Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Ядерне послідовне кластерування потоків даних
на основі еволюційних нейронних мереж
(тема)

Виконав:
студент 2 курсу, групи СШІм-19-2
Григор`єв Д. С.
(прізвище, ініціали)

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

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

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник к.т.н., доцент, Шевченко О. Ю.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

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

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

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

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

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Григор`єву Дмитру Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Ядерне послідовне кластерування потоків даних на основі еволюційних нейронних мереж

затверджена наказом університету від 29 березня 2021 р. № 390Ст

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

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет та відомих наукових проектів, електронні документації, тестові набори даних

4. Перелік питань, що потрібно опрацювати в роботі аналіз предметної області та постановка задачі дослідження, навчання нейронних мереж, засноване на оптимізації, нейронні мережі, що миттєво навчаються, самоорганізовні нейронні мережі, гібридні системи обчислювального, ядерні функції активації та їх види, адаптивне навчання узагальненої регресійної нейронної мережі, середовище реалізації, опис алгоритму, аналіз отриманих результатів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1 – Архітектура стандартної конкурентної нейронної мережі, Рисунок 2 – Архітектура ядерної самоорганізовної мапи Т. Кохонена, на основі узагальненої регресійної нейронної мережі, Рисунок 3 – Результати кластеризації штучно згенерованої вибірки, Рисунок 4 – Результати кластеризації тестової вибірки «Iris», Рисунок 5 – Зміщення центроїдів тестової вибірки «Iris», Рисунок 6 – Результати кластеризації тестової вибірки «Wine», Рисунок 7 – Зміщення центроїдів тестової вибірки «Wine», Рисунок 8 – Середня точність кластеризації розглянутих методів

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	доц. каф. ШІ Шевченко О.Ю.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів	Примітка
1.	Отримання завдання на кваліфікаційне проектування	29.03.2021	виконано
2.	Аналіз завдання та пошук літератури за темою	30.03-05.04	виконано
3.	Опрацювання літератури та аналіз об'єкту	06.04-12.04	виконано
4.	Вибір програмних засобів для розробки системи	13.04-19.04	виконано
5.	Розробка програмного засобу	20.04-03.05	виконано
6.	Аналіз отриманих результатів	04.04-06.05	виконано
7.	Оформлювання пояснювальної записки	06.05-11.05	виконано
8.	Оформлення презентаційних матеріалів	12.05.2021	виконано
9.	Представлення на рецензування	13.05.2021	виконано
11.	Представлення кваліфікаційної роботи	18.05.2021	

Дата видачі завдання 29 березня 2021 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Шевченко А.Ю.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 80 с., 16 рис., 3 табл., 42 формули, 2 дод, 36 джерел

ДИНАМІЧНИЙ АНАЛІЗ ДАНИХ, ГІБРИДНА СИСТЕМА, КЛАСТЕРУВАННЯ ДАНИХ, УЗАГАЛЬНЕНА РЕГРЕСІЙНА НЕЙРОННА МЕРЕЖА, ЯДЕРНА НЕЙРОННА МЕРЕЖА.

Об'єкт дослідження – процес обробки даних, що надходять в online режимі за допомогою ядерної кластерувальної мережі в умовах кластерів довільної форми та невідомої їх кількості.

Предмет дослідження – методи кластеризації на основі ядерних функцій в задачах інтелектуального аналізу даних.

Метою даного дослідження є розробка методу кластеризації на основі ядерної нейронної мережі, яка налаштовує свою архітектуру в процесі навчання-самонавчання, в умовах кластерів довільної форми, що перетинаються.

Методи дослідження – теорія обчислювального інтелекту і м'яких обчислень; теорія штучних нейронних мереж; теорія нечіткої логіки; теорія оптимізації і статистичний аналіз; імітаційне моделювання.

Припускається, що задача розробки нового методу ядерної кластеризації, призначеного для обробки даних в online режимі, коли дані надходять на обробку послідовно, одне за одним, а кластери можуть перекриватися і мати довільну форму на сьогоднішній день є актуальною і такий підхід може бути використано для вирішення широкого класу задач динамічного аналізу даних.

РЕФЕРАТ

Пояснительная записка: 80 с., 16 рис., 3 табл, 42 формулы, 2 прил., 36 источников.

ДИНАМИЧЕСКИЙ АНАЛИЗ ДАННЫХ, ГИБРИДНАЯ СИСТЕМА, КЛАСТЕРИЗАЦИЯ ДАННЫХ ОБОБЩЕННАЯ РЕГРЕССИОННАЯ НЕЙРОННАЯ СЕТЬ, ЯДЕРНАЯ НЕЙРОННАЯ СЕТЬ.

Объект исследования – процесс обработки данных, поступающих в online режиме с помощью ядерной кластерувальнои сети в условиях кластеров произвольной формы и неизвестной их количества.

Предмет исследования – методы кластеризации на основе ядерных функций в задачах интеллектуального анализа данных.

Целью данного исследования является разработка метода кластеризации на основе ядерной нейронной сети, которая настраивает свою архитектуру в процессе обучения-самообучения, в условиях кластеров произвольной формы, пересекаются.

Методы исследования – теория вычислительного интеллекта и мягких вычислений; теория искусственных нейронных сетей; теория нечеткой логики; теория оптимизации и статистический анализ; имитационное моделирование.

Предполагается, что задача разработки нового метода ядерной кластеризации, предназначенного для обработки данных в online режиме, когда данные поступают на обработку последовательно, одно за другим, а кластеры могут перекрываться и иметь произвольную форму на сегодняшний день является актуальной и такой подход может быть использован для решения широкого класса задач динамического анализа данных.

ABSTRACT

Explanatory note: 80 p., 16 fig., 3 tabl., 42 formulas, 2 ann., 36 sources

DATA CLUSTING, DYNAMIC DATA MINING, GENERAL REGRESSION NEURAL NETWORK, HYBRID SYSTEM, KERNEL NEURAL NETWORK.

The object of the research is the process of processing data coming online using a nuclear cluster network in the conditions of clusters of arbitrary shape and their unknown number.

The subject of research is clustering methods based on nuclear functions in data mining tasks.

The purpose of the research is to develop a clustering method based on a nuclear neural network, which adjusts its architecture in the process of learning-self-learning, in the conditions of clusters of arbitrary shape, intersect.

Research methods – theory of computational intelligence and soft computing; theory of artificial neural networks; fuzzy logic theory; optimization theory and statistical analysis; simulation modeling.

It is assumed that the task of developing a new method of nuclear clustering designed for online data processing, when data is sent for processing sequentially, one after another, and the clusters can overlap and have an arbitrary shape, is currently relevant and such an approach can be used to solve a wide class of problems in dynamic data analysis.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

НРБНМ – нормалізованими радіально-базисними нейронними мережами;

РБФ – радіальними базисними функціями;

ШНМ – штучна нейрона мережа;

DDM – dynamic data mining – динамічний аналіз даних;

GRNN – general regression neural network – узагальнена регресійна нейронна мережа;

RBFN – radial basis function network – радіально-базисна нейронна мережа;

SOM – self-organizing Map – самоорганізовна мапа.

ВСТУП

На сьогоднішній день технології штучних нейронних мереж (ШНМ) досить часто і досить успішно використовуються у вирішенні складних задач, які, як правило, не мають аналітичного рішення. На сьогодні нейронні мережі стають все поширенішими для вирішення різних задач обробки сигналів, оптимізації, оптимального і адаптивного управління, розпізнавання образів, ідентифікації, прогнозування в реальному часі і т.п. Створено реальні системи обробки зображень та комп'ютерного зору, управління аерокосмічними об'єктами, технічної та медичної діагностики, в економіці і фінансах (планування, управління, аналіз ринків, прогнозування курсів, технічний аналіз, пошук інформації, ідентифікація кредитних карт), у військовій справі, управлінні рухом, в енергетиці (оцінка станів, виявлення розладнань, прогнозування навантажень), в криміналістиці, аналізі сигналів різної природи та ін., причому цей перелік розширюється практично щодня. Однією з основних завдань в області інтелектуального аналізу даних є задача кластеризації, тобто задача розбиття вихідних даних на однорідні в деякому сенсі групи, в режимі навчання без вчителя (самонавчання).

На сьогоднішній день існує велика кількість методів кластеризації, що відрізняються між собою як математичними постановками задач так і результатами. При цьому більшість популярних алгоритмів припускають, що кластери мають опуклу форму і лінійно роздільні, але, на жаль, в реальних задачах – це далеко не так, оскільки кластери можуть мати довільну форму. Для вирішення таких задач також існує ряд алгоритмів, одним з найбільш популярних підходів при цьому є методи ядерної кластеризації [1]. В рамках цього підходу передбачається, що кластери можуть мати довільну форму, але вони не перетинаються і не перекриваються, а вся вибірка, що має бути кластеризована, задається апріорі.

На даний момент є достатня кількість інформації про діяльність підприємств, лікарень, фірм, яка відображає діяльність цих об'єктів. Проаналізувавши дану інформацію, можна знайти об'єктивні закономірності за умови того, що сформована таблиця відображає фактичні дані, що відображають причинно-наслідкові залежності.

Дана інформація збирається протягом багатьох років (наприклад, темпи інфляції, рівень доходів населення, структура витрат населення, вартість послуг житлово-комунального господарства, стан промислового і сільськогосподарського виробництва, своєчасність виплати заробітної плати та пенсій тощо). Зрозуміло, такі дані складно аналізувати вручну через великий обсяг інформації і складних нелінійних причинно-наслідкових залежностей. Тому і виникла необхідність в розробці нових методів аналізу і кластеризації, до яких відносяться машинні методи виявлення закономірностей.

Подальшою розбудовою інтелектуального аналізу даних є динамічний аналіз потоків даних, який має на увазі, що дані надходять послідовно в online режимі, а обсяг вибірки заздалегідь не відомий. В реальних задачах кластери можуть перекриватися, а кожне спостереження може належати кільком кластерам з певною ймовірністю. Для вирішення таких завдань ефективно застосовувати методи обчислювального інтелекту, і, перш за все, такі підходи, які пов’язані зі штучними нейронними мережами та нечіткими методами аналізу даних. Використання нейронних мереж дозволить визначати параметри кластерів в online режимі, а нечіткий підхід дозволить вирішити задачу кластеризації в умовах, коли кластери перекриваються.

Кінець XX століття характеризується помітним сплеском досліджень в області штучних нейронних мереж завдяки тому, що, з одного боку, у другій половині 1980-их років був відкритий алгоритм зворотного поширення похибки, внаслідок чого вдалося подолати критичні зауваженняМінського і Пайперта, а з іншого – через те, що з року у рік

справджувався закон Муру, дозволяючи персональним комп'ютерам проводити дедалі складніші обчислення. У 1990-ті роки теорія штучних нейронних мереж стрімко розвивається, а отримані результати успішно застосовуються для вирішення широкого кола завдань ідентифікації, прогнозування, управління, кластерування та класифікування. Однак, в той же час, стають чітко зрозумілими недоліки традиційних нейромережевих архітектур: велика обчислювальна складність, абсолютна неінтерпретовність результатів, емпіричний характер вибору архітектури мережі для вирішення будь-якої задачі. У зв'язку з цим застосування нейронних мереж у певному ряді випадків є недоцільним.

З середини 1990-их років у світі активно проводяться дослідження з розробки методів, що дозволяють подолати зазначені недоліки. Останнім часом все більшої популярності набувають так звані гібридні нейронні мережі, що об'єднують в собі переваги різних архітектур в єдину нейромережеву систему. Під гібридними нейронними мережами розуміють штучні нейронні мережі з можливістю у той чи інший спосіб отримувати знання про те, за якими правилами проводиться генерація вихідного сигналу. Традиційні гібридні нейромережеві системи є потужним інструментом для вирішення проблеми неінтерпретовності результатів, однак, слід зазначити, що вони не здатні працювати у послідовному режимі опрацювання даних, а крім того часто є адаптивними лише з тієї точки зору, що можуть налаштовувати свої синаптичні вагові коефіцієнти в процесі навчання, не маючи при цьому механізмів структурної адаптації.

На даний час задачі Data Mining широко зустрічаються в багатьох областях: медицині, економіці та електроенергетиці і т.д. На сьогоднішній день розроблено потужний математичний апарат для вирішення цих задач. Разом з тим, на перший план виходять задачі, пов'язані з Dynamic Data Mining, Data Stream Mining і Big Data, тобто задачі, де дані надходять не в формі пакета, а в формі потоку інформації.

1 ОГЛЯД СТАНУ ПРОБЛЕМИ І ГАЛУЗІ ДОСЛІДЖЕННЯ

Здатність навчатися є основною властивістю біологічного мозку, а штучна нейронна мережа в деякому сенсі моделює мозок, поняття «навчання» посідає щонайперше місце в теорії штучних нейронних мереж. Математичні проблеми, що пов’язані з навчанням, вивчають у напрямі загальної теорії штучних нейронних мереж, який дістав назву «нейроматематика». Із точки зору нейроматематики, навчання тлумачать як завдання адаптувати параметри, а можливо, й архітектуру мережі, щоби, оптимізуючи прийнятий критерій якості, розв’язати поставлену задачу. Таке визначення є узвичаєним та неявно припускає, що нейроматематика ґрунтується на методах оптимізації та ідентифікації.

Зазвичай припускають, що навчання має перманентний характер та з часом мережа покращує свої характеристики, постійно «наближаючись» до оптимального розв’язку поставленої задачі. Тип та характер навчання обумовлені, насамперед, обсягом попередньої та поточної інформації про довкілля, в яке «занурили» мережу, а також критерієм якості (цільовою функцією), що характеризує рівень відповідності нейронної мережі до розв’язуваної нею задачі. Інформацію про довкілля здебільшого задають у вигляді навчальної вибірки образів або зразків, оброблюючи їх мережа дістає відомості, необхідні для отримання шуканого розв’язку. Саме характер та обсяг цієї інформації визначають тип і методнавчання.

З погляду математики, навчання нейронних мереж – це багато-параметрична задача нелінійної оптимізації. Більшість методів навчання можна розділити на два класи: навчання з учителем (із заохоченням) та навчання без учителя (без заохочення, або самонавчання).

Методи навчання з учителем застосовують у випадках, коли відома бажана реакція системи в кожну мить часу, себто відомий навчальний сигнал, який впливає на налаштування параметрів системи, що навчається. Рівень «навченості» системи формально визначають за значенням цільової

функції, тобто за тим станом, якого має набути системав результаті навчання.

Як відомо, на сьогоднішній день розроблено безліч алгоритмів і методів кластеризації, при цьому більшість з них є по своїй суті чіткими процедурами, де передбачається, що кластери лінійно-роздільні і мають опуклу форму. У практичних задачах досить часто виникає ситуація, коли кластери можуть перекриватися і мати довільну форму, в таких випадках можуть бути використані алгоритми нечіткої кластеризації в їх рекурентній формі. Разом з тим, досить часто виникає задача, коли кластери не тільки перетинаються, але і мають довільну форм.

Дана проблема може бути вирішена на основі ядерного підходу, пов'язаного з гіпотезою, сформульованою Кавером [2], [3], яка говорить про те, що якщо задача лінійно нероздільна у вихідному просторі, то вона може бути вирішена в просторі підвищеної розмірності.

Таким чином, на сьогоднішній день актуальною є задача розробки нового методу ядерної кластеризації, призначеного для обробки даних в online режимі, коли дані надходять на обробку послідовно, одне за одним, а кластери можуть перекриватися і мати довільну форму.

Метою даного дослідження є розробка методу кластеризації на основі ядерної нейронної мережі, яка налаштовує свою архітектуру в процесі навчання-самонавчання, в умовах кластерів довільної форми, що перетинаються. Досягнення поставленої мети здійснюється шляхом вирішення наступних основних завдань:

– аналіз існуючих методів і підходів для кластеризації даних різної фізичної природи;

– розробка ядерної кластерувальної нейронної мережі, яка є гібридом узагальненої регресійної мережі і самоорганізовної мапи Т. Кохонена, яка дозволяє опрацьовувати потоки даних в умовах невідомої кількості кластерів;

– вирішення за допомогою розробленої архітектури тестових задач різної фізичної природи.

Об'єкт дослідження – процес обробки даних, що надходять в online режимі за допомогою ядерної кластерувальної мережі в умовах кластерів довільної форми та невідомої їх кількості.

Предмет дослідження – методи кластеризації на основі ядерних функцій в задачах інтелектуального аналізу даних.

Методи дослідження – теорія обчислювального інтелекту і м'яких обчислень; теорія штучних нейронних мереж; теорія нечіткої логіки; теорія оптимізації і статистичний аналіз; імітаційне моделювання.

1.1 Основні принципи навчання штучних нейронних мереж

На сьогоднішній день штучні нейронні мережі (ШНМ) широко застосовуються при вирішенні задач інтелектуального аналізу даних таких, як розпізнавання образів, прогнозування, інтелектуальне управління і т.п., в умовах невизначеності, нелінійності, стохастичності, хаотичності, різного роду збурень і перешкод, завдяки своїм універсальним апроксимуючої властивостям і можливості навчатися за даними, що характеризують функціонування явища, яке досліджується.

Штучні нейронні мережі можуть вирішувати практичні задачі такі, як стиснення відеоінформації, розпізнавання рукописного тексту, обробка медичних зображень, аналіз опитувань, оцінка ризиків неповернення кредитів, безпеку транзакцій по пластикових картах – і це далеко не все.

