

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ БІЗНЕС-КОЛЕДЖ  
кафедра комп'ютерної інженерії та інформаційних технологій

## КВАЛІФІКАЦІЙНА РОБОТА

на тему

---

**ДОСЛІДЖЕННЯ СУЧАСНИХ СИСТЕМ НА ОСНОВІ НЕЙРОПОДІБНИХ  
ОБЧИСЛЕНЬ**

Виконав: студент групи 2к-20  
спеціальності  
123 Комп'ютерна інженерія

Сластьєн Н. В.

Керівник Фальченко Н.Г.

Черкаси 2024

## АНОТАЦІЯ

Кваліфікаційна робота присвячена дослідженню архітектур та методів навчання нейроподібних систем. Особливу увагу приділено дослідженню ефективності різних архітектур у вирішенні конкретних завдань, а також проведено порівняльний аналіз їхніх переваг та недоліків. Окремо було розглянуто впровадження цих систем у транспортну галузь, де було виявлено суттєві переваги нейроподібних систем над традиційними методами.

Однією з ключових переваг нейроподібних систем є їх здатність до навчання та аналізу ситуацій, що дозволяє оптимізувати транспортні потоки набагато ефективніше, ніж стандартні алгоритми. Наприклад, при застосуванні в автопілоті автомобіля нейроподібні системи демонструють значну перевагу завдяки своїй здатності приймати рішення в нестандартних умовах, аналізуючи велику кількість факторів у реальному часі.

Крім того, нейроподібні системи показали високу ефективність у передбаченні та запобіганні аварійних ситуацій, що значно підвищує безпеку на дорогах. Завдяки своїй адаптивності, такі системи можуть швидко реагувати на змінні умови, що робить їх незамінними в сучасних транспортних технологіях.

Таким чином, результати нашого дослідження підтверджують, що нейроподібні системи мають численні переваги над традиційними алгоритмами. Їх впровадження в транспортну сферу може значно покращити ефективність та безпеку транспортних процесів. Висновки нашої роботи свідчать про перспективність подальших досліджень у цьому напрямку, а також про потенціал широкого застосування нейроподібних систем у різних галузях транспорту.

Ключові слова: *нейронна мережа, машинне навчання, архітектура нейроподібної системи*

## ABSTRACT

The qualification work is devoted to the study of architectures and training methods of neural systems. Particular attention is paid to the study of the effectiveness of different architectures in solving specific tasks, as well as a comparative analysis of their advantages and disadvantages. The implementation of these systems in the transport industry was separately considered, where significant advantages of neural systems over traditional methods were revealed.

One of the key advantages of neural systems is their ability to learn and analyze situations, which allows optimizing traffic flows much more efficiently than standard algorithms. For example, when used in a car autopilot, neural systems demonstrate a significant advantage due to their ability to make decisions in non-standard conditions, analyzing a large number of factors in real time.

In addition, neural-like systems have shown high efficiency in predicting and preventing accidents, which significantly increases road safety. Due to their adaptability, such systems can quickly respond to changing conditions, which makes them indispensable in modern transport technologies.

Thus, the results of our study confirm that neural-like systems have numerous advantages over traditional algorithms. Their implementation in the transport sector can significantly improve the efficiency and safety of transport processes. The conclusions of our work indicate the prospects for further research in this direction, as well as the potential for widespread use of neural-like systems in various transport sectors.

**Keywords:** neural network, machine learning, neural-like system architecture

## ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1 ОГЛЯД НЕЙРОПОДІБНИХ СИСТЕМ.....	5
1.1. Види нейроподібних систем.....	5
1.2. Архітектури нейронних мереж.....	8
1.3. Алгоритми навчання нейронних мереж.....	15
Висновки до розділу 1.....	20
РОЗДІЛ 2 АНАЛІЗ СУЧАСНИХ СИСТЕМ НА ОСНОВІ НЕЙРОПОДІБНИХ ОБЧИСЛЕНЬ.....	21
2.1. Системи розпізнавання образів.....	21
2.2. Системи прийняття рішень.....	28
2.3. Порівняльний аналіз.....	35
Висновки до розділу 2.....	42
РОЗДІЛ 3 ЗАСТОСУВАННЯ НЕЙРОПОДІБНИХ СИСТЕМ У ТРАНСПОРТНІЙ ГАЛУЗІ.....	43
3.1. Автономне водіння автомобілем.....	43
3.2. Система оптимізації транспортних потоків.....	47
Висновки до розділу 3.....	51
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53

## ВСТУП

**Актуальність теми дослідження** полягає у тому, що нейроподібні обчислення є одним із найбільш перспективних напрямків у сучасній науці та технологіях. Застосування нейроподібних обчислень охоплює широкий спектр галузей, від розпізнавання образів та мовних перекладів до прогнозування та прийняття рішень у реальному часі. Зокрема, в контексті зростаючої кількості даних та потреби в їхньому аналізі нейроподібні обчислення надають можливість ефективно впоратися з великими обсягами інформації та забезпечити нові рівні розуміння та прогнозування складних явищ. Таким чином, дослідження сучасних систем на основі нейроподібних обчислень є важливим кроком у розвитку науки та технологій, спрямованим на створення більш ефективних та інтелектуальних систем, що відповідають потребам сучасного суспільства.

**Метою цієї дипломної роботи** є систематизація та аналіз сучасних комп'ютерних систем на основі нейроподібних обчислень, а також порівняльний аналіз систем розпізнавання образів та систем прийняття рішень за виділеними категоріями.

### **Завдання дослідження:**

1. Провести огляд архітектури і алгоритмів нейроподібних систем
2. Провести аналіз систем розпізнавання образів
3. Провести аналіз систем прийняття рішень
4. Порівняти алгоритми та архітектури.

**Об'єктом дослідження** є сучасні комп'ютерні системи на основі нейроподібних обчислень. Це включає в себе різноманітні методи та технології, які базуються на імітації принципів роботи людського мозку.

**Предметом дослідження** є архітектури, алгоритми та застосування нейроподібних обчислень. Основна увага зосереджена на вивченні їхньої ефективності та потенціалу в різних сферах, таких як розпізнавання образів, прогнозування та прийняття рішень.

## РОЗДІЛ 1 ОГЛЯД НЕЙРОПОДІБНИХ СИСТЕМ

### 1.1 Види нейроподібних систем

Штучні нейронні мережі (ШНМ) виконують важливу роль у машинному навчанні та штучному інтелекті, демонструючи значний потенціал у рішенні складних завдань, які стосуються аналізу даних, розпізнавання образів, класифікації та прогнозування. Основною особливістю ШНМ є їх здатність до моделювання роботи нейронних систем у мозку людини за допомогою математичних моделей.

Штучні нейронні мережі складаються з набору штучних нейронів, зв'язків між ними та функцій активації. Нейрони у таких мережах виконують роль обчислювальних одиниць, які приймають вхідні сигнали, оброблюють їх та генерують вихідний сигнал, який передається до інших нейронів.

Кожен нейрон отримує вхідні дані, які представлені числовими значеннями, із зваженими зв'язками від інших нейронів. Ці зв'язки мають вагу, яка визначає вплив вхідних сигналів на активність нейрона. Після прийняття вхідних даних, нейрон обчислює взважену суму цих даних та ваг, після чого використовує активаційну функцію для генерації вихідного сигналу. Цей вихідний сигнал служить вхідним сигналом для інших нейронів у мережі.

Цей процес повторюється через всю мережу, дозволяючи їй адаптуватися до різних вхідних даних та приймати відповідні рішення. Взаємодія між нейронами та їхніми зв'язками створює складні обчислювальні структури, які можуть використовуватися для вирішення різноманітних завдань у багатьох галузях, від обробки природних мов до обчислення складних математичних моделей.

Різноманітність архітектур ШНМ включає в себе прямі та рекурентні мережі, а також різноманітні функції активації, такі як сигмоїдальні, гіперболічні тангенси та функції ReLU.

Використання ШНМ в сучасних застосунках, таких як системи автоматичного розпізнавання, рекомендаційні системи, автономні транспортні засоби та фінансовий аналіз, відображає їхню ефективність у розв'язанні складних проблем.

Еволюційний алгоритм - це метод оптимізації, що базується на біологічній метафорі еволюції та використовує природні механізми відбору, рекомбінації та мутації для пошуку оптимальних рішень у просторі можливих варіантів. В основі цього алгоритму лежить популяція індивідів, кожен з яких представляє собою потенційне рішення задачі оптимізації. Початкова популяція генерується випадковим чином або за допомогою певних евристик, після чого застосовуються оператори еволюції для покращення якості рішень у кожній новій популяції.

Процес еволюції складається з кількох ітерацій, або поколінь, під час кожної з яких відбувається оцінка та відбір найкращих індивідів, розмноження з використанням операторів рекомбінації та мутації, а також видалення менш ефективних рішень. Цей цикл продовжується до досягнення критерію зупинки, такого як досягнення заданої якості рішення або вичерпання встановленої кількості ітерацій.

Еволюційні алгоритми широко застосовуються в різних сферах, де потрібно вирішувати складні задачі оптимізації та пошуку рішень. У інженерії та конструюванні вони допомагають знайти оптимальні параметри для дизайну складних систем, включаючи автомобілі, літаки, будівлі та електричні мережі. У фінансовій сфері вони застосовуються для розробки інвестиційних стратегій, оптимізації портфелів та прогнозування руху фінансових ринків. В галузі машинного навчання та штучного інтелекту вони використовуються для навчання моделей та оптимізації параметрів нейронних мереж та інших

алгоритмів. Еволюційні алгоритми також знаходять застосування у біології, транспорті, логістиці, енергетиці, екології та інших галузях. Вони є особливо ефективними в задачах, де немає зручної аналітичної форми цільової функції або коли простір пошуку є складним та має багато локальних оптимумів.

Рекурентні нейронні мережі (РНМ) є надзвичайно потужним інструментом у сфері обробки послідовних даних, здатним ефективно вирішувати завдання, пов'язані з аналізом тексту, розпізнаванням мови, машинним перекладом, генерацією тексту, а також моделями часових рядів та іншими задачами, де важлива часова залежність та контекстуальна інформація.

Принцип роботи РНМ базується на їхній здатності зберігати та використовувати попередні стани для аналізу та обробки нових вхідних даних. Під час кожного кроку обчислень, кожен нейрон в мережі приймає вхідні дані та свій власний попередній стан, який відображає інформацію, отриману на попередніх кроках часу. Цей механізм дозволяє інформації переноситися вперед через часові кроки, що дозволяє мережі враховувати інформацію з попередніх моментів часу при обробці нових даних.

Під час навчання РНМ, ваги нейронів оновлюються з урахуванням різниці між прогнозованим та фактичним значеннями вихідних даних. Цей процес дозволяє мережі вивчати залежності між вхідними даними та попередніми станами, забезпечуючи досягнення бажаного результату.

Такий підхід дозволяє РНМ адаптуватися до широкого спектру задач, забезпечуючи високу точність та ефективність у вирішенні завдань, що вимагають обробки та аналізу послідовних даних у реальному часі.

## 1.2 Архітектури нейронних мереж

Хоча окремі нейрони можуть виконувати прості когнітивні операції, справжня сила нейронної мережі полягає у зв'язках між нейронами. Найпростіші мережі складаються з груп нейронів, що утворюють шари, як показано на правому боці (рис. 1.1). Варто зазначити, що вузли в колі зліва не вважаються частиною шару, оскільки вони використовуються лише для розподілу вхідного сигналу і не виконують жодних обчислень. Ця особливість слугує для того, щоб відрізнити їх від нейронів, які виконують обчислення, позначені квадратиками. Кожен елемент вхідної множини  $X$  з'єднується з кожним штучним нейроном з відповідною вагою. Потім кожен нейрон обчислює зважену суму вхідних даних мережі. Деякі зв'язки можуть бути відсутніми для штучних та біологічних мереж, але для загальності наведено всі зв'язки. Також можуть бути зв'язки між виходами та входами елементів у шарі.

