

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Ймовірнісна нейро-фаззі система та її гібридне навчання в задачах аналізу
потоків даних
(тема)

Виконав:
студент 2 курсу, групи СШМ-19-1
Чала О. С.
(прізвище, ініціали)

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

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного
інтелекту
(повна назва спеціалізації)

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

Допускається до захисту

Зав. кафедри _____
(підпис)

В.О. Філатов
(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

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

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_____» _____ 20__ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Чалій Ользі Сергіївні _____
(прізвище, ім'я, по батькові)

1. Тема роботи Ймовірнісна нейро-фаззі система та її гібридне навчання в задачах аналізу потоків даних

затверджена наказом університету від 30 жовтня 2020_р. № 1497Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 20__ р.

3. Вихідні дані до роботи Науково-технічні публікації щодо дослідження та розробки ймовірнісних і нечітких нейронних мереж, а також нейро-фаззі систем; дані відомих наукових проектів щодо розробки та дослідження алгоритмів аналізу потоків даних, опис тестових вибірок із UCI-репозиторія, інші Інтернет-джерела та література з вказаної теми.

4. Перелік питань, що потрібно опрацювати в роботі Мета роботи; аналіз предметної області і постановка задачі дослідження; порівняльний аналіз можливостей ймовірнісних та нечітких нейронних мереж до класифікації потоків даних; розробка архітектури ймовірнісної нейро-фаззі системи; експериментальна перевірка розробленої нейро-фаззі системи.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1 – Архітектура ймовірнісної нейронної мережі; Рисунок 2 – Архітектура ймовірнісної нечіткої нейромережі; Рисунок 3 – Приклад спостережень з набору даних «Fashion MNIST»; Рисунок 4 – Характеристики набору даних «ML hand-written digits»; Рисунок 5 – Приклад спостережень з набору даних «ML hand-written digits»; Рисунок 6 – Приклад перетинних класів з набору даних «ML hand-written digits»; Рисунок 7 – Приклад перетинних класів з «Fashion MNIST»; Рисунок 8 – Графік збільшення часу класифікації в залежності від розміру набору даних; Рисунок 9 – Графік залежності часу класифікації від розміру набору даних

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 |
| Основна частина | проф. Бодяньський Є.В. | | |
| | | | |

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Терміни виконання етапів роботи | Примітка |
|----|--|---------------------------------|----------|
| 1 | Отримання завдання на атестаційну роботу | 02.11.20 | виконано |
| 2 | Аналіз предметної області і постановка завдання | 03.11.20-05.11.20 | виконано |
| 3 | Порівняльний аналіз можливостей ймовірнісних та нечітких нейронних мереж до класифікації потоків даних | 06.11.20-08.11.20 | виконано |
| 4 | Розробка архітектури ймовірнісної нейро-фаззі системи | 09.11.20-13.11.20 | виконано |
| 5 | Експериментальна перевірка розробленої нейро-фаззі системи | 14.11.20-18.11.20 | виконано |
| 6 | Обробка і оформлення результатів | 19.11.20-21.11.20 | виконано |
| 7 | Оформлення пояснювальної записки | 23.11.20-30.11.20 | виконано |
| 8 | Нормоконтроль | 2.12.20 | виконано |
| 9 | Попередній захист | 11.12.20 | виконано |
| 10 | Захист перед ЕК | 15.12.20 | виконано |

Дата видачі завдання 2 листопада 2020 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Бодяньський Є.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Записка пояснювальна: 79 с., 9 рис., 2 табл., 2 дод., 59 джерел.

АРХІТЕКТУРА НЕЙРОННОЇ МЕРЕЖІ, ГІБРИДНЕ НАВЧАННЯ,
ЙМОВІРНІСНІ НЕЙРОННІ МЕРЕЖІ, ЗАДАЧА АНАЛІЗУ ПОТОКІВ
ДАННИХ, КЛАСИФІКАЦІЯ ДАНИХ.

Метою даної атестаційної роботи є розробка архітектури ймовірнісної нейро-фаззі системи для підвищення ефективності класифікації послідовності даних, що надходять у онлайн-режимі та мають різні типи вхідних даних.

Об'єкт дослідження – процес класифікації потоків даних, які можуть бути представлені у числовій, номінальній (бінарній) та ранговій шкалах.

Предмет дослідження – архітектури ймовірнісних нейронних мереж та методи їх навчання.

В ході роботи було спроектована та реалізована за допомогою мови програмування Python архітектура ймовірнісної нейро-фаззі системи.

Запропонована система вирізняється швидкою класифікацією потоку даних, які можуть бути представлені у номінальній (бінарній), ранговій та числовій шкалах. Система забезпечує класифікацію у випадку, коли класи даних можуть довільно перетинатися у просторі ознак.

РЕФЕРАТ

Пояснительная записка: 79 с., 9 рис., 2 табл., 2 прил., 59 источников.

АРХИТЕКТУРА НЕЙРОННОЙ СЕТИ, ВЕРОЯТНОСТНЫЕ НЕЙРОННЫЕ СЕТИ, ГИБРИДНОЕ ОБУЧЕНИЕ, ЗАДАЧА АНАЛИЗА ПОТОКОВ ДАННЫХ, КЛАСИФИКАЦИЯ ДАННЫХ.

Целью данной аттестационной работы является разработка архитектуры вероятностной нейро-фаззи системы для повышения эффективности классификации последовательности данных, поступающих в онлайн-режиме и имеют различные типы входных данных.

Объект исследования – процесс классификации потоков данных, которые могут быть представлены в числовой, номинальной (бинарной) и ранговой шкалах.

Предмет исследования – архитектуры вероятностных нейронных сетей и методы их обучения.

В ходе работы было спроектирована и реализована с помощью языка программирования Python архитектура вероятностной нейро-фаззи системы.

Предложенная система отличается быстрой классификации потока данных, которые могут быть представлены в номинальной (бинарной), ранговой и числовой шкалам. Система обеспечивает классификацию в случае, когда классы данных могут произвольно пересекаться в пространстве признаков.

ABSTRACT

Explanatory note: 79 pages, 9 figures, 2 tables, 2 annexes, 59 sources.

**DATA CLASSIFICATION, DATA STREAM MINING TASKS,
HYBRID LEARNING, NEURAL NETWORK ARCHITECTURE,
PROBABILISTIC NEURAL NETWORKS**

The aim of this certification work is to develop an architecture of probabilistic neuro-fuzzy system to increase the efficiency of classification of data sequences that come online and have different types of input data.

The object of study – the process of classification of data streams, which can be represented in numerical, nominal (binary) and rank scales.

The subject of research – the architecture of probabilistic neural networks and methods of their training.

In the course of the work, the architecture of the probabilistic neuro-fuzzy system was designed and implemented using the Python programming language.

The proposed system will feature a rapid classification of the data stream, which can be represented in nominal (binary), ranking and numerical scales. The system provides classification in the case when data classes can arbitrarily intersect in the feature space.

ЗМІСТ

| | |
|---|----|
| Перелік умовних позначень, символів, одиниць, скорочень і термінів | 8 |
| Вступ..... | 9 |
| 1. Аналіз предметної області і постановка задачі дослідження | 11 |
| 1.1 Задача аналізу потоків даних..... | 11 |
| 1.2 Поняття нейронних мереж | 14 |
| 1.3 Ймовірнісні нейронні мережі | 19 |
| 1.4 Нечіткі нейронні мережі..... | 24 |
| 1.5 Постановка задачі дослідження..... | 32 |
| 2 Ймовірнісна нейро-фаззі система та її гібридне навчання в задачах аналізу потоків даних | 34 |
| 2.1 Еволюційна нечітка ймовірнісна нейронна мережа та її онлайн- навчання | 34 |
| 2.2 Адаптивна ймовірнісна нейронна мережа з нечітким висноуванням та її онлайн навчання | 40 |
| 2.3 Ймовірнісна нечітка нейромережа та її гібридне навчання | 50 |
| 3 Експериментальна перевірка ймовірнісної нейро-фаззі системи при класифікації потоків даних | 57 |
| 3.1 Опис вхідних наборів даних | 57 |
| 3.2 Аналіз результатів експериментів..... | 62 |
| Висновки | 67 |
| Перелік використаних джерел | 69 |
| Додаток А Програмна реалізація..... | 76 |
| Додаток Б Відомість атестаційної роботи | 79 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ANN – Artificial Neural Network – штучна нейронна мережа;

APNN – Adaptive Probabilistic Neural Network – Адаптивна ймовірнісна нейронна мережа з нечітким висноуванням;

EFPNN – Evolving Fuzzy-Probabilistic Neural Network – Еволюційна нечітка ймовірнісна нейронна мережа;

NFS – Neuro-Fuzzy System – нейро-фаззі система;

NN – Neural Network – нейронна мережа;

PNFS – Ймовірнісна нечітка нейромережа з одновимірними функціями активації-належності;

PNN – Probabilistic Neural Network – ймовірнісна нейронна мережа.

ВСТУП

Вирішення задач обробки потоків даних є однією з найактуальніших проблем у сфері штучного інтелекту. Проте в реальних задачах, які пов'язані з обробкою великих масивів даних, вхідні дані надходять з високою швидкістю, а алгоритми, які повинні їх обробляти, мають дуже жорсткі обмеження на час та обчислювальні ресурси. З урахуванням наведених обмежень, існує ряд методів аналізу з низькими вимогами до пам'яті та незначним часом, що витрачається на обчислення кожного елемента з набору даних. Одним із методів, за допомогою якого досягається така перевага, є зменшення часу є ковшного вікна. Однак через зміни даних у часі, виникає необхідність до адаптації, яка потребує реалізації швидкого навчання.

Штучні нейронні мережі є досить популярним методом для обробки інформації для вирішення складних задач у сфері обчислювального інтелекту. На їх основі можна успішно вирішувати задачі розпізнавання природної мови та зображень, здійснення прогнозів, виявлення шахрайства (fraud detection), діагностики захворювань людини й багато інших.

Штучні нейронні мережі мають ряд переваг, що забезпечують їх популярність в задачах аналізу даних: здатність до навчання, тобто вони можуть підлаштовуватися під набір даних, що обробляється, тим самим підвищуючи точність класифікації; толерантність до шуму та можливість паралельно обробляти інформацію, що дозволяє мережі вирішувати ті завдання, які погано виконувались традиційними комп'ютерами, а також прискорює її роботу.

Однак, незважаючи на вагомі переваги, традиційні штучні нейронні мережі мають недолік, який ускладнює їх використання при аналізі даних з реальних об'єктів: вони не дозволяють класифікувати об'єкти, які із різним ступенем достовірності належать до різних класів (або кластерів).

З іншого боку, за останні роки, нечіткі системи стали досить популярним підходом у багатьох наукових додатках та інженерних системах, в системах управління та розпізнавання образів. Використання нечіткості дає можливість реалізувати більш гнучкий підхід до вирішення задач аналізу та обробки даних, а також дає можливість легко пояснити результати, отримані за допомогою систем нечітких правил. Недоліком нечітких мереж є те, що вони не можуть автоматично навчатись і у подальшому використати автоматично отримані знання у механізмі висновування, що обмежує вирішення складних задач, в тому числі задач аналізу потоків даних.

Таким чином, вирішення задачі аналізу потоків даних потребує розробки швидкої нейронної мережі для обробки інформації в онлайн режимі. Для її вирішення доцільно використовувати швидкі ймовірнісні нейронні мережі. Застосування нечіткої логіки дозволить прискорити роботу системи та підвищити її точність. Проте, через неперервний потік даних система може стати досить громіздкою, тому важливо використовувати метод ковзного вікна з метою зменшення витрат часу та потреб у пам'яті.

У магістерській атестаційній роботі виконано удосконалення ймовірнісної нечіткої нейронної мережі для підвищення її швидкодії та забезпечення можливостей навчання із вибіркою менших розмірів, що створює умови для їх ефективного використання при виконанні класифікації в режимі онлайн.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

ДОСЛІДЖЕННЯ **Equation Section 1**

1.1 Задача аналізу потоків даних

Останні десятиліття дослідження у сфері машинного навчання були орієнтовані на пакетне навчання, переважно з використанням невеликих наборів даних. При пакетному навчанні доступним є повний набір навчальних даних. Вважається, що приклади для навчання генеруються випадковим чином відповідно до деякого стаціонарного розподілу ймовірностей. При використанні малих наборів даних основною проблемою є зменшення дисперсії. З іншого боку, навчання з використанням великих наборів даних може бути більш ефективним при управлінні зміщенням.

У найбільш складних задачах алгоритми навчання функціонують у динамічних середовищах, де дані накопичуються з часом. Такі алгоритми [1] мають включати нових даних у процесі навчання. Ряд алгоритмів навчання забезпечують можливості включення нових даних, наприклад KNN та наївні Байєси. Інші алгоритми, наприклад дерева рішень, вимагають суттєвих змін для того, щоб доповнити вхідні дані, зробити поступову індукцію. Можливість додаткового навчання є необхідною властивістю при аналізі потоків даних, але не є недостатньою. Необхідно мати можливості виключення застарілих даних та використовувати більш актуальні набори нових даних [2], [3].

Потоки даних відрізняються тим, що їх у більшості випадків формують комп'ютерні системи. Наприклад, системи моніторингу мереж, веб-майнінгу, управління телекомунікаційними даними, а також фінансові фінансові дані, маркетингові дані, медичні дані, демографічні дані тощо.

Потоки даних можна розглядати як стохастичні процеси, в яких події відбуваються безперервно і незалежно. Робота з моделлю потоку даних не виключає використання звичайних збережених даних, оскільки дані з потоку можуть бути тимчасовими.

Ключовими відмінностями поточкових є такі:

- елементи даних у потоці надходять онлайн;
- інтелектуальна система не контролює порядок надходження даних потоку;
- потоки даних можуть мати необмежений розмір;
- швидкість подання інформації досить велика, так що кожен конкретний елемент повинен бути оброблений в реальному часі, а потім викинутий;
- після обробки даних з потоку даних їх відкидають або зберігають в архіві; оброблені дані неможливо знову отримати, якщо вони явно не збережені в пам'яті;
- розподіл, що генерує елементи, може змінюватися з часом. Таким чином, дані минулого можуть стати неактуальними (або навіть шкідливими) для результатів.

Найважливіші проблеми аналізу потоків даних пов'язані із обмеженими ресурсами на їх обробку.

По-перше, виникають проблеми, пов'язані з обсягом даних. Дані генеруються та збираються постійно, тому їх обсяг стає дуже великим.

Аналіз великих обсягів даних характеризується обмеженими ресурсами часу та пам'яті [4]. Аналіз даних традиційно виконується над статичними наборами даних, де алгоритми видобутку даних можуть дозволити зчитувати вхідні дані кілька разів. Коли джерелом елементів даних є відкритий потік даних, не всі дані можна завантажувати в пам'ять, а офлайн-майнінг із набором даних фіксованого розміру більше не є технічно здійсненним через унікальні особливості поточкових даних. Отже, ми маємо обмеження часу, хоча традиційні методи передбачають, що ми

можемо зберігати всі дані в пам'яті, і немає обмеження за часом. Важливим фактом є те, що дані розвиваються з часом, тому нам потрібні методи, які адаптуються автоматично, без необхідності перезапуску кожен раз, коли виявляється зміна даних.

По-друге, виникають проблеми, спричинені складністю типів даних. На сьогоднішній день дані генеруються та збираються з багатьох джерел: наукові дані, фінансові дані, маркетингові дані, медичні дані, демографічні дані тощо. Вони можуть бути представлені у складному вигляді: деревами XML; послідовностями ДНК; часовою та просторовою інформацією GPS.

Для того, щоб була можливість працювати з такими даними, необхідні відповідні методи. Аналіз даних використовує статистичний пошук інформації, машинне навчання та розпізнавання образів-патернів [5]. Аналіз даних використовує такі інструменти, як класифікація, видобуток правил асоціацій, кластеризація тощо.

