Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ЙМОВІРНІСНА НЕЙРО-ФАЗІ СИСТЕМА ТА ЇЇ НАВЧАННЯ
(тема)

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

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Руденко Д.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Лотвіновій Вікторії Василівні
(прізвище, ім'я, по батькові)

1. Тема роботи Ймовірнісна нейро-фазі система та її навчання

затверджена наказом по університету від 3 листопада 2023 року № 1280Ст

2. Термін подання студентом роботи до екзаменаційної комісії 25 грудня 2023 р.

3. Вихідні дані до роботи опис тестових вибірок UCI репозиторію, науково-технічні публікації щодо дослідження та розробки нейро-фазі систем, теоретичні відомості про методи кластеризації та аналізу потоків даних.

4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз предметної області.
2. Порівняльний аналіз можливостей ймовірнісних та нечітких нейронних мереж до кластеризації потоків даних.
3. Розробка архітектури ймовірнісної нейро-фазі системи.
4. Експериментальні дослідження розробленої ймовірнісної нейро-фазі системи.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) аналіз предметної області, постановка задачі, ймовірнісна нейро-фазі система та її навчання, аналіз отриманих результатів.

6. Консультанти розділів роботи (п.6 включається до завдання 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

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	03.11.2023	
2	Аналіз завдання, підбір літератури	04.11.23-06.11.23	
3	Аналіз літератури з досліджуваної проблеми	06.11.23-08.11.23	
4	Аналіз технічних засобів	08.11.23-09.11.23	
5	Розробка ймовірнісної нейро-фазі системи	09.11.23-10.11.23	
6	Програмна реалізація	10.11.23-14.11.23	
7	Оформлення пояснювальної записки	15.11.23-30.11.23	
8	Перевірка на плагіат	10.12.2023	
9	Рецензування	20.12.2023	
10	Підготовка презентації та доповіді	25.12.2023	
11	Занесення роботи в електронний архів	04.01.2024	
12	Попередній захист кваліфікаційної роботи	04.01.2024	

Дата видачі завдання 3 листопада 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Руденко Д.О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 60 с., 4 табл., 19 рис., 60 джерел.

НЕЙРОННА МЕРЕЖА, ГІБРИДНЕ НАВЧАННЯ, ЙМОВІРНІСНА НЕЙРО-ФАЗІ СИСТЕМА, КЛАСТЕРИЗАЦІЯ ДАНИХ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ.

Об’єктом дослідження є кластеризація даних, що надходять на обробку послідовно, в онлайн режимі.

Метою дослідження є розробка ймовірнісної нейро-фаззі системи та її навчання для підвищення якості кластеризації даних, що надходять на обробку послідовно в режимі реального часу (онлайн).

В ході виконання кваліфікаційної роботи була спроєктована і реалізована нейро-фаззі ймовірнісна система за допомогою мови Python. Запропонована система дозволяє вирішувати завдання кластеризації в онлайн режимі. Система забезпечує кластеризацію у випадку, коли класи даних можуть довільно перетинатися у просторі ознак.

NEURAL NETWORK, HYBRID LEARNING, PROBABILISTIC NEURO-FUZZY SYSTEM, DATA CLUSTERING, INTELLIGENT DATA ANALYSIS.

The object of the work is the clustering of data that fed for processing sequentially, in online mode.

The purpose of the work is to develop a probabilistic neuro-fuzzy system and its training to improve the quality of data clustering, which is received for processing sequentially in real time (online).

During the qualification work was designed and implemented the neuro-fuzzy probabilistic system using the Python language. The proposed system allows to solve the problem of clustering in online mode. The system provides clustering in the case when data classes can arbitrarily overlap in the feature space.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШНМ – штучні нейронні мережі

ННМ – нейронні нечіткі мережі

РБФ – радіально-базисні функції

ANFIS – Adaptive-Network-Based Fuzzy Inference Systems (адаптивні мережеві системи нечіткого виводу)

NEFPROX – Neuro-Fuzzy function approximator (функція нейро-фаззі апроксиматора)

NEFCLASS – NEuro Fuzzy CLASSifier (нейро-фазі класифікатор)

PNN – Probabilistic Neural Network (ймовірнісна нейронна мережа)

PNFS – Probabilistic Neuro-Fuzzy System (ймовірнісна нейро-фаззі система)

NFS – Neuro-Fuzzy System (нейро-фаззі система)

EFPNN – Evolutionary Fuzzy Probabilistic Neural Network (еволюційна нечітка ймовірнісна нейронна мережа)

WTM – Winner Take More (переможець отримує більше)

SI – Silhouette Index (індекс силуету)

CHI – Calinski-Harabasz Index (індекс Калінські-Харабаса)

DBI – Davies-Bouldin Index (індекс Девіса-Болдуїна)

ВСТУП

Розв’язання завдань обробки потоків даних є однією з найважливіших проблем у сфері штучного інтелекту. У реальних задачах, пов’язаних із обробкою великих обсягів інформації, вхідні дані надходять надзвичайно швидко, а алгоритми, які повинні їх обробляти, обмежені часом та обчислювальними ресурсами. З урахуванням цих обмежень виникає потреба у методах аналізу з низькими вимогами до пам’яті та обчислювальних можливостей. Один із способів досягнення цієї мети – використання ковзних вікон. Однак через те, що дані постійно змінюються з часом, виникає потреба у реалізації швидких методів навчання для адаптації до змін і забезпечення ефективності аналізу.

Штучні нейронні мережі є важливим і популярним інструментом у сфері обчислювального інтелекту. Вони дозволяють вирішувати складні задачі, такі як розпізнавання природної мови, обробка зображень, прогнозування, виявлення шахрайства (fraud detection), діагностика захворювань людини та багато інших.

Штучні нейронні мережі мають кілька переваг, які роблять їх популярними у сфері аналізу даних. Вони можуть навчатися, адаптуючись під конкретний набір даних, що дозволяє підвищити точність кластеризації. Вони також толерантні до шуму та можуть паралельно обробляти великі обсяги інформації, що дозволяє їм вирішувати завдання, які не ефективно виконуються традиційними комп’ютерами, та робити це швидше. Ці переваги роблять їх потужним інструментом для вирішення різних завдань у світі обробки інформації.

Так, традиційні штучні нейронні мережі мають свої обмеження, особливо при аналізі даних з реальних об’єктів. Однією з основних недоліків є їх неспроможність класифікувати об’єкти, які належать до різних класів або кластерів з різним ступенем достовірності. Це може ускладнювати точність

та надійність результатів аналізу, особливо коли відомий точний клас або кластер об’єкта.

Справді, нечіткі системи стали надзвичайно популярними в наукових та інженерних застосуваннях, зокрема у системах управління та розпізнавання образів. Використання нечіткості відкриває можливості для гнучкого підходу до вирішення завдань аналізу та обробки даних, а також спрощує пояснення отриманих результатів за допомогою нечітких правил.

Проте, нещодавні дослідження показують, що нечіткі мережі мають свої обмеження. Наприклад, вони не можуть автоматично навчатись та використовувати отримані знання у механізмі висновування. Це обмеження ускладнює їх використання при розв’язанні складних задач, включаючи аналіз потоків даних. Це викликає потребу у подальших дослідженнях та розробці нових методів для подолання цих обмежень і вдосконалення можливостей нечітких систем у сучасних застосуваннях.

Таким чином, вирішення задачі аналізу потоків даних вимагає розробки швидкої нейронної мережі для обробки інформації в режимі реального часу. Для цього доцільно використовувати швидкі ймовірнісні нейронні мережі. Використання нечіткої логіки може значно прискорити роботу системи та підвищити її точність. Однак через неперервний потік даних система може стати досить громіздкою, тому важливо використовувати метод ковзного вікна для зменшення часових витрат та витрат пам’яті. Цей підхід дозволяє ефективно вирішувати завдання аналізу потоків даних у реальному часі, забезпечуючи швидку та точну обробку великих обсягів інформації.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Аналіз потоків даних

Аналіз потоків даних – це захоплююча область, де досліджуються інноваційні методи взаємодії з даними, які надходять неперервним потоком в режимі реального часу. Ця сфера науки і технологій дозволяє нам зануритися у світ інформації, розкрити її та виявити значущі закономірності, відмінності та аномалії. З врахуванням великих обсягів даних та високої швидкості їх надходження, аналіз потоків даних відкриває нові можливості у сфері реагування на події в реальному часі, допомагає зрозуміти сутність даних та забезпечує швидке прийняття рішень в умовах постійної динаміки [1-6].

Аналіз потоків даних – це процес отримання, інтерпретації і взаємодії з даними, що надходять в режимі реального часу або неперервним потоком.

Цей аналіз дозволяє виявляти закономірності, шаблони, аномалії та іншу корисну інформацію даних, що надходять та які не завжди можуть бути відзначені за допомогою традиційних методів аналізу даних.

Основні аспекти аналізу потоків даних включають:

– спостереження за реальним часом: аналізується інформація, яка надходить миттєво, без затримок, що дозволяє вчасно реагувати на зміни чи події;

– великий обсяг та висока швидкість: дані надходять у великих обсягах і часто з великою швидкістю. Тому аналітичні методи повинні бути ефективними для обробки великої кількості інформації швидко;

– потоковий аналіз та вікна: для аналізу використовуються концепції вікон, де дані обмежуються за певний час чи кількість подій для аналізу;

– виявлення патернів та асоціацій: шукання корисних патернів і взаємозв’язків в потоках даних;

– аналіз аномалій: виявлення незвичайних, відхилення від звичних патернів чи аномалій, що можуть вказувати на проблеми чи цікаві події;

– візуалізація та передача знань: подання результатів аналізу у зрозумілій та доступній формі для рішень.

Дані можна подати на обробку у різних форматах та структурах, залежно від природи даних та завдання аналізу.

Текстовий формат. Дані можуть бути подані у вигляді текстових файлів, які містять рядки тексту. Цей формат часто використовується для обробки надлишкової інформації, такої як новини, соціальні мережі або вебсторінки.

Табличний формат. Дані можуть бути представлені у вигляді таблиць з рядками та стовпцями, де кожен рядок представляє окремий запис, а кожний стовпець відповідає конкретній властивості або ознаки. Такі дані легко обробляються за допомогою баз даних чи електронних таблиць.

Графічний формат. Дані можуть бути представлені у вигляді зображень, відео або аудіофайлів. Це типовий формат для обробки медіа-даних, таких як фотографії, відеоролики, музика тощо.

JSON/XML формат. Дані можуть бути структуровані в форматі JSON (JavaScript Object Notation) або XML (eXtensible Markup Language). Ці формати дозволяють представляти дані у вигляді структурних об'єктів, що включають вкладені об’єкти, масиви та інші структури даних.

Лінгвістичний формат. У випадку аналізу мови, дані можуть бути викладені у форматі тексту, який додатково анотований з метою визначення частин мови, сутностей, або інших лінгвістичних аспектів.

Сенсорні дані. Це дані, зібрані з сенсорних пристроїв, таких як акселерометри, гіроскопи, GPS-пристрої тощо. Ці дані можуть бути числовими чи векторними та використовуються для вимірювання фізичних параметрів або відстеження руху.

Дані великих обсягів (Big Data). Це надзвичайно великі обсяги даних, які вимагають спеціальних підходів до зберігання, обробки та аналізу. Вони

можуть бути у будь-якому з вищезазначених форматів, але відрізняються за своєю обсягом та потребують високоефективних алгоритмів обробки.

Стрімінгові дані. Це дані, які постійно надходять у реальному часі, такі як дані з сенсорів, транзакції, відгуки користувачів тощо. Ці дані потребують негайної обробки та можуть бути представлені у будь-якому форматі, що дозволяє їхню обробку у реальному часі.

1.2 Нейро-фаззі системи

Задачі комплексного забезпечення адекватного представлення взаємодіючих динамічних нечітких процесів, а також оптимізації ресурсів та вибору альтернатив розвитку нечітких процесів на безлічі обмежень є важливими і в даний час не мають рішень, які б знайшли ефективне застосування в більшості практичних реалізацій і часто носять емпіричний, вузькоспеціальний характер.

Підходи на основі штучних нейронних мереж (ШНМ), нейронних нечітких мереж (ННМ) є універсальним засобом моделювання складних процесів великої розмірності, але менш ефективні при моделюванні процесів і процедур малої та середньої розмірності. Вони також потребують додаткових ресурсів для машинного навчання моделей [7-10].

Динамічні процеси в реальних системах характеризуються складними багатофункціональними залежностями та суттєвою нелінійністю. Моделювання таких процесів у задачах прогнозування, діагностики, ідентифікації, класифікації на основі існуючих класичних підходів викликає певні труднощі.

Теоретико-системний аналіз на основі представлень процесів з математичними рівняннями дозволяє створювати досить точні моделі, але досить трудомісткий і складний в реальних розробках. Експериментальний системний аналіз чи ідентифікація базуються на моделях, параметри яких

ґрунтуються на виміряних даних. Перевагами таких моделей є коротші терміни розробки, але їх якість істотно залежить від коректного вибору структури та інтерпретації результатів вимірювань.

Особливістю нейронних нечітких мереж є те, що вони відносяться до непараметричних моделей, але це викликає труднощі у визначенні відповідності (інтерпретації) їх параметрів у термінах реальних процесів. В задачах ідентифікації на основі апроксимуючих властивостей штучних нейронних мереж найефективнішими є моделі на основі багатошарового персептрону та мереж з радіально-базисними функціями (РБФ). Архітектури моделей на основі мереж Кохонена або Хопфілда та деяких інших, більшою мірою використовуються в задачах групування та класифікації [1, 11, 12].

Нечіткі системи як і нейронні мережі мають універсальні апроксимуючі властивості, з їх допомогою можна моделювати довільні функціональні залежності.