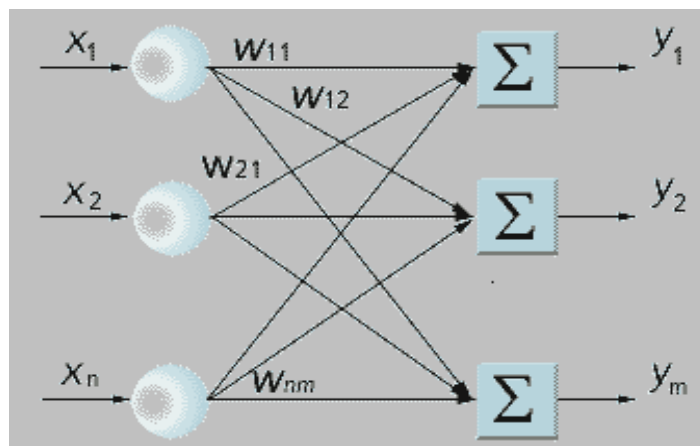


Рисунок 1.1 - Одношарова нейронна мережа

Сучасні великі, складні нейронні мережі мають великий потенціал у вирішенні складних проблем. Їхній успіх частково залежить від здатності відтворювати біологічну організацію мозку, де нейрони розташовані шарами.

Ця пошарова структура, що імітує анатомію мозку, виявилася дуже ефективною у вирішенні найрізноманітніших завдань. Найпоширенішими є багатошарові нейронні мережі, що складаються з шарів (рис. 1.2). У цих мережах вихід одного шару стає входом наступного шару, і відбувається послідовна обробка даних.

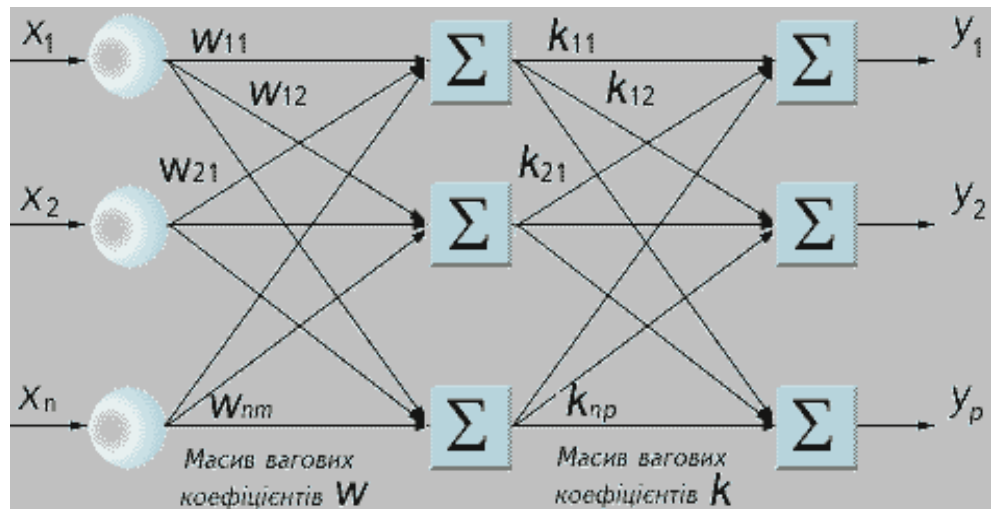


Рисунок 1.2 - Багатошарова нейронна мережа

Підтримка такої структури шарів виявилася ключовим фактором, що гарантує високу обчислювальну потужність таких мереж. В останні роки було досягнуто значного прогресу в розробці алгоритмів навчання та оптимізації багатошарових нейронних мереж. Такий підхід відкриває широкі можливості для використання нейронних мереж для вирішення складних завдань у різних галузях.

Згорткова нейронна мережа (Convolutional Neural Network, CNN) - це тип нейронної мережі, який використовується переважно для завдань обробки зображень, таких як розпізнавання об'єктів, класифікація зображень, розпізнавання облич, виявлення об'єктів і т.д.

Перш ніж ми перейдемо до роботи зі згортковими нейронними мережами, давайте розглянемо що таке зображення, і як воно представлене. RGB-зображення - це ніщо інше, як матриця значень пікселів із трьома

площинами (рис 1.3), тоді як в чорно-білому зображенні вона має одну площину рис.

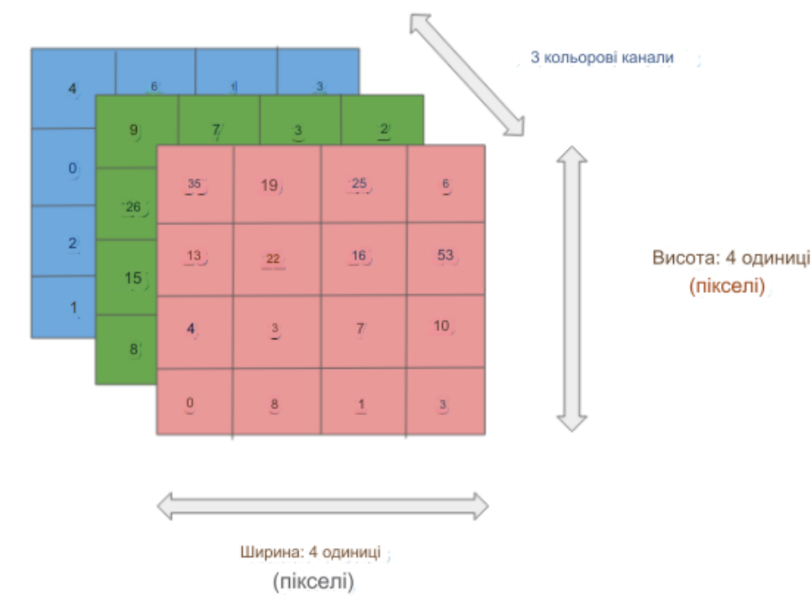


Рисунок 1.3 - Матриця RGB зображення

Для згортки використовується матриця  $3 \times 3$ , яка застосовується до вхідного зображення, щоб отримати згорнуту функцію, яка слідом передається на наступний шар (рис. 1.4).

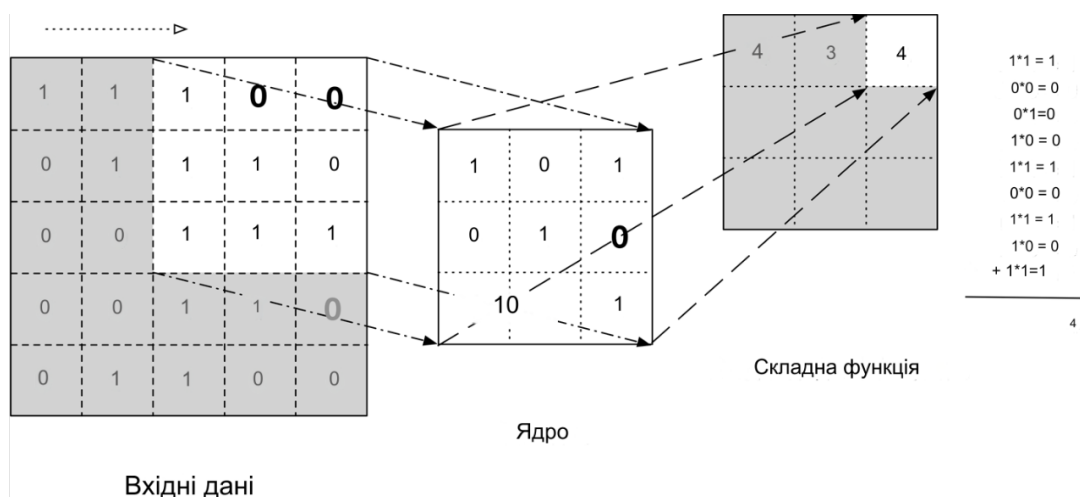


Рисунок 1.4 - Згортка чорно-білого зображення

Перший шар зазвичай виконує функцію екстракції базових ознак, таких

як горизонтальні або діагональні контури. Його вихід передається наступному шару, який спроможний виявляти більш складні особливості, такі як кути або комбіновані контури. Під час проходження через мережу на більш глибоких рівнях вона навчається розпізнавати ще більш складні образи, такі як об'єкти, обличчя та інші визначні атрибути (рис. 1.5), завдяки поступовому збагаченню навчальних фільтрів та активаційних шаблонів.

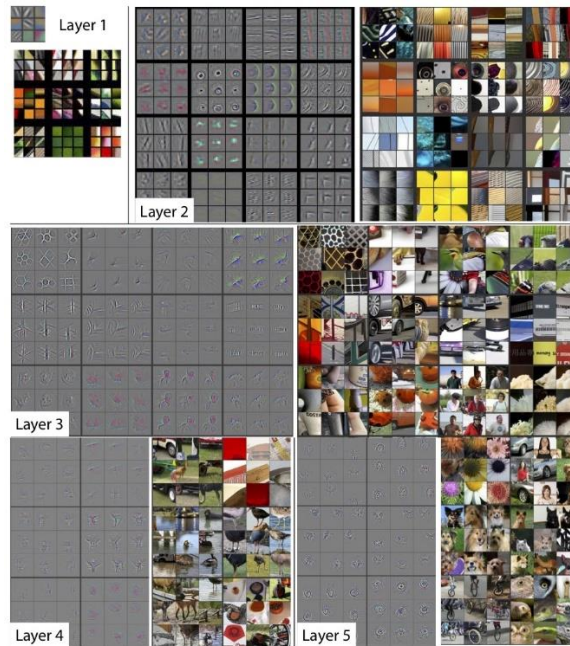


Рисунок 1.5 - Приклад шарів

На основі активаційної карти останнього згорткового шару, шар класифікації формує набір значень довіри (від 0 до 1), що відображають ймовірність належності вхідного зображення до певного "класу". Наприклад, якщо ми використовуємо згорткову нейронну мережу для розпізнавання котів, собак та коней, вихід останнього шару вказує на ймовірність того, що в зображенні присутній будь-який з цих видів тварин. Ці значення довіри можуть інтерпретуватися як ступінь упевненості моделі у відповідності зображення до кожного класу, допомагаючи визначити головний об'єкт або об'єкти на зображенні та їхню кількість (рис. 1.6).

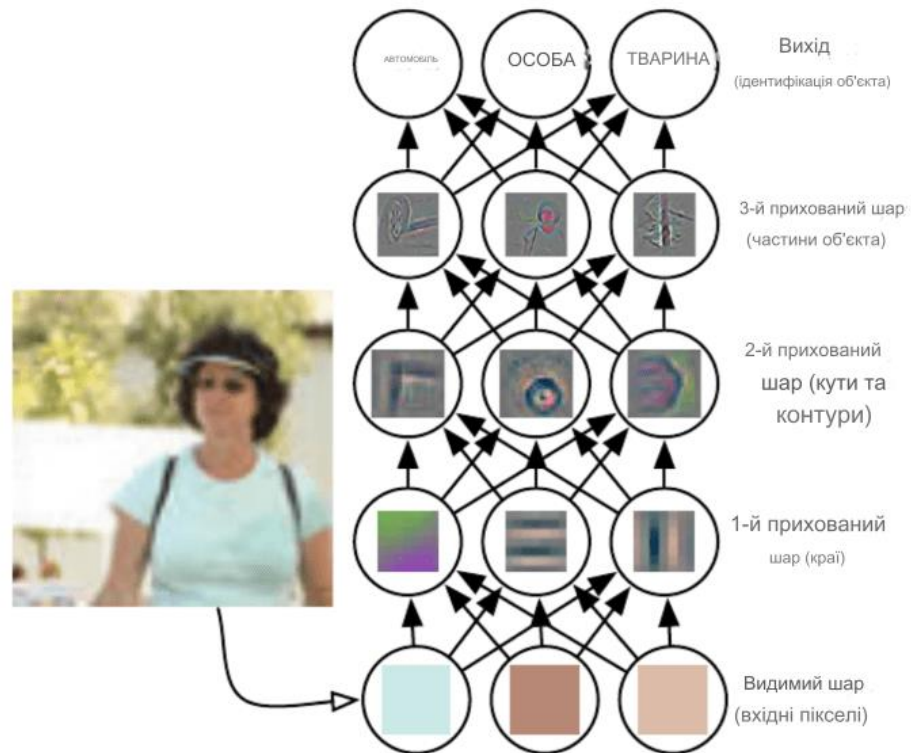


Рисунок 1.6 - Структура згорткової мережі

Різноманітність архітектур нейронних мереж дозволяє моделювати різні типи даних та вирішувати різні завдання, від класифікації до генерації. Кожен тип мережі має свої унікальні характеристики та властивості, які використовуються залежно від конкретної задачі та типу даних.

