МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

І.А. Терейковський,
Д.А. Бушуєв,
Л.О. Терейковська

ШТУЧНІ НЕЙРОННІ МЕРЕЖІ

БАЗОВІ ПОЛОЖЕННЯ

Навчальний посібник

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра
за освітньою програмою «Системне програмування та спеціалізовані комп'ютерні системи»
спеціальності 123 Комп'ютерна інженерія

Електронне мережне навчальне видання

Київ
КПІ ім. Ігоря Сікорського
2022

Рецензент *Толюпа С. В.* доктор технічних наук, професор, професор кафедри кібербезпеки та захисту інформації факультету інформаційних технологій Київський національний університет імені Тараса Шевченка

Відповідальний редактор Тарасенко В.П., доктор технічних наук, професор

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
(протокол № Х від DD.MM.YYYY р.)
за поданням Вченої ради факультету прикладної математики
(протокол № 1 від 01.09.2022 р.)

Навчальний посібник містить матеріали для самостійної роботи здобувачів ступеня бакалавр за освітньою програмою «Системне програмування та спеціалізовані комп'ютерні системи» спеціальності 123 «Комп'ютерна інженерія» при вивченні розділу «Базові положення в області штучних нейронних мереж» з дисципліни «Штучні нейронні мережі». Також стане у нагоді розробникам програмного забезпечення, аспірантам та студентам технічних снеціальностей.

Реєстр. № НП ХХ/ХХ-ХХХ. Обсяг Х,Х авт. арк.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Перемоги, 37, м. Київ, 03056
https://kpi.ua
Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів
і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

ЗМІСТ

Список скорочень

АРТ - нейронна мережа адаптивної резонансної теорії

БШП - багатошаровий персептрон

ДАП - двонаправлена асоціативна пам'ять

ДШП - двохшаровий персептрон

НМ - нейрона мережа

ОС - операційна система

ПК - пружна карта (пружинна карта)

РБФ - нейронна мережа з радіальними базисними функціями

СЛД - синхронізоване лінійне дерево

СМ - семантична мережа

СНМ - семантична нейронна мережа

СШН - схований шар нейронів

ШНМ - швидка нейронна мережа

SOM (Self-Organizing Maps) - карта Кохонена

Вступ

В теперішній час в різних галузях науки, техніки, економіки та медицини збільшився інтерес до використання штучних нейронних мереж (НМ). Багато в чому популярність НМ мереж пояснюється можливістю їх ефективного застосування в випадках, коли класичні аналітичні методи не спрацьовують. В теоретичних роботах присвячених НМ наголошується, що їх використання доцільне в задачах класифікації та кластеризації образів, апроксимації функцій, прогнозування, оптимізації, управління, створення інформаційно-обчислювальних систем з асоціативною пам'яттю. Частково або в комплексі вирішувати перераховані задачі доводиться при розробці методів і засобів технічного контролю, управління та захисту комп'ютерних систем.

Слід відзначити, що на сьогодні у відкритому доступі знаходиться достатньо науково-практичної літератури, присвяченій або проблемі вдосконалення новітніх типів НМ, або застосуванню таких НМ для вирішення актуальних задач в різних галузях. В той же час помічено, що в багатьох випадках механізми вдосконалення базуються на застосуванні класичних типів НМ, опис яких наведено у багатьох розрізнених джерелах. Ці обставини значно ускладнюють отримання цілісного уявлення про можливості класичних типів НМ, а відповідно, звужують перспектив їх використання. Виправленню вказаного недоліку присвячено даний навчальний посібник, в якому з єдиних позицій наведено опис класичних архітектур НМ, особливості їх функціонування, переваги та недоліки. Також наведено опис підходу для оцінки ефективності типу НМ при її практичному застосуванні.

Розділ 1. Передумови застосування штучних нейронних мереж

В загальному випадку під терміном штучні НМ розуміють мережу елементів (штучних нейронів), пов'язаних між собою синаптичними зв'язками. Нейрони та зв'язки між ними утворюють структуру НМ. НМ з довільною структурою показана на рис.1.

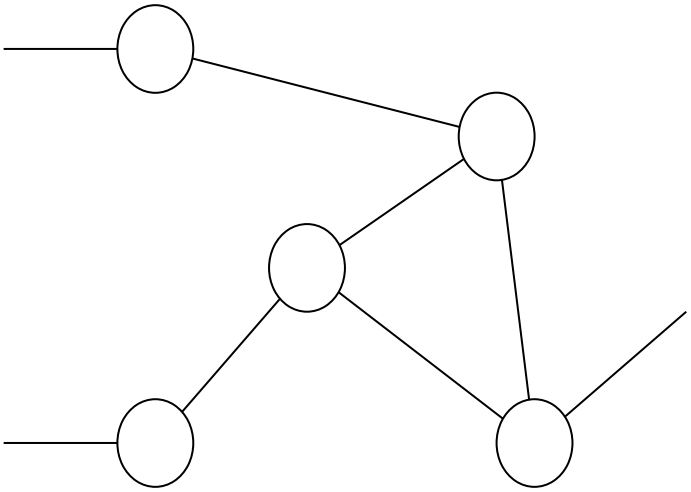


Рис. 1. Приклад НМ з довільною структурою

З точки зору механізму реалізації обчислювальних процесів НМ моделюють функціонування біологічних процесів, які відбуваються в людському мозку. Однак в порівнянні з людським мозком сучасні НМ представляють собою значно спрощену абстракцію. Робота НМ полягає в перетворенні вхідної інформації у певну сукупність вихідних сигналів. Перетворення відбувається за рахунок зміни внутрішнього стану НМ. При цьому НМ, як правило, оперують цифровими величинами.

Зв'язки, по яких інформація передається в напрямку вхід - вихід, називаються прямими. Зв'язки, по яких інформація передається в напрямку вихід - вхід, називаються зворотніми.

Мережі, в яких існують тільки прямі зв'язки, називаються мережами з прямим розповсюдженням сигналу. Мережі з зворотніми зв'язками називаються рекурентними.

Досить часто в структурі НМ виділяють групу нейронів з однаковими зв'язками - нейронний шар. Приклад НМ, що складається із двох шарів нейронів, показаний на рис. 2. Загальновідомим прикладом НМ, яка складається із декількох шарів нейронів, є БШП.

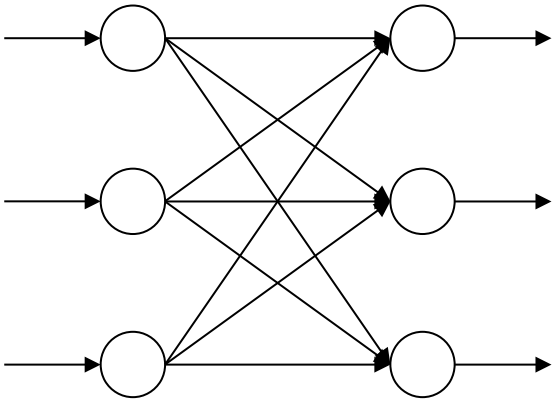


Рис. 2. Приклад двохшарової НМ

Шаблон, що визначає наявність зв'язків між окремими нейронами, називається топологією мережі. Розрізняють повнозв'язну та не повнозв'язну топологію НМ.

Нейрони, з яких складається НМ, представляють собою прості процесори, обчислювальні параметри яких обмежуються деякими правилами комбінування вхідних сигналів и правилом активації, яке дозволяє визначити вихідний сигнал по сукупності вхідних. Вихідний сигнал нейрону може передаватись іншим нейронам мережі по синаптичним (зваженим) зв'язкам, кожному із яких відповідає ваговий коефіцієнт, що також називається вагою зв'язку.

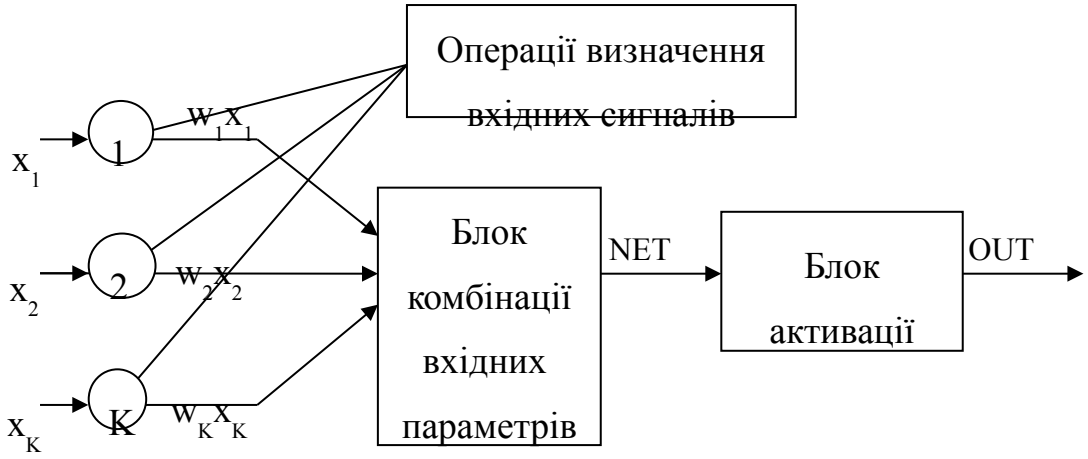
Вхідні зв'язки нейронів отримали назву дендритів, а вихідний зв'язок - аксону. Нейрони, призначені для безпосереднього прийому інформації із зовнішнього середовища, називаються вхідними. Нейрони, що віддають інформацію безпосередньо у зовнішнє середовище, називаються вихідними. Інші нейрони називаються проміжними або схованими. Вони утворюють один або декілька СШН.

Типова формальна модель нейрону показана на рис.3. Комбінування вхідних сигналів (зв'язків) нейрону полягає в розрахунку суми їх зважених

значень та деякої константи, яка дістала назву зсуву. Сумарний вхідний сигнал нейрону (*NET*) розраховується так:

$$NET = \sum_{i=1}^{K} x_i w_i , \quad (1)$$

де K - кількість вхідних зв'язків, x_i - величина і-го зв'язку, w_i - вага і-го зв'язку.



x_i - і-ий вхідний сигнал, K - кількість вхідних зв'язків (сигналів), $w_i x_i$ - і-ий зважений вхідний сигнал, NET - сумарний вхідний сигнал, OUT - вихідний сигнал

Рис. 3. Типова формальна модель нейрону

В загальному випадку вхідні сигнали, зсув та вагові коефіцієнти можуть приймати будь-які значення із діапазону дійсних чисел, а на практиці їх величини визначається специфікою конкретної задачі. Зв'язки, яким призначені від'ємні вагові коефіцієнти називаються гальмуючими, а зв'язки з додатніми ваговими коефіцієнтами називаються збуджуючими. Блок активації нейрону призначений для розрахунку вихідного сигналу нейрону. Як правило, для цього сумарний вхідний сигнал підлягає нелінійному перетворенню:

$$OUT = F(NET - \theta), \quad (2)$$

де θ - гранична величина або зсув, F -функція активації.

Досить часто зсув інтерпретують як зв'язок з ваговим коефіцієнтом, що дорівнює w_0. В цьому випадку вирази (1, 2) можна записати так:

$$NET = \sum_{i=0}^{K} x_i w_i , \quad (3)$$

$$OUT = F(NET) \quad (4)$$

Відзначимо, що механізм обробки інформації в формальній моделі нейрону, заданій виразами (1-4), багато в чому відрізняється від свого біологічного прототипу. Основні відмінності полягають в наступному:

- Не існує механізму визначення затримки реалізації вихідного сигналу.
- Відсутня модуляція рівня вхідного сигналу щільністю нервових імпульсів.
- В більшості НМ не використовується ефект синхронізації функціонування нейронів.
- Відсутній сторонній механізм типу гормональної регуляції активностей нейронів, що регулює функціонування НМ в цілому.
- Не використовується механізм динамічної настройки активаційного порогу та вагових коефіцієнтів в процесі функціонування НМ.
- Використовується тільки збуджуючі та гальмуючі зв'язки між нейронами.

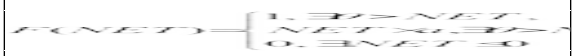
За рахунок вказаних відмінностей використання НМ для моделювання динамічних систем потребує додаткових елементів, які не входять до складу мережі. Також слід розраховувати, що пластичність НМ та її адаптація до зміни зовнішніх умов значно поступаються біологічним аналогам.

Характеристики найбільш відомих функцій активації для нейронів, що входять до складу класичних НМ, представлені в табл. 1. Зазначимо, що в даній таблиці символом θ позначено порогове значення (зсув), символом a - коефіцієнт крутизни, а символом σ - радіус функції Гауса.

В багатьох сучасних НМ використовуються складні активаційні функції. Наприклад, в СНМ в якості функцій активації використовуються складні функції нечіткої логіки, а в глибоких повнозв’язних НМ та в згорткових НМ використовуються функції активації типу ReLU та softmax. Також відомі приклади застосування складених функцій активації.

Таблиця 1

Функції активації штучних нейронів

Назва	Формула	Область використання
Лінійна	$F(NET) = NET \times a$	Вхідні нейрони всіх типів НМ.
Логістична (сигмоїдальна)	$F(NET) = \frac{1}{1 + e^{-a \times NET}}$	Всі типи мереж з прямим розповсюдженням сигналу, включаючи БШП
Лінійна з погашенням від'ємних імпульсів	$F(NET) = \begin{cases} NET \times a, \exists NET > \theta \\ 0, \exists NET \le \theta \end{cases}$	Вхідні нейрони всіх типів НМ.
Порогова	$F(NET) = \begin{cases} 1, \exists NET \ge \theta \\ 0, \exists NET < \theta \end{cases}$	Одношаровий перспетрон, НМ Хопфілда, ДАП
Гіперболічний тангенс	$F(NET) = \frac{e^{a \times NET} - e^{-a \times NET}}{e^{a \times NET} + e^{-a \times NET}}$	Всі типи мереж з прямим розповсюдженням сигналу, включаючи БШП
Гаусова крива	$F(NET) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{NET}{2\sigma^2}}$	Проміжні нейрони для ймовірністних НМ та РБФ
Гістерезис		НМ Хеммінга

На практиці вибір функції активації обумовлюється специфікою задачі, ефективністю комп'ютерної реалізації та алгоритмом навчання НМ. Загальноприйнятого алгоритму вибору функції активації на сьогодні не існує, при цьому відомі деякі обмеження використання певних видів цієї функції.

Більшість моделей НМ потребують навчання, в процесі якого визначаються такі внутрішні параметри мережі, при яких вона найкраще вирішує поставлену задачу. Найчастіше навчання НМ полягає в розрахунку

вагових коефіцієнтів синаптичних зв'язків між нейронами, а структура НМ (кількість нейронів та наявність зв'язків між нейронами) визначається перед навчанням.

В процесі навчання мережі пред'являються навчальні приклади, кожному з яких відповідає власний вектор ознак. При цьому вагові коефіцієнти змінюються так, щоб НМ найкраще відповідала цим прикладам. Зміна коефіцієнтів реалізується відповідно наперед заданому алгоритму навчання. В деяких алгоритмах, наприклад, "нейронний газ" крім модифікації коефіцієнтів передбачено зміну кількості нейронів в мережі.

Розрізняють два основних типи навчання НМ - безпосередньої обробки навчальних даних та ітераційний. В першому випадку вагові коефіцієнти визначаються шляхом безпосередньої одноразової обробки параметрів навчальних прикладів. Другий випадок характеризується багатократним пред'явленням НМ навчальних прикладів. Вагові коефіцієнти уточнюються під час показу кожного прикладу доти, доки мережа не буде виконувати свої функції з заданою якістю.

Ітераційне навчання що базується на прикладах, до складу яких входять тільки вхідні дані НМ, називається навчанням "без вчителя". Якщо ж в прикладах крім вхідних є очікувані вихідні дані, то таке навчання називається навчанням "з вчителем". Крім того, існують менш відомі проміжні методики навчання, наприклад - "з підкріпленням". При апріорно заданих показниках якості, основною характеристикою методики навчання є термін її проведення, який напряму залежить від кількості ітерацій.

На сьогодні найбільш потужними є НМ, які навчаються по методиці навчання "з вчителем". Відзначимо, що можливість використання тієї чи іншої методики навчання залежить від наявності навчальних даних, топології НМ, правил комбінування вхідних сигналів нейрону та виду функції активації. Наприклад, НМ з нейронами в яких використовується порогова функція активації не можливо навчати за допомогою методу "зворотнього розповсюдження помилок", який є найбільш відомим серед методів навчання "з вчителем". Після навчання НМ може розпізнавати невідомі вхідні дані, або нести якесь інше змістовне навантаження. Інформація про отриманий під час

навчання досвід зберігається у вигляді вагових коефіцієнтів зв'язків.

Основними конструктивними параметрами НМ є кількість вхідних, схованих і вихідних нейронів, структура зв'язків (топологія мережі), правила розповсюдження сигналів в мережі, правила комбінування сигналів, що входять в нейрон, правила обчислення вихідного сигналу нейрона та правила навчання, що коректують зв'язки в мережі. Ці параметри використовують в якості критеріїв класифікації НМ. Наприклад, по критерію структури зв'язків, розрізняють одношарові та багатошарові НМ. Крім того, застосовуються цілий ряд додаткових критеріїв класифікації НМ. Наприклад, серед багатошарових НМ виділяють монотонні мережі. Сукупність вказаних параметрів визначають архітектуру мережі.

Відомий ряд архітектур, що вже стали класичними - мережа пошуку максимума, вхідна та вихідна зірка, одношаровий перспетрон, БШП, мережа РБФ, мережі Хопфілда, Хеммінга, Коско, Маккаллока-Питтса, Кохонена та Гросберга. Достатньо відомі ймовірністні мережі, ДАП та мережа АРТ. Крім того, розроблена значна кількість специфічних архітектур - рекурсивна автоасоціативна пам'ять, модульні НМ, когнітрон, неокогнітрон, мережі, що використовують апарат нечіткої логіки, СНМ, різні типи рекурентних мереж та багато інших. При цьому для кожного класу прикладних задач використовується своя архітектура НМ.

З точки зору теорії технічного контролю, найбільш важливою характеристикою НМ, яка взагалі визначає можливість її практичного використання, є помилка контролю мережі. Під цим терміном будемо розуміти помилку при класифікації мережею вхідного образу (вектору), як одного із еталонних образів. В теорії НМ аналогом помилки контролю є помилка узагальнення мережі. Зазначимо, що властивість узагальнення характеризує можливості НМ проводити правильну класифікацію вхідних образів, що не були представлені в навчальних даних. Розрахунковий вираз помилки узагальнення складається із двох частин - помилки опису моделі та помилки апроксимації навчальних даних. Таким чином, помилка узагальнення характеризує не тільки помилку розпізнавання невідомих образів, але й помилку НМ на навчальних даних.

При визначеній моделі НМ помилка апроксимації в першу чергу залежить від методу та алгоритму навчання мережі. У випадку використання ітераційних алгоритмів навчання помилка апроксимації також залежить від максимально допустимої кількості ітерацій, що в свою чергу залежить від максимально допустимого терміну навчання, від потужності апаратного забезпеченння та від ефективності програмної реалізації НМ. Для сучасних НМ можливо досягнути достатньо низьких величин помилки апроксимації. Наприклад, максимальна відмінність між модельними та вхідними даними при апроксимації нелінійних функцій за допомогою БШП становить близько 1%. Зазначимо, що в багатьох випадках зменшення помилки апроксимації пропорційне збільшенню потужності НМ. Тобто для досягнення необхідної помилки апроксимації рекомендується збільшити кількість нейронів, шарів нейронів та кількість синаптичних зв'язків.

Помилка опису моделі характеризує адекватність побудованої НМ тим процесам, що лежать в основі формування вхідних образів. Величина помилки опису залежить від формальної моделі нейрону, топології НМ, потужності НМ, адекватності навчальної інформації. Наведемо загальноприйняті шляхи зменшення помилки опису моделі:

- Використання тієї архітектури НМ, яка найбільш повно відповідає специфіці прикладної задачі. На сьогодні вибір архітектури відбувається емпірично та значною мірою залежить від традиційної сфери її застосування та наявного програмно-апаратного забезпечення. Найчастіше використовують НМ з однією із класичних архітектур. Інколи розробляють НМ з оригінальною архітектурою, що включає формальну модель нейрону, яка відрізняється від загальновідомої моделі (1-4).
- Використання із декількох можливих НМ з заданою топологією найменш потужної. При цьому мінімально допустима потужність мережі визначається максимально допустимою помилкою апроксимації навчальних даних. Водночас помилку апроксимації можливо розрахувати тільки при навчанні вже побудованої НМ. Тому досить часто визначення достатньої потужності НМ реалізується експериментально.

- Невідомі вхідні образи не повинні значно відрізнятись від навчальних даних. Наприклад, при апроксимації функції виду $y = f(x)$ інтервал навчальних даних $X_н = [a, b]$ повинен перекривати інтервал невідомих даних $X_р = [c, d]$. Однак в загальному випадку чіткого алгоритму визначення відповідності навчальних та невідомих вхідних даних на сьогодні не існує.

- Основний закон, який повинен моделюватись мережею повинен добре прослідживатись в навчальних даних, а не затінюватись в них несуттєвими закономірностями. Для цього навчальні дані перед використанням в НМ проходять попередню обробку. Під цією обробкою розуміється нормалізація даних, їх фільтрація та перекодування.

Вважається, що в багатьох практичних сферах НМ дозволяють досягти помилки узагальнення 90% при одночасній помилці апроксимації ≈98-100%. При цьому однією із основних передумов використання НМ є складність формалізації практичної задачі, що призводить до неефективності застосування класичних математичних методів для її вирішення.

В теоретичних роботах, присвячених НМ, наголошується, що використовувати їх доцільне в задачах:

- Класифікації образів. Задача полягає в розрахунку приналежності вхідного образу, представленого вектором ознак, одному або декільком попередньо визначеним класам.

- Кластеризації/категоризації. Задача відрізняється від класифікації образів тільки тим, що класи наперед не визначені, хоча у багатьох випадках кількість класів все-таки заздалегідь вказується.

- Апроксимації функцій. Задача полягає в знаходженні оцінки функції по відомій вибірці її параметрів і значень. НМ рекомендується використовувати у випадках, коли вибірка спотворена шумом і знайти аналітичне рішення важко. Одночасно з цим розв'язується задача фільтрації даних, тобто виділення корисного сигналу з фонового шуму.

- Прогнозу. Необхідно на підставі множини дискретних відліків $\{f(t_1), f(t_2), f(t_j)\}$ передбачити значення $f(t_{j+1})$ у момент часу t_{j+1}.

- Оптимізації, тобто знаходження рішень, які задовольняють системі обмежень і максимізують або мінімізують цільову функцію. НМ рекомендується використовувати при неможливості сформувати явні функціональні залежності для обмежень та/або для цільової функції.

- Управління з еталонною моделлю. В цих задачах метою управління є розрахунок такої вхідної управляючою дії на керовану систему, при якій вона слідує по бажаній траєкторії, що визначається еталонною моделлю.

- Створення інформаційно-обчислювальних систем з пам'яттю, що адресується за змістом (асоціативної пам'яті). В таких системах пам'ять може бути відновлена по частковому або спотвореному змісту. Використання асоціативної пам'яті дозволяє вирішувати задачі стиснення інформації, відновлення даних та підвищує живучість обчислювальних систем.

Перелік традиційних передумов та сфер застосування НМ підтверджують доцільність їх використання для розв'язання задач контролю, управління та захисту комп’ютерної системи.

По перше, контроль параметрів безпеки є важкоформалізуємою задачею по причині суб'єктивних процесів, що є основою зміни цих параметрів. По друге, розпізнавання небезпечного стану контрольованих параметрів комп’ютерної системи та оптимізація управління параметрами захисту відносяться до тих задач, де НМ вже довели свою ефективність. Водночас слід врахувати обмеження на використання НМ. В першу чергу це стосується тих задач, для розв'язання яких існує формалізований математичний апарат. Крім того, деякі фахівці застерігають, що НМ багато в чому є аналогом статистичних методів аналізу інформації і, як наслідок, схильні помилятись при застосуванні зловмисником нестандартних прийомів. Однак в багатьох випадках появі нестандартних прийомів можливо запобігти як при постановці задачі, так і за допомогою стандартних засобів моніторингу та управління. Також в науково-прикладних роботах зазначається, що представлення НМ у вигляді простого статистичного фільтру є дещо поверхневим. Разом з тим в новітніх типах НМ додатково реалізована аналогія з засобами класичного штучного інтелекту, наприклад, з семантичними мережами. Тому слід сподіватись, що сучасні типи НМ

дозволять правдиво діагностувати ситуації, які не були представлені в початкових статистичних даних. Окреслюючи сферу застосування НМ слід врахувати, що можливості мережі значною мірою залежать від її архітектури. Результати досліджень вказують на те, що розвиток сучасних НМ йде шляхом пристосування базових архітектур для вирішення практичних задач. При цьому ряд архітектур вже втратили свої передові позиції і використовуються тільки в якості допоміжних. Тому слід зосередити увагу на адаптації НМ з найбільш перспективною базовою архітектурою до задачі моніторингу діагностичних параметрів піддослідної системи та до спорідненої задачі управління параметрами такої системи. Базуючись на висновках літературних робіт та аналізі вказаних науково-прикладних задач, в навчальному посібнику розглянуто НМ типу БШП, РБФ, АРТ, мережі Хеммінга, Хопфілда, Коско (ДАП) та Кохонена. Відзначимо, що обрані мережі погано пристосовані для аналізу тексту. Тому, крім базових архітектур, розглянуто СНМ, яка є однією із найбільш досконалих класичних НМ в галузі обробки текстової інформації.

Розділ 2. Багатошаровий персептрон

В загальному випадку БШП представляє собою НМ, яка складається із декількох послідовно з'єднаних між собою шарів штучних нейронів. Типова структура БШП показана на рис. 4. Зовнішня інформація спочатку поступає у вхідний шар, що складається тільки із сенсорних елементів (вхідних нейронів). Основними завданнями цього шару є прийом та розповсюдження вхідної інформації по іншим шарам НМ. Далі знаходиться один або декілька СШН, в яких власне і відбувається основна обробка інформації. Результати цієї обробки відображаються у вихідному шарі. Відзначимо, що при підрахунках кількості шарів вхідний шар не враховують. Наприклад, ДШП складається із вхідного, одного схованого та вихідного шару.

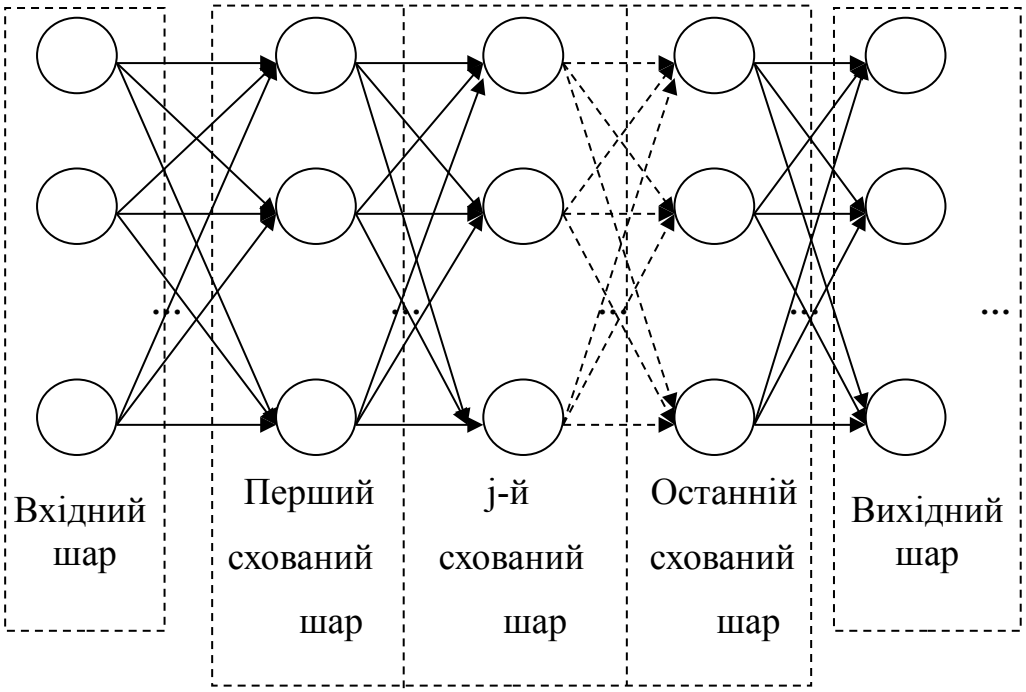


Рис. 4. Типова структура багатошарового персептрону

Як правило, кожен нейрон СШН приймає всі вихідні сигнали попереднього шару, а його вихідний сигнал надсилається всім нейронам наступного шару. Особливістю БШП є наявність тільки прямих гальмуючих або збуджуючих зв'язків між сусідніми шарами нейронів. При цьому кожен нейрон в СШН характеризується унікальним вектором вагових коефіцієнтів.

Для вхідних нейронів досить часто використовується лінійна, лінійна з погашенням від'ємних імпульсів та порогова функції активації. Для схованих нейронів монотонні функції активації не впливають на результати розпізнавання даних. Але використання певного типу функції може підвищити інформативність результатів розпізнавання. Наприклад, сигмоїдальна функція активації дозволяє трактувати результати класифікації, як ймовірність віднесення вхідного образу до відповідного класу. Тому для схованих нейронів найчастіше використовують порогову та сигмоїдальну функцію активації.

В більшості випадків вихідні елементи БШП виконують тільки розрахунок власних вхідних сигналів, тому функція активації для них не потрібна.

Розрахунок основних параметрів *j*-го нейрону в *l*-му СШН можна провести так:

$$NET_j^l = \sum_{i=1}^{K_j^l} w_{ij}^l x_{ij}^l , \tag{5}$$

$$OUT_j^l = F(NET_j^l - \theta_j^l), \tag{6}$$

$$x_{ij}^{l+1} = OUT_j^l, \tag{7}$$

де i - номер входу, j - номер нейрону в шарі, l - номер СШН, K_j^l - кількість вхідних зв'язків j-го нейрону в l-му шарі, w_{ij}^l - ваговий коефіцієнт i-го входу j-го нейрону в l-му шарі, θ_j^l - пороговий рівень активації j-го нейрону в l-му шарі, x_{ij}^l - i-й вхідний сигнал нейрону в l-му шарі, F - функція активації, OUT_j^l - вихідний сигнал j-го нейрону в l-му шарі, NET_j^l - сумарний вхідний сигнал j-го нейрону в l-му шарі.

Відзначимо, що для першого СШН кількість вхідних зв'язків нейрону дорівнює кількості нейронів у вхідному шарі. Для інших СШН, K_j^l дорівнює кількості нейронів в попередньому СШН. Як видно із (1.5-1.7) кожен СШН виконує нелінійне перетворення від лінійної комбінації сигналів попереднього шару. В цілому БШП може сформувати на виході довільну багатовимірну функцію (f), від множини вхідних параметрів $\{x\}$:

(8)

$$f(x) = F\left(\sum_{i=1}^{K^N} w_i^N \ldots \left(\sum_{i=1}^{K^2} w_i^2 F\left(\sum_{i=1}^{K^1} w_i^1 x - \theta^1\right) - \theta^2\right) \ldots - \theta^N\right),$$

де x - вектор вхідних параметрів, w_i^N - вектор вагових коефіцієнтів нейронів в N-му шарі, θ^N - вектор порогів активації нейронів в N-му шарі, K^N - кількість вхідних зв'язків в N-му шарі, N-кількість схованих шарів.