Нейронна мережа може навчатися на вхідних та вихідних даних для визначення поведінки системи. Ці знання можуть бути використані для створення нечітких правил та функцій належності, що суттєво зменшує час, потрібний на розробку. Таке об’єднання також допомагає вирішити проблему неінтерпретованості результатів, що отримуються за допомогою нейронних мереж. Вираз ваг нейронної мережі за допомогою нечітких правил забезпечує розуміння роботи нейронної мережі, що, своєю чергою, допомагає створювати ефективніші програми та додатки до них.

Нейро-фаззі системи можуть генерувати нечіткі логічні правила та функції належності для складних систем, яким недостатньо стандартного нечіткого підходу [9, 11-13]. Для таких систем стандартна нечітка логіка використовує складні ієрархічні правила, зменшення кількості самих правил, що відповідно знижує ефективність і точність рішення.

Дані штучної нейронної мережі використовують нелінійні функції належності. Перевагою таких функцій є рівномірний розподіл знань між базою правил та базою функцій власності, що відображається у скороченні

розміру бази правил. Важливим є той факт, що властивості узагальнення та уточнення нейро-фаззі систем дозволяють генерувати правила та функції належності, що забезпечують більш достовірне та точне рішення порівняно з альтернативними методами [14-17]. За допомогою коректної комбінації нечіткої логіки та нейронних мереж є можливим повне відображення знань нейронної мережі нечіткою логікою, якщо алгоритми нечіткої логіки повністю засновані на архітектурі нейронної мережі. Це дозволяє генерувати нечіткі логічні рішення із заздалегідь заданою точністю виходу.

Алгоритми, що дозволяють на основі даних спостережень будувати в кінцевому підсумку нечітку систему, на початкових етапах використовують або нечітку кластеризацію, або покриття простору багатовимірними гіперкубами, або оптимізацію багатовимірних гратчастих структур, що виникають при розбитті координатних осей n-вимірного простору на нечіткі підмножини. Основними проблемами в цьому випадку є забезпечення єдності уявлення одного й того ж лінгвістичного значення й інтерпретованості індукованого набору правил приведення одержуваних дискретних функцій належності до однієї зі стандартних форм, і навіть уникнення серйозних інформаційних втрат після проєктування одержуваних в результаті кластеризації областей.

Один із напрямків розвитку систем нечіткого виведення був запропонований в [18], де розглядаються адаптивні мережеві системи нечіткого виводу (Adaptive-Network-Based Fuzzy Inference Systems, ANFIS), побудовані на базі архітектури Такагі-Сугено, а також гібридний метод налаштування параметрів антецедента (тобто параметрів функції приналежності). У [19], доведено еквівалентність ANFIS та радіально-базисних нейронних мереж. Основною властивістю ANFIS є те, що апріорно задані правила налаштовуються в процесі навчання для більш точних результатів.

В іншій архітектурі нейро-фаззі систем NEFPROX (Neuro-Fuzzy function approximator) основна увага приділяється отриманню правил, що добре інтерпретуються. Платою за це є зниження точності апроксимації.

Метою моделі NEFCLASS (NEuro Fuzzy CLASSifier) є отримання нечітких правил з багатьох даних, які можна розділити на різні класи. Нечіткі множини та лінгвістичні правила представляють апроксимацію та визначають результат системи NEFCLASS. Вони виходять із безлічі вибірок шляхом навчання. Обов’язково має виконуватися правило, що для кожного лінгвістичного значення може існувати лише одне уявлення нечіткої множини.

1.3 Ймовірнісні нейронні мережі

В сучасних умовах одним із ключових вимог при виборі методів класифікації є їхнє швидке виконання, особливо в онлайн-режимі. Серед різних архітектур нейронних мереж надзвичайно швидкою є ймовірнісна нейронна мережа (PNN) [20].

Ймовірнісна нейронна мережа застосовується тоді, коли важливий пріоритет має саме швидкодія в роботі з даними. Це особливо актуально в випадках, коли класифікація повинна здійснюватися в режимі реального часу, і обробка інформації, яка надходить на вхід системи, має обмежений час для обробки. Ймовірнісна нейронна мережа дозволяє швидко та ефективно класифікувати дані в таких умовах, що робить її ідеальним вибором для онлайн-класифікації в реальному часі.

Ймовірнісна нейронна мережа є унікальною тим, що вона може динамічно змінювати границі областей рішень (decision boundaries) при надходженні нових спостережень. Це досягається завдяки паралельній обробці та використанню концепції «нейронів в точках даних».

Ймовірнісна нейронна мережа була вперше запропонована в 1990 році Дональдом Шпехтом [21]. Її унікальність полягає у тому, що вона поєднує статистичну теорію з теорією нейронних мереж, що дозволяє ефективно та швидко вирішувати задачі класифікації в умовах потоку даних, що змінюється.

Навчання PNN базується на статистичних методах та Баєсівському висовуванні, але не використовує зворотне поширення (backpropagation), завдяки чому, робота пришвидшується.

$$P(x \mid y) = \frac{P(y \mid x) P(x)}{P(y)}. \tag{1.1}$$

На основі цієї формули, у ймовірнісній нейронній мережі розраховуються умовні імовірності.

При вирішенні задачі класифікації за допомогою методу Байєса, спочатку формується оцінка щільності розподілу ймовірностей для кожного класу. Ця оцінка розраховується на етапі навчання моделі. Під час тестування або класифікації нових елементів, класифікуються ті, у яких ймовірність належності даного елемента до класифікованого класу є найвищою порівняно з іншими класами. Це дозволяє визначити найімовірніший клас для кожного тестового елемента на основі ймовірностей, зібраних під час навчання.

Ймовірнісна нейронна мережа (PNN) класифікує дані за допомогою таких ознак:

– щільність розподілу в області нового сигналу *x(k)*. PNN використовує щільність розподілу ймовірностей для кожного класу, щоб визначити ймовірність того, що новий сигнал *x(k)* належить до конкретного класу;

– ціна помилки класифікації. Це величина, яка вказує на вартість помилкової класифікації конкретного сигналу. PNN може враховувати цю

величину при класифікації, що допомагає зменшити помилки в процесі навчання та тестування;

– апріорна ймовірність. Це ймовірність того, що конкретний сигнал належить до певного класу, незалежно від власне щільності розподілу. Апріорна ймовірність може бути використана разом із щільністю розподілу для підрахунку остаточної ймовірності класифікації.

Ці параметри допомагають PNN визначити, до якого класу належить конкретний сигнал, з урахуванням ймовірності та витрат при класифікації.

Архітектура ймовірнісної нейронної мережі наведена на рисунку 1.1.

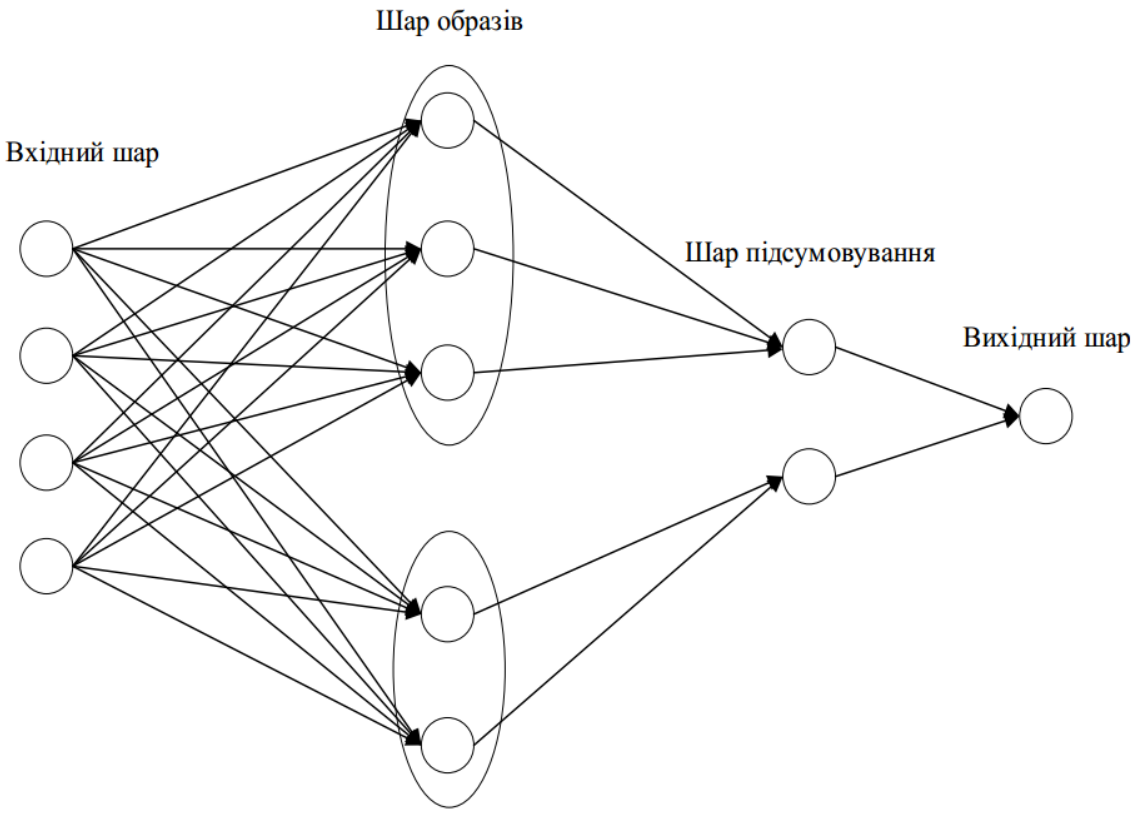

Рисунок 1.1 – Архітектура ймовірнісної нейронної мережі

Архітектура ймовірнісної нейронної мережі вперше демонструє можливість розбиття Баєсівського класифікатора на прості процеси, які можуть виконуватися паралельно. Це важливе досягнення, оскільки такий підхід дозволяє ефективно вирішувати задачі класифікації за допомогою розділення складної задачі на менші та більш керовані процеси. Кожен

процес відповідає за певний класифікаційний варіант, і їх паралельне виконання сприяє швидкодії та ефективності обробки даних.

Такий підхід особливо корисний при роботі з великими об’ємами даних, де швидкодія та паралельна обробка можуть визначати продуктивність та точність класифікації. PNN відкриває можливості для оптимізації роботи Баєсівських класифікаторів та їх застосування в широкому спектрі завдань аналізу даних.

Ймовірнісна нейронна мережа має чотири шари у своїй архітектурі:

– вхідний шар. Цей шар відповідає за приймання вхідних сигналів. Кожен вхід відповідає окремій ознаці (або характеристиці) даних;

– шар шаблонів. У цьому шарі зберігаються шаблони даних, які вивчені під час тренування мережі. Кожен вузол у цьому шарі представляє один шаблон;

– шар суматорів. Кожен вузол у цьому шарі вираховує суму квадратів відстаней між вхідними даними та шаблонами у відповідних шарах шаблонів. Ці суми використовуються для розрахунку ймовірностей;

– вихідний шар. У цьому шарі проводиться класифікація на основі розрахованих ймовірностей. Кожен вузол у вихідному шарі представляє один клас, і ймовірність належності до кожного класу розраховується на основі суматорів.

Ця чотиришарова архітектура дозволяє ймовірнісній нейронній мережі ефективно та швидко вирішувати задачі класифікації за допомогою відомих шаблонів та ймовірностей.

Ймовірнісні нейронні мережі (PNN) мають свої переваги і недоліки. До переваг слід віднести:

– висока точність класифікації. PNN зазвичай надає високу точність при класифікації, особливо коли використовуються великі та репрезентативні тренувальні набори даних;

– здатність враховувати розподіл даних. PNN може ефективно моделювати складні розподіли даних, що робить її корисною для задач класифікації, де дані мають складний та нелінійний характер;

– швидка класифікація нових даних. Після навчання, класифікація нових даних відбувається швидко через використання заздалегідь вивчених шаблонів;

– можливість роботи з неперервними та категоріальними даними: PNN може працювати як з неперервними, так і з категоріальними даними без значних модифікацій.

Окрім переваг, PNN має ще недоліки:

– пам’ять. PNN потребує збереження всіх навчених шаблонів у пам’яті, що може стати проблемою для великих наборів даних;

– обчислювальні витрати: Обчислювальні витрати при навчанні можуть бути великими, особливо якщо тренувальний набір даних великий та складний;

– нееластичність. Після навчання PNN не може адаптуватися до умов, що змінюються або додавання нових класів без повного перетренування;

– підбір гіперпараметрів. Вибір оптимальних значень для гіперпараметрів (наприклад, ширина ядра) може бути нетривіальною задачею.

У підсумку, ймовірнісні нейронні мережі є потужним інструментом для класифікації даних, особливо в умовах, де розподіл даних складний і нелінійний. Однак вони мають свої обмеження, які варто враховувати при їх використанні.

1.4 Кластерний аналіз

Кластерний аналіз – це статистичний метод аналізу даних, який спрямований на групування схожих об’єктів у колективності, відомі як

кластери. Цей метод має довгу історію, і він був застосований у різних галузях знань.

Кластерний аналіз походить з антропології та психології, де він був вперше запропонований Драйвером і Крьобером у 1932 році. Пізніше, у 1938 та 1939 роках, цей метод був введений в психологію Зубіним і Робертом Тріоном. Відомість кластерного аналізу взяла початок завдяки його використанню Кеттелем для класифікації теорії ознак в психології особистості, починаючи з 1943 року.

Кластерний аналіз – це процес розподілу елементів даних на групи так, що елементи в одній групі максимально схожі між собою, тоді як елементи різних груп є якнайменш схожими. Залежно від типу даних і мети кластерного аналізу використовуються різні метри подібності, такі як відстань, зв’язок та інтенсивність. Цей метод має застосування у різних галузях, включаючи маркетинг, біологію, медицину, соціологію та інші, дозволяючи виявляти приховані закономірності та розуміти структуру даних.