Нейронна мережа Хопфілда є одним з найвідоміших прикладів рекурентної штучної нейронної мережі. Вона була запропонована Джоном Хопфілдом у 1982 році як модель з метою пошуку рішень для задач асоціативного запам'ятовування та оптимізації.

Основною ідеєю є використання розподіленого представлення даних у вигляді шаблонів, а також можливості мережі відновлювати ці шаблони з неповних або спотворених вхідних даних. У нейронній мережі Хопфілда кожен нейрон пов'язаний з кожним іншим у мережі, створюючи так званий "повністю зв'язаний" шар (рис 1.7).

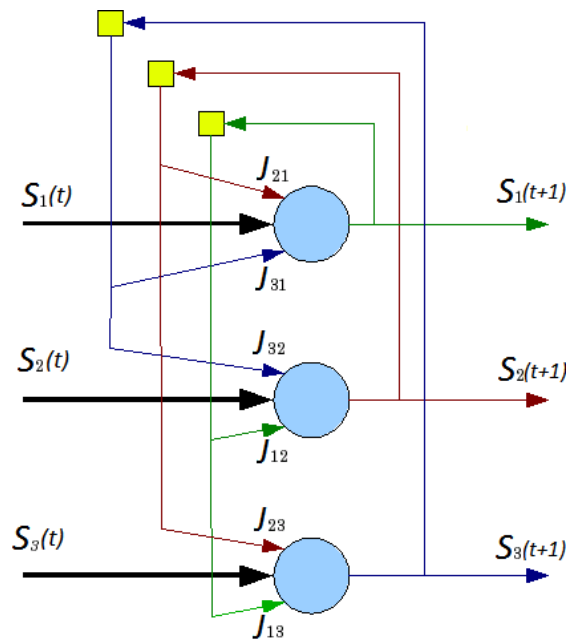


Рисунок 1.7 - Нейронна мережа хопфілда

Кожен нейрон може знаходитись в одному з двох станів:

$$S(t) \in \{-1; +1\}$$

Де  $S(t)$  – стан нейрону в момент  $t$ . Нейрон може приймати два стани: активований (+1) або неактивований (-1).

Зв'язки між нейронами визначаються вагами, які відповідають силі зв'язку між ними. У нейронній мережі Хопфілда ваги є симетричними і визначаються за допомогою правила Гіббса-Больцмана або методу Гайкена-Стродахана.

Процес роботи мережі полягає у послідовному оновленні станів нейронів до досягнення стабільного стану, який відповідає певному запам'ятованому шаблону. Цей процес включає в себе ітеративну подачу сигналів в мережу і корекцію їхніх ваг на основі активованих нейронів та їх зв'язків.

Нейронна мережа Хопфілда знайшла застосування у багатьох областях, зокрема в асоціативному запам'ятовуванні, оптимізації, розпізнаванні образів та розв'язанні комбінаторних проблем. Вона є важливим кроком у розвитку ідеї нейромереж та відіграє значну роль у сучасних дослідженнях з області машинного навчання та штучного інтелекту.

### 1.3 Алгоритми навчання нейронних мереж

Зворотне поширення є ключовим алгоритмом навчання для багат шарових нейронних мереж. Цей метод дозволяє моделі аналізувати складні залежності в наборі даних, шляхом коригування внутрішніх параметрів, що відбувається у зворотньому напрямку від виходу до входу мережі. Завдяки зворотньому поширенню модель може самостійно адаптуватися до різних умов та навчатися вирішувати завдання більш ефективно.

Під час навчання нейронної мережі ми стикаємося з завданням оптимізації, яке полягає у визначенні того, наскільки ефективно мережа може розв'язати поставлену перед нею задачу. Цю ефективність зазвичай оцінюється за допомогою певної оціночної функції, що залежить від вихідного сигналу мережі та всіх її параметрів, включаючи параметри, які визначають її функціонування. Одним із найпоширеніших прикладів такої оціночної функції є сума квадратів відстаней між вихідним сигналом мережі та його очікуваним значенням:

$$H = \frac{1}{2} \sum_{\tau \in v_{out}} (Z(\tau) - Z^*(\tau))^2, \text{ де } Z^*(T) \text{ — необхідне значення вихідного сигналу.}$$

Алгоритм зворотного поширення використовується для навчання багат шарового персептрона. У цій мережі є множина входів  $x_1, \dots, x_n$ , множина виходів Outputs і множина внутрішніх вузлів. Всі вузли, що містять входи та виходи, пронумеровані від 1 до  $N$  (по порядку, незалежно від топології шарів); ваги зв'язків між  $i$ -м та  $j$ -м вузлами позначені через  $W_{i,j}$ , а

вихід  $i$ -го вузла позначений через  $O_i$ . Якщо відомий навчальний приклад з правильними відповідями мережі  $t_k$ . Де  $k \in \text{Outputs}$ , що містить правильний розв'язок мережі, то функція помилки за методом найменших квадратів визначається наступним чином:

$$E(\{w_{i,j}\}) = \frac{1}{2} \sum_{k \in \text{Outputs}} (t_k - o_k)^2$$

Для модифікації ваг ми будемо використовувати стохастичний градієнтний спуск, що передбачає коригування ваг після кожного навчального прикладу. Це означає, що ми будемо просуватись в багатовимірному просторі ваг, спрямовуючись до мінімуму помилки. Для цього нам потрібно коригувати ваги в напрямку, протилежному градієнту функції помилки. Іншими словами, на підставі кожної групи правильних відповідей ми будемо додавати до кожної ваги  $W_{i,j}$  корекцію.

$$\Delta w_{i,j} = -\eta \frac{\partial E}{\partial w_{i,j}}$$

Де  $0 < \eta < 1$  — множник швидкості руху

Цей процес зворотного поширення повторюється для кожного зображення або бачення, що входить у навчальний набір даних, і це відбувається протягом декількох епох навчання. Протягом кожної епохи мережа аналізує велику кількість прикладів, виконує передачу сигналів від входу до виходу, обчислює втрати і коригує свої внутрішні параметри за допомогою алгоритму градієнтного спуску. Цей процес триває досягнення певного рівня функції втрати або до завершення всього процесу навчання.

Зворотне поширення помилки - це ефективний метод навчання, який дозволяє нейронним мережам адаптуватися до складних завдань і робити точні прогнози.

Звичайний градієнтний спуск використовується для оновлення параметрів моделі, обчислюючи градієнт функції втрати відносно всього навчального набору даних. Однак цей підхід може бути надто обчислювально витратним, особливо при роботі з великими обсягами даних. Метод стохастичного градієнтного спуску (SGD) створений, щоб подолати це обмеження, роблячи оновлення параметрів на кожному кроці навчання з використанням лише одного міні-пакета даних. Це робить процес навчання ефективнішим та швидшим.

Давайте розглянемо кроки методу стохастичного градієнтного спуску більш детально:

1. **Ініціалізація параметрів:** Початкові значення параметрів моделі (ваги і зсуви) встановлюються зазвичай випадковим чином або за допомогою певних евристик.
2. **Вибір міні-пакетів даних:** Навчальний набір даних розділяється на міні-пакети фіксованого розміру. Ці міні-пакети випадковим чином обираються з навчального набору на кожному кроці навчання.
3. **Проходження вперед (forward pass):** Кожен міні-пакет даних передається через мережу, і обчислюються вихідні значення для кожного шару, починаючи з вхідного і закінчуючи вихідним.
4. **Обчислення втрат (loss calculation):** Порівнюються прогнозовані значення з фактичними, і обчислюється величина втрат за допомогою функції втрати.
5. **Обчислення градієнтів (gradient computation):** Використовуючи зворотне поширення помилки, градієнти втрати відносно параметрів мережі обчислюються для кожного міні-пакету.
6. **Оновлення параметрів (parameter update):** Параметри моделі оновлюються в напрямку, протилежному градієнту, з використанням

швидкості навчання (learning rate). Кожен параметр оновлюється окремо для кожного міні-пакету.

- 7. Повторення процесу (repeat):** Ці кроки повторюються для кожного міні-пакету даних протягом однієї епохи навчання. Після завершення кожної епохи може виконуватися оцінка продуктивності моделі на валідаційному наборі даних, і, за потреби, параметри можуть бути скориговані.

Метод стохастичного градієнтного спуску (Stochastic Gradient Descent, SGD) - це один із способів оптимізації параметрів моделі шляхом мінімізації функції втрат. Він базується на ідеї крокового руху в напрямку негативного градієнту функції втрат з метою знаходження локального мінімуму. Основна відмінність SGD від класичного градієнтного спуску полягає в тому, що на кожному кроці оновлення ваг використовується лише випадковий піднабір даних (міні-пакет), що дозволяє швидше знаходити локальний мінімум, особливо в великих наборах даних.

Для ефективності і збіжності методу SGD важливо правильно налаштувати два основні параметри: швидкість навчання (learning rate) і розмір міні-пакетів (batch size). Швидкість навчання визначає, наскільки швидко змінюються ваги моделі під час навчання, тоді як розмір міні-пакетів визначає, скільки прикладів даних використовується для обчислення градієнту на кожному кроці. Оптимальні значення цих параметрів зазвичай визначаються емпірично та залежать від конкретного завдання та властивостей даних.

Метод моменту (Momentum) є інноваційним підходом до оптимізації, що вирішує деякі проблеми, які можуть виникнути при застосуванні простого стохастичного градієнтного спуску (SGD). Основна проблема SGD - це повільна збіжність та можливість застрягання в локальних мінімумах.

Метод моменту розв'язує ці проблеми, використовуючи інерцію. У кожному кроці навчання вектор градієнту додається до попереднього вектора

градієнту з певним коефіцієнтом, який відображає швидкість накопичення моменту (інерції). Це означає, що параметри моделі будуть оновлюватися з урахуванням не тільки поточного градієнту, але й попередніх його значень, що дозволяє прискорити навчання та уникнути застрягання в локальних мінімумах.

Одним із ключових переваг методу моменту є його здатність зберігати інформацію про напрямок градієнту з попередніх кроків. Це допомагає уникнути різких змін в параметрах моделі, що можуть спричинити нестабільність навчання. Замість цього, метод моменту забезпечує більш плавний та стабільний процес оптимізації.

Крім того, метод моменту дозволяє врахувати швидкість навчання для кожного параметра окремо. Це означає, що для швидко змінюваних параметрів швидкість навчання може бути збільшена, тоді як для менш змінюваних параметрів вона може бути зменшена, що дозволяє більш ефективно збігатися до оптимуму.

У цілому, метод моменту є потужним інструментом оптимізації, який доповнює методи стохастичного градієнтного спуску, дозволяючи навчанням нейронних мереж прискорити та стабілізувати свій процес.

Регуляризація є важливим методом управління перенавчанням (overfitting) у нейронних мережах. Вона полягає в додаванні додаткових обмежень або штрафів до функції втрати для зменшення ваг параметрів моделі.

Одним з найпоширеніших методів регуляризації є L1 та L2 регуляризація. L1 регуляризація додає штраф до функції втрати, що пропорційний абсолютній величині параметрів моделі, тоді як L2 регуляризація використовує квадрат величини параметрів. Ці методи допомагають уникнути перенавчання шляхом зменшення абсолютних значень параметрів.

