МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет водного господарства та природокористування
Навчально–науковий інститут автоматики, кібернетики та обчислювальної техніки
Кафедра комп'ютерних технологій та економічної кібернетики

До захисту допущений
Завідувач кафедри
комп'ютерних технологій та
економічної кібернетики
д. е. н., проф. П. М. Грицюк
_____
«____» _____ 2021 р.

КВАЛІФІКАЦІЙНА РОБОТА НА ЗДОБУТТЯ ОСВІТНЬО-КВАЛІФІКАЦІЙНОГО РІВНЯ «БАКАЛАВР»

на тему:

«РЕАЛІЗАЦІЯ ЗАДАЧ КЛАСИФІКАЦІЇ ТА ПРОГНОЗУВАННЯ НА ПЛАТФОРМІ PYTHON»

Виконав:
здобувач вищої освіти за ОПП «Інформаційні системи і технології» спеціальності 126 «Інформаційні системи та технології», групи ІСТ–41
Котвицький Олексій Миколайович

Керівник:
проф. **Грицюк П. М.**

Рівне – 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет водного господарства
та природокористування
Навчально-науковий інститут автоматики, кібернетики та
обчислювальної техніки

Кафедра комп'ютерних технологій та економічної кібернетики

Освітньо-кваліфікаційний рівень - бакалавр
Освітньо-професійна програма «Інформаційні системи і технології»
Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри
комп'ютерних технологій та
економічної кібернетики
д.е.н., професор П.М. Грицюк
_____
"__"_____ 2021 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

(прізвище, ім'я, по батькові)

1.Тема роботи: *«Реалізація задач класифікації та прогнозування на платформі Python»*

керівник роботи: *Грицюк Петро Михайлович, д.е.н.,професор, завідувач кафедри комп'ютерних технологій та економічної кібернетики*

затверджена наказом по університету *С №262 від 02.04.2021 року*

2. Термін здачі студентом закінченої роботи *18.06.2021 року*

3. Вихідні дані до роботи *Вихідні дані – фрагменти відео, які містять групи людей. Об'єкт дослідження – задачі класифікації та прогнозування. Предмет дослідження – методика реалізації класифікації об''єктів на платформі Python.*

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

Розділ 1. Огляд теорії сучасних інтелектуальних технологій. Поняття про штучні нейронні мережі. Архітектура та принип роботи і навчання нейронних мереж.

Розділ 2. Огляд середовища Python. Побудова штучних нейронних мереж у Python. Використання ШНМ для вирішення задач класифікації та прогнозування.

Розділ 3. Прогнозування ціни криптовалют з використанням Python. Ідентифікація та класифікація динамічних об''єктів (людських фігур) на платформі Python.

5. Перелік графічного матеріалу

Презентація. Таблиці. Рисунки. Фрагменти відео.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Розділ 1	*Грицюк П.М.*		
Розділ 2	*Грицюк П.М.*		
Розділ 3	*Грицюк П.М.*		

Дата видачі завдання _9 березня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Термін виконання етапів (роботи)	Примітка
1	Робота над Розділом 1.	*9.03.2021 – 21.03.2021*	
2	Робота над Розділом 2.	*22.03.2021 – 4.04.2021*	
3	Робота над Розділом 3.	*5.04.2021 – 25.04.2021*	
4	Підготовка пояснювальної записки	*26.04.2021 – 19.05.2021*	
5	Написання вступу, висновків, реферату	*20.05.2021 - 30.05.2021*	
6	Попередній захист роботи	*01.06.2021 - 06.06.2021*	
7	Підготовка презентації роботи	*9.06.2021*	
8	Відгук керівника, рецензування роботи, перевірка на плагіат	*11.06.2021 - 14.06.2021*	
9	Допуск до захисту	*15.06.2021*	

Студент(ка) _____ .
(підпис) (прізвище і ініціали)

Керівник кваліфікаційної роботи
_____ ____
(підпис) (прізвище і ініціали)

РЕФЕРАТ

Кваліфікаційна робота на здобуття освітньо-кваліфікаційного рівня «бакалавр»: 70 стор., 40 рис., 2 табл., додатки на 3 стор., 27 літературних джерел.

Актуальність теми даної кваліфікаційної роботи пов’язана з важливістю динамічної класифікації людей у натовпі з видачею попереджень для груп ризику. Ця задача набула актуальності з розвитком пандемії коронавірусу та введенням карантинних обмежень у всіх країнах світу. Ідентифікація людських фігур на відеозображеннях та класифікація їх за групами ризику дозволить автоматизувати процес моніторингу поведінки та переміщень людей у громадських місцях.

Об’єктом дослідження кваліфікаційної роботи є задачі класифікації та прогнозування.

Предметом дослідження кваліфікаційної роботи є методика реалізації задачі ідентифікації людських фігур на відеозображеннях та їх класифікація за групами ризику.

Метою даної кваліфікаційної роботи є написання та реалізація програми у середовищі Python для для ідентифікації людських фігур на відеозображеннях та їх класифікація за групами ризику.

У кваліфікаційній роботі були використані наступні методи дослідження: бібліографічний метод – для пошуку джерел даних для досліджень (фото та відео фрагменти); метод зворотного поширення помилки – для навчання нейронної мережі; алгоритмічний – для побудови алгоритму та реалізації його у середовищі Python; метод згорткової фільтрації для визначення горизонтальних країв зображень; графічний – для ілюстрації алгоритму та отриманих результатів.

КЛЮЧОВІ СЛОВА: PYTHON, КЛАСИФІКАЦІЯ, ПРОГНОЗУВАННЯ, ІДЕНТИФІКАЦІЯ, ВІДЕО, ГРУПА РИЗИКУ

ЗМІСТ

ВСТУП..... 7
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ..... 9
1.1 Сучасний стан розвитку інтелектуальних технологій..... 9
1.2 Основні моделі та методи інтелектуальних технологій..... 11
1.3 Штучні нейронні мережі 22
Висновки по розділу 1 29
РОЗДІЛ 2. РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ З ВИКОРИСТАННЯМ PYTHON 30
2.1. Принципи за якими навчаються штучні нейронні мережі..... 30
2.2. Використання штучних нейронних мереж у задачах класифікації та прогнозування..... 38
2.3. Побудова штучних нейронних мереж з використанням Python 41
Висновки по розділу 2 47
РОЗДІЛ 3. ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ВИРІШЕННЯ ЗАДАЧ КЛАСИФІКАЦІЇ ТА ПРОГНОЗУВАННЯ..... 48
3.1. Прогнозування ціни крипто валют з використанням Python..... 48
3.2. Класифікації бази даних «Титанік»..... 53
3.3. Класифікація динамічних об’єктів 60
Висновки по розділу 3 65
ВИСНОВКИ..... 66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... 68
ДОДАТКИ..... 71

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IT - Інтелектуальні технології

ІКТ - Інформаційних і комунікаційних технологій

RL - Reinforcement Learning

AI – Artificial Intelligence

ML – Machine Learning

GPU - Graphics Processing Unit

IDE - Integrated Drive Electronics

BTC - Bitcoin

ВСТУП

Поява інтересу до машинного навчання пов'язане з тими ж факторами, які роблять інтелектуальний аналіз даних і Байєсове виведення все більш популярними. Це в першу чергу зростаючі обсяги і різноманітність доступних даних, більш дешева і потужна обчислювальна обробка даних, а також доступні сховища даних.

Все це означає, що можна швидко і автоматично створювати моделі, які можуть аналізувати більші і складні дані і надавати більш швидкі і точні результати – навіть в дуже великому масштабі. А завдяки побудові точних моделей у організації чи підприємства з'являється більше шансів визначити прибуткові можливості або уникнути невідомих ризиків.

Актуальність дослідження. Сьогодні Python став одним з найпопулярніших мов програмування серед розробників по всьому світу - від автоматизації процесів до створення сценаріїв, веб-розробки і машинного навчання - він використовується всюди. Більшість галузей, що працюють з великими обсягами даних, усвідомили цінність технології машинного навчання. Збираючи ідеї з цих даних - часто в режимі реального часу, - підприємства та організації можуть працювати більш ефективно або отримати перевагу над конкурентами. Однак в деяких випадках використання нейронних мереж може викликати деякі складності, як от, наприклад, у випадку відсутності достатніх даних.

Мета наукової роботи – охарактеризувати та проаналізувати методи реалізації інтелектуальних технологій з використанням платформи Python, а також запропонувати методи застосування нейронних мереж для вирішення задач класифікації та прогнозування.

Завдання дослідження:

– сформувати поняття інтелектуальних технологій і описати методи їх реалізації з використанням Python;

– розглянути основні моделі та методи інтелектуальних технологій;

– проаналізувати тенденції розвитку та використання штучних нейронних мереж в задачах класифікації та прогнозування;

– проаналізувати та удосконалити методику застосування нейронних мереж до вирішення задач класифікації та прогнозування;

– розробити методику виділення та класифікації динамічних об’єктів з використанням штучних нейронних мереж.

Об’єкт дослідження – задачі класифікації та прогнозування на платформі Python.

Предмет дослідження – реалізація задач класифікації та прогнозування на платформі Python з використанням штучних нейронних мереж.

Методи дослідження. При виконанні роботи були використані такі методи наукового дослідження: опис, аналіз, пояснення, узагальнення інформації. Робота базується на теоретичному вивченні інтелектуальних технологій з використанням Python, ознайомлення з можливостями використання нейронних мереж, а також практичному створенні прикладів використання машинного навчання з допомогою мови програмування Python. Застосовуються комп’ютерні програми та системи для прогнозування та класифікації.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ

1.1 Сучасний стан розвитку інтелектуальних технологій

Через постійне збільшення об’ємів даних та взаємозв’язків між ними стало дуже важко (а інколи і неможливо) знайти відповіді на запитання в інформаційному просторі. Для розуміння людиною деякої предметної області з’явилась необхідність отримати додатковий інструмент, роль якого зараз відіграють комп’ютери та автоматизовані інформаційні системи.

Звернення до теми нових інтелектуальних технологій пов'язано з ідеєю переходу людства в нову еру. Її тривалість і характер визначаються швидкістю і глибиною перетворень інтелектуальної культури. Значний ризик, який при цьому виникає – протиріччя між потребами суспільства в інтелектуальній революції. З одного боку, за рахунок цього суспільство з новою культурою і в зміненій цивілізаційній оболонці має перейти в нову еру, з іншого боку, зростають процеси масової інтелектуальної деградації, бюрократизації та падіння творчих здібностей. У той же час людство живе в інформативно пересиченому середовищі і, при цьому, інтелектуальна культура суспільства масового споживання трагічно відстає за складністю систем життєзабезпечення. Створення нових інтелектуальних технологій, в свою чергу, вимагає осмислення причин і механізмів, що викликали застій в інтелектуальній культурі, науці та освіті.

Конструктивними в роботі над інтелектуальними технологіями є цілі синтезу, в тому числі асоціації науково-дослідницької, навчальної, проектної діяльності, які доцільно відкривати на базі будь-якого університету і факультету [1].

Отже, інтелектуальні технології (ІТ) – це способи та методи використання комп'ютерів для зберігання, вилучення, передачі та управління даними або інформацією. ІТ зазвичай використовуються в контексті бізнес-операцій, на відміну від персональних або розважальних технологій. ІТ вважаються різновидом інформаційних і комунікаційних технологій (ІКТ). Система інформаційних технологій (ІТ-система), як правило, являє собою інформаційну систему, систему зв'язку

або, більш конкретно, комп'ютерну систему, включаючи все обладнання, програмне забезпечення та периферійні пристрої, керовану обмеженою групою IT-користувачів [2].

Будь-яка комп`ютерна інформаційна система зазвичай реалізовує інформаційний процес, виконуючи наступні функції: приймає інформаційні запити розв'язання задачі (1) і необхідні початкові дані, обробляє запити і дані (2), що зберігаються в системі, у відповідності з відомими алгоритмами і формує необхідну вихідну інформацію (3) (рис. 1.1) [3, с 13].

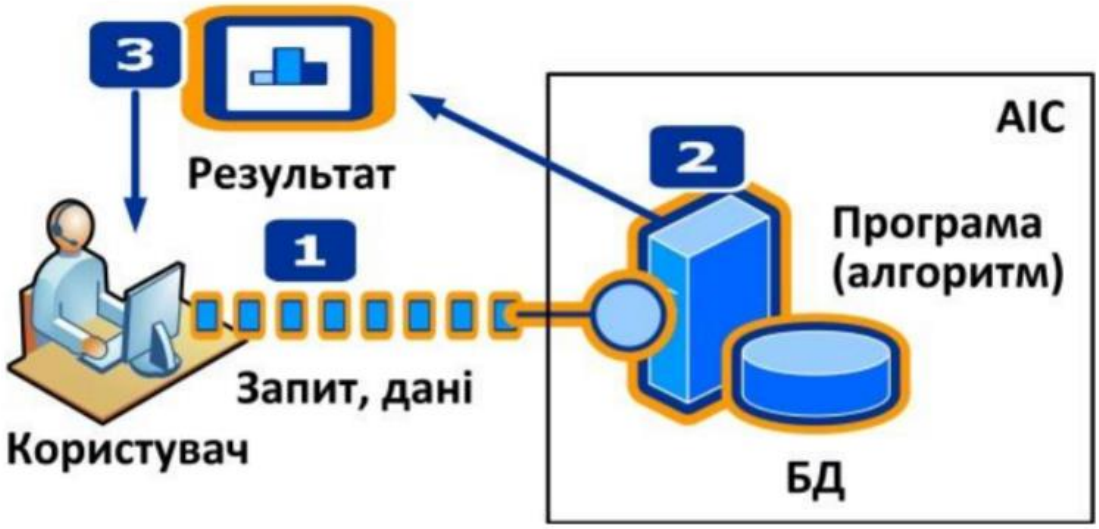

Рис. 1.1 – Інформаційний процес, що реалізується комп’ютерною системою

Двома найбільшими рушіями недавнього прогресу були [4, с. 10]:

- Наявність даних. Зараз люди проводять більше часу на цифрових пристроях (ноутбуках, мобільних пристроях). Їхня цифрова діяльність генерує величезні обсяги даних, які ми можемо подавати до своїх алгоритмів навчання.
- Обчислювальна можливості. Лише нещодавно ми почали мати можливість навчати нейронні мережі, які є досить потужними, щоб скористатися величезними наборами даних, які ми маємо зараз.

Інтелектуальна система може припускати зовнішнє керування, але для неї характерною є самокерованість. Система має певну мету і прагне так планувати

свої дії, щоб досягати цієї мети. Як вхідні стимули системи можна розглядати поточну ситуацію, що сприймається і аналізується системою. Результатом реакції системи стає зміна зовнішньої ситуації, і поведінка системи коригується в залежності від того, бажаною чи небажаною є ця зміна [5, с. 14].

Вихідні дані навчання моделі можуть використовуватися для виведення, що означає прогнозування нових даних. Модель – це дистильоване уявлення про те, чого навчилася система машинного навчання. Моделі машинного навчання на кшталт математичних функцій – приймають запит у вигляді вхідних даних, роблять прогноз на основі цих вхідних даних, а потім видають відповідь.

Таким чином, перехід до нової епохи супроводжується постійним накопиченням знань і методів, відбувається «стиснення» знань, яке супроводжується їхніми непоправними втратами. Відбувається надзвичайно стрімкий розвиток інтелектуальних технологій в усіх сферах виробничої, господарської та інтелектуальної діяльності. У цих умовах потрібні нові інтерфейси для перетворення інформаційних потоків в знання (зрозумілу, інтерпретовану інформацію), тощо. Робота над новими інтелектуальними технологіями передбачає створення когнітивних інструментів, що змінюють основні підходи до сприйняття, обробки, використання інформації, при застосуванні візуалізації різних типів і рівнів жорсткості.

1.2 Основні моделі та методи інтелектуальних технологій

Зазвичай буває важко, а іноді і неможливо простежити за поведінкою реальних систем в різних умовах або змінити ці системи. Вирішити дану проблему допомагають моделі. Побудувавши модель системи, можна багаторазово повертатися до початкового її стану, а також спостерігати за її поведінкою в умовах, що змінюються.

Модель (лат. "Modulus" - міра) - об'єкт-заступник об'єкта-оригіналу, що забезпечує вивчення деяких властивостей останнього; спрощене уявлення системи

для її аналізу і передбачення, а також отримання якісних і кількісних результатів, необхідних для прийняття правильного управлінського рішення.

При вирішенні конкретної задачі, коли необхідно виявити певну властивість досліджуваного об'єкта, модель виявляється не тільки корисним, але і часом єдиним інструментом дослідження. Один і той же об'єкт може мати безліч моделей, а різні об'єкти можуть описуватися однією моделлю.

Під терміном "моделювання" зазвичай розуміють процес створення точного опису системи; метод пізнання, що складається з створення і дослідження моделей.

Моделювання полегшує вивчення об'єкта з метою його створення, подальшого перетворення і розвитку. Воно використовується для дослідження існуючої системи, коли реальний експеримент проводити недоцільно через значні фінансові і трудові витрати, а також при необхідності проведення аналізу проектованої системи, тобто яка ще фізично не існує в даній організації.

Для проектування інформаційної системи використовують інформаційні моделі, що представляють об'єкти і процеси в формі малюнків, схем, креслень, таблиць, формул, текстів і т.п.

За допомогою формальних мов будують інформаційні моделі певного типу - формально-логічні моделі. Процес побудови інформаційних моделей за допомогою формальних мов називають формалізацією.

Моделі, побудовані з використанням математичних понять і формул, називають математичними моделями.

Концептуальне моделювання об’єкта зазвичай передує математичному моделюванню. Воно являє собою структурований процес створення систем, що складається з наступних етапів:

– Аналіз;

– Проектування;

– Програмування;

– Тестування;

– Впровадження.

Існує декілька методів і принципів побудови інформаційних систем (автоматизованих ІС), серед яких можна виділити: методи "знизу-вгору", "зверху-вниз", принципи "дуалізму", багатокомпонентний та інші.

Метод "знизу-вгору".

