Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Програмні засоби кластеризації даних з використанням штучних нейронних мереж

(тема)

Виконав:
здобувач 4 року навчання,
групи КІУКІ-21-4
Олег РУДЕНКО
(власне ім'я, прізвище)

Спеціальність
123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

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

Освітня програма
Комп'ютерна інженерія
(повна назва освітньої програми)

Керівник: доц. Антон СОРОКІН
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

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

Освітня програма Комп’ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Руденку Олегу Дмитровичу
(прізвище, ім’я, по батькові)

1. Тема роботи Програмні засоби кластеризації даних з використанням штучних нейронних мереж

затверджена наказом по університету від “ 26 ” травня 2025 р. № 426 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 16 червня 2025 р.

3. Вхідні дані до роботи
мережна аномалія
Google Colab
Python
розпізнавання зображень
програмні засоби

4. Перелік питань, що потрібно опрацювати у роботі
Огляд літературних джерел по штучним нейронним мережам
Методи кластеризації даних
Програмна реалізація розглянутих методів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 13 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання та аналіз літератури	26.05.2025–30.05.2025	
2	Огляд існуючих засобів виявлення аномалій	31.05.2025–03.06.2025	
3	Вибір алгоритмів	04.06.2025–06.06.2025	
4	Вибір програмних засобів	07.06.2025–08.06.2025	
5	Програмна реалізація	09.06.2025–11.06.2025	
6	Аналіз отриманих результатів	12.06.2025–13.06.2025	
7	Оформлення записки	14.06.2025–16.06.2025	

Дата видачі завдання " 26 " травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Антон СОРОКІН
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 58 с., 21 рис., 2 дод., 19 джерел.

АВТОЕНКОДЕР, КЛАСТЕРИЗАЦІЯ, UMAP, ГЛИБИННЕ НАВЧАННЯ, МАШИННЕ НАВЧАННЯ, ЗНИЖЕННЯ РОЗМІРНОСТІ, SCIKIT-LEARN, ORANGE, РУКОПИСНІ ЦИФРИ, PYTHON.

Метою кваліфікаційної роботи є є створення програмного інструменту для обробки даних з використанням алгоритмів кластерного аналізу, побудованих на основі штучних нейронних мереж.

У ході виконання кваліфікаційної роботи було здійснено глибокий аналіз кластеризації рукописних цифр із використанням автоенкодера, алгоритмів зниження розмірності та методів кластерного аналізу. Автоенкодер успішно навчився відтворювати вхідні зображення, що підтверджується візуальною подібністю між оригіналами та їх реконструкціями, а також зменшенням функції втрат протягом епох навчання. Це свідчить про здатність моделі ефективно кодувати й відновлювати інформацію.

ABSTRACT

Bachelor’s thesis: 58 pages, 21 figures, 2 appendices, 19 sources.

AUTOENCODER, CLUSTERING, UNIFORM MANIFOLD APPROXIMATION AND PROJECTION (UMAP), DEEP LEARNING, MACHINE LEARNING, DIMENSIONALITY REDUCTION, SCIKIT-LEARN, ORANGE, HANDWRITTEN DIGITS, PYTHON.

The major goal of this thesis is to develop a software tool for data processing using clustering algorithms based on artificial neural networks.

In order to an in-depth analysis of handwritten digits clustering was performed using an autoencoder, dimensionality reduction algorithms, and clustering methods.

The autoencoder successfully learned to reconstruct input images, as demonstrated by the visual similarity between the original samples and their reconstructions, as well as the reduction of the loss function across training epochs. This confirms the model's ability to effectively encode and reconstruct information.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ШНМ – штучна нейронна мережа

GAN – Generative Adversarial Network

GRU – Gated Recurrent Unit

LSTM – Long Short-Term Memory

MLP – Multilayer Perceptron

MSE – Mean Squared Error

RNN – Recurrent Neural Network

SOM – Self-Organizing Map

UMAP – Uniform Manifold Approximation and Projection

ВСТУП

Застосування кластеризації даних у поєднанні з технологіями машинного навчання відіграє важливу роль у сучасному аналізі великих інформаційних масивів. Цей підхід дозволяє ефективно структурувати дані шляхом об’єднання об’єктів із подібними характеристиками в однорідні групи, що сприяє глибшому розумінню внутрішніх закономірностей та динаміки досліджуваних процесів. Завдяки використанню алгоритмів машинного навчання вдається підвищити точність і продуктивність процесу кластеризації, зменшити вплив людського фактора й автоматизувати обробку даних на всіх етапах.

Концепція кластеризації ґрунтується на ідеї максимального зближення об'єктів у межах одного кластеру при водночас чіткому розмежуванні між різними кластерами. Для досягнення цього ефекту застосовуються різні математичні моделі, серед яких помітне місце займає метод K-середніх. Цей алгоритм, завдяки своїй простоті й наочності, часто використовується як базовий інструмент кластерного аналізу. Його суть полягає у багаторазовому уточненні центрів кластерів та перерозподілі об’єктів між ними до досягнення стійкої конфігурації.

Ієрархічна кластеризація, на відміну від K-середніх, дозволяє формувати багаторівневу структуру групування без необхідності попереднього визначення кількості кластерів. Цей підхід забезпечує гнучкий інструментарій для аналізу даних на різних рівнях деталізації, що особливо цінно в дослідницьких задачах.

Метод головних компонент, який зазвичай застосовується перед кластеризацією, дає змогу зменшити розмірність даних, зберігаючи при цьому основні характеристики варіативності. Така трансформація значно полегшує аналіз і підвищує якість кластерного розподілу.

Інтеграція кластеризації з методами машинного навчання, зокрема

нейронними мережами, відкриває нові можливості для виявлення складних і нелінійних залежностей у даних. Автоматичне навчання на великих вибірках сприяє формуванню більш релевантних та адаптивних моделей кластеризації, що надає вагомі переваги в таких галузях, як економіка, медицина, біоінформатика та соціальні науки.

У межах даною роботи передбачено створення програмного інструменту для обробки великих обсягів інформації з використанням алгоритмів кластерного аналізу, побудованих на основі штучних нейронних мереж. Зокрема, акцент буде зроблено на порівняльному аналізі ефективності різних підходів до кластеризації, їх практичній реалізації в програмному середовищі, а також на моделюванні кластерів з метою ідентифікації схожих об’єктів у досліджуваних вибірках.

1 ШТУЧНІ НЕЙРОННІ МЕРЕЖІ

Штучні нейронні мережі (ШНМ) становлять фундамент сучасних технологій машинного та глибокого навчання. Ці системи моделюють принципи функціонування біологічних нейронних мереж, створюючи багаторівневу архітектуру, що складається з взаємопов'язаних обчислювальних елементів – нейронів. Типова структура ШНМ включає вхідний шар, декілька прихованих шарів і вихідний шар, кожен з яких виконує специфічні функції у процесі перетворення інформації. Вхідний шар фіксує дані, що надходять до мережі, приховані шари здійснюють багатоступеневу обробку сигналів, тоді як вихідний шар формує кінцеві результати обчислень.

Важливою складовою ефективної роботи нейронної мережі є активаційні функції, які визначають ступінь активації нейронів та сприяють нелінійності моделі. Серед найпоширеніших функцій активації виділяються Sigmoid, що мапує значення у діапазон [0, 1], ReLU, яка пригнічує негативні значення, пропускаючи позитивні без змін, та Tanh, що забезпечує симетричне відображення значень у межах [-1, 1].

Процес навчання нейронної мережі реалізується через механізм зворотного поширення помилки, в рамках якого визначається відхилення фактичного результату від очікуваного. Це відхилення поширюється у зворотному напрямку через усі шари мережі з метою корекції вагових коефіцієнтів, що визначають вплив кожного нейрону. Така адаптація виконується ітеративно, що дозволяє моделі поступово знижувати рівень похибки та досягати оптимальної точності передбачень.

Завдяки своїй гнучкості та здатності виявляти складні залежності у великих масивах даних, ШНМ знайшли широке застосування у різноманітних сферах. У галузі комп'ютерного зору вони слугують основою для розпізнавання об'єктів на зображеннях та відео; в обробці природної

мови забезпечують функціонування систем машинного перекладу, синтезу мовлення та аналізу семантики текстів. У медицині ШНМ використовуються для автоматизованої діагностики патологій на основі медичних візуалізацій, а також у геномних дослідженнях. У фінансовому секторі вони застосовуються для прогнозування ринкових коливань та ідентифікації підозрілої активності.

Таким чином, штучні нейронні мережі посідають провідне місце в арсеналі інструментів сучасного аналізу даних, забезпечуючи інтелектуальну підтримку прийняття рішень у багатьох наукових і прикладних галузях. Їхній подальший розвиток відкриває нові перспективи для вирішення задач, що вимагають високого рівня аналітичної складності.

1.1 Історія виникнення і розвитку нейронних мереж

Історія формування та еволюції штучних нейронних мереж (ШНМ) відображає тривалий і динамічний процес наукового поступу, який бере свій початок у середині XX століття й охоплює низку фундаментальних відкриттів, що стали визначальними для розвитку сучасного штучного інтелекту.

Зародження концепції ШНМ припадає на 1943 рік, коли нейрофізіолог Воррен МакКаллоч та математик Волтер Піттс вперше запропонували формальну модель штучного нейрона, здатного до виконання базових логічних операцій. Ця праця заклала теоретичне підґрунтя для подальших досліджень у галузі обчислювальних моделей, що імітують діяльність людського мозку.

Важливою віхою у становленні нейронних мереж стала розробка Франком Розенблаттом у 1958 році перцептрона – першої адаптивної моделі, що навчалася на основі вхідних даних. Незважаючи на свою інноваційність, перцептрон мав суттєві обмеження й не міг розв’язувати нелінійно роздільні задачі, що викликало критичну реакцію наукової спільноти. Зокрема, у праці Марвіна Мінського та Сеймура Пейперта було

теоретично обґрунтовано неспроможність перцептронів моделювати такі задачі, як логічна функція XOR, що призвело до зниження інтересу до тематики на декілька десятиліть.

Ситуація кардинально змінилася з появою у 1980-х роках нових підходів до тренування багатошарових мереж. Джон Хопфілд запропонував модель рекурентної мережі з асоціативною пам’яттю, а Джеффрі Хінтон, Девід Румелхарт та Рональд Вільямс розробили алгоритм зворотного поширення помилки (backpropagation), що дозволив ефективно навчати багатошарові перцептрони, значно розширивши можливості ШНМ для обробки складних даних.

У 1990-х роках розвиток архітектур нейронних мереж тривав інтенсивними темпами: з’явилися нові типи, такі як радіально-базисні мережі та самоорганізуючі карти Кохонена. Поступовий прогрес в обчислювальній техніці та накопичення великих обсягів даних створили передумови для нового етапу – епохи глибокого навчання (deep learning).

Суттєвим проривом стало впровадження багатошарових згорткових нейронних мереж (CNN), які виявилися надзвичайно ефективними для аналізу зображень. Архітектура AlexNet, що перемогла у престижному змаганні ImageNet, продемонструвала переваги глибоких моделей у задачах розпізнавання образів. У сфері обробки послідовностей знайшли широке застосування рекурентні мережі (RNN) та їх удосконалення, зокрема LSTM, які суттєво покращили результати у мовних та часових задачах.

Значущим внеском у розвиток генеративних моделей стала архітектура Generative Adversarial Networks (GAN), запропонована Яном Гудфеллоу, яка відкрила можливості для синтезу реалістичних зображень та інших даних. Ще одним етапом у технологічному прогресі стало створення трансформерних моделей, зокрема у роботі "Attention is All You Need", що започаткувала новий підхід до обробки природної мови, кульмінацією якого стали моделі сімейства GPT.