Крім того, існують інші методи регуляризації, такі як dropout, який випадковим чином вимикає деякі нейрони в мережі під час навчання, тим самим змушуючи мережу використовувати більше різних шляхів для навчання.

Регуляризація є важливим інструментом управління складністю моделей і забезпеченням їх загальної здатності до нових даних. Цей підхід допомагає покращити загальну здатність моделі до уникнення перенавчання і покращення її загальних результатів на тестових даних.

## **Висновок до розділу 1**

У цьому розділі було проведено детальний огляд нейроподібних систем, що охоплює їх різновиди, архітектури та алгоритми навчання. Ми розглянули основні типи нейроподібних систем, зокрема штучні нейронні мережі, еволюційні алгоритми та рекурентні нейронні мережі. Кожен з цих типів відіграє важливу роль у сучасних дослідженнях та застосуваннях штучного інтелекту, забезпечуючи різноманітні підходи до вирішення складних задач.

Далі ми проаналізували архітектури нейронних мереж, включаючи одношарові, багатшарові, згорткові нейронні мережі та мережі Хопфілда. Кожна з цих архітектур має свої особливості та застосування. Одношарові мережі є основою для розуміння більш складних структур, багатшарові мережі підвищують обчислювальні можливості, згорткові нейронні мережі спеціалізуються на обробці зображень, а мережа Хопфілда використовується для моделювання асоціативної пам'яті.

Огляд алгоритмів навчання нейронних мереж включав розгляд зворотнього поширення, методу стохастичного градієнтного спуску (SGD),

методу моменту (Momentum) та технік регуляризації. Ці алгоритми є ключовими для ефективного навчання та оптимізації нейронних мереж, забезпечуючи точність та стабільність у їх роботі.

Таким чином, цей розділ надав всеохоплюючий огляд нейроподібних систем, що дозволяє зрозуміти основи та підходи до створення і навчання нейронних мереж. Подальші розділи будуть зосереджені на детальному аналізі конкретних аспектів та застосувань цих систем у різних галузях.

## РОЗДІЛ 2 АНАЛІЗ СУЧАСНИХ СИСТЕМ НА ОСНОВІ НЕЙРОПОДІБНИХ ОБЧИСЛЕНЬ

### 2.1 Системи розпізнавання образів

Розпізнавання шаблонів аналізує вхідні дані з метою виявлення закономірностей. Визначення регулярностей у даних може використовуватися для прогнозування, категоризації інформації та вдосконалення процесів прийняття рішень. Під час дослідницького розпізнавання образів здійснюється загальний пошук закономірностей у даних, тоді як описове розпізнавання образів починається з категоризації виявлених образів. Таким чином, розпізнавання образів охоплює обидва ці сценарії, і різні методи розпізнавання застосовуються залежно від конкретного випадку та форми даних. Відповідно, розпізнавання образів - це не один метод, а широка сукупність знань і технік, які часто мають слабкі зв'язки між собою. Здатність до розпізнавання образів є часто необхідною умовою для створення інтелектуальних систем. Основу системи розпізнавання образів становлять комп'ютерні алгоритми, призначені для аналізу та інтерпретації даних. Вхідними даними можуть бути слова або тексти, зображення чи аудіофайли. Таким чином, розпізнавання образів є ширшим поняттям у порівнянні з комп'ютерним зором, який зосереджується на розпізнаванні зображень. Автоматичне та машинне розпізнавання, опис, класифікація і групування шаблонів є важливими завданнями в різних інженерних і наукових дисциплінах, включаючи біологію, психологію, медицину, маркетинг, комп'ютерний зір та штучний інтелект.

Образ - це концепція, яка визначається як об'єкт або сутність, яка має повторювані характеристики або властивості у різних формах даних. У 1985 році Сатосі Ватанабе описав образ як "протилежність хаосу", що може бути ідентифіковано та названо. Це може бути будь-що, від відбитка пальця до зображення обличчя (рис 2.1). Образ може бути спостережуваним як фізично, так і математично, залежно від типу даних, які він представляє. Його

важливість полягає в тому, що він надає змогу впізнавати та ідентифікувати певні об'єкти або явища у навколишньому середовищі.



Рисунок 2.1 - Приклади образів

Статистичне розпізнавання образів є важливим напрямком в області штучного інтелекту та машинного навчання. Цей підхід використовує статистичні методи для аналізу та інтерпретації образів або даних, що представлені у вигляді зображень. Основною метою статистичного розпізнавання образів є автоматичне визначення та класифікація об'єктів, що представлені у вигляді зображень, на основі їх характеристик та структури.

Цей процес включає в себе кілька етапів, включаючи попередню обробку даних, витягування характеристик, вибір моделі та класифікацію (рис. 2.2) Під час попередньої обробки зображень зазвичай використовуються методи фільтрації та підготовки даних для видалення шуму та підвищення якості образів. Наступним етапом є витягування характеристик, під час якого алгоритми аналізують особливості зображення та виділяють різні ознаки, які можуть бути корисними для подальшої класифікації.

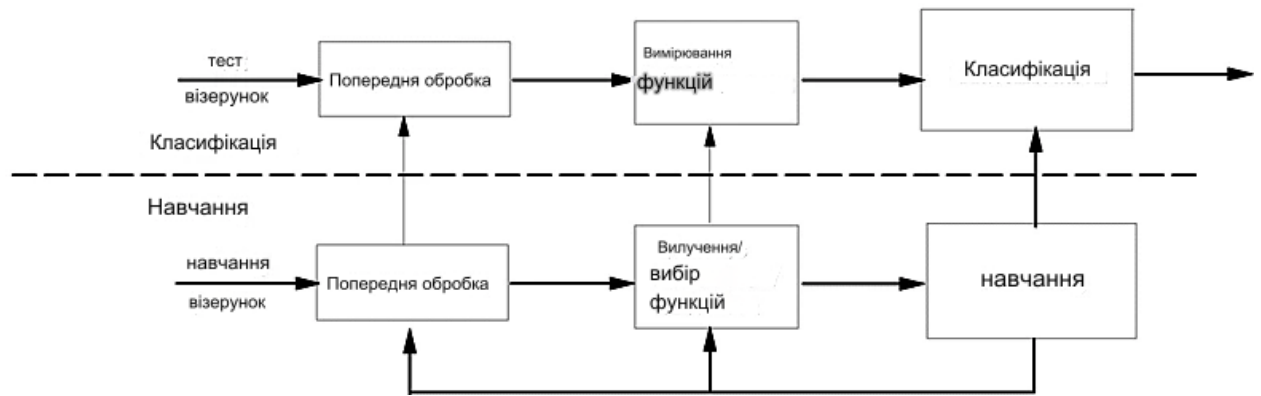


Рисунок 2.2 - Схема роботи статистичного розпізнавання образів

Після витягування характеристик вибирається відповідна модель, яка може включати в себе різні методи машинного навчання, такі як згорткові нейронні мережі (CNN), метод опорних векторів (SVM) чи навчання з учителем. Нарешті, класифікація зображень відбувається шляхом подальшої обробки та порівняння характеристик об'єктів з відомими шаблонами або класами.

Щодо застосування, то він успішно використовувався для вирішення таких проблем, як статистичне розпізнавання образів, розпізнавання рукописного вводу та автоматична медична діагностика. У випадку розпізнавання рукописного тексту алгоритм працює, витягуючи функції та порівнюючи їх із поточними параметрами моделі. Той самий підхід використовується для вирішення більш складних проблем, таких як класифікація зображень, і глибоке навчання може замінити традиційні методи, такі як дискримінантний аналіз. Подібний метод також використовується в системах штучного зору для ідентифікації об'єктів на зображеннях та їх класифікації за певними критеріями.

Зіставлення шаблонів є технікою машинного зору на високому рівні, що виявляє частини на зображенні, що відповідають попередньо визначеному шаблону. Сучасні алгоритми зіставлення шаблонів можуть знаходити відповідності шаблону незалежно від їх орієнтації та місцевої яскравості.

Техніки зіставлення шаблонів є гнучкими та відносно простими для використання, що робить їх одними з найбільш популярних методів локалізації об'єктів. Їх застосовність переважно обмежена наявною обчислювальною потужністю, оскільки ідентифікація великих та складних шаблонів може займати багато часу.

Наївне зіставлення шаблонів. Уявімо, що ми маємо зображення роз'єму і наше завдання - знайти його контакти. Ми маємо шаблонне зображення, яке представляє еталонний об'єкт, що ми шукаємо, та вхідне зображення для аналізу. Ми будемо проводити пошук у досить простий спосіб – розміщуватимемо шаблон на зображенні в усіх можливих положеннях і кожного разу обчислюватимемо числову міру схожості між шаблоном і сегментом зображення, з яким він наразі перекривається. Врешті-решт, ми визначимо положення, які показують найвищі показники схожості, як ймовірні місця розташування шаблону.

Однією з підзадач, що виникає в описі вище, є обчислення міри схожості між вирівняним зображенням-шаблоном і перекритим сегментом вхідного зображення, що еквівалентно обчисленню міри схожості двох зображень однакових розмірів. Це класичне завдання, і числову міру схожості зображень зазвичай називають кореляцією зображень.

Основний метод обчислення кореляції зображень називається взаємною кореляцією. Він полягає у простому додаванні попарних множень відповідних значень пікселів зображень. Хоча можна помітити, що значення кореляції дійсно відображає схожість порівнюваних зображень, метод взаємної кореляції далекий від досконалості. Його основний недолік полягає в

чутливості до змін глобальної яскравості зображень – збільшення яскравості одного зображення може значно підвищити його взаємну кореляцію з іншим зображенням, навіть якщо друге зображення зовсім не схоже.

$$\text{Cross-Correlation}(\text{Image1}, \text{Image2}) = \sum_{x,y} \text{Image1}(x, y) \times \text{Image2}(x, y)$$

Нормалізована взаємна кореляція (НВК) є вдосконаленою версією класичного методу взаємної кореляції, яка забезпечує два важливі покращення (рис.2.3):

$$\text{NCC}(\text{Image1}, \text{Image2}) = \frac{1}{N\sigma_1\sigma_2} \sum_{x,y} (\text{Image1}(x, y) - \overline{\text{Image1}}) \times (\text{Image2}(x, y) - \overline{\text{Image2}})$$

- 1. Незалежність від змін глобальної яскравості:** Результати НВК не залежать від постійного збільшення або зменшення яскравості будь-якого зображення. Це досягається шляхом віднімання середньої яскравості зображення від кожного значення пікселя, що дозволяє уникнути впливу глобальних змін освітлення.
- 2. Масштабування кінцевого значення кореляції:** Кінцеве значення кореляції масштабується в діапазоні від -1 до 1. Таким чином, НВК двох ідентичних зображень дорівнює 1.0, а НВК зображення та його негативу дорівнює -1.0.

Зображення 1	Зображення 2	
		-0,417
		0,553
		0,844

Рисунок 2.3 - Нормалізована крос-кореляція

Повернемося до нашої задачі. Після введення нормалізованої взаємної кореляції (НВК) як надійної міри схожості зображень, ми тепер можемо

визначити, наскільки добре шаблон підходить для кожної з можливих позицій. Ми можемо представити результати у вигляді зображення, де яскравість кожного пікселя відображає значення НВК для шаблону, розташованого над цим пікселем (чорний колір представляє мінімальну кореляцію -1.0, білий колір представляє максимальну кореляцію 1.0).

На цьому етапі залишається визначити, які точки на зображенні кореляції шаблону є достатньо відповідними, щоб вважатися справжніми збігами (рис. 2.4). Зазвичай ми вважаємо збігами ті позиції, які одночасно відповідають двом критеріям:

1. Кореляція шаблону перевищує певне попередньо задане порогове значення (наприклад, вище 0.5).
2. Є локальними максимумами (сильніші за кореляцію шаблону в сусідніх пікселях).