Для вирішення наведених проблем з урахуванням властивостей потоків даних було розроблено ряд методів для аналізу потоків даних [6], з низькими вимогами до обсягу пам'яті та часу на елемент даних. Також, виникає необхідність адаптації до змін у часі та швидкого навчання.

Типовий підхід для роботи з такими проблемами базується на використанні так званого «Ковзного вікна»: алгоритм зберігає вікно розміром W , що містить останні елементи даних W , що надійшли (скажімо, на останніх етапах часу W). Коли надходить новий елемент, найстаріший елемент у вікні видаляється, щоб звільнити місце. Короткий зміст потоку даних щоразу обчислюється або перебудовується лише з даних у вікні. Якщо W має невеликий розмір, це по суті опікується вимогою використовувати мало пам'яті.

У більшості випадків величина W вважається зовнішньо визначеною та фіксованою при виконанні алгоритму. Гіпотеза, на якій базується такий підхід, полягає в тому, що користувач може вгадати W , так що розподіл даних можна вважати по суті постійним у більшості інтервалів розміру W .

У більшості випадків користувач не знає заздалегідь, якою буде швидкість змін, яка впливає на вибір розміру вікна, тому W навряд чи буде оптимальним. Також, швидкість змін може сама змінюватися з плином часу, тому оптимальна величина W може змінюватися.

На сьогодні найпоширенішим способом роботи з безперервними потоками даних, що розвиваються в часі, була побудова початкової моделі з ковзним вікном та відновлення моделі періодично або всякий раз, коли її продуктивність (наприклад, помилка класифікації) погіршується на поточному вікні останніх прикладів.

Також при аналізі потоків даних використовуються такі підходи, які підвищують ефективність їх обробки:

- методи попередньої обробки запитів для виявлення запитів, які потребують значного або необмеженого обсягу пам'яті.

- відбір для обробки підмножини даних у випадку, коли швидкість вхідного потоку вище, ніж швидкість обробки запитів;

- реалізація процедур блокування, агрегування, сортування за наявності нескінченних потоків.

1.2 Поняття нейронних мереж

Серед існуючих підходів нейронні мережі є досить популярним і найбільш перспективним засобом обробки сигналів інформації у сфері обчислювального інтелекту [7], [8], [9], [10], [11]. Вони являють собою пристрій, особливість якого полягає в можливості паралельного обчислення одержуваних даних. Ця характеристика є досить важливою, адже саме можливість паралельно обробляти дані, свідчить про широкі можливості нейронних мереж і має величезний потенціал для практичного використання [12]. Наприклад, у випадку, коли кількість зв'язків між нейронами досить велика, обробка даних пришвидшується, завдяки

паралельній обробці. Оскільки при послідовній обробці тих самих даних, витрачаємий на обчислення час різко збільшується.

Термін штучні нейронні мережі з'явився на початку двадцятого сторіччя, але широку відомість отримав лише у 90-х роках через те, що з'явилися більш сучасні й потужніші обчислювальні системи [10], а також були вирішені деякі теоретичні питання. У даному випадку слово «штучні» підкреслює той факт, що ці мережі являють собою штучний пристрій, а не біологічний, хоча варто зауважити, що штучні нейронні мережі були створені на основі принципів роботи біологічного мозку в спробі зрозуміти як саме він працює.

У спрощеному вигляді, нервова система біологічного мозку має вигляд системи [13], де нейронні клітини пов'язані між собою. Кожна з них представляє собою тіло та відгалуження від нього: дендрити і аксони. Кожне тіло нейронної клітини, за допомогою дендритів, отримує сигнали від інших нейронів. При достатній силі отриманих сигналів клітина збуджується та за допомогою аксону передає цей сигнал збудження до іншої, пов'язаних з нею, клітин. Клітини, що отримали збуджуючий сигнал, поширюють цей сигнал далі по всьому мозку.

Проте, незважаючи на історію виникнення штучних нейронних мереж, кожен елемент цієї мережі наділяється певними властивостями для вирішення відповідних задач, а не імітування роботи біологічних процесів у мозку людини.

Штучні нейронні мережі, як і ті, що знаходяться у мозку людини, складаються з множини простих елементів які зветься нейронами. На вхід кожного такого нейрона надходить або один, або декілька вхідних сигналів, від подібних елементів, або сигнали самих даних, що піддаються обробці.

На відміну від біологічних, кожен з штучних нейронів являє собою процес, а саме обчислювальну функцію комбінації вхідних сигналів за певним правилом і правилом активації, що дозволяє розрахувати вихідний

сигнал. Далі отриманий сигнал елементу може передаватися на вхід іншим нейронам по зв'язкам, кожен з яких мають свій ваговий коефіцієнт (вага). В залежності від значення цього коефіцієнта, він впливає на сигнал, що передається для подальшої обробки, підсилюючи його, або навпаки, роблячи його менш впливовим на подальшу обробку.

Головною особливістю штучних нейронних мереж є здатність до навчання. Він характеризується налаштуванням параметрів, або архітектури мережі для вирішення поставленої задачі шляхом оптимізації заданого критерію. Іншими словами з часом мережа покращує якість своєї роботи, налаштовуючи свої характеристики таким чином, щоб максимально наблизитись до оптимального рішення задачі. В залежності від початкових умов [14] (початкові дані, їх об'єм та вигляд, у якому вони представлені, цільова функція), визначається тип навчання та алгоритм за яким дані будуть оброблені.

Навчання нейронної мережі полягає у налаштуванні вагових коефіцієнтів мережі. В загалом навчання ділиться на навчання із вчителем та без вчителя.

У випадку навчання із учителем присутня розмічена навчальна вибірка, тобто та, що містить деякі ознаки та «відповідь» яку повинна отримати система у результаті класифікації. Цей тип навчання полягає у тому, щоб знайти значення ваг, при яких помилка класифікації була б найменшою [17], [18], [19], [20]. Під помилкою класифікації мається на увазі різниця очікуваних значень класифікації та тих, що отримали у результаті роботи системи.

Ідеально розмічені та чисті дані досить нелегко отримати, тому постає задача класифікації без очікуваних відповідей. Підхід, що дає змогу працювати за цих умов називається навчання без учителя. В цьому підході у моделі є набір даних та немає явних вказівок, що з ними робити, тому нейронна мережа намагається самостійно знайти кореляції в даних.

Існує досить багато різних штучних нейронних мереж та з них можна виділити декілька найбільш поширених архітектур:

- персептрони;
- нейронні мережі прямого поширення;
- радіально-базисні нейронні мережі;
- рекурентні нейронні мережі;
- глибокі нейронні мережі;
- самоорганізовані мапи Кохонена [22].

Модель персептрону також відома як одношарова нейронна мережа. Вона містить лише два шари: вхідний та вихідний. У цьому типі нейронної мережі немає прихованих шарів. Він бере вхід і обчислює зважений вхід для кожного вузла. Згодом він використовує функцію активації (переважно сигмовидної функції) для задачі класифікації.

Нейронна мережа прямого поширення – це штучна нейронна мережа, в якій всі вузли не мають циклів та персептрони, що знаходяться у ній згруповані у шари. У цьому типі нейронній мережі [23], кожен персептрон в одному шарі пов'язаний з кожним вузлом у наступному шарі та для того щоб мінімізувати помилку класифікації, зазвичай використовується алгоритм зворотного розповсюдження для оновлення значень ваги.

Радіально-базисні нейронні мережі зазвичай використовуються для задач апроксимації функцій. Вони вирізняються серед інших нейронних завдяки швидшому навчанню та універсальному апроксимації. Основна різниця між радіально-базисними нейронними мережами (RBFN) та мережами зворотного поширення полягає в тому, що RBFN використовують радіально-базисну функцію як активаційну.

Рекурентні нейронні мережі (RNN) – різновидом мереж зворотного поширення. Цей тип мережі представляють інший тип нейронів, які зветься рекурентними – коли кожна прихований нейрон отримує власний вихідний сигнал із фіксованою затримкою (одну або кілька ітерацій). Зазвичай цей тип нейронної мережі використовується у випадках, коли

важливий контекст, тобто коли рішення з минулих етапів або вибірки можуть впливати на рішення наступних ітерацій. Найпоширенішими прикладами такого контексту є текстові дані, де слово в контексті попередніх слів чи речень.

Самоорганізовані мапи Кохонена [16], [21] – це тип штучної нейронної мережі, який навчається за принципом навчання без учителя для створення низьковимірного (як правило, двовимірного), дискретизованого представлення вхідного простору навчальних прикладів, що називається картою. Цей метод застосовується для зменшення розмірності, кластеризації та візуалізації даних. Самоорганізовані мапи, відрізняються від інших нейронних мереж, тому що цей підхід застосовує конкурентне навчання на відміну від навчання з виправленням помилок (наприклад, зворотне розповсюдження з градієнтним спуском), також він використовує функцію сусідства для збереження топологічних властивостей вхідного простору.

Не менш важливою частиною є програмна реалізація обраної, в залежності від поставленої задачі, архітектури мережі. Більша частина штучних нейронних мереж реалізується за допомогою програмних засобів для перевірки теоретичних досліджень, що більшості випадків є достатнім для вирішення поставленої задачі та забезпечує достатньо дешевий та гнучкий підхід. На перший погляд програмна реалізація штучних нейронних мереж виглядає й веде себе як звичайна програма, але на відміну від іншого типу програмного забезпечення, при реалізації штучних нейронних мереж розрахунки виконуються шляхом навчання, а не безпосередньо програмується.

Нейронні мережі використовуються у тому випадку, коли неможливо розробити програму з таким алгоритмом, який би повністю описував всі необхідні розрахунки, або рішення у різних ситуаціях. Наприклад у кожній області діяльності людини завжди існують такі експерти, які можуть пояснити ті чи інші закономірності що впливають на необхідні результати,

проте ці фактори у низці випадків важко формально описати й запрограмувати. У таких випадках використовуються нейронні мережі, адже її рішення є більш гнучким тому що в подальшому система може вдосконалюватися з накопиченням досвіду.

Рішення нейронних мереж становляться кращими та більш ефективними й, з часом, можливості розробки відповідних пристроїв збільшується з покращенням розуміння основоположні принципи тому ця область є досить перспективною для подальшого розвитку у великій кількості різних сфер, а саме розпізнавання природної мови та образів, здійснення прогнозів, виявлення шахрайства (fraud detection), діагностика захворювання людини й багато інших. Таким чином, на сьогоднішній день, штучні нейронні мережі поширюються на все більшу кількість сфер, удосконалюючи потік процесів у них.

1.3 Ймовірнісні нейронні мережі

На сьогоднішній день одним із важливих критеріїв для оцінки методів класифікації, зокрема при роботі в онлайн режимі є швидкодія. Серед існуючих архітектур нейронних мереж, одна за найбільш швидких є ймовірнісна нейронна мережа.

Ймовірнісна нейронна мережа (PNN) використовується у випадку, коли критерій швидкодії є пріоритетним при вирішенні задачі класифікації. Іншими словами мова йдеться про онлайн-класифікацію, що накладає деякі обмеження на час обробки інформації яка надходить на вхід системи.

Завдяки паралельній обробці, ймовірнісна нейронна мережа може змінювати границі областей рішень (decision boundaries) з приходом нових спостережень. На відміну від звичайних нейронних мереж, PNN не використовує зворотне поширення, натомість її навчання засновується на

лінійному навчанні та концепції «нейрони в точках даних», що значно прискорює її роботу.

У 1990 році Дональдом Шпехтом [24] була запропонована ймовірнісна нейронна мережа, що заснована на Байєсівському висовуванні (1.1), та поєднує в собі статистичну теорію із теорією нейронних мереж:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}. \quad (1.1)$$

На основі цієї формули, у ймовірнісній нейронній мережі розраховуються умовні імовірності. Даний підхід полягає у тому, що умовна ймовірність $P(x/y)$ може бути визначена за допомогою апостеріорної ймовірності $P(y/x)$, відомої ймовірності $P(x)$ події x та ймовірності $P(y)$.

Також, подія x може бути представлена як спостереження з навчальної вибірки, а клас, у який може потрапити елемент, що класифікується, може бути представлений як y .

При вирішенні задачі класифікації за допомогою методу Байєса, спочатку формується оцінка щільності розподілу ймовірностей для кожного класу. Вона розраховується на етапі навчання. Виграє клас, у якого ймовірність належності до класифікуемого елемента тестової вибірки найбільша, ніж у інших класів.

Ймовірнісна нейронна мережа класифікує за такими ознаками:

- щільність розподілу в області нового сигналу $x(k)$;
- ціна помилки класифікації;
- апіорна ймовірність.

Наприклад, при бінарній класифікації правило вибору має вигляд:

$$P_A C_A p_A(x) > P_B C_B p_B(x), \quad (1.2)$$

де P_A, P_B – апіорна ймовірність для класів A та B відповідно;

C_A, C_B – ціна помилки класифікації для відповідних класів;

p_A, p_B – значення функції щільності розподілу ймовірностей для відповідних класів.

Значення C_A, C_B залежать від задачі, що вирішується, і тому в більшості випадків ці значення є однаковими для обох класів. Значення функцій p_A, p_B розраховуються методом Парзена.

Навчальна вибірка для ймовірнісної нейронної мережі містить набір n -вимірних векторів $x(1), x(2), \dots, x(N)$ із заданою належністю до класів. Ці вектори розташовані у довільному порядку. Загальна кількість елементів навчальної вибірки N дорівнює сумі кількості елементів N_A та N_B класів A та B відповідно.

Архітектура ймовірнісної нейронної мережі наведена на рисунку 1.1.

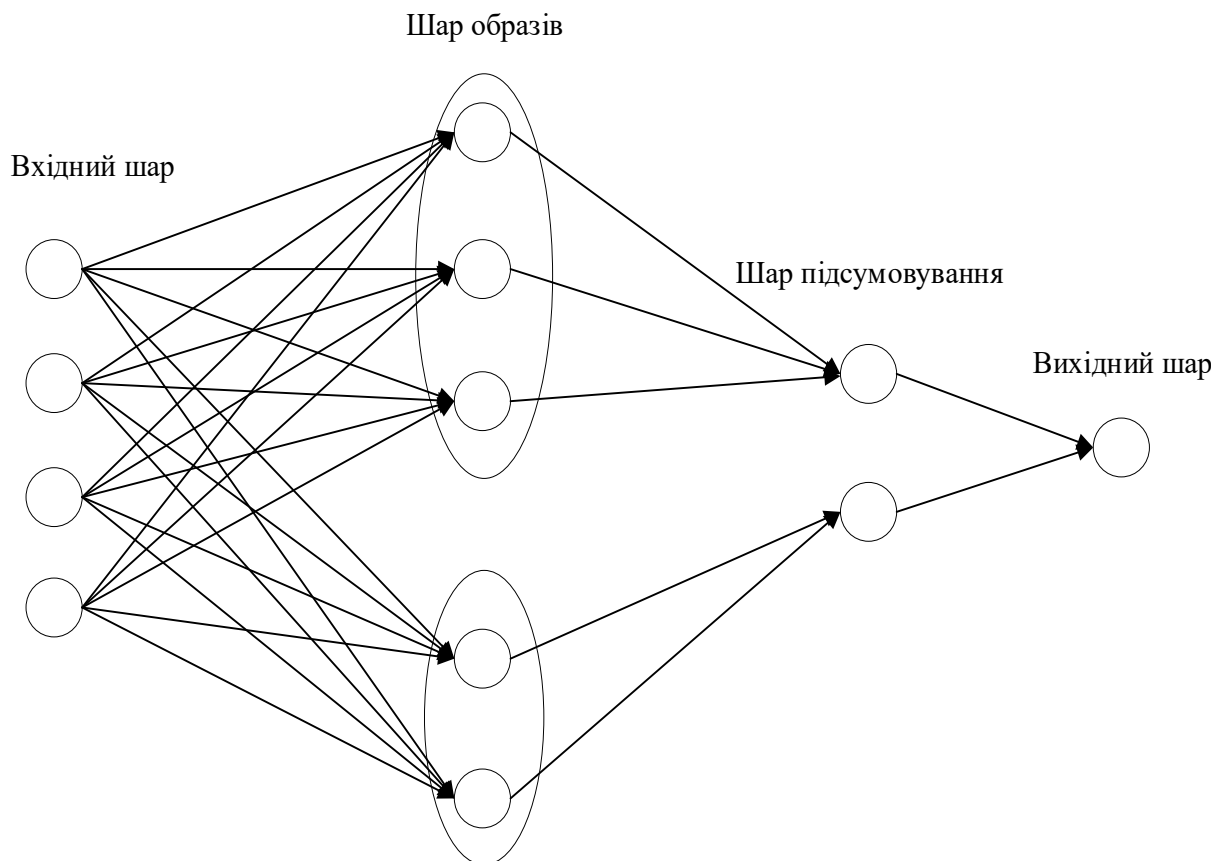


Рисунок 1.1 – Архітектура ймовірнісної нейронної мережі

Архітектура цієї нейронної мережі вперше показує, що Баєсівський класифікатор може бути розбитий на прості процеси, які виконуються паралельно.

