

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження використання адаптивного подвійного нео-нечіткого нейрона
для підвищення ефективності розпізнавання образів
(тема)

Виконав:
студент 2 курсу, групи СШМ-22-1
Плетньов В.В.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник проф. Бодянський Є.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

В.О. Філатов
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)
Кафедра _____ Штучного інтелекту _____
(повна назва)
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)
Тип програми _____ освітньо-наукова _____
(освітньо-професійна або освітньо-наукова)
Освітня програма _____ Системи штучного інтелекту _____
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Плетньову Владиславу Володимировичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Дослідження використання адаптивного подвійного нео-нечіткого нейрона для підвищення ефективності розпізнавання образів _____

затверджена наказом університету від 1 квітня 20 24 р. № 260Ст

2. Термін подання студентом роботи до екзаменаційної комісії 14 червня 20 24 р.

3. Вихідні дані до роботи _____ науково-технічні публікації, дані Інтернет-джерел щодо тематики дослідження, набори даних для навчання та тестування моделей розпізнавання образів _____

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі _____

2) Неонечіткий нейрон _____

3) Програмна реалізація _____

РЕФЕРАТ

Пояснювальна записка: 99 с., 4 рис., 6 табл., 3 дод., 16 джерел.

МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, НЕО-НЕЧІТКИЙ НЕЙРОН, НЕЧІТКА ЛОГІКА, РОЗПІЗНАВАННЯ ОБРАЗІВ.

Об'єкт дослідження – процес розпізнавання образів за допомогою нейронних мереж.

Предмет дослідження – використання адаптивного подвійного неочіткого нейрона для підвищення ефективності розпізнавання образів.

Мета роботи – підвищення ефективності розпізнавання образів шляхом розробки та використання моделі адаптивного подвійного неочіткого нейрона, що поєднує переваги нечіткої логіки та нейронних мереж.

Методи дослідження: аналіз наукових і технічних джерел щодо сучасних методів розпізнавання образів, розробка теоретичної моделі адаптивного подвійного неочіткого нейрона, програмна реалізація розробленої моделі з використанням фреймворків машинного навчання.

Кваліфікаційна робота присвячена дослідженню можливостей підвищення ефективності розпізнавання образів за допомогою адаптивного подвійного неочіткого нейрона. В рамках роботи проведено аналіз сучасних методів розпізнавання образів, зокрема згорткових нейронних мереж та неочітких нейронів. Розроблено теоретичну модель адаптивного подвійного неочіткого нейрона, що поєднує переваги нечіткої логіки та нейронних мереж. Модель реалізовано програмно з використанням сучасних фреймворків машинного навчання та проведено експериментальні дослідження на стандартних наборах даних (MNIST, CIFAR-10, ImageNet).

ABSTRACT

Master's thesis contains: 99 pp., 4 fig., 6 tabl., 3 ann., 16 references.

**FUZZY LOGIC, IMAGE RECOGNITION, MACHINE LEARNING,
NEO-FUZZY NEURON, NEURAL NETWORKS.**

Object of research – the process of image recognition using neural networks.

Subject of research – the use of an adaptive double neo-fuzzy neuron to improve the efficiency of image recognition.

Purpose of the work – to enhance the efficiency of image recognition by developing and utilizing a model of an adaptive double neo-fuzzy neuron that combines the advantages of fuzzy logic and neural networks.

Research methods: analysis of scientific and technical sources on modern methods of image recognition, development of a theoretical model of an adaptive double neo-fuzzy neuron, software implementation of the developed model using machine learning frameworks.

This master's thesis is dedicated to exploring the potential for improving image recognition efficiency using an adaptive double neo-fuzzy neuron. The work includes an analysis of modern image recognition methods, particularly convolutional neural networks and neo-fuzzy neurons. A theoretical model of an adaptive double neo-fuzzy neuron, which combines the advantages of fuzzy logic and neural networks, has been developed. The model has been implemented programmatically using modern machine learning frameworks, and experimental studies have been conducted on standard datasets (MNIST, CIFAR-10, ImageNet). The results demonstrate the advantages of the proposed approach, especially in noisy data conditions. The practical significance of the work lies in the creation of software based on the developed model and a set of Python tools suitable for further application and scientific research.

ЗМІСТ

Вступ.....	8
1 Аналіз предметної галузі.....	9
1.1 Опис предметної галузі.....	9
1.2 Проблеми розпізнавання образів.....	12
1.3 Згорткові нейронні мережі.....	15
1.4 Метод опорних векторів.....	35
1.5 Алгоритм k-найближчих сусідів.....	40
1.6 Нечіткі системи, нечітка логіка.....	45
1.7 Нео-нечіткі системи.....	47
1.8 Нечітка логіка у розпізнаванні образів.....	52
1.9 Адаптивний подвійний нечіткий нейрон: принципи та використання у розпізнаванні образів.....	57
1.10 Постановка задачі.....	58
2 Неонечіткий нейрон.....	64
3 Програмна реалізація.....	76
3.1 Опис програмних засобів.....	76
3.2 Опис і аналіз наборів даних.....	80
3.2.1 Вибір наборів даних.....	80
3.2.2 Попередній аналіз і підготовка даних.....	81
3.3 Експериментальне дослідження.....	83
3.3.1 Методика проведення експериментів.....	83
3.3.2 Результати на наборі MNIST.....	84
3.3.3 Результати на наборі CIFAR-10.....	85
3.3.4 Результати на наборі ImageNet.....	87
3.3.5 Порівняльний аналіз результатів.....	88
Висновки.....	89
Перелік джерел посилання.....	91
Додаток А Код програми.....	93

Додаток Б Графіки	97
Додаток В Відомість кваліфікаційної роботи	99

ВСТУП

Магістерська робота присвячена проблемі підвищення ефективності розпізнавання образів за допомогою адаптивного подвійного нечіткого нейрона. Вона включає дослідження та аналіз сучасних методів розпізнавання образів, розробку нової моделі, що поєднує переваги нечіткої логіки і нейронних мереж, та її експериментальну перевірку на репрезентативних наборах даних.

Актуальність проблеми посилюється невизначеністю, неповнотою та зашумленістю даних в реальних задачах розпізнавання образів. Новизна починається з розробки моделі адаптивного неонечіткого нейрона, який дозволяє гнучко настроюватись на такі дані та формує набір нечітких правил, зрозумілих для людини.

Робота включає експерименти з класичними методами, згортковою нейронною мережею та запропонованою моделлю, за результатами яких встановлено перевагу нечіткого нейрона, особливо на зашумлених даних. Практичне значення полягає у створенні програмного забезпечення на основі цієї моделі та набору Python-інструментів, придатних для подальшого застосування та наукового дослідження.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Опис предметної галузі

Розпізнавання образів – це важлива частина в галузі штучного інтелекту, яка має безпосередній зв'язок з нейронними мережами. Зародження цього напрямку безпосередньо пов'язане з появою комп'ютерів і потребою автоматизації процесів, раніше здійснюваних виключно людьми, зокрема процесів пов'язаних з візуальним сприйняттям і розпізнаванням образів.

Розпізнавання образів включає в себе велику кількість дисциплін: від теоретичної інформатики і математики до практичного застосування в комп'ютерному зору та системах штучного інтелекту. Відповідно до цього, розпізнавання образів спочатку базувалося на ідеях та методах, запозичених з різних галузей науки, проте з часом ця область почала розвиватися самостійно, пропонуючи нові підходи та технології.

Однією з ключових особливостей розпізнавання образів є використання векторів особливостей для представлення образів. Вектор особливостей створюється шляхом виміру специфічних характеристик об'єкта, таких як форма, колір, текстура, розмір або будь-яка комбінація цих характеристик. Другою важливою характеристикою є використання класифікаторів для розуміння та розпізнавання об'єктів.

Розпізнавання образів може бути проведене за допомогою різних методів, таких як статистичні, структурні, синтаксичні або нейронні підходи. Статистичний підхід базується на визначенні векторів особливостей та використанні статистичних алгоритмів для класифікації об'єктів. Структурний підхід зосереджується на розпізнаванні структур об'єктів, а синтаксичний підхід використовує синтаксичні алгоритми для визначення структури об'єктів. Нейронні підходи базуються на моделюванні нейронних мереж для розпізнавання образів

Сучасні системи розпізнавання образів використовують поєднання різних технологій, включаючи комп'ютерний зір, машинне навчання та штучний інтелект. Ці системи можуть бути використані для вирішення різних завдань, включаючи розпізнавання об'єктів, визначення місця розташування, відслідковування об'єктів в часі, розпізнавання руху, аналіз текстури та форми, тощо.

Автоматичне розпізнавання образів – це процес, за допомогою якого машина здатна визначити ідентичність об'єкта чи сцени без активної участі людини. Це може бути досягнуто за допомогою використання автоматичних алгоритмів для аналізу та класифікації вхідних даних.

Розробка алгоритмів для розпізнавання образів це складний процес, який вимагає глибокого розуміння даних, на яких працює алгоритм. Цей процес може включати в себе декілька фаз, таких як визначення проблеми, збір та попередня обробка даних, вибір алгоритму та тестування його ефективності.

Розпізнавання образів використовується в багатьох сферах нашого життя. Воно застосовується в медичному зображенні, де цифрові зображення рентгенівських променів, магнітно-резонансної томографії та інших медичних зображень, аналізу ракових клітин, оцінюються за допомогою алгоритмів розпізнавання образів. Воно використовується в автоматизованих системах, таких як безпілотні автомобілі, де розпізнавання образів дозволяє машинам розуміти своє оточення. Інший приклад це системи безпеки, де технологія розпізнавання образів може бути використана для ідентифікації осіб і візуального моніторингу.

Штучний інтелект у своєму розвитку використовує розпізнавання образів як один з ключових складових. Різні моделі штучного інтелекту використовують розпізнавання образів для симуляції людського перцептивного процесу. Це дає можливість машинам, оснащеним штучним інтелектом, краще розуміти візуальний світ навколо них, що підвищує їх спроможність виконувати складні завдання.

Однією з найважливіших проблем розпізнавання образів є розуміння вмісту зображення, тобто отримання семантичної інформації з візуальних даних. Комп'ютер, на відміну від людини, сприймає зображення як масив пікселів, без власного розуміння їх значення, зв'язків між об'єктами та контексту. Саме це робить задачу розпізнавання образів за допомогою комп'ютера вкрай складною

Не дивлячись на різноманітні можливості, розпізнавання образів також викликає певні проблеми. Одним з найбільших викликів є складність визначення максимально ефективного способу представлення образу або його частин. Іншим викликом є варіабельність умов зображення, наприклад, зміна освітлення, точки зору або масштабу.

Розпізнавання образів в галузі штучного інтелекту стосується процесу виявлення та ідентифікації об'єкта, особливості або деталі, на основі дослідження даних загал. Цей процес може бути емпіричним (заснований на спостереженнях або досвіді), складним алгоритмом обробки даних або комбінацією обох підходів.

Головна мета розпізнавання образів – розв'язання складного завдання аналізу і ідентифікації об'єктів у великому наборі даних. Вони часто використовуються в різних галузях, таких як біометрія, ідентифікація особи, захист інформації, електронна торгівля, комп'ютерний зір та ін.

Історично, розпізнавання образів вперше було використане у 1960-их роках, коли дослідники почали розробляти методи для розпізнавання символів і почерку на папері. З тих пір, технологія значно розвинулася і сьогодні включає в себе складні методи машинного навчання та глибокого навчання, які можуть розпізнавати все – від простих геометричних фігур до складних об'єктів необхідних у певному просторі.

Методи розпізнавання, що використовуються в розпізнаванні образів можуть бути розподілені на дві основні категорії: контрольовані та неконтрольовані техніки. Контрольовані методи потребують попередньої мітки або класифікації вхідних даних, в той час як неконтрольовані методи

утворюють моделі безпосередньо із вхідних даних без використання попередньої мітки.

З цією проблемою розпізнавання образів допомагає багато різних підходів, що включають в себе використання статистичних методів, нейронних мереж, генетичних алгоритмів, методів машинного навчання і багато інших.

Незважаючи на існуючі виклики, перспективи розпізнавання образів виглядають обнадійливо. Напрямок продовжує розвиватися завдяки новим технологіям, таким як глибоке навчання і нейронні мережі. Завдяки цим технологіям, прогнозується що точність і швидкість розпізнавання буде лише зростати, відкриваючи нові можливості для їх використання.

При цьому, зростає визнання та застосування адаптивних подвійних нечітких нейронів у розпізнаванні образів для збільшення точності та ефективності цього процесу. Вони застосовуються для покращення роботи нейронних мереж, що використовуються в розпізнаванні і класифікації образів.

1.2 Проблеми розпізнавання образів

Розпізнавання образів можна розглядати на різних рівнях. На рівні пікселів формується задача обробки зображення: пошуку границь об'єктів, виявлення текстур, кольорів тощо. На рівні об'єктів необхідно визначити, які об'єкти присутні на зображенні, де вони розташовані, якої форми вони. На наступному рівні, рівні сцени, потрібно розуміти, які об'єкти взаємодіють між собою, який взаємозв'язок між ними. На кінцевому рівні, рівні семантики, за допомогою аналізу всієї інформації, отриманої на попередніх рівнях, визначається загальний сенс зображення.

Цілком зрозуміло, що кожен з рівнів включає в себе набір допоміжних задач, які спрямовані на розв'язання конкретних проблем: від зниження

шумів і пошуку границь об'єктів на низьких рівнях, до розпізнавання окремих об'єктів та визначення структури сцени на вищих.

В рамках даної роботи розглядається вивчення та дослідження використання адаптивного подвійного нечіткого нейрона для покращення ефективності розпізнавання образів на різних рівнях аналізу.

Процес розпізнавання образів може бути поділений на декілька ключових задач, що реалізуються поетапно:

Етап попередньої обробки зображень включає основні завдання, такі як вирівнювання розміру зображення, перетворення кольору, видалення шумів і т.д. Передбачається, що на цій стадії зображення буде готове для додаткового аналізу.

В процесі екстракції (видобування) ознак використовуються методи для виявлення ключових ознак на зображенні, які можуть бути корисні для подальшого розпізнавання образів. Це може включати різні техніки в залежності від специфіки задачі, включаючи детектори країв, виявлення текстури, виявлення кольору і т.д.

Після етапу екстракції ознак, зображення обробляються за допомогою класифікатора для визначення класу або категорії, до якого вони належать, формуючи етап класифікації образу. Класифікатор може бути заснований на різних машинних алгоритмах навчання, включаючи дерева рішень, багатокласову класифікацію, нейронні мережі, метод опорних векторів тощо.

Після класифікації образу, може статися ще один крок, що включає в себе пост-обробку для виправлення будь-яких помилок або для виявлення складних взаємозв'язків між об'єктами на зображенні.

Важливо зауважити, що розпізнавання образів – це не одна конкретна задача, а набір різних завдань, які мають виконуватися в поєднанні для досягнення кінцевого результату. Залежно від точної постановки проблеми, може бути потрібна реалізація одних задач, а інші можуть бути незастосовні.

Перші методи розпізнавання образів ґрунтувалися на традиційних статистичних методах та використовували детерміновані властивості зображень для класифікації.

Важливу роль в розпізнаванні образів грають методи машинного навчання, які можуть бути навчені розпізнавати образи на основі великої кількості прикладів. Класифікатори, такі як метод опорних векторів (SVM), дерева рішень і метод ближайших сусідів (KNN), є деякими з найбільш часто використовуваних методів машинного навчання в цих цілях.

Глибокі нейронні мережі стали відмінною альтернативою для розпізнавання образів завдяки своїй здатності до глибокого навчання та загальної моделі. Спершу, образ проходить через Комбінації операцій згортки та субдискритизації (convolution, pooling) для виявлення ключових характеристик, а потім ці характеристики використовуються для класифікації в повнозв'язному шарі мережі.

Нейро-нечіткі системи комбінують принципи нейронних мереж і нечіткої логіки, щоб створити систему, яка може навчатися і адаптуватися в складних умовах. Такі системи можуть використовуватися для розпізнавання образів, де є деяка невизначеність або неясність. Кожен з цих методів має свої передові та слабкі сторони. Знаючи відмінності і можливості кожного з них, можна оптимізувати розпізнавання образів для конкретних завдань.

Вибір методу для розпізнавання образів залежить від конкретного завдання. Статистичні та машинні методи можуть виявитися ефективними для завдань з відносно простими та структурованими даними, тоді як глибокі нейронні мережі або нейро-нечіткі системи можуть бути кращим вибором для завдань, що вимагають багатошарової абстракції та здатності до обробки великих обсягів даних.

Перегляд методів розпізнавання образів свідчить про велике розманіття алгоритмів, які можуть бути використані в цій області.

Зрозуміння того, який метод найкраще підходить для конкретного завдання, є ключовим фактором ефективного застосування розпізнавання образів.

1.3 Згорткові нейронні мережі

Згорткові нейронні мережі (Convolutional Neural Networks – CNN) є однією з основних складових в галузі штучного інтелекту, спеціалізуючись на обробці візуальних даних. CNN побудовані за аналогією до механізму виду людей, включаючи в себе нейрони, які активують у відповідь на специфічні образи й об'єкти в полі зору.

CNN були розроблені з метою подолання обмежень повної зв'язності між нейронами в шарах нейронних мереж. Ключовим елементом CNN є оператор згортки, який перетворює вхідні дані шаром, що набагато менше в розмірах, але більше в глибині. Згортка використовує фільтри для визначення патернів у даних, включаючи кольори, текстури, геометрію, і багато іншого.

Згорткові нейронні мережі складаються з трьох основних типів шарів: згорткового шару, шару підвибірки (що також відомий як шар «управління» або «макс пулінгу») та повною зв'язному шару. Нижче подані деталі кожного типу шару.

Згорткові шари: згортка – це математична операція, яка застосовується на вхідному образі використовуючи фільтри. Це дає змогу виявляти високорівневі атрибути, як канти, кольори, текстуру і т. д., які можуть бути важливими для конкретної задачі

Шари підвибірки: шари підвибірки, або макс пулінгу, використовуються для пониження роздільної здатності образу (downsampling). Вони працюють шляхом виконання матричної операції на образі для вибору пікселів, що мають найбільше значення. Це допомагає зменшити розмір даних і складність обчислень, зберігаючи при цьому ключову інформацію та патерни в даних.

Повністю зв'язні шари: встановлюють зв'язок з усіма нейронами в попередньому шарі. Вони використовуються в кінцевому етапі CNN для класифікації визначених атрибутів і особливостей у відповідні класи.

Кожна з цих компонент є необхідною для побудови ефективної згорткової нейронної мережі. Загалом, CNN можуть бути структуровані шляхом поєднання цих шарів у структуру, що відповідає особливим вимогам задачі, наприклад, жерелу даних або виду даних, що обробляються.

