

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_  
(повна назва)

Кафедра \_\_\_\_\_ Штучного інтелекту \_\_\_\_\_  
(повна назва)

**АТЕСТАЦІЙНА РОБОТА**  
**Пояснювальна записка**

Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_  
(рівень вищої освіти)

\_\_\_\_\_ Нейро-фаззі самоорганізовна мапа \_\_\_\_\_  
\_\_\_\_\_ для пошуку глобального екстремума \_\_\_\_\_  
(тема)

Виконав:  
студент 2 курсу, групи СШМ-18-1  
Сербіна Д.В. \_\_\_\_\_  
(прізвище, ініціали)

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

Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)  
Освітня програма Системи штучного  
інтелекту (СШ) \_\_\_\_\_  
(повна назва освітньої програми)

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

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

В.О. Філатов  
(прізвище, ініціали)

2019 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_

Кафедра \_\_\_\_\_ Штучного інтелекту \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

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

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

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

ЗАТВЕРДЖУЮ:  
Зав. кафедри \_\_\_\_\_  
(підпис)  
«\_\_\_\_» \_\_\_\_\_ 2019 р.

**ЗАВДАННЯ**

НА АТЕСТАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Сербіній Дар'ї Володимирівні \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Нейро-фаззі самоорганізовна мапа для пошуку глобального екстремума

затверджена наказом по університету від 04 листопада 2019 р. № 1623Ст

2. Термін подання студентом роботи до екзаменаційної комісії 18 грудня 2019 р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проектів щодо розробки та дослідження нечіткої нейронної мережі для пошуку глобального екстремуму, дані відомих наукових проектів щодо розробки та дослідження алгоритмів стрибків з локальних екстремумів, науково-технічні публікації стосовно розробки самоорганізовних мап для вирішення різних задач, тестові виборки з UCI-репозиторію, деталі технічної реалізації

4. Перелік питань, що потрібно опрацювати в роботі Аналіз предметної галузі та постановка задачі, нечіткі нейронні мережі і системи з нечіткою логікою в задачах кластерного аналізу, метрики, подібність їх застосування, алгоритми самоорганізацій, алгоритми кластерування, конкурентні нейронні мережі, нейромережі Кохонена, нейро-фаззі самоорганізовна мапа, імітаційне моделювання.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)  
Рисунок 1 – Порогова функція, Рисунок 2 – Лінійно-порогова функція, Рисунок 3 – Сигмоїдальна функція, Рисунок 4 – Схема нейрона, Рисунок 5 – Найпростіша штучна мережа, Рисунок 6 – Функції належності лінгвістичної змінної, Рисунок 7 – Дендрограма агломеративних і дивізимних методів, Рисунок 8 – 1D-мапа Когонена, Рисунок 9 – Топологія мап Когонена, Рисунок 10 – Геометрична інтерпретація налаштування ваг мережі Когонена, Рисунок 11 – Найпростіша конкурентна мережа, Рисунок 12 – Кластерування на основі конкурентного навчання, Рисунок 13 – Адаптивна фаззі-самоорганізовна нейронна мережа, Рисунок 14 – Результат роботи системи на вибірці Wine, Рисунок 15 – Результат роботи системи на вибірці Dermatorogy, Рисунок 16 – Результат роботи системи на вибірці Ionosphere.

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	проф. Бодянський Є.В.		

#### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	04.11.19	виконано
2	Аналіз предметної області і постановка завдання	04.11.19-12.11.19	виконано
3	Дослідження методів навчання	13.11.19-16.11.19	виконано
4	Створення імітаційної моделі	17.11.19-21.11.19	виконано
5	Тестування і відладка імітаційної моделі	22.11.19-28.11.19	виконано
6	Обробка і оформлення результатів	29.11.19-01.11.19	виконано
7	Оформлення пояснювальної записки	02.12.19-10.12.19	виконано
8	Нормоконтроль	11.12.19	виконано
9	Попередній захист	16.12.19	виконано
10	Захист перед ЕК	18.12.19	

Дата видачі завдання 04 листопада 2019 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ проф. Бодянський Є.В.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Записка пояснювальна: 82 сторінка, 16 рисунків, 3 таблиць, 38 формул, 21 джерело, 2 додатки.

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, КЛАСИФІКАЦІЯ, КЛАСТЕРИЗАЦІЯ, НЕЙРОННА МЕРЕЖА, НЕЧІТКІ АЛГОРИТМИ, ПЕРЕМОЖЕЦЬ ОТРИМУЄ БІЛЬШЕ, ПЕРЕМОЖЕЦЬ ОТРИМУЄ ВСЕ, САМООРГАНІЗОВНА МАПА КОГОНЕНА, ШТУЧНИЙ ІНТЕЛЕКТ

В атестаційній роботі розглядається задача створення алгоритму нечіткої кластеризації даних з використанням процедури глобального випадкового пошуку, що працює в послідовному режимі.

Метою атестаційної роботи є створення послідовного алгоритму, з використанням процедури випадкового пошуку, що справляється з локальними екстремумами, в умовах пересічних класів.

Об'єктом дослідження є процес обробки даних, що надходять одне за одним, в послідовному режимі за допомогою нейро-фаззі самоорганізовної мапи.

Предметом дослідження є нечіткий online кластер, захищений від застрягання в локальних екстремуму з використанням штучних нейронних мереж.

Методи дослідження базуються на математичному апараті Data Mining, методах обчислювального інтелекту, теорії оптимізації та самонавчання нейронних мереж і нейро-фаззі систем.

Імітаційне моделювання застосовується для перевірки якості виконаної роботи з використанням алгоритму послідовно нечіткої кластеризації даних.

## РЕФЕРАТ

Пояснительная записка: 82 страницы, 16 рисунков, 3 таблицы, 38 формул, 21 источник, 2 приложения.

ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ,  
ИСКУССТВЕННЫЙ ИНТЕЛЕКТ, КЛАССИФИКАЦИЯ,  
КЛАСТЕРИЗАЦИЯ, НЕЙРОННАЯ СЕТЬ, НЕЧЕТКИЕ  
АЛГОРИТМЫ, ПОБЕДИТЕЛЬ ПОЛУЧАЕТ БОЛЬШЕ,  
ПОБЕДИТЕЛЬ ПОЛУЧАЕТ ВСЕ, САМООРГАНИЗУЮЩАЯСЯ  
КАРТА КОХОНЕНА

В аттестационной работе рассматривается задача создания алгоритма нечеткой кластеризации данных с использованием процедуры глобального случайного поиска, работающего в последовательном режиме.

Целью аттестационной работы является создание алгоритма работающего последовательно, с использованием процедуры случайного поиска, справляющегося с локальными экстремумами.

Объектом исследования является процесс обработки данных, поступающих одно за одним, в последовательном режиме при помощи нейро-фаззи самоорганизующейся карты.

Предметом исследования является нечеткий online кластер, защищенный от застреваний в локальных экстремумах с использованием искусственных нейронных сетей.

Методы исследования базируются на математическом аппарате Data Mining, нейронных сетей и нейро-фаззи систем.

Имитационное моделирование применяется для проверки качества проделанной работы с использованием алгоритма последовательно нечеткой кластеризации данных.

## ABSTRACT

Explanatory note: 82 pages, 16 figures, 3 tables, 38 formulas, 21 sources, 2 annex.

ARTIFICIAL INTELLIGENCE, CLASSIFICATION, CLUSTERING, DATA MINING, FUZZY ALGORITHMS, KOHONEN'S SELF-ORGANIZING MAPS, NEURAL NETWORK, WINNER TAKES ALL, WINNER TAKES MORE

In the attestation work addresses the task of creating a fuzzy data clustering algorithm using a global random search procedure operating in sequential mode.

The goal of certification work is to create an algorithm that works sequentially using a random search procedure that copes with local extremums under intersecting classes.

The object of the study is the processing of data, arriving one by one, in a sequential mode using the neuro-fuzzy self-organized map.

The subject of the study is a fuzzy online cluster, protected from being stuck in local extremes using artificial neural networks.

Research methods are based on the mathematical apparatus of Data Mining, methods of computational intelligence, the theory of optimization and self-learning of neural networks and neuro-fuzzy systems. Simulation is used to check the quality of the work done using an algorithm sequentially fuzzy clustering of data.

## ЗМІСТ

<b>Перелік умовних позначень, символів, одиниць, скорочень і термінів</b>	<b>3</b>
<b>Вступ</b> .....	<b>4</b>
<b>1 Аналіз предметної галузі та формалізована постановка задачі</b> .....	<b>7</b>
1.1 Поняття штучної нейронної мережі та її складових .....	8
1.2 Навчання штучних нейронних мереж.....	12
1.3 Нечітка логіка та нейро-фаззі системи .....	15
1.4 Загальна постановка задачі кластерування. Застосування нейронних мереж для кластерування масивів.....	19
1.5 Постановка завдання дослідження.....	21
<b>2 Нечіткі нейронні мережі і системи з нечіткою логікою в задачах кластерного аналізу</b> .....	<b>23</b>
2.1 Формалізована постановка задачі кластерування .....	24
2.2 Метрики, подібність та їх застосування .....	25
2.3 Методи кластерного аналізу .....	26
<b>3 Нейро-фаззі самоорганізовна мапа для пошуку глобального</b> .....	<b>30</b>
3.1 Алгоритм самоорганізації K-середніх .....	31
3.2 Алгоритм кластерування Fuzzy C-means.....	32
3.3 Алгоритм кластерування J-means.....	34
3.4 Алгоритм кластерування Fuzzy J-means.....	35
3.5 Алгоритми «стрибку» для пошуку глобального екстремума.....	37
3.6 Самоорганізованні мапи Т. Когонена .....	40
3.7 Конкурентні нейронні мережі .....	46
3.8 Online нейро-фаззі самоорганізовної мапи для пошуку глобального екстремума.....	49
<b>4 Імітаційне моделювання</b> .....	<b>55</b>
4.1 Імітаційне моделювання на виборці Wine.....	55
4.2 Імітаційне моделювання на виборці Dermatology .....	57
4.3 Імітаційне моделювання на виборці Ionosphere.....	59

4.4 Висновки імітаційного моделювання .....	61
<b>Висновки</b> .....	63
<b>Перелік посилань</b> .....	65

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

ШНМ – штучна нейрона мережа;

FCM – fuzzy c-means – нечіткий алгоритм c-середніх;

FKCN – fuzzy Kohonen clustering network;

SOM – Kohonen's Self-Organizing Maps – самоорганізовані мапи Т.

Когонена;

WTA – winner takes all – переможець отримує все;

WTM – winner takes more – переможець отримує більше.

## ВСТУП

Однією з важливих проблем сучасності є збір та обробка даних. Обсяги інформації зростають дуже швидко, і тому важливим є вибір методів для її опрацювання. Саме це питання висвітлюється у розділах інтелектуального аналізу даних і машинного навчання [1].

Дуже актуальним напрямом є задача кластерування даних, що є одним із методів аналізу та обробки множин об'єктів. Його мета полягає у групуванні елементів на окремі підгрупи – кластери, що схожі за певними ознаками. Спектр застосувань кластерного аналізу дуже широкий, бо його використання можливе майже у всіх сферах людського життя.

Так як в умовах новітнього часу даних стає дедалі більше, потреба у адекватному групуванні зростає пропорційно [2].

Всі методи кластерування можна умовно поділити на ієрархічні та неієрархічні. В основі неієрархічних методів лежить попередньо задана кількість кластерів або існує можливість використання складних алгоритмів для знаходження їх кількості. Ієрархічне ж кластерування реалізується за допомогою послідовного об'єднання менших кластерів до більших (агломеративне кластерування) чи навпаки розділення більших на менші (дивізімне кластерування). Різниця цих методів полягає у тому, що на відміну від неієрархічних, алгоритми ієрархічного кластерування будують розбиття для всіх можливих варіантів. Проблемою кластерування є знаходження точки зупинки алгоритму [3].

На сьогоднішній день запропоновано багато підходів для отримання якісних результатів кластерування, але не існує однозначного рішення, яке б задовольняло всі вимоги. Серед підходів, що використовуються, особливо ефективними показали себе нейронні мережі та нейро-фаззі системи, завдяки, насамперед своїм універсальним апроксимуючим та екстраполуючим можливостям та здатності навчатися в умовах суттєвої структурної і параметричної невизначенності.

В задачах інтелектуального аналізу даних, що пов'язані з кластеруванням широкого розповсюдження набули самоорганізовані мапи Когонена. Властивості самоорганізації SOM пов'язані з тим, що налаштування синаптичних ваг відбувається за відсутності зовнішнього навчального сигналу, тобто в режимі самонавчання, при цьому кожен вхідний образ викликає налаштування тих чи інших параметрів.

У випадку перетинаючихся та неопуклих кластерів рішення про належність вхідного образу до одного з кластерів, що приймається згідно до правила «переможець отримує все», може давати неточне кластерування. Це зумовлено тим, що деякі образи можуть належати в той самий час до декількох кластерів з певним ступенем належності [4]. Саме у зв'язку з цим доцільно забезпечити SOM можливості і властивості нечіткого кластерування.

Традиційні методи вирішення задач класифікації і кластерування базуються на деяких апріорних припущеннях про характер розподілу даних і інші властивості вибірки, і тому для ситуацій, в яких потрібна послідовна обробка зростаючого масиву даних, властивості яких змінюються з часом, стає виправданим застосування складних гібридних і нейромережових методів.

