів регуляризувального шару екстракції ознакового опису найвищого рівня. Композиція операторів $\psi \circ \gamma \circ \varphi \circ d$ утворює контур оптимізації геометричних параметрів вирішувальних правил. Композиція операторів $\delta_1 \circ \delta_2 \circ \Phi_5 \circ \Phi_6$ утворює контур оптимізації параметрів регуляризувального шару екстракції компактного ознакового подання найвищого рівня. Композиція операторів $w_1 \circ w_2 \circ \Phi_3 \circ \Phi_4$ утворює контур селекції та точного налаштування шарів запозиченої нейромережі згідно з принципом перенесення знань. Композиція операторів $f_1 \circ f_2 \circ \Phi_2$ утворює контур оптимізації гіперпараметрів моделі виділення пояснювальних факторів.

Отже, формування ознакового опису полягає в застосуванні моделей і методів, що дозволяють утилізувати всю доступну апріорну інформацію для підвищення ефективності навчання. До апріорної інформації відносять інформацію про топологію даних, інформацію про доменну область застосування системи (для пошуку сторонніх моделей, акумульовані знання яких можна запозичити згідно з технікою перенесення), а також інформацію про структуру даних у вигляді множини пояснювальних факторів. Існує

велика кількість варіантів щодо вибору моделі виділення пояснювальних факторів із вибірки нерозмічених даних, велика кількість конфігурацій рецептивних полів нейронів або кодерів та способів компактного й завадозахищеного подання ознакового опису найвищого рівня. Під час синтезу моделі виділення пояснювальних факторів перевагу варто надавати моделям і методам, що ґрунтуються на принципах розрідженого кодування. При цьому налаштування параметрів і гіперпараметрів моделі екстрактора ознак ґрунтується на максимізації функціональної ефективності вирішувальних правил.

2.3. Модель і метод синтезу класифікаційних вирішувальних правил

Синтез класифікаційних вирішувальних правил пропонують здійснювати в межах так званої інформаційно-екстремальної інтелектуальної технології (IEI-технології), що ґрунтується на таких принципах:

– двійкове кодування класів розпізнавання самокоректувальними кодами (Error-Correcting Output Coding) з урахуванням внутрішньої структури класів розпізнавання;

– трансформація простору ознак для зменшення відстані між зразками однакових класів та збільшення відстані між зразками різних класів (подібно до сіамських нейронних мереж) у межах двійкового простору Хеммінга;

– оптимізація в інформаційному розумінні вирішувальних правил, що відновлюються в радіальному базисі простору Хеммінга, з метою врахування габаритів розподілу

кожного класу та підвищення стійкості до шуму й новизни в даних;

– інкрементальне нарощування складності моделі;

– використання обчислювально ефективних операцій як будівельних блоків моделі.

У межах IEI-технології процес навчання моделі класифікаційного аналізу полягає в реалізації ітераційної процедури оптимізації генотипних та фенотипних параметрів функціонування, що впливають на функціональну ефективність навчання. Одним з основних генотипних параметрів навчання моделі класифікаційного аналізу є система порогів для двійкового кодування спостережень. У найпростішому випадку ця система може бути побудована у вигляді полів контрольних допусків, а в більш складному випадку – у вигляді ансамблю дерев рішень.

Система полів контрольних допусків (СКД) на ознаки розпізнавання, що визначає координати спостережень у бінарному просторі ознак, напряму впливає на геометричні параметри замкнених роздільних гіперповерхонь, що зазвичай називають контейнерами класів розпізнавання. Оптимізація СКД спрямована на таку зміну розподілу векторів спостережень у бінарному просторі, що дозволяє оптимально в інформаційному розумінні описати розподіл класів контейнерами найпростішої (гіперсферичної) форми. Тобто алгоритм навчання повинен забезпечити в найгіршому випадку мінімальний перетин контейнерів класів, а в найкращому випадку – максимальний відступ між межами контейнерів класів (рис. 2.11). На рисунку 2.11 показано межі гіперсферичних контейнерів і розподілу спосте-

режень двох класів, де зразки одного з класів показані у вигляді чорних точок, а зразки іншого класу – у вигляді незафарбованих кілець.

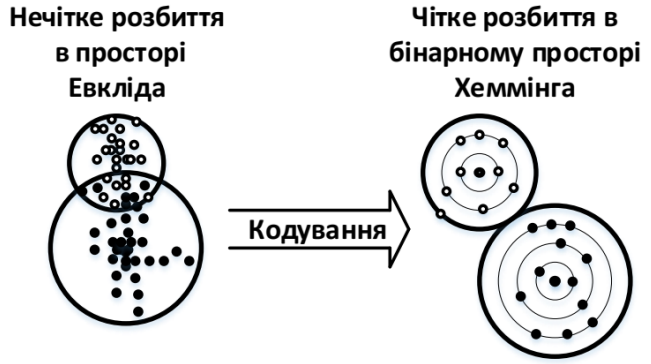


Рисунок 2.11 – Ілюстрація для пояснення ідеї трансформації простору ознак у межах IEI-технології

На рисунку 2.12 показано поле контрольних допусків на значення i-ї ознаки $x_{m,i}^{(j)}$, $i=\overline{1,N}$, межі яких відрахову-ються від усередненого значення ознаки в базовому класі $X_{Б}^{o}\in\{X_{m}^{o}\}$, що обирає розробник інформаційного забезпечення. У діагностичних системах базовому класу $X_{Б}^{o}$ відповідає клас, що характеризує нормальний стан, щоб решту класів можна було розглядати як різноманітні відхилення від норми [42].

На рисунку 2.12 взято такі позначення: $\bar{x}_{Б,i}$ – усереднене значення ознаки в базовому класі; $A_{\min,i}$, $A_{\max,i}$ – нижній та верхній нормовані допуски відповідно, які зада-

ють область значень i-ї ознаки й відповідних контрольних допусків; $A_{H,l,i}$, $A_{B,l,i}$ – нижня та верхня межі контрольних допусків l-го інтервалу; $\delta_{K,l,i}$ – поле контрольних допусків l-го інтервалу.

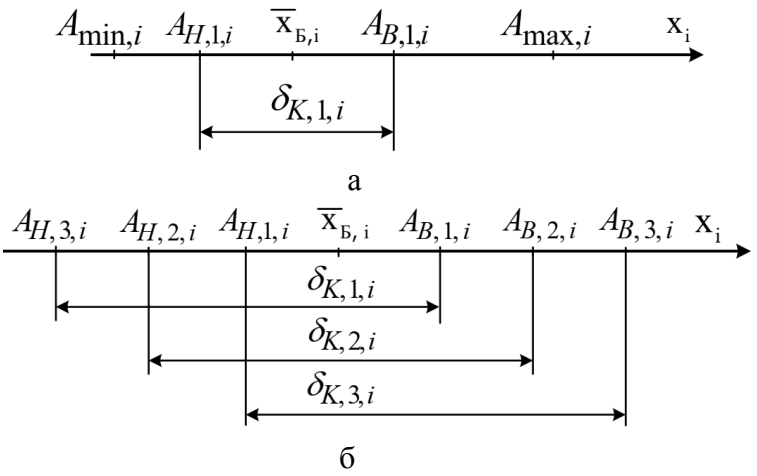


Рисунок 2.12 – Поле контрольних допусків на значення ознаки: а – одноінтервальне поле; б – триінтервальне поле

Інформаційно-екстремальне машинне навчання ґрунтується на адаптивному двійковому кодуванні ознак розпізнавання за допомогою порівняння значення i-ї ознаки з відповідними нижнім $A_{H,l,i}$ та верхнім $A_{B,l,i}$ контрольними допусками l-го інтервалу, розрахованими за такими формулами:

$$A_{H,l,i} = \overline{x}_{Б,i}\left[1-\frac{\delta_i}{\delta_{\max}}\right]^l,$$

$$A_{B,l,i} = \bar{x}_{Б,i}\left[1+\frac{\delta_i}{\delta_{\max}}\right]^l,$$

де δ_i – параметр поля контрольних допусків;

$\delta_{\max}$ – максимальне значення параметра поля контрольних допусків.

У разі використання СКД формування бінарної навчальної матриці $\{b^{(j)}_{m,f} \mid f=\overline{1,L*N};\ l=\overline{1,L};\ i=\overline{1,N};\ j=\overline{1,n};m=\overline{1,M}\}$ для L-інтервальної СКД здійснюють за правилом

$$b^{(j)}_{m,L*i-L+l} = \begin{cases} 1, & \text{якщо } A_{H,L-l+1,i} \le x^{(j)}_{m,i} \le A_{B,l,i}; \\ 0, & \text{інакше} \end{cases} \quad (2.4)$$

Межі контрольних допусків поділяють область можливих значень ознаки розпізнавання на *2*L+1* областей, кожній із яких відповідає окремий двійковий код i-ї ознаки, що складається з L розрядів. Кодова відстань між кодами сусідніх областей дорівнює одній кодовій одиниці, а кодова відстань між кодами областей, розміщених через одну чи більше областей, рівна двом і більше кодовим одиницям. Запропонована схема кодування (2.4) дозволяє збільшити різноманітність двійкових векторів-реалізацій та враховувати напрям відхилення розподілу векторів-реалізацій образів від базового класу, що відповідає найбільш бажаному функціональному стану.

Одним зі способів реалізації оперативного визначення оптимальних параметрів СКД є застосування популяційних алгоритмів пошукової оптимізації. Ці алгоритми не потребують початкових наближень і дозволяють знайти оптимальне рішення за невелику кількість ітерацій. Водночас одна ітерація популяційного алгоритму потребує n_a запусків базового алгоритму інформаційно-екстремального машинного навчання, де n_a – кількість агентів популяції [43]. Базовий алгоритм інформаційно-екстремального навчання полягає в оптимізації геометричних параметрів контейнерів класів при незмінній СКД для оцінювання функціональної ефективності моделі.

Одним із найпростіших у реалізації популяційних алгоритмів пошукової оптимізації є алгоритм рою частинок (Particle Swarm Optimization) [43]. Ефективність кожної частинки популяційного алгоритму, тобто близькість до глобального оптимуму, вимірюють за допомогою наперед визначеної фітнес-функції J. Роль фітнес-функції виконує функція інформаційного критерію ефективності машинного навчання. Кожна *j*-та частинка, крім її позиції *Pj,* зберігає таку інформацію: *Vj* – поточна швидкість частинки, *Pbestj* – краща персональна позиція частинки. Краща персональна позиція *j*-ї частинки – це позиція *j*-ї частинки, у якій значення фітнес функції для частинки було максимальним на поточний момент часу. Зокрема, з метою пошуку глобального екстремуму фітнес-функції найкращу частинку шукають в усьому рої, а позицію позначають як *Gbest*. Розглянемо основні кроки реалізації алгоритму рою части-

нок для оптимізації вектора параметрів поля СКД.

1. Ініціалізація рою частинок (агентів):

а) ініціалізація кількості частинок n_a;

б) ініціалізація розмірності кожної частинки N та ініціалізація меж зміни i-ї координати j-ї частинки $\delta_{j,i}$;

в) ініціалізація початкових позицій частинок $P_j[0] := 100 \cdot U(0,1)$, де $U(0,1)$ – генератор випадкових чисел з діапазону $(0,1)$;

г) ініціалізація початкових швидкостей частинок $V_j(0) := 0$;

д) ініціалізація максимальної швидкості частинок у $V_{\max,i}$;

е) ініціалізація вагових коефіцієнтів для формули швидкості, тобто ваги інерції w та констант прискорення c_1 і c_2.

2. Інкремент номера ітерації: $k := k + 1$.

3. Інкремент номера частинки: $j := j + 1$.

4. Інкремент номера координати в позиції: $i := i + 1$.

5. Розрахування нового стану частинки:

а) розрахування i-ї компоненти швидкості для j-ї частинки за правилами

$$V_{j,i}[k+1] := wV_{j,i}[k] + c_1 a_{1,i}[k] * (Pbest_{j,i}[k] - P_{j,i}[k]) + c_2 a_2[k] * (Gbest_j - P_{j,i}[k]);$$

$$V_{j,i}[k+1] := \begin{cases} V_{j,i}[k+1] & if \quad V_{j,i}[k+1] < V_{\max,i}, \\ V_{\max,i} & if \quad else, \end{cases}$$

де $a_1[k]$ = U(0,1), $a_2[k]$ = U(0,1);

б) оновлення позиції частинки $P_j[k+1] := P_j[k] + V_j[k+1]$;

в) обчислення цільової функції $J_j[k+1]$;

г) оновлення значень найкращої персональної *Pbest* та глобальної *Gbest* позицій агентів пошуку

$$Pbest_j[k+1] := \begin{cases} Pbest_j[k], if \ J(P_j[k+1]) \le J(Pbest_j[k]); \\ P_j[k+1], if \ else; \end{cases}$$

$$Gbest[k+1] := \arg\max_j \{J(Pbest_j[k+1])\}.$$

6. Перевірка умови зупинення: якщо $k < K_{\max}$, де $K_{\max}$ – максимальна кількість ітерацій пошуку, і $J(Gbest[k+1]) < 1,0$, то відбувається перехід до кроку 2, інакше – до кроку 7.

7. Зупинення.

Використання ансамблю дерев рішень є ще одним способом двійкового кодуванням ознак. При цьому двійкове кодування вектора x_j здійснюють методом конкатенації результатів рішень деревами ансамблю T_1, .., T_L. При цьому кожне дерево рішень формує бінарний код, де кожен

ненульовий біт відповідає вузлу на шляху прийняття рішень від кореня дерева до термінального вузла). Дерева рішень будуються за принципом багінгу або бустингу (рис. 2.13) предиктивних кластеризувальних дерев рішень (Predictive Clustering Tree) [41].

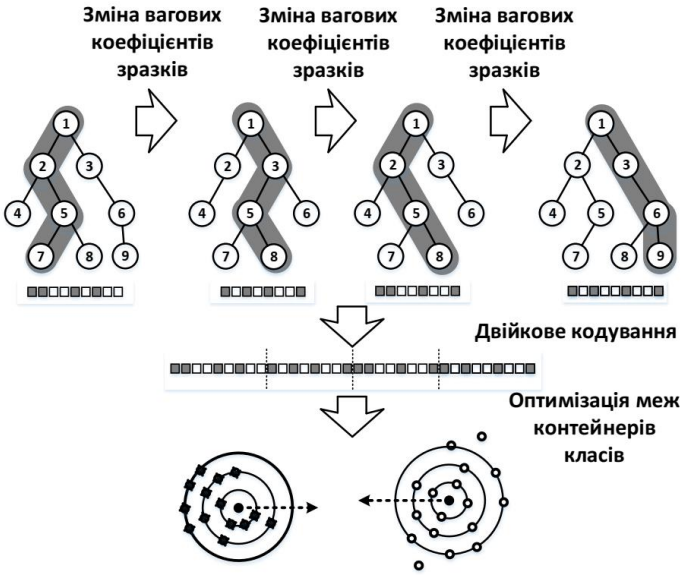


Рисунок 2.13 – Схема двійкового кодування спостережень із використанням бустингу кластеризувальних дерев рішень

У результаті вхідна навчальна вибірка кодується в бінарну навчальну матрицю $\{b_{m,s,i} \mid i = \overline{1, N_2};\ s = \overline{1, n_z};\ m = \overline{1, Z}\}$, де N_2 – кількість індукованих двійкових ознак; n_m – кількість навчальних зразків, що відповідають класу X_m^o.