Прикладом прикладного успіху глибокого навчання є система AlphaGo

від компанії DeepMind, яка змогла перемогти професійного гравця у го – гру, що довгий час вважалась викликом для штучного інтелекту.

Сьогодні штучні нейронні мережі застосовуються у широкому спектрі галузей – від медичної діагностики до фінансового прогнозування, від автономного транспорту до індустрії розваг. Постійне вдосконалення архітектур, зокрема розвиток трансформерів та методів підкріплювального навчання, забезпечує подальше розширення можливостей ШНМ та відкриває нові горизонти у вирішенні складних міждисциплінарних завдань.

Таким чином, історія ШНМ є не лише хронікою технологічних інновацій, а й свідченням еволюції наукової думки, що постійно трансформується під впливом нових відкриттів, практичних викликів та суспільного попиту.

1.2 Класифікація нейронних мереж

Штучні нейронні мережі (ШНМ) становлять основу сучасних інтелектуальних систем і класифікуються відповідно до різних характеристик, таких як архітектурна будова, тип навчання та сфера застосування. Архітектурно ШНМ відрізняються кількістю шарів, характером зв’язків між нейронами та напрямком передачі інформації. Найпростішими є одношарові моделі, на кшталт перцептрона, тоді як багатошарові перцептрони (MLP) містять кілька рівнів обробки: від вхідного шару до одного чи більше прихованих і вихідного. Передача сигналів у таких мережах може бути прямолінійною (feedforward), або рекурентною, що передбачає зворотні зв’язки між нейронами.

Серед спеціалізованих архітектур вирізняються згорткові нейронні мережі (CNN), які стали стандартом у завданнях комп’ютерного зору. Вони інтегрують згорткові шари для виявлення локальних ознак та шари субдискретизації (pooling) для зменшення розмірності й підвищення

обчислювальної ефективності. Для обробки послідовностей, таких як текст чи часові ряди, ефективними виявились рекурентні нейронні мережі (RNN), що дозволяють зберігати контекст попередніх елементів. Модифікації типу LSTM (Long Short-Term Memory) усувають обмеження базових RNN, пов’язані з втратою градієнта, забезпечуючи збереження інформації протягом тривалих інтервалів.

Генеративні змагальні мережі (GAN), що поєднують два компоненти – генератор і дискримінатор, – відкрили нові горизонти у синтезі даних. Під час взаємодії ці дві моделі удосконалюють одна одну, створюючи дані, що імітують реальні з високим ступенем достовірності. Така архітектура знайшла застосування у генерації зображень, відео та аудіо.

Однією з найреволюційніших архітектур останнього десятиліття стали трансформери. Завдяки механізму самоуваги (self-attention), трансформерні мережі ефективно враховують взаємозв’язки між усіма елементами вхідної послідовності незалежно від їх положення, що зробило їх незамінними в задачах обробки природної мови, машинного перекладу, генерації текстів і створення мовних моделей нового покоління.

Типи навчання ШНМ охоплюють три основні підходи: навчання з учителем, без учителя та підкріплювальне навчання. У першому випадку модель вчиться на основі маркованих даних, у другому – самостійно виявляє приховані структури у немаркованих даних. Підкріплювальне навчання базується на взаємодії агента з середовищем і моделює поведінку на основі винагороди, що є особливо актуальним для систем автономного управління та робототехніки.

Застосування ШНМ є надзвичайно широким і охоплює численні сфери. У комп’ютерному зорі вони використовуються для ідентифікації об’єктів, у мовних технологіях – для перекладу, розпізнавання голосу, аналізу тональності, у фінансах – для прогнозування динаміки ринку та виявлення шахрайства, у медицині – для діагностики на основі медичних зображень і генетичних досліджень. Вони стали складовою частиною систем

автономного транспорту, розважальних платформ, цифрових асистентів та багатьох інших інноваційних технологій.

З огляду на стрімкий розвиток, постійне вдосконалення архітектур і методів навчання, штучні нейронні мережі залишаються ключовою технологією в епоху інтелектуальних систем, демонструючи безпрецедентний потенціал у розв'язанні завдань підвищеної складності.

Рисунок 1.1 – Класифікація штучних нейронних мереж

Процес навчання штучної нейронної мережі (ШНМ) є ключовим етапом, що визначає здатність моделі ефективно вирішувати поставлені задачі. Цей процес охоплює низку взаємопов'язаних кроків, які забезпечують

адаптацію параметрів мережі – зокрема вагових коефіцієнтів – до характеру вхідних даних з метою досягнення максимальної точності прогнозів або класифікацій.

Початковим етапом є підготовка даних, що передбачає їх очищення, перетворення та нормалізацію. Зазвичай застосовується масштабування значень до діапазонів [0, 1] або [-1, 1], що сприяє стабільності обчислень та запобігає таким проблемам, як вибух або зникнення градієнтів. Також обов'язковим є розподіл даних на тренувальний, валідаційний і тестовий підмножини, які виконують різні функції: навчання, налаштування гіперпараметрів та фінальну оцінку здатності моделі до узагальнення.

Наступним критичним компонентом є вибір архітектури ШНМ. Вона визначається кількістю шарів, кількістю нейронів у кожному з них та типами зв'язків між елементами мережі. Наприклад, для аналізу візуальної інформації зазвичай використовуються згорткові нейронні мережі (CNN), які дозволяють автоматично виділяти релевантні ознаки. Натомість для обробки послідовностей перевагу надають рекурентним архітектурам (RNN, LSTM) або трансформерам, які забезпечують ефективну обробку залежностей у довгих ланцюжках даних.

Невід'ємною частиною навчального процесу є визначення функції втрат – метрики, що відображає ступінь розбіжності між передбаченнями моделі та реальними значеннями. У задачах класифікації типовим вибором є функція крос-ентропії, тоді як у регресійних задачах частіше використовують середньоквадратичну помилку. Обрана функція втрат слугує орієнтиром для оптимізаційного алгоритму, який модифікує ваги моделі з метою мінімізації помилки.

Оптимізація вагових коефіцієнтів відбувається за допомогою алгоритмів градієнтного спуску, серед яких стохастичний градієнтний спуск (SGD) є одним із базових. Для підвищення ефективності використовуються модифікації на кшталт Adam або RMSprop, які враховують адаптивну швидкість навчання для кожного параметра. Ці методи значно покращують

швидкість збіжності та стабільність результатів.

Регуляризація виступає важливим засобом боротьби з перенавчанням – ситуацією, коли модель демонструє високу точність на тренувальних даних, але слабко узагальнює на нові. До найпоширеніших методів регуляризації належать Dropout, що тимчасово "вимикає" частину нейронів під час тренування, а також L1 і L2 регуляризації, які вводять додаткові обмеження на ваги. Рання зупинка навчання за результатами валідаційної метрики також є ефективним інструментом.

Упродовж усього навчального циклу здійснюється моніторинг продуктивності моделі на валідаційному наборі, що дозволяє своєчасно коригувати гіперпараметри або зупинити тренування. Заключний етап передбачає тестування моделі на незалежному наборі даних для оцінки її здатності до узагальнення в реальних умовах.

Навчання ШНМ – це комплексний процес, що поєднує попередню обробку даних, архітектурне моделювання, вибір функції втрат, оптимізацію, регуляризацію та валідацію. Ретельне налаштування кожного з етапів є запорукою створення ефективної, стабільної та надійної моделі, здатної розв’язувати задачі підвищеної складності в різноманітних прикладних контекстах.

На рисунку 1.2 подано класифікацію нейронних мереж залежно від типу застосованої моделі.

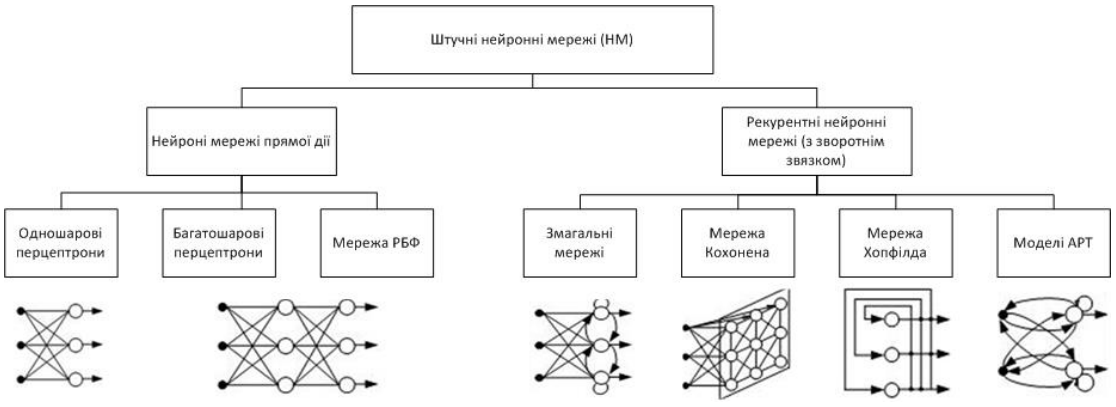


Рисунок 1.2 – Класифікація ШНМ

Рекурентні нейронні мережі (RNN) становлять окремий клас штучних

нейронних мереж, спеціально адаптованих для обробки послідовнісних даних. Їхня принципова відмінність від традиційних (feedforward) моделей полягає у здатності зберігати та використовувати інформацію про попередні стани, що забезпечує аналіз контексту у часово впорядкованих послідовностях. Така властивість робить RNN ефективними для вирішення завдань, у яких критичну роль відіграє порядок елементів, зокрема в обробці природної мови, розпізнаванні мовлення, аналізі часових рядів і машинному перекладі.

Архітектура RNN передбачає наявність зворотних зв'язків між елементами мережі, що дозволяє інформації циркулювати в системі, формуючи тимчасову «пам'ять». На кожному часовому кроці модель отримує не лише нові вхідні дані, а й приховане представлення з попереднього кроку, що забезпечує збереження історії обчислень. Однак класичні RNN мають низку обмежень, зокрема схильність до проблеми зникнення або вибуху градієнтів під час навчання, що ускладнює збереження інформації на великих часових відстанях.

Для подолання цих недоліків було запропоновано вдосконалені архітектури, зокрема довготривалу короткострокову пам'ять (LSTM) та мережі з керованими воротами (GRU). LSTM-моделі інтегрують спеціальні комірки пам'яті й систему воріт (вхідні, забування, вихідні), які забезпечують гнучке управління інформаційними потоками, дозволяючи зберігати релевантні дані протягом тривалих періодів. GRU-мережі, своєю чергою, є спрощеним варіантом LSTM, у якому зменшено кількість параметрів шляхом об'єднання деяких воріт, що сприяє швидшому навчанню за збереження основної функціональності.

RNN знаходять численні застосування в задачах, де важливо враховувати послідовність елементів. У сфері обробки природної мови вони ефективно реалізуються для машинного перекладу, генерації тексту, аналізу тональності та розпізнавання мовлення. Наприклад, у системах перекладу контекст усього речення дозволяє формувати граматично та семантично

коректні відповідники. У генеративних застосуваннях RNN здатні створювати зв’язні текстові фрагменти, а в задачах тонального аналізу – враховувати не лише лексичні одиниці, а й їх взаємозв’язки в межах повідомлення.

У галузі обробки аудіосигналів RNN дозволяють здійснювати розпізнавання мовлення в режимі реального часу, перетворюючи звукову інформацію на текст. У фінансовому секторі та прогнозуванні часових рядів вони застосовуються для моделювання динаміки змін економічних показників на основі історичних даних.

Попри те, що трансформерні моделі, які використовують механізм самоуваги, у багатьох випадках витіснили RNN у сфері обробки природної мови, останні все ще залишаються релевантними в задачах, де необхідна покрокова обробка даних та збереження довготривалої залежності.