Одним з найбільш популярних алгоритмів кластеризації є метод k-середніх, завдяки своїй простоті, наочності результатів і можливості їх ясної інтерпретації. Цей метод відноситься до алгоритмів, ґрунтованих на обчисленні прототипів-центроїдів. З розвитком сучасних інформаційних технологій в класичному інтелектуальному аналізі даних з'явилася необхідність роботи в online режимі, коли дані на обробку поступають послідовно. Такого роду завдання вирішуються у рамках напряму, який дістав назву Data Stream Mining. У подібній ситуації стандартний метод K-середніх неефективний, оскільки предзначен для роботи в пакетному режимі і заздалегідь відомою кількістю класів на яке необхідно класифікувати дані. У такому разі з успіхом можуть бути використані кластеризуючі нейронні мережі Т. Когонена (SOM - Kohonen's Self - Organizing Maps) [5], що вирішують задачу в online режимі.

У випадку перетинаючихся та неопуклих кластерів рішення про належність вхідного образу до одного з кластерів, що приймається згідно до правила «переможець отримує все», може давати неточне кластерування. Це зумовлено тим, що деякі образи можуть належати в той самий час до декількох кластерів з певним ступенем належності. Саме у зв'язку з цим доцільно забезпечити SOM можливості і властивості нечіткого кластерування.

У зв'язку з цим, у цій атестаційній роботі розглядається нейро-фаззі самоорганізовна мапа для пошуку глобального екстремума.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ФОРМАЛІЗОВАНА ПОСТАНОВКА ЗАДАЧІ

Штучні нейронні мережі (ШНМ), що являють собою один із напрямів обчислювального інтелекту – це потужний інструмент для вирішення широкого кола завдань інтелектуального аналізу даних. З плином часу ця методологія активно і успішно використовувалася дослідниками в різних сферах діяльності людини [6].

Важливою особливістю нейронних мереж є їх здатність до навчання чи самонавчання. Під процесом навчання розуміють адаптацію параметрів та архітектури мережі для рішення поставленої задачі шляхом оптимізації обраного критерію якості. Зазвичай до параметрів, що адаптуються відносять синаптичні вагові коефіцієнти нейронів, рідше – архітектуру мережі та параметри активаційної функції.

Всі алгоритми навчання можна поділити на дві групи: детерміновані та стохастичні. Для алгоритмів першої групи характерною особливістю є наявність жорстких правил дій при адаптації параметрів, а для другої притаманні дії, що підкоряються деякому випадковому процесу. На поточний час відомо три парадигми навчання нейронних мереж, в основу яких покладено особливості машинного навчання: навчання з вчителем (supervised learning), навчання без вчителя (unsupervised) (або навчання на основі самоорганізації (self-organized)) та навчання з підкріпленням.

На сьогоднішній день розроблено безліч типів штучних нейронів, архітектур ШНМ з різною топологією, алгоритмів навчання, що дозволяють робити налаштування синаптичних ваг за короткі проміжки часу і обробляти великі об'єми даних в потоковому режимі, постійно «донавчатися» на даних, що надходять, при цьому традиційні нейронні мережі в загальному випадку є системою типу «чорний ящик» і не дозволяють інтерпретувати отримані результати.

Метою даного дослідження є розробка методу кластерування на основі

саморганізовної мапи Когонена, яка налаштовує свою архітектуру в процесі самонавчання, в умовах пересічних кластерів довільної форми.

Для досягнення поставленої мети необхідно буде вирішити ряд завдань:

- проаналізувати базові поняття проблемної області штучних нейронних мереж;
- розглянути існуючі методи навчання-самонавчання;
- проаналізувати базові поняття нечіткої логіки і принципи роботи з нечіткими множинами даних;
- розробити глибинну нейро-фаззі систему послідовного кластерування на основі саморганізовної мапи Когонена;
- вирішувати за допомогою розробленої архітектури тестові і реальні задачі різної фізичної природи.

### 1.1 Поняття штучної нейронної мережі та її складових

Людський мозок можна описати як біологічну нейронну мережу – взаємопов'язану мережу нейронів, що передає складні схеми електричних сигналів. Дендрити отримують вхідні сигнали і, ґрунтуючись на цих входах, подають вихідний сигнал через аксон [6].

Штучна нейронна мережа – це парадигма обробки інформації, яка базується на біологічних системах, таких як мозок, інформація про процес. Ключовим елементом цієї парадигми є нова структура системи обробки інформації. Вона складається з великої кількості взаємопов'язаних елементів (нейронів), що працюють для вирішення конкретних завдань. Штучна нейронна мережа вчиться на прикладах. Вона налаштовується для конкретного завдання, такі як кластерування або класифікація даних, за допомогою навчального процесу. Навчання в такого роду системах включає в себе коригування синаптичних зв'язків, що існують між нейронами.

Нейронні мережі мають здатність витягувати релевантну інформацію зі складних або неточних даних і можуть використовуватися для отримання

шаблонів і виявлення тенденцій, які занадто складні, щоб їх помітили люди або інші машинні методи. Навчену нейронну мережу можна розглядати як «експерта» в області, яка піддавалася аналізу. Потім отримані знання можуть бути використані для надання прогнозів з урахуванням нових ситуацій і відповідей на питання «що, якщо».

На ряду з цим є й інші переваги, такі як:

- адаптивне навчання: здатність навчитися виконувати завдання на основі даних, наданих для навчання або початкового досвіду;
- самоорганізація: мережа може створити свою власну організацію або подання інформації, яку вона отримує під час навчання;
- операції в режимі реального часу: обчислення мережі можуть виконуватися паралельно, розробляються і виробляються спеціальні апаратні пристрої, які використовують цю можливість;
- стійкість до відмов за допомогою надлишкового інформаційного кодування.

Часткове руйнування мережі призводить до відповідного зниження продуктивності. Однак деякі мережеві можливості можуть зберігатися навіть при великому мережевому пошкодженні.

Базовим складовим штучної нейронної мережі є нейрон, головна функція якого, формувати вихідний сигнал, в залежності від сигналів, що надходять на його входи. В найпоширенішій конфігурації штучного нейрона вхідні сигнали множаться на певні ваги і обробляються адаптивним сумматором, потім вихідний сигнал суматора надходить в перетворювач, де перетворюється активаційною функцією, і результат подається на вихід (в точку розгалуження).

Активаційна функція, яку також називають характеристичною – це нелінійна функція, що обчислює вихідний сигнал формального нейрона.

Часто використовувані активаційні функції:

- жорстка порогова функція;
- лінійний поріг;
- сигмоїдальна функція.

На рисунках 1.1–1.3 наведені приклади активаційних функцій.

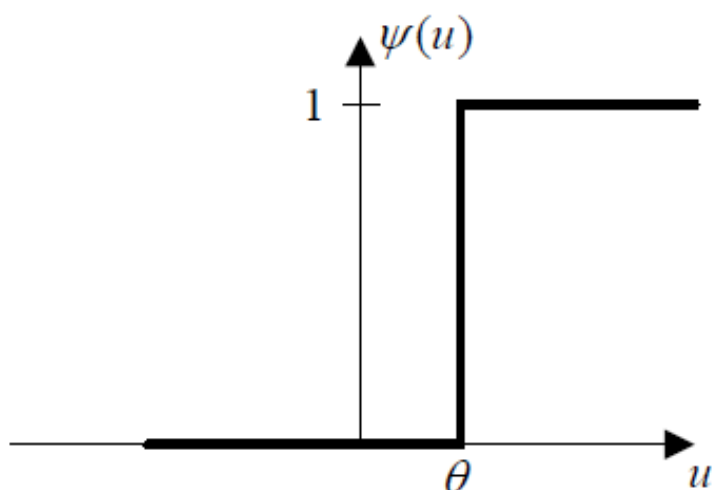


Рисунок 1.1 – Порогова функція

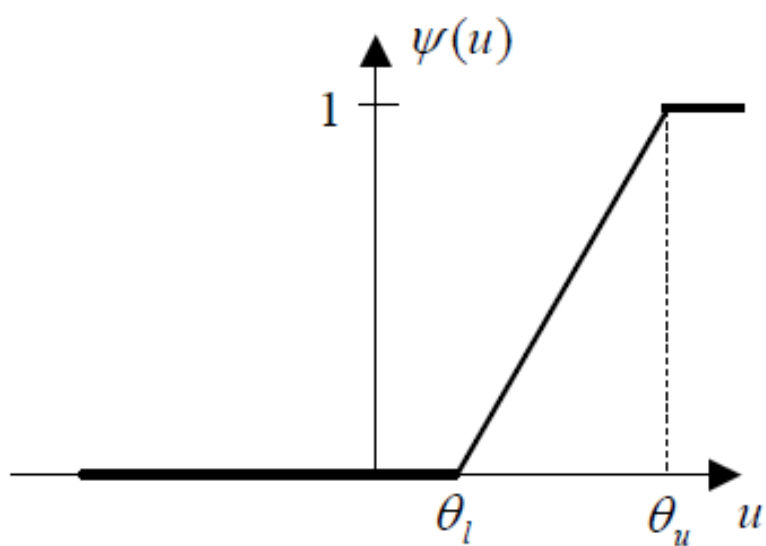


Рисунок 1.2 – Лінійно–порогова функція

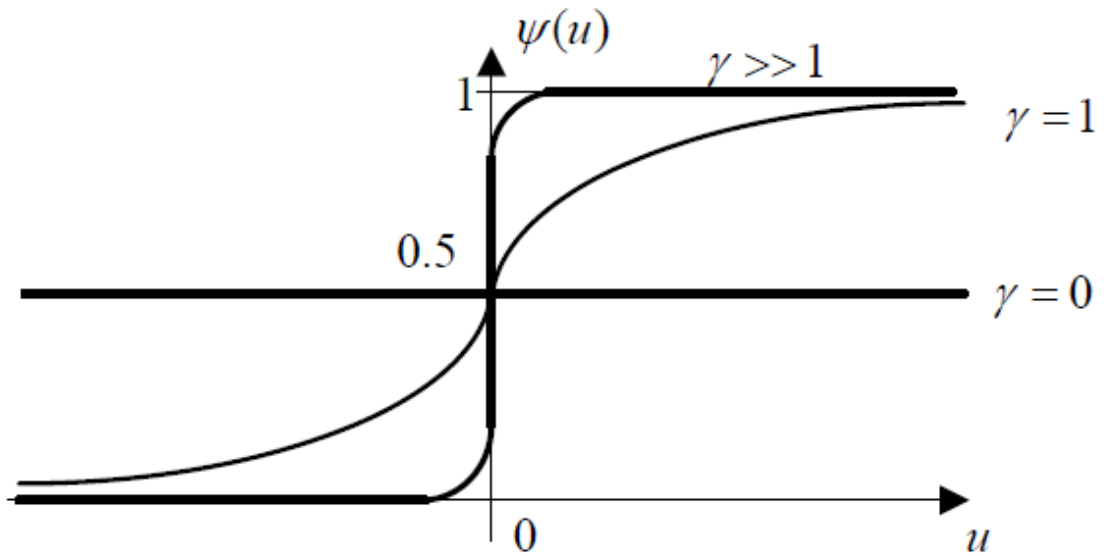


Рисунок 1.3 – Сигмоїдальна функція

Вибір активаційної функції визначається специфікою поставленого завдання або обмеженнями, що накладаються деякими алгоритмами навчання. На рисунку 1.4 наведена загальна схема нейрона.

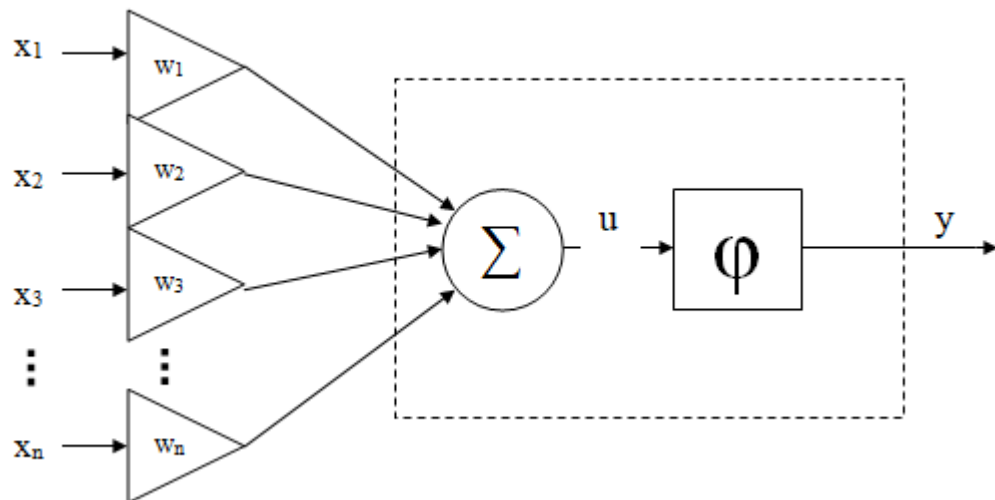


Рисунок 1.4 – Схема нейрона

Групи нейронів в свою чергу формують шари нейронної мережі.

Найбільш поширений тип штучної нейронної мережі складається з трьох шарів: шар «вхідних» нейронів прив'язаний до шару «прихованих» нейронів, який підключений до шару «вихідних» нейронів.

Вхідний шар являє собою вихідну інформацію, яка подається в мережу. Кожен нейрон прихованого шару визначається діями блоків введення і вагами на з'єднаннях між вхідним і прихованим шарами. Поведінка вихідного шару залежить від активності прихованих шарів і ваг між прихованими і вихідними шарами .

На рисунку 1.5 представлена найпростіша штучна нейронна мережа.