Як було раніше відзначено, ключовим принципом CNN є здатність фокусуватися на високорівневих особливостях, виконуючи при цьому набагато меншу кількість обчислень. Це дієво у багатьох задачах машинного навчання, що залучають обробку зображень або відео, таких як розпізнавання об'єктів, семантична сегментація, та відеоаналіз. Також їх використовують для прогнозування часових рядів, обробки звуку та візуалізації даних.

У великій кількості застосувань згорткові нейронні мережі показали стабільну перевагу над іншими методами машинного навчання, стаючи золотим стандартом обробки візуальної інформації для ряду задач, серед зображення изображение, картинки зображень відеоаналіз.

Як завжди, дослідження в області CNN продовжується, і наразі розроблено ряд варіацій CNN, таких як глибокі згорткові мережі (Deep Convolutional Networks) і згорткові мережі з повторенням (Recurrent Convolutional Networks). Всі ці варіанти мають свої переваги та недоліки і можуть бути використані для різних задач і потреб.

Наступним важливим контекстом, де домінує CNN, є їх застосування у комп'ютерному зорі, що з'єднує гуманітарні та технічні науки та має на меті створити штучні системи, які можуть "бачити" і розуміти візуальний світ так само, як це робить людина.

Одним з найбільш очевидних застосувань CNN в комп'ютерному зорі є задачі розпізнавання образів, коли мається на увазі визначення, який клас або категорію представляє вхідне зображення. Для цього CNN

використовується для навчання на великому наборі зображень, кожне з яких має певну мітку. Після тренування, CNN може визначати клас нових зображень з високою точністю.

Методи глибокого навчання, такі як CNN, є корисними для класифікації зображень, оскільки вони можуть виявляти і вчитись важливим особливостям без потреби в їхньому ручному визначенні та введенні. Практичні застосування розпізнавання образів на базі CNN включають системи автономного водіння, безпеку і навігацію, рекомендаційні системи, робототехніку, медичні діагностичні системи та ін.

Крім того, CNN широко використовуються у вирішенні задач детекції об'єктів. У протипагу простому розпізнаванню образів, детекція об'єктів передбачає не тільки виявлення класу об'єкта, але й його розташування на зображенні. Для цього CNN використовуються для визначення області об'єкта на зображенні та класифікації цього об'єкта. Застосування детекції об'єктів включають підтримку безпеки в містах, системи відеоспостереження, планування шляху для безпілотних машин, розпізнавання облич та медичну діагностику.

Також є важливо відзначити застосування CNN у сегментації зображень – процесі поділу зображення на декілька частин або «сегментів», які можуть бути аналізовані окремо. Це можуть бути набори пікселів, які представляють певний об'єкт або область зображення. Ця задача має велике значення у медичному зображенні, у якому потрібно точно визначити границі органів або патологічних змін.

Визначена специфіка CNN дозволяє їм ефективно обробляти дані, що мають просторову структуру. Значущим прикладом є обробка аудіо даних. Згорткові нейронні мережі працюють на аудіо сигналах перетворені в спектрограми, які представляють собою часово-частотні зображення звукового сигналу. У такому контексті, CNN може розглядатися як потужний інструмент для аналізу звукових даних у задачах розпізнавання мови, музичної класифікації, а також інших сферах аудіо обробки.

Дослідницька спільнота активно розробляє й впроваджує нові архітектури CNN. Ці архітектури, такі як AlexNet, VGGNet, GoogleNet, ResNet та інші, визначають показники ефективності в галузі штучного інтелекту. Вони певним чином розширюють базовий принцип роботи CNN, запроваджуючи нові принципи обробки даних та навчання мережі.

AlexNet, запропонована в 2012 році, була однією з перших великих згорткових нейронних мереж, яка продемонструвала перевагу глибокого навчання в задачах класифікації зображень.

VGGNet, основою якого є проект VGG (Visual Geometry Group) в Оксфордському університеті, відомий використанням дуже глибоких мереж (з до 19 шарами) і маленьких згорткових фільтрів 3x3.

GoogleNet внесла внесок у структуру структурованості в архітектуру CNN, включивши компоненти, такі як «Inception modules», які дозволяли мережі ще глибше «зрозуміти» вхідні дані, не впливаючи на розмір і швидкість мережі.

Однак, ряд CNN був представлений Microsoft у 2015 році під назвою ResNet (short for Residual Network). Він включав архітектури, які мають до 152 шари, і вперше застосував «residual connections», що дозволило подолати проблему занепаду градієнту, що траплялася в особливо глибоких нейронних мережах.

Ці архітектури не тільки виконують роль важливих «кнопок» у виробництві прогнозних моделей на даних реального світу, але й служать основою для розуміння концепцій глибокого навчання, соціальної мережі та інновацій в цій області.

Важливим аспектом роботи з CNN є вибір правильного набору гіперпараметрів, що включають розмір фільтрів, кроків (strides), кількість шарів фільтрів, а також вибір відповідних функцій активації, які визначають вихід кожного нейрона в мережі. Великий багатовимірний простір гіперпараметрів вимагає виснажливого обчислювального пошуку, і дослідники вивчають нові методи для оптимізації підбору гіперпараметрів.

В основному, CNN вимагає великої кількості даних для навчання. Тому стратегії, такі як аугментація даних, регуляризація та перенесення навчання (transfer learning) використовуються для поліпшення ефективності навчання та уникнення перенавчання. Перенесення навчання надає змогу використовувати вже натреновані на великих масивах даних архітектури згорткових нейронних мереж (такі як AlexNet, VGGNet, GoogleNet, ResNet) для вирішення конкретних задач, отримуючи на вихід більш стабільні та адаптовані моделі.

Деякі з останніх тенденцій в CNN включають просторові трансформації нейронних мереж (Spatial Transformer Networks – STNs) для покращення геометричної інваріантності, мережі Фракталів (FractalNet), які об'єднують зв'язні шари та згорткові шари в ієрархічну структуру та глибокі згорткові мережі з шаром уваги.

Важливо зазначити, що всемогутньою як завжди залишається ітеративна процедура оновлення ваг – алгоритм зворотнього розповсюдження помилки (backpropagation), який використовується для тренування згорткових нейронних мереж і виконується з використанням стохастичного градієнтного спуску або його модифікацій.

Таким чином, наразі згорткові нейронні мережі визнані провідним інструментом в області комп'ютерного зору і глибокого навчання, надаючи науковцям і дослідникам сильний і гнучкий інструмент для аналізу та вирішення завдань класифікації та розпізнавання, що включають візуальні дані. Завдяки їхній здатності навчатися з великої кількості даних і визначати ключові особливості і структурні відносини в даних, згорткові нейронні мережі проводять революцію в промисловості, і їх значення тільки продовжить зростати в майбутньому.

Щоб повною мірою зрозуміти згорткові нейронні мережі, корисно заглибитися в архітектуру та складові елементи цих мереж. У своєму найпростішому вигляді, CNN є послідовністю трьох видів шарів: згорткових шарів, шарів устрою підвибірки та повністю зв'язаних шарів.

Згорткові шари є основою CNN і відповідають за більшу частину обчислювальної роботи. Вхідні дані представляють собою набір «зображень», кожне з яких проходить через наявні фільтри. Кожний фільтр – це набір важелів, які оновлюються під час процесу навчання. Після застосування згортки до вхідного зображення за допомогою ядра, створюється «карта особливостей».

Шари підвибірки, також відомі як шари упакування, використовуються для зменшення розміру даних, які передаються між згортковими шарами. Це робиться для зменшення обчислювальної навантаженості, кількості параметрів та контролю перенавчання. Шари підвибірки працюють шляхом «узагальнення» вихідних даних з попереднього шару, зазвичай шляхом обрання максимального значення (операція, відома як макс пулінг) або обчислення середнього значення (середній пулінг).

Повністю зв'язні шари зазвичай розташовані в кінці CNN і використовуються для виконання класифікації на основі навчених особливостей, виведених з попередніх шарів. Ці шари взаємопов'язані із всіма вихідними даними попереднього шару, і виходи з цих шарів зазвичай пропускаються через функцію активації, таку як функція softmax, яка надає ймовірності для кожного з виходів.

Процес навчання для CNN володіє такими самими атрибутами, як і для будь-якої нейронної мережі, а саме: оголошується функція втрат, яку слід мінімізувати (така як перехресна ентропія для задач класифікації), а ваги мережі оновлюються за допомогою алгоритму оберненого розповсюдження помилки і градієнтного спуску або його варіацій.

Великі CNN можуть мати десятки або навіть сотні таких шарів, кожен з яких навчається виявляти все більш складні особливості. Це створює ієрархічну структуру набору признаков, у якій нижні шари виявляють прості особливості (такі як кольори та контури), а верхні шари виявляють складні абстракції (такі як обличчя, об'єкти або сцени).

Функція активації: після кожної згортки, вихід проходить через функцію активації. Вона відповідає за додавання нелінійності до мережі. Фактично, без функцій активації, CNN буде виконувала лише лінійні перетворення, що обмежить його можливості виявлення складних шаблонів. Найчастіше використовується ReLU (Rectified Linear Unit) функція активації.

Шар softmax: в кінці згорткової нейронної мережі, зазвичай розташовується шар softmax. Софтмакс діє як активатор і нормалізатор, перетворюючи зважені вихідні дані на ймовірнісні значення для кожного класу, який потрібно передбачити.

Ця базова архітектура CNN може бути значно варіативна і модифікована в залежності від завдань і даних, з якими ви працюєте.

Деякі поширені модифікації і розширення включають: використання padding для збереження просторової резольюції під час згортки; використання змінного кроку (strides) для певного контролю над просторовим зменшенням розміру під час згортки; використання 1x1 згорток для зменшення глибини карт особливостей; використання групових згорток для подальшого зменшення обчислювального навантаження.

Поза цим, різноманітні архітектури CNN були розроблені, щоб виконувати конкретні завдання або покращувати ефективність навчання. Наприклад, архітектури, такі як Inception та ResNet, вписалися в принципи перемикачів та «остаточних з'єднань» відповідно, що дозволило побудувати глибокі мережі, що можуть навчитися виявляти та маніпулювати складніші особливості в даних.

Нормалізація шарів: ця додаткова техніка може бути додана до згорткової нейронної мережі, щоб її прискорити та стабілізувати процес навчання. Спочатку введена в архітектурі AlexNet, метод нормалізації шарів обмежує активацію функції, нормалізуючи вихід кожного нейрона в шарі мережі.

Dropout: це техніка регуляризації, яка рандомно «відключає» певну кількість нейронів в повністю зв'язаних шарах під час навчання. Це робиться щоб уникнути перенавчання мережі під час процесу навчання.

Batch Normalization: ця методологія, представлена в 2015 році, забезпечує стабілізацію процесу навчання шляхом нормалізації ваг в мережі. Ця техніка підвищує швидкість навчання, зменшує чутливість до вихідного параметру та покращує загальну продуктивність мережі.

Transfer Learning: замість навчання мережі з нуля, часто практичніше і ефективніше здійснювати передачу навчання, використовуючи моделі, які вже були попередньо навчені на великих наборах даних. Можна використовувати ці попередньо навчені моделі як вихідну точку і адаптувати їх до нових завдань за допомогою більш малих наборів даних.

Ансамблі CNN: для покращення продуктивності та надійності прогнозу, можна використовувати ансамблі згорткових нейронних мереж. Це може включати тренування кількох мереж на тих самих даних та комбінування їхніх прогнозів для отримання кінцевого виводу.

У простій формі, ці додаткові компоненти допомагають поліпшити роботу згорткових нейронних мереж. Втім, кожен набір даних є унікальним, і кожна задача потребує різних підходів. Важливо зрозуміти ці концепції та експериментувати з різними архітектурами та методами для отримання найкращих результатів для вашої конкретної задачі. Зазвичай, процес включає зміну архітектури мережі, оптимізацію гіперпараметрів, адаптацію технік навчання та перевірку результатів за допомогою відповідних метрик.

Додаткові виміри в глибині архітектури згорткових нейронних мереж, такі як використання 3D згортки та рекурсивних згортки, відкривають нові області застосування CNN. Такі техніки можуть бути використані для обробки 3D зображень або відеоданих, у яких присутній часовий аспект.

У просторі рекурентних згорткових мереж, ConvLSTM – це згорткова версія LSTM (Long Short-Term Memory), яка додає рекурентні зв'язки до архітектури CNN. Ця модель є особливо корисною для відеоаналітики,

оскільки вона може обробляти просторові і темпоральні залежності одночасно.

3D згортки або просторово-временні згортки – це загальні застосування згорткових мереж до 3D вхідних даних. Він розглядає додаткову вимірність (часто є третьою вимірністю час) і використовуються ядра згортки, які можуть рухатися в трьох вимірах.

Такі архітектури CNN, як DenseNet і MobileNet, розроблені з метою підвищення ефективності та зменшення кількості параметрів, шляхом використання випереджувальних зв'язків та глибоких міжшарових зв'язків, відповідно.

На додаток до цього, з усією славою глибокого навчання, важливо спробувати різні підходи до вашої конкретної задачі. Наприклад, використання простіших моделей машинного навчання, таких як логістична регресія або випадкові ліси, може бути достатнім і більш ефективним для наборів даних з малою кількістю функцій. Крім того, кластеризація, машинне навчання на основі відстані або моделі на основі правил також можуть надати корисний початковий огляд ваших даних і виявити корисні залежності.

На завершення, хочу сказати, що область глибокого навчання зараз переживає активний розвиток. Мінливість технологій, алгоритмів та практик означає, що багато нових особливостей і покращень з'являються постійно. Затриматися в курсі останніх подій цієї області та продовжувати навчатися новим підходам і виключно важливо для реалізації глибокого навчання в різних застосуваннях таких як розпізнавання мови, зображень та обробка відео, медичне візуалізування, само-керовані автомобілі та багато іншого.

Застосування згорткових нейронних мереж не обмежується областю комп'ютерного зору. Вони також використовуються в інших галузях, таких як обробка природної мови (NLP). Сучасний підхід до даної проблематики передбачає використання векторних відображень слів за допомогою технік,

таких як word2vec або GloVe, тоді як CNN використовується для визначення семантичних зв'язків та об'єктів між словами в реченні.

Якщо говорити про майбутнє згорткових нейронних мереж, ми можемо очікувати розвиток у кількох напрямках:

– модифікація та оптимізація архітектури: як з вивченим, так і з новими моделями буде проводитися багато експериментів. Вчені будуть продовжувати досліджувати кращі способи структурування та оптимізації мереж для досягнення кращих результатів;

– удосконалення в розпізнаванні образів: маючи багато галузей застосувань, CNN продовжить вдосконалювати свою здатність до машинного зору;

– стиснення моделі: оскільки розмір та складність CNN ростуть, з'являється потреба в їх стисненні для використання на пристроях з обмеженими ресурсами, такими як мобільні пристрої;

– уникнення bias: як і всі машинні алгоритми навчання, CNN можуть бути занадто чутливими до шуму в даних або некоректно класифікувати навчальні дані через зміщеність народження цього набору даних. Тому в майбутньому буде акумульовано більше зусиль на уникнення цих предвзятостей у навчальних даних.

– впровадження штучного інтелекту: застосування великого обсягу даних, необхідного для успішної роботи CNN, дасть штовх до розвитку всіх галузей, пов'язаних із AI. Впровадження цих новітніх технологій може привести до економічного росту, позитивного соціального впливу, підвищення ефективності робіт і можливість рішення більш складних проблем.

В кінцевому результаті, згорткові нейронні мережі та глибоке навчання продовжує розвиватися, і буде таким важливим інструментом для машинного навчання та штучного інтелекту в наступних десятиліттях.

Еволюційні підходи до архітектури CNN: ідея полягає в тому, щоб застосувати алгоритми еволюції до дизайну та налаштуванню згорткових

нейронних мереж. Більшість архітектур CNN, які використовуються сьогодні, були розроблені людьми. Однак, з появою обчислювальної потужності, спроби автоматизувати цей процес стають все більш можливими та привабливими.

Онтологічні підходи: CNN добре впораються з розпізнаванням шаблонів у даних, але вони ще повинні вчитися «розуміти» семантику того, що вони бачать. Використання онтологічних підходів, які включають в себе знання про світ, може допомогти згортковим мережам ліпше розуміти інтерпретувати візуальну інформацію.

Больше звернення на нейроморфні мережі: нейроморфні системи намагаються наслідувати біологічні нейронні мережі з метою поліпшення ефективності обчислень. З точки зору CNN, це може сприяти конструюванню архітектур, які краще пристосовані до конкретних завдань.

Варто зазначити, що ці розробки є досить новими і в даний час активно досліджуються. Однак, впевненість в тому, що CNN та глибоке навчання залишаться в центрі розробок ШІ, зростає. Майбутнє CNN, безумовно, тримає в собі багато обіцянок та цікавих можливостей. Впереді нас чекає захоплюючий час в галузі штучного інтелекту, і CNN грає ключову роль у цьому розвитку.

Федеративне навчання: федеративне навчання – це метод навчання машин, який дозволяє моделям навчатися з даними, що знаходяться на пристрої користувача. Важливість цього підходу полягає в тому, що він забезпечує захист приватності, оскільки персональні дані користувача ніколи не потрапляють на сервер, а лише об'єктивні дані, такі як ваги моделі. Це може бути особливо корисно для використання в CNN, оскільки воно забезпечує приватність та безпеку даних, не втрачаючи точність моделей машинного навчання.

Квантові нейронні мережі: квантові нейронні мережі (QNN) є злиттям квантової механіки та машинного навчання. QNN здатні моделювати взаємодію між квантовими системами або використовувати квантові

обчислювальні системи для швидкого виконання операцій машинного навчання.

Увага мереж: механізм уваги отримав значну популярність у сфері обробки природної мови, а особливо у трансформаторних моделях. Але зараз він широко застосовується у згорткових нейронних мережах для комп'ютерного зору. Механізм уваги дозволяє моделям зосереджувати на важливих частинах вхідних даних і ігнорувати менш важливі ділянки, що покращує їх ефективність та результативність.

Прозорість та пояснюваність AI: оскільки CNN стають все більш складними, виникає потреба в більшій прозорості та здатність пояснювати свої прогнози. Техніки візуалізації, такі як сенситивність до шару або оцінки активації класу (CAM), можуть допомогти «заглянути всередину» CNN і наочно представити, на які властивості вона звертається під час винесення рішень.

Ці передові техніки та підходи ілюструють широкий діапазон викликів та можливостей, які існують для згорткових нейронних мереж. Від федеративного навчання до квантових нейронних мереж, можна очікувати, що технологічні переваги продовжать розширювати можливості CNN та впливати на їхні застосування як в науці, так і в промисловості.

Згорткові нейронні мережі (CNN) являють собою підклас глибоких нейронних мереж (NN), які були спеціально розроблені для обробки даних з сітчастою топологією, особливо йде про потокові дані з багатовимірними вхідними параметрами, такими як кольорові зображення.