Штучні нейронні мережі виникли на основі знань про функціонування нервової системи живих істот. Вони являють собою спробу використання процесів, що відбуваються в нервових системах, для вироблення нових технологічних рішень.

Відомо, що важлива властивість, якою володіє біологічний мозок, є здатність до навчання, а так як штучна нейронна мережа і є модель мозку,

поняття «навчання» є ключовим у теорії ШНМ. У загальній теорії штучних нейронних мереж існує напрям, який займається проблемами процесів навчання, яке отримало назву «нейроматематика» [4].

З точки зору нейроматематики процес навчання може розглядатися як адаптація параметрів, а можливо і архітектури мережі для вирішення поставленої задачі шляхом оптимізації прийнятого критерію якості. Дане формулювання є загальновідомим і передбачає, що в основі нейроматематики лежать методи оптимізації та ідентифікації.

Зазвичай вважається, що процес навчання має перманентний характер і з плином часу мережа покращує свої характеристики, поступово «наближаючись» до оптимального рішення поставленого завдання.

Тип і характер навчання визначаються насамперед обсягом апріорної і поточної інформації про середовище, в яку «занурена» нейромережа, а також критерієм навчання (цільовою функцією), що характеризує ступінь відповідності нейромережі до задачі, яка нею розв’язується. Інформація про зовнішнє середовище задана, як правило, у вигляді навчальної вибірки образів або прикладів, обробляючи яку, мережа витягує відомості, необхідні для отримання шуканого рішення. Саме характер і обсяг цієї інформації визначають як тип навчання, так і конкретний алгоритм.

Найбільш популярною в даний час є парадигма навчання «зі вчителем». Парадигму навчання «зі вчителем» схематично можна представити як на рисунку 1.1

В даній схемі «вчителю» відома інформація про зовнішнє середовище, яка задана у вигляді послідовності вхідних векторів x, а також «правильна реакція» на ці сигнали, позначена бажаним сигналом y.

Очевидно, що реакція ненавченої мережі $\hat{y}$ буде відрізнятися від «вірної» реакції вчителя, в зв'язку з чим, виникне помилка $e = y - \hat{y}$. Метою є настройка параметрів ШНМ так, щоб деяка скалярна функція від помилки $E(e)$ (критерій навчання) досягла мінімального значення. Так як

дані про зовнішнє середовище мають нестаціонарний характер, процес навчання триває безперервно, для чого і використовуються ті чи інші рекурентні процедури.

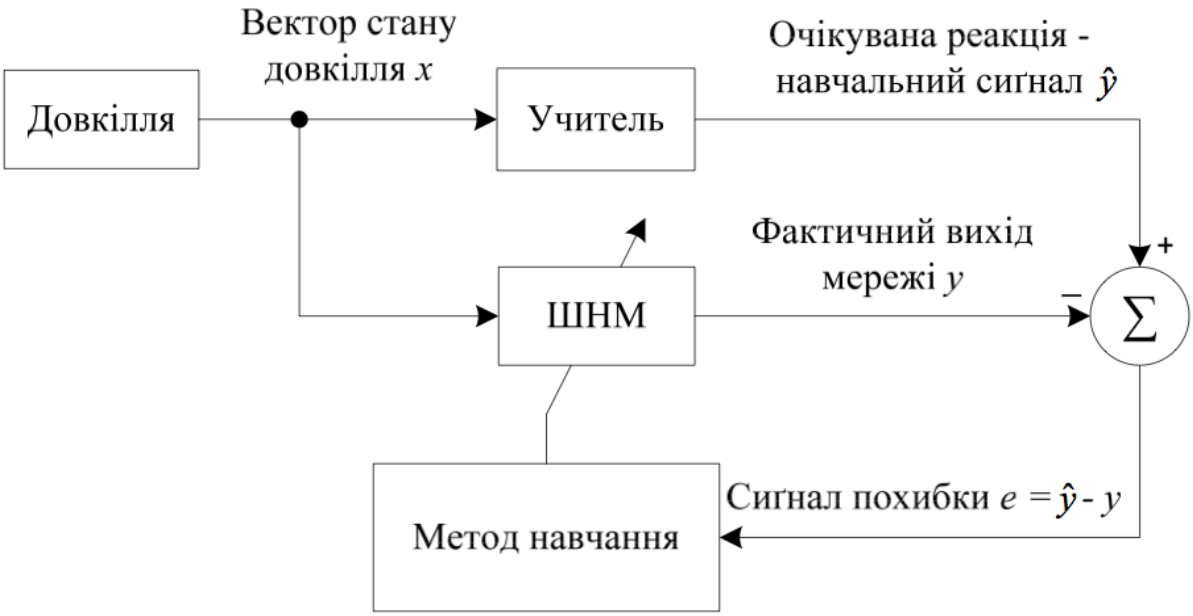

Рисунок 1.1 – Схема навчання зі вчителем

При навчанні «без вчителя» або самонавчання ми не можемо прогнозувати вихідні дані. Мережі, які реалізують парадигму самонавчання, призначені, для аналізу латентної структури вхідних даних і вирішують задачу автоматичної класифікації (кластеризації), компресії даних.

З вищеописаними парадигмами тісно пов'язані правила навчання, які лежать в основі конкретних алгоритмів. С. Хайкін [5] визначає п'ять основних правил: навчання на основі корекції помилок, навчання по Больцману, навчання за Хебом, навчання пам'яті та конкурентне навчання.

Своєрідним компромісом між двома парадигмами є навчання з підкріпленням [6], при якому доступна лише непряма інформація про

правильну реакцію на входовий сигнал мережі x. Досить відомою також є парадигма змішаного навчання, коли частина параметрів мережі налаштовується за допомогою навчання зі вчителем, а друга частина або архітектура в цілому – за допомогою самонавчання. Цей підхід набув широкого поширення в навчанні радіально-базисних нейронних мереж.

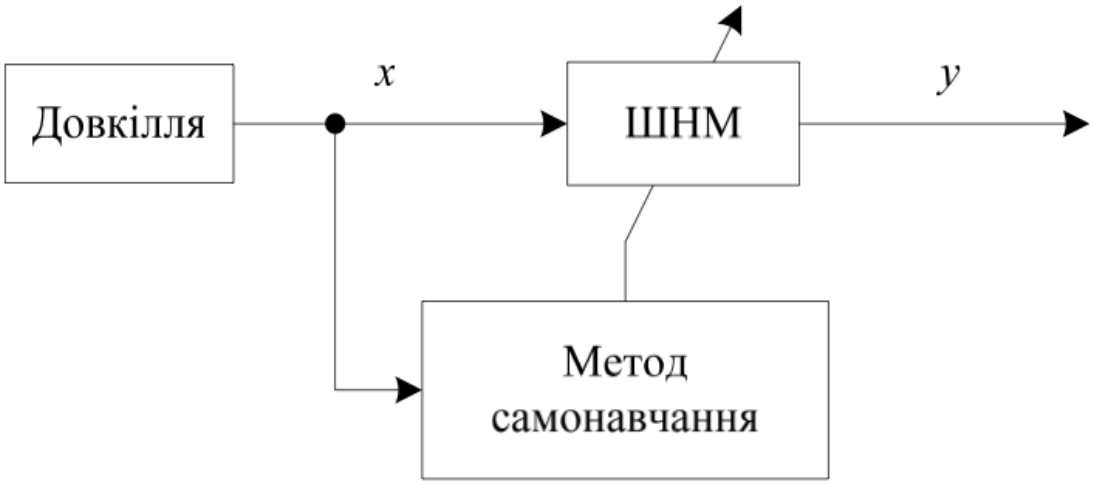

Рисунок 1.2 – Схема навчання без вчителя

Правило корекції помилки – типовий випадок навчання з вчителем, при цьому за допомогою тих чи інших процедур оптимізації та адаптивної ідентифікації мінімізується апріорі задана скалярная цільова функція $E(e)$. Найбільша кількість відомих алгоритмів ґрунтується саме на цьому правилі. В основі навчання по Больцману лежать принципи теоретичної термодинаміки, при цьому налаштування синаптичних ваг стохастичної мережі забезпечує необхідний (бажаний) розподіл ймовірностей станів окремих нейронів. У певному сенсі навчання по Больцману може розглядатися як розширення ідей навчання зі вчителем на стохастичний випадок. З самонавчанням тісно пов'язані правило Хебба і навчання, засноване на пам'яті, в основі яких лежить нейрофізіологічний постулат,

згідно з яким, коли нейрони з обох сторін синапсу знаходяться в збудженому стані, сила зв'язку між ними зростає (збільшується синаптична вага) і, навпаки, коли сусідні нейрони знаходяться в різних станах, зв'язок між ними слабшає.

Своєрідним компромісом між двома цими парадигмами є навчання з підкріпленням, при якому доступна лише непряма інформація щодо правильної реакції на вхідний сигнал *x.* Досить широке поширення набула також парадигма змішаного навчання, коли частина параметрів мережі налаштовується за допомогою навчання з учителем, а інша частина або архітектура в цілому – за допомогою самонавчання. Цей підхід набув найбільшого поширення у навчанні радіально-базисних штучних нейронних мереж.

Для поліпшення якості навчання і прискорення збіжності ітераційне навчання на підставі будь-якого з вищевказаних правил можна повторювати циклічно на, так званому, «вікні» – наборі послідовних значень навчального сигналу або проміжку часу. Одним з варіантів є пакетне навчання і навчання в послідовному режимі. Під навчанням в пакетному режимі мається на увазі випадок, коли вся вибірка відома, а навчання виконується «епохами». У послідовному режимі повторення зазвичай відсутні.

Сучасні обчислювальні технології дозволяють накопичувати і обробляти досить великі масиви спостережень, що надходять одне за одним, тому в даній роботі увага приділяється методам навчання з найбільшою швидкодією, які зможуть вирішити завдання кластерування в умовах коли масив даних не відомий заздалегідь.

1.2 Навчання нейронних мереж, засноване на оптимізації

За останні декілька років відчутно зріс інтерес до застосування нейронних мереж (НМ) в засобах технічного контролю та діагностики.

Нейронні мережі переважно використовуються в якості управляючого елемента в блоках розпізнавання стану технічних систем. Доведено, що ефективність застосування багато в чому залежить від обчислювальних можливостей НМ, які в свою чергу визначаються їх архітектурою. Однією із перспективних архітектур визнано мережу з радіальними базисними функціями (РБФ) [1;2]. Очевидно, що дана мережа має певні перспективи і в засобах діагностики параметрів захисту програмного забезпечення комп'ютерних систем. Визначення цих перспектив і є основною проблемою даної статті. Проблема безпосередньо пов’язана з таким важливим науково-практичним завданням, як забезпечення надійності функціонування розподілених комп’ютерних систем та мереж.

Загальними передумовами застосування мережі РБФ є: простота структури, яка зумовлює простоту програмної реалізації та висока швидкість навчання.

До загальних обмежень мережі РБФ належать: обмеженість обчислювальних можливостей у порівнянні з багатошаровим перспетроном, велика кількість емпіричних параметрів, що використовуються при навчанні схованого шару та погана екстраполяція результатів за межами області навчальних даних. Тому в навчальній виборці повинен бути представлений практично весь діапазон можливих вхідних даних.

Застосування РБФ доцільне в задачах захисту інформації за необхідності проведення швидкого оперативного аналізу даних з метою подальшого використання результатів в більш потужних системах. Наприклад, за допомогою лабораторного аналізу РБФ сигнатур комп’ютерних вірусів можна приблизно визначити характеристики багатошарового перспетрону, призначеного для використання в блоці розпізнавання антивірусних засобів.

1.3 Нейронні мережі, що миттєво навчаються

Ефективною альтернативою мережам, заснованим на оптимізації, є, так звані, нейро-моделі які навчаються миттєво, в їх основі лежить принцип «нейрони в точках даних» [9]. В рамках парадигми навчання штучних нейронних мереж подібні моделі також називають «лінивими» моделями навчання. Важливою характерною рисою миттєвих нейромереж вважається те, що етап формування фактичної моделі триває до того моменту, поки існує потреба в даній моделі. Іншими словами, це триває до тих пір, поки для цього входу необхідний виходовий сигнал, а до цього здійснюється тільки збір і зберігання даних.

Найчастіше, навчання класичних моделей відбувається в режимі offline і для оцінки застосовується виключно фіксована модель. Тобто, обробка всіх даних, необхідних для навчання, спочатку здійснюється в пакетному режимі. Подібна навчальна процедура відрізняється обчислювальною складністю і в окремих ситуаціях є практично неможливою. У зв'язку з цим використовують методи стиснення даних. Можна зробити ще один висновок на базі цього класичного підходу, для процедури навчання необхідна велика кількість обчислювальних етапів, в той час як оцінювання здійснюється дуже швидко або моментально.

Більш того, в якості додаткового варіанту може застосовуватися online адаптація, яка служить для налаштування і адаптації моделі за допомогою зміни її характеристик (параметрів) протягом фази оцінки. Необхідність в цьому обумовлена важливістю забезпечення слідкуючих властивостей.

Для нейромоделей, які навчаються миттєво, фаза навчання відбувається одночасно з фазою накопичення даних. З цього випливає, що розрахунок параметрів моделі здійснюється в період оціночної фази.

Нейронним мережам, які навчаються миттєво, притаманні локальні параметри, що описують входовий сигнал, зокрема, його поточний стан.

Отже, є можливість вибрати найбільш примітивну структурну моделі, наприклад, лінійну.

Необхідно відзначити, що етап оптимізації моделі і вибір даних проходить в індивідуальному порядку щодо кожного вхідного образу. В результаті, з'являється шанс модифікувати архітектуру моделі, складність її алгоритму та чинники, які необхідно враховувати для вибору даних, зберігає послідовність та враховуючи поточну ситуацію. Можна враховувати характерну якість і кількість даних, відповідні обмеження, стан і положення об'єкта.

Очевидно, миттєві моделі спочатку мають адаптивність. Отже, оновлені дані, що надходять на обробку в почерговому порядку, далі зважуються і зберігаються в базі даних, а попередні по закінченню часу забуваються.

Узагальнена регресійна нейронна мережа (УРНМ), яку запропонував Д. Шпехт [10], є найяскравішим представником моделей, які навчають на основі цього принципу. Вона базується на принципах ядерних оцінок Надарая-Ватсона [11], непараметричних моделей [12] і парзеновських вікон [13], а процес її навчання, в кінцевому рахунку, зводиться до одноразової установки багатовимірних радіально-базисних функцій (РБФ) в точках одиничного центрованого гіперкуба, які задаються в однозначному порядку за допомогою навчальної вибірки. Це говорить про те, що подібні мережі можна цілком віднести до тих самих нейронних мереж, що навчаються миттєво [14], [15], настройка яких проводиться за допомогою єдиного проходу навчального алгоритму.

За рахунок збігу в плані архітектури з нормалізованими радіально-базисними нейронними мережами (НРБНМ), навчання УРНМ відбувається в більш швидкому темпі, при одночасному встановленні центрів РБФ в точках, координати яких визначаються за допомогою вхідних сигналів об'єкта відповідно до принципу «нейрони в точках даних». При цьому «висота» РБФ повністю збігається з певними параметрами входового

сигналу об'єкта. Завдяки оперативній швидкості навчання УРНМ стало можливим їх ефективне застосування в реальних задачах.

Основним «недоліком» УРНМ є: необхідність великих обсягів пам'яті і використання складних обчислювальних дій, які потрібні для оцінки. Подібні труднощі можуть мати місце для online режиму оцінювання, до їх числа відносяться пропущені дані, час, необхідний для оцінки відгуку. Однак, в рамках підвищення можливостей в області обробки даних, моделі, що навчаються миттєво, можуть стати більш привабливою альтернативою.

1.4 Самоорганізовні нейронні мережі

Деякі методи навчання, які використовуються для традиційних архітектур штучних нейронних мереж, вимагають значення оптимальних вихідних сигналів мережі в навчальній вибірці інформації, тобто відносяться до класу навчання зі вчителем.

Але, іноді мають місце реальні задачі, для цілей яких в навчальній вибірці немає необхідних бажаних значень, а присутній виключно набір спостережень $x(k)$. Тому, для навчання мережі, слід витягувати важливі дані, які ґрунтуються виключно на даному наборі $x(k)$. У більшості випадків, процедури навчання без вчителя для ШНМ базуються на конкуренції між нейронами.

Охарактеризувати вищезгадані задачі найкращим чином, допоможуть наступні приклади:

– кластеризація – в разі, коли вхідні дані об’єднуються в кластери, їх необхідно виділити для подальшої обробки, як чіткої, так і нечіткої при істотному перетині кластерів;

– векторне квантування – дана задача має місце, коли входовий векторний простір потрібно відобразити в дискретному вигляді

оптимальним способом, за допомогою його розбиття на неперетинні підпростори;

– зниження розмірності – в ситуаціях, коли навчальні дані знаходяться в меншому по розмірності підпросторі, в порівнянні з входовою вибіркою. Для зменшення розмірності необхідно побудувати оптимальне відображення R^n в менший по розмірності простір з мінімальними втратами даних (найчастіше інформативними ознаками вважаються ознаки з найбільшою дисперсією даних);

– вибір ознак – в порівнянні з завданням зниження розмірності в даному випадку необхідно знайти кілька найбільш важливих ознак без їх перетворення.