Рекурентні нейронні мережі продовжують активно розвиватися, демонструючи значний потенціал у розв’язанні широкого кола завдань, пов’язаних із часовими та послідовними структурами даних, і залишаються важливою складовою арсеналу сучасного штучного інтелекту.

1.3 Біологічний прототип нейрона

Біологічний нейрон є ключовою структурною та функціональною одиницею нервової системи, яка забезпечує обробку, інтеграцію та передачу інформації в організмі. Його будова включає низку спеціалізованих елементів: дендрити, сому (тіло клітини), аксонний горбик, аксон і синапси, кожен з яких виконує специфічну роль у нейрональній активності.

Дендрити, як численні розгалужені відростки, виступають основними рецепторами вхідних сигналів від інших клітин. Вони сприймають електрохімічні імпульси через синаптичні з’єднання і передають їх до тіла нейрона. Високий ступінь розгалуження дендритів дозволяє нейрону інтегрувати інформацію від великої кількості джерел.

Сома, або тіло нейрона, містить ядро клітини та органели, необхідні для її метаболічного забезпечення. У сомі відбувається підсумовування сигналів, що надходять через дендрити. Якщо сумарна активація перевищує порогове значення, генерується вихідний потенціал дії.

Аксонний горбик є критичною зоною на межі між сомою та аксоном. Саме тут ініціюється потенціал дії – електричний сигнал, який далі передається по аксону. Цей процес відбувається у разі досягнення збудження, необхідного для активації натрієвих каналів.

Аксон – це довгий відросток, що слугує шляхом передачі електричного імпульсу від тіла нейрона до термінальних структур. Його ефективність забезпечується мієліновою оболонкою, яка, завдяки своїй ізоляційній здатності, значно підвищує швидкість передачі сигналу через механізм сальтаторного проведення – стрибкоподібного переміщення імпульсу між перехватами Ранв'є.

Синапси є кінцевими елементами аксона, через які нейрон передає сигнал до інших нейронів, м'язових чи залозистих клітин. Структурно синапс включає пресинаптичну мембрану, синаптичну щілину та постсинаптичну мембрану. Передача сигналу здійснюється за допомогою нейромедіаторів – хімічних речовин, які вивільняються у відповідь на електричний імпульс та активують рецептори на постсинаптичній клітині, спричиняючи зміну її мембранного потенціалу.

Функціонування нейронів тісно пов'язане з дією нейромедіаторів, серед яких особливу роль відіграють глутамат, ГАМК, дофамін і серотонін. Їхній вплив на постсинаптичний нейрон може бути як збуджувальним, так і гальмівним, що забезпечує пластичність нервової системи та точну регуляцію її реакцій на зовнішні та внутрішні стимули.

Таким чином, біологічний нейрон функціонує як високоспеціалізований інформаційний інтегратор та передавач, забезпечуючи координацію діяльності організму на різних рівнях. Завдяки комплексній взаємодії його структурних елементів та хімічних процесів,

нервова система виконує широкий спектр завдань – від базових рефлекторних реакцій до складної когнітивної діяльності.

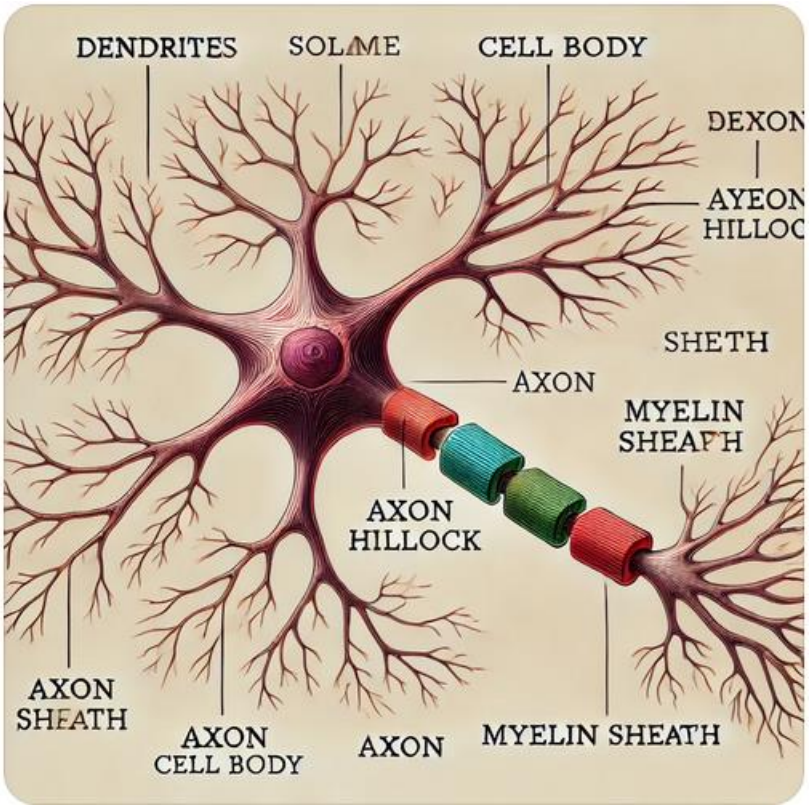


Рисунок 1.3 – Схема біологічного нейрона

1.4 Штучний нейрон

Штучний нейрон є базовим елементом архітектури штучних нейронних мереж і побудований за аналогією до біологічного нейрона. Його призначення полягає у прийомі, перетворенні та передачі сигналів, що забезпечує процес обчислення всередині моделі. Структурно штучний нейрон включає декілька функціональних компонентів: вхідні зважені сигнали, сумуючий блок (суматор), активаційну функцію та генерацію вихідного сигналу.

На першому етапі нейрон приймає числові значення з вхідного простору, кожне з яких помножується на відповідну вагу – коефіцієнт, що відображає значущість цього входу. Далі всі зважені сигнали

підсумовуються, і результуюче значення подається на активаційну функцію. Ця функція виконує роль нелінійного перетворення, завдяки чому мережа здатна моделювати складні залежності у даних. Вихід нейрона, сформований після активації, передається на наступний шар мережі або використовується як фінальний результат моделі.

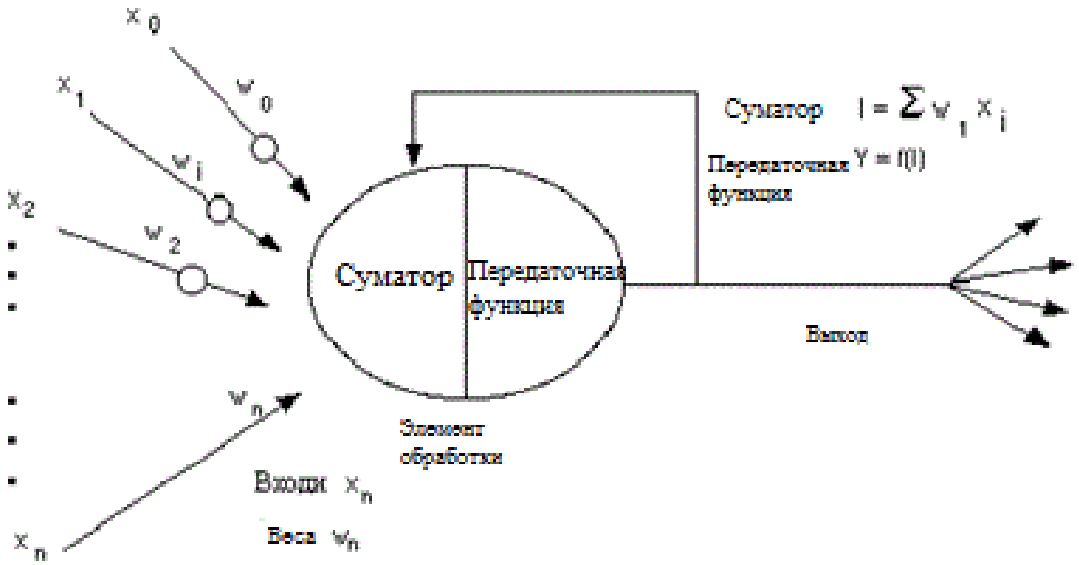


Рисунок 1.4 – Базовий штучний нейрон

1.5 Функції активації

Активаційні функції становлять невід’ємну складову штучних нейронних мереж, оскільки саме вони визначають спосіб обчислення вихідного сигналу нейрона на основі суми вхідних зважених значень. Їх головне призначення полягає у внесенні нелінійності в процес обробки даних, що дозволяє моделі ефективно апроксимувати складні функціональні залежності та вирішувати широкий спектр задач, які не піддаються розв’язанню за допомогою лише лінійних перетворень.

Однією з класичних активаційних функцій є сигмоїда, яка трансформує вхідне дійсне число у значення з інтервалу (0, 1). Її характерна S-подібна крива робить цю функцію особливо зручною для задач, що потребують інтерпретації виходу як ймовірності, зокрема у двійковій класифікації.

До категорії частково-лінійних активацій належать функції, які поєднують лінійні й нелінійні області у своїй структурі. Найбільш поширеною серед них є функція випрямленого лінійного блоку (ReLU, Rectified Linear Unit). Вона характеризується тим, що для додатних вхідних значень функціонує як лінійна, а для від'ємних – приймає нульове значення. Такий підхід сприяє простоті обчислень, зменшенню проблеми зникнення градієнтів та пришвидшенню навчання моделей.

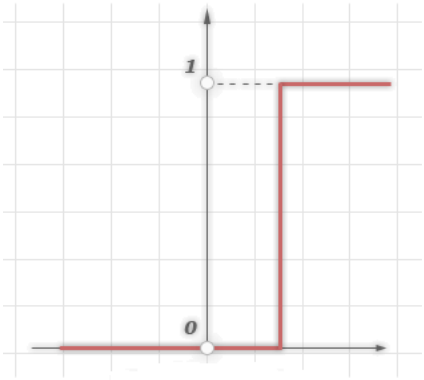


Рисунок 1.5 – Частково-лінійна функція

Серед частково-лінійних активаційних функцій найбільшою популярністю користується ReLU (Rectified Linear Unit), яка завдяки своїй простоті та ефективності стала стандартом у багатьох архітектурах нейронних мереж. Її головною перевагою є здатність забезпечувати лінійне зростання для додатних значень та нульову відповідь для від'ємних, що пришвидшує процес навчання та зменшує ризик зникнення градієнтів. Однак використання ReLU може призводити до появи «мертвих» нейронів – таких, що втрачають здатність до активації внаслідок постійно негативних входів.

Для подолання цього недоліку було запропоновано низку модифікацій, серед яких особливої уваги заслуговує функція Leaky ReLU. Вона відрізняється від класичної ReLU тим, що для від'ємних значень повертає невелике, але ненульове значення. Така властивість дозволяє підтримувати активацію навіть у випадках, коли сигнал має негативне значення, що сприяє збереженню повноцінної участі нейрона у процесі навчання.

Частково-лінійні функції активації характеризуються низькою обчислювальною складністю, що робить їх придатними для реалізації у великих моделях. Їх використання сприяє прискоренню збіжності мережі та дозволяє уникнути деяких типових труднощів, зокрема пов’язаних із градієнтним затуханням. Водночас, попри ці переваги, вони не є універсальними: надто великі значення ваг можуть спричинити нестабільність у вигляді вибуху градієнтів, що негативно позначається на навчальному процесі.

Незважаючи на певні обмеження, частково-лінійні функції залишаються важливим компонентом у проєктуванні глибоких нейронних мереж і знаходять широке застосування в сучасних системах машинного навчання завдяки оптимальному балансу між простотою реалізації та високою продуктивністю.