Загальна та більш широка мета кластерного аналізу полягає в глибшому розумінні структури даних і виявленні подібності або відмінностей між об’єктами чи спостереженнями в наборі даних. Основні аспекти цієї мети включають:

– групування та класифікація. Кластерний аналіз допомагає автоматично групувати подібні об’єкти, визначаючи структуру великих масивів даних;

– виявлення паттернів. Кластерний аналіз може виявляти паттерни, які можуть бути непомітними при звичайному огляді даних;

– подальший аналіз та прогнозування. Кластерний аналіз може служити вихідною точкою для більш глибокого аналізу. Пошук внутрішніх відмінностей у кластерах може допомогти розробити точніші моделі для прогнозування майбутніх подій або трендів;

– сегментація ринку та клієнтів. У бізнесі, кластерний аналіз застосовується для розуміння сегментації ринку та поведінки клієнтів. Це

дозволяє бізнесу краще спілкуватися з різними сегментами аудиторії та виробляти більш ефективні стратегії маркетингу;

– виявлення аномалій. Кластерний аналіз може виявляти аномалії або викиди в даних, що можуть бути важливими для виявлення проблем чи незвичайних явищ.

Отже, мета кластерного аналізу полягає в глибокому розкритті структури та взаємозв'язків у даних, що може вести до кращого розуміння явищ, більш точних прогнозів та ефективніших стратегій в прийнятті рішень.

1.5 Постановка задачі дослідження

Сучасні системи та методи обчислювального інтелекту стали невід'ємною частиною інтелектуального аналізу даних (Data Mining) [22, 23] і знаходять застосування для розв'язання різноманітних завдань, таких як прогнозування, класифікація, кластеризація, вирішення оптимізаційних задач і багато інших. Ці методи дозволяють виявляти закономірності в даних, знаходити корисну інформацію та використовувати її для прийняття управлінських рішень, прогнозування майбутніх подій та здійснення більш точного аналізу. Такий інтелектуальний аналіз даних стає дуже важливим інструментом в сучасному світі, де обсяги даних надзвичайно великі і різноманітні.

Аналіз потоків даних є однією з ключових та актуальних задач обчислювального інтелекту, яка широко застосовується на практиці. Вирішення цієї задачі зазвичай вимагає значних обчислювальних витрат та потребує системи, що здатна швидко обробляти дані, що надходять та швидко давати відповіді. У великих потоках даних, вхідні послідовності можуть мати величезну або навіть необмежену довжину, що ставить вимоги до розробки ефективних та швидкодіючих методів класифікації для таких обсягів інформації. Покращення швидкодії та ефективності обробки цих

великих та неперервних потоків даних стає важливою задачею у розробці методів аналізу потоків даних.

В умовах обробки необмеженого обсягу даних у режимі реального часу, де дані надходять послідовно, важливо контролювати «вікно», яке обирається для обробки. В таких випадках застосування методу ковзного вікна є ключовим для ефективної обробки потоків даних. Це свідчить про актуальність розробки швидкодіючих методів online-кластеризації потоків даних [24-27] з урахуванням різних рівнів належності вхідних даних до кількох класів. Використання ймовірнісних нейронних мереж може допомогти в цьому контексті, дозволяючи точно та швидко класифікувати дані в режимі реального часу відповідно до їхніх ймовірностей належності до різних класів.

Об’єктом дослідження є кластеризація даних, що надходять на обробку послідовно, в онлайн режимі.

Метою дослідження є розробка ймовірнісної нейро-фаззі системи та її навчання для підвищення якості кластеризації даних, що надходять на обробку послідовно в режимі реального часу (онлайн).

Для досягнення мети необхідно вирішити такі завдання:

– аналіз та опис предметної області;

– постановка завдання;

– розробка архітектури ймовірнісної нейро-фаззі системи;

– експериментальна перевірка розробленої нейро-фаззі системи шляхом розпізнавання послідовностей даних, які надходять на обробку.

2 ЙМОВІРНІСНА НЕЙРО-ФАЗІ СИСТЕМА ТА ЇЇ НАВЧАННЯ

Ймовірнісні нейронні мережі є одними з важливих інструментів у сфері машинного навчання та штучного інтелекту. Ці мережі засновані на байєсівському висновуванні та використовують гаусіанські функції для обробки даних. Радіально-базисні функції (RBFN) та узагальнені регресійні нейронні мережі (GRNN) є їх близькими родичами. Однак ймовірнісні нейронні мережі мають свої особливості, що робить їх потужними та ефективними для вирішення задач класифікації та регресії.

Ймовірнісні нейронні мережі мають важливу перевагу у швидкості порівняно з глибинними нейронними мережами (DNN), особливо в режимі реального часу. Ця швидкість досягається завдяки лінивому навчанню та концепції «нейронів у точках даних». Деякі з їх важливих переваг включають:

– швидкість навчання. Ймовірнісні нейронні мережі швидше навчаються, оскільки вони використовують ліниве навчання та не вимагають повного проходження через всі дані під час навчання;

– режим реального часу. У випадках, коли потрібно проводити класифікацію в режимі реального часу, швидкість ймовірнісних нейронних мереж є критичною, оскільки вони здатні швидко обробляти вхідні дані;

– ефективність з обмеженими ресурсами. Ймовірнісні нейронні мережі можуть бути більш ефективними з точки зору ресурсів, оскільки вони вимагають менше обчислень для навчання та передбачення;

– працездатність у відсутності повних даних. Вони можуть ефективно працювати з неповними або неструктурованими даними, оскільки їхні моделі дозволяють обробляти дані навіть у випадку відсутності конкретних значень.

Однак важливо враховувати, що вибір між ймовірнісними нейронними мережами та глибинними нейронними мережами повинен залежати від конкретної задачі, вимог до точності та доступних ресурсів.

Важливо підкреслити, що глибокі нейронні мережі, а також традиційні плоскі (мілінні) нейронні мережі, спеціалізуються на чіткій класифікації спостережень. Це означає, що вони намагаються розділити дані на чітко визначені класи, які не перетинаються у просторі ознак. Це особливо важливо у багатьох завданнях, таких як розпізнавання об’єктів або визначення категорій.

Проте, в багатьох реальних сценаріях даних може виникати проблема, відома як «невизначеність класу» або «розділова зона нечітка». Це означає, що деякі області в просторі ознак можуть бути неоднозначні, і може бути складно однозначно визначити, до якого класу вони належать.

У таких випадках ймовірнісні нейронні мережі можуть виявитися корисними. Вони можуть надати ймовірності того, що спостереження належать до кожного з класів, замість того, щоб надавати тільки один конкретний клас. Це дозволяє враховувати невизначеність і нечіткість у вхідних даних, що може бути важливим у деяких застосуваннях, наприклад, в медичних діагностиках чи розпізнаванні образів.

Задача нечіткої класифікації може бути вирішена за допомогою нейро-фаззі систем (NFS). Ці системи враховують неоднозначність та невизначеність у вхідних даних і надають нечіткі оцінки належності кожного спостереження до можливих класів.

Проте, NFS мають певні обмеження, зокрема, у великій обчислювальній складності, яка сповільнює їх роботу, особливо у режимі онлайн опрацювання інформації. Це може стати серйозною проблемою у сучасних застосуваннях, де час відгуку та швидкість обробки даних дуже важливі.

У відповідь на це виникає потреба в розвитку більш ефективних методів та алгоритмів для нечіткої класифікації, які можуть працювати у режимі реального часу та оптимізувати ресурси обчислень. Це може включати в себе поєднання нейронних мереж із фаззі логікою, а також

застосування оптимізованих методів навчання та апаратної реалізації для підвищення ефективності нечіткої класифікації в реальному часі.

Тут, на перший план виходить еволюційна оптимізація нейронних мереж, що є причиною виникнення нового напряму нейронних мереж, таких як еволюційні нечіткі нейронні мережі.

2.1 Еволюційна нечітка ймовірнісна нейронна мережа

Еволюційна нечітка ймовірнісна нейронна мережа (Evolutionary Fuzzy Probabilistic Neural Network, EFPNN) – це складна архітектура нейронної мережі, яка використовується для рішення завдань класифікації, прогнозування та апроксимації в умовах невизначеності та нечіткості вхідних даних. Ця мережа поєднує в собі концепції нечіткої логіки, ймовірнісних моделей та еволюційних алгоритмів для досягнення оптимальних результатів [28-30].

Основні характеристики та компоненти еволюційної нечіткої ймовірнісної нейронної мережі включають:

- нечітка логіка;
- ймовірнісні моделі;
- еволюційні алгоритми;
- паралельність та розподіленість;
- вибір оптимальних параметрів.

Зупинимось на кожній характеристиці окремо.

Еволюційна нечітка ймовірнісна нейронна мережа використовує нечітку логіку для обробки нечітких та невизначених даних.

Нечітка логіка – це математичний підхід, який дозволяє враховувати нечіткі або неявні концепції в прийнятті рішень. У контексті EFPNN, нечітка логіка використовується для формулювання правил класифікації та прийняття рішень на основі нечітких умов.

В рамках EFPNN, нечітка логіка використовується для наступних завдань:

– формулювання нечітких правил. Експертні знання можуть бути виражені у вигляді нечітких правил, де вказуються умови та висновки з урахуванням нечіткості або відсутності даних;

– обчислення нечітких відносин. Нечіткі логічні операції, такі як «І» (логічне «І»), «АБО» (логічне «АБО»), використовуються для обчислення ступеня належності об’єкта до різних класів;

– розмивання даних. Вхідні дані можуть бути розмиті, що дозволяє враховувати невизначеність у вхідних даних та ускладнює класифікацію;

– агрегація правил. Нечіткі правила можуть бути об’єднані для визначення висновків на основі вхідних даних.

Еволюційна нечітка ймовірнісна нейронна мережа також враховує ймовірності входження об’єктів в різні класи [31, 32]. Це може бути корисно у випадках, коли деякі об’єкти можуть мати високий рівень невизначеності щодо своєї класифікації.

EFPNN використовує нейронні мережі для навчання та адаптації до вхідних даних. Це дозволяє автоматично визначати ваги та зв’язки між вхідними та вихідними даними для оптимальної класифікації.

Еволюційні нечіткі ймовірнісні нейронні мережі можуть використовувати еволюційні алгоритми для оптимізації структури та параметрів нейронних мереж. Це дозволяє знаходити оптимальні рішення в умовах невизначеності та зміни вхідних умов.

EFPNN може бути використана в різних областях, де потрібно моделювати та передбачати явища з урахуванням невизначеності в даних. Це може включати в себе фінансовий аналіз, медичні діагностики, управління виробництвом тощо.

2.2 Адаптивний нейро-фаззі метод для кластеризації даних

Задача кластеризації масивів даних, що описуються наборами векторів-образів, досить часто зустрічається в багатьох додатках, пов'язаних з інтелектуальним аналізом даних, при цьому останнім часом особлива увага приділяється так званій нечіткій кластеризації [31-33], коли оброблюваний вектор-образ ознак з різними рівнями ймовірностей, можливостей чи належностей може відноситися одночасно до кількох класів [33-36].

Вихідною інформацією є $N \times n$ таблиця «об'єкт-властивість», яка містить інформацію про N об'єктах, кожен з яких описується $(1 \times n)$ - вектором-рядком ознак $X = \{x_1, x_2, ..., x_N\} \subset R^n, x_k \in X, k = 1, 2, ..., N$.

Результат кластеризації є розбиття вихідного масиву даних на m класів $(1 \le m \le N)$ з деяким рівнем належності $U_q(k)$ k-го вектора ознак до q-го кластера $(1 \le q \le m)$. Дані, що надходять на обробку, дані попередньо центруються і стандартизуються за всіма ознаками так, щоб усі спостереження належали гіперкубу $[-1,1]^n$. Таким чином дані, що підлягають кластеризації, утворюють масив $\tilde{X} = \{\tilde{x}_1, ..., \tilde{x}_k, ..., \tilde{x}_N\} \subset R^n$, $\tilde{x}_k = (\tilde{x}_{k1}, ..., \tilde{x}_{ki}, ..., \tilde{x}_{kn})^T$, $-1 \le \tilde{x}_{ki} \le 1$, $1 < m < N$, $1 \le q \le m$, $1 \le i \le n$, $1 \le k \le N$.

Ситуація різко ускладняється, якщо зв'язок між стовпцями $\vec{x}_j$ не може бути адекватно описаний лінійними співвідношеннями, а носить більш складний, нелінійний характер, причому характер цієї нелінійності апріорі невідомий. Звичайно, в загальному випадку довільний тип нелінійності може бути як завгодно точно відновлений за допомогою нейромережевих технологій [37-40], проте, по-перше, штучні нейронні мережі вимагають для свого навчання досить великих за обсягом навчальних вибірок, нехай і з пропусками [41-44], а, по-друге, різко ускладняється реалізація системи відновлення та знижується її швидкодія, не кажучи вже про неможливість роботи такої мережі в режимі on-line.

Подолати це ускладнення можна, скориставшись в якості «будівельних блоків» системи відновлення, так званими, нео-фаззі нейронами [45, 46], які, з одного боку є нейро-фаззі системами, що володіють високими апроксимуючими властивостями.

Структурна схема нео-фаззі нейрона наведена на рисунку 2.1 і, як видно, крім настроюваних синаптичних ваг w_{li}, вона містить нелінійні функції належності $\mu_{li}(x_i)$, які забезпечують нео-фаззі нейрону необхідні апроксимуючі властивості.

При поданні на вхід нео-фаззі нейрона векторного сигналу $\tilde{x}_k = (\tilde{x}_{k1}, \tilde{x}_{k2}, ..., \tilde{x}_{kn})^T$ (тут вхідні дані попередньо перетворюються так, що $0 \le \tilde{x}_{ki} \le 1$ для всіх $1 \le k \le N$ і $1 \le i \le n$), на його виході з'являється скалярне значення

$$\hat{y}_k = \sum_{i=1}^{n} f_i(\tilde{x}_{ki}) = \sum_{i=1}^{n} \sum_{l=1}^{h} w_{li}(k-1)\mu_{li}(\tilde{x}_{ki}) \qquad (2.1)$$

визначається як настроюваними синаптичними вагами $w_{li}(k-1)$, так і функціями належності $\mu_{li}(\tilde{x}_{ki})$, в якості яких можуть бути використані будь-які прийняті в теорії нечітких систем конструкції: трикутні, дзвонуваті, сплайн-функції і т.п.