Інформаційно-екстремальний класифікатор у режимі екзамену визначає належність вхідного вектора x із відповідним двійковим поданням b до одного з класів $\{X_m^o \mid m=\overline{1,M}\}$ відповідно до максимального значення функції належності $\mu_m(b)$ на основі процедури $\arg\max\limits_m\{\mu_m(b)\}$. При цьому функцію належності $\mu_m(b)$ до контейнера з опорним вектором b_m^* та радіусом d_m^*, обчислюють за однією з формул:

$$\mu_m(b)=\exp\left(-\sum_{i=1}^{N_2} b_i \oplus b_{m,i}^* / d_m^*\right), \tag{2.5}$$

$$\mu_m(b)=1-\sum_{i=1}^{N_2} b_i \oplus b_{m,i}^* / d_m^* \tag{2.6}$$

Отже, інформаційно-екстремальний класифікатор, що оцінює належність j-го набору даних $x^{(j)}$ з N_1 ознаками до одного з класів Z, виконує кодування ознак за допомогою композиції дерев рішень і побудову вирішувальних правил у радіальному базисі двійкового простору Хеммінга. На вхід алгоритму надходить максимально допустима кількість кодувальних дерев K, навчальний набір $D=\left\{x^{(j)}, y^{(j)} \mid j=\overline{1,n}\right\}$, де n розмір набору даних, а $y^{(j)}$ це мітка j-го навчального зразка, що належить множині

$\left\{X_z^{\circ} \mid z=\overline{1,Z}\right\}$.

Навчання інформаційно-екстремального класифікатора з використанням бустингу виконують так:

1) Ініціалізація вагових коефіцієнтів $w^{(j)}=1/n$.

2) Для $k=1,...,K$ виконуються наступні кроки.

3) Генерація D_k набору навчальних даних з усього набору D з використанням функції розподілу ймовірності $P(X=x^{(j)})=w^{(j)}$.

4) Навчання дерева рішень T_k на наборі D_k.

5) Бінарне кодування вектора $x^{(j)}$ з набору даних D методом об'єднання способів рішення в деревах $T_1,...,T_k$. Результатом роботи цього етапу є бінарна матриця $\left\{b_{z,i}^{(s)} \mid i=\overline{1,N_2}; s=\overline{1,n_z}; z=\overline{1,Z}\right\}$, де N_2 кількість індукованих бінарних ознак та n_z кількість реалізацій відповідного класу X_z°, що задовольняє рівність $n=\sum_z n_z$.

6) Побудова інформаційно-екстремальних вирішувальних правил у радіальному базисі бінарного простору Хеммінга та обчислення інформаційного критерію

$$E_z^*=\max_{\{d\}} E_z(d), \qquad (2.7)$$

де $\{d\}=\left\{0,1,...,\left(\sum_i b_{z,i}\oplus b_{c,i}-1\right)\right\}$ – набір концентричних

радіусів із центром b_z;

b_z, b_c – опорні вектори розподілу даних у класі X_z° та сусідньому до нього класі X_c° відповідно, які можна розрахувати за правилом

$$b_{z,i} = \begin{cases} 1, if \ \frac{1}{n_z}\sum_{s=1}^{n_z} b_{z,i}^{(s)} > \frac{1}{Z}\sum_{c=1}^{Z}\frac{1}{n_c}\sum_{s=1}^{n_c} b_{c,i}^{(s)}; \\ 0, otherwise. \end{cases} \qquad (2.8)$$

де E_z – критерій ефективності побудови вирішувальних правил в режимі навчання для класу X_z° [44].

Для підвищення ефективності навчання загальноприйнятим способом є зведення проблеми багатокласової класифікації до серії двокласових класифікацій за принципом «один проти всіх». Для уникнення проблеми незбалансованості класів розпізнавання, обумовленої переважанням у навчальному наборі даних негативних зразків, здійснюють уведення синтетичного класу, що є альтернативним для X_z°. Синтетичний клас представлено n_z векторами з інших класів, найбільш близькими до опорного вектора b_z, де n_z – обсяг навчального набору даних класу X_z°.

7) Тестування одержаних інформаційно-екстремальних вирішувальних правил на наборі даних D та розрахування коефіцієнта помилок для кожної реалізації з D. Во-

дночас у режимі екзамену прийняття рішення про належність вектора b до одного з класів розпізнавання з алфавіту $\left\{X_z^{\circ} \mid z=\overline{1,Z}\right\}$ здійснюють за максимальним значенням функції належності $\mu_b(b)$ відповідно до виразу $\arg\max_z\{\mu_z(b)\}$. У цьому разі функцію належності $\mu_z(b)$ бінарного подання b вхідного вектора даних x до класу X_z°, оптимальний контейнер якого має опорний вектор b_z^* та радіус d_z^*, розраховують за формулою (2.5).

8) Оновлення вагових коефіцієнтів $\left\{w^{(j)}\right\}$ пропорційно помилкам розпізнавання вектора $x^{(j)}$:

$$w^{(j)} = 1 - \mu_{m'}(x^{(j)}),\ m' = y^{(j)};$$
$$w^{(j)} = \frac{w^{(j)}}{\sum_j w^{(j)}}$$

9) Якщо $\left|E_k^* - E_{k-1}^*\right| < \varepsilon$ то вихід із циклу.

Центральним питанням інформаційного синтезу вирішувальних правил є оцінювання функціональної ефективності процесу навчання моделі, що визначає максимальну асимптотичну достовірність рішень, що ухвалюють на екзамені. Як критерій функціональної ефективності (КФЕ) в методах ІЕІ-технології можуть використовуватися різні критерії, що задовольняють такі властивості інформацій-

них мір:

– інформаційна міра є величина дійсна та знакододатна як функція від імовірності;

– кількість інформації для детермінованих змінних ($p_i = 1$ або $p_i = 0$) дорівнює нулю;

– інформаційна міра має екстремум при значенні ймовірності $p_i = \frac{1}{m}$, де m – кількість якісних ознак розпізнавання.

Серед інформаційних мір для оцінювання функціональної ефективності вирішувальних правил перевагу потрібно надавати статистичним логарифмічним критеріям, що дозволяють працювати з навчальними вибірками порівняно малих обсягів. Серед таких критеріїв найбільшого використання набули ентропійні міри та інформаційна міра Кульбака [44].

Подамо нормований ентропійний КФЕ навчання системи розпізнавати реалізації класу X_m^o у вигляді:

$$E_m^{(k)} = \frac{I_m^{(k)}}{I_{\max}^{(k)}} = \frac{H_m^{(k)} - H_m^{(k)}(\gamma)}{H_m^{(k)}}, \qquad (2.9)$$

де $I_m^{(k)}$ – кількість умовної інформації, що обробляється на k-му кроці навчанні моделі розпізнавати реалізації класу X_m^o;

$I_{\max}^{(k)}$ – максимальна можлива кількість умовної інфор-

мації, одержаної на k-му кроці навчання розпізнавати реалізації одного з класів із заданого алфавіту $\{X_m^o\}, m = \overline{1, M}$;

$$H_m^{(k)} = -\sum_{l=1}^{M} p(\gamma_{l,k}) \log_2 p(\gamma_{l,k}) \tag{2.10}$$

апріорна (безумовна) ентропія, що існує на k-му кроці навчання системи розпізнавати реалізації класу X_m^o;

$$H_m^{(k)}(\gamma) = \\ = -\sum_{l=1}^{M} \sum_{m=1}^{M} p(\gamma_{l,k}) p(\mu_{m,k} / \gamma_{l,k}) \log_2 p(\mu_{m,k} / \gamma_{l,k}) \tag{2.11}$$

апостеріорна (умовна) ентропія, що характеризує залишкову невизначеність після k-го кроку навчання системи розпізнавати реалізації класу X_m^o;

d – дистанційна міра, що визначає радіуси гіперсферичних контейнерів, побудованих у радіальному базисі простору Хеммінга;

$p(\gamma_{l,k})$ – безумовна ймовірність прийняття на k-му кроці навчання гіпотези $\gamma_{l,k}$;

$p(\mu_{m,k} / \gamma_{l,k})$ – апостеріорна ймовірність прийняття на k-му кроці навчанні рішення $\mu_{m,k}$ за умови, що прийнята гіпотеза $\gamma_{l,k}$.

Для двохальтернативної системи оцінювання (M = 2) та рівноймовірних гіпотез, що характеризує найбільш важкий у статистичному сенсі випадок прийняття рішень, після відповідної підстановки ентропій (2.10) і (2.11) у вираз (2.4.1) та заміни відповідних апостеріорних імовірностей на апріорні за формулою Байєса ентропійний критерій набирає вигляду:

$$E_m^{(k)} = 1 + \frac{1}{2}\left(\frac{\alpha_m^{(k)}(d)}{\alpha_m^{(k)}(d) + D_{2,m}^{(k)}(d)} \log_2 \frac{\alpha_m^{(k)}(d)}{\alpha_m^{(k)}(d) + D_{2,m}^{(k)}(d)} + \right.$$
$$+ \frac{\beta_m^{(k)}(d)}{D_{1,m}^{(k)}(d) + \beta_m^{(k)}(d)} \log_2 \frac{\beta_m^{(k)}(d)}{D_{1,m}^{(k)}(d) + \beta_m^{(k)}(d)} +$$
$$+ \frac{D_{1,m}(d)}{D_{1,m}^{(k)}(d) + \beta_m^{(k)}(d)} \log_2 \frac{D_{1,m}(d)}{D_{1,m}^{(k)}(d) + \beta_m^{(k)}(d)} +$$
$$\left. + \frac{D_{2,m}^{(k)}(d)}{\alpha_m^{(k)}(d) + D_{2,m}^{(k)}(d)} \log_2 \frac{D_{2,m}^{(k)}(d)}{\alpha_m^{(k)}(d) + D_{2,m}^{(k)}(d)} \right), \quad (2.12)$$

де $\alpha_m^{(k)}(d)$ – ймовірність помилок першого роду – точністна характеристика рішення на k-му кроці навчання;

$\beta_m^{(k)}(\mathrm{d})$ – ймовірність помилок другого роду;

$D_{1,m}^{(k)}(\mathrm{d})$ – перша достовірність (чутливість);

$D_{2,m}^{(k)}(\mathrm{d})$ – друга достовірність (специфічність);

d – дистанційна міра, що визначає радіуси гіперсферичних контейнерів, побудованих у радіальному базисі

простору Хеммінга.

Оскільки точнісні характеристики є функціями відстані вершин еталонних векторів-реалізацій образу від геометричних центрів контейнерів відповідних класів розпізнавання, то критерій (2.12) в IEI-технології необхідно розглядати як нелінійний і взаємно неоднозначний функціонал від точнісних характеристик, що потребує знаходження в процесі навчання робочої (допустимої) області для його визначення.

Розглянемо модифікацію диференційної інформаційної міри Кульбака, що подають як добуток відношення правдоподібності Λ на міру відхилень відповідних розподілень імовірностей.

У праці [42] розглянуто логарифмічне відношення повної ймовірності $P_{t,m}^{(k)}$ правильного прийняття рішень щодо належності реалізацій класів X_m^o і X_c^o контейнеру $K_{m,k}^o \in X_m^o$ до повної ймовірності помилкового прийняття рішень $P_{f,m}^{(k)}$, що для двохальтернативної системи оцінювання рішень має такий вигляд:

$$\Lambda = \log_2 \frac{P_{t,m}^{(k)}}{P_{f,m}^{(k)}} = ,$$

$$= \log_2 \frac{p(\mu_m)\,p(\gamma_{1,k}/\mu_m) + p(\mu_c)\,p(\gamma_{2,k}/\mu_c)}{p(\mu_m)\,p(\gamma_{2,k}/\mu_m) + p(\mu_c)\,p(\gamma_{1,k}/\mu_c)} \qquad (2.13)$$

де $p(\mu_m)$ – безумовна ймовірність появи реалізації класу X_m^o;

$p(\mu_c)$ – безумовна ймовірність появи реалізації класу X_c^o;

$\gamma_{1,k}$ – гіпотеза про належність контейнеру $K_{m,k}^o \in X_m^o$ реалізації класу X_m^o;

$\gamma_{2,k}$ – альтернативна гіпотеза.

Із врахуванням (2.13) у разі допущення згідно з принципом Лапласа – Бернуллі, що $p(\mu_m) = p(\mu_c) = 0,5$ і після переозначення апріорних умовних імовірностей відповідними точнісними характеристиками загальна міра Кульбака остаточно набуває вигляду:

$$J_m^{(k)} = \log_2 \frac{P_{t,m}^{(k)}}{P_{f,m}^{(k)}} * [P_{t,m}^{(k)} - P_{f,m}^{(k)}] =$$

$$= 0,5 \log_2 \left(\frac{D_{1,m}^{(k)}(d) + D_{2,m}^{(k)}(d)}{\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)} \right) *$$

$$* \left[(D_{1,m}^{(k)}(d) + D_{2,m}^{(k)}(d)) - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)) \right] =$$

$$= \log_2 \left(\frac{2 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d))}{\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)} \right) * \quad (2.14)$$

$$* \left[1 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)) \right]$$

Нормовану модифікацію критерію (2.14) можна подати у вигляді:

$$E_{K,m}^{(k)} = \frac{J_m^{(k)}}{J_{\max}^{(k)}},$$

де $J_{\max}^{(k)}$ – значення критерію при $D_{1,m}^{(k)}(d) = D_{2,m}^{(k)}(d) = 1$ і відповідно $\alpha_m^{(k)}(d) = \beta_m^{(k)}(d) = 0$ для формули (2.14).

У задачах оптимізації параметрів функціонування моделі в процесі навчання за IEI-технологією нормування критеріїв оптимізації не є обов'язковим, оскільки тут розв'язують задачу пошуку екстремальних значень параметрів навчання, що відповідають глобальному максимуму КФЕ в робочій області його визначення. Але нормування критеріїв оптимізації є доцільним під час порівняльного аналізу результатів досліджень та під час оцінювання ступеня близькості реальної системи до потенційної.

Розглянемо процедуру обчислення модифікації ентропійного інформаційного КФЕ за Шенноном для двохальтернативного рішення при рівноймовірних гіпотезах згідно з формулою (2.14). Оскільки інформаційний критерій є функціоналом від точнісних характеристик, то при мінімальному обсязі репрезентативної навчальної вибірки необхідно користуватися їх оцінками:

$$D_{1,m}^{(k)}(d) = \frac{K_{1,m}^{(k)}}{n_{\min}},\ \alpha_m^{(k)}(d) = \frac{K_{2,m}^{(k)}}{n_{\min}},$$

$$\beta_m^{(k)}(d) = \frac{K_{3,m}^{(k)}}{n_{\min}},\ D_{2,m}^{(k)}(d) = \frac{K_{4,m}^{(k)}}{n_{\min}}, \qquad (2.15)$$

де $K_{1,m}^{(k)}$ – кількість подій, що означають належність реалізацій образу контейнера $K_{1,mk}^{o}$, якщо дійсно $\{x_1^{(j)}\} \in X_1^o$;

$K_{2,m}^{(k)}$ – кількість подій, що означають неналежність реалізацій образу контейнера $K_{1,m}^{o}$, якщо дійсно $\{x_1^{(j)}\} \in X_1^o$;

$K_{3,m}^{(k)}$ – кількість подій, що означають належність реалізацій образу контейнера $K_{1,m}^{o}$, якщо вони насправді належать класу X_2^o;

$K_{4,m}^{(k)}$ – кількість подій, що означають неналежність реалізацій образу контейнера $K_{1,m}^{o}$, якщо вони насправді належать класу X_2^o;

n_{min} – мінімальний обсяг репрезентативної навчальної вибірки.