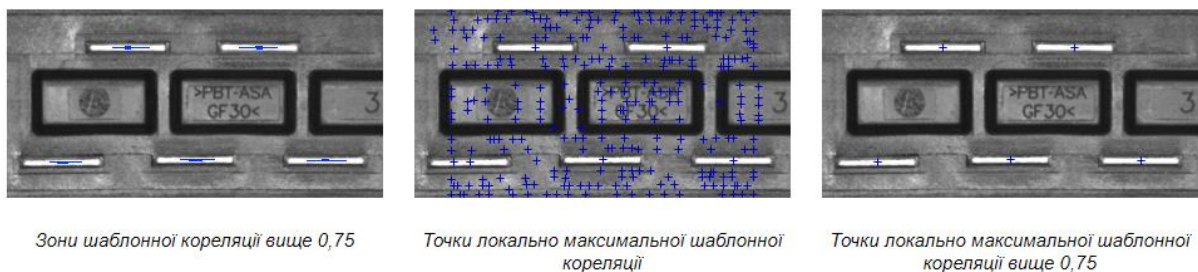


Рисунок 2.4 - Ідентифікація збігів

Зображення-піраміди можна використовувати для підвищення ефективності виявлення шаблонів на основі кореляції. Важливою є спостереження, що шаблон, зображений на еталонному зображенні, зазвичай залишається розпізнаваним навіть після значного зменшення розміру зображення (хоча, природно, при цьому втрачаються дрібні деталі). Тому ми можемо визначити кандидатів на збіг у зменшеному (і тому набагато швидшому для обробки) зображенні на найвищому рівні нашої піраміди, а

потім повторити пошук на нижчих рівнях піраміди, кожного разу розглядаючи лише ті позиції шаблону, які отримали високі оцінки на попередньому рівні.

На кожному рівні піраміди нам потрібно відповідно зменшене зображення еталонного шаблону, тобто повинні бути обчислені як піраміда вхідного зображення (рис. 2.5), так і піраміда зображення шаблону.

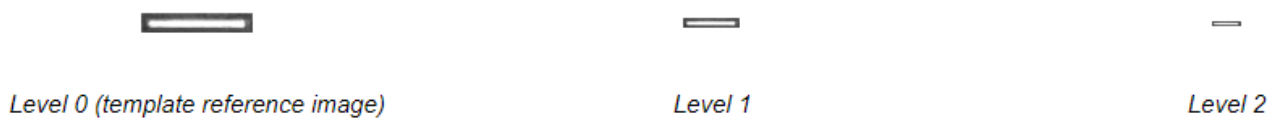


Рисунок 2.5 - Приклад піраміди шаблону

## 2.2 Системи прийняття рішень

Системи прийняття рішень (СПР) є невід'ємною частиною сучасних комп'ютерних технологій, що використовуються для підтримки процесу прийняття рішень в різних галузях. Ці системи розроблені для допомоги користувачам у прийнятті складних рішень шляхом аналізу великої кількості даних, ідентифікації варіантів і надання рекомендацій на основі цих даних.

Процес прийняття рішень включає в себе вибір найкращого курсу дій з кількох можливих альтернатив на основі певних критеріїв. Цей процес може бути досить складним, особливо коли необхідно враховувати численні фактори та величезні обсяги інформації. Системи прийняття рішень надають структурований підхід до аналізу цих даних і допомагають у визначенні найбільш оптимальних рішень.

Нейроподібні системи, зокрема нейронні мережі та алгоритми глибокого навчання, відіграють ключову роль у сучасних системах прийняття рішень. Завдяки своїй здатності навчатися на великих наборах даних і

виявляти складні закономірності, ці системи можуть забезпечувати високу точність і ефективність у процесі прийняття рішень. Нейроподібні системи використовуються для прогнозування, класифікації та оптимізації різних процесів, що дозволяє покращити якість рішень та скоротити час, необхідний для їх прийняття.

Таким чином, системи прийняття рішень на основі нейроподібних обчислень представляють собою потужний інструмент, здатний значно покращити процес прийняття рішень у різних сферах людської діяльності, роблячи його більш точним, швидким та надійним.

Системи прийняття рішень (СПР) можна класифікувати за різними критеріями, такими як функціональність, структура, методи обробки даних та область застосування. В цьому підрозділі розглянемо основні типи систем прийняття рішень: експертні системи, системи підтримки прийняття рішень (DSS) та інтелектуальні агентні системи.

Експертні системи (ЕС) є одним із найстаріших типів СПР, розроблених для моделювання процесу прийняття рішень людськими експертами. Вони використовують базу знань, яка складається з правил, фактів та евристик, отриманих від експертів у певній галузі. Основні компоненти експертних систем включають базу знань, механізм виведення та інтерфейс користувача. База знань зберігає факти та правила, що використовуються для прийняття рішень. Механізм виведення застосовує правила бази знань для аналізу ситуацій і генерації рекомендацій, а інтерфейс користувача забезпечує взаємодію з користувачем, дозволяючи вводити дані та отримувати результати. Експертні системи ефективно використовуються в медичній діагностиці, технічній підтримці та інших галузях, де необхідні глибокі спеціалізовані знання.

Системи підтримки прийняття рішень (Decision Support Systems, DSS) призначені для допомоги менеджерам та іншим користувачам у прийнятті

складних рішень на основі аналізу великих обсягів даних. Вони включають базу даних, модельну базу, програмний модуль та інтерфейс користувача. База даних зберігає релевантні дані з різних джерел. Модельна база містить моделі для аналізу даних, такі як статистичні моделі, моделі оптимізації та симуляційні моделі. Програмний модуль забезпечує інструменти для аналізу даних та генерації звітів, а інтерфейс користувача дозволяє користувачам взаємодіяти з системою, виконувати аналіз та отримувати результати. DSS широко використовуються у фінансовому аналізі, плануванні ресурсів, управлінні ланцюгами поставок та інших галузях, де необхідно приймати обґрунтовані рішення на основі великої кількості даних.

Інтелектуальні агентні системи (IAS) є більш сучасними СПР, які використовують автономні програмні агенти для виконання певних завдань та прийняття рішень. Ці агенти можуть працювати незалежно або співпрацювати з іншими агентами для досягнення спільної мети. Основні характеристики IAS включають автономність, адаптивність, комунікативність та цільову орієнтацію. Автономність означає, що агенти можуть самостійно виконувати завдання без втручання користувача. Адаптивність передбачає здатність агентів навчатися та адаптуватися до змін у середовищі. Комунікативність забезпечує можливість взаємодії з іншими агентами та системами, а цільова орієнтація означає, що агенти діють відповідно до визначених цілей та стратегій. Інтелектуальні агентні системи застосовуються в різних сферах, таких як електронна комерція (персоналізовані рекомендації), мережеве управління (оптимізація трафіку) та робототехніка (координація дій між роботами).

Таким чином, класифікація систем прийняття рішень відображає різноманіття підходів і технологій, що використовуються для підтримки процесу прийняття рішень у різних галузях. Кожен тип СПР має свої унікальні особливості та переваги, що дозволяють ефективно вирішувати специфічні завдання.

Алгоритми прийняття рішень у нейроподібних системах відіграють ключову роль у забезпеченні ефективності та точності цих систем. Нейроподібні системи, зокрема нейронні мережі, використовують різні підходи до прийняття рішень, які дозволяють їм адаптуватися до змінних умов і обробляти великі обсяги даних. У цьому підрозділі розглянемо основні алгоритми, що використовуються в нейроподібних системах для прийняття рішень: дерева рішень, баєсові мережі, підсилювальне навчання та алгоритми глибокого навчання.

Дерева рішень є одним із найпоширеніших методів у нейроподібних системах для прийняття рішень. Це алгоритми машинного навчання, які використовують модель у вигляді дерева для прийняття рішень на основі вхідних даних (рис. 2.6). Кожна внутрішня вершина дерева відповідає за перевірку певного атрибута, кожна гілка представляє результат цієї перевірки, а кожна листова вершина відповідає за кінцеве рішення. Дерева рішень прості у візуалізації та інтерпретації, що робить їх ефективним інструментом для багатьох застосувань.

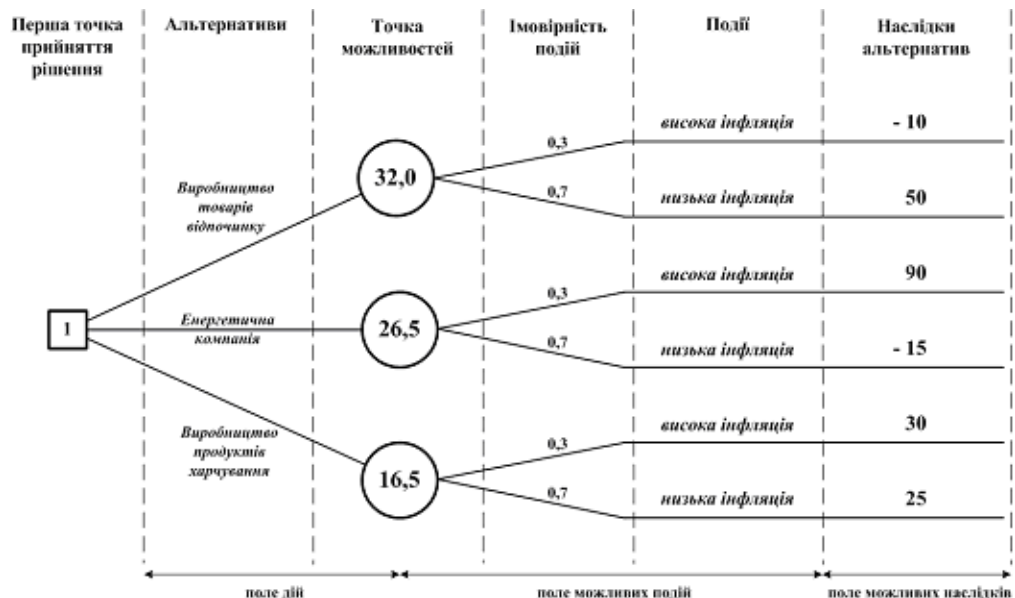


Рисунок 2.6 - Дерево прийняття рішень

Баєсові мережі є ще одним важливим методом прийняття рішень у нейроподібних системах. Вони представляють собою графічні моделі, що використовують теорію ймовірностей для моделювання набору змінних і їхніх залежностей. Баєсові мережі дозволяють ефективно працювати з неповними даними та враховувати невизначеність, що робить їх корисними для складних систем, де важлива обробка ймовірнісної інформації. Вони широко застосовуються в області прогнозування, діагностики та розпізнавання образів.

Підсилювальне навчання є методом машинного навчання, який базується на взаємодії агента з середовищем для досягнення певної мети. Агент приймає рішення, виконуючи дії, які змінюють стан середовища, і отримує зворотній зв'язок у вигляді нагород або штрафів. Цей процес дозволяє агенту навчатися оптимальним стратегіям для досягнення максимальної довгострокової нагороди. Підсилювальне навчання особливо ефективно для завдань, де важлива послідовність дій, таких як управління роботами, ігри та автономні транспортні системи.

Алгоритми глибокого навчання, які включають глибокі нейронні мережі, є одним із найсучасніших підходів у нейроподібних системах. Глибокі нейронні мережі складаються з багатьох шарів нейронів, що дозволяє їм виявляти складні закономірності у великих наборах даних. Ці алгоритми використовуються для різних завдань, включаючи класифікацію, регресію, кластеризацію та генерацію даних. Глибоке навчання є основою для багатьох сучасних систем штучного інтелекту, таких як розпізнавання мови, зображень і тексту, а також для систем, що приймають складні рішення в реальному часі.

Таким чином, алгоритми прийняття рішень у нейроподібних системах включають різноманітні методи, які дозволяють цим системам ефективно працювати з великими обсягами даних, враховувати невизначеність і адаптуватися до змінних умов. Кожен з цих алгоритмів має свої унікальні

особливості та переваги, що робить їх відповідними для різних застосувань у багатьох галузях.