Досвід і методи роботи вітчизняних програмістів сформувалися у великих обчислювальних центрах, основною метою яких було не створення тиражованих продуктів, а виконання завдань конкретної установи. Сучасні керівники часто вдаються до нього, вважаючи, що їм зручно мати своїх фахівців. Розробка програм "знизу-вгору", що здійснюється кваліфікованими програмістами, дозволяє автоматизувати, як правило, окремі робочі процеси. Такий метод досить витратний і все рідше використовується, особливо в малих і середніх підприємствах.

Метод "зверху-вниз".

Розвиток комерційних та інших сучасних структур послужило підставою до формування ринку різних програмних засобів. Найбільший розвиток отримали ІС, орієнтовані на автоматизацію ведення бухгалтерського аналітичного обліку і технологічних процесів. В результаті з'явилися ІС, розроблені сторонніми, як правило, спеціалізованими організаціями та групами фахівців "зверху", в припущенні, що одна ІС зможе задовольняти потреби багатьох користувачів.

Принципи "дуалізму" і багатокомпонентності.

Розвиток систем і підприємств, збільшення числа їх філій і клієнтів, підвищення якості обслуговування і інше викликали істотні зміни в розробці і функціонуванні автоматизованих ІС (АІС), в основному базуються на збалансованому поєднанні двох попередніх методів.

Багатокомпонентна система забезпечує дотримання основоположного принципу побудови АІС - відсутність дублювання введення вихідних даних.

Майже необмежена кількість доступних даних, доступне зберігання даних та зростання менш дорогої та потужнішої обробки сприяли зростанню машинного навчання. Зараз багато галузей розробляють більш надійні моделі машинного навчання, здатні аналізувати більші та складніші дані, забезпечуючи при

цьому швидші та точніші результати в широких масштабах. Інструменти машинного навчання дозволяють організаціям швидше визначати вигідні можливості та потенційні ризики.

Практичне застосування машинного навчання зумовлює ділові результати, які можуть суттєво вплинути на суть компанії. Нові методи в цій галузі швидко розвиваються і розширили застосування машинного навчання майже до безмежних можливостей. Галузі, які залежать від великої кількості даних – їм потрібна система для їх ефективного та точного аналізу, сприйняли машинне навчання як найкращий спосіб побудови моделей, стратегій та планування.

Машинне навчання (Machine Learning – ML) – це великий підрозділ штучного інтелекту, що вивчає методи побудови алгоритмів, здатних навчатися (рис. 1.2). Машинне навчання – це процес, під час перебігу якого система опрацьовує велику кількість прикладів, виявляє закономірності і використовує їх, щоб прогнозувати вихідні характеристики для нових вхідних даних [6, с. 6].

Рис. 1.2 – Машинне навчання і його складові

У машинному навчанні з учителем і без вчителя модель описує сигнал в шумі або патерн, виявлений на основі даних навчання.

У навчанні з підкріпленням модель описує найкращий можливий спосіб дій в конкретній ситуації.

Остаточний набір параметрів навчання (інформація, яку містить модель) залежить від конкретного типу моделі – в глибоких нейронних мережах модель є кінцевим станом, в регресії вона містить коефіцієнти, а в рішенні дерева вона містить розділені локації.

Всі моделі машинного навчання діляться на контрольовані і неконтрольовані. Якщо модель є контрольованою, вона потім поділяється на регресійну або класифікаційну модель. Ми розглянемо, що означають ці терміни, і відповідні моделі, які потрапляють в кожну категорію.

Десять описаних методів пропонують огляд і основу, на які можна спиратися, відточуючи свої знання і навички в області машинного навчання:

– Регресія;

– Класифікація;

– Кластеризація;

– Зменшення розмірності;

– Ансамблеві методи;

– Нейронні мережі і глибоке навчання;

– Трансферне навчання;

– Навчання з підкріпленням;

– Обробка природної мови;

– Вкладення слів.

Регресія відноситься до категорії контрольованого машинного навчання. Вона допомагає передбачити або пояснити конкретне числове значення на основі набору попередніх даних, наприклад, прогнозування ціни власності на основі попередніх даних про ціни на аналогічні об'єкти.

Найпростіший варіант – це лінійна регресія, де використовується математичне рівняння лінії (y = m * x + b) для моделювання набору даних. Навчається модель лінійної регресії на основі навчальної вибірки з багатьма парами даних (x, y). При цьому обчислюючи положення і нахил лінії регресії, які мінімізують загальну відстань між усіма точками даних і лінією. Іншими словами,

обчислюється нахил (m) і точку перетину по осі y (b) для лінії, яка найкращим чином апроксимує дані спостережень (рис. 1.3).

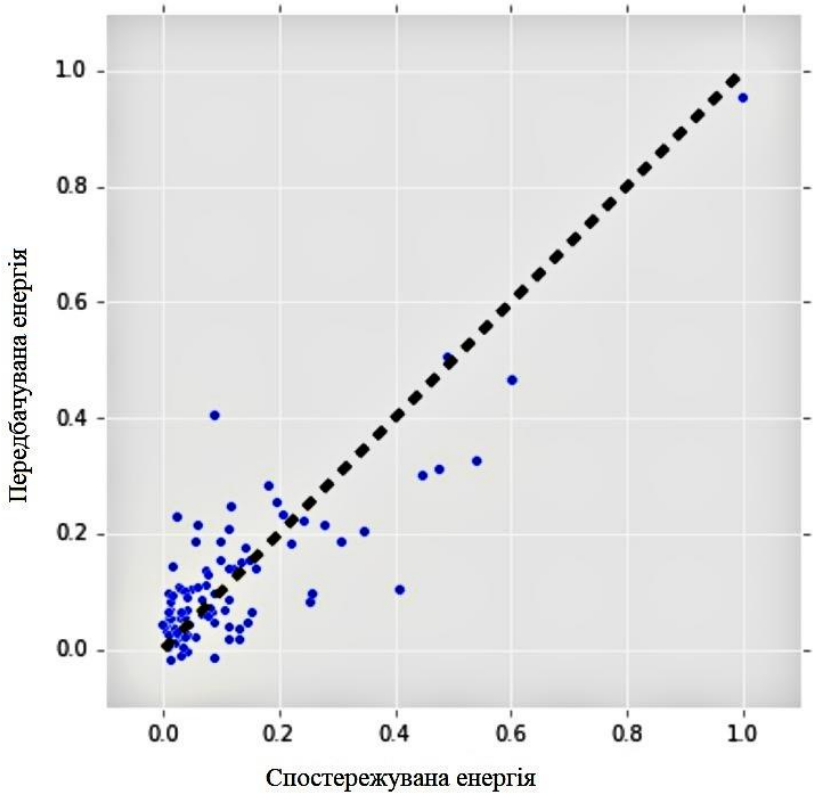

Рис. 1.3. – Модель лінійної регресії для оцінки енергоспоживання будівлі (кВтг) [21].

Методи регресії охоплюють діапазон від простих (наприклад, лінійна регресія) до складних (наприклад, регуляризована лінійна регресія, поліноміальна регресія, дерева рішень і випадкові регресії, нейронні мережі та інші).

Наступні методи – класифікації, передбачають або пояснюють значення класу. Наприклад, вони можуть допомогти передбачити, чи купить продукт онлайн-покупець. Висновок може бути так чи ні. Але методи класифікації не обмежуються двома класами. Наприклад, метод класифікації може допомогти оцінити, чи містить дане зображення автомобіль або вантажівку. У цьому випадку на виході будуть 3 різних значення: 1) зображення містить автомобіль, 2) зображення містить вантажівку, або 3) зображення не містить ні автомобіля, ні вантажівки.

На зображенні нижче показані оцінки попередніх студентів і їх зарахування. Логістична регресія дозволяє нам провести лінію, яка представляє кордон прийняття рішення (рис. 1.4.).

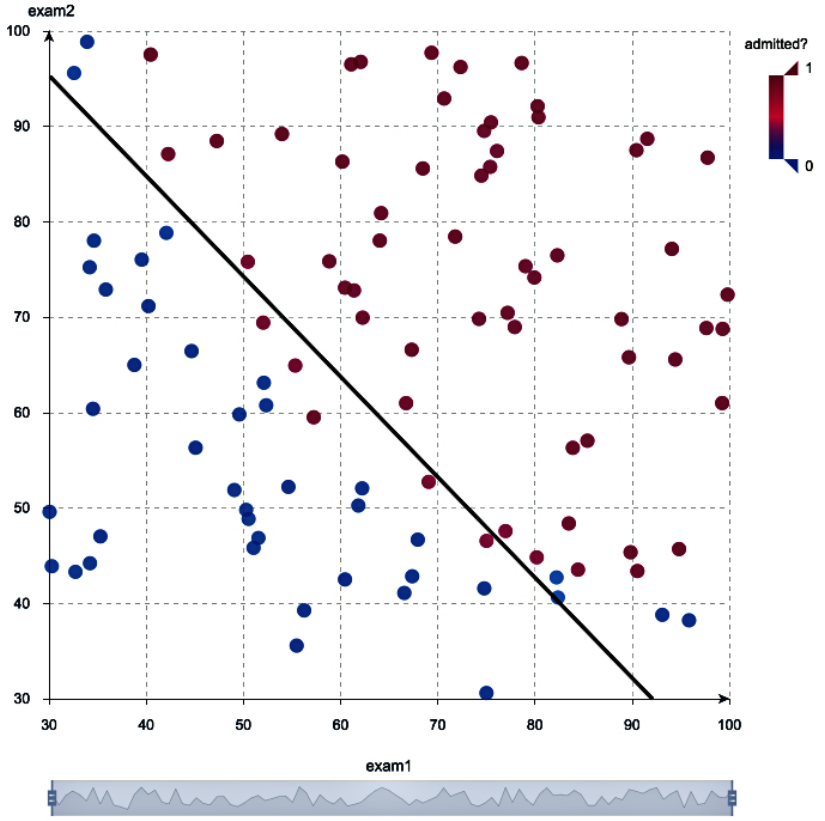

Рис. 1.4. - Кордон рішення логістичної регресії: вступати до коледжу чи ні [21]?

За допомогою методів кластеризації потрапляємо в категорію машинного навчання без учителя, бо їхня мета – згрупувати або кластеризувати спостереження, що мають схожі характеристики. Методи кластеризації не використовують вихідну інформацію для навчання, а замість цього дозволяють алгоритму визначати вихідні дані. В методах кластеризації можна використовувати тільки візуалізації для перевірки якості рішення.

Одним з найбільш поширених методів кластеризації є метод К-середніх [22]. На наступній діаграмі К-середні застосовуються до набору даних про будівлі. Кожен стовпець на графіку показує енергетичну ефективність кожної будівлі. Чотири виміри споживання енергії відносяться до кондиціонування повітря. та роботі підключеного до електромережі обладнання (мікрохвильові печі,

холодильники тощо). Було обрано K = 2 для кластеризації, що дозволяє легко інтерпретувати один із кластерів як групу ефективних будівель, а інший кластер як групу неефективних будівель. На рисунку 1.5. зображено план розташування будівель.

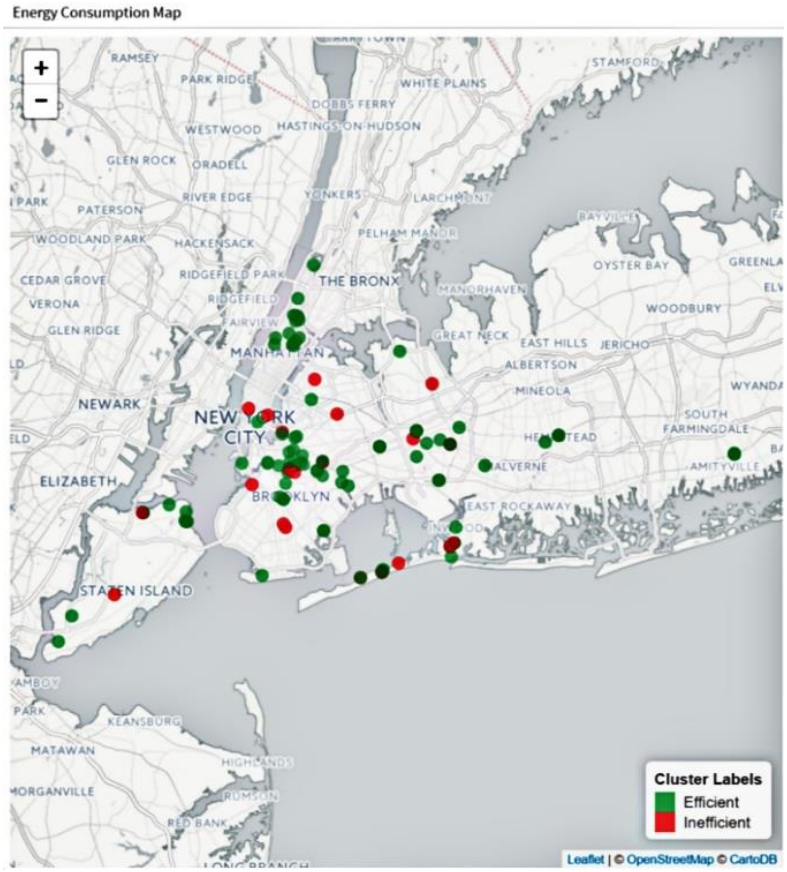

Рис. 1.5. – План розташування будівель

На рисунку 1.6. відображена залежність двох з чотирьох показників, які було використано в якості вхідних: підключене устаткування і опалювальний газ.

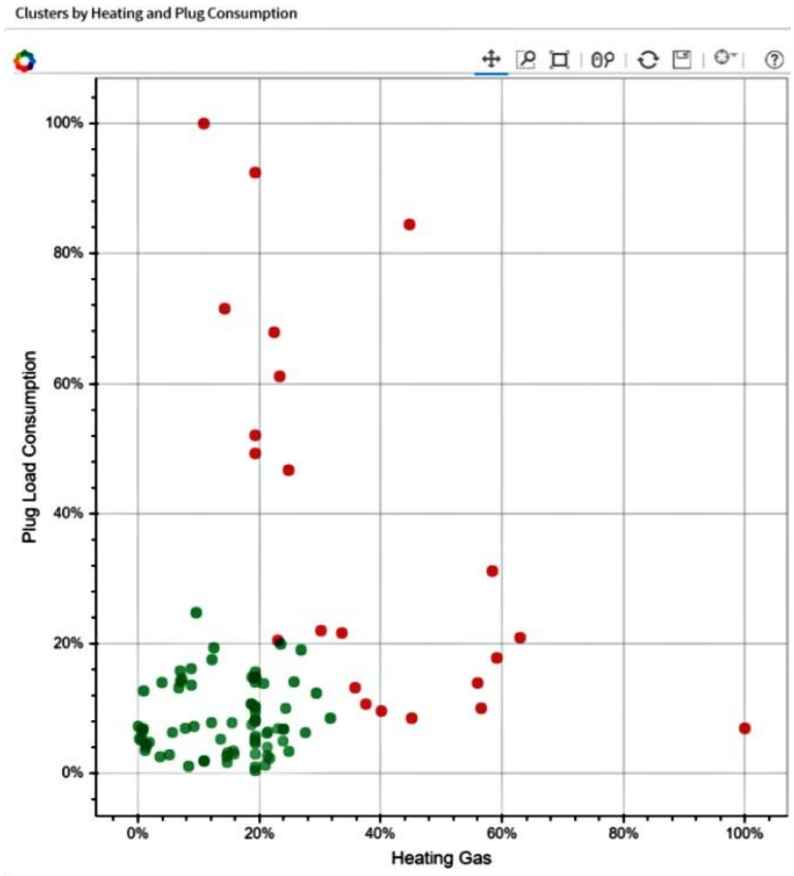

Рисунок 1.6. – Кластеризація будівель в ефективні (зелені) і неефективні (червоні) групи.

Метод зменшення розмірності використовується, щоб видалити найменш важливу інформацію (іноді надлишкові стовпці) з набору даних. На практиці часто є набори даних з сотнями або навіть тисячами стовпців, тому скорочення загального числа показників є дуже важливим. Наприклад, зображення можуть включати тисячі пікселів, не всі з яких мають значення для аналізу. Або при тестуванні мікро чіпів в процесі виробництва до кожного чіпу можуть бути застосовані тисячі вимірювань і тестів, багато з яких надають надлишкову інформацію. У цих випадках знадобляться алгоритми зменшення розмірності, щоб зробити набір даних керованим.

У ансамблевих методах використовується ідея комбінування декількох прогнозних моделей (контрольованого машинного навчання) для отримання більш якісних прогнозів. Наприклад, алгоритми випадкового лісу - це метод

ансамблю, який об'єднує множину дерев рішень, навчених з різними вибірками наборів даних. В результаті якість прогнозів випадкового лісу є вищою, ніж якість прогнозів, оцінених за допомогою одного дерева рішень.

Нейронні мережі і глибоке навчання. На відміну від лінійних і логістичних регресій, які вважаються лінійними моделями, мета нейронних мереж - вловлювати нелінійні закономірності в даних шляхом додавання в модель прихованих шарів параметрів. На наступному зображенні проста нейронна мережа має три входи, один прихований шар з п'ятьма параметрами і вихідний шар з одним вихідним нейроном (рис. 1.7.).

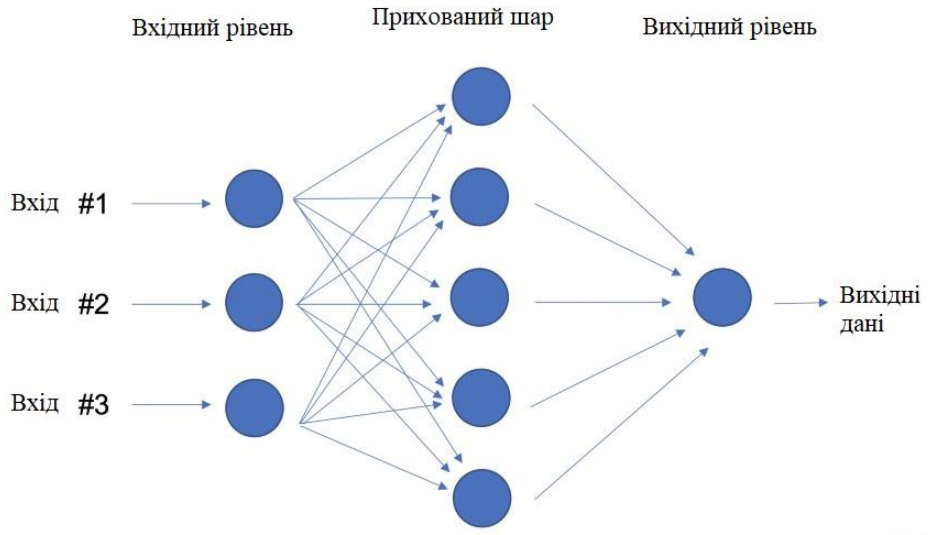

Рис. 1.7. – Нейронна мережа з одним прихованим шаром.