Головна відмінність методу кластеризації від векторного квантування полягає в тому, що під час кластеризації визначаються області входового простору, які включають «схожі» спостереження [16], а в іншому випадку – основне завдання полягає в розбитті всього входового простору.

Конкурентне навчання. В даному випадку вивчаються процедури самонавчання, спрямовані на вирішення завдань кластеризації інформації в класичному розумінні [17].

Процедура самоорганізації заснована на методах конкурентного навчання, а її робота починається з ініціалізації синаптичних ваг мережі, які обираються або у випадковому порядку, або за допомогою будь-якого широко поширеного методу. Як правило, реалізація процедури самоорганізації включає три важливих етапи: конкуренція, кооперація і синаптична адаптація.

Стандартна архітектура конкурентної мережі (competitive neural network) являє собою повнозв'язний шар з m нейронів, які містять по n рецепторів, що характеризуються n–вимірними векторами синаптичних ваг

$$w_j(k)=\left(w_{j,1}(k),\ w_{j,2}(k),...,\ w_{j,n}(k)\right)^T,\ j=1,2,...,m. \qquad (1.1)$$

На вхід мережі подається входовий вектор-спостереження $x(k)$, $k=1,2,....$ На етапі конкуренції, під час подачі входового вектора, відбувається активація лише одного нейрона, який має назву «нейрон-переможець». У випадку з коректно навченою мережею для всіх векторів $x(k)$, які належать єдиному кластеру, активується один і той же нейрон-переможець $w^*(k)$.

В якості функцій активації нейронів конкурентної мережі, в більшості випадків, використовуються лінійні функції. Також, нейрони включають ваги w_j, які задають топологію карти, і, за винятком безпосереднього значення $y(k)$ виходу мережі, як результат класифікації може виступати номер (координати w^*) нейрона-переможця. Графічно архітектура конкурентної мережі зображена на рисунку 1.4.

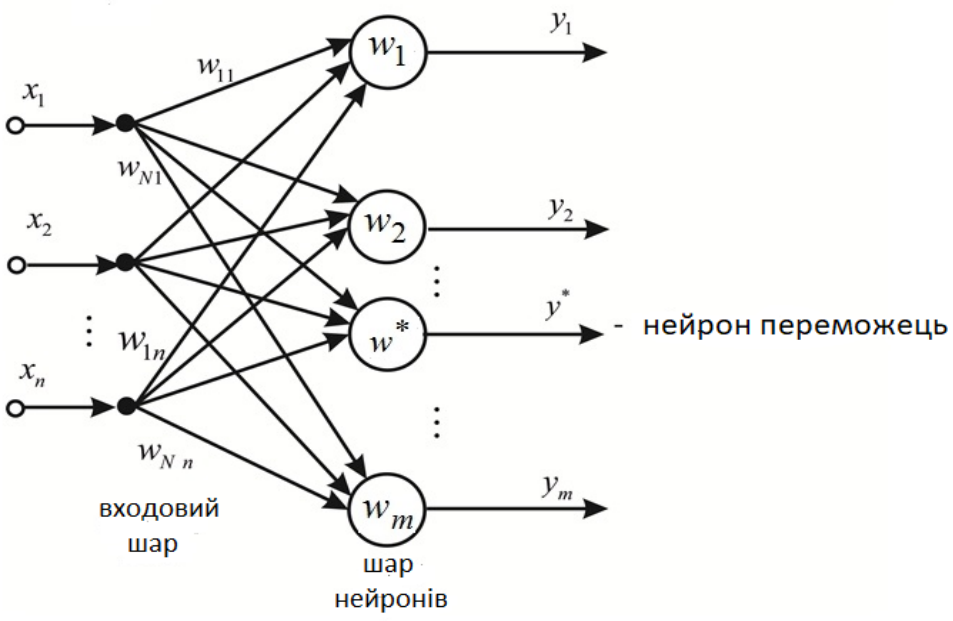

Рисунок 1.3 – Архітектура стандартної конкурентної нейронної мережі

На даний момент відомо два основні методи знаходження нейрона-переможця як «найближчого» до вектору поточного входового спостереження $x(k)$ відповідно до прийнятої метрики. Розглянемо кожен з них більш детально.

Здійснення вибору нейрона-переможця відповідно до відстанні в евклідовому просторі. Скалярний добуток векторів не відображає ступінь їх близькості у випадку з ненормованими даними і векторами синаптичних ваг. Таким чином, щоб вибрати переможця, доцільно застосовувати метрику Евкліда:

$$D_E(a,b) = \|a-b\| = \sqrt{(a-b)^T (a-b)}, \tag{1.2}$$

відповідно до якої

$$w^*(k) = w_O(k),\ k = \arg\max_{j=1,\ldots,m} \left\| w_j(k) - x(k) \right\|. \tag{1.3}$$

Налаштування синаптичних ваг. У загальновідомих випадках вибір «ближнього» для входового спостереження нейрона-переможця згідно з прийнятою метрикою записується так:

$$D\left(w^*(k), x(k)\right) = \min_j D\left(w_j(k), x(k)\right). \tag{1.4}$$

Наступним етапом після завершення фази вибору переможця є налаштування синаптичних ваг, згідно подальших процедур синаптичної адаптації.

Для нормованого простору ваг:

$$w^*(k+1) = w^*(k) + \eta(k)\left(x(k) - w^*(k)\right). \quad (1.5)$$

Для ненормованого простору ваг:

$$w^*(k+1) = \frac{w^*(k) + \eta(k)\left(x(k) - w^*(k)\right)}{\left\| w^*(k) + \eta(k)\left(x(k) - w^*(k)\right)\right\|}. \quad (1.6)$$

В даному випадку процедури налаштування ваг (1.1), (1.2) «підтягують» вектор синаптичних ваг нейрона-переможця до входового вектору $x(k)$ на відстань, що визначається величиною параметру кроку пошуку $\eta(k)$. У разі нормованого простору налаштування ваги виражається в його повороті в бік $x(k)$. Таким чином реалізується принцип навчання «переможець отримує все», хоча відомий і широко використаний другий принцип самонавчання «переможець отримує більше».

1.5 Гібридні системи обчислювального

Алгоритми та методи навчання, які базуються на екземплярах (їх також називають на основі пам’яті, на основі вибору, або «ліниві» методи) вже багато років використовуються для вирішення завдання класифікації. Зазвичай ці підходи діють, зберігаючи кожен навчальний екземпляр у певній формі, а потім порівнюючи нові екземпляри з тими, що зберігаються. У деяких системах (наприклад, радіально-базисні функціональні мережі) навчальні дані попередньо обробляються непідтримуваною системою навчання для досягнення певного рівня стиснення перед збереженням екземплярів.

Однак система без вчителя створює кластери, групуючи подібні навчальні екземпляри разом, але ця міра подібності не враховує

контрольовану класифікацію кожного примірника. В інших системах керована кластеризація може використовуватися для досягнення стиснення, зберігаючи при цьому врахування завдання класифікації. Однак окремі кластери, які створюються, можуть бути не оптимальними з точки зору їх використання для класифікації.

На додаток до питань масштабованості, наприклад, описаних вище класифікаторів, заснованих на парадигмі навчання зі вчітелем, всі обчислювальні підходи зазнають так званого «прокльону розмірності». Це стосується різкого підвищення кількості активаційних функцій і складності класифікації. Зі збільшенням розмірів навчальної вибірки збільшується і складність розробкі сістеми для класифікації.

Основна проблема збільшення розмірності полягає в тому, що функції, визначені у просторі більш високого розміру, як правило, набагато складніші, ніж функції у вхідного простору ознак. Для цих складних функцій для точного моделювання основної функції необхідно використовувати щільніші зразки точок даних. Однак такі щільні зразки набагато складніше знайти, оскільки розмірність збільшується [5].

Для того щоб уникнути цих проблем в розробці систем для кластерування доцільним є використання нейронних мереж які навчаються на основі принципу «нейронів в точках даних».

Цей підхід дозволяє в деякій мірі поліпшити систему та обійти проблему «прокльону розмірності». Але, якщо розглядати суто класичні нейромережі на основі лінивого навчання все ж таки з часом опрацювання даних та накопиченням їх у системі, побудована модель також буде схильна до «прокльону розмірності».

Отже, на даний час, актуальною є ідея використання нейронних мереж які використовують для навчання принцип «нейронів в точках даних», але в поєднанні з самонавчанними системами, які будуть контролювати кількість активаційних функцій в мережі.

В магістерській роботі пропонується гібридна мережа на основі самоорганізовної карти та узагальненої регресійної нейронної мережі для вирішення завдання кластерування потоків даних.

1.6 Аналіз предметної галузі

В даний час системи і методи обчислювального інтелекту набули широкого розповсюдження для вирішення різного кола задач, що виникають в рамцях інтелектуального аналізу даних (Data Mining) таких, як прогнозування, класифікація, асоціація і кластеризація і т.п. При цьому кластеризація в цьому ряду займає особливе місце, оскільки рішення цієї задачі ґрунтується на парадигмі самонавчання (навчання без вчителя), що істотно ускладнює процес пошуку рішення. На сьогоднішній день існує велика кількість методів, що вирішують завдання кластеризації, всі вони можуть відрізнятися як структурою, формою так і апріорними умовами.

Метою цієї роботи є створення гібридної системи обчислювального інтелекту, яка призначена для кластеризації даних в послідовному режимі.

Найбільш відомими системами для потокової кластеризації є самоорганізовні мапи Т. Кохонена. Як відомо ці нейросистеми, за великим рахунком, тісно пов'язані з методом К-середніх, який, по своїй суті, є пакетним методом кластеризації Т. Кохонена.

Однак, мапа Т. Кохонена, як і всі традиційні методи, засновані на центроідах, мають одне істотне обмеження: передбачається, що дані класу є лінійно-розділимими і мають опуклу форму, між якими може бути побудована в процесі самонавчання розділяюча лінійна гіперплощина. Оскільки в реальних ситуаціях такі умови практично ніколи не виконуються, є доцільним розробка такої архітектури нейромережі для кластеризації даних, в яких кластери мали б довільну форму, а розділяючі їх поверхні не були гіперплощинами, а мали як завгодно складну форму.

Зараз в рамках обчислювального інтелекту, існують такі системи кластеризації, як ядерні самоорганізовні карти Т. Кохонена, які є, по суті, машинами опорних векторів, свого часу запропонованими Вапніком і заснованими на використанні, так званих, ядер Мерсера.

З математичної точки зору це досить потужний апарат, але, справа в тому, що в нейронній мережі опорних векторів кількість ядерних функцій в першому прихованому шарі дорівнює кількості спостережень і призводить до такого явища, як «прокляття розмірності», тому використання цих нейромереж в рамках потокової обробки великих масивів даних недоречно. Тому в даній роботі пропонується замість традиційного підходу, підхід, пов'язаний з оцінками Парзена, які близькі до оцінок Надарая-Ватсона, але в основі яких, лежить гіпотеза, сформульована Кавером, що говорить про те, що задача лінійно-нероздільна в входовому просторі, може бути вирішена в просторі підвищеної розмірності. Тобто, суть даного підходу полягає в тому, що спочатку ми повинні за допомогою тих чи інших ядерних функцій, підняти розмірність, далі в просторі підвищеної розмірності вирішити задачу кластеризації за допомогою самоорганізовних мап, де кількість входів перевищує розмірність початкового простору.

Проведені в рамцях магістерської роботи дослідження показують, що еволюційний підхід до скорочення кількості нейронів в прихованому шарі є ефективним, особливо в системах опрацювання великих даних які надходять послідовно в онлайн режимі. Еволюційних підхід також дозволяє навчати не тільки параметри мережі (її синаптичні ваги), а також і архітектури системі в цілому, контролюючи кількість нейронів у прихованому шарі.

2 КЛАСТЕРНИЙ АНАЛІЗ І ПОСТАНОВКА ЗАДАЧІ КЛАСТЕРИЗАЦІЇ

На сьогоднішній день одним з основних завдань в області інтелектуального аналізу даних є завдання кластеризації, тобто завдання розбиття входових даних на однорідні в деякому сенсі групи в режимі навчання без вчителя (самонавчання) [18].

2.1 Задача кластеризації

Кластеризація – одна з найбільш важливих проблем неконтрольованого навчання. Як і кожна проблема такого типу, вона має справу з виявленням прихованої структури в сукупності даних. Об'єднання в кластери – це процес організації об'єктів в групи, члени яких подібні до певної міри. Кластер – це безліч об'єктів, близьких між собою в сенсі деякої міри подібності. У просторі змінних кластери являють собою скупчення точок (об'єктів) різної форми (рисунок 2.1).

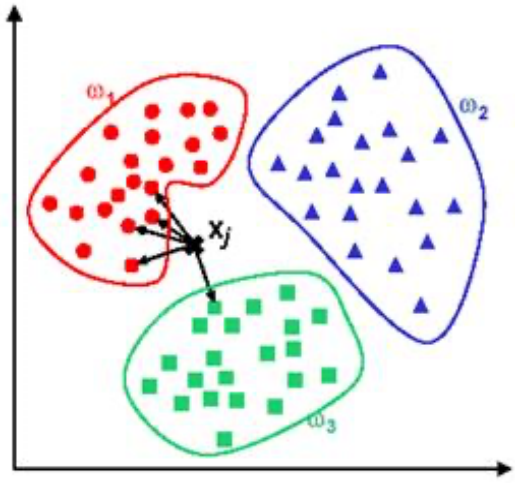

Рисунок 2.1 – Кластеризація лінійно-роздільних даних

У разі, представленому на рисунку 2.1, мірою подібності є відстань: два або більше об'єктів належать до одного кластеру, якщо вони «близькі» відповідно до заданого відстанню. Це кластеризація на підставі відстані.

Інший тип кластеризації – концептуальна кластеризація: два або більше об'єктів належать до одного кластеру, якщо вони визначають поняття, загальне для всіх об'єктів. Іншими словами, об'єкти групуються так, щоб відповідати певному концепту.

Кластерний аналіз займає одне з центральних місць серед методів аналізу даних і являє собою сукупність підходів, методів і алгоритмів, призначених для знаходження деякого розбиття досліджуваної сукупності об'єктів на підмножини щодо подібних, схожих між собою об'єктів [19]. При цьому вихідним допущенням для виділення таких підмножин, які отримали спеціальну назву «кластер», які так само іноді називають «таксонами» або просто класами, служить лише неформальне припущення про те, що об'єкти, що відносяться до одного кластеру, повинні мати більшу схожість між собою, ніж з об'єктами з інших кластерів [20].

2.2 Класифікація алгоритмів кластеризації

Алгоритми кластеризації можуть бути класифіковані в такий спосіб:

– чітка / нечітка кластеризація – до цієї групи методів кластеризації можна віднести методи засновані на центроїдах та методи кластеризації великих баз даних;

– ієрархічна / неієрархічна кластеризація – тут можна виділити методи засновані на щільності та так звані «решітчасті» методи кластеризації;

– імовірнісна кластеризація – на даний час розширенням цієї групи методів кластеризації є можливістні методи.

У разі чіткої кластеризації дані групуються особливим чином так, що якщо якесь спостереження належить до певного кластеру, то воно не може

належати до іншого кластеру. Навпаки, другий тип – нечітка (фаззі, fuzzy) кластеризація – використовує нечіткі набори для кластеризації даних таким чином, що кожна точка може належати до двох або більше кластерів з різним ступенем приналежності. Результатом фаззі-кластеризації є побудова матриці нечіткого розбиття. Ієрархічні алгоритми застосовуються, коли необхідно вирішити задачу таксономії. В ході роботи алгоритму великі кластери розбиваються на більш дрібні (або дрібні кластери об'єднуються у великі). Результатом таксономії є деревоподібна ієрархічна структура.

2.3 Алгоритми засновані на центроїдах

Одним з найбільш поширених алгоритмів кластеризації з групи методів кластеризації заснованих на центроїдах є алгоритм k-середніх (k-means), також званий швидким кластерним аналізом. Повний опис алгоритму можна знайти в роботі Хартігана і Вонга [21]. На відміну від ієрархічних методів, які не вимагають попередніх припущень щодо числа кластерів, для можливості використання цього методу необхідно мати інформацію про найбільш ймовірну кількість кластерів.

Алгоритм К-середніх будує k кластерів, розташованих на можливо великих відстанях один від одного. Основний тип задач, які вирішує алгоритм К-середніх – наявність припущень (гіпотез) щодо числа кластерів, при цьому вони повинні бути різні настільки, наскільки це можливо. вибір числа k може базуватися на результатах попередніх досліджень, теоретичних міркуваннях або інтуїції.

Загальна ідея алгоритму: набір спостережень зіставляється k кластерам так, що середнє значення об'єктів в кластері (для всіх спостережень) максимально можливо відрізняються один від одного для кожного з кластерів.

Перевагами алгоритму К-середніх є: простота використання; швидкість використання; зрозумілість і прозорість алгоритму.

Недоліки алгоритму К-середніх:

– алгоритм занадто чутливий до викидів, які можуть спотворювати середнє. Можливим вирішенням цієї проблеми є використання модифікації алгоритму – алгоритм К-медіан (k-medians);

– алгоритм може повільно працювати на великих базах даних. Можливим вирішенням цієї проблеми є використання частини вибірки даних.

2.4 Застосування нейронних мереж в задачі кластеризації

Методи кластеризації за допомогою нейронних мереж є розвитком класичних методів кластеризації. Метод кластеризації за допомогою мережі Т. Кохонена містить в своїй основі розглянутий вище метод К-середніх.