Системи прийняття рішень, особливо ті, що базуються на нейроподібних обчисленнях, мають значний потенціал для покращення процесів аналізу та прийняття рішень у різних галузях. Однак, як і будь-які інші технології, вони стикаються з певними викликами та обмеженнями, що можуть вплинути на їхню ефективність та надійність.

Одним із основних викликів для систем прийняття рішень є проблема з інтерпретацією та прозорістю. Багато нейроподібних систем, зокрема нейронні мережі, часто сприймаються як "чорні скриньки", оскільки їхні внутрішні механізми складні для розуміння навіть для фахівців. Це ускладнює процес пояснення, як система дійшла до певного рішення, що може бути критично важливим у деяких галузях, таких як медицина чи фінанси, де прозорість і довіра є ключовими факторами.

Іншим важливим обмеженням є необхідність великих обсягів даних для навчання. Нейроподібні системи потребують великих та різноманітних наборів даних для ефективного навчання та генерації точних рішень. Однак збирання, зберігання та обробка таких даних може бути ресурсомісткою задачею. Крім того, у деяких випадках дані можуть бути обмежені або недоступні через питання конфіденційності, законодавчі обмеження або технічні труднощі.

Етичні питання, пов'язані з системами прийняття рішень на основі нейроподібних обчислень, стають все більш актуальними з поширенням цих технологій в різних сферах життя. Нейроподібні системи можуть виявляти певні упередження або біаси внаслідок некоректно підібраних або неповних даних для навчання. Якщо система навчалася на даних, які відображають соціальні або культурні стереотипи, це може призвести до несправедливих рішень та дискримінації за расовими, гендерними або іншими ознаками.

Багато нейроподібних систем важко пояснити, як саме вони приймають рішення. Це може стати перешкодою для розуміння та довіри до системи, прозорість стає критично важливою для запобігання дискримінації та несправедливості.

У випадку неправильних рішень системи або наявності упередженості, виникає питання, хто несе відповідальність за ці помилки.

Зростаюче використання нейроподібних систем підкреслює потребу у регулюванні та стандартизації цих технологій. Необхідність встановлення етичних стандартів, які б забезпечували безпеку, прозорість та справедливість у використанні нейроподібних систем, стає все більш актуальною.

Ці етичні питання вимагають уважного аналізу та розробки етичних принципів та рекомендацій для розробників, користувачів та регуляторів систем прийняття рішень на основі нейроподібних обчислень. Тільки через ретельне врахування цих питань можна забезпечити ефективне та етичне використання цих технологій у суспільстві.

Іншим викликом є забезпечення надійності та безпеки систем прийняття рішень. Нейроподібні системи можуть бути вразливими до різних типів атак, таких як атаки на цілісність даних або маніпуляції з вхідними даними. Забезпечення захисту від таких атак є критично важливим для підтримання довіри до цих систем і забезпечення їх безперебійної роботи.

Зрештою, питання інтеграції систем прийняття рішень у існуючі процеси та інфраструктуру також є важливим обмеженням. Впровадження нових технологій часто вимагає значних змін у існуючих робочих процесах, що може бути пов'язано з високими витратами та потребою в додатковому навчанні персоналу.

Отже, хоча системи прийняття рішень на основі нейроподібних обчислень мають великий потенціал, вони стикаються з численними викликами та обмеженнями. Подолання цих викликів вимагатиме

міждисциплінарного підходу, включаючи технічні, етичні, правові та соціальні аспекти. Лише завдяки комплексному підходу можна забезпечити ефективне, надійне та справедливе використання цих систем у різних галузях.

### 2.3 Порівняльний аналіз

Мета порівняльного аналізу архітектури та функціональності нейроподібних систем полягає в розумінні основних принципів архітектури і їх впливу на функціональність. Оцінка точності та ефективності систем допоможе виявити переваги та недоліки різних підходів до побудови нейроподібних систем.

Системи розпізнавання образів та системи прийняття рішень мають різні архітектури, що відповідають їх специфічним завданням і функціям.

Системи розпізнавання образів часто базуються на Convolutional Neural Networks (CNN) і Recurrent Neural Networks (RNN) або Long Short-Term Memory (LSTM) мережах. CNN складаються з шарів згортки, підвибірки, повнозв'язних шарів та функцій активації. Вони використовують згорткові операції для автоматичного вилучення ознак з вхідних зображень, зменшуючи розмірність даних за допомогою підвибірки, що дозволяє зберегти важливі ознаки. CNN забезпечують високу точність у задачах класифікації та детекції об'єктів, але мають високі обчислювальні вимоги під час тренування і потребують великої кількості даних для досягнення високої точності.

RNN та LSTM використовуються для обробки послідовних даних, таких як відео або серії зображень. Вони складаються з рекурентних шарів, де вихід одного кроку є входом для наступного. LSTM включають спеціальні блоки для зберігання та управління пам'яттю, що дозволяє їм враховувати контекст інформації у часових рядах. Ці архітектури ефективні для обробки послідовних даних, але мають проблеми з тренуванням через зникаючі градієнти та високі обчислювальні витрати.

Системи прийняття рішень часто базуються на методах Reinforcement Learning (RL), Decision Trees та Bayesian Networks. RL складається з агента, середовища, політики, функції винагороди та функції цінності. Агент взаємодіє зі середовищем, отримуючи винагороди або покарання за свої дії, що дозволяє йому навчитися оптимальним діям. Основні методи RL включають Q-learning та Policy Gradient. RL дозволяє навчання складних стратегій у динамічних середовищах та адаптацію до змін, але має високі обчислювальні витрати і потребує великої кількості епізодів для ефективного навчання.

Decision Trees складаються з вузлів, де кожен вузол представляє тест на ознаку, кожна гілка - результат тесту, і кожен лист - клас або значення. Ці дерева розділяють дані на підмножини на основі ознак, роблячи послідовні рішення до досягнення кінцевого листа. Random Forest використовує ансамбль дерев для підвищення стабільності та точності. Decision Trees є прозорими і легкими для інтерпретації, але можуть бути схильні до перенавчання і потребують обмеження глибини дерева для уникнення цього.

Bayesian Networks представляють орієнтований ациклічний граф, де вузли представляють змінні, а ребра - залежності між ними. Вони використовують байєсівську ймовірність для обчислення вірогідності подій на основі відомих залежностей та свідчень. Bayesian Networks дозволяють обробку невизначеності та неповних даних, а також ефективні у моделюванні причинно-наслідкових зв'язків, але складні у побудові та налаштуванні, особливо для великих мереж з багатьма залежностями.

Обидва типи систем мають свої унікальні архітектурні особливості та області застосування. Системи розпізнавання образів, такі як CNN та RNN/LSTM, є потужними інструментами для обробки візуальних даних і послідовностей, забезпечуючи високу точність розпізнавання. Водночас, системи прийняття рішень, засновані на RL, Decision Trees або Bayesian Networks, є ключовими для задач, що вимагають оптимального прийняття

рішень в умовах невизначеності та динамічних змін. Кожна архітектура має свої переваги та недоліки, що визначає її придатність для конкретних завдань та умов експлуатації.

Системи розпізнавання образів, такі як Convolutional Neural Networks (CNN) та Recurrent Neural Networks (RNN)/Long Short-Term Memory (LSTM) мережі, мають обмежену здатність до адаптації до змін у вхідних умовах. Їхня ефективність значною мірою залежить від якості та обсягу тренувальних даних, використаних під час первинного навчання. Для адаптації до нових або змінних умов необхідно провести додаткове навчання або fine-tuning на нових даних.

1. Convolutional Neural Networks (CNN): Для адаптації CNN до нових умов або змінених даних, часто використовують методику transfer learning, де вже натреновану модель налаштовують на новий набір даних. Це може бути ефективно, якщо нові умови не сильно відрізняються від початкових, але потребує додаткових ресурсів та часу для перенавчання.
2. Recurrent Neural Networks (RNN) та Long Short-Term Memory (LSTM): Ці мережі використовуються для обробки послідовних даних і можуть зберігати контекст інформації у часових рядах, що дає їм певну перевагу в адаптації до нових умов. Проте, для значних змін у вхідних умовах також потрібне додаткове тренування або налаштування, щоб мережа могла ефективно інтегрувати нову інформацію.

Системи прийняття рішень, такі як Reinforcement Learning (RL), Decision Trees та Bayesian Networks, загалом мають кращу здатність до адаптації до змін у вхідних умовах, оскільки вони можуть динамічно навчатися та оновлювати свої моделі на основі нової інформації.

1. Reinforcement Learning (RL): RL системи особливо добре підходять для адаптації до змін у динамічних середовищах. Вони можуть навчатися нових стратегій у режимі реального часу, коригуючи свої дії на основі

отриманих винагород або покарань. Це дозволяє RL агентам адаптуватися до нових умов без необхідності повного перенавчання.

2. Decision Trees: Древа рішень можуть адаптуватися до нових даних шляхом оновлення або створення нових вузлів, але це може призвести до зниження загальної узагальнювальної здатності моделі. Оновлення дерева рішення може вимагати періодичного перебудування, щоб запобігти перенавчанню та зберегти ефективність.
3. Bayesian Networks: Bayesian Networks добре підходять для роботи з невизначеністю та змінами у вхідних умовах, оскільки вони можуть оновлювати ймовірності залежностей між змінними на основі нових даних. Це дозволяє їм адаптуватися до нових умов, зберігаючи точність та стабільність моделей.

Здатність до адаптації до змін у вхідних умовах є критичним фактором для вибору типу системи залежно від специфіки завдань. Системи розпізнавання образів потребують додаткового навчання для адаптації до нових умов, що може бути ресурсомістким. Натомість, системи прийняття рішень, особливо засновані на RL, мають вищу здатність до адаптації завдяки своїм інтерактивним та динамічним методам навчання, що дозволяє їм ефективно працювати у змінних середовищах.

Системи розпізнавання образів, такі як Convolutional Neural Networks (CNN) та Recurrent Neural Networks (RNN)/Long Short-Term Memory (LSTM) мережі, можуть зіткнутися з викликами обробки великих обсягів даних через їхню складність та обчислювальну вимогливість.

Convolutional Neural Networks (CNN): CNN можуть бути великими та складними моделями, особливо якщо вони тренуються на великих обсягах даних або мають глибокі архітектури. Обробка великих обсягів даних зачасту вимагає великої обчислювальної потужності та ресурсів для тренування, а також швидких обчислень для прогнозування на нових даних.

Recurrent Neural Networks (RNN) та Long Short-Term Memory (LSTM): RNN та LSTM мережі можуть виявляти складність при обробці великих обсягів послідовних даних через проблему зниклих градієнтів та довгих залежностей. Тренування та інференція з використанням RNN та LSTM може вимагати значних обчислювальних ресурсів і часу, особливо для довгих послідовностей.

Системи прийняття рішень можуть також зустрічати виклики обробки великих обсягів даних, але їхній підхід може бути менш обчислювально вимогливим, залежно від конкретного методу.

Reinforcement Learning (RL): RL агенти можуть взаємодіяти з великими обсягами даних через їхню інтерактивну природу. Проте, тренування RL моделей може вимагати значних обчислювальних ресурсів, особливо при використанні складних алгоритмів навчання.

Decision Trees: Древа рішень можуть бути ефективними для обробки великих обсягів даних, оскільки їхнє тренування і використання не вимагає значних обчислювальних ресурсів. Вони можуть бути швидкими та масштабованими для прогнозування.

Bayesian Networks: Bayesian Networks можуть виявити складність при обробці великих обсягів даних через необхідність оцінки параметрів та структури моделі. Проте, їх можна застосовувати до великих обсягів даних у випадках, коли вони можуть бути ефективно розбиті на менші підзадачі.

Обробка великих обсягів даних є важливим викликом для обох типів систем. Системи розпізнавання образів, такі як CNN та RNN/LSTM, можуть мати складності через свою велику обчислювальну вимогливість та складність моделей. У той час, системи прийняття рішень, такі як RL, Decision Trees та Bayesian Networks, можуть мати менші обчислювальні вимоги, але все ще потребують уваги до ефективного управління великими обсягами даних.