Таким чином, структура нейронних мереж досить гнучка, щоб побудувати добре відому лінійну і логістичну регресію. Але, для відображення нелінійних зв’язків та залежностей необхідно використовувати нейронні мережі з декількома прихованими шарами. Термін «глибоке навчання» походить від нейронної мережі з багатьма прихованими шарами (рис. 1.8.) і включає в себе широкий спектр архітектур [6, с. 13].

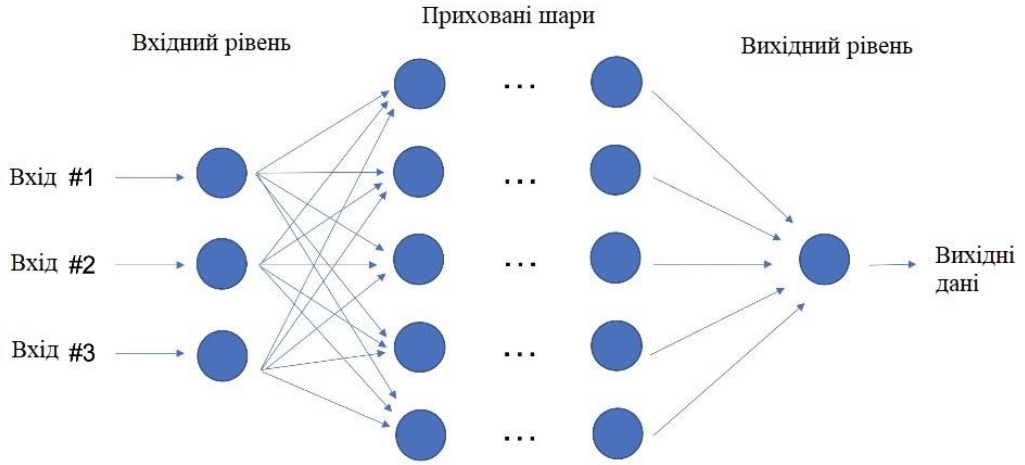

Рис. 1.8. – Нейронна мережа з багатьма прихованими шарами

Для оптимальної роботи методи глибокого навчання вимагають великої кількості даних і великої обчислювальної потужності.

Трансферне навчання відноситься до повторного використання частини раніше навченої нейронної мережі та адаптації її до нової, але аналогічної задачі. Зокрема, після навчання нейронної мережі з використанням даних для завдання можна передати частину навчених шарів і об'єднати їх з декількома новими шарами, які можна навчати, використовуючи дані нового завдання. Додавши кілька шарів, нова нейронна мережа зможе швидко навчитися і адаптуватися до нової задачі.

Одним з ефективних методів моделювання складних систем є агентне моделювання. Під агентом у агентній моделі процесів управління розуміється елемент моделі, який може мати поведінку, пам’ять (історію), контакти та взаємодіяти зі іншими агентами системи. Агентні системи можуть моделювати поведінку елементів системи. **Навчання з підкріпленням** (Reinforcement Learning - RL) – це метод машинного навчання, який допомагає агенту навчатися на власному досвіді. Реєструючи дії і використовуючи метод проб і помилок в заданому середовищі, метод RL може максимізувати сукупну винагороду. Наприклад: миша - це агент, а лабіринт - це середовище. Набір можливих дій для миші: рух вперед, назад, вліво або вправо. Нагорода - сир.

Обробка природної мови (NLP) - це не метод машинного навчання як такого, а, скоріше, широко використовуваний метод підготовки тексту для машинного навчання. На даний момент найпопулярнішим пакетом для обробки тексту є NLTK (Natural Language ToolKit), створений дослідниками з Стенфорда.

Вкладення слів – це метод, заснований на нейронних мережах, який зіставляє слова в корпусі з числовим вектором. Потім можна використовувати ці вектори для пошуку синонімів, виконання арифметичних операцій зі словами або буде представляти текстових документів (взявши середнє значення всіх векторів слів в документі).

Інформаційна модель - це модель об'єкта, процесу або явища, в якій представлені інформаційні аспекти модельованого об'єкта, процесу або явища. Вона є основою розробки моделей інформаційних систем.

Поряд з природними мовами (російська, англійська і т.д.) розроблені і використовуються формальні мови: системи числення, алгебра висловлювань, мови програмування та ін.

Машинне навчання – це не тільки математична, а й практична інженерна дисципліна. Машинне навчання – важлива тема в дослідженнях і промисловості, і постійно розробляються нові методології. Швидкість і складність цієї області ускладнює освоєння нових технік навіть для експертів – і інколи пригнічує новачків.

1.3 Штучні нейронні мережі

Великий інтерес сьогодні викликають обчислювальні структури нового типу, а саме – штучні нейронні мережі (ШНМ).

На назву “нейронні мережі” зараз претендують усі обчислювальні структури, які в тій чи іншій мірі моделюють роботу мозку. Але таке моделювання, здебільшого, є дуже фрагментарним, і говорити про створення у найближчому майбутньому штучного мозку або навіть деякої його моделі, яка дублювала б роботу мозку найпримітивніших живих створінь, ще зарано [7, с. 12].

Сьогодні не існує загальновизнаного означення штучної нейронної мережі, оскільки їх використанням займаються спеціалісти в різних галузях науки. Взаємному розумінню заважають методологічні та термінологічні бар'єри.

Штучна нейронна мережа є структурою, яка складається з великої кількості процесорних елементів, кожен з яких має локальну пам'ять і може взаємодіяти з іншими процесорними елементами за допомогою комунікаційних каналів з метою передачі даних, що можуть бути інтерпретовані довільним чином. Процесорні елементи незалежно в часі обробляють локальні дані, що поступають до них через вхідні канали. Зміна параметрів алгоритмів такої обробки залежить тільки від характеристик даних (рис.1.9.).

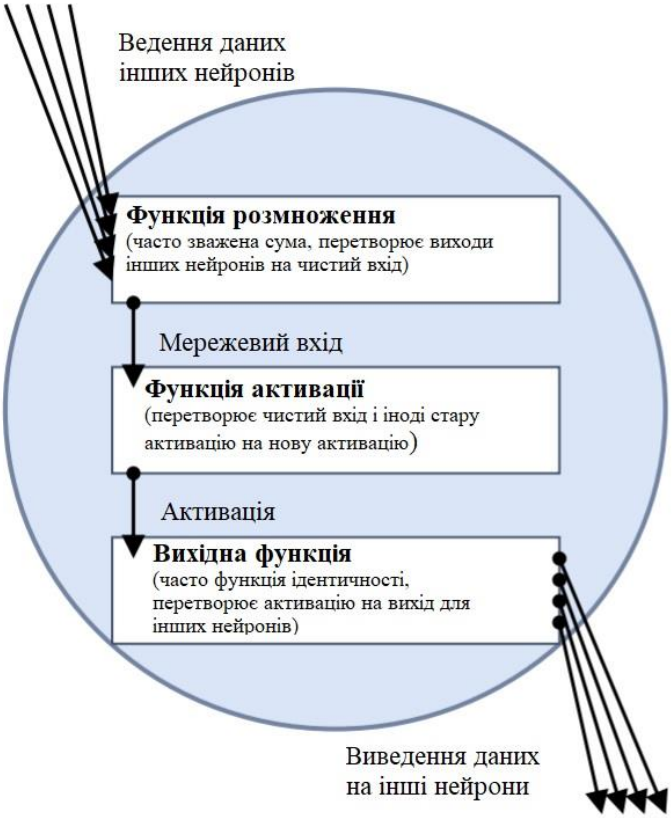

Рис. 1.9. – Обробка даних нейрона. Функція активації нейрона передбачає порогове значення [8, с 35]

Іншими словами, штучні нейронні мережі – це обчислювальні парадигми, які реалізують спрощені моделі біологічних нейронних мереж (БНМ). Під БНМ

будемо розуміти локальні ансамблі біологічних нейронів, які об’єднані синаптичними зв’язками. Сукупність таких ансамблів формує мозок із його різноманітними функціональних можливостями [7, с. 37].

Загалом відома велика кількість нейронних структур та їх модифікацій, найбільш відомі з яких показані нижче (рис 1.10.).

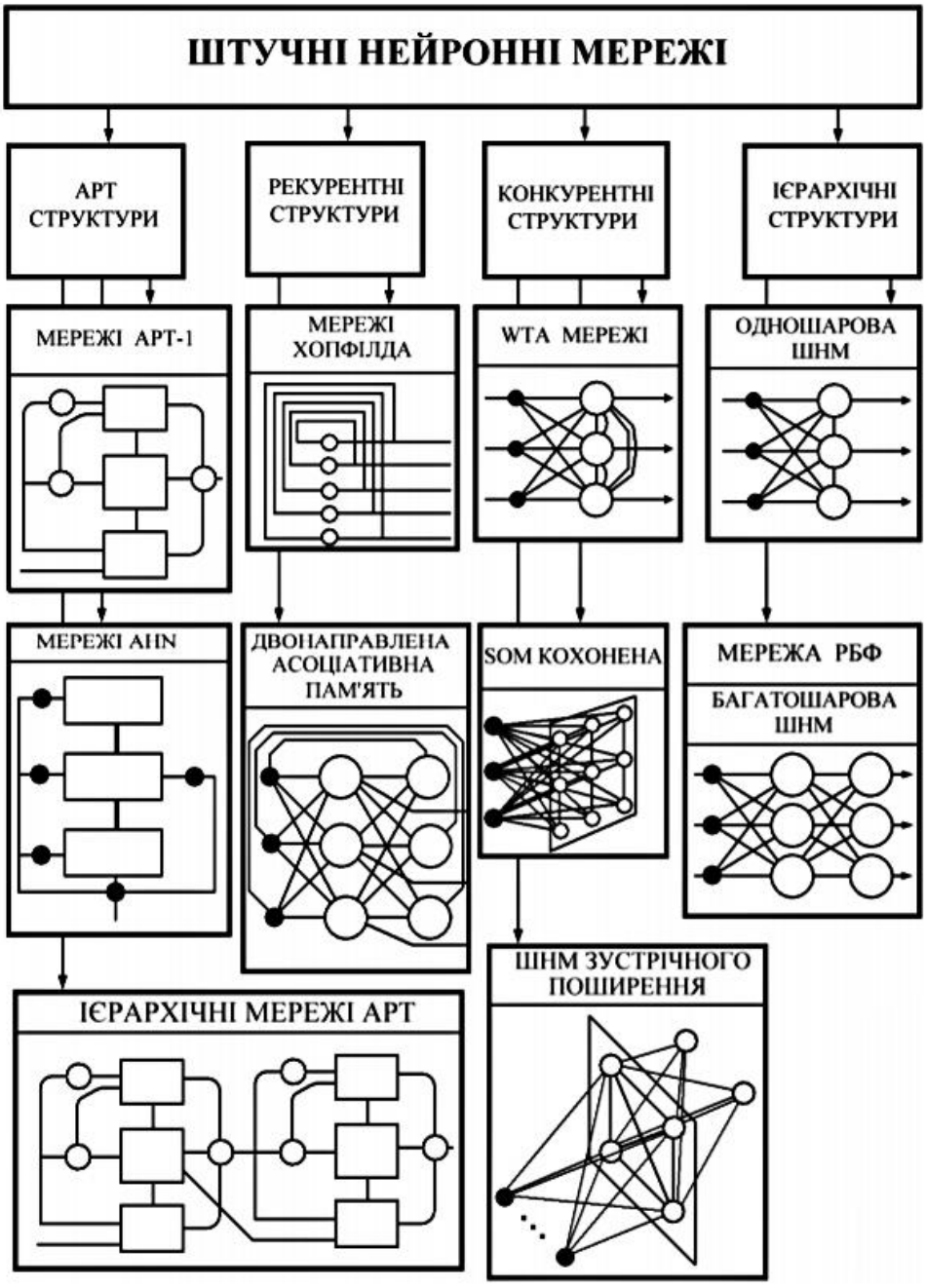

Рис. 1.10. – Нейронні структури [7, с.38]

Нейронні мережі покладаються на дані навчання, щоб навчитися та підвищувати їх точність з часом. Однак, як тільки ці алгоритми навчання налаштовані на точність, вони є потужними інструментами в галузі інформатики та штучного інтелекту, що дозволяє класифікувати та кластеризувати дані з великою швидкістю. Завдання з розпізнавання мови чи розпізнавання зображень можуть зайняти хвилини проти годин у порівнянні з ручним визначенням людськими експертами. Однією з найбільш відомих прикладів застосування нейронних мереж є алгоритм пошуку інформації у Google.

На зображенні нижче (рис. 1.11.) зображено типову структуру штучної нейронної мережі. Існують мережі, що мають лише один шар чи елемент, проте більшість реалізацій використовують мережі з мінімум трьома шарами – вхідний, прихований і вихідний [9, с. 49].

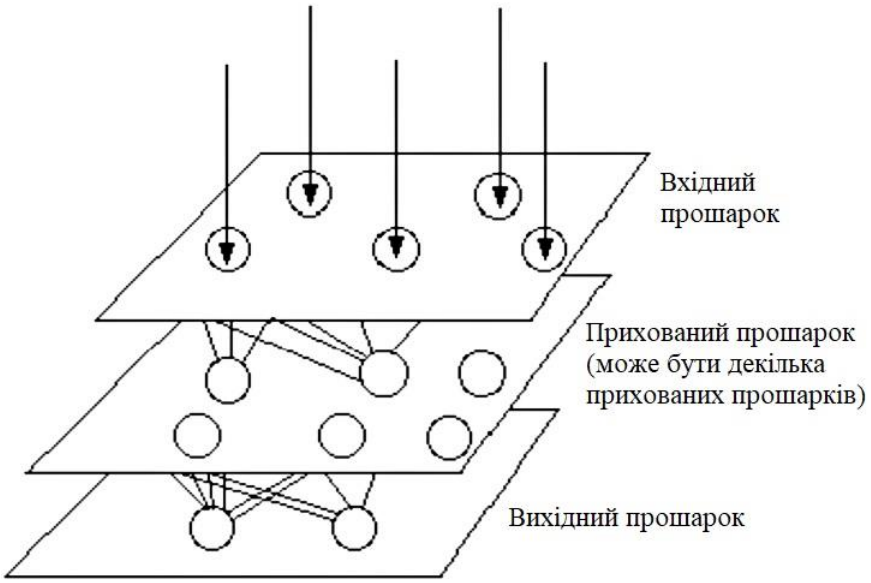

Рис. 1.11. – Діаграма простої нейронної мережі

Вихідний шар пересилає інформацію в зовнішнє середовище, до вторинного комп'ютерного процесу, або до іншого пристрою. Між цими двома шарами може бути кілька прихованих шарів, що містять багато по різному зв'язаних нейронів. Входи і виходи кожного зі схованих нейронів з'єднані з іншими нейронами. Напрямок зв'язку від одного нейрона до іншого є важливим аспектом нейронних мереж. У більшості мереж кожен нейрон прихованого шару одержує

сигнали від усіх нейронів попереднього шару і звичайно від нейронів вхідного шару. Після виконання операцій над сигналами, нейрон передає свій вихід усім нейронам наступних шарів, забезпечуючи передачу сигналу вперед (feedforward) на вихід. При зворотному зв'язку, вихід нейронів шару направляється до нейронів попереднього шару (рис. 1.12).

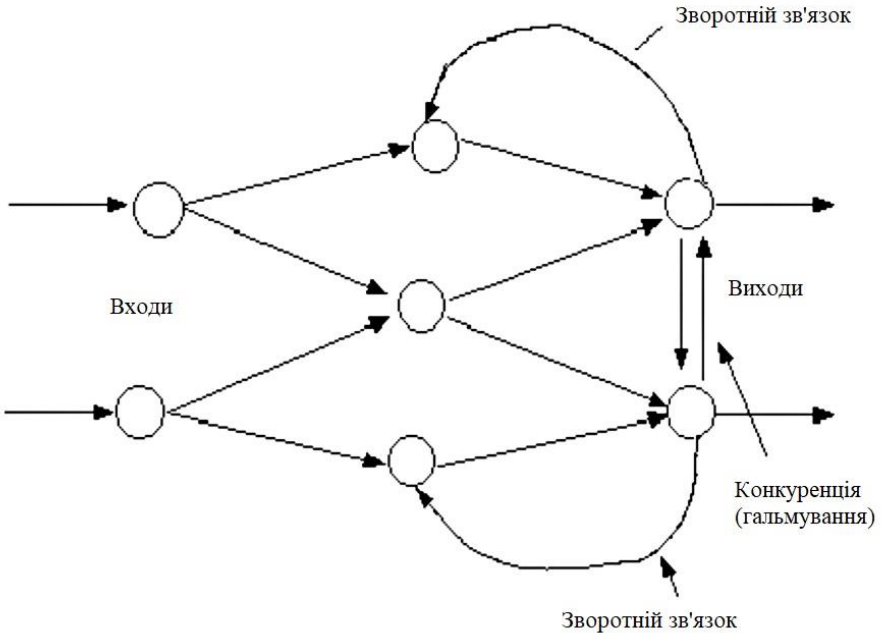

Рис. 1.12 – Повернення нейронів до попереднього шару

З’єднання між нейронами має великий вплив на роботу мережі. Більшість пакетів програмних реалізацій нейронних мереж дозволяють користувачеві додавати, віднімати і керувати з'єднаннями як завгодно. Постійно кореговані параметри зв'язку можна зробити як посилюючими так і гальмуючими [10, с. 58].

Поглиблене навчання - це назва, яка використовується для «нейронних мереж із накопиченням», тобто для мереж, які складаються з декількох шарів.