Цільова функція кластеризації має вигляд

$$E(U_q(k), w_q, \mu_q) = \sum_{k=1}^{N} \sum_{q=1}^{m} U_q^{\beta}(k) D^2(\tilde{x}_k, w_q) + \sum_{q=1}^{m} \mu_q \sum_{k=1}^{N} (1 - U_q(k))^{\beta} \qquad (2.2)$$

скалярний параметр $\mu \ge 0$ визначає відстань, на якій рівень належності набуває значення 0,5, тобто якщо $D^2(\tilde{x}_k, w_q) = \mu_q$, то $w_q(k) = 0,5$.

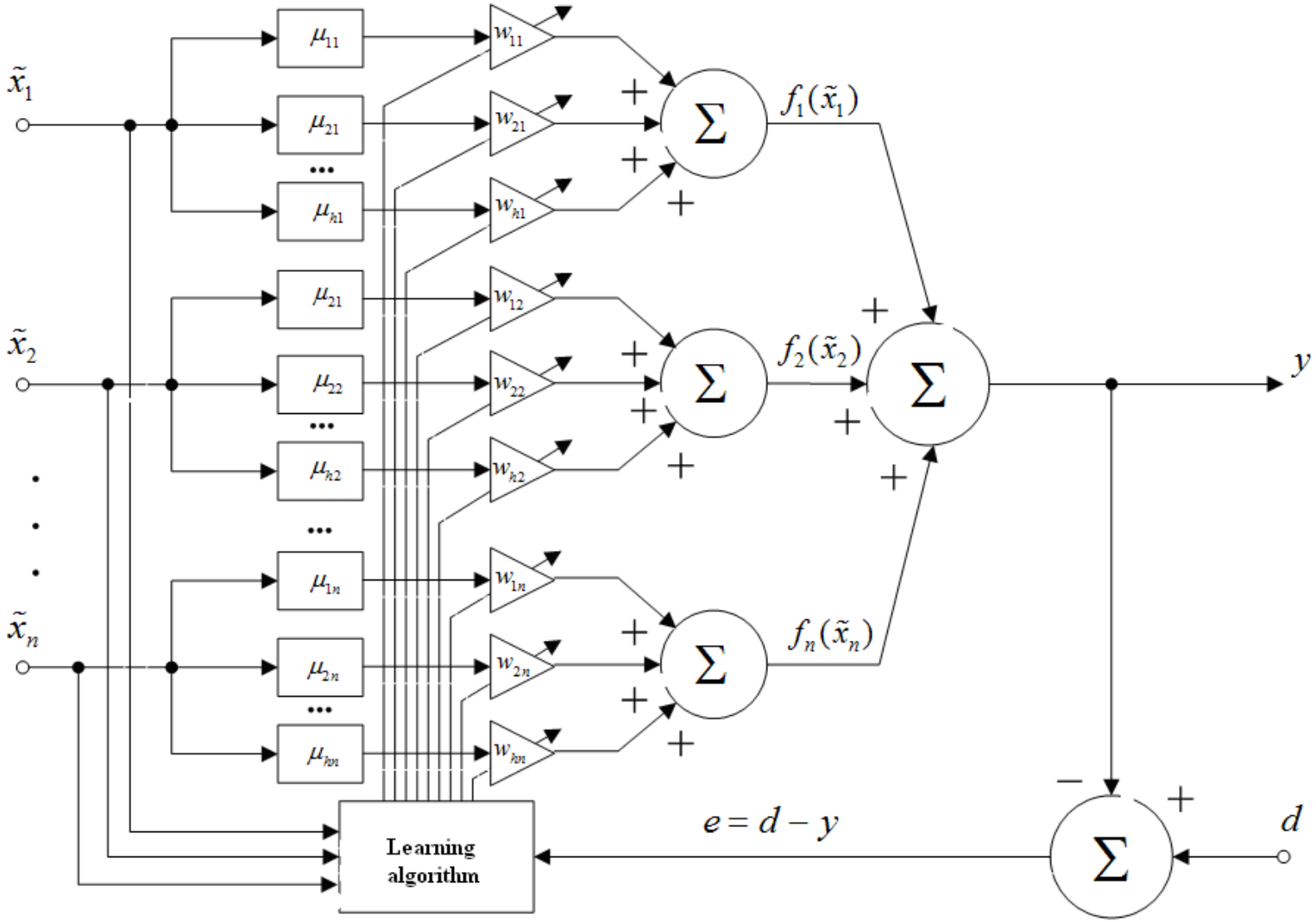

Рисунок 2.1 – Нео-фаззі нейрон

Мінімізація по $U_q(k), w_q$ та μ_q дає наступне рішення

$$\begin{cases} U_q^{(\tau+1)}(k) = \dfrac{1}{1+\left(\dfrac{D^2(\tilde{x}_k, w_q^{(\tau)})}{\mu_q^{(\tau)}}\right)^{\frac{1}{\beta-1}}}, \\ w_q^{(\tau+1)} = \dfrac{\sum_{k=1}^{N}(U_q^{(\tau+1)}(k))^{\beta}\tilde{x}_k}{\sum_{k=1}^{N}(U_q^{(\tau+1)}(k))^{\beta}}, \\ \mu_q^{(\tau+1)} = \dfrac{\sum_{k=1}^{N}(U_q^{(\tau+1)}(k))^{\beta} D^2(\tilde{x}_k, w_q^{(\tau+1)})}{\sum_{k=1}^{N}(U_q^{(\tau+1)}(k))^{\beta}}. \end{cases} \quad (2.3)$$

В режимі on-line обробки інформації співвідношення (2.3) може бути переписане у вигляді

$$\begin{cases} U_q(k+1) = \dfrac{1}{1+\left(\dfrac{D^2(\tilde{x}_{k+1}, w_q(k))}{\mu_q(k)}\right)^{\frac{1}{\beta-1}}}, \\ w_q(k+1) = w_q(k) + \eta(k+1)U_q^{\beta}(k+1)(\tilde{x}_{k+1} - w_q(k)). \\ \mu_q(k+1) = \dfrac{\sum_{p=1}^{k+1} U_q^{\beta}(p) D^2(\tilde{x}_p, w_q(k+1))}{\sum_{p=1}^{k+1} U_q^{\beta}(p)} \end{cases} \quad (2.4)$$

Неважко помітити, що другі співвідношення в (2.3), (2.4) є не що інше, як WTM-правила самонавчання Кохонена з кошианами як функції сусідства.

2.3 Ймовірнісна нечітка нейромережа та її гібридне навчання

При вирішенні задачі обробки потоків даних була виконана низка модифікацій PNN, що забезпечила можливість опрацювання потоків нестаціонарних даних за умов перетинних класів на основі ідей нечіткої кластеризації [47, 48]. В той же час слід відзначити і деяку громіздкість цих мереж, як ми бачимо на рисунку 2.2 , тому що стандартна PNN може бути дещо складною, оскільки її перший прихований шар образів у загальному

випадку містить кількість R-нейронів, що дорівнює числу спостережень у навчальні вибірці.

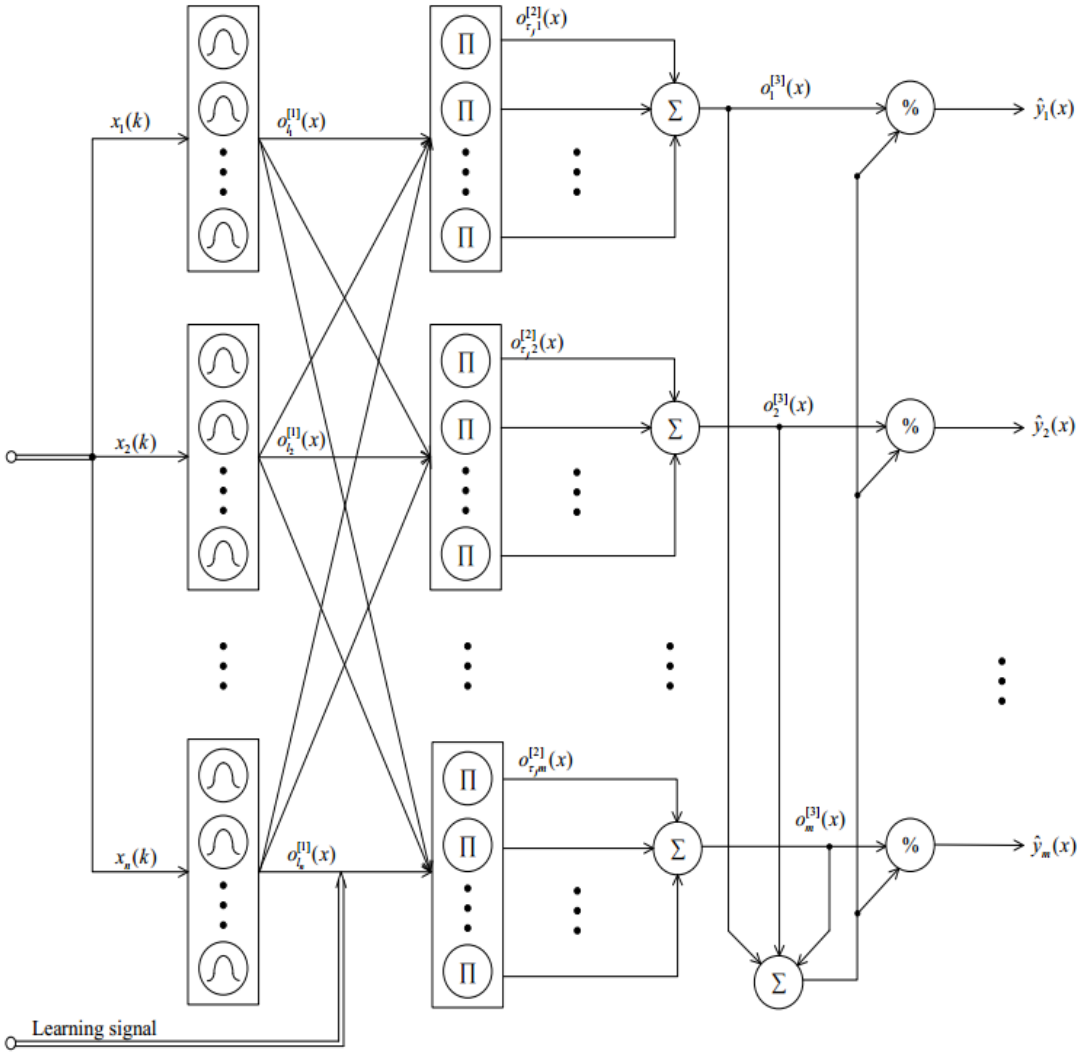

Рисунок 2.2 – Ймовірнісна нейронна мережа

Для подолання цієї незручності можна скористатися підходом, що ґрунтується на гібридних системах обчислювального інтелекту [16], зокрема, на нейро-фаззі системах [12, 28]. Ці системи мають численні переваги перед класичними нейронними мережами, оскільки зберігають універсальні властивості апроксимації, а водночас дозволяють використовувати можливості навчання у режимі реального часу.

На рисунку 2.2 наведено архітектуру ймовірнісної нечіткої нейромережі, що містить чотири шари обробки інформації.

На нульовий (рецепторний) шар системи послідовно надходять вектори спостережень, що формують навчальну вибірку у вигляді $X = \{x_1, x_2, ..., x_N\} \subset R^n, x_k \in X, k = 1, 2, ..., N$.

Спостереження надходять на перший прихований шар фаззіфікації покомпонентно, де вони обробляються за допомогою нечітких функцій належності $\mu_{li}(\tilde{x}_{ki})$, в якості яких можуть бути використані будь-які прийняті в теорії нечітких систем конструкції: трикутні, дзвонуваті, сплайн-функції і т.п.

Точна кількість цих функцій належності на кожному вході x_i може варіюватись і залежить від конкретної задачі. Наприклад, у випадку бінарної вхідної змінної, яка може приймати лише два значення («так» або «ні»), можуть використовуватись дві функції належності. У стандартній ймовірнісній нейронній мережі PNN кількість цих функцій може дорівнювати N, де N – це кількість спостережень у навчальній вибірці.

Вихідні сигнали першого шару

$$o_{l_i}^{[1]} = \mu_{li}(x_i). \qquad (2.5)$$

Спостереження подаються на входи другого прихованого шару агрегації, який складається з N стандартних блоків множення. Ці блоки формують дзвонуваті (bell-shaped) функції активації у формі багатовимірних Гауссіанів. Такий підхід дозволяє створити гібридну систему, яка комбінує нечітку логіку та ймовірнісні методи, забезпечуючи ефективну обробку навчальних даних у реальному часі.

Такий підхід має свої переваги у випадку, якщо деякі компоненти вхідних векторів навчальної вибірки співпадають. Це часто відбувається, коли окремі компоненти вхідних сигналів є бінарними, ранговими або номінальними змінними – ситуація, яка зустрічається у багатьох реальних

задачах. У такому випадку кількість функцій належності на кожному вході x_i може бути менше, ніж кількість навчальних даних N, що полегшує обробку навчальних даних та робить систему більш ефективною у вирішенні задач.

Третій прихований шар утворений $m+1$суматорами, при цьому перші m з яких розраховують Парзенівські оцінки щільності розподілу даних у кожному класі:

$$o_j^{[3]} = \sum_{\tau_j = N_1 + N_{2+\ldots}+1}^{N_1 + N_{2+\ldots}} o_{\tau_j j}^{[2]} = p_j(x), \tag{2.6}$$

а $m+1$-й підсумовує всі вихідні сигнали цього шару.