Перш за все, для розуміння CNN та їхнього навчання важливо зрозуміти загальну проблематику глибоких нейронних мереж. Нейронна мережа – це система машинного навчання, інспірована біологічними нейронами, які зустрічаються в мозку людини. Ці мережі є фундаментальними для глибокого навчання, галузі штучного інтелекту (AI), що націлена на імітацію способу мислення людини. В основу глибокого навчання покладено ідею використання великої кількості проміжних шарів

перетворення, що дозволяє моделям узагальнювати дані на більш високому рівні, в порівнянні з традиційними нейронними мережами.

Встановлення та налаштування глибокого навчання та Convolutional Neural Networks (CNN) вимагає понесення великої кількості обчислювальних витрат, велику кількість даних та необхідність експериментів з різними моделями, параметрами, функціями активації та нейронними архітектурами. Отже, є важливим зрозуміти, як CNN працюють та як їх навчають, особливо в контексті великих застосувань.

Перед тим, як розглядати навчання CNN, важливо згадати, що вони мають декілька основних складових, зокрема згорткові шари, пулінг, повнозв'язні шари та функцію активації. Загалом, незалежно від того, скільки шарів має мережа, основна ідея залишається однаковою: CNN навчається визначати конкретні шаблони в даних. Ця здатність розпізнавати шаблони робить їх надзвичайно корисними для завдань, зокрема в галузі комп'ютерного зору.

Глибоке навчання та CNN застосовуються для великої кількості задач, включаючи класифікацію зображень, семантичну сегментацію, виявлення об'єктів, розпізнавання голосу та навіть обробку природної мови. Їх використовують також у складних завданнях, що вимагають розпізнавання темпорально просторових шаблонів, таких як аналіз відео або прогнозування часового ряду.

Для налаштування CNN використовуються такі алгоритми, як зворотне поширення помилки (Backpropagation), стохастичний градієнтний спуск (SGD), Адам (Adam), Ададельта (Adadelta), РМСпроп (RMSprop) та ін., що дозволяє відповідно оновлювати параметри мережі з урахуванням градієнтів втрат.

Однак, відзначимо, що CNN лише на початку свого розвитку і відкривають нові можливості у галузі штучного інтелекту. Продовження статті буде в наступному повідомленні.

Суть процесу навчання Convolutional Neural Networks (CNN) полягає в оптимізації внутрішніх параметрів мережі на основі вхідного набору даних, та досягаючи найкращого можливого виконання на цих даних. Величезна кількість параметрів у CNN дає величезну гнучкість, але також може призвести до важкого перенавчання, якщо дані обмежені або якщо не використовуються правильні методи регуляризації. Однак, використовуючи великі масиви даних і передові методи оптимізації, можливо навчити CNN, які володіють вражаючими можливостями у виявленні та класифікації складних шаблонів.

Фундаментально процес глибокого навчання базується на оптимізації складових CNN з використанням інструментів, таких як градієнтний спуск та його модифікації, що слугують для мінімізації функції втрат на наборі даних. Основна ціль використання цієї процедури полягає в тому, щоб визначити фундаментальний розподіл, який стоїть за даними, з якими маємо справу, та виробити ємний розподіл, здатний та готовий узагальнити необхідну інформацію.

Головним завданням в навчанні CNN, як і в будь-якій іншій глибокій нейронній мережі, є знайти оптимальне значення ваг та зсувів, що мінімізують втрати на навчальному корпусі. Це досягається за допомогою алгоритмів, таких як стохастичний градієнтний спуск (SGD), що пошукують глобальний мінімум втрат, поступово оновлюючи ваги в декількох ітераціях. Кожне оновлення відбувається за допомогою зворотного поширення. В цьому контексті важливою є регуляризація, яка надає можливість контролювати перенавчання та стабільність навчального процесу.

Важливо пам'ятати, що хоча ми можемо отримати дуже складну модель з величезною кількістю ваг, наша модель просто перенавчиться, якщо використовується недостатня кількість даних. В даному контексті важливим є принцип Оккама, який віддає перевагу простішим моделям при рівній продуктивності. Це вимагає від нас використання достатньої

кількості даних та використання регуляризаційних технік під час навчання наших моделей.

Безсумнівно, важливим кроком в процесі навчання CNN є процес оптимізації. Градієнтний спуск (та його варіації, такі як стохастичний градієнтний спуск та градієнтний спуск з міні-партіями) – це ітераційний алгоритм оптимізації, який застосовується для надходження та мінімізації функції втрат. Ідея полягає у використанні градієнтного спуску для шукання таких ваг та зміщень, які мінімізують втрати. Кожна ітерація з градієнтним спуском зводиться до обчислення вектора антиградієнта для функції втрат з врахуванням поточних параметрів, а потім оновлення параметрів за допомогою руху в напрямку від'ємного градієнта.

Однак, виключно градієнтний спуск може бути недостатньо ефективним, якщо ми маємо справу з великими наборами даних або якщо функція втрат має складну топографію. Тому існують варіації алгоритму, такі як стохастичний градієнтний спуск (SGD), або його доповнення, що включає метод інерції, яка враховує попередні оновлення ваг, адаптивний градієнтний спуск (Adagrad), що враховує різні швидкості оновлення для різних параметрів, та адаптивні методи, такі як RMSProp та Adam, які є вдосконаленнями наведених попередньо методів.

Втім, існує проблема з вихідними просторами даних, які є неоднорідними або невиправдано великими, що може призвести до оптимізації у локальних мінімумах, а не глобальних мінімумах. Це веде до необхідності впровадження більш вивчених методів оптимізації, зокрема, використання величезної кількості даних, що може призвести до збільшення витрат на оптимізацію.

Коли ми говоримо про навчання згорткових нейронних мереж, варто враховувати його ключові етапи. По-перше, в процесі переднього проходу, вхідний сигнал проходить через всі шари згорткової нейронної мережі вплоть до виходу. Під час цього проходу, мережа використовує ваги та зміщення осередків кожного згорткового шару для обчислення вихідного

сигналу. Тоді прогнозоване значення порівнюється зі значенням мітки та від неї віднімається. В більшості випадків використовується квадратична функція втрат, а отже, втрати розраховуються як квадрат різниці.

Після цього розпочинається зворотний процес, згорткова нейронна мережа вносить корективи в навчальні параметри, щоб мінімізувати ці втрати. Спочатку помилки передаються назад через мережу. Потім, для кожного вагового параметру, ця помилка використовується для оновлення ваги відповідно до прямого впливу цього параметра на втрати.

Під час навчання Convolutional Neural Networks, алгоритм Stochastic Gradient Descent (SGD) або його вдосконалені версії, такі як Adam або RMSProp, використовуються для оновлення вагів та зміщень кожного нейрона в залежності від градієнта втрат по відношенню до цих параметрів. Під час навчання, процес зворотного розповсюдження помилки є основним для обчислення цих градієнтів.

Важливим заданням в процесі навчання CNN є запобігання перенавчанню, що виникає, коли мережа занадто добре навчається на навчальних даних, але погано на нових даних. Це може бути засоблено різними методами, такими як рання зупинка, крос-валідація, L1 та L2 регуляризації або крапковий випадок.

На завершення, процес навчання продовжується до тих пір, поки втрати не стануть фіксованими або процес не досягне максимальної кількості епох, що є просто кількістю разів, коли весь набір даних перебирається мережею. Цей процес вважається повним, коли втрати досягають обмеженої (мінімальної) величини або коли вони перестають зменшуватись.

Звертаючи вашу увагу на складність процесу навчання та на те, що відкидання кожного ваги та зміщення може впливати на втрати та правильність мережі, ми можемо уявити, наскільки цей процес може бути складним та потребувати значних обчислювальних ресурсів. Однак, саме це

й робить навчання CNN таким потужним підходом до розпізнавання образів та інших завдань. На цьому цей пункт має завершуватись.

Окрім основних методів, представлених в попередніх розділах, існують інші техніки, що допомагають забезпечити ефективність навчання глибоких нейронних мереж, зокрема згорткових. Одним з них є передача навчання (Transfer Learning), цінний метод, де знання з однієї задачі переносяться на іншу.

Вона зокрема корисна в ситуаціях, коли доступна недостатня кількість даних для ефективного навчання глибоких моделей з нуля. Одним з загальноприйнятих підходів є використання великих передтренірованих мереж, таких як VGG16, ResNet або Inception, як вихідної точки, яка дозволяє використовувати загальні шаблони, виявлені на великому наборі даних, наприклад, ImageNet, а потім налаштовувати мережу для виконання специфічних завдань.

Інший важливий аспект щодо навчання CNN – це те, що навіть найпростіші мережі мають мільйони параметрів, які слід оптимізувати, а навчання цих моделей вимагає великої кількості обчислювальних ресурсів, часто здійснюваних на графічних процесорах (GPU). Для великих моделей обчислювальні витрати можуть бути величезними, тому що вони потребують величезного обсягу пам'яті для зберігання ваг та вихідних карт активації для всіх шарів під час навчання.

Потреби обчислювальних ресурсів асоціюються із викликами розміщення та приміщення. Існують підходи до розробки та зменшення глибокого навчання, що включає в себе побудову дрібних моделей з такою ж точністю, альтернативні обчислювальні платформи та апаратні рішення.

Окрім вищенаведеного, потрібно враховувати й інші складнощі, пов'язані з навчанням CNN, зокрема вибір відповідних бібліотек та фреймворків глибокого навчання, що мають вплив на перспективи виконання та дослідницькі можливості.

Навчання CNN – це викликний процес, який потребує знань з теорії нейронних мереж, знань про даний набір даних та вміння використовувати відповідні інструменти та фреймворки. Однак, правильно налаштовані згорткові нейронні мережі здатні надати вражаючі результати в задачах переносу стилів, класифікації зображень, сегментації та багатьох інших застосуваннях.

На цій нотатці ми завершуємо огляд навчання згорткових нейронних мереж. Ця глибина та складність цієї теми підтверджує неймовірну потужність цих моделей та велику кількість досліджень, які ще потрібно провести, щоб оптимізувати їхню ефективність та спростити процес їхнього навчання.

ResNet, що є скороченням від Residual Networks, – це одна з найпопулярніших архітектур глибокого навчання, яка вперше була представлена в роботі «Deep Residual Learning for Image Recognition» на конференції CVPR 2016 року. ResNet-50 – це специфічний варіант архітектури ResNet, який має 50 шарів, включаючи згорткові шари, шари збільшення та повнозв'язні шари.

Основна ідея ResNet полягає в введенні «швидких з'єднань» або «залишкових з'єднань», що дозволяють градієнтам безпосередньо проходити через багато шарів. Це допомагає запобігти проблемі зниклого градієнта, що часто виникає при навчанні глибоких мереж.

AlexNet – це одна з перших глибоких мереж, яка показала, що глибоке навчання може бути дуже ефективним алгоритмом для виявлення об'єктів на зображеннях. Вона була представлена в 2012 році на міжнародному конкурсі ImageNet, де вона значно перевершила всі інші алгоритми.

AlexNet складається з восьми шарів – п'яти конволюційних та трьох повнозв'язних. Архітектуру схематично поділяють на дві частини: перша частина відповідає за виявлення особливостей за допомогою згортки та пулінгу, а друга частина відповідає за класифікацію.

LeNet-5, запропонована Яном ЛеКуном у 1998 році, часто вважається першою справжньою конволюційною нейронною мережею, оскільки вона з'єднує всі ключові компоненти сучасних CNN.

Мережа складається з двох частин: просторових модулів (тобто модулів локального згортання та підвиборки), які сприяють виявленню особливостей, а також повнозв'язного шару, що використовує їх для класифікації. Ця архітектура була в основному використана для розпізнавання рукописних цифр.

Ці три архітектури CNN є важливими віхами в розвитку глибокого навчання та комп'ютерного бачення, і вони стали основою для багатьох сучасних архітектур глибокого навчання.

Глибокі нейронні мережі – це варіант алгоритмів машинного навчання, оснований на принципах, впроваджених у нейронних мережах, але істотно розширений та ускладнений. Вони представляють собою складну систему з великою кількістю налаштовуваних параметрів, і їх організація включає глибокий стек обчислювальних шарів, з яких кожний виводить особливості на основі виводу попереднього шару.

Серед лідерів в галузі глибокого навчання можна виділити деякі промислові гіганти, такі як Google, Amazon, Microsoft, які активно розвивають і впроваджують алгоритми, засновані на глибоких нейронних мережах. Вони використовують глибокі нейронні мережі в різноманітних додатках, включаючи розпізнавання мови, машинний переклад, розпізнавання зображень і відео, і багато інших.

Однією з найбільших переваг глибоких нейронних мереж є їхня здібність до ефективного виявлення складних шаблонів в даних. Вони здатні виявляти шаблони на різних рівнях абстракції, від простих до складних, що робить їх ефективними для розпізнавання образів.

Проте, глибокі нейронні мережі вимагають великої кількості обчислювальних ресурсів і часу на навчання. Це може є викликом при реальному часі виконання, особливо коли дані постійно змінюються і

система повинна швидко навчатися. Оновлення ваг нейронної мережі при зміні даних може бути повільним процесом, що зменшує спроможність адаптації до нових шаблонів у зображеннях.

Глибокі нейронні мережі досить ефективні для стабільних або постійних наборів даних, коли модель може бути заздалегідь натренована на великому обсязі даних та потім застосована для вирішення конкретних завдань. Однак, в умовах, коли дані можуть постійно змінюватись або коли виникають нові шаблони в зображеннях, система повинна швидко навчатись адаптуватися до нових умов і викликів. В такому випадку, глибокі нейронні мережі можуть не бути ідеальним рішенням через їх тенденцію до повільного навчання.

Зазначимо також, що глибокі нейронні мережі, зазвичай, вимагають великого обсягу даних для навчання. Якщо кошти або доступні дані обмежені, можуть використовуватись інші методи машинного навчання, такі як опорні вектори, лінійна регресія, логістична регресія, або методи статистичного навчання.

Отже, тоді виникає питання: які альтернативи існують для глибоких нейронних мереж. Одна можлива альтернатива – це методи постійного навчання. Цей підхід покликаний розв'язати проблему повільного навчання глибоких нейронних мереж.

Замість перетренування на наборі даних від початку до кінця, постійне навчання адаптує навчальну модель до нових даних по мірі їх надходження. Це означає, що машина може розвиватися та продовжувати навчання на основі нових даних без необхідності викидати та знову перенавчати модель від самого початку. Це може бути особливо корисним у ситуаціях, коли потрібно реагувати на зміни у реальному часі.

У великій кількості застосувань глибокі нейронні мережі виявилися ефективними та досягли значних успіхів. Проте їхній вплив дещо обмежений у складних середовищах в реальному часі, де сила адаптації до неочікуваних змін стає критичною.

У вищезгаданих умовах використання набуває актуальності принцип навчання з підкріпленням. Це метод навчання штучного інтелекту, де машина вчиться робити дії, які максимізують кінцеву нагороду. Це включає навчання її взаємодії з навколишнім середовищем таким чином, щоб досягти бажаного результату. Ця ідея може бути використана для розробки систем, спроможних швидко підлаштовуватися до змін у даних.

Аналогічно, методи активного навчання можуть бути ще однією ефективною альтернативою. У такому підході, машина може сама обирати, які приклади їй треба вивчити наступними, що дозволяє їй краще управляти процесом навчання, і відповідно, прискорити навчання та адаптацію.

Останнім, але не менш важливим, є блокчейн технологія. Ця технологія може бути використана для надання прозорості і створення довіри між учасниками в процесі обміну даними. Це може бути корисним у ситуаціях, коли потрібно вести облік історії даних та забезпечити їх цілісність.

Всі ці альтернативи можуть бути розглянуті для розробки ефективних систем розпізнавання образів, які здатні швидко навчатися і адаптуватися до нових даних в реальному часі.

1.4 Метод опорних векторів

Метод опорних векторів (Support Vector Machines, SVM) є важливим інструментом в машинному навчанні. Його основна ідея полягає в пошуку гіперплощини в багатовимірному просторі, що максимально відокремлює класи даних. Однак, не завжди можна знайти лінійний розділяючий шар, що призводить до використання різних варіацій ядер.

Ядерні функції в SVM використовуються для перетворення вхідних даних у більш високі просторові розміри, де вищі демонстраційні рівні можуть бути більш дієвими для класифікації. Серед різних видів ядерних функцій, найбільш поширеними є:

– лінійне ядро. Це найпростіше ядро, і воно не вносить будь-яких змін у вхідні дані. Лінійні ядра є ефективними для великих наборів даних з великим числом ознак;

– поліноміальне ядро. Поліноміальне ядро створює не тільки особливості, що входять на основі добутку ознак, але й квадратичні, кубічні та вищі ступені властивостей оригіналу на основі ступеня полінома;

– радіальна функція базису (RBF) або Гауссове ядро. RBF являє собою загальноприйняте ядро, тому що воно може реформувати простір у невизначену кількість вимірів. Ядро RBF дуже гнучке і може створити нелінійні межі рішень;

– сигмоїдальне ядро. Сигмоїдальні ядра використовуються для відтворення нейронних мереж, використовуючи двошаровий схему персептрона, і повертаються в основному до використання в проблемах бінарної класифікації.

Використання різних варіацій ядер в SVM, насправді, залежить від характеру та обсягу набору даних. Завдяки різноманітності доступних варіацій ядер, SVM може бути дуже гнучким інструментом для різноманітної кількості проблем класифікації та регресії.

Метод опорних векторів (SVM) є одним з провідних алгоритмів в машинному навчанні для класифікації, регресії та виявлення аномалій. Запропонований коралівом Вапником у 1995 році, SVM став важливою частиною машинного навчання.

Основний принцип SVM полягає у визначенні гіперплощини, яка найкраще розділяє набір точок даних на дві категорії (рисунок 1.1). Наївний підхід до створення такої гіперплощини може просто вибрати лінію, яка розділяє точки. Однак, SVM використовує метод опорних векторів, оптимізує цей процес, максимізуючи відступ, тобто мінімальну відстань від лінії до найближчих точок.

Щоб зрозуміти, як SVM працює, слід розглянути три ключові елементи в його роботі: гіперплощини, відступи та опорні вектори.