Шари нейронної мережі складаються з нейронів (вузлів). Вузол - це просто місце, де відбувається обчислення, вільно намальоване на нейроні людського мозку, який спрацьовує, коли зустрічає достатню кількість подразників. Вузол поєднує вхідні дані із набором вагових коефіцієнтів, які або підсилюють, або гасять цей вхідний сигнал, тим самим надаючи більшого або меншого значення вхідним даним щодо задачі, яку намагається вирішити алгоритм; наприклад, який вхід є

найбільш корисним - класифікація даних без помилок? Вхідні дані перемножуються на ваги і підсумовуються, а потім отримана сума передається через так звану функцію активації вузла, щоб визначити, чи повинен і в якій мірі цей сигнал просуватися далі по мережі, щоб вплинути на кінцевий результат, скажімо, на акт класифікації. Якщо сигнали проходять, нейрон «активується».

Нейрони проводять математичні розрахунки, щоб з'ясувати, чи достатньо інформації для передачі інформації наступному нейрону. Простіше кажучи, вони зчитують всі дані і з'ясовують, де існують найсильніші зв'язки. У найпростішому типі мережі отримані вхідні дані додаються, і якщо сума перевищує певне порогове значення, нейрон «спрацьовує» і активує нейрони, з якими він підключений. У складнішому випадку розраховується сума зважених сигналів (сигналів, помножених на ваги).

Зі збільшенням кількості прихованих шарів у нейронній мережі утворюються глибокі нейронні мережі. Архітектури глибокого навчання виводять прості нейронні мережі на новий рівень. Використовуючи ці шари, вчені можуть створити власні мережі глибокого навчання, які можуть навчити комп'ютер точно наслідувати людські завдання, такі як розпізнавання мови, ідентифікація зображень або прогнозування. Не менш важливо, що комп'ютер може вчитися самостійно, розпізнаючи закономірності в багатьох шарах обробки.

Дані надходять у нейронну мережу через вхідний рівень, який передає дані прихованим шарам. Обробка відбувається в прихованих шарах через систему зважених з'єднань. Потім вузли в прихованому шарі поєднують дані з вхідного шару з набором коефіцієнтів і присвоюють відповідним вагам вхідні дані. Потім ці вхідні вагові продукти підсумовуються. Сума передається через функцію активації вузла, яка визначає ступінь подальшого просування сигналу через мережу, щоб впливати на кінцевий вихід. Нарешті, приховані шари посилаються на вихідний рівень - там, де отримуються вихідні дані.

Нейронні мережі також ідеально підходять, щоб допомогти людям вирішувати складні проблеми в реальних ситуаціях. Вони можуть вивчати та моделювати зв'язки між вхідними та вихідними даними, які є нелінійними та складними;

робити узагальнення та висновки; розкрити приховані стосунки, закономірності та передбачення; і моделювати дуже нестабільні дані (такі як дані фінансових часових рядів) та відхилення, необхідні для прогнозування рідкісних подій (наприклад, виявлення шахрайства) [11, с. 25].

Як результат, нейронні мережі можуть покращити процеси прийняття рішень у таких сферах, як:

– Виявлення шахрайства з кредитною карткою;

– Оптимізація логістики транспортних мереж;

– Розпізнавання символів та голосу;

– Медицина та діагностика хвороби;

– Цільовий маркетинг;

– Фінансові прогнози;

– Робото технічні системи управління;

– Прогнозування електричного навантаження та споживання енергії;

– Контроль процесу та якості;

– Ідентифікація хімічної сполуки;

– Оцінка екосистеми;

– Розпізнавання обличчя та будь-яких об’єктів.

Отже, розуміння майбутнього нейронних мереж та їх застосування допоможуть дослідникам зрозуміти їх важливість. Дослідження штучного інтелекту спрямовані на розвиток штучного життя, яке неможливо створити без штучного мозкоподібного процесора, що не входить у його поведінку та фізичну структуру.

Намагаючись вижити та отримати успіх у висококонкурентному світі, ми можемо багато виграти від використання нейронних мереж. Їх здатність вчитися на кращих прикладах робить їх потужними та гнучкими. Більше того, не потрібно розробляти алгоритм для виконання певного завдання. Для цього не потрібні внутрішні механізми. Штучні нейронні мережі добре підходять для систем

реального часу, оскільки швидко реагують на найкращі обчислювальні періоди завдяки своїй паралельній архітектурі.

Нейронні мережі також вносять вклад в інші галузі досліджень, такі як психологія та неврологія. У неврології їх використовують для дослідження внутрішніх механізмів мозку та моделювання частин живих організмів. Найбільш захоплюючим аспектом нейронних мереж є те, що існує ймовірність виникнення «свідомих» мереж.

Висновки по розділу 1

У першому розділі ми розглянули теоретичні основи сучасних теоретичних технологій. Введено та проаналізовано поняття інтелектуальних технологій. Розглянуті основні методи та моделі інтелектуальних технологій. Проаналізовані методи та принципи побудови інформаційних систем - методи "знизу-вгору", "зверху-вниз", принципи "дуалізму", багатокомпонентний. Розглянута сутність машинного навчання та основні методи його реалізації – метод регресії, методи класифікації та кластеризації, метод зменшення розмірності, нейронні мережі. Проаналізовано поняття штучної нейронної мережі, її структура, принцип її дії та сфери застосування ШНМ.

РОЗДІЛ 2. РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ З ВИКОРИСТАННЯМ PYTHON

2.1. Принципи за якими навчаються штучні нейронні мережі

Штучні нейронні мережі є перспективною та практичною формою реалізації методів машинного інтелекту. Розуміння того, як працюють нейронні мережі є досить складним. Навіть дослідницькі групи у великих компаніях, таких як Google і Facebook, намагаються зрозуміти, як взаємодіють рівні нейронних мереж і як алгоритми "навчаються" або покращують свою ефективність у виконанні завдань з часом.

Нейронні мережі складаються з шарів обчислювальних одиниць (нейронів), що мають зв’язки між нейронами в різних шарах. Ці мережі трансформують дані - як-от пікселі на зображенні або слова в документі - до тих пір, поки не зможуть класифікувати їх як вихідні дані, наприклад, присвоєння імені об’єкту на зображенні або позначення неструктурованих текстових даних.

Кожен нейрон у мережі перетворює дані, використовуючи ряд обчислень: нейрон множить початкове значення на деяку вагу, підсумовує результати з іншими значеннями, що надходять у той самий нейрон, регулює отримане число за зміщенням нейрона, а потім нормалізує вихід з функція активації. Упередження - це специфічне для нейрона число, яке регулює значення нейрона, як тільки всі зв’язки обробляються, а функція активації забезпечує, що передані значення лежать у настроюваному очікуваному діапазоні. Цей процес повторюється до тих пір, поки кінцевий вихідний рівень не зможе надати оцінки або передбачення, пов'язані з завданням класифікації, наприклад, ймовірність того, що собака знаходиться на зображенні.

При контрольованому навчанні ми маємо екземпляр даних *i*, що включає вектор атрибутів X_i та цільовий вектор Y_i. Ми обробляємо X_i з мережею, щоб отримати вихід y_i, який має той же вигляд, що і цільовий вектор Y_i. Параметри мережі w модифіковані для оптимізації збігу між виходами та цілями, як правило, шляхом мінімізації загальної квадратичної помилки:

$$E = \frac{1}{2}\sum_i (y_i - Y_i)^2 . \quad (2.1)$$

Може здатися більш природним використання відсотка похибки помилкової класифікації в задачах класифікації, але сумарна квадратична помилка має корисні властивості неперервності та диференційованості.

Активація нейрона Мак-Калок-Піттса була узагальнена до формули:

$$y_i = f_i\left(\sum_i w_{ji} X_i\right). \quad (2.2)$$

Функцією активації f_i може бути будь-яка нелінійна функція але, найчастіше, використовують функції експоненційного типу. Вузли мережі розділяють на вхідний шар I і вихідний шар O. Пороговий рівень або зміщення рівняння було включено до суми, припускаючи додатковий компонент у векторі X, значення якого зафіксовано на 1 [12, с.86].

У методі радіальних базових функцій [24], мережа базових функцій складається з шару одиниць, що виконують лінійні або нелінійні функції атрибутів, а потім слідує шар зважених з'єднань з вузлами, виходи яких мають таку ж форму, що і цільові вектори. Таким чином, мережа має структуру з одним прихованим шаром, за винятком того, що кожен вузол прихованого шару обчислює довільну функцію входів, а передавальна функція кожного вихідного вузла є тривіальною функцією ідентичності. Замість “синоптичної сили” прихований шар має параметри, які відповідають будь-яким використовуваним функціям, наприклад, Гауссову ширину. Ця мережа пропонує ряд переваг перед багатошаровим персептроном за певних умов, хоча ці дві моделі обчислювально еквівалентні.

За останні роки інтерес до навчання без контролю значно зріс. Він пропонує можливість дослідити структуру даних без настанов у формі інформації про клас і часто може виявити особливості, на які раніше не очікували або не знали. Сюди може входити розподіл даних, які раніше вважалися єдиним кластером, на ряд менших груп, кожна з яких має окремі властивості, які можна ідентифікувати. Знайдені кластери пропонують модель даних із точки зору кластерних

центрів, розмірів та форм, які часто можуть бути описано з використанням меншої кількості інформації та з меншою кількістю параметрів, ніж було потрібно для зберігання всього набору навчальних даних. Це має очевидні переваги для зберігання, кодування та передачі стохастично сформованих даних; якщо його розподіл у просторі атрибутів відомий, еквівалентні дані можна генерувати з моделі, коли це потрібно.

Як відомо, неконтрольовані методи навчання, такі як машини Больцмана є обчислювально дорогі, ітеративні алгоритми кластеризації, такі як мережі Кохонена, кластеризація K-методом та моделі Гауссової суміші пропонують рівносильну потужність моделювання із значно коротшим часом навчання. Хоча мітки класів не використовуються для обмеження структури, вивченої моделями, звільнення від цього обмеження в поєднанні з ретельною ініціалізацією моделей з використанням будь-якої попередньої інформації, доступної про дані, може дати дуже швидкі та ефективні моделі. Ці моделі, відомі в сукупності як векторні квантування, можуть бути використані як нелінійна частина моделей навчання під контролем. У цьому випадку лінійна частина додається і навчається пізніше для реалізації відображення від активації в різних частинах моделі до ймовірних класів подій, що генерують дані.

Мережевий алгоритм Кохонена (Kohonen, 1984) [23] також забезпечує класифікацію у вигляді діаграми Вороного вхідного простору на патчі з відповідними кодовими векторами. Він має ту додаткову особливість, що центри розташовані в структурі з низькою розмірністю (зазвичай це рядок або квадратна сітка), такою що сусідні точки в топологічній структурі (рядок або сітка) відображаються на сусідні точки в просторі атрибутів. Вважається, що структури такого роду зустрічаються в природі, наприклад, при картографуванні від вуха до слухової кори або ж при побудові ретинотопної карті від сітківки до зорової кори.

При навчанні з учителем нейронні мережі, як правило, виконують контрольовані навчальні завдання, будуючи знання з наборів даних, де правильна відповідь надається заздалегідь. Потім мережі навчаються, налаштовуючись на те,

щоб самостійно знаходити правильну відповідь, підвищуючи точність своїх прогнозів.

Навчання штучної нейронної мережі полягає у правильному підборі ваги зв’язків між нейронами мережі. Для цього мережа порівнює вихідні результати із наданою правильною відповіддю. Метод, який називається функцією витрат, використовується для модифікації початкових результатів на основі ступеня, на який вони відрізнялися від цільових значень. Необхідно виконати зворотній хід по всіх нейронах мережі від її кінця до початку, щоб відрегулювати зміщення та ваги зв’язків.

Цей метод налаштування мережі називається «метод зворотного поширення помилки» (англ. *Backpropagation*) – і він є ключем до того, як нейронна мережа вивчає та моделює певну ситуацію (рис. 2.1.).

Спочатку алгоритм зворотного розповсюдження був запроваджений в 1970-х роках, але його значення не було повністю оцінено до публікації 1986 року Девіда Румельхарта, Джеффрі Хінтона, та Рональда Вільямса [25]. Ця стаття описує кілька нейронних мереж, де зворотне поширення працює набагато швидше, ніж попередні підходи до навчання, що дозволяє використовувати нейронні мережі для вирішування проблем, які раніше були нерозв'язними [13, с. 39].

$$w'_{14} = w_{14} + \eta\delta_4 \frac{df_4(e)}{de} y_1$$

$$w'_{24} = w_{24} + \eta\delta_4 \frac{df_4(e)}{de} y_2$$

$$w'_{34} = w_{34} + \eta\delta_4 \frac{df_4(e)}{de} y_3$$

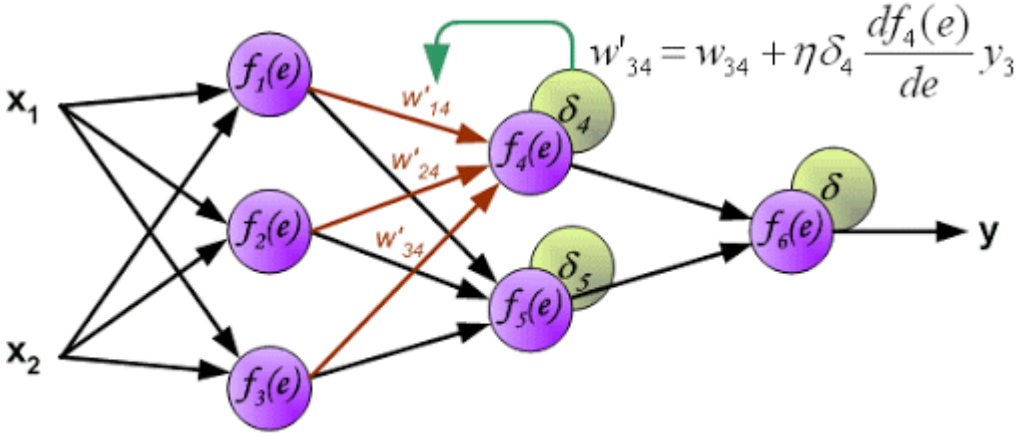

Рис. 2.1 – Метод зворотного поширення помилки

Зворотне поширення помилки базується на чотирьох основних рівняннях [13, с. 44]. Разом, ці рівняння дають нам спосіб обчислення як похибки δ^l.

Рівняння похибки вихідного рівня, δ^L:

$$\delta_j^L = \frac{\partial C}{\partial a_j^L}\sigma'\left(z_j^L\right) \tag{2.3}$$

Перший множник правої частини, $\partial C / \partial a_j^L$, просто вимірює, наскільки швидко змінюється сумарна квадратична помилка при зміні активації j-го виходу. Якщо, наприклад, C не сильно залежить від конкретного вихідного нейрона, j, тоді δ_j^L буде малим. Другий множник правої частини, $\sigma'(z_j^L)$, вимірює, наскільки швидко змінюється функція активації σ при зміні аргумента z_j^L.

Рівняння для похибки δ^l з точки зору похибки в наступному шарі, δ^{l+1}:

$$\delta^l = \left((w^{l+1})^T \delta^{l+1}\right) \odot \sigma'(z^l) \tag{2.4}$$

Де (w^{l+1}) T - транспонування вагової матриці w^{l+1} для (l + 1) - го шару. Це рівняння виглядає складним, але кожен елемент має хорошу інтерпретацію. Припустимо, ми знаємо помилку δ^{l+1} на (l + 1) - му шарі. Коли ми застосовуємо матрицю ваги транспонування $(w^{l+1})^T$, ми можемо інтуїтивно думати про це як про переміщення помилки назад через мережу, даючи нам якусь міру похибки на виході l-го шару. Потім беремо добуток Адамара $\odot\sigma'(z^l)$. Це переміщує помилку назад через функцію активації шару l, даючи нам помилку δ^l у зваженому вході до шару l.

Рівняння швидкості зміни сумарної похибки щодо будь-якого упередження в мережі:

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l \tag{2.5}$$

Тобто похибка δ_j^l точно дорівнює швидкості зміни $\partial C / \partial b_j^l$. Ми можемо переписати скорочено як $\partial C/\partial b = \delta$, де розуміється, що δ оцінюється в тому ж нейроні, що і зміщення b.

Рівняння швидкості зміни похибки відносно будь-якої ваги в мережі:

$$\frac{\partial C}{\partial w_{jk}^{l}} = a_k^{l-1}\delta_j^l \tag{2.6}$$

Це рівняння показує, як обчислити часткові похідні $\partial C / \partial w_{jk}^{l}$ через величини δ^l та a^{l-1}.

Маючи на увазі чотири рівняння, можна пояснити, чому такі модифікації відбуваються та який вплив вони можуть мати.

Використання цих рівнянь покращує ефективність роботи штучної нейронної мережі після застосування цих правил до мережі. Таким чином, правила навчання оновлюють ваги та рівні упередженості мережі. Процес навчання мережі – це ітераційний процес. Після кожної зміни ваг відбувається пряме проходження вхідних сигналів та розраховується нове значення похибки. Після цього знову відбувається зміна ваг з використанням правил. Це допомагає нейронній мережі вчитися на існуючих умовах та покращувати свою роботу.

Різні правила навчання нейронних мереж:

– Правило навчання Гебба (англ. *Hebbian learning rule*) – воно визначає, як змінювати ваги вузлів мережі.

– Правило навчання перцептрону (англ. *Perceptron learning rule*) – мережа починає своє навчання, присвоюючи випадкове значення кожній вазі.

– Правило «дельта»-навчання (англ. *Delta learning rule*) – модифікація симпатричної ваги вузла дорівнює множенню похибки та вхідних даних.

– Правило кореляційного навчання (англ. *Correlation learning rule*) – правило кореляційного навчання є контрольованим навчанням.

– Правило навчання поза зіркою (англ. *Outstar learning rule*) – ми можемо використовувати його, коли ми припускаємо, що вузли або нейрони в мережі розташовані в шар.

Правило Гебба було першим правилом навчання. У 1949 році Дональд Гебб розробив його як алгоритм навчання некерованої нейронної мережі. Правило навчання Гебба передбачає, що якщо два сусідні нейрони активуються та деактивуються одночасно, тоді вага зв’язку, що з'єднує ці нейрони, повинна

збільшитися. Для нейронів, що діють у протилежній фазі, вага між ними повинна зменшуватися. Якщо кореляція сигналу відсутня, вага не повинна змінюватися.