Після підстановки відповідних позначень (2.15) в (2.12) отримаємо робочу модифіковану формулу для обчислення в межах IEI-технології ентропійного інформаційного КФЕ навчання моделі розпізнаванню реалізацій класу X_m^o

$$E_m^{(k)} = 1 + \frac{1}{2}\left(\frac{K_{1,m}^{(k)}}{K_{1,m}^{(k)} + K_{3,m}^{(k)}} \log_2 \frac{K_{1,m}^{(k)}}{K_{1,m}^{(k)} + K_{3,m}^{(k)}} + \right.$$

$$+ \frac{K_{2,m}^{(k)}}{K_{2,m}^{(k)} + K_{4,m}^{(k)}} \log_2 \frac{K_{2,m}^{(k)}}{K_{2,m}^{(k)} + K_{4,m}^{(k)}} +$$

$$+ \frac{K_{3,m}^{(k)}}{K_{1,m}^{(k)} + K_{3,m}^{(k)}} \log_2 \frac{K_{3,m}^{(k)}}{K_{1,m}^{(k)} + K_{3,m}^{(k)}} +$$

$$\left. + \frac{K_{4,m}^{(k)}}{K_{2,m}^{(k)} + K_{4,m}^{(k)}} \log_2 \frac{K_{4,m}^{(k)}}{K_{2,m}^{(k)} + K_{4,m}^{(k)}} \right) \qquad (2.16)$$

Робоча модифікація критерію Кульбака після відповідної підстановки оцінок (2.15) у вираз (2.14) набуває вигляду

$$J_m^{(k)} = \frac{1}{n} \log_2 \left\{ \frac{2n + 10^{-r} - \left[K_{2,m}^{(k)} + K_{3,m}^{(k)} \right]}{\left[K_{2,m}^{(k)} + K_{3,m}^{(k)} \right] + 10^{-r}} \right\} *$$

$$\left[n - (K_{2,m}^{(k)} + K_{3,m}^{(k)}) \right], \qquad (2.17)$$

де $K_{2,m}^{(k)}$ – кількість реалізацій класу X_m^o, що не знаходяться в побудованому на k-му кроці навчання контейнері цього класу;

$K_{3,m}^{(k)}$ – кількість реалізацій класу X_c^o, що знаходяться в побудованому на k-му кроці навчання контейнері класу

X_m^o.

Отже, інформаційні критерії (2.12) і (2.14) є функціоналами як від точнісних характеристик ухвалюваних рішень, так і від дистанційних критеріїв, тобто є узагальненням відомих статистичних і детермінованих критеріїв оптимізації параметрів функціонування системи прийняття рішень. До того ж, метод синтезу вирішувальних правил ґрунтується на використання обчислювально-ефективних операцій порівнювання з порогом та обчислення відстані Хемінга. Це дозволяє підвищити оперативність оцінювання функціональної ефективності моделі, що особливо актуально під час налаштування екстрактора ознак.

2.4. Модель і метод синтезу регресійних вирішувальних правил

У найпростішому випадку регресійні вирішувальні правила будують у вигляді лінійної моделі, вагові коефіцієнти якої знаходять згідно з методом найменших квадратів. Однак у багатозадачних моделях зі спільним ознаковим описом більш ефективним вважають використання регресійної моделі як мінімум з одним прихованим шаром.

Розглянемо регресійну модель у вигляді нейромережі прямого поширення з одним прихованим шаром. Водночас набір навчальних даних описано множиною $\{(x^{(j)}, y^{(j)}) \mid x^{(j)} \in R^N, \quad y^{(j)} \in R^M, 1 \le j \le n\}$, де $x^{(j)} = (x_1^{(j)}, x_2^{(j)}, \ldots, x_N^{(j)})^T$. Вихідна змінна $y^{(j)} \in R^M$ від-

повідає вектору цілі, тобто є міткою навчального зразка. Нейромережа з R адитивними прихованими вузлами та функцією активації $\varphi(x)$ може бути подана у вигляді системи рівнянь

$$\sum_{r=1}^{R} \beta_j \varphi(w_r^T x^{(j)} + b_r) = o^{(j)}, 1 \le j \le n,$$

де $w_r = (w_{r,1}, w_{r,2}, \ldots, w_{r,N})^T$ – вектор вагових коефіцієнтів, що зв'язує вхідний шар із r-м прихованим вузлом;

b_r – зміщення r-го прихованого вузла;

$\beta_r = (\beta_{r,1}, \beta_{r,2}, \ldots, \beta_{r,M})$ – вектор вагових коефіцієнтів, що зв'язує вихідний шар з r-м прихованим вузлом;

$o^{(j)}$ – вихід мережі для вхідного вектора $x^{(j)}$;

$\varphi(x)$ – функція активації.

Нейромережа з R прихованими вузлами може відтворити ці n зразків із заданою точністю, якщо всі параметри можуть вільно коригуватися, тобто існують, β_r, w_r і b_r. Вищенаведені вирази можуть бути компактно переписані як матрична рівність

$$H\beta = Y, \qquad (2.18)$$

де

$$H = \begin{bmatrix} \varphi(w_1^T x^{(1)} + b_1) & \dots & \varphi(w_R^T x^{(1)} + b_R) \\ \dots & \dots & \dots \\ \varphi(w_1^T x^{(n)} + b_1) & \dots & \varphi(w_R^T x^{(n)} + b_R) \end{bmatrix}_{n \times R},$$

$$\beta = \begin{bmatrix} \beta_1 \\ \dots \\ \beta_R \end{bmatrix}_{R \times M}, \; Y = \begin{bmatrix} y^{(1)} \\ \dots \\ y^{(n)} \end{bmatrix}_{n \times M}$$

де H – матриця виходу прихованого шару.

Навчання цієї мережі можливо здійснити з використанням алгоритму зворотного поширення помилки. Однак нестабільність результатів, чутливість до гіперпараметрів навчання та повільна збіжність обумовлюють непридатність даного алгоритму за умов ресурсних обмежень. До того ж, важливим гіперпараметром є кількість нейронів прихованого шару, що найважче піддаються оцінюванню. Одним із рішень є інкрементальне ускладнення моделі до моменту досягнення необхідної точності за навчальною та тестовою вибірками. Для прискорення навчання можна скористатися принципами машин екстремального навчання (Extreme Learning Machine), де вагові коефіцієнти прихованого шару ініціалізуються за допомогою генерації випадкових чисел або ортогональних випадкових матриць. Вагові коефіцієнти вихідного шару визначаються шляхом псевдоінверсії, що може бути обчислена на основі алгоритму ортогоналізації Грамма – Шмідта. Збіжність ітераційного алгоритму ортогоналізації Грамма – Шмідта для по-

шуку розв'язання задачі найменших квадратів було доведено в праці Li Ying [28]. Розглянемо основні кроки навчання машини екстремального навчання з інкрементальним збільшенням кількості прихованих вузлів і визначенням вагових коефіцієнтів вихідного шару на основі алгоритму ортогоналізації Грамма – Шмідта:

1. Встановити максимальну кількість ітерацій $L_{\max}$, початкову залишкову помилку $E = [y^{(1)}, ..., y^{(j)}]^T$ та допустиму помилку E_0.

2. Для $L = 1 \ldots L_{\max}$ виконати наступні кроки.

3. Збільшити кількість прихованих вузлів на один : $r = r + 1$.

4. Випадково згенерувати один прихований нейрон та розрахувати вектор його виходу h_r.

5. Якщо r = 1 тоді $v_r = h_r$ інакше

$$v_r = h_r - \frac{\langle v_1, h_r \rangle}{\langle v_1, v_1 \rangle} v_1 - \frac{\langle v_2, h_r \rangle}{\langle v_2, v_2 \rangle} v_2 - \ldots$$

$$- \frac{\langle v_{r-1}, h_r \rangle}{\langle v_{r-1}, v_{r-1} \rangle} v_{r-1} \qquad (2.19)$$

6. Якщо $\| v_r \| \geq \varepsilon$, то виконувати обчислення вихідної ваги для нового прихованого вузла $\beta_r = v_r^T E / (v_r^T v_r)$ та розрахувати нову залишкову помилку $E = E - v_r \beta_r$ інакше

$r = r - 1$.

7. Якщо $\| E \| < E_0$ або помилка на тестовій вибірці не зменшується упродовж заданої кількості ітерацій, то потрібно вийти з основного циклу роботи.

Наведений алгоритм повинен успішно працювати за умови добре підготованого ознакового опису, інакше може знадобитися велика кількість прихованих вузлів. Тому потрібно розглянути також алгоритми навчання багатошарових регресійних моделей, що дозволяють зменшити загальну кількість нейронів. Одночасно для економії оперативної пам'яті необхідно розглянути можливість навчання в міні-пакетному режимі або в режимі он-лайн (стохастичний алгоритм). Крім алгоритму зворотного поширення помилки, що ґрунтується на алгоритмі градієнтного спуску, можливі альтернативні варіанти. Найпростішим варіантом є використання псевдоінверсії Мура – Пенроуза для розв'язання лінійних рівнянь типу $Ax = b$ на основі сингулярного розкладу матриці. У разі навчання в міні-пакетному чи онлайн режимах обчислення псевдоінверсії є досить швидкою операцією, оскільки розмір матриць буде порівняно невеликим. Це дозволяє знайти необхідну корекцію параметрів нейрона для досягнення бажаного виходу та одночасно оновити ціль для попереднього нейронного шару [45]. Однак у цьому разі існує потреба у використанні функцій активації $activation(\bullet)$, що мають обернене відображення з такою самою областю визначення. До таких функцій можуть належати ReLU-подібні функції типу LeakyReLU, Softplus, S-ReLU та інші.

На рисунку 2.14 показано псевдокод алгоритму прямого поширення сигналу для нейронного шару з K-нейронів.

для k від 1 *до* K :
    для i *від* 1 *до* N :
        $o_k = o_k + x_i * w_i + b_i$
    $o_k = activation(\ o_k\ )$

Рисунок 2.14 – Псевдокод алгоритму прямого поширення сигналу для нейронного шару

На вхід нейронного шару надходить N-вимірний вектор спостереження $x = [x_1, .., x_i, .., x_N]$. Вихід нейронного шару формує K-вимірний вектор $o = [o_1, .., o_k, .., o_K]$. Водночас кожен i-й вхід нейрона має ваговий коефіцієнт w_i та коефіцієнт зміщення b_i.

На рисунку 2.15 показано псевдокод алгоритму зворотного поширення сигналу для корекції вагових коефіцієнтів та оновлення цілі для попереднього шару нейромережі. На вхід алгоритму зворотного поширення сигналу в заданому нейронному шарі надходить вхідний вектор нейронного шару $x = [x_1, .., x_i, .., x_N]$ та очікуваний (бажаний) вектор вихідного сигналу нейронного шару $y = [y_1, .., y_k, .., y_K]$. Водночас вихід нейрона формується під впливом кожного його входу, тому цільове значення $y_{k,i}$ для і-го входу k-го нейрона є складовою результату застосування оберненої

функції активації до цільової змінної y_k. Для простоти обчислень цільова змінна рівномірно розподілена між кожним входом нейрона, тобто $y_{k,i} = activation^{-1}(y_k) / N$.

для k від 1 до K:

для i від 1 до N:

$$y_{k,i} = activation^{-1}(y_k) / N$$

$$o_{k,i} = x_i * w_{k,i} + b_{k,i}$$

$$\begin{bmatrix} \Delta w_{k,i} \\ \Delta b_{k,i} \end{bmatrix} = \begin{bmatrix} x_i & 1 \end{bmatrix}^{\dagger} \times \left[\left(o_{k,i} - y_{k,i} \right) \right]$$

$$\tilde{w}_{k,i} = w_{k,i} - \alpha \Delta w_{k,i}$$

$$\tilde{b}_{k,i} = b_{k,i} - \alpha \Delta b_{k,i}$$

$$\tilde{x}_{k,i} = \frac{y_{k,i} - \tilde{b}_{k,i}}{\tilde{w}_{k,i}}$$

для i від 1 до N:

$$\tilde{x}_i = \frac{1}{K} \sum_{k=1}^{K} \tilde{x}_{k,i}$$

Рисунок 2.15 – Псевдокод алгоритму зворотного поширення сигналу для нейронного шару мережі

Якщо ввести бажаний ваговий коефіцієнт $\tilde{w}_{k,i}$ та бажане зміщення $\tilde{b}_{k,i}$ і-го входу k-го нейрона, що забезпечують бажану складову виходу $y_{k,i}$, тоді можна скласти таку си-

стему рівнянь:

$$\begin{cases} x_i * w_{k,i} + b_{k,i} = o_{k,i} \\ x_i * \tilde{w}_{k,i} + \tilde{b}_{k,i} = y_{k,i} \end{cases} \Rightarrow \begin{cases} x_i * \tilde{w}_{k,i} + \tilde{b}_{k,i} = y_{k,i} \\ \tilde{w}_{k,i} = w_{k,i} - \Delta w_{k,i} \\ \tilde{b}_{k,i} = b_{k,i} - \Delta b_{k,i} \end{cases}$$

$$\Rightarrow x_i * \Delta w_{k,i} + \Delta b_{k,i} = (o_{k,i} - y_{k,i}),$$

де $\Delta w_{k,i}$, $\Delta b_{k,i}$ – необхідний сигнал корекції для одержання бажаної складової вихідного сигналу $y_{k,i}$.

Вище наведені рівняння дозволяють записати задачу тренування параметрів нейрона у формі, що дозволяє застосувати псевдоінверсію Мура – Пенроуза.

$$\begin{bmatrix} x_i & 1 \end{bmatrix} \times \begin{bmatrix} \Delta w_{k,i} \\ \Delta b_{k,i} \end{bmatrix} = \begin{bmatrix} (o_{k,i} - y_{k,i}) \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} \Delta w_{k,i} \\ \Delta b_{k,i} \end{bmatrix} = \begin{bmatrix} x_i & 1 \end{bmatrix}^{\dagger} \times \begin{bmatrix} \left(o_{k,i} - y_{k,i}\right) \end{bmatrix}$$

Для забезпечення стабільності навчання та уникнення помітного впливу шумового навчального зразка $x = [x_1, .., x_i, .., x_N]$ на результат навчання задають порівняно мале значення швидкості навчання α. Значення α залежить від кількості епох навчання, обсягу навчальних даних і для більшості практичних задач може бути обране з діапазону [0,01-0,0001].

Скориговані значення вагового коефіцієнта $\tilde{w}_{k,i}$ та

зміщення $\tilde{b}_{k,i}$ можуть бути використані для обчислення скорегованого вхідного сигналу $\tilde{x}_{k,i}$, що забезпечуватиме бажане значення складової виходу $y_{k,i}$. Водночас цільовий сигнал попереднього шару нейронної мережі може бути обчислений як усереднене за нейронами даного шару значення скоригованих входів

$$\tilde{x}_i = \frac{1}{K}\sum_{k=1}^{K}\tilde{x}_{k,i}\ .$$

Отже, у найпростішому випадку регресійні вирішувальні правила можуть бути сформовані у вигляді лінійної комбінації ознак розпізнавання, але в більш складному випадку існує потреба в багатошаровій архітектурі. Синтез цих правил полягатиме у розв'язанні задачі найменших квадратів на основі алгоритму градієнтного спуску, псевдо-інверсії Мура – Пенроуза або ортогоналізації Грамма – Шмідта. Водночас безградієнтні алгоритми навчання менш чутливі до гіперпараметрів.

2.5. Методи точного налаштування моделі аналізу даних

Традиційним підходом до точного налаштування глибоких моделей аналізу даних є використання алгоритму зворотного поширення помилки. Однак умовою його застосування є диференційованість цільової функції та

функцій, якими описують складові частини моделі. Це ускладнює його застосування до гібридних моделей з недиференційованими елементами. Тому неможливо здійснювати точне налаштування моделей із пороговими функціями та бінарними відношеннями прямо попри те, що саме ці елементи найбільш є обчислювально-ефективними. До того ж, ефективність методів зворотного поширення помилки істотно залежить від оптимальності вибору швидкості навчання на кожному етапі навчання. Тому значну увагу приділяють планувальникам навчання з різними політиками зміни швидкості навчання, більш просунуті версії яких є інтелектуальними алгоритми мета-навчання.

У завданнях глобальної оптимізації набули популярності мета-евристичні пошукові алгоритми, що не висувають умов щодо диференційованості чи опуклості досліджуваної функції. Найбільш ефективними з точки зору точності рішень вважають популяційні алгоритми, до яких належать алгоритм косяку риб (Fish School Search Algorithm), алгоритм зграї птахів (Bird Swarm Algorithm), алгоритм рою частинок (Particle Swarm Optimization Algorithm), гравітаційний алгоритм (Gravitational Search Algorithm), еволюційний алгоритм (Evolutionary Algorithm), алгоритм диференційної еволюції (Differential Evolution Algorithm) та інші [46].