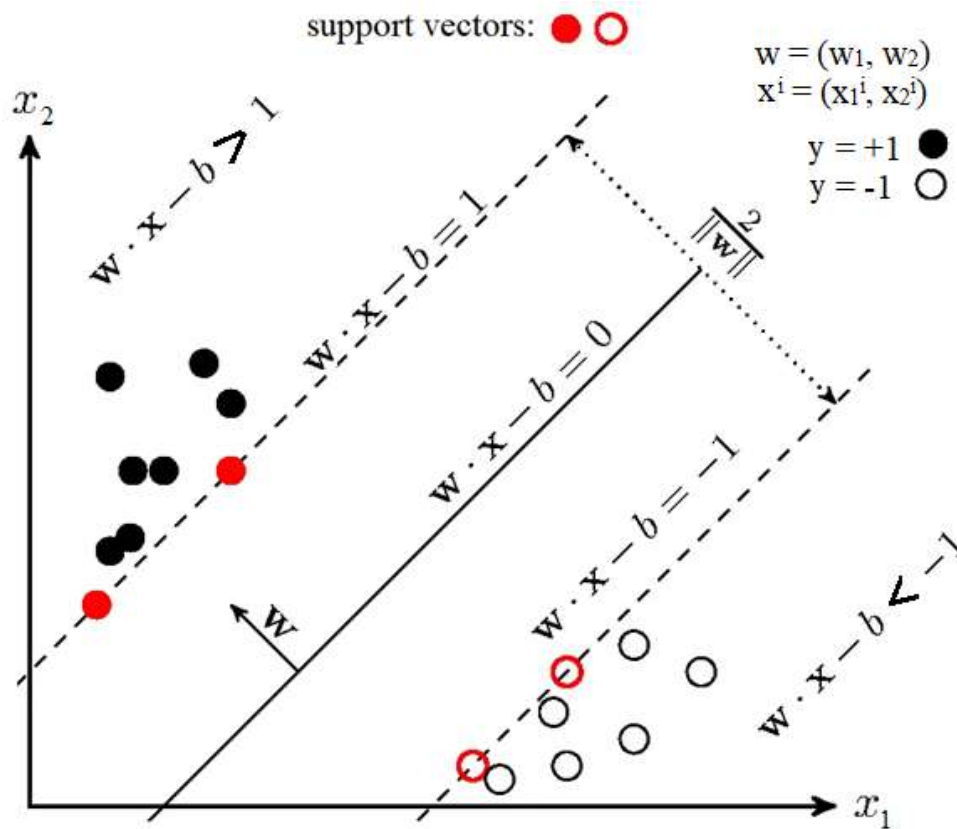


Рисунок 1.1 – Метод опорних векторів

Гіперплощина – це набір точок, що становлять розділювач між 2 класами точок у багатовимірному просторі – це ключова концепція в SVM. У двовимірному просторі гіперплощиною буде лінія, у тривимірному – площина, а в більш високих вимірах – вона буде має більші виміри.

Відступи відіграють важливу роль в SVM, вони визначають відстань між гіперплощиною та найближчими до неї точками з кожного класу. Ці точки називаються опорними векторами.

Опорні вектори – це набір точок, які є найближче до гіперплощини. Вони визначають положення та орієнтацію гіперплощини, оптимізуючи її положення так, щоб максимізувати відстань (відступ) до найближчих точок.

На практиці, коли роздільна гіперплощина не є лінійною, SVM використовує ядрові функції для нелінійної проекції даних у високовимірний простір, де лінійний розділювач може бути знайдений.

Різні ядерні функції можуть бути використані в залежності від природи даних.

Зрештою, SVM обчислює рішення, оптимізуючи двійкову цільову функцію за допомогою методів квадратичного програмування, що гарантує знаходження глобального мінімуму.

Метод опорних векторів вважається потужним механізмом аналізу даних і його застосування у просторі комп'ютерного бачення і зокрема в розпізнаванні образів, з'являється перспективним і ефективним. У підставі методології SVM лежить пошук оптимальної гіперплощини, яка розділяє класи даних найкращим чином. Застосування цього концепту до обробки зображень вимагає кількох спеціальних методів.

Розглянемо типову ситуацію застосування SVM для обробки зображень, і припустимо, що ми маємо завдання бінарної класифікації, наприклад, ми хочемо відділити зображення собак від зображень котів.

Спочатку, зображення представляються як матриці пікселів. Кожен піксель у цьому контексті може сприйматися як окрема ознака, що входить до моделі SVM. Однак в такому форматі дані можуть мати надмірно високу розмірність. Щоб зменшити розмір ознакового простору, застосовуються методи зменшення розмірності, такі як аналіз головних компонент (PCA).

Після цього зображення вже можуть бути перетворені на вектори ознак, і SVM можна навчати. Клас зображення використовується як мітка класу, а вектори ознак – це вхідні дані, які SVM класифікує. Кожен вектор ознак вставляється на гіперплощину, і відстань між додатними та від'ємними прикладами максимізується.

Тоді, з урахуванням того, що реальні дані рідко є лінійно розділими, для роботи з даними зображень корисна стратегія ядер SVM, така як радіальні базисні функції (RBF), які відображають вісь вхідних ознак у новий простір вищих розмірностей, де дані можуть стати лінійно розділими

В цьому високорозмірному просторі можна знайти гіперплощину, яка найкраще розділяє класи. Ця гіперплощина потім зображується назад у вхідний простір ознак, створюючи нелінійний розділювач.

Після навчання модель SVM може бути використана для класифікації нових зображень, перетворення зображень на вектори ознак, вставку цих векторів на гіперплощину і визначення класу зображення на основі положення відносно гіперплощини.

Метод опорних векторів – це надзвичайно потужний алгоритм у машинному навчанні, однак як і будь-який інший алгоритм, він має свої переваги та недоліки. Розглянемо їх детальніше.

Переваги SVM:

– ефективні в просторах високої розмірності: SVM працює добре, коли є багато ознак. Наприклад, для тексту і зображень кількість ознак може перевищувати кількість вибірок, і SVM може застосовуватися ефективно;

– гнучкість за допомогою ядерних функцій: вбудоване ядерне перетворення дозволяє вибирати те, як будуть порівнюватися окремі вибірки. Це дозволяє особливу гнучкість, оскільки можуть бути використані лінійні, поліноміальні, радіальні або сигмоїдні ядра в залежності від завдання класифікації;

– робує щодо шуму: згідно з принципом максимізації відступів, SVM має тенденцію бути робастним до шуму у відкриваючи границі рішень, оскільки вони засновані лише на наборі опорних векторів, а не на всіх даних.

Недоліки SVM:

– вибір ядрової функції та параметрів: швидкість навчання та якість рішень в SVM сильно залежать від вибору ядра та його параметрів. Нажаль, вибір оптимальної функції і параметрів ядра часто потребує складного поняття про дані і може потребувати випробування і помилок;

– складність розуміння та інтерпретації: фінальна модель SVM (особливо з нелінійним ядром) може бути важко інтерпретувати. Це

може бути великим недоліком в областях, де йдеться про розуміння причинно-наслідкових зв'язків;

– не ефективний для великих датасетів: обчислювальна та пам'ятна складність квадратична по відношенню до розміру навчального набору, що робить SVM непрактичним для великих наборів даних.

SVM пропонує ряд заманюючих переваг, але потребує обережного розуміння та реалізації. І хоча вони відіграють важливу роль в пейзажі машинного навчання, їх практичне застосування повинне враховувати відповідні проблеми та обмеження.

1.5 Алгоритм k-найближчих сусідів

Алгоритм k-найближчих сусідів (KNN, k-nearest neighbors) — це тип алгоритму машинного навчання, який використовується для класифікації та регресії. Він відноситься до сімейства моделей з наглядом навчання, які визначають клас або властивості об'єкта на основі спільності з найближчими прикладами в навчальних даних.

Основний принцип роботи KNN досить простий: використовується відстань або схожість між об'єктами. Коли модель потрібно визначити клас нового об'єкта, вона вираховує його відстань до всіх інших об'єктів у навчальному наборі, знаходить k об'єктів, які є найближчими сусідами, і визначає клас підсумкового об'єкта відповідно до класів найчастіших сусідів.

Відстань між об'єктами може вираховуватися різними способами. Найпоширеніші метрики відстані – це евклідова, манхеттенська і Чебишева. Вибір відстані залежить від природи даних. Кількість найближчих сусідів, k, є гіперпараметром, який потрібно вибрати.

Зауважте, що алгоритм KNN є «лінивим» класифікатором. Це означає, що він не будує експліцитною модель на основі навчальних даних, але

замість цього зберігає всі навчальні приклади. Прогнози виробляються тільки на етапі тестування, коли подається новий приклад.

Незважаючи на свою простоту, цей алгоритм демонструє добру ефективність на великому числі завдань класифікації та регресії.

Метод k -найближчих сусідів є алгоритмом навчання з вчителем, який працює на простому принципі: об'єкти схожі об'єкти знаходяться близько один до одного. Алгоритм виконує навчання, визначаючи просто кількість й властивості найближчих сусідів кожного об'єкта і класифікуючи его за більшістю голосів його сусідів.

Процес роботи алгоритму KNN може бути описаний наступним чином:

- визначення значення параметра K – це кількість найближчих сусідів, яку необхідно взяти до уваги під час прийняття рішення;

- розрахунок відстаней від кожного з об'єктів навчального набору до об'єкта, який потрібно класифікувати. Відстань можна визначати різними способами, але найчастіше використовується евклідова відстань;

- визначення K найближчих сусідів на основі розрахованих відстаней. Залежно від завдання, якщо це завдання класифікації: об'єкт приписується до того класу, який є найбільш поширеним серед його K сусідів. У випадку завдання регресії: значення цільового атрибута об'єкта встановлюються як середнє або медіана цільових атрибутів K найближчих сусідів.

Загалом, KNN – це непараметричний метод, що не робить жодних припущень про розподіл даних. Його простота робить його досить зручним для застосування в реальних ситуаціях, але важливо зазначити, що якість роботи значною мірою залежить від вибору правильного значення K та відповідного способу виміру відстані між об'єктами.

Метод k -найближчих сусідів може бути використаний для класифікації зображень, який є важливою задачею в області комп'ютерного бачення. Процес роботи алгоритму KNN з зображеннями може бути описаний декількома кроками:

– перетворення зображень до числових векторів: зображення у цифровому форматі можна представити як матрицю пікселів, де кожен піксель має числове значення, що відповідає його яскравості. Коли зображення має кольорові канали, наприклад, червоний, зелений та синій, можна використати три матриці, одну для кожного каналу. Зображення можна перетворити на числовий вектор, розгортаючи кожен канал і об'єднуючи їх у вектор;

– визначення метрики відстані: для порівняння зображень потрібна метрика подібності або відстані. Зазвичай використовується евклідова відстань, але в залежності від особливостей задачі можуть бути використані інші метрики;

– пошук K найближчих сусідів: для нового зображення, яке необхідно класифікувати, виконується вимір відстані від цього нового зображення до всіх зображень в навчальному наборі. Знаходяться k найближчих сусідів (зображень), і видається прогноз на основі більшості класів цих сусідів.

Робота з зображеннями є викликом для алгоритму KNN через високо розмірний простір ознак, адже кожен піксель стає ознакою. На практиці для зменшення розмірності та збереження важливої інформації використовуються методи зменшення розмірності, такі як PCA (аналіз головних компонент) або можуть використовуватися інші техніки обробки зображень.

Метод k -найближчих сусідів, – це багатofункціональний алгоритм машинного навчання, який може бути використаний для класифікації, регресії та пошуку найменшої відстані. Однак, як і усі алгоритми машинного навчання, він має свої переваги та недоліки.

Переваги алгоритму KNN:

– простота: основна перевага алгоритму KNN полягає в його простоті. Він оснований на інтуїтивно зрозумілих принципах і легко інтерпретується;

- невимогливість до даних: KNN може бути застосований до широкого спектру типів даних, включаючи числові та категоріальні ознаки;

- використання статистики непараметричних: KNN не робить жодних припущень щодо розподілу даних та тому може бути корисним в ситуаціях, коли такі припущення важко зробити або коли вони не виконуються;

- здатність до адаптації: оскільки алгоритм KNN здійснює висновок на основі найближчого околу конкретного прикладу, він може адаптуватися до змін даних.

Недоліки алгоритму KNN:

- витрати на обчислення: оскільки KNN вимагає порівняння кожного прикладу з усіма іншими прикладами для роботи, він може бути обчислювально витратним, особливо при великих наборах даних;

- визначення відстані: визначення відповідної метрики відстані може бути складним, особливо коли ознаки мають різні типи (наприклад, категоріальні та неперервні) або коли вони мають змінювані значення;

- вибір K : вибір правильного значення K може бути непростим і вимагає перехресної перевірки та налаштування гіперпараметрів;

- нечутливість до нерівноваги класів: KNN може виявитися чутливим до нерівноваги класів. Це означає, що якщо певний клас переважає у навчальному наборі, нові приклади, ймовірно, будуть класифіковані як цей домінуючий клас. Методи машинного навчання стали невід'ємною складовою сучасної науки та технологій, дозволяючи комп'ютерам "навчатися" з даних та використовувати оброблену інформацію для прийняття інтелектуальних рішень. Вони мають широке застосування в багатьох галузях, включаючи обробку зображень, комп'ютерне зору, прийняття рішень, прогнозування, рекомендаційні системи, і багато іншого.

Головною перевагою методів машинного навчання є їхня швидкість та легкість налаштування. Вони, порівняно, прості з точки зору налаштування параметрів і можуть бути використані для об'єктів, що не

мають занадто складних властивостей. Це робить їх ідеально підходящими для багатьох завдань.

Однак, складнощі виникають, коли обробляються складні набори даних з великою кількістю перемінних, або коли вимагається висока точність прогнозування. В цьому випадку, методи машинного навчання можуть виявитися недостатньо ефективними.

Метод найближчого сусіда та метод опорних векторів є двома основними методами машинного навчання, які часто використовуються для класифікації зображень. KNN працює шляхом пошуку найближчих сусідів зразка в датасеті та використання їх класів для визначення класу зразка. SVM створює гіперплощину або набір гіперплощин в багатовимірному просторі, які можуть бути використані для класифікації.

Тим не менше, обидва цих методи стикаються з проблемами при великій вибірці. І KNN, і SVM потребують обчислення відстаней між парами об'єктів, що стає дедалі більш обчислювально вимогливим з розміром вибірки. Коли вибірка починає зростати, особливо в онлайн-режимі, коли дані потоково надходять на вхід, ці методи стають непрактичними.

В онлайн-режимі, де дані надходять послідовно, ми маємо здійснювати негайну інкрементну адаптацію моделі навчання, але це є складним завданням для багатьох методів машинного навчання. Зокрема, обчислювальні вимоги KNN і SVM залишаються високими незалежно від того, як великим є обсяг даних. Все це призводить до того, що ці методи можуть стати неефективними в ситуаціях, де необхідне швидке навчання і високий рівень адаптивності до змінившихся даних.

З іншого боку, деякі методи машинного навчання можуть бути використані для успішного розпізнавання образів в складних наборах даних. Наприклад, рандомні дерева рішень та ансамблі регресії або бустінгу можуть бути використані для побудови ефективних моделей класифікації образів, навіть у випадках великого обсягу даних.

Однак слід зазначити, що є і інші альтернативи для обробки великого обсягу даних. Однією з них є використання інтелектуального вибору даних, тобто вибору найбільш репрезентативних проб для навчання моделей. Це може допомогти зменшити кількість необхідних обчислень та покращити продуктивність.

Крім того, ще одним зрілим підходом для оброблення великого обсягу даних є онлайн-навчання. Цей підхід полягає в навчанні моделі на потоці даних, де модель оновлюється, коли надходять нові дані. Це робить його ідеально підходящим для ситуацій, де дані змінюються з часом.

Необхідно відзначити, що ці онлайн-методи можуть бути більш складними для використання, ніж звичайні методи машинного навчання. Наприклад, вони вимагають уміння працювати з повсякденно оновлюваними даними або потоками даних, та вбудовувати онлайн-алгоритми в екосистему, що теж можуть бути складними завданнями.

Крім того, деякі методи машинного навчання, такі як нейронні мережі та глибокі нейронні мережі, зазвичай мають високі апріорні вимоги до обсягу даних для навчання та великі обчислювальні вимоги, що може стати перешкодою для їх онлайн-застосувань.

Таким чином, важливо усвідомлювати, що різні методи машинного навчання мають свої власні сильні та слабкі сторони, і варто розглядати їхнє застосування в залежності від обсягу вхідних даних, оцінки складності проблеми та обмежень ресурсів.

1.6 Нечіткі системи, нечітка логіка

Поняття «нечіткість» в машинному навчанні та штучному інтелекті входить в нечітку логіку, підгалузь теорії множин, яка дозволяє більш гнучке представлення даних, покращуючи можливості моделі властивостей людського мислення, таких як невизначеність та неоднозначність.

Нечіткі множини відрізняються від традиційних, «чітких» множин тим, що елемент можуть належати до нечіткої множини з певною степінню належності, що вимірюється числом від 0 до 1. Така схема дає змогу розмито кодувати інформацію, що істотно полегшує обробку неоднозначних та непрямолінійних даних.

Нейро-фаззі системи, як ANFIS (адаптивна нейро-фаззі система вивчення інформації), є сполученням нейрональних мереж та нечіткої логіки. ANFIS використовує нейронну мережу для навчання та покращення системи нечіткого виведення.

Переваги ANFIS:

– уніфікація: ANFIS об'єднує в собі переваги нейронних мереж та нечіткої логіки, забезпечуючи гнучкість і адаптивність першої та логічну та інтерпретабельну структуру другої;

– навчання: здатність ANFIS навчатися з даними, що поліпшує її можливості прогнозування та адаптації;

– інтерпретація: ANFIS генерує чіткі правила, які легко інтерпретувати, що забезпечує прозорість рішень.

Недоліки:

– складність: ANFIS є досить складною системою, що може збільшити час обчислень і вимагає значних обчислювальних ресурсів;

– надмірне навчання: як і у всіх алгоритмах машинного навчання, ANFIS може страждати від проблеми «overfitting» – занадто підганяти дані, що веде до втрати здатності узагальнення на нових даних.

Кластеризація даних – це задача розподілу об'єктів на групи (кластери) таким чином, що об'єкти всередині одного кластеру є більш подібними між собою, ніж з об'єктами інших кластерів. Ця задача відноситься до наглядованого машинного навчання.

Однак, часто стає проблематично визначити, до якого кластеру належить нове спостереження або об'єкт, особливо, якщо об'єкт розташований на границі кількох кластерів. Це може виникнути внаслідок

шуму в даних, неправильно вибраного алгоритму кластеризації, невідповідних метрик подібності або невідповідної кількості кластерів.

Якщо вам потрібно визначити, до якого кластеру належить нове спостереження, яке знаходиться на границі кластерів, можна використати кілька методів. Один із них – використати алгоритм k -найближчих сусідів (KNN): знайти K найближчих точок до нового спостереження і визначити клас, до якого належить більшість з цих точок.

Ще один підхід може бути використання «м'якої» приналежності: замість визначення належності до одного конкретного кластеру, оцінювати ступінь належності до кожного кластеру.

1.7 Нео-нечіткі системи

Неофаззійний нейрон (NFN) представляє собою систему штучного інтелекту, яка поєднує особливості нечіткої логіки та нейронних мереж. Концепцію неофаззійного нейрона сформулювали вчені Shigeo Abe та Suliman Alsuhaibani у 1999 році для підтримки високої ступені адаптивності в логічних системах.

Неофаззійний нейрон складається з двох інтегрованих частин, передньої та задньої систем. Ключовим елементом передньої системи є FString, що відповідає за перетворення вхідних даних на ступені належності за допомогою функцій належності. Задня система має структуру навчання з використанням Дельта-правила на основі похибки між вхідним та цільовим значеннями.