Нейронні мережі Т. Кохонена або самоорганізовні мапи Т. Кохонена (Kohonen's Self-Organizing Maps) призначені для вирішення задач класифікації без вчителя [22]. Це двошарова нейронна мережа, яка містить вхідний шар (шар вхідних нейронів) і шар Кохонена (шар активних нейронів). Шар Кохонена може бути одновимірним, двовимірним або тривимірним. У першому випадку активні нейрони розташовані в ланцюжок. У другому випадку вони утворюють двовимірну сітку (зазвичай у формі квадрата або прямокутника), а в третьому випадку вони утворюють тривимірну конструкцію.

2.5 Метрики кластерного аналізу

Подібність або відмінність між об'єктами кластеризації встановлюється в залежності від обраної метричної відстані між ними.

Якщо об'єкт описується n властивостями (ознаками), то він може бути представлений як точка в $n-$вимірному просторі, і схожість з іншими спостереженнями буде визначатися як відповідне відстань. Мірою подібності називається будь-яка функція, яка може визначити значення подібності або відстань між двома об'єктами в даному просторі. Подібність між двома об'єктами зворотно пропорційна відстані між ними. Метрика – функція, що визначає відстань в метричному просторі. В даному просторі визначено відстань між будь-якою парою елементів.

У таблиці 2.1 наведені основні метрики, які використовуються для вирішення задачі кластеризації.

Таблиця 2.1 – Метрики, що використовуються в кластерному аналізі

Найменування метрики	Тип ознак	Формула оцінки міри близькості
Евклідова відстань	кількісні	$d_{ik}=\left(\sum_{j=1}^{N}\left(x_{ij}-x_{kj}\right)^2\right)^{\frac{1}{2}}$
Манхеттенська метрика	кількісні	$d_{ik}=\sum_{j=1}^{N}\left\|x_{ij}-x_{kj}\right\|$
відстань Махаланобіса	кількісні	$d_{ik}^{M}=\left(x_{ij}-x_{kj}\right)^{T}W^{-1}\left(x_{ij}-x_{kj}\right)$, де $W-$ коваріаційна матриця вибірки $X=\{X_1,X_2,...,X_n\}$

2.6 Ядерні функції активації та їх види

У низькорозмірних просторах лінійні моделі накладають вельми жорсткі обмеження, оскільки лінії і гиперплоскости мають обмежену гнучкість. Один із способів зробити лінійну модель більш гнучкою – додати нові ознаки, наприклад, додати взаємодії або поліноми вхідних ознак. Однак часто ми не знаємо, які ознаки необхідно додати, і додавання більшої кількості ознак (наприклад, розгляд всіх можливих взаємодій в

100-вимірному просторі ознак) може дуже сильно збільшити вартість обчислень.

На щастя, є хитрий математичний трюк, який дозволяє нам навчити класифікатор в багатовимірному просторі, практично не вдаючись до обчислення нового, можливо, дуже багатовимірного простору. Цей трюк відомий під назвою «ядерний трюк» (kernel trick) і він безпосередньо обчислює евклідові відстані (точніше, скалярні твори точок даних), щоб отримати розширену простір ознак без фактичного додавання нових ознак.

У 1992 році в роботі Бернарда Бозера, Изабелл Гийон і Владимира Вапника був запропонований спосіб адаптації машини опорних векторів для нелінійного розділення класів [5]. Існують декілька засобів помістити дані у високорозмірний простір, які найчастіше використовуються машиною опорних векторів: лінійне ядро, сигмоїдальне ядро, та найпопулярніші серед усіх поліноміальне ядро та ядро гауса (RBF).

Лінійне ядро – це найпростіша функція ядра. Він задається внутрішнім твором (x_i, x_j) плюс необов'язкова константа c. Алгоритми ядра, що використовують лінійне ядро, часто еквівалентні своїм неядерних аналогам.

$$K(x_i, x_j) = {x_i}^T x_j \tag{2.1}$$

Лінійне ядро використовується, коли дані є лінійно розділяються, тобто їх можна розділити за допомогою одного рядка. Це одне з найбільш часто використовуваних ядер. Він в основному використовується, коли в конкретному наборі даних є велика кількість функцій.

Сигмоїдальне ядро відбувається з області нейронних мереж, де біполярна сигмовидная функція часто використовується в якості функції активації для штучних нейронів. Це ядро було досить популярно для

опорних векторних машин через свого походження з теорії нейронних мереж.

Цікаво відзначити, що модель SVM, що використовує функцію сигмоїдального ядра, еквівалентна дворівневої нейронної мережі персептрона. Сигмоїдальна функція повертає два значення, 0 і 1, тому вона більше підходить для задач двійковій класифікації.

$$K(x_i, x_j) = tanh(\gamma_0 {x_i}^T x_j + \gamma_1), \quad (2.2)$$

де γ_0 та γ_1 – позитивні параметри

Поліноміальне ядро обчислює усі можливі поліноміальні комбінації початкових ознак до певної міри. Додавання поліноміальних ознак просто в реалізації і може чудово працювати з усіма видами алгоритмів МН (не тільки з методами SVM), але при низькій поліноміального ступеня воно не здатне справлятися з дуже складними наборами даних, а при високій поліноміального ступеня воно створює величезну кількість ознак, роблячи модель вкрай повільної. Поліноміальний ядро "дивиться" не тільки на задані властивості вхідних даних, щоб визначити їх схожість, а й на їх комбінації.

У поліноміальному ядрі ми просто обчислюємо скалярний твір, збільшуючи потужність ядра.

$$K(x_i, x_j) = \left(c + {x_i}^T x_j\right)^P \quad (2.3)$$

де c – вільний параметр, коли c = 0, ядро називається однорідним;

xi, xj – вектори у вхідному просторі;

p – ступінь полінома.

Ядро RBF (радіальна базисна функція), також відоме як ядро гауса. Ядро гауса відповідає нескінченному простору ознак. Пояснити ядро гауса

можна так: воно розглядає усі можливі поліноми усіх мір, проте важливість ознак знижується із зростанням міри.

Щоб отримати прогноз для нової точки, вимірюється відстань до кожного опорного вектору. Класифікаційне рішення приймається, виходячи з відстаней до опорних векторів, а також важливості опорних векторів, отриманих в процесі навчання.

Відстань між точками даних вимірюється за допомогою ядра гауса, яка зображена формулою 2.4.

$$K(x_i, x_j) = exp\left(-\gamma \left\| x_i - x_j \right\|^2\right) = -exp\left(-\frac{\left\| x - x' \right\|^2}{2\sigma^2}\right) \qquad (2.4)$$

де x_i, x_j– точки даних;

$\left\| x_i - x_j \right\|$ – відстань Евкліда;

γ – параметр, який регулює ширину ядра Гауса.

2.7 Постановка задачі дослідження

Оскільки, як зазначалося раніше, сучасні обчислювальні технології дозволяють накопичувати і обробляти досить великі масиви інформації, то на перший план виходить швидкість обробляння даних, а також можливість роботи з ними в послідовному режимі. Крім того варто зазначити, що інформація, яка обробляється, може характеризуватися нелінійним і нестаціонарним характером даних.

У таких випадках доцільно використання штучних нейронних мереж, які володіють універсальними апроксимуючими властивостями. Застосування апарату нечіткої логіки дозволяє розширити функціональні можливості штучних нейронних мереж і коло вирішуваних завдань.

Як показав проведений аналіз, на сьогоднішній день в рамцях інтелектуального аналізу даних існує велика кількість методів

кластеризації, що відрізняються між собою як математичними постановками задач так і результатами. При цьому більшість найбільш відомих методів передбачають, що кластери мають опуклу форму і лінійно-роздільні, але, на жаль, в реальних задачах – це далеко не так, оскільки кластери можуть мати довільну форму.

Для вирішення таких завдань існує ряд методів, одним з найбільш популярних підходів при цьому є методи ядерної кластеризації [23], [24]. В рамцях цього підходу передбачається, що кластери можуть мати довільну форму, але вони не перетинаються і не перекриваються, а вся вибірка, що підлягає кластеризації задана апріорі.

На сьогоднішній день існує безліч методів кластеризації, але всі вони допускають, що кластери лінійно-роздільні, а дані надходять в пакетному режимі. В реальних задачах кластери можуть перекриватися. Щоб подолати проблеми масштабованості, пов'язані з навчаннязі вчителем, доцільним є використання еволюційних підходів для зменшення як кількості навчальних екземплярів, так і кількості функцій, що використовуються в таких системах.

Основна ідея цієї методики – створити дві бінарні маски, одну для навчального набору та одну для набору функцій. Кожен елемент маски являє собою включення або виключення. Потім алгоритм еволюційного обчислення використовується для пошуку оптимальних компонентів маски, які одночасно мінімізують помилку тренувань, кількість навчальних екземплярів та кількість функцій. Очевидно, що ці три критерії оптимізації пов'язані і, таким чином, вимагають компромісів. Наприклад, зменшення кількості навчальних екземплярів може призвести до збільшення невірно відкласифікованих прікладів.

Таким чином, на даний час стоїть актуальне завдання по розробці online методу кластеризації на основі ядерних функцій, коли кластери можуть перекриватися, а дані на входи системи можут надходити послідовно, одне за одним. Використання спеціальних ядерних функцій

дозволить підвищити аппроксимуючі властивості запропонованої гібридної нейронної мережі і методу кластеризації.

Для досягнення зазначеної задачі необхідно: проаналізувати існуючі методи і підходи для кластеризації даних різної фізичної природи; розробити архітектури гібридної нейронної мережі на основі узагальненої регресійної мережі і самоорганізовної мережі Т. Кохонена; провести імітаційне моделювання розробленого методу і вирішити з його допомогою ряд тестових завдань потокової кластеризації даних.

3 ЯДЕРНА ПОСЛІДОВНА КЛАСТЕРИЗАЦІЯ ПОТОКІВ ДАНИХ НА ОСНОВІ ЕВОЛЮЦІЙНОЇ УЗАГАЛЬНЕНОЇ РЕГРЕСІЙНОЇ НЕЙРОННОЇ МЕРЕЖІ

На даний час штучні нейронні мережі [25], [26] набули широкого поширення для вирішення різного кола задач, що виникають в рамцях інтелектуального аналізу даних (Data Mining) таких, як прогнозування, класифікація, кластеризація і т.п. При цьому кластеризація в цьому ряду займає особливе місце, оскільки рішення цієї задачі грунтується на парадигмі самонавчання [27], що істотно ускладнює процес пошуку рішення. Тут найбільш популярними є BSB– і ART-штучні нейронні мережі, призначені для обробки інформації в пакетному режимі, і самоорганізовні мапи Т. Кохонена (SOM) [28], призначені для вирішення задачі кластеризації великих масивів інформації, завдяки простоті обчислювальної реалізації та можливості послідовної online обробки даних.

3.1 Узагальнена регресійна штучна нейронна мережа

Ще одним класом ШНМ з використанням ядерних функцій активації, що дозволяє вирішувати задачу інтерполяції і нелінійного регресійного аналізу на основі непараметричного підходу, при цьому з меншими обчислювальними затратами, є узагальнені регресійні ШНМ (General Regression Neural Network – GRNN), запропоновані Д. Ф. Шпехтом [29], [30].

Дана архітектура подібна до архітектури радіально-базисної нейронної мережі, але відрізняється тільки тим, що в неї додатково вводяться блоки ділення, хоча цей блок можна ввести і до радіально-базисної нейронної мережі – тоді вона буде називатися нормалізована радіально-базисна нейронна мережа.

На рисунку 3.1 приведена схема узагальненої регресійної нейронної мережі з $n-$ входами і $h-$ виходами. Ця мережа подібно тришаровому персептрону містить три шари обробки інформації, проте, в якості активаційних функцій використовує радіально-базисні конструкції в першому прихованому шарі. Другий прихований шар містить $h+1$ нейронів, m з яких є адаптивними лінійними асоціаторами, а $(h+1)$-й – стандартний блок підсумовування $\sum^{*}$. Вихідний блок мережі утворено h блоками ділення.

$$\hat{y}_l^G = F_{li}(x) = \frac{\sum_{l=0}^{h} w_l^G \varphi_l^G(x)}{\sum_{l=0}^{h} \varphi_l^G(x)}, \quad l = 1,2,\ldots,\mathrm{h}. \tag{3.1}$$

Навчання узагальненої регресійної ШНМ є комбінованим процесом самоорганізації центрів в першому прихованому шарі, і навчання зі вчителем синаптичних ваг лінійних асоціаторів.

Суттєвою особливістю даної мережі є те, що число нейронів першого шару h жорстко не фіксується і може змінюватися в процесі навчання. Перший вхідний вектор навчальної вибірки $x(1)$ утворює центр першого нейрона першого шару $x(1) = c_1 = (c_{11}, c_{12}, \ldots, c_{1n})^T$. Наступний вхідний вектор $x(2)$ порівнюється з c_1 і якщо відстань між ними перевищує деякий заздалегідь заданий поріг, стає центром другого нейрона першого шару c_2. Ця процедура повторюється до вичерпання всієї навчальної вибірки, причому кожен новий вектор $x(k)$ порівнюється з усіма раніше сформованими центрами c_1, c_2, c_3,…. Таким чином формується перший прихований шар, званий також шаром образів, синаптичні ваги якого є по суті параметрами центрів радіально-базисних функцій. Зауважимо також, що параметри рецепторних полів даних мереж не настроюються, а

задаються апріорно, визначаючи в значній мірі узагальнюючі властивості цих нейронних мереж.

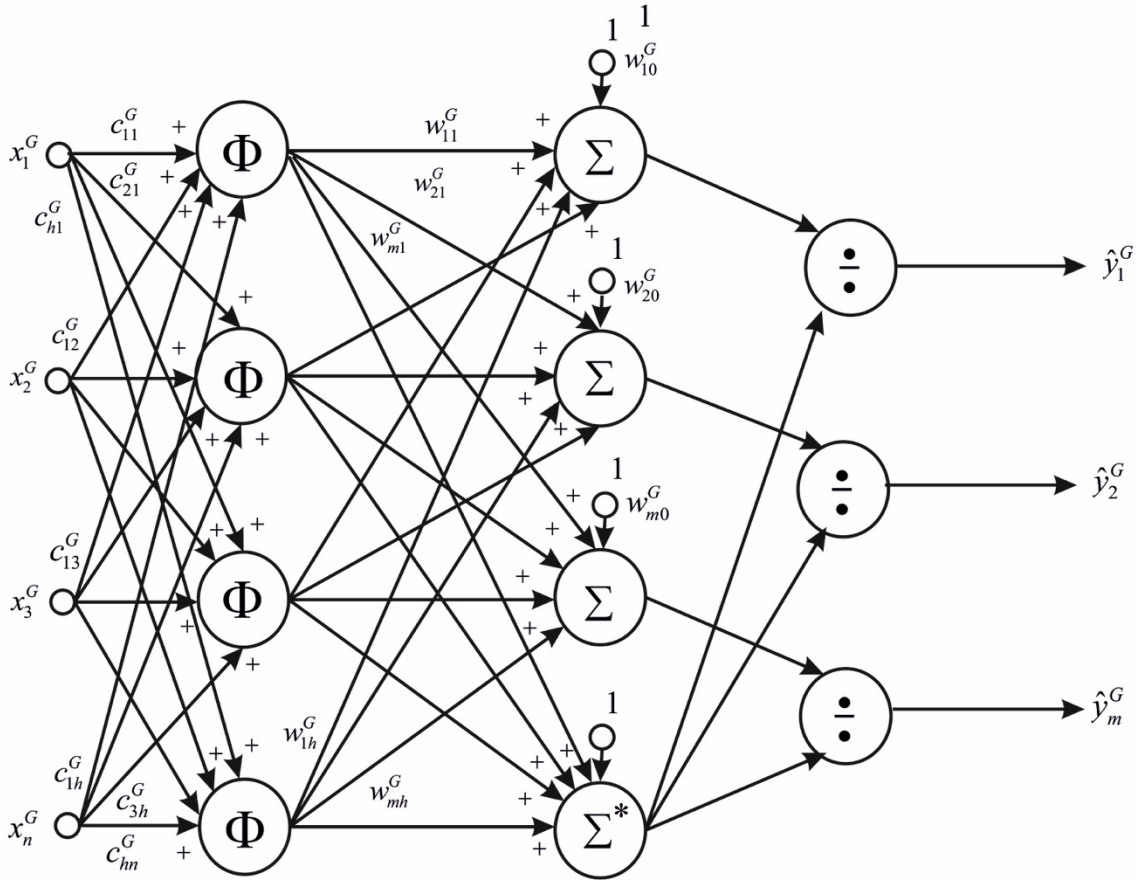

Рисунок 3.1 – Узагальнена регресійна ШНМ

Другий прихований шар, це шар підсумовування, налаштовується абсолютно аналогічно до процесу навчання радіально-базисних мереж. При цьому на виході m адаптивних лінійних асоціаторів формуються сигнали

$$o_l^{[2]}(k) = \sum_{i=0}^{h} w_{li}^G(k)\varphi_l^G(x(k)), \quad l = 1,2,\ldots,\mathrm{h}, \tag{3.2}$$

а на виході суматора $\sum^*$ з'являється сума $\sum_{i=0}^{h} \varphi_l^G(x(k))$.