Ймовірнісна мережа має чотирьохшарову архітектуру:

- вхідний шар;
- перший прихований шар – шар образів;
- другий прихований – шар підсумовування;
- вихідний шар.

Кількість нейронів вхідного шару фактично дорівнює розміру тренувальної вибірки, що подається в якості вхідного сигналу.

В подальшому вхідні сигнали передаються на кожен із радіальних елементів прихованого шару. Кожен із нейронів шару образів представляється з використанням гаусової функції. Нейрони групуються по класам. Кількість нейронів у даному шарі дорівнює кількості образів. Їх ваги w_{ij} визначаються для компонент цих образів:

$$w_{ij} = x_i(j), i = \overline{1, n}, j = \overline{1, N}. \quad (1.3)$$

Навчання мережі полягає у визначенні ваг w_{ij} . Для нейронів шару образів розраховується зважена сума вхідних значень, а потім перетворюється з використанням ядерної функції:

$$O_j = \exp\left(-\frac{\sum (x - x_i)^2}{\sigma^2}\right) \quad (1.4)$$

де x – спостереження з навчальної вибірки, що є центром активаційної функції;

x_i – елемент вектору вхідного сигналу;

σ – параметр ширини активаційної функції.

Другий прихований шар є шаром підсумовування, оскільки його елементи сформовані як сума виходів шару образів. На виході цього шару отримуємо значення функції щільності розподілу ймовірностей для всіх екземплярів кожного класу:

$$\begin{cases} o_A^{[2]}(k) = \sum_{j=1(A)}^{N_A(A)} o_j^{[1]}(k), \\ o_B^{[2]}(k) = \sum_{j=N_A+1(B)}^{N_B(B)} o_j^{[1]}(k). \end{cases} \quad (1.5)$$

Вихідний шар порівнює виходи шару підсумовування. Результатом буде клас із найбільшим значенням розподілу ймовірностей.

Відмінність навчання PNN полягає у тому, що число елементів ймовірнісної мережі, а також значення їх ваг, визначаються навчальною вибіркою.

Ймовірнісні нейронні мережі мають такі переваги: швидке навчання, що обумовлює можливість її використання для класифікації потоків даних; ймовірнісний характер вихідного значення [25], що створює умови для реалізації нечіткої класифікації.

Недоліки ймовірнісних нейронних мереж пов'язані із швидким зростанням кількості нейронів у шарі образів по мірі збільшення навчальної вибірки. Значне збільшення кількості нейронів може сповільнити роботу мережі. Також слід зазначити, що базова ймовірнісна мережа орієнтована лише на чітку класифікацію. Як правило, у реальних задачах, спостереження, що класифікується може одночасно належати до декількох класів з різним рівнем належності [26]. Іншими словами, сформовані класи можуть перетинатися, утворюючи ситуацію нечіткої класифікації.

Виконаний аналіз властивостей PNN свідчить про необхідність їх удосконалення при вирішенні задач обробки потоків даних з урахуванням обмежень роботи в онлайн режимі.

1.4 Нечіткі нейронні мережі

За останнє десятиліття нечіткі системи стали досить популярним підходом у багатьох наукових додатках та інженерних системах, в системах управління та розпізнавання образів. Також нечітка логіка активно використовується в широкому спектрі споживчих товарів та промислових систем. Нечіткі алгоритми застосовуються в пральних машинах, відеокамерах, автофокусних камерах, кондиціонерах, пирососах, автомобільних трансмісіях, судових навігаторах, поїздах метро, регуляторах горіння. Також вони використовуються в інформаційних технологіях підтримки прийняття рішень, в експертних системах з потужними можливостями висновування з використанням невеликої кількості правил.

Нечіткі множини, запроваджені Лотфі Заде в 1965 р. як математичний спосіб представити невизначеність у лінгвістиці, можна вважати узагальненням класичної теорії множин [27], [28]. У класичному (нечіткому) наборі елемент Всесвіту або належить, або не належить до набору. В даному випадку розглядається чітка належність елемента, тобто це або так (у наборі), або ні (не в наборі). В свою чергу нечіткий набір – це узагальнення звичайного набору, що дозволяє ступінь належності кожного елемента визначити в рамках одиничного інтервалу $[0,1]$. Основна особливість нечіткої логіки полягає в тому, що кожен нечіткий набір має нескінченну кількість функцій належності, на відміну від чітких наборів, де кожен з них має лише одну таку функцію. Така властивість дає змогу

нечітким системам налаштовуватись на максимальну корисність у певній ситуації.

Досить часто нечіткість плутають з ймовірністю. Принципова різниця між ними полягає в тому, що нечіткість має справу з детермінованою правдоподібністю (*deterministic plausibility*), тоді як ймовірність стосується подібності стохастичних недетермінованих подій. Нечіткість – один із аспектів невизначеності. Саме двозначність (невизначеність) виявляється у визначенні поняття або у значенні такого поняття, як «молода людина» чи «велика кімната». Однак невизначеність ймовірності загалом відноситься до поняття випадковості. Тобто, твердження є імовірнісним, якщо воно виражає якусь вірогідність або ступінь визначеності, або якщо воно є результатом чітко визначених, але випадкових подій. Наприклад, висловлювання «Існує 50 на 50 шансів, що він буде там», «Завтра буде дощ», «Киньте кубики і отримайте чотири» мають невизначеність випадковості.

Отже, нечіткість і випадковість відрізняються за своєю суттю, де перший передає «суб'єктивне» людське мислення, почуття або мову, а другий вказує на «об'єктивну» статистику в природничих науках. З точки зору моделювання, нечіткі моделі та статистичні моделі також мають різні види інформації: нечіткі членства представляють подібність об'єктів до неточно визначених властивостей, тоді як ймовірності передають інформацію про відносні частоти [29].

Формалізація нечіткості виконується з використанням нечіткої логіки. Доречно її використовувати у наступних випадках:

– коли процес стосується неперервних явищ (наприклад, одна або декілька змінних управління є безперервними), які нелегко розбити на окремі сегменти;

– коли математична модель процесу не існує, або існує, але занадто складна для кодування, або занадто складна, щоб оцінюватися досить

швидко для роботи в режимі реального часу, або включає занадто багато пам'яті на призначеній архітектурі мікросхеми;

– коли потрібно боротися з високим рівнем навколишнього шуму або важливо використовувати недорогі датчики та / або низькоточні мікроконтролери;

– коли процес передбачає взаємодію людини (наприклад, описове чи інтуїтивне мислення людини);

– коли доступний експерт, який може вказати правила, що лежать в основі поведінки системи, та нечіткі набори, що представляють характеристики кожної змінної.

Методи нечіткої логіки знаходять своє застосування в таких областях, як:

– контроль параметрів різноманітних процесів (найбільш широко застосовувана область);

– розпізнавання образів (наприклад, зображення, аудіо, обробка сигналів);

– кількісний аналіз (наприклад, дослідження операцій, управління);

– вивід (наприклад, експертні системи для діагностики, планування та прогнозування, обробка природної мови; інтелектуальний інтерфейс; розумні роботи; програмна інженерія);

– пошук інформації (наприклад, бази даних).

В свою чергу, штучні нейронні мережі, на сьогоднішній день, є досить популярним методом для обробки інформації для вирішення різного класу задач. Нейронні мережі мають велику кількість взаємопов'язаних вузлів обробки інформації, що працюють паралельно і демонструють здатності навчатися, згадувати та узагальнювати на основі тренувальних моделей або даних.

Нечіткі системи та нейронні мережі застосовуються в різних областях, наприклад споживчих товарів, контроль промислових процесів, медичні прилади, інформаційні системи та аналіз рішень. Нечітка логіка

базується на способі роботи мозку з неточною інформацією, тоді як нейронні мережі моделюються по аналогії з фізичною архітектурою мозку.

Незважаючи на те, що ці два поняття брали початок від різних ідей, існує ряд паралелей, які вказують на їх подібність. Обидві системи мають здатність працювати у невизначеному, неточному середовищі та спроможні моделювати складні нелінійні процеси з довільним ступенем точності.

Незважаючи на спільні риси, між ними також є суттєві відмінності. Нечіткі системи – це структуровані числові оцінювачі, які починають із формалізованих уявлень про структуру предметної області, а потім формулюють нечіткі правила «Якщо-То» (IF-THEN) як своєрідні знання експертів. У результаті ці правила поєднуються у нечіткі набори для створення загальної складної нелінійної поведінки. розбиття множин даних на окремі області. Це зазвичай лімітує кількість змінних, що подаються на вхід нечітким системам [30], [31], [32]. Нечіткі системи дозволяють легко пояснити отримані з їх допомогою результати.

Однак ці системи не можуть навчатись і у подальшому використовувати отримані знання у механізмі висновування. Для користувача нечітка система представляється «білою скринєю», однак вона має обмеження при вирішенні складних задач, що характеризуються значною кількістю змінних.

Нейронні мережі це динамічні системи, які здатні навчатись, тим самим здобуваючи толерантність до шуму та можливість узагальнення, яка утворюється із взаємопов'язаних структур (нейронів), динамічного та розподіленого представлення даних. Елементи нейронної мережі мають здатність навчатись та узагальнювати за допомогою навчальних моделей або даних. Їх перевагою також є те, що вони можуть автоматично здобувати знання, навчаючись. Проте цей процес, як правило, потребує досить багато часу. Однак вони не дають пояснення, як здійснюється класифікація. Також досить складно використати яку-небудь попередньо

визначену інформацію для оптимізації процесу навчання нейронної мережі. Ще один недолік традиційних нейро-систем полягає в тому, що вони не дозволяють класифікувати об'єкти, які із різним ступенем достовірності належать до різних класів. В цілому нейронна мережа є «чорною скринею» для користувача, тому її аналіз є досить складним.

З огляду на подібність та розбіжність нейронних мереж та нечітких систем, можна визначити, що вони вирішують багато однакових проблем та недоліки одного математичного апарату компенсуються перевагами іншого, і навпаки. Їх розбіжності призвели до об'єднання їх у функціональну систему для подолання їхніх індивідуальних недоліків. Також це поєднання призводить до нової системи, яка має переваги обох цих систем: здібність до навчання нейронних мереж, а нечіткі системи забезпечують нейромережі структурною структурою з нечіткими високими рівнями IF-THEN правила мислення та міркувань [33].

У цілому, нечітка логіка та нейронні мережі є складовими дослідницької області, яка називається м'якими обчисленнями (soft computing), термін створений Лотфі Заде. Основною метою м'яких обчислень є використання такої толерантності для досягнення контрольованості, надійності, високого рівня машинного інтелекту та низької вартості в практичному застосуванні.

Ще однією основною складовою м'яких обчислень є імовірнісне міркування. Воно включає в себе генетичні алгоритми, еволюційне програмування, мережі переконань, хаотичні системи та частини теорії навчання. Серед приведеного переліку, генетичні алгоритми та еволюційне програмування схожі на нейронні мережі. Вони базуються на низькорівневих мікроскопічних біологічних моделях, та розвиваються в напрямку пошуку кращих рішень проблем, подібно до того, як види еволюціонують до кращої адаптації до свого середовища.

Нечітка логіка, нейронні мережі та ймовірнісне міркування відповідають за різні властивості: неточністю та наближеними

міркуваннями нечіткої логіки, навчання нейронних мереж та невизначеність ймовірнісних міркувань. Оскільки нечітка логіка, нейронні мережі та ймовірнісні міркування доповнюють, а не конкурують, часто вигідно використовувати їх у комбінації.

Для поєднання їх переваг та подолання недоліків було запропоновано методи їх інтеграції та синтезу. Це поєднання двох різних технологій може бути реалізоване у трьох напрямках, в результаті чого утворюються моделі з різними характеристиками [34]:

- нечіткі нейро-системи: використання нейронних мереж як інструментів у нечітких системах;
- нечіткі нейронні мережі: нечіткість звичайних моделей NN;
- нечіткі нейронні гібридні системи: включення технології нечіткої логіки NN в гібридних системах.

Перші дві системи представляють комбінації, де одна технологія допомагає іншій, а третя система демонструє спільну комбінацію [35], де дві технології тісно інтегровані для виконання спільного завдання.

Нейро-нечіткі, або нейро-фаззі системи базуються на застосуванні принципів нечіткої логіки для оптимізації процесу налаштування відносно часу або інших обраних параметрів. В даній ситуації нечітка логіка використовується як засіб NN, що представляє собою «чорну скриню» і тому не може бути визначена у вигляді множини нечітких правил.

Нейро- фаззі системи також класифікуються за тим, як вони відображаються у архітектурі NN. Виділяють такі підходи:

- вибіркоче визначення функцій належності;
- зберігання функцій належності в нейронах;
- зберігання функцій належності у зв'язках між нейронами.

Мета нейро-нечітких, або нейро-фаззі NN забезпечити нечіткі системи з можливостями автоматичного налаштування, тобто, навчання

або налаштуванні функцій членства та нечітких правил нечітких систем.

Вони використовують багат шарову архітектуру:

- на вхід першого шару надходять вхідні змінні;
- у прихованих шарах визначаються нечіткі правила;
- останній, вихідний шар – вихідні дані, де визначити різні рівні належності до окремих класів;
- ваги – нечітка множина вхідних і вихідних змінних;
- процедура навчання може враховувати змістові властивості систем що розглядаються. Це дає можливість обмежити можливі модифікації налаштованих параметрів;
- нейро-нечітка система апроксимує невідому $m \times n$ -вимірну функцію, яка частково опиється навчальними.

Згідно першого підходу нейро-нечіткі системи базуються на нечітких множинах. Вони мають такі особливості:

- ступені належності описані лише для деяких значень і функція належності представлена як вектор;
- кожному ступеню належності відповідає лише один вхідний або вихідний нейрон.

На сьогодні існує наступні варіанти імплементації нейро-нечітких систем згідно даного підходу:

- апроксимація виходів відповідним змінним, що надходять на вхід, тобто використовується модель «чорної скрині».
- втілення відповідних нечітких правил, у архітектуру створеної системи.

Підхід до зберігання параметризованих функцій належності у нейронах реалізовано у адаптивній мережі на основі нечіткого висновування Такагі-Сугено. Вивід у такій системі відповідає виводу на if-then правилах, що носять нечіткий характер.

Такий підхід дозволить суттєво скоротити спожитий час на розробку, одночасно покращуючи продуктивність нечітких систем. Також варто

зазначити, що у багатьох випадках, нейро-нечіткі системи, швидші, ніж звичайні NN. Нечіткі NN зберігають основні властивості та функції NN, наприклад, корекція функцій належності використовуючи методи навчання NN, наприклад, одного з найпопулярніших методів – методу зворотного поширення помилки [36].