У вихідному шарі системи обчислюється ймовірність належності кожного спостереження x_i, що не належить до навчальної вибірки, до кожного класу за формулою (2.1).

У системі, вихідний шар виконує операцію, яка схожа на процес дефаззіфікації в відомих нейро-фаззі системах. Однак у нашому випадку ця операція має суто ймовірнісний сенс. Це означає, що вихідні значення системи вказують ймовірність належності вхідних даних до різних класів або категорій. Такий підхід дозволяє враховувати ймовірнісний аспект при вирішенні задачі класифікації.

Для налаштування цієї системи використовується модифікована процедура лінивого навчання, що базується на концепції «Нейрони в точках даних». Цей підхід передбачає, що навчання відбувається лише тоді, коли системі потрібно взяти рішення, замість того, щоб навчати всю систему заздалегідь. Це дозволяє системі адаптуватися до конкретних умов та змінюваних даних в режимі реального часу.

Використання нейро-фаззі підходу дозволяє зменшити кількість одновимірних функцій належності, які використовуються в другому прихованому шарі для формування багатовимірних функцій активації. Це

спрощує обчислення та поліпшує продуктивність системи, забезпечуючи при цьому ефективність та точність у вирішенні завдань.

Ймовірнісна нечітка нейронна мережа (probabilistic neuro-fuzzy system) розроблена для вирішення завдань розпізнавання образів та класифікації. Ця система є узагальненням ймовірнісної мережі Шпехта на випадок перетинних класів і може працювати як з короткими, так і з довгими навчальними вибірками. За своєю архітектурою, ця система подібна до нейро-фаззі системи Такагі-Сугено-Канга нульового порядку, але має значно вищу швидкість навчання, що базується на принципі «Нейрони в точках даних». Процес налаштування цієї системи полягає у встановленні центрів функцій належності у шарі фаззіфікації, реконфігурації блоків множення у шарі агрегації та відбувається майже миттєво. Ця система є простою у чисельній реалізації, що полегшує її впровадження.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ЙМОВІРНІСНОЇ НЕЙРО-ФАЗЗІ СИСТЕМИ ПРИ КЛАСТЕРИЗАЦІЇ ДАНИХ

У цьому розділі наведені результати експериментальної перевірки ефективності ймовірнісної нейро-фаззі системи при вирішенні задачі нечіткої кластеризації даних, які надходять поступово і представлені у бінарній, номінальній та числовій формах.

Отримані результати класифікації були порівняні з результатами роботи відомих методів як за якістю, так і за швидкістю роботи. Порівняння проводилося на основі зіставлення отриманих вихідних даних з результатами роботи розробленої нейро-фаззі мережі в рамках задачі нечіткої класифікації потоку вхідних даних.

Для виконання експериментальної перевірки роботи мережі було розроблено програмне забезпечення за допомогою високорівневої мови програмування Python. Ця мова програмування є досить популярною в сфері Data Science, оскільки вона відзначається простим синтаксисом та величезним вибором різних бібліотек. Python допомагає підвищити продуктивність під час розробки та аналізу написаного коду завдяки своїй зручності та потужним функціональним можливостям.

3.1 Опис вхідних наборів даних

Розроблена ймовірнісна нейро-нечітка система призначалася для опрацювання різноманітних типів даних, включаючи числові та бінарні дані, що можуть бути представлені у вигляді як довгих, так і коротких наборів даних. З цією метою для експериментальної оцінки були обрані вибірки з репозиторію UCI [49, 50], які включали дані різних типів.

3.1.1 Тренувальна вибірка «Heart Disease»

Перший набір даних під назвою «Heart Disease» містить 303 спостереження та 76 атрибутів, проте у цьому випадку використовується лише підмножина, що складається з 14 атрибутів, як зображено на рисунку 3.1.

Дані включають числові та категоріальні атрибути, які характеризують фізіологічні параметри та інші показники здоров’я пацієнтів.

Головна мета цього набору даних полягає в тому, щоб розробити модель кластеризації, яка може передбачити наявність чи відсутність серцевих захворювань на основі вхідних параметрів пацієнтів.

Деякі з атрибутів, які можуть бути включені у цей набір даних, включають в себе вік пацієнта, стать, артеріальний тиск, рівень холестерину, рівень цукру в крові, електрокардіографічні показники та інші клінічні параметри.

	age	sex	ChestPain	RestBP	Cholestoral	Blood Sugar	ECG	MAXHeartRate	ExerciseAngina	oldpeak	slope	MajorVessels	thal	Target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

Рисунок 3.1 – Приклад вибірки «Heart Disease»

Цей набір даних є важливим для досліджень у галузі кардіології та медичної статистики, оскільки він дозволяє розробити моделі для прогнозування ризику серцевих захворювань на основі вхідних медичних

даних пацієнтів. Фізіологічні параметри мають числову форму, а симптоми, як правило, мають бінарну форму.

3.1.2 Тренувальна вибірка «Diabetes 130-US hospitals for years 1999-2008»

Набір даних «Diabetes 130-US hospitals for years 1999-2008» – це набір медичних даних, який містить інформацію про пацієнтів із цукровим діабетом, яких лікували в різних лікарнях США протягом періоду з 1999 по 2008 рік.

Набір містить дані про пацієнтів з 130 різних лікарень. Цей набір є довгим набором даних, який містить 100 000 випадків.

Дані включають числові, категоріальні та бінарні атрибути, такі як вік пацієнта, стать, тривалість госпіталізації, лікування та діагностика, лабораторні показники, а також інші характеристики здоров’я. Головна мета цього набору даних полягає в аналізі та вивченні факторів, що впливають на лікування пацієнтів із цукровим діабетом в лікарнях. Це може включати в себе вивчення частоти госпіталізацій, тривалості лікування, ефективності лікування та зв'язку між різними факторами та прогнозом пацієнтів. Цей набір даних може бути використаний для проведення досліджень у галузі медичної статистики та аналізу ефективності лікування пацієнтів із цукровим діабетом. Він також може використовуватися для розробки моделей прогнозування та визначення оптимальних методів лікування.

3.1.3 Тренувальний набір даних «Fashion MNIST»

Набір даних «Fashion MNIST» – це популярний набір даних у галузі машинного навчання, який представляє собою велику колекцію зображень

одягу та аксесуарів. Цей набір даних є аналогом вибірки MNIST із рукописними цифрами, але замість цифр від 0 до 9 містить зображення різних типів одягу та взуття.

Набір містить 60,000 тренувальних зображень та 10,000 тестових зображень, включає 10 класів одягу та аксесуарів, включаючи футболки, спідниці, плаття, кросівки, сумки, футболки з довгим рукавом та інші предмети одягу.

Усі зображення у наборі мають однаковий розмір 28x28 пікселів.

Кожен піксель у зображенні представляється значенням від 0 до 255, що відображає інтенсивність сірого кольору.

Головна мета цього набору даних – навчити моделі класифікації розпізнавати різні типи одягу на зображеннях. Це може бути використано для розробки системи рекомендацій у сфері моди, автоматичної класифікації товарів у магазинах та інших задач комп'ютерного зору.

Приклади даних із набору «Fashion MNIST» представлено на рисунку 3.2 та рисунку 3.3.

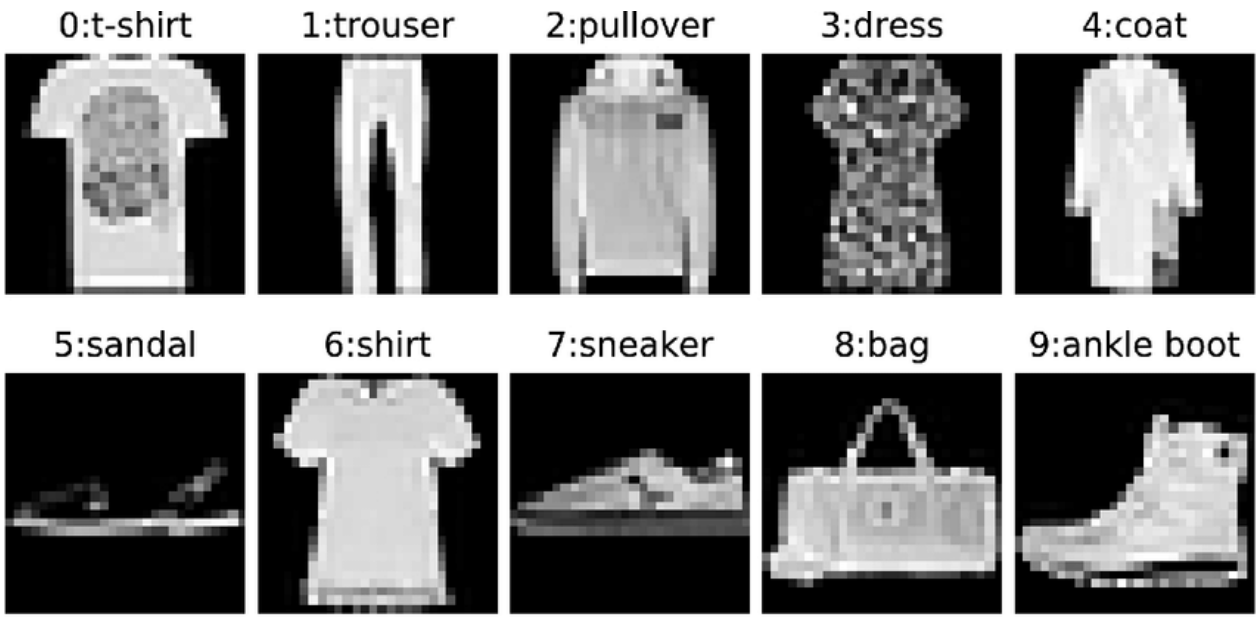

Рисунок 3.2 – Екземпляри вибірки «Fashion MNIST»

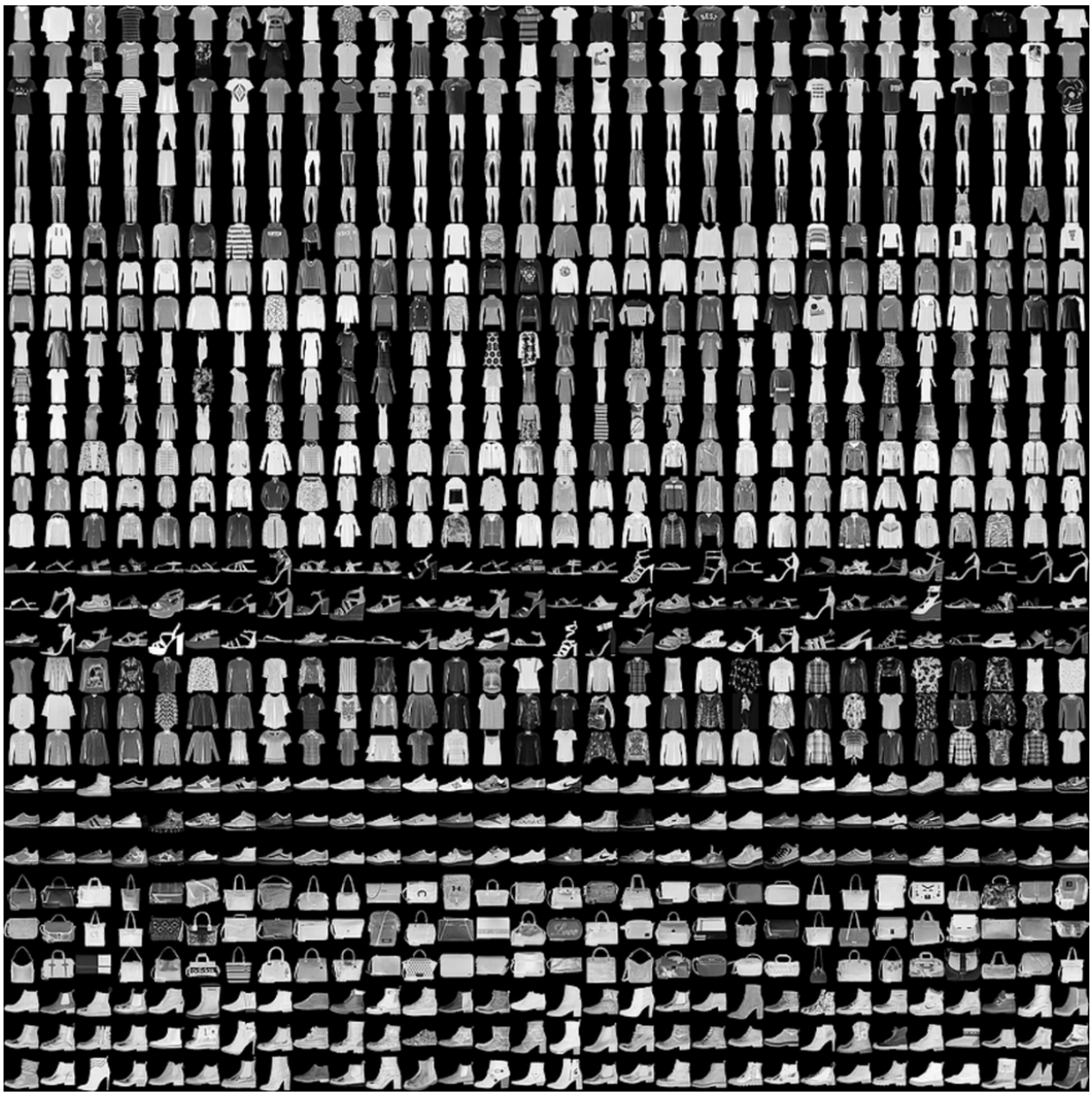

Рисунок 3.3 – Дані із вибірки «Fashion MNIST»

3.1.4 Тестова вибірка «ML hand-written digits»

Тестова вибірка «ML Hand-written Digits» – це набір даних, який часто використовується у галузі машинного навчання для валідації та тестування алгоритмів кластеризації та класифікації. Цей набір даних містить зображення рукописних цифр від 0 до 9, які були написані людиною. Кожне зображення має розмір 28 на 28 пікселів.