Вихідний шар виробляє нормування вихідного вектору так, що

$$\hat{y}_l^G(k) = \frac{\sum_{i=0}^{h} w_{li}^G(k)\varphi_l^G(x(k))}{\sum_{i=0}^{h} \varphi_l^G(x(k))}, \quad \varphi_0^G(x(k)) \equiv 1, \tag{3.3}$$

виключаючи тим самим вплив змінного числа вузлів шару образів на кінцевий результат.

Реалізуючи ідеї нелінійної непараметричної регресії, узагальнені регресійні мережі є, мабуть, найбільш гнучкими і перспективними в класі функціонально зв’язаних ШНМ.

За рахунок того, що в основі цих мереж лежить ідея вікон Парзена і ядерних оцінок Надарая-Ватсона, GRNN здатна до практично миттєвого навчання. Але, якщо за основу брати гаусіан, то параметр ширини буде обраний внаслідок тільки емпіричних міркувань, тобто він буде абсолютно суб'єктивний. Замість гаусіанів логічніше буде використовувати інші, більш прості з обчислювальної точки зору ядерні конструкції. Позитивною стороною такого роду мереж є те, що, наприклад, в GRNN центри розміщуються в точках з координатами спостережень активаційних функцій, тоді функція є значенням відгуку її сигналу. В результаті інтерполяції функція являє собою зважену суму.

За великим рахунком, GRNN – це мережа, яка вирішує задачу інтерполяції, а не апроксимації. У цьому випадку буде відбуватися наступне – добуток активаційних функцій і значень, одержуваних на виході мережі, будуть прямо залежати від стану об'єкта, значення якого подаються на вхід системи. При цьому, якщо значення реального об'єкта забруднені випадковими збуреннями, то забезпечити високу якість інтерполяції неможливо.

Завдяки тому, що в основі GRNN лежить принцип навчання «нейрони в точках даних», ця мережа здатна навчатися практично миттєво,

не вимагаючи великої кількості даних в навчальній вибірці. У такій ситуації нейронні мережі, засновані на оптимізації, не встигатимуть навчатися.

3.2 Адаптивне навчання узагальненої регресійної нейронної мережі

Узагальнені регресійні нейронні мережі Д. Шпехта отримали досить широке поширення для вирішення завдань прогнозування та ідентифікації, завдяки своїм високим інтерполяційним властивостями, простоті і швидкості навчання за принципом «нейрони в точках даних». Ці мережі мають досить просту архітектуру, сформовану R-нейронами, двома блоками підсумовування і одним блоком розподілу, при цьому багатовимірний входовий сигнал $x(k)=(x_1(k),\ x_2(k),...,\ x_n(k))^T \in R^n$ (тут k –номер спостереження в вибірці тестових даних або поточний дискретний час) в шарі образів, сформованому R-нейронами з ядерними активаційними функціями (зазвичай гаусіанів $\varphi_l^G(x(k)=\varphi_l^G(k))$, $l=1,2,...,k$ переводиться в простір підвищеної розмірності, після чого за допомогою блоків підсумовування і ділення формується вихідний сигнал мережі $\hat{y}^G(k)$.

Оскільки навчання мережі відбувається миттєво шляхом встановлення центрів активаційних функцій в точках з координатами векторів навчальної вибірки, реакція мережі на довільний входовий сигнал може бути записана у вигляді

$$\hat{y}^G(x)=\frac{\sum\limits_{l=1}^{k} y(l)\varphi_l^G(x)}{\sum\limits_{l=1}^{k}\varphi_G^l(x)}=\frac{\sum\limits_{k} y(k)\exp\left(-\dfrac{\|x-x(k)\|^2}{2\sigma^2}\right)}{\sum\limits_{k}\exp\left(-\dfrac{\|x-x(k)\|^2}{2\sigma^2}\right)}=\frac{NG(k)}{DG(k)}, \qquad (3.4)$$

де $y(l)$, $l=1,2,...,k$ – зовнішній навчальний сигнал,

σ^2 – параметр рецепторного поля дзвонуватої функції активації.

Нескладно помітити, що навчання мережі відбувається в online режимі по мірі надходження спостережень навчальної вибірки $x(k)$, $y(k)$, при цьому кількість R-нейронів в мережі дорівнює k. При значній кількості даних в навчальній вибірці, мережа стає занадто громіздкою, що ускладнює її чисельну реалізацію.

Для скорочення кількості нейронів в мережі Д. Шпехтом запропоновано проводити попередню кластеризацію даних за допомогою методу k-середніх, а в процесі подальшого надходження даних на обробку проводити уточнення координат центроїдів сформованих кластерів за допомогою експоненціального згладжування.

Досить просто і ефективно управляти кількістю нейронів в online режимі можна за допомогою або ковзного вікна, сформованого з останніх s спостережень, при цьому замість (3.4) використовується оцінка

$$\hat{y}^G(x) = \frac{NG(k) + y(k+1)\exp\left(-\frac{\|x - x(k+1)\|^2}{2\sigma^2}\right) - y(k-s)\exp\left(-\frac{\|x - x(k-s)\|^2}{2\sigma^2}\right)}{DG(k) + \exp\left(-\frac{\|x - x(k+1)\|^2}{2\sigma^2}\right) - \exp\left(-\frac{\|x - x(k-s)\|^2}{2\sigma^2}\right)}, \tag{3.5}$$

або за допомогою правила самонавчання Т. Кохонена, також реалізованого на ковзному вікні.

Тоді, якщо мережа містить s нейронів, при надходженні, при надходженні $(k+1)$-го спостереження відбувається корекція одного з центрів активаційних функцій $x(1),...,x(l),...,x(s)$:

$$\begin{cases} x(l) = x^*(l) + \eta(k+1)\Big(x(k+1) - x^*(l)\Big), l = 1,2,...,s, \\ l = \arg\max \cos\Big(x(k+1), x^*(l)\Big), \Big\|x^*(l)\Big\| = \Big\|x(k+1)\Big\| = 1. \end{cases} \tag{3.6}$$

При цьому згідно з принципом «переможець отримує все», близький до $x(k+1)$ центр $x^*(l)$ «підтягується» до останнього спостереження на величину кроку $\eta(k+1)$. Цікаво також зауважити, що, якщо центри активаційних функцій розглядати як прототипи кластерів, правило (3.6) перетворюється в звичайний алгоритм навчання самоорганізується карти Т. Кохонена, яка в процесі своєї настройки збігається з оцінками-середніх:

$$w_l\left(k+1\right) = \begin{cases} w_l\left(k\right) + \eta\left(k+1\right)\Big(x\left(k+1\right) - w_l\left(k\right)\Big), \ \textit{якщо } w_l\left(k\right) = \ \textit{переможець}, \\ w_l\left(k\right) \ \textit{в протилежному випадку}, \end{cases} \tag{3.7}$$

де параметр кроку $\eta(k+1)$ обирається згідно до умов стохастичної апроксимації.

При цьому центри активаційних функцій мережі визначаються координатами центрів кластерів, які уточнюються в послідовному режимі на відміну від пакетної процедури кластеризації, використаної Д. Шпехтом.

Таким чином, використання процедур (3.5), (3.6), (3.7) дозволяє проводити навчання узагальненої регресійної мережі з обмеженою кількістю нейронів у прошарку образів в режимі послідовної обробки інформації без попередньої пакетної обробки вихідних даних навчальної вибірки.

3.3 Ядерна кластеризація на основі узагальненої регресійної нейронної мережі та самоорганізовної мапи Т. Кохонена

Як вже зазначалося вище, для вирішення завдань кластеризації в ситуаціях, коли класи мають довільну форму, можуть бути використані, так звані, ядерні самоорганізовні мапи (KSOM) [31], побудовані з використанням ядер Дж. Мерсера і засновані на мінімізації критерію емпіричного ризику [32], що лежить в основі, так званих, машин опорних векторів (SVM) [33], введених В.Н. Вапніком. Недоліком нейронної мережі опорних векторів, є те, що кількість нейронів такої мережі визначається обсягом вибірки даних, тому дана мережа не підходить для online обробки.

Архітектура ядерної самоорганізовної мапи на основі узагальненої регресійної нейронної мережі. Нижче на рисунку 3.2 приведена архітектура даної ядерної самоорганізовної мапи на основі узагальненої регресійної нейронної мережі. Вихідною інформацією для даної мережі є вибірка (можливо зростаюча) векторів спостережень $x(1), x(2), ..., x(k), ..., x(N), ...; \quad x(k) = (x_1(k), x_2(k) ..., x_i(k), ..., x_N(k))^T \in R^n$, яка повинна бути розбита на m кластерів довільної форми, при цьому k тут може бути як номером спостереження, так і моментом поточного часу.

Вектори спостережень $x(k)$ послідовно надходять на перший шар радіально-базисних функцій (R-нейронів), повністю співпадає за структурою з першим шаром (шаром образів) стандартної узагальненої регресійної мережі Д. Шпехта і сформований ядерними дзвонуватими функціями активації $\varphi_1, ..., \varphi_k, ..., \varphi_N$, за допомогою яких здійснюється підвищення розмірності входового простору.

В якості таких функцій зазвичай використовуються традиційні Гаусіани, а налаштування цього шару забезпечується за допомогою «лінивого» навчання на основі концепції «нейрони в точках даних» [34]. При цьому в якості центрів активаційних функцій приймаються самі

оброблювані вектори-образи. Таким чином, при подачі на вхід нейронної мережі деякого некласифікованого образу x, на виходах R-нейронів першого шару з'являються значення

$$\varphi_k(x) = e^{-\frac{\|x-x(k)\|^2}{2\sigma^2}}, k = 1,2,...,N \tag{3.4}$$

(тут σ^2 – параметр рецепторного поля дзвонуватої функції), а на виході GRNN в цілому – сигнал

$$\hat{y}(x) = \frac{\sum_{k=1}^{N} y(k)\varphi_k(x)}{\sum_{k=1}^{N} \varphi_k(x)}, \tag{3.5}$$

де $y(k)$ зовнішній навчальний сигнал, відповідний до образу $x(k)$. Зрозуміло, що в задачі кластеризації навчальний сигнал відсутній як такий, а сама GRNN в загальному випадку орієнтована на вирішення задачі інтерполяції, а не кластеризації.

Другий прихований шар розглянутої мережі – шар нормалізації реалізує елементарне перетворення

$$\tilde{\varphi}(x) = \frac{\varphi(x)}{\|\varphi(x)\|} \tag{3.6}$$

(тут $(\varphi(x) = (\varphi_1(x), \varphi_2(x), ..., \varphi_N(x))^T$), необхідне для обробки інформації вихідним шаром, що є по суті кластеризувальною. нейронною мережею Т. Кохонена, налаштування параметрів якої проводиться на основі конкурентного самонавчання.

У цьому вихідному шарі вирішується завдання розбиття послідовності образів підвищеної розмірності $\tilde{\varphi}_1,...,\tilde{\varphi}_2,...,\tilde{\varphi}_k,...,\tilde{\varphi}_N$ на m кластерів з находженням прототипів центроїдів $\tilde{c}_1^{\,K},\tilde{c}_2^{\,K},...,\tilde{c}_m^{\,K} \in R^N$.

Незважаючи на певну простоту, при реалізації цього підходу можуть виникати суттєві обчислювальні проблеми при великому обсязі N оброблюваної вибірки, оскільки мережа, яка містить N нейронів, стає занадто громіздкою.

У зв'язку з цим є доцільним замість традиційної процедури навчання GRNN ввести метод, що дозволяє не тільки налаштувати параметри мережі, а й істотно скоротити число її R-нейронів.

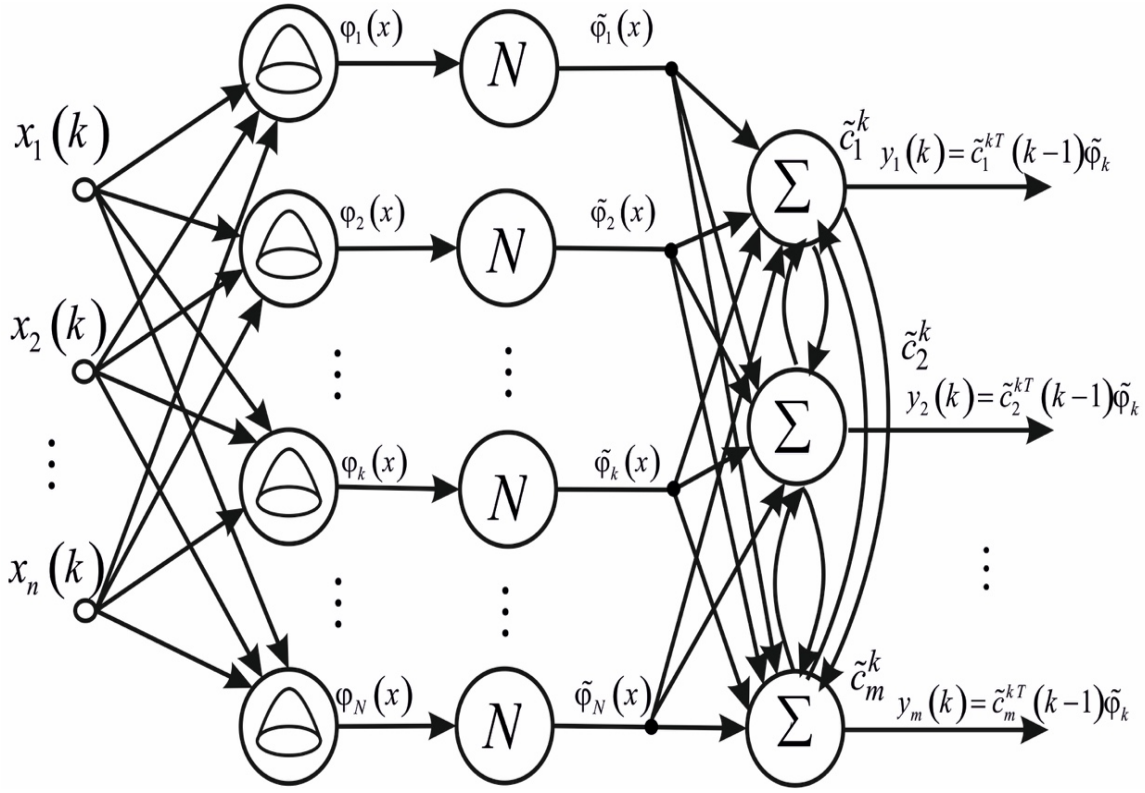

Рисунок 3.2 – Архітектура ядерної самоорганізовної мапи Т. Кохонена, на основі узагальненої регресійної нейронної мережі

3.4 Навчання ядерної самоорганізовної мапи Т. Кохонена на основі узагальненої регресійної нейронної мережі Д. Шпехта

Для формування першого шару даної гібридної нейронної мережі можна скористатися ідеями, що лежать в основі еволюційних систем обчислювального інтелекту [35], [36], адаптованими до online обробки інформації, що послідовно надходить в систему. Реалізація цього підходу проводиться в формі наступної послідовності кроків:

Крок 0: задати поріг нерозрізненості векторів центрів активаційних функцій Δ, максимально допустиму кількість нейронів в першому шарі $H \le N$ і параметр ширини рецепторного поля σ^2.

Крок 1: при надходженні спостереження $x(1)$ формується перший центр $H \le N$ і сама активаційна функція

$$\varphi_1(x) = e^{-\frac{\|x-c_1\|^2}{2\sigma^2}} = e^{-\frac{\|x-x(1)\|^2}{2\sigma^2}}. \qquad (3.7)$$

Крок 2: при надходженні спостереження $x(2)$ перевіряється умова

$$\|x(2)-c_1\|^2 \le \Delta, \qquad (3.8)$$

якщо воно виконується, то спостереження $x(2)$ не формує новий центр і автоматично вважається, що це спостереження належить тому ж кластеру, що і $x(1)$, Якщо виконується умова

$$\Delta < \|x(2)-c_1\| \le 2\Delta, \qquad (3.9)$$

то відбувається корекція c_1 згідно WTA-правила самонавчання Т. Кохонена в формі

$$c_1(2) = c_1(1) + \eta(2)(x(2) - c_1(1)) \tag{3.10}$$

(тут $0 < \eta(2) < 1 -$ параметр кроку настройки), якщо ж

$$2\Delta \le \left\| x(2) - c_1 \right\|, \tag{3.11}$$

то формується друга активаційна функція

$$\varphi_2(x) = e^{-\frac{\|x - c_2\|^2}{2\sigma^2}} = e^{-\frac{\|x - x(2)\|^2}{2\sigma^2}}. \tag{3.12}$$

Крок N: Якщо до $k -$го моменту надходження N-го вектору-образу $x(N)$ сформовано $h \le H$ активаційних функцій і виконується умова (2), процес нарощування числа R-нейронів першого шару закінчується і надалі структура цього шару залишається незмінною.

Оцінити якість функціонування першого шару можна було б, скориставшись виразом, яке для $h = H = N$ набуває вигляду

$$\hat{x}(k) = \frac{\sum_{k=1}^{N} x(k)\varphi_k(x(k))}{\sum_{k=1}^{N} \varphi_k(x(k))}, \tag{3.13}$$

після чого необхідно оцінити похибку відновлення вхідних образів:

$$\varepsilon = \frac{1}{N} \sum_{k=1}^{N} \frac{\left\| x(k) - \hat{x}(k) \right\|}{\left\| x(k) \right\|}. \tag{3.14}$$

Однак, оскільки в процесі формування першого шару за допомогою описаного вище підходу деякі з центрів активаційних функцій не збігаються з векторами спостережень, використання виразу, коректного для

«класичної» GRNN, в випадку, що розглядається, може виявитися неправомірним.