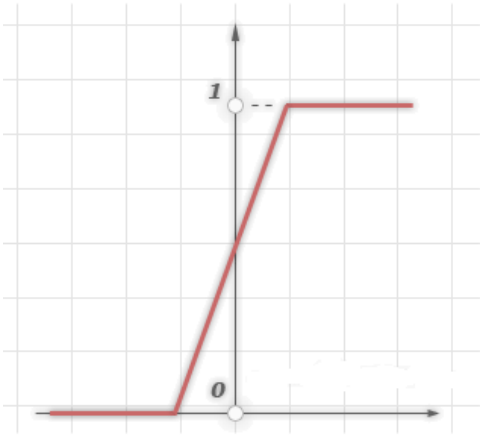


Рисунок 1.6 – Нескладна частково-лінійна функція

ReLU (Rectified Linear Unit) є однією з найпростіших частково-лінійних функцій активації, що вирізняється надзвичайною обчислювальною ефективністю завдяки використанню лише базових математичних операцій. Вона вводить у модель необхідну нелінійність, що дозволяє нейронним мережам апроксимувати складні функціональні залежності у вхідних даних. Її лінійна поведінка на додатній піввісі запобігає зникненню градієнтів, сприяючи швидкій та стабільній збіжності мережі під час навчання. Саме ця комбінація простоти реалізації та здатності до ефективної генералізації

зумовлює її поширення в більшості сучасних архітектур машинного навчання.

Водночас логістична функція, або сигмоїда, є іншим популярним видом активації, яка перетворює довільні вхідні значення на вихід у межах інтервалу від 0 до 1. Ця властивість робить її надзвичайно корисною у задачах, що передбачають оцінку ймовірності належності до певного класу, зокрема в контексті бінарної класифікації. Вона дозволяє інтерпретувати вихід моделі як ймовірнісну міру, що є важливим у статистичних та ймовірнісних підходах до машинного навчання.

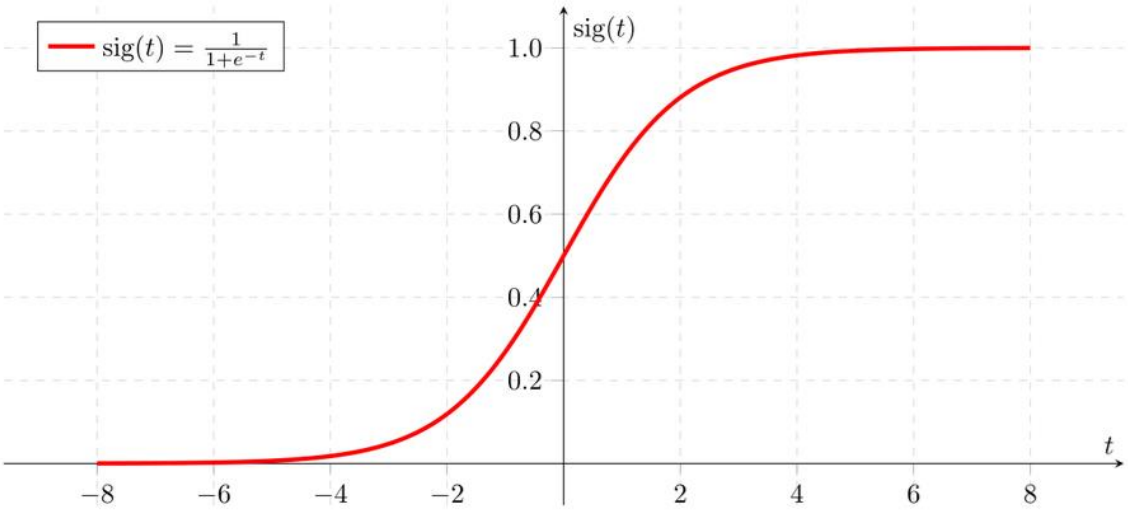


Рисунок 1.7 – Логістична функція (уніполярна)

Логістична функція, або сигмоїда, є однією з базових активаційних функцій, яка широко використовується в нейронних мережах, особливо у задачах бінарної класифікації. Її головною властивістю є строго зростаюча залежність: з підвищенням вхідного значення відповідно зростає й вихідне значення. Завдяки своїй характерній S-подібній формі логістична функція здатна моделювати складні нелінійні взаємозв’язки між змінними. Вона є гладкою, тобто має безперервні перші та другі похідні, що є критично важливим для застосування диференційованих методів оптимізації, зокрема градієнтного спуску.

У контексті нейронних мереж логістична функція часто використовується на вихідному шарі моделі для інтерпретації результату у вигляді ймовірності належності до певного класу. У таких випадках вихідний сигнал трактується як ймовірність, а остаточне рішення приймається шляхом порівняння з фіксованим порогом (наприклад, 0.5). Важливою особливістю є те, що похідна логістичної функції безпосередньо застосовується в процесі оновлення ваг під час навчання, що забезпечує адаптивність моделі.

Попри численні переваги, логістична функція має й певні обмеження. Зокрема, при дуже великих або дуже малих вхідних значеннях її градієнт прямує до нуля, що може спричинити проблему зникнення градієнтів. Це негативно впливає на ефективність навчання, особливо у глибоких мережах. Проте завдяки своїй здатності відображати ймовірнісну інтерпретацію та моделювати нелінійність, вона залишається важливим інструментом у побудові нейронних моделей.

Альтернативною активаційною функцією є гіперболічний тангенс (tanh), який має схожу S-подібну форму, але повертає значення у межах від -1 до 1. Завдяки тому, що вихід цієї функції центрований відносно нуля, tanh забезпечує кращу симетрію у поширенні сигналу по мережі. Це сприяє швидшій збіжності моделей, особливо на ранніх етапах навчання. Як і у випадку з сигмоїдою, функція є строго зростаючою, має гладкі похідні та широко застосовується в алгоритмах градієнтної оптимізації для оновлення вагових коефіцієнтів. Відмінність полягає також у тому, що tanh дозволяє нейронам генерувати як позитивні, так і негативні значення, що додає гнучкості в моделюванні складних залежностей.

2 МЕТОДИ КЛАСТЕРИЗАЦІЇ ДАНИХ

2.1 Поняття кластеризаціїї

Кластеризація є одним із фундаментальних методів аналізу даних, який дозволяє виявляти приховані структури у вибірках шляхом об’єднання об’єктів у групи за принципом подібності. Основна ідея полягає в тому, щоб сформувати такі кластери, у межах яких об’єкти були б максимально подібними між собою, а міжкласова відмінність, навпаки, була якомога більшою. Цей підхід особливо цінний у випадках, коли попередня інформація про структуру даних відсутня, що робить кластеризацію ефективним інструментом для дослідження та виявлення закономірностей у невідомих наборах даних.

На початку процесу кластеризації здійснюється формалізація поняття подібності між об’єктами. Для цього застосовуються різноманітні метрики відстані, такі як евклідова, манхеттенська чи косинусна. Конкретний вибір метрики обумовлюється характером даних та метою дослідження. Розраховані відстані формують основу для подальшого розподілу об’єктів на групи відповідно до обраного алгоритму.

Серед найбільш уживаних методів кластеризації вирізняється алгоритм K-середніх, який передбачає попереднє визначення кількості кластерів. Центри кластерів ініціалізуються випадковим чином, після чого кожен об’єкт асоціюється з найближчим до нього центром. Далі центри перераховуються як середнє значення координат об’єктів, що потрапили до відповідного кластеру. Цей процес ітеративно повторюється до досягнення стабільного стану, коли положення центрів перестає змінюватися.

Ієрархічний підхід до кластеризації, на відміну від K-середніх, не потребує попередньої фіксації кількості кластерів. Він реалізується у вигляді дендрограми, що відображає поступове злиття або поділ груп об’єктів.

Ієрархічна кластеризація може бути реалізована агломеративно, коли злиття починається з одиничних об’єктів, або дивізивно – шляхом поступового розбиття одного великого кластеру.

Методи кластеризації на основі щільності, як-от DBSCAN, визначають кластери як області простору з високою концентрацією об’єктів, відокремлені зонами низької щільності. Цей підхід дозволяє моделювати кластери складної форми та виділяти шуми, які не належать жодному кластеру, що особливо корисно в реальних наборах даних із великою варіативністю.

Завдяки своїй універсальності кластеризація активно використовується в багатьох практичних контекстах. У сфері маркетингу вона слугує для сегментації клієнтів, у біології – для аналізу генетичної інформації та класифікації біомолекул. У соціальних мережах кластеризація допомагає ідентифікувати спільноти на основі поведінкових характеристик, а в комп’ютерному зорі застосовується для розпізнавання структур на зображеннях, таких як об’єкти чи текстури.

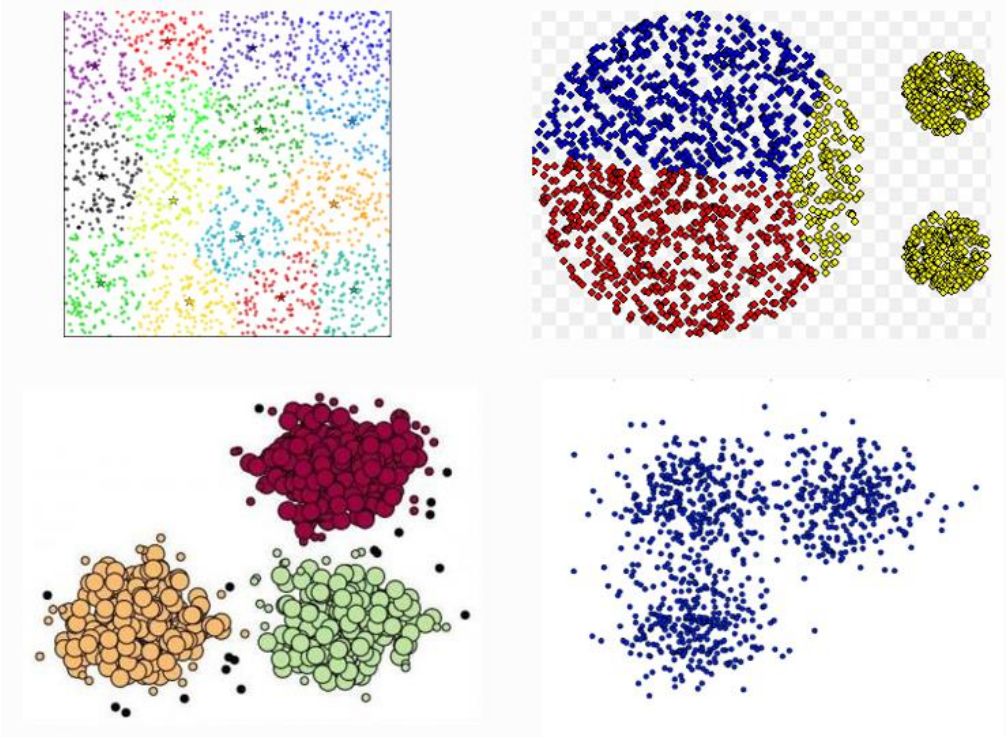


Рисунок 2.1 – Кластеризація

Процедура кластерного аналізу нерідко передбачає проведення етапу попередньої обробки даних, що є критично важливим для забезпечення точності та надійності подальших результатів. До таких підготовчих заходів належать нормалізація або стандартизація ознак, що дозволяє усунути масштабні відмінності між змінними, а також очищення вибірки від шуму та викидів, які можуть суттєво впливати на результат кластеризації, особливо при використанні метрик відстані.

Після завершення кластеризації важливим завданням є оцінювання якості отриманого розподілу даних. У випадках, коли апріорна інформація про структуру даних відсутня, застосовуються внутрішні критерії, зокрема коефіцієнт силуету, що дозволяє оцінити ступінь відповідності об’єктів своєму кластеру у порівнянні з іншими. Якщо ж існує можливість порівняння з відомими еталонними групами, використовуються зовнішні метрики, які дозволяють більш об’єктивно оцінити точність кластеризації.