Системи розпізнавання образів:

Convolutional Neural Networks (CNN): CNN відомі своєю високою точністю розпізнавання зображень. Завдяки своїм глибоким архітектурам та здатності до автоматичного вивчення ознак, вони здатні до ефективного розпізнавання об'єктів у великому спектрі умов, включаючи зміну розміру, розміщення та освітлення.

Recurrent Neural Networks (RNN) та Long Short-Term Memory (LSTM): RNN та LSTM можуть бути ефективними у розпізнаванні послідовностей та залежностей між даними. Вони здатні до прогнозування на основі попередніх даних та контексту, але їхню точність може бути обмеженою в разі довгих або складних послідовностей.

Системи прийняття рішень:

Reinforcement Learning (RL) системи можуть досягати високої точності у прийнятті рішень у динамічних середовищах. Вони вчаться оптимальним стратегіям через взаємодію з середовищем та отримання винагороди або покарання за свої дії.

Decision Trees Дерева рішень можуть мати високу точність у класифікації та регресії, особливо коли дані мають чітко виражені ознаки. Вони можуть швидко та ефективно приймати рішення на основі відомих залежностей.

Bayesian Networks можуть досягати високої точності у моделюванні ймовірностей подій та прогнозуванні на основі статистичних залежностей. Їхній підхід до управління невизначеністю дозволяє їм робити точні припущення навіть при обмежених даних.

Точність розпізнавання та рішень є критичними характеристиками для оцінки ефективності систем. Велика точність розпізнавання може забезпечити надійність та стабільність у рішеннях, що робить системи більш ефективними та придатними для різних застосувань. Важливо враховувати контекст та особливості завдань при виборі найбільш підходящої системи для конкретної ситуації.

Системи розпізнавання образів відомі високою точністю розпізнавання, особливо CNN, які забезпечують надійне визначення об'єктів на зображеннях. Однак, для досягнення високої точності необхідна велика кількість тренувальних даних, що може бути складно зібрати і обробити. Крім того, тренування складних моделей, таких як CNN, може вимагати значних обчислювальних ресурсів та тривалого часу.

Системи прийняття рішень можуть бути гнучкими та здатними до адаптації до змін в середовищі, особливо методи на основі Reinforcement Learning. Вони можуть навчатися на ходу та оновлювати свої моделі. Однак, деякі методи, зокрема Reinforcement Learning, можуть бути складними у розумінні та налаштуванні, а також вимагати великих обчислювальних витрат під час тренування.

## **Висновки до розділу 2**

У цьому розділі ми дослідили сучасні комп'ютерні системи, що використовують нейроподібні обчислення, зокрема системи розпізнавання образів та прийняття рішень. Згорткові нейронні мережі (CNN) є ключовими в розпізнаванні образів, забезпечуючи високу точність у таких галузях, як медична діагностика та автономні транспортні засоби. Системи прийняття рішень, що базуються на нейроподібних обчисленнях, ефективно автоматизують аналіз складних даних в різних сферах.

Порівняльний аналіз виявив, що успіх цих систем залежить від якості даних та алгоритмів навчання. Подальші дослідження мають зосередитися на підвищенні стабільності, масштабованості та енергоефективності цих технологій. Загалом, нейроподібні обчислення демонструють великий потенціал і перспективи розвитку, що сприятиме вирішенню складних задач у різних сферах.

## РОЗДІЛ 3 ЗАСТОСУВАННЯ НЕЙРОПОДІБНИХ СИСТЕМ У ТРАНСПОРТНІЙ ГАЛУЗІ

### 3.1 Автономне водіння автомобілем

Одним із ключових викликів автономного водіння є прийняття рішень в умовах невизначеності складного оточення транспортного засобу. На сьогоднішній день основною метою прийняття рішень є надання всебічних, надійних і стійких рішень, адаптованих до всіх можливих обставин, враховуючи невизначеність та непередбачувані ситуації. Важливим завданням є забезпечення безпеки користувачів доріг. Крім того, транспортний засіб має адаптувати свої рішення до навколишнього середовища, використовуючи датчики та внутрішню пам'ять для аналізу та оновлення свого уявлення як про власний стан, так і про об'єкти в навколишньому середовищі. Для даних цілий чудово підходять нейроподібні системи.

Метод навчання "від кінця до кінця" розглядає весь процес як одне завдання машинного навчання. У сфері автономних транспортних засобів "від кінця до кінця" означає систему, де нейронна мережа приймає основні рішення щодо водіння без визначення, якими мають бути вхідні та вихідні дані або скільки етапів навчання необхідно.

Розширене навчання (Reinforcement Learning, RL) полягає в навчанні прийняттю рішень через взаємодію транспортного засобу з оточенням. Використовуючи відомі сенсори (Lidar, Radar, RGB-камери тощо), транспортний засіб спостерігає за середовищем, що дозволяє захопити поточний стан і прийняти відповідні активні

рішення. Одна з методик навчання пропонує вирішення проблеми уникнення перешкод для автономних транспортних засобів у динамічно змінюваному середовищі з декількома цілями. Запропонована методика базується на рамках навчання з підкріпленням з кількома цілями, де використовували Q-навчання для визначення рішень на основі взаємодії з іншими транспортними засобами щодо досягнення цікавих цілей. Поєднуючи RL та Q-навчання, вдалося досягти ефективного прийняття рішень для проблеми обгону, що дозволило системі виконувати правильні дії, уникати зіткнень, підтримувати оптимальну швидкість і дотримуватися напрямку руху. Ієрархічне RL було представлено для вирішення виклику лівого повороту на неконтрольованих міських перехрестях, особливо з урахуванням зустрічного руху. На відміну від традиційних моделей прийняття рішень з фіксованою траєкторією, їх алгоритм прийняття рішень з множинними траєкторіями, що включає горизонтальні та вертикальні стратегії, покращує ефективність автономних транспортних засобів під час перетину перехресть, забезпечуючи безпеку.

Метод зворотного навчання з підкріпленням (Inverse Reinforcement Learning, IRL) використовують для навчання індивідуальних стилів водіння для самокерованих автомобілів через демонстрацію. Для захоплення відповідних аспектів руху по шосе було запропоновано набір характеристик, включаючи відстані до інших транспортних засобів, відстань до бажаної смуги, швидкість і прискорення. Підхід співвідносить спостережувані емпіричні значення характеристик з очікуваними значеннями моделі, щоб

навчити політику, яка представляє індивідуальний стиль водіння. Експериментальні результати з використанням реальних траєкторій транспортних засобів демонструють здатність підходу надійно навчати політики на основі демонстрацій, що робить їх придатними для автономної навігації.

Метод глибокого навчання з підкріпленням пропонується для прийняття рішень автономних транспортних засобів. Результати симуляцій використовуються для оцінки ефективності політики прийняття рішень за трьома аспектами. По-перше, наводиться порівняння та пояснення процесів навчання Deep Q-Learning (DQL) та дуельного DQL. Потім натреновані мережі, отримані з різних завдань водіння, застосовуються до трьох цільових завдань: правого повороту, лівого повороту та руху прямо. Тестова винагорода та відсоток успіху використовуються як метрики оцінки ефекту навчання з перенесенням (Transfer Reinforcement Learning, TRL). Нарешті, проводиться порівняння дуельного DQL з і без навчання з перенесенням, щоб підтвердити його ефективність.

Система прийняття рішень на основі глибокого навчання запропонована для автономних транспортних засобів для прийняття рішень перед в'їздом на круговий рух. Цей підхід допомагає транспортним засобам приймати правильні рішення, включаючи рішення щодо в'їзду або очікування при в'їзді на круговий рух.

Контролер на основі штучних нейронних мереж (ШНМ) відіграє ключову роль у автономному водінні, забезпечуючи адаптивне і точне прийняття рішень. Вони широко застосовуються

для автономного водіння, де автомобіль вибирає правильний рух і досягає мети без зіткнень, використовуючи дані з датчиків, хоча через динамічне середовище це завдання складне і вимагає тривалого навчання. Нечітка логіка, яка дозволяє приймати рішення на основі неповних або неточних даних, також використовується в нейронних мережах для контролю поведінки в складних умовах, що ефективно підтверджено як у симуляціях, так і в реальних тестах. Традиційні методи планування в штучному інтелекті мають обмеження через необхідність точної інформації про стан середовища, що часто неможливо через динамічність середовища і ненадійні сенсорні дані, тому адаптивні нейронні мережі, які можуть підлаштовуватися під зміни середовища, використовуються для вирішення проблем нелінійного відображення і розробки контролерів руху. Для зменшення складності планування використовують Q-навчання, метод навчання з підкріпленням, що включає такі дії: рух вперед, поворот праворуч, поворот ліворуч, обертання на  $180^\circ$  та зупинка, що дозволяє транспортним засобам навчатися і адаптуватися до середовища. Щоб покращити уникнення перешкод, використовується метод глибокого навчання з багат шаровими прямоперадавальними нейронними мережами, який забезпечує, що відстань між роботом і перешкодами залишається в допустимих межах, і контролер працює плавно та ефективно. Ієрархічні системи керування на основі глибокого навчання, що розподіляють завдання між різними рівнями для підвищення ефективності, використовуються для навігації мобільних роботів в невизначених середовищах, де верхній рівень

планує рух, а нижній рівень відстежує точки маршруту з використанням алгоритмів глибокого навчання для уникнення зіткнень. Також розроблені структури для комфортного автономного водіння на нерівних дорогах, що використовують інформацію про дороги від інших транспортних засобів, і стратегія на основі глибоких нейронних мереж, яка допомагає прогнозувати оптимальні команди руху під час паркування, враховуючи початкові умови з шумами.

### **3.2 Система оптимізації транспортних потоків**

Більшість сучасних систем керування світлофорами працюють за заданими алгоритмами та інтервалами часу. Через це система не здатна ефективно реагувати на змінювану ситуацію на дорозі. Перспективним рішенням цієї проблеми є оптимізація системи за допомогою штучних нейронних мереж. Їхня перевага полягає у можливості самонавчання, що дозволяє системі адаптуватися до змін дорожніх умов.

У роботі "Гнучке керування системою світлофорів на перехресті на основі нейронних мереж" І. А. Сєдих та Д. С. Демахіної (2017) розглядається алгоритм навчання штучної нейронної мережі для системи світлофорів, заснованої на багат шаровому персептроні з одним прихованим шаром. Згідно з запропонованим і реалізованим алгоритмом, було сформовано вибірку, на якій навчалася нейронна мережа. Використання нейронної мережі для керування світлофорами обумовлене її здатністю самонавчатися в умовах, що змінюються. Це так зване неконтрольоване навчання, коли попередньо налаштовані ваги починають змінюватися для максимізації критерію якості, яким у цьому випадку є пропускна здатність перехрестя. Таким чином, у

майбутньому можлива реалізація автоматичної адаптації системи керування до змінюваних умов руху, наприклад, під час ремонтних робіт, залежно від пори року тощо.

Розглянемо алгоритм роботи реального перехрестя, зображений на рис. 3.1. Повний цикл функціонування системи світлофорів на перехресті складається з чотирьох етапів. Кожен етап включає одночасну роботу декількох світлофорів, об'єднаних у групи, які, згідно з порядком включення, асоційовані з номером від 1 до 4. Кожен етап - це період часу від увімкнення зеленого сигналу в одній групі світлофорів до увімкнення зеленого сигналу в наступній групі світлофорів.

На рис. 3.1 номери в колах вказують належність світлофора до групи, а номери в квадратах позначають смуги, по яких рухаються автомобілі. Для кожної групи світлофорів цикл роботи складається з чотирьох стадій: жовтий (1) → зелений (2) → жовтий (3) → червоний (4).