Таким чином БШП може розрахувати вихідний вектор y для деякого вхідного вектору x. Відповідно, умовою задачі, яка може бути поставлена БШП, повинна бути множина вхідних векторів (вхідних образів):

$$x \in \left\{ x^1, \ldots x^i, \ldots x^S \right\}, \tag{9}$$

де i - номер вхідного вектору, а S - кількість вхідних векторів.

Кожен із вхідних векторів складається із N_1 компонент, тобто:

$$x^i = \begin{pmatrix} x_1^i \\ \ldots \\ x_{N_1}^i \end{pmatrix}, \tag{10}$$

Як правило кількість компонент вхідного вектору дорівнює кількості нейронів у вхідному шарі. Вирішенням задачі з умовою (9) буде множина вихідних векторів (вихідних образів):

$$y \in \left\{ y^1, \ldots y^i, \ldots y^S \right\}. \tag{11}$$

Кожен з вихідних векторів складається із N_0 компонент:

$$y^i = \begin{pmatrix} y_1^i \\ \ldots \\ y_{N_0}^i \end{pmatrix}, \tag{12}$$

де N_0 - кількість нейронів у вихідному шарі персептрону.

Навчання БШП виконується методом "навчання з вчителем" та полягає в визначенні таких вагових коефіцієнтів зв'язків нейронів СШН, які дозволяють найкраще вирішувати поставлену задачу.

Процес навчання починається з ініціалізації вказаних вагових коефіцієнтів випадковими величинами. Після цього на вхід НМ подаються параметри, що відповідають відомим образам. Відзначимо, що з точки зору БШП відомий образ означає відомий набір значень вихідних параметрів. Якщо реальні вихідні параметри відрізняються від цих значень, то вагові

коефіцієнти нейронів схованих шарів уточнюються за допомогою спеціальних алгоритмів. Найбільш популярним із них є алгоритм оберненого розповсюдження помилок, що базується на оцінці помилок нейронів і-го СШН, як зваженої суми помилок наступного (і+1) шару. При цьому помилки останнього (вихідного) шару нейронів відомі.

Під час навчання інформація розповсюджується від нижніх шарів до верхніх, а оцінки помилок мережі в зворотному напрямку. Процес навчання багатоітераційний і полягає в мінімізації функції помилки персептрона на всій множині навчальної вибірки. Пошук мінімуму помилки може реалізуватись методом градієнтного спуску.

Хоча алгоритм оберненого розповсюдження помилок знайшов широке практичне застосування, він має декілька серйозних недоліків. Основним недоліком є низька збіжність алгоритму, яка пояснюється тим, що в багатьох випадках локальний напрям градієнту не співпадає з напрямком до глобального мінімуму. При цьому уточнення вагових коефіцієнтів виконується незалежно для кожної пари образів із навчальної вибірки. Відповідно зменшення помилки перспетрону для деякої пари образів може призвести до збільшення цієї ж помилки для інших пар. З цієї точки зору взагалі немає ніяких гарантій знаходження мінімальної помилки. Крім того, мтод оберненого розповсюдження помилок може застосовуватись тільки при використанні гладких функцій активації нейронів СШН.

Для зменшення вказаних недоліків використовуються модифікації методу, які полягають в застосуванні різних функцій оцінки помилки БШП та процедур визначення напрямку та величини кроку пошуку оптимуму. Достатньо відомі та апробовані методи пов'язаних градієнтів, Левенберга-Маркара, швидкого навчання за допомогою зменшення розмірності обчислень на базі використання теорії рядів Вольтера, швидкого розповсюдження та дельта- метод.

Також відомі вдалі спроби проводити навчання БШП за допомогою генетичних алгоритмів. В багатьох випадках перевагою цих методів є більш висока швидкість навчання відносно методу оберненого розповсюдження помилок. Однак не достатня точність цих методів, велика кількість

управляючих параметрів, а також деякі обмеження на структуру БШП ускладнюють їх практичне використання. Крім того, в випадку великого обсягу навчальних даних, серед яких є надлишкові, точність визначення вагових коефіцієнтів нейронів за допомогою методу оберненого розповсюдження помилок є суттєво вищою.

Доведено, що помилку апроксимації ДШП (ε_a) можливо оцінити так:

$$\varepsilon_a \sim N_1 / L_w, \tag{13}$$

де N_1 - кількість компонент вхідного вектора (розмірність вхідного вектору), L_w - кількість синаптичних зв'язків.

Аналіз (13) вказує на те, що збільшення кількості синаптичних зв'язків, а значить і нейронів в СШН, призводить до більш точної апроксимації невідомої функції. Негативною стороною збільшення кількості нейронів є виникнення перенавчання.

Суть явища перенавчання полягає в тому, що в процесі навчання вагові коефіцієнти настроюються для мінімізації помилки на деякій навчальній вибірці. В випадку відсутності ідеальної та нескінченної навчальної вибірки ця помилка може суттєво відрізнятись від помилки в наперед невідомій множині нових образів. Тому потрібно так настроїти вагові коефіцієнти, щоб БШП міг адекватно узагальнювати результати навчання на нові вхідні дані. Іншими словами необхідно мінімізувати помилку узагальнення (ε), яка складається із помилки апроксимації (ε_a) та помилки опису моделі (ε_o):

$$\varepsilon = \varepsilon_a + \varepsilon_o. \tag{14}$$

В першому наближенні помилку опису БШП можливо оцінити так:

$$\varepsilon_o \sim L_w / P, \tag{15}$$

де P - кількість навчальних прикладів.

Відзначимо, що на відміну від помилки апроксимації помилка опису зростає пропорційно кількості схованих нейронів. Підстановка (13) та (15) в (14) дозволяє отримати вираз для приблизної оцінки помилки узагальнення:

$$\varepsilon \sim \left(N_1 / L_w + L_w / P \right). \tag{16}$$

Після відповідних перетворень (16) отримаємо вираз для приблизної

оцінки оптимальної кількості синаптичних (настроюваних) зв'язків ДШП, що відповідає мінімуму помилки узагальнення:

$$L_w \sim \sqrt{N_1 \times P}. \tag{17}$$

Враховуючи, що кількість нейронів в СШН (L) для ДШП розраховується:

$$L = {L_w}/{N_1}. \tag{18}$$

Приблизну оптимальну кількість схованих нейронів (L_{opt}) в ДШП можливо оцінити так:

$$L_{opt} \sim \sqrt{{P}/{N_1}}. \tag{19}$$

Також відомі дещо інші формули для оцінки оптимальної кількості синаптичних зв'язків та кількості схованих нейронів в БШП з сигмоїдальними функціями активації:

$$\frac{N_0 \times P}{1+\log_2 P} \le L_w \le N_1 \times \left(\frac{P}{N_1}+1\right) \times (N_1 + N_0 + 1) + N_1 \tag{20}$$

$$\frac{P}{10} - N_1 - N_0 \le L_w \le \frac{P}{2} - N_1 - N_0, \tag{21}$$

$$L_w < P \times \varepsilon_{\max}, \tag{22}$$

де L_w - кількість зв'язків між нейронами СШН та вхідними нейронами, ε_{max} - максимальна допустима помилка узагальнення.

Крім того, відома формула для визначення максимальної кількості образів (P), яку може запам'ятати ДШП з пороговою функцією активації СШН:

$$\frac{L_w}{N_0} < P < \frac{L_w}{N_0}\log_2\left(\frac{L_w}{N_0}\right), \tag{23}$$

де N_0 - розмірність вихідного сигналу (кількість вихідних) нейронів.

Місткість ДШП з сигмоїдальною функцією активації виду дещо більша, а місткість ДШП з кількістю СШН більше одного теоретично не визначена, хоча вважається дещо вищою місткості ДШП з тими ж показниками L_w та N_0. При цьому представлено залежність необхідної кількості навчальних прикладів від загальної кількості зв'язків (вагових коефіцієнтів) в БШП і помилки узагальнення та вираз для розрахунку кількості схованих нейронів:

$$P = \frac{L_w}{\varepsilon}, \tag{24}$$

$$L < 2 \times N_1. \tag{25}$$

Враховуючи (18), (24) та (25) отримаємо залежність між кількістю навчальних образів, величиною помилки узагальнення та розмірністю вхідного сигналу:

$$P < \frac{2 \times N_1^2}{\varepsilon}. \tag{26}$$

Відзначимо, що наведені вирази дозволяють визначити тільки наближені значення помилки узагальнення та оптимальної кількості схованих нейронів в БШП. Крім того, формули (23-25) представлені в літературних джерелах без належного теоретичного обґрунтування. В той же час, власні експериментальні дослідження вказують на те, що для моделювання достатньо складних і неоднорідних процесів помилка опису моделі не відповідає виразу (15). В цих випадках кількість схованих нейронів буде більшою, ніж кількість, розрахована за допомогою (19-23, 26), та повинна визначатись експериментально.

Досить часто на практиці для вирішення проблеми оцінки помилки узагальнення (якості навчання) використовується емпіричний механізм контрольної крос-перевірки. Цей механізм передбачає розділ навчальної вибірки на дві множини - навчальну та контрольну.

Контрольна множина не використовується в процесі навчання по алгоритму оберненого розповсюдження а застосовується тільки для незалежного контролю результатів навчання. На початку навчання помилка БШП на навчальних та контрольних даних повинна бути приблизно однакова. Якщо це не так, то очевидно, що розподіл даних між дві множини був неоднорідний. В процесі навчання помилка БШП на навчальних даних буде зменшуватись. До тих пір поки навчання зменшує помилку узагальнення, помилка на контрольних даних також буде зменшуватись. Стабілізація або збільшення помилки на контрольних даних вказує на виникнення перенавчання і необхідність закінчення навчання. При цьому, якщо помилка навчання не досягла необхідного мінімальної величини, значить БШП є занадто потужним для вирішення даної задачі. В цьому випадку рекомендують зменшити кількість СШН та/або кількість нейронів в

них. Якщо ж потужності БШП недостатньо, що моделювати потрібну функцію то перенавчання не відбудеться, але обидві помилки не досягнуть необхідної мінімальної величини.

Необхідність проведення багатоітераційного навчання призводить до того, що контрольні дані можуть мати вирішальне значення в побудові моделі БШП. Тим самим значно послаблюється роль цих даних як незалежного критерію якості моделі, бо при великій кількості експериментів є ризик побудувати НМ з низькою помилкою на контрольних даних. З метою надання кінцевій моделі БШП достатньої надійності рекомендується крім контрольної зарезервувати ще тестову множину. Кінцева модель повинна бути протестована на даних цієї множини з метою перевірки того, що результати досягнуті на навчальній та контрольній множині даних достовірні, а не являються артефактами процесу навчання. Для того, щоб досягнути бажаного результату тестові данні повинні застосовуватись тільки один раз. Якщо їх використовувати багатократно, то фактично вони перетворяться в контрольні дані. Однак застосування тестових даних доцільно тільки у випадку великого обсягу початкових даних, що не завжди можливо в практичній діяльності.

Крім розглянутої ранньої зупинки навчання, для зменшення помилки узагальнення та кількості синаптичних зв'язків використовують методи розрідження і поетапного нарощення зв'язків. В методах розрідження зв'язків відбувається видалення зв'язків з малими ваговими коефіцієнтами без суттєвого погіршення апроксимуючих властивостей НМ. Для цього в функціонал помилки апроксимації вводиться штрафна складова, використання якої не впливає на зміну зв'язків з великими ваговими коефіцієнтами, але експоненціально зменшує малі вагові коефіцієнти.

Методи нарощення зв'язків (конструктивні алгоритми) базуються на динамічному збільшенні кількості схованих нейронів в процесі навчання. Особливістю цих методів є те, що невеликі зміни в структурі мережі не призводять до необхідності її повного перенавчання. Оскільки складність навчання БШП пропорційна квадрату кількості вагових коефіцієнтів, навчання по частинам більш вигідне ніж навчання великої мережі:

$$L_w^2 = \left(\sum_{i=1}^{K} L_{w,i} \right)^2 < \sum_{i=1}^{K} L_{w,i}^2 , \qquad (27)$$

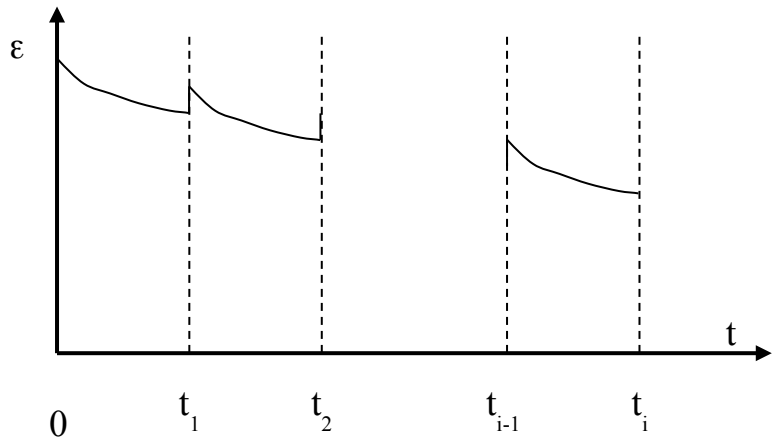
де K - кількість синаптичних зв'язків, $L_{w,i}$ - i-й синаптичний зв'язок, $\sum_{i=1}^{K} L_{w,i}^2$ - кількість вагових коефіцієнтів при навчанні по частинам.

За рахунок цього можливо досягнути високих темпів навчання НМ. Одним із найбільш поширених є алгоритм динамічного додавання нейронів, який передбачає, що початково використовується НМ з кількістю нейронів заздалегідь недостатньою для вирішення задачі. Навчання відбувається до тих пір, поки помилка не перестане зменшуватись і не буде виконуватись умова:

$$\begin{cases} \dfrac{E(t) - E(t - \delta)}{E(t_0)} < \Delta, \\ t \geq t_0 + \delta. \end{cases} \qquad (28)$$

де t -термін навчання, Δ - порогова величина помилки навчання, δ - мінімальний термін навчання між приєднанням нового нейрону, E - помилка навчання, t_0 - момент приєднання нового нейрону.

Після виконання умови (28) в СШН БШП додається новий нейрон, вагові коефіцієнти зв'язків якого ініціюються невеликими числами. Навчання НМ відбувається знову до виконання умови (28). При цьому помилка БШП з початку різко збільшується, а потім швидко сходиться до меншого значення. Залежність помилки при приєднанні нового нейрону показана на рис. 5.



ε - помилка БШП, t - термін навчання, $t_1, t_2, \ldots, t_{i-1}$ - моменти додавання нових нейронів, t_i - момент завершення навчання.

Рис. 5. Залежність помилки БШП від кількості нейронів при використанні динамічного алгоритму додавання нейронів

Додавання нейронів відбувається доки загальна помилка БШП не досягне заданої величини. Доведено, що при використанні алгоритму динамічного додавання загальний час навчання БШП приблизно в 1,4 рази більший, від часу навчання з необхідною кількістю нейронів.

Після навчання БШП може розпізнавати вхідні дані, або нести інше змістовне навантаження. Інформація про отриманий в процесі навчання досвід зберігається у вигляді вагових коефіцієнтів зв'язків схованих нейронів. Відзначимо, що достатньо часто при вирішенні задачі класифікації, вхідний образ співвідноситься з деяким еталоном якщо:

$$\sqrt{\sum_{i=1}^{N_0} \frac{(y_i - z_i)^2}{z_i^2}} < \eta, \qquad (29)$$

де y_i - i-й вихідний параметр, що відповідає вхідному образу, z_i - i-й вихідний параметр, що відповідає еталону образу, η - поріг розпізнавання.

Вважається, що поріг розпізнавання $\eta \in [0{,}01...0{,}05]$.

Інколи для класифікації використовують більш складні вирази, наприклад:

$$\begin{cases} X^k \in Z^j, \forall\{\partial^j\} \subset \{\eta\} \\ \partial_i^j = y_i - z_i, \partial_i^j \in \{\partial^j\}, \end{cases} \qquad (30)$$

де X^k - k-й вхідний образ, Z^j - j-й еталон, $\{\eta\}$ - множина порогів розпізнавання вихідних параметрів.

Дослідження показують, що для представлення довільного функціонального відображення, заданого навчальною вибіркою достатньо всього двох СШН. Цей результат відомий як теорема Колмогорова. Також доведено, що одного СШН з сигмоїдальною функцією активації достатньо для апроксимації будь-якої випуклої функції із наскільки завгодно високою точністю. Цього достатньо для моделювання більшості реальних задач класифікації образів. В той же час для моделювання складних функціоналів рекомендують використовувати БШП з більшою кількістю СШН.

На практиці найчастіше використовують БШП з кількістю СШН від 1 до 3, хоча промислові програмні пакети можуть реалізувати більш 1000 схованих шарів. Вважається, що збільшення кількості СШН дозволяє зменшити загальну кількість нейронів, необхідних для адекватного відображення.

Негативними факторами цього збільшення є теоретична невизначеність точної кількості схованих нейронів, складність програмної реалізації, відносно низька швидкість функціонування та занадто висока точність підгонки апроксимаційної функції до навчальних даних. Класичні методи розробки БШП передбачають виконання наступних етапів:

1. Визначити номенклатуру та допустимі величини вхідних параметрів.
2. Підготувати тестову, контрольну та навчальну вибірку.
3. Визначити максимальну та мінімальну межу загальної кількості схованих нейронів.
4. В межах допустимої області вибрати загальну кількість схованих нейронів.
5. Вибрати кількість СШН та кількість нейронів в кожному з цих шарів.
6. Вибрати вид та параметри функцій активації для всіх типів нейронів.
7. Провести навчання.
8. Провести тестування.
9. Якщо результати тестування не задовільні - змінюємо параметри БШП. Для цього повторити п.4-8.

Таким чином, для розв'язання практичної задачі необхідно сформувати множину вхідних параметрів (п.1), розробити архітектуру БШП (п. 2-6) та провести його навчання (п. 2, 7-9). Для повноти оцінки придатності використання БШП в задачах комп’ютерної діагностики розглянемо підхід до визначення обчислювальної складності навчання НМ такого типу. Як відомо, в більшості розповсюджених методів навчання оптимальний розподіл вагових коефіцієнтів шукається за допомогою градієнтних методів пошуку мінімуму помилки на всій множині навчальних даних. Приблизну кількість обчислювальних операцій (ξ_1) потрібних для розрахунку градієнта функції

помилки $\partial E / \partial L_w$ можливо визначити так:

$$\xi_1 \sim P \times L_w, \tag{31}$$

де L_w - кількість зв'язків, P - кількість навчальних прикладів.

Враховуючи, що швидкість сходження найкращих методів навчання пропорційна кількості синаптичних зв'язків, загальну кількість обчислювальних операцій (ξ) потрібних для визначення мінімуму помилки, можливо розрахувати так:

$$\xi \sim P \times L_w^2. \tag{32}$$

Вираз (32) дозволяє провести оптимістичну оцінку кількості операцій необхідних для навчання БШП.

Приблизну оцінку кількості операцій (ξ_{opt}) необхідних для навчання ДШП з оптимальною кількістю нейронів в СШН отримаємо на основі (1.19) та (1.32):

$$\xi_{opt} \sim N_1 \times P^2, \tag{33}$$

де N_1 - розмірність вхідного сигналу (кількість вхідних нейронів).

Слід врахувати, що для БШП з регулярною структурою кількість синаптичних зв'язків пропорційна добутку числа вхідних та вихідних нейронів:

$$L_w \sim N_1 \times N_0, \tag{34}$$

де N_0 - розмірність вихідного сигналу (кількість вихідних нейронів).

Після підстановки (34) в (32) отримаємо:

$$\xi \sim P \times N_1^2 \times N_0^2 = \frac{P \times N_1^4}{\Omega^2}, \tag{35}$$

де $\Omega = N_1 / N_0$ коефіцієнт стиснення інформації персептроном.

В багатьох практичних задачах очікувана розмірність вхідного (N_1) та вихідного сигналів (N_0) не буде перевищувати 10^3, а загальновживаний термін навчання НМ повинен знаходитись в межах однієї доби ($\approx 10^5$ с). Прикладом такої задачі може бути застосування НМ в системі розпізнавання скриптових комп'ютерних вірусів. При цьому кількість параметрів, що діагностують найбільш поширені скриптові віруси написані на мові JS, $N_1 \leq 100$.

Вихідними сигналами системи розпізнавання скриптових вірусів можуть бути: вірусу немає, вірус є, підозра на вірус, виявлено певний тип вірусу. Таким чином, розмірність вихідного сигналу відповідає очікуваній. Термін навчання НМ вибрано на основі власного практичного досвіду з позицій надійної роботи комп'ютерної системи. При використанні персонального комп'ютера з потужністю приблизно 3×10^7 операцій в секунду, вказаному терміну навчання відповідатиме $\xi\approx3\times10^{12}$ обчислювальних операцій. На основі (26), визначимо, що максимальний обсяг навчальної бази даних на основі класичного БШП становитиме $P\approx5*10^4$ прикладів, що перевищує обсяг баз даних сучасних антивірусних засобів, систем виявлення атак та систем виявлення вторгнень. Крім того, можливо ще підвищити обсяг навчальної бази даних або зменшити термін навчання завдяки навчання НМ на декількох комп'ютерах. При цьому, розрахована помилка узагальнення класичного ДШП з оптимальною кількістю синаптичних зв'язків знаходиться в діапазоні [0,1...0,3], а приблизний оптимальний діапазон величин коефіцієнта стиснення інформації $\Omega\in[1..300]$. Відзначимо, що необхідний коефіцієнт стиснення, який відповідає очікуваним на практиці розмірностям вхідного і вихідного сигналів $\Omega_m = 1000/10 = 100$ належить оптимальному діапазону. Таким чином, отримані результати підтверджують доцільність використання БШП при вирішенні практичних задач.

Суттєвою перевагою БШП є наявність методів отримання знань у вигляді набору класифікуючих правил. Вказані методи отримали назву вербалізації БШП. За їх допомогою з навченого БШП можливо отримати правила виду:

$$\begin{cases} якщо\ (x_1 \Theta q_{1,1}) \wedge (x_2 \Theta q_{2,1}) \wedge \ldots i (x_j \Theta q_{j,1}) \wedge \ldots \wedge (x_{N_1} \Theta q_{N_1,1}) \Rightarrow z_1 \\ \ldots \\ якщо\ (x_1 \Theta q_{1,Z}) \wedge (x_2 \Theta q_{2,Z}) \wedge \ldots \wedge (x_j \Theta q_{j,Z}) \wedge \ldots \wedge (x_{N_1} \Theta q_{N_1,Z}) \Rightarrow z_Z \end{cases}, \qquad (36)$$

де x_i - i-й вхідний параметр, z_i - i-й клас, N_1 - кількість вхідних параметрів, Z - кількість визначених класів, θ - оператори відношення ($\leq$, $\geq$, $=$, $<$, $>$), $q_{i,j}$ - константа, що відповідає i-му вхідному параметру в j-му правилі.

Одним із найбільш відомих методів отримання знань є NeuroRule. Даний метод пристосований для отримання знань із ДШП в якому функцією активації схованих нейронів є гіперболічний тангенс, а функцією активації нейронів вихідного шару є функція Фермі. Отримання знань за допомогою NeuroRule розділяється на три етапи: навчання НМ, розрідження НМ та формування правил (36). Навчання БШП відбувається за допомогою модифікованого методу оберненого розповсюдження помилок.

Особливістю навчання NeuroRule є використання таких функцій помилки НМ, мінімізація яких призводить не тільки до спрямування процесу навчання в сторону правильної класифікації навчальних образів, але й до зменшення вагових коефіцієнтів зв'язків між багатьма нейронами. Зменшення вагових зв'язків необхідно для полегшення процесу розрідження. Розрідження НМ полягає в знищенні нейронів та зв'язків між нейронами, які мало впливають на класифікацію.

Вважається, що зв'язок ($w_{i,j}$) між деяким *i*-тим та *j*-тим нейронами можливо знищити, якщо його значення належить деякому діапазону:

$$w_{i,j} = \begin{cases} w_{i,j}, |w_{i,j}| > \alpha, \\ 0, |w_{i,j}| < \alpha \end{cases}, \qquad (37)$$

де α - константа, що співвідноситься з порогом розпізнавання η.

Нейрон можна знищити, якщо зв'язків з ним не існує. Після знищення малозначущих зв'язків та нейронів необхідно перевірити правильність класифікації НМ, та при необхідності уточнити її структуру (додати нейрони та зв'язки між нейронами). Якщо вхідні параметри представляють собою неперервні величини, то для їх представлення в вигляді дискретних величин використовують бінарні нейрони та кодування типу “термометр”. Наприклад, для дискретного представлення неперервного вхідного параметру $x_i \in [0..90]$, допустимий діапазон його значень розбивають на 3 однакових інтервали - [0..30],]30..60],]60.90]. Кожному із цих інтервалів відповідає власний бінарний нейрон. Якщо величина вхідного параметру x_i належить, наприклад, першому інтервалові, то вихід першого бінарного нейрону буде 1, а виходи другого та третього нейронів 0. Після цього проводиться дискретизація неперервних величин активностей нейронів схованого шару.

Для дискретизації неперервних величин можливий діапазон їх значень кластеризується і замінюється значеннями, що відповідають центрам кластерів. Далі проводиться перевірка точності класифікації. Якщо точність недостатня то процес кластеризації повторюється, але вже з більшою кількістю кластерів. Відзначено, що процес дискретизації неперервних вхідних сигналів та величин активностей нейронів СШН негативно впливає на процес формування правил виводу за рахунок значного збільшення кількості вхідних нейронів та зв'язків між ними та схованими нейронами.

Після проведення дискретизації, використовуючи структуру зв'язків НМ, можливо побудувати матрицю зв'язків між дискретними величинами вхідних сигналів та дискретними значеннями активностей схованих нейронів. Це дозволяє побудувати правила відповідностей між дискретними значеннями вхідних сигналів та дискретними значеннями активностей цих нейронів. Крім того, результати дискретизації дозволяють створити матрицю зв'язків між дискретними значеннями активностей нейронів СШН та величинами виходів НМ. На базі цієї матриці будуються правила відповідності між дискретними значеннями активностей нейронів СШН та заданими класами. Комбінація розглянутих правил дозволяє побудувати набір остаточних класифікуючи правил.

У випадку потужного БШП, для якого навіть після розрідження характерна велика кількість нейронів та зв'язків між нейронами, кількість класифікуючих правил може бути занадто великою. Це значно ускладнює процес виводу та інтерпретації знань із НМ. Ще один недолік процесу отримання знань із НМ пов'язаний з тим, що НМ необхідно попередньо навчити проводити класифікацію. Оскільки для великих баз даних термін навчання достатньо довгий, то і термін отримання знань із НМ потребує багато часу. Однак, якщо отримання правил класифікації можливе, то низька помилка класифікації та робастність НМ дають їм певні переваги перед іншими методами отримання знань при вирішенні практичних задач. Наприклад, вказані правила класифікації можливо використовувати при створенні інструкцій користувачів щодо моніторингу та управління комп’ютерних систем.

На сьогодні практично всі вдосконалення БШП спрямовані на зменшення обчислювальної складності його навчання, яка головним чином залежить від розмірності вхідного і вихідного сигналів, кількості навчальних образів, кількості синаптичних зв'язків НМ та методики навчання.

Для зменшення розмірності вхідного сигналу рекомендується проводити попередню обробку навчальних даних за допомогою методів статистичного аналізу та НМ менш потужних ніж БШП. До таких мереж відносять мережу РБФ, мережу Кохонена, ймовірнісні НМ. Результатом попередньої обробки має бути визначення номенклатури вхідних параметрів, достатньої для вирішення даної задачі.

Вдосконалення методів навчання БШП відбувається по двом основним напрямкам: зменшення кількості обчислювальних ітерацій в методах, які базуються на алгоритмі оберненого поширення помилок та використанні безітераційних алгоритмів навчання. Цікавим прикладом безітераційних алгоритмів навчання є алгоритм, що базується на аналогії між НМ та рядами Вольтера. Це дозволило звести навчання НМ, що використовуються в системах біометричної аутентифікації, до вирішення системи лінійних рівнянь. Однак перешкодою застосуванню цього алгоритму може стати постулат про не корельованість вхідних параметрів та недостатня апробованість результатів досліджень. Також для зменшення кількості синаптичних зв'язків пропонується використовувати ШНМ.

ШНМ це різновид багатошарових НМ прямого розповсюдження, висока обчислювальна ефективність яких досягається за рахунок обмежень на структурну організацію. В ШНМ шари діляться на нейронні ядра. Нейронне ядро це група нейронів, які мають загальне рецепторне поле, тобто отримують один і той же вхідний сигнал. Аналог нейронного ядра можна представити у вигляді ДШП малої потужності, що використовується для розпізнавання окремої частини вхідного образу. Проектування архітектури ШНМ може здійснюватись на основі методів розрідження зв'язків БШП з врахуванням особливостей конкретної задачі. Широкому застосуванню ШНМ заважає недостатня дослідженість методики адаптації їх топології до умов конкретної задачі. Тому використання ШНМ доцільне тільки при

наявності досконалого алгоритму розділу вхідних образів на частини яким відповідають окремі нейронні ядра.

Проведений аналіз типових прикладів дозволяє сформувати висновок про те, що використання в засобах комп'ютерної діагностики модифікацій БШП спрямованих на покращення обчислювальних затрат на навчання потребує серйозного доопрацювання. При цьому для багатьох практичних задач обчислювальна складність навчання БШП не є критичною перепоною.

Розділ 3. Мережа з радіальними базисними функціями

Використання НМ типу РБФ базується на посиланні про те, що для підвищення ймовірності лінійного поділу образів на класи, необхідно розмістити ці образи в просторі високої розмірності деяким нелінійним чином. В найбільш простій формі РБФ представляє собою НМ, що складається із трьох шарів: вхідного, схованого та вихідного. Спрощена схема РБФ з одним нейроном у вихідному шарі показана на рис. 6.