Деякі елементи такої нейронної мережі нечіткі, наприклад, функції висновування можуть бути реалізовані засновуючись на апараті нечіткої логіки. Такі системи мають наступні переваги:

- використання апріорної інформації;
- здобування нові знань;
- є логічно прозорими.

У цьому підході знання про предметну область стають формалізованими з точки зору нечітких наборів, які пізніше застосовуються для покращення навчання мережі та розширення її можливостей інтерпретації. Наприклад, нейронна мережа може бути нечіткою таким чином, що вона формує відображення між нечіткими наборами вводу-виводу. Крім того, нечітка логіка може бути використана для визначення кроку навчання нейро мереж.

Використовуючи нечітку логіку у нейронних мережах, досягається деякий ступінь гнучкості підходу, який впливає на якість роботи системи і вона стає більш надійною.

У нечітко-нейронній гібридній системі як методи нечіткої логіки, так і нейронні мережі використовуються окремо для створення двох незалежних підсистем, які виконують власні завдання, виконуючи різні функції в комбінованій системі. Архітектура нечітко-нейронних гібридних систем зазвичай орієнтована на практичне застосування. Використовуючи свої індивідуальні сильні сторони, нечітка логіка та підсистеми нейронної мережі доповнюють одна одну ефективно та ефективно для досягнення спільної мети.

1.5 Постановка задачі дослідження

На сьогодні системи і методи обчислювального інтелекту використовуються для ефективного вирішення широкого кола задач в рамках інтелектуального аналізу даних (Data Mining) таких, як прогнозування, класифікація, кластеризація, вирішення оптимізаційних задач і т.п.

Однією з актуальних задач обчислювального інтелекту, що широко застосовується на практиці, є задача аналіз потоків даних. Її вирішення зазвичай потребує значних обчислювальних витрат, а також такої системи, яка б змогла швидко обробляти поступаючі дані і оперативно надавати відповідь. Відповідно, вхідна послідовність даних може мати значну, і навіть необмежену довжину, що потребує удосконалення швидкодіючих підходів до виконання такої класифікації.

В той же час, серед нейронних мереж, ймовірнісні нейронні мережі мають ряд характеристик, що забезпечують збільшення швидкості класифікації: не вимагають окремої фази навчання; додаткові навчальні приклади можуть бути легко включені в PNN, при їх появі; забезпечують надійність класифікації в областях із зашумленими даними.

Одна з головних особливостей послідовності даних пов'язана із тим, що вони можуть одночасно належати до декількох класів із різним ступенем достовірності [37], [38], [39], [40], що потребує врахувати додаткову властивість нечіткості при виконанні їх класифікації.

Також варто зазначити, що використання нечіткості також може прискорити та вдосконалити роботу системи.

Треба зазначити, що як нейронна мережа, PNN може бути інтегрована із нечіткими залежностями, що дає можливість врахувати належність образів-зображень до декількох класів.

У ситуаціях, якщо обсяг вибірки даних, що підлягають опрацюванню, є необмеженим, а самі ці дані надходять у систему у послідовному online-режимі, необхідно контролювати «вікно», яке захоплює дані для обробки. У цьому випадку необхідно використати типовий метод роботи з потоками даних – метод ковзного вікна.

Зазначене свідчить про актуальність розробки швидкодіючого методу online-класифікації потоків даних з урахуванням різних рівнів належності вхідних даних до декількох класів на основі використання ймовірнісних нейронних мереж.

Мета магістерської атестаційної роботи полягає у створенні ймовірнісної нейро-фаззі системи для підвищення ефективності класифікації послідовності даних, що надходять у онлайн-режимі та мають різні типи вхідних даних.

Для досягнення поставленої мети в роботі вирішуються такі задачі:

- виконання порівняльного аналізу можливостей ймовірнісних та нечітких нейронних мереж до класифікації потоків даних з обмеженнями режиму онлайн;
- розробка архітектури ймовірнісної нейро-фаззі системи;
- експериментальна перевірка розробленої нейро-фаззі системи шляхом розпізнавання послідовностей даних, які надходять у різних формах.

2 ЙМОВІРНІСНА НЕЙРО-ФАЗЗИ СИСТЕМА ТА ЇЇ ГІБРИДНЕ НАВЧАННЯ В ЗАДАЧАХ АНАЛІЗУ ПОТОКІВ ДАННИХ **Equation Section 2**

2.1 Еволюційна нечітка ймовірнісна нейронна мережа та її онлайн-навчання

Ймовірнісні нейронні мережі [24], [25] призначені для вирішення задач класифікації та є «найближчими родичами» радіально-базисних (RBFN) та узагальнених регресійних нейронних мереж (GRNN). Ці архітектури використовують байєсівське висновування, вікна Парзена та оцінки Надарая-Ватсона з ядерними активаційними функціями. В якості таких функцій в більшості випадків використовують гаусіани. Проте, ці конструкції можуть біти замінені на будь-які радіально-базисні функції, що мають дзвонувану форму.

Хоча за точністю класифікації, ймовірнісні нейронні мережі поступаються більш просунутим, глибинним нейронним мережам (DNN) [23], [33], [34], [35], у них є досить важлива перевага у швидкості у порівнянні з DNN. Ймовірнісні нейронні мережі відносяться до класу «just-in-time-models» та мають дуже велику швидкість навчання завдяки лінійному навчанню, базуючись на концепції нейрони у точках даних («neurons at data points»). Така характеристика багатьох випадках дає суттєву перевагу у порівнянні з іншими підходами до опрацювання інформації. Суттєвою ця перевага є при вирішенні задачі класифікації у режимі реального часу.

Слід відзначити, що відмінною рисою як глибоких нейронних мереж, а також традиційних мілінних нейронних мерж є те, що вони вирішують задачу чіткої класифікації. Іншими словами, вони розподілять спостереження по класам, які не перетинаються у просторі ознак. У

випадку, коли ця умова не виконується, тоді постає задача нечіткої класифікації, яка потребує інших підходів до її вирішення.

Задача нечіткої класифікації може бути вирішена за допомогою нейро-фаззі систем (NFS) [26], [28]. Ці системи, по-перше, визначають, чи належить те, чи інше спостереження до класу. По-друге, вони оцінюють рівень нечіткої належності кожного спостереження до кожного з можливих класів. Даний підхід, незважаючи на його переваги, має головний недолік: обчислення даної системи досить громіздкі, що сповільнює її роботу. Тому виникають серйозні обмеження у їх використанні при рішенні задачі нечіткої класифікації у онлайн режимі опрацюванні інформації [39], [40], [41].

Модифікована EFPNN забезпечує високу швидкість вирішення задач розпізнавання образів у випадках, коли класи даних довільним чином перетинаються [42], [43]. Модифікована нечітка ймовірнісна нейронна мережа має чотиришарову архітектуру.

Вхідною інформацією для системи є навчальна вибірка образів, що утворена пакетом n -вимірних векторів

$$X = \{x(1), x(2), \dots, x(k), \dots, x(N)\}, x(k) = (x_1(k), \dots, x_i(k), \dots, x_n(k))^T \in R^N,$$

при цьому N_1 спостережень відносяться до першого класу Cl_1 , N_2 – до другого класу Cl_2 , ..., N_m – до Cl_m , тобто:

$$\underbrace{k=1, 2, \dots, N_1}_{Cl_1 \text{ містить } N_1 \text{ спостережень}}, \underbrace{N_1+1, \dots, N_1+N_2, \dots, N_1+N_2+N_3, \dots, N_m}_{Cl_2 \text{ містить } N_2 \text{ спостережень}, \dots, Cl_m \text{ містить } N_m \text{ спостережень}}.$$

Кількість нейронів у першому прихованому шарі – шарі образів дорівнює N . Активаційна функція кожного нейрона позначається як $\varphi_{\tau_j}(x, c_{\tau_j})$, де $\tau_j = N_1 + N_2 + \dots + N_{j-1} + 1, \dots, N_1 + N_2 + \dots + N_j, j = 1, 2, \dots, m$ – номер класу, $c_{\tau_j} = (c_{\tau_j,1}, \dots, c_{\tau_j,i}, \dots, c_{\tau_j,n})^T$ центр дзвонуватої активаційної функції. Налаштування мережі відбувається лише на рівні шару образів за

допомогою лінивого навчання, а саме концепції «Нейрони в точках даних» так, що $c_{\tau_j} = x(k) = x(\tau_j)$, якщо $x(k) \in Cl_j$.

При надходженні на вхід системи першого прихованого шару вектора x з невідомою класифікацією на виходах нейронів першого шару з'являються сигнали

$$o_j^{[1]}(x) = \varphi_{\tau_j}^j(x, c_{\tau_j}), \quad (2.1)$$

які у випадку застосування гаусіанів можна записати у вигляді

$$\varphi_{\tau_j}^j(x, c_{\tau_j}) = \exp\left(-\frac{\|x - c_{\tau_j}\|^2}{2\sigma^2}\right), \quad (2.2)$$

тут σ^2 – параметр ширини активаційної функції, а якщо вихідні дані попередньо відцентровані та пронормовані так, що

$$\|x(k)\| = \|c_{\tau_j}\| = \|x\| = 1, \quad (2.3)$$

то

$$o_j^{[1]}(x) = \exp(\sigma^{-2}(x^T c_{\tau_j} - 1)). \quad (2.4)$$

Другий прихований шар містить m елементарних підсумовувачів, на виходах яких формуються сигнали

$$o_j^{[2]}(x) = \sum_{\tau_j=N_1+N_2+\dots+N_{j-1}+1}^{N_1+N_2+\dots+N_j} o_j^{[1]}(x) = p_j(x), \quad (2.5)$$

що є парзенівськими оцінками невідомих щільностей розподілу ймовірностей $p_j(x)$.

Третій прихований шар враховує апіорні ймовірності $P_j = N_j N^{-1}$ та ціну помилок класифікації S_j так, що

$$o_j^{[3]}(x) = o_j^{[2]}(x) N_j N^{-1} S_j. \quad (2.6)$$

У вихідному шарі визначається клас-«переможець»

$$j^* = \arg \max_j o_j^{[3]}(x), \quad (2.7)$$

а також нечіткий рівень належності образу x до класу-переможця j^*

$$\mu_{j^*}(x) = \frac{\left\| N_{j^*}^{-1} \sum_{\tau_{j^*}} c_{\tau_{j^*}} - x \right\|^{-2}}{\sum_{j=1}^m \left\| N_j^{-1} \sum_{\tau_j} c_{\tau_j} - x \right\|^{-2}}. \quad (2.8)$$

Головною перевагою цієї мережі є простота її синтезу та навчання, а основним недоліком – суттєве збільшення кількості нейронів у першому прихованому шарі із зростанням кількості спостережень N у навчальній вибірці.

Обмежити кількість нейронів у першому прихованому шарі можна, скориставшись ідеями теоріями еволюційних систем обчислювального

інтелекту. Припустимо, що загальна кількість нейронів у цьому шарі не може перевищувати деякого числа $M < N$, при цьому на кожен клас Cl_j припадає $m^{-1}M$ нейронів. Розглянемо процес формування нейронів – еволюції шару для будь-якого класу $j=1, 2, \dots, m$, при цьому для спрощення позначень конкретного нейрону будемо позначати індексом $\tau_j = 1, 2, m^{-1}M$. Процес формування шару запишемо у вигляді такої послідовності кроків.

Крок 0. Задається максимальна кількість нейронів у шарі M та поріг нерозрізненості Δ між двома сусідніми центрами активаційних функцій c_{τ_j} та $c_{\tau_{j+1}}$.

Крок 1. Коли спостереження $x(1)$, що належить класу j , надходить у мережу, формується перша активаційна функція класу y_1^j з центром $c_1^j = x(1)$ та параметром ширини σ^2 , який обирається досить довільним чином. При цьому, якщо вихідні дані відцентровані та про нормовані так, що $-1 \leq x_i(k) \leq 1$, досить зручно прийняти значення $\sigma^2 \approx 0,33$.

Крок 2. Коли друге спостереження $x(2)$ з того же класу надходять у мережу, перевіряється виконання умови:

$$\|x(2) - c_2^j\| \leq \Delta. \quad (2.9)$$

Якщо умова, що $x(2)$ не формує нового центру задовольняється, то далі перевіряється умова

$$0 < \|x(2) - c_2^j\| \leq 2\Delta. \quad (2.10)$$

Якщо ця умова виконується, то відбувається корегування центру c_1^j згідно із правилом самонавчання Кохонена [8] “Winner takes all”:

$$c_1^j(2) = c_1^j(1) + \eta(2)(x(2) - c_1^j(1)). \quad (2.11)$$

Тобто сформований центр $c^j(1)$ “підтягується” до нового спостереження $x(2)$, де $0 < \eta(2) < 1$ – параметр кроку.

Якщо виконується нерівність:

$$2\Delta < \|x(2) - c_1^j\|, \quad (2.12)$$

то формується нова активаційна функція φ_2^j з центром $c_2^j = x(2)$.

Крок h : коли спостереження $x(h)$ ($h < m^{-1}M < N$) надходить у мережу, спочатку відшукується нейрон-переможець найближчий у сенсі прийнятої метрики до цього спостереження. Далі перевіряються умови

$$\|x(h) - c_{h-1}^{j*}\| \leq \Delta, \quad (2.13)$$

$$0 < \|x(h) - c_{h-1}^{j*}\| \leq 2\Delta, \quad (2.14)$$

$$2\Delta < \|x(h) - c_{h-1}^{j*}\|. \quad (2.15)$$

Після цього або це спостереження ігнорується, або центр c_{h-1}^{j*} підтягується до спостереження $x(h)$, або формується нова активаційна функція з центром $c_h^j = x(h)$.

Процес побудови виконується до того часу, поки у кожному шарі буде сформовано по $m^{-1}M$ нейронів. Після цього цей шар вважається сформованим, а нові спостереження, що надходять у мережу, можуть лише змінювати розташування центрів згідно з правилом самонавчання Кохонена.

Таким чином, еволюційна нечітка ймовірнісна нейронна мережа реалізує принципи навчання за Д. Шпехтом, еволюційність систем П. Ангелов - Н. Касабов [44], [45], та самонавчання Кохонена [15], [22]. Ця мережа призначена для класифікації спостережень, що надходять на обробку у послідовному онлайн режимі у формі вектора. Важливою перевагою цієї мережі є можливість роботи за умов перетинних класів довільної форми. Також вона не потерпає від «прокльону розмірності». Завдяки обмеженій кількості нейронів, ця мережа має можливість швидко навчатися, що дає їй перевагу при обробці даних в онлайн режимі, вигідно відрізняючи її від глибинних та традиційних мілтинних багат шарових мереж, а також дозволяє обробляти велику кількість даних в рамках загальної проблеми Data Mining.

2.2 Адаптивна ймовірнісна нейронна мережа з нечітким висновуванням та її онлайн навчання

В основі ймовірнісних нейронних мереж лежать Байєсівська оцінка та вікна Парзена, та в якості активаційних функцій використовуються радіально-базисні, з двозонуватою формою функції. Хоча вони поступаються за точністю більш складним системам, проте простота їх чисельної реалізації і висока швидкодія в ряді випадків забезпечують їм перевагу. Це особливо важливо в задачах, коли інформація на обробку подається послідовно в онлайн режимі, а обсяги цієї інформації можуть бути досить великі.

Також важливо зазначити, що ймовірнісні нейронні мережі чіткі системи. Априорі передбачається, що класи спостережень, що формуються, взаємно не перетинаються, хоча в загальному випадку це не так. Саме тому у [46], [47], [48], [49], [50] була запропонована ймовірнісна нейронна мережа з нечітким висновком, де замість визначення належності

спостереження до одного класу, визначається його рівень належності до кожного з класів.