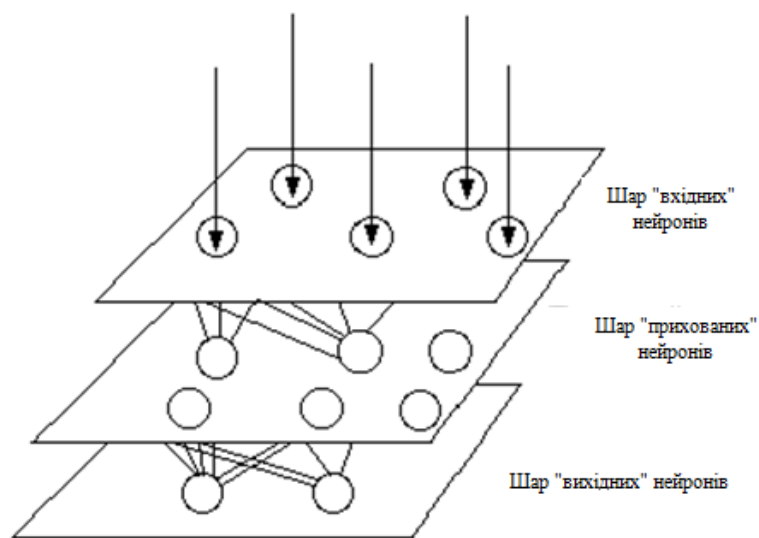


Рисунок 1.5 – Найпростіша штучна мережа

## 1.2 Навчання штучних нейронних мереж

Однією з основних особливостей інтелектуальних систем є їх здатність проводити узагальнення, ґрунтуючись на вже наявних і нових даних, або – іншими словами – їх здатність до навчання [7].

Навчання – це процес, в якому вільні параметри нейронної мережі налаштовуються за допомогою моделювання середовища, в яку ця мережа вбудована. Тип навчання визначається способом підлаштування цих

параметрів. Якщо ж говорити з точки зору математики, то навчання нейронних мереж можна визначити як параметричну задачу нелінійної оптимізації.

Дотепер не існує такого універсального алгоритму навчання, який підходив би для всіх нейронних мереж. Існують три парадигми навчання, на підставі яких проводиться налаштування параметрів нейронних мереж:

- навчання з учителем (supervised learning). При застосуванні цієї парадигми вважається, що для кожного вектора  $x(k)$  з заданої навчальної вибірки вже відомий бажаний результат  $y(k)$ . Таким чином, наприклад, навчається перцептрон. Налаштування параметрів мережі здійснюється з використанням навчального вектора і сигналу помилки. Одним з недоліків цієї парадигми є те, що для навчання мережі не завжди є достатня кількість образів з відповідями;

- навчання без учителя (unsupervised learning). При застосуванні цієї парадигми вважається, що бажані результати роботи нейронної мережі не відомі, а алгоритм навчання налаштовує ваги самостійно. Важливо відзначити те, що, на відміну від навчання «з учителем», в даному випадку вхідні образи, за якими проводиться навчання нейронної мережі, не марковані, тобто для них не заданий бажаний вихід системи. При цьому підході здійснюється пошук закономірностей в навчальній вибірці і проводиться групування схожих вхідних векторів за неявними ознаками. Відмінною особливістю цієї парадигми є наявність міри якості представлення, якому повинна навчитися система, а параметри нейронної мережі повинні оптимізуватися по відношенню до цієї міри. Одним із прикладів застосування цієї парадигми є алгоритм навчання Когонена;

- навчання «з підкріпленням» (навчання методом критики, reinforcement learning). При застосуванні цієї парадигми вважається, що є можливість оцінювати коректність роботи мережі і вказувати бажаний напрям навчання, тобто формувати сигнал підкріплення. Цей підхід можна позначити як проміжний між навчанням «з учителем» і навчанням «без вчителя».

Архітектура нейронних мереж пов'язана з алгоритмами, які використовуються для її навчання. Відповідно до класифікації, даної Хайкіним, виділяють 3 основні класи архітектур нейронних мереж:

- одношарові мережі прямого поширення. Вони містять єдиний шар нейронів. Інформація при цьому передається від вхідного шару на вихідний;
- багатошарові мережі прямого поширення. Вони містять один або кілька прихованих шарів. Зазвичай нейрони кожного з шарів мережі в якості вхідних сигналів використовують тільки вихідні сигнали нейронів попереднього шару;
- рекурентні мережі. Їх відмінною особливістю є наявність однієї або декількох зворотних зв'язків.

Також виділяють повнозв'язні нейронні мережі (коли всі вузли одного шару з'єднані з усіма вузлами суміжних шарів) і неповнозв'язні (коли деякі синаптичні зв'язки відсутні);

Згідно до Хайкіна існує 5 основних моделей навчання:

- навчання на основі корекції помилок (реалізація методу оптимальної фільтрації);
- навчання з використанням пам'яті (з явним використанням навчальних даних);
- Хеббовське навчання (ця інформація базується на нейробіологічних принципах);
- конкурентне навчання (також засноване на нейробіологічних принципах);
- навчання за методом Больцмана (ця інформація базується на ідеях статистичної механіки).

Одним з найбільш поширених є навчання, засноване на корекції помилок. Воно використовує спеціальний механізм управління: до синаптичних ваг нейрона застосовується послідовність коригувань з метою наблизити вихідний сигнал системи до бажаного.

Коригування залежить від отриманого сигналу помилки, що видно з формули 1.1:

$$e(k) = y(k) - \hat{y}(k), \quad (1.1)$$

де  $k$  – поточний дискретний час або порядковий номер спостереження з навчальної вибірки,

$y(k)$  – бажаний вихід нейрона,

$\hat{y}(k)$  – фактичний вихідний сигнал нейрона.

Ще однією часто використовуваною моделлю навчання є конкурентне навчання. Як випливає з назви, в даному випадку в нейронній мережі вихідні нейрони конкурують між собою за «право» бути активізованими. На відміну від нейронної мережі, заснованої на навчанні Хебба, в кожен момент часу в конкурентній мережі тільки один нейрон може бути активним. Його називають нейроном–переможцем, а принцип конкурентного навчання зазвичай формулюється як «переможець отримує все» (winner takes all).

### 1.3 Нечітка логіка та нейро-фаззі системи

Однією з актуальних проблем сучасної науки є побудова і використання систем, робота яких заснована на моделі міркування людини. Лотфі А. Заде здійснив значне просування в цьому напрямку. Він ввів поняття нечіткої множини (fuzzy set), для якого функція належності елемента множини (характеристична функція) може приймати значення в інтервалі  $[0, 1]$ . Він визначив ряд операцій над нечіткими множинами, а також ввів поняття лінгвістичної змінної, значеннями якої є нечіткі множини. Таким чином, його послідовники заснували нову теорію і створили апарат для опису інтелектуальної діяльності, що включає невизначеність і нечіткість [8].

Формально чітку множину  $A_c$  з універсальної множини  $E$  можна описати як безліч впорядкованих пар (формула 1.2)

$$A_c = \{\mu_{A_c}(x)/x\} \quad (1.2)$$

де  $\mu_{A_c}(x)$  – характеристична функція, яка приймає значення 1, якщо  $x$  задовольняє деяку властивість  $R$ , або 0 – в іншому випадку.

Аналогічно нечітку підмножину  $A_f$  з універсальної множини  $E$  можна визначити як безліч впорядкованих пар (формула 1.3)

$$A_f = \{\mu_{A_f}(x)/x\} \quad (1.3)$$

де  $\mu_{A_f}(x)$  – функція належності, що приймає значення із множини належностей  $M = [0,1]$ .

Слід зазначити, що характеристична функція належності визначає ступінь належності елемента  $x$  нечіткій підмножини  $A$ .

На основі правил нечіткої логіки будуються системи нечіткого логічного висновку.

Умовно нечіткий логічний висновок можна розбити на 4 етапи:

- введення нечіткості (фазифікації). На цьому етапі функції належності, визначені на вхідних змінних, застосовуються до їх фактичних значень з метою визначити ступінь істинності кожної передумови кожного правила;

- логічний висновок. На цьому етапі обчислене значення істинності для передумов кожного правила застосовується до висновків кожного правила. Кожній змінній виведення для кожного правила буде призначена нечітка підмножина. Як правила логічного висновку зазвичай використовуються операції  $\min$  (мінімуму) або  $\text{prod}$  (множення). Для  $\min$  функція належності виведення «відсікається» по висоті, відповідного обчислювального ступеня істинності передумови правила (нечітка логіка «і»). Для  $\text{prod}$  функція належності виведення масштабується за

допомогою обчисленої ступеня істинності передумови правила;

- композиція. На цьому етапі всі нечіткі підмножини, призначені до кожної змінної виводу, об'єднуються з метою сформувати одну нечітку підмножину для всіх змінних виведення. При такому об'єднанні зазвичай використовуються операції  $\max$  (максимуму) або  $\text{sum}$  (суми). При  $\max$  комбінований висновок нечіткої підмножини конструюється як покроковий максимум за всіма нечітким підмножинами (нечітка логіка «або»). При  $\text{sum}$  комбінований висновок нечіткої підмножини формується як покрокова сума за всіма нечіткими підмножинами, призначеними змінній виводу правилами логічного висновку;

- приведення до чіткості (дефазифікація). Проводиться, якщо потрібно перетворити нечіткий набір висновків в точну цифру.

Основною перевагою нечітких множин є можливість лінгвістичної інтерпретації. А саме, використовуючи нечіткі множини, вдається легко визначати різні лінгвістичні терміни, які можуть приймати нечіткі значення. Кожне з цих значень описується своєю функцією належності:  $\mu_1$ ,  $\mu_2$ ,  $\mu_3$  відповідно. Можливі значення цих функцій показані на рисунку 1.6.

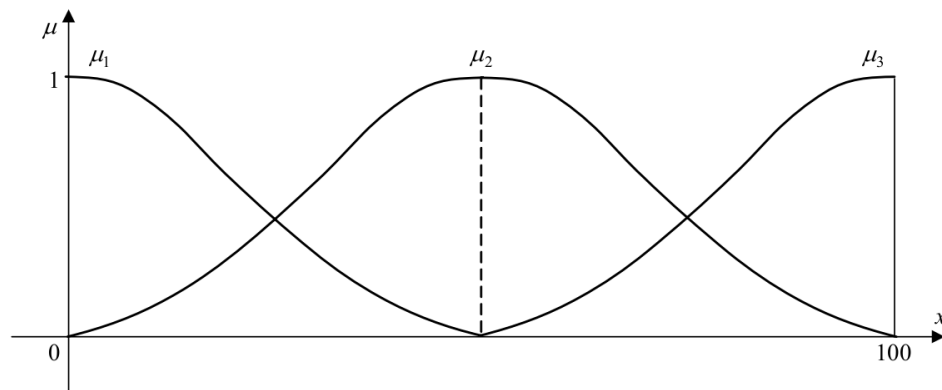


Рисунок 1.6 – Функції належності лінгвістичної змінної

Найбільшого поширення набула гаусова функція належності наведена в формулі 1.4:

$$\mu_A(x) = \exp\left(-\frac{(x-c_A)^2}{2\sigma_A^2}\right), \quad (1.4)$$

де  $c_A$  – центр функції;

$\sigma_A$  – параметр «ширини» функції.

Серед переваг застосування апарату нечітких множин та нечіткої логіки можна також виділити наступні:

- умови і методи розв'язання задачі можуть бути описані мовою, близькою до природньої;
- системи, засновані на нечіткій логіці, є універсальними: відповідно до теореми про нечітку апроксимацію (fuzzy approximation theorem), будь-яка математична система може бути апроксимована системою, заснованою на нечіткій логіці;
- системи, засновані на нечіткій логіці, є ефективними, що доведено переліком теорем, які аналогічні теоремам про повноту для штучних нейронних мереж.
- разом з тим, у нечітких систем можна виділити і певні недоліки:
- набір нечітких правил може виявитися неповним або суперечливим;
- функції належності, їх вид і параметри можуть не цілком відображати реальну дійсність, оскільки вони вибираються суб'єктивно.

Для того щоб усунути зазначені недоліки, нечіткі системи можна зробити адаптивними, тобто нечіткі правила і параметри функцій належності будуть налаштовуватися по мірі роботи такої системи. Одними з найпоширеніших і ефективних варіантів таких систем є гібридні нейронні системи.

Гібридні нейронні системи за своєю структурою схожі на багат шарові нейронні мережі, але приховані шари в них відповідають етапам

функціонування нечіткої системи. У загальному вигляді перший шар гібридних нейронних систем вводить нечіткість на основі заданих функцій належності. Далі вводиться сукупність нечітких правил. І, нарешті, останні шари приводять результат до чіткої відповіді.

Слід відзначити, що кожен з представлених шарів має набір параметрів, що підлягають налаштуванню. До цих параметрів відносять функції належності, активаційні функції, а також синаптичні ваги.

#### 1.4 Загальна постановка задачі кластерування. Застосування нейронних мереж для кластерування масивів

Кластерування відрізняється від класифікації тим, що для проведення аналізу потрібно мати виділену цільову змінну, з цієї точки зору відноситься до класу *unsupervised learning*. Це завдання вирішується на початкових етапах дослідження, коли про дані мало що відомо. Його рішення допомагає краще зрозуміти дані, і з цієї точки зору завдання кластеризації є описовим завданням.

Для етапу кластерування характерна відсутність будь-яких відмінностей як між змінними, так і між записами. Навпаки, метою є знайти групи найбільш близьких, схожих записів. Методи автоматичного розбиття на кластери рідко використовуються самі по собі, просто для отримання груп схожих об'єктів. Аналіз тільки починається з розбиття на кластери. Після визначення кластерів використовуються інші методи *Data Mining*, для того щоб спробувати встановити, значення такого розбиття на кластери і чим воно викликане.

Значною перевагою кластерного аналізу є те, що він дозволяє зробити розгрупування об'єктів не за одним параметром, а за цілим набором ознак. Крім того, кластерний аналіз, на відміну від більшості математико-статистичних методів, не накладає ніяких обмежень на вид розглянутих об'єктів і дозволяє розглядати безліч вихідних даних довільної природи.

Завдання кластеризації полягає в поділі досліджуваної множини об'єктів на групи «схожих» об'єктів, які називаються кластерами.