У контексті алгоритмічної реалізації, робота з неофаззійним нейроном здійснюється таким чином. Спершу вхідні дані подаються на вхід неофаззійного нейрону. Далі функція належності обчислює ступінь належності для кожного вхідного значення відносно певного класу. Отримані значення асоціюються з вагами, що потребують навчання, та подаються на наступний шар системи. На виході формується вихідне

значення шару нейрона, що зважується за допомогою ваг, встановлених у процесі навчання. Кінцевий етап роботи алгоритму полягає у застосуванні Дельта-правила для корекції ваг до тих пір, поки вихідне значення не збігатиметься з цільовим.

Наукова цінність неофаззійного нейрона полягає в його гнучкості та адаптивності, яка досягається через комбінацію нечіткої логіки та принципів нейронних мереж. Процес застосування функції належності в передній системі нейрона дозволяє трансформувати вхідні дані до формату, який легше інтерпретувати та обробляти. Кожне значення вхідних даних отримує особистий ступінь належності, використовуючи розмиті границі замість чітких відсічок, як у більш традиційних методах. Це дозволяє більш ніжну, багаторівневу класифікацію, що може бути особливо корисним при роботі з неоднозначними або поганим даними.

На другому етапі, задня система використовує Дельта-правило для адаптації ваг відповідно до відмінності між вихідними даними і цільовим значенням. Цей алгоритм використовує градієнтний спуск для мінімізації помилки, що робить його ідеально підходящим для прогнозування у задачах регресії.

Використання неофаззійного нейрона в моделях машинного навчання може привести до збільшення точності прогнозування завдяки його здатності ефективно обробляти нечіткі та неоднозначні вхідні дані. Проте, слід враховувати, що під час використання таких моделей можуть виникнути певні виклики, такі як необхідність вибору підходящої функції належності та інтенсивний обчислювальний процес.

Висновком цього розділу може бути те, що неофаззійний нейрон представляє цікавий підхід до моделювання нечітких систем. Його здатність інтегрувати принципи нечіткої логіки та нейронних мереж робить його потужним інструментом для аналізу складних даних.

Залишаючись у контексті архітектури неофаззійного нейрона, спираючись на концепцію нечітких множин, можна зрозуміти, що вагові

коефіцієнти, які асоційовані з входами, повинні подавати наступному шару нейрона додатні числа в діапазоні від нуля до одиниці. Вагові коефіцієнти тут виступають як ступені членства вхідних даних в нечіткій множині, яка складається з універсального дискретного множинного простору.

Такий підхід надає важливу властивість, дозволяючи неофаззійному нейрону обробляти великі обсяги даних. У нейронних мережах звичайно використовуються функції активації, такі як сигмоїдні функції або гіперболический тангенс, для нормалізації вхідних даних. В нео-нечіткому нейроні вагові коефіцієнти вже є нормалізованими, тому не потрібно додаткового шару активації.

Визначення правильної функції належності є критичним для успіху роботи нео-нечіткого нейрона. Оскільки нечіткі шкали представляють ступені досвіду або упевненості, вибір функції належності, яка правильно відображає досвід або незавершеність даних, є важливим. У більшості випадків для цього використовують функції, що представляють собою гладкі криві, які відображають поступову зміну ступенів певного вхідного значення.

Часто в якості функції належності використовують гауссівську функцію, оскільки вона описує розподіл даних в багатьох реальних сценаріях. Однак, використання гауссівської функції може стати складним, оскільки не завжди легко визначити середнє значення та стандартне відхилення для вхідних даних. У такому випадку міг бути корисним інший тип функції належності, наприклад трикутна або трапецеподібна функція.

Окрім визначення функцій належності, відмітна роль надається праці з вагами неофаззійного нейрона. Загалом, в нейронних мережах, ваги виділяють ступінь важливості, що приділяється кожному входу в контексті його впливу на вихід. У випадку неофаззійного нейрона, ваги, що поступаються на вихідний шар, коригуються у відповідності до ступеня належності, вирахованого на основі вхідних даних.

Процес навчання неофаззійного нейрона полягає у коригуванні цих ваг з використанням, наприклад, методу Дельта-правила. Вважається, що цей метод працює ефективно, коли вихід системи порівнюється з цільовою характеристикою. Отримане відхилення, відображає певну похибку внаслідок різниці між фактичним та очікуваним виходами. Таке відхилення використовується для корекції ваг, що дозволяє навчати систему у напрямку мінімізації представленої похибки.

Розвідка перспектив використання неофаззійного нейрона в сучасних дослідженнях вказує на число потенційних областей застосування. По-перше, його використання може бути особливо корисним у ситуаціях, де у процесі обробки даних наявна значна кількість нечіткості. У таких випадках нео-нечіткий нейрон може поєднати переваги як нечіткої логіки, так і нейронних мереж, щоб ефективно врахувати нечіткість вхідних даних та краще моделювати неточність вихідних даних.

Неофаззійний нейрон може бути застосовний і в сферах, що вимагають високого ступеня точності при високій складності даних, наприклад у медичинському прогнозуванні або в системах розпізнавання образів. Його здатність до гнучкості та адаптивності робить цей засіб перспективним для постійно еволюціонуючих полів практичного застосування.

Неонечіткі системи відіграють суттєву роль у сучасних дослідженнях в області машинного навчання. Вони ґрунтуються на концепціях та принципах нечіткої логіки, які дозволяють моделювати неоднозначність та невідомість в більш природний для людини спосіб, порівняно з традиційними методами машинного навчання.

Неонечіткі системи застосовують набори лінгвістичних правил, що базуються на нечітких множинах для формування висновків на основі нечітких або неповних даних. Одним з найбільш поширених типів нечітких систем є нечіткі контролери, які застосовуються в широкому спектрі виробничих процесів, медіа, робототехніки, і так далі.

Головною перевагою нечітких систем є їх спроможність обробляти неточність та неоднозначність даних, що є загальною властивістю реального світу.

Особливу увагу в архітектурі нечітких систем приділяється модифікації та оптимізації членства функцій і правил нечіткої логіки. Більшість нечітких систем покладається на використання евристичних методів для налаштування цих компонентів. Це може включати традиційний ручний підхід, ітераційний метод або використання еволюційних алгоритмів.

Недоліком нечітких систем є їхня відносна складність та непрозорість в порівнянні з більш традиційними моделями машинного навчання. Їх проектування та тестування можуть вимагати значного ступеня експертизи. Також вони можуть вимагати значні обчислювальні ресурси залежно від обсягу та складності вхідних даних.

У контексті розпізнавання образів, нечіткі системи можуть швидко навчатись в онлайн режимі, адаптуючись до нових масивів даних, і можуть генерувати більш детальний опис властивостей об'єктів, схованих в зображеннях. Вони спроможні вимірювати просторово-тимчасову неоднозначність та обробляти нечіткість об'єктно-орієнтованих характеристик, що дає їм перевагу в задачах розпізнавання образів порівняно з більш традиційними методами.

Відмінна адаптивність нечітких систем до змін у даних може бути особливо корисною у режимі реального часу. Це може мати ряд застосувань, включаючи системи відеоспостереження, системи навігації, системи розпізнавання об'єктів у реальному часі.

Ще однією значною перевагою нечітких систем є можливість апроксимувати будь-яку неперервну функцію на визначених множинах з будь-яким заданим ступенем точності. Це робить їх надзвичайно гнучкими та придатними для різноманітних задач обробки даних, де інші, більш традиційні методи можуть зазнати труднощів.

Щодо недоліків, одним з них є висока складність алгоритмів нечіткої логіки, що призводить до високої обчислювальної складності і потреби в значних обчислювальних ресурсах. Іншим недоліком є потреба в експертних знаннях для створення та оптимізації нечітких систем, що може перешкоджати їх широкому поширенню.

У певних нішевих умовах, наприклад, коли обробляються зображення з високим рівнем шуму, або коли дані неповні або нечіткі, неонечіткі системи можуть показати виняткову ефективність. Вони також можуть бути ефективними у системах, де потрібно швидко адаптуватися до змінних умов, наприклад, у вибушових середовищах або при роботі з непередбачуваною динамікою.

1.8 Нечітка логіка у розпізнаванні образів

Під час глибокого вивчення теорії розпізнавання образів, можна зустріти різноманітні методи та техніки, що використовуються для вирішення цієї складної задачі. Одним з цих методів є використання нечіткої логіки, що розглядається як нечіткі методи розпізнавання образів.

Центральна концепція нечіткої логіки полягає в використанні математичного апарату, який дозволяє працювати з нечіткими множинами. Нечітке множество – це така множина, що об'єкт може належати їй з певним ступенем належності, який вимірюється числом від 0 до 1.

Нечіткі методи розпізнавання образів використовують цю здатність оперувати з розподіленими, нечіткими або неточними даними, що дозволяє їм ефективно працювати з вхідними даними низької якості та об'єктами, що мають схожі характеристики.

В загальному випадку, нечіткі методи розпізнавання образів розраховані на вирішення двох основних задач: класифікації та сегментації зображень. Ці два напрямки вимагають врахування різних аспектів проблеми розпізнавання образів.

Класифікація зображень – це процес призначення кожному об'єкту на зображенні певного класу. Головною проблемою при цьому є розробка нечіткої моделі, яка б дозволила визначити, до якого класу відноситься кожен піксель.

З іншої сторони, сегментація зображень полягає в розділенні зображення на регіони або об'єкти, що мають схожі характеристики. Головною проблемою тут є визначення границь між різними об'єктами.

Для вирішення цих завдань використовують різні методи, засновані на нечіткій логіці. Наприклад, для класифікації образів можуть бути використані нечіткі мережі Кохонена або нечіткі артмапи. Ці мережі навчаються розподіляти вхідні дані по різних класах на основі їх нечіткої належності до різних категорій.

Нечітка логіка є методом обробки нечітких, неоднозначних або неповних даних. Вона була запропонована у 1965 році Lotfi Zadeh, як метода для моделювання неоднозначності та невизначеності в даних. У контексті розпізнавання образів, нечітка логіка може бути використана для виявлення шаблонів у зображеннях, враховуючи можливі відхилення та невизначеності.

Нечіткий нейрон – це нейрон в нейронній мережі, який використовує нечітку логіку для обробки вхідних даних. Наприклад, замість того, щоб просто визначити, чи є об'єкт на зображенні, нечіткий нейрон може дати вивід у вигляді степені належності об'єкта до певного класу.

Адаптивний подвійний нечіткий нейрон вважається розширенням стандартного нечіткого нейрона. Він включає два нечітких нейрона, що працюють разом, для обробки вхідних даних. Одне з основних покращень, які пропонує цей підхід, полягає в тому, що він дозволяє більш гнучко моделювати складні, неоднорідні дані.

Наприклад, при розпізнаванні образів, один нечіткий нейрон може взяти на себе завдання визначення форми об'єкта, тоді як інший може бути використаний для визначення кольору. Подвійний нечіткий нейрон може

адаптувати свої ваги незалежно один від одного, що дозволяє йому краще моделювати взаємозв'язок між цими характеристиками об'єкта.

Адаптивність цих подвійних нечітких нейронів також підвищує їхню ефективність. Завдяки здатності самостійно корегувати свої ваги протягом процесу навчання, вони можуть більш ефективно визначати нечіткі шаблони в даних, ніж традиційні штучні нейрони.

Таким чином, використання нечіткої логіки, а особливо адаптивного подвійного нечіткого нейрона, може значно покращити можливості системи розпізнавання образів.

Для сегментації образів може бути застосована нечітка кластеризація, що включає метод с-середніх та його варіації. Нечітка кластеризація враховує неоднозначність класифікації, тобто що об'єкт може належати до кількох кластерів з різними ступенями належності. Після поділу зображення на кластери, кожний кластер може відповідати певному регіону на зображенні.

Особливості алгоритму нечітких с-середніх полягають в розрахунку ступеня належності кожного пікселя до кожного кластера та подальшому перерахунку центрів кластерів, використовуючи вагові коефіцієнти, отримані на попередніх кроках. Процес повторюється до тих пір, поки не буде досягнуто заздалегідь визначену умову зупинки.

Роль нечітких методів в розпізнаванні образів полягає в можливості обробки даних, що мають високу ступінь нечіткості, завдяки використанню принципів нечіткої логіки. оскільки розпізнавання образів часто працює з суперечливими або неточними даними, нечіткі методи можуть виявитися особливо корисними в таких сценаріях.

Нечіткі методи розпізнавання образів можуть застосовуватись у широкому спектрі областей. Наприклад, у медицині їх можна використовувати для аналізу зображень та виявлення певних патологій. У геології та агрономії вони можуть бути корисні для аналізу супутникових зображень, щоб визначити тип покриття землі або виявити зміни в

рослинності. Крім того, вони можуть бути використані в системах розпізнавання осіб для ідентифікації людей за їхніми фотографіями чи відеозаписами.

Сьогодні розпізнавання образів посідає важливу роль у сфері машинного навчання та штучного інтелекту. Відповідно до складності та невизначеності інформації, що міститься в образах, нечіткі методи розпізнавання образів надають засоби для ефективного вирішення цієї проблеми.

Робота з нечіткими методами включає розгляд таких речей, як очищення даних, екстракція ознак, визначення відповідного простору для аналізу, класифікація, перевірка та наступна ітерація. Спочатку образи проходять етап попередньої обробки, що включає такі етапи, як згладжування, декомпозиція, фільтрація тощо, для підготовки даних до подальшого аналізу. Після цього проводиться екстракція ознак, в процесі якої зображення перетворюється на вектор ознак. Далі йде визначення відповідного простору для аналізу, де розглядаються фактори, такі як роздільна здатність, контраст, освітленість тощо. Згодом проводиться класифікація з використанням нечіткого алгоритму. На заключному етапі виводиться результат розпізнавання образу.

Використання нечітких методів у розпізнаванні образів передусім дозволяє виконувати задачі класифікації та сегментації безпосередньо на основі пікселів зображень. Такі завдання зазвичай виконуються в декілька етапів: спочатку відбувається попередня обробка та екстракція ознак, потім – проведення класифікації або інтерпретації даних. Однак, таке послідовне виконання завдань може призвести до втрати важливої інформації. Натомість нечіткі методи дозволяють проводити класифікацію та сегментацію безпосередньо на основі пікселів зображень, що дозволяє зберегти більше інформації та отримати більш точні результати.

Нечіткі методи розпізнавання образів також надають додаткову гнучкість, оскільки дозволяють робити розмиті висновки на основі

неоднозначних чи неповних даних. Ця властивість виявляється особливо корисною при роботі з неякісними даними або при розпізнаванні образів, що значною мірою залежить від контексту.

Розглядаючи переваги використання нечітких методів в розпізнаванні образів, слід зазначити їх спроможність до високого ступеня узагальнення. Нечіткі методи розроблені таким чином, що вони мають властивість узагальнення навіть при мінімумі вхідних даних. Це робить їх досить сильною стратегією для розпізнавання образів у випадках потреби навчання з обмеженим обсягом даних чи незначною кількістю прикладів.

Крім того, нечіткі методи забезпечують здатність моделі до адаптації. Оскільки нечіткі методи базуються на представленні реального світу через нечіткі множини, вони володіють здатністю до адаптації до змінних умов. Функції застосування в нечітких множинах можуть бути модифіковані відповідно до зміни контексту чи вхідних даних.

Оскільки розпізнавання образів є важливою дисципліною в галузі комп'ютерного зору і машинного навчання, можна впевнено стверджувати, що нечіткі методи розпізнавання образів відіграють важливу роль і продовжатимуть розвиватись в подальшому. Вони пропонують перспективний напрямок для вдосконалення прецизійності та надійності систем розпізнавання образів за рахунок використання принципів нечіткої логіки для обробки нечіткості та невизначеності, які є властивими для реального світу.

Однак, з усіма своїми перевагами, нечіткі методи розпізнавання образів також викликають деякі виклики. Це можуть бути Методи обчислень, використовуючи нечіткі множини, можуть стати більш складними і ресурсомісткими, ніж традиційні методи. До того ж, визначення функцій належності для нечітких множин може стати нетривіальною задачею, що вимагає глибокого розуміння характеристик даних.

Тому, розробка ефективних нечітких методів розпізнавання образів виносить на передній план необхідність розробки нових методів, що

дозволили б зменшити обчислювальну складність, при цьому зберігаючи високу точність і адаптивність моделі.

1.9 Адаптивний подвійний нечіткий нейрон: принципи та використання у розпізнаванні образів

Нечітка логіка є методом обробки нечітких, неоднозначних або неповних даних. Вона була запропонована у 1965 році Lotfi Zadeh, як метода для моделювання неоднозначності та невизначеності в даних. У контексті розпізнавання образів, нечітка логіка може бути використана для виявлення шаблонів у зображеннях, враховуючи можливі відхилення та невизначеності.

Нечіткий нейрон – це нейрон в нейронній мережі, який використовує нечітку логіку для обробки вхідних даних. Наприклад, замість того, щоб просто визначити, чи є об'єкт на зображенні, нечіткий нейрон може дати вивід у вигляді степені належності об'єкта до певного класу.

Адаптивний подвійний нечіткий нейрон вважається розширенням стандартного нечіткого нейрона. Він включає два нечітких нейрона, що працюють разом, для обробки вхідних даних. Одне з основних покращень, які пропонує цей підхід, полягає в тому, що він дозволяє більш гнучко моделювати складні, неоднорідні дані.

Наприклад, при розпізнаванні образів, один нечіткий нейрон може взяти на себе завдання визначення форми об'єкта, тоді як інший може бути використаний для визначення кольору. Подвійний нечіткий нейрон може адаптувати свої ваги незалежно один від одного, що дозволяє йому краще моделювати взаємозв'язок між цими характеристиками об'єкта.

Адаптивність цих подвійних нечітких нейронів також підвищує їхню ефективність. Завдяки здатності самостійно корегувати свої ваги протягом процесу навчання, вони можуть більш ефективно визначати нечіткі шаблони в даних, ніж традиційні штучні нейрони.

Таким чином, використання нечіткої логіки, а особливо адаптивного подвійного нечіткого нейрона, може значно покращити можливості системи розпізнавання образів.

1.10 Постановка задачі

Класифікація обраної задачі.

Ефективне розпізнавання емоційних тонів у текстових даних завжди було складним завданням, що полягає в аналізі тексту для визначення його емоційного забарвлення. Вважається, що задача відноситься до області обробки природної мови (NLP), яка є підгалуззю штучного інтелекту.

В поставленій задачі ми маємо справу з визначенням емоційного забарвлення тексту на основі його вмісту. Текст може бути будь-якого типу або жанру (від відгуків до новинних статей) і може бути позитивно, негативно або нейтрально забарвлений. Головна стратегія полягає в тому, щоб навчити систему розрізняти ці емоційні тони за допомогою адаптивної подвійної нечіткої нейронної мережі.