Однією з основних особливостей цих мереж є їх висока швидкодія, що досягається завдяки способу їх навчання за принципом «Neurons at data points», коли центр зформованої дзвонуватої активаційної функції визначається векторними спостереженнями з навчальної вибірки. Звичайно, при великому обсязі навчальної вибірки мережа може стати занадто громіздкою з точки зору кількості сформованих активаційних функцій R-нейронів. Однак це проблема досить просто вирішується за допомогою ковзного вікна, коли ці активаційні функції формуються на основі невеликого числа спостережень з навчальної вибірки, що останніми прийшли в синтезовану мережу.

Складнішою є проблема вибору параметрів ширини формованих активаційних функцій. Якщо активаційні функції занадто вузькі, то, можливо, виникли «дірки» в просторі входів. При цьому можлива поява ситуації, коли навчальна мережа не буде реагувати на деякі спостереження, що надходять. При занадто широких дзвонуватих функціях, спостереження, що класифікуються, «розмазуються» по різних класах з практично однаковими рівнями належності. У зв'язку з цим в [51] була запропонована стратегія самоадаптації вибору кращого параметру ширини, що представляє собою набір емпіричних правил аналізу даних у пакетному режимі. І хоча цей підхід забезпечив поліпшення якості класифікації, швидкодія процесу навчання природно знизилася.

У зв'язку з цим, доцільною є розробка адаптивної імовірнісної нейро-нечіткої системи, що дозволяє налаштовувати параметри ширини активаційних функцій в онлайн режимі під час формування системи, коли на вхід надходить навчальна вибірка.

Імовірнісна нейронна мережа з нечітким висовуванням має трьохшарову архітектуру, при цьому всі шари виконують різні функції.

На вхідний (нульовий) шар мережі послідовно поступають дані навчальної вибірки $X = \{x(1), x(2), \dots, x(k), \dots, x(N)\}$, $x(k) = (x_1(k), \dots, x_i(k), \dots, x_n(k))^T \in R^N$, при цьому кількість цих спостережень, що визначають значення N , може змінюватися з часом. Передбачається, що також в цьому наборі N_1 спостережень належить першому класу Cl_1 , N_2 спостережень належить другому класу Cl_2 , N_j спостережень належить j -му класу Cl_j й, нарешті N_m спостереження належить класу Cl_m . Іншими словами, у вхідній вибірці дані належать m класів, при цьому m також може змінюватися в процесі навчання – формування мережі. Також навчальна вибірка може бути представлена у формі пакета спостережень, проте порядок надходження даних з різних класів значення не має.

Таким чином, перший прихований шар (шар патернів) містить N , так званих, R -нейронів, що реалізують нелінійні дзвонуваті активаційні функції

$$\varphi_{\tau_j}(x, c_{\tau_j}, \sigma_{\tau_j}), \quad (2.16)$$

де τ_j – індекс для j -го класу, що набуває значення $\tau_j = N_1 + N_2 + \dots + N_{j-1} + 1, \dots, N_1 + N_2 + \dots + N_j$, $j = 1, 2, \dots, m$;

c_{τ_j} – векторний центр-вектор дзвонуватої функції активації

$$c_{\tau_j} = (c_{\tau_j 1}, \dots, c_{\tau_j i}, \dots, c_{\tau_j n})^T;$$

σ_{τ_j} – скалярний параметр ширини відповідної активаційної функції, який в подальшому буде налаштовуватися,

Найбільш часто в якості дзвонуватих функцій використовується Гауссіан, так що вихідні сигнали нейронів прихованого шару j -го класу можуть бути представлені у вигляді

$$o_j^{[1]}(x) = \varphi_{\tau_j}^j(x, c_{\tau_j}, \sigma_{\tau_j}) = \exp\left(-\frac{\|x - c_{\tau_j}\|^2}{2\sigma_{\tau_j}^2}\right). \quad (2.17)$$

Якщо вихідні дані попередньо перетворені так, що

$$\|x\| = \|c_{\tau_j}\| = 1, \quad (2.18)$$

нормування на гіперсферу, то вираз (2.17) може бути перетворено до форми

$$o_j^{[1]}(x) = \exp\left(\sigma^{-2}(x^T c_{\tau_j} - 1)\right), \quad (2.19)$$

де скалярний добуток $x^T c_{\tau_j}$ – косинус кута між центром відповідної функції і спостереженням, що підлягають класифікації.

Сигнали $o_j^{[1]}$ можуть змінюватися в діапазоні

$$0 < \exp(-2\sigma_{\tau_j}^{-2}) < o_j^{[1]}(x) \leq 1. \quad (2.20)$$

Діаметр кожного з сформованих класів описується співвідношенням

$$0 \leq D_j = \max\|c_{\tau_j} - c_{\tau_j'}\| \leq 2, \quad (2.21)$$

де τ_j' й τ_j приймають значення з інтервалу $N_1 + N_2 + \dots + N_{j-1} + 1, \dots, N_1 + N_2 + \dots + N_j$, $j = 1, 2, \dots, m$.

Другий прихований шар утворений $m + 1$ елементами $N_1 + N_2 + \dots + N_j$ суматорами, а вихідні сигнали перших m суматорів цього шару мають вигляд:

$$o_j^{[2]}(x) = \sum_{\tau_j=N_1+N_2+\dots+N_{j-1}+1}^{N_1+N_2+\dots+N_j} o_j^{[1]}(x) = p_j(x), \quad (2.22)$$

де $p_j(x)$ – парзенівська оцінка щільності розподілу даних в j -му класі. Останній $m + 1$ -й суматор цього шару об'єднує всі сигнали першого прихованого шару так, що

$$o_j^{[2]}(x) = \sum_{j=1}^m o_j^{[2]}(x). \quad (2.23)$$

Третій вихідний шар мережі складається з двох незалежних блоків оцінки ймовірності того, що спостереження x належить j -му класу, і оцінки рівня нечіткої приналежності спостереження x до j -го класу. Перший з цих блоків утворений m елементарними дільниками, що реалізують операцію ділення, а на його виході з'являються векторні сигнали

$$\hat{y}(x) = (\hat{y}_1(x), \dots, \hat{y}_j(x), \dots, \hat{y}_m(x))^T, \quad (2.24)$$

де

$$0 \leq \hat{y}_j(x) = \frac{o_j^{[2]}(x)}{o^{[2]}(x)} \leq 1, \quad \sum_{j=1}^m \hat{y}_j(x) = 1 \quad (2.25)$$

описують ймовірність того, що класифікується спостереження належить j -го класу.

У другому блоці вихідного шару оцінюється рівень належності за Дж. Бездеком [52]:

$$\mu_j(x) = \frac{\left\| N_j^{-1} \sum_{\tau_j} c_{\tau_j} - x \right\|^{-2}}{\sum_{l=1}^m \left\| N_l^{-1} \sum_{\tau_l} c_{\tau_l} - x \right\|^{-2}} = \frac{\|c_j - x\|^{-2}}{\sum_{l=1}^m \|c_l - x\|^{-2}}, \quad (2.26)$$

де

$$c_j = \frac{1}{N_j} \sum_{\tau_j} c_{\tau_j} = \frac{1}{N_j} \sum_{x(k) \in Cl_j} x(k) \quad (2.27)$$

є центроїдом-прототипом j -го класу, а τ_j номер спостереження з навчальної вибірки, що відноситься до j -го класу.

Навчання PNN реалізується за принципом «Neurons at data points» [42] (ліниве навчання), що означає, у тому випадку, якщо спостереження $x(k)$ з навчальної вибірки належить класу Cl_j , то формується центр активаційної функції $\varphi_{\tau_j}^j(x, c_{\tau_j}, \sigma_{\tau_j})$, що також можна записати у наступному вигляді:

$$c_{\tau_j} = x(k) \text{ if } x(k) \in Cl_j. \quad (2.28)$$

Зрозуміло, що якщо навчальна вибірка досить велика за обсягом, нейронна мережа що стає занадто громіздкою. Подолати цю проблему можна, використовуючи принцип ковзного вікна. Припустимо, що на основі навчальної вибірки сформована мережа, що складається з N нейронів, причому на j -й клас припадає N_j нейронів. Передбачається

також, що далі мережа не повинна збільшувати кількість нейронів першого прихованого шару. Припустимо також, що в навчальну вибірку надійшло спостереження, що відноситься до j -го класу. Тоді слідуючи стандартному підходу, індекс τ_j перебирає значення, тобто

$$\begin{aligned} \tau_j = N_1 + N_2 + \dots + N_{j-1} + 1, \dots, N_1 + N_2 + \dots \\ + N_j, N_1 + N_2 + \dots + N_j + 1, \end{aligned} \quad (2.29)$$

повинен бути доданий додатковий нейрон. Використовуючи ідею «ковзного вікна», нейрон виключається з $\tau_j = N_1 + N_2 + \dots + N_{j-1} + 1$ й замість цього формується нейрон з $\tau_j = N_1 + N_2 + \dots + N_j + 1$. Це означає, що в j -му класі залишається N_j нейронів з $\tau_j = N_1 + N_2 + \dots + N_{j-1} + 2, \dots, N_1 + N_2 + \dots + N_j + 1$; при надходженні спостереження $x(N+2) \in Cl_j$ виключається нейрон з $\tau_j = N_1 + N_2 + \dots + N_{j-1} + 2$ й додається нейрон з $\tau_j = N_1 + N_2 + \dots + N_j + 2$.

Задаючи максимально можливе число нейронів у j -му класі N_j^* , можна сформувати процес лінивого навчання ймовірнісної нейронної мережі, в задачах коли навчальний набір даних надходить в online режимі, а її розмір попередньо не визначений.

Також можна реалізувати рекуррентне уточнення центроїдів класів (2.27) для розрахунку рівнів належності (2.26). Переписавши вираз (2.27) у наступній формі:

$$c_j = \frac{1}{N_j} \sum_{\substack{\tau_j = N_1 + N_2 + \dots + N_{j-1} + 1 \\ x(k) \in Cl_j}}^{N_1 + N_2 + \dots + N_j} x(k), \quad (2.30)$$

й додаючи спостереження $x(N_j + 1)$, виходить:

$$c_j = \frac{1}{N_{j+1}} \sum_{\substack{\tau_j=N_1+N_2+\dots+N_{j-1}+1 \\ x(k) \in Cl_j}}^{N_1+N_2+\dots+N_{j+1}} x(k), \quad (2.31)$$

або у рекурентній онлайн формі:

$$c_j(N_{j+1}) = c_j(N_j) + \frac{1}{N_{j+1}} (x(N+1) - c_j(N_j)), \quad (2.32)$$

для задач, де в першому прихованому шарі кількість нейронів не обмежена.

Якщо кількість нейронів в j -му класі обмежена числом N_j^* , то «віконна» форма процедури може бути записана у вигляді

$$c_j(N_{j+1}) = c_j(N_j) + \frac{1}{N_j^*} (x(N+1) - x(N_1 + N_2 + \dots + N_{j-1} + 1)). \quad (2.33)$$

Процес налаштування параметрів ширини активаційних функцій σ_{τ_j} можна реалізувати на основі навчання з учителем(контрольоване навчання) заданої цільової функції. В [50] був запропонований алгоритм навчання квадратичної активаційної функції на основі квадратичного критерію. Оскільки в PNN найбільш часто використовуються гаусіани, доцільно ввести метод налаштування параметра ширини – дисперсії саме для цих функцій.

Введемо в розгляд цільову функції параметр кроку:

$$E(x, \sigma^2) = \sum_{j=1}^m E_j(x, \sigma^2) = \frac{1}{2} \|e(x, \sigma^2)\|^2, \quad (2.34)$$

де

$$E(x, \sigma^2) = \frac{1}{2} e_j^2(x, \sigma^2) = \frac{1}{2} (y_j(x) - \hat{y}_j(x))^2, \quad (2.35)$$

$$e(x, \sigma^2) = (e_1(x, \sigma_1^2), \dots, e_j(x, \sigma_j^2), \dots, e_m(x, \sigma_m^2))^T, \quad (2.36)$$

$$y_j(x) = \begin{cases} 1, & \text{if } x(k) \in Cl_j, \\ 0 & \text{інакше} \end{cases} \quad (2.37)$$

навчальний опорний сигнал, сформований нулями та одиницями, так зване «one-hot coding».

Вводячи далі похідні

$$\frac{\partial E(x, \sigma^2)}{\partial \sigma_l^{-2}} = -e_j(x, \sigma_j) \frac{1 - \hat{y}_j(x)}{o^{[2]}(x)} \sum_{\tau_j} \frac{\partial o_j^{[1]}(x)}{\partial \sigma_l^{-2}}, \quad (2.38)$$

$$\frac{\partial E(x, \sigma^2)}{\partial \sigma_l^{-2}} = e_l(x, \sigma_l) \frac{1 - \hat{y}_l(x)}{o^{[2]}(x)} \sum_{\tau_l} \frac{\partial o_l^{[1]}(x)}{\partial \sigma_l^{-2}}, l \neq j, \quad (2.39)$$

або те саме:

$$\frac{\partial E(x, \sigma^2)}{\partial \sigma_j^{-2}} = e_j(x, \sigma_j) \frac{\hat{y}_j(x) - \delta_{jl}}{o^{[2]}(x)} \sum_{\tau_j} \frac{\partial o_j^{[1]}(x)}{\partial \sigma_j^{-2}}. \quad (2.40)$$

Тут

$$\delta_{jl} = \begin{cases} 1, & \text{if } j = l, \\ 0 & \text{інаше,} \end{cases} \quad (2.41)$$

$$\begin{aligned} \frac{\partial o_j^{[1]}(x)}{\partial \sigma_j^{-2}} &= -\frac{1}{2} \|x - c_{\tau_j}\|^2 \exp\left(-0,5\sigma_j^{-2} \|x - c_{\tau_j}\|^2\right) = \\ &= (x^T c_{\tau_j} - 1) \exp\left(\sigma_j^{-2} (x^T c_{\tau_j} - 1)\right). \end{aligned} \quad (2.42)$$

Алгоритм навчання параметра ширини активаційної функції може бути записаний у формі:

$$\begin{aligned} \sigma_j^{-2}(k+1) &= \sigma_j^{-2}(k) - \eta(k) \frac{\partial E_j(x(k), \sigma_j^2)}{\partial \sigma_j^{-2}} = \\ &= \sigma_j^{-2}(k) + \eta(k) \sum_{j=1}^m e_j(x(k), \sigma_j^2) \frac{\hat{y}_j(x(k)) - \delta_{jl}}{o^{[2]}(x(k))} \sum_{\tau_i} \frac{\partial o_j^{[1]}(x)}{\partial \sigma_j^{-2}} \end{aligned} \quad (2.43)$$

де $0 < \eta(k) < 1$ – параметр кроку градієнтного пошуку.

У результаті перетворень отримуємо:

$$\begin{aligned} &\sigma_j^{-2}(k) + \eta(k) \sum_{j=1}^m e_j(x(k), \sigma_j^2(k)) \frac{\hat{y}_j(x(k)) - \delta_{jl}}{o^{[2]}(x(k))} \times \\ &\times \sum_{\tau_j} \exp\left(-\frac{\|x(k) - c_{\tau_j}\|^2}{2\sigma^2(k)}\right) \|x(k) - c_{\tau_j}\|^2, \end{aligned} \quad (2.44)$$

Алгоритм навчання (2.43) істотно ускладнює чисельну реалізацію PNN, а тому його використання доцільне після того, як нейронна мережа вже побудована і потрібно уточнити параметр ширини. Перевагою ж цього підходу є те, що параметри і число R-нейрони, що утворюють перший прихований шар, можуть уточнюватися в online режимі по мірі надходження даних навчальної вибірки.

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 класифікації [53], [54], [55]. В той же час слід відзначити і деяку громіздкість цих мереж, 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.