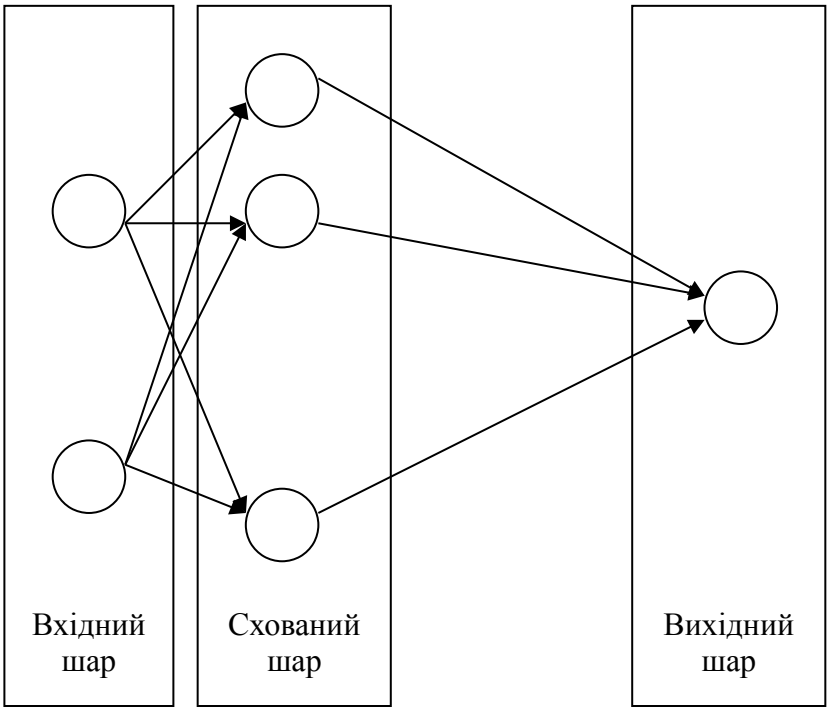


Рис.6. Спрощена схема РБФ

В задачу вхідного шару входить розподіл вхідних даних по нейронам СШН. Сховані нейрони мають радіально-симетричну функцію активації. Кожен з них призначений для зберігання окремого еталонного образу, який відповідає окремому класу. Досить часто кількість схованих нейронів більша кількості вхідних нейронів.

Для j-го нейрону в СШН сумарний вхідний сигнал (*net*) від деякого вхідного вектора (x) розраховується як евклідова норма:

$$net_j = \sqrt{\sum_{i=1}^{N} (x_i - w_{ij})^2}, \quad (38)$$

де x_i - i-а компонента вхідного вектору x, w_{ij} - ваговий коефіцієнт j-го схованого нейрону з i-м вхідним нейроном, N - кількість вхідних нейонів.

В якості функції активації (φ) схованих нейронів досить часто використовують функцію Гауса:

$$\varphi_j(net) = exp\left(-\frac{1}{2\sigma}\sum_{i=1}^{N}(c_i - x_i)^2\right), \quad (39)$$

де $\varphi_j(net)$ -функція активації j-го нейрону, *net* - сумарний вхідний сигнал, x - вхідний вектор, c - центр функції Гауса, σ - радіус функції Гауса.

В цьому випадку вагові коефіцієнти вхідних зв'язків схованих нейронів відповідають центрам функції Гауса.

Після нелінійного перетворення сигнали від СШН потрапляють у вихідний шар нейронів, які мають лінійні функції активації. Розрахунок сумарного вхідного сигналу для нейрону вихідного шару можливо провести відповідно (1.38). Відзначимо, що сукупність значень активностей всіх схованих нейронів визначає вектор на який відображається вхідний вектор:

$$\varphi(x) = |\varphi_1(x), \varphi_2(x), ..., \varphi_M(x)|, \quad (40)$$

де x - вхідний вектор, $\varphi(x)$ - вихідний вектор, $\varphi_i(x)$ - компонента вихідного вектору, пов'язана з i-м схованим нейроном, M - кількість схованих нейронів.

Оскільки функції активації нейронів схованого шару - $\varphi_i(x)$ є нелінійною, то для моделювання будь-якої вхідної інформації достатньо одного СШН з достатньо великою кількістю нейронів. Загальну кількість синаптичних зв'язків (Z_Σ) мережі РБФ можливо розрахувати так:

$$Z_\Sigma = Z_1 + Z_2, \quad (41)$$

$$Z_1 = N \times M, \quad (42)$$

$$Z_2 = M \times K, \quad (43)$$

де Z_1 - кількість синаптичних зв'язків схованих нейронів, Z_2 - кількість синаптичних зв'язків вихідних нейронів.

Навчання РБФ проводиться поетапно. На першому етапі розраховуються кількість нейронів в СШН та коефіцієнти (центр і радіус

функції Гауса) для функцій активації схованих нейронів. Для розрахунку центру функції Гауса рекомендується використовувати метод “К-середніх” або метод навчання мережі Кохонена - “переможець забирає все”.

Метод “К-середніх” спрямований на визначення оптимальної кількості точок, які є центроїдами кластерів в навчальних даних. При К радіальних елементах вказані центри розташовуються так, щоб кожна навчальна точка відносилась до одного центру кластера і лежала до нього ближче, ніж до іншого. Необхідно, щоб кожен центр кластера був центроїдом множини навчальних точок, які до нього відносяться. Спрощеною модифікацією методу є рівномірний розподіл центрів кластерів на всій множині навчальних прикладів.

Наступним етапом навчання мережі РБФ є розрахунок радіусів функцій Гауса. Якщо радіуси будуть занадто короткими, то РБФ не зможе проводити інтерполяцію даних між відомими точками та втратить можливості узагальнення результатів. Якщо ж радіуси будуть занадто довгими, то мережа не буде сприймати дрібні деталі. Для розрахунку радіусів функції Гауса можливо використовувати метод “К найближчих сусідів”.

Після розрахунку параметрів функції Гауса, які по своїй суті представляють вагові коефіцієнти нейронів схованого шару необхідно визначити вагові коефіцієнти нейронів вихідного шару. Визначення можливо реалізувати методом “навчання з вчителем” відповідно правилу Відроу-Хофа:

$$\Delta w_j = \eta \times net_j \times \delta_j, \tag{44}$$

де Δw_j - корекція вагових коефіцієнтів j-го нейрону вихідного шару, η - норма навчання, δ_j - помилка вихідного сигналу j-го вихідного нейрону , net_j - сумарний вхідний сигнал j-го вихідного нейрону.

Помилка вихідного сигналу для j-го нейрону розраховується так:

$$\delta_j = w_j^f - w_j^o, \tag{45}$$

де w^f_j - фактичний вихід, w^o_j - очікуваний вихід j-го вихідного нейрону.

В багатьох випадках при розрахунках (44, 45) вважається, що всі зв’язки вихідних нейронів потребують однакової величини корекції вагових коефіцієнтів. Тому для РБФ з одним вихідним нейроном (44, 45) можливо переписати так:

$$\Delta w_{\mathrm{j}} = \frac{\eta \times net \times \delta}{\mathrm{M}}, \tag{46}$$

$$\delta = w^{f} - w^{\mathrm{j}}, \tag{47}$$

де M – кількість схованих нейронів, w^f - фактичний вихід, w^o - очікуваний вихід мережі, *net* - сумарний вхідний сигнал вихідного нейрону.

Використання (44-47) вказує на ітераційний процес навчання. При цьому аналітичної залежності оптимальної кількості ітерацій навчання від параметрів РБФ не знайдено. Водночас кількість ітерацій (i), кількість схованих (M), вхідних (N) та вихідних нейронів (K) безпосередньо впливають на тривалість навчання (T), яка є однією із головних характеристик НМ:

$$T \approx i \times K \times N \times M . \tag{48}$$

Після навчання необхідно перевірити якість розпізнавання РБФ на тестових прикладах, що не входять до навчальної вибірки. Якщо якість незадовільна то проводиться корегування вагових коефіцієнтів, спочатку схованого, а потім вихідного шару нейронів.

Відомі деякі модифікації мережі РБФ. Наприклад, були спроби застосування у вихідних нейронах нелінійних функцій активації та навчання вихідного шару за допомогою методу оберненого розповсюдження помилок. Проте широкого розповсюдження такі модифікації не отримали.

Відзначені деякі переваги мережі РБФ в порівнянні з БШП. По-перше, РБФ дозволяє моделювати довільну функцію за допомогою всього одного СШН, що в деякій мірі спрощує архітектуру мережеі. По-друге, навчання СШН та вихідного шару нейронів РБФ можливо проводити за допомогою достатньо апробованих методів лінійного моделювання. По цій причині РБФ навчається на порядок швидше.

Водночас мережа РБФ має і цілий ряд суттєвих недоліків. В першу чергу, це велика кількість емпіричних параметрів, що використовуються при навчанні СШН. Навіть невелика зміна цих параметрів може суттєво вплинути на результати класифікації РБФ. При цьому ні метод К-середніх, ні алгоритм Кохонена не гарантують достовірного визначення оптимальної кількості схованих нейронів, яка є однією із найважливіших характеристик мережі РБФ.

Ще одним важливим недоліком мережі РБФ є погана екстраполяція результатів за межею області відомих даних, тобто мережа якісно розпізнає тільки образи близькі до навчальних. Тому в навчальній виборці РБФ повинен бути представлений практично весь діапазон можливих вхідних даних. Крім того, повинна бути заздалегідь відома кількість схованих нейронів, яка відповідає кількості еталонних образів. Як наслідок, РБФ доцільно використовувати в випадках, коли кількість класів, приблизний вид апроксимованої функції та діапазон вхідних даних наперед відомий. Цим умовам відповідають задачі класифікації приблизно відомих образів. Вказані обмеження значно звужують сферу використання мережі РБФ для розв'язання практичних задач.

Результати порівняння обчислювальних можливостей РБФ та БШП вказують на те, що для моделювання складних функції мережа РБФ потребує дещо більшого числа нейронів. Це пояснюється тим, що в процесі апроксимації даних біля будь-якої точки в БШП задіяні всі сховані нейрони, а в РБФ задіяні тільки найближчі. Тому для РБФ кількість нейронів, необхідних для апроксимації функції з заданою точністю зростає експоненційно з ростом розмірності вхідного сигналу. Як наслідок, програмна реалізація РБФ буде проводити класифікацію довше, витрачати більше ресурсів, але навчатись швидше, ніж програмна реалізація багатошарового перспетрону. Разом з тим важливою перевагою РБФ перед БШП є проста програмна реалізація яка досягається за рахунок більш простого алгоритму навчання.

Традиційною сферою застосування мережі РБФ є проведення оперативного аналізу інформації в процесі якого питання швидкості приблизного визначення класів превалюють над задачами точності класифікації. Досить часто такий аналіз проводиться з метою приблизної оцінки архітектури БШП, що буде вже більш точно вирішувати аналогічну задачу.

Проведені числові експерименти спрямовані на верифікацію моделі РБФ та на визначення оптимальної кількості навчальних ітерацій. В першій серії експериментів було проведено апроксимацію функції $y = 0,5x + 2x^2 - x^3$.

Застосована мережа РБФ та навчальні приклади з такими параметрами:

- Радіус функції Гауса σ=0,5.
- Норма навчання η=0,1.
- Кількість схованих нейронів - 9.
- Кількість вхідних нейронів – 1.
- Кількість вихідних нейронів – 1.
- Кількість навчальних прикладів в яких $x \in [-1;1]$ - 30.
- Центри функцій Гауса розташовані в точках -0,88889; -0,66667; -0,44444; -0,22222; 0; 0,222222; 0,444444; 0,666667; 0,888889.

Розраховані показники РБФ при 1, 10, 50, 100 та 1000 навчальних ітерацій. Величини виходу РБФ та вагових коефіцієнтів зв'язків вихідного нейрону показані в табл. 2 та табл. 3.

Таблиця 2

Величина виходу РБФ для різної кількості навчальних ітерацій

№ прикладу	Кількість ітерацій						Фактичне значення
	1	10	50	100	500	1000	
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
1	-0,8365	0,7660	1,6720	1,1255	1,9626	2,0503	2,5
2	-0,9034	0,6625	1,3861	1,8632	1,5348	1,5626	1,392
3	-0,8655	0,5050	0,9393	1,4190	0,9477	0,9246	0,636
4	-0,6997	0,3526	0,4555	0,4050	0,3680	0,3194	0,184
5	-0,4083	0,2720	0,0833	-0,0140	-0,0321	-0,0752	-0,012
6	-0,0233	0,3095	-0,059	-0,0279	-0,1416	-0,1573	0
7	0,3978	0,4699	0,0656	0,0277	0,0468	0,0624	0,172
8	0,7853	0,7122	0,4065	0,5215	0,4397	0,4728	0,456

Таблиця 2 (продовження)

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	1,0756	0,9647	0,8409	1,3200	0,8875	0,9154	0,804
10	1,2282	1,1524	1,2281	1,7931	1,2454	1,2483	1,168
11	1,2349	1,2246	1,4607	1,8798	1,4227	1,3924	1,5

Таблиця 3

Величини вагових коефіцієнтів РБФ для різної кількості навчальних ітерацій

№ схованого нейрону	Кількість ітерацій					
	1	10	50	100	500	1000
1	-0,18802	0,95915	2,13254	2,43328	3,48993	4,18081
2	-0,32088	0,38270	0,85880	0,87474	0,59617	0,32934
3	-0,35721	-0,14207	-0,29341	-0,43454	-1,21845	-1,74135
4	-0,29017	-0,46679	-0,99735	-1,14324	-1,63726	-1,86594
5	-0,13523	-0,50705	-1,09894	-1,15253	-1,02151	-0,83788
6	0,07581	-0,27337	-0,66131	-0,61717	-0,07266	0,26688
7	0,30527	0,14092	0,09261	0,17153	0,63723	0,80934
8	0,51980	0,60449	0,89260	0,93172	0,92580	0,82485
9	0,69716	0,99989	1,52483	1,48519	0,93647	0,72793

Аналіз даних табл. 2 та табл. 3 вказує на те, що максимальна відносна помилка та середня відносна помилка виходу РБФ стабілізується при кількості навчальних ітерацій більше 100. Величини помилок відповідно 1,01 та 0,63. Збільшення кількості навчальних ітерацій не зменшує величин цих помилок, хоча й значно впливає на величини вагових коефіцієнтів вихідного нейрону.

Для збільшення наглядності отриманих результатів на рис. 7 показано графік функції апроксимованої РБФ, яка навчалась 1000 ітерацій та фактичний графік функції $y = 0,5x + 2x^2 - x^3$. Можна відзначити близькість модельного та фактичного графіків. Крім того, модельний графік практично не відрізняється від відомих теоретичних результатів, що є підтвердженням точності моделі. Крім того проведені експерименти з метою апроксимації РБФ лінійних та квадратичних функцій, що залежать від двох та трьох змінних. Навчання моделі проводилось в діапазоні навчальних прикладів від 10 до 100 при кількості схованих нейронів від 10 до 20. Модель показала достатню точність апроксимації лінійних функцій після 100 навчальних ітерацій, квадратичних функцій – після 1000 ітерацій. При цьому відносна

середня похибка апроксимації знаходилась в межах 1%, а відносна максимальна похибка – в межах 3%.

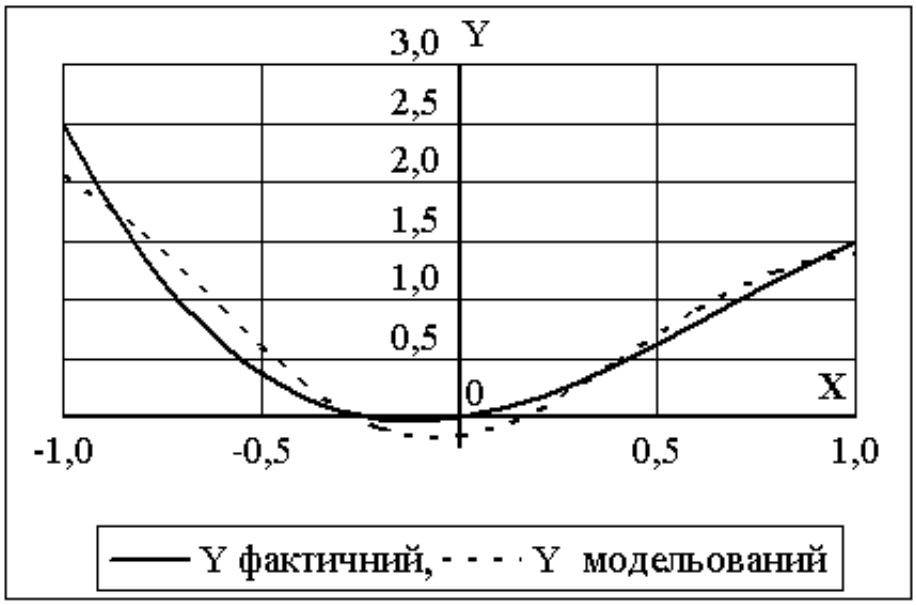


Рис. 7. Фактичний та модельований РБФ графік функції $y = 0,5x + 2x^2 - x^3$

Можна зробити висновок, що крім іншого, оптимальна кількість навчальних ітерацій безпосередньо залежить від кількості вхідних параметрів та виду модельованої функції. При цьому обчислювальна складність ітерації пропорційна кількості схованих нейронів. Тому, в складних випадках (велика кількість вхідних параметрів, складні функціональні залежності), навчання РБФ може вимагати значної кількості обчислень, що суперечить відомим теоретичним висновкам. Це вказує на необхідність експериментальної перевірки ефективності РБФ при її використанні в високовідповідальних комп'ютерних засобах.

Розділ 4. Нейронні мережі, що самонавчаються

Принципи побудови НМ, що самонавчаються розглянемо на основі моделі Ліппмана-Хеммінга. Вказану модель можливо розглянути в якості базової моделі для НМ, що самонавчаються. Відзначимо, що в якості основи можливо застосувати і інші моделі. Однак розгляд моделі Ліппмана-

Хеммінга рекомендований по причині її простоти та можливості розгляду всіх основних аспектів процесу самонавчання.

В НМ, що базуються на вказаній моделі, кожен з образів однозначно описується відповідним K-вимірним вектором з бінарними компонентами, а критерієм класифікації є відстань Хеммінга від піддослідного образу до бібліотечних образів. Класифікація полягає в пошуку бібліотечного образу, що є найближчим до піддослідного.

Відстань між образами (векторами) розраховується відповідно правилу Хеммінга, як кількість неоднакових компонент:

$$\begin{cases} \rho_k = \sum_{i=1}^{K} \rho_i(x_i^k, \xi_i), \\ \forall (x_i^k = \xi_i) \Rightarrow \rho_i = 1, \\ \forall (x_i^k \neq \xi_i) \Rightarrow \rho_i = 0. \end{cases} \quad (49)$$

де ρ_k - відстань від бібліотечного образу $x_к$ до піддослідного образу ξ, $\rho_i(x_i^k, \xi_i)$ - відстань між i-ми компонентами бібліотечного образу $x_к$ та піддослідного образу ξ, K - кількість компонент.

На рис. 8 показана структура НМ Ліппмана-Хеммінга для розпізнавання образу, що описуються двома параметрами, як одного із трьох бібліотечних класів.

Структурно модель складається із вхідного шару нейронів та шару образів. В задачу вхідних нейронів входить лише розподіл вхідної інформації між нейронами шару образів. Кількість вхідних нейронів дорівнює кількості компонент в образі, тобто кількості параметрів, що характеризують образ.

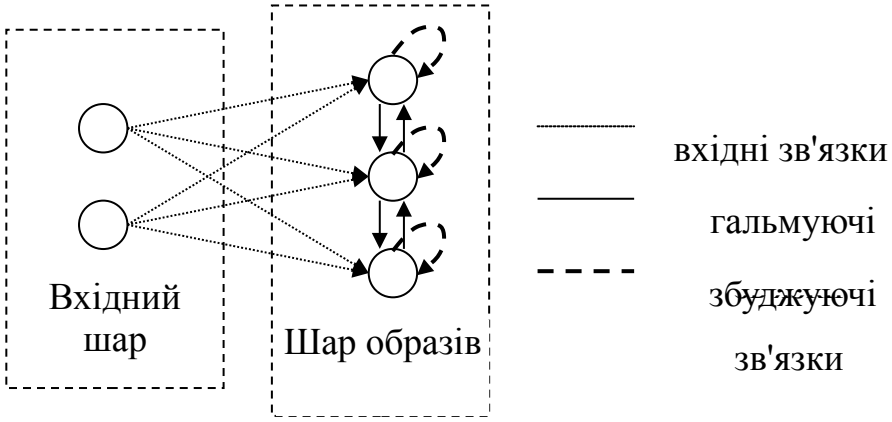


Рис. 8. Приклад структури НМ Ліппмана-Хеммінга

Кожен вхідний нейрон пов'язаний з кожним нейроном шару образів, кількість яких дорівнює кількості кластерів. Нейронам шару образів відповідає функція порогова

активації виду:

$$F(NET) = \begin{cases} NET, \exists NET > 0, \\ 0, \exists NET \le 0 \end{cases}, \quad (50)$$

де *NET* - вхід нейрону, що в загальному випадку розраховується за формулою (1.3).

Процес навчання НМ Ліппмана-Хемінга полягає в тому, що вагові коефіцієнти нейронів шару образів встановлюються рівними нормованим компонентам бібліотечних образів:

$$w_n^m = \frac{x_n^m}{\sum_{i=1}^{K} x_i^m}, \quad (51)$$

де w_n^m - ваговий коефіцієнт *n*-го входу для *m*-го нейрону шару образів, x_n^m - *n*-та компонента для *m*-го нейрону (образу), *K* - кількість компонент.

Компоненти невідомого образу також нормуються:

$$s_n = \frac{\xi_n}{\sum_{i=1}^{K} \xi_n}, \quad (52)$$

де s_n - нормована n-та компонента вхідного вектору ξ.

Подача невідомого образу призводить до того, що нейрони шару образів приймають початкові рівні активації:

$$y^m(0) = f\left(0{,}5\sum_{n=1}^{M}\left(s_n w_n^m\right)\right), \quad (53)$$

де $y^m(0)$ - початковий рівень активації *m*-го нейрону, *M* - кількість бібліотечних образів, *f* - функція активації нейрону шару образів виду (50).

Після цього відбувається ітераційний процес вибору нейрону, який є найближчим до невідомого образу. Вибір реалізується за рахунок того, що по гальмуючим зв'язкам кожен з нейронів отримує негативне збудження від всіх інших нейронів. Величина негативного збудження від будь-якого нейрону пропорційна величині активації цього нейрону. Водночас кожен з нейронів отримує позитивне збудження від самого себе.

Розрахунок рівня активації m-го нейрону шару образів на *t*-ій ітерації можливо здійснити так:

(54

$$OUT^m(t) = f\left(OUT^m(t-1) - \frac{\sum_{n=1, n\neq m}^{M} OUT^n(t)}{M+1}\right). \qquad)$$

Після деякої кількості ітерацій зостається єдиний активний нейрон-переможець, що вказує на клас до якого належить піддослідний образ. Такий механізм вибору нейрону, отримав назву “переможець забирає все” (Winner Take All - WTA). При цьому базовим методом навчання нейрону-переможця є метод Хебба.

НМ Ліппмана-Хеммінга є моделлю з прямою структурою пам'яті. Інформація, що міститься з бібліотечних образах не узагальнюється а безпосередньо запам'ятовується в синоптичних зв'язках. За рахунок цього навчання НМ такого типу відбувається набагато швидше ніж навчання БШП. Принциповим недоліком розглянутого варіанту НМ є жорстко апріорно задана кількість кластерів, що відповідає кількості нейронів в шарі образів. В багатьох практичних задачах апріорно визначити кількість кластерів не можливо. Тому, було б краще, якби НМ сама розраховувала кількість кластерів, виходячи із реальних навчальних даних. Для вирішення цієї проблеми застосовують модифіковані алгоритми, наприклад “зростаючий нейронний газ”. Зміст цих алгоритмів полягає в послідовному збільшенні кількості кластерів до тих пір доки помилка розпізнавання не досягне наперед заданої величини. В класичному вигляді НМ Ліппмана-Хеммінга практично не застосовується, але вона є базовим компонентом інших архітектур НМ, які можуть знайти своє застосування в практичних задачах, наприклад, НМ Кохонена. Необхідно зазначити, що НМ, в основу яких покладена модель Ліппмана-Хеммінга багато в чому еквівалентні добре вивченим методам статистичного аналізу головних компонент. Однак НМ можливо використовувати вже під час збору статистичної інформації. Тобто межі кластерів можливо оперативно адаптувати до зміни вхідного потоку інформації. Крім того, така НМ має переваги в при нелінійнму характері задачі, коли явні рішення знайти неможливо.

Як і мережа Ліппмана-Хеммінга, НМ Кохонена складається із двох шарів нейронів, вхідного і вихідного (топографічного). Кількість нейронів

вхідного шару дорівнює кількості компонент вхідних образів. Кожен вхідний нейрон пов'язаний з кожним топографічним нейроном, який відповідає певному класу образів.

На відміну від моделі Ліппмана-Хеммінга, де кожен клас характеризується заданими зовні параметрами одного бібліотечного образу, в SOM кожному класу можуть відповідати декілька образів. SOM самостійно розділяє бібліотечні образи на класи, для чого самостійно розраховує вагові коефіцієнти зв'язків топографічних нейронів. Кількість класів є параметром зовнішнім, по відношенню до НМ і визначається точністю з якою необхідно виконати кластеризацію набору бібліотечних образів.

Навчання SOM відбувається методом “без вчителя”, за допомогою механізму “переможець забирає все”. На противагу моделі Ліппмана-Хеммінга де положення нейрона-переможця в шарі образів не мало нічого спільного з координатами його вагових коефіцієнтів у вхідному просторі, в SOM близьким нейронам топографічного шару відповідають близькі вхідні образи. Це дозволяє будувати так звані топографічні карти Кохонена, які широко застосовуються для візуалізації багатовимірних даних. Для виявлення кореляції між топографічними нейронами SOM має деякі особливості в структурі і в алгоритмі навчання. Особливістю структури є те, що зв'язки між нейронами топографічного шару складають не повноз'язну структуру, а побудовані по певним правилам. Лінійна, квадратна та гексагональна сітки зв'язків між нейронами топографічного шару показані на рис. 9-11.

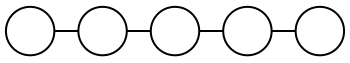


Рис. 9. Лінійне розміщення нейронів в топографічному шарі

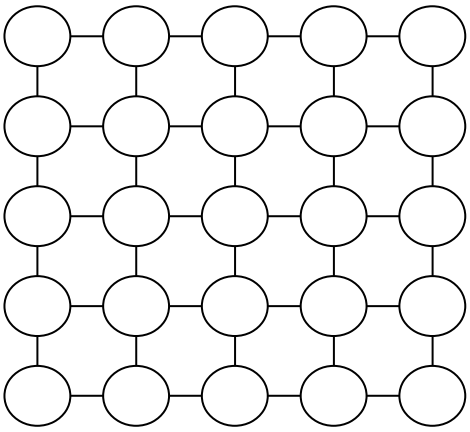


Рис. 10. Квадратна сітка зв'язків

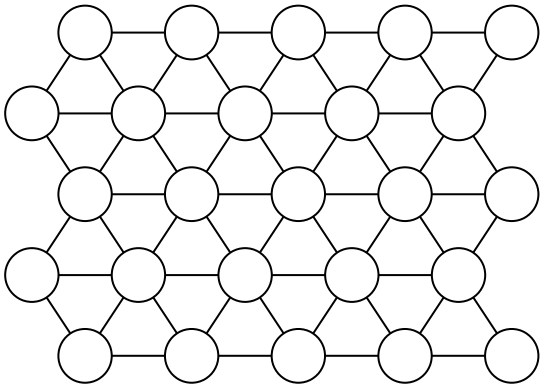


Рис. 11. Гексагональна сітка зв'язків

Навчання мережі Кохонена відбувається методом послідовних наближень. Починається навчання з призначення матриці вагових коефіцієнтів випадкових значень. Після цього на вхід мережі послідовно подаються вектори, що відповідають навчальним образам. Для кожного з векторів розраховується відстань від нього до кластерного елементу:

$$d_m^j = \left(\sum_{i=1}^{N} \left(w_i^m(t) - x_i \right)^2 \right)^{0,5}, w_i^m \in W^m, \quad (55)$$

де d_m^j - відстань від *j*-го вхідного вектору до *m*-го нейрону, t - номер кроку навчання, N - кількість компонент вхідного вектору, $w_i^m(t)$ - вага зв'язку між *i*-м входом та *m*-м нейроном на *t*-му кроці навчання, W^m - матриця вагових коефіцієнтів *m*-го нейрону, x_i - *i*-а компонента вхідного вектору.

Знаходиться нейрон для якого ця відстань мінімальна, тобто нейрон-переможець. Після цього відповідно (56, 57) змінюються вагові коефіцієнти нейрону-переможця та сусідніх з ним нейронів:

$$w_i^k(t+1) = w_i^k(t) + \eta(t) \times (x_i - w_i^k(t)), \tag{56}$$

$$\begin{cases} w_i^n(t+1) = w_i^n(t) + \eta(t,n) \times (x_i - w_i^n(t)), \\ n \in \{k - r(t), \ldots k + r(t)\} \wedge n \in \{m\}. \end{cases} \tag{57}$$

де k - номер нейрону-переможця, η - коефіцієнт (норма) швидкості навчання, n - номер сусіднього нейрону, r - радіус навчання, $\{m\}$ - множина нейронів в топографічному шарі.

Відзначимо, що сусідніми вважаються нейрони, які мають між собою зв'язки, що входять в круг з центром, що відповідає нейрону-переможцю та заданим радіусом навчання. Саме в розрахунку сусідніх нейронів виявляється різниця між різними структурами зв'язків в топологічному шарі. Наприклад при радіусі 1 для лінійної структури зв'язків будуть змінюватись вагові коефіцієнти нейрону-переможця, та двох найближчих до нього нейронів. Для квадратної структури крім нейрону-переможця змінюються вагові коефіцієнти чотирьох найближчих нейронів, а для гексагональної структури - шести найближчих нейронів. Радіус навчання поступово зменшується на кожній новій епосі навчання, на протязі якої мережі в випадковому порядку подаються всі навчальні вектори. На початку навчання радіус може охоплювати досить велику кількість нейроні, можливо навіть всю топографічну карту. В кінці навчання радіус прирівнюється нулю, тобто навчається тільки нейрон-переможець. За рахунок цього на кожній наступній епосі корекція вагових коефіцієнтів стає все більш тонкою. Швидкість навчання також зменшується після кожної епохи. Тобто, на перших епохах навчання НМ формує грубу топографічну структуру на якій схожі навчальні образи активують групи нейронів, що розміщені неподалік на топографічній карті. З кожною епохою швидкість і радіус навчання зменшуються, що призводить до точної настройки топографічної карти. В [17, 24, 27, 29, 68] відзначено, що для навчання НМ Кохонена потребує обсяг навчальної вибірки, як мінімум в 5-10 разів більшої кількості вхідних параметрів. При цьому в навчальних даних можуть бути помилки (шум), якщо вони не носять

систематичний характер.

Класична НМ Кохонена не використовує явного критерію якості навчання. В сучасних модифікаціях НМ таким критерієм є мінімізація сумарної середньої відстані (Θ) від кожного навчального образу до найближчого вузлу карти:

$$\Theta \to min. \tag{58}$$

Розрахунок сумарної середньої відстані може бути проведений так:

$$\Theta = \frac{1}{J} \sum_{j=1}^{J} \sqrt{\left(\sum_{i=1}^{N} x_i^j - y(x_i^j) \right)^2}, \tag{59}$$

де J - кількість навчальних образів, що відповідають даному кластеру, N - кількість компонент вхідного вектору, x_i^j - і-а компонента j-го навчального образу, що відповідають даному кластеру, $y(x_i^j)$ - відстань по осі i від j-го навчального образу до відповідного кластеру.