Необхідно відзначити ряд особливостей, властивих задачі кластерування. По-перше, рішення сильно залежить від природи об'єктів даних і їх атрибутів. По-друге, результат залежить також і від представлення класів (кластерів) і передбачуваних відносин об'єктів даних і класів. Також, необхідно враховувати такі властивості, як можливість або неможливість належності об'єктів окремих класів. Є потреба визначення самого поняття належності класу: однозначна (належить чи не належить), імовірнісна (ймовірність належності), нечітка (ступінь належності).

Формально задачу кластерування можна описати наступним чином. Дано безліч об'єктів даних  $I$ , кожен з яких представлений набором атрибутів. Потрібно побудувати мнодину кластерів  $C$  і відображення  $F$  множини  $I$  на множини  $C$ , тобто  $F: I \rightarrow C$ . Відображення  $F$  задає модель даних, що є рішенням поставленого завдання. Якість рішення задачі визначається кількістю вірно класифікованих об'єктів даних.

Логічним розвитком класичних методів кластерування стали методи на основі штучних нейронних мереж. В даній роботі буде розглянутий метод послідовного кластерування на основі самоорганізовної мапи Когонена. Нейронні мережі Когонена або самоорганізованні мапи Когонена призначені для вирішення задач класифікації без вчителя.

У класичному випадку – це двошарова нейронна мережа, яка містить вхідний шар і шар Когонена (шар активних нейронів). Шар Когонена 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.

На відміну від методу  $k$ -середніх нейронні мережі Когонена дозволяють вирішувати задачу кластерування в послідовному режимі, але все-одно вимагають апріорно задавати число класів, на якому необхідно провести

кластерування.

Таким чином, в даній роботі розглядається задача послідовного кластерування даних в умовах апріорно невідомої кількості кластерів.

Метою магістерської атестаційної роботи є створення глибинної кластерувальної нейро-фаззі системи та її послідовне самонавчання.

Об'єктом дослідження є процес обробки даних, що надходять одне за одним, в послідовному режимі за допомогою нечітких самоорганізованих мап Т. Когонена.

Предметом дослідження є методи потокового кластерування з будь-якою кількістю кластерів в задачах інтелектуального аналізу даних.

Дослідницька робота була виконана у рамках науково-дослідної держбюджетної теми №326 «Глибинні гібридні системи обчислювального інтелекту для аналізу потоків даних та їх швидке навчання».

Апробація основних результатів атестаційної роботи були представлені і обговорені на XXIII Міжнародному молодіжному форуму – Радіоелектроніка та молодь у XXI столітті [10] та V Міжнародній науково-практичній конференції – Обчислювальний інтелект (Результати, проблеми, перспективи) (Ужгород, 2019) [11].

### 1.5 Постановка завдання дослідження

Оскільки, як зазначалося раніше, сучасні обчислювальні технології дозволяють накопичувати і обробляти досить великі масиви інформації, то на перший план виходить швидкість оброблення даних, а також можливість роботи з ними в послідовному режимі.

Для роботи з такими даними найбільш ефективними представляються штучні нейронні мережі, нейро-фаззі системи, гібридні системи, відомі своїми універсальними апроксимуючими властивостями і здатністю до навчання, під яким зазвичай розуміється можливість налаштування їх параметрів шляхом оптимізації деякого критерію якості (цільової функції, критерію навчання) . У

більш широкому сенсі налаштовувати можна не тільки параметри, але і архітектуру системи.

Крім того варто зазначити, що інформація, яка обробляється, може характеризуватися нелінійним і нестационарним характером даних. Застосування апарату нечіткої логіки дозволяє розширити функціональні можливості штучних нейронних мереж і коло вирішуваних завдань. Завдання магістерської роботи зводиться до послідовного нечіткого кластерування даних для пошуку глобального екстремума.

Для досягнення поставленої мети необхідно розглянути наступні питання:

- провести аналіз існуючих методів обчислювального інтелекту для вирішення завдання послідовного кластерування потоків даних;
- проаналізувати існуючі проблеми в даній проблемній області і визначити шляхи їх вирішення;
- розробити систему здатну кластерувати дані за допомогою алгоритму глобального випадкового пошуку та самоорганізовної мапи в online режимі;
- провести імітаційне моделювання та порівняльний аналіз розробленої нейронної системи з існуючими на даний момент алгоритмами.

## **2 НЕЧІТКІ НЕЙРОННІ МЕРЕЖІ І СИСТЕМИ З НЕЧІТКОЮ ЛОГІКОЮ В ЗАДАЧАХ КЛАСТЕРНОГО АНАЛІЗУ**

Термін кластерний аналіз містить у собі набір різних алгоритмів класифікації без вчителя. Загальне питання, що задається дослідниками з різних областей, полягає у тому, як організувати спостережувані дані в наочні структури, тобто розгорнути таксономію. Наприклад, біологи задаються метою розбити тварин на відмінні види, щоб змістовно описати різницю між ними.

Згідно до сучасної системи, що є прийнятою в біології, людина належить до приматів, ссавців, хребетних і тварин. Слід зазначити, що в даній класифікації, чим вище рівень агрегації, тим менше схожості між членами у відповідному класі. Людина має більше схожості з іншими приматами, ніж з «віддаленими» членами сімейства ссавців(наприклад, собаками), тощо.

Техніка кластерування застосовується у найрізноманітніших областях. Хартиган у своїх працях дав змістовний огляд багатьох опублікованих досліджень, що містять результати, отримані методами кластерного аналізу.

Наприклад в області медицини кластерування захворювань чи їх симптомів призводить до широковикористовуваних таксономій. В області психіатрії правильна діагностика кластерів симптомів, таких як параноя, шизофренія і тому подібні, є вирішальною для проведення успішної терапії. В археології за допомогою кластерного аналізу дослідники намагаються встановити таксономію кам'яної зброї чи інших об'єктів старини.

Відомі широкі використання кластерного аналізу у маркетингових дослідженнях. Останні роки кластерний аналіз широко використовується в інтелектуальному аналізі даних і Data Mining, як один з найважливіших методів.

## 2.1 Формалізована постановка задачі кластерування

Кластерний аналіз є математичним методом, призначеним для вирішення завдань класифікації і його мета полягає в поділі емпіричної вибірки на ряд підмножин (зазвичай непересічних), які називаються кластерами, а іноді – групами, класами, таксонами.

Термін «кластер» означає множину елементів, які характеризуються будь-якою загальною властивістю. Йдеться про те, що елементи, об'єднані в один кластер, є в тому чи іншому сенсі більш схожими один на одного, більш близькими за будь-кими характеристиками в порівнянні з іншими. При цьому кластерний аналіз спирається, як правило, тільки на дані про саму емпіричну вибірку і не використовує ніяких додаткових апріорних припущень: наприклад, про характер розподілу ймовірностей у генеральній сукупності. Окрім того, після побудови класифікації її результати вважаються остаточними і не переглядаються для даної емпіричної вибірки даних і застосованого конкретного методу кластерного аналізу. В силу цього кластерний аналіз часто називають класифікацією без навчання (на відміну від дискримінантного аналізу, який називають класифікацією з навчанням).

Метою кластерування можна позначити наступні пункти:

- розуміння даних шляхом виявлення кластерної структури. Розбиття вибірки на групи схожих об'єктів дозволяє спростити подальшу обробку даних і процедуру прийняття рішень, застосовуючи до кожного кластеру свій метод аналізу;
- стиснення даних. Якщо вихідна вибірка надлишково велика, то можна стиснути її, залишивши по одному найбільш типовому представникові від кожного кластера;
- виявлення новизни (novelty detection). Виділяються нетипові об'єкти, які не вдається приєднати до жодного з кластерів.

Вихідною інформацією для класичної задачі кластерування є вибірка даних, сформована з  $N$   $n$ -мірних векторів (образів) ознак  $X = \{x_1, x_2, \dots, x_N\} \subset \mathbb{R}^n$ ,  $x_k \in X$ ,  $k = 1, 2, \dots, N$  – або номер спостереження в таблиці «об'єкт-властивість», або поточний дискретний час у задачах on-line обробки.

Результатом кластерування є розбиття  $X$  на  $m$  класів, що перетинаються з деяким рівнем належності  $U_q(k)$   $k$ -го вектора ознак  $x_k$  до  $q$ -го кластера.

Вихідні дані в процесі обробки центруються і стандартизуються по всіх компонентах так, щоб усі спостереження належали гіперкубу  $[-1, 1]^n$ . Таким чином вихідні дані набувають вигляду  $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N\} \subset \mathbb{R}^n$ , при цьому  $N$  в загальному випадку не обмежено і з плином часу може збільшуватись. Слід зазначити, що традиційно прийняте у мапах Когонена (SOM) перетворення даних до вигляду  $\|\tilde{x}_k\| = 1$  у данному випадку беззмістовно, оскільки якщо  $\tilde{x}_k$  містить пропуск – розрахунок норми такого вектора є неможливим, а якщо  $x_k$  містить викид в одній з компонент –  $\tilde{x}_k$  буде практично збігатися з відповідним ортом простору ознак. Перетворення  $-1 \leq \tilde{x}_{ki} \leq 1$  веде до того, що неспотворені дані концентруються біля нуля, а дані з викидами – біля  $-1$  и  $+1$ .

## 2.2 Метрики, подібність та їх застосування

До безпосередньо кластерування настає фаза яка визначає використання тих чи інших метрик для даних або певні вимірювання для спостережень. Для групування даних потрібен спосіб оцінки елементів та їх відстані відносно один одного, щоб визначити, які елементи належать до тієї чи іншої групи. Це може бути подібністю, хоча в багатьох випадках вимірювання несхожості або «сильніша» відстань [12].

Відстань між будь-якою парою векторів або точок  $i, j, k$  повинна задовольняти наступним властивостям: симетрії,  $d(i, j) = d(j, k)$ ; позитивної визначеності,  $d(i, j) > 0$  і  $d(i, j) = 0$ , якщо  $i = j$ ; і трикутній нерівності,  $d(i, j) \leq d(i, k) + d(k, j)$ .

Якщо трикутна нерівність не виконується, існує можливість

використати несхожості. Нарешті, подібність визначається як  $s(i, j) = \max_{i, j} (d(i, j)) - d(i, j)$ .

Працюючи в векторному просторі традиційним способом вимірювання відстаней є метрика Мінковського, яка є сімейством метрик, визначених як наведено у формулі 2.1:

$$L_p(x_a, x_b) = \left( \sum_{i=1}^n |x_{i,a} - x_{i,b}|^p \right)^{\frac{1}{p}}; \forall p \geq 1, p \in Z^+, \quad (2.1)$$

де  $Z^+$  являє собою набір натуральних чисел.

Манхеттенська, евклідова і чебишівська (максимальної відстані) метрики – особливі випадки метрики відстані Мінковського при  $p = 1$ ,  $p = 2$  і  $p \rightarrow \infty$  відповідно.

Як зразок подібності можна привести косинусну подібність, яка дає кут між двома векторами. Це широко використовується при пошуку в тексті, щоб домогтися відповідності вектору запиту до вибірки. Чим менше кут між вектором запиту та вектором тексту, тим точніший запит до документа. Нормалізована косинусна подібність визначається як наведено у формулі 2.2.

$$s(x_a, x_b) = \cos(\theta) = \frac{x_a * x_b}{\|x_a\| \|x_b\|}, \quad (2.2)$$

де  $x_a * x_b$  – скалярний добуток векторів,

$\|x_a\| \|x_b\|$  – норма вектора.

Існують також інші відповідні метрики, такі як відстань Хелінгера, варіаційна відстань, Махаланобіса та Хеммінга.

При відображенні вхідних даних у евклідовому просторі, де кожен об'єкт є рівнозваженим, існує можливість використовувати евклідову відстань для послідовного кластерування.

### 2.3 Методи кластерного аналізу

Усі відомі методи кластерного аналізу можна умовно поділити на дві великі групи:

- ієрархічні методи;
- неієрархічні методи.

Кожна з груп включає в себе безліч підходів і алгоритмів. Використовуючи різні методи кластерного аналізу, є можливість отримання різних рішень для одних і тих самих даних. Це вважається нормальним явищем.

Ієрархічні методи кластерного аналізу [13], що в свою чергу підрозділяються на агломеративні та дивізімні методи призначені переважно для побудови ієрархічних дерев відносно невеликих за обсягом сукупностей. Іноді їх використовують для задач класифікації і в цьому випадку реалізацію ієрархічного алгоритму продовжують до досягнення кількості класів, яка дорівнює заздалегідь заданому числу  $k$ , або досягненню екстремуму одного з критеріїв якості розбиття.

Перевагами ієрархічних методів є можливість більш повного і тонкого аналізу структури досліджуваної сукупності порівняно з іншими методами, а також наочність подання результатів кластерування.

Їх основними недоліками є громіздкість обчислювальної процедури, яка пов'язана з перерахунком усієї матриці відстаней на кожному кроці, а також «скінчена неоптимальність» гранично оптимальних алгоритмів у багатьох випадках.

Агломеративна група методів характеризується послідовним об'єднанням вихідних елементів і відповідним зменшенням числа кластерів. На початку роботи алгоритму кожен об'єкт є окремим кластером. На першому кроці найбільш схожі об'єкти об'єднуються в кластер. На наступних кроках об'єднання триває до тих пір, доки всі об'єкти не будуть складати один кластер.

Інший варіант дивізімного кластерування полягає, навпаки, у виділенні в окремий кластер елементів, що мають найменші показники схожості, при

тому, що спочатку вся вибірка розглядається як окремий кластер, який на наступних кроках ділиться на менші кластери, в результаті утворюється послідовність розщеплюваних груп.

Ієрархічні алгоритми пов'язані з побудовою дендрограм, які є результатом ієрархічного кластерного аналізу. Дендрограма описує близькість окремих точок і кластерів один до одного, представляє в графічному вигляді послідовність об'єднання (поділу) кластерів.