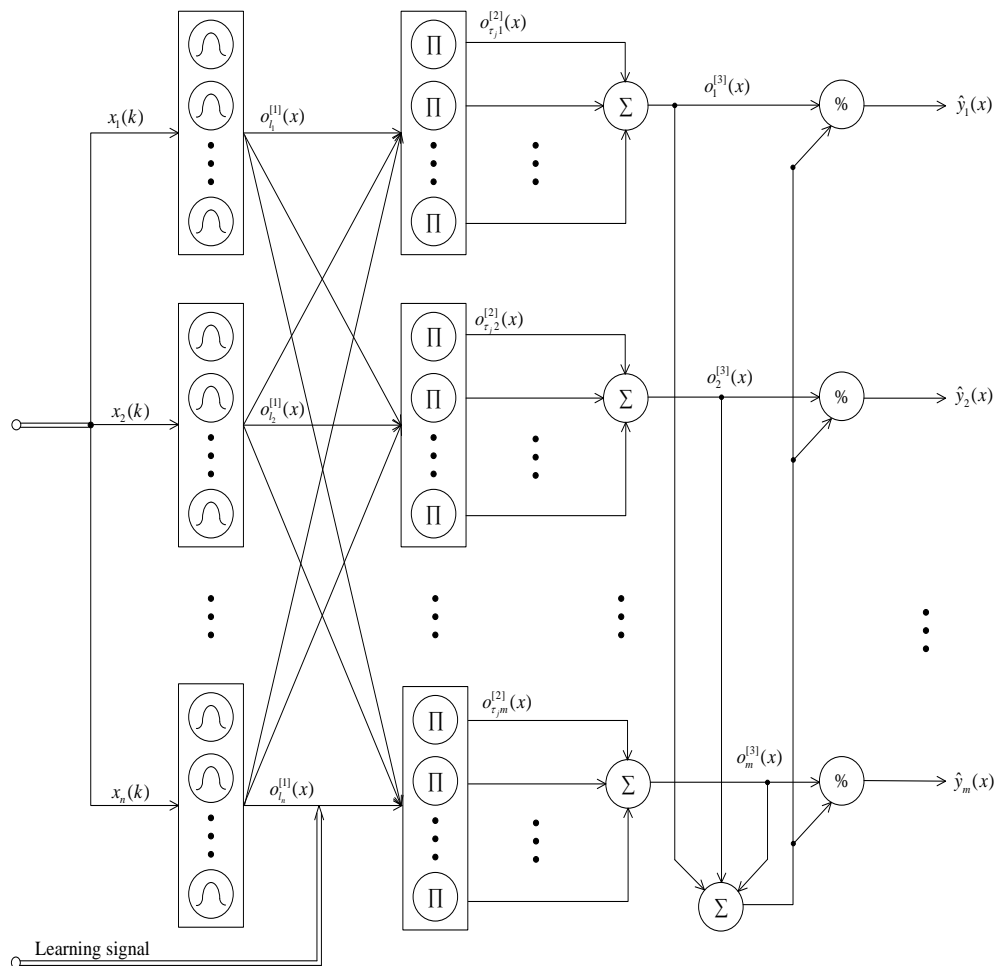


Рисунок 2.1 – Архітектура ймовірнісної нечіткої нейромережі

Подолати цю незручність можна скориставшись підходом, заснованим на гібридних системах обчислювального інтелекту [7], [8], [9], [10] і, перш за все, нейро-фаззі системами [26], що мають низку переваг перед класичними нейронними мережами, зберігаючи при цьому універсальні апроксимуючі властивості в поєднанні з можливостями навчання у онлайн режимі.

На рисунку 2.1 наведена архітектура ймовірнісної нечіткої нейромережі, що містить чотири шари обробки інформації.

На нульовий (рецепторний) шар системи послідовно надходять вектори спостережень, що формують навчальну вибірку у вигляді $X = \{x(1), x(2), \dots, x(k), \dots, x(N)\}$, $x(k) = (x_1(k), x_2(k), \dots, x_i(k), \dots, x_n(k))^T \in R^n$ (всього N спостережень, при цьому ця кількість може змінюватися у часі) також припускається, що N_1 спостережень відноситься до класу Cl_1 , N_2 до Cl_2 , N_j до Cl_j і нарешті N_m до m -го класу Cl_m .

При цьому вихідні дані можна упорядкувати так, що:

$$\left\{ \begin{array}{l} \text{якщо } x(k) \in Cl_1 \text{ тоді } k = \tau_1 = 1, 2, \dots, N_1, \\ f \ x(k) \in Cl_2 \text{ тоді } k = \tau_2 = N_1 + 1, \dots, N_1 + N_2, \\ \vdots \\ f \ x(k) \in Cl_j \text{ тоді } k = \tau_j = N_1 + N_2 + \dots + N_{j-1}, \dots, N_1 + N_2 + \dots + N_j, \\ \vdots \\ f \ x(k) \in Cl_m \text{ тоді } k = \tau_m = N_1 + N_2 + \dots + N_{m-1} + 1, \dots, N_1 + N_2 + \dots + N_m = N. \end{array} \right. \quad (2.45)$$

Ці спостереження покомпонентно надходять на перший прихований шар фаззифікації, що утворений нечіткими функціями належності $\varphi_i(x_i, c_i)$, $l_i = 1, 2, h_i$, $i = 1, 2, \dots, n$, в якості яких використовуються звичайні одновимірні функції Гаусса:

$$\varphi_i(x_i, c_i) = \left(\frac{(x_i - c_i)^2}{2\sigma^2} \right), \quad (2.46)$$

де c_i – центр відповідної функції належності;

σ – параметр ширини.

Кількість цих функцій належності на кожному вході x_i може бути різною: від двох у випадку бінарної вхідної змінної, що приймає тільки два значення «так», або «ні», до N як у стандартній ймовірнісній нейронній мережі PNN, тобто:

$$2 \leq h_i \leq N. \quad (2.47)$$

Для кожного входу, або для системи в цілому кількість функцій належності має вигляд:

$$2n \leq h = \sum_{i=1}^n h_i \leq N_n. \quad (2.48)$$

Вихідні сигнали першого шару

$$o_i^{[1]} = \varphi_i(x_i, c_i) \quad (2.49)$$

надходять на входи другого прихованого шару агрегації, що утворений N стандартними блоками множення, що формують дзвонуваті (bell shaped) функції активації у формі багатовимірних Гауссіанів:

$$\varphi_{\tau_j}^j(x_i, c_{\tau_j}) = \prod_{i=1}^n \exp\left(-\frac{(x_i - c_{\tau_j})^2}{2\delta^2}\right) = \exp\left(-\frac{\|x - c_{\tau_j}\|^2}{2\delta^2}\right), \quad (2.50)$$

де вектор-центр $c_{\tau_j} = (c_{l_1}, \dots, c_{l_i}, \dots, c_{l_n})^T$ формується на основі одновимірних центрів для кожного класу j .

При цьому кількість блоків множення для кожного класу складає N_j . Таким чином, у другому шарі формується N багатовимірних ядерних активаційних функцій, а на його виході з'являються сигнали

$$o_{\tau_j i}^{[2]} = \varphi_{\tau_j}^j(x(\tau_j), c_{\tau_j}). \quad (2.51)$$

З цієї формули можна бачити, що перші два шари системи, що пропонується, співпадають з вхідними шарами популярних нейро-фаззі-систем Такагі-Сугено-Канга або ANFIS [30] реалізують ту ж функцію, що й шар образів PNN.

Переваги системи що розглядається, виникають у випадку якщо деякі компоненти вхідних векторів-образів навчальної вибірки співпадають. Така ситуація досить часто виникає у ситуаціях, коли окремі компоненти вхідних сигналів є або бінарними, або ранговими, або номінальними змінними, що достатньо часто виникає у реальних задачах. У цьому випадку кількість функцій належності на відповідному вході h_i є меншою, ніж кількість навчальних даних N .

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

$$o_j^{[3]} = \sum_{\tau_j=N_1+N_2+\dots+N_{j-1}+1}^{N_1+N_2+\dots+N_j} o_{\tau_j}^{[2]} = p_j(x), \quad (2.52)$$

а $(m+1)$ -й підсумовує всі вихідні сигнали цього шару:

$$o^{[3]} = \sum_{j=1}^m o_j^{[3]} = \sum_{j=1}^m \sum_{\tau_j=N_1+N_2+\dots+N_{j-1}+1}^{N_1+N_2+\dots+N_j} o_{\tau_j}^{[2]}. \quad (2.53)$$

У вихідному шарі системи обчислюється ймовірність належності кожного спостереження x , що не належить до навчальної вибірки, до кожного класу у вигляді:

$$\hat{y}_j(x) = \frac{o_j^{[3]}(x)}{o^{[3]}(x)} = \frac{o_j^{[3]}(x)}{\sum_{j=1}^m o_j^{[3]}(x)}. \quad (2.54)$$

Нескладно таж побачити, що вихідний шар виконує операцію, що дуже схожа на процес дефаззифікації у відомих нейро-фаззі системах, але у нашому випадку вона має суто ймовірнісний сенс.

Для налаштування цієї системи використовується модифікована процедура лінивого навчання, в основі якої полягає концепція «Нейрони в точках даних» [47]. Саме на основі цієї концепції відбувається навчання стандартної ймовірнісної нейронної мережі, при тому коли на її вхід надходить початковий образ $x(\tau_j)$ формується багатовимірний активційний функція $\varphi_{\tau_j}^j(x, c_{\tau_j})$ центр якої визначається саме цим образом, тобто $c_{\tau_j} = x(\tau_j)$. Тобто навчання відбувається практично миттєво, але кількість активційних функцій визначається обсягом навчальної вибірки N .

Використання нейро-фаззі підходу дозволяє скоротити кількість одновимірних функцій належності, на основі яких у другому прихованому шарі формуються багатовимірні функції активації.

У найпростішому випадку компоненти образу $x(\tau_j)$ формують по N функцій належності $\varphi_{\tau_j}^j(x, c_{\tau_j})$, де центри c_{τ_j} визначаються окремими координатами $x(\tau_j)$ і ніякого виграшу у кількості функцій ми не отримуємо. Але при великих значеннях N ці одновимірні центри c_{τ_j} можуть розташовуватися на осі x_i дуже близько і практично співпадають.

У рамках нейро-фаззі-підходу всі координати вхідних сигналів передоброблені і закодовані так, що

$$0 \leq x_i(\tau_j) \leq j. \quad (2.55)$$

Виграш у кількості функцій належності виникає у випадку коли у різних навчальних образів $x(\tau_j)$ та $x(\delta_j)$ ($\tau_j \neq \delta_j$) співпадають окремі координати, тобто

$$x_i(\tau_j) = x_i(\delta_j). \quad (2.56)$$

Така ситуація досить часто виникає коли, наприклад, одна з координат є бінарною змінною 0 або 1. Тоді необхідно сформулювати всього дві функції належності з центрами у точках 0 та 1. Виграш у кількості функцій також виникає у випадку номінальних та порядкових змінних. Тоді необхідно сформувати функції належності, кількість яких дорівнює кількості імен, або рангів.

Скоротити кількість функцій належності можна також якщо задати їх максимально можливу кількість h_i по кожній координаті x_i . Тоді відстань між двома сусідніми функціями задається величиною Δ_i :

$$\Delta_i = \frac{1}{h_i - 1}. \quad (2.57)$$

Для бінарних змінних $\Delta_i = 1$, тобто центри у цьому випадку розташовуються у точках 0 та 1. Для неперервних числових змінних центри розташовуються у точках $0, \Delta, 2\Delta, \dots, 1$. Якщо якесь значення з навчальної вибірки $x_i(\tau_j)$ потрапляє в інтервал між двома центрами $l_i\Delta, (l_i + 1)\Delta$, то воно «приписується» до найближчого центру. Далі на основі цих одновимірних функцій у другому прихованому шарі формуються багатовимірні функції активації, при цьому кількість блоків множення у цьому шарі визначається числом N . При цьому передбачається, що у навчальній вибірці відсутні абсолютно однакові образи.

Таким чином, навчання PNFS зводиться до установки центрів одновимірних функцій належності та реконфігурації блоків множення у шарі агрегації. Таке налаштування системи відбувається практично миттєво.

Ймовірнісна нечітка нейронна мережа (probabilistic neuro-fuzzy system) призначена для вирішення задач розпізнавання образів класифікації, що є узагальненням ймовірнісної мережі Шпехта на випадок перетинних класів та призначена для роботи за умови як коротких, так і довгих навчальних вибірок. За своєю архітектурою запропонована система є досить близькою до нейро-фаззі системи Такагі-Сугено-Канга нульового порядку, але суттєво перевищує її за швидкістю навчання в основі якого полягає принцип «Нейрони в точках даних». Процес налаштування системи зводиться до встановлення центрів функцій належності у шарі фазифікації – реконфігурації блоків множення у шарі агрегації та відбувається практично миттєво. Запропонована система є простою у чисельній реалізації.

3 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ЙМОВІРНІСНОЇ НЕЙРО-ФАЗЗИ СИСТЕМИ ПРИ КЛАСИФІКАЦІЇ ПОТОКІВ ДАНИХ

У цьому розділі приведені результати експериментальної перевірки роботи ймовірнісної нейро-фаззи системи при вирішенні задачі нечіткої класифікації даних, що надходять поступово та представлені у бінарній, номінальній та числовій формах.

Отримані вихідні дані у результаті класифікації були зіставлені з результатами роботи популярних методів для порівняння як якості, так і швидкості роботи розробленої мережі в рамках задачі нечіткої класифікації потоку вхідних даних.

Програмне забезпечення при виконанні експериментальної перевірки роботи мережі було написано високорівневою мовою програмування Python. Ця мова є досить поширеною при використанні у сфері Data Science, оскільки завдяки простому синтаксису та широкому вибору різних бібліотек Python підвищує продуктивність при розробці та аналізі написаного коду.

3.1 Опис вхідних наборів даних

Ймовірнісна нейро-нечітка система була розроблена для роботи з різними типами даних, такими як числові та бінарні дані, які представлені у довгих та коротких наборах даних.

Тому для експериментальної оцінки з репозиторію UCI були взяті наступні вибірки з різними типами даних.

Перший набір даних «Heart Disease» містить 303 спостережень та 76 атрибутів, проте використовується підмножина – лише 14 з них. Кожен із спостережень містить детальну інформацію про пацієнта, його фізіологічні параметри та симптоми захворювання.

Мета використання цієї вибірки – знаходження наявності хвороби серця у пацієнта, що є цілим числом, значення якого варіюється між 0 та 4, де нуль – відсутність захворювання. Загальне використання цієї вибірки націлене на знаходження наявності, або відсутності захворювання.

Приведений набір даних використовується в рамках вирішення даної задачі як коротка вибірка, що поєднує в собі різні типи даних, такі як числові та бінарні дані.

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

Другий набір даних, «Diabetes 130-US hospitals for years 1999-2008», є довгим набором даних, який містить 100 000 випадків. Він включає значення параметрів, що представляють результати лікування пацієнтів: тривалість перебування в лікарні, інформацію про лабораторні дослідження та ліки, що вводяться, коли пацієнти перебували в лікарні. Цей набір даних, як і перший датасет, містить числові та двійкові дані.

Наступні набори даних, що використовуються для перевірки роботи мережі, представлені у формі зображень, що зберігаються у векторній формі. Також важливо зазначити, що для коректної перевірки роботи мережі в умовах нечткості необхідно брати зображення різної якості. Тому, гарним рішенням у цьому випадку буде взяти один датасет що містить зображення з низькою якістю, та інший з якістю вище, ніж попередній вибір.

Було обрано два набори даних, що містять зображення та відповідають зазначеним критеріям.

Перший набір даних «Fashion MNIST» містить зображення предметів одягу. Кожне зображення одягу представлене матрицею 28x28, а довжина навчального набору включає 60 000 зображень.

Приклад даних із набору «Fashion MNIST» представлено на рисунку 3.1.