Однак застосуванню вказаного критерію заважає те, що теоретично не обґрунтовані умови закінчення навчання мережі Кохонена навіть в точці локального мінімуму функції (58).

Часто навчання спеціально розділяють на дві фази. Перша коротка фаза характеризується великою швидкістю та радіусом навчання. Друга фаза довга, характеризується малою швидкістю навчання та близьким до нуля радіусом. Рекомендована тривалість першої фази навчання становить біля 1000 епох, а тривалість другої фази 10000-100000 епох. Разом з цим необхідно враховувати, що кількість навчальних епох має бути як мінімум в 10 разів більша ніж кількість навчальних прикладів. В деяких джерелах пропонується використовувати в алгоритмі навчання параметр “совісті”, який запобігає частому вибору одного і того ж нейрона-переможця.

В ідеальному варіанті зупинка навчання відбувається тоді, коли навчальні вектори не змінюють своїх кластерів при переході від однієї епохи до іншої. Інколи, в складних випадках, навчання закінчують після певної кількості епох. В результаті положення кожного центу кластеру встановлюється в деякій позиції, яка найкраще відповідає тим навчальним прикладам, для яких нейрон є переможцем. При цьому мережа організуються

таким чином, що нейрони, які відповідають образам, розміщеним близько в просторі входів будуть розміщені близько один від другого і на топографічній карті.

Якщо в навчальному наборі даних були характерні точки з призначеними їм маркерами, тоді відповідні вузли карти також можуть бути маркірованими. Крім того, мережа вузлів може бути різнокольоровою, відповідно деякої ознаки, що дозволяє будувати трьохвимірні карти. Популярним способом показу на картах самих даних є застосування діаграм Хінтона, які передбачають зображення на кожному вузлі карти квадрату, розмір якого пропорційний кількості навчальних образів, найближчих до цього вузла.

По своїй суті навчання мережі Кохонена є кластерзація невідомих даних, тобто зменшення різнорідності даних. Вказаний факт є передумовою для використання НМ даного типу для розвідувального аналізу даних. Запропоновано цілий ряд модифікованих алгоритмів навчання SOM. Модифікації полягають в застосуванні двох технологічних прийомів:

- Зміни напрямку руху вузлів. Крім руху в сторону точки вхідного образу нейрони пересуваються ще й в іншому напрямку.
- Специфічному розрахунку радіусу визначення нейронів, сусідніх з нейроном-переможцем.

Метою більшості модифікацій є прискорення навчання, підвищення точності апроксимації при фіксованій кількості вузлів, динамічне визначення кількості вузлів, покращення гладкості та регулярності топографічної сітки. Крім того, запропоновано доповнити мережу додатковим шаром нейронів - зіркою Гроссберга, навчання якої відбувається методом “з вчителем”. Такі НМ дістали назву - мережі зустрічного розповсюдження. За рахунок використання зірки Гроссберга НМ отримала деякі додаткові можливості, наприклад проводити калібровку отриманих результатів, відображати результати функціонування шару Кохонена в вихідні образи, реалізовувати асоціативний пошук. Проте розвиток інших типів НМ нівелювало отримані переваги і на сьогодні описана модифікація самостійно не використовується.

Досить перспективною модифікацією карти Кохонена є НМ, що дістала назву ПК, яка передбачає використання обґрунтованого критерію оптимізації архітектури мережі. Застосування цієї технології передбачає розгляд процесу створення НМ, що самонавчається як вирішення задачі оптимізації заданого функціоналу від взаємного розміщення вузлів карти і навчальних даних.

Висуваються три основні вимоги до критерію оптимізації:

- залежність від положень вузлів мережі в просторі навчальних даних;
- квадратичність по відношенню до положень вузлів мережі;
- процес мінімізації критерію повинен призводити до самоорганізації вузлів карти.

Передумовою першої вимоги є можливість оптимізації відносно невеликої кількості параметрів:

$$\omega = l \times N, \tag{60}$$

де ϖ - кількість оптимізуємих параметрів, l - кількість вузлів мережі, N - кількість точок навчальної вибірки.

В випадку прямокутної мережі кількість вузлів розраховується як:

$$l = p \times q, \tag{61}$$

де p та q - число вузлів мережі по горизонталі та вертикалі.

Передумовою другої вимоги є можливість знаходження мінімуму критерію оптимізації в результаті вирішення системи лінійних рівнянь. Для того, щоб критерій задовольняв першим двом вимогам до його складу повинна входити середня евклідова відстань від точки даних до найближчого вузла пружної карти.

Щоб задовольнити третю вимогу в критерій повинні бути добавлені складові, що відповідають сумарним лінійним та кутовим деформаціям мережі. Використання цих складових зумовлено аналогією між НМ та пружною пластиною, що при лінійних та кутових деформаціях намагається зберегти свою початкову форму.

В остаточному вигляді критерій оптимізації представляє собою суму середнього квадрату відстані до вузла мережі та коефіцієнтів пружності з відповідними ваговими коефіцієнтами мережі. Завдяки такому критерію

вузли НМ з однієї сторони будуть притягуватись до точок даних, а з іншої намагатимуться мінімізувати своє розтягнення та прийняти максимально гладку форму (стати більш регулярними). При цьому розрахунок значень коефіцієнтів пружності карти потребує додаткового розгляду.

Чим більш пружна карта, тим більш гладку модель даних вона представляє, але і тим гірше описує невеликі відхилення даних. Менш пружна карта точніше описує дані, але погано відділяє випадковий шум, що негативно впливає на узагальнення інформації. Тому для настройки коефіцієнтів пружності запропонована процедура їх послідовного зменшення від максимуму до величин, що задовольняють вимогам точності і узагальнення конкретної задачі.

Розглянемо алгоритм побудови прямокутної пружної сітки. Нехай p кількість вузлів сітки по горизонталі, q кількість вузлів по вертикалі. Пронумеруємо вузли мережі за допомогою індексів - $y_{i,j}$, $i=1..p$, $j=1..q$. Розділимо всю множину вхідних даних X на $p \times q$ підмножин (таксонів) $K_{i,j}(i=1..p, j=1..q)$, в межах кожної із яких точки знаходяться ближче до вузла мережі $y_{i,j}$, ніж до будь-якого іншого вузла:

$$K_{i,j} = \left\{ x \in X, \forall \left| y^{i,j} - x \right|^2 \to \min_{i,j} \right\}. \quad (62)$$

При формуванні (62) в якості міри близькості вузлів мережі до даних використано величину середнього квадрату відстані від точки до найближчого вузла сітки. Кожен вузол (крім граничних) має чотирьох сусідів, з кожним із яких він з'єднаний відповідним ребром мережі. Чим більша середня довжина вузла, тим сильніше мережа розтягнута, що зумовлює необхідність мінімізації цієї величини. Відповідно, в мінімізуємий функціонал повинні ввійти різниці між положеннями сусідніх вузлів.

Ступінь згину можливо визначити за допомогою точечної оцінки величини другої похідної. В результаті критерій оптимізації можливо записати так:

$$D = \frac{D_1}{N} + \lambda \times \frac{D_2}{p \times q} + \mu \times \frac{D_3}{p \times q} \to \min \ , \quad (63)$$

де N - кількість точок навчальної вибірки X, λ та μ коефіцієнти пружності, що відповідають за лінійну та кутову деформацію мережі, D_1 -

міра близькості розміщення вузлів мережі до даних, D_2 - міра розтягнутості мережі, D_3 - міра кривизни мережі.

Розрахунок складових D_1, D_2, D_3 можливо здійснити так:

$$D_1 = \sum_{i,j} \sum_{X_k \in K_{i,j}} \left\| X_k - y^{i,j} \right\|^2, \tag{64}$$

$$D_2 = \sum_{i=1}^{p} \sum_{j=1}^{q-1} \left\| y^{i,j} - y^{i,j+1} \right\|^2 + \sum_{i=1}^{p-1} \sum_{j=1}^{q} \left\| y^{i,j} - y^{i+1,j} \right\|^2, \tag{65}$$

$$D_3 = \sum_{i=1}^{p} \sum_{j=1}^{q-1} \left\| 2y^{i,j} - y^{i,j-1} - y^{i,i+1} \right\|^2 + \sum_{i=1}^{p-1} \sum_{j=1}^{q} \left\| 2y^{i,j} - y^{i-1,j} - y^{i+1,j} \right\|^2, \tag{66}$$

де $K_{i,j}$ - підмножина точок із множини X, для яких вузол мережі є найближчий (таксони).

Межі додавання в (65, 66) вибрані таким чином, щоб в функціоналах D_2 та D_3 ребро входило в суму тільки один раз. В теоретичних роботах відзначено, що функціонал D є квадратичним по положенню вузлів $y^{i,j}$. Це дозволяє при заданому розділу множини точок на таксони провести його мінімізацію шляхом вирішення системи рівнянь розміром $pq \times pq$. Таким чином, для розрахунку мінімального значення функціоналу D необхідно:

1. Розмістити вузли мережі довільним чином.
2. При заданих положеннях вузлів мережі провести розділ множини даних на таксони - $K_{i,j}$.
3. При заданому розділі множин точок н таксони провести мінімізацію функціоналу D.
4. Етапи 1 та 2 слід повторювати доти, доки функціонал D не перестане змінюватись в межах заданої точності.

Процес мінімізації сходиться по причині того, що на кожному етапі величина D буде зменшуватись, при цьому вона обмежена знизу 0. Оскільки число варіантів розділу точок на таксони обмежене, хоча і може бути достатньо великим, то і процес мінімізації сходиться за обмежену кількість етапів. Крім того, запропоновані методики адаптації розміру та структури пружної карти до нових навчальних даних, що з'являються в процесі її експлуатації.

В наслідок того, що методика побудови ПК відрізняється від SOM розміщення вузлів цих НМ також будуть дещо відрізнятися. Так в ПК

кількість вузлів, розміщених в областях згустків даних не завжди буде пропорційна потужності згустків, що характерно для карти Кохонена. Так деякі точки даних можуть бути розміщені між вузлами, на значній відстані від них. Відзначено, що процедура проектування даних в найближчий вузол може давати похибку більшу ніж у карти Кохонена. Разом з тим для ПК характерним недоліком є наявність крайових ефектів, тобто хибного групування даних в периферійних областях.

Методом усунення цих недоліків може бути використання кусочно-лінійних методів проектування даних в найближчу точку карти, або використання декількох НМ, що доповнюють одна одну.

Перша НМ буде картографувати та апроксимувати самі дані, друга - відмінності від першої моделі, тобто вектори, що починаються в точках даних та закінчуються в точках відповідних проекцій на першу карту. Ці відмінності можуть мати дві складові - випадковий шум та помилки першої моделі. Для третьої карти можуть бути використані відмінності від проекцій перших відмінностей. При цьому, якщо більшість навчальних точок будуть адекватно описані за допомогою першої моделі, то більша частина даних буде розміщена біля нуля в просторі координат відмінностей. Якщо ж домінуючим фактором відмінностей є випадковий шум, то закон розподілу відмінностей буде близьким до нормального. Тому використання декількох карт дозволяє досягти високої точності моделювання не втрачаючи при цьому узагальнюючих властивостей.

Після закінчення навчання на вхід мережі можна подавати нові образи для розпізнавання. При цьому можливо застосовувати так званий поріг доступу, який дорівнює максимальному рівню активації нейрона-переможця. Якщо рівень активації нейрона-переможця для класифікуємого образу нижчий ніж вказаний поріг, то класифікація проведена успішно. В випадку рівня активації вищого від порогу доступу, вважається, що мережа не прийняла рішення про класифікацію. Це може відбутись коли класифікуємий образ значно відрізняється від навчальної вибірки.

Таким чином, крім розвідувального аналізу даних, мережа Кохонена може використовуватись як детектор нових явищ. Цінність мережі Кохонена

підтверджують досить чисельні приклади її застосування в програмних додатках, що використовуються при розв'язанні економічних та лінгвістичних задач.

Серія експериментів з картою Кохонена, спрямованих на її верифікацію, полягала в кластеризації десяти абстрактних фігур, еталони яких показані на рис. 12.

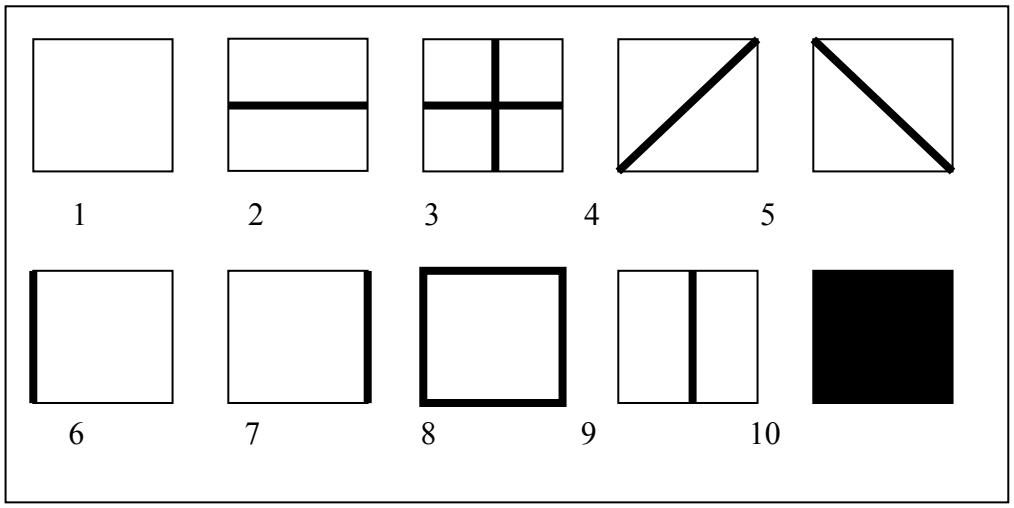


Рис. 12. Еталони абстрактних фігур навчальної вибірки

Для навчання використовувались як зашумлені, так і еталоні фігури. Кожна з фігур була вписана в прямокутник розміром 5×4 одиниці, а тому характеризувалась набором із 20 дискретних бінарних (0 або 1) параметрів. З точки зору комп'ютерної діагностики, дані фігури можливо інтерпретувати, наприклад, як певні образи, що діагностуються за допомогою 20 дискретних параметрів.

Відзначимо, що реалізовану мережу Кохонена не відразу вдалось навчити проводити правильну кластеризацію. Так, при рекомендованих значеннях параметрів навчання мережа розділяла навчальну вибірку на 5 кластерів:

$$\begin{cases} \{\Phi(1)\} \in K(1), \\ \{\Phi(10)\} \in K(2), \\ \{\Phi(2),\Phi(3)\} \in K(3), \\ \{\Phi(4),\Phi(5)\} \in K(4), \\ \{\Phi(6),\Phi(7),\Phi(8),\Phi(9)\} \in K(5). \end{cases} \quad (67)$$

де $\{\Phi(i)\}$ - множина фігур типу $i \in [1..10]$, $K(j)$ - j-ий кластер, $j \in [1..10]$

При цьому кластеризація відбувалась максимум за 10 епох навчання, після чого вагові коефіцієнти уточнювались без переходу образів між кластерами. Якісної кластеризації вдалось досягти завдяки:

- Емпіричному початковому розподілу вагових коефіцієнтів близькому до значень параметрів, що характеризують еталонні образи.
- Використання функціоналу норми навчання виду $\eta = f(r,t)$, де η - коефіцієнт (норма) швидкості навчання, t - номер епохи, r - радіус навчання.

Відзначимо, що за допомогою карти Кохонена, реалізованої в промисловому пакеті DataMining Deductor, так і не вдалось провести правильну кластеризацію фігур. Основною причиною невдачі є неможливість настройки в Deductor більшості стандартних параметрів навчання. Результати експериментів підтвердили теоретичні висновки, що значним недоліком НМ типу карта Кохонена є велика кількість емпіричних параметрів навчання - кількість кластерів, максимальна та мінімальна величина норми навчання, функціонал зміни норми навчання, максимальна та мінімальна величина радіусу навчання, функціонал радіусу навчання, сигнал зупинки навчання.

Разом з цим проявляється висока залежність якості розпізнавання від початкового розподілу вагових коефіцієнтів. У випадках, коли початковий розподіл вагових коефіцієнтів значно відрізняється від розподілу ознак навчальної вибірки, якість кластеризації виявляється незадовільною. Найпростіший шлях вирішення цієї проблеми полягає у випадковому або рівномірному початковому розподілі вагових коефіцієнтів, що відповідає очікуваному випадковому або рівномірному розміщені кластеризуємих образів в просторі ознак. Але через можливий суб'єктивний, а значить і не рівномірний і не випадковий розподіл образів, що повинні бути розпізнанні в засобах комп’ютерної діагностики, даний прийом не ефективний.

Крім того, для отримання початкового розподілу вагових коефіцієнтів відповідного до розподілу вхідних векторів використовуються такі методи:

- випуклої комбінації,
- зашумлення вхідних векторів,
- присвоєння нейронам-переможцям “відчуття справедливості”.

Суть найбільш апробованого методу випуклої комбінації зводиться до того, що нормовані компоненти кожного вхідного вектору піддаються перетворенню виду:

$$\bar{x}_i = \beta(t) \times \bar{x}_i + \frac{1 - \beta(t)}{\sqrt{N_1}}, \quad (68)$$

$$\bar{x}_i = \frac{x_i \times N_1}{\sqrt{\sum_{j=1}^{N_1} x_j^2}}, \quad (69)$$

де $\beta(t)$ - коефіцієнт, що змінюється в процесі навчання від 0 до 1, $\bar{x}_i$ - і-а нормована компонента вхідного вектору, N_1 - розмірність вхідного вектору, x_i - i-а компонента вхідного вектору.

В результаті на початку навчання на вхід мережі подаються практично однакові образи. З часом перетворені вхідні образи наближаються до істинних. Завдяки цьому в деяких випадках вдається розділити близькі вхідні образи між різними нейронами.

Відзначимо, що метод можливо використовувати тільки для компонент, що характеризуються неперервними числовими величинами. При цьому реалізація всіх вказаних методів супроводжується використанням емпіричних параметрів та значним ускладненням програмної реалізації НМ. Можливим шляхом усунення перерахованих недоліків є побудова декількох десятків карт Кохонена, що мають однакову структуру, але навчаються з різними параметрами. Із побудованих вибирається та карта, що найбільш повно відповідає критерію якості.

Оцінку обчислювальної складності навчання карти Кохонена та ПК, можливо провести за допомогою методики, використаної в розділі що присвячений БШП. При цьому врахуємо особливості архітектури карт.

Кількість операцій (ξ) необхідних для навчання обох типів карт:

$$\xi \sim L_w \times P = N_1 \times N_0 \times P, \quad (70)$$

де L_w - кількість синаптичних зв'язків, N_0 - розмірність вхідного сигналу, N_1 - розмірність вихідного сигналу, P - кількість навчальних прикладів.

Коефіцієнт стиснення інформації (Ω):

$$\Omega = N_1 \times b \Big/ \ln N_0, \quad (71)$$

де b - розрядність даних (кількість біт), яка визначає можливу різноварінтність приймаємих ними значень.

Залежність кількості операцій від коефіцієнту стиснення:

$$\xi \sim N_1 \times P \times 2^{bN_0/\Omega}. \quad (72)$$

Порівняння обчислювальних можливостей мереж, що самонавчаються з БШП дозволяє вказує на значно більший термін навчання останніх. Про те узагальнюючі можливості БШП, а також кількість образів, що можуть ними запам'ятовуватись набагато вищі.

Традиційними галузями застосування мереж Кохонена та ПК є візуалізація даних, групування об'єктів, оцінка значимості ознак, відбір найбільш інформативних ознак, відновлення пропущених значень в даних, прогнозування значень окремих ознак, побудова регресивних залежностей.

Візуалізація даних дозволяє в автоматизованому режимі наглядно аналізувати розподіл самих даних та похибок їх опису. Групування об'єктів можливо проводити як в автоматичному, так і в автоматизованому режимах. В другому випадку групування виконується оператором візуально за допомогою оцінки компактності і форми згустків даних. Крім того, можливо використовувати кольорове виділення точок. Точки, що розміщенні близько в просторі ознак будуть мати схожий колір.

Формалізація цих же процедур дозволяє проводити групування в автоматичному режимі. Оцінка значимості ознак дозволяє виявити в наборі даних ознаки, що мають між собою значні кореляції та ознаки, що не мають інформаційної цінності, тобто шум.

Значимість ознаки можливо оцінити за допомогою величини зміни помилки навчання НМ при заміні i-ої ознаки на деяку константу:

$$\chi^i = \frac{1}{N}\sum_{j=1}^{N}\left(\varepsilon_j^i - \varepsilon_j\right)^{0,5}, \quad (73)$$

де N - кількість об'єктів в навчальних даних, i – номер ознаки, що змінюється на константу, ε_j - початкова помилка, ε^i_j - помилка після заміни.

Значимість групи ознак можливо оцінити так:

$$\chi^{|K|} = \frac{1}{K}\sum_{i=1}^{K}\chi^i, \quad (74)$$

де $\{K\}$ - множина ознак, що оцінюються, K - кількість ознак в множині.

Очевидно, що в більшості випадків малоінформативні ознаки використовувати в практичних задачах недоцільно. З ознаками, що мають кореляційні зв'язки ситуація дещо інша. Якщо замінити одну або декілька із взаємокорельованих ознак на фіксовану величину, то значимість інших ознак може різко збільшитись.

Відбір найбільш інформативних ознак можливо реалізувати за таким алгоритмом:

1. Провести оцінку всієї множини ознак.
2. Виявити найменш інформативну ознаку - *min*.
3. Замінити ознаку *min* на деяку константу.
4. Розрахувати загальна помилку розпізнавання НМ.
5. Повторювати п.1-4 доти, доки загальна помилка знаходиться в допустимих межах, або кількість ознак не досягне заданої величини.

Відновлення пропущених значень в даних базується на тому, що при наявності R побудованих НМ кожному вектору X з області даних відповідає множина модельних векторів Y:

$$\{Y\} = \{Y_1, Y_2 \ldots Y_R\}. \qquad (75)$$

Навіть якщо в деякому векторові з простору даних є невідомі компоненти, то їх можливо визначити за допомогою відповідного модельного вектора. Крім того, модельні вектори можливо використовувати для прогнозування окремих ознак та вирішення задачі регресії даних.

Відома методика використання карти Кохонена для розв'язання оптимізаційних задач типу класичної задачі комівояжера. В цій задачі необхідно знайти замкнений маршрут по якому комівояжер повинен так об'їхати M міст, щоб довжина маршруту була мінімальною. Будь-яке місто можна відвідати тільки один раз. При цьому кожне місто характеризується двома координатами x та y.

Передумовою використання є твердження, що після навчання карти Кохонена розміщення топографічних нейронів відповідають оптимальному маршруту комівояжера.

Застосовується карта Кохонена, яка містить два шари нейронів. Вхідний шар складається із трьох нейронів. Кількість нейронів у вихідному (топографічному) шарі дорівнює кількості міст - *M*. Кожен нейрон вхідного шару пов'язаний з кожним нейроном топографічного шару.

Топографічний шар нейронів вважається закільцьованим, тобто перший та *M*-ий нейрони цього шару з'єднані між собою. Вхідний вектор, що відповідає місту складається із трьох компонентів. Дві компоненти відповідають координатам *x* та *y*. Третя компонента представляє нормуючий параметр, який розраховується так, щоб всі вхідні вектори мали однакову евклідову довжину і не були колінеарні.

Розрахунок вихідного сигналу *y* для *j*-го вихідного нейрону при подачі на вхід мережі *i*-го вхідного образу розраховується як скалярний добуток вхідного вектору та вектору вагових коефіцієнтів зв'язків:

$$y^{(j)} = x^{(i)} \times w^{(j)} , \qquad (76)$$

де $x^{(i)}$ - вхідний вектор, що відповідає *i*-му місту, $w^{(i)}$ - вектор вагових коефіцієнтів зв'язків *j*-го нейрону.

Вихідний нейрон, для якого значення *y* є максимальним при подачі і-го вхідного вектора називається образом *i*-го міста.

Навчання карти Кохонена по своїй суті є формуванням оптимального маршруту відвідування міст. На підготовчій стадії (нульові ітерації) навчання емпірично вибираються значення коефіцієнта швидкості навчання η та радіусу навчання r, а всі зв'язки між нейронами ініціюються випадковими величинами. Після цього відбувається наступний ітераційний процес:

1. Випадковим чином вибирається місто *x*. На вхід мережі подається відповідний вхідний образ.

2. За допомогою (76) розраховуються виходи всіх нейронів топографічного шару.

3. Визначається номер нейрону, який відповідає образу міста - i_x. Для цього розраховується вихідний нейрон з максимальним виходом.

4. Відповідно (77) модифікується вектор вагових коефіцієнтів $w^{(j)}$ нейрону образу міста та нейронів, що знаходяться в межах радіусу *r*.

$$w^{(j)}(t) = \frac{w^{(j)}(t-1) + x^{(i)} \times \eta(t-1)}{\left\| w^{(j)}(t-1) + x^{(i)} \times \eta(t-1) \right\|}, \quad (77)$$

де t - номер ітерації, $\|x\|$ - евклідова норма вектору x.

Відзначимо, що на протязі навчання радіус r поступово зменшується до певної величини. Після цього r зостається постійним. Наприклад, можливо спочатку встановити $r=0,1 \times M$. На протязі перших 10% ітерацій зменшити його значення до 1, після чого рідіус не змінювати.

5. Параметр η зменшується по наперед визначеній залежності:

$$\eta(t) = \eta(t-1) - f(t), \quad (78)$$

де $f(t)$ - деяка функція.

Вважається, що вид функції f мало впливає на якість навчання, тому в багатьох випадках $f(t) = \Delta = const$. Кількість ітерацій в описаній методиці визначається апріорно за допомогою параметру Δ. При цьому η та Δ повинні співвідноситись так, щоб кількість ітерацій навчання була більшою від M^2.

6. Якщо $\eta(t) > 0$, то відбувається перехід на наступну ітерацію. В протилежному випадку процес навчання закінчується.

Після завершення навчання, положення міста в маршруті визначається положенням його образу в топографічному шарі. Якщо декільком різним містам відповідає один образ, то вважається, що порядок відвідин цих міст не має значення.

Доведена ефективність використання карт Кохонена в оптимізаційних задачах дозволяє зробити висновок про перспективність їх використання при керуванні процесами в комп'ютерній системі. Запропоновано ряд модифікацій мережі Кохонена, пристосованих для вирішення оптимізаційних задач. Наприклад, відома мережа SOM з адаптивною структурою. Її перевагами відносно мережі Кохонена є більш висока теоретична обчислювальна ефективність алгоритму пошуку оптимального маршруту. Проте для таких модифікацій характерна велика кількість емпіричних параметрів навчання та складність програмної реалізації. Дані фактори зменшують надійність пошуку оптимуму та зменшують реальний термін оптимізаційних розрахунків, що значно ускладнює процес використання мереж SOM з адаптивною структурою в задачах керування комп'ютерними

системами.

Розділ 5. Ймовірнісні нейронні мережі

Функціонування ймовірнісних НМ базується на передумові, що вирішення задач класифікації та регресії можливе завдяки оцінці щільності ймовірності сумісного розподілу вхідних та вихідних даних. В задачах класифікації виходи НМ інтерпретуються як оцінки ймовірності того, що образ належить деякому класу.

Для вирішення таких задач НМ повинна оцінити щільність ймовірності віднесення образу до кожного із класів, порівняти ці ймовірності між собою та вибрати найбільш ймовірний клас. В задачах регресії виходи НМ розглядаються як очікуване найбільш ймовірне значення моделі у вказаній точці можливого ймовірного простору входів. При розв'язані обох типів задач розрахунок щільності ймовірності відбувається за допомогою методу ядерних оцінок.

Ідея методу. Якщо наблюдение знаходиться в певній точці простору ознак класів то це свідчить про те, що в даній точці простору є деяка не нульова щільність ймовірності. Причому, поблизу точки величина щільності більша ніж далі від неї. Для кожної точки наблюдений величина щільності розподілу змінюється відповідно деякій простій функції.

Сумарну функціональну оцінку щільності ймовірності можливо розрахувати як суму вказаних функцій. Найчастіше в якості ядерної функції використовують функцію Гауса з формою у вигляді дзвона. При великому обсязі спостережень метод ядерних оцінок дозволяє достатньо точно розрахувати щільність ймовірності належності образу певним класам. Розрізняють два основних типи ймовірнісних НМ:

- PNN - використовується для вирішення задачі класифікації.
- GRNN - використовується для вирішення задачі регресії.

Базова модель мережі PNN має дві модифікації. Перша модифікація моделі передбачає, що пропорції класів в навчальній множині образів відповідають їх пропорціям на множині всіх можливих образів.

Наприклад, якщо серед всіх запитів до Web-сервера 1% складають запити з метою отримання НСД, то в навчальній виборці також потрібно передбачити 1% відповідних образів. Для багатьох практичних задач, в тому числі і для задач моніторингу комп'ютерних систем, вказане передбачення є неприйнятним.

Друга модифікація PNN враховує той факт, що використання реальних зашумлених даних як для навчання так і для розпізнавання неминуче призводить до виникнення помилок класифікації.

В багатьох випадках доцільно вважати, що деякі види помилок класифікації важливіші ніж інші. Важливість цих помилок можливо врахувати за допомогою вагових коефіцієнтів. Таким чином, формальним правилом відповідності невідомого образу x k-му класу є вираз:

$$h_k c_k f_k(x) > h_i c_i f_i(x), \exists i \in \{N\}, \quad (79)$$

де $\{N\}$-множина всіх класів, i - довільний клас, h_k та h_i - апріорні ймовірність класифікації образу, c_k та c_i - ціна помилки классифікації образу, $f_k(x)$ і $f_i(x)$ - функції щільності ймовірності для класів k та i.

На практиці розрахунок апріорних ймовірностей та помилок класифікації в багатьох випадках достатньо складний. Тому, часто ці величини вибираються однаковими для всіх класів. Оцінка функції щільності ймовірності виставляється на основі учбових образів з використанням метода Парцена. При цьому застосовується вагова функція (ядро), що має центр в точці, яка представляє учбовий образ. Як вже було відмічено, найчастіше в якості ядра використовують функцію Гауса.

Мережа складається із чотирьох шарів нейронів, кількість яких визначається структурою учбових даних.

Кількість вхідних нейронів дорівнює кількості ознак класу.

Кількість елементів шару образів дорівнює кількості учбових образів. Вхідний шар та шар образів складають повнозв'язну структуру.

Кількість елементів шару додавання дорівнює кількості класів. Елемент шару образів пов'язаний тільки з тим елементом шару додавання, якому відповідає клас образу.

Архітектурна схема мережі PNN, що розподілу образів на два класи А та Б показана на рис.13. При цьому вектор образу складається із 3 компонент, а кількість учбових образів дорівнює 4. Образи, що відповідають нейронам №1, №2 та №3 відносяться до класу А, а образ №4 відноситься до класу Б. Позначення ВЕ означає вихідний елемент (нейрон) мережі.