У цій ситуації можна скористатися модифікованою GRNN, чия архітектура приведена на рисунку 3.3.

Дана мережа подібно тришаровому персептрону містить три шари обробки інформації, проте, в якості активаційних функцій використовує радіально-базисні конструкції в першому прихованому шарі.

Другий прихований шар містить $n+1$ вузлів, n з яких є адаптивними лінійними асоціаторами, а $(n+1)-$й стандартним блоком підсумовування $\sum^*$. Вихідний сигнал мережі утворено n блоками ділення $\div$.

Навчання такої мережі є комбінованим процесом установки центрів радіально-базисних функцій на основі методів еволюційних систем і навчання зі вчителем синаптичних ваг лінійних асоціаторів. При цьому в якості навчального, тут використовується сам вхідний сигнал, тобто мережа налаштовується в автоасоціативному режимі.

Другий прихований шар мережі налаштовується аналогічно процесу навчання радіально-базисних нейронних мереж. При цьому на виходах n адаптивних лінійних асоціаторів формуються сигнали

$$o_i(k) = \sum_{l=0}^{h} w_{il}(k)\varphi_l(x(k)),\ i = 1,2...,n, \qquad (3.15)$$

а на виході суматорів $\sum^*$ з'являється сума $\sum_{l=0}^{h} \varphi_l(x(k))$.

Вихідний шар – це шар нормування вихідного сигналу по типу до нормованої радіально-базисної нейронної мережі так, що

$$\hat{x}_i(k) = \frac{\sum_{l=0}^{h} w_{il}(k)\varphi_l(x(k))}{\sum_{l=0}^{h} \varphi_l(x(k))} = \sum_{l=0}^{h} w_{il}(k) \frac{\varphi_l(x(k))}{\sum_{l=0}^{h} \varphi_l(x(k))} = \\ = \sum_{l=0}^{h} w_{il}(k)\varphi_l^*(x(k)), \tag{3.16}$$

$$\varphi_0(x(k)) = 1,\ \varphi_l^*(x(k)) = \varphi_l(x(k))(\sum_{l=0}^{h} \varphi_l(x(k)))^{-1}, \tag{3.17}$$

або

$$\hat{x}(k) = W(k)\varphi^*(k), \tag{3.18}$$

де

$$\hat{x}_i(k) = (\hat{x}_1(k),...\hat{x}_i(k),...,\hat{x}_n(k))^T, \tag{3.19}$$

$$\varphi^*(k) = (\varphi_0^*(x(k)),\varphi_1^*(x(k)),...,\varphi_h^*(x(k)))^T, \tag{3.20}$$

де $\mathrm{W}(k) - (n \times (h+1)) -$ матриця синаптичних ваг, що підлягає визначенню.

Для настройки матриці синаптичних ваг $\mathrm{W}(k)$ може бути використаний рекурентний метод найменших квадратів, який є по суті оптимальною за швидкодією гаусівсько-ньютонівською процедурою оптимізації виду:

$$\begin{cases} W(k) = W(k-1) + \dfrac{(x(k) - W(k-1)\varphi^*(k))\varphi^{*T}(k)P(k-1)}{1 + \varphi^{*T}(k)P(k-1)\varphi^*(k)}, \\ P(k) = P(k-1) - \dfrac{P(k-1)\varphi^*(k)\varphi^{*T}(k)P(k-1)}{1 + \varphi^{*T}(k)P(k-1)\varphi^*(k)}. \end{cases} \tag{3.21}$$

Таким чином, можна оцінити якість роботи першого шару, використовуючи у виразі (3.11) замість стандартного співвідношення (3.10) введені вище оцінки (3.12), (3.13).

Вихідний сигнал синтезованого першого шару в формі $(h\times 1)$ – вектору $\varphi(x)=(\varphi_1(x)...,\varphi_l(x),...,\varphi_h(x))^T$ у другому прихованому шарі нормалізується до виду

$$\tilde{\varphi}(x)=\varphi(x)\left\|\varphi(x)\right\|^{-1}, \tag{3.22}$$

тобто проектується на h-вимірну гіперсферу одиничного радіусу, після чого у вигляді послідовності $\tilde{\varphi}_1,\tilde{\varphi}_2,...,\tilde{\varphi}_k,...,\tilde{\varphi}_N$ надходить на вхід самоорганізовної мапи Т. Кохонена.

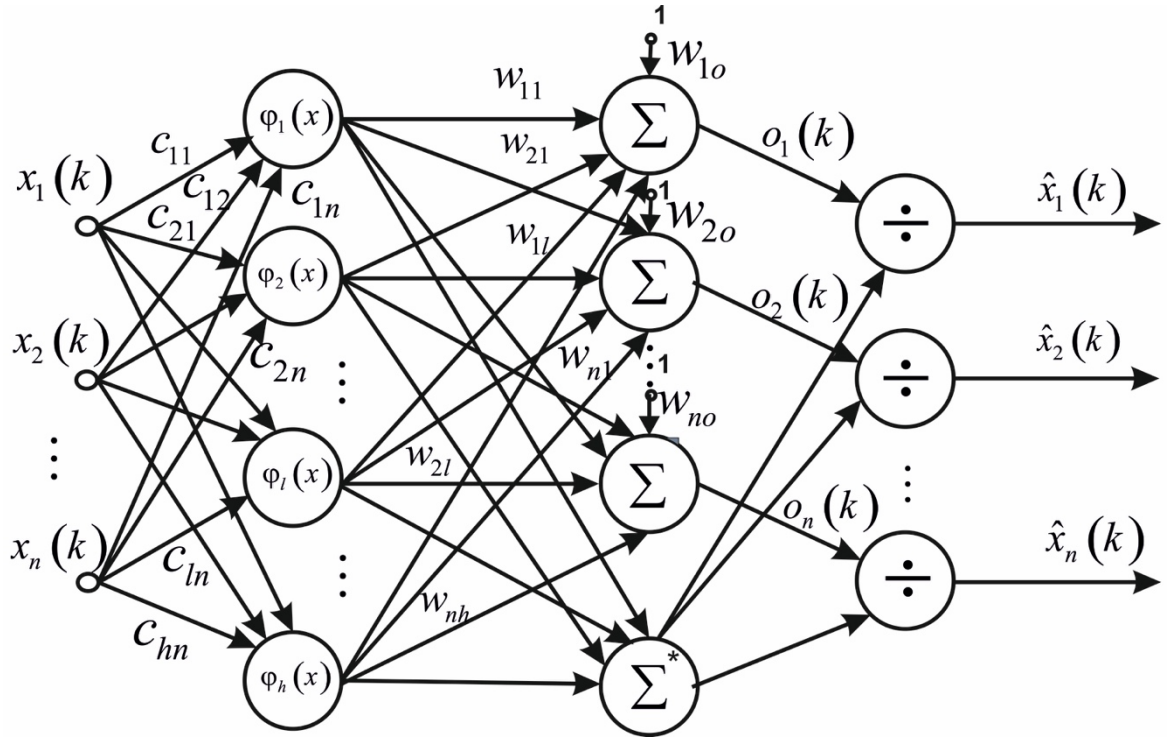

Рисунок 3.3 – Модифікована узагальнена регресійна нейронна мережа

Налаштування мапи Т. Кохонена, утвореної m адаптивними лінійними асоціаторами, виконується на основі WTM-правила самонавчання («Переможець отримує більше») і полягає в розбитті послідовності нормованих векторів-образів $\tilde{\varphi}_1, \tilde{\varphi}_2, ..., \tilde{\varphi}_N$ на m кластерів, кожен з яких характеризується власним прототипом-центроїдом $\tilde{c}_j^K \in R^h$, $j = 1,2,...,m$, що безперервно уточнюється при надходженні чергового образу підвищеної розмірності $\tilde{\varphi}_k$.

Процес самонавчання починається з ініціалізації синаптичних ваг вихідного шару, в якості яких виступають досить довільно вибрані початкові значення прототипів $\tilde{c}_j^K(0)$ такі, що

$$\left\| \tilde{c}_j^K(0) \right\| = 1. \tag{3.23}$$

При подачі на вхід третього шару сигналу $\tilde{\varphi}_1$ обчислюється m відстаней

$$D(\tilde{\varphi}_1, \tilde{c}_j^K(0)) = \left\| \tilde{\varphi}_1 - \tilde{c}_j^K(0) \right\|$$
$$\forall j = 1,2,...,m, \tag{3.24}$$

на підставі яких оцінюється нейрон-переможець, для якого

$$D(\tilde{\varphi}_1, \tilde{c}_*^K(0)) = \min_j D(\tilde{\varphi}_1, \tilde{c}_j^K(0)). \tag{3.25}$$

Після цього проводиться перший крок налаштування ваг-центроїдів

$$\tilde{c}_l^K(1) = \frac{\tilde{c}_l^K(0) + \eta(1)\psi(\tilde{c}_*^K(0), \tilde{c}_l^K(0))(\tilde{\varphi}_1 - \tilde{c}_l^K(0))}{\left\| \tilde{c}_l^K(0) + \eta(1)\psi(\tilde{c}_*^K(0), \tilde{c}_l^K(0))(\tilde{\varphi}_1 - \tilde{c}_l^K(0)) \right\|},$$

$$\forall l = 1, 2, ..., m. \tag{3.26}$$

Аналогічним чином можна записати правило самонавчання для *k*-го вектору-образу

$$\tilde{c}_l^K(k) = \frac{\tilde{c}_l^K(k-1) + \eta(k)\psi(\tilde{c}_*^K(k-1), \tilde{c}_l^K(k-1)(\tilde{\varphi}_k - \tilde{c}_l^K(k-1))}{\left\| \tilde{c}_l^K(k-1) + \eta(k)\psi(\tilde{c}_*^K(k-1), \tilde{c}_l^K(k-1))(\tilde{\varphi}_k - \tilde{c}_l^K(k-1)) \right\|},$$

$$\forall l = 1, 2, ..., m, \tag{3.27}$$

де $\psi(\tilde{c}_*^K(k-1), \tilde{c}_l^K(k-1))$ – так звана, функція сусідства, яка визначає локальну область топологічного сусідства, в якій налаштовуються не тільки нейрон-переможець $\tilde{c}_*^K$, але і його найближче оточення, при цьому більш близькі до переможця нейрони підтягуються по вхідному вектора $\tilde{\varphi}_k$ більше ніж віддалені від $\tilde{c}_*^K$ центроїди $\tilde{c}_l^K$.

В якості функції сусідства зазвичай використовується все той же гаусіан (хоча можна використовувати будь-яку ядерну функцію), який приймає в даному випадку вигляд

$$\psi(\tilde{c}_*^K(k), \tilde{c}_l^K(k)) = e^{-\frac{\left\| \tilde{c}_k^*(k) - \tilde{c}_l^K(k) \right\|^2}{2\sigma_c^2}}, \tag{3.28}$$

де σ_C^2 визначає розміри області сусідства, при цьому в процесі навчання цей параметр повинен монотонно зменшуватися.

Для навчання самоорганізовної мапи в деяких випадках пропонується взагалі не визначати переможця, а в якості функції сусідства використовувати вихідний сигнал кожного нейрона

$$y_l(k) = \tilde{\varphi}_k^T \tilde{c}_l^K, \tag{3.29}$$

при цьому правило (3.9) може бути переписано у вигляді

$$\tilde{c}_l^K(k) = \frac{\tilde{c}_l^K(k-1) + \eta(k) y_l(k)(\tilde{\varphi}_k - \tilde{c}_l^K(k-1))}{\left\| \tilde{c}_l^K(k-1) + \eta(k) y_l(k)(\tilde{\varphi}_k - \tilde{c}_l^K(k-1)) \right\|}, \quad \forall l = 1,2,...,m. \tag{3.30}$$

Треба відмітити, що

$$\left\| \tilde{\varphi}_k \right\| = \left\| \tilde{c}_l^K(k-1) \right\| = 1, \tag{3.31}$$

нескладно встановити, що вираз (3.10) є не що інше, як косинус кута між вхідним чином $\tilde{\varphi}_k$ і вектором-центроїдом $\tilde{c}_l^K(k-1)$, тобто $\cos(\tilde{\varphi}_k, \tilde{c}_l^K(k-1))$. тоді з урахуванням невід'ємності функції сусідства можна остаточно переписати правило самонавчання у вигляді

$$\tilde{c}_l^K(k) = \frac{\tilde{c}_l^K(k-1) + \eta(k)[\cos(\tilde{\varphi}_k, \tilde{c}_l^K(k-1))]_+ (\tilde{\varphi}_k - \tilde{c}_l^K(k-1))}{\left\| \tilde{c}_l^K(k-1) + \eta(k)[\cos(\tilde{\varphi}_k, \tilde{c}_l^K(k-1))]_+ (\tilde{\varphi}_k - \tilde{c}_l^K(k-1)) \right\|}, \tag{3.32}$$

$$\forall l = 1,2,...,m$$

де $[\bullet]_+$ – проектор на позитивний ортант.

Процес кластеризації закінчується або після вичерпання вибірки, що містить N спостережень, або відбувається безперервно, якщо дані надходять у формі потоку в online режимі.

Розроблена система призначена вирішувати задачу online кластеризації в умовах, коли утворені вихідними даними класи мають довільну форму. Запропонована архітектура еволюційної узагальненої регресійної нейромережі несхильна до «прокльону розмірності» та дозволяє опрацьовувати великі потоки даних. Завдяки використанню ядерних активаційних функцій ця нейронна мережа дозволяє кластеризувати данні які не є лінійно роздільними у вихідному просторі ознак.

Гібридна нейронна мережа об’єднує в собі всі переваги узагальненої регресійной мережі та самонавчанної мапи. Ця нейронна мережа проста в реалізації і дозволяє вирішувати досить широкий клас задач динамічного інтелектуального аналізу даних і інтелектуального аналізу потоків даних.

4 ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ І РІШЕННЯ ЗАДАЧ НА ТЕСТОВИХ ВИБІРКАХ

В даному розділі наведені результати моделювання розробленого методу ядерної кластеризації. Імітаційне моделювання виконувалося на тестових вибірках. Результати моделювання запропонованого методу були порівняні зі стандартними методами кластеризації з метою оцінки порівняльної якості вирішення розглянутих завдань.

В якості основних засобів для реалізації синтезованої в попередньому розділі гібридної нейронної мережі було використано мову програмування «Python 3.8». на сьогодні це найпопулярніша мова для розробок систем які пов’язані з обчислювальним інтелектом. Також були використані декілька бібліотек, а саме numpy – використання для опрацювання масивів даних: бібліотеки pandas та sklearn містять всі необхідні методи та функції для реалізації синтезованої системи, а також було використано декілька пакетів для візуалізації даних – matplotlib, seaborn. В якості менеджеру пакетів було використано Google CoLab.

4.1 Імітаційне моделювання ядерної кластеризувальної нейронної мережі

Для кращого опрацювання даних на початку було проведено препоцесінг всіх вибірок, що використовувались. Всі вибірки були попередньо перевірені на наявність викидів та дір. Далі було проведено кодування даних в інтервал [-1;1]. А для подальшої візуалізації даних був використаний метод головних компонент, який проводив компресію розкластерізованих даних на дві та три компоненти.

Спершу розроблена модель еволюційної нейронної мережі пройшла тестування своєї працездатності на штучно синтезованій вибірці яка була згенерована на основі довільного нормального розподілу. Як

можна побачити на рисунку 4.1 розроблена нейромережа провела кластерування даних коректно.

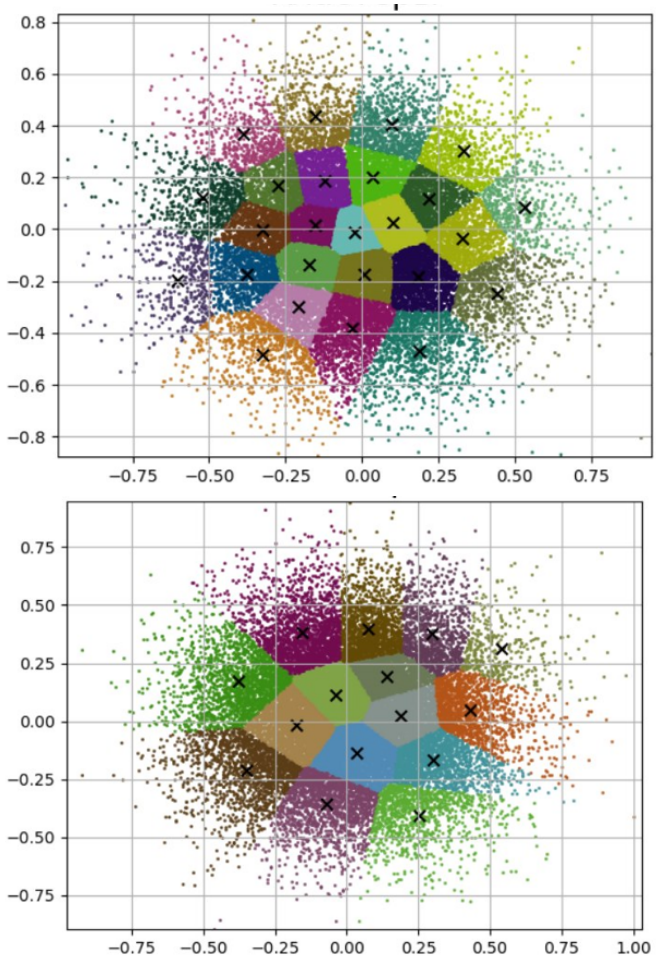

Рисунок 4.1 – Результати кластеризації штучно згенерованої вибіркина основі ядерної кластерувальної нейронної мережі (KSOM-GRNN)

Для підтвердження працездатності розробленої ядерної кластеризувальної нейронної мережі, заснованої на узагальненій регресійній нейронній мережі, була вирішена задача кластеризації на основі тестових вибірок з UCI-репозіторія [36], [37]. Всі вибірки, що використовувались буди попередньо опрацьовані та закодовані. Всі змінні були перевірені на пусті значення, категоріальні змінні були трансформовані в чисельні та закодовані, бінарні змінні також бути

перекодовані в числовий вигляд. Всі вибірки було перевірено на можливу наявність викідів через розрахунок середньо квартильних відстаней та розрахунку верхнього та нижнього персентилів. Всі знайдені по вибіркам викиди було виключено з тестових підсетів, але в навчальних даних викиди були навмисно залишені для того щоб система мала в майбутньому більш узагальнюючи властивості. Перед початком вибірки було поділено на три частини – навчальну, тренувальну та валідаційну вибірку в співвідношенні 70 на 30. Валідаційна вибірка складалася з 5 крос-валідаційних сетів з даниз, що було віддібрано до навчальної вибірки даних.