На рисунку 2.1 приведений принцип роботи описаних вище груп методів у вигляді дендрограми.

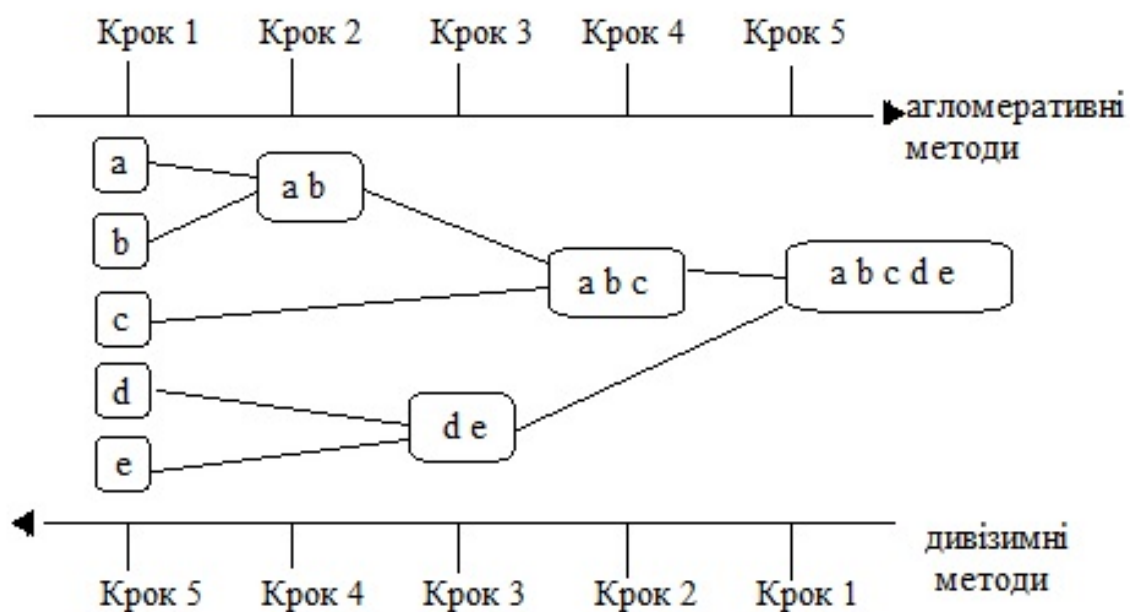


Рисунок 2.1 – Дендрограма агломеративних і дивізімних методів

Ієрархічні методи кластерування розрізняються правилами побудови кластерів. Як правила виступають критерії, які використовуються при вирішенні питання про «схожість» об'єктів при їх об'єднанні в групу (агломеративні методи) або поділу на групи (дивізімні методи).

Агломеративні ієрархічні алгоритми кластерування можна охарактеризувати як жадібні, в алгоритмічному сенсі. Послідовність незворотних кроків алгоритму використовується для побудови необхідної

структури даних.

Припустимо, що пара кластерів, в тому числі можливо, поодиноких, об'єднуються або агломеруються на кожному кроці алгоритму. Тоді наступні є еквівалентним представленням однієї і тієї ж побудованої структури виводу на  $n$  об'єктів: набір  $n-1$  розділів, починаючи з першого розділу, що складається з  $n$  класів і закінчується тривіальним поділом, що складається всього з одного класу, весь набір об'єктів; двійкове дерево (один або два дочірні вузли на кожному нетермінальному вузлі, який зазвичай називають дендрограмою; частково впорядкована множина, яка представляє собою підмножину потужності набору з  $n$  об'єктів; і ультраметрична топологія на  $n$  об'єктів.

### 3 НЕЙРО-ФАЗЗИ САМООРГАНІЗОВНА МАПА ДЛЯ ПОШУКУ ГЛОБАЛЬНОГО

Проблема нечіткого кластерування (Fuzzy clustering problems, F-CP) складається з присвоєння набору шаблонів заданому числу кластерів, щодо деяких критеріїв так, що кожен з них може належати більш ніж одному кластеру з різними ступенями членства. Для того, щоб вирішити цю проблему буде запропонован алгоритм кластерування, який дозволяє знайти глобальний екстремум, завдяки «вистрибуванням» з локальних [14], де околиця визначається всіма можливими переміщеннями центроїда до шаблону. Розчин «цілочисельного» переміщують на безперервну одну по черзі альтернативну стадію, тобто шляхом пошуку центроїди та ступені членства для всіх моделей і кластерів. Для полегшення труднощів застрягання в локальних мінімумах поганого значення, цей локальний пошук потім вбудовується в метаевристичний пошук змінного сусідства.

Однак, окрім проблеми застрягання, виникає ще одна проблема – робота лише у пакетному режимі. Тому, потрібно розробити метод, який буде працювати у послідовому (online) режимі.

Таким чином, необхідно розглянути існуючі методи, виявити недоліки, опрацювати їх. Назамість запропонувати методи які перевершують недоліки. Дойти до ідеального рішення. Отже, в основі метода використані знання о таких алгоритмах:

- k – Means;
- fuzzy C – means;
- j – Means;
- Самоорганізовна мапа Когонена;
- правило «Переможець отримує все»;
- правило «Переможець отримує більше»;
- fuzzy J – means.

Далі, по черзі, розглянемо кожен з алгоритмів.

### 3.1 Алгоритм самоорганізації K-середніх

Розглянемо нейронну мережу з самоорганізацією, де навчання проводиться без наявності вчителя [15].

Алгоритм самоорганізації відносить вектор  $x$  до відповідного кластеру даних, які заявлені центром  $C_i$  і використовують конкурентне навчання.

Базова форма алгоритму самоорганізації дозволяє точно знайти положення центрів  $C_i$  відповідних груп даних(кластерів), на які розбивається вихідних багатовимірний простір. Дані центри далі можуть використовуватися в гібридному алгоритмі навчання нечітких штучних мереж в якості початкових значень, що суттєво прискорює процес навчання і гарантує сходження до глобального мінімуму.

Одним з найбільш поширених алгоритмів кластерування є алгоритм k-середніх (k-means), який також називають швидким кластерним аналізом. Для можливості використання цього методу необхідно мати інформацію про найбільш ймовірну кількість кластерів. Алгоритм k-середніх будує k кластерів, розташованих на можливо великих відстанях один від одного.

Основний тип задач – наявність припущень (гіпотез) щодо числа кластерів, при цьому вони повинні відрізнятися настільки, наскільки це можливо. Вибір числа k може базуватися на результатах попередніх досліджень, теоретичних міркуваннях або інтуїції.

Маємо масив спостережень, кожен з яких має певні значення по ряду ознак. Відповідно до цих значень об'єкт розташовується у багатовимірному просторі:

- дослідник визначає кількість кластерів, що необхідно утворити;
- випадковим чином обирається k спостережень, які на цьому кроці вважаються центрами кластерів;
- кожне спостереження «приписується» до одного з n кластерів — того, відстань до якого найкоротша;
- розраховується новий центр кожного кластера як елемент, ознаки якого

розраховуються як середнє арифметичне ознак об'єктів, що входять у цей кластер;

- відбувається така кількість ітерацій (повторюються кроки 3-4), поки кластерні центри стануть стійкими (тобто при кожній ітерації в кожному кластері опинятимуться одні й ті самі об'єкти), дисперсія всередині кластера буде мінімізована, а між кластерами — максимізована.

Вибір кількості кластерів відбувається на основі дослідницької гіпотези. Якщо її немає, то рекомендують створити 2 кластери, далі 3,4,5, порівнюючи отримані результати.

Принцип алгоритму полягає в пошуку таких центрів кластерів та наборів елементів кожного кластера при наявності деякої функції, що виражає якість поточного розбиття множини на  $k$  кластерів, коли сумарне квадратичне відхилення елементів кластерів від центрів цих кластерів буде найменшим як наведено у формулі 3.1:

$$V = \sum_{i=1}^k \sum_{x_j \in S_i} (x_j - \mu_i)^2, \quad (3.1)$$

де  $k$  – число кластерів,  $S_i$  – отримані кластери,  $i = 1, 2, \dots, k$ ,  $\mu_i$  – центри мас векторів  $x_i \in S_i$ .

В початковий момент роботи алгоритму довільним чином обираються центри кластерів, далі для кожного елемента множини, ітеративно обраховується відстань від центрів з приєднанням кожного елемента до кластера з найближчим центром. Для кожного з отриманих кластерів обчислюються нові значення центрів, намагаючись при цьому мінімізувати функцію після чого повторюється процедура перерозподілу елементів між кластерами.

Але у цього алгоритма є недолік, він працює лише при чітко подільних кластерах.

### 3.2 Алгоритм кластерування Fuzzy C-means

Алгоритм нечіткої кластеризації називають FCM-алгоритмом (Fuzzy Classifier Means, Fuzzy C-Means). Метою FCM-алгоритму кластеризації є автоматична класифікація множини об'єктів, які задаються векторами ознак у просторі ознак. Алгоритм визначає кластери і відповідно класифікує об'єкти. Кластери представляються нечіткими множинами, і, крім того, межі між кластерами також є нечіткими [16]. FCM-алгоритм кластеризації припускає, що об'єкти належать усім кластерам з певною функцією належності. Ступінь належності визначається відстанню від об'єкта до відповідних кластерних центрів. Даний алгоритм ітераційно обчислює центри кластерів і нові ступені належності об'єктів. Задача кластерування за допомогою алгоритму FCM для кількісних характеристик вирішується шляхом мінімізації цільової функції як наведено у формулі 3.2:

$$Q = \sum_{i=1}^c \sum_{j=1}^N w_{i,j}^\beta \|x_j - v_i\|^2, \quad (3.2)$$

при обмеженнях наведених у формулі 3.3:

$$\begin{aligned} w_{i,j} &\geq 0, \forall i = 1, \dots, c; \forall j = 1, \dots, N, \\ \sum_{i=1}^c w_{i,j} &= 1, \forall j = 1, \dots, N, \\ \sum_{j=1}^N w_{i,j} &> 0, \forall i = 1, \dots, c, \end{aligned} \quad (3.3)$$

де  $w_{i,j}$  – рівень належності  $j$ -го спостереження до  $i$ -го кластеру, а  $\beta$  – невідомий параметр фаззифікації.

При цьому рівень належності і прототипи кластерів обчислюються за формулами 3.4 та 3.5:

$$w_{t,j} = \frac{1}{\sum_{i=1}^c \left( \frac{\|x_j - v_t\|}{\|x_j - v_i\|} \right)^{\frac{2}{\beta-1}}}, \quad (3.4)$$

$$\forall t = 1, \dots, c; \forall j = 1, \dots, N;$$

$$v_t = \frac{\sum_{j=1}^N w_{t,j}^\beta x_j}{\sum_{j=1}^N w_{t,j}^\beta}, \quad (3.5)$$

$$\forall t = 1, \dots, c.$$

З формул 3.4 та 3.5 видно, що при обчисленні рівня належності конкретного спостереження до кластеру  $w_{i,j}$  використовується відстань між наглядом и відповідними центроїдами кластеру  $v_i$ . Далі перераховується  $v_i$  на основі рівнів належності до кластерів  $w_{i,j}$ . Обчислення проводяться ітераційно, поки не буде виконана умова зупинки алгоритму.

Ознайомившись з алгоритмом роботи Fuzzy C-means очевидно, що він застряє в локальних екстремумах, тому з'явився алгоритм кластерування J-means, який може впоротись з цією проблемою.

Таким чином, алгоритм k-means та FCM, добре кластерують, але застряють у локальних екстремумах. Тому виник алгоритм J-means – нова евристика локального пошуку, названа J-середніми, для вирішення завдання кластеризації мінімальної суми квадратів. Сусідство поточного рішення визначається всіма можливими переміщеннями центроїда до об'єкта з подальшими відповідними змінами призначень. Рух здійснюється в таких околицях, поки не буде досягнутий локальний оптимум.

### 3.3 Алгоритм кластерування J-means

У деяких випадках існуючі точки можуть бути центроїдами деяких кластерів у поточному рішенні. Будемо називати їх зайнятими. Далі представлені правила евристики J-Means [17]. Для того, щоб отримати сусіднє рушення поточного, центроїд  $x_i$  кластеру  $C_i$  переміщується до місця розташування незайнятого об'єкта і всі сутності  $C_i$  переносяться до свого найближчого центроїда. Всі такі можливі рухи утворюють «стрибок» сусідства поточного рішення.

Крок перший – ініціалізація.  $P_M = \{C_i\}$ , ( $i = 1, 2, \dots, M$ ),  $x_i$ , ( $i = 1, \dots, M$ ),

де  $M$  – кількість кластерів та  $f_{opt}$  є початковим розбиттям множини  $X$ , відповідних центроїдів і струму значень об'єктивної функції відповідно.

Крок другий – зайняті точки. Потрібно знайти незайняті точки, тобто об'єкти, які не збігаються з центроїдом кластера (в межах невеликого допуску).

Крок третій – «стрибок» до сусідства. Необхідно знайти найкращий розділ  $P_{M'}$  і відповідне значення  $f'$  в околиці «стрибка» поточного рішення  $P_M$ .

Дослідження сусідства. Для кожного  $j$  ( $j = 1, \dots, N$ ) повторити крок переселення, де необхідно додати новий центроїд кластера  $x_{M+1}$  на певному незайнятому розташуванні об'єкта  $x_j$  і знайти індекс найкращого віддалення центроїда, позначили через  $v_{ij}$  зміну в значенні цільової. А також повторити крок зберігання найкращого. Для цього, необхідно взяти пару індексів  $i'$   $j'$ , де  $v_{ij}$  мінімальна. У під кроку, під назвою, переміщення, необхідно змінити центроїди  $x_{i'}$  на  $x_{j'}$  і відповідно оновлювати призначення для отримання нового розбиття  $P_{M'}: f' := f_{opt} + v_{i'j'}$ .