Коли входи обох вузлів є або позитивними, або негативними, тоді між вузлами існує сильна позитивна вага. Якщо вхідні дані вузла є позитивними та негативними для інших, між вузлами існує сильна негативна вага.

Математична формула правила навчання Гебба в штучній нейронній мережі:

$$W_{ij} = x_i * x_j \tag{2.7}$$

Кожне з'єднання в нейронній мережі має відповідну вагу, яка змінюється в процесі навчання. Згідно з цим прикладом контрольованого навчання, мережа починає своє навчання, присвоюючи випадкове значення кожній вазі.

Якщо обчислити вихідне значення на основі набору записів, для яких ми можемо знати очікуване вихідне значення, отримаємо об’єкт, який називається навчальним зразком.

Потім мережа порівнює розраховане вихідне значення з очікуваним значенням. Далі обчислює функцію помилки $\in$, яка може бути сумою квадратів помилок, що трапляються для кожного зразка навчальної вибірки.

Математична формула правила навчання перцептрону в штучній нейронній мережі має наступний вигляд:

$$\sum_i \sum_j (E_{ij} - O_{ij})^2 \tag{2.8}$$

Оскільки призначення ваги вузлів залежить від користувачів, це приклад контрольованого навчання.

Правило «дельта», розроблене Відроу та Гоффом [26, с. 2-3], є одним із найпоширеніших правил навчання. Воно відноситься до методів контрольованого навчання. Це правило стверджує, що модифікація симпатичної ваги вузла дорівнює множенню похибки та вхідних даних.

Математична формула правила навчання дельта в штучній нейронній мережі:

$$\triangle w = \eta(t - y)x_i \quad (2.9)$$

Якщо різниця дорівнює нулю, навчання не відбувається; в іншому випадку потрібно так регулювати ваги зв’язків, щоб зменшити цю різницю.

Зміна ваги від ui до uj’ становить: dwij = r * ai * ej. Тут r - швидкість навчання, ai являє собою активацію ui, а ej - різниця між очікуваним результатом та фактичним виходом uj.

Ми можемо використовувати дельта-правило навчання як з одним вихідним блоком, так і з кількома вихідними блоками.

Застосовуючи правило «дельта», припустимо, що похибка може бути безпосередньо виміряна.

Метою застосування правила «дельта» є зменшення різниці між фактичним та очікуваним результатом, що є помилкою.

Правило кореляційного навчання базується на подібному принципі, як правило навчання Гебба. Передбачається, що ваги між нейронами, що відповідають, повинні бути більш позитивними, а ваги між нейронами з протилежною реакцією повинні бути більш негативними.

У математичній формі правило кореляційного навчання таке:

$$\triangle w_{ij} = \eta x_i d_j \quad (2.10)$$

Де d_j - бажане значення вихідного сигналу. Цей навчальний алгоритм зазвичай починається з ініціалізації ваг до нуля. Оскільки присвоєно бажану вагу користувачами, правило кореляційного навчання є прикладом контрольованого навчання.

Правило навчання *Outstar* використовується, коли припускається, що вузли або нейрони в мережі розташовані в шар. Тут ваги, підключені до певного вузла, повинні дорівнювати бажаним виходам для нейронів, підключених через ці ваги. Правило вихідного запуску виробляє бажану відповідь t для шару з n вузлів.

Математична формула для правила *Outstar* :

$$W_{jk} = \begin{cases} \eta(y_k - w_{jk}) \text{якщо вузол } j \text{ виграє змагання} \\ 0 \text{ якщо вузол } j \text{ програє змагання} \end{cases} \quad (2.11)$$

Ця процедура виконується під наглядом, оскільки необхідні результати повинні бути відомі.

На закінчення огляду методів та правил навчання в нейронній мережі можна відзначити, що найбільш перспективною особливістю штучної нейронної мережі є її здатність вчитися. Процес навчання мозку змінює його нервову структуру. Збільшення або зменшення сили синаптичних зв'язків залежно від їх активності, покращує ефективність роботи штучної нейронної мережі. Більш відповідна інформація буде мати сильніший синаптичний зв'язок.

2.2. Використання штучних нейронних мереж у задачах класифікації та прогнозування

В даний час нейронні мережі використовуються для вирішення цілого ряду завдань, одним з яких є завдання прогнозування.

Прогнозування - це передбачення майбутніх подій. Нехай задані n дискретних відліків {y (t1), y (t2) ..., y (tn)} в послідовні моменти часу t1, t2, ..., tn. Тоді задача прогнозування полягає в передбаченні значення y (t_n + 1) в деякий майбутній момент часу t_n + 1 [14].

Метою прогнозування є зменшення ризику при прийнятті рішень. Прогноз зазвичай виходить помилковим, але помилка залежить від використовуваної прогнозуючої системи. Надаючи прогнозу більше ресурсів, можна збільшити точність прогнозу і зменшити збитки, пов'язані з невизначеністю при прийнятті рішень. Типовими прикладами техніки прогнозу є передбачення цін на фондовій біржі, прогноз погоди, прогноз споживання електроенергії, прогноз відмов технічних систем та ін.

Класифікація – одна із задач, яка часто зустрічається в людській діяльності, пов'язаній із прийняттям рішень. Проблема класифікації виникає, коли об'єкт потрібно присвоїти заздалегідь визначеній групі або класу на основі кількості

спостережуваних атрибутів, пов'язаних з цим об'єктом. Багато проблем у бізнесі, науці, промисловості та медицині можна розглядати як проблеми класифікації. Приклади включають передбачення банкрутства, кредитний баланс, медичну діагностику, контроль якості, рукописне розпізнавання символів та розпізнавання мови.

Традиційні статистичні процедури класифікації, такі як дискримінантний аналіз, побудовані на Байєсівській теорії рішень. Теорема Байєса в загальному виді дозволяє визначити апостеріорну імовірність, залежить від апріорної імовірності і від інформації, що надійшла:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_{j=1}^{N} P(B|A_i)P(A_i)}, i = 1,2 \dots, N \qquad (2.12)$$

Тут $\{A_i\}$ — послідовність непересічних подій (гіпотез), B - довільна подія (симптом), для якої P(B)> O, N — число можливих гіпотез, $P(A_i|B)$ – апостеріорна імовірність гіпотези A,- за наявності симптому B; $P(B|A_i)$ — імовірність появи симптому B за наявності гіпотези A_i; $P(A_i)$ - апріорна імовірність гіпотези A_i.

Імовірності $P(B|A_i)$ та $P(A_i)$, i = 1, …, N, задаються експертом і не змінюються в процесі вирішення задачі [15, с. 36].

У цих процедурах для обчислення задньої ймовірності, за якою приймається рішення про класифікацію, повинна бути прийнята основна модель ймовірності. Одним із основних обмежень статистичних моделей є те, що вони добре працюють лише тоді, коли виконуються основні припущення.

Ефективність цих методів значною мірою залежить від різних припущень або умов, за яких розробляються моделі. Користувачі повинні добре знати як властивості даних, так і можливості моделей, перш ніж моделі зможуть бути успішно застосовані. Нейронні мережі стали важливим інструментом для класифікації. Нещодавня велика наукова діяльність з нейронної класифікації встановила, що нейронні мережі пропонують альтернативу різним класичним методам класифікації. Перевага нейронних мереж полягає у наступних теоретичних аспектах. По-перше, нейронні мережі є методами само адаптації, керованими даними,

оскільки вони можуть коригувати їх до даних без будь-якої чіткої специфікації функціональної форми розподілу для базової моделі. По-друге, вони є універсальними функціональними апроксиматорами, оскільки нейронні мережі можуть наближати будь-яку функцію з довільною точністю.

Оскільки будь-яка процедура класифікації прагне функціонального взаємозв'язку між членством у групі та атрибутами об'єкта, точне визначення цієї основної функції, безсумнівно, важливе. По-третє, нейронні мережі є нелінійними моделями, що робить їх гнучкими при моделюванні складних взаємозв’язків у реальному світі. Нарешті, нейронні мережі здатні оцінити задні ймовірності, що забезпечує основу для встановлення правила класифікації та проведення статистичного аналізу [16, с. 451].

Отже, класифікація та прогнозування є найбільш дослідженими темами нейронних мереж. Використання нейронних мереж як способу моделювання економічних процесів є актуальним в розрізі проблеми, коли обробка та управління великою кількістю даних людським інтелектом є малоефективною, а використання традиційних обчислень стає трудомістким процесом.

Застосування нейронних мереж є досить потужним методом прогнозування, який дозволяє відтворювати досить складні залежності. Воно вирішує багато актуальних проблем у сфері економіки та фінансів.

Однією з досить важливих областей застосування нейронних мереж у фінансовій сфері є прогнозування на фондовому ринку. Застосування нейронних мереж є досить потужним методом прогнозування, який дозволяє відтворювати досить складні залежності.

Різноманітність програмного забезпечення для реалізації штучних нейронних мереж дає спеціалістам з найрізноманітніших сфер широкий спектр вибору, який буде здійснюватися у залежності від поставлених завдань, а також області дослідження.

2.3. Побудова штучних нейронних мереж з використанням Python

Наша мета полягає в тому, щоб побудувати нейронну мережу, яка зможе виконувати деякі функції. Для виконання цієї задачі буде використано мову програмування Python.

Мови програмування дуже схожі на людські мови тим, що вони мають правила синтаксису. Ці правила диктують відповідні розташування слів, пунктуації тощо. Наприклад, у багатьох мовах програмування команди або оператори закінчуються крапкою з комою (як і більшість речень закінчуються крапкою). Це приклад “пунктуації” мови програмування. В англійській мові абзаци розділяються рядками, в мовах програмування блоки коду відокремлюються фігурними дужками.

Блок коду – це розділ коду, який був логічно згрупований. Багато мов дозволяють визначити блок, включаючи згрупований код навколо відкриття та закриття фігурних дужок. Блоки можуть бути вкладені один в одного для формування підблоків [17, с. 13].

Python є високорівневою мовою програмування, яка для виведення результатів використовує інтерпретатор. Python містить велику стандартну бібліотеку модулів протестованого коду, які можуть бути включені у власні програми.

Мова Python, розроблена Гвідо Ван Россумом в кінці вісімдесятих - початку дев'яностих років в Національному науково-дослідному інституті математики і комп'ютерних наук в Нідерландах, є похідною від багатьох інших мов, в тому числі C, C ++ і командної оболонки Unix. Сьогодні Python підтримується командою розробників ядра в інституті, хоча Гвідо Ван Россум як і раніше відіграє важливу роль у визначенні напрямку розвитку мови [18, с. 3-4].

Оскільки Python орієнтований на читаність коду, в ньому часто використовуються ключові слова англійською мовою там, де інші мови програмування зазвичай використовують розділові знаки. Особлива його відмінність полягає в тому, що для угруповання інструкцій в блоці коду Python використовує відступи, а не ключові слова або коми. У мові Pascal. наприклад, початок блоків позначається

ключовим словом *begin* і закінчується ключовим словом *end*, той час як програмісти на C використовують фігурні дужки, для позначення блоків коду.

Основними характерними рисами мови Python є [19, с. 11]:

– Python безкоштовний – це вільне програмне забезпечення з відкритим вихідним кодом;

– Python легкий у вивченні – має простий синтаксис;

– Python дозволяє створювати код, що легко читається, не перегружений;

– Python легкий в обслуговуванні – має модульну структуру;

– Python має велику стандартну бібліотеку, що легко інтегрується в програми;

– Python можна запустити на великій кількості платформ і всюди буде мати один і той же інтерфейс;

– Python підтримує як об’єктний так і процедурний методи програмування.

Проекти штучного інтелекту (Artificial Intelligence - AI) відрізняються від традиційних проектів програмного забезпечення. Відмінності полягають у наборі технологій, навичках, необхідних для проекту на основі ШІ, та необхідності глибоких досліджень. Для реалізації прагнень до штучного інтелекту слід використовувати мову програмування, яка є стабільною, гнучкою та має доступні інструменти. Python пропонує все це, саме тому сьогодні спостерігається безліч проектів AI для Python [20, с. 7].

Від розробки до розгортання та обслуговування Python допомагає розробникам бути продуктивними та впевненими в тому, яке програмне забезпечення вони створюють. Переваги, які роблять Python найкращим чином придатним для машинного навчання та проектів на основі ШІ, включають простоту та послідовність, доступ до чудових бібліотек та фреймворків для ШІ та машинного навчання (ML), гнучкість, незалежність від платформи та широке співтовариство. Це додає загальної популярності мові.

Python пропонує стислий і читабельний код. Хоча складні алгоритми та універсальні робочі процеси стоять за машинним навчанням та ШІ, простота Python дозволяє розробникам писати надійні системи. Розробники докладають усіх своїх зусиль для вирішення проблеми, замість того щоб зосередитись на технічних нюансах мови.

Крім того, Python приваблює багатьох розробників, оскільки його легко вивчити. Код Python зрозумілий людям, що полегшує побудову моделей машинного навчання.

Сьогодні багато програмістів обирають Python для створення програм із стислою, чистою та читабельною базою коду. Вони навіть можуть прискорити розробку спеціальних програм, використовуючи низку інтегрованих середовищ розробки (IDE) для Python. PyCharm - одна з найбільш широко використовуваних середовищ розробки для мови програмування Python (рис. 2.2.). В даний час IDE Python використовується великими підприємствами, такими як Twitter, Pinterest, HP, Symantec та Groupon.

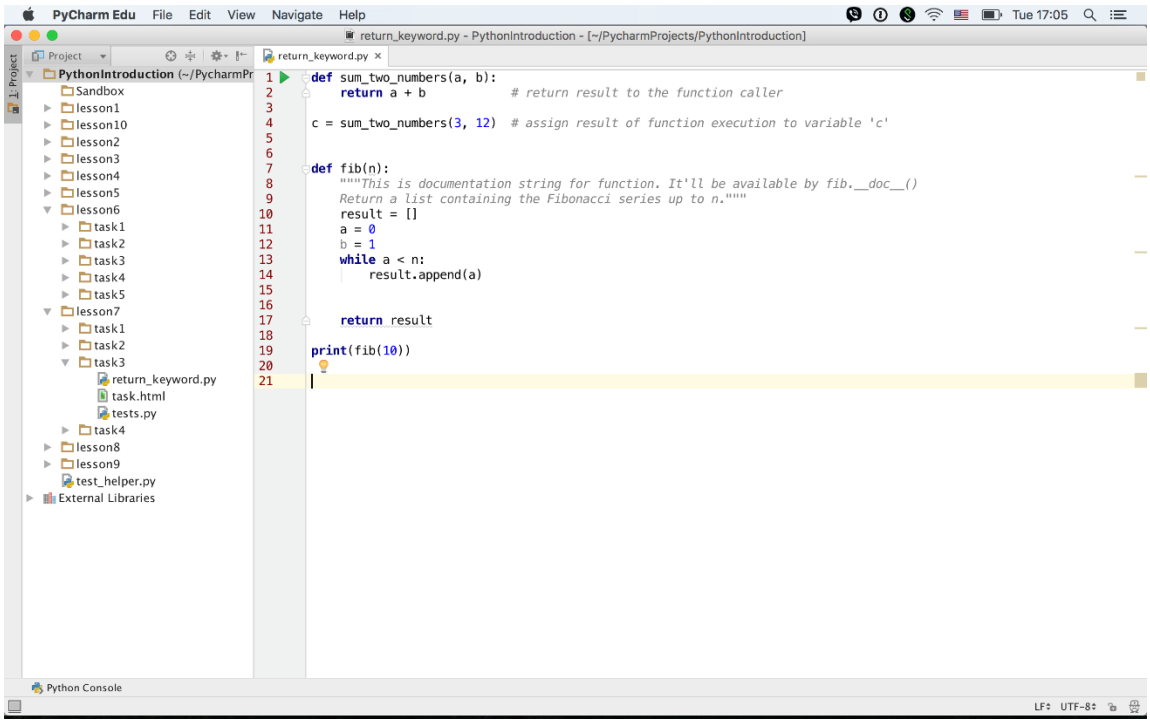

Рисунок 2.2. – Інтерфейс Pycharm

Багато програмістів кажуть, що Python є більш інтуїтивно зрозумілим, ніж інші мови програмування. Інші вказують на безліч фреймворків, бібліотек та розширень, які спрощують реалізацію різних функціональних можливостей. Загальновизнано, що Python підходить для спільної реалізації, коли залучено кілька розробників. Оскільки Python є мовою загального призначення, він може виконувати набір складних завдань машинного навчання та дозволяти швидко створювати прототипи, що дозволяють протестувати ваш продукт для цілей машинного навчання.

Впровадження алгоритмів AI та ML може бути складним і вимагає багато часу. Тому дуже важливо мати добре структуроване та перевірене середовище, щоб розробники могли запропонувати найкращі рішення для кодування.

Щоб скоротити час розробки, програмісти звертаються до ряду фреймворків та бібліотек Python. Бібліотека програмного забезпечення - це заздалегідь написаний код, який розробники використовують для вирішення загальних завдань програмування. Python, з його багатим набором технологій, має великий набір бібліотек для штучного інтелекту та машинного навчання. Ось деякі з них:

– Keras, TensorFlow та Scikit-learn для машинного навчання;

– NumPy для високопродуктивних наукових обчислень та аналізу даних;

– SciPy для вдосконалених обчислень;

– Pandas для загального аналізу даних;

– Seaborn для візуалізації даних.

За допомогою цих рішень можна швидше розвивати свій продукт, використовувати існуючу бібліотеку для реалізації необхідних функцій.

Незалежність від платформи відноситься до мови програмування або фреймворку, що дозволяє розробникам реалізовувати речі на одній машині та використовувати їх на іншій машині без будь-яких (або лише з мінімальними) змін. Одним із ключових факторів популярності Python є те, що це незалежна від платформи мова. Python підтримується багатьма платформами, включаючи Linux,

Windows та macOS. Код Python може бути використаний для створення автономних виконуваних програм для більшості поширених операційних систем, а це означає, що програмне забезпечення Python можна легко розповсюджувати та використовувати в цих операційних системах без інтерпретатора Python.

Більше того, розробники зазвичай використовують послуги Google або Amazon для своїх обчислювальних потреб. Однак часто можна зустріти компанії та науковців, які використовують власні машини з потужними графічними процесорами (GPU) для навчання своїх моделей ML. І той факт, що Python не залежить від платформи, робить це навчання набагато дешевшим та простішим.