Істотним недоліком популяційних алгоритмів є значні ресурсні потреби для розв'язання задач з великою кількістю параметрів. Наприклад, у задачі оптимізації глибоких нейронних мереж популяційному алгоритму знадобиться зберігати в пам'яті популяцію таких мереж із модифікова-

ними параметрами. Сучасні глибокі мережі займають великий обсяг пам'яті та потребують істотних обчислювальних ресурсів для обчислення фітнес-функції. Однак у задачі точного налаштування моделі область пошуку (exploration space) дещо звужена. Тому можливий ряд модифікацій алгоритму, обмежень розміру популяції та кроку модифікації рішень, що можуть забезпечити практичну придатність популяційних алгоритмів для точного налаштування глибоких моделей. Одним з характерних представників цього підходу є еволюційний алгоритм з обмеженою оцінкою (limited evaluation evolutionary algorithms). Зокрема, зменшити область пошуку можна також за рахунок пошарової оптимізації (рис. 1.6 б) [47].

Еволюційний алгоритм із обмеженою оцінкою може здійснювати оцінювання популяції на обмеженій кількості вхідних навчальних зразків подібно до алгоритму стохастичного градієнтного спуску, що здійснює корекцію вагових коефіцієнтів мережі після оброблення міні-пакета (mini-batch). Використання міні-пакета замість одного зразка зменшує ймовірність значного впливу на результат навчання шумових зразків. При цьому в еволюційному алгоритмі передбачено модифікацію фітнес-функції для уникнення втрати на етапі селекції агентів популяції, що є більш ефективними в глобальному сенсі, але менш пристосованими до конкретного вхідного міні-пакета. Під час обчислення фітнес-функції враховують як ефективність агентів для поточного міні-пакета, так і ефективність предків окремих агентів на інших міні-пакетах.

В еволюційному алгоритмі з обмеженою оцінкою ви-

користовують окремо фітнес-функцію для випадку репродукції схрещуванням (2.20) та окремо фітнес-функція для асексуальної репродукції (2.21) відповідно:

$$f' = \frac{f_{p_1} + f_{p_2}}{2}(1-d) + f, \tag{2.20}$$

$$f' = f_{p_1}(1-d) + f, \tag{2.21}$$

де f' – скоригована індивідуальна фітнес-функція агента популяції;

f – індивідуальна фітнес-функція агента популяції для поточного міні-пакета;

f_{p_1} – фітнес-функція першого предка;

f_{p_2} – фітнес-функція другого предка;

d – константне значення затухання впливу історії предків.

Агенти популяції прямо кодують параметри моделі $W \in \mathbb{R}^C$, де C – кількість параметрів моделі. Операції репродукції та мутації напряму модифікують вектор параметрів кожного агента. Оператор репродукції схрещуванням може бути рівномірний (uniform crossover), під час якого нащадок $W_{u,i}$ утворюється методом випадкового копіювання елементів обох батьківських агентів (2.22). Також оператор репродукції схрещуванням може бути поданий у вигляді процедури арифметичного усереднення (Arithmetic crossover) параметрів від батьківських агентів

(2.23) для формування дочірнього агента W_a. Формули операторів репродукції показано нижче:

$$W_{u,i} = \begin{cases} W_{1,i}, \ \textit{з ймовірністю} \ 0,5; \\ W_{2,i}, \ \textit{у протилежному випадку}; \end{cases} \quad (2.22)$$

$$W_a = \frac{1}{2}\left(W_1 + W_2\right). \quad (2.23)$$

Оператор мутації здійснює додавання шуму з Гаусовим розподілу $\mathrm{N}(0,1)$ до параметрів батьківського агента $W_m = W_1 + \sigma \mathrm{N}(0,1)$, де σ – сила мутації. При цьому сила мутації σ є важливим гіперпараметром, який бажано зменшувати з кожною епохою, подібно до того, як в алгоритмі градієнтного спуску зменшується швидкість навчання. На рисунку 2.16 показано псевдокод еволюційного алгоритму з обмеженим оцінюванням.

Для реалізації точного налаштування моделі пропонують першу популяцію формувати методом застосування оператора мутації до наявного рішення. При цьому один з агентів першої популяції пропонується сформувати за допомогою простого копіювання наявного (вхідного) рішення.

Оскільки точне налаштування параметрів екстрактора ознакового опису вимагає використання зменшеного кроку модифікації рішень, то процес пошуку глобального оптимуму цільової функції зазвичай триває упродовж великої кількості ітерацій. У разі трудомісткого обчислення цільо-

вої функції та за великої кількості параметрів доцільніше використовувати траєкторні метаевристичні алгоритми, що оперують лише одним рішенням на кожній ітерації пошуку.

Доки $iter < \max\_iter$ *виконувати*:
Вибірку міні-пакету навчальних зразків
Оцінювання кожного агента популяції
Обчислення модифікації фітнес – функції для кожного агента, що враховує ефективність предків
Селекцію методом рулетки
Формування нащадків методом схрещування або мутації
Зменшення сили мутації методом помноження на константу послаблення
$iter \leftarrow iter + 1$.

Рисунок 2.16 – Псевдокод еволюційного алгоритму з обмеженим оцінюванням

Серед траєкторних алгоритмів найбільш ефективними вважають алгоритм симуляції відпалу (Simulated Annealing), його макро- та мікроканонічні реалізації, алгоритм погодження порогу (Threshold accepting) та алгоритм сходження на пагорб (Hill Climbing) [46, 47]. До того ж, алгоритм симуляції відпалу є більш універсальним і при деяких налаштуваннях придатний як для глобальної, так і для локальної оптимізації параметрів моделі.

На рисунку 2.17 показано псевдокод алгоритму симу-

ляції відпалу, на кожній ітерації якого виконують обчислення цільового критерію f() методом пропускання розміченого навчального набору даних через модель аналізу даних та розрахування критерію ефективності [46].

$s_{current} \leftarrow$ *ініціалізація _ початкового _ рішення*()
$s_{best} \leftarrow s_{current}$
$T \leftarrow T_0$
$c \leftarrow \varepsilon,\ 0 < \varepsilon < 1$
для i від 1 *до epochs* _ max)
    $s_i \leftarrow$ *формування _ сусіднього _ рішення*$(s_{current})$
    якщо $f(s_i) \geq f(s_{current})$, *то*
        $s_{current} \leftarrow s_i$
        якщо $f(s_i) \geq f(s_{best})$, *то* $s_{best} \leftarrow s_i$

    інакше $\exp\left(\frac{f(s_{current}) - f(s_i)}{T}\right) > uniform\_random(0,1)$ *і* $s_{current} \leftarrow s_i$
    $T \leftarrow c \times T$
повернення s_{best}

Рисунок 2.17 – Псевдокод алгоритму симуляції відпалу

Початкове рішення утворюється за допомогою процедури «ініціалізація_початкового_рішення()», що може бути реалізована на основі алгоритмів навчання без учителя. Водночас ефективність алгоритму симуляції відпалу залежить від реалізації процедури «формування_сусіднього_рішення()» для формування нового рішення Si на і-ій ітерації алгоритму.

Аналіз псевдокоду на рисунку 2.17 показує, що поточне рішення $S_{current}$, щодо якого відбувається пошук нових кращих рішень S_{best}, оновлюється у разі знаходження нового рішення, що збільшує критерій, або випадково з розпо-

ділу Гіббса. Для формування нового рішення пропонують використовувати найпростіший неадаптивний алгоритм, який можна надати у вигляді такої формули:

$$S_{current} = S_{current} +, \\ +uniform\_random(-1,1) \cdot step\_size \qquad (2.24)$$

де uniform_random – функція генерації випадкових чисел із рівномірного розподілу з заданого діапазону; step_size – це розмір діапазону пошуку нових рішень, сусідніх з $S_{current}$.

Отже, у разі виконання диференційованості функції, якою описують модель аналізування даних, ефективним методом точного налаштування є використання алгоритмів зворотного поширення помилки. У разі порушення цієї умови, що часто буває під час синтезу гібридних моделей, пошук необхідно здійснювати метаевристичними популяційними або траєкторними алгоритмами. Ці алгоритми здатні розв'язувати оптимізаційну задачу типу «чорної шухляди» і гармонізувати різнорідні частини гібридної моделі під час точної настройки.

2.6. Критерії та методи оптимізації параметрів функціонування системи аналізу даних

Система аналізу даних містить такі елементи: засоби збору, зберігання та підготовки даних, моделі аналізу даних і алгоритми навчання, обчислювальне середовище роз-

гортання моделей аналізу даних, а також алгоритми оцінювання ефективності процесу й результату навчання. Система аналізу даних має ряд регульованих параметрів, що впливають на ефективність в інформаційному та вартісному сенсах. На рисунку 2.18 показано, що покращення точнісних характеристик системи аналізу даних пов'язане зі збільшенням накладних витрат, наприклад, на розмітку та зберігання даних або на оптимізацію моделі. До того ж, обсяг цих витрат обмежується ресурсами, доступних у системі, та допустимим рівнем втрат, що виникають унаслідок помилкових рішень.

Необхідно враховувати, що навіть при одній і тій самій точності можна регулювати чутливість моделі так, щоб віддавати перевагу помилкам першого роду (внаслідок пропуску події) чи помилкам другого роду (внаслідок хибних спрацювань). Якщо ці помилки не є рівнозначними, тобто мають різну ціну, то це дозволяє здійснювати додаткову оптимізацію втрат.

Параметри функціонування системи аналізу даних можна поділити на чотири основні групи:

– гіперпараметри, що впливають на ємність моделі аналізу даних;

– гіперпараметри, що впливають на поведінку алгоритму навчання;

– параметри збору, зберігання та попереднього оброблення даних;

– параметри середовища розгортання моделей.

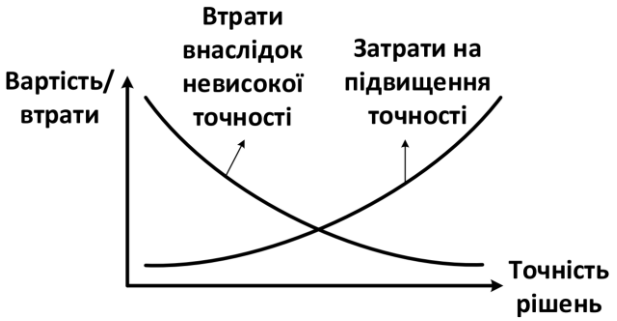


Рисунок 2.18 – Ілюстрація до знаходження оптимального відношення рівня точності системи аналізу даних та вартості досягнення цього рівня

До гіперпараметрів, що впливають на процес попередньої обробки, можуть належати частота опитування сенсорів, розподільна здатність сенсорних систем, параметри аугментації даних тощо. До гіперпараметрів, що впливають на ємність моделі, належать кількість нейронів, кількість шарів, розмір композиції моделей тощо. До параметрів, що впливають на поведінку алгоритму, належать швидкість навчання, коефіцієнт зниження швидкості навчання, параметри регуляризації та ін. Під час оптимізації гіперпараметрів оцінювання їх ефективності здійснюють на зовнішній валідаційній вибірці, що не брала участі в навчанні. На рисунку 2.19 показано, що проста модель може бути занадто грубою, для коректного опису даних або занадто складною, що призводить до втрати узагальнювальної здатності.

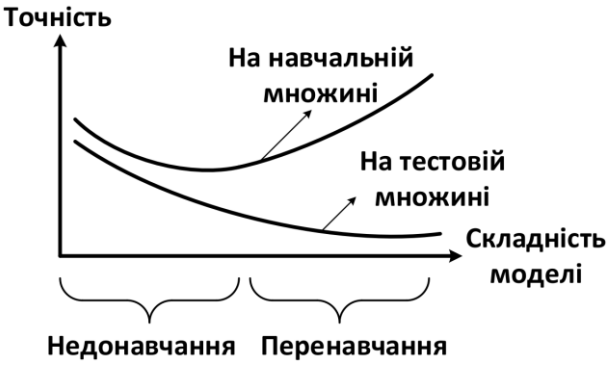


Рисунок 2.19 – Ілюстрація до знаходження оптимального відношення рівня якості сервісу та вартості досягнення цієї якості

Оптимізація складності моделі полягає в забезпеченні максимальної достовірності рішень на валідаційній вибірці за умов обмеженого обсягу ресурсів. Одночасно регулювання швидкості навчання й розміру міні-пакета дозволяє знайти компроміс між швидкістю збіжності та ймовірністю застрявання в локальних екстремумах функції оптимізації.

Для системи аналізу даних параметри збору, зберігання і попереднього оброблення даних є не менш важливими, ніж гіперпараметри моделі. Наприклад, регулюючи частоту опитування та роздільну здатність сенсорів можна впливати як на інформаційну спроможність системи, так і на ресурсні потреби алгоритмів. Крім того, збільшення кількості розмічених даних дозволяє збільшити узагальнювальну здатність глибоких моделей, хоча й потребує накладних витрат, пов’язаних з розміткою та обробленням.

Надмірне збільшення накладних затрат на досягнення малопомітного приросту точнісних характеристик системи в більшості практичних задач є неприйнятним. Тому загалом критерій оптимізації параметрів функціонування системи аналізу даних повинен мати вигляд:

$$Q = (ефект) / (витрати). \qquad (2.25)$$

Критерій ефективності (2.25) можна розглядати як основу синтезу різноманітних критеріїв ефективності системи аналізу даних. До того ж, якщо оцінювати ефективність лише з точки зору отримуваного прибутку, то загальну ефективність моделі аналізу даних можна подати у вигляді формули

$$Q_c = (C_v - C_s) / C_{vi},$$

де C_v – результат використання системи (реальний дохід);

C_s – витрати на створення та експлуатацію системи;

C_{vi} – результат застосування системи в разі виконання всіх функцій і за відсутності витрат на їх здійснення (ідеальний випадок).

Загалом завдання оптимізації параметрів системи аналізу даних є багатокритеріальним, де частинні критерії є попарно суперечливими, мають різну розмірність та є нелінійними функціями контрольованих характеристик і конфігурацій системи. Тому необхідно розглянути можливість приведення (згортки) вектора частинних критеріїв до

одного комплексного критерію.

Для надання рівномірності впливу кожного з частинних критеріїв на значення згортки необхідно вирівняти діапазони зміни значень частинних критеріїв методом масштабування та зведення їх значень до безрозмірної шкали [0,1] за правилом

$$k_i^{'} = \begin{cases} 0, k_i \le k_i^{\min}; \\ \dfrac{(k_i - k_i^{\min})}{(k_i^{\max} - k^{\min})}, k_i^{\min} < k_i < k_i^{\max} \\ 1, k_i > k_i^{\max}; \end{cases}$$

де $k_i^{\min}$, $k_i^{\max}$ – відповідно нижня та верхня межі допустимої області значень i-го частинного критерію.