Крок четвертий, може відбуватись два випадки: припинення або переміщення. Якщо  $f' > f_{opt}$  алгоритм зупиняється, тому що локальний мінімум був знайдений на попередній ітерації. В іншому випадку, потрібно перейти до сусіднього рішення  $P_{M'}$  ( $P_M := P_{M'}, f_{opt} = f'$ ) і повернутись до другого кроку. Слід зазначити, що ефективність евристики J-Means значною мірою залежить від того, що етап переміщення може бути реалізований за час  $O(N)$ .

Але J-Means та K-means абсолютно точні процедури, був створен алгоритм Fuzzy J-means, який повністю базується на алгоритмі Fuzzy C-means та, на відмінність, може подолати застрягання у локальних екстремумах.

### 3.4 Алгоритм кластерування Fuzzy J-means

У цьому підрозділі буде розглянутий алгоритм кластерування Fuzzy J-Means, який базується на локальному J-means. У алгоритмі, так само, рухаються визначаючі сусідства, які побудовані за всіма можливими перенесеннями до центроїда по шаблону. Найпростіша конструкція розглядає

присвоєння одного центроїда будь-якій незайнятій точці (тобто точки, яка не має центроїда, що з нею співпадає). Алгоритм кластерування Fuzzy J-means складається з 6 кроків [18].

Крок перший – ініціалізація. Необхідно навмання або за деякою ознакою конструктивної евристики, початковий набір центроїдів кластера  $V = \{v_1, v_2, \dots, v_c\}$  і  $R$  – відповідне значення знайденої цільової функції. Розрахунок належності і центроїдів обчислюється за формулою 3.6 і 3.7 відповідно:

$$w_{ij}^{(t)} = \begin{cases} \frac{1}{\sum_{q=1}^c \left\{ \frac{\|x_i - v_j^{(t-1)}\|}{\|x_i - v_q^{(t-1)}\|} \right\}^{\frac{2}{m-1}}}, \\ 1, \\ 0, \end{cases} \quad (3.6)$$

$$v_q^{(t-1)} \neq x_i, j = 1, \dots, c, v_j^{(t-1)} = x_i, v_j^{(t-1)} = x_k, k \neq i.$$

$$v_j^{(t)} = \frac{\sum_{i=1}^n (w_{ij}^{(t)})^m x_i}{\sum_{i=1}^n (w_{ij}^{(t)})^m} \quad (3.7)$$

Крок другий – видалення. Потрібно знайти центроїд ( $v^{\text{del}}$ ), який потрібно видалити за допомогою якогось критерію, всі інші центроїди залишаються у своїх поточних місцях розташування. Кожен центроїд видаляється по черзі і значення цільової функції отримуються для решти центроїдів за допомогою формул 3.5 – 3.7. Центроїд для видалення відповідає мінімальному значенню  $R_j$ , тобто за формулою 3.8:

$$v^{\text{del}} = \arg \min_{j=1, \dots, c} \{R_j\}, \text{ де } R_j = R(v - \{v_j\}). \quad (3.8)$$

Крок третій – зайняті точки. Необхідно знайти незайняті точки, тобто точки, які не збігаються з центроїдом кластера в межах невеликого допуску.

Крок четвертий – додавання. Потрібно обрати незайняту точку ( $x^{ins}$ ) відповідно до критеріїв вказаних у формулі 3.9 и отримати нове рішення:

$$V := V \cup \{x^{ins}\} \setminus \{v^{del}\}. \quad (3.9)$$

Далі потрібно вставити вилучений центроїд на незайняту точку. Обчислити значення цільової функції для нового центроїду, отриманий шляхом зміни  $v^{del}$  з кожною незайнятою точкою, як зазначено у формулі 3.10:

$$x^{ins} = \arg \min_{j=1, \dots, n} R(R_j), \text{ де } V_j = V \cup \{x_j\} \setminus \{v^{del}\}. \quad (3.10)$$

Крок п'ятий – локальне поліпшення. Необхідно перемістити чітке рішення (тобто, шляхом виявлення членства  $W$ , як вказано у формулі 3.11 та потім центроїд  $V$  з урахуванням використання належності за формулою 3.6. Далі потрібно оновити відповідне значення цільової функції  $R$ .

$$(\min_V) R_m(V) = \sum_{i=1}^n \left( \sum_{j=1}^c \|x_i - v_j\|^{2(1-m)} \right)^{1-m}, \quad (3.11)$$

$$\|x_i - v_j\| \neq 0, i = 1, \dots, n, j = 1, \dots, c.$$

Крок шостий – припинення або переміщення. Якщо ( $R < R_{opt}$ ) зберегти нове найкраще рішення і його значення, тобто встановити  $V_{opt} := V$  та  $R_{opt} := R$ , необхідно буде повернутись до другого кроку. Інакше розрахувати матрицю належностей  $W_{opt}$  і зупинитись у локальному мінімумі ( $V_{opt}, W_{opt}$ ).

Далі буде зрозуміло яким чином робити «стрибки» з локального екстремума для пошуку глобального.

### 3.5 Алгоритми «стрибку» для пошуку глобального екстремума

Є три основні типи алгоритмів глобального пошуку, а

саме – незалежний, який імітує незалежну випадкову виборку відповідно до заданого закону розподілу. Блукаючий, який імітує багатовимірний блукаючий об’єкт як динамічні системи. Змішаний, який поєднує властивості перших двох типів [19].

Незалежним алгоритмом сліпого глобального пошуку є випадковий перебір який зводиться до наступного. На кожному кроці в залежності с заданою просторною щільністю розподілу  $P(X)$  незалежно й випадкового визначається станом  $X_i$ , вираховується критерій якості у цьому стані  $Q(X_i)$  і отриманне значення зіставляється з береженням у пам’яті  $Q_{i-1}^0$ . Якщо  $Q(X_i) \geq Q_{i-1}^0$ , то здійснюється черговий випадковий експеримент, якщо ж  $Q(X_i) < Q_{i-1}^0$ , то нове значення показника якості  $Q(X_i)$  зафіксується, як и стан  $X_i$ , який привів до зниження функції якості. Після цього робиться дуже рідкісна випадкова проба відповідно з тією самою заданою у просторі параметрів щульністю розподілу випадкових проб  $P(X)$ .

Алгоритм цього пошуку (формула 3.12) може бути зафіксован у вигляді наступної рекурентної формули для вмісту пам’яті:

$$\begin{aligned} X_i^0 &= \begin{cases} X_{i-1}^0, & \text{якщо } Q(X_i) \geq Q_{i-1}^0, \\ X_i, & \text{якщо } Q(X_i) < Q_{i-1}^0; \end{cases} \\ Q_i^0 &= \begin{cases} Q_{i-1}^0, & \text{якщо } Q(X_i) \geq Q_{i-1}^0, \\ Q(X_i), & \text{якщо } Q(X_i) < Q_{i-1}^0, \end{cases} \end{aligned} \quad (3.12)$$

де  $X_i$  – випадковий стан, обраний у відповідності с заданою щільністю розподілу  $P(X)$ ;  $X_i^0$ .

$Q_i^0 = Q(X_i^0)$  – вміст пам’яті на  $i$ -тому кроці пошуку.

Цілком ймовірно, що цей алгоритм гарантує пошук глобального екстремума при достатньо розумному будуванні розподілу  $P(X) \neq 0$  для  $X \in S$  та  $N \rightarrow \infty$ . Блукаючий алгоритм глобального пошуку пов’язан з випадковими «стрибками» у процесі градієнтного пошуку. Виявляється, якщо спеціальним образом організувати момент появи випадкових «стрибків», то при

визначених умовах можливо гарантувати пошук глобального екстремума. Сенс такого пошуку зводиться до наступного.

Нехай процес градієнтної оптимізації у певний момент часу починається з нових початкових умов, визначених заданою щільністю розподілу  $p(X_0)$ , яка не дорівнює нулю по всій області пошуку. Це означає, що час від часу градієнтний пошук починається спочатку з випадкової точки  $X_0$ . Момент цього «стрибка» визначається значенням деякої випадкової скалярної функції  $\eta(t)$ , яка дискретна у часі й рівню  $i$  задається по формулі 3.13:

$$\eta(t+1) = \begin{cases} i (i = 2, \dots, M), & \text{якщо } \eta(t) = 1, \text{ з ймовірністю } P(Q) \\ \eta(t) - 1, & \text{якщо } 1 < \eta(t) \leq M, \text{ з ймовірністю } P(Q) \\ \eta(t) + 1, & \text{якщо } 1 < \eta(t) < M, \text{ з ймовірністю } 1 - P(Q) \\ M, & \text{якщо } \eta(t) = M, \text{ з ймовірністю } 1 - P(Q) \end{cases} \quad (3.13)$$

де  $M$  – задане ціле число,

$q_i$  – задовільні ймовірності з природними обмеженнями, де сума  $q_i$  по  $M$  дорівнює одиниці при  $i=2$ ,

$P(Q)$  функція – суворо монотонно зростаюча в межах  $0 < P(Q) < \delta < 1/2$ , а у свою чергу  $\delta$  – деяке мале число, яке визначається з міркувань сходимості процесу.

Момент «стрибку» настає коли  $\eta(t)$  дорівнює одиниці. Доведено, що при  $t \rightarrow \infty$  та  $M \rightarrow \infty$  має місце сходимость до глобального екстремуму функції  $Q(X)$ , а саме  $X(t) \rightarrow X^{**}$ . На формулі 3.13 зазначено, що випадковий процес  $\eta(t)$  у середньому, має тенденцію прагнути до  $M$ . Якщо число  $M$  невелике, то на кожному кроці, з ймовірністю  $P(Q) < 1/2$  можливо зменшення  $\eta$ , отже, час від часу створюються умови для введення випадкового «стрибка». Ймовірність появи «стрибка» залежить від  $M$  та  $Q$ . При малому  $M$  «стрибки» будуть проходити частіше, ніж при великому. При великому  $Q$ , аналогічно як із  $M$  – «стрибки» бувають частіше, ніж при великому. Отже, з глибоких

екстремумів система «вістрибує» рідше ніж з дрібних. Якщо збільшувати  $M$ , то різниця між глибоким та дрібним екстремумами, куда систему заводять градієнтний пошук, виявиться ще більше. Частота «вістрибувань» система із глибокого екстремума зменшується (так само як і з маленького). Проте з глибокого система буде «вистрибувати» відносно маленького, ще рідше. Наприклад, якщо при  $M=M_1$  ці частоти відносяться як 1:2, то при  $M=M_2 > M_1$  відношення стане 1:5. Очевидно, що при досить великом  $M$  – відношення стане досить малим й застрягне у самому глибокому (глобальному) екстремуму, доведено.

Але процедури Fuzzy J-means Fuzzy та C-means працюють тільки в пакетному режимі, тому існують онлайнві процедури, які вписуються у, так звані, правила самонавчання по Когонену winner takes all та winner takes more, які будуть розглянуті окремо.

### 3.6 Самоорганізованні мапи Т. Когонена

Мережа Когонена відноситься до самоорганізованих мереж. це означає, що вони не отримують бажаний вихідний сигнал при надходженні вхідного навчального вектора, а в результаті навчання мережу розділяє вхідні сигнали на класи, таким чином формуючи топологічні карти [20].

Однією з найважливіших властивостей навченої мережі Когонена є її здатність до узагальнення. Суть роботи мережі Когонена полягає в тому, що відбувається кластерування вхідних векторів в групи схожих між собою векторів. При цьому ваги мережі налаштовуються так, щоб вхідні образи, які належать одному кластеру, активізували один і той ж вихідний нейрон.

Вектор кожного з нейронів мережі Когонена може замінити групу відповідних йому векторів, що класифікуються. Таким чином, ця властивість дозволяє використовувати даний вид мережі в області стиснення даних. Варто відзначити, що самоорганізована мапа Т. Когонена реалізує відображення вхідного простору розмірності  $n$  в вихідний простір розмірності  $m$ .

Самоорганізовна мапа має дуже просту архітектуру з прямою передачею інформації. Крім нульового (рецепторного) шару, вона містить єдиний шар нейронів, який дуже часто називають шаром Когонена.

Саме завдяки такій організації, кожен нейрон мережі отримує всю інформацію по аналізованому образу і генерує на свій вихід відповідний відгук.

Після цього в шарі Когонена виникає режим конкуренції, в результаті якої визначається єдиний нейрон-переможець з максимальним вихідним сигналом. Цей сигнал по латеральним зв'язкам забезпечує збудження найближчих «сусідів» переможця і придушення реакції далеко віддалених вузлів.

Самоорганізовані мапи можуть мати різну топологію. Однак найбільш часто рецептори і нейрони розташовуються у вузлах одно- або двовимірної решітки.

Розглянемо докладніше архітектуру самоорганізовної мапи. На вхід мережі надходить  $n$ -мірний вхідний сигнал. мережа містить єдиний шар з  $m$  нейронів, які утворюють на площині прямокутні решітки. Нейрони характеризуються своїм місцем розташування в мережі. кожен нейрон шару Когонена пов'язаний з кожним входом нульового (вхідного) шару прямими зв'язками, а також з усіма іншими нейронами поперечними (латеральними) зв'язками. На рисунку 3.1 представлена 1D-мапа Когонена.