Онлайн репозиторії містять понад 140 000 спеціальних програмних пакетів Python. Наукові пакети Python, такі як Numpy, Scipy та Matplotlib, можуть бути встановлені в програмі, що працює на Python. Ці пакети обслуговують машинне навчання та допомагають розробникам виявляти закономірності у великих наборах даних. Python настільки надійний, що Google використовує його для сканування веб-сторінок, Pixar - для створення фільмів, а Spotify - для рекомендування пісень.

Загальновідомий факт, що спільнота Python AI зросла по всьому світу. Є форуми на Python та активний обмін досвідом, пов’язаний з рішеннями машинного навчання. Для будь-якого завдання, яке може виникнути, досить висока ймовірність того, що хтось інший мав справу з тією ж проблемою. Можна отримати поради та рекомендації від розробників.

Штучний інтелект швидко розвивається і його популярність зростає, існує кілька мов, які також використовуються для його створення:

1. ***R***, як правило, застосовується, коли потрібно аналізувати дані та маніпулювати ними для статистичних цілей. У R є такі пакети, як Gmodels, Class, Tm та RODBC, які зазвичай використовуються для побудови проектів машинного навчання. Ці пакети дозволяють розробникам впроваджувати алгоритми машинного навчання без зайвих клопотів і дозволяють швидко реалізовувати бізнес-логіку. У порівнянні з Python, R має репутацію повільного та відстаючого, що стосується великомасштабних продуктів передачі

даних. Для фактичної розробки продукту краще використовувати Python або Java з його гнучкістю.

2. ***Scala*** безцінна, коли йдеться про великі дані. Ця мова пропонує вченим з обробки даних безліч інструментів, таких як Saddle, Scalalab та Breeze. Scala має чудову підтримку одночасності, що допомагає обробляти великі обсяги даних. Оскільки Scala працює на JVM, вона виходить за межі, поруч із Hadoop, розподіленою обробною системою з відкритим кодом, яка управляє обробкою та зберіганням даних для додатків великих даних, що працюють у кластерних системах. Незважаючи на меншу кількість інструментів машинного навчання порівняно з Python та R, Scala також придатна для цього.

3. ***Julia***. Має синтаксис, подібний до Python, і була розроблена для роботи з числовими обчислювальними завданнями. Забезпечує підтримку глибокого навчання через TensorFlow.jl та фреймворк Mocha. Однак мова не підтримується багатьма бібліотеками і досі не має такої сильної спільноти, як Python, оскільки вона відносно нова.

4. ***Java***. Java є об’єктно-орієнтованою, портативною, підтримуваною та прозорою. Підтримується численними бібліотеками, такими як WEKA та Rapidminer. Java широко поширена, коли йдеться про обробку природної мови, алгоритми пошуку та нейронні мережі. Це дозволяє швидко будувати масштабні системи з чудовими характеристиками. Але для статичтичного моделювання та візуалізації Java – остання мова, яку варто використовувати. Хоча є деякі пакети Java, які підтримують статистичне моделювання та візуалізацію, їх недостатньо. Python, навпаки, має передові інструменти, які добре підтримуються спільнотою.

Отже, фільтри спаму, рекомендаційні системи, пошукові системи, особисті помічники та системи виявлення шахрайства - все це стало можливим завдяки штучному інтелекту та машинному навчанню, і, безумовно, буде ще багато речей. Власники продуктів хочуть створювати ефективні програми. Для цього

потрібно придумати алгоритми, які обробляють інформацію розумно, змушуючи програмне забезпечення діяти як людина.

Попри існування ще кількох мов, які можна використовувати для створення штучної нейронної мережі, Python залишається швидшим і кращим засобом для вирішення цієї задачі.

Висновки по розділу 2

У розділі 2 ми розглянули принципи побудови та навчання нейронних мереж. Розглянуто основний метод навчання ШНМ - метод зворотного поширення помилки. Проаналізовано використання штучних нейронних мереж у задачах класифікації та прогнозування. Представлені основні характеристики мови Python. Розглянута методика побудови штучних нейронних мереж з використанням Python. Вказано на переваги використання Python для вирішення задач класифікації та прогнозування.

РОЗДІЛ 3. ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ВИРІШЕННЯ ЗАДАЧ КЛАСИФІКАЦІЇ ТА ПРОГНОЗУВАННЯ

3.1. Прогнозування ціни крипто валют з використанням Python

Крипто валюта подібна до будь-якої іншої валюти. Ви можете купувати товари та послуги з ними або торгувати з ними. Там, де вони відрізняються від традиційних паперових валют, які ми маємо в своїх гаманцях, є відсутність фізичних монет або купюр - гроші абсолютно віртуальні.

Одиниці крипто валюти можна придбати у брокерів або сформувати за допомогою онлайн-процесу, який називається «майнінг», і використовувати для здійснення платежів або зберігання грошей анонімно. Анонімність це дуже великий плюс для цієї крипто валюти, біткойн одночасно прозорий і анонімний. Він заснований на технології Blockchain. Blockchain це - розподілена база даних, що зберігає впорядкований ланцюжок записів [27].

Хоча вони в народі зображені як монети, схожі на фішки казино, фізична монета нічого не коштує без надрукованого коду всередині неї.

Біткойн (BTC), про який ви, можливо, чули, був оригінальною крипто валютою, і він продовжує залишатися найбільш широко визнаним людьми. З огляду на свій успіх у світовому визнанні, біткойн став мірилом для всіх інших альткойнів, або «альтернативних монет», що з’явилися після нього.

Подібно до золота або діамантів, крипто валюта - це просто ще один товар, що підлягає торгівлі. Багато людей, які використовують крипто валюти, віддають перевагу їм традиційним валютам, оскільки вони не контролюються і не регулюються урядами чи банками, а транзакції є анонімними.

Хоча деякі країни, такі як Північна Македонія та Алжир, заборонили використовувати біткойн, інші більш сприйнятливі до його використання. Наприклад, Банк Сінгапуру дав знак, що може замінити золото біткойнами як запас вартості.

Крипто валюти ще не широко використовуються бізнесом чи магазинами. Але постійно розширюваний список компаній вирішує прийняти крипто валюти як законний платіжний засіб. Виробник автомобілів Tesla і компанія, що займається кредитними картками Visa, оголосили в березні 2021 року, що визнають їх способом оплати. У той же час авіакомпанія airBaltic також оголосила, що розширює свій перелік прийнятих крипто валют.

Сервіс онлайн-платежів PayPal також заявив у жовтні минулого року, що дозволить своїм клієнтам купувати та продавати за допомогою біткойнів.

Деякі компанії, такі як Google і Amazon, планують випускати власні крипто валюти для операцій, пов'язаних із товарами та послугами, спеціально наданими ними.Нам відомо, що крипто валюта має дуже великий вплив на багато сфер бізнесу і не тільки, тому спробувати передбачити курс крипто валюту дуже важливо. Для вирішення цієї проблеми було створено нейронну модель яка аналізує попередні дні і навіть місяці зміни курсу крипто валюти, на основі теорії того що ринок циклічний.

Дані було використано з пошукової системи Yahoo за допомогою веб-парсингу (рис. 3.1), значення змінної crypto_currency дорівнює BTC, а значення змінної against_currency дорівнює USD. Всі дані про ціну використовують тільки одну валюту це USD.

```
data = web.DataReader(f'{crypto_currency}-{agaisnt_currency}', 'yahoo', start, end)
```

Рис. 3.1. Парсинг даних про ціну біткоїна

Для навчання моделі було взято дані з початку 2016 року до 2020. Для для тестування відбуваються з початку 2020 до теперішньої дати запуску програми.

Ми використали TensorFlow (відкрита бібліотека для машинного навчання) для навчання моделі, на таблиці наведеній нижче (табл. 3.1).За основну крипто валюту було використано біткоїн (Bitcoin). Було пройдено 25 кроків навчання нейронної мережі.

Таблиця 3.1.

Дані про навчання нейронної мережі при апроксимації динаміки ціни біткоїна

Епохи	Час на навчання	Похибка апроксимація
1	12s 66ms	0.0074
2	4s 68ms	0.0022
3	4s 65ms	0.0019
4	4s 66ms	0.0013
5	4s 65ms	0.0015
6	4s 64ms	0.0014
7	4s 65ms	0.0013
8	4s 68ms	0.0016
9	4s 69ms	0.0013
10	4s 64ms	0.0013
…	…	…
25	4s 68ms	8.9866e-04

Для навчання нейронної мережі використовували 25 кроків, в кінцевому шарі похибка апроксимації складала 8.9866e-04 , і після цього було спроектовано передбачення і відобразили дані за допомогою бібліотеки matplotlib, графік зображений на рисунку нижче (рис. 3.2).

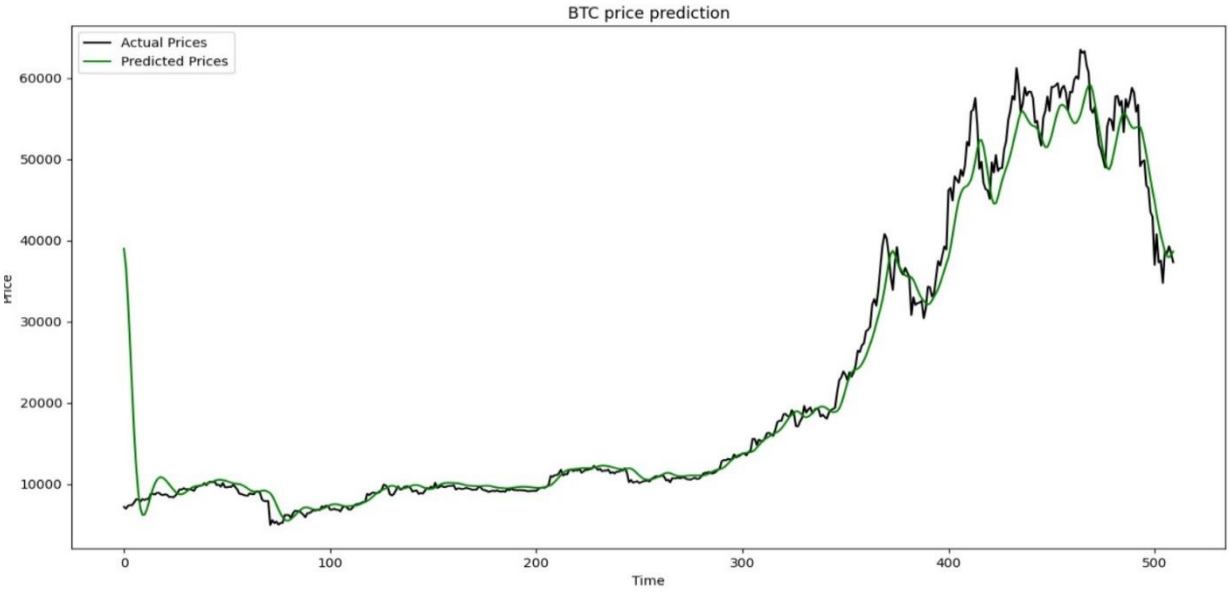

Рис. 3.2. Рисунок передбачення ціни на крипто валюти біткоїн (25 кроків)

Для більшої точності похибки апроксимації мережа була навчена за 100 кроків, дані представленні нижче (табл. 3.2).

Таблиця 3.2

Дані про навчання нейронної мережі

Номер епохи	Час на навчання	Похибка апроксимації
1	12s 66ms	0.0074
2	4s 68ms	0.0022
3	4s 65ms	0.0019
4	4s 66ms	0.0013
5	4s 65ms	0.0015
…	…	…
50	3s 50ms	6.0915e-04
…	…	…
75	3s 47ms	5.4812e-04
…	…	…
100	3s 46ms	3.9255e-04

Як видно з таблиці 3.2, з кожною наступною епохою час на навчання моделі зменшується, і похибка апроксимації стає меншою. На рисунку нижче (рис. 3.3.) відображені результати передбачення ціни крипто валюти.

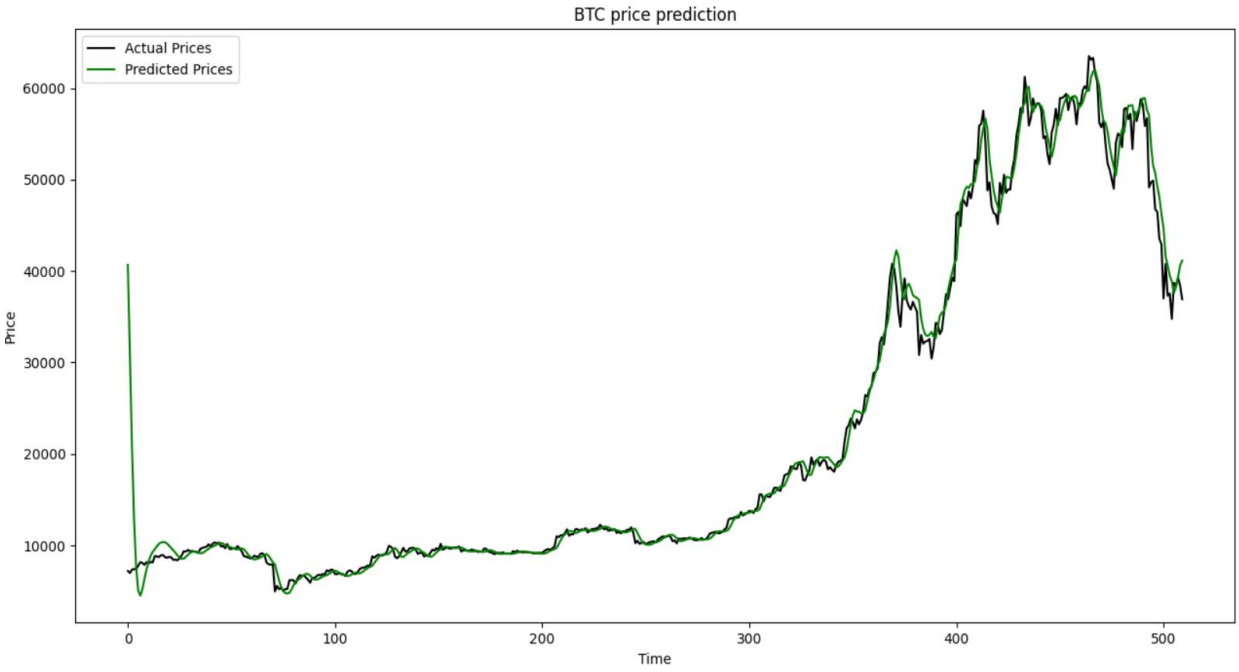

Рис. 3.3. Рисунок передбачення ціни на крипто валюти (100 кроків)

При побудові нейронної мережі була використанна модель Sequential. Модель Sequential - це (зазвичай) граф шарів. Найбільш поширеним видом моделей є стек шарів. Послідовна модель підходить для простого стеку шарів, де кожен шар має рівно один вхідний і один вихідний тензор, реалізація наведена на рисунку нижче (рис. 3.4.)

```
model = Sequential()

model.add(LSTM(units=50, return_sequences=True, input_shape=(x_train.shape[1], 1)))
model.add(Dropout(0.2))
model.add(LSTM(units=50, return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(units=50))
model.add(Dropout(0.2))
model.add(Dense(units=1))
```

Рис. 3.4. Реалізація моделі Sequential

Adam це оптимізатор імпульсу. Застосовуваний градієнт для кожного кроку залежить від змінного ковзкого середнього і стандартного відхилення градієнтів попередніх кроків (рис. 3.5.)

```
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(x_train, y_train, epochs=100, batch_size=32)
```

Рис. 3.5. Оптимізатор імпульсу “Adam”

Для навчання нейронної моделі було використано: Intel® Core™ i5-8500H @ 2.30GHz і відеокарта: Nvidia GeForce GTX 1060 6GB.

3.2. Класифікації бази даних «Титанік»

Затоплення Титаніка - одна з найвідоміших корабельних аварій в історії. 15 квітня 1912 року, під час свого першого плавання, широко розрекламований "непотоплюваний" Титанік затонув після зіткнення з айсбергом. На жаль, на борту не вистачило рятувальних шлюпок, що призвело до загибелі 1502 із 2224 пасажирів та екіпажу. Незважаючи на те, що в виживанні було задіяно якийсь елемент удачі, схоже, деякі групи людей виживали частіше, ніж інші.

Було побудовано модель прогнозування, яка відповідає на запитання: „які люди з більшою ймовірністю виживуть”, використовуючи дані пасажирів (тобто ім’я, вік, стать, соціально-економічний клас тощо).

Навчальний набір даних і набір даних тестування даються нам, ми можемо завантажити з Kaggle (Kaggle — платформа для змагань з аналітики та передбачувального моделювання).

Нами було імпортовано необхідні файли заголовків та бібліотеки, необхідні для вирішення цієї задачі, які наведені на рисунку нижче (рис. 3.6).

```
import seaborn as sns
import numpy as np
import pandas as pd


from matplotlib import pyplot as plt
from matplotlib import style

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import cross_val_score, KFold
from sklearn.feature_selection import SelectFromModel
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from xgboost import XGBClassifier
```

Рис. 3.6. Список використаних бібліотек

Фрагмент програми дослідницького аналізу даних наведений на рис 3.7.

```
train_df = pd.read_csv('train.csv')
#test_df = pd.read_csv('processed_test.csv')
print('Successfully loaded data')
```

Рис. 3.7. Дослідницький аналіз даних

Для кращого розуміння задачі наведемо словник ознак даних:

- Survived: 0 = ні, 1 = так
- Pclass: Клас квитків 1 = 1-й, 2 = 2-й, 3 = 3-й
- Sibsp: Кількість братів і сестер / подружжя на борту "Титаніка"
- Parch: Кількість батьків / дітей на борту "Титаніка"
- Ticket: Номер квитка
- Cabin: Номер кабіни
- Embarked: порт посадки C = Cherbourg, Q = Queenstown, S = Southampton

Ми бачимо, що в нашому навчальному наборі даних є 891 рядок та 12 стовпців, дані зображено на рисунку нижче (рис. 3.8).