Під час нормування та формування формули згортки необхідно враховувати, що часткові критерії не є односпрямованими: частина часткових критеріїв має бути максимізована, частина – мінімізована. Тому часткові критерії можна поділити на стимулятори (що мають бути максимізовані) та дестимулятори (які повинні мінімізуватися). Формула нормування стимуляторів може бути спрощена та мати вигляд

$$k_i^{'} = \frac{k_i}{k_i^{\max}}. \qquad (2.26)$$

Формула нормування дестимуляторів може бути спро-

щена аналогічно

$$k_i^{'} = \frac{k_i^{\min}}{k_i}. \qquad (2.27)$$

Для охоплення широкого кола задач розроблено велику кількість різноманітних згорток, однак найбільш просту формулу обчислення й мінімальну кількість параметрів мають адитивно-мультиплікативна та ентропійна формули згорток [48]. Частина складових цих формул може не використовуватися залежно від наявності стимуляторів або дистимуляторів. Крім того, якщо частинні критерії природньо можуть компенсувати один одного, то мультиплікативно-адитивна згортка може спроститися до адитивної. Аналогічно адитивно-мультиплікативна згортка може бути спрощена до мультиплікативної, якщо існує потреба лише одночасного покращання всіх частинних критеріїв. З урахуванням (2.26) та (2.27) адитивно-мультиплікативну згортку можна подати у вигляді

$$F = \sum_{i=1}^{K_1} \omega_i \frac{k_i}{k_i^{\max}} + \sum_{i=K_1+1}^{K_2} \omega_i \frac{k_i^{\min}}{k_i} + \\ + \prod_{i=1}^{K_1} \left(\frac{k_i}{k_i^{\max}} \right)^{\omega_i} \prod_{i=K_1+1}^{K_2} \left(\frac{k_i^{\min}}{k_i} \right)^{\omega_i}, \qquad (2.28)$$

де ω_i – вага (пріоритет) i-го критерію, для якого повинна

виконуватися умова

$$\sum_{i=1}^{K_1+K_2} \omega_i = 1.$$

Формула згортки, побудована за принципом інформаційної ентропії, відбиває змістовне наповнення поняття корисності як інформаційної категорії та має такий вигляд: [48]

$$F = \sum_{i=1}^{K_1} \omega_i \left(\frac{k_i}{k_i^{\max}} \right)^{\omega_i} + \sum_{i=K_1+1}^{K_2} \omega_i \left(\frac{k_i^{\min}}{k_i} \right)^{\omega_i}. \qquad (2.29)$$

Значення ваги кожного частинного критерію може бути обчислене на основі методів, що ґрунтуються на попарному порівнянні критеріїв чи аналітичній залежності показників важливості критеріїв, та формальних методів, таких як метод базового критерію чи метод Черчмена – Акоффа [48]. Проте в умовах апріорної невизначеності відповідно до принципу Бернулі – Лапласа можна взяти вагу критеріїв однаковою та рівною:

$$\omega_i = \frac{1}{K_1 + K_2}.$$

Згідно з працею [49] узагальнену ефективність системи

аналізу даних можна визначити її двома складовими: інформаційною спроможністю системи та зведеною вартістю створення, експлуатації, зберігання та ліквідації системи аналізу даних. При цьому узагальнений функціонально-статистичний критерій ефективності І. В. Кузьміна має такий вигляд:

$$E_{I,C} = K_I / K_{I0}, \qquad (2.30)$$

де K_I – узагальнена функціонально-статистична характеристика реальної системи:

$$K_I = I_{\max} / C \ , \qquad (2.31)$$

де $I_{\max}$ – максимальна інформаційна спроможність моделі; C – зведені витрати на створення, експлуатацію та ліквідацію системи.

Узагальнену функціонально-статистичну характеристику потенційної (ідеальної) системи визначають як

$$K_{I0} = I^0_{\max} / C_{\min}, \qquad (2.32)$$

де $I^0_{\max}$ – максимальна інформаційна спроможність потенційної системи;

$C_{\min}$ – зведені витрати для потенційної системи.

Максимізація інформаційної спроможності системи

аналізу даних передбачає зняття залишкової невизначеності на виході моделі аналізу даних. Вирішувальні правила системи аналізу даних можна одержати в процесі машинного навчання за вхідними даними. При цьому процес навчання системи полягає в пошуку оптимальних значень координат вектора просторово-часових параметрів функціонування, що забезпечують максимальне значення критерію ефективності системи, який з урахуванням (2.26) можна подати в такому вигляді

$$J = \frac{\overline{E}}{E_{\max}} \cdot \frac{C_{\min}}{C_{training} + C_{error}}, \qquad (2.33)$$

де $\overline{E}$ – усереднене значення інформаційного критерію ефективності машинного навчання моделі;

$E_{\max}$ – максимальне граничне значення критерію ефективності навчання моделі;

$C_{\min}$ – мінімальне граничне значення витрат, пов'язаних із експлуатацією системи аналізу даних;

$C_{training}$ – значення затрат на експлуатацію системи, зокрема витрати на формування вхідного математичного опису та вартість системних ресурсів, задіяних під час навчання (донавчання);

C_{error} – розраховані за матрицею штрафів втрати сервісу на базі системи аналізу даних, пов'язані з помилковим прийняттям рішень.

Найбільш простим методом оптимізації параметрів

функціонування системи аналізу даних є метод пошуку за сіткою (Grid Search), що виконує повний перебір по заданій вручну підмножині простору параметрів. Пошук за сіткою повинен супроводжуватися частковим вимірюванням продуктивності за комплексним критерієм або за точнісною характеристикою, обчисленою на валідаційних вибірках.

У разі низької внутрішньої розмірності задачі оптимізації, коли лише невелика кількість гіперпараметрів впливає на продуктивність системи аналізу даних, більш виправданим є використання методу випадкового пошуку (Random Search). У межах цього методу повний перебір усіх комбінацій замінюють на їх вибірку випадковим чином. Цей метод може застосовуватися як для дискретної множини значень так і для неперервних чи змішаних просторів.

Також існує практика використання методу Байєсовської оптимізації (Bayesian Optimization), що полягає в побудуванні стохастичної моделі функції відображення значень гіперпараметра на цільову функцію. Алгоритм має ітеративну структуру, на кожній ітерації якої перспективна конфігурацію параметрів моделі тестують та оновлюють. Одночасно в алгоритмі передбачено балансування між зондуванням простору рішень і використанням зібраної інформації про наближення до оптимуму. Цей метод перевершує результати пошуку за сіткою та випадкового пошуку як з точки зору кількості обчислень, так і з точки зору оптимальності знайденого рішення.

Ще одним ефективним і гнучким методом оптимізації

параметрів функціонування є використання популяційних метаевристичних пошукових алгоритмів. Ці алгоритми дуже легко розпаралелюються, оскільки кожен агент популяції може здійснювати оцінювання простору в окремому потоці. Тому ефективність використання цих алгоритмів часто визначається доступними обсягами пам'яті та обмеженнями на простір пошуку. За умов обмежених ресурсів ці алгоритми використовують для налаштування невеликої кількості параметрів. Проте вони алгоритми не потребують початкових наближень і дозволяють знайти близьке до оптимального рішення за порівняно невелику кількість ітерацій. До того ж, одна ітерація популяційного алгоритму потребує n_a обчислень цільової функції (критерію оптимізації), де n_a – кількість агентів популяції. Одним з найпростіших у реалізації популяційних алгоритмів пошукової оптимізації є алгоритм рою частинок (Particle Swarm Optimization).

Отже, параметри функціонування системи аналізу даних впливають на інформаційну спроможність системи та накладні витрати, пов'язані з її експлуатацією. Тому оптимізацію параметрів функціонування системи потрібно здійснювати за комплексними критеріями, що в загальному випадку є адитивно-мультиплікативними або ентропійними згортками частинних критеріїв. При цьому найпростішим алгоритмом оптимізації параметрів є пошук за сіткою, а найбільш просунутим є метаевристичний популяційний пошук.

РОЗДІЛ 3
ПРИКЛАДИ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ БАГАТОВИМІРНИХ ДАНИХ

3.1. Інтелектуальна система детектування об'єктів інтересу на аерозображенні

Малогабаритні безпілотні апарати часто використовують у задачах моніторингу території та об'єктів. Одночасно безпілотний апарат часто розглядають як мобільний сенсор для збору даних для їх подальшого оброблення в хмарному сервісі. Однак існує тенденція до розвитку крайових обчислень із метою розвантаження комунікаційних каналів інфокомунікаційного середовища й захисту інформації. Тому багато сучасних розробок пов'язані з розширенням функціональних можливостей та підвищенням рівня автономності безпілотних апаратів. Однією з найбільш затребуваних функціональних можливостей безпілотних апаратів є пошук і класифікація об'єктів інтересу. Детектування об'єктів інтересу та їх класифікаційний аналіз можуть бути корисними в таких практичних завданнях, як відео-аналітика систем рятувально-пошукових заходів, охорони периметру, прицілювання та наведення зброї.

Для прискорення режиму екзамену в системах детектування об'єктів інтересу на місцевості розроблено різноманітні апаратні прискорювачі на основі графічних обчислювальних пристроїв та нейропроцесорів. Однак процес навчання таких моделей є досить ресурсомістким, що усклад-

нює їх удосконалення в автономному режимі. Це обумовлює актуальність розроблення моделі та методу навчання системи розпізнавання об'єктів інтересу для адаптації до нових умов функціонування за умов ресурсних та інформаційних обмежень.

Для забезпечення ефективної роботи бортової системи аналізу даних у завданнях детектування об'єктів важливу роль відіграє екстрактор ознакового опису. Його ефективність визначається обчислювальною складністю, що оцінюють за критерієм $J_{Complexity}$, та ефективністю класифікаційних чи регресійних вирішувальних правил, що можуть бути оцінені за критеріями J_{Cls} та $J_{\mathrm{Re}g}$ відповідно. Водночас мультиплікативна згортка дозволяє зменшити ефект компенсації одного частинного критерію за рахунок іншого. Тому як комплексний критерій ефективності навчання бортової системи розглядають таку згортку частинних критеріїв:

$$J = J_{Cls} \cdot J_{\mathrm{Re}g} \cdot J_{Complexity} \qquad (3.1)$$

Для задачі класифікаційного аналізу образів нормований інформаційний критерій функціональної ефективності J_{Cls} обчислюють за формулою

$$J_{Cls} = \frac{1}{E_{\max}} \frac{1}{M} \sum_{m=1}^{M} E_m, \qquad (3.2)$$

де E_m – інформаційний критерій ефективності навчання

класифікатора розпізнавати реалізації класу X_m^o;

$E_{\max}$ – максимальне граничне значення інформаційного критерію ефективності навчання класифікатора.

Критерій ефективності регресійного аналізу меж обмежувального прямокутника детектованого на зображенні об'єкта обчислюють за формулою

$$J_{\mathrm{Re}g} = \frac{1}{\hat{n}} \sum_{i=1}^{\hat{n}} IoU_i \,, \qquad (3.3)$$

де IoU_i – міра перетину (коефіцієнт Жаккарда) правильної рамки об'єкту інтересу з відповідною *i*-ю прогнозованою рамкою об'єкта інтересу [50];

$\hat{n}$ – кількість пікселів карти ознак детектора, у яких міра перетину їх проекції на вхідне зображення з прямокутником об'єкта інтересу більша за порогове значення *Th*.

Для підвищення інформативності ознакового подання зображень важливим є використання додаткової інформації щодо контексту в якому знаходиться об'єкт розпізнавання. На рисунку 3.1 зображено запропоновану архітектуру моделі детектування малорозмірних об'єктів інтересу на основі комбінації техніки перенесення знань (Transfer learning) та інформації про контекст, одержаної за допомогою об'єднання карт ознак одержаних із різних шарів штучної нейронної мережі.

Пропонують запозичити нижні шари наперед навченої нейронної мережі Squeezenet, що складається з Fire

модулів і характеризується високою обчислювальною ефективністю [50]. Попереднє навчання цієї мережі здійснювалося на великому наборі даних ImageNet, що дозволило їй акумулювати інформацію про аналіз візуальних образів. Верхні шари пропонують реалізувати у вигляді розріджено кодувального шару, під час навчання якого можна утилізувати доступний обсяг нерозмічених даних для виділення пояснювальних факторів доменної області використання.

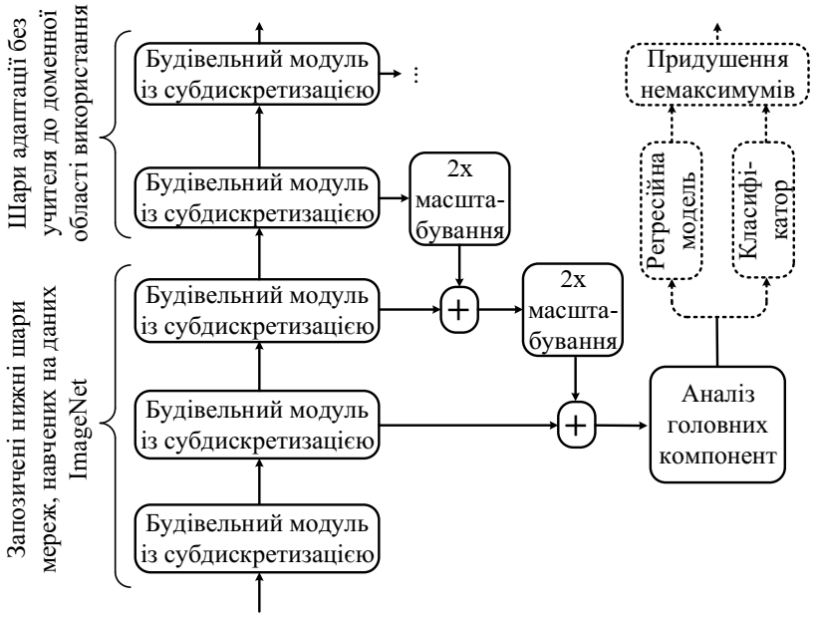


Рисунок 3.1 – Модель ієрархічного екстрактора ознак для детектування об’єктів на місцевості

Використання шару масштабування забезпечує однако-

вий розмір кожного каналу карти ознак. Конкатенація та масштабування розглядають як один шар «масштабування-конкатенації». Проте об'єднання карт ознак з різних шарів штучної нейронної мережі призводить до проблеми прокляття розмірності ознак. Для її усунення даної проблеми пропонують використовувати метод аналізу головних компонент. Його використання дозволяє знизити розмірність, не беручи до розгляду ознаки, які є нечутливими до цільової доменної області застосування. Вибір кількості головних компонент пропонують здійснювати відповідно до критерію Кайзера: вибір лише основних компонентів із власними значеннями, що перевищують одиницю.

Глибина згорткових фільтрів кожного рівня дорівнює кількості каналів вхідного зображення для першого шару та кількості каналів карти активації – для решти. Активацію пікселя кожного каналу карти ознак пропонують обчислювати на основі алгоритму швидкого невід'ємного ортогонального узгодженого переслідування (Non-negative Orthogonal Matching Pursuit) [21]. Цей алгоритм здійснює пошук невід'ємних коефіцієнтів $x_1,..., x_N$ лінійної комбінації навченого словника атомів D з метою мінімізації норми помилки апроксимації вхідного патча y. Вхідним параметром алгоритму є очікувана частка ненульових коефіцієнтів розрідженого коду, що за замовчуванням дорівнює 10 %.

Перший етап машинного навчання високорівневих шарів моделі ієрархічного екстрактора ознакового опису (рис. 2.12) пропонують виконувати з використанням алго-

ритму розріджено кодувального нейронного газу. Опис цього алгоритму наведено в підрозділі 2.2.

Задачу класифікаційного аналізу ієрархічного подання ознак пропонують виконати за допомогою інформаційно-екстремального алгоритму, основні етапи якого наведені в підрозділі 2.3. Водночас локалізацію та визначення меж об'єкта інтересу на зображенні пропонується здійснювати на основі регресійної моделі, що ґрунтується на нейронній мережі прямого поширення з одним прихованим шаром. Навчання цієї моделі пропонується здійснювати згідно з принципами машини екстремального навчання з інкрементальним додаванням нейронів прихованого шару. Деталі алгоритму описано в підрозділі 2.4.

Точне налаштування параметрів екстрактора ознакового опису пропонується здійснювати на основі алгоритму симуляції відпалу (Simulated Annealing) [46]. Алгоритм симуляції відпалу є універсальним і за деяких налаштувань придатний як для глобальної, так і для локальної оптимізації параметрів моделі.