Ця тестова вибірка може включати будь-яку кількість зображень рукописних цифр для тестування моделей класифікації. Кожне зображення представляє одну з десяти цифр від 0 до 9.

Усі зображення мають однаковий розмір 28 на 28 пікселів. Кожен піксель у зображенні представляється значенням від 0 до 255, що відображає інтенсивність сірого кольору.

Головна мета цієї тестової вибірки – перевірити якість та точність роботи моделі класифікації на нових, раніше не бачених даних. Це дає змогу оцінити ефективність моделі поза тренувальним набором даних.

Приклади даних із набору «ML Hand-written Digits» представлено на рисунку 3.4 та рисунку 3.5.

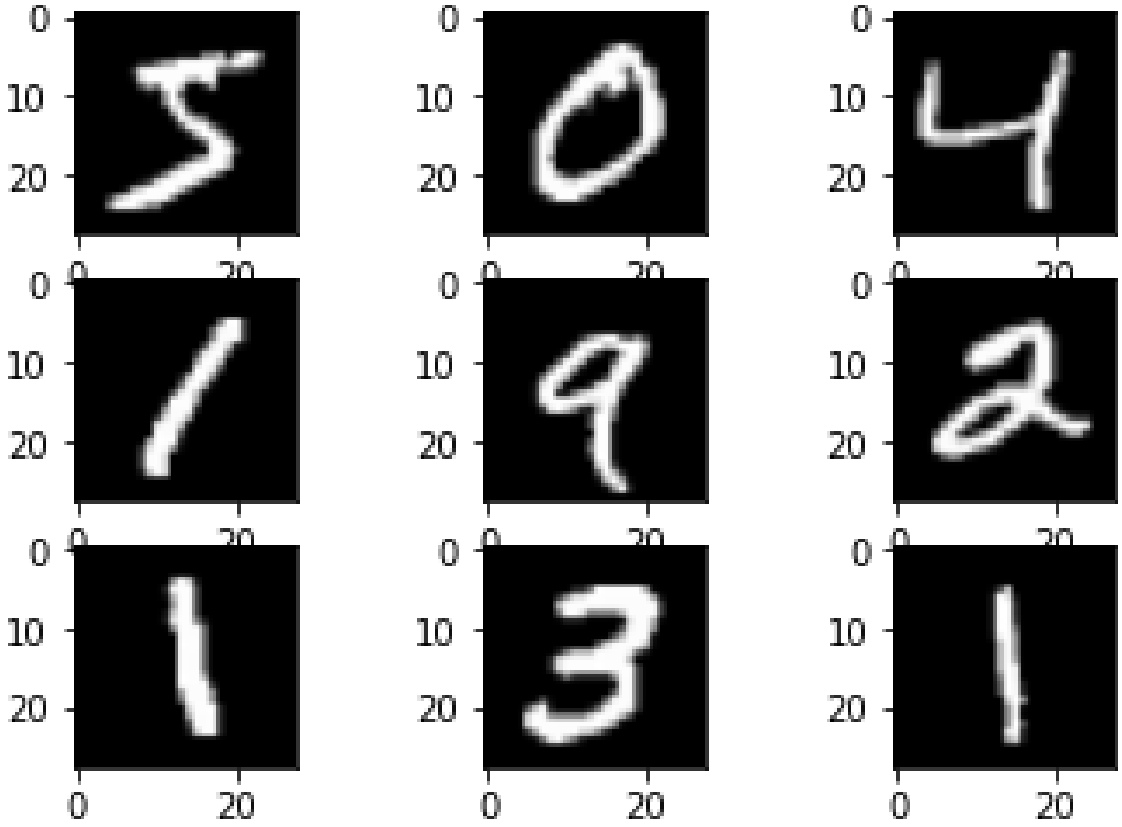

Рисунок 3.4 – Дані із вибірки «ML Hand-written Digits»

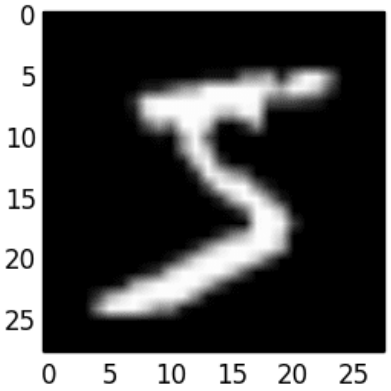

Рисунок 3.5 – Екземпляр із вибірки «ML Hand-written Digits»

Зрозуміло, в цій роботі передбачено, що зображення монохромні, тобто мають тільки один канал (наприклад, сірий колір).

Представлені набори даних містять класи, що перетинаються у просторі ознак, які представлені на рисунку 3.6 та на рисунку 3.7.

Усі приведені характеристики вибірок, підтверджують актуальність використання нечіткої ймовірнісної мережі, що розглядається.

Рисунок 3.6 – Приклад класів, що перетинаються із вибірки «ML Hand-written Digits»

Рисунок 3.7 – Приклад класів, що перетинаються із вибірки «Fashion MNIST»

3.2 Експериментальні дослідження

Експериментальна перевірка ймовірнісної нейро-фаззі системи була проведена для порівняння обраних критеріїв оцінки її роботи, а саме точності кластеризації та швидкості обробки вхідних даних. Ці критерії є важливими для визначення ефективності системи та можуть бути використані для оцінки її відповідності вимогам конкретного застосування.

Точність кластеризації вимірює, наскільки точно система може кластеризувати вхідні дані у відповідні класи чи категорії. Висока точність кластеризації свідчить про те, що система добре впоралася із завданням.

Швидкість обробки вхідних даних вимірює, як швидко система може обробити та кластеризувати вхідні дані. Швидка обробка даних є важливою, особливо в реальному часі або у великих масштабах, коли потрібно обробляти великий обсяг інформації.

Перший експеримент був проведений з коротким набором медичних даних «Heart Disease». Після цього набір даних був розбитий на підмножини з метою визначення мінімальної кількості елементів вибірки, яка потрібна для отримання певної якості результатів. Цей підхід може допомогти визначити оптимальний обсяг даних для аналізу та кластеризації в рамках даної задачі медичної діагностики. Розбиття набору даних на підмножини дозволяє здійснити більш глибокий аналіз та оптимізацію роботи системи в залежності від розміру набору даних, який доступний для аналізу.

Для того, щоби почати роботу з вибіркою даних, їх необхідно імпортувати, як показано на рисунку 3.8, а результат завантажених даних наведено на рисунку 3.9.

```
import pandas as pd
import matplotlib.pyplot as mp
import nuphy ad np

data = pd.read_csv('C:/Users/User/Desktop/Data_sets/Heart_Disease.csv')
df = pd.DataFrame(data)
df.head
```

Рисунок 3.8 – Імпорт вибірки даних «Heart Disease» до Python

	age	sex	ChestPain	RestBP	Cholestoral	Blood Sugar	ECG	MAXHeartRate	ExerciseAngina	oldpeak	slope	MajorVe
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.0
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.7
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.0
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.0
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.0
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.0
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.0
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.0

Рисунок 3.9 – Вибірка даних «Heart Disease» до Python

Були порівняні значення точності кластеризації ймовірнісної нейро-фаззі системи та популярного методу машинного навчання KNN (K-найближчих сусідів). Отримані результати експериментальної оцінки наведені в Таблиці 3.1. Це порівняння може допомогти визначити, яка система або метод краще справляється з кластеризацією медичних даних та має вищу точність. Таблиця 3.1 демонструє результати роботи експериментального дослідження.

Таблиця 3.1 – Порівняння точності кластеризації алгоритмів у випадку короткого набору даних «Heart Disease»

Метод кластеризації	**Точність кластеризації**				**Час(с)**
	250	**200**	**150**	**100**	
PNFS	77,52	69,34	57,69	52,10	0,11
EFPNN	79,02	71,84	62,0	56,06	0,12
KNN	49,03	50,69	51,66	51,1	0,15

Як видно із таблиці, найкращій результат демонструє ймовірнісна нейро-фаззі мережа при порівнянні точності і часу, незалежно від кількості спостережень.

При однаковій апаратній реалізації (зокрема, швидкість обробки даних) для всіх методів часові витрати, які стосуються швидкості обробки, будуть схожими. Однак точність ймовірнісної нейро-фаззі системи вища, що робить її більш ефективною та точною для класифікації медичних даних порівняно з методом *K*-найближчих сусідів (KNN).

Зокрема, було проведений якісний показник кластеризації даних за основними характеристиками, такими як індекс силуету – даний коефіцієнт не передбачає знання істинних міток об’єктів, і дозволяє оцінити якість кластеризації, використовуючи тільки саму (нерозмічену) вибірку і результат кластеризації [51]. Спочатку силует визначається окремо для кожного об’єкта. Позначимо через *n*-середня відстань від даного об’єкта до об’єктів з

того ж кластера, через *m*-середня відстань від даного об’єкта до об’єктів з найближчого кластера (відмінного від того, в якому лежить сам об’єкт). Тоді силуетом даного об’єкта називається величина:

$$SI = \frac{m - n}{\max(m, n)}.$$

Індекс Калінські-Харабаса – це метод оцінки якості кластеризації, який поєднує в собі дві метрики: компактність та розділеність кластерів. Компактність визначається за допомогою середньої відстані між кожною точкою в кластері та її центроїдом (це відстань між точкою та центроїдом кластера, до якого ця точка належить). Чим менше середні відстані, тим компактніше кластери. Розділеність визначається за допомогою відстані між центроїдами кластерів та глобальним центроїдом (центроїдом, що представляє усі дані). Ця відстань вказує на те, наскільки далеко розташовані кластери один від одного:

$$CH(C) = \frac{N - K}{K - 1} * \frac{\sum_{c_k \in C} |c_k| * \|\overline{c}_k - \overline{X}\|}{\sum_{c_k \in C} \sum_{x_i \in C} |c_k| * \|x_i - \overline{c}_k\|}.$$

Індекс Девіса-Болдуїна є однією з найбільш використовуваних мір оцінки якості кластеризації. Ця метрика враховує як компактність кластерів (відстань від об’єктів кластера до їхніх центроїдів), так і окремість кластерів (відстань між центроїдами різних кластерів).

Індекс Девіса-Болдуїна розраховується за наступною формулою:

$$DB(C) = \frac{1}{K} \sum_{c_k \in C} \max_{c_l \in C \setminus c_k} \left\{ \frac{S(c_k) + S(c_l)}{\|\overline{c}_k - \overline{c}_l\|} \right\}.$$

В таблиці 3.2 наведено якість кластеризації за основними характеристиками.

Таблиця 3.2 – Оцінка якості кластеризації

Методи кластеризації	**SI**	**CHI**	**DBI**
PNFS	0,2324	921,01	1,22
EFPNN	0,3324	964,42	1,08
KNN	0,3655	1418,28	1,08

Другий експеримент проведено на тестовій вибірці з UCI репозиторію «Diabetes 130-US hospitals for years 1999-2008»

Завантаження вибірки продемонстровано на рисунку 3.10 та результат виконання програмної реалізації на рисунку 3.11