Були взяті такі набори даних.

1. Вибірка «Іриси». Вибірка складається з даних про 150 примірників ірису, по 50 примірників з трьох видів – Ірис щетинистий (Iris setosa), Ірис віргінський (Iris virginica) і Ірис різнокольоровий (Iris versicolor). Для кожного екземпляра вимірювалися чотири характеристики (в сантиметрах): довжина чашолистки (англ. Sepal length); ширина чашолистка (англ. sepal width); довжина пелюстки (англ. petal length); ширина пелюстки (англ. petal width).

2. Вибірка «Вино». Вибірка включає 178 векторів спостережень, розділених на 3 класу, кожне спостереження містить 13 ознак.

Для обраних з UCI-репозіторія вибірок було проведено детальний EDA аналіз. На прикладі вибірка «Іриси» розглянемо основні його етапи. Всі змінні було перевірено на викиди за допомогою побудови боксплотів та розрахунку інтерквартильних відстаней та визначення верхнього та нижнього персентилів.

Візуалізація боксплотів наведена на рисунку 4.2. Кількість викидів не є критичною та і для якісного навчання моделі їх неможна видаляти з навчальної підмножини.

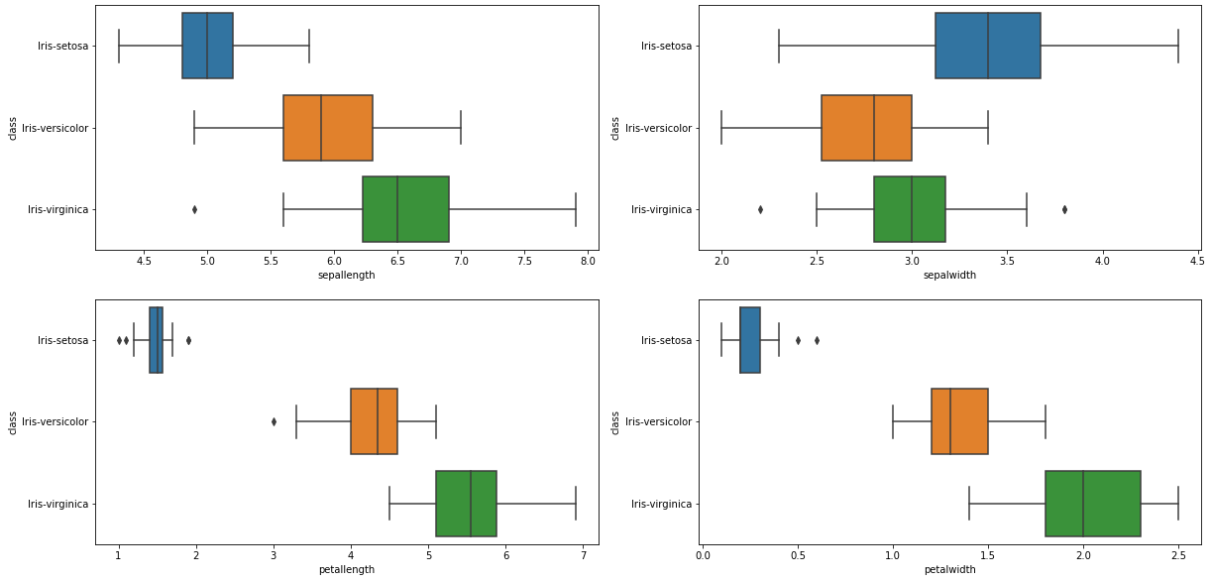

Рисунок 4.2 – Візуалізація знаходження викидів в вибірці «Іриси»

Після перевірки на викиди по всім змінним, що містить обрана вибірка було побудовано графіки щільності розподілів (рисунок 4.3), по відтвореним графікам можна зробити припущення про те, що розподіл по змінним довжина чашолистки, довжина пелюстки та ширина пелюстки є бімодальними (тут може бути дійсною гіпотеза про те, що ця вибірка містить 3 кластери). А розподіл по змінній ширина чашолистка є нормальним, хоча правий хвіст цього розподілу є важчим за лівий. Цей факт є підтвердженням того, що саме по цій змінній є викиди.

На рисунку 4.4 зображено попарне співвідношення по всім змінним відносно класу ірису. Головна діагональ – це щільності розподілу по змінним, лівий нижній кут – це також щільність розподілу, але це вертикальні зрізи розподілів, а верхній правий кут – це попарне співвідношення по парам змінних. Ця візуалізація також є підтвердження того, що вибірка містить три кластери, з яких один є лінійно роздільним, а два інших перетинаються у вихідному просторі ознак.

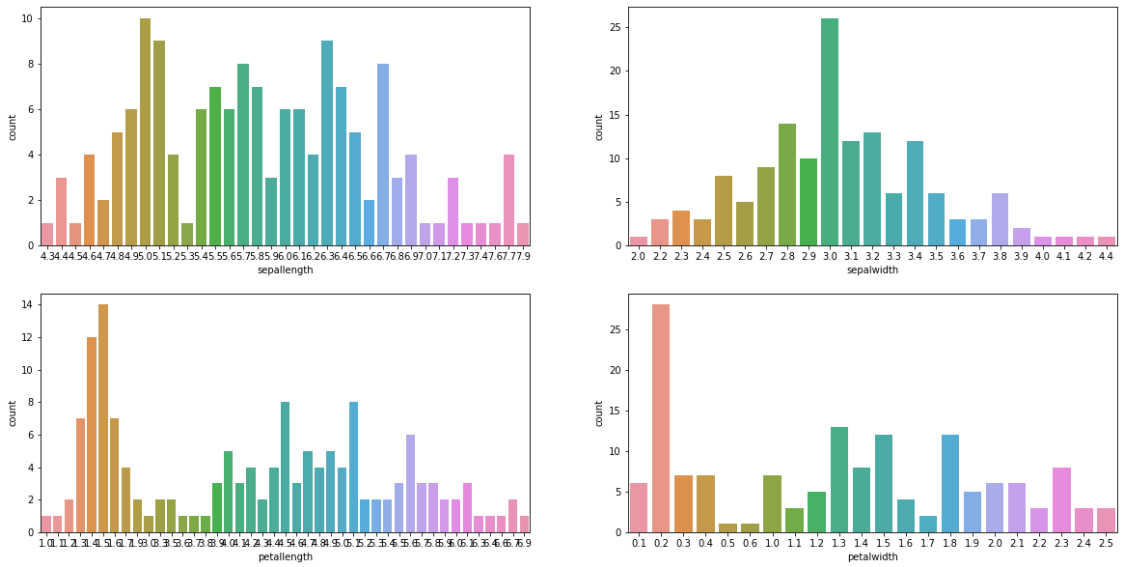

Рисунок 4.3 – Графік щільності розподілу по всім змінним вибірки «Іриси»

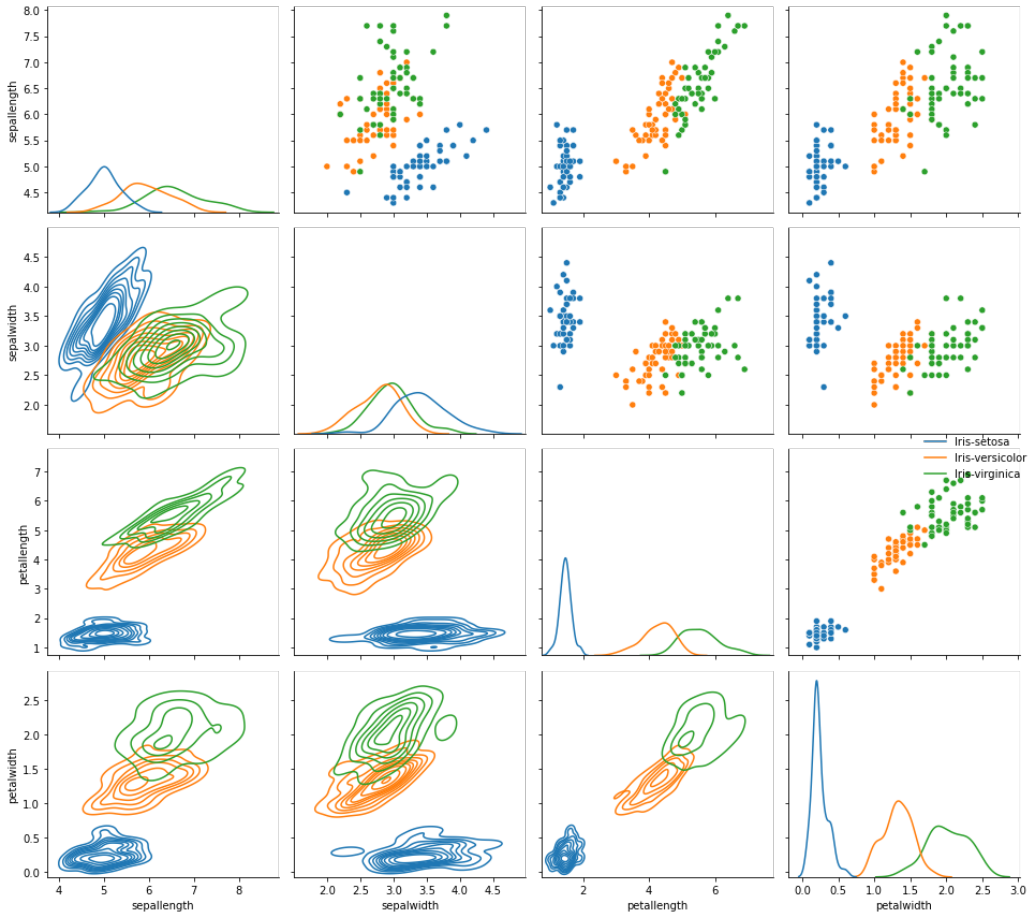

Рисунок 4.4 – Попарне співвідношення по всім змінним відносно класу квітів ірису

З метою оцінки ефективності ядерної кластеризувальної нейронної мережі (KSOM-GRNN) результати кластеризації були порівняні зі стандартним методом С-середніх (FCM) з параметром фаззіфікації 2, стандартною мапою Т. Кохонена (SOM) і узагальненою регресійною нейронною мережею (GRNN).

Оскільки для кожної вибірки існують мітки вірної класифікації, ефективність кластеризації вимірювалася у відсотках точності щодо еталонного значення. У кожній клітинці таблиці 4.1 наведено середній, мінімальний і максимальний результат для серії з 50 експериментів. Якість всіх наведених кластерувальних систем вимірювалась та порівнювалась на тестових вибірках.

Результати показують, що ефективність ядерної кластеризувальної нейронної мережі (KSOM-GRNN) вища і стабільніша, ніж у SOM, FCM і GRNN.

Таблиця 4.1 – Порівняння точності кластеризації на тестових вибірках

Методи, що будо досліджено	Iris			Wine		
	avg	max	min	avg	max	min
Стандартна мапа Кохонена (SOM)	83%	92%	80%	85%	94%	77%
Метод нечітких С-середніх (FCM з параметром $\beta = 2$)	92%	95%	84%	93%	96%	85%
Узагальнено регресійна нейронна мережа (GRNN)	82%	94%	76%	80%	89%	76%
Ядерна кластерувальна нейронна мережа (KSOM-GRNN)	95%	97%	88%	95%	97%	90%

У таблиці 4.2 представлена еволюція центроїдів розробленої нейронної мережі.

Таблиця 4.2 – Перші три етапи процедури навчання центроїдів

Номер центроїда	Координати центроїда
C11	-1.5886 0.4635 0.6624
C12	1.3848 0.3887 1.8371
C13	0.6149 0.5035 -1.9144

На рисунках 4.5-4.8 наведені результати кластерування за допомогою розробленою кластерувальною мережею вибірок «Іриси» та «Вино».

Для того щоб показати як змінювались координати центроїдів кластерів була зроблена додаткова візуалізація даних на якій пунктирною лінією відображається зміщення центрів (рисунок 4.6)

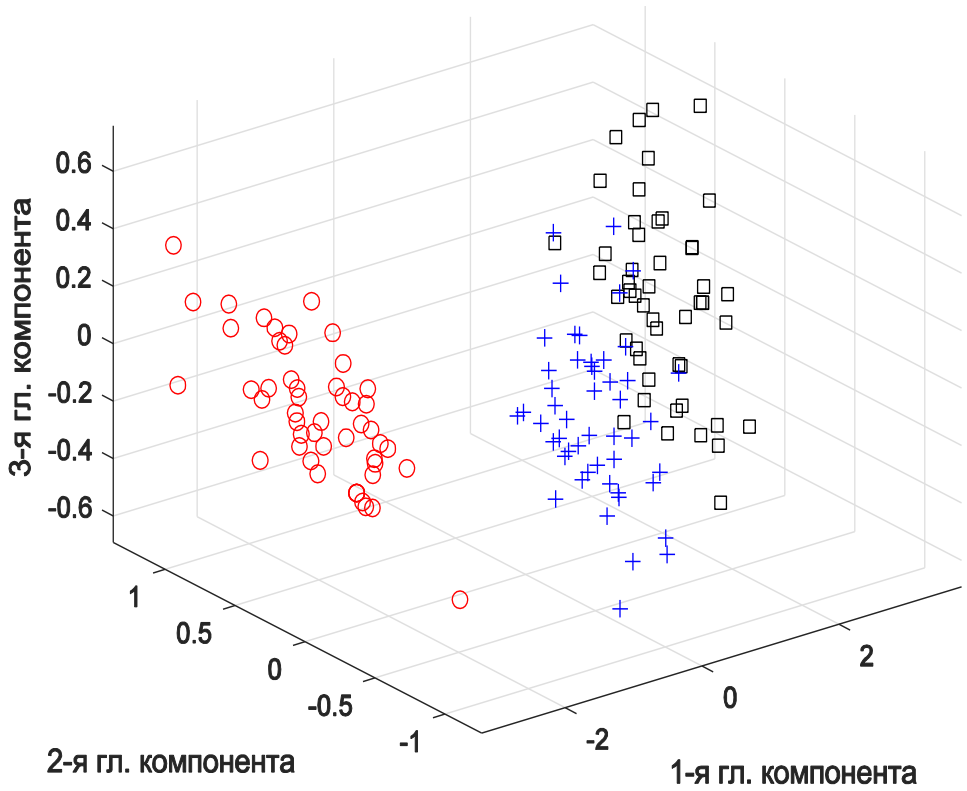

Рисунок 4.5 – Результати кластерування вибірки «Іриси» розробленою кластерувальною мережею

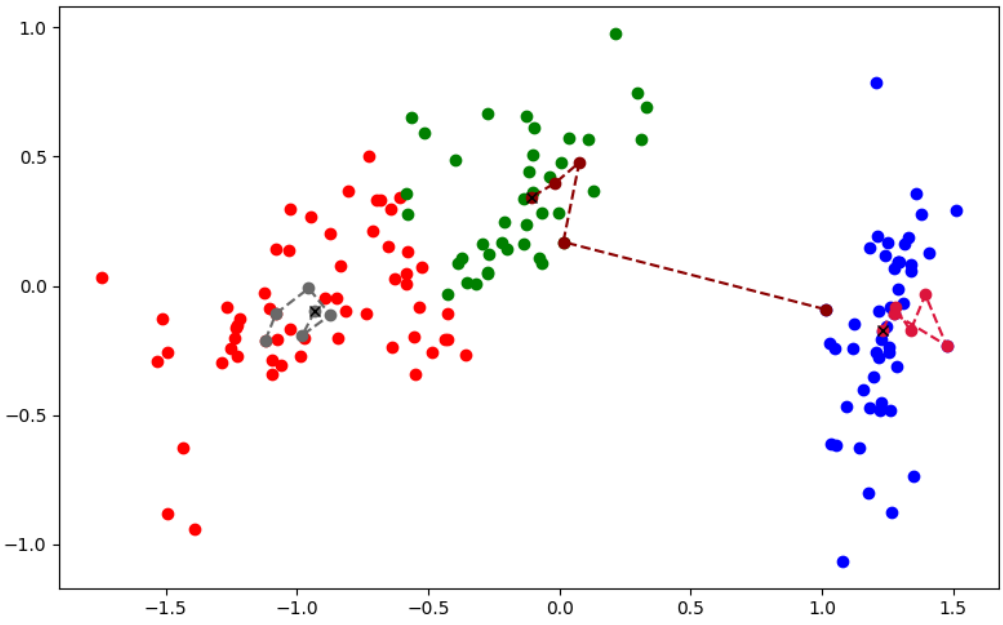

Рисунок 4.6 – Зміщення центроїдів вибірки «Іриси»