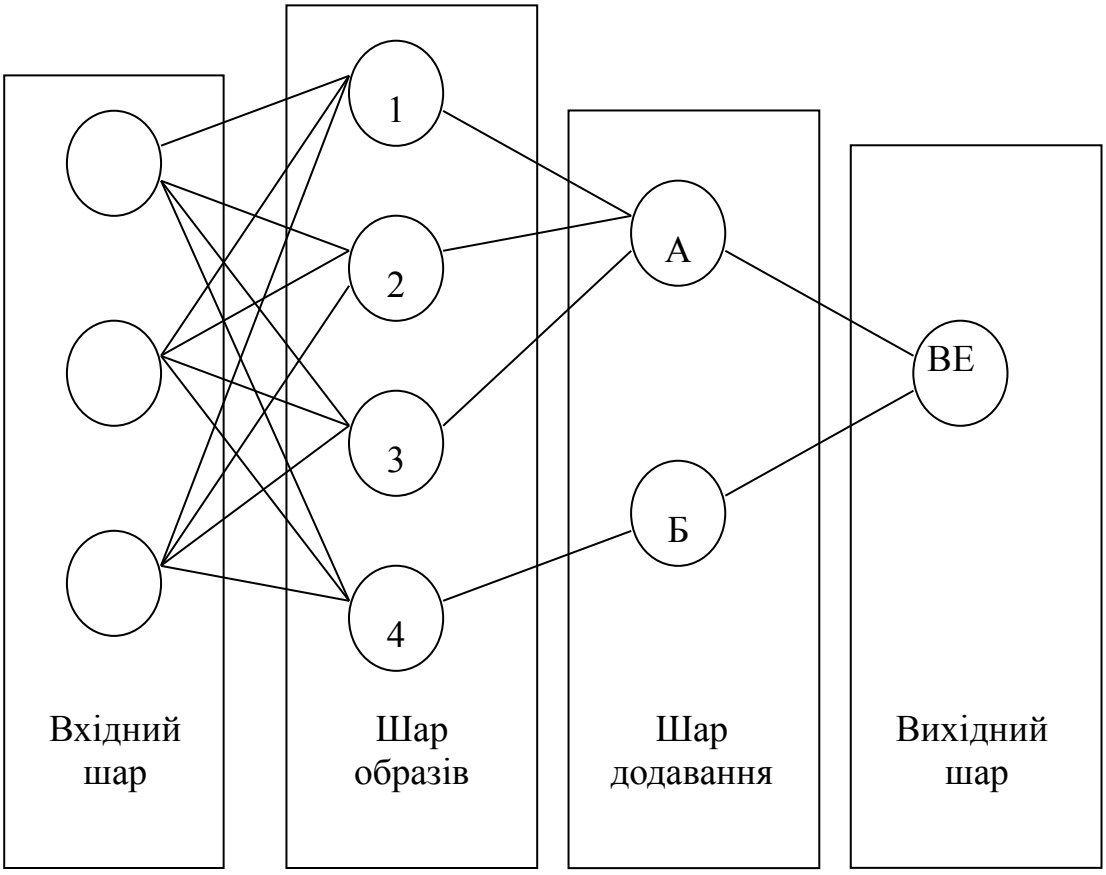


Рис.12. Архітектурна схема мережі PNN

Активність j-го нейрону шару образів (θ_j^o) розраховується так:

$$\theta_j^o = \sum_{i=1}^{N} exp\left(\frac{-(w_{i,j} - x_i)^2}{\sigma^2} \right), \qquad (80)$$

де x - невідомий образ, x_i - i-а компонента невідомого образу, N - кількість компонент вхідного вектора-образу, σ - радіус функції Гауса.

Як правило, при вирішенні задач із області комп’ютерної діагностики

величина радіусу функції Гауса лежить в межах від 0,1 до 1. Для спрощення розрахунків можливо прийняти, що σ^2=0,5.

Для зв'язків, що входять в елемент шару образів вагові коефіцієнти встановлюються такими ж, як складові частини відповідного учбового вектора-образу. Таким чином, всі параметри мережі PNN безпосередньо визначаються учбовими даними.

Тому навчання мережі відбувається відносно швидко. Вагові коефіцієнти зв'язків, що входять в нейрони шару додавання та в вихідний елемент дорівнюють 1.

Нейронам шару додавання характерна лінійна функція активації. Активність *j*-го нейрону шару сумування (θ_j^s) розраховується так:

$$\theta_j^s = \frac{\sum_{i=1}^{N} \theta_i^o}{N}, \tag{81}$$

де N - кількість нейронів шару образів, пов'язаних з *j*-им нейроном шару додавання, θ_j^o- активність *i*-ого нейрону шару образів, пов'язаного з *j*-им нейроном шару додавання.

Значення активності нейрону шару сумування дорівнює ймовірності віднесення вхідного образу до класу, що відповідає даному нейрону. Задачею вихідного елементу є тільки визначення нейрону шару сумування з максимальною активністю. Тому на практиці вихідний елемент може бути реалізований не тільки як нейрон.

Важливим позитивним моментом процесу навчання мережі PNN є наявність тільки одного управляючого параметру навчання, значення якого вибирається користувачем. Фактично цим параметром є радіус функції Гауса. Вказано, що мережі PNN мало чуттєві до величини радіусу функції Гауса.

До переваг мережі PNN відноситься:

- Можливість проведення якісної класифікації на невеликих наборах учбових даних.
- Низька чутливість до помилкових даних в учбових наборах.
- Простота програмної реалізації та ймовірністний зміст класифікації.

Вказані переваги значно полегшуютьнтерпретацію вихідних результатів.

Загальними недоліками мережі PNN є:

- Якісна класифікація образів можлива тільки в діапазоні навчальних даних. В класичному вигляді мережа не здатна проводити узагальнення та не володіє асоціативними властивостями.
- Потенційно висока обчислювальна ресурсоємкість. Причиною цього є те, що мережа PNN містить в своєму складі весь навчальний матеріал, а через це вона потребує великого обсягу пам'яті та повільно працює.
- Можливість використання тільки в задачах класифікації.

Відзначимо, що вказані недоліки не є критичним в багатьох задачах контролю та діагностики комп’ютерних систем. Наприклад, для вирішення проблеми ресурсоємкості мережу можна реалізувати апаратними засобами.

Недоліки, пов'язані з поганим узагальненням результатів можна нівелювати за рахунок оптимізації множини навчальних даних та модифікації архітектури НМ. При цьому слід враховувати, що традиційною сферою використання мережі PNN є попередня обробка даних для виділення із них найбільш інформативних параметрів. Тому використання мережі PNN для вирішення практичних задач в області комп’ютерних систем має хороші перспективи. Однак для цього необхідно організувати ефективну системи збору та обробки статистичної інформації, адаптувати мережу до розпізнавання як можна більш широкої номенклатури класів, пристосувати її до донавчання в процесі експлуатації для розпізнавання нових класів та інтегрувати PNN до сумісного використання з іншими типами НМ.

Ще одним представником ймовірністних НМ є загально-регресивна нейронна мережа (GRNN), архітектура якої подібна PNN. Призначенням мережі GRNN є вирішення задач регресії. Принцип функціонування такої мережі полягає у встановленні зв'язку між кожною точкою вхідних даних та деякою функцією Гауса. Вважається, що наявність даних в точці свідчить про певну ймовірність величини функції Гауса в цій точці. Причому ймовірність зменшується при віддаленні від точки.

В процесі навчання GRNN записує в себе всі точки навчальної вибірки та використовує її для оцінки відгуку в довільній точці. Сумарна вихідна оцінка мережі розраховується як зважене середнє виходів по всім навчальним даним. Величини вагових коефіцієнтів означають відстань від точок навчальних даних до точки класифікуємих даних.

Мережа складається із 4 шарів нейронів. Задача вхідного шару є тільки прийом зовнішнього сигналу та його розподіл між всіма нейронами першого проміжного шару, з гаусівською функцією активації. Другий проміжний шар містить два нейрони для розрахунку складових середнього зваженого. Вихідний шар призначений для остаточного визначення середнього зваженого. Можлива модифікація GRNN для того, щоб радіальні елементи відповідали не окремим навчальним даним, а їх класам. Це дозволяє зменшити розміри мережі та підвищити швидкість навчання та розпізнавання. Центри класів можливо розрахувати за допомогою методу К-середніх або за допомогою мережі Кохонена.

Недоліки та переваги мережі GRNN в основному ті ж самі, що і у мережі PNN. Найважливіша відмінність полягає в сфері застосування - вирішення задач регресії, тому використання мережі даного типу доцільне в засобах розслідування причин порушення технічного стану комп'ютерної системи.

Розділ 6. Рекурентні нейронні мережі

До рекурентних відносять НМ, в яких вхідна інформація передається між нейронами не тільки в напрямку вхід-вихід, але і в зворотньому напрямку. Для цього в рекурентних НМ використовуються зворотні зв'язки між нейронами. Завдяки зворотнім зв'язкам нейрони можуть повторно виконувати свої функції. Тим самим інформація багаторазово проходить через НМ. Фрагмент рекурентної НМ показаний на рис.14.

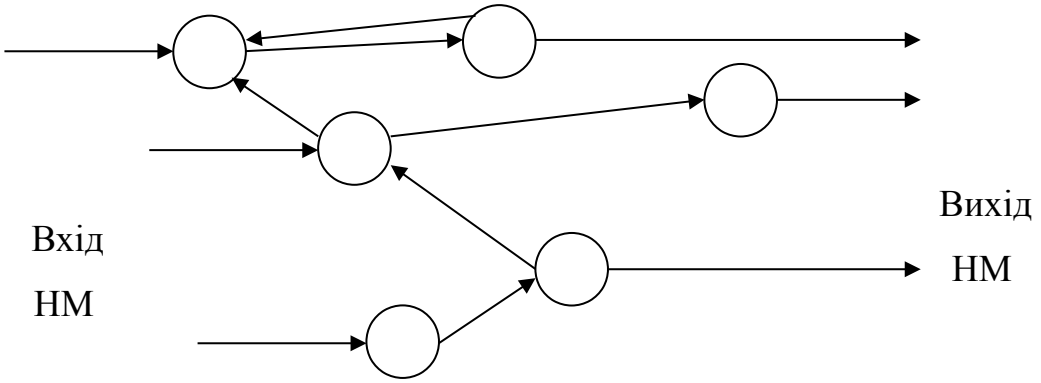


Рис.14. Фрагмент рекурентної НМ

Основою перевагою рекурентних НМ є динамічність та ітераційність обробки даних, що в перспективі повинно позитивно вплинути на узагальнюючі та обчислювальні можливості. Однак потенційна нестійкість та недостатня дослідженість НМ довільної архітектури із зворотніми зв'язками є серйозними перешкодами для їх широкого застосування.

Нестійкість НМ полягає в постійній зміні стану нейронів без виникнення стаціонарного стану мережі. Наслідком нестійкості може бути як колапс процесу навчання, так і не визначеність вихідної інформації в процесі розпізнавання невідомого образу.

Широкого розповсюдження набули рекурсивні НМ типу асоціативної пам'яті, базовою архітектурою яких є мережа Хопфілда. До відомих типів асоціативних НМ відносяться також мережі Коско та Хеммінга, деякі модифікації якої не мають зворотніх зв'язків, а тому не відносяться до

рекурсивних мереж.

Крім класичних асоціативних мереж Хопфілда, Хеммінга, та Коско заслуговує на увагу рекурсивна автоассоціативна мережа (RAAM - Recursive Autoassociative Memory), яку можна вважати розвитком мережі Джордано і простої рекурентної мережі (SRN - Simple Recurrent Network) та подібна до неї семантична нейронна мережа.

Специфічним призначенням RAAM та семантичної нейронної мережі є розпізнавання семантики тексту, що значною мірою вплинуло на їх архітектуру та методи навчання. Відзначимо, що з точки зору наявності зворотніх зв'язків між нейронами, мережі типу карти Кохонена також відносяться до рекурентних. Але, по причині значної відмінності в архітектурі, сфері застосування, методів навчання та теоретичній базі НМ типу карти Кохонена розглядаються окремо від рекурентних. Крім того, не розглядаються малодосліджені НМ та типи мереж з обмеженими можливостями (лінійний асоціатор).

Розглянемо класичні асоціативні НМ типу Хопфілда, Хемінга та Коско. НМ такого типу застосовуються для відновлення за допомогою набору певних асоціативних ознак образів, збережених в пам'яті мережі у вигляді множини векторних пар:

$$\{\langle x_i, y_i \rangle\}, \quad (82)$$

де X, Y - множини асоційованих між собою образів, що зберігаються в пам'яті мережі, P - кількість асоційованих образів.

Для кожної асоційованої пари $\langle x_i, y_i \rangle$ образ x_i є асоціацією для відновлення образу y_i. Розрізняють три класи асоційованої пам'яті:

1. Гетероасоціативна. Вхідний вектор x по наперед визначеним правилам співвідноситься з деяким вектором x_i, який вже зберігається в пам'яті НМ у вигляді (82). Після цього вектор x асоціюється з вектором y_i. Цей клас пам'яті використовується для відновлення одного із збрігаємих образів.

2. Автоасоціативна. Для кожної асоційованої пари (82) $x_i = y_i$. Така пам'ять використовується для відновлення повного вектору по його частині.

3. Інтерполятивна. Для вхідного вектору x по наперед визначеним правилам розраховується:

$$x = x_i + \delta_i, \tag{83}$$

де x_i - еталонний образ, що зберігається в пам'яті НМ, δ_i - відмінність між вхідним та еталонними образами.

Розрахунок вектору y асоційованого з x реалізується так:

$$y = y_i + f(\delta_i), \tag{84}$$

де $f(\delta)$ - деяка функція.

Цей класс пам'яті використовується для моделювання вихідного вектору, що відрізняється від еталонної асоціації.

Базою НМ Хопфілда є аналогія з відомим фізичним об'єктом - спиновим склом. Як і спинове скло, мережа Хопфілда характеризується симетричністю зв'язків між нейронами та може мати декілька стаціонарних конфігурацій активностей нейронів, до яких сходиться динаміка НМ.

Симетричність зв'язків означає, що матриця вагових коефіцієнтів $\overline{w}$ є повною та симетричною, тобто $w_{i,j} = w_{j,i}$. При цьому взаємодія нейрону з самим собою відсутня $w_{i,i} = 0$. Це означає рівність нулю діагональних елементів матриці $\overline{w}$. Аналогічно спиновому склу, для нейронів використовується порогова функція активації з величиною порогу рівною θ.

Найбільш дослідженою є бінарна мережа Хопфілда, в якій нейрон може мати два стани: –1 або 1. Сумарний вхід для i-го нейрону розраховується так:

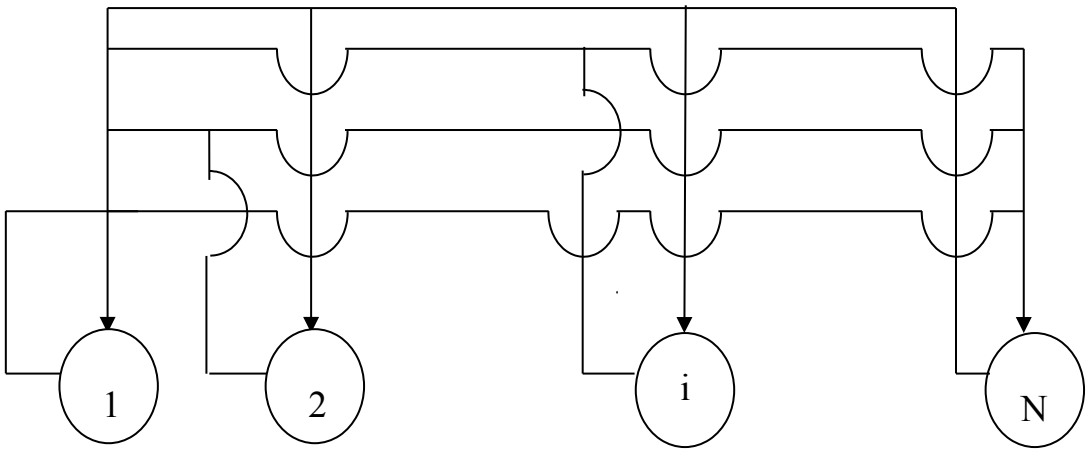
$$NET_i(t) = \sum_{j=1}^{N} \left(s_j(t) \times w_{j,i} \right), \tag{85}$$

$$s_i(t+1) = \begin{cases} 1, \exists (NET_i(t) - \theta_i) \geq 0, \\ -1 \exists (NET_i(t) - \theta_i) < 0. \end{cases}, \tag{86}$$

де $NET_i(t)$ - сумарний вхід i-го нейрону в момент часу t, N- кількість нейронів, $s_i(t)$ - стан нейрону з номером i, в момент часу t, θ_i - пороговий рівень активації j-го нейрону.

Кількість нейронів в мережі відповідає кількості компонент вхідного сигналу.

Структура мережі Хопфілда показана на рис. 15.



1,2,...,i,...N - номера нейронів

Рис. 15. Структура мережі Хопфілда

Для мережі Хофілда всі нейрони вважаються одночасно вхідними, схованими та вихідними. При цьому в якості входу використовується початковий розподіл станів нейронів, а в якості виходу - кінцевий (стаціонарний) розподіл станів. Кількість нейронів дорівнює розмірності вхідного/вихідного сигналу.

Відповідно (87) складові вхідного вектору можуть приймати тільки два дискретні значення: –1 або 1 (в деяких модифікаціях 0 або 1).

Процес розпізнавання починається з подачі вхідного вектору $X=(x_1,x_2,...x_N)$, який відповідає невідомому образу. Подача вектору X означає призначення кожному з нейронів одного із двох можливих станів: –1 або 1. Після цього в дискретні моменти часу починають оновлюватись стани нейронів.

Нейрони можуть оновлюватись незалежно один від другого або всі разом. В першому випадку динаміку НМ називають послідовною, а в протилежному випадку - паралельною. Теоеретичні результати вказують, що властивості НМ практично не залежать від виду нейродинаміки. При цьому для НМ, орієнтованих на однопроцесорні комп'ютери, зручніше використовувати послідовну динаміку мережі.

Нейрон для оновлюеннія вибирається або послідовно, один за іншим, або випадковим чином. Якщо нейрони оновлюються випадковим чином, то в середньому кожен нейрон повинен пройти оновлення однакову кількість разів.

Обраний нейрон, відповідно (85), отримує сигнали від всіх інших нейронів і переходить в стан, визначений умовою (86). Оскільки кожен *j*-й нейрон змінює свій стан відповідно виразу $(net_j(t) - \theta_j)$, то справедливим є твердження:

$$\Delta E_i = -(s_i(t+1) - s_i(t)) \times (h_i(t) - \theta_i) \le 0. \quad (87)$$

Тобто, кожна процедура процесу оновлення нейрону призводить до зменшення його власної енергії.

Загальна енергія мережі Хопфілда, що розраховується відповідно (88) також зменшується.

$$E(S) = -0{,}5 \times \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} (w_{i,j} \times s_i \times s_j) + \sum_{i=1}^{N} (s_i \times \theta_i), \quad (88)$$

де $E(S)=f(t)$ - енергія (енергетична функція) мережі Хопфілда, $S = (s_1, s_2, \ldots s_N)$ - вектор станів НМ, $S=f(t)$.

Доведено, що динаміка системи (87) є стійкою та закінчується в одному із її мінімумів при довільному початковому векторі станів S і довільній матриці вагових коефіцієнтів зв'язків $\overline{w}$.

Стани, в яких сходиться динаміка мережі, називаються атракторами. Поверхня функції енергії $E(S)$ в просторі ознак має достатньо складну форму з великою кількістю локальних мінімумів. Стаціонарні стани НМ, що відповідають локальним мінімумам інтерпретуються як образи в пам'яті НМ. Динаміку мережі, що визначається оновленням станів нейронів можливо інтерпретувати як процес розпізнавання образу, що запам'ятався. При реалізації НМ процес оновлення закінчується, коли при оновленні стан будь-якого нейрону не змінюється.

Для внесення в пам'ять мережі потрібних образів необхідно в процесі навчання визначити матрицю вагових коефіцієнтів. Для класичної мережі Хопфілда використовується правило навчання Хебба, результатом якого є вираз для розрахунку вагових коефіцієнтів зв'язків між нейронами:

$$W = \sum_{n=1}^{P} \left(X_n \times X_n^T \right), \tag{89}$$

де P - кількість навчальних образів, W - матриця вагових коефіцієнтів, X_n (X_n^T) - матриця (транспонована матриця), n-го навчального образу.

Для програмної реалізації більш зручною формою (89) є:

$$w_{i,j} = \begin{cases} \sum_{n=1}^{P} \left(x_{i,n} \times x_{j,n} \right), \forall i \neq j, \\ 0, \forall i = j, \\ i, j \in [1..P]. \end{cases} \tag{90}$$

де $w_{i,j}$ - ваговий коефіцієнт зв'язку між і-м та j-м нейронами, x_i, x_j - i-а та j-а компонента n-го навчального образу.

Відзначимо, що в (90) вектору x_n відповідає вектор s_n стану нейронів мережі. При цьому враховано відсутність взаємодії нейрону з самим собою. Використання (89, 90) призводить до того, що мережа Хопфілда навчається шляхом безпосередньої обробки навчальних даних, що позитивно впливає на швидкість навчання та ефективність програмної реалізації. Однак обсяг образів, які можуть бути збережені в мережі, є відносно невеликим в порівнянні з БШП.

Це пояснюється виникненням атракторів не пов'язаних із зберігаємими образами. Так в теоретичних роботах наведені оцінки максимальної кількості образів (p_{max}), що можуть бути збережені в достатньо великій мережі при умові безпомилкового розпізнавання більшості із них:

$$p_{max} < \frac{N}{2 \times ln\, N}, \tag{91}$$

$$p_{max} \approx (0{,}14 \div 0{,}15) \times N, \tag{92}$$

де N - кількість нейронів в мережі.

Оцінку максимального обсягу збережених образів при умові безпомилкового розпізнавання всього обсягу пам'яті можливо визначити так:

$$p_{max} \leq (0{,}05 \times N). \tag{93}$$

При цьому навчальні образи повинні бути слабо корельовані між собою. В протилежному випадку можливо виникнення перехресних асоціацій при їх пред'явленні на вході мережі. Достатня умова слабкої кореляції між навчальними образами:

$$\left| (x_k, x_j) \right| < \frac{P}{N}, \tag{94}$$

де x_k та x_j - k-ий та j-ий навчальні образи, P - кількість образів, що записані в пам'ять мережі, (x_k, x_j) - відстань Хеммінга між k-м та j-м навчальними образами, що розраховується так:

$$(x_k, x_j) = \sum_{i=1}^{N} (x_{k,i} - x_{j,i})^2 . \qquad (95)$$

До недоліків класичної мережі Хопфілда відносять:

- відносно невелику ємність НМ,
- можливість зациклювання в процесі розпізнавання при використанні корельованих еталонів,
- неможливість навчання на зашумлених образах,
- квадратичне зростання кількості міжнейронних зв'язків при збільшенні розмірності вхідного вектору.

Для збільшення обсягу пам'яті були запропоновані різноманітні модифікації правила навчання Хебба. Найбільш відомі з них:

- процедура Кріка-Мітчісона,
- метод Кінцеля,
- ортогоналізація навчальних даних,
- методи, що базуються на принципі модельного загартування.

Процедура Кріка-Мітчісона використовується для зменшення кількості атракторів не пов'язаних із зберігаємими образами, тобто для забування мережею хибних образів. Процедура полягає в багаторазовому пред'явленні вже навченій мережі Хопфілда довільним чином генерованих образів. Пред'явлення будь-якого з цих образів призводить до переходу вектору станів нейронів мережі з s_i в s_j, При цьому вектор s_j є локальним мінімумом енергії мережі (88) та в загальному випадку може відповідати як істинній так і хибній пам'яті. Однак при великому обсязі навчальних даних вектор s_j найчастіше буде відповідати саме хибній пам'яті. Незалежно від цього вектор вагових коефіцієнтів зв'язків між нейронами змінюється на величину:

$$\delta w_{i,j} = -\varepsilon \times s_i \times s_j , \qquad (96)$$

де ε - деяке невелике позитивне число.

Хоча використання (96) впливає на всі локальні мінімуми енергії мережі, але в більшості випадків вплив здійснюється на локальні мінімуми,

що відповідають хибній пам'яті. При цьому атракторам хибної пам'яті відповідають менші енергетичні мінімуми. Тому процедура (96) призводить до зменшення обсягу хибної пам'яті мережі. Однак використання даної процедури не можливе при необхідності запам'ятовування мережею корельованих образів.

Метод Кінцеля застосовується для мережі Хопфілда з нейронами, які мають нульові пороги активації та міжнейронні зв'язки, величини яких мають Гаусів розподіл з нульовим середнім. Суть методу полягає в тому, що після навчання (90) в мережі знищуються всі зв'язків, для яких $w_{i,j} \times s_{n,j} \times s_{n,i} < 0$. В результаті всі стани, що кодуються векторами навчальних образів, є стаціонарними.

Однак, метод Кінцеля ефективний тільки для сильно корельованих навчальних образів. В випадку слабо корельованих образів в мережі майже всі стани стають стабільними, що призводить до великого обсягу хибних образів.

Процедура ортогоналізації призводить до суттєвого збільшення обсягу пам'яті НМ - $p_{max} \approx N$. Проте має суттєві недоліки. Для визначення будь-якого вагового коефіцієнта зв'язку необхідно знати стан всіх нейронів мережі, а це значно ускладнює реалізацію НМ. Крім того, всі навчальні образи повинні бути відомі до початку процесу навчання. Навіть незначна зміна навчальної вибірки вимагає повного перенавчання мережі.

Методи модельного загартування базуються на аналогії з процесом загартування металів, під час якого метал спочатку сильно нагрівається, а потім поступово охолоджують. Завдяки цьому метал стає більш гнучким, що в свою чергу дозволяє надати йому потрібно форми.

В задачах оптимізації метод модельного загартування використовують для визначення глобального оптимуму функції енергії системи, який співвідноситься з вирішенням поставленої задачі.

Відомою модифікації мережі Хопфілда, що використовує метод модельного загартування для навчання НМ є машина Больцмана. Процедура пошуку глобального мінімуму починається з визначення діапазону в якому буде проведена оцінка функції. Після цього значення функції оцінюються в

деякій кількості випадково вибраних точок. Визначаються точки, в яких значення функції енергії найменші. Нова оцінка функції проводиться в діапазоні навколо цієї точки. Процес повторюється до визначення глобального оптимуму з заданою точністю.

Вибір початкового діапазону і наступне його зменшення аналогічне визначенню початково високої температури металу з наступним поступовим охолодженням. Крім того, можуть бути використані точки в яких енергія системи не є мінімальною. Ймовірність використання цих точок (q) розраховується так:

$$q = exp\left(- \frac{\Delta E}{T} \right), \tag{97}$$

де ΔE - зміна енергії системи, T - температура системи.

Перевірка може бути реалізована для різних температурних діапазонів. Відносно мережі Хопфілда (97) з урахуванням (88) трансформується в:

$$q = \frac{1}{1 + exp\left(2 \times s_j \times \frac{\sum_{i=1}^{P} (s_i \times w_{i,j})}{T} \right)}. \tag{98}$$

Принциповою перешкодою використання методів модельного загартування є недостатня апробованість, наявність емпіричних коефіцієнтів, що використовуються в процесі навчання та складність як програмної, так і апаратної реалізації.

Мережа Хопфілда призначена для віднесення невідомого образу до одного із наперед визначених класів. Процес класифікації відбувається за допомогою методу максимальної правдоподібності.

В якості міри правдоподібності використовується відстань Хеммінга (95) між невідомим образом та визначеними в мережі еталонами. Мережа повинна визначити еталон з мінімальною відстанню Хеммінга до невідомого образу, в результаті чого буде активовано тільки один вихід мережі, що відповідає вказаному еталону.

Структура мережі Хеммінга представлена на рис. 16.

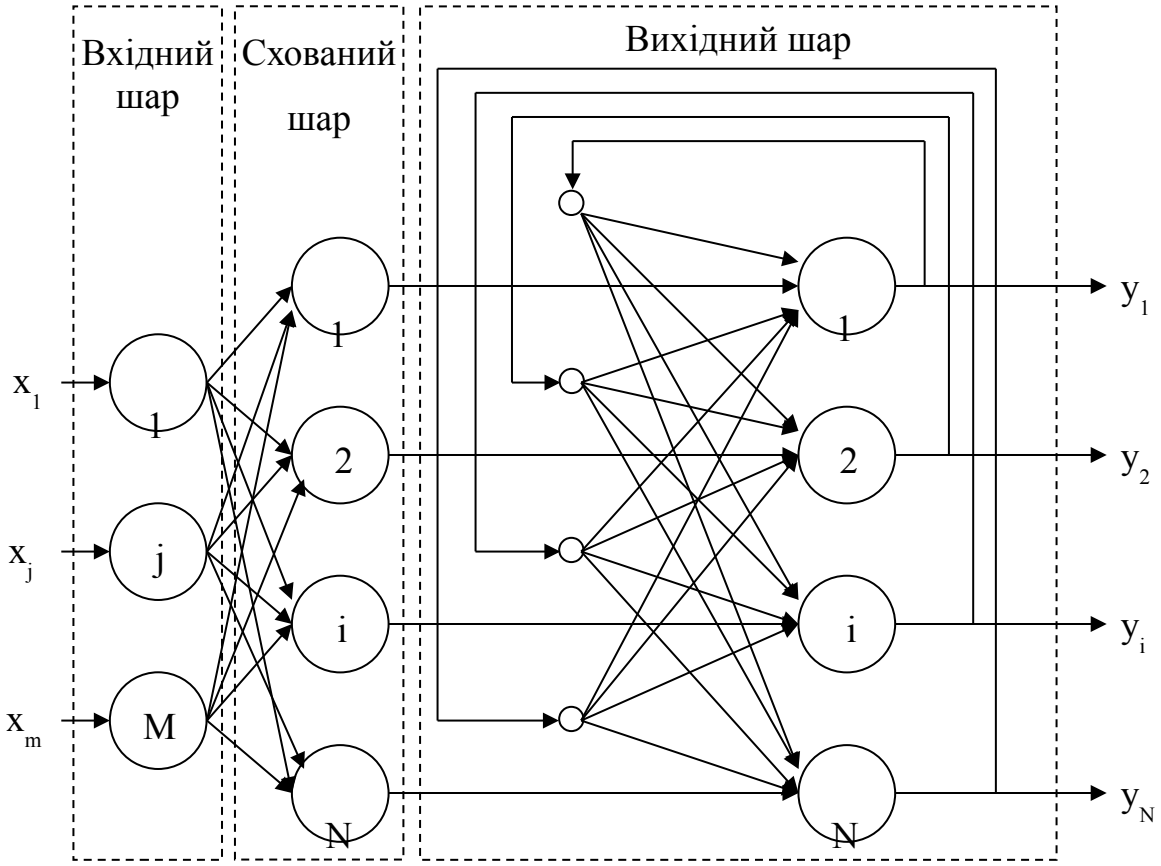


Рис.16. Структура мережі Хеммінга

Мережа складається із трьох шарів нейронів: вхідного, схованого та вихідного. Кількість нейронів вхідного шару (*M*) відповідає розмірності вхідного сигналу мережі. Задачею вхідних нейронів є тільки розповсюдження вхідної інформації по НМ, тому при розрахунках параметрів мережі даний шар не враховується. Кількість нейронів в схованому та вихідному шарі (*N*) дорівнює кількості еталонних образів мережі.