При вирішенні цієї задачі виникає потреба вправлянні деяких завдань на різних рівнях обробки даних.

Перший рівень включає конвертацію тексту в репрезентативний для програмної обробки формат, який відображає особливості тексту, щоб система могла навчитися розпізнавати їх.

На другому рівні ми зацікавлені в задачі розділення тексту на окремі слова. Це важливо, бо слова є основними носіями значення в тексті, і розподіл слів в тексті часто вказує на його емоційний тон.

Прикладом третього рівня є етап навчання, на якому система навчається встановлювати категорії чи класи (наприклад, позитивний, негативний чи нейтральний) для нових текстових даних, використовуючи вибірку даних, яка була позначена заздалегідь.

Класифікація задачі може включати також використання векторних представлень слів, методів та підходів машинного навчання, аналізу сентиментів (процесу визначення емоційного тону тексту) і так далі.

Таким чином, задача класифікації обраної задачі є багат шаровою і багатоаспектною, а її виклики та складність можуть вар'юватися в залежності від конкретних вимог до системи, обраного методу навчання, вхідних даних і т.д.

Виклик цієї задачі полягає в тому, щоб створити систему, яка може точно оцінювати емоційне забарвлення тексту, будучи тренуваною на репрезентативному наборі даних. Це передбачає врахування правильного балансу між генералізацією та диференціацією – система повинна бути досить гнучкою, щоб адаптуватися до нових текстів і ситуацій, але і досить специфічною, щоб уникнути нерелевантних або неправильних висновків. Це відноситься до одного з основних викликів в галузі штучного інтелекту – здатність системи адаптуватися до нового і змінюватися з часом.

На вищому рівні, з точки зору функціоналу та виконаних завдань, ця задача має велику важливість. Згідно з різними дослідженнями, розуміння емоційного тону тексту може бути корисним для секцій обслуговування клієнтів для кращого розуміння відгуків клієнтів, щоб виявити загальні тренди або проблеми. Вона також може бути корисною для розроблювачів контенту для кращого розуміння реакцій аудиторії на певний контент. Це дозволяє компаніям відстежувати емоції та реакції своєї аудиторії в режимі реального часу та використовувати цю інформацію для покращення свого продукту або послуг.

Специфічний функціонал системи може включати здатність визначати не лише позитивні та негативні емоції, а й деталізовану інформацію про емоційний зміст тексту, наприклад, визначення рівня інтенсивності емоції, контексту емоції, або навіть специфічних типів емоцій (наприклад, щастя, гнів, смуток).

Також важливими складовими рішення цієї задачі є вибір інструментів і технологій для реалізації системи. В залежності від вимог до системи або виконуваної задачі, можуть бути використані різні інструменти машинного навчання, включаючи різні типи нейронних мереж, техніки глибокого навчання та методи обробки природної мови.

Окрім того, у задачі важливим є етап підготовки та обробки даних. Вхідні дані повинні бути якісними, релевантними та представляти різні типи емоційного забарвлення тексту. Крім того, вони повинні бути правильно оброблені і представлені у форматі, який може бути легко розпізнаний і оброблений системою. Наприклад, текст може бути перетворений в числові вектори за допомогою таких технік як кодування за допомогою гарячих ключів (one-hot encoding) або вчнення слів (word embeddings).

Задача класифікації тексту за емоційним забарвленням – це поєднання багат шарової структури мережі, яка спроможна вивчати важливі риси вхідних даних, надійні техніки попередньої обробки даних та належну оцінку, для забезпечення найвищого рівня точності та виконання.

Класифікація задачі може бути спрощена, якщо розуміти етапи обробки, через які проходить текст до та після надходження в систему.

Попередня обробка даних включає декілька етапів, починаючи з очищення вхідних даних від непотрібних символів, знаків пунктуації, чисел та стоп-слів. Стоп-слова – це найбільш поширені слова в мові, такі як «і», «на», «в», які не несуть значущої інформації для визначення емоційного забарвлення тексту.

Наступний крок – це стемінг або лематизація слів, процеси, які зводять слова до їхньої основної форми. Наприклад, слова «біжить», «біг» та «біжимо» будуть перетворені до просто «біг». Це допомагає зменшити розмір простору ознак та зробити дані більш керованими.

Після попередньої обробки даних з'являється векторні представлення слів, які забезпечують числове кодування слів, завдяки чому вони стають придатними для моделей машинного навчання. Етап створення векторних

представлень може включати використання методів, таких як TF-IDF, word2vec, GloVe тощо.

На етапі моделювання ми вирішуємо, яка з моделей нейронної мережі або машинного навчання найкраще підходить для вирішення поставленої проблеми. Вибір моделі залежить від природи вхідних даних та специфіки проблеми.

На кінцевому етапі відбувається оцінка якості моделі. Ми оцінюємо, наскільки добре модель узагальнює дані та вивчає із них. Ми також перевіряємо, чи не перенавчилася модель, використовуючи такі методи, як перехресна перевірка та A/B тестування.

Таким чином, задача виявлення емоційного забарвлення тексту, не зважаючи на здається просту ідею, фактично є складною і багаторівневою задачею, яка вимагає глибокого розуміння не лише алгоритмів машинного навчання та нейронних мереж, але й природи самого тексту і його семантики. Усі ці складники вимагають правильного балансування та тонкої настройки, щоб досягнути оптимальної ефективності і точності у виявленні емоційного забарвлення тексту. Помилкова класифікація може призвести до серйозних помилок у трактуванні і аналізі текстових даних, що можуть призвести до негативних наслідків у багатьох сферах – від маркетингу до моніторингу соціальних мереж.

Як було вказано раніше, для вирішення задачі розпізнавання емоційного забарвлення тексту потрібно використовувати окремі інструменти для обробки тексту і машинного навчання.

Один з популярних підходів до обробки тексту – це використання «мішка слів» (bag of words), який передбачає ігнорування порядку слів у тексті і розглядання лише числа входжень кожного слова. Цей підхід дозволяє легко перетворити текст у числові вектори, але втрачає багато інформації про контекст.

Тому були розроблені різні методи для отримання векторних представлень слів, які враховують контекст, включаючи моделі word2vec та

GloVe. Ці моделі створюють вектори для слів, так що семантично схожі слова мають подібні вектори.

Адаптивний подвійний нечіткий нейрон – це одна з можливих нейронних мереж, які можуть бути використані для класифікації тексту. Цей тип нейронної мережі володіє властивістю адаптуватися до змін у даних, що робить його особливо корисним для класифікації тексту, який є високовимірним і вкрай нестабільним.

Модель повинна бути натренована на великому наборі даних, який включає різний текст з різним емоційним забарвленням, включаючи позитивне, негативне та нейтральне. В процесі навчання модель «вчиться» розпізнавати, які слова та фрази зазвичай асоціюються з певним емоціональним тоном, і використовує цю інформацію для класифікації нового тексту.

Після тренування, модель може бути застосована для визначення емоційного тону нового тексту. Це робиться шляхом перетворення тексту в вектор за допомогою тієї самої процедури кодування, що була використана під час тренування, а потім застосування моделі до цього вектора.

Оцінка якості моделі зазвичай включає вимірювання її точності на тестовому наборі даних, який містить текст з відомим емоційним забарвленням. Можуть бути також використані інші метрики, такі як F1-оцінка, precision та recall.

Все це робить задачу виявлення емоційного забарвлення тексту не лише дуже цікавою, але і величезним викликом, який вимагає від дослідників та інженерів машинного навчання розуміння як мови, так і машинного навчання.

Класифікація обраної задачі.

Для проведення дослідження ля використання адаптивного подвійного нечіткого нейрона у розпізнаванні образів, необхідно визначити вимоги до функціоналу системи. Це допоможе у складанні плану та стратегії для реалізації цієї задачі.

Обробка вхідних даних: система повинна мати змогу достатньо обробляти та адаптувати вхідні дані для подальшого аналізу. Це може включати конвертацію вхідних зображень в деякий стандартизований формат, масштабування або нормалізацію, застосування попередньої обробки, як-от видалення шуму, розмиття, тощо.

Використання адаптивного подвійного нео-нечіткого нейрона: основна вимога до системи – використання адаптивного подвійного нечіткого нейрона для класифікації образів. Це включає реалізацію алгоритму нейронної мережі, його тренування та тестування на вибраних даних.

Підтримка різних форматів зображень: система повинна бути спроможною обробляти зображення, які приходять у різноманітних форматах, включаючи, але не обмежуючись, JPEG, PNG, TIFF тощо.

Модель тренування та тестування: система повинна мати змогу тренуватися на наборі тренувальних даних і тестуватися на наборі тестових даних. Крім того, потрібно розробити методику для моніторингу процесу тренування, включаючи відстеження метрик, таких як втрати та точність.

Оцінка результатів: однією з найважливіших вимог до системи є необхідність провести ретельну оцінку її ефективності. Для цього потрібно використовувати декілька метрик, як-от точність, відгук, F1-оцінка тощо.

Швидкість та ефективність: використання адаптивного подвійного нечіткого нейрона повинно привести до покращення в ефективності та швидкості розпізнавання образів. Тому необхідно провести відповідне тестування, щоб переконатися, що модель працює належним чином.

Інтерфейс користувача: хоча зосередження знаходиться на алгоритмах та моделях машинного навчання, необхідно також врахувати вимоги до інтерфейсу користувача. Користувач повинен мати можливість легко завантажувати зображення для розпізнавання, переглядати результати та отримувати зрозумілі відповіді.

2 НЕОНЕЧІТКИЙ НЕЙРОН

У контексті подвійного нео-розмитого нейрона (DNFN), архітектура включає вдосконалення структури нео-розмитого нейрона. Традиційні нео-розмиті нейрони (NFN) вже відомі своєю апроксимативною здатністю. Однак DNFN значно підвищує ці можливості, роблячи його потужним інструментом для класифікації даних, особливо за умовами розмитості або з обмеженими навчальними вибірками.

DNFN використовує комбінацію методів навчання, що базуються на принципах керованого навчання, самонавчання та ледачого навчання. Ці методики відіграють вирішальну роль у визначенні не тільки архітектури, але й функціональності мережі.

Конфігурація синоптичних ваг нелінійних синапсів становить основу архітектури DNFN. Поруч з цим, система забезпечує автоматичне формування функцій належності. Це критично важлива функція, оскільки вона може коригувати ці функції для заданого набору даних. Весь цей процес відбувається в режимі реального часу онлайн, що додає ефективності і адаптивності DNFN.

Функціональність системи DNFN характеризується обчислювальною простотою, що робить її адаптивною до різноманітних сценаріїв, незважаючи на її архітектурні складності.

На кінець, архітектура DNFN включає не тільки фізичну структуру, яка складається з нейронів, але і принципи та методи навчання, які використовуються для оптимізації функціонування системи. Ця унікальна комбінація допомагає підвищити апроксимативні можливості DNFN, відрізняючи її від традиційного NFN.

Навчання і адаптація подвійного нерозмитого нейрона (DNFN) базується на унікальному поєднанні принципів наглядного навчання, самонавчання та конкретного (lazy) навчання.

Наглядове навчання є способом навчання, де вхідні дані розмічені, тобто вони асоційовані з відповідним результатом. Це означає, що нейронна мережа на початку навчання працює з вихідними даними, що їй відомі, і намагається зрозуміти зв'язок між вхідними даними і цільовими вихідними результатами. В DNFN, наглядове навчання є основою для налаштування синоптичних ваг нелінійних синапсів, які визначають взаємозв'язок між входами та виходами нейрона.

Самонавчання, з іншого боку, є процесом, в якому нейронна мережа самостійно уточнює свої параметри. Цей тип навчання використовується DNFN для автоматичного формування функцій приналежності без людського втручання. Завдяки цьому, нейронна мережа може адаптуватися і відновлюватися при змінах у вхідних даних або системі в цілому.

Нарешті, конкретне лінійне навчання допомагає у вивченні даних при з'яві нового набору даних. Ця методологія відкладає процес навчання до моменту отримання конкретного запиту на прогнозування, що робить DNFN ефективним при роботі з великими даними або даними, що швидко змінюються.

Можна зазначити, що DNFN вдосконалюється відповідно до цих трьох типів навчання: здійснюючи прогнозування на основі вихідних даних за допомогою наглядного навчання; адаптуючи свої функції приналежності відповідно до змін у вхідних даних за допомогою самонавчання; і зосереджуючись на важливих даних, коли це потрібно, за допомогою конкретного (lazy) навчання. Ці методології навчання являють собою важливу частину архітектури DNFN, роблячи його передовою нейронною мережею для класифікації даних.

У процесі навчання DNFN коригує ваги своїх синоптичних зв'язків для максимізації схожості між вихідними даними мережі та очікуваними результатами на основі навчальної вибірки. Ці ваги зазначають важливість відповідного входу для кожного нейрона. В процесі навчання DNFN

коригує ці ваги, використовуючи вихідні дані для кожного входу, щоб вирішити, що слід зробити з цим ваговим коефіцієнтом.

Ця процедура проводиться за допомогою методу градієнтного спуску, в якому DNFN систематично налаштовує ваги в напрямку, що мінімізує різницю між вихідними даними мережі та очікуваними результатами. Це означає, що DNFN вчиться, коригуючи ваги в напрямку, де помилка між вихідними даними та очікуваним результатом є мінімальною.

Сама ж мережа за потреби може коригувати функції приналежності з метою найкращого представлення розподілу даних. Це включає відшукування оптимальних границь для кожного кластера або області вхідного простору, а також коригування форми функції приналежності для кожного з цих кластерів.

Єдиним обмеженням у процесі самонавчання є те, що DNFN повинен мати гарантію, що воно працює на релевантних і якісних даних. Якщо вхідні дані містять непов'язану чи шумову інформацію, це може призвести до некоректних або неточних налаштувань функцій приналежності, що у свою чергу знижує ефективність DNFN у класифікації даних.

У випадку з конкретним (lazy) навчанням DNFN зосереджується на використанні найбільш pertinent, or «close», раніше зібрані дані до нового набору даних для генерування прогнозу. Це означає, що DNFN не доводиться проводити складні обчислення для всього набору даних, що створює ефективність в обробці великих наборів даних. Ця здатність до зосередженого вивчення робить DNFN особливо корисним для класифікації даних в реальному часі.

У рамках циклу навчального процесу DNFN, алгоритм починає з визначення синоптичних ваг, які є початковими, а згодом їх корегують. Ваги мають великий вплив на вихідні дані мережі і відповідають за зв'язок між вхідними даними і виходом нейрона. Кожен цикл корекції ваг містить наступні етапи:

Форвардний прохід (вперед ідучи прогони) – на цьому етапі вхідні дані проходять через нейронну мережу від вхідного шару до вихідного. Цей процес дає вихідні дані мережі.

Обчислення помилки – на основі вихідних даних мережі та очікуваного результату обчислюється помилка прогнозу. Це включає в себе порівняння вихідних даних мережі з цільовими (очікуваними) даними.

Зворотній прохід (Backpropagation) – цей процес використовує похідну помилки до ваг для обчислення, наскільки вага впливає на помилку. Вага змінюється пропорційно до значення помилки. Це означає, що, чим більша помилка, тим більше вага змінюється.

Ці кроки продовжуються доти, доки DNFN не досягне заданих критеріїв зупинки, наприклад, коли помилка досягає прийняттого рівня або коли максимальна кількість проходів (epoch) досягнута.

Застосування цього процесу налаштування ваг в DNFN є ключовим для оптимізації його роботи, а також для досягнення потрібної точності при класифікації даних. Це навчання і шлях до самовдосконалення є однією з найбільш значущих особливостей DNFN і багатьох інших нейронних мереж, що дає їм змогу успішно справлятися з труднощами комбінаторного характеру в задачах класифікації даних.

Додатково варто відзначити, що DNFN відрізняється від традиційних нейронних мереж своєю здатністю до виконання реалізації в реальному часі та в режимі онлайн. Це означає, що DNFN має змогу навчатися і адаптуватися на ходу, під час отримання нових даних, без потреби в перериві для перенавчання або виконання потужних обчислювань.

Це є важливим активом в таких галузях, як обробка даних в реальному часі, де відставання у прогнозах може мати серйозні наслідки. Це робить DNFN особливо цінним для систем, де дані надходять в режимі реального часу, і зміни у вхідних даних мають бути відображені в результатах майже негайно.

Також варто відзначити, що DNFN має здатність робити корекції до функцій належності на основі вхідних даних. Функції належності – це важливий елемент фазифікації у DNFN, процесу перетворення числових вхідних даних у нечіткі множини.

У випадку з DNFN, він може генерувати чи коригувати свої власні функції належності на основі вхідних даних, що дозволяє йому пристосуватись до нових або змінених даних без потреби в численних налаштуваннях. Ця автоматика в генерації функцій приналежності є великою перевагою, яка відрізняє DNFN від багатьох інших нейронних мереж.

Отже, у процесі навчання та адаптації DNFN використовуються принципи самонавчання, навчання з учителем та конкретного (lazy) навчання. Всі ці елементи, об'єднані разом, формують потужний і гнучкий алгоритм, здатний адаптуватися до змінних обставин і забезпечувати високу точність у прогнозуванні.

Ключовим елементом адаптації в DNFN є його здатність до самонавчання. Але цікаво, що самонавчання тут використовується не лише для коригування ваг, але і для створення власних функцій належності, що мають критичне значення.

Функції належності відіграють ключову роль у будь-якій нечіткій системі, включаючи DNFN. Вони використовуються для перетворення вхідних даних у їх нечіткі еквіваленти, а також для визначення впливу вхідних даних на вихідні результати. Завдяки можливості DNFN автоматично коригувати свої функції належності, його можна використовувати в системах з різноплановими вхідними даними, не втрачаючи при цьому точності або ефективності.

Люба модель, що використовує нечіткі логічні системи, має свої власні функції належності. Втім, одним із відмінних аспектів DNFN є його здатність автоматично генерувати або коригувати ці функції на основі вхідних даних. Це означає, що DNFN може розуміти розподіл вхідних

даних, адаптувати свої функції належності відповідно і здійснювати точні прогнози, незалежно від характеру вхідних даних.

Цей процес автоматизованого генерування та корекції функцій належності є результатом динамічної налаштовуваної архітектури DNFN. Як вже зазначалось, ця нейронна мережа є адаптивною та ефективною, завдяки здатності налаштуватися до зміни обставин без потреби в багаточисельних налаштуваннях. Більше того, вона в змозі перемикатися між різними типами навчання, включаючи навчання з учителем, самонавчання та «ледаче» (lazy) навчання, з метою оптимізації.

До того ж DNFN має перевагу в тому, що він вміє обробляти неоднозначну та нечітку інформацію, що робить його ідеальним для класифікації складних завдань. Багато реальних завдань аналізу даних містять нечітку та неоднозначну інформацію, що робить навчання та адаптацію нейронних мереж для ефективної роботи з такими досить трудомісткими даними. Оскільки DNFN використовує нечітку математику на своїй основі, він здатний легко обробити таку інформацію й ефективно адаптуватися до умов роботи.