Загалом, кластерний аналіз виступає потужним інструментом у дослідницькому арсеналі, що сприяє виявленню прихованих патернів і структур у даних. Завдяки здатності оперувати без попередньої інформації про класи, цей метод дозволяє здійснювати глибинне дослідження складних і неоднорідних наборів даних, відкривати нові знання та підтримувати прийняття рішень на основі емпірично отриманих кластерів.

2.2 Етапи кластерного аналізу

Кластерний аналіз є поетапним процесом, який охоплює всі ключові складові від попередньої обробки даних до інтерпретації отриманих результатів. Першим кроком виступає підготовка даних, що передбачає збирання, очищення та трансформацію інформації для подальшого аналізу. На цьому етапі зазвичай усуваються пропущені значення, виконується нормалізація ознак з метою уніфікації масштабів змінних, а також, за необхідності, здійснюється зменшення розмірності, що дозволяє знизити

складність обчислень без втрати важливої інформації.

Після підготовки здійснюється вибір методу кластеризації, який визначається особливостями даних і аналітичними цілями. Наприклад, метод K-середніх доцільно використовувати, коли кількість кластерів відома наперед, а самі кластери мають приблизно сферичну форму. У випадках, коли структура даних є складнішою або неоднорідною, доцільним може бути використання ієрархічної кластеризації, яка не потребує фіксації кількості груп і дозволяє виявляти кластери довільної форми.

На подальшому етапі визначається метрика відстані, яка використовується для оцінки ступеня подібності між об’єктами. Найпоширенішими є евклідова, манхеттенська відстань та косинусна схожість. Вибір метрики залежить від характеру вхідних даних: числових, категоріальних чи змішаних, а також від того, як саме слід інтерпретувати близькість між об’єктами в контексті дослідження.

Виконання кластеризації включає застосування обраного алгоритму до нормалізованого набору даних. Алгоритм розподіляє об’єкти на групи, виходячи з їхньої внутрішньої схожості. У випадку K-середніх може додатково застосовуватися процедура ітеративного оновлення центрів кластерів до досягнення стійкої конфігурації.

Завершальним етапом є інтерпретація результатів кластерного аналізу. Вона передбачає змістове осмислення отриманих кластерів і перевірку їх відповідності реальним закономірностям у даних. Для цього використовуються засоби візуалізації, зокрема графіки розсіювання або дендрограми, а також методи кількісної оцінки, зокрема коефіцієнт силуету, який відображає якість формування кластерів з огляду на їхню компактність і роздільність.

Таким чином, кластерний аналіз є комплексною аналітичною процедурою, яка поєднує в собі технічну обробку даних, алгоритмічну реалізацію кластеризації та аналітичне осмислення результатів, спрямоване на виявлення прихованих структур у даних і побудову нових гіпотез для

подальших досліджень.

2.3 Класичний аппарат карт Кохонена

Класичний підхід до побудови карт Кохонена, також відомих як самонавчальні карти (Self-Organizing Maps, SOM), належить до методів штучних нейронних мереж, які реалізують механізм конкурентного навчання з метою зниження розмірності даних та виявлення прихованих структур у багатовимірному просторі. Ця концепція була запропонована Тейво Кохоненом у 1980-х роках і відтоді здобула широке визнання як ефективний інструмент для кластеризації, візуалізації та інтерпретації складних даних.

Самонавчальна карта являє собою двовимірну регулярну сітку нейронів, де кожен нейрон характеризується власним вектором ваг, розмірність якого відповідає кількості ознак у вхідному векторі. Просторова організація нейронів на карті визначається їхніми координатами у вигляді сітки, наприклад (i, j), що забезпечує топологічну впорядкованість елементів мережі. Початкове значення вагових коефіцієнтів встановлюється або випадковим чином, або з використанням малих стохастичних відхилень від середнього значення вхідних даних, що дозволяє забезпечити необхідну варіативність для процесу навчання.

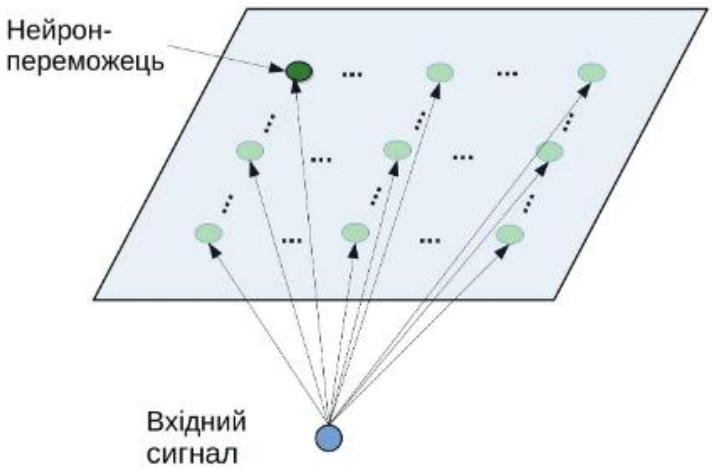


Рисунок 2.2 – Структура карт Кохонена

Процес навчання карт Кохонена є ітеративним та охоплює низку послідовних кроків, що повторюються протягом багатьох циклів. На кожній ітерації мережа отримує випадково вибраний вхідний вектор із навчальної вибірки. Цей вектор порівнюється з ваговими векторами усіх нейронів, розташованих на двовимірній решітці карти, з метою визначення так званого нейрона-переможця (Best Matching Unit, BMU). Визначення здійснюється шляхом обчислення відстані між вхідним вектором і кожним з вагових векторів, причому найчастіше використовується евклідова метрика як критерій подібності. Нейроном-переможцем вважається той, чий ваговий вектор має найменшу відстань до поданого вектора.

Після ідентифікації BMU відбувається адаптація його вагових параметрів, а також ваг сусідніх до нього нейронів. Оновлення виконується таким чином, щоб ці вектори стали ближчими до вхідного сигналу. При цьому ступінь оновлення залежить від відстані між нейроном і переможцем у межах топології карти, а також від значення функції навчання, яка зменшується з часом. Таким чином, на ранніх етапах навчання адаптація охоплює ширшу зону, тоді як з наближенням до завершення процесу вона стає більш локалізованою. Такий механізм дозволяє формувати впорядковане відображення багатовимірного простору даних у двовимірному форматі карти, зберігаючи топологічні відношення між об’єктами.

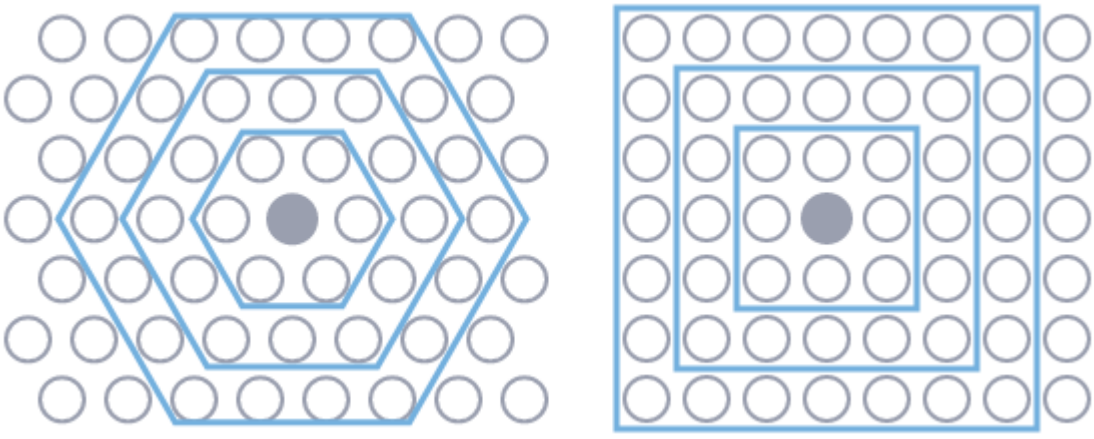


Рисунок 2.3 – Решітка карти Кохонена

Процес навчання карт Кохонена триває протягом значної кількості ітерацій і завершується тоді, коли або досягається стабілізація вагових коефіцієнтів, або вичерпується заздалегідь задана кількість циклів. Протягом усього періоду навчання відбувається поступове зниження як коефіцієнта навчання, так і радіуса сусідства. Це забезпечує спочатку грубу адаптацію топології карти до структури вхідних даних, а згодом – її точне налаштування на фінальних етапах навчального процесу.

Після завершення навчання карта Кохонена може бути ефективно використана для перетворення багатовимірних даних у двовимірне представлення, що значно полегшує їх візуалізацію. Кожен вхідний вектор проєктується на поверхню карти через нейрон-переможець, що дозволяє чітко ідентифікувати групи схожих об’єктів. Така проєкція відкриває можливість для виявлення кластерів, патернів і взаємозв’язків, які могли б залишитися непоміченими у багатовимірному просторі.

Карти Кохонена отримали широке застосування в численних галузях науки і техніки. Вони ефективно використовуються у кластерному аналізі, візуалізації складних структур даних, зниженні розмірності, обробці сигналів, розпізнаванні образів, а також в інших сферах, де важливо представити високорозмірні дані у зрозумілому вигляді. Особливо актуальним є їх використання у випадках, коли обсяг даних є великим, а їх структура – складною й неочевидною.

Таким чином, класичний апарат самонавчальних карт Кохонена є дієвим інструментом для самоорганізації та аналізу даних, який, завдяки здатності до збереження топології та ефективної візуалізації, зберігає свою актуальність і практичну цінність у розв’язанні сучасних задач інтелектуального аналізу даних.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗГЛЯНУТИХ МЕТОДІВ

3.1 Вибір програмних засобів

Вибір інструментального середовища, що включає мову програмування Python та бібліотеки Pandas, Matplotlib, NumPy, Scikit-learn і Orange, обґрунтовується їх високою функціональністю, продуктивністю, зручністю у використанні, а також активною підтримкою спільноти розробників. Python виступає як універсальна мова програмування з потужною екосистемою для наукових обчислень і аналізу даних. Його синтаксис є простим і читабельним, що значно спрощує створення прототипів і реалізацію складних алгоритмічних рішень, особливо в контексті обробки великих даних та реалізації методів кластеризації.

Бібліотека Pandas є одним з основних інструментів для роботи з табличними структурами даних. Вона забезпечує високу гнучкість при виконанні типових завдань підготовки даних, включаючи очищення, перетворення, фільтрацію та агрегацію. Pandas дозволяє ефективно обробляти великі обсяги інформації, що є передумовою для якісного попереднього аналізу перед проведенням кластеризації.

Matplotlib, як одна з ключових бібліотек для візуалізації, надає широкі можливості для побудови графіків, діаграм і візуальних інтерпретацій результатів кластерного аналізу. Завдяки високій точності графічного представлення, дослідник має змогу краще розуміти структуру даних, виявляти приховані патерни і закономірності, а також перевіряти коректність виконаних обчислень.

NumPy є основою для чисельних обчислень у Python і забезпечує підтримку ефективних операцій з багатовимірними масивами, що є критично важливим для реалізації алгоритмів кластеризації. Завдяки широкому набору вбудованих математичних функцій, включаючи засоби для обчислення

відстаней, операцій лінійної алгебри та статистичних розрахунків, NumPy відіграє ключову роль у побудові продуктивних алгоритмічних рішень.