Кожен з нейронів вхідного шару сполучений прямим зв'язком з кожним із нейронів СШН. Ваговий коефіцієнт зв'язку між *j*-м вхідним та *i*-м схованим нейроном позначимо $w_{j,i}^{(1)}$.

Для нейронів СШН характерна порогова функція активації виду одиничного стрибка. Кожен з нейронів СШН пов'язаний прямим зв'язком тільки з одним нейроном вихідного шару. Всі нейрони вихідного шару пов'язані між собою зворотніми зв'язками.

Вагові коефіцієнти зворотніх зв'язків між j-м та і-м вихідними нейронами позначимо - $w_{j,i}^{(2)}$. При цьому зв'язок вихідного нейрону з самим собою позитивний ($w_{j,i}^{(2)} > 0, \exists j = i$), всі ж інші зворотні зв'язки негативні ($w_{j,i}^{(2)} < 0, \exists j \neq i$). Нейронам вихідного шару призначається гістерезисна функція активації.

Навчання мережі Хеммінга полягає в безпосередній обробці навчальних еталонних даних. Вагові коефіцієнти вхідних зв'язків *i*-го схованого нейрону, що відповідає *i*-му еталонному образу розраховуються за допомогою виразу:

$$\begin{cases} w_{j,i}^{(1)} = x_j^{(i)} / 2, \\ j = 1..M, i = 1..N. \end{cases} \quad (99)$$

де $x_j^{(i)}$ - j-й компонента вхідного вектору для і-го еталонного образу.

Вагові коефіцієнти вхідних зв'язків схованих нейронів навчених всім еталонним образам визначаються відповідно (89, 90) з врахуванням (99). Активаційні пороги у всіх схованих нейронів ($\theta^{(1)}$) однакові:

$$\theta^{(1)} = M / 2 \quad (100)$$

Вихідна інформація схованих нейронів без обробки поступає на вхід нейронів вихідного шару. Таким чином:

$$NET_i^{(2)} = OUT_i^{(1)}, \quad (101)$$

де $OUT_i^{(1)}$ - вихідний сигнал *i*-го нейрону в схованого шару, $NET_i^{(2)}$ - сумарний вхідний сигнал *i*-го вихідного нейрону.

Визначаються вагові коефіцієнти та величини порогів вихідних нейронів:

$$\begin{cases} 0 > w_{j,i}^{(2)} > -1/N, \exists j \neq i, \\ w_{j,i}^{(2)} = 1, \exists j = 1. \end{cases} \quad (102)$$

Відзначимо, що вагові коефіцієнти зворотніх негативних зв'язків та величина порогу функції активації вихідних нейронів визначаються емпірично. При цьому в багатьох випадках вагові коефіцієнти негативних зв'язків ($w^{(2)}$) та величини порогів ($\theta^{(2)}$) для всіх вихідних нейронів призначаються однаковими:

$$\begin{cases} w^{(2)} = const, \\ \theta^{(2)} = const. \end{cases} \quad (103)$$

Як правило, величина ($\theta^{(2)}$) призначається так, щоб при будь-якому допустимому значенню вхідного сигналу не наступало насичення вихідної функції. Досить часто $\theta^{(2)}$ призначається рівним кількості навчальних еталонів.

Для класифікації невідомого вхідного образу $X=\{x_1,..x_m\}$ використовується ітераційний алгоритм. На підготовчому етапі відповідно (104) розраховуються вихідні сигнали кожного з нейронів СШН:

$$OUT_i^{(1)} = \sum_{j=1}^{M} \left(w_{j,i} \times x_j \right) + \theta^{(1)}, i = 1..N. \quad (104)$$

Розрахованими величинами ініціюються виходи нейронів другого шару:

$$OUT_i^{(2)} = OUT_i^{(1)}. \quad (105)$$

Після цього починається ітераційний обчислювальний процес. Кожна ітерація складається із трьох етапів:

1. Використовуючи величини виходів схованих нейронів, розраховуються стани кожного із вихідних нейронів:

$$s_i^{(2)}(k) = OUT_i^{(2)}(k-1) - w^{(2)} \times \sum_{n=1}^{N} OUT_n^{(2)}(k-1), \quad (106)$$

де k - номер ітерації, $s_i^{(2)}$ - стан i-го вихідного нейрону .

Відзначимо, що ітераційний процес починається з k=1. При цьому $OUT_i^{(2)}(0) = OUT_i^{(1)}$.

2. Розраховуються виходи НМ:

$$OUT_i^{(2)}(k) = f\left[s_i^{(2)}(k) \right], i = 0...N, \quad (107)$$

де $f(x)$ - функція активації вихідних нейронів типу гістерезис.

3. Якщо вихідний вектор після виконання ітерації не змінився, тобто умова (108) виконується, то вважається, що невідомий образ класифіковано:

$$OUT_i^{(2)}(k) = OUT_i^{(1)}(k-1). \quad (108)$$

В протилежному випадку виконується нова ітерація. Відзначимо, що в випадку закінчення процесу класифікації всі компоненти вихідного вектору крім одного дорівнюють 0. Номер ненульового компоненту вихідного вектору є номером еталону, який відповідає вхідному образу.

Аналіз структури мережі Хеммінга, процесів навчання та розпізнавання дозволяють зробити висновок про те, що фактичним завданням вихідного шару є тільки визначення номеру вихідного вектору на який поступає максимальне збудження при подачі на вхід мережі невідомого образу. При цьому вихідний сигнал нейрону з максимальним збудженням прирівнюється 1, а сигнали всіх інших нейронів прирівнюються 0.

Таким чином, при програмній реалізації мережі Хеммінга вихідний шар можливо замінити компонентом який безпосередньо розраховує номер нейрону з максимальним виходом. Використання такої модифікації дозволить підвищити швидкість алгоритму розпізнавання за рахунок вилучення ітераційної складової та зменшити кількість зв'язків між нейронами. При цьому мережа Хеммінга перетвориться із рекурсивної в НМ з прямим розповсюдженням сигналу між нейронами.

Після вилучення вихідного шару кількість міжнейронних зв'язків (Z) буде дорівнювати:

$$Z = N \times M , \tag{109}$$

де N - кількість еталонів, M - розмірність вхідного вектору.

У випадку апаратної реалізації мережі питання заміни вихідного шару потребує додаткового вивчення.

Традиційними перевагами мережі Хеммінга вважаються простота програмної та апаратної реалізації, висока швидкість як навчання так і розпізнавання невідомих образів та відносно великий обсяг пам'яті мережі, який дорівнює кількості нейронів в схованому шарі.

До недоліків мережі, як правило, відносять можливість розпізнавання тільки бінарних образів, визначення тільки номеру еталону при класифікації та можливість розпізнавання тільки слабо зашумлених сигналів.

До цих недоліків також слід додати неможливість навчання на зашумлених образах та занадто просту модель процесу класифікації, що лежить в основі навчання і функціонування мережі. Адже далеко не завжди подібність між навіть бінарними образами можливо оцінити тільки кількістю співпадаючих компонент. Наприклад, вектори, що відповідають образам можуть мати різну довжину, а їх компоненти - не однакову інформативну

цінність. Наслідками вказаних нами недоліків є недостатні узагальнюючі можливості мережі та необхідність спеціальної процедури підготовки вхідної інформації.

Мережа Коско, яка є в певному розумінні розвитком НМ Хопфілда, призначена для вирішення задачі встановлення асоціації між невідомим вхідним образом з деяким еталонним образом. В загальному випадку вхідний і еталонний образи можуть бути не корельовано між собою. В літературі мережу Коско ще називають ДАП та мережею BAM (Bidirectional Associative Memory).

Структура класичної мережі Коско, призначеної для визначення асоціації вхідних векторів розмірності M з вихідними векторами розмірності N показана на рис. 17.

На відміну від структури мережі Хопфілда в мережі Коско є другий шар нейронів, призначенням якого є відображення вихідного вектору. В загальному випадку в НМ Коско складається з трьох шарів: вхідного, схованого та вихідного.

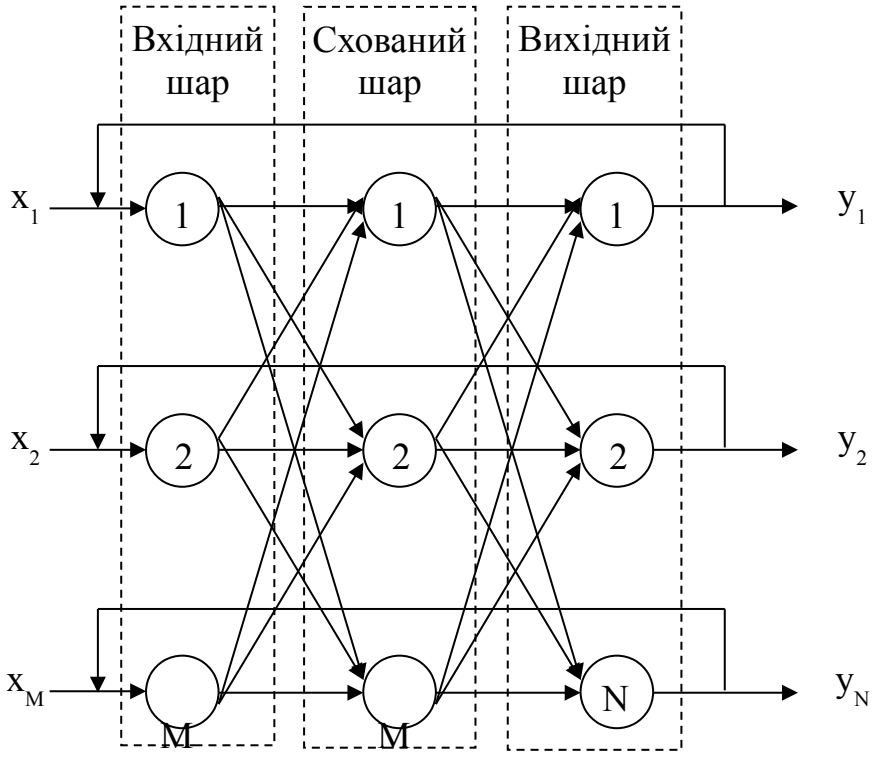


Рис. 17. Структура класичної мережі Коско

Вхідний шар тільки розподіляє інформацію між нейронами СШН, тому досить часто його виключають із розгляду. Призначенням СШН є відображення вхідного вектору. Кожен з нейронів цього шару з'єднаний з кожним із нейронів вихідного шару.

Зв'язки між нейронами в середині шару відсутні, але нейрон сполучений з самим собою. Завдяки цьому в мережі Коско діагональні елементи матриці міжнейронних зв'язків можуть відрізнятися від 0.

На сьогодні найбільш дослідженими та апробованими є дискретні гомогенні бінарні мережі Коско. В таких мережах стани нейронів змінюються дискретно. Це значить, що на протязі деякого терміну часу вхідні сигнали одного шару, які є вихідними сигналами іншого шару стоять в черзі на обробку.

Після закінчення вказаного терміну одночасно по всім нейронам відбувається обробка вхідних сигналів і нові вихідні сигнали займають місце в черзі на обробку. При цьому відповідність невідомого образу з еталонним оцінюється за допомогою відстані Хеммінга.

Для нейронів схованого та вихідного шарів використовується сигмоїдальна (логістична) або порогова функції активації. Розрахунок стану нейронів відбувається з використанням нульового порогу активації.

Навчання мережі відбувається за допомогою правила Хеба і полягає в розрахунку вагових коефіцієнтів прямих та зворотніх зв'язків між нейронами схованого та вихідного шарів з метою визначення асоціацій між всіма парами вхідних (X) та вихідних векторів (Y) - $<X_i, Y_i>$.

Для розрахунку вагових коефіцієнтів прямих зв'язків між нейронами схованого та вихідного шару використовується наступна процедура:

$$W = \sum_{i=1}^{P} \left(X_i^T \times Y_i \right), \qquad (110)$$

де W - матриця вагових коефіцієнтів зв'язків між нейронами, X_i^T - транспонована матриця компонент i-го вхідного образу, Y_i - матриця компонент i-го вихідного образу, P - кількість навчальних образів.

Матриця вагових коефіцієнтів зворотніх зв'язків між схованим та вихідним шаром представляє собою транспоновану матрицю вагових

коефіцієнтів прямих зв'язків W^T. Таким чином, як і мережа Хопфілда, мережа Коско навчається шляхом безпосередньої обробки навчальних даних.

Якщо матриця вагових коефіцієнтів є квадратною і симетричною ($W=W^T$), то мережа Коско зводиться до мережі Хопфілда.

Розрахунок матриці станів нейронів мережі Коско можливо провести за допомогою виразів:

$$S^{(2)} = f\left(W \times S^{(1)}\right), \tag{111}$$

$$S^{(1)} = f\left(W^T \times S^{(2)}\right), \tag{112}$$

де $S^{(1)}$, $S^{(2)}$ - матриці станів нейронів схованого та вихідного шарів, $f(x)$ - функція активації нейронів.

Відповідно (111, 112) розрахунок стану окремого нейрону відбувається за допомогою виразів (113, 114).

$$s_j^{(2)} = f\left(\sum_{i=1}^{M}\left(w_{i,j} \times s_i^{(1)}\right)\right), \tag{113}$$

$$s_i^{(1)} = f\left(\sum_{j=1}^{N}\left(w_{j,i} \times s_j^{(2)}\right)\right), \tag{114}$$

де $s_j^{(2)}$ - стан j–го нейрону вихідного шару, $s_i^{(1)}$ - стан i-го нейрону схованого шару, $w_{i,j}$ - ваговий коефіцієнт зв'язку між i-м та j–м нейронами схованого та вихідного шару.

Алгоритм функціонування мережі Коско в режимі встановлення асоціації для невідомого вхідного образу подібний мережі Хопфілда.

1. Вектор X поступає на вхід мережі. При цьому компоненти X без обробки встановлюються в якості виходів нейронів схованого шару. Після цього вектор X знімається з входу мережі.

2. Інформація подається до вихідного шару, після чого відповідно (113, 1.111) розраховуються стани вихідних нейронів.

3. Інформація подається до схованого шару, після чого відповідно (114, 112) розраховуються стани схованих нейронів.

4. Ітерації 2 та 3 виконуються до досягнення стабільного стану мережі, при якому стани всіх схованих та вихідних нейронів не змінюються. Величини стабільних станів нейронів схованого та вихідного шарів визначають пару асоційованих образів $<X,Y>$.

Відзначимо, що встановлення асоціації можливе не тільки в напрямку $X \rightarrow Y$, але й в протилежному напрямку $Y \rightarrow X$.

В режимі розпізнавання НМ Коско функціонує в напрямку мінімізації енергії мережі, а використання транспонованих матриць вагових коефіцієнтів гарантує досягнення стабільного стану.

Обсяг пам'яті мережі оцінюється за допомогою виразів:

$$K \leq \frac{A}{2 \times ln\, A}, \tag{115}$$

$$K \leq \frac{A}{4 \times ln\, A}, \tag{116}$$

де K - максимальна кількість еталонних образів, що можуть зберігатись в пам'яті мережі, A - кількість нейронів в меншому шарі.

Вираз (115) використовується, якщо достатньо провести правильну асоціацію з більшістю еталонних образів. Вираз (116) використовується при необхідності встановлення безпомилкової асоціації.

Існує досить велика кількість різноманітних модифікацій НМ Коско: негомогенні, безперервні, адаптивні та конкуруючі ДАП. Потенційні можливості таких мереж з точки зору обсягу пам'яті НМ, дещо вищі. Але недостатня теоретична дослідженість та апробованість не дозволяють використовувати їх для вирішення відповідальних задач.

Розглянемо сферу використання, переваги та недоліки класичних рекурентних мереж.

Мережі Хопфілда і Хеммінга відносяться до класу автоасоціативних НМ, а мережа Коско відноситься до класу гетероасоціативних. Як і для БШП, традиційною сферою використання цих НМ є вирішення задач класифікації зашумлених даних та виділення прототипів. Відомі вдалі приклади їх застосування для пошуку промоторів в ДНК, розпізнавання символів та аномалій в електроенцефалограмах.

Доведена доцільність обробки статистичних даних за допомогою асоціативних мереж перед їх використанням в БШП. Доцільність визначається подібністю обробки даних за допомогою асоціативних мереж з статистичним аналізом. Наприклад, автоасоціативні мережі по своїй суті проводять розрахунок голових компонент вибірки статистичних даних. В

свою чергу, аналіз головних компонент в багатьох випадках дозволяє запобігти надмірності даних. Тобто за допомогою автоасоціативних мереж можливо зменшити кількість параметрів, що використовуються при класифікації образів за допомогою БШП. Вказані напрямки використовують специфічні особливості архітектури та навчання вказаних типів мереж, не характерні ні для БШП, ні для мереж типу карти Кохонена.

Крім того, перспективними напрямками застосування класичних асоціативних НМ є активна кластеризація та вирішення задач комбінаторної оптимізації. Активна кластеризація передбачає, що мережа навчена розпізнавати задану кількість еталонів може класифікувати невідомий вхідний образ як новий клас, не визначений в процесі навчання. Процес активної кластеризації базується на тому, що всі аттрактори мережі Хопфілда, вагові коефіцієнти зв'язків якої сформовані відповідно правилу Хебба на основі навчального набору (117) можуть бути проінтерпретовані як найбільш ймовірні версії деякого повідомлення, що було передане P разів через канал з шумом. При цьому варіанти передачі відповідають навчальним образам.

$$\begin{cases} \{x_n\}, \\ n = 1,.., P.' \end{cases} \qquad (117)$$

де P - кількість навчальних образів.

Якщо після навчання провести дослідження всього простору станів мережі, подаючи на її вхід випадковим чином генеровані образи, то в ньому можуть виявитись аттрактори, що не відповідають ні одному навчальному образу. Ці аттрактори отримали назву порожніх класів.

В багатьох випадках порожні класи не співпадають з помилковою пам'яттю мережі, а представляють нову інформацію. Таким чином, за допомогою мережі Хопфілда можливо передбачити існування нових класів, які не були визначені в навчальній вибірці.

Потенційно застосування описаної властивості мережі Хопфілда може знайти своє застосування в області захисту комп'ютерної інформації для визначення нових видів атак або для визначення непередбачених вразливостей комп'ютерної системи.

Вирішення задач комбінаторної оптимізації базується на тому, що мережа Хопфілда може бути розглянута як алгоритм оптимізації цільової функції, представленої у вигляді енергії НМ. Для вирішення таких задач необхідно сформулювати цільову функцію у вигляді функції енергії мережі. Зазначимо, що в загальному випадку методології відображення умов задачі в функцію енергії мережі не існує. Разом з тим, сформовано загальні рекомендації та наведено приклади щодо використання мережі Хопфілда при розв'язанні задач типу транспортно-орієнтованої оптимізації та оптимального розподілу ресурсів. Наведено детальний опис методики адаптації мережі Хопфілда до класичної задачі комівояжера, в якій він повинен об’їхати по замкненому маршруту всі M міст так, щоб повна довжина його шляху була мінімальною. Таким чином, єдиним критерієм оптимізації є мінімізація загальної довжини маршруту комівояжера, а обмеження полягають в проходженні маршруту через будь-яке місто тільки один раз та в необхідності відвідин кожного з міст.

Адаптація НМ складається з таких етапів:

1. Розробка структури мережі.
2. Кодуванні маршруту за допомогою станів нейронів.
3. Формуванні цільової оптимізаційної функції, до складу якої входять параметрами НМ.
4. Встановлення взаємозв'язку між цільовою функцією та енергією мережі.
5. Розрахунку порогових рівнів активації та вагових коефіцієнтів зв’язків між нейронами.

Розглянемо більш детально кожен перерахованих етапів.

1. Використовується НМ Хопфілда, яка складається із $M \times M$ бінарних нейронів. Стан кожного з нейронів розраховується так:

$$s_{i,m} \in \{0,1\} \ \forall (i = 1,..,M; m = 1,..M), \qquad (118)$$

де $s_{i,m}$ - стан нейрону, i - номер міста, m - номер міста в маршруті.

Для представлення мережі використовується квадратна матриця з довжиною сторони - M. Номер рядка матриці відповідає номеру міста, а номер колонки - номеру міста в маршруті.

2. Нейрон вважається активним $s_{i,m}$=1, якщо номер *i*-го міста в маршруті відповідає *m*. В протилежному випадку нейрон не активний $s_{i,m} = 0$.

3. Враховуючи (118) цільову функцію *C(s)* та обмеження на оптимізацію сформовано так:

$$C(s) = \frac{1}{2} \sum_{i,j,m=1}^{M, i \neq j} \left(d_{i,j} s_{i,m} \times \left(s_{j,m-1} + s_{j,m+1} \right) \right), \tag{119}$$

$$\sum_{i=1}^{M} s_{i,m} = 1, \forall m = 1,.., M, \tag{120}$$

$$\sum_{m=1}^{M} s_{i,m} = 1, \forall i = 1,.., M, \tag{121}$$

де $d_{i,j}$ - довжина маршруту між містами *i* та *j*.

Вирази (120) та (121) відповідають першому та другому обмеженням. З врахуванням обмежень визначена загальна цільова функція, що враховує обмеження оптимізації за рахунок використання штрафних складових:

$$\begin{aligned} C'(s) = & \frac{1}{2} \sum_{i,j,m=1}^{M, i \neq j} \left(d_{i,j} s_{i,m} \times \left(s_{j,m-1} + s_{j,m+1} \right) \right) \\ & + \frac{\gamma}{2} \left(\sum_{m=1}^{M} \left(\sum_{i=1}^{M} s_{i,m} - 1 \right)^2 + \sum_{i=1}^{M} \left(\sum_{m=1}^{M} s_{i,m} - 1 \right)^2 \right), \end{aligned} \tag{122}$$

де γ - множник Лагранжа який регулює розмір штрафних складових, що збільшують цільову функцію при відхиленні від обмежень.

4. Цільову функцію (122) прирівняно до енергії мережі (89). За рахунок цього отримано залежність енергії мережі від довжини маршруту та обмежень встановлюється. По причині того, що параметрами функції енергії (89) є вагові коефіцієнти зв'язків та активаційні пороги, отримана залежність дозволяє перейти до встановлення взаємозв'язку між ними та довжиною маршруту комівояжера.

5. Наведено остаточні вирази для розрахунку вагових коефіцієнтів зв'язків між нейронами та активаційними порогами:

$$w_{im,jn} = - d_{i,j} \times (\delta_{m-1,n} + \delta_{m+1,n}) - \theta \times \delta_{m,n} - \theta \times \delta_{i,j}, \tag{123}$$

$$\delta_{i,j} = \begin{cases} 1, i = j, \\ 0, i \neq j. \end{cases} \tag{124}$$

де θ - активаційні пороги нейронів $\theta = -\gamma$, $w_{im,jn}$ - ваговий коефіцієнт зв'язку між нейронами з координатами *(i, m)* та *(j, n)*.

Функціонування мережі в режимі оптимізації починається з деякого випадкового стану та закінчується в стаціонарній конфігурації, що відповідає мінімуму енергетичної функції.

Оптимальному рішенню повинен відповідати стаціонарний стан мережі в якому тільки *M* нейронів будуть активними, при цьому в кожному рядку і в кожній колонці матриці станів нейронів $\|w_{i,j}\|$ буде знаходитись тільки один активний нейрон.

Слід зазначити, що для визначення конфігурації, яка відповідає глобальному мінімуму запропоновано ряд модифікацій мережі - використання неперервних нейронів, мережі Поттса, використання нейронів з нелінійним суматором вхідних сигналів. Хоча їх використання покращує результати пошуку оптимального рішення, про те відзначається складністю розрахунків та потребує подальших досліджень. При цьому для пошуку оптимуму доцільно використовувати достатньо досліджені та надійні методи модельного загартування (98, 99), наприклад, машину Больцмана. В науковій літературі проведено порівняння ефективності вирішення задачі комівояжера за допомогою мережі Хопфілда, методом модельного загартування та мережі Кохонена. Результати порівняння вказують на те, що мережа Хопфілда визначає більш короткі маршрути, проте обчислювальна ефективність мережі Кохонена дещо вища.

Відзначимо, що в сфері комп’ютерних систем є достатньо багато актуальних задач, вирішення яких базується на комбінаторній оптимізації. Наприклад, оптимізація навантаження між різноманітними серверами.

По відношенню до БШП перевагами асоціативних НМ є швидкість навчання, сумісність з аналоговими системами та простоту програмної і апаратної реалізації. До загальних недоліків асоціативних мереж відносять обмеженість пам'яті, квадратичну залежність кількості зв'язків від розмірності вхідного сигналу та деяку непередбачуваність функціонування за рахунок можливих помилок та нестабільної класифікації образів. Для мережі Хопфілда характерним є недолік пов’язаний з тим, що розмірність і тип вхідного сигналу повинні повністю співпадати з розмірністю і типом вихідного сигналу. Ще одним важливими недоліком всіх класичних

асоціативних мереж є неможливість навчання на зашумлених та сильно корельованих образах, що значно звужує можливу сферу їх застосування.

Розділ 7. Мережі адаптивної резонансної теорії

АРТ є вдалою спробою створення мережі, яка поєднує в собі пластичність сприйняття нової інформації з стабільністю збереження старої пам'яті. Мережа АРТ може динамічно запам'ятовувати нові образи без повного перенавчання та втрати інформації про образи, що вже були в ній збережені.

Динамічне запам'ятовування означає невідривність процесів навчання та функціонування мережі. Тобто НМ може одночасно як запам'ятовувати зовсім нові образи, так і уточнювати інформацію про старі образи. Для цього в НМ використовується детектор новизни, який представляє собою специфічний по відношенню до класичних НМ механізм порівняння вхідного образу з вмістом пам'яті. Відзначимо, що в процесі порівняння використовуються не всі ознаки образу, а тільки набір найбільш інформативних.

Формування так званого шаблону критичних рис відбувається автоматично під час функціонування мережі. Випадок успішного порівняння відповідає виникненню адаптивного резонансу деякого нейрону, в якому зберігається певний еталонний образ. Вхідний образ класифікується як відомий еталон. Одночасною модифікуються вагові коефіцієнти зв'язків нейрону, що виконав класифікацію. Якщо резонанс не виникнув, то образ сприймається мережею в вигляді нового класу.

Наслідком сприйняття є додавання в мережу нового нейрону, вагові коефіцієнти зв'язків якого запам'ятовують новий еталон. Вагові коефіцієнти зв'язків нейронів, що не брали участі в резонансі не модифікуються. Таким чином, результатом роботи мережі АРТ є не тільки класифікація вхідного образу, але й уточнення вагових коефіцієнтів, що відповідають збереженим образам.

В базовій конфігурації мережа АРТ дозволяла проводити розпізнавання тільки бінарних образів. Модифікації мережі дозволяють розпізнавати неперервні вхідні вектори, використовувати елементи нечіткої логіки, класифікувати не тільки вхідні, але й вихідні вектори.

Метою більшості відомих модифікацій АРТ було використання мережі для вирішення окремих практичних задач та/або сумісного застосування нейронних мереж та інших засобів вирішення задач в галузі штучного інтелекту. Наприклад, мережі АРТ-2 та АРТ-3 в основному призначені для розпізнавання зображень в складі ієрархічних нейромережевих систем. В мережах типу FART використовуються елементи нечіткої логіки. При цьому принципи функціонування та архітектурні рішення модифікацій залишилися тими, що і в базовій конфігурації мережі АРТ-1.

Блок-схема НМ АРТ-1, що складається із п'яти функціональних блоків, показана на рис. 18.

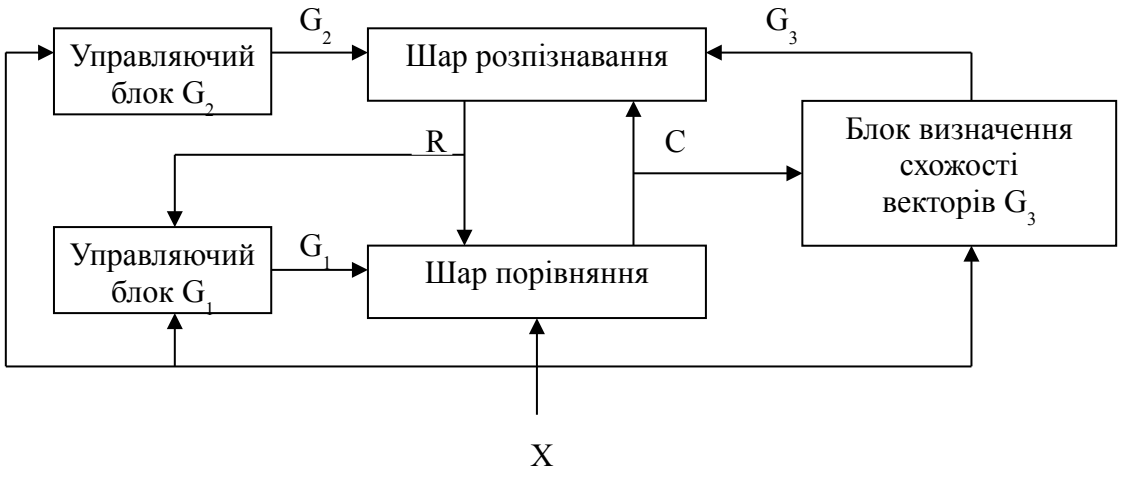


Рис. 18. Блок-схема АРТ-1

На вхід мережі подається *N*-вимірний бінарний вектор *X*. Основну роботу по розпізнаванню та навчанню АРТ виконують нейронні шари порівняння та розпізнавання. При цьому в шарі розпізнавання зберігається *M* еталонів, кожному із яких відповідає власний нейрон. Призначенням інших блоків є виконання однотипних логічних операцій. Методика функціонування цих блоків незмінна, а тому їх реалізація можлива у вигляді

не тільки нейроподібних структур, але й у вигляді звичайних логічних схем. Завданням управляючих блоків G_1 та G_2 є реалізація одиничного бінарного (0 або 1) вихідного сигналу. Вихід блоку G_1 дорівнює 1, якщо на вхід мереж подано ненульовий вектор X, а вихід шару розпізнавання R дорівнює 0. Тобто сигнал блоку G_1 =1 сповіщає блок порівняння про отримання вхідного образу.

Завдання блоку G_2 полягає в інформуванні шару розпізнавання про подачу нового вхідного вектору. Тому вихід G_2 дорівнює 1 тільки тоді, коли на вхід мережі подається новий образ. Метою використання блоку G_3 є визначення подібності вхідного вектору з найбільш схожим на нього вектором, що міститься в пам'яті мережі. Подібність векторів визначається виразом:

$$\frac{\|C\|}{\|X\|} = \frac{\sum_{i=1}^{N} C_i}{\sum_{i=1}^{N} X_i} < \rho, \qquad (125)$$

де X - вхідний вектор, X_i - i-а компонента вхідного вектору, C - вихідний вектор шару порівняння, C_i - i-а компонента вектору C, N - розмірність вхідного сигналу та виходу шару порівняння, ρ - параметр подібності.