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import time

# load the csv file
df=pd.read_csv('diabetic_data.csv')

print('Number of samples:' len(df))
Number of samples: 101766
```

Рисунок 3.10 – Завантаження тестової вибірки UCI репозиторію «Diabetes 130-US hospitals for years 1999-2008»

Для аналізу швидкості роботи методів, тестову вибірку даних було розбито на декілька різного розміру масиви від 3000 до 30000 екземплярів.

За результатами першого експерименту стало зрозуміло, що у подальших порівняльних дослідженнях доцільно фокусуватися на двох нейромережах – PNFS та EFPNN, оскільки вони забезпечують більш високу точність кластеризації. Експеримент, продемонстрований на рисунку 3.12, також показав, що ймовірнісна нейро-фаззі мережа вимагає менших обчислювальних витрат, ніж EFPNN.

	race_Asian	race_Caucasian	race_Hispanic	race_Other	race_UNK	gender_Male	gender_Unknown/Invalid	max_
0	0	1	0	0	0	0	0	
1	0	1	0	0	0	0	0	
2	0	0	0	0	0	0	0	
3	0	1	0	0	0	1	0	
4	0	1	0	0	0	1	0	

5 rows × 133 columns

Рисунок 3.11 – Результат виконання програмної реалізації

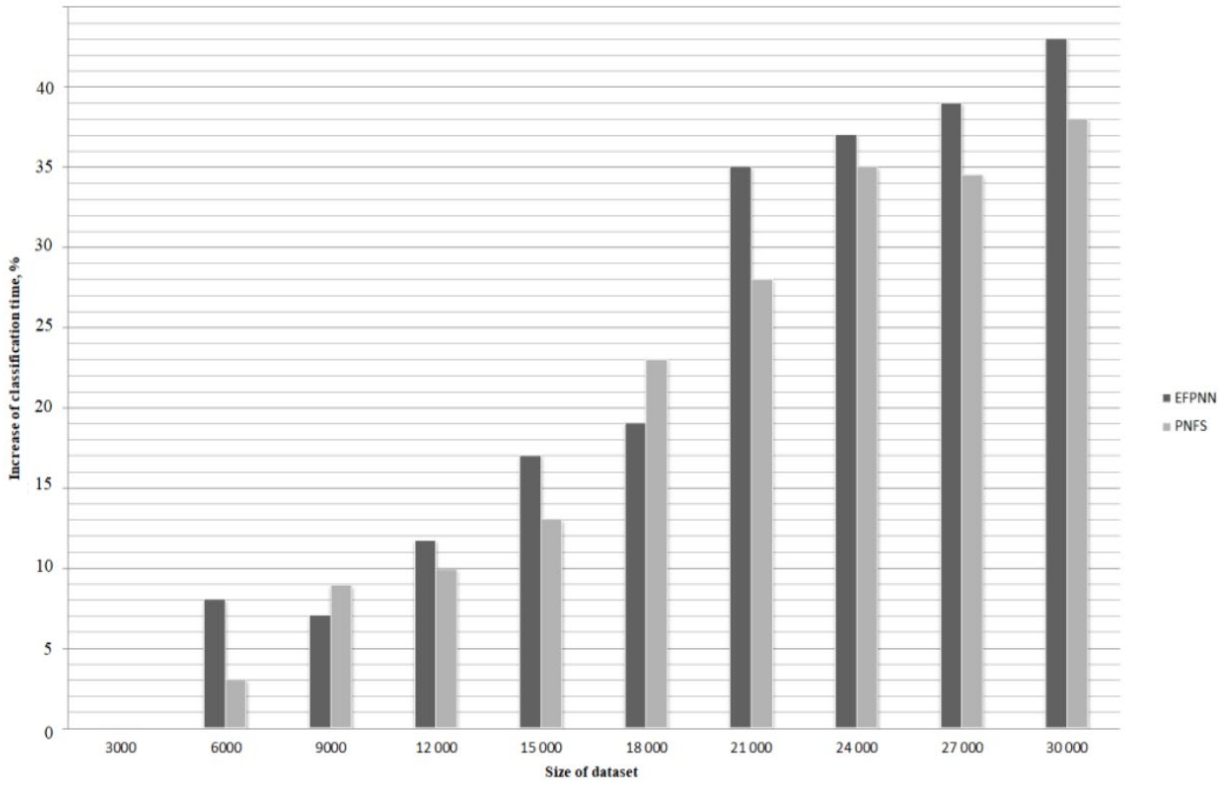

Рисунок 3.12 – Залежність швидкості роботи алгоритмів кластеризації від кількості спостережень

Приріст часу, необхідного для обробки підмножин великого розміру, значно збільшується порівняно з підмножинами маленького розміру. Це пов’язано з тим, що менші вибірки зазвичай можуть бути повністю завантажені в оперативну пам’ять комп’ютера, що дозволяє швидше та ефективніше обробляти їх.

У той час як довгі набори даних, які не вміщуються повністю в оперативну пам’ять, вимагають обміну даними із зовнішньої пам’яті (наприклад, з твердотільного накопичувача або з бази даних). Це призводить до затримок у читанні та записі даних, що може значно уповільнити процес обробки.

У випадку великих наборів даних, оптимізація роботи з пам’яттю та ефективний обмін даними можуть бути ключовими факторами для забезпечення продуктивності та ефективності алгоритмів обробки даних.

Отже, згідно з результатами двох експериментів видно, що запропонований підхід у порівнянні з EFPNN має свої переваги та недоліки. З одного боку, він забезпечує дещо нижчу точність кластеризації для малих наборів даних. Однак з іншого боку, він вимагає значно менших обчислювальних витрат, коли розмір набору даних зростає.

Це означає, що цей підхід може бути особливо корисним у випадках, коли важливо зберегти обчислювальні ресурси при обробці великих наборів даних, і ви готові прийняти деякий відхил від точності кластеризації. Однак для менших наборів даних, де точність дуже важлива, можливо, буде доцільно розглянути інші методи, такі як EFPNN, які забезпечують більшу точність, навіть за вартістю вищих обчислювальних витрат.

В наступному експерименті, який розглядається у даній роботі, використовувалися два набори даних: «Fashion MNIST» і «ML hand-written digits».

Для порівняльного аналізу були взяті популярний метод машинного навчання KNN (*K*-найближчих сусідів) [52, 53], Evolving Fuzzy-Probabilistic Neural Network (EFPNN) [54-56], та ймовірнісна нейро-фаззі система (PNFS) [57, 58]. Ці методи використовувалися для аналізу та класифікації даних у контексті обраних наборів «Fashion MNIST» та «ML hand-written digit».

Для того, аби завантажити вибірки, необхідно імпортувати необхідні бібліотеки та модулі як показано на рисунку 3.13:

```
import keras
from keras.datasets
import mnist
from keras.models
import Sequential
from keras.layers
import Dense, Dropuot, Flatten
from keras.layers
import Conv2D, maxPooling2D
from keras
import backend as K
```

Рисунок 3.13 – Імпорт необхідних бібліотек та модулів

Далі, розділення набору даних «Fashion MNIST» на тренувальну та тестову, як показано на рисунку 3.14

```
(x_train, y_train), (x_test, y_test) = mnist.load.data()
```

Рисунок 3.14 – Розділення набору даних «Fashion MNIST» на тренувальну та тестову

Після того, як вибірка буда розділена, необхідно переробити дані для подальшої роботи. На рисунку 3.15 продемонстрований препроцесінг даних «Fashion MNIST».

```
num_of_trainImg = x_train.shape[0]
num_of_testImg = x_test.shape[0]
img_width = 28
img_height = 28

x_train = x_train.reshape(x_train.shape[0], img_height, img_width, 1)
x_test = x_test.reshape(x_test.shape[0], img_height, img_width, 1)
input_shape = (img_height, img_width, 1)

x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train / =255
x_test /=255
```

Рисунок 3.15 – Попередня обробка вхідних даних

Під час першої частини даного експерименту був взятий набір даних «ML hand-written digits», а отримана точність кластеризації разом із затрачуваним часом фіксувалися. Програмна реалізація наведена на рисунку 3.16. Отримані результати представлені в таблиці 3.3.