Для реалізації навчання класифікатора, регресійної моделі та точного налаштування екстрактора ознак сформовано навчальні набори в процесі зіставлення реальних обмежувальних рамок об'єктів інтересу з рамками за замовчуванням. Обмежувальні рамки за замовчуванням визначають як карту ознак пік селів, спроектовану на вхідне зображення. Ми розглядаємо стан аерознімків у якому камера орієнтована вниз під прямим кутом і розташована на великій висоті (висота більша ніж 100 м).

Установлену за замовчуванням обмежувальну рамку

ставлять у відповідність кожній реальній рамці об'єкту, якщо перетин із нею за метрикою Жаккарда більший ніж 0,4. Регресійна модель навчається тільки на позитивних зразках (зіставлені обмежувальні рамки за замовчуванням).

Для навчання детектора об'єктів було використано 200 зображень із розміченого набору даних Inria Aerial Image Labeling Dataset [51]. Кожне зображення має розподільну здатність 5000×5000. До того ж, 500 нерозмічених зображень із розподільною здатністю 224x224 було згенеровано з використанням техніки випадкового вирізання з поворотом для навчання без учителя. Також 200 розмічених зображень із розподільною здатністю 224x224 було згенеровано для навчання з учителем. Розмічений набір навчальних даних було аугментовано до 1 000 зразків за допомогою додавання до зображень шуму, повороту та вирізання, зміни їхнього контрасту.

У наборі навчальних даних Inria Aerial Image Labeling Dataset подано велику кількість транспортних засобів у міській місцевості. Транспортні засоби було обрано як об'єкти інтересу, при цьому міська зона розглядається як доменну область застосування. У цьому разі алфавіт класів розпізнавання дорівнює $Z = 3$, де перший клас розпізнавання відповідає автомобілям, другий клас розпізнавання відповідає вантажівкам і третій – фоновим зображенням. Розмір об'єкта інтересу на випадковому зображенні знаходиться в діапазоні [7×7, …, 10×10] пікселів.

Відповідно до техніки перенесення знань перші 7 fire модулів попередньо навченої згорткової нейронної мережі Squeezenet були запозичені. У результаті кожне зображен-

ня було перекодовано в карту ознак із розміром 13×13×384. Наступні шари навчаються без учителя на нерозміченому наборі навчальних даних із цільової доменної області. При цьому ядра фільтрів дорівнюють 3x3, крок сканування – 1. Вихідна карта ознак сформована методом об'єднання карт ознак, сформованих модулями Fire6 та Fire7, а також карти ознак, сформованої останнім згортковим шаром.

Спочатку пропонують виконувати навчання детектора з попередньо навченим останнім шаром без учителя з використанням розріджено кодувального нейронного газу без точного налаштування. В алгоритмі навчання інформаційно-екстремального класифікатора пікселів карти ознак кількість вузлів у деревах рішень обмежена до 16. Водночас глибина кожного дерева встановлена на рівні 6.

З метою покращення результатів машинного навчання детектора, інформативність ознакового опису підвищена методом використання точного налаштування високорівневих згорткових шарів. У цьому разі поточні параметри алгоритму симуляції відпалу було встановлено так: $c = 0{,}98$, $T_0 = 10$, $epochs\_max = 6000$, $step\_size = 0{,}001$. На кожному етапі точного налаштування виконується перенавчання регресора та класифікатора. Для максимізації узагальнювальної здатності моделі та мінімізації обчислювальної складності, було виконано машинне навчання з послідовним, рівномірним нарощуванням кількості згорткових фільтрів (нейронів) із кроком 16.

Варто розглянути вплив параметрів алгоритму розріджено-кодувального нейронного газу, використаного на

етапі навчання без вчителя, на результати навчання з учителем. У таблиці 3.1 наведено залежність показників ефективності навчання детектора від параметра Nc, що дорівнює кількості атомів розрідженого кодера, які покривають навчальний набір даних. При цьому навчання здійснювалося на одноплатному комп'ютері Raspberry Pi v3 і час навчання було виміряно саме на цій платформі.

Таблиця 3.1 – Результати навчання детектора об'єктів із використанням різних значень кількості атомів Nc

N_c	J_{Cls}	J_{Loc}	$J=J_{Cls}* J_{Loc}$	Час навчання моделі, *с*	Час розпізнавання одного зображення, *с*	Відсоток розпізнаних об'єктів на тестовій вибірці, %
16	0,091 2	0,450	0,041 04	300	0,07	0,67
32	0,150 2	0,511	0,076 75	479	0,09	0,73
64	0,250 8	0,621	0,155 74	600	0,11	0,87
128	0,420 3	0,700	0,294 21	689	0,14	0,93
256	1,000 0	0,921	0,921 00	780	0,17	0,96
512	1,000 0	0,923	0,923 00	921	0,46	0,95

Аналіз таблиці 3.1 показує, що зі збільшенням гіперпараметра Nc, що визначає довжину розрідженого коду, зростає значення часткових і загального критеріїв оптимізації. Одночасно зростає як час, необхідний для навчання, так і час, необхідний для розпізнавання об'єктів на зображенні. При значенні $Nc < 256$ точність моделі за тестовим набором даних зростає зі збільшенням параметра Nc, проте подальше збільшення цього параметра призводить до погір-

шення результатів через перенавчання моделі.

Для оцінювання переваг у використанні попереднього навчання без учителя розглянемо результати машинного навчання з використанням симуляції відпалу до та після навчання за алгоритмом розріджено-кодувального нейронного газу. На рисунку 3.2 показано динаміку зміни критерію ефективності навчання детектора $J = J_{Cls} * J_{Loc}$ під час точного налаштування за алгоритмом симуляції відпалу з параметрами $c = 0{,}998$, $T_0 = 10$, $epochs\_max = 5000$, $step\_size = 0{,}001$.

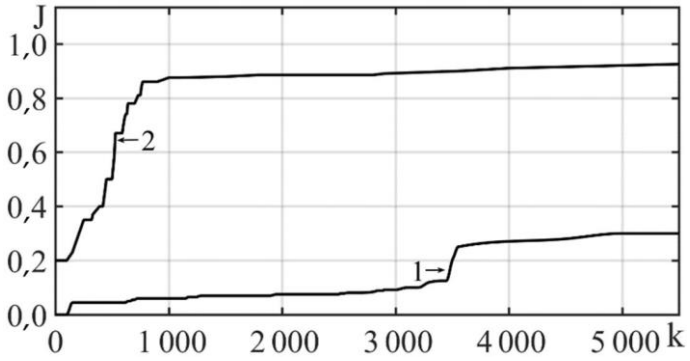


Рисунок 3.2 – Графіки залежності критерію оптимізації від кількості епох навчання: 1 – до застосування попереднього навчання без учителя; 2 – після застосування попереднього навчання без учителя

Аналіз рисунка 3.2 показує, що попереднє навчання без учителя на основі розріджено-кодувального нейронного газу дозволяє покращити кінцевий результат точного налаштування за допомогою алгоритму симуляції відпалу. У

разі використання попереднього навчання без учителя критерій, розрахований для навчальної вибірки рівний J_{train} = 0,921, забезпечує частку правильно детектованих об'єктів на тестовій вибірці, що дорівнює 96 %. Без використання попереднього навчання відповідні критерії значно відрізняються, J_{train} = 0,3011 із часткою правильно детектованих об'єктів на тестовій вибірці, що дорівнює 85 %.

Отже, запропонований алгоритм допомагає успішно вирішити завдання навчання детектора об'єктів на місцевості зі значною варіацією умов спостереження, що сформовані з набору даних Inria Aerial Image Labeling Dataset. Помічено, що використання попереднього навчання високорівневих шарів на основі розріджено-кодувального нейронного газу дозволяє значно покращити кінцевий результат. Зокрема, попереднє навчання пришвидшує більш ніж в 5 разів процес досягнення комплексним критерієм максимального значення під час точного налаштування моделі на розмічених навчальних даних. Час навчання прийнятний для автономного виконання на платформі Raspberry Pi v3 під час заряджання безпілотного апарату без використання додаткового обладнання.

3.2. Інтелектуальна система візуальної навігації

Автономна навігація має важливе значення для пошукових та рятувальних заходів, а також інших завдань віддаленого спостереження. Ручне керування безпілотним апаратом на основі відеопотоку є надзвичайно складним під час польоту поруч із будівлями, деревами або в примі-

щенні.

Використання глобальної (супутникової) системи позиціювання (GPS) може бути ненадійним рішенням в умовах поганого покриття або внаслідок проблем багатопроменевого поширення сигналу. Альтернативним підходом є використання компактного лазерного сканера (лідару). Проте лазерні сканери дорогі й характеризуються низькою частотою вимірювання. Перспективним напрямком розвитку навігаційних систем є використання візуальних сенсорів із точки зору невеликої ваги, невисокої ціни та інформативності. За допомогою відео камери можна одночасно здійснювати оцінювання переміщення та одержувати інформацію про зовнішнє середовище.

У межах геометричного підходу розроблено багато алгоритмів прямої (Direct), напівпрямої (Semi-direct) та візуальної навігації для монокулярної та стерео камер, що ґрунтується на локальних ознаках (Feature-based). Однак алгоритми в межах геометричного підходу характеризуються недостатньою стійкістю до ряду дуже поширених ефектів: зміна освітленості, наявність динамічних об’єктів у полі зору камери, чутливість до калібрувальних параметрів камери, низька текстурованість середовища, наявність шуму та розмиття рухом.

Візуальна одометрія, що основана на використанні недорогих камер та ідей і методів глибокого навчання є найбільш перспективним напрямком. У межах цього підходу існує принципова можливість знизити чутливість системи до калібрувальних параметрів та складних умов середовища за рахунок машинного навчання. Однак для навчання

глибоких моделей існує потреба у великих обсягах обчислювальних ресурсів та розмічених даних. Зниження обчислювальної складності можливе за рахунок звуження задач і місця застосування. Однак навіть спрощені глибокі моделі для успішного навчання потребують значних ресурсів, що уповільнюють процес адаптації до нових задач і середовища. Тому підвищення ефективності навчання системи візуальної навігації за умов обмеженого обсягу актуальних розмічених даних та обчислювальних ресурсів є актуальним завданням.

Процес машинного навчання і оптимізації навігаційної системи орієнтований на визначення оптимального вектора параметрів g, що забезпечує максимум комплексного критерію [52]:

$$J = J_{Cls} \cdot J_{\mathrm{Re}g} \cdot J_{Complexity}, \tag{3.4}$$

$$g^* = \arg\max_G \{J(g)\}, \tag{3.5}$$

де J_{Cls} – критерій ефективності класифікаційних вирішувальних правил для оцінювання перешкод і необхідного кута повороту;

$J_{\mathrm{Re}g}$ – критерій ефективності регресійного аналізу для оцінювання переміщення апарату за відеопослідовністю з бортової відеокамери;

$J_{Complexity}$ – критерій обчислювальної ефективності екстрактора ознак.

G – допустима область значень параметрів, що впливають на екстракцію ознак та прийняття рішень.

Критерій ефективності регресійного аналізу в задачі візуальної одометрії можна подати у вигляді відношення мінімально досягнутої (зокрема іншими дослідниками) усередненої за відеопослідовністю середньоквадратичної помилки реконструкції траєкторії руху $\varepsilon_{\min}$ до фактично виміряного її значення ε, тобто

$$J_{\mathrm{Re}g} = \frac{\varepsilon_{\min}}{\varepsilon}. \qquad (3.6)$$

Критерій обчислювальної ефективності можна подати у вигляді відношення мінімально можливого значення критерію трудомісткості моделі (або найбільш трудомісткої частини моделі) $C_{\min}$ до фактичного значення критерію трудомісткості C, тобто

$$J_{Complexity} = \frac{C_{\min}}{C}. \qquad (3.7)$$

У нейромережевих моделях трудомісткість C зазвичай вимірють кількістю операцій множення (Mul) та додавання (Add), що виконуються під час прямого проходження сигналу.

Критерій ефективності класифікаційного аналізу можна розрахувати за формулою (3.2).

Отже, критерій функціональної ефективності моделі

аналізу даних має як інформаційну та точнісну, так і вартісну природу. До того ж для одночасного врахування частинних критеріїв комплексний критерій формується на основі мультиплікативної згортки.

Схема інтелектуальної навігаційної системи для малогабаритного літального апарату показана на рисунку 3.3. Спроектована модель візуальної навігації повинна забезпечувати ухилення від перешкод та контроль власної позиції за допомогою візуальної одометрії.

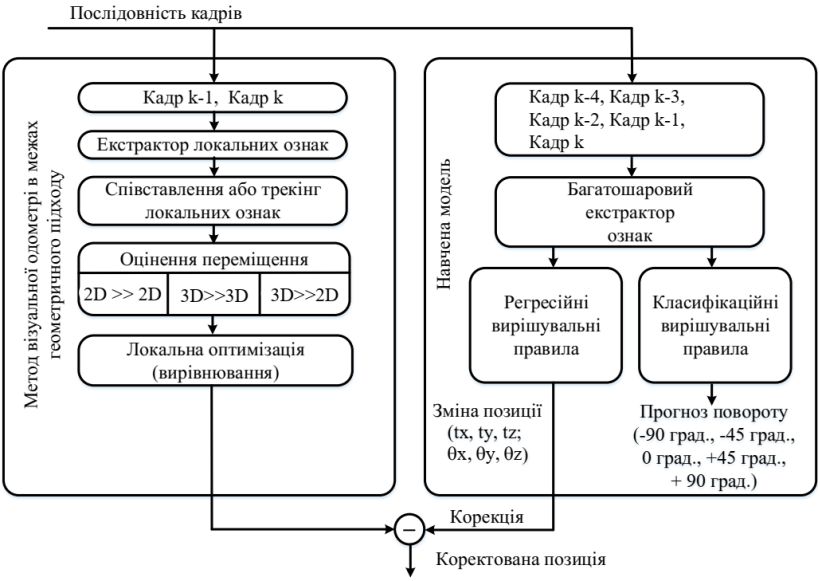


Рисунок 3.3 – Модель геометричної візуальної одометрії з моделлю паралельного коректора помилок

Запропонована модель візуальної навігації навчається як допоміжна система у вигляді паралельного коректора помилок для геометричної одометрії, що ґрунтується на

використанні екстрактора локальних та глобальних ознак. Обидві моделі повинні бути синхронізовані.

Для того, щоб виділити ознаки візуальних спостережень, пропонують використовувати 4-х шарову модель, що працює на розрідженому кодуванні з локальним рецептивним полем. На вхід моделі надходить багатоканальне зображення, сформоване із серії K_1 послідовних відео кадрів у градації сірого. На виході моделі формується високорівневий ознаковий опис візуальних спостережень.

Словник атомів розрідженого кодера пропонують попередньо навчати без учителя, послідовно шар за шаром. Для виявлення перешкод та формування відповідної реакції використовують класифікатор, що навчається з учителем на навчальних зразках, закодованих відповідними високорівневими ознаками. Регресійна модель використовується для відображення візуальних ознак у відповідну оцінку переміщення для визначення положення відеокамери в просторі.

На рисунку 3.4 показано архітектуру моделі, на першому шарі якої використовують кодери з рецептивним полем $5\text{x}5\text{x}K_1$, $3\text{x}3\text{x}K_1$ та $1\text{x}1\text{x}K_1$. Кількість фільтрів регулюють параметром K_2. Для збереження одного й того самого розміру карт ознак, утворених різномасштабними рецептивними полями, використовують техніку заповнення нулями.

Рисунок 3.4 – Архітектура моделі для екстракції візуальних ознак у навігаційній системі

У другому та третьому шарах параметр кроку сканування карти ознак із різномасштабними рецептивними полями становить 3 та 2 відповідно. Одночасно пропонують використовувати алгоритм невід'ємного ортогонального узгодженого переслідування для обчислення багатоканальних пікселів карти ознак [52].