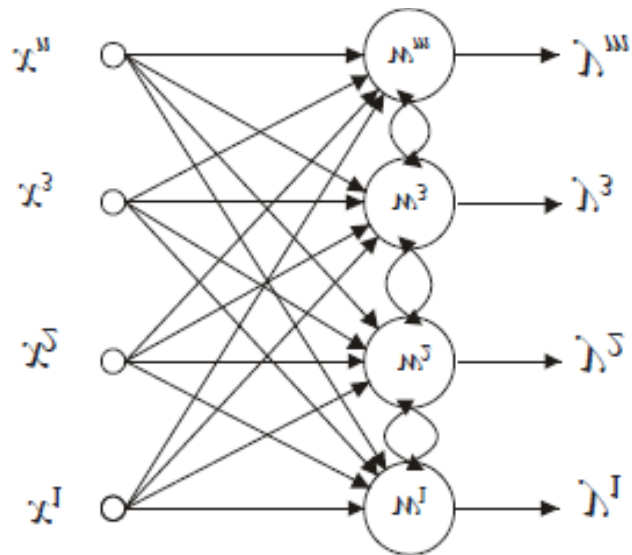


Рисунок 3.1 – 1D-мапа Когонена

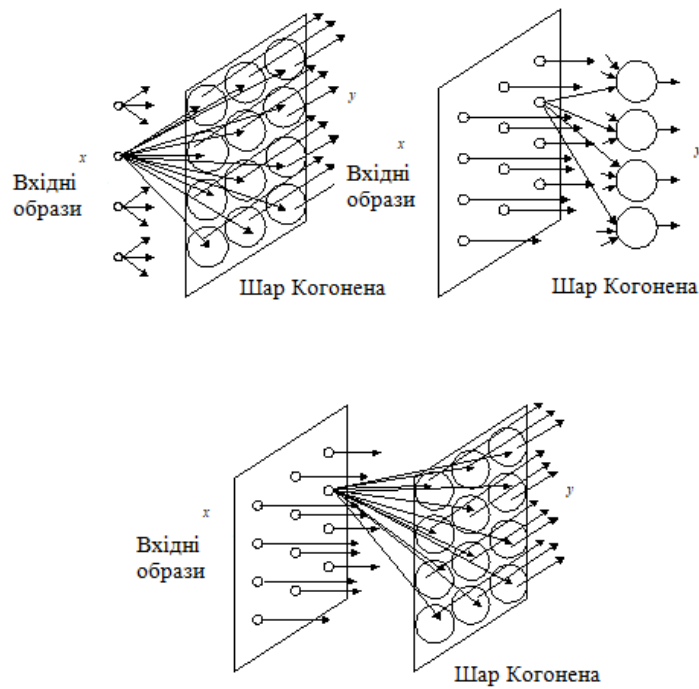


Рисунок 3.2 – Топологія мап Когонена

Самоорганізованні мапи можуть мати різну топологію, хоча найчастіше рецептори і нейрони розташовуються у вузлах одно (1D)- або двовимірної (2D) решітки так як це показано на рисунку 3.2

В основі алгоритму самоорганізації мапи Когонена лежать принципи

конкурентного самонавчання. Як і будь-яка інша процедура навчання, робота алгоритму починається з ініціалізації синаптичних ваг мережі, які зазвичай вибираються за допомогою генератора випадкових чисел.

В процесі навчання сусідні нейрони впливають один на одного сильніше, ніж ті, які розташовані далі. Саме латеральні зв'язки в мережі забезпечують збудження одних нейронів і гальмування інших. Кожен нейрон з шару Когонена формує зважену суму сигналів  $f(x, w) = \sum_{i=1} w_i x_i$ . При цьому, якщо синапси прискорюють, то  $w_{ij} > 0$ . Якщо ж синапси гальмуючі, то  $w_{ij} < 0$ .

Перед початком роботи алгоритму навчання мережі Кохонена вхідні вектори попередньо нормалізуються як показано у формулі 3.14:

$$\tilde{x}_i = \frac{x_i}{\sqrt{\sum_i x_i^2}} = \frac{x_i}{\|x\|}, i = 1, 2, \dots, N \quad (3.14)$$

Сам алгоритм навчання мережі Кохонена може бути описаний як послідовність кроків:

Крок 1: проводиться ініціалізація початкових значень синаптичних ваг  $w_{ij}$  (0). Одним з часто застосовуваних способів ініціалізації є присвоєння синаптичним вагам значень, рівних випадково обраним векторам з безлічі спостережень. В цілому, існує три основних способи ініціалізації початкових ваг:

- ініціалізація випадковими значеннями, коли всім вагам даються малі випадкові величини;
- ініціалізація прикладами, коли в якості початкових значень задаються значення випадково вибраних прикладів з навчальної вибірки;
- лінійна ініціалізація. В цьому випадку ваги ініціюються значеннями векторів, лінійно впорядкованих вздовж лінійного підпростору, що проходить між двома головними власними векторами вихідного набору даних.

Крок 2: на вхід системи подається нормалізований вектор сигналів  $\tilde{x}$  і

вибирається вектор ваг (нейрон), найближчий до  $\tilde{x}$ , тобто вектор, для якого Евклідова відстань до  $\tilde{x}$  буде найменшою (формула 3.15):

$$\arg \min_j \|\tilde{x} - w_j\|, j = 1, 2, \dots, l. \quad (3.15)$$

Цей вираз також може бути записаний у наступному вигляді (формула 3.16):

$$c = \arg \max_j (\tilde{x}^T w_j), j = 1, 2, \dots, l. \quad (3.16)$$

Крок 3: проводиться корекція (налаштування) вектора синаптичних ваг  $w_{ij}$  за правилом (формула 3.17):

$$w_{ij}(k+1) = w_{ij}(k) + \eta(k) f_{ij}(k) (\tilde{x}(k) - w_{ij}(k)), \quad (3.17)$$

де  $w_{ij}(k+1)$  – нове значення ваги;

$\eta(k)$  – коефіцієнт підсилення, що змінюється з плином часу (на першій ітерації  $\eta = 1$  і поступово зменшується до нуля,  $\eta(k) \in (0,1]$ );

$f_{ij}(k)$  – монотонно спадна функція (функція сусідства) виду (формула 3.18):

$$f_{ij}(k) = f(\|r_i - r_j\|, k), \quad (3.18)$$

де  $r_i$  - вектор, що визначає положення  $i$ -го нейрона в решітці;

$r_j$  - вектор, що визначає положення  $j$ -го нейрона в решітці.

Найчастіше в якості функції сусідства обирають Гауссіан, «Мексиканський капелюх», косінусоїдну функцію, функції Епанечнікова і т.п. При цьому вихідні сигнали мережі визначаються як було наведено раніше.

Далі кроки 1 і 2 повторюються до тих пір, поки вихідні значення мережі не стабілізуються із заданою точністю.

Сенс налаштування синаптичних ваг зводиться до того, що відбувається

мінімізація різниці між вхідним вектором  $\tilde{x}_k$  і вектором синаптичних ваг нейрона-переможця.

Іншими словами можна сказати, що цей алгоритм в процесі налаштування «підтягує» вектор синаптичних ваг нейрона переможця до поточного вхідного образу  $\tilde{x}_k$ .

На рисунку 3.3 представлена геометрична інтерпретація налаштування синаптичних ваг для двовимірного випадку.

В даному випадку знаходиться  $\tilde{x}_k - w_{ij}(k)$ , який потім зменшується на величину  $\eta(k)$ , яка задає швидкість навчання.

Таким чином, можна сказати, що навчання зводиться до обертання вектора ваг нейрона в напрямку вхідного вектора без істотної зміни його довжини.

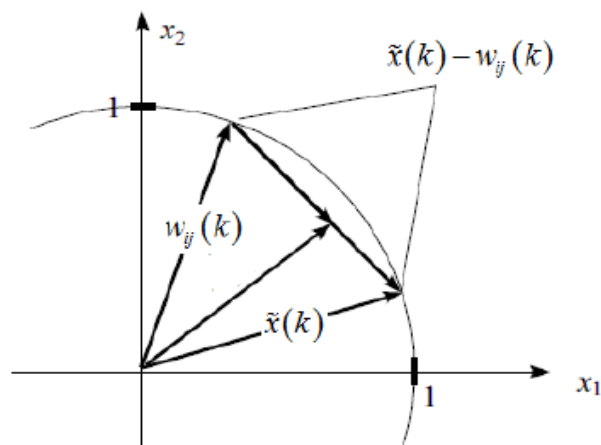


Рисунок 3.3 – Геометрична інтерпретація налаштування ваг мережі Когонена

### 3.7 Конкурентні нейронні мережі

Особливим видом самонавчання є, так зване, конкурентне навчання, коли всі нейрони мережі «змагаються» між собою за право бути активним, реалізуючи принципи «Переможець отримує все» (Winner takes all) та «Переможець отримує більше» (Winner takes more).

Перший – веде до того, що в мережі може активуватися тільки один нейрон. Саме ця особливість конкурентного навчання забезпечила йому широке використання в задачах класифікації та кластерування.

На рисунку 3.4 наведена найпростіша нейронна мережа, що використовує конкурентну схему навчання з одним шаром нейронів.

Особливістю даної мережі є наявність поперечних (латеральних) зв'язків, за якими відбувається змагання, в результаті якого «перемагає» тільки один нейрон.

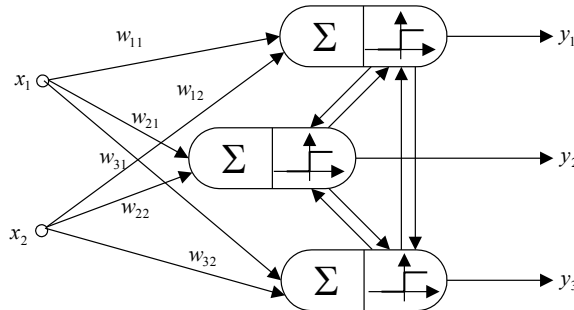


Рисунок 3.4 – Найпростіша конкурентна мережа

Процес конкуренції може бути представлений у формі наведеної у формулах 3.19- 3.20:

$$y_q(k) = \begin{cases} 1, \text{ якщо } w_q^T(k)\tilde{x}_k > w_p^T(k)\tilde{x}_k \text{ для всіх } p \neq q, \\ 0 \text{ в протилежному випадку,} \end{cases} \quad (3.19)$$

$$\tilde{x}_k = \frac{\vec{x}_k}{\|\vec{x}_k\|}, \quad \|w_q(k)\| = 1, \quad (3.20)$$

При цьому в кожен момент часу налаштовується тільки нейрон-переможець за допомогою алгоритму відомого як правило навчання Т. Когонена (формула 3.21):

$$w_q(k+1) = \begin{cases} w_q(k) + \eta(k) (\tilde{x}_k - w_q(k)), & \text{якщо } q \text{ – нейрон переміг} \\ w_q(k) & \text{в протидежному випадку} \end{cases} \quad (3.21)$$

Рисунок 3.5 ілюструє рішення задачі кластерування за допомогою найпростішої конкурентної ШНМ.

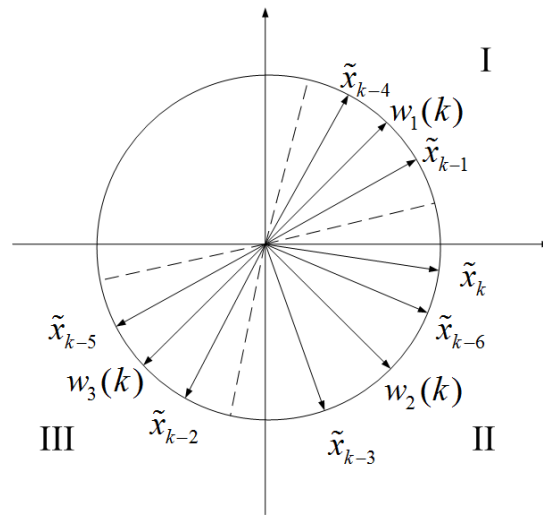


Рисунок 3.5 – Кластерування на основі конкурентного навчання

Можна побачити, що поточний нормований вектор  $\tilde{x}_k$  найближчий до вектора синаптичних ваг  $w_2(k)$ , в результаті чого «перемагає» третій нейронякий і налаштовує свої параметри за допомогою правила Когонена, «підтягуючи»  $w_2(k)$ , до  $\tilde{x}_k$  на відстань пропорційну параметру кроку  $\eta(k)$  (формула 3.22):

$$\begin{cases} w_2^T(k) \tilde{x}_k > w_1^T(k) \tilde{x}_k, \\ w_2^T(k) \tilde{x}_k > w_3^T(k) \tilde{x}_k, \end{cases} \quad (3.22)$$

Конкурентне навчання лежить в основі ряду нейромереж, що набули широкого поширення в задачах обробки інформації та інтелектуального аналізу даних. Правило «Переможець отримує більше» полягає в розбитті послідовності нормованих векторів-образів  $\phi_1, \phi_2, \dots, \phi_N$  на  $m$  кластерів, кожен з яких характеризується власним прототипом-центроїдом  $c_j^k \in R^h, j=1,2, \dots, m$ , безперервно уточнюється при надходженні чергового образу підвищеної розмірності  $\phi_k$ . Процес самонавчання починається з ініціалізації синаптичних ваг вихідного шару, в якості яких виступають довільно вибрані початкові значення прототипів  $c_j^k(0)$ , таки як наведені у формулі 3.23:

$$\|c_j^k(0)\| = 1. \quad (3.23)$$

При подачі на вхід третього шару сигналу  $\phi_1$  обчислюється  $m$  відстань, яка наведена у формулі 3.24:

$$D(\phi_1, c_j^k(0)) = \|\phi_1 - c_j^k(0)\| \forall j = 1, 2, \dots, m, \quad (3.24)$$

на підставі яких оцінюється нейрон переможець, для якого відстань обчислюється по формулі 3.25:

$$D(\phi_1, c_*^k(0)) = \min_j D(\phi_1, c_j^k(0)). \quad (3.25)$$

Після цього проводиться перший крок налаштування ваги центроїдів, як наведено у формулі 3.26:

$$c_l^k(1) = \frac{c_l^k(0) + \eta(1)\psi(c_*^k(0), c_l^k(0))(\phi_1 - c_l^k(0))}{\|c_l^k(0) + \eta(1)\psi(c_*^k(0), c_l^k(0))(\phi_1 - c_l^k(0))\|} \quad (3.26)$$

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

Аналогічним чином можна записали правило самонавчання для  $k$ -го

вектора-образа, за формулою 3.27:

$$c_1^k(1) = \frac{c_1^k(k-1) + \eta(k)\psi\left(c_*^k(k-1), c_1^k(k-1)\right)\left(\varphi_k - c_1^k(k-1)\right)}{\left\|c_1^k(k-1) + \eta(k)\psi\left(c_*^k(k-1), c_1^k(k-1)\right)\left(\varphi_k - c_1^k(k-1)\right)\right\|} \quad (3.19)$$

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

де – так звана, функція сусідства, яка визначає локальну область топологічного сусідства, в якій налаштовується не тільки нейрон переможець  $c_*^k$ , але і його найближче оточення, при цьому більш близькі до переможця нейрони підтягуються за вхідним вектором  $\varphi_k$  більш ніж віддалені від  $c_*^k$  центроїди  $c_1^k$ .