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

Рис. 3.8. Навчальні дані

Аналіз показує, що значення віку пасажира відсутнє для багатьох рядків. Дані були досліджені на рисунку (рис. 3.9) і рисунку (рис. 3.10)

```
total = train_df.isnull().sum().sort_values()
percent = total / train_df.isnull().count().sort_values()
pd.concat([total,percent],axis = 1,keys = ['Count','%'])
```

Рис. 3.9. Код для визначення даних з пропущеними характеристиками

	Count	%
Age	177	0.198653
Cabin	687	0.771044
Embarked	2	0.002245
Fare	0	0.000000
Name	0	0.000000
Parch	0	0.000000
PassengerId	0	0.000000
Pclass	0	0.000000
Sex	0	0.000000
SibSp	0	0.000000
Survived	0	0.000000
Ticket	0	0.000000

Рис. 3.10. Процентне відношення пропущених даних

Із 891 рядків значення віку присутнє лише в 714 рядках. Подібним чином, значення номер каюти також відсутні у багатьох рядках. Лише 204 із 891 рядків мають значення каюти.

Проведено одновимірний аналіз даних за такими характеристиками:

- Стать.
- Клас квитка.
- Кількість братів і сестер / подружжя на борту "Титаніка".
- Кількість батьків / дітей на борту "Титаніка".
- Порт посадки.

Одновимірний аналіз даних за критерієм статі показано нижче на рисунку (рис. 3.11).

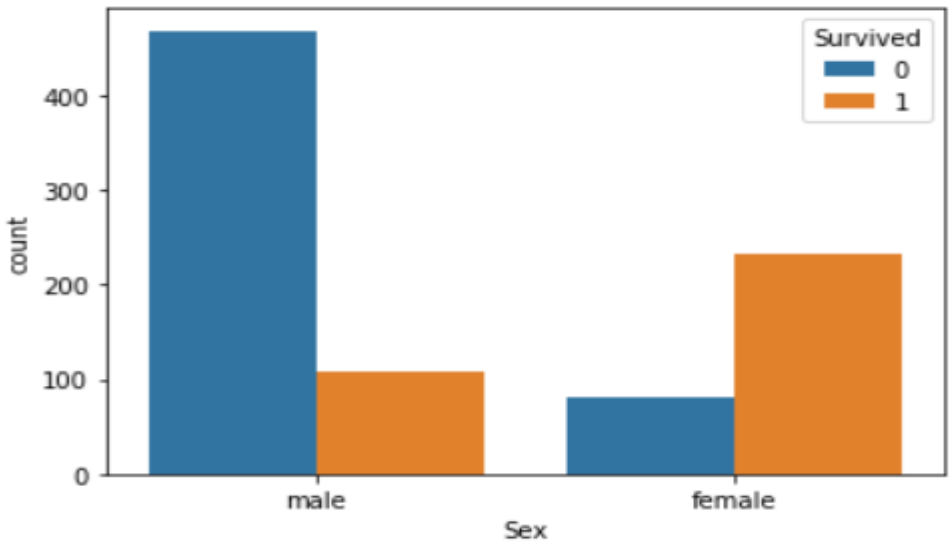

Рис. 3.11. Відношення вцілілих людей за критерієм статі

Діаграма підтверджує, що жінки виживали частіше, ніж чоловіки.

Одновимірний аналіз даних за критерієм класу квитка показано нижче (рис. 3.12.).

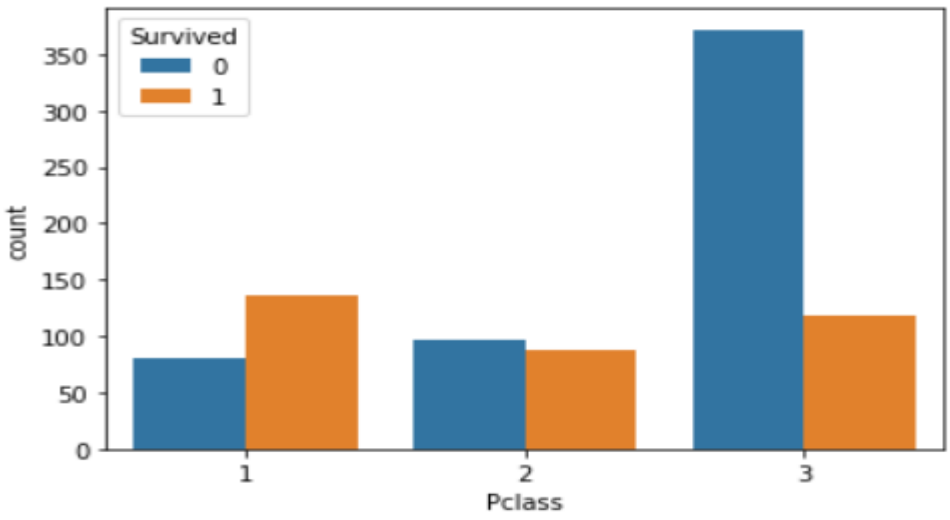

Рис. 3.12. Відношення вцілілих людей за критерієм класу квитка

Діаграма 3.12 підтверджує, що пасажири 1-го класу виживали частіше, ніж пасажири інших класів. Діаграма підтверджує, що пасажири 3-го класу помирали частіше, ніж пасажири інших класів.

Одновимірний аналіз даних за критерієм «кількість братів і сестер / подружжя» на борту "Титаніка" показано нижче на рисунку (рис. 3.13).

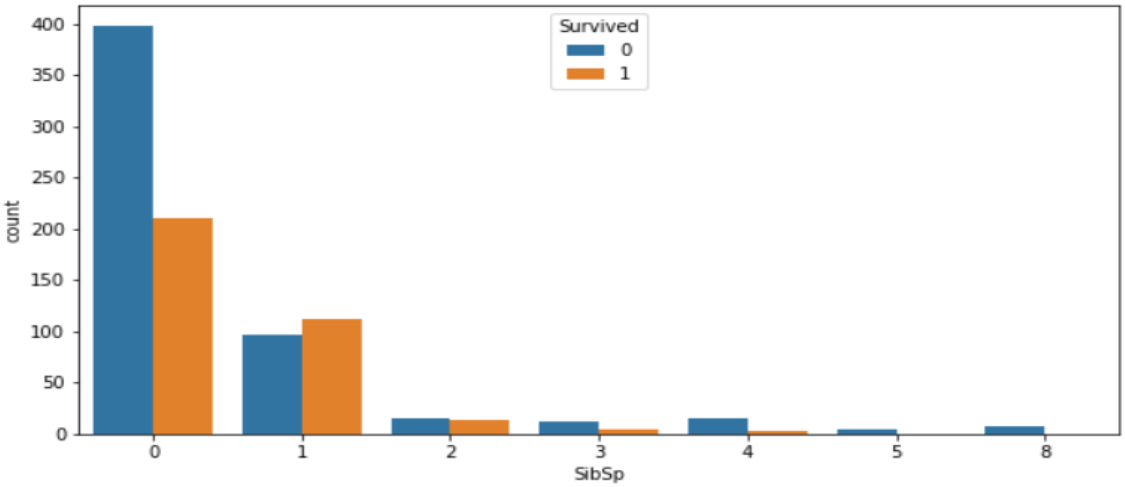

Рис. 3.13. Відношення вцілілих людей за критерієм кількість братів і сестер / подружжя на борту "Титаніка"

Діаграма підтверджує, що людина, яка перебувала на борту без братів і сестер чи подружжя, з більшою ймовірністю померла на борту корабля.

Одновимірний аналіз даних за критерієм «кількість батьків / дітей на борту "Титаніка"» показано на рис. 3.14.

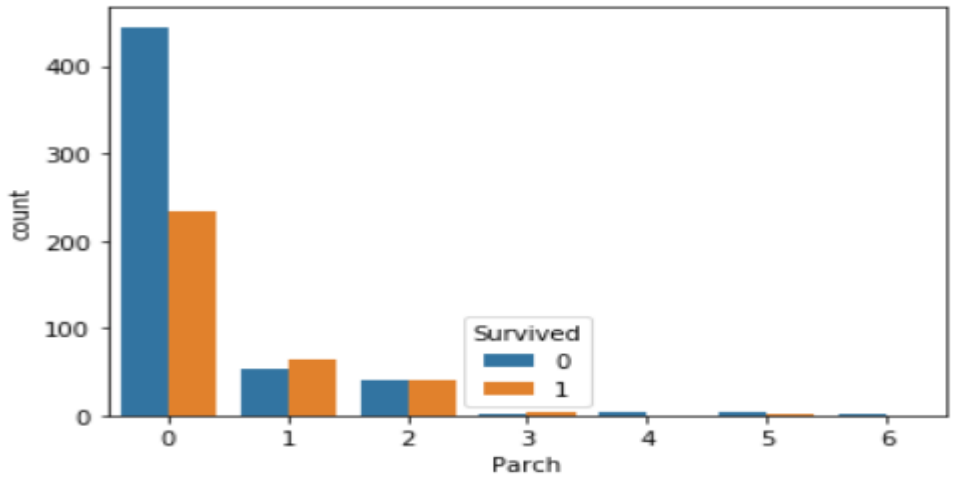

Рис. 3.14. Відношення вцілілих людей за критерієм кількість батьків / дітей на борту "Титаніка"

Одновимірний аналіз даних за критерієм порт посадки показано нижче на рисунку (рис. 3.15).

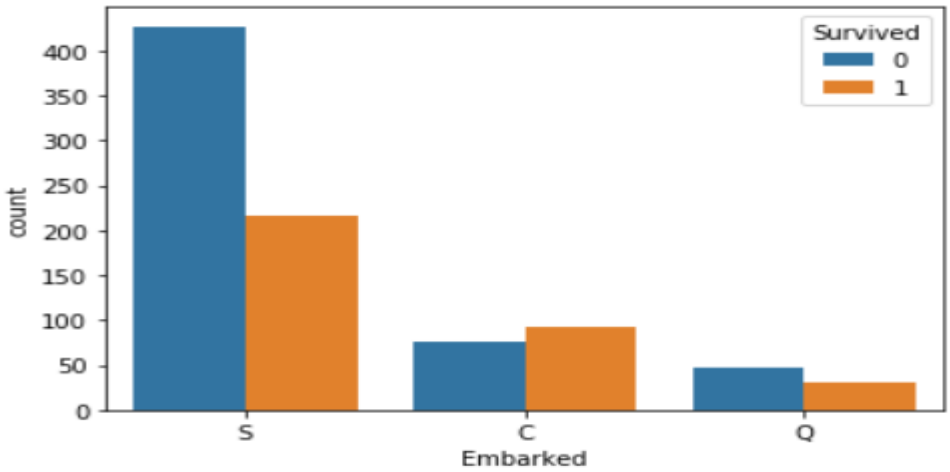

Рис. 3.15. Відношення вцілілих людей за критерієм порт посадки

Діаграма підтверджує, що людина, яка сідала на борт з порта Cherbourg, виживала з більшою вірогідністю.

Кореляційний аналіз даних про пасажирів Титаніка за допомогою кореляційної матриці зображено на рисунку нижче (рис. 3.16).

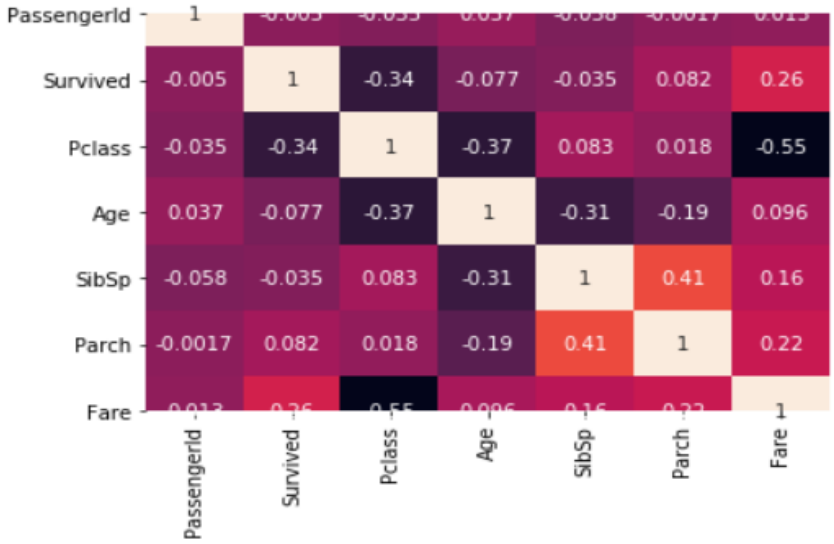

Рис. 3.16. Кореляційна матриця

Аналізуючи кореляційну матрицю можна зробити висновок: вік має негативну кореляцію з класом квитка, вік має негативну кореляцію з кількість братів і сестер / подружжя на борту та кількість батьків / дітей на борту.

Використаємо модель логістичної регресії для моделювання ймовірності існування певного класу або події, наприклад проходження / невдача, перемога / програш, живий / мертвий або здоровий / хворий. Це може бути розширено для моделювання декількох класів подій, таких як визначення, чи містить зображення кота, собаку, людини тощо. Кожному об'єкту, який виявляється на зображенні, буде присвоєна ймовірність від 0 до 1 із сумою одиниці. У нашому випадку модель логістичної регресії можна використати для передбачення ймовірності виживання дюдини, яка перебувала на борту Титаніка. При побудові моделі ми використовували всі дані про людину, які знаходилися у базі даних. Фрагмент коду, який реалізує логістичну регресію показано на рис. 3.17. Точність передбачень достатньо висока і становить 82.04 %.

```
sc = StandardScaler()
t = sc.fit_transform(train_reduced)
classifier = LogisticRegression()
score = cross_val_score(classifier, t, Y_train, cv=k_fold, n_jobs=1, scoring = 'accuracy')
print(np.mean(score)*100)
scores.append(np.mean(score))
model_name.append("Logistic Regression")
```

Рис. 3.17. Реалізація логістичної регресії

3.3. Класифікація динамічних об’єктів

Після початку пандемії коронавірусу було розроблено цілий комплекс заходів, спрямованих на запобігання поширенню цієї хвороби. Одним з таких заходів є соціальне дистанціювання. Соціальне дистанціювання, яке також називають "фізичним дистанціюванням", означає збереження безпечного простору між собою та іншими людьми, які не є вашими домочадцями. Щоб практикувати соціальну або фізичну дистанцію, потрібно дотримуватись принаймні 2 метри або на відстані витягнутої руки від інших людей, які не є вашими людьми з якими ви живете, як у приміщенні, так і на відкритому просторі.

COVID-19 поширюється переважно серед людей, які протягом тривалого періоду знаходяться в тісному контакті (на відстані приблизно 2 метрів). Поширення трапляється, коли заражена людина кашляє, чхає або розмовляє, а краплі з рота або носа потрапляють у повітря і потрапляють у рот або ніс людей поблизу. Краплі також можна вдихати в легені. Недавні дослідження показують, що люди, які інфіковані, але не мають симптомів, ймовірно, також відіграють певну роль у розповсюдженні COVID-19. Оскільки люди можуть поширювати вірус, перш ніж вони зрозуміють, що хворі, важливо триматися на відстані щонайменше 2 метрів від інших, коли це можливо, навіть якщо у вас - або у них - відсутні симптоми. Соціальна дистанція особливо важлива для людей, які мають більший ризик важкої хвороби через COVID-19.

Якщо ви хворі на COVID-19, маєте симптоми, що відповідають COVID-19, або були в тісному контакті з кимось, хто страждає COVID-19, важливо залишатися вдома і подалі від інших людей, поки не буде безпечно бути поруч з іншими. COVID-19 може жити на поверхні годинами або днями, залежно від таких факторів, як сонячне світло, вологість та тип поверхні. Можливо, людина може отримати COVID-19, торкнувшись поверхні або предмета, на якому є вірус, а потім торкнувшись власного рота, носа чи очей. Однак вважається, що це не основний спосіб поширення вірусу. Соціальна дистанція допомагає обмежити можливості контакту із забрудненими поверхнями та інфікованими людьми поза домом.

Незважаючи на те, що ризик важкої хвороби може бути різним для кожного, кожен може отримати та поширити COVID-19. Кожна людина повинна зіграти свою роль в уповільненні поширення та захисті себе, своєї родини та своєї громади. Окрім практичних повсякденних заходів щодо запобігання COVID-19, збереження простору між вами та іншими людьми є одним із найкращих інструментів, яким ми маємо уникати контакту з цим вірусом та уповільнення його поширення в громадах.

Нами було розроблено застосунок який може аналізувати дистанцію між людьми, даний програмний продукт використовує технологію комп'ютерного зору, в даному програмному продукті було використано OpenCV (OpenCV - це бібліотека з відкритим кодом комп'ютерного зору, яка призначена для аналізу, класифікації та обробки зображень) та навчений штучний інтелект який здатен аналізувати відстань між людьми і виводити дані в реальному часі, даний програмний продукт буде показаний нижче (рис. 3.18).