Параметр подібності визначає необхідну кількість ненульових компонент вхідного вектору та найбільш подібного йому вектору, що міститься в пам'яті мережі. Параметр ρ є зовнішнім по відношенню до мережі, повинен знаходитись в межах]0,1[та визначається емпірично. Виходом блоку є бінарний вектор G_3, що містить M компонент. Якщо вираз (125) є справедливим, то подібність векторів вважається недостатньою і блок реалізує вихідний сигнал, що є негативним для нейрону переможця в шарі розпізнавання.

Блок визначення схожості пам'ятає свій стан впродовж однієї класифікації. Тому негативний сигнал також зберігається. Цим самим нейрон переможець, для якого подібність визначена недостатньою, далі не приймає участі в класифікації. Основним завданням шару порівняння, структура якого показана на рис. 19 є визначення шаблону критичних рис.

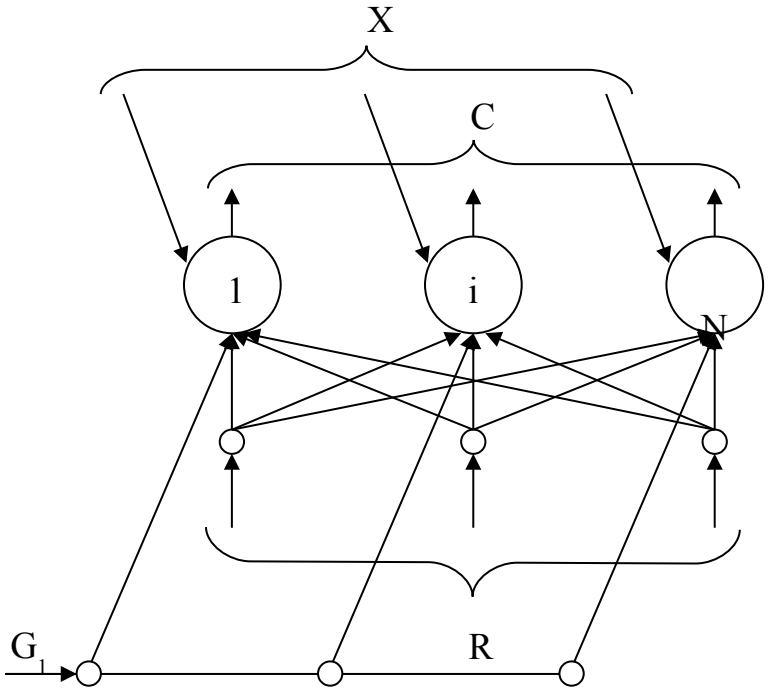


Рис. 19. Структура шару порівняння

Вхід будь-якого *i*-го нейрону шару порівняння сполучений з відповідною і-ою компонентою вхідного вектору *X*, виходом блоку G_1 та виходом кожного із нейронів шару розпізнавання. На рис.1.18 останні зв'язки позначені літерою *R*, а виходи нейронів - літерою *C*. Вагові коефіцієнти вхідного зв'язку та зв'язку з блоком G_1 дорівнюють 1. Вагові коефіцієнти зв'язків з виходами нейронів шару розпізнавання складають матрицю вагових коефіцієнтів $W^{(1)}$. Вхід кожного із нейронів шару порівняння можливо розрахувати так:

$$NET_i^p = X_i + G_1 + \sum_{j=1}^{M} \left(R_j \times w_{j,i}^{(1)} \right), \tag{126}$$

де R_j - вихід *j*-го нейрону шару розпізнавання, $w_{j,i}^{(1)}$ - ваговий коефіцієнт зв'язку *i*-го нейрону шару порівняння з *j*-им нейроном шару розпізнавання.

Нейрони шару порівняння використовують порогову активаційну функцію з величиною θ=2. Тому вони активізуються тільки тоді, коли як мінімум два вхідні зв'язки відрізняються від 0:

$$C_i = \begin{cases} 1, \exists NET_j \geq 2, \\ 0, \exists NET_j < 2. \end{cases} \tag{127}$$

де C_i - вихід *i*-го нейрону шару порівняння.

Структура шару розпізнавання показана на рис. 20.

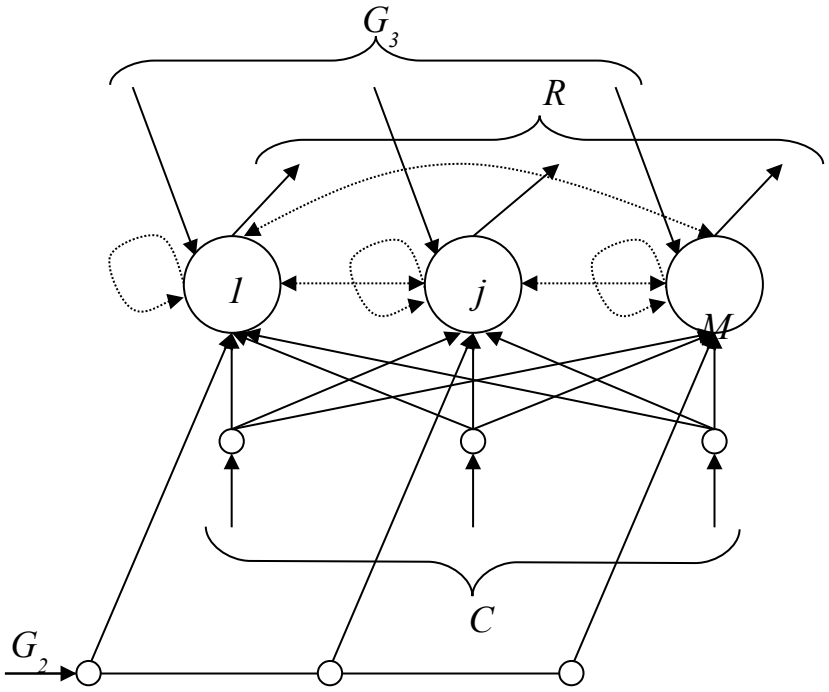


Рис. 20. Структура шару розпізнавання

Особливістю структури шару є наявність негативних зв'язків між сусідніми нейронами та позитивного зв'язку нейрона з самим собою. На рис. 20 ці зв'язки показані пунктиром. Крім того, вхід будь-якого *j*-го нейрону шару розпізнавання з'єднаний з виходом блоку G_2, відповідною вихідною компонентою блоку G_3 та виходом кожного із нейронів шару порівняння.

На рис. 20 останні зв'язки позначені літерою *C,* а виходи нейронів - літерою *R*. Вагові коефіцієнти зв'язків з блоками G_2 та G_3 дорівнюють 1. Вагові коефіцієнти зв'язків з виходами нейронів шару порівняння складають матрицю вагових коефіцієнтів $W^{(2)}$, компоненти якої розраховуються в процесі навчання. Вхід *j*-го нейрону (NET_j^R) шару розпізнавання можливо розрахувати так:

$$NET_j^R = G_2 + G_{3,j} + \sum_{i=1}^{N} \left(C_i \times w_{i,j}^{(2)} \right), \qquad (128)$$

де R_k - вихід *k*-го нейрону шару розпізнавання, $w_{i,j}^{(2)}$ - ваговий коефіцієнт зв'язку між і-им нейроном шару порівняння та *j*-им нейроном

шару розпізнавання, G_{3j} - j-ий компонент матриці зв'язків з блоком G_3, C_i - вихід i-го нейрону шару порівняння.

Наявність в (128) великої негативної складової вектору G_3 дозволяє заморозити функціонування будь-якого нейрону шару розпізнавання. Крім того тільки наявність сигналу $G_2 \neq 0$ призводить до спрацювання нейрону:

$$\begin{cases} R_j = 0, \forall G_2 = 0, \\ R_j = F(NET_j), \forall G_2 = 1. \end{cases} \quad (129)$$

де $F(x)$ - лінійна функція.

Особливістю шару розпізнавання є взаємодія нейронів по схемі "переможець забирає все", відповідно якій в кожен момент часу збуджується тільки один нейрон з найбільшим рівнем активації. Принципово взаємодія відбувається за рахунок використання зв'язків між нейронами в середині шару. В програмній реалізації мережі схему "переможець забирає все" можливо реалізувати іншим способом, більш ефективним з точки зору обчислювальних можливостей. Тому в (128, 129) матриця вагових коефіцієнтів зв'язків між нейронами в середині шару не використовується. Функціонування мережі АРТ можливо розділити на п'ять етапів: ініціалізацію, розпізнавання, порівняння, пошук та навчання.

Під час ініціалізації визначаються:

- Кількість нейронів в шарі порівняння (N) та розпізнавання (M). N - дорівнює розмірності вхідного образу, M - дорівнює кількості навчальних образів

- Величина параметру подібності ρ. Критерієм визначення величини ρ є необхідна ступінь детальності класифікації. Чим більша величина ρ, тим більш подібними мають бути образи, що відносяться до одного класу.

- Матриця вагових коефіцієнтів нейронів шару порівняння:

$$W^{(1)} = 1. \quad (130)$$

- Матриця вагових коефіцієнтів вхідних зв'язків нейронів шару розпізнавання:

$$W^{(2)} < \frac{H}{H - 1 + N}, \quad (131)$$

де H - константа, як правило H=2, N - розмірність вхідного образу.

Етап розпізнавання. На вхід мережі подається нормований вектор X, що відповідає деякому образу. До цього моменту G_2=0, тому R=0. Вектор X містить відмінні від 0 компоненти, а вихід R шару розпізнавання дорівнює 0, тому G_1=1. Шар порівняння активізується відповідно (126) та (127) і сигнал X без змін проходить через шар порівняння на вхід шару розпізнавання - $C=X$. G_2=1, а значить активізується шар розпізнавання. За допомогою (128) розраховуються величини активностей нейронів цього шару та визначається номер нейрону переможця - v, вихід якого дорівнює 1. Виходи всіх інших нейронів дорівнюють 0:

$$\begin{cases} R_i = 1, \forall i = v, \\ R_i = 0, \forall i \neq v. \end{cases} \quad (133)$$

де R_i - i-а компонента матриці виходів нейронів шару розпізнавання.

Етап порівняння. $R \neq 0$, тому G_1=0. Компонента R_v поступає на вхід кожного з нейронів шар порівняння. За допомогою (1.126) розраховується вхід кожного з нейронів. Вихідний сигнал шару, розрахований за допомогою (127), поступає на вхід блоку визначення схожості векторів. Якщо вираз (125) справджується, то приймається рішення про належність вхідного образу X класу v. В протилежному випадку відбувається перехід до етапу пошуку.

Етап пошуку. В шар розпізнавання поступає сигнал G_3, v-а компонента якого дорівнює 1. Завдяки цьому v-й нейрон шару переможця виділяється, тобто вибуває (R_v=0) із процесу класифікації даного вхідного образу. Після цього сигнал шару розпізнавання R=0, що призводить до G_1=1. На вхід мережі знову подається нормований вектор X, який призводить до відновлення етапів розпізнавання та порівняння. Процес повторюється доки не буде виконуватись критерій схожості або не залишиться не використаних нейронів. Доведено, що процес пошуку завжди закінчується на невиділеному нейроні. Якщо в мережі не залишилось не виділених нейронів, тоді вона розширюється за рахунок додавання в шар розпізнавання нового нейрону з відповідною модифікацією зв'язків та зміною блоку визначання схожості векторів.

Етап навчання. Навчання АРТ виконується методом “без вчителя”, відбувається незалежно від результатів розпізнавання та полягає в корекції

вагових коефіцієнтів всіх зв'язків нейрону переможця. Тобто корекції підлягають обидві матриці вагових коефіцієнтів $W^{(1)}$ та $W^{(2)}$. При цьому розрізняють повільне та швидке навчання. Для повільного навчання характерним є пред'явлення мережі вхідних образів на короткий термін часу. Завдяки цьому вагові коефіцієнти не досягають своїх асимптотичних величин. В цьому випадку їх величини визначаються статистичними характеристиками всіх вхідних векторів, а не характеристиками окремого вектору. Динаміка повільного навчання описується диференційними рівняннями виду:

$$w_{i,v}^{(2)} \to \alpha \times \Delta w_{i,v}^{(2)} + (1 - \alpha) \times w_{i,v}^{(2)}, \tag{134}$$

$$w_{v,j}^{(1)} \to \alpha \times C_j + (1 - \alpha) \times w_{v,j}^{(1)}, \tag{135}$$

де α - емпірично розрахований коефіцієнт швидкості навчання, C_j - j-а компонента вектору C, $w_{i,v}^{(2)}$ - ваговий коефіцієнт вхідного зв'язку нейрону переможця з i-им нейроном шару порівняння, $w_{v,j}^{(1)}$ - ваговий коефіцієнт вхідного зв'язку j-го нейрону шару порівняння з нейроном-переможцем.

В випадку швидкого навчання, характерного для практичного використання АРТ, образи подаються на вхід мережі достатньо довго. Вагові коефіцієнти зв'язків нейрону-переможця розраховуються так:

$$w_{i,v}^{(2)} = \frac{H \times C_i}{H - 1 + \sum_{j=1}^{N} C_j}, \tag{136}$$

$$w_{v,j}^{(1)} = C_j, \tag{137}$$

де H - константа, як правило H=2, N - розмірність вхідного образу.

Вагові коефіцієнти $w_{v,j}^{(1)}$ бінарні. При ініціації $w_{v,j}^{(1)}$=1. Змістом швидкого навчання по відношенню до вектора $W^{(1)}$ нейрона-переможця є обнулення несуттєвих компоненти. Якщо ж на деякій ітерації навчання компонента стала рівною 0, то вона вже ніколи не зможе стати рівною 1. Це призводить до серйозних негативних наслідків при навчанні АРТ на зашумлених образах.

Особливістю класифікації мережі АРТ є використання двох критеріїв подібності образів. Перший критерій, використовується на етапі розпізнавання і полягає у визначенні бібліотечного еталону, що є найбільш

схожим на класифікуємий образ. Другий критерій за допомогою шаблону критичних рис порівнює визначений еталон з вхідним образом. В багатьох випадках шаблон критичних рис змінюється в процесі функціонування, що і призводить до необхідності розділу етапів розпізнавання та порівняння. В теорії адаптивного резонансу доводиться можливість стабілізації вказаного шаблону. Після цього реалізація етапу порівняння не змінює результатів класифікації, тобто відбувається прямий доступ до пам'яті мережі.

По відношенню до інших архітектур основною перевагою АРТ є можливість динамічного запам'ятовування нових образів без втрати старої пам'яті та без необхідності довготривалого навчання. До інших позитивних рис АРТ можна віднести швидкий доступ до бібліотечних образів, стабільність та закінченість процесів навчання та розпізнавання, досить короткий термін навчання, зрозумілість функціонування та простоту програмної реалізації. В той же час широкому застосуванню мережі перешкоджає неможливість довготривалої класифікації зашумлених образів, яка призводить до паралічу мережі. Іншими недоліками мережі є чутливість навчання до порядку пред'явлення вхідних векторів та відносно високі обчислювальні затрати в процесі функціонування. Описані недоліки мережі зумовлюють необхідність її модифікації, спрямованої на адекватність класифікації зашумлених образів. Перспективним напрямком модифікації може бути застосування в складі АРТ компонентів НМ інших типів, вдосконалення методики формування шаблону критичних рис.

Розділ 8. Мережі, призначені для розпізнавання змісту тексту

НМ, призначені для обробки текстової інформації, значно відрізняються від мереж, що використовуються при вирішенні інших практичних задач. Причинами відмінностей є як структура вхідної інформації, так і характер задач обробки тексту.

Особливістю структури вхідної інформації є неможливість апріорного визначення довжини текстового фрагменту, що призначений для обробки НМ. Довжина фрагменту може бути довільною. Якщо ж розділити текст на

декілька частин, то в результаті можливо частково або й повністю втратити його зміст. В свою чергу не визначеність довжини викликає труднощі при формуванні структури НМ.

Насамперед, складно розрахувати кількість вхідних нейронів та нейронів, що призначені для обробки тексту. Стандартним шляхом вирішення цієї проблеми є використання такої архітектури НМ, в якій кількість нейронів та зв'язків між ними розраховуються в процесі обробки конкретного текстового фрагменту. Найчастіше застосовується методика динамічної зміни архітектури НМ типу рекурсивної автоасоціативної пам'яті (RAAM).

Характер задач комп'ютерної обробки тексту в багатьох випадках полягає у автоматизованому або автоматичному визначенні змісту документу. При цьому зміст може бути представлений у вигляді:

- Вихідного документу, що в уніфікованій та компактній формі описує основні змістовні атрибути початкового тексту.
- Порівняння з одним або декількома зразками. Результатом порівняння є класифікація початкового документу по наперед визначеному критерію. При автоматизованій обробці така класифікація досить часто реалізується за допомогою НМ з архітектурою на основі карти Кохонена.

Типова методика обробки тексту з метою визначення його змісту полягає в переведенні тексту на природній мові в внутрішнє представлення системи та можливому виведенні із цього представлення нових даних.

Процес обробки реалізується на трьох рівнях:

- морфологічному,
- синтаксичному,
- семантичному.

На морфологічному та синтаксичному рівнях із тексту виділяються окремі слова, які розділяються на окремі морфеми. Після цього визначаються синтаксичні зв'язки між словами. Мета перших двох етапів полягає в отриманні словоформ з відрізаними закінченнями. Кожній словоформі ставиться у відповідність значення певних граматичних ознак.

Метою семантичного аналізу є виявлення змісту слів та

словосполучень. Для цього необхідно використати імітаційну модель предметної області тексту. В процесі семантичного аналізу знання, що закладені в імітаційній моделі порівнюються із фактами, представленими в тексті. Результатом порівняння виступає внутрішнє, з точки зору системи обробки, формалізоване представлення тексту. Тобто визначення змісту тексту це процес співставлення кодованих позначень явищ предметної області, що містяться безпосередньо в тексті з відповідними цим явищам фрагментами імітаційної моделі тієї ж предметної області.

Інформація про синтаксичну структуру тексту є критерієм вирішення можливих неоднозначностей вказаного співставлення. Відзначимо, що обсяг та якість формалізованого представлення тексту залежить від характеру прикладної задачі. Так, в представлення можуть бути включені дані та знання з імітаційної моделі або вилучені деякі несуттєві частини тексту. Таким чином, зміст тексту це семантично зв'язана сукупність фрагментів відповідної імітаційній моделі. При цьому граматична структура речень виражається на деякій формальній мові.

Остаточним результатом аналізу тексту є СМ, яка традиційно вважається найбільш повним та достовірним описом його змісту. Відзначимо, що в класичному розумінні СМ - це форма представлення знань у вигляді графа, вузли якого відповідають фактам або поняттям, а дуги - відношенням або асоціаціям між поняттями. СМ дозволяє абстрагуватись від малоінформативних елементів та представити структуру тексту в термінах описаних в тексті ситуацій (предикатів) та їх учасників (аргументів) в визначених семантичних ролях.

В теперішній час відомо багато різних типів СМ, пристосованих для аналізу змісту текстової інформації. Недоліком більшості з них є складність їх самонавчання в процесі практичного використання. Тому заслуговують на увагу СМ, створені з використання нейромережевих технологій. В них поєднуються переваги СМ та НМ. В теоретичних роботах запропоновано використовувати на всіх етапах розбору тексту СНМ, яку можна розглядати як розвиток активних СМ та НМ Маккаллока-Питтса.

В загальному випадку під поняттям СНМ розуміють мережу динамічно

пов'язаних між собою нейронів, що паралельно або квазіпаралельно виконують операції нечіткої логіки, обмінюються між собою інформацією та організовані в єдине ціле за допомогою визначених механізмів. Як і для інших типів НМ, базові операції обробки даних виконуються окремими нейронами. При цьому для сприйняття СНМ інформації із зовнішнього середовища використовуються рецептори (вхідні нейрони).

Для передачі інформації із мережі на зовні використовуються ефектори (вихідні нейрони). Організуючі механізми можуть мати власну обчислювальну активність та взаємодіяти з нейронами. Нейронам СНМ призначається відповідність деяких елементів семантики предметної області або моделі тексту. При цьому елемент представляє собою окремий символ, сукупність деяких символів тексту, понять і відношень між поняттями, яку можна абстрагувати як єдине ціле. Таким чином, агрегована сукупність елементів може бути представлена як окремий елемент.

Відзначається, що елементами можуть бути всі текстові символи, всі форми всіх слів, що складаються із вказаних символів, словосполучення, речення, абзац, весь фрагмент тексту. Різним етапам обробки тексту повинні відповідати різні рівні агрегації елементів. Наприклад, при морфологічному розборі окремі символи можуть бути агреговані в склади та слова, а при синтаксичному слова можуть бути агреговані в речення.

У випадку наявності відповідного елементу в тексті нейрон приймає значення - “істина”, а в протилежному випаду - “не правда”. Зв'язки між нейронами представляють собою відношення між елементами семантики. Фактори впевненості представляються у вигляді градієнтних величин, що оброблюються і передаються нейронами. Змістом обробленої частини тексту є миттєвий стан частини СНМ, що відповідає за сприйняття інформації із вхідного потоку символів. Миттєвий стан СНМ включає в себе параметри нейронів та зв'язків між ними. Таким чином, змістом тексту, що оброблюється СНМ, є стан цієї мережі.

СНМ можливо розділяти на окремі мережі, що виконують різні функції. Обмін даними між цими під мережами, як і в випадку обміну даними з зовнішнім середовищем, виконується за допомогою нейронів ефекторів та

рецепторів. При цьому нейрон-рецептор розпізнає тільки одну постійну ситуацію.

Рецептор повертає градієнтне значення пропорційне ступню впевненості в тому, що дана величина зустрічається в пред'явленій ситуації. Максимальне значення відповідає повній впевненості в наявності розпізнаної величини в ситуації, мінімальне значення відповідає повній впевненості у відсутності даної величини в ситуації, а середнє значення відповідає повній невизначеності.

Зміст тексту, представлений станом СНМ, оброблюється мережею як потік градієнтних даних, що передається між нейронами. Відзначається, що градієнтні дані представляють собою цілі числа, задані на деякому діапазоні. Якщо провести аналогію з класичними СМ, то вузлам графа мережі будуть відповідати нейрони, а ребрам - зв'язки НМ. Логічне значення “істина”, що оброблюється семантичним графом, представляється в вигляді максимального градієнтного значення, що оброблюється нейроном. Логічне значення “не правда” відповідно представляється у вигляді мінімального градієнтного значення. Логічні операції, що виконуються вузлом графа моделюються в мережі операціями нечіткої логіки, що виконуються нейроном.

Розглянемо СНМ, пристосовану для реалізації на цифровій комп'ютерній техніці. При обробці СНМ текстової інформації з метою проведення морфологічного, синтаксичного та семантичного аналізу нейронам достатньо виконувати такі операції нечіткої логіки як диз'юнкція, кон’юнкція та інверсія. Операція диз'юнкції “$\vee$” виконується нейроном, коли на його виході потрібно отримати істину, якщо хоча б по одному із входів надійшло істинне повідомлення. Операція конюкції “$\wedge$” виконується нейроном, коли на його виході потрібно отримати істину, якщо всі вхідні повідомлення істинні. Операція інверсії “$\neg$” виконується нейроном коли на його виході потрібно отримати істину (неправду), якщо вхідний сигнал хибний (істинний). Операція інверсії є унарною.

Доведено, що СНМ можливо розглядати як формальну мову, що дозволяє оброблювати зміст тексту як функції алгебри логіки, що

складаються із окремих нейронів. Структура СНМ визначає порядок спрацювання нейронів, а отже і порядок обробки вхідних даних базовими операціями алгебри. Відома структура СНМ, яка дістала назву СЛД. При цьому доведено, що НМ з такою структурою дозволяє проводити всі необхідні етапи обробки тексту.

Особливостями СЛД є наявність синхронізованих та несинхронізованих нейронів, а також динамічно змінювана кількість нейронів. Використання різних типів нейронів обумовлене необхідністю синхронізації паралельно протікаючи процесів при реалізації НМ на сучасній комп'ютерній техніці.

Не синхронізовані нейрони безперервно оброблюють вхідну інформацію та видають результати цієї обробки. Синхронізовані нейрони також безперервно видають результати, але оброблюють вхідну інформацію тільки в визначені кванти часу. Момент активізації синхронізованих нейронів визначається за допомогою додаткового синхронізуючого входу. Відзначимо, що опис реалізації синхронізованих та несинхронізованих нейронів виходить за межі опису СЛД.

В СЛД зовнішня інформація подається на рецептори мережі кількість яких відповідає сумарній кількості символів алфавіту природної мови тексту та спецсимволів, призначених для відокремлення слів, речень, абзаців, то що. За один такт роботи НМ рецепторам для розпізнавання подається тільки один символ. Тому вихід тільки одного рецептору буде мати значення “істина”. Далі інформація передається на шар обробки, який містить синхронізовані нейрони, що виконують операцію кон’юнкції та несинхронізовані нейрони, що виконують операції диз’юнкції та інверсії.

Відзначимо, що шар обробки складається із декількох окремих синхронізованих та несинхронізованих шарів нейронів. Після обробки інформація потрапляє до ефекторів. Таким чином, подача окремої букви спричиняє в НМ хвилю активності нейронів, що розповсюджується від рецепторів до ефекторів. Кожному фронту хвилі активації відповідає власний синхронізований шар обробки. При цьому перший шар містить нейрони, що розпізнають першу букву, другий шар розпізнає перші дві букви, третій -

перші три і так далі. По причині того, що спецсимволи використовуються тільки для відокремлення різних елементів тексту та для керування активацією нейронів, загальна кількість шарів обробки дорівнює максимальній кількості букв в одному слові. Наприклад, спецсимвол “нове слово” може використовуватись в якості активаційного сигналу для синхронізованих нейронів.

Фрагмент шару обробки СЛД, призначений для розпізнавання слів “захисту”, “захист”, “хист”, “хисту”, “за”, “з” показаний на рис. 21. Як показано на рис. 21, кожен із нейронів шару обробки є синхронізованим кон’юнктором, що має один вхідний зв'язок з нейроном із попереднього шару та один зв'язок із рецептором, що відповідає поточній букві. Кількість вихідних зв'язків нейронів обробки необмежена.

Для спрощення рисунку на ньому не показано синхронізуючого сигналу “новий символ” та агрегуючих шарів диз'юнкторів та інверторів. Вказаний сигнал ініціюється шаром рецепторів, після розпізнавання ними нового символу. Сигнал подається на синхронізуючі входи всіх нейронів шару обробки та призводить до їх активації. Активація нейронів означає обробку ними вхідних сигналів, які є результатом попереднього кванту активності.

Таким чином, подача нового символу викликає в мережі ефект хвилі обробки. Відзначимо, що в СЛД, показаній на рис. 21, кількість хвиль обробки відповідає кількості шарів обробки і дорівнює 7.

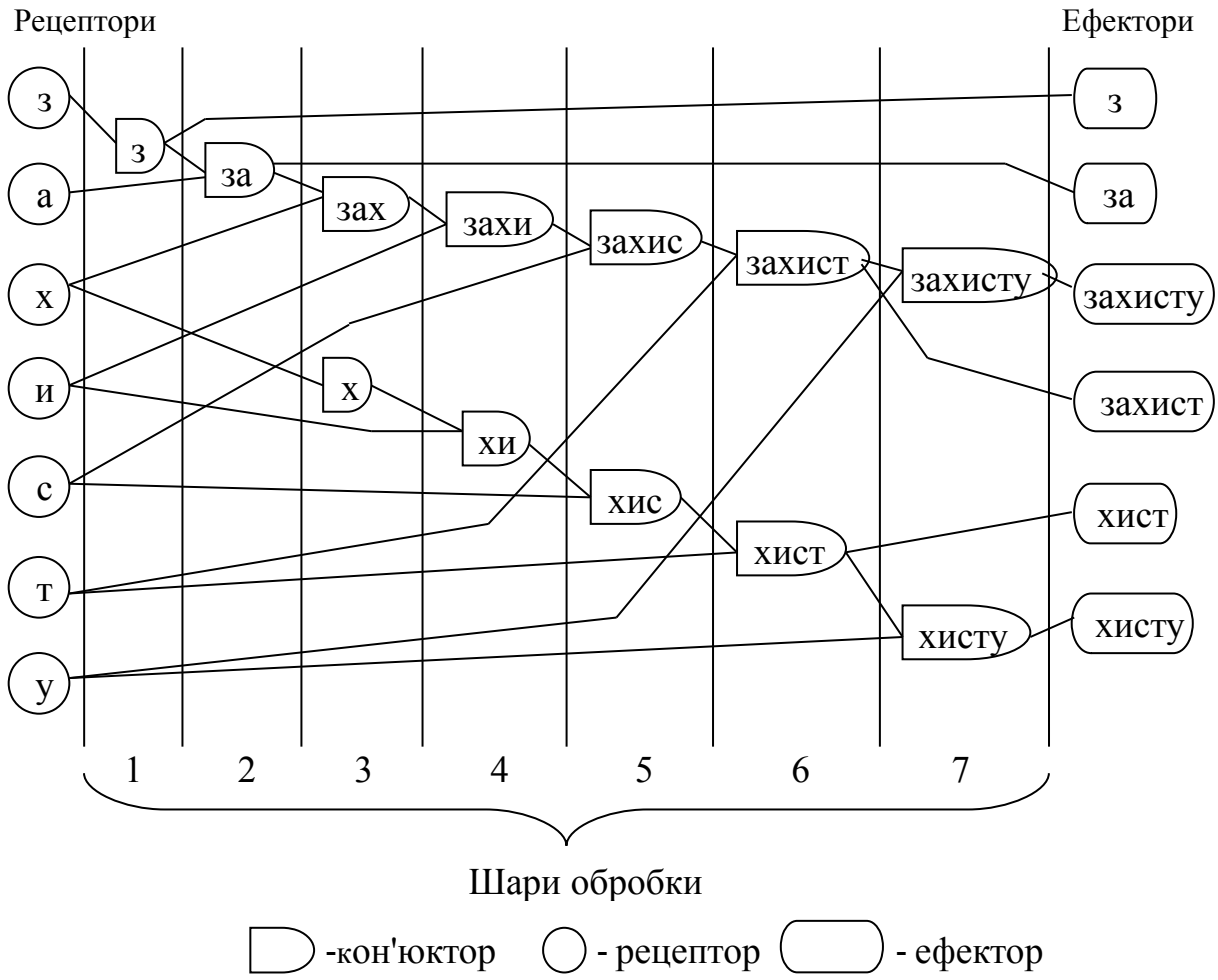


Рис. 21. Фрагмент шарів обробки СЛД

Агрегуючі шари складаються із несинхронізованих нейронів та використовуються в СЛД для класифікації понять, виділених із тексту за допомогою шарів обробки. Агрегуючий шар призначений для часткової класифікації слів “захисту”, “захист”, “хист”, “хисту”, “за”, “з” показаний на рис. 22. По причині простоти задачі застосовується всього один агрегуючий шар, що складається тільки з диз'юнкторів.