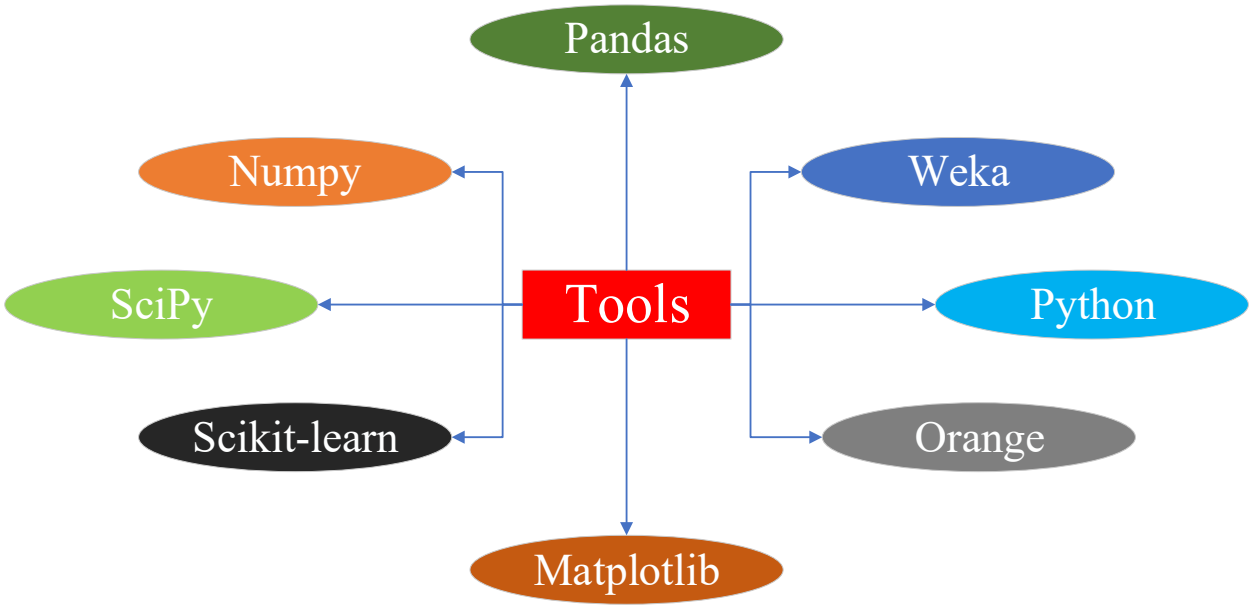


Рисунок 3.1 – Вибір ПЗ

Scikit-learn посідає провідне місце серед бібліотек машинного навчання в середовищі Python завдяки широкому спектру реалізованих алгоритмів, які охоплюють задачі класифікації, регресії та кластеризації. Інтерфейс цієї бібліотеки вирізняється простотою та зручністю, що дозволяє дослідникам швидко впроваджувати, тестувати й порівнювати різні підходи до кластерного аналізу, зокрема метод K-середніх, ієрархічну кластеризацію та алгоритм DBSCAN. Крім того, Scikit-learn забезпечує потужний інструментарій для попередньої обробки даних, оцінювання ефективності моделей та їх подальшої візуалізації, що робить її незамінною у реалізації повного циклу машинного навчання.

Orange, у свою чергу, є гнучким інструментом для візуального аналізу даних, який надає користувачам можливість створення аналітичних робочих процесів за допомогою інтуїтивно зрозумілого графічного інтерфейсу. Це особливо корисно для фахівців, які не мають глибоких навичок

програмування, проте прагнуть експериментувати з різноманітними методами кластеризації, візуалізації та іншими алгоритмами штучного інтелекту. Зручність у використанні, а також підтримка інтерактивної роботи з даними, робить Orange ефективним інструментом для прикладного аналізу.

У сукупності використання Python з бібліотеками Pandas, Matplotlib, NumPy, Scikit-learn та середовищем Orange формує повноцінну інфраструктуру для реалізації сучасних програмних засобів кластеризації. Такий набір інструментів дає змогу здійснювати повний спектр операцій – від попередньої підготовки даних до їх глибокого аналізу та інтерпретації, відкриваючи широкі можливості для виявлення складних структур і закономірностей у великих обсягах інформації.

3.2 Реалізація в Google Colab

Код реалізації представлений в додатку Б.

```
# Кластеризація з використанням нейронної мережі та візуалізацій
!pip install umap-learn --quiet

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.datasets import load_digits
from sklearn.metrics import normalized_mutual_info_score, adjusted_rand_score
import tensorflow as tf
from tensorflow.keras import layers, models
import umap

# Завантаження та нормалізація даних
data = load_digits()
X = data.data
y = data.target
X = X / 16.0

# Побудова автоенкодера
input_dim = X.shape[1]
encoding_dim = 10

input_layer = layers.Input(shape=(input_dim,))
encoded = layers.Dense(64, activation='relu')(input_layer)
encoded = layers.Dense(32, activation='relu')(encoded)
latent = layers.Dense(encoding_dim, activation='relu')(encoded)
```

Рисунок 3.2 – Фрагмент коду реалізації

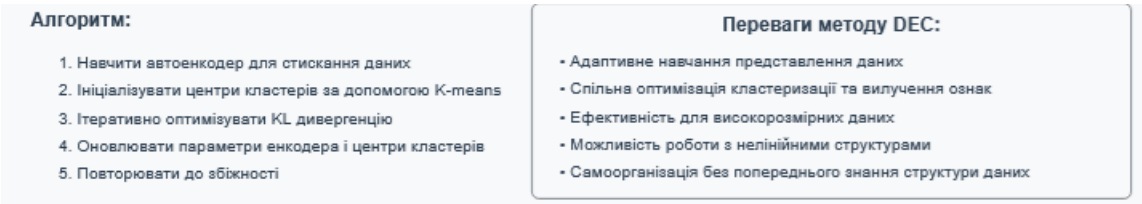


Рисунок 3.3 – Алгоритм використовуваного методу

На рисунку 3.2 представлений фрагмент реалізації коду в Colab.

На рисунку 3.3 алгоритм використовуваного методу.

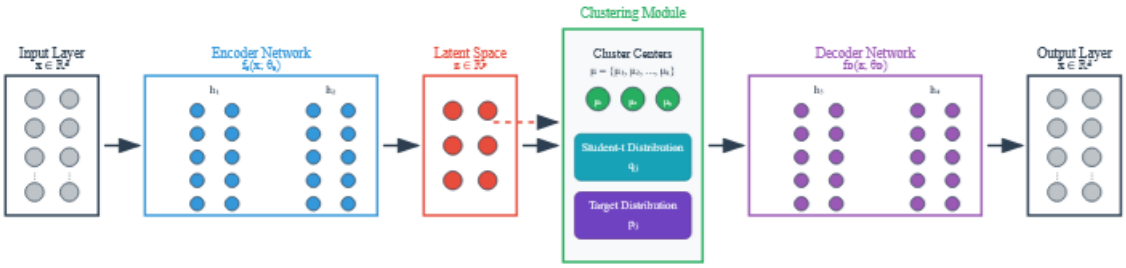


Рисунок 3.4 – Алгоритм DEC

Рисунок 3.4 являє собою візуалізацію цього алгоритму.

3.2 Аналіз результатів роботи

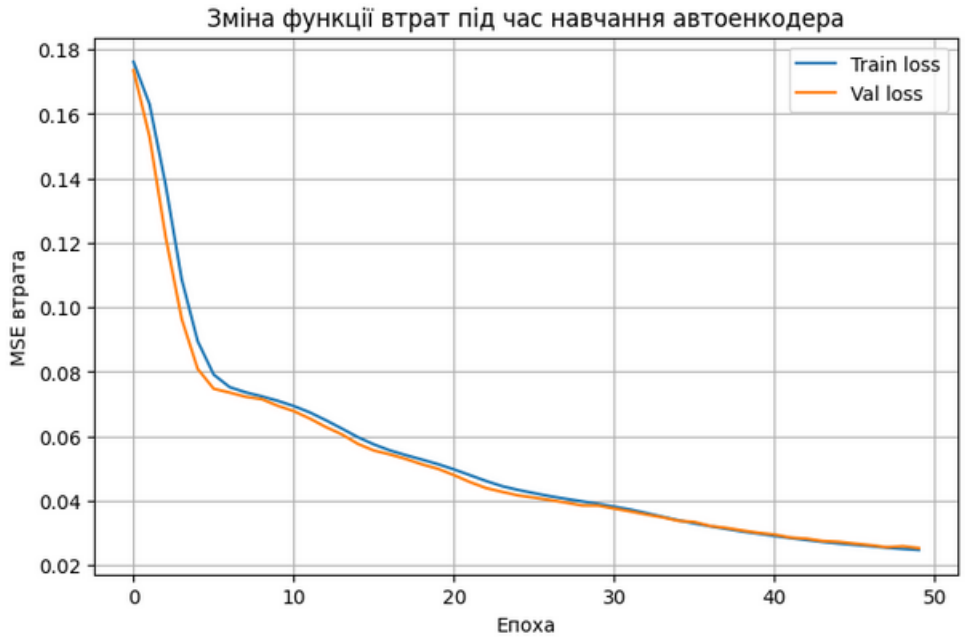


Рисунок 3.5 – Результати роботи

На рисунку 3.5 представлено графік зміни функції втрат під час навчання автоенкодера. По осі абсцис відкладено кількість епох, а по осі ординат – значення середньоквадратичної помилки (MSE). Дві криві відображають динаміку втрат на тренувальній (Train loss) та валідаційній (Val loss) вибірках. Спостерігається поступове зниження обох кривих, що свідчить про ефективне навчання моделі та відсутність значного перенавчання.

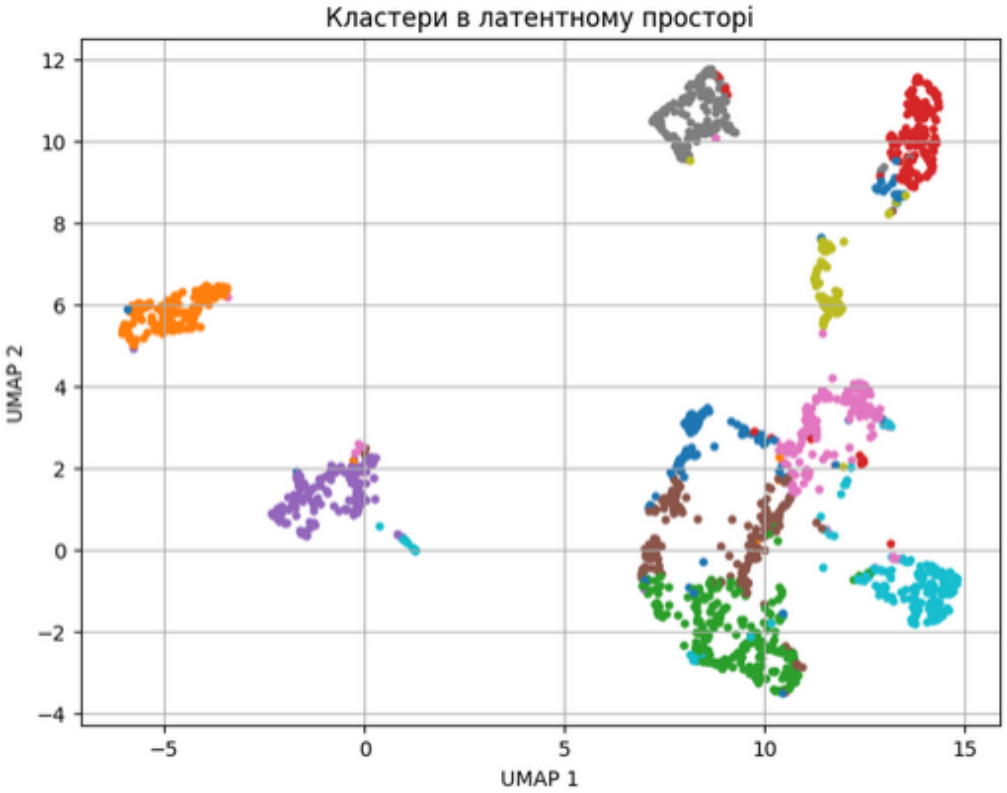


Рисунок 3.6 – Результати роботи

На рисунку 3.6 зображено результат кластеризації даних у латентному просторі, зниженому до двох вимірів за допомогою методу UMAP. Кожна крапка відповідає одному об'єкту з вибірки, а різні кольори позначають належність до окремих кластерів. Така візуалізація дозволяє оцінити структуру та розподіл даних після їхньої трансформації автоенкодером, а також якість кластеризації у зниженому просторі ознак.