```
score = model.evaluate (x_test, y_test, verbose = 0)
print ('Test loss:' score[0])
print('Test accuracy:' score[1])
```

Рисунок 3.16 – Програмна реалізація точності кластеризації

Таблиця 3.3 – Порівняння точності кластеризації алгоритмів

Методи кластеризації	**Точність кластеризації**	**Час (с)**
PNFS	81,98	0,18
EFPNN	96,39	7,02
KNN	93,07	5,52

Ці результати показують, що EFPNN має найвищу точність класифікації (96,39%), але вимагає значно більше часу на виконання

(7,02 секунди). PNFS має меншу точність (93,07%), але працює трохи швидше (5,52 секунди). KNN має найменшу точність (81,98%), але працює дуже швидко (0,18 секунди).

У другій частині експерименту використовувався набір даних «Fashion MNIST». У цьому етапі дослідження вихідний набір даних був розділений на 9 різних наборів, кожен із яких мав власний обсяг даних.

З початкового набору даних «Fashion MNIST» був сформований набір, який містить 15 000 спостережень. Крім того, параметр ширини активаційної функції був емпірично обраний для початкового набору даних з метою отримання найкращих результатів. Для довгого набору даних, цей параметр широкого поширення дорівнює 0,74. Результати експерименту представлені на рисунку 3.17.

Згідно з графіку, видно, що алгоритм KNN забезпечує найменше збільшення часу обчислення при збільшенні розміру набору даних. Однак, варто враховувати, що цей алгоритм має меншу точність порівняно з алгоритмами, заснованими на імовірнісних нейронних мережах, такими як EFPNN та ймовірнісна нейро-фаззі система (PNFS). Вибір між часом обчислення та точністю класифікації залежить від конкретних потреб і вимог до конкретної задачі.

Алгоритм, який розглядається у цій роботі – PNFS, вимагає менших обчислювальних навантажень порівняно з алгоритмом EFPNN, при цьому зберігаючи подібну точність класифікації. На наборах даних з більшим обсягом, які перевищують 8000 елементів, цей алгоритм проявляє майже лінійний ріст обчислювальних витрат зі збільшенням розміру вибірки.

Якість кластеризації вибірки «Fashion MNIST» був проведений в декілька етапів на 5000, 10000 та 15000 спостережень. Результати наведені в таблиці 3.4.

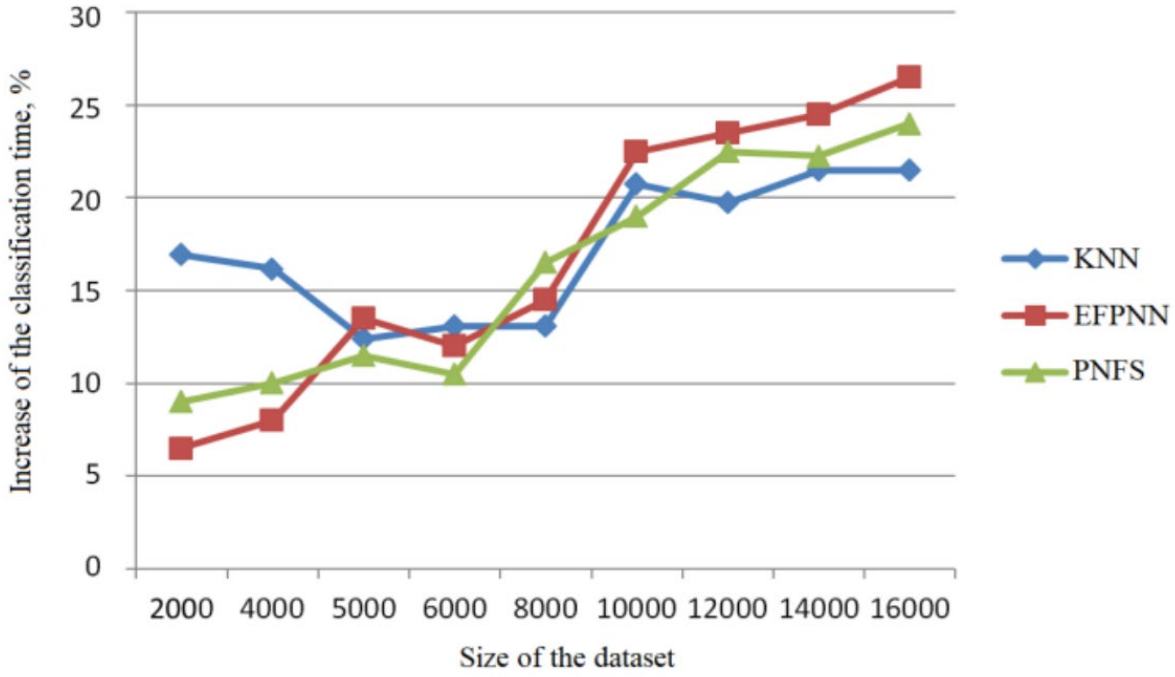

Рисунок 3.17 – Графік залежності часу кластеризації від розміру вхідних даних

Таблиця 3.4 – Оцінка якості кластеризації вибірки «Fashion MNIST»

Методи кластеризації	**SI**	**CHI**	**DBI**
	5000 спостережень		
PNFS	0,3324	928,01	1,3
EFPNN	0,3335	974,42	1,10
KNN	0,3665	1420,28	1,15
	10000 спостережень		
PNFS	0,6045	1460,65	1,5
EFPNN	0,6122	1945,5	1,25
KNN	0,7222	2800,05	1,32
	15000 спостережень		
PNFS	0,8324	1560,00	1,8
EFPNN	0,8735	2000,28	1,15
KNN	0,9885	2985,2	1,28

Індекс силуету (SI) вимірює, наскільки кожен об’єкт в кластері подібний до інших об’єктів у тому ж кластері порівняно з об’єктами інших кластерів. За значенням від 0 до 1, вищі значення SI вказують на кращу якість кластеризації. В таблиці 3.4 PNFS має найвищі значення SI для всіх трьох обсягів даних, що свідчить про його здатність ефективно відокремлювати кластери навіть у великих обсягах даних.

Індекс Калінські-Харабаса (CHI) вимірює, наскільки добре кластери відокремлені один від одного. Високі значення CHI вказують на кращу якість кластеризації. В таблиці 3.4 PNFS також має найвищі значення CHI для всіх обсягів даних, що означає, що кластери, сформовані PNFS, добре відокремлені один від одного.

Індекс Девіса-Болдуїна (DBI) вимірює середню відстань між кластерами і внутрішньокластерну відстань. Низькі значення DBI свідчать про кращу якість кластеризації. У вас KNN має найменші значення DBI для 5000 спостережень, вказуючи на кращу якість кластеризації для цього обсягу даних. Однак для більших обсягів даних (10000 та 15000 спостережень), EFPNN та PNFS мають менші значення DBI, що свідчить про їхню кращу якість кластеризації.

Отже, з цих результатів можна вивести висновок, що метод PNFS показує кращу або аналогічну якість кластеризації порівняно з EFPNN та KNN для великих обсягів даних, а також добре справляється з меншими обсягами даних.

ВИСНОВКИ

В кваліфікаційні роботі представлено результати роботи з удосконалення ймовірнісної нейро-фаззі системи та її гібридного навчання. Головна увага була спрямована на вирішення актуальної проблеми обробки потоків даних у режимі онлайн. Розглянули різні аспекти цієї задачі, включаючи методи кластеризації, оцінку якості кластеризації за допомогою різних індексів, а також порівняння різних методів машинного навчання, таких як KNN, EFPNN та PNFS.

Проведено аналіз сучасного стану теорії штучних нейронних мереж, спрямованих на швидку обробку потоків даних у режимі онлайн, підкреслив необхідність інтеграції швидкодіючих ймовірнісних нейронних мереж для вирішення обмежень у часі при послідовній обробці великих обсягів даних. Також визначено переваги використання нечіткої логіки для виділення класів, які перетинаються в просторі ознак.

Цей аналіз підкреслив потребу у розробці інноваційних підходів, які поєднують в собі швидкодіючі ймовірнісні нейронні мережі та нечітку логіку. Ця інтеграція може забезпечити не тільки швидку обробку великих обсягів даних, але й здатність впоратися з задачами, де класи перетинаються у просторі ознак, що є типовою ситуацією в багатьох реальних даних.

Підкреслено важливість розвитку нових методів, які можуть оптимально використовувати можливості штучних нейронних мереж та нечіткої логіки для вирішення цих викликів у реальному часі. Це може мати значущий вплив на області, де швидка обробка та точність кластеризації є критичними, такі як медична діагностика, фінансові ринки та інші сфери, де важливо швидко реагувати на поточні події та тренди.

Розглянуто аспекти гібридного навчання, де ймовірнісна нейро-фаззі система була поєднана з іншими методами для досягнення кращих результатів у вирішенні задачі обробки потоків даних. Розроблена

архітектура ймовірнісної нейро-фаззі системи із гібридним навчанням містить кілька шарів, які спільно працюють для кластеризації даних.

Експериментальна перевірка можливостей ймовірнісної нейро-фаззі системи була проведена за критеріями точності кластеризації та швидкості обробки вхідних даних. Під час експериментів використовувалися як числові, так і бінарні вхідні дані. Результати досліджень свідчать, що розроблена ймовірнісна нейро-фаззі система демонструє кращу або аналогічну якість кластеризації порівняно з EFPNN та KNN для великих обсягів даних, а також добре справляється з меншими обсягами даних. Це створює умови для швидкої адаптації системи при зміні типів вхідних даних.

Важливо відзначити, що при збільшенні розміру вибірки розроблена мережа вимагає менше часу для обробки даних порівняно з адаптивною ймовірнісною нейронною мережею. Це свідчить про ефективність та швидкість ймовірнісної нейро-фаззі системи, зокрема при роботі з об’ємними наборами даних.

Узагальнюючи, запропонована ймовірнісна нейро-фаззі система з гібридним навчанням демонструє ефективну та швидку обробку потоків даних у режимі онлайн. Ця система виявилася особливо корисною при роботі з об’ємними даними, забезпечуючи високу точність кластеризації та ефективність в умовах зміни типів вхідних даних. Її гнучкість та здатність до адаптації роблять її потужним інструментом для різноманітних завдань у сфері обробки даних.

Результати роботи апробовано у вигляді тез доповідей під час IV Міжнародної науково-теоретичної конференції «МОДЕРНІЗАЦІЯ СУЧАСНОЇ НАУКИ: ДОСВІД ТА ТЕНДЕНЦІЇ», Сінгапур, [59] та VIII міжнародній науковій конференції «Розвиток науки у XXI столітті», Дортмунд, Німеччина [60].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Kohonen, T. (1991). Self-organizing maps: Ophmization approaches. In *Artificial neural networks* (pp. 981-990). North-Holland.

2. Gorban, A. N., Kégl, B., Wunsch, D. C., & Zinovyev, A. Y. (Eds.). (2008). *Principal manifolds for data visualization and dimension reduction* (Vol. 58, pp. 96-130). Berlin: Springer.

3. Ротштейн, О. П. (1999). Інтелектуальні технології ідентифікації: нечіткі множини, генетичні алгоритми, нейронні мережі. *Вінниця: Універсум-Вінниця*.

4. Han, J., Pei, J., & Tong, H. (2022). *Data mining: concepts and techniques*. Morgan kaufmann.

5. Han, J., Kamber, M., & Mining, D. (2006). Concepts and techniques. *Morgan kaufmann*, *340*, 94104-3205.

6. Uchino, E., & Yamakawa, T. (1997). Soft computing based signal prediction, restoration, and filtering. *Intelligent hybrid systems: fuzzy logic, neural networks, and genetic algorithms*, 331-351.

7. Kasabov, N. K. (2015). Evolving connectionist systems for adaptive learning and knowledge discovery: Trends and directions. *Knowledge-Based Systems*, *80*, 24-33.

8. de Jesús Rubio, J., & Bouchachia, A. (2017). MSAFIS: an evolving fuzzy inference system. *Soft Computing*, *21*, 2357-2366.

9. Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

10. Tkacz, M. (2005). Artificial neural networks in incomplete data sets processing. In *Intelligent Information Processing and Web Mining: Proceedings of the International IIS: IIPWM'05 Conference held in Gdansk, Poland, June 13–16, 2005* (pp. 577-583). Springer Berlin Heidelberg.

11. Hoppner, F. (1999). *Fuzzy cluster analysis: methods for classification, data analysis and image recognition*. John Wiley.

12. Lughofer, E. (2011). *Evolving fuzzy systems-methodologies, advanced concepts and applications* (Vol. 53). Berlin: Springer.

13. Pal, J., & Bhattacherjee, V. (2012). Application of fuzzy clustering on software quality using max-min method. *Procedia Technology*, *6*, 67-73.

14. De Almeida, C. W., De Souza, R. M., & Candeias, A. L. (2013). Fuzzy Kohonen clustering networks for interval data. *Neurocomputing*, *99*, 65-75.

15. Волкова, В. В., & Шафроненко, А. Ю. (2011). Нечітка кластеризація масивів даних з пропущеними значеннями. *Індуктивне моделювання складних систем*.

16. А. Шафроненко Є. Бодянський, І. Плісс (2022). *Нечіткі методи інтелектуального аналізу даних.* GlobeEdit.

17. Shafronenko, A., Dolotov, A., Bodyanskiy, Y., & Setlak, G. (2018, August). Fuzzy clustering of distorted observations based on optimal expansion using partial distances. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)* (pp. 327-330). IEEE.

18. Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, *23*(3), 665-685.

19. Jang, J. S., & Sun, C. T. (1993). Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE transactions on Neural Networks*, *4*(1), 156-159.

20. Krishnapuram, R., & Keller, J. M. (1993). A possibilistic approach to clustering. *IEEE transactions on fuzzy systems*, *1*(2), 98-110.

21. Specht, D. F. (1990). Probabilistic neural networks and the polynomial adaline as complementary techniques for classification. *IEEE Transactions on Neural Networks*, *1*(1), 111-121.

22. Marwala, T. (Ed.). (2009). *Computational Intelligence for Missing Data Imputation, Estimation, and Management: Knowledge Optimization Techniques: Knowledge Optimization Techniques*. IGI Global.

23. Gan, G., Ma, C., & Wu, J. (2020). *Data clustering: theory, algorithms, and applications*. Society for Industrial and Applied Mathematics.

24. Shafronenko, A. Y., Kasatkina, N. V., Bodyanskiy, Y. V., & Shafronenko, Y. O. (2023). CREDIBILISTIC ROBUST ONLINE FUZZY CLUSTERING IN DATA STREAM MINING TASKS. *Radio Electronics, Computer Science, Control*, (3), 93-97.

25. Bodyanskiy, Y. V., Shafronenko, A., & Rudenko, D. (2019). Online Neuro Fuzzy Clustering of Data with Omissions and Outliers based on Completion Strategy. In *CMIS* (pp. 18-27).

26. Bodyanskiy, Y., Shafronenko, A., & Mashtalir, S. (2020). Online robust fuzzy clustering of data with omissions using similarity measure of special type. In Lecture Notes in Computational Intelligence and Decision Making*: Proceedings of the XV International Scientific Conference "Intellectual Systems of Decision Making and Problem of Computational Intelligence"(ISDMCI'2019), Ukraine, May 21–25, 2019 15 (pp. 637-646*). Springer International Publishing.

27. Shafronenko, A. Y., & Rudenko, D. A. (2020). ONLINE RECURRENT METHOD OF CREDIBILISTIC FUZZY CLUSTERING. *BBK 91*, 37.

28. Shafronenko, A., Bodyanskiy, Y., & Rudenko, D. (2020). *Neuro-fuzzy clustering of Distorted Data Using Cat Swarm Optimization*. LAP LAMBERT Academic Publishing.

29. Bodyanskiy, Y., Shafronenko, A., & Pliss, I. (2021). Правдоподібна нечітка кластеризація даних на основі еволюційного методу божевільних котів. *System research and information technologies*, (3), 110-119.

30. Xu, R., Xu, J., & Wunsch, D. C. (2012). A comparison study of validity indices on swarm-intelligence-based clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42(4), 1243-1256

31. Шафроненко, А. Ю., & Бодянський, Є. В. (2023). Адаптивний підхід до нечіткої кластеризації на основі еволюційної оптимізації алгоритму сірих вовків. *Збірник наукових праць Харківського національного університету Повітряних Сил*, (1 (75)), 77-81.

32. Шафроненко, А. Ю., Бодянський, Є. В., & Руденко, Д. О. (2023). Модифікований рекурентний метод достовірної нечіткої кластеризації з

використанням оптимізаційної процедури на основі косяків риб. *Системи обробки інформації*, (1 (172)), 92-96.

33. Dracopoulos, D. C. (2013). *Evolutionary learning algorithms for neural adaptive control*. Springer.

34. Bodyanskiy, Y., Pliss, I., & Shafronenko, A. (2022). Adaptive neuro-fuzzy clustering of distorted data based on prototype-centroid strategy using evolutionary procedures. *Artificial Intelligence*, *27*, 239-244.

35. Шафроненко, А. Ю., & Бодянський, Є. В. (2023). Нечітка достовірна кластеризація великих масивів даних з гіпереліпсоїдальними класами з довільною орієнтацією осей. *Наука і техніка Повітряних Сил Збройних Сил України*, (1 (50)), 93-99.

36. Bodyanskiy, Y. V., Shafronenko, A., & Klymova, I. (2021, April). Adaptive Recovery of Distorted Data Based on Credibilistic Fuzzy Clustering Approach. In *COLINS* (pp. 6-15).

37. Wunsch, D. C., Rossiev, A., & Gorban, A. N. (2000). Neural Network Modeling of Data with Gaps.

38. Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

39. Haykin, S. (1998). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

40. Braun, H. (2013). *Neuronale Netze: Optimierung durch Lernen und Evolution*. Springer-Verlag.

41. Плісс, І. П., Шевякова, А. Ю., & Шевякова, Ю. Ю. (2011). Нейромережеве відновлення пропусків у таблицях даних. *Наукові праці [Чорноморського державного університету імені Петра Могили]. Сер.: Комп'ютерні технології*, (160, Вип. 148), 59-61.

42. Bodyanskiy, Y., Shafronenko, A., & Volkova, V. (2012). Adaptive fuzzy probabilistic clustering of incomplete data. *INFORMATION MODELS & ANALYSES*, *112*.

43. Shafronenko, A. Y., Bodyanskiy, Y. V., & Rudenko, D. A. (2020).

Adaptive Neuro-Fuzzy Methods for Distorted Data Clustering.

44. Kalton, G. (1986). The treatment of missing survey data. *Survey methodology*, *12*, 1-16.

45. Bodyanskiy, Y., Shafronenko, A., & Volkova, V. (2013, September). Neuro fuzzy Kohonen network for incomplete data clustering using optimal completion strategy. In *Proceedings 20th East West Fuzzy Colloquium 2013, Zittau* (pp. 214-223).

46. Yamakawa, T. (1992). A neo fuzzy neuron and its applications to system identification and prediction of the system behavior. In *Proc. of the 2nd Int. Conf. on Fuzzy Logic & Neural Networks* (pp. 477-483).

47. Bodyanskiy, Y. V., Pliss, I. P., Shafronenko, A. Y., & Kalynychenko, O. V. (2022). НЕЧІТКА ДОВІРЧА КЛАСТЕРИЗАЦІЯ ДАНИХ НА ОСНОВІ АНАЛІЗУ ЩІЛЬНОСТІ РОЗПОДІЛУ ДАНИХ ТА ЇХ ПІКІВ. *Radio Electronics, Computer Science, Control*, (3), 58-58.

48. Bodyanskiy, Y., Shafronenko, A., Klymova, I., & Polyvoda, V. (2022). Robust Recurrent Credibilistic Modification of the Gustafson-Kessel Algorithm. In *Lecture Notes in Computational Intelligence and Decision Making: 2021 International Scientific Conference" Intellectual Systems of Decision-making and Problems of Computational Intelligence", Proceedings* (pp. 613-623). Springer International Publishing.

49. Asuncion, A., & Newman, D. (2007). UCI machine learning repository.

50. Frank, A. (2010). UCI machine learning repository. *http://archive. ics. uci. edu/ml*.

51. Lenssen, L., & Schubert, E. (2022, September). Clustering by Direct Optimization of the Medoid Silhouette. In *International Conference on Similarity Search and Applications* (pp. 190-204). Cham: Springer International Publishing.

52. Cayton, L. (2008, July). Fast nearest neighbor retrieval for bregman divergences. In *Proceedings of the 25th international conference on Machine learning* (pp. 112-119).

53. Weber, R., Schek, H. J., & Blott, S. (1998, August). A quantitative

analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB* (Vol. 98, pp. 194-205).

54. Bodyanskiy, Y., Gorshkov, Y., Kolodyazhniy, V., & Wernstedt, J. (2003). A learning probabilistic neural network with fuzzy inference. In *Artificial Neural Nets and Genetic Algorithms: Proceedings of the International Conference in Roanne, France, 2003* (pp. 13-17). Vienna: Springer Vienna.

55. Bodyanskiy, Y. V., Gorshkov, Y., & Kolodyazhniy, V. (2003, September). Resource-allocating probabilistic neuro-fuzzy network. In *EUSFLAT Conf.* (pp. 392-395).

56. Bodyanskiy, Y., Gorshkov, Y., Kolodyazhniy, V., & Wernstedt, J. (2003, September). Probabilistic neuro-fuzzy network with non-conventional activation functions. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems* (pp. 973-979). Berlin, Heidelberg: Springer Berlin Heidelberg.

57. Rutkowski, L. (2004). Adaptive probabilistic neural networks for pattern classification in time-varying environment. *IEEE transactions on neural networks*, *15*(4), 811-827.

58. Kaski, S., Kangas, J., & Kohonen, T. (1998). Bibliography of self-organizing map (SOM) papers: 1981–1997. *Neural computing surveys*, *1*(3&4), 1-176.

59. Руденко, Д., Лотвінова, В., & Безверха, Є. (2023). НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ В ЗАДАЧАХ ОБРОБКИ ДАНИХ. *Collection of scientific papers «SCIENTIA»*, (September 22, 2023; Singapore, Singapore), 94-95.

60. Rudenko, D. O., Bezverkha, Ye. V., Lotvinova, V. V. (2023). NEURAL NETWORK RECOVERY OF OMISSIONS IN DATA SETS. *VIII international scientific conference*, (September 14-15, 2023; Dortmund, Germany).