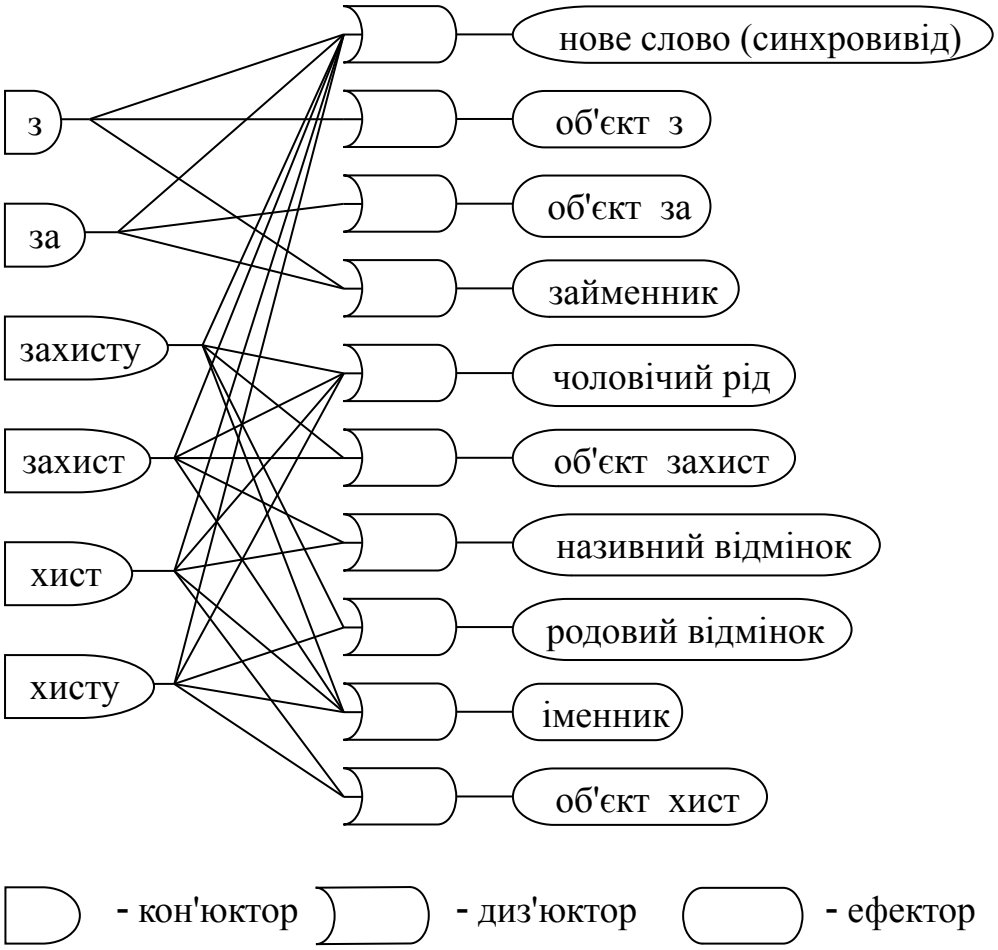


Рис. 21. Структура агрегуючого шару нейронів СЛД

В загальному випадку перед кожним шаром обробки може знаходитись агрегуючий шар нейронів. При цьому всі виходи нейронів шару обробки, що відносяться до одного класу, зв'язані з входами нейронів агрегуючого шару, призначеними для розпізнавання цього класу. Об'єднання послідовних шарів (агрегуючого та обробки) з одним загальним синхронізуючим входом отримало назву СЛД. Для проведення повної обробки тексту можливо використовувати декілька пов'язаних між собою СЛД. За рахунок використання несинхронізованих нейронів в СЛД є можливість одночасного отримання декількох альтернативних рішень. Таким чином, в тексті можливо визначити всі варіанти його змісту.

Для спрощення структури СЛД та підвищення її швидкодії запропоновано об'єднати в одному об'єкті (дизкон'юкторі) несинхронізовані диз'юнктори із підшару агрегування та пов'язані з ними синхронізовані кон'юнктори. Крім того, для спрощення реалізації в СЛД введено групу

нейронів (лінію часу) яка відповідає за збудження дизкон'юкторів в кожну кванті обробки тексту. Всі дизкон'юктори, збуджені в певному кванті обробки тексту, пов'язані з відповідним цьому кванту нейроном лінії часу. За рахунок цього виключається обмін даними між диз'юнкторами, а сам диз'юнктор позбавляється свого внутрішнього стану.

Навчання СНМ з структурою типу СЛД полягає в запам'ятовуванні нею нової інформації. При цьому виділення окремих слів із тексту доцільно проводити за допомогою окремого від НМ препроцесора. В режимі навчання дані посимвольно подаються на вхід СЛД. На кожному такті навчання в НМ проводиться пошук збуджених нейронів. Якщо такі нейрони є, то вважається, що мережа вже навчена даній символьній послідовності і вносити в неї додаткову інформацію не потрібно.

В протилежному випадку, в мережу додаються нові дизкон'юктори, що відразу ж переводяться в збуджений стан. Таким чином, при навчанні СЛД змінюються не тільки вагові коефіцієнти зв'язків, але й сама структура НМ. Відзначимо, що навчальні дані подані на вхід мережі повинні представляти базу даних та базу знань імітаційної моделі предметної області.

Сама імітаційна модель буде сформована в результаті навчання СНМ. В якості бази даних та бази знань імітаційної моделі для СЛД, пристосованих для морфологічного та синтаксичного розбору тексту можуть бути використані граматичні словники мови на якій написано текст. Проведені розрахунки показують, що обсяг сховища даних СЛД для розбору тексту на російській мові становить менше 10 гігабайт. За основу розрахунків було взято розмір одного з найбільш повних граматичних словників природньої мови.

Процес морфологічного та синтаксичного розборів тексту базується на тому, що основою кожної словникової статті є лексема та група пов'язаних з нею словоформ. Під лексемою розуміють форму слова, яка відповідає за основний зміст статті. Пов'язані словоформи створюються із лексеми шляхом словозміни. Наприклад, для лексеми «захищати», пов'язаними словоформами можуть бути захищеність та захищає. Тому в СНМ кожна словникова стаття представляє собою окреме СЛД, що складається із дизкон'юнкторів які

відповідають окремим словоформам.

Лексемі відповідає головний дизкон'юктор СЛД. При успішному розпізнанні будь-якої словоформи, що входить до складу статті крім головного збуджується допоміжний дизкон'юктор, що відповідає цій словоформі. Випадок одночасного збудження декількох допоміжних дизкон'юкторів відповідає морфологічній та/або синтаксичній багатозначності слова, що розпізнається.

Крім дизкон'юкторів, що відповідають окремим словниковим статтям в складі загального СЛД морфологічного та синтаксичного розбору передбачене використання дизкон'юкторів, що відповідають синтаксичним ознакам характерним для всіх статей. До цих ознак відносяться число, час, рід, відмінок. Дизкон'юктор синтаксичної ознаки пов'язаний з дизкон'юнктором словоформи, в якій є ця ознака. Синтаксичний дизкон'юктор збуджуються при збудженні будь-якої словоформи з відповідною ознакою. За рахунок цього можливо полегшити розпізнавання синонімів та омонімів.

Відзначимо, що синонімами називаються слова, що пишуться однаково, але мають різний зміст. Омонімами називаються слова, що пишуться по різному, але мають однаковий зміст. Крім задачі класифікації вхідної символьної послідовності у вигляді відомої словникової статті, СЛД морфологічного та синтаксичного розбору можуть вирішувати задачу формування коректних словозмін вказаної послідовності. Для цього необхідно перевести стани дизкон'юкторів із положень, що відповідають початковій словоформі в положення, що відповідають необхідній словоформі.

Особливістю семантичного аналізу тексту є необхідність використання в СЛД імітаційної моделі предметної області. При цьому представити імітаційну модель можливо представити у вигляді продуктивної експертної системи. Основою таких систем є мережа правил прийняття рішень, в якій окремим правилам відповідають окремі вузли, а елементам, що входять до умов та наслідків правил - зв'язки між вузлами.

Продуктивні експертні системи складаються із двох частин: механізму логічного виводу та бази знань. В спрощеному вигляді механізм логічного виводу визначає які правила виводу будуть використовуватись, а також послідовність цього використання. По відношенню до СЛД механізмом логічного виводу є структура СЛД. База знань експертної системи представляє собою набір логічних правил виду:

$$Якщо\ \Omega = \chi,\ то\ \Psi, \tag{138}$$

де $\Omega,\ \Psi$ - деякі логічні вирази, χ - операнд, що може приймати логічні значення "істина" або "неправда".

В загальному випадку до складу $\Omega,\ \Psi$ можуть входити інші логічні вирази, а при розрахунку складових (138) використовуються операції нечіткої логіки. За рахунок застосування нечіткої логіки підвищується відповідність експертної системи реальним процесам. Відзначимо, що заповнення бази знань відбувається за допомогою експертів в даній предметній області.

Відома методологія перетворення правил (138) у вигляд, придатний для програмування вузлів СЛД. Методологія передбачає розділ правил (138) на елементарні операції диз'юнкції, кон'юнкції та інверсії. Кожній елементарній операції призначається своя група нейронів, яка пов'язується з іншими групами за допомогою дендритів та аксонів. Таким чином, навчання СЛД зводиться до адаптації її структури до правил експертної системи. Зазначено, що блоки СЛД, які відповідають базі знань експертної системи, можуть послідовно включатись один за другим або входити до складу СЛД, які виконують функції морфологічного та синтаксичного розборів.

В своєму закінченому варіанті до заявлених функцій СНМ входить синтез текстової інформації, яка відповідає вхідному текстовому запитанню. В той же час аналіз реалізації СНМ пристосованої для обробки документів фінансової звітності, а також апробація представленого ПЗ вказує на високу якість проведення синтаксичного та морфологічного аналізу, середню якість семантичного аналізу та необхідність додаткових досліджень для отримання синтезованих відповідей. Також слід враховувати що проблема обробки тексту на природній мові далека від вирішення. По цій причині при адаптації СНМ до задач контролю та комп'ютерної діагностики слід піти шляхом

спрощення її архітектури, яка багато в чому відповідає перспективам підвищення якості семантичного аналізу та синтезу текстових відповідей.

Ще однією передумовою спрощення СНМ є постулат про те, що повне представлення змісту тексту в формі СМ є надмірним та непродуктивним для комп'ютерних систем, що мають обмеження на обчислювальні ресурси та швидкодію. Крім того, необхідно розробити більш досконалі методи формування імітаційної моделі прикладної області, що використовується при семантичному аналізі тексту.

В практичній діяльності використання експертів при розробці правил виводу може бути недопустимим по багатьом причинам. Наприклад, при розробці системи розпізнаванні спаму неможливо збирати групу експертів, що визначать критерії класифікації електронних листів кожного користувача. При цьому імітаційна модель може бути побудована самою НМ шляхом її навчання на прикладах, характерних для електронної пошти конкретного користувача.

Розділ 9. Визначення доцільності застосування типу нейронної мережі

Проведений аналіз сучасного стану найромережевих технологій дозволяє сформулювати висновок про те, що доцільність застосування

конкретного типу НМ слід визначати на основі співставлення характеристик мережі з умовами прикладної задачі. До вказаних характеристик та умов відносяться:

1 - параметри навчальних даних,

2 - загальні обмеження процесу навчання,

3 - вимоги до обчислювальних потужностей,

4 - вимоги до вихідної інформації,

5 - обмеження технічної реалізації НМ,

6 - сфера застосування.

Розглянемо вказані характеристики в ракурсі використання НМ в області технічного контролю та захисту інформації в комп'ютерній системі.

1. До основних параметрів навчальних даних відносяться:

- Кількість параметрів, що характеризують навчальний приклад.
- Вид параметрів, дискретний (символьний) чи безперервний (числовий).
- Загальна кількість навчальних прикладів.
- Наявність помилок (шуму) в навчальних прикладах.
- Наявність кореляції навчальних прикладів.
- Можливість та необхідність попередньої обробки вхідних даних з метою їх нормалізації та видалення шуму.
- Повнота виборки, тобто можливість відображення в ній всіх аспектів процесу, що моделюється. Наприклад, чи можливо відобразити в навчальній виборці сигнатури всіх вірусів, або сигнатури мережевих атак певного типу.
- Пропорційність навчальних прикладів, що відповідають різним аспектам процесу, що моделюється.

2. Загальні обмеження процесу навчання обумовлюються:

- Максимальним терміном навчання.
- Необхідністю представлення в навчальних даних очікуваного вихідного сигналу НМ. Цим визначається тип навчання - з вчителем або без вчителя.

- Можливістю автоматизації процесу навчання, яка визначається кількістю та важливістю емпіричних параметрів. Вказана можливість багато в чому визначає умови застосування НМ. Мережі, в яких процес навчання не автоматизовано, можуть використовуватись тільки в лабораторних умовах.
- Можливістю донавчання в процесі експлуатації.
- Вимогами до якості навчання, яке звичайно оцінюють по величині максимальної та середньої помилки розпізнавання навчальних та тестових даних. При цьому тестові дані повинні не значно відрізнятись від навчальних.
- Можливістю навчання НМ в лабораторних умовах. Наприклад, в лабораторних умовах потенційно можливо навчити НМ розпізнавати мережеві атаки певного типу. В той же час неможливо навчити НМ класифікувати електронні листи відповідно інтересам конкретного користувача. Доцільність навчання в лабораторних умовах пояснюється потребами оптимального механізму створення та оновлення бази знань НМ.

3. На практиці вимоги до обчислювальних потужностей визначаються максимальною кількістю прикладів (обсяг пам'яті), яку може запам'ятати мережа для досягнення необхідної достовірності прийняття рішення. В свою чергу достовірність прийняття рішення характеризується допустимими величинами максимальної та середньої помилки мережі на реальних даних які в загальному випадку можуть виходити за межі множини навчальних даних. Відповідно виникає задача екстраполяції результатів навчання НМ за межі навчальних прикладів. Відзначимо, що обчислювальна потужність мережі залежить від її типу та алгоритму навчання. Ще однією вимогою може бути незмінність виходу мережі для різних прикладів з однаковими параметрами.

4. Вимоги до вихідної інформації НМ вказують на те, в якому вигляді має бути представлена ця інформація. Наприклад, при розпізнавані вірусів може виникнути необхідність не тільки визначення ситуації “вірус А присутній”, але й розрахунку ймовірності цієї ситуації. Стосовно класифікації електронних листів вихідною інформацією НМ може бути відображення листів на площину, що дозволить провести остаточну

класифікацію користувачеві. Ще однією вимогою може бути необхідність визначення вербальних залежностей між вхідною та вихідною інформацією.

5. Обмеження технічної реалізації НМ стосуються: швидкості прийняття рішення, інтеграції в існуючі засоби технічного контролю та захисту інформації, обсягу та складності програмної реалізації. Для зменшення обсягу можливо розділити програмний код для навчання мережі від коду, що відповідає за її функціонування.

6. Сфера застосування визначає засоби технічного контролю та захисту інформації, в яких буде використовуватись НМ.

7. На сьогодні достатньо дослідженим є використання НМ для розпізнавання образів та при проведенні оптимізаційних розрахунків. Відзначимо, що системи розпізнавання образів принципово відрізняються від систем аналізу тексту тим, що в них кількість вихідних та кількість комбінацій вхідних параметрів принципово обмежена.

В системах аналізу тексту ця кількість принципово необмежена. Відповідно, в системах визначення атак на комп'ютерну систему та в системах виявлення вразливостей слід використовувати НМ, призначені для розпізнавання образів. В системах захисту від спаму можливо використати НМ, призначені для аналізу тексту. В системах керування параметрами комп'ютерної системи слід застосувати НМ, призначені для проведення оптимізаційних розрахунків.

В перспективі доцільно застосувати НМ з метою реалізації паралельних розрахунків в комп'ютерній системі, що дозволить значно підвищити їх стійкість від багатьох типів атак з метою відмови в обслуговуванні. Крім того, сфера застосування визначається пристосованістю мережі до автономного функціонування. Для цього в архітектурі НМ повинно бути передбачено можливість повної автоматизації процесу донавчання на експлуатації.

Якісні оцінки відповідності основних характеристик НМ умовам задачі технічного контролю та захисту інформації в комп'ютерній системі для описаних в типів мереж наведені в табл. 5.

В табл. 5 відсутні характеристики, які хоча і застосовуються при побудові мережі, але не впливають на вибір типу НМ. Оцінки відповідності виставлені в числовому вигляді по трьохбальній системі: -1 - мінімальна, 0 - середня, 1 - максимальна.

Величини оцінок розраховані в результаті проведеного порівняльного аналізу описаних класичних типів НМ.

Відсутність оцінки означає, що для її визначення потрібні додаткові дослідження.

Таблиця 5

Якісні оцінки відповідності НМ умовам задачі технічного контролю та захисту інформації

Умова	БШП	РБФ	РБФ	АРТ	СНМ	PNN	Асоціа-тивні
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Навчальні дані							
Допустимість шуму	1	0	1	-1	1	0	-1
Допустимість кореляції	1	1	1	1	1	1	-1
Повнота виборки	-1	1	1	-1	-1	1	0
Пропорційність прикладів	1	-1	-1	-1	-1	-1	0
Загальні обмеження процесу навчання							
Короткий термін навчання	-1	0	1	1	0	1	1
Представлення в навчальних прикладах очікуваного виходу	1	1	-1	-1	-1	1	1
Автоматизація навчання	1	-1	0	1	1	1	0

Таблиця 5 (продовження)

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Можливість донавчання	0	1	1	1	1	1	0
Якість навчання	1	0	0	1	1	1	1
Обчислювальні потужності							
Обсяг пам'яті	1	-1	-1	-1		-1	0

Екстраполяції результатів навчання	1	-1	-1	-1		-1	1
Незмінність результатів	1	1	0	1	1	1	0
Вихідна інформація							
Інтерпретації виходу у вигляді ймовірності	0	0	-1	-1	-1	1	0
Інтерпретації виходу у графічному вигляді	-1	-1	1	-1	-1	-1	-1
Можливість вербалізації	1	0	-1	-1	-1	0	-1
Обмеження технічної реалізації НМ							
Швидкості прийняття рішення	1	1	1	1	0	1	-1
Обсяг програмної реалізації	-1	1	-1	0	-1	-1	0
Сфера застосування							
Системи розпізнавання образів	1	1	1	1	0	1	1
Системи аналізу тексту	-1	-1	1	0	1	0	-1
Системи управління	-1	-1	1	-1	-1	-1	1
Автономність функціонування	-1	-1	-1	1	1	-1	-1

Використання даних табл. 5 дозволяє визначити принципову доцільність застосування того чи іншого типу НМ для вирішення практичної задачі. Остаточне рішення про використання конкретного типу НМ із декількох можливих повинно бути прийняте після проведення порівняльних експериментів. Відзначимо, що в задачах, які зводяться до розпізнавання образів при відсутності обмежень на використання методу навчання “з вчителем”, термін навчання, донавчання, автономність функціонування, представлення результатів розпізнавання, обсяг програмної реалізації, кількість та якість навчальних даних найбільш ефективним є використання багатошарового персептрону. Його ефективність пояснюється найбільшою обчислювальною потужністю, можливістю автоматизації процесу навчання та вербалізації отриманих результатів. При цьому інші типи НМ доцільно застосувати для оперативного попереднього аналізу або в специфічних

випадках, що характеризуються певними обмеженнями.

Розглянемо перспективи практичного використання розглянутих типів НМ в області захисту інформації. Можна зробити висновок про те, що основними напрямками застосування НМ в галузі комп'ютерного забезпечення технічних та економічних систем є розпізнавання образів, визначення оптимальних управляючих рішень та створення асоціативної пам'яті. До першого напрямку віднесемо задачі класифікації образів, кластеризації образів та апроксимації функцій. Зазначимо, що до групи задач апроксимації функції слід віднести розрахунок параметрів процесів, що відбуваються в технічних системах. Адже по своїй суті оцінка регресивних або прогнозованих значень параметрів деякого процесу є апроксимацією функції, що описує цей процес. До другого напрямку віднесемо власне задачі оптимального управління та задачі управління з еталонною моделлю. До третього напрямку входять задачі створення інформаційно-обчислювальних систем з пам'яттю, що адресується за змістом.

Оцінювання перспектив використання НМ слід починати із визначення, до якого із вказаних напрямків відноситься поставлена задача захисту інформації. В теперішній час найбільш актуальними та важливими задачами захисту є створення систем виявлення вразливостей, систем виявлення мережевих атак, антивірусів, антикейлогерів, систем протидії спаму та фішінгу, систем управління функціональними параметрами та параметрами безпеки, систем резервування даних.

Типовий алгоритм функціонування відзначених систем захисту такий:

- Проводиться початкова настройка параметрів системи захисту. Як правило, в початкових настройках відображається режим контролю, підконтрольні параметри та деякі параметри захисних заходів. Наприклад, в антивірусних системах може настроюватись період контролю об'єктів файлової системи, режим функціонування постійного захисту, номенклатура заходів проти заражених файлів (блокування, лікування та знищення).

- З визначеною періодичністю реєструються певні параметри комп’ютерної системи. Наприклад, в системах виявлення вразливостей реєструються відкриті порти операційної системи, імена користувачів, версія

операційної системи, права користувачів та ін. В системах виявлення атак можуть реєструватись параметри мережених запитів, обсяг мережевого трафіку, кількість мережевих запитів за певний проміжок часу. В антивірусах та антикейлогерах можуть реєструватись фрагменти програмного коду, що відповідають сигнатурам вірусів та/або програмні події які супроводжують функціонування вірусу (звернення до системного реєстру, звернення до поштової програми) або кейлогеру (перехват натиску клавіш, запис змісту екрану). В антиспамових системах реєструються адреси відправника електронної пошти та окремі слова електронного листа.

- Проводиться первинна обробка зареєстрованих параметрів. Наприклад, в системах виявлення атак підраховується частота мережевих запитів за одиницю часу. В антиспамових системах підраховується частота зустрічі кожного із зареєстрованих слів. При необхідності первинна обробка проводиться до реєстрації або паралельно з нею. Наприклад, в антивірусах та антикейлогерах програмний код при необхідності дешифрується.

- На основі зареєстрованих даних за допомогою спеціального алгоритму приймається рішення про безпеку комп'ютерної системи. Наприклад, в системах виявлення вразливостей це рішення про потенційні вразливості, в системах виявлення атак це рішення про наявність атаки.

- Про виявлену загрозу або вразливість інформується адміністратор комп'ютерної системи.

- Спрацьовує захисний модуль системи. При цьому спочатку приймається рішення про захисний захід, а потім цей захід реалізується. В простих випадках рішення про захисний захід приймається на основі тільки початкових настройок системи, а в складних випадках - за допомогою набору спеціальних правил. Наприклад, заражений вірусом файл необхідно спробувати вилікувати. Якщо ж спроба лікування не вдалась, то файл необхідно знищити. В антиспамових системах електронний лист класифікований як спам може бути знищений або тільки помічений як нецільовий. Водночас адреса відправника цього листа може бути помічена як підозріла або повністю заблокована.

- Якщо в розглянутій системі модуль реакції на загрозу відсутній, то висновок про небезпеку може бути переданий іншій системі, що управляє параметрами захисту. Наприклад, в системах виявлення атак рішення про наявність атаки може бути передане системі захисту від мереженої атаки.

Як свідчить практичний досвід, основні труднощі при розробці вказаних систем полягають у розробці ефективного контуру розпізнавання. З точки зору теорії НМ задачі, що вирішується за допомогою даного блоку відносяться до найбільш дослідженого напрямку розпізнавання образів. Це вказує на беззаперечні перспективи використання НМ для розпізнавання вірусів, кейлогерів, спаму, мережевих атак та вразливостей комп'ютерної системи. Другий та третій напрямок застосування НМ (визначення оптимальних управляючих рішень та створення систем з асоціативною пам'яттю) менш досліджені. Крім зазначених задач визначення захисних заходів в системах виявлення атак, системах виявлення вразливостей, антивірусах, антикейлогерах, системах протидії спаму та фішінгу до другого напрямку можна віднести задачі визначення функціональних параметрів та параметрів політики безпеки конкретної комп'ютерної системи. Наприклад, за допомогою НМ можливо визначити розподіл навантаження декількох комп'ютерів-серверів, необхідність та тривалість блокування ресурсу, захищеного паролем, або необхідність та тривалість блокування доступу до ресурсу з певної IP-адреси.

Відзначимо, що до параметрів політики безпеки слід включити параметри режиму контролю комп'ютерної системи за допомогою конкретного засобу захисту. Наприклад, на практиці доцільно визначити оптимальний період контролю антивірусом та/або антикейлогером конкретної локальної мережі. Третій напрямок застосування НМ може знайти своє відтворення в системах резервного збереження даних та відновлення пошкодженої інформації. Однак на сьогодні недостатня теоретична база перешкоджає практичному використанню НМ для вирішення задач другого та третього напрямків. Крім того, створення ефективної комп'ютерної системи з асоціативною пам'яттю багато в чому є проблемою розробки оригінального, а значить і достатньо дорогого апаратного забезпечення.

Характер застосування НМ при вирішенні деяких актуальних задач захисту інформації в комп'ютерній системі можна оцінити за допомогою даних наведених в табл. 6.

Таблиця 6

Оцінка перспектив використання НМ в системах захисту

Назва системи	Мета застосування НМ	Функціональний блок	Вид задачі
1	*2*	*3*	*4*
Антивірус	Розпізнавання вірусів	Розпізнавання атак (загроз)	Розпізнавання образів
Антикейлогер	Розпізнавання кейлогерів		
Система виявлення атак	Розпізнавання мережевих та локальних атак		
Антиспамова система	Класифікація електронних листів		
Система виявлення вразливостей	Розпізнавання неправильних настройок та параметрів	Розпізнавання вразливостей	
Антивірус	Визначення параметрів протидії розпізнаним вірусам	Прийняття рішення про захисні заходи	Визначення управляючих рішень

Таблиця 6 (продовження)

1	*2*	*3*	*4*
Антикейлогер	Визначення параметрів протидії розпізнаним кейлогерам	Прийняття рішення про захисні заходи	Визначення управляючих рішень
Система виявлення вразливостей	Визначення величини корекції параметрів		

Система виявлення атак (з системою протидії)	Визначення параметрів протидії атаці		
Антиспамова система	Визначення параметрів протидії спаму та підозрілим листам		
Система парольного захисту	Визначення параметрів протидії спробі зламу паролю		
Система захисту від несанкціонованого доступу	Визначення прав користувачів, визначення параметрів протидії спробі несанкціонованого доступу		
Система балансування навантаження серверів	Визначення серверу, який буде виконувати черговий запит		
Система резервування даних	Підвищення живучості даних	Зберігання даних	Система з асоціативною пам'яттю

Післямова

В навчальному посібнику наведено опис, особливості функціонування, переваги та недоліки класичних типів НМ, що можуть використовуватись в якості бази механізмів вдосконалення новітніх нейромережевих рішень. Використано результати авторських досліджень, присвячених вирішенню проблеми підвищення ефективності методів та засобів контролю, управління і захисту комп'ютерних систем за рахунок використання нейромережевих технологій. Базою досліджень послужили результати аналізу теоретичних робіт присвячених загальним принципам функціонування та таким типам НМ як багатошаровий персептрон, мережа радіальних базисних функцій, ймовірнісні мережі (PNN, GRNN), мережі адаптивної резонансної теорії, мережі, що самонавчаються (мережа Ліпмана-Хемінга, карта Кохонена, пружна карта), мережі для розпізнавання змісту тексту та асоціативні мережі (Хопфілда, Хеммінга, Коско).

В результаті аналізу вдалось окреслити механізми побудови та застосування вказаних мереж, визначити обчислювальну потужність, очікувану помилку результатів функціонування, обмеження, традиційні сфери застосування кожної з них. Також описана концепція визначення доцільності застосування конкретного типу НМ для вирішення практичної задачі. Рішення про доцільність використання базуються на результатах співставлення характеристик мережі з умовами прикладної задачі. До вказаних характеристик та умов відносяться: параметри навчальних даних, загальні обмеження процесу навчання, вимоги до обчислювальних потужностей, вимоги до вихідної інформації, обмеження технічної реалізації, сфера застосування. Наведені якісні оцінки відповідності основних характеристик класичних типів НМ умовам поставленої задачі.

Автори висловлюють подяку рецензенту навчального посібника за зауваження та поради, які сприяли покращенню видання.

Список використаної літератури

1. Haykin S. Neural networks. A comprehensive foundations. McMillan College Publ.Co. N.Y., 1994. 696 pp.

2. Барский А.Б. Нейронные сети: распознавание, управление, принятие решений. М.: Финансы и статистика, 2004. 176 с.

3. Бондаренко М.Ф., Осыка А.Ф. Автоматическая обработка информации на естественном языке. К.: УМК ВО, 1991. 140 с.

4. Галушкин А.И. Теория нейронных сетей. М.: ИПРЖР, 2000. 416 с.

5. Вороновский Г.К., Махотило К.В., Петрашев С.Н., Сергеев С.А. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальності. Харьков: Основа, 1997. 112 с.

6. Дорогов А.Ю. Структурный синтез двухслойных быстрых нейронных сетей. Кибернетика и системный анализ. 2000. №4. – С. 47-56.

7. Каллан Р. Основные концепции нейронных сетей. М.: Вильямс, 2003. 288 с.

8. Корченко А., Терейковский И., Карпинский Н., Тынымбаев С. Нейросетевые модели, методы и средства оценки параметров безопасности интернет-ориентированных информационных систем : монография. Київ : Наш Формат, 2016. 273 с

9. Круглов В.В., Борисов В.В. Искусственные нейронные сети. М.: Горячая линия-Телеком, 2002. 382 с.

10. Люгер Ф. Искусственный интелект: стратегии и методы решения сложных проблем, 4-е издание.: Пер. с англ. М.: Вильямс, 2003. 864 с.

11. Міхайленко В. М., Терейковська Л. О., Терейковський І. А., Ахметов Б. Б. Нейромережеві моделі та методи розпізнавання фонем в голосовому сигналі в системі дистанційного навчання : монографія. Київ : Компринт, 2017. 252 с.

12. Терейковский И.А. Использование искусственных нейронных сетей в задачах распознавания атак на компьютерные системы. Захист інформації. 2006. №3. С.57-65.

пошти з використанням нейронних мереж. Захист інформації. 2013. Т. 15, № 2. С. 115-121.

14. Терейковський І. Нейронні мережі в засобах захисту комп'ютерної інформації: монографія. К. : ПоліграфКонсалтинг. 2007. 209 с.

15. Терейковський І., Терейковська Л. Оптимізація захисту відкритих корпоративних мереж. Вісник КНТЕУ. 2004. №1. С.103-112.

16. Терейковський І.А. Вдосконалення методики захисту інформації в корпоративних мережах, що використовують ресурси internet. Вісник національного транспортного університету. 2003. №8. С.13-16.

17. Терейковський І.А. Використання нейронних мереж при розпізнаванні макровірусів. Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. 2006. Випуск 2(13). С.176-183.

18. Терейковський І.А. Використання нейронної мережі Кохонена для розпізнавання спаму. Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. 2007. Випуск 1(14). С.106-114.

19. Хорошко В.О., Кудінов В.А. Методичний підхід до формалізації задачі оцінювання ефективності системи захисту інформаційної системи ОВС України. Захист інформації. 2004. №4. С.11-18.

20. Шуклін Д. Є. Моделі семантичних нейронних мереж та їх застосування в системах штучного інтелекту: 05.13.23.. Дис. …канд. техн. наук. Харків, 2003. 196 с.