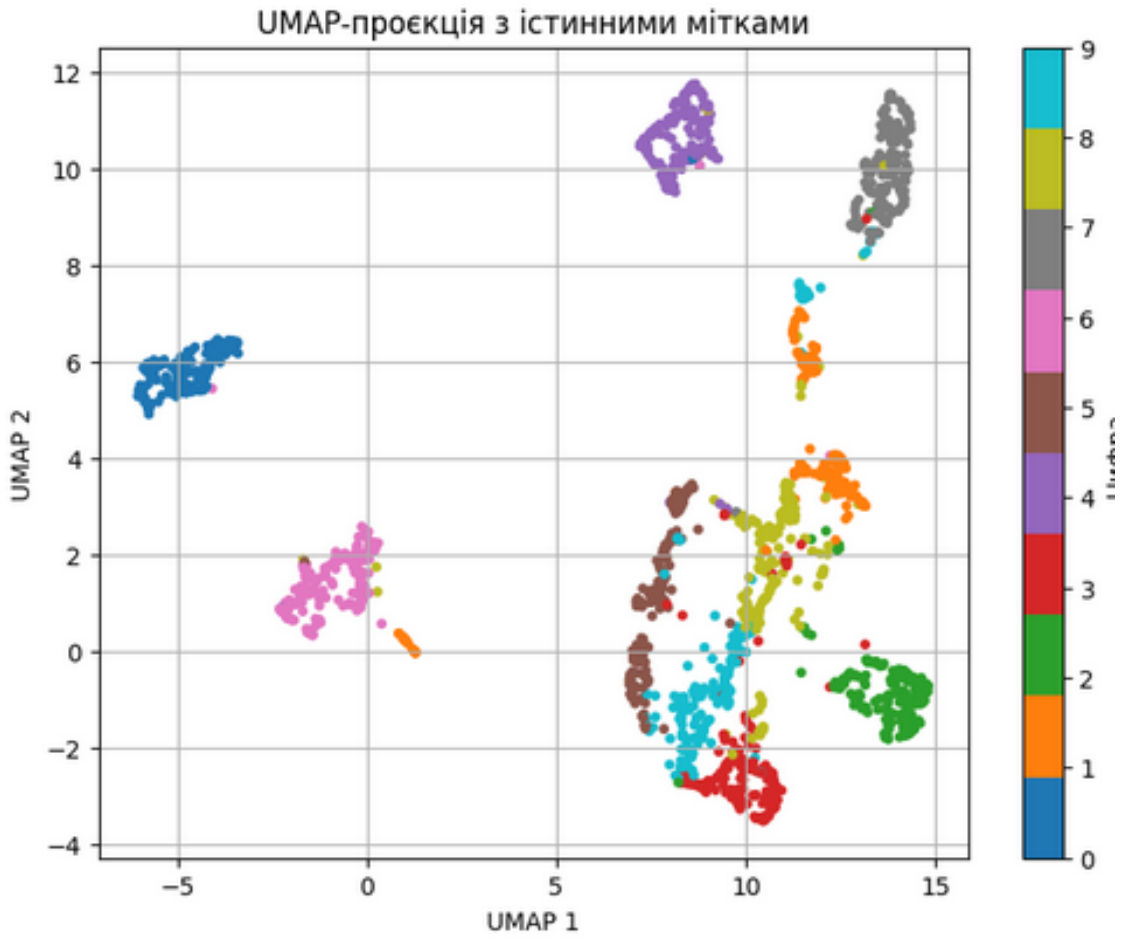


Рисунок 3.7 – Результати роботи

На рисунку 3.7 представлено UMAP-проєкцію багатовимірних даних у двовимірний простір з урахуванням істинних міток класів. Кожна точка відповідає об’єкту, а колір позначає класову належність відповідно до реальних (апріорних) міток. Така візуалізація дозволяє оцінити, наскільки добре класи розділені в латентному просторі та чи є перетин між класами після проєкції.

На рисунку 3.8 зображено результат кластеризації у просторі, зменшеному за допомогою UMAP, з використанням кластерних міток. Кожна точка представляє об’єкт, а колір вказує на кластер, до якого цей об’єкт було віднесено алгоритмом кластеризації. Графік демонструє, як об'єкти групуються відповідно до знайдених кластерів, що дозволяє візуально оцінити якість кластеризації та її відповідність структурі даних у латентному просторі.

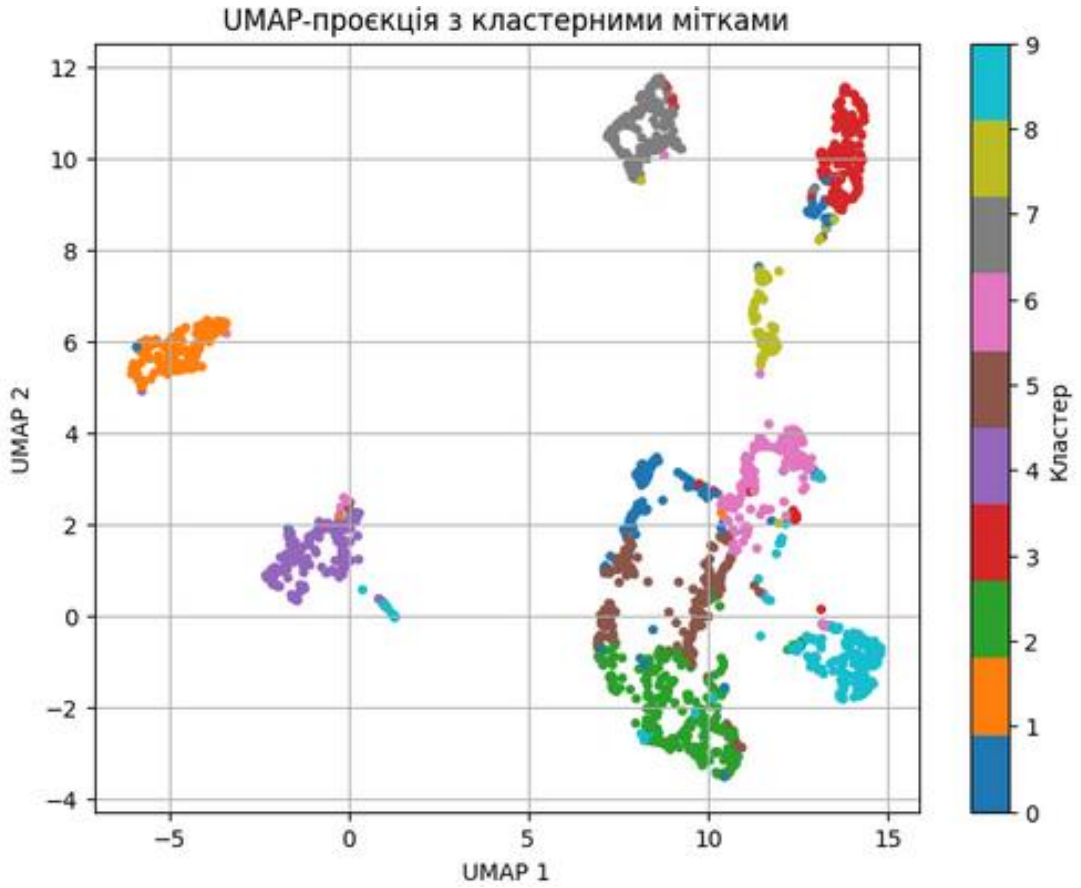


Рисунок 3.8 – Результати роботи

На рисунку 3.8 зображено результат кластеризації у просторі, зменшеному за допомогою UMAP, з використанням кластерних міток. Кожна точка представляє об’єкт, а колір вказує на кластер, до якого цей об’єкт було віднесено алгоритмом кластеризації. Графік демонструє, як об'єкти групуються відповідно до знайдених кластерів, що дозволяє візуально оцінити якість кластеризації та її відповідність структурі даних у латентному просторі.

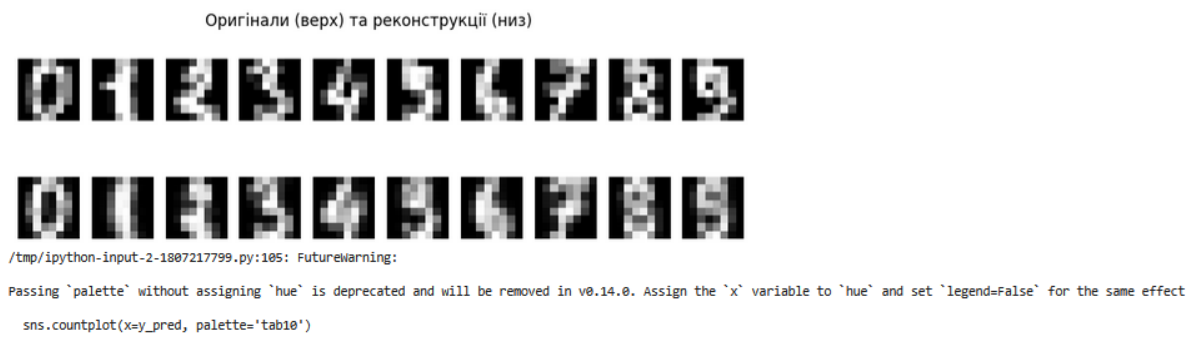
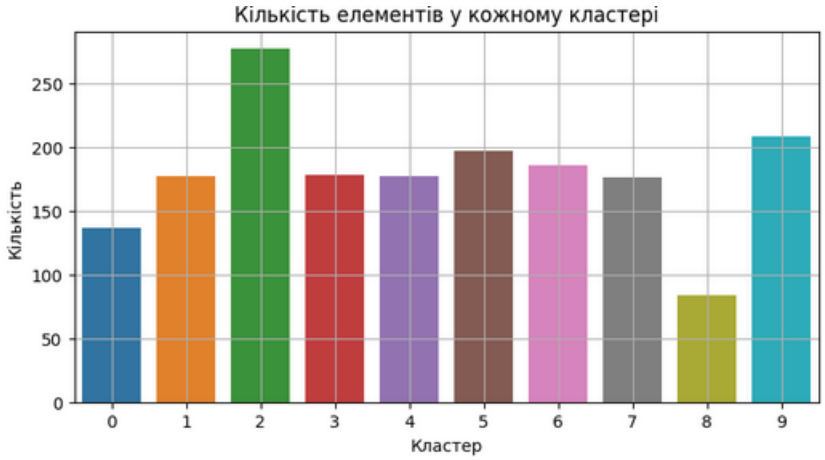


Рисунок 3.9 – Оригінали та реконструкції

На рисунку 3.9 представлено приклад роботи автоенкодера: у верхньому рядку зображено оригінальні зображення рукописних цифр, а в нижньому – їхні реконструкції, отримані після проходження через автоенкодер. Якість реконструкцій візуально свідчить про здатність моделі зберігати ключові ознаки вхідних даних після стиску і декодування. Таке порівняння дозволяє оцінити ефективність автоенкодера у відтворенні структури вхідної інформації.