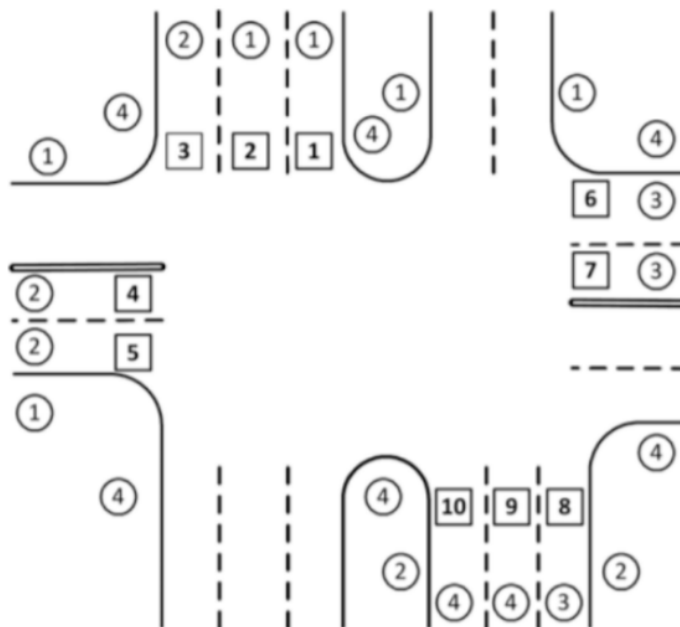


Рисунок 3.1 - Графічна схема світлофорів

У запропонованому алгоритмі керування перехрестям номери груп світлофорів залишилися незмінними, але послідовність їх активації більше не

фіксована. Іншими словами, після першої групи може бути включена третя група, що забезпечує гнучке керування перехрестям. Очевидною проблемою цієї модифікації є можливість повторного ігнорування будь-якої групи світлофорів. Внаслідок цього автомобілі, які перебувають у смугах, що належать до цієї групи, не можуть проїхати перехрестя, а пішоходи не можуть перейти дорогу. Для вирішення цієї проблеми в алгоритм введено обмеження часу очікування - 250 секунд. Якщо час очікування в будь-якій групі світлофорів досягне граничного значення, ця група буде включена наступною.

Модифікований алгоритм роботи світлофорів враховує запропоновану зміну. Жовтий сигнал (1) включається на фіксований час 3 секунди. Його увімкнення зупиняє червоний сигнал. Після завершення цього часу вмикається зелений (2) сигнал. Час роботи зеленого сигналу залежить від кількості автомобілів, що стоять у потоці. Група з найбільшою кількістю автомобілів у смугах вибирається для активації, якщо час простою будь-якої з груп не перевищує граничного значення. Інакше вибирається група з найвищим часом очікування. У вибраній групі вмикається жовтий сигнал (1). Далі в поточній групі світлофорів вмикається жовтий сигнал (3), який блимає протягом фіксованого часу 3 секунди. Після цього вмикається червоний сигнал (4). Під час роботи червоного сигналу світлофор знаходиться в режимі очікування, поки не ввімкнеться жовтий сигнал (1). Повертаємося до кроку 1.

Згідно з запропонованим і реалізованим алгоритмом, було сформовано вибірку, на якій навчалася нейронна мережа. Використання нейронної мережі для керування світлофорами обумовлене її здатністю самонавчатися в умовах, що змінюються. Це так зване неконтрольоване навчання, коли попередньо налаштовані ваги починають змінюватися для максимізації критерію якості, яким у цьому випадку є пропускна здатність перехрестя. Таким чином, у майбутньому можлива реалізація автоматичної адаптації системи керування до змінюваних умов руху, наприклад, під час ремонтних робіт, залежно від пори року тощо.

Параметри системи включають максимальну кількість машин у кожній відповідній групі ( $x_1 - x_4$ ), лічильники часу очікування для кожної групи ( $x_5 - x_8$ ), що відображають час, проведений групою у стані жовтого та червоного сигналів, а також одиничний сигнал ( $x_9$ ), який при увімкненні зеленого сигналу скидає лічильник часу очікування цієї групи до нуля (рис. 3.2).

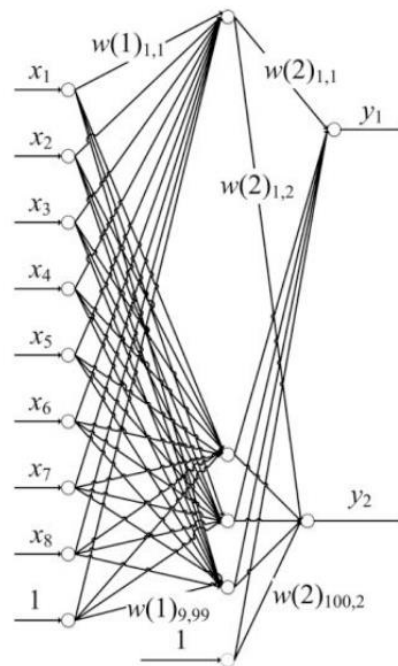


Рисунок 3.2 - Графічна схема нейронної мережі

Після вхідного шару розташований прихований шар, що містить 100 нейронів. Завдяки цьому шару та великій кількості нейронів у ньому, система здатна вирішувати поставлену задачу. Наступним є вихідний шар, який має два нейрони. Один нейрон визначає номер групи світлофорів, яка буде активована наступною, а інший визначає тривалість зеленого сигналу. Вхідні сигнали, проходячи через відповідні зв'язки, множаться на вагові коефіцієнти цих зв'язків і підсумовуються на вході нейрона прихованого шару.

Результати числового експерименту показали, що запропонований алгоритм керування ефективно вирішує завдання оптимізації транспортного

потоків в міських умовах. Оптимізація регулювання світлофорів за допомогою нейронних мереж є дієвим способом підвищення безпеки дорожнього руху. Слід приділяти більше уваги впровадженню нейронних мереж у системи керування світлофорами, оскільки їх використання може значно покращити ситуацію на дорогах.

Розглянемо, які результати дало застосування подібної системи в центрі Манхеттена:

1. Середній час затримки на ключових перехрестях зменшився на 10-20%, що призвело до покращення трафіку в години пік.
2. Швидкість руху транспорту зросла приблизно на 10%, покращуючи загальну ефективність дорожнього руху і зменшуючи час поїздки.
3. Кількість аварій на ділянках, де була впроваджена система, зменшилась на 8-12%, що підвищило безпеку на дорогах.
4. Ефективність громадського транспорту підвищилася, оскільки автобуси та інші засоби почали рухатися швидше і точніше за графіком.

### **Висновки до розділу 3**

Застосування нейронних мереж для оптимізації транспортних потоків та систем автономного водіння є перспективним напрямком у розвитку сучасних транспортних систем. На основі даних спостережень за реальним перехрестям були обрані оптимальні параметри нейронної мережі, включаючи архітектуру, функцію активації, кількість нейронів у проміжному шарі, вхідні та вихідні сигнали. Навчання нейронної мережі проводилося за допомогою алгоритму зворотного поширення помилки, що дозволило досягти значного підвищення ефективності управління транспортним потоком в міських умовах.

Оптимізація регулювання світлофорів за допомогою нейронних мереж сприяє підвищенню пропускної здатності перехресть та безпеки дорожнього руху,

завдяки здатності системи до самонавчання і динамічної адаптації до змін дорожньої ситуації. Це важливо не лише для покращення управління міським транспортом, але й для розвитку систем автономного водіння, які також використовують нейронні мережі для прийняття рішень у складних і змінних умовах.

## ВИСНОВКИ

У цій роботі було розглянуто різні архітектури та методи навчання нейроподібних систем. Особливу увагу приділено дослідженню ефективності різних архітектур у вирішенні конкретних завдань, а також проведено порівняльний аналіз їхніх переваг та недоліків. Окремо було розглянуто впровадження цих систем у транспортну галузь, де було виявлено суттєві переваги нейроподібних систем над традиційними методами.

Однією з ключових переваг нейроподібних систем є їх здатність до навчання та аналізу ситуацій, що дозволяє оптимізувати транспортні потоки набагато ефективніше, ніж стандартні алгоритми. Наприклад, при застосуванні в автопілоті автомобіля нейроподібні системи демонструють значну перевагу завдяки своїй здатності приймати рішення в нестандартних умовах, аналізуючи велику кількість факторів у реальному часі.

Крім того, нейроподібні системи показали високу ефективність у передбаченні та запобіганні аварійних ситуацій, що значно підвищує безпеку на дорогах. Завдяки своїй адаптивності, такі системи можуть швидко реагувати на змінні умови, що робить їх незамінними в сучасних транспортних технологіях.

Таким чином, результати нашого дослідження підтверджують, що нейроподібні системи мають численні переваги над традиційними алгоритмами. Їх впровадження в транспортну сферу може значно покращити ефективність та безпеку транспортних процесів. Висновки нашої роботи свідчать про перспективність подальших досліджень у цьому напрямку, а також про потенціал широкого застосування нейроподібних систем у різних галузях транспорту.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Симчак В. С. «Нейронні мережі» URL:  
<https://wiki.tntu.edu.ua/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D1%96%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D1%96>
2. Дипломна робота «Згорткові нейронні мережі та їх застосування до розпізнавання дорожніх об'єктів в умовах зашумленості» Женчук О. В.
3. Що таке зворотнє поширення? Деніел Н. URL:  
<https://www.unite.ai/uk/what-is-backpropagation/>
4. Лекція 1. Основні поняття розпізнавання образів URL:  
[http://om.univ.kiev.ua/users\\_upload/15/upload/file/pr\\_lecture\\_01.pdf](http://om.univ.kiev.ua/users_upload/15/upload/file/pr_lecture_01.pdf)
5. Рекурентна нейронна мережа URL:  
<https://www.wikidata.uk-ua.nina.az/%D0%A0%D0%B5%D0%BA%D1%83%D1%80%D0%B5%D0%BD%D1%82%D0%BD%D1%96%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D1%96%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D1%96.html>
6. Introduction to Convolutional Neural Networks (CNN) Manav M.
7. An Introduction to Convolutional Neural Networks (CNNs) Zoumana K. URL:  
<https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns>
8. Tesla показала, как видит нейронная сеть Autopilot 3.0 URL:  
<https://habr.com/ru/news/486600/>
9. Pattern Recognition in Machine Learning [Basics & Examples] Pragati B. URL:  
<https://www.v7labs.com/blog/pattern-recognition-guide>

10. Neuromorphic Hardware and Computing By — Dhriti Parikh , Gouri Kanade , Sambhav Bhasin , Prantik Chakraborty URL:  
[https://medium.com/@IEEE\\_Computer\\_Society\\_VIT/neuromorphic-hardware-and-computing-f7cc8f71ed58](https://medium.com/@IEEE_Computer_Society_VIT/neuromorphic-hardware-and-computing-f7cc8f71ed58)
11. Тема 9. Методи дерев рішень, класифікації та прогнозування URL:  
[https://moodle.znu.edu.ua/pluginfile.php?file=/486136/mod\\_resource/content/1/%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F%209.pdf](https://moodle.znu.edu.ua/pluginfile.php?file=/486136/mod_resource/content/1/%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F%209.pdf)
12. Dynamic Control System Design for Autonomous Car.  
By - Shoaib A., Farzeen M., Moongu J.
13. Design of a Control System for an Autonomous Vehicle Based on Adaptive-PID. By - Pan. Zh. URL:  
<https://journals.sagepub.com/doi/10.5772/51314>
14. An Autonomous Vehicle Driving URL:  
<https://www.ijee.ie/articles/Vol21-5/Ijee1674.pdf>
15. Traffic flow prediction using neural network. By Mouna J., Imad L., Ali Ya.  
URL:  
[https://www.researchgate.net/publication/325028512\\_Traffic\\_flow\\_prediction\\_using\\_neural\\_network](https://www.researchgate.net/publication/325028512_Traffic_flow_prediction_using_neural_network)
16. A novel traffic optimization method using GRU based deep neural network for the IoV system. Wu Wen, Dongliang Xu, and Yang Xia. URL:  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10280423/>