Пропонують провести навчання екстрактора ознак без учителя, з використанням розглянутого в підрозділі 2.2 алгоритму розріджено-кодувального нейронного газу. Класифікаційний аналіз вихідної карти ознак пропонують здійснювати з використанням розглянутого у підрозділі 2.3 інформаційно-екстремального класифікатора. Так само

регресійну модель пропонують реалізовувати на основі інкрементальної машини екстремального навчання. Отже, обрано регресійну модель і класифікатор, що характеризуються високою оперативністю навчання та ґрунтуються на послідовному нарощуванні складності моделі аналізу даних до моменту досягнення необхідної достовірності рішень. Для апроксимації глобального оптимуму комплексного критерію під час навчання вирішувальних правил та точного налаштування екстрактора ознак пропонують використовувати метаевристичний алгоритм імітації відпалу [46].

Різні значення параметрів K_1 та K_2 впливають як на інформативність ознак, так і на обчислювальну трудомісткість. Трудомісткість пропонують вимірювати за кількістю операцій множення (Mul) та додавання (Add), що виконують під час згорткових операцій із зображенням або картою ознак. Для архітектури мережі, показаної на рисунку 3.4, трудомісткість може бути розрахована за формулою

$$C = K_2(2706472K_1 + 4438784K_2). \qquad (3.8)$$

Для класифікатора та регресійної моделі оптимальна конфігурація згорткового екстрактора може бути різною, оскільки вона відповідає за різні завдання. Тому комплексний критерій (3.4) пропонує компроміс із точки зору точності правил прийняття рішень та обчислювальної складності екстрактора візуальних ознак.

Множина класів розпізнавання $\{X_z^o\}$ описує характерні перешкоди та відповідні команди реакції та має потужність $Z = 5$. Перший клас розпізнавання X_1^o характеризує рух уперед без повороту. Класи X_2^o і X_3^o відповідають лівому повороту 45 і 90 градусів відповідно. Класи X_4^o і X_5^o відповідають правому повороту 45 і 90 градусів відповідно. Обсяг навчальних та тестових даних кожного класу $n_z = 300$.

Для виявлення тенденції зміни середніх значень часткових та комплексного критеріїв у разі зростання параметрів K_1 та K_2, що впливають на розмір екстрактора ознак (рис. 3.4), було виконано симуляцію для трьох фіксованих значень кожного з цих параметрів (табл. 3.2). До того ж, оптимальні значення цих параметрів визначаються для відкритого набору даних KITTI-07 [53].

В алгоритмі симуляції відпалу для точного налаштуванняекстрактора ознак використовуються такі параметри : c = 0.98, T_0 = 10, epochs_max = 5000, step_size = 0,001. На кожному кроці точного налаштування передбачається побудову класифікаційних та регрусійних вирішувальних правил із нуля.

Аналіз таблиці 3.2 показує, що збільшення значень параметрів K_1 і K_2 загалом призводить до збільшення обчислювальної складності екстрактора ознак (3.8) та достовірності вирішувальних правил. Водночас збільшення параметра K_1 мало впливає на ефективність

класифікатора внаслідок зниження ефективності пошукового алгоритму зі значним збільшенням розміру простору пошуку, тоді як помилка регресії однаково чутлива до значення параметрів K_1 і K_2.

Таблиця 3.2 – Залежність часткових та комплексного критерію від параметрів екстрактора ознакового опису K_1 і K_2

K_1	K_2	$\bar{E} / E_{\max}$	$\varepsilon_{\min} / \varepsilon$	$C_{\min} / C$	J
3	4	0,083	0,112	1,000	0,009296
5	4	0,101	0,188	0,827	0,015703
7	4	0,098	0,200	0,705	0,013818
3	8	0,28	0,688	0,297	0,057214
5	8	0,29	0,756	0,264	0,057879
7	8	0,29	0,775	0,238	0,053491
3	16	0,39	0,968	0,082	0,030957
5	16	0,55	1,000	0,077	0,04235
7	16	0,51	1,000	0,072	0,03672

З огляду на те, що під час зростання K_1 та K_2 достовірність правил прийняття рішень зростає повільніше, ніж обчислювальна складність, то використання комплексного критерію J забезпечує компромісне рішення. Тобто наступні значення параметрів $K_1^* = 5$ та $K_2^* = 8$ є оптимальними. В оптимальній конфігурації екстрактора ознак середнє значення інформаційного критерію функціональної ефективності

дорівнює $\bar{E} = 0,29$. Це відповідає точності 95,2 % для навчального набору та 94 % для тестового набору.

На рисунку 3.5 показано графік зміни середнього значення інформаційного критерію ефективності (3.2) залежно від кількості ітерацій алгоритму точного налаштування екстрактора ознак.

Аналіз рисунку 3.5 показує, що після 1 000-ї ітерації зростання усередненого за алфавітом класів інформаційного критерію (2.5) почало сповільнюватися, а після 2 500-ї ітерації практично не змінилося. Така зміна критерію свідчить про те, що подальше збільшення інформаційного критерію можливе лише за рахунок збільшення значень K_1 та K_2, або за допомогою вдосконалення структури екстрактора (рис. 3.4).

Для очної ілюстрації результатів машинного навчання навігаційної системи можна виконати порівняння реальної та реконструйованої за допомогою запропонованої моделі. За реальну траєкторію беруть ту, що вимірюється за допомогою системи супутникового позиціювання (GPS) та лазерного сканування (LiDaR). Запропонована модель паралельного коректора навчена на наборі даних KITTI-09 [53]. На рисунку 3.6 а показано результат використання геометричної одометрії ORB-SLAM, що ґрунтується на зіставленні ключових точок, на наборі даних KITTI-07 [54].

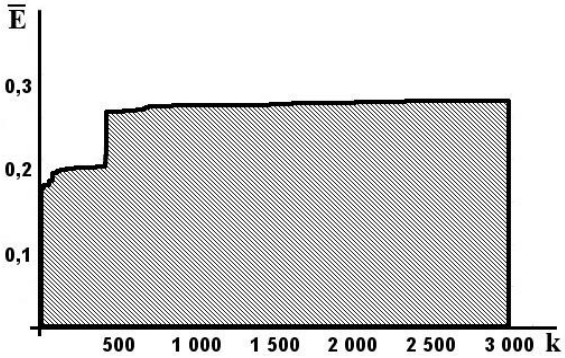
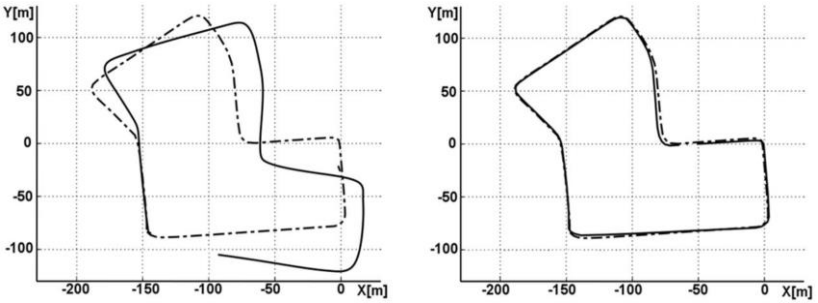


Рисунок 3.5 – Графік зміни усередненого значення інформаційного критерію ефективності (3.2) залежно від кількості ітерацій алгоритму імітації відпалу



а б

Рисунок 3.6 – Реальна траєкторія (пунктирна лінія) та реконструйована (суцільна лінія) для KITTI-07 [53]:
а – монокулярна одометрія на основі локальних ознак;
б – паралельна корекція помилок для монокулярної одометрії на основі локальних ознак із використанням запропонованої моделі

Максимальна кількість ключових точок обмежена до

100, щоб забезпечити необхідну частоту кадрів (10 кадрів/секунда) у разі бортової системи без застосування обчислювальних можливостей графічного процесора. Тестування відбувалося на комп'ютері Raspberry Pi 3+. Процес навчання тривав близько години.

Аналіз рисунка 3.6 а показує, що реконструйована траєкторія з використанням лише алгоритму геометричної одометрії має значну похибку. При цьому на рисунку 3.6 б зображено результат використання геометричної одометрії з паралельною корекцією на основі запропонованої моделі. Результат показує, що точність реконструкції стала значно вищою внаслідок ефективнішого розпізнавання поворотів. Зокрема, помилку оцінювання переміщення вдалося скоротити приблизно на 65,6 % при тій самій частоті кадрів.

Отже, запропоновані модель та алгоритм навчання автономної навігаційної системи безпілотного апарату були перевірені на відкритому наборі даних KITTI. Результати тестування підтверджують придатність розроблених алгоритмів для практичного використання. Розроблені модель та метод навчання дозволяють за прийнятний час переконфігурувати безпілотний апарат до нових умов функціонування без використання додаткових ресурсів.

3.3. Інтелектуальна система детектування шкідливого мережевого трафіку

Наявні системи виявлення шкідливого мережевого трафіку досі не забезпечують високої достовірності ві-

шень, що обумовлено постійним зростанням кількості та різноманітності нових джерел шкідливого трафіку та малою кількістю актуальних розмічених даних. Водночас використання вручну сконструйованих ознак для описання спостережень призводить до зниження з плином часу інформативності ознакового опису та ефективності навчання вирішувальних правил для детектування шкідливого трафіку. Тому найбільш перспективним підходом до синтезу екстрактора ознакового опису є використання моделі розділення пояснювальних факторів за нерозміченими даними. При цьому ієрархічні моделі ознакового подання для розділення пояснювальних факторів характеризуються більшою ємністю та інформативністю, ніж однорівневі моделі.

Серед моделей ієрархічного ознакового подання найбільш обчислювально ефективними є моделі з локальними рецептивними полями кодерів. Саме такі моделі показали найбільший успіх у задачах аналізу візуальних образів. При цьому в разі достатньої кількості шарів ці моделі є ефективними не лише для даних із локальною просторовою зв'язаністю, але й для довільних топологій.

Перш ніж проектувати ієрархічний екстрактор ознак та вирішувальні правила необхідно розглянути спосіб кодування зразків трафіку в багатоканальне вхідне зображення. Внутрішні характеристики одиниці трафіку (потоку пакетів чи сесії) найкраще відображаються в передній частині її байтів, де містяться дані про з'єднання й деякі дані контенту. Процес перетворення pcap-файла на навчальний набір даних містить три основні етапи: розділення трафіку на

дискретні одиниці з урахуванням деякої гранулярності, очищення трафіку методом видалення пустих і дублюючих одиниць, формування навчальних зображень. У разі розділення трафіку на дискретні одиниці можна розглядати такі гранулярності: TCP-з'єднання, потік, сесія, сервіс та хост. У цій роботі пропонують розділяти вхідний трафік на потоки, де ряд пакетів мають однаковий кортеж із п'яти елементів: IP-адреса джерела та отримувача, номер портів джерела та одержувача, номер протоколу. Одночасно довжина потоку обмежена 784 байтами, тому довші потоки обрізають, а коротші доповнюють нульовими байтами. Послідовні 10 потоків об'єднують у канали вхідного зображення. У результаті маємо 10-канальне зображення 28x28 пікселів, що надходитиме на вхід згорткової мережі. Яскравість кожного пікселя нормалізується до діапазону [0, 1].

Для побудови моделі екстрактора ознак пропонують використати два шари розрідженого кодування з оператором агрегації max_pooling між ними. Водночас розмір рецептивного поля сканувального кодера дорівнює 5x5xd, де d – кількість каналів вхідного зображення. Для оператора агрегації max_pooling використане ядро 2x2. Активацію пікселя кожного каналу карти ознак пропонують обчислювати на основі алгоритму ортогонального узгодженого переслідування (Orthogonal Matching Pursuit).

Кількість атомів розрідженого кодера наперед не задають, а визначають у процесі машинного навчання за алгоритмом зростаючого розріджено-кодувального

нейронного газу, описаного в підрозділі 2.2. Водночас вхідні дані для алгоритму формують методом декомпозиції вхідного зображення на 3D-патчі згідно зі схемою, зображеною на рисунку 2.3, та їх перетворення на 1D-вектори. Навчені атоми розріджених кодерів потім можуть бути перетворені на 3D-габарит.

Класифікаційні вирішувальні правила пропонується формувати в процесі інформаційно-екстремального машинного навчання. До того ж, двійкове кодування спостережень можна здійснювати на основі багатоінтервальної системи порогів (рис. 2.11).

Навчальна вибірка, сформована зі CTU-Mixed, для навчання екстрактора ознак становить 10 000 зразків. Для навчання інформаційно-екстремального класифікатора сформовано по 1 000 розмічених зразків на клас у навчальній та тестовій вибірках. В алгоритмі зростаючого розріджено кодувального нейронного газу обрано такі параметри $\varepsilon_b = 0,5$, $\varepsilon_b = 0,05$, $a_{\max} = 100$, $\eta_0 = 1$ та $\eta_{final} = 0,01$. Параметр порогу фіксації нейронів v та параметр кількості порогів на ознаку L системи контрольних допусків класифікатора налаштовують за допомогою перебору значень. У таблиці 3.3 показано залежність кількості нейронів у першому M_1 і другому M_2 розріджено-кодувальних шарах моделі, усередненого за класами інформаційного критерію ефективності навчання $\bar{E}$ та точності за валідаційною вибіркою від параметрів v та L [54].

Аналіз таблиці 3.3 показує, що збільшення порогу v приводить до збільшення кількості атомів розрідженого

кодера в процесі навчання екстрактора ознак без учителя. Одночасно збільшення порогу з 0,8 до 0,9 практично не впливає на точність вирішувальних правил.

Таблиця 3.3 – Залежність результатів машинного навчання від параметрів моделі детектора шкідливого трафіка

ν	L	M_1	M_2	$\bar{E}$	Точність за тестовою вибіркою
0,6	1	27	41	0,36	90,0
0,7	1	45	52	0,39	91,0
0,8	1	49	300	0,42	92,0
0,9	1	320	1 500	0,42	92,0
0,6	2	27	41	0,46	93,0
0,7	2	45	52	0,54	95,0
0,8	2	49	300	0,65	97,0
0,9	2	320	1 500	0,65	97,0
0,6	3	27	41	0,46	93,0
0,7	3	45	52	0,55	95,2
0,8	3	49	300	0,81	98,9
0,9	3	320	1 500	0,83	99,0
0,6	4	27	41	0,46	93,0
0,7	4	45	52	0,55	95,3
0,8	4	49	300	0,74	98,1
0,9	4	320	1 500	0,83	99,0

Значення $\nu^* = 0,8$ є оптимальним і дозволяє сформувати більш компактне ознакове подання (компресію), у той час як $\nu = 0,9$ дозволяє сформувати розріджене подання на основі надповного базису атомів розрідженого кодера.

Оптимальне значення гіперпараметра L дорівнює $L^* = 3$. Подальше збільшення параметра L не приводить до зростання точності вирішувальних правил. За оптимальних параметрів екстрактора та класифікатора точність детектування шкідливого трафіку становить 98,9 %.

На рисунку 3.7 показано графік зміни максимумів усередненого за алфавітом класів нормованого інформаційного критерію (2.17) від кількості ітерацій пошуку за алгоритмом симуляції відпалу при $L^* = 3$ та $v^* = 0{,}8$. При цьому задано такі параметри алгоритму симуляції відпалу: c = 0,998, $T_0 = 10$, epochs_max = 10000, step_size = 0,001.