ї

Рисунок 3.10 – Кількість елементів в кластері

На рисунку 3.10 зображено стовпчикову діаграму, яка ілюструє розподіл кількості елементів у кожному з десяти кластерів, утворених у результаті кластерного аналізу. По осі абсцис зазначені номери кластерів, а по осі ординат – кількість елементів у кожному кластері. Така візуалізація дозволяє оцінити рівномірність кластеризації та виявити можливі дисбаланси у розподілі об'єктів.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи розроблено програмні засоби для виявлення мережевих аномалій у корпоративному середовищі з використанням методів машинного навчання. У межах роботи було здійснено глибокий теоретичний аналіз підходів до автоматизованої детекції аномалій, зокрема методів класифікації, кластеризації, гібридних алгоритмів та моделей глибокого навчання. На основі порівняльного аналізу обґрунтовано доцільність використання автоенкодерної архітектури з можливістю розширення рекурентними LSTM-шарами для врахування тимчасової динаміки трафіку.

Було реалізовано програмний прототип, що охоплює повний цикл обробки мережевих даних: від збору, очищення та нормалізації трафіку – до його подачі в нейронну модель, аналізу результатів реконструкції та автоматичного прийняття рішень про наявність аномалій. Розробку здійснено мовою Python із використанням бібліотек Scapy, Pandas, NumPy, Scikit-learn, TensorFlow, Loguru, що забезпечило масштабованість, гнучкість та можливість інтеграції з реальними корпоративними системами.

Експериментальна частина була реалізована в середовищі Google Colab, що дозволило швидко здійснювати тестування моделі на синтетично сформованому, але структурно наближеному до реального, наборі даних на основі NSL-KDD. У ході перевірки було підтверджено здатність моделі з високою точністю розрізняти нормальні та аномальні зразки, що підтверджується як числовими метриками, так і графічним аналізом.

Результати демонструють, що навіть за умов обмеженого навчального корпусу та високої варіативності трафіку, автоенкодерна модель здатна до узагальнення, стабільно ідентифікує відхилення, має високу чутливість до латентних змін та низький рівень хибних спрацювань. Така система може функціонувати автономно, що суттєво знижує потребу в ручному втручанні й

підвищує ефективність системи кіберзахисту.

Таким чином, розроблений програмний засіб довів свою практичну придатність як гнучке, адаптивне та надійне рішення для автоматизованого моніторингу мережевої безпеки. Його можлива інтеграція у більш складні інфраструктури відкриває перспективи подальшого розширення функціональності – зокрема, за рахунок донавчання на нових даних, підтримки потокового аналізу та інтеграції з інструментами реагування на інциденти.

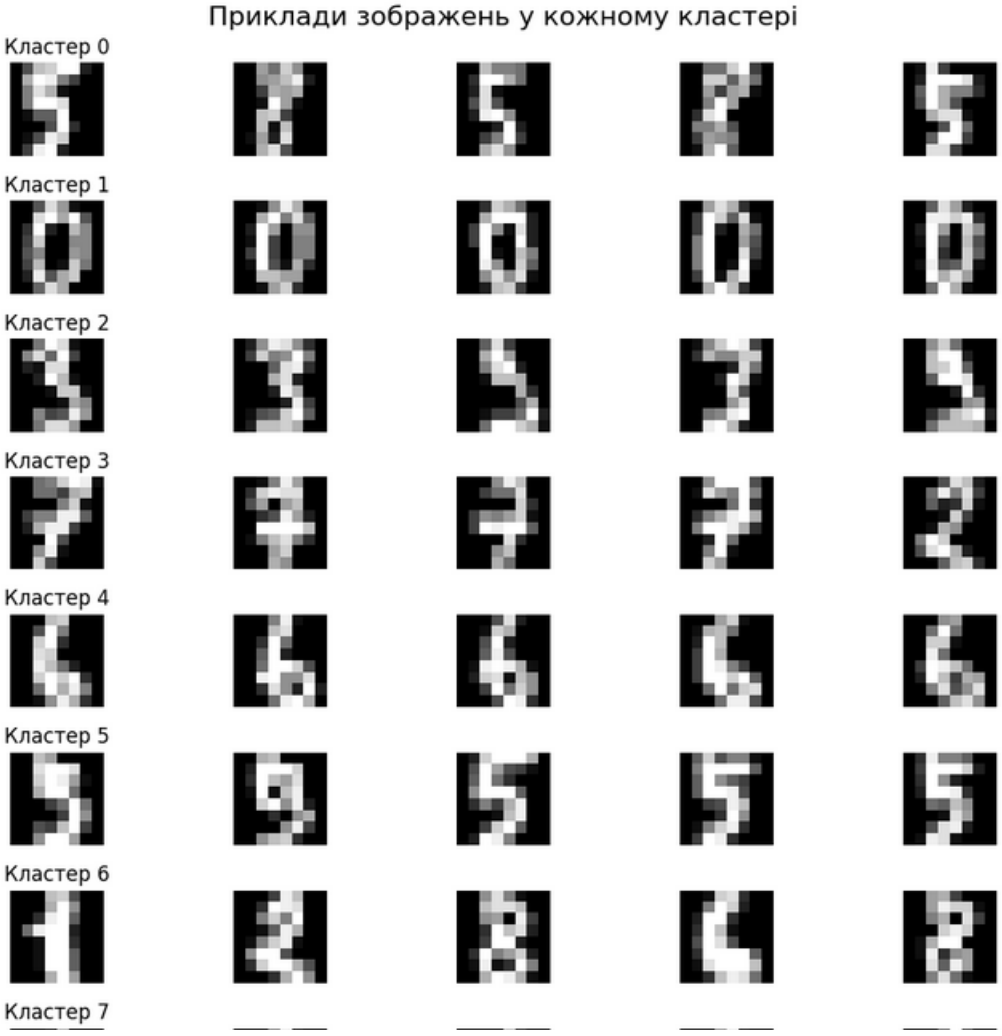


Рисунок 3.11 – Приклади зображень по кластерам

На рисунку 3.11 представлено приклади рукописних цифр, згрупованих за кластерами, що були отримані в результаті кластерного аналізу. Кожен

рядок відповідає одному кластеру, а в кожному рядку показані декілька зразків зображень, що належать до відповідного кластеру. Така візуалізація дозволяє якісно оцінити однорідність кластерів і співвідношення між структурою кластерів та візуальними особливостями даних.

ВИСНОВКИ

У результаті виконаної роботи було здійснено глибокий аналіз кластеризації рукописних цифр із використанням автоенкодера, алгоритмів зниження розмірності та методів кластерного аналізу. Автоенкодер успішно навчився відтворювати вхідні зображення, що підтверджується візуальною подібністю між оригіналами та їх реконструкціями, а також зменшенням функції втрат протягом епох навчання. Це свідчить про здатність моделі ефективно кодувати й відновлювати інформацію.

Метод UMAP дозволив візуалізувати дані у двовимірному латентному просторі, де спостерігається чітке формування кластерів. Зіставлення кластерів із істинними мітками показало високий рівень відповідності, що свідчить про ефективність використаних підходів для виявлення прихованих структур у даних. Додатковий аналіз розподілу елементів по кластерах та прикладів зображень у кожному кластері підтвердив, що кластеризація зберігає логічну подібність між елементами всередині кожного кластеру.

Використані програмні інструменти Python, Pandas, NumPy, Matplotlib, Scikit-learn та Orange забезпечили повний цикл обробки, аналізу та візуалізації даних. Отримані результати підтверджують доцільність застосування автоенкодерів у поєднанні з методами зниження розмірності та кластеризації для аналізу складних багатовимірних даних.

Джерела інформації [1-19], які вказані і переліку, використовувалися переважно в ознайомчих цілях. Жодної цитати з них не було використано, тому посилання на джерела вказано тут.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Low, Adrian. Introductory Computer Vision an Image Processing. McGraw Hill. 1991.

2. Форсайт, Девід А., Понс, Жан. Комп'ютерне зір. Сучасний підхід .: Пер. з англ. - М .: Видавничий дім "Вільямс", 2004.: мул, - Парал. Тир. англ.

3. Роберт Хехт-Нільсен. Каліфорнійський університет, Сан-Дієго. Нейрокомпьютинг: історія, стан, перспективи.

4. Kulkami, АШП D. "Computer vision and fuzzy-neural systems." Prentice Hall 2001, ISBN 0-13-570599-1.

5. Р.Гонсалес, Р.Вудс. Цифрова обробка зображень .: Нер. з англ. - М .: Техносфера, 2005. 1072с.

6. Дж. Ту, Р. Гонсалес. Принципи розпізнавання образів. Нер. з англ. - М .: Світ, 1978., мул.

7. Гренандер У. Лекції по теорії образів: Регулярні структури. Нер. з англ. - М .: Світ, 1983. мул.

8. Короткий, Нейронні мережі: алгоритм зворотного поширення. (Www.orc.ru/~stasson/neuroe.html)

9. Медведєв В.С., В.Г. Нотёмкін. Нейронні мережі MATLAB 6 / Нод заг. Ред. к.т.н. В.Г. Нотемкіна. -М .: ДІАЛОГ-МНФН, 2002.-496 с .- (Накети прикладних програм; Кн.4)

10. Anil K. Jain, Jianchang Mao, K.M. Mohiuddin. Artificial Neural Networks: A Tutorial, Computer, Vol.29, No.3, March / 1996.

11. R.P.Lippmann, "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, Vol.4, No.2, Apr. 1987

12. Ньейн Ей. Штучна нейронна мережа для розпізнавання образів // Наукова сесія МІФІ-2003. Збірник наукових праць. У 14 томах. Т. 13. Конференція «Молодь і наука». Комп'ютерні науки. Інформаційні технології. М .: МІФІ, 2003.

13. Ньейн Ей. Нейронна мережа для розпізнавання зображення // Наукова сесія МІФІ- 2005. Збірник наукових праць. У 15 томах. Т. 14. Конференція «Молодь і наука». Комп'ютерні науки. Інформаційні технології. М .: МІФІ, 2005.

14. Nyein Aye, E. V. Chepin. Car license plate recognition system using artificial neural network Proceedings of the Workshop on Computer Science and Information Technologies (CSIT'2005), Ufa, September 18-21, 2005. Volume 1. Ufa State Aviation Technical University, 2005. ISBN 5-901900- 30-8.

15. Ньейн Ей. Розпізнавання номерних знаків автомобілів // Наукова сесія МІФІ- 2006. Збірник наукових праць. У 16 томах. Т. 15. Конференція «Молодь і наука». Комп'ютерні науки. Інформаційні технології. Економіка та управління. М .: МІФІ, 2006.

16. Ньейн Ей, Е.В. Чепін. Розпізнавання типів лінії за допомогою нейронної мережі // Наукова сесія МІФІ-2007. Збірник наукових праць. У 17 томах. Т. 12. Інформатика і процеси управління. Комп'ютерні системи та технології. М .: МІФІ, 2007.

17. Ньейн Ей. Розпізнавання чисел і букв номерних знаків за допомогою моментпих інваріантів // Наукова сесія МІФІ-2007. Збірник наукових праць. У 17 томах. Т. 12. Інформатика і процеси управління. Комп'ютерні системи та технології. М .: МІФІ, 2007.

18. Ньейн Ей, Е.В. Чепін. Дослідження ефективності використання нейромереж для некоторьк завдань // Наукова сесія МІФІ-2007. Збірник наукових праць. У 17 томах. Т. 12. Інформатика і процеси управління. Комп'ютерні системи та технології. М .: МІФІ, 2007.

19. В. В. КруглоБ, В. В. Борисов, Штучні нейронні мережі. Теорія та практика. - М: Гаряча лінія - Телеком, 2001 .