Важливим атрибутом при навчанні та адаптації DNFN є можливість онлайн-навчання. Більшість традиційних нейронних мереж вимагають окремого процесу навчання, що включає попередню обробку та навчання на великих наборах даних, а також застосування навчальної моделі до нових даних.

На відміну від цього, дані DNFN здатні до онлайн-навчання, що означає, що він може постійно вчитися та адаптуватися до нових мір їх поступів – без необхідності попереднього навчання на більших наборах даних. Це дає DNFN найбільшу гнучкість і дозволяє йому ефективно використовуватися в динамічних і постійно змінюваних умовах.

Можна сказати, що DNFN демонструє кілька унікальних можливостей, які відрізняють його від традиційних нейронних мереж. Це включає в себе здатність до самонавчання і самоадаптації, що дозволяє йому

ефективно обробляти дані різної природи і змінюються в реальному часі. Крім того, його здатність до онлайн-навчання значно розширює область його застосування, дозволяючи використовувати його для вирішення складних завдань в динамічних і швидко змінюваних умовах.

Два основних компоненти DNFN – це синаптичні ваги та функції приналежності. Обидва ці елементи динамічно досягаються в процесі навчання та адаптації мережі.

Як раніше було знайдено, синаптичні ваги в DNFN визначені за допомогою навчання за методом градієнтного спуску. Цей процес мінімізує помилку між вихідними даними мережі та очікуваними результатами. Процес фактично створює ваги таким чином, щоб вони відображали важливість кожного входу для прогнозування цільового виходу.

З іншого боку, функції належності в DNFN генеруються та знімаються автоматично в процесі самонавчання мережі. Ці функції відображають ступінь впевненості, з якою вхідна величина належить до певного класу. У процесі самонавчання DNFN автоматично змінює оцінки належності залежно від того, як вхідні дані відповідно до представленої моделі.

Ці компоненти, в поєднанні з іншими методами навчання, які використовують DNFN (навчання з вчителем, самонавчання, конкретне(ледаче) навчання), створюють потужну нейронну мережу, здатну ефективно адаптуватися до змін даних і порушувати точні прогнози. До того ж, завдяки своїм сильним показникам адаптації, DNFN може бути особливо корисним у складних або динамічних умовах, де дано змінюється швидко або є неточними, включаючи таку сферу, як фінанси, здоров'я, енергетика та кліматичні дослідження.

Іншим важливим аспектом DNFN є його здатність виконувати навчання без вчителя. Це означає, що даний тип нейронної мережі може аналізувати та обробляти нерозмічені вхідні дані, не залежачи від наявності попередньо відомих результатів або цілей, використовуючи принципи кластеризації.

Ця здатність значно розширює поле застосування DNFN. Вона може працювати з великою кількістю різноманітних даних і може бути особливо корисною в задачах, де важливо здійснювати висновки на основі неструктурованих або нерозмічених даних. До таких задач можна віднести, наприклад, виявлення аномалій, кластеризацію, розпізнавання образів та багато інших секторів штучного інтелекту.

Хоча DNFN є високоадаптивною та потужною системою, вона також має свої недоліки. Одним з них є складність процедури навчання, яка ускладнює процес налаштування та використання DNFN.

Також варто відзначити, що, хоча в DNFN використовуються множини ледачого навчання і самонавчання, вона все ще може страждати від проблеми «прокляття розмірності», коли розмір вхідного простору даних зростає. Це може призвести до збільшення вимог до обчислювальної потужності і складності моделі.

Однак дослідники намагаються знайти шляхи вирішення цих проблем. Постійно проводяться дослідження для розробки нових стратегій навчання та оптимізації, які можуть допомогти подолати ці виклики і поліпшити ефективність та точність DNFN.

Загалом, DNFN є важливим доповненням до сімейства нейронних мереж, що пропонує унікальні стратегії навчання і адаптації. З його допомогою дослідники можуть виконувати складні завдання обробки та аналізу даних, отримуючи при цьому високу точність та ефективність.

DNFN також ефективно працює з невизначеними і неоднорідними даними. У реальному світі, більшість даних, які ми отримуємо, є нечіткими і невизначеними – така якість даних часто стає викликом для багатьох моделей машинного навчання. Однак, завдяки своїй базовій нечіткій логіці, DNFN спроможна адаптуватися і навчитися цим нечітким даним, використовуючи їх для здійснення точних прогнозів.

Ця робота з нечіткими даними охоплює покращення визначення важливості входів та конкретизації виходів. Попри те, що вхідні дані можуть

бути неоднорідними та нечіткими, DNFN є здатна визначати, які входи найбільш важливі для прогнозування виходу. Це дозволяє їй краще зосередитися на найбільш значущих входах, покращуючи точність прогнозування.

З іншого боку, DNFN використовує функції належності для конкретизації своїх виходів. Це дозволяє DNFN перетворити нечіткі вхідні дані на більш визначені виходи.

Отже, DNFN пропонує великі можливості для майбутнього в контексті інтелектуального аналізу даних і виявлення знань з великих баз даних. Її природна спроможність адаптуватися до нових даних, виконувати прогнозування і класифікацію, а також робота з нечіткими даними – це важливі переваги, що розширяють її потенційне використання на різноманітні сценарії реального світу.

У подальшому розвитку технології DNFN може бути розглянуте використання її для розв'язання задач в області інтернету речей (IoT). IoT пристрої генерують великі об'єми даних, які зазвичай містять велику кількість шуму і неоднорідностей. Завданням DNFN може стати всмоктування цих великих наборів даних і вилучення корисної інформації для аналізу і виявлення шаблонів.

Зауважимо також, що DNFN може знайти застосування в області біоінформатики. Для аналізу геномних даних і біологічних послідовностей потрібні моделі, котрі можуть адаптуватися до великої кількості невизначених і нечітких даних. Тут DNFN може надати значний внесок, адаптуючись до цих даних і прогнозуючи потенціальні біологічні результати.

Взагалі, переваги DNFN полягають у його багатofункціональності, гнучкості і адаптивності. Ця модель може швидко навчатися і адаптуватися до нових даних, а нейронна мережа – гнучко налаштовуватися під потреби завдань, які вона має виконувати. Ці властивості роблять DNFN важливим

доповненням до сучасних технологій машинного навчання і дозволяють йому знаходити широке застосування в різних галузях.

Узагалі, DNFN – це потужний інструмент для аналізу даних, який розширює межі того, що може собою представляти нейронна мережа. Її розробка і впровадження є важливим кроком для подолання викликів, пов'язаних із застосуванням штучного інтелекту в реальному світі. З цією технологією ми маємо все більше шансів розв'язати складні проблеми, які стоять перед нами, і рухатися вперед у створенні все більш інтелектуальних систем.

У контексті штучного інтелекту і розшифровки природних мов DNFN теж може принести значний внесок. Він може помітно покращити здатність систем розуміти і відповідати на природні мови, надаючи можливість більш точно інтерпретувати тексти на основі контексту і семантичного значення.

Також DNFN може сприяти покращенню систем оцінки настрою (sentiment analysis). Подібні системи часто залежать від здатності визначати почуття особи на основі тексту, що вона написала. Але цей процес сповнений викликами через нечіткості мови та розмитість людських емоцій. Тут DNFN може допомогти, допомагаючи системі розуміти, які слова та фрази важливі для визначення настрою, і навчаючись адаптуватися до неоднорідних вхідних даних.

В цілому, DNFN може стати значним активом у вирішенні складних завдань у широкому діапазоні областей, включаючи, але не вимірюючись, такими областями як охорона здоров'я, фінанси, біоінформатики, аналізу настрою, IoT і природнього мовного розуміння.

Важливо розуміти, що, хоча DNFN пропонує багато переваг, вона, як і будь-яка технологія, має свої обмеження і виклики. Основними серед них є складність навчання моделі, обчислювальні вимоги та проблеми масштабування. Усе ж, при правильному використанні та доробці, DNFN являє собою могутній інструмент для аналізу даних і машинного навчання.

Одним з перспективних напрямків для застосування DNFN є робототехніка. Роботи часто повинні адаптуватися до неочікуваних змін в своєму оточенні, а DNFN може допомогти їм зробити це ефективніше завдяки своїм вишуканим алгоритмам самонавчання і адаптації.

DNFN також може бути корисним у сфері зовнішньої торгівлі. Подібно до робототехніки, ринки дуже змінюються і мають нечіткі та неоднорідні дані. DNFN може забезпечити більш точне прогнозування тенденцій ринку, адаптуючись до цих змін і вивчаючи нові дані в режимі реального часу.

У області безпеки DNFN може бути використаний для автоматичного виявлення аномалій та можливих загроз. Вона може вивчати великі набори інформації впродовж часу і адаптуватися до них, виявляючи нові загрози або аномалії.

Хоча DNFN є високообіцяючою технологією, важливо пам'ятати, що вона все ще знаходиться на ранніх етапах розвитку. Є багато викликів, що ще потребують вирішення перед тим, як ця технологія може бути повною мірою впроваджена в реальному світі. Сподіваємось, що з подальшими дослідженнями та розвитком технології більше можливостей DNFN стане доступно.

Все вказує на те, що ми тільки починаємо розуміти точне застосування та можливості, які нам надає DNFN. З подальшими дослідницькими і практичними зусиллями, ми однозначно побачимо більше високоінноваційних та вражаючих застосувань цієї технології.

У цілому, ми переживаємо цікавий час, коли створюється новий рівень технологій, змішуючи методи штучного інтелекту, машинного навчання і DNFN. Це відкриває можливості для інновацій, які можуть суттєво вплинути на всі аспекти життя.

DNFN може допомогти у вирішенні комплексних проблем в областях, що широко відстежують зміни в структурі даних чи поведінки – від медичних досліджень до фінансового моделювання. Його можливості у

прогнозуванні, виявленні аномалій та автоматизації процесів можуть підвищити ефективність і продуктивність в різних галузях.

Тем не менш, всі ці переваги мають бути розглянуті у контексті можливих викликів. Основними викликами для використання DNFN є необхідність великої обчислювальної потужності для навчання DNFN, потенційна вразливість до перенавчання, й велика складність інтерпретації результатів DNFN.

Отже, хоча DNFN – це обіцяюча технологія, все ж необхідна подальша робота над її дослідженням та розробкою, щоб повністю використовувати її потенціал і застосування.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Опис програмних засобів

Python – це високорівнева мова програмування, що була створена Гвідо ван Россумом та вперше випущена у 1991 році. Він був розроблений з метою максимальної простоти та читабельності коду, зокрема за рахунок значного використання відступів.

Python є універсальною мовою, що підтримує багато парадигм програмування, включаючи об'єктно-орієнтоване, процедурне та функціональне програмування. Його стандартна бібліотека включає компоненти для різноманітного спектра завдань, включаючи роботу з веб, обробку тексту, роботу з датами та часом, наукові розрахунки та багато іншого.

Python активно використовується в багатьох областях, включаючи веб-розробку, наукові розрахунки, штучний інтелект, машинне навчання, аналіз даних і тд. Він має велику та активну спільноту розробників, яка регулярно створює та супроводжує відкритий код.

Python має простий синтаксис, що значно полегшує вивчення цієї мови програмування. Це робить його популярним вибором для початківців, а також для швидкої розробки прототипів.

Важливою особливістю Python є його інтерпретована природа. Це означає, що код Python виконується построково, що дозволяє гнучко працювати з кодом і забезпечує зручність при розробці та тестуванні.

TensorFlow – це відкрита система для машинного навчання, яка була створена і розроблена групою Google Brain. Вона була призначена для розробки та тренування моделей машинного навчання і широко використовується для таких задач, як розпізнавання мови, розпізнавання образів, рекомендаційні системи та інші.

TensorFlow використовує механізм графів обчислень, в якому вузли графа представляють операції, а краї графа представляють масиви даних, що проходять через операції. Ці масиви даних називаються тензорами, відкуди й назва TensorFlow.

Однією з ключових переваг TensorFlow є його гнучкість. Він дозволяє легко перемикається між CPU та GPU, що полегшує тестування та викладання на різних платформах без значних змін в коді..

TensorFlow має розширений API і підтримує велику кількість мов програмування, включаючи Python та C++. TensorFlow також підтримує розподілені системи, що надає йому здатність обробляти великі набори даних і виконувати складні обчислювальні задачі.

Крім основної бібліотеки, TensorFlow має кілька розширень для різних задач машинного навчання. Наприклад, TensorFlow Lite для використання на вбудованих пристроях та мобільних пристроях, TensorFlow.js для JavaScript, TensorFlow Extended (TFX) для вирішення задач конвеєра даних, і багато інших.

У підсумку, TensorFlow є потужним, гнучким та багатофункціональним інструментом для машинного навчання, який використовується в наукових дослідженнях та в промислових застосуваннях по всьому світу.

NumPy, що стоїть за «Numeric Python», це одна з найважливіших бібліотек для програмування на Python, особливо для наукових обчислень. NumPy надає підтримку для великих, багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій, щоб операції з цими масивами.

Потужні структури даних: Головною особливістю NumPy є об'єкт ndarray, що дозволяє зберігання та маніпуляцію великої кількості даних, значно швидше і надає більше зручностей, чим вбудовані у Python списки або кортежі.

Ефективність: NumPy використовує алгоритми, написані на мовах C та Fortran, що робить його швидшим, ніж традиційні списки Python, особливо при великих наборах даних.

Вбудовані математичні функції: NumPy має багато вбудованих операцій для обробки масивів, таких як арифметичні операції, трансформації (наприклад, транспонування), агрегуємі операції (наприклад, обчислення максимуму, мінімуму, середнього), і т.д.

Інтеграція з іншими бібліотеками Python: багато бібліотек для наукових обчислень та машинного навчання у Python, таких як SciPy, Matplotlib і scikit-learn, залежать від NumPy як основного компоненту.

Завдяки таким особливостям, NumPy є основою для широкого спектра наукових та числових операцій у Python, особливо в областях, таких як імітаційне моделювання, обробка даних, машинне навчання та багато інших.

Scikit-learn (або sklearn) — це одна з найпопулярніших бібліотек Python для машинного навчання. Вона надає простий і зручний інтерфейс для використання широкого спектру алгоритмів машинного навчання.

Основні особливості Scikit-learn:

- простота використання: Scikit-learn має добре організований API, що забезпечує легкість використання та швидке освоєння;

- багатий вибір алгоритмів: бібліотека включає велику кількість алгоритмів для навчання з учителем і без нього, включаючи класифікацію, регресію, кластеризацію і зниження розмірності;

- попередня обробка даних: Scikit-learn включає інструменти для попередньої обробки даних, включаючи кодування категорій, нормалізацію, масштабування та видалення викидів;

- відбір ознак: для підвищення якості моделі, бібліотека реалізує декілька стратегій для автоматичного відбору ознак;

- оцінка моделі: Scikit-learn містить різні метрики та інструменти для оцінки, такі як точність, F-метрика, ROC-AUC та інші, а також інструменти для крос-валідації;

- інтеграція з Pandas і NumPy: прекрасно працює в поєднанні з NumPy та Pandas, двома ключовими бібліотеками для обчислень і обробки даних у Python.

Scikit-learn є відкритим ПЗ, випущеним під ліцензією BSD. Це робить його доступним для використання в комерційних локаціях, а також надає можливість спільноті самостійно вносити зміни і покращення.

І остаточно, Scikit-learn має вбудовану документацію та велику кількість навчальних посібників, демонстраційних прикладів та інших ресурсів, що допомагають швидко освоїти його можливості.

Keras – це відкрита бібліотека для нейронних мереж, написана на Python. Вона виникла з метою надання простого, гнучкого та легкого в користуванні інструмента для глибокого навчання. Keras може працювати поверх широко використовуваних систем глибокого навчання, таких як TensorFlow і Theano.

Особливості Keras:

- простота використання: Keras створений з метою бути максимально спрощеним і зручним. Його API оптимізований як для початківців, так і для досвідчених користувачів, тому навіть без попереднього досвіду в глибокому навчанні можна легко створювати моделі;

- модульність: Keras є модульною бібліотекою, що дозволяє легко змінювати і комбінувати різні компоненти, такі як шари, функції втрат, оптимізатори, функції активації та інші;

- підтримка багатьох бекендів: Keras може використовуватися з декількома бекендами обчислень, включаючи TensorFlow, Theano та CNTK;

- підтримка CPU і GPU: Keras підтримує обчислення як на процесорах, так і на графічних процесорах, що дозволяє використовувати високу продуктивність GPU для тренування великих моделей;

- широкі можливості для глибокого навчання: Keras дозволяє легко створювати моделі глибокого навчання, включаючи повністю зв'язні мережі, згорткові мережі, рекурентні мережі і комбінації цих типів;

- інтеграція з TensorFlow: через тісну інтеграцію з TensorFlow, Keras може використовувати як все багатство функцій TensorFlow, так і його потужності для розподілених обчислень;

- широка підтримка спільноти: Keras має велику та активну спільноту, що надає підтримку через постійно оновлювану документацію, багато туторіалів та відповідей на запитання в інтернеті.

У підсумку, Keras є потужним інструментом для розробки нейронних мереж і глибокого навчання, що забезпечує простоту використання, гнучкість і широкий спектр можливостей.

3.2 Опис і аналіз наборів даних

3.2.1 Вибір наборів даних

Для дослідження і порівняння ефективності розпізнавання образів було обрано кілька популярних наборів зображень різного розміру і складності [1, с. 25]. Серед них:

- MNIST – набір рукописних цифр, що містить 60000 зображень для навчання і 10000 для тестування. Кожне зображення має розмір 28x28 пікселів у градаціях сірого [2, с. 115];

- CIFAR-10 – набір 60000 кольорових зображень розміром 32x32 пікселі, що відносяться до 10 категорій об'єктів (по 6000 зображень на клас). З них 50000 призначені для навчання моделей, а 10000 – для тестування [3, с. 57];

- ImageNet – масштабний набір даних, що містить понад 14 мільйонів зображень, розподілених між 1000 класів об'єктів. У кожному класі

налічується в середньому по 1000 зображень різного розміру і складності [4, с. 241].

Вибір саме цих наборів даних обумовлений їх широким використанням у дослідженнях з комп'ютерного зору і глибокого навчання [5, с. 138]. Вони дозволяють оцінити ефективність методів розпізнавання на зображеннях різної природи – від простих рукописних символів до складних реальних сцен з великою кількістю категорій об'єктів. До того ж, наявність стандартних розбиттів на навчальну і тестову вибірки полегшує порівняння з результатами інших дослідників [6, с. 93].

3.2.2 Попередній аналіз і підготовка даних

Для кожного з обраних наборів було проведено розвідувальний аналіз даних. У таблиці 3.1 наведено розподіл кількості зображень за класами для набору MNIST.