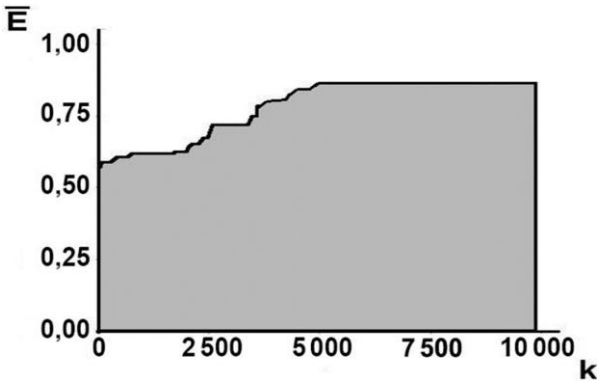
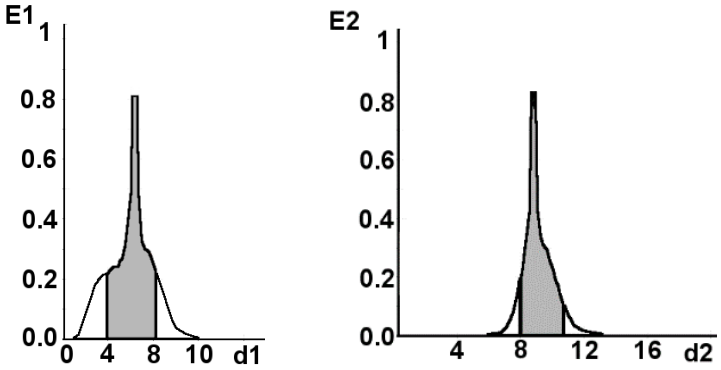


Рисунок 3.7 – Графік залежності максимального значення усередненого інформаційного критерію (2.17) від кількості ітерацій пошукового алгоритму

Аналіз рисунка 3.7 показує, що алгоритму знадобилося лише 5 000 ітерацій для досягнення глобального максимуму, що свідчить про інформативність ознакового опису спостережень.

На рисунку 3.8 показано залежність нормованого інформаційного критерію (2.17) від кодового радіуса контейнера кожного з класів [55].



а б

Рисунок 3.8 – Графіки залежності нормованого інформаційного критерію (2.17) від радіуса контейнерів класів: а – клас нормального трафіку; б – клас шкідливого трафіку

Аналіз рисунка 3.8 показує, що максимальне значення інформаційного критерію ефективності навчання розпізнавати спостереження першого й другого класів дорівнюють $E_1^* = 0{,}80$ і $E_2^* = 0{,}82$ відповідно. Оптимальні значення радіусів відповідних контейнерів класів дорівнюють $d_1^* = 6$, $d_2^* = 10$ (в кодових одиницях). Водночас міжцентрова кодова відстань дорівнює 18, що свідчить про компактність розподілу векторів і чіткість розбиття в просторі Хемінга.

Отже, за результатами моделювання на даних із набо-

рів CTU-Mixed та CTU-13 доведено придатність запропонованих моделі та методу навчання системи розпізнавання шкідливого мережевого трафіку для практичного використання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ding J. Model Selection Techniques: an Overview / J. Ding, V. Tarokh, Y. Yang // IEEE Signal Processing Magazine. – 2018. – Vol. 35, No. 6. – P. 16–34.

2. Vreeken J. Modern MDL meets Data Mining Insights, Theory, and Practice / J. Vreeken, K. Yamanishi // Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. – New York : ACM, 2019. – P. 3229–3230.

3. Jain P. Non-convex Optimization for Machine Learning / P. Jain, P. Kar // Journal Foundations and Trends in Machine Learning archive. – 2017. – Vol. 10, I. 3–4. – P. 142–336.

4. Big data preprocessing: methods and prospects / G. Salvador, R.-G. Sergio, L. Julián, B. José, H. Francisco // Big Data Analytics. – 2016. – No. 1. – P. 1–22.

5. Saima B. A Survey of Data Clustering Methods / B. Saima, K. Naeem // International Journal of Advanced Science and Technology. – 2018. – Vol. 113. – P. 133–142.

6. Fast dimensionality reduction and classification of hyperspectral images with extreme learning machines / J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza // Journal of Real-Time Image Processing. – 2018. – Vol. 15, Issue 3. – P. 439–462.

7. Mhaskar H. N. Deep vs. Shallow Networks: an Approximation Theory Perspective [Electronic resource] / H. N. Mhaskar, T. Poggio // CBMM Memo. – 2017. – Vol. abs1608.03287, No. 054. – P. 1–39. – Access mode: https://arxiv.org/pdf/1608.03287.pdf.

8. Christiane L. Metalearning: a survey of trends and

technologies / L. Christiane, B. Marcin, G. Bogdan // Artificial Intelligence Review. – 2015. – Vol. 144. – P. 117–130.

9. Jolliffe I. T. Principal component analysis: a review and recent developments / I. T. Jolliffe, J. Cadima // Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. – 2016. – Vol. 374. – P. 1–16.

10. Bengio Y. Representation Learning: a review and New Perspectives / Y. Bengio, A. Courville, P. Vincent // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2014. – Vol. 35 (8). – P. 1798–1828.

11. Kitsuchart P. A comparison between shallow and deep architecture classifiers on small dataset / P. Kitsuchart, S. Wisuwat // Proceedings of the 8th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, 2016. – P. 1–6.

12. Forbus K. D. Representation and Computation in Cognitive Models / K. D. Forbus, C. Liang, I. Rabkina // Topic Continuation: Visions of Cognitive Science: Edited by Wayne D. Gray – Game XP: Action Games as Experimental Paradigms for Cognitive Science. – 2017. – Vol. 9, Issue 2. – P. 255–536.

13. Privileged Semi-Supervised Learning / X. Chen, C. Gong, C. Ma, X. Huang el al. // 25th IEEE International Conference on Image Processing (ICIP). – Athens, Greece, 2018. – P. 2999–3003.

14. Patel H. Dictionary Properties for Sparse Representation: Implementation and Analysis / H. Patel, H. Mewada // Journal of Artificial Intelligence. – 2018. – Vol. 11, Issue 1. – P. 1–8.

15. Zhou Z.-H. Deep forest / Z.-H. Zhou, J. Feng // National Science Review. – 2019. – Vol. 6, Issue 1. – P. 74–86.

16. Lopes E. Estimating The Algorithmic Variance Of

Randomizedensembles Via The Bootstrap / E. Lopes // The Annals of Statistics. – 2019. – Vol. 47, No. 2. – P. 1088–1112.

17. Utkin L. V. A deep forest classifier with weights of class probability distribution subsets / L. V. Utkin, M. S. Kovaleva, A. A. Meldo // Knowledge-Based Systems. – 2019. – Vol. 173. – P. 15–27.

18. Mostafa H. Deep Supervised Learning Using Local Errors / H. Mostafa, V. Ramesh, G. Cauwenberghs // Frontiers in Neuroscience. – 2018. – Vol. 16, Issue 608. – P. 1–16.

19. Hadsell R. Dimensionality Reduction by Learning an Invariant Mapping / R. Hadsell, S. Chopra, Y. LeCun // Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). – New York, NY, USA, 2006. – P. 1735–1742.

20. Längkvist M. A Review of Unsupervised Feature Learning and Deep Learning for Time-Series Modeling / M. Längkvist, L. Karlsson, A. Loutfi // Pattern Recognition Letters. – 2014. – Vol. 42. – P. 1–62.

21 Tariyal S. Greedy Deep Dictionary Learning / S. Tariyal, A. Majumdar, R. Singh, M. Vatsa // IEEE Access. – 2016. – Vol. 99.

22. Chang H. Stacked Predictive Sparse Decomposition for Classification of Histology Sections / H. Chang, Y. Zhou, A. Borowsky, K. Barner, el al // International Journal of Computer Vision. – 2015. – Vol. 113. – P. 3–18.

23. Renton G. Fully convolutional network with dilated convolutions for handwritten text line segmentation / G. Renton, Y. Soullard, C. Chatelain, S. Adam et al. // International Journal on Document Analysis and Recognition. – 2018. – Vol. 21, Issue 3. – P. 177–186.

24. Shameem F. Categorized image classification using CNN features with ECOC framework / F. Shameem,

M. Seshashayee // International Journal of Recent Technology and Engineering (IJRTE). – 2019. – Vol. 8, Issue 2. – P. 145–150.

25. Christiani T. Set similarity search beyond MinHash / T. Christiani, R. Pagh // Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing. – June 19–23, Montreal, Canada, 2017. – P. 1094–1107.

26. A Comparative study of distance metric learning to find sub-categories of minority class from imbalance data / Md. Mahin, Md. J. Islam, A. Khatun, B. C. Debnath // Preprint. – DOI: 10.13140/RG.2.2.33869.49121.

27. Zanaty E. A. Generalized Hermite kernel function for support vector machine classifications / E. A. Zanaty, A. Afifi // International Journal of Computers and Applications. – 2018.

28. Ying L. Orthogonal incremental extreme learning machine for regression and multiclass classification / L. Ying // Neural Computing and Applications. – 2016. – Vol. 27, Issue 1. – P. 111–120.

29. Saleh A. Analysis of Accurate Learning in Radial Basis Function Neural Network Using Cosine Similarity on Leaf Recognition / A. Saleh, T. Tulus, S. Efendi // Proceedings of the 1st Workshop on Multidisciplinary and Its Applications Part 1, WMA-01. – 19–20 Jan., Aceh, Indonesia, 2018. – P. 1–10.

30. Song Y. Decision tree methods: applications for classification and prediction / Y. Song, Y. Lu. // Shanghai Archives of Psychiatry. – 2015. – Vol. 27, No. 2. – P. 130–135.

31. Schapire R. E. The strength of weak learnability / R. E. Schapire // Machine Learning. – 1990. – Vol. 5, Issue 2. – P. 197–227.

32. Sagi O. Ensemble learning: a survey / O. Sagi,

L. Rokach // Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. – 2018. – Vol. 8, Issue. 4. – P. 1–18.

33. Zhang C.-X. RotBoost: A technique for combining Rotation Forest and AdaBoost / C.-X. Zhang, J.-S. Zhang // Pattern Recognition Letters. – 2008. – Vol. 29, Issue 10. – P. 1524–1536.

34. Xu S. Bayesian Naïve Bayes classifiers to text classification / S. Xu // Journal of Information Science. – 2018. – Vol. 44. – P. 48–59.

35. Li H.-T. Layer-Level Knowledge Distillation for Deep Neural Network Learnin / S.-C. Lin, C.-Y. Chen, C.-K. Chiang // Applied Sciences. – 2019. – Vol. 9. – P. 1966.

36. Zhou Y. Approximation Trees: Statistical Stability in Model Distillation / Y. Zhou, Z. Zhou, G. Hooker // ArXiv. – 2018. – Vol. abs/1808.07573.

37. Labusch K. Sparse coding neural gas: learning of overcomplete data representations / K. Labusch, E. Barth, T. Martinetz // Neurocomputing. – 2009. – Vol. 72, Issue 7–9. – P. 1547–1555.

38. Palomo J. The Growing Hierarchical Neural Gas Self-Organizing Neural Network / J. Palomo, E. López-Rubio // IEEE Transactions on Neural Networks and Learning Systems. – 2017. – Vol. 28, No. 9. – P. 2000–2009.

39. Improving the effectiveness of training the on-board object detection system for a compact unmanned aerial vehicle / V. Moskalenko, S. Dovbysh, I. Naumenko, A. Moskalenko el al // Eastern-European Journal of Enterprise Technologies. – 2018. – No. 4/9(94). – P. 19–26.

40. Tim S. C. W. / Multi-layer Dictionary Learning for Image Classification / S. C. W. Tim, M. Rombaut, D. Pellerin // Proceedings of the 17th International Conference

on Advanced Concepts for Intelligent Vision Systems. – Lecce, Italy, Oct. 24–27. – 2016. – P. 522–533.

41. Vens C. Random Forest Based Feature Induction. / C. Vens, F. Costa // Proceedings of the IEEE 11th International Conference on Data Mining. – Vancouver Canada, 11–14 Dec, 2011. Piscataway, NJ, 2011. – P. 744–753.

42. V.V. Moskalenko. Learning decision making support system for control of nonstationary technological process / V. V. Moskalenko, A. S. Rizhova, A. S. Dovbysh // Journal of Automation and Information Sciences. – 2016. – Vol. 48, Issue 6. – P. 39–48.

43. Development of the method of features learning and training decision rules for the prediction of violation of service level agreement in a cloud-based environment / V. Moskalenko, A. Moskalenko, S. Pimonenko, A. Korobov // Eastern-European Journal of Enterprise Technologies. – 2017. – Vol 5, No. 2 (89). – P. 26–33.

44. Moskalenko V. V. Information-Extreme Method for Classification of Observations with Categorical Attributes / V. V. Moskalenko, A. S. Rizhova, A. S. Dovbysh // Cybernetics and Systems Analysis. – 2016. – Vol. 52. – P. 224–231.

45. Ranganathan V. A New Backpropagation Algorithm without Gradient Descent [Electronic resource] / V. Ranganathan, S. Natarajan // Access mode: https://arxiv.org/pdf/1802.00027v1.pdf. – 2018.

46. Moskalenko V.V. Information-extreme algorithm of the system for recognition of objects on the terrain with optimization parameter feature extractor / A. G. Korobov, V. V. Moskalenko // Radio Electronics, Computer Science, Control. – 2017. – No. 2. – P. 61–69.

47. Prellberg J. Limited Evaluation Evolutionary

Optimization of Large Neural Networks [Electronic resource] / J. Prellberg, O. Kramer. – Access mode: https://arxiv.org/abs/1806.09819. – 2018.

48. Moskalenko V. Optimizing the parameters of functioning of the system of management of data center it infrastructure / V. Moskalenko, S. Pimonenko // Eastern-European Journal of Enterprise Technologies. – 2016. – Vol. 5, No. 2 (83). – P. 21–29.

49. Москаленко В. В. Вступ до інформаційного аналізу і синтезу інфокомунікаційних систем / В. В. Москаленко, А. С. Довбиш. – Суми : СумДУ, 2016. – 226 с.

50. Moskalenko V. V. Information-extreme algorithm of the system for recognition of objects on the terrain with optimization parameter feature extraction / V. V. Moskalenko, A. G. Korobov // Radio Electronics, Computer Science, Control. – 2017. – № 2. – С. 38–45.

51. A Model And Training Method Of Small-Sized Object Detection System For A Compact Aerial Drone // V. V. Moskalenko, A. S. Moskalenko, A. G. Korobov, M. O. Zaretsky / Radio Electronics, Computer Science, Control. – 2019. – No. 1. – P. 110–121.

52. Москаленко В. В. Моделі і методи інтелектуальної інформаційної технології автономної навігації для малогабаритних безпілотних апаратів / В. В. Москаленко, А. С. Москаленко, А. Г. Коробов // Радіоелектроніка, інформатика, управління. – 2018. – № 3. – С. 68–77.

53. Model and training methods of autonomous navigation system for compact drones / V. Moskalenko, A. Moskalenko, A. Korobov, O. Boiko et al. // Proceedings of the IEEE Second International Conference on Data Stream Mining & Processing (DSMP) (Lviv, Ukraine, 21–25 August 2018). – Lviv Polytechnic Publishing House: Lviv, Ukraine, 2018. – P. 503–

508.

54. The Model and Training Algorithm of Compact Drone Autonomous Visual Navigation System / V. Moskalenko, A. Moskalenko, A. Korobov, V. Semashko // Data. – 2019. – Vol. 4 (1). – P. 1–14.

55. Москаленко В. В. Модель і алгоритм навчання детектора шкідливого трафіку на основі модифікації зростаючого нейронного газу / В. В. Москаленко, А. С. Москаленко, М. О. Зарецький // Радіоелектронні і комп'ютерні системи. – 2018. – № 3. – С. 260–271.

Наукове видання

Москаленко В’ячеслав Васильович

МОДЕЛІ І МЕТОДИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ БАГАТОВИМІРНИХ ДАНИХ ЗА УМОВ АПРІОРНОЇ НЕВИЗНАЧЕНОСТІ

Монографія

Художнє оформлення обкладинки А. С. Москаленко
Редактор О. Ф. Дубровіна
Комп’ютерне верстання В. В. Москаленка

Формат 60×84/16. Ум. друк. арк. 10,92. Обл.-вид арк. 8,58. Тираж 300 пр. Зам. №

Видавець і виготовлювач
Сумський державний університет,
вул. Римського-Корсакова, 2, м. Суми, 40007
Свідоцтво суб’єкта видавничої діяльності ДК № 3062 від 17.12.2007.