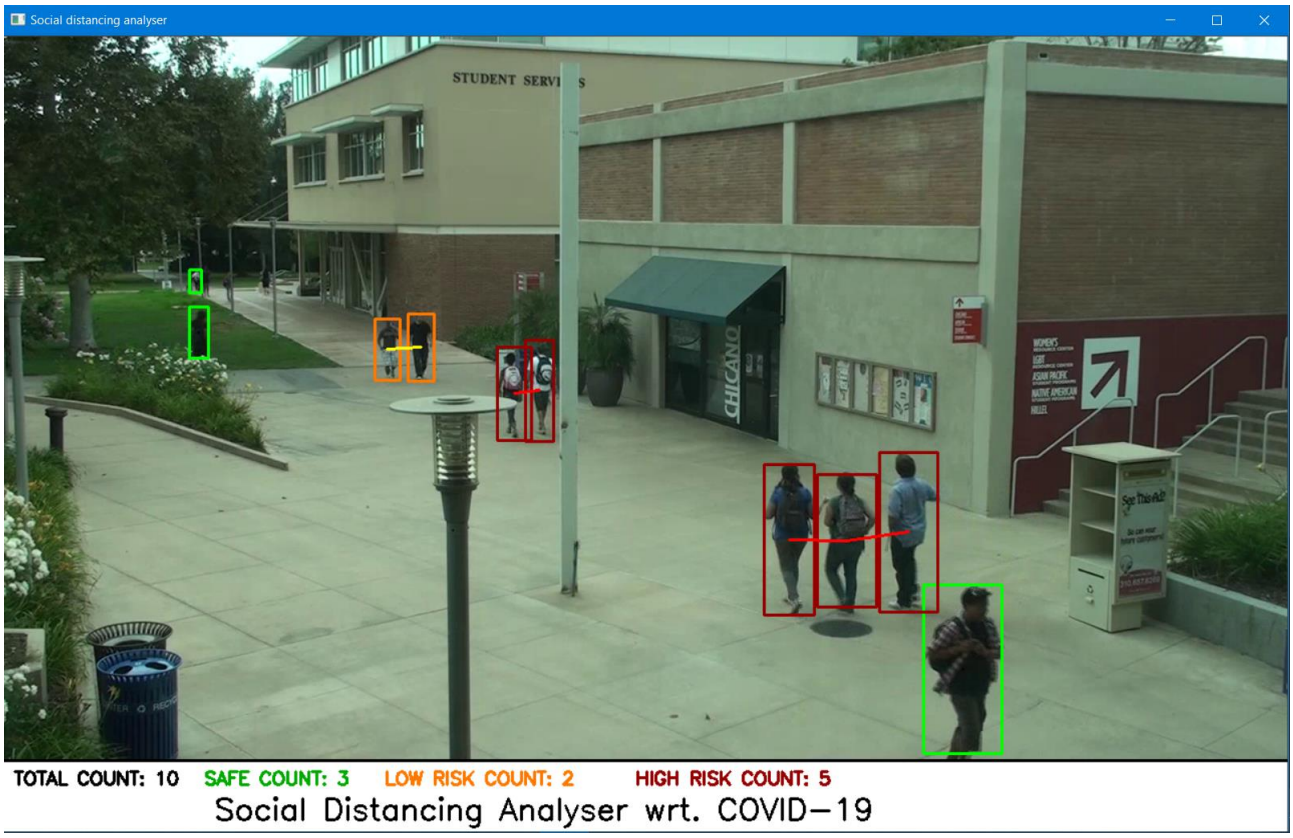

Рис. 3.18. Робота програмного продукту «Covid Distance Analizer»

Даний програмний продукт аналізує кожен кадр і оновлює дані зі швидкістю 30 кадрів в секунду (FPS, Frames Per Second), показано нижче (рис. 3.19).

```
    writer = cv2.VideoWriter("op_"+vname, fourcc, 30,
                             (frame.shape[1], frame.shape[0]), True)
```

Рис. 3.19. Параметри які використовуються для обробки відео

Для визначення відстані між об'єктами було використано атрибут shape (shape[0] – висота, shape[1] – ширина, shape[2] – кількість каналів), реалізація відображена нижче (рис. 3.20.)

```
if W is None or H is None:
    (H, W) = frame.shape[:2]
    FW=W
    if(W<1075):
        FW = 1075
    FR = np.zeros((H+210,FW,3), np.uint8)

    col = (255,255,255)
    FH = H + 210
FR[:] = col
```

Рис. 3.20. Визначення відстані між обєктами

Нейронна мережа використовується тут для класифікації динамічних об'єктів, у ролі яких виступають людські фігури. Для навчання програми потрібно було надати багато зображень із людськими фігурами, причому для підвищення рівня точності потрібно більше даних. Під час обробки зображення, ми використали метод згорткової фільтрації для визначення горизонтальних країв (рис. 3.21).

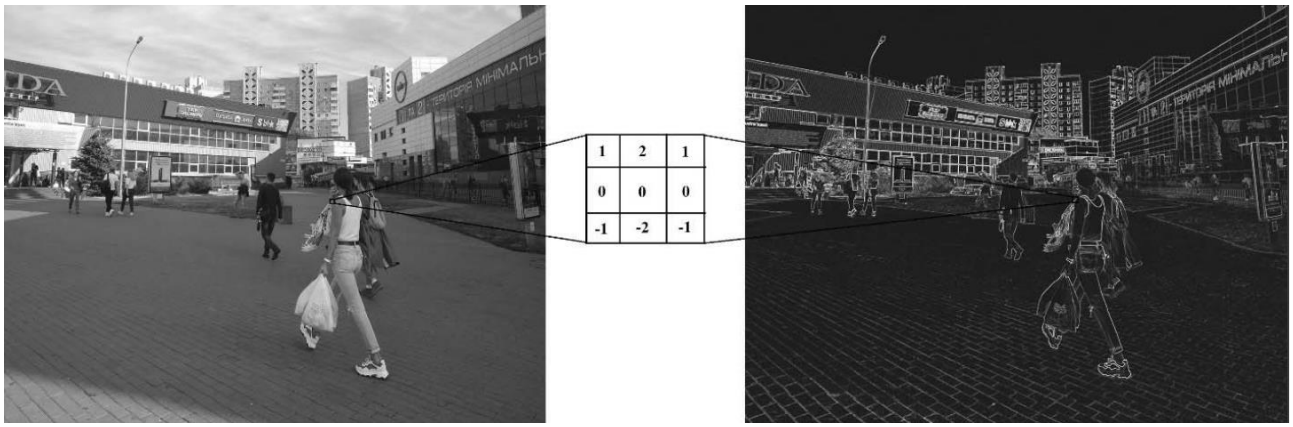

Рис. 3.21. Визначення горизонтальних країв методом згорткової фільтрації

Рис. 3.22. Зображення для тренування нейронної мережі

Для навчання нейронної мережі, яка була використана в нашому програмному продукту, було використано відкриту базу даних зображень MS-COCO. На зображенні нижче (рис. 3.22.) наведено декілька прикладів зображень, які використовуються для навчання та тренування нейронної мережі.

Після того, як нейронна мережа була навчена, нами було розроблено інтерфейс для зручного використання розробленого нами програмного продукту. Для зберігання файлів ми використовували відео в форматі .mp4, які зберігаються в папці videos, на рисунку нижче представлено (рис. 3.23).

Рис. 3.23. Відео які тестувались

Під час роботи, розробленої нами програми, аналізується відстань і для зручного використання програмного продукту, будуть створюватись прямокутні рамки та лінії які позначають відстань між людьми : червоний колір лінії означає, що люди знаходяться в умовах високого ризику для зараження (рис. 3.24), а оранжевий колір лінії означає що існує помірний ризик зараження (рис. 3.25).

Рис 3.24. Люди в умовах високого ризику зараження

Рис. 3.25. Люди в умовах низької ймовірності зараження

Зелений колір рамки навколо людської фігури інформує про безпечну відстань даного динамічного об’єкта від інших об’єктів (з точки зору інфікування). Кожна людина яка розпізнається програмним продуктом зберігається в окремій змінній. Загальні дані про кількість спостережених людей розміщено в нижній частинні програми (рис. 3.26). Тут представлені наступні дані для відображення: загальна кількість людей, кількість людей які знаходяться на безпечній відстані, кількість людей які знаходяться на відстані з низькою імовірністю зараження, і кількість людей які знаходяться на відстані з високою імовірністю зараження.

Рис. 3.26. Результати класифікації групи людей (кількість людей, віднесені до кожної з груп ризику)

Висновки по розділу 3

В третьому розділі описано декілька задач які вирішують методи класифікації та прогнозування за допомогою нейронних мереж у середовищі Python. Наведено процес навчання мереж, продемонстровані результати на тестових наборах даних. Зокрема розглянуті та вирішені задача прогнозування ціни криптовалюти, задача про класифікацію та прогнозування виживання пасажирів на борту лайнера «Титанік», задача ідентифікації та класифікації людських фігур у натовпі.

ВИСНОВКИ

Методи штучного інтелекту відіграють все більшу роль у нашому повсякденному житті. Вони проникають у роботу побутових приладів та механізмів, реалізують комфортні умови проживання у «розумних» будинках тощо. З початком пандемії коронавірусу стала актуальною задача про поведінку людей у місцях масового скупчення та на вулицях міст. Зокрема, дуже важливим є дотримання вимоги про «соціальну дистанцію» між людьми.

У даній кваліфікаційній роботі було реалізовано задачі класифікації та прогнозування на мові програмування Python. В першому розділі розглянуто теоретичні основи інтелектуальних технологій, їхній сучасний стан та розвиток, а також методи та моделі інтелектуальних технологій, штучні нейронні мережі. Проаналізовано поняття інтелектуальних технологій, а також методи та принципи побудови інформаційних систем, як от методи "знизу-вгору", "зверху-вниз", принципи "дуалізму", багатокомпонентний. Також було розглянуто сутність машинного навчання та методи його реалізації - метод регресії, методи класифікації та кластеризації, метод зменшення розмірності, нейронні мережі.

Нами було проаналізовано роботу нейронних мереж, а також методи їх навчання, як от контрольоване, метод радіальних базових функцій, неконтрольований, метод зворотного поширення помилки. Зокрема проаналізовано різні правила навчання нейронних мереж. Також визначено, що класифікація та прогнозування є найбільш дослідженими темами нейронних мереж, які можна використовувати в різноманітних сферах людської діяльності.

Для створення нейронних мереж найбільш зручною мовою програмування є Python. Попри існування ще кількох мов, які можна використовувати для створення штучної нейронної мережі, Python залишається швидшим і кращим засобом для вирішення цієї задачі.

В третьому розділі було розглянуто декілька задач, які можна вирішити із застосуванням нейронних мереж. Перша задача – це прогнозування ціни криптовалют. Для вирішення задачі було створено нейронну модель, що аналізує попередні дані зміни курсу крипто валюти з початку 2016 до 2020 року, на основі

теорії того, що ринок циклічний. За основну крипто валюту було використано біткоїн, а для навчання нейронної моделі відкриту бібліотеку TensorFlow.

Наступна задача, яку ми розглянули – це задача про загибель пасажирів лайнера «Титанік». Нами було класифіковано базу даних "Титанік", та проведено дослідницький аналіз даних, згідно з яким визначено, що вік має негативну кореляцію з класом квитка, вік має негативну кореляцію з кількість братів і сестер / подружжя на борту та кількість батьків / дітей на борту. Побудована прогнозна модель виживання особи типу логістична регресія, точність передбачення виживання за якою становить 82 %.

Для вирішення задачі класифікації динамічних об'єктів (зокрема людських фігур) було розроблено застосунок який може аналізувати дистанцію між людьми. Даний програмний продукт використовує технологію комп'ютерного зору, в даному програмному продукті було використано OpenCV (OpenCV - це бібліотека з відкритим кодом комп'ютерного зору, яка призначена для аналізу, класифікації та обробки зображень) та навчений штучний інтелект який здатен аналізувати відстань між людьми і виводити дані в реальному часі. Під час обробки зображення було використано метод згорткової фільтрації для визначення горизонтальних країв. Для зручного використання програмного продукту у програмі створюються прямокутні рамки та лінії які позначають відстань між людьми : червоний колір лінії означає, що люди знаходяться в умовах високого ризику для зараження, а оранжевий колір лінії означає що існує помірний ризик зараження. Зелений колір рамки інформує про безпечну відстань між людьми.

Таким чином, нами було охарактеризовано та проаналізовано методи реалізації інтелектуальних технологій з використанням платформи Python, а також запропоновано методи застосування нейронних мереж для вирішення задач класифікації та прогнозування динамічних об'єктів. Написана та реалізована програма для ідентифікації людських фігур з фрагментів відеозображень з наступною класифікацією їх за групами ризику. Робота може представляти інтерес для керівників міських служб та для керівників сфери охорони здоров'я.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. New intellectual technologies and actual promblems of the modern information society. URL: http://revistaespacios.com/a18v39n39/a18v39n39p08.pdf
2. Information technology. URL: https://en.wikipedia.org/wiki/Information_technology
3. Нестеренко О. В., Ковтунець О. В., Фаловський О. О. Інтелектуальні системи і технології. Ввідний курс: Навч. Посібник. – К.: Національна академія управління, 2017. – 90 с.
4. Ng Andrew, Machine learning yearning. Technical strategy for AI Engineers, In the Era of Deep Learning, 2018. 118 p.
5. Коцовський В. М. Методи та системи штучного інтелекту. Конспект лекцій. – МОН ДВНЗ «Ужгородський національний університет», 2016. – 76 с.
6. Могильний С. Б. Машинне навчання з використанням мікропомп’ютерів: Навч. Посібник. – Національна академія наук України, 2019. – 226 с.
7. Новотарський М. А., Нестеренко Б. Б. Штучні нейронні мережі: обчислення. // Праці Інституту математики НАН України. – Т50. – Київ: Ін-т математики НАН України, 2004. – 408 с.
8. David Kriese, A Brief Introduction to Neural Networks, 2007. 244 p.
9. Кононюк А.Ю. Нейроні мережі і генетичні алгоритми – К.: «Корнійчук», 2008. – 446 с.
10. Francois Chollet Deep Learning with Python, - Manning Publications, 2018. 361 p. URL: http://faculty.neu.edu.cn/yury/AAI/Textbook/Deep%20Learning%20with%20Python.pdf
11. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2015. – 400 с.
12. Donald Michie, Donald Michie, Donald Michie Machine Learning, Neural and Statistical Classification. Ellis Horwood, 1994. 298 p.

13. Neural Networks and Deep Learning, Michael Nielsen. URL: https://static.latexstudio.net/article/2018/0912/neuralnetworksanddeeplearning.pdf
14. Применение нейронных сетей для решения задач прогнозирования. URL: http://elibrary.lt/resursai/Uzsienio%20leidiniai/MFTI/2006/136.pdf
15. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: Навчальний посібник. – Запоріжжя: ЗНТУ, 2008. – 341 с.
16. IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews) 30(4):451 – 462 с.
17. Dr. Chris Bourke Computer Science I. Department of Computer Science & Engineering University of Nebraska – Lincoln, NE 68588, USA, 2018. 613 p.
18. Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers How to Think Like a Computer Scientist: Learning with Python 3, 2020. 384 p.
19. Программирование на Python для начинающих: [перевод с англ. М. А. Райтмана] / Майк МакГрат. – Москва: Эскмно, 2015. – 192 с.
20. Lisa Tagliaferri, Brian Boucheron Python Machine Learning Projects, - DigitalOcean, 2019. 135 p.
21. 10 Machine Learning Methods that Every Data Scientist Should Know. URL: https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0eeee9
22. K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks. URL: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a
23. Lucas Brocki Kohonen Self-Organizing Map for the Traveling Salesperson Problem, - Polish–Japanese Institute of Information Technology, 2007. 116-119 p.
24. Radial basis function interpolation. URL: https://en.wikipedia.org/wiki/Radial_basis_function_interpolation#:~:text=Radial%20basis%20function%20(RBF)%20interpolation,sum%20of%20radial%20basis%20functions
25. David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams. Learning representations by back-propagating errors, - Institute for Cognitive Science,

University of California San Diego, California 92093, USA, 1986. 533-536 p. URL: https://www.iro.umontreal.ca/~vincentp/ift3395/lectures/back-prop_old.pdf

26. Christopher H. Bennett, Vivek Parmar, Laurie E. Calvet, Jacques-Olivier Klein, Manan Suri, Matthew J. Marinella, Damien Querlioz. Contrasting advantages of learning with random weights and backpropagation in non-volatile memory neural networks, - Centre de Nanosciences et de Nanotechnologies, Univ. Paris-Sud, Univers e Paris-Saclay, France, 2016. 16 p.
27. Блокчейн. URL: https://uk.wikipedia.org/wiki/Блокчейн

ДОДАТКИ

Код реалізації контурів які позначають рівень безпеки людей.

```
for i in idf:
    cv2.line(FR,(0,H+1),(FW,H+1),(0,0,0),2)
    cv2.putText(FR, "Social Distancing Analyser wrt. COVID-19", (210, H+60),
            cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 2)
    cv2.rectangle(FR, (20, H+80), (510, H+180), (100, 100, 100), 2)
    cv2.putText(FR, "Connecting lines shows closeness among people. ", (30,
H+100),
            cv2.FONT_HERSHEY_SIMPLEX, 0.6, (100, 100, 0), 2)
    cv2.putText(FR, "-- YELLOW: CLOSE", (50, H+90+40),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 170, 170), 2)
    cv2.putText(FR, "--    RED: VERY CLOSE", (50, H+40+110),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)
    # cv2.putText(frame, "--    PINK: Pathway for Calibration", (50, 150),
    #            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (180,105,255), 1)

    cv2.rectangle(FR, (535, H+80), (1060, H+140+40), (100, 100, 100), 2)
    cv2.putText(FR, "Bounding box shows the level of risk to the person.", (545,
H+100),
            cv2.FONT_HERSHEY_SIMPLEX, 0.6, (100, 100, 0), 2)
    cv2.putText(FR, "-- DARK RED: HIGH RISK", (565, H+90+40),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 150), 2)
    cv2.putText(FR, "--   ORANGE: LOW RISK", (565, H+150),
```

```
        cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 120, 255), 2)

cv2.putText(FR, "--    GREEN: SAFE", (565, H+170),
        cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 150, 0), 2)

tot_str = "TOTAL COUNT: " + str(total_p)
high_str = "HIGH RISK COUNT: " + str(high_risk_p)
low_str = "LOW RISK COUNT: " + str(low_risk_p)
safe_str = "SAFE COUNT: " + str(safe_p)

cv2.putText(FR, tot_str, (10, H +25),
        cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 0), 2)
cv2.putText(FR, safe_str, (200, H +25),
        cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 170, 0), 2)
cv2.putText(FR, low_str, (380, H +25),
        cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 120, 255), 2)
cv2.putText(FR, high_str, (630, H +25),
        cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 150), 2)

(x, y) = (boxes[i][0], boxes[i][1])
(w, h) = (boxes[i][2], boxes[i][3])
if status[kk] == 1:
```

```
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 150), 2)


        elif status[kk] == 0:
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)


        else:
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 120, 255), 2)


        kk += 1
    for h in close_pair:
        cv2.line(frame, tuple(h[0]), tuple(h[1]), (0, 0, 255), 2)
    for b in s_close_pair:
        cv2.line(frame, tuple(b[0]), tuple(b[1]), (0, 255, 255), 2)
    FR[0:H, 0:W] = frame
    frame = FR
    cv2.imshow('Social distancing analyser', frame)
    cv2.waitKey(1)


if writer is None:
    fourcc = cv2.VideoWriter_fourcc(*"MJPG")
    writer = cv2.VideoWriter("op_"+vname, fourcc, 30,
                             (frame.shape[1], frame.shape[0]), True)
```