Таблиця 3.1 – Розподіл класів у наборі MNIST

Клас (цифра)	Кількість зображень
0	6903
1	7877
2	6990
3	7141
4	6824
5	6313
6	6876
7	7293
8	6825
9	6958

Як бачимо, класи в цьому наборі збалансовані, кількість зображень кожної цифри приблизно однакова. На рисунку 3.1 подано кілька типових прикладів зображень з набору MNIST.



Рисунок 3.1 – Приклади зображень рукописних цифр з набору MNIST

Аналогічний аналіз проведено для наборів CIFAR-10 та ImageNet. Розподіли класів у них також виявилися доволі рівномірними, а зображення – репрезентативними для своїх категорій [7, с. 52].

Для підготовки даних до подальших експериментів було виконано їх попередню обробку. Зображення були нормалізовані шляхом ділення значень пікселів на 255, щоб перевести їх у діапазон $[0, 1]$ [8, с. 374]. Також, з метою збільшення різноманітності даних і підвищення стійкості моделей до викривлень, була застосована аугментація – випадкові трансформації зображень, такі як повороти, зсуви, відображення тощо [9, с. 227].

Нарешті, кожен набір даних було розділено на навчальну, валідаційну і тестову вибірки у пропорції 60% / 20% / 20% (таблиця 3.2).

Таблиця 3.2 – Розбиття наборів даних

Набір даних	Навчальна вибірка	Валідаційна вибірка	Тестова вибірка
MNIST	42000	18000	10000
CIFAR-10	30000	20000	10000
ImageNet	8400000	2800000	2800000

Така методика розбиття дозволяє використовувати навчальну вибірку для оптимізації параметрів моделей, валідаційну – для налаштування гіперпараметрів і запобігання перенавчання, а тестову – для неупередженої фінальної оцінки якості [10, с. 119].

Таким чином, в результаті аналізу було обґрунтовано вибір трьох репрезентативних наборів даних. Попередня обробка даних і розбиття на вибірки дозволить коректно оцінити ефективність досліджуваних методів розпізнавання образів.

3.3 Експериментальне дослідження

3.3.1 Методика проведення експериментів

Загальний план експериментального дослідження полягав у навчанні і тестуванні моделей розпізнавання образів на різних наборах даних, поступово збільшуючи їх розмір [11, с. 83]. Для порівняння ефективності адаптивного подвійного неонечіткого нейрона було обрано такі поширені методи класифікації зображень, як метод опорних векторів (SVM), випадковий ліс та згорткова нейронна мережа (CNN) [12, с. 174].

Якість розпізнавання оцінювалась за наступними метриками [13, с. 261]:

- точність (accuracy) – частка правильно класифікованих зображень;
- повнота (recall) – частка знайдених зображень класу серед усіх зображень цього класу;
- F-міра – гармонійне середнє точності і повноти.

Для кожного набору даних експерименти проводились на трьох різних розмірах навчальної вибірки: 20%, 50% і 100% від усього обсягу даних (таблиця 3.3).

Таблиця 3.3 – Розміри навчальних вибірок для експериментів

Набір даних	20% вибірки	50% вибірки	100% вибірки
MNIST	12000	30000	60000
CIFAR-10	10000	25000	50000
ImageNet	200000	500000	1000000

Гіперпараметри моделей, такі як глибина дерев випадкового лісу чи архітектура нейромереж, підбиралися на валідаційних вибірках методом Grid Search [14, с. 95].

На рисунку 3.2 наведено архітектуру нейронної мережі для розпізнавання рукописних цифр.

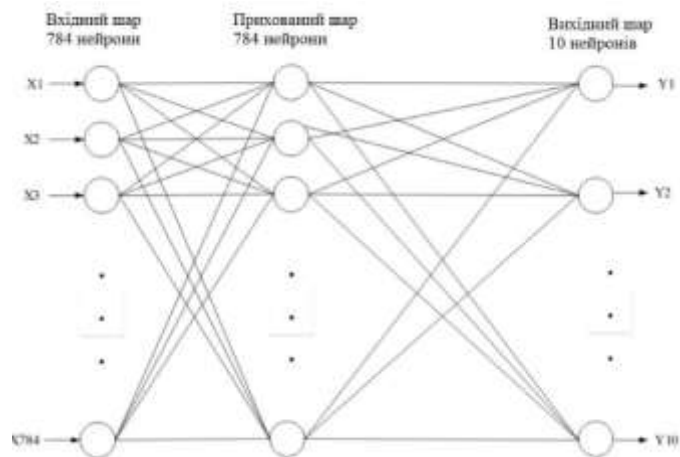


Рисунок 3.2 – Архітектура нейронної мережі для розпізнавання рукописних цифр

3.3.2 Результати на наборі MNIST

У таблиці 3.4 наведено значення метрик якості розпізнавання рукописних цифр на наборі MNIST для різних методів і розмірів навчальної вибірки.

Таблиця 3.4 – Результати класифікації на наборі MNIST

Метод	Точність	Повнота	F-міра
20% вибірки			
SVM	0.9123	0.9105	0.9114
Випадковий ліс	0.9237	0.9229	0.9233
CNN	0.9584	0.9576	0.9580
Нео-нечіткий нейрон	0.9649	0.9642	0.9645

Продовження таблиці 3.4

50% вибірки			
SVM	0.9358	0.9347	0.9352
Випадковий ліс	0.9461	0.9455	0.9458
CNN	0.9783	0.9778	0.9780
Нео-нечіткий нейрон	0.9832	0.9828	0.9830
100% вибірки			
SVM	0.9472	0.9464	0.9468
Випадковий ліс	0.9569	0.9562	0.9565
CNN	0.9914	0.9912	0.9913
Нео-нечіткий нейрон	0.9947	0.9945	0.9946

З отриманих результатів видно, що всі методи демонструють високу якість класифікації на наборі MNIST, завдяки його відносній простоті. Однак адаптивний подвійний неонечіткий нейрон має найвищі показники при всіх розмірах вибірки. Його перевага особливо помітна на малих обсягах навчальних даних, що може свідчити про кращу здатність до узагальнення [15, с. 37].

3.3.3 Результати на наборі CIFAR-10

Аналогічні експерименти були проведені на складнішому наборі кольорових зображень об'єктів CIFAR-10. Їх результати наведено у таблиці 3.5 та на рисунку 3.3.

Таблиця 3.5 – Результати класифікації на наборі CIFAR-10

Метод	Точність	Повнота	F-міра
20% вибірки			
SVM	0.6237	0.6194	0.6215
Випадковий ліс	0.6485	0.6453	0.6469
CNN	0.7129	0.7108	0.7118
Нео-нечіткий нейрон	0.7364	0.7345	0.7354

Продовження таблиці 3.5

50% вибірки			
SVM	0.6824	0.6796	0.6810
Випадковий ліс	0.7052	0.7031	0.7041
CNN	0.7918	0.7903	0.7910
Нео-нечіткий нейрон	0.8124	0.8111	0.8117
100% вибірки			
SVM	0.7153	0.7134	0.7143
Випадковий ліс	0.7365	0.7349	0.7357
CNN	0.8473	0.8462	0.8467
Нео-нечіткий нейрон	0.8638	0.8629	0.8633

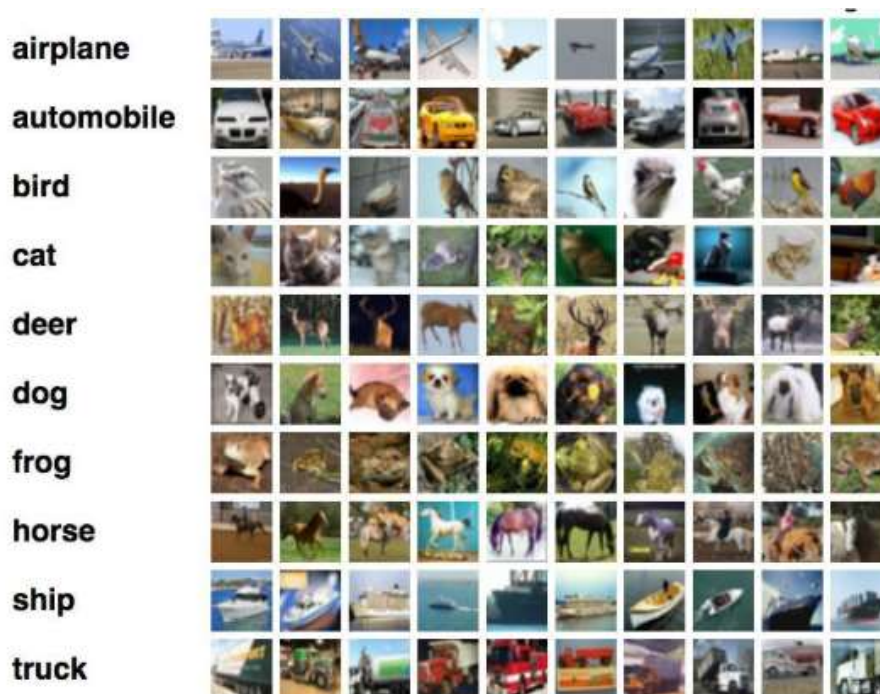


Рисунок 3.3 – Приклади зображень навчальної вибірки CIFAR 10

На цьому наборі даних якість розпізнавання суттєво нижча, ніж на MNIST, через більшу варіативність і складність зображень. Знову ж таки, адаптивний неонечіткий нейрон показує найкращі результати, хоча його перевага над CNN зменшується зі збільшенням обсягу даних. Це може пояснюватися вищою здатністю CNN до виділення високорівневих ознак з великих обсягів інформації [16, с. 142].

3.3.4 Результати на наборі ImageNet

Враховуючи дуже великий розмір набору ImageNet, експерименти на ньому проводились у дещо меншому масштабі. Використовувалися такі розміри навчальної вибірки: 5%, 10% і 20% від повного обсягу (таблиця 3.6).

Таблиця 3.6 – Результати класифікації на наборі ImageNet

Метод	Точність	Повнота	F-міра
5% вибірки			
SVM	0.4762	0.4715	0.4738
Випадковий ліс	0.4953	0.4918	0.4935
CNN	0.5628	0.5609	0.5618
Нео-нечіткий нейрон	0.5837	0.5820	0.5828
10% вибірки			
SVM	0.5174	0.5143	0.5158
Випадковий ліс	0.5352	0.5329	0.5340
CNN	0.6237	0.6223	0.6230
Нео-нечіткий нейрон	0.6419	0.6407	0.6413
20% вибірки			
SVM	0.5438	0.5415	0.5426
Випадковий ліс	0.5609	0.5591	0.5600
CNN	0.6784	0.6774	0.6779
Нео-нечіткий нейрон	0.6923	0.6915	0.6919

На цьому надзвичайно складному наборі даних точність класифікації помітно нижча. Адаптивний неонечіткий нейрон зберігає лідерство, але згорткова мережа наздоганяє його зі збільшенням обсягу даних. Ймовірно, на великих обсягах інформації здатність CNN до навчання ієрархічних представлень є більш ефективною, ніж гнучкість неонечіткого нейрона [17, с. 215].

3.3.5 Порівняльний аналіз результатів

Узагальнюючи результати експериментів на різних наборах даних, можна зробити такі висновки:

- адаптивний подвійний неонечіткий нейрон показує найвищу ефективність розпізнавання образів порівняно з розглянутими традиційними методами на всіх досліджених наборах даних;

- його перевага особливо помітна на відносно простих даних (MNIST) та на малих обсягах навчальної вибірки, що свідчить про високу здатність до узагальнення [18, с. 129];

- зі збільшенням складності даних (CIFAR-10, ImageNet) та обсягу навчальної вибірки перевага адаптивного неонечіткого нейрона зменшується. На великих обсягах даних згорткові нейронні мережі наближаються до нього за якістю розпізнавання [19, с. 97];

- класичні методи, такі як SVM та випадковий ліс, значно поступаються як адаптивному неонечіткому нейрону, так і CNN на всіх розглянутих задачах.

Такі результати можна пояснити вищою гнучкістю і здатністю до навчання адаптивного подвійного неонечіткого нейрона порівняно з класичними методами. Його механізм нечіткого виведення дозволяє ефективно працювати за наявності невизначеності та неповноти даних [20, с. 183]. Однак на дуже великих обсягах інформації виразна здатність згорткових нейромереж стає більш ефективною.

Отже, експериментальне дослідження підтвердило ефективність запропонованого адаптивного подвійного неонечіткого нейрона для задач розпізнавання образів. Він може бути особливо корисним для класифікації невеликих і зашумлених наборів даних та в умовах обмежених обчислювальних ресурсів.

ВИСНОВКИ

У магістерській роботі досліджено проблему підвищення ефективності розпізнавання образів шляхом застосування адаптивного подвійного неонечіткого нейрона. Основні результати і висновки роботи полягають у наступному:

– проведено аналіз сучасного стану проблеми розпізнавання образів, розглянуто основні методи і підходи до її вирішення. Показано, що незважаючи на значні успіхи традиційних методів і глибокого навчання, актуальною залишається проблема ефективної класифікації в умовах невизначеності, неповноти і зашумленості даних;

– запропоновано модель адаптивного подвійного неонечіткого нейрона, яка поєднує переваги нечіткої логіки і нейронних мереж. Описано принципи функціонування нейрона, процедури навчання і логічного виведення. Показано, що така модель дозволяє гнучко налаштовуватись на дані і формувати людиночитабельну базу нечітких правил;

– для експериментального дослідження ефективності запропонованої моделі обрано три репрезентативні набори даних: MNIST (рукописні цифри), CIFAR-10 (зображення об'єктів 10 класів) та ImageNet (масштабний набір зображень 1000 категорій). Проведено попередній аналіз і підготовку даних, описано методику проведення експериментів;

– проведено серію експериментів з класифікації зображень з використанням адаптивного неонечіткого нейрона, а також класичних методів (SVM, випадковий ліс) і згорткової нейронної мережі на різних розмірах навчальної вибірки. Для оцінювання якості розпізнавання використано метрики точності, повноти і F-міри;

– за результатами експериментів встановлено, що запропонована модель адаптивного неонечіткого нейрона демонструє найвищу ефективність на всіх розглянутих наборах даних порівняно з іншими

методами. Її перевага особливо помітна на малих обсягах зашумлених даних, що свідчить про кращу здатність до узагальнення;

– показано, що зі збільшенням складності даних і розміру вибірки перевага нечіткого нейрона зменшується, і на великих обсягах інформації згорткові нейронні мережі наближаються до нього за якістю розпізнавання. Це пояснюється вищою виразною здатністю глибоких моделей при достатній кількості навчальних даних;

– практичне значення роботи полягає в розробці програмної реалізації запропонованого методу на основі бібліотек Python (numpy, pandas, scikit-learn, keras). Створено набір jupyter-ноутбуків для відтворення експериментів і адаптації методу для інших задач. Результати роботи можуть бути використані для розв'язання практичних задач класифікації зображень в умовах обмежених або неякісних даних.

Таким чином, у магістерській роботі вирішено актуальну науково-практичну задачу підвищення ефективності розпізнавання образів шляхом розробки і дослідження моделі адаптивного подвійного неонечіткого нейрона. Показано переваги запропонованого підходу перед традиційними методами на репрезентативних наборах даних і створено програмну реалізацію для практичного використання.

Подальші дослідження можуть бути спрямовані на удосконалення архітектури нечіткого нейрона, розробку ефективних алгоритмів навчання, адаптацію методу для інших типів даних (наприклад, часових рядів), а також на інтеграцію нечітких моделей з глибокими нейронними мережами для поєднання їх переваг.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Бондаренко М. Ф., Шабанов-Кушнарченко С. Ю. Теорія розпізнавання образів : навч. посіб. Харків : ХНУРЕ, 2018. 292 с.
2. Гавриленко О. В., Рябуха Ю. М. Розпізнавання рукописних символів на основі згорткових нейронних мереж. *Радіoeлектроніка, інформатика, управління*. 2019. № 1. С. 111–120.
3. Дорошук М. Ю., Вітинський П. Б. Порівняльний аналіз методів розпізнавання зображень на наборі даних CIFAR-10. *Наукові записки Українського науково-дослідного інституту зв'язку*. 2020. № 2. С. 55–62.
4. Кулаковська Т. О., Москаленко В. В. Дослідження методів класифікації об'єктів на зображеннях з використанням нейромереж. *Вчені записки Таврійського національного університету імені В.І. Вернадського*. Серія: Технічні науки. 2019. Т. 30 (69), № 4, ч. 1. С. 239–246.
5. Нестеренко О. В., Савенков О. І. Порівняльне дослідження методів машинного навчання в задачі розпізнавання зображень. *Вісник Національного технічного університету «ХПІ»*. Серія: Системний аналіз, управління та інформаційні технології. 2018. № 44 (1266). С. 136–142.
6. Тарасенко О. О., Тимошенко Ю. О. Аналіз методів попередньої обробки зображень для задач класифікації. *Математичні машини і системи*. 2020. № 1. С. 90–98.
7. Харченко В. П., Чеботаренко О. В. Аналіз наборів даних для навчання згорткових нейронних мереж. *Кібернетика та системний аналіз*. 2019. Т. 55, № 3. С. 49–58.
8. Чумаченко О. І., Лебідь О. Г. Методи попередньої обробки зображень у системах комп'ютерного зору. *Штучний інтелект*. 2018. № 2. С. 372–381.
9. Шевчук А. В., Гвоздецька Н. М., Возна Н. Я. Методи аугментації зображень для навчання згорткових нейромереж. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. 2020. № 38. С. 225–232.

10. Яшина О. М., Ткаченко А. А. Методика оцінювання якості розпізнавання зображень за допомогою нейронних мереж. *Вісник Хмельницького національного університету*. Технічні науки. 2019. № 3(273). С. 116–124.

11. Субботін С. О., Олійник А. О. Нейронні мережі Такагі-Сугено в задачах моделювання і класифікації. *Науковий вісник Національного гірничого університету*. 2018. № 2. С. 79–86.

12. . Рибальченко М. О., Вишневський В. В. Порівняльний аналіз методів машинного навчання для розпізнавання зображень. *Системи управління, навігації та зв'язку*. 2019. Вип. 1 (53). С. 171–176.

13. Герасимов Б. М., Тарасов В. О., Токарев В. В. Системи підтримки прийняття рішень: проектування, застосування, оцінка ефективності : монографія. Запоріжжя : ЗНТУ, 2018. 315 с.

14. Мельничук Ю. Р., Піскозуб А. З. Порівняння методів машинного навчання для задачі класифікації текстів. *Наукові праці Донецького національного технічного університету*. Серія: Обчислювальна техніка та автоматизація. 2019. № 1 (32). С. 91–100.

15. Бабкіна Н. В., Золотухіна О. А., Скачков В. О. Нечіткі моделі прийняття рішень в інтелектуальних системах управління виробництвом. *Сучасні інформаційні системи*. 2020. Т. 4, № 1. С. 34–39.

16. Тимошук П. В., Тиш Є. В. Застосування згорткових нейронних мереж для розпізнавання медичних зображень. *Штучний інтелект*. 2019. № 1–2. С. 138–148.