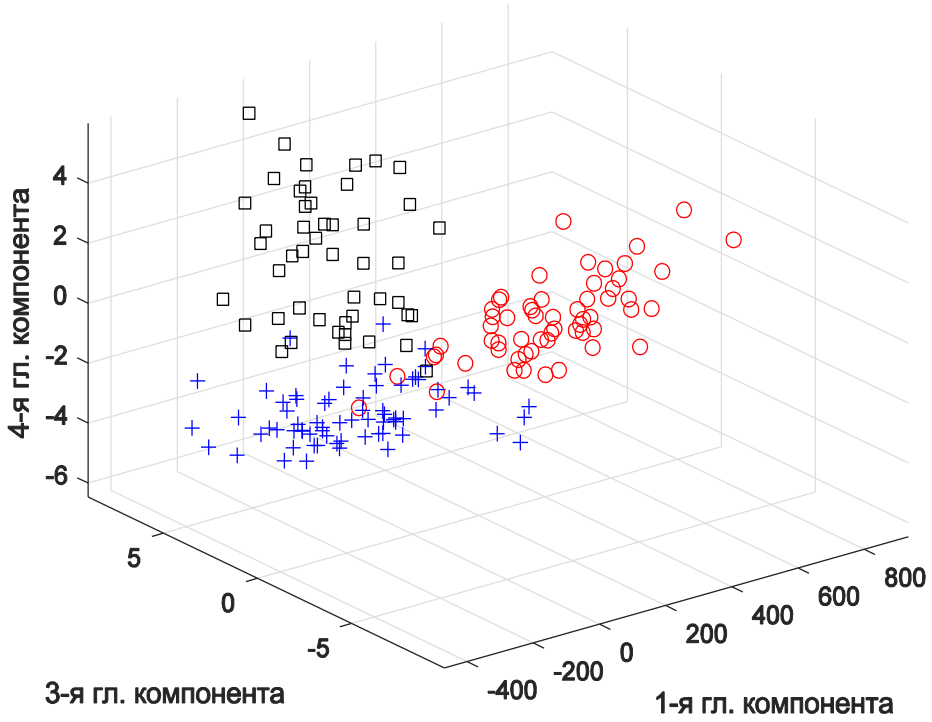

Рисунок 4.7 – Результати кластеризації вибірки «Вино» розробленою кластерувальною мережею

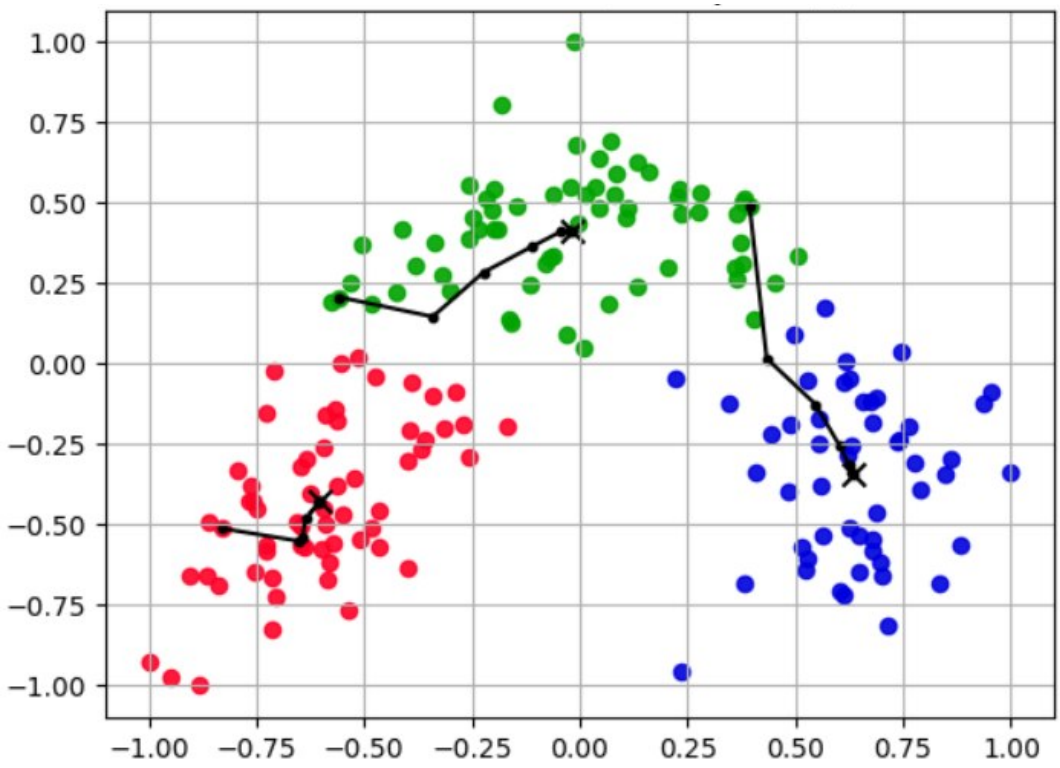

Рисунок 4.8 – Зміщення центроїдів вибірки «Вино»

4.2 Аналіз отриманих результатів

З огляду на те, що точність якості кластеризації розглянутих вище архітектур ідентичні їх пакетним аналогам, найбільш цікавою для експериментального дослідження була визнана швидкість навчання системи. В якості запобіжного модельного часу, як мінімальний загальний квант пакетних і рекурентних форм даних методів, прийнято кількість проходів (епох, ітерацій) по всій доступній вибірці спостережень. У проведеної серії експериментів тестувався час, за який система досягає заданої точності кластеризації. Для тестування використовувалася традиційна вибірка «Вино» (178 13-вимірних спостережень, розділених на 3 класи). З кожним з розглянутих методів і архітектур була проведена серія з 50 експериментів. Кожен експеримент включав 25 ітерацій навчання.

Спочатку система тестувалася на навчальній множині, що включає 66% вибірки, після чого вимірювалася точність кластеризації на всій вибірці. На графіках на рисунку 4.9 наведена середня точність кластеризації кожної з архітектур в залежності від кількості проходів по вибірці.

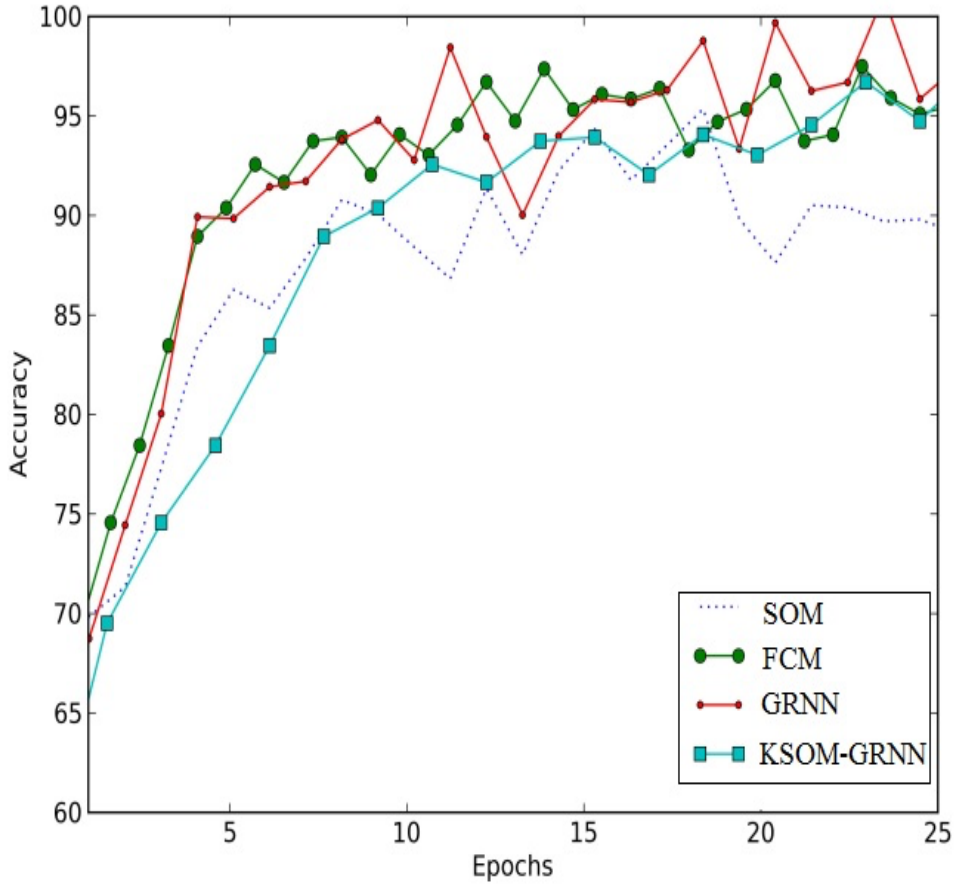

Рисунок 4.9 – Порівняльна точність кластеування відносно епох навчання

Зауважимо, що через велику кількість обчислюваних параметрів, ядерна кластерувальна нейронна мережі вимагає для якісної настройки більшої кількості спостережень, ніж пакетні форми, що особливо вигідно виділяє синтезовану архітектуру на тлі інших вже відомих систем, які традиційно відрізняються нестабільністю результатів. Розроблена модель еволюційної нейронної мережі об’єднала в собі всі переваги узагальненої регресійної нейронної мережі та самонавчанної карти Кохонена.

Розроблена модель позволяє опрацьовувати набори даних в послідовному режимі, не схильна до «прокльону розмірності» та завдяки ядерним функціям активації вирішує завдання розпізнавання образів в умовах коли вхідний простір не є лінійно роздільним. Завдяки тому, що модель навчалась на даних з викидами вона має узагальнюючі властивості і не потребує детального перенавчання для інших даних. Чисельні експерименти підтвердили, що еволюційна мережа вирішує завдання послідовної кластеризації і демонструє точність вище 90%.

ВИСНОВКИ

В роботі представлені результати, що є відповідно до поставленої мети рішенням актуальної задачі обробки багатовимірних масивів даних в умовах невизначеності за допомогою методу ядерної кластеризації на основі об’єднання узагальнено регресійної нейромережі та самонавчанної мапи Кохонена. Проведені дослідження дозволили зробити наступні висновки.

1. Проаналізовано стан проблеми кластеризації даних і розглянуті існуючі підходи до її вирішення. Проаналізовано стан сучасної теорії гібридних систем обчислювального інтелекту, призначених для вирішення задач обробки даних в тому числі і задач інтелектуального аналізу даних, а також розглянуто основні підходи до її реалізації.

2. Відповідно до поставленої мети було встановлено основні етапи аналізу та створення кластерувальної ядерної системи.

3. Запропоновано архітектуру гібридної нейронної мережі та метод її самонавчання, призначені для ядерної кластеризації потоку спостережень, які послідовно в online режимі надходять на обробку. Запропоновано систему яка побудована на основі еволюціїної узагальненої регресійної нейронної мережі та самоорганізовної мапи Т. Кохонена.

4. Було вирішено задачу кластеризації на основі тестових вибірок з UCI-репозіторія за допомогою розробленої ядерної кластерувальної нейронної мережі, заснованої на узагальненій регресійній нейронній мережі.

5. Запропонована система дозволяє вирішувати задачу online кластеризації в умовах, коли утворені вихідними даними класи мають довільну форму та їх кількість невідома апріорі. Введена нейронна мережа проста в реалізації і дозволяє вирішувати досить широкий клас задач динамічного інтелектуального аналізу даних (DDM) і інтелектуального аналізу потоків даних (DSM).

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Braun H. Neuronale Netze. Optimierung durch Lernen und Evolution. Berlin : Springer-Verlag, 1997. 279 p.

2. Dracopoulos D.C. Evolutionary Learning Algorithms for Neural Adaptive Control. Evolutionary Learning Algorithms for Neural Adaptive Control. Berlin : Springer-Verlag, 1997. 211 p.

3. Shepherd A.J. Second-Order Methods for Neural Networks. London : Springer-Verlag, 1997. 145 p.

4. Haykin S. Neural Networks. A Comprehensive Foundation. Upper Saddle River, NJ : Prentice Hall, Inc., 1999. 842 p.

5. Schalkoff R.J. Artificial Neural Networks. NY : The McGraw-Hill Comp., Inc., 1997. 422 p.

6. Rojas R. Neural Networks. A Systematic Introduction. Berlin : Springer-Verlag, 1996. 502 p.

7. Ципкин Я. З. Основи теорії навчаються систем. М. : Наука, 1970. 252 с.

8. Фомін В.Н., Фрадков А.Л., Якубович В.А. Адаптивне керування динамічними об'єктами. М. : Наука, 1981. 448 с.

9. Вапник В.Н. Відновлення залежностей за емпіричними даними. М. : Наука, 1979. 448 с.

10. Moody J., Darken C.J. Fast learning in networks of locally-tuned processing units. *Neural Computation*. 1989. №1. P. 281-294.

11. Zahirniak D., Chapman R., Rogers S., Suter B., Kabritsky M., Piati V. Pattern recognition using radial basis function network. proc 6th Ann. Aerospace Application of Artificial Intelligence Conf. Dayton, OH, 1990. P. 249-260.

12. Park J., Sandberg I.W. Universal approximation using radial-basis-function networks. *Neural Computation*. 1991. №3. P.246-257.

13. Schilling, R.J., Carrol J.J., Al-Ajlouni A.F. Approximation of nonlinear systems with radial basis function neural networks. IEEE Trans. on Neural Networks. 2001. №12. P. 1-15.

14. Klawonn F., Höppner F., Jayaram B. What are clusters in high dimensions and are they difficult to find. *Lecture Nodes in Computer Science*. Berlin Heidelberg : Springer – Verlag, 2015. Vol. 7627 P. 14-33.

15. Parzen E. On the estimation of a probability density function and the mode. Ann. Math. Stat. 1962. № 38. P.1065-1076.

16. Bishop C.M. Neural Networks for Pattern Recognition. Oxford : Clarendon Press, 1995. 482 p.

17. Borgelt C. Prototype-based Classification and Clustering. Magdeburg, 2005. 350 p.

18. Ball G.H., Hall D.J. A Clustering Technique for Summarizing Multivariate Data. *Behavioral Science*. 12(2). 1967. P. 153-155.

19. Abonyi J., Feil B. Cluster Analysis for Data Mining and System Identification. Basel: Birkhauser, 2007. 303 p.

20. Hartigan J.A., Wong M.A. A k-means clustering algorithm. *Applied Statistics.* 1979. №28. P. 100-108.

21. Kohonen T. Self-Organizing Maps. Berlin: Springer-Verlag, 1995. 362 p.

22. Aggarwal C.C., Reddy C.K. Data Clustering. Algorithms and Application. Boca Raton: CRC Press, 2014. 648 p.

23. Xu R., Wunsch D.C. Clustering. IEEE Press Series on Computational Intelligence. Hoboken, NJ: John Wiley & Sons, Inc., 2009, 370 p.

24. Rutkowski L. Computational Intelligence. Methods and Techniques. Berlin-Heidelberg: Springer-Verlag, 2008. 514 p.

25. Du K.-L., Swamy M.N.S. Neural Networks and Statistical Learning. London: Springer-Verlag, 2014. 824 p.

26. Haykin S. Neural Networks. A Comprehensive Foundation. Upper Saddle River, N.J.: Prentice Hall, Inc., 1999. 842 p.

27. Бодянский Е.В., Руденко О.Г. Искусственные нейронные сети: архитектуры, обучение, применение. Харьков: ТЕЛЕТЕХ, 2004. 372 с.

28. Nelles O. Nonlinear System Identification. Berlin: Springer, 2001. 785 p.

29. Specht D.E. A general regression neural network. IEEE Trans. on Neural Networks. 1991. №2. P. 568-576.

30. Girolami M. Mercer kernel based clustering in feature space. IEEE Trans. on Neural Networks. 2002. №13. V. 3. P. 789-784.

31. Вапник В.Н., Червоненкис А.Я. Теория распознавания образов (Статистические проблемы обучения). М.: Наука, 1974. 416 с.

32. Cortes C., Vapnik V. Support Vector Networks. *Machine Learning*. 1995. №20. P. 273-297.

33. Nelles O. Nonlinear System Identification. Berlin: Springer, 2001. 785 p.

34. Angelov P. Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems. Heidelberg, New York : Springer-Verlag, 2002. 227 p.

35. Lughofer E. Evolving Fuzzy Systems – Methodologies, Advanced Concepts and Applications. Berlin, Heidelberg : Springer-Verlag, 2011. 456 p.

36. Murphy P. M. UCI Repository of machine learning databases. CA: University of California, Department of Information and Computer Science. 1994. URL : http://www.ics.uci.edu/mlearn/MLRepository.html (дата звернення 15.05.2018).