Рисунок 3.1 – Приклад спостережень з набору даних «Fashion MNIST»

Наступний набір даних – це «ML hand-written digits» із UCI репозиторію, Він містить 1797 розмічених спостережень. Кожна цифра представлена у матричній формі з роздільною здатністю 8x8 пікселів. Більш детальний опис вхідної вибірки представлений на рисунку 3.2. Кожне зображення має низьку якість, що дуже корисно у випадку перетинних класів. Також у реальних задачах початкові вибірки не завжди мають якісні вхідні дані. Приклади з цього набору даних наведені на рисунку 3.3.

Each datapoint is a 8x8 image of a digit.

| | |
|-------------------|---------------|
| Classes | 10 |
| Samples per class | ~180 |
| Samples total | 1797 |
| Dimensionality | 64 |
| Features | integers 0-16 |

Рисунок 3.2 – Характеристики набору даних «ML hand-written digits»

Ряд зображень, що представлений на рисунку 3.3, містить елементи вибірки «ML hand-written digits» з низькою якістю зображення, що у деяких випадках робить їх досить схожими між собою.

Наприклад, перше та друге зображення у другому ряду відносяться до різних класів, проте з точки зору людини вони подібні.

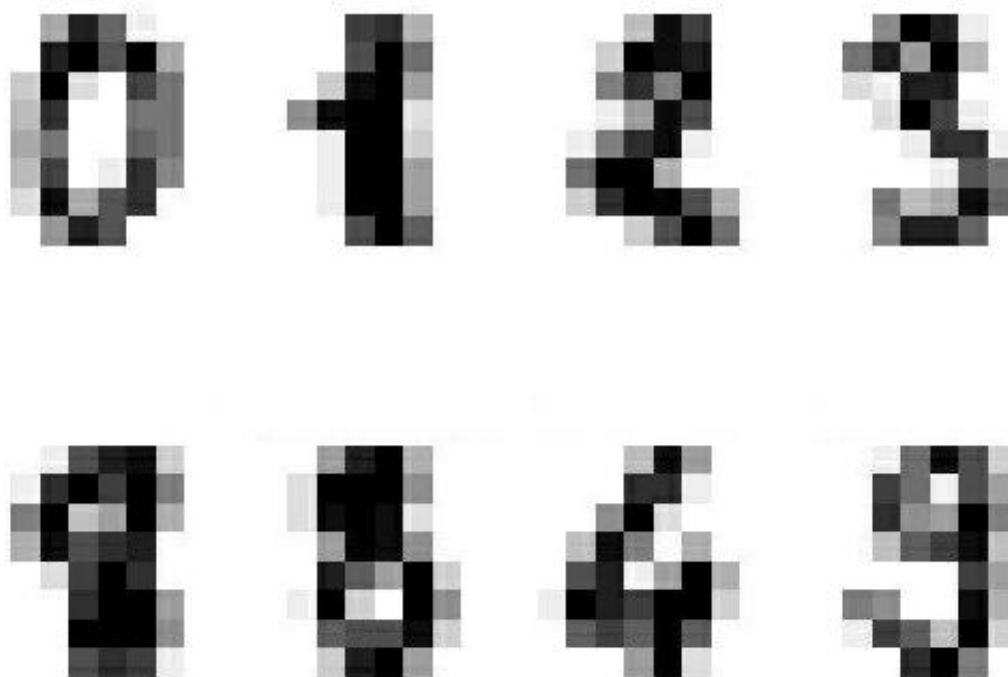


Рисунок 3.3 – Приклад спостережень з набору даних «ML hand-written digits»

Варто зауважити, що в цій роботі також передбачається, що зображення монохромні, проте потенційно, ця мережа спроможна до розпізнавання кольорових зображень.

Представлені набори даних містять класи, що перетинаються у просторі ознак, які представлені на рисунку 3.4 та на рисунку 3.5.

Рисунок 3.4 показує клас числа 8; однак серед основного кластера екземплярів ви також можна побачити багато зразків класу числа 1.

На рисунку 3.5 легко побачити три різні класи, екземпляри яких присутні не лише серед власних зразків класу, але також у кластерах інших класів. Ці візуалізації показують випадок перекривання класів у цих наборах даних.



Рисунок 3.4 – Приклад перетинних класів з набору даних «ML hand-written digits»

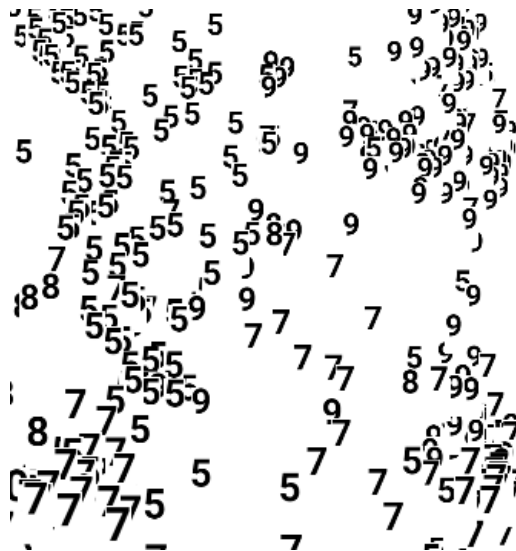


Рисунок 3.5 – Приклад перетинних класів з «Fashion MNIST»

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

3.2 Аналіз результатів експериментів

Експериментальна перевірка ймовірнісної нейро-фаззі системи проведена з метою порівняння обраних критеріїв оцінки її роботи, а саме точності класифікації та швидкості обробки вхідних даних.

Перший експеримент був проведений з коротким набором медичних даних – «Heart Disease». Далі цей набір даних був додатково розбитий на підмножини з метою визначення мінімальної кількості елементів вибірки для отримання певної якості результатів класифікації. Були порівняні отримані значення точності класифікації ймовірнісної нейро-фаззі системи і популярного методу машинного навчання KNN – K – найближчих сусідів. Отримані результати експериментальної оцінки представлені в таблиці 3.1.

Таблиця 3.1 – Порівняння точності класифікації алгоритмів у випадку короткого набору даних

| Алгоритми | Точності класифікації | | | | Максимальний час, сек |
|---------------------------------|-----------------------|-------|-------|-------|-----------------------|
| | 250 | 200 | 150 | 100 | |
| Ймовірнісна нейро-фаззі система | 77.52 | 69.34 | 57,7 | 51.14 | 0.79 |
| EFPNN | 79.02 | 71.83 | 61.9 | 56.07 | 0.1 |
| KNN | 49.03 | 50.7 | 51,63 | 50.24 | 0.03 |

Як можна бачити з представленої таблиці, алгоритм KNN швидкий за інші піходи, але його точність близько 50%. Таким чином, приведений алгоритм не призначений для класифікації дуже коротких зразків. На відміну від KNN, точність класифікації запропонованої мережі зростає із збільшенням кількості елементів у вибірці. Навіть на дуже малих зразках він досягає точності 77%. EFPNN також забезпечує більшу точність із

збільшенням обсягу вибірки. Однак це значно, більш ніж на 20% повільніше, ніж запропонована мережа. Найшвидшим методом є KNN, але слід врахувати, що нейронні мережі були реалізовані за допомогою мови програмування Python і працюють на центральному процесорі, а не на графічному процесорі, як KNN. Тобто, при однаковій апаратній реалізації тимчасові витрати для всіх методів будуть схожими, проте точність ймовірнісної нейро-фаззи системи вища.

Другий експеримент був проведений за допомогою довгого набору даних, який називається «Diabetes 130-US hospitals for years 1999-2008». Результати експерименту представлено на рисунку 3.6.

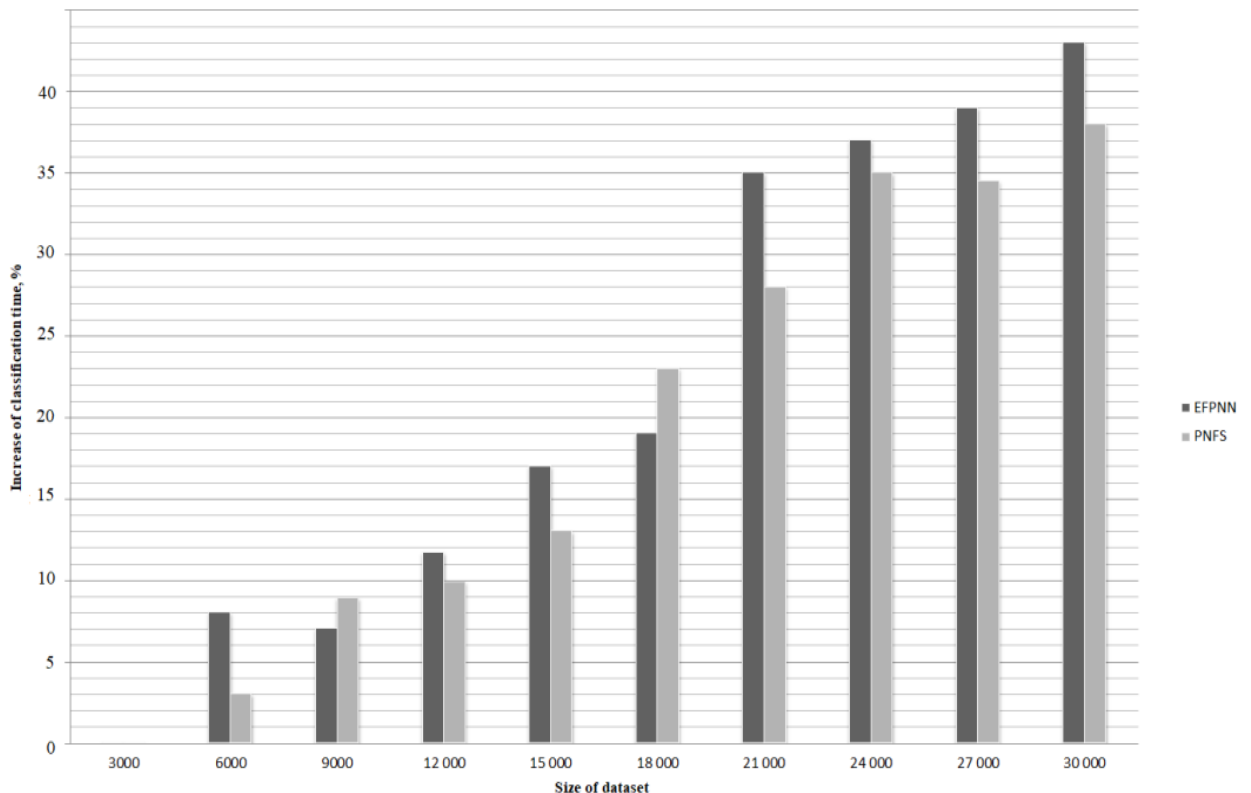


Рисунок 3.6 – Графік збільшення часу класифікації в залежності від розміру набору даних

Як і в першому експерименті, з початкового датасету було сформовано ряд підмножин, що мають різні розміри, від 3000 до 30000

екземплярів. Метою експерименту є порівняння приросту часу класифікації із збільшенням розміру набору даних, оскільки абсолютне споживання часу залежить від комп'ютерної платформи та використовуваного процесора (CPU, GPU). Проаналізувавши результати першого експерименту, в подальшому аналізі доцільно порівнювати дві нейромережі PNFS та EFPNN, які забезпечують більш високу точність класифікації. Експеримент показав, що ймовірнісна нейро-фаззі мережа вимагає менших обчислювальних витрат, ніж EFPNN. Приріст часу, необхідного для підмножин великого розміру, значно збільшується порівняно з підмножинами маленького розміру. Менші датасети зазвичай розміщуються в оперативній пам'яті, тоді як довгі набори даних вимагають обміну даними із зовнішньої пам'яті.

Загалом, згідно з результатами двох експериментів, запропонований підхід у порівнянні з EFPNN забезпечує дещо нижчу точність класифікації для малих наборів даних, але вимагає значно менших обчислювальних витрат, коли розмір набору даних зростає.

Наступний експеримент із підходом, що розглядається у даній роботі, проводився з використанням наступних наборів даних: «Fashion MNIST» та «ML hand-written digits». Для порівняльного аналізу були взяті популярний метод машинного навчання KNN, Evolving fuzzy-probabilistic neural network (EFPNN), яка була запропонована у [39], та ймовірнісна нейро-фаззі система (PNFS).

Таблиця 3.2 – Точність та часове споживання алгоритмів з використанням набору даних «Ml Hand-Written Digits»

| Algorithms for comparison | Classification accuracies | Time, sec |
|---------------------------|---------------------------|-----------|
| KNN | 81.98 | 0.18 |
| EFPNN | 96.39 | 7.02 |
| PNFS | 93.07 | 5.52 |

Під час першої частини даного експерименту був взятий набір даних «ML hand-written digits», а отримана точність класифікації разом із затраченим часом фіксувалися. Отримані результати представлені в таблиці 3.2.

Як можна бачити з таблиці 3.2, імовірнісна нейро-фаззі система працює швидше, ніж EFPNN; однак вона поступається EFPNN в точності класифікації. Слід врахувати, що алгоритм KNN був виконаний на графічному процесорі, а ймовірнісна нейро-нечітка система на центральному процесорі, що призвело до невідповідності між витраченим часом кожного з підходів.

Тим не менше, якби імовірнісна нейро-фаззі система працювала на графічному процесорі, витрачений час був би порівнянним, оскільки швидкість обчислень з плаваючою точкою у сучасного графічного процесора вище у п'ять-десять разів.

Для другої частини даного експерименту було взято набір даних «Fashion MNIST». Під час цієї частини експерименту початковий набір даних був розділений на дев'ять наборів даних з різними обсягами, і відстежувалася точність класифікації. Кожен із цих наборів даних був сформований випадковим чином.

Із початкового набору даних «Fashion MNIST», найбільший із сформованих наборів даних має 15 000 спостережень. Крім того, параметр ширини активаційної функції був обраний емпірично для початкового набору даних для отримання найкращих результатів. Для довгого набору даних параметр широкого поширення дорівнює 0,74. Результати представлені на рисунку 3.7.

Як видно з графіку, алгоритм KNN забезпечує найменше збільшення часу обчислення, залежно від розміру набору даних. Однак цей алгоритм має меншу точність порівняно з алгоритмами, заснованими на імовірнісних нейронних мережах.

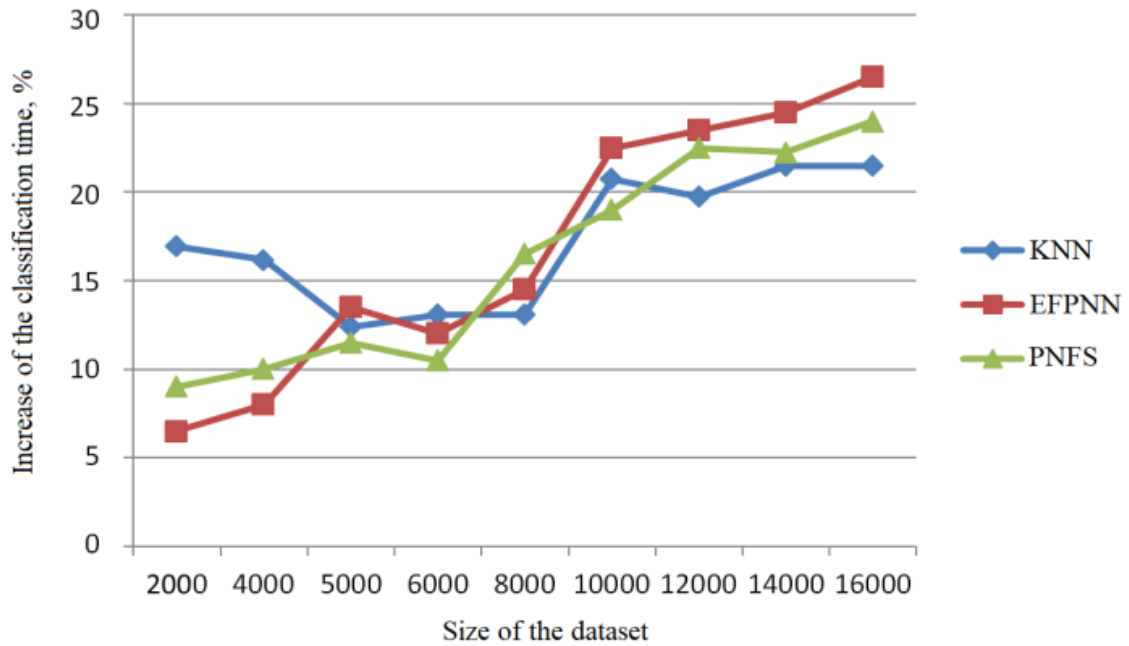


Рисунок 3.7 – Графік залежності часу класифікації від розміру набору даних

Алгоритм, що розглядається у даній роботі – PNFS вимагає менших обчислювальних навантажень порівняно з алгоритмом EFPNN з подібною точністю класифікації.

На наборах даних із більшим розміром, що перевищує 8000 елементів, цей алгоритм показує майже лінійне збільшення обчислювальних витрат із збільшенням обсягу вибірки.

ВИСНОВКИ

В магістерській атестаційній роботі представлені результати вирішення задачі удосконалення ймовірнісної нейро-фаззі система, а також її гібридного навчання, що орієнтована на вирішення актуальної задачі обробки потоків даних у режимі онлайн.

В процесі виконання атестаційної роботи отримано такі результати.

1. Виконано аналіз сучасного стану теорії штучних нейронних мереж, орієнтованих на швидку обробку потоків даних різних типів у режимі онлайн. Проведений аналіз показав, що розв'язання даної задачі потребує інтеграції можливостей швидкодіючих ймовірнісних нейронних мереж для задоволення обмежень у часі при послідовній обробці великих об'ємів даних, а також переваг нечіткої логіки для виділення класів, що перетинаються в просторі ознак.

2. Розроблено архітектуру ймовірнісної нейро-фаззі системи з гібридним навчанням, що містить рецепторний шар, на який надходять вектори спостережень, перший прихований шар, що утворений нечіткими функціями належності, шар агрегації, третій прихований шар із суматорів, які розраховують Парзенівські оцінки щільності розподілу даних у кожному класі, а також вихідний шар, у якому обчислюється ймовірність належності кожного спостереження до відповідного класу. Перші два шари розробленої системи реалізують функцію шару образів традиційної ймовірнісної мережі і є аналогічними вхідним шарам відомих нейро-фаззі-систем. Вихідний шар виконує операцію, аналогічну процесу дефаззіфікації, але для даної архітектури вона має суто ймовірнісний характер.

3. Виконано експериментальну перевірку можливостей ймовірнісної нейро-фаззі системи за критеріями точності класифікації та швидкості обробки вхідних даних. При проведенні експериментів використовувались як числові, так і бінарні вхідні дані. Результати експериментів показали,

що розроблена ймовірнісна нейро-фаззі система забезпечує підвищену точність на вибірках малого розміру, що створює умови для швидкої адаптації системи при зміні типів вхідних даних. При збільшенні розміру виборки розроблена мережа потребує менше часу у порівнянні з адаптивною ймовірнісною нейронною мережею.

В цілому запропонована ймовірнісна нейро-фаззі система з гібридним навчанням забезпечує швидку обробку потоків даних у режимі онлайн.

4. Результати досліджень доповідались на низці міжнародних конференцій [38], [39], [40], [42], [55] та публіковалися у низці публікацій [58], [59].

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Lemaire V., Salperwyck C., Bondu A. A Survey on Supervised Classification on Data Streams. *Business Intelligence*. European Business Intelligence Summer School. 2014. P. 88-125.
2. Stonebraker, M., Çetintemel, U., Zdonik, S. The 8 requirements of real-time stream processing. *ACM SIGMOD*. 2005. № 34(4). P. 42–47.
3. Gama J. Knowledge Discovery from Data Streams. Chapman and Hall: CRC Press, 2010. 256 p.
4. Domingos P., Hulten G. Mining high-speed data streams. KDD '00. Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, August, 2000. P. 71–80. DOI: <https://doi.org/10.1145/347090.347107>.
5. Gaber M., Zaslavsky A., Krishnaswamy Sh. Mining Data Streams: A Review. *ACM SIGMOD*. 2005. № 34(2). P. 18-26.
6. Heinz C. Density estimation over data streams: Dissertation zur Erlangung des Doktorgrades der Naturwissenschaften/ Fachbereich Mathematik und Informatik der Philipps-Universität Marburg. Marburg, 2007. 201 p.
7. Kacprzyk J., Pedrycz W., Springer Handbook of Computational Intelligence, Berlin Heidelberg: Springer, Verlag, 2015. 1179 p.
8. Kruse R., Borgelt C., Klawonn F., Moewes C., Steinbrecher M., Held P. Computational Intelligence. A Methodological Introduction. Berlin: Springer-Verlag, 2013. 490 p.
9. Mumford C., Jain L. Computational Intelligence, Collaboration, Fuzzy and Emergence. Berlin: Springer, Verlag, 2009. DOI: 10.1007/978-3-642-01799-5.
10. Mumford C.L., Jane L.C. Computational Intelligence. Berlin: Springer-Verlag, 2009. 1539 p.

11. Rutkowski L. Computational Intelligence. Methods and Techniques. Berlin-Heidelberg: Springer-Verlag, 2008. 514 p.
12. Бодянский Е.В., Руденко О.Г. Искусственные нейронные сети: архитектуры, обучение, применение. Харьков: ТЕЛТЕХ, 2004. 372 с.
13. Осовский С. Нейронные сети для обработки информации/Пер. с польского И.Д. Рудинского. М.: Финансы и статистика, 2002. 344 с.
14. Schalkoff R.J. Artificial Neural Networks. NY: The McGraw-Hill Comp., Inc., 1997. 448 p.
15. Kohonen T. Improved version of learning vector quantization. *Neural Networks: Proceedings of the 1990 IJCNN International joint conference on Neural Networks*, San Diego, CA, 1990. P. 545-550.
16. Kasabov N. Evolving Connectionist Systems. London:Springer-Verlag, 2003. 273 p.
17. Kazman R., Kominek J. Information organization in multimedia resources. *ACM: Proceedings of the 11th annual international conference on Systems documentation*, 1993. P. 149–162.
18. Diallo B., Dolidon F., Travers J.-M., Mazoyer B. B-spide: An object-relational database architecture to store, retrieve, and manipulate neuroimaging data. *Human brain mapping*, 1999. Vol. 7, No. 2. P. 136–150.
19. Martinez M., Koenen R., Pereira F. Mpeg-7: the generic multimedia content description standard. *Computer Science. MultiMedia*, IEEE, 2002. Vol. 9, No. 2. P. 78–87.
20. Stankovic M., Falkowski B. J., Jankovic D., Stankovic R. S. Calculation of the paired haar transform through shared binary decision diagrams. *Computers & Electrical Engineering*, 2003. Vol. 29, Iss. 1. 1324 p.
21. Syerov Yu., Shakhovska N., Fedushko S. Method of the Data Adequacy Determination of Personal Medical Profiles. *Advances in Artificial Systems for Medicine and Education II*. Springer Nature Switzerland AG, 2018, P. 333-343. doi: 10.1007/978-3-030-12082-5_31.

22. Kohonen T. Self-Organizing Maps Berlin: Springer-Verlag, 1995. 362 p.
23. Graupe D. Deep Learning Neural Networks: Design And Case Studies. N.Y.: World Scientific, 2016.
24. Specht D. F. Probabilistic neural networks Neural Networks. 1990. P. 109-118.
25. Specht D. F. Probabilistic neural networks and polynomial ADALINE as complementary techniques to classification. *IEEE Trans. on Neural Networks*, 1990. Vol.1, Iss. 1. P.111-121.
26. Souza P.V.C. Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. *Applied Soft Computing*. 2020. Vol. 92.
27. Yoichi H. Neural expert system using fuzzy teaching input and its application to medical diagnosis. *Information Sciences Applications*. Vol.1, Iss.1. January, 1994. P. 47-58.
28. Jang J.-S., Sun C. T., Mizutani E. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. *IEEE Transactions on automatic control*. 1997. Vol. 42, Iss. 10. P. 1482-1484.
29. Lughofer E. Evolving Fuzzy Systems-Methodologies, Advanced Concepts and Applications. Berlin Heidelberg: Springer-Verlag, 2011. 456 p.
30. Jang, J.-S.R. ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics*, 1993. Vol.23, Iss3.
31. Jang, Jyh-Shing R. Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm. *AAAI-91: Proceedings of the 9th National Conf. on Artificial Intelligence*. Anaheim, CA, USA. July 14–19, 1991. P. 762–767.
32. Nelles O. Nonlinear Systems Identification. Berlin: Springer, 2001. 785 p.
33. Bengio Y., Le Cun Y., Hinton G. Deep learning Nature, 2015. P. 436-444.

34. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. 787 p.
35. Schmidhuber J. Deep learning in neural networks: An overview *Neural Networks*, 2015. P. 85-117.
36. Bodyanskiy Ye., Pliss I., Volkova V. Modified probabilistic neuro-fuzzy network for text document processing. *International Journal of Computing*. 2012. Vol. 11, Iss. 4. P. 391-396.
37. Bodyanskiy Ye. Tyshchenko O., Deineko A. Evolving Neuro-fuzzy Systems with Kernel Activation Functions. Their Adaptive Learning for Data Mining Tasks, volume 58, Saar-brücken: LAP LAMBERT Academic Publishing, 2015. 64 p.
38. Bodyanskiy Ye., Deineko A., Pliss I. Chala O. Adaptive probabilistic neural network with fuzzy inference and its online learning. *CSIT'20: Proceedings of the XV International Scientific and Technical Conference Computer Science and Information Technologies*, Zbarazh Castle, Ukraine, 23-26 September, 2020.
39. Bodyanskiy Ye., Deineko A., Pliss I., Chala O. Evolving fuzzy-probabilistic neural network and its online learning. *ACIT 2020: Proceedings of the 10th International conference on advanced computer information technologies*, Deggendorf, Germany, Germany 16-18 September 2020. P. 465-468. DOI: 10.1109/ACIT49673.2020.9208904.
40. Bodyanskiy Ye., Deineko A., Pliss I., Chala O. Probabilistic neuro-fuzzy system in medical diagnostic task and its lazy learning-selflearning. *IDDM'20: Proceedings of the 3rd International Conference on Informatics & Data-Driven Medicine*, Växjö, Sweden, November 19-21, 2020. P.29-35.
41. Жернова П.Є., Плісс І.П., Чала О.С. Модифікована нечітка ймовірнісна нейронна мережа. *Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту: матеріали міжнародної наукової конференції*. Херсон: Видавництво ФОП Вишемирський В. С., 2018. С. 228-230.

42. Кулішова Н.Є., Бодянський Є.В., Плісс І.П., Чала О.С. Нео-фаззі система та її оптимальне навчання у завданні розпізнавання образів у реальному часі. *Міжнародний науковий симпозиум “Інтелектуальні рішення” Обчислювальний інтелект (результати, проблеми, перспективи)*: Матеріали V міжнародної науково-практичної конференції, м. Ужгород, 15–20 квітня 2019 р. Ужгород, 2019. С.106-107.

43. Мірошніченко Н.С., Чала О.С. Медичне діагностування захворювань щитоподібної залози за допомогою нео-фаззі нейрону. *Міжнародний науковий симпозиум “Інтелектуальні рішення” Обчислювальний інтелект (результати, проблеми, перспективи)*: Матеріали V міжнародної науково-практичної конференції, м. Ужгород, 15–20 квітня 2019 р. Ужгород, 2019. С. 115-117.

44. Angelov P. *Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems*. Heidelberg, New York: Springer-Verlag, 2002. 197 p.

45. Angelov P. Kasabov N. *Evolving computational intelligence systems*. Proceedings of the 1st International Workshop on Generic Fuzzy Systems, Granada, Spain, 2005. P.76-82.

46. Yi J.-H., Wang J., Wang G.-G. Improved probabilistic neural networks with self-adaptive strategies for transformer fault diagnosis problem- advances. *Mechanical Engineering*. 2016. Vol.8, Iss. 1. P. 1-13.

47. Zahirniak D.R., Chapman R, Rogers S.K., Suter B.W., Kabriski M., Pyatti V. Pattern recognition using radial basis function network. *Aerospace Application of Artificial Intelligence: 6 International Conference (5-8 June 1990, Proceedings, Dayton, Ohio)*. P. 249–260.

48. Bodyanskiy Ye., Gorshkov Ye., Kolodyazhniy V., Wernstedt J. A learning of probabilistic neural network with fuzzy inference. *ICANN'03: Proceedings Sixth International Conference on Artificial Neural Nets and Generic Algorithms*, Wien:Springer-Verlag, 2003. P.13-17.

49. Bodyanskiy Ye., Gorshkov Ye., Kolodyazhniy V. Resource-Allocating Probabilistic Neuro-Fuzzy Network. *EUSFLAT 2003: Proceedings of the 2nd Conference of European Union Society for Fuzzy Logic and Technology*, Zittan, Germany, 10-12 September 2003. P. 392-395.

50. Bodyanskiy Ye., Gorshkov Ye., Kolodyazhniy V., Wernstedt J. Probabilistic neuro-fuzzy network with non-conventional activation functions. *Lecture notes in Artificial Intelligence*, vol. 2773, Berlin Heidelberg New York:Springer, 2003. P. 973-979.

51. Parzen E. On the estimation of a probability density function and the mode, *Ann. Math.Statistics*, vol.33, 1962, pp.1065-1076.

52. Bezdek J.C. Pattern recognition with fuzzy objective function algorithms. N.Y.: Plenum Press, 1987. 250 p.

53. Чалая О.С. Матричная вероятностная нейро-фаззи система в задаче распознавания образов. *Радіоелектроніка та молодь у XXI столітті: Матеріали 23-го Міжнародного молодіжного форуму*. Т.6. Харків: ХНУРС. 2019. С. 5-6.

54. Bodyanskiy Y., Kulishova N., Chala O. The Extended Multidimensional Neo-Fuzzy System and Its Fast Learning in Pattern Recognition Tasks. *Data*. 2018. Vol.3, Iss.4, №63 DOI: 10.3390/data3040063

55. Bodyanskiy Ye., Deineko A., Pliss I. Chala O. Matrix fuzzy-probabilistic neural network in image recognition task. *DSMP'20: Proceedings of the IEEE Third International Conference on Data Stream Mining & Processing*, Lviv, Ukraine, 21-25 August, 2020. P. 33– 36.

56. Rutkowski L. Adaptive probabilistic neural networks for pattern classification in time-varying environment. *IEEE Trans. on Neural Networks*. 2004. P. 811-827. DOI: 10.1109/TNN.2004.828757.

57. Zhernova P., Pliss I., Chala O. Modified fuzzy probabilistic neural network. *ISDMCI'2018: Proceedings of the XIV International Scientific Conference Intellectual Systems For Decision Making And Problems Of Computational Intelligence*, Kherson: PP Vyshemirsky V. S. 2018. P. 228-230.

58. Bodyanskiy Ye., Deineko A., Pliss I. Chala O. Fast Probabilistic Neuro-Fuzzy System for Pattern Classification Task. *ITMS 2020: The 61st International Scientific Conference on Information Technology and Management Science of Riga Technical University, Riga, Latvia, October 15-16, 2020.*

59. Bodyanskiy Ye., Deineko A., Pliss I. Chala O. Adaptive Learning of Probabilistic Neural Network in Situation of Overlapping Classes in Classification Task. *CSIT2020: XV International Scientific and Technical Conference on Computer Science and Information Technologies, Springer in publish.*