Після розглянення усі існуючих алгоритмів яки знадобилися для створення послідовної нейро-фаззі SOM для оминання локальних екстремумів, тому, що були виявлені усі недоліки, та шляхи подолання недоліків. Отже, можна перейти до заключного підрозділу, та розглянути нейронну мережу для подолання усі проблем оговорених вище.

### 3.8 Online нейро-фаззі самоорганізовної мапи для пошуку глобального екстремума

Під час обробки реальних даних часто виникають ситуації, коли один образ належить різним класам, а самі ці класи взаємно перетинаються. У межах традиційних самоорганізовних мап формально ця обставина ніяк не враховується, однак може бути розглянута з пропозиції нечіткого кластерного аналізу [21].

Доцільно об'єднати простоту і швидкість з можливістю роботі за умов взаємного перетину класів.

Так було запропоновано модифікацію SOM, в якій нейрони початкової архітектури, що є по суті адаптивними лінійними асоціаторами, замінено

нечіткими множинами і нечіткими правилами.

До основи фаззі-кластерувальна мережа Кохонена (FKCN) покладено алгоритм FCM Бездека. Особливості цієї нейро-фаззі системи є пакетний режим навчання, коли весь масив даних, зо має бути опрацьований, задається заздалегідь. Тобто FKCN не може працювати у реальному часі, аналізуючи інформацію у міру її надходження.

Тому пропонується як альтернатива SOM і FKCN адаптивна фаззі-самоорганізовна нейронна мережа, що налаштовується за допомогою рекурентного алгоритму самонавчання, що є узагальненням правила навчання Кохонена і дає змогу знаходити у реальному часі не лише прототипи (центроїди) кластерів, зо формуються, але і оцінити рівень належності кожного образу, що надійшов, до конкретного кластера.

Архітектуру нечіткої нейронної мережі наведено на рисунку 3.6. Вона має єдиний шар нейронів  $N_i$ ,  $i=1,2,\dots,m$ , що є адаптивними лінійними асоціаторами, та утворює мапу Кохонена.

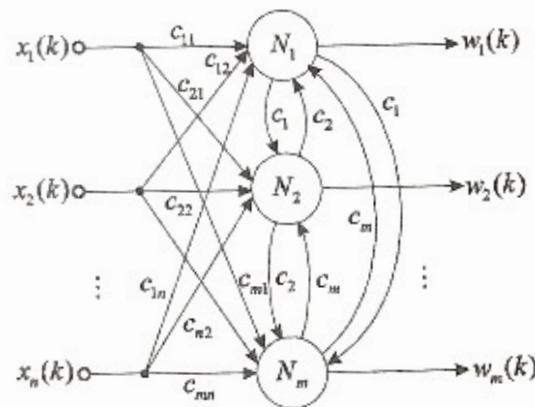


Рисунок 3.6 – Адаптивна фаззі-самоорганізовна нейронна мережа

На рецепторний шар мережі послідовно надходять образи, що мають бути кластеризовані, у вигляді  $(n \times 1)$ -векторів ознак  $x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T$ .

Налаштовані синаптичні ваги  $c_{ij}$ ,  $j=1,2,\dots,m$ ,  $i=1,2,\dots,n$  визначають

координати центрів  $m$  кластерів, що взаємно перетинаються  $c_j(k) = (c_{j1}(k), c_{j2}(k), \dots, c_{jn}(k))^T$ , а виходом мережі, на відміну від SOM, вихідний сигнал якої визначається тільки нейроном-переможцем, є  $(m \times 1)$ -векторів  $w(k) = (w_1(k), w_2(k), \dots, w_m(k))^T$ , що визначає рівень належності образу  $x(k)$  до кожного з  $m$  сформованих кластерів та обчислюється нейронами  $N_j$ . По латеральним зв'язкам нейрони обмінюються координатами  $c_j(k)$ , що необхідні для обчислення належностей  $w_j(k)$ . Алгоритм, відповідальний за формування рівнів належності образів до сформованих кластерів, починається з ініціалізації синаптичних ваг мережі, сформованих генератором псевдовипадкових чисел. За такого кормування карта однак спочатку не має будь-якого порядку ознак. Після коректної ініціалізації мережі для формування рівнів належності образів до кластерів запускаються такі основні процеси:

- конкуренція – для кожного вхідного образу нейрона мережі обчислюються відносні значення дискримінантної функції. Ця функція є основою конкуренції серед нейронів;

- кооперація – нейрон-переможець визначає просторове положення топологічного околу нейронів, забезпечуючи тим самим базис для кооперації між цими нейронами;

- синаптична адаптація – останній механізм дає змогу збудженим нейронам збільшувати власні значення дискримінантних функцій відносно вхідних образів за допомогою відповідних коригувань синаптичних ваг. Коригування відбувається так, що відгук нейрона-переможця на подальше застосування аналогічних прикладів посилювався.

В основу самонавчання адаптивної фаззі-самоорганізованої нейронної мережі покладено імовірнісний метод кластерування, оснований на оптимізації цільової функції на формулі 3.20 з обмеженнями зазначеними на формулах 3.21 та 3.22:

$$E(w_j, c_j) = \sum_{k=1}^N \sum_{j=1}^m w_j^\beta(k) \|x(k) - c_j\|, \quad (3.20)$$

$$\sum_{j=1}^m w_j(k) = 1, k = 1, 2, \dots, N, \quad (3.21)$$

$$0 \leq \sum_{k=1}^N w_j(k) \leq N, j = 1, 2, \dots, m, \quad (3.22)$$

де  $w_j(k) \in [0, 1]$ ,  $N$  – кількість оброблюваних образів.

Застосування нестандартного апарату невизначених множників Лагранжа та процедури нелінійного програмування Ерроу-Гурвіца, що є процедурою пошуку сідлової точки функції Лагранжа, веде до імовірнісного рекурентного методу нечіткої кластеризації (формула 3.23):

$$\begin{cases} c_j(k+1) = c_j(k) + \eta(k)w_j^\beta(k) (x(k+1) - c_j(k)), j = 1, 2, \dots, m; \\ w_j(k+1) = \frac{(\|x(k+1) - c_j(k+1)\|^2)^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (\|x(k+1) - c_l(k+1)\|^2)^{\frac{1}{1-\beta}}}, \end{cases} \quad (3.23)$$

де  $\eta(k)$  – параметр кроку пошуку.

Якщо у формулі 3.23 розглянути множник  $w_j^\beta(k)$  як функцію сусідства  $\varphi(j, l, k)$ , можна отримати формулу 3.24, що є правилом самонавчання Кохонена на основі WTM-принципу:

$$c_j^{\text{WTA}}(k+1) = c_j^{\text{WTA}}(k) + \varphi(j, l, k)(x(k+1) - c_j^{\text{WTA}}(k)). \quad (3.24)$$

Якщо припустити, що параметр  $\beta=0$ , формула 3.25 відповідає стандартному WTA-правилу Кохонена для нейрона-переможця:

$$c_j^{\text{WTA}}(k+1) = c_j^{\text{WTA}}(k) + \eta(k)(x(k+1) - c_j^{\text{WTA}}(k)). \quad (3.25)$$

Нескладно помітити також, що рекурентна процедура на формулі 3.25 мінімізує цільову функцію вигляду (формула 3.26):

$$E(c_j) = \sum_{x(k) \in Cl_j} \|x(k) - c_j^{WTA}\|^2, \quad (3.26)$$

де  $Cl_j$  –  $j$ -й кластер,  $j=1, \dots, m$ .

А її пряма оптимізація веде до звичайної оцінки середнього арифметичного, запис якої в рекурентній формі приводить до співвідношення 3.27:

$$c_j^{WTA}(k+1) = c_j^{WTA}(k) + \frac{1}{k+1} (x(k+1) - c_j^{WTA}(k)), \quad (3.27)$$

де  $\eta(k) = 1/k+1$ . Такий вибір параметру кроку  $\eta(k)$  узгоджується з вимогами стохастичної апроксимації і нажає результати прозорого фізичного змісту.

Отже, адаптивний алгоритм самонавчання нечуткої самоорганізовної мережі можна записати у вигляді 3.28 та 3.29:

$$c_j(k+1) = c_j(k) + \frac{w_j^\beta(k)}{k+1} (x(k+1) - c_j(k)), \quad j = 1, 2, \dots, m, \quad (3.28)$$

$$w_j(k+1) = \frac{(\|x(k+1) - c_j(k+1)\|^2)^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (\|x(k+1) - c_l(k+1)\|^2)^{\frac{1}{1-\beta}}}, \quad (3.29)$$

що об'єднує обчислювальну простоту і послідовну обробку кохоненівського самонавчання з можливостями нечіткої кластеризації.

Цільова функція нечіткої кластеризації показано на формулі 3.30:

$$E_k(w_j(k), c_j(k)) = \sum_{j=1}^m w_j^\beta(k) D^2(x(k), c_j) + \sum_{j=1}^m \mu_j (1 - w_j(k))^\beta. \quad (3.30)$$

Таким чином, алгоритм працює у послідовному режимі, при нелінійно поділених кластерах, але застряє у локальних екстремумах. Тому, додамо до

формули пошуку центроїдів 3.28 «стрибки», у час потрапляння до екстремума. На формулі 3.31 відображена оновлена формула вирахування центроїдів, а формули приналежності (3.29) і цільової функції (3.30) залишаються незмінними.

$$c_j(k+1) = c_j(k) + \frac{w_j^\beta(k)}{k+1} (x(k+1) - c_j(k)) + \xi, j = 1, 2, \dots, m, \quad (3.28)$$

$$\xi(k) \neq 0, \|c_j(k) - c_j(k-1)\| < \xi,$$

$\xi(k) = 0$ , якщо алгоритм не потрапив до екстремума

Для оцінки якості кластерування було використано індекс Ксі-Бені (Хіє-Бені index), який для послідовної обробки має вигляд, як зазначено у формулі 3.32:

$$XB^{[m]}(k) = \frac{NXB^{[m]}(k-1) + \frac{1}{k} (\sum_{j=1}^m u_j^2(k) \|x(k) - w_j(k)\|^2 - NXB^{[m]}(k-1))}{\min_{l \neq j} \|w_j(k) - w_l(k)\|^2} \quad (3.32)$$

## 4 ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ

Для реалізації онлайн нейро-фаззі мапи для пошуку глобального екстремума, при умовах невідомої кількості кластерів, був обран язык програмування Python 3.0 та бібліотеки python scikit, а за для візуалізації даних була обрана бібліотека python matplotlib. Процес обробки інформації відбувається наступним чином:

- користувач задає налаштування системі;
- завантажує у систему дані;
- система нормалізує дані;
- система задає початкові параметри нейронів;
- система створює нейро-фаззі мапу на основі параметрів заданих користувачем;
- система проводить кластерування;
- система розраховує якість проведеного навчання за допомогою індексу Ксі-Бені;
- якщо системі на вхід надходить новий приклад, то Вона не перебудовує заново всю модель кластеризації, а лише трохи оновлює синаптичні ваги нейронів і заново підсчитиваваает значення індексу Ксі-Бені;
- система відображає для користувача графік розбиття на кластери для кожної нейронної мережі.

Для перевірки роботи алгоритму була обрані три популярних вибірки з UCI-репозиторії - Wine, Dermatology, Ionosphere.

### 4.1 Імітаційне моделювання на виборці Wine

Вибірка Wine вдає із себе данні, які були отриманні у результаті хімічного аналізу вина, зробленого з винограду трьох різних сортів, вирощеного в одному і тому ж регіоні Італії. Всього приклів у вибірці – 178.

Вибірка має 13 атрибутів: алкоголь, малинова кислота, зола, алкаїн золи,

магній, всього фенолів, флавоноїди, нефлаваноїдні феноли, проантоціани, інтенсивність кольору, відтінок, OD280/OD315 розведених вин, пролин. Вибірка має один цільовий атрибут, який показує приналежність до певного класу. Всього три класи. Розбиття по класах: у першому класі – 59 прикладів, у другому класі – 71 приклад, у третьому класі – 48 прикладів. Усі атрибути у вибірці є безперервними, порожніх місць немає. Данна вибірка досить популярна, при тестуванні різного роду алгоритмів машинного навчання, тому неможливо було її обійти.

На початку усі данні було пронормовано на гіперкулю  $[-1;1]$  та відцентровано відносно середнього значення.

Для початку потрібно завантажити вибірку з UCІ репозиторію, після цього задати параметри для системи. При бажанні можна також задати такі гіперпараметри як, кількість ітерацій, початковий коефіцієнт оновлення ваг (швидкість навчання), а також правило, за яким буде змінюватися швидкість навчання. Однак, цей крок можна і пропустити, тоді ці гіперпараметри матимуть значення задані в системі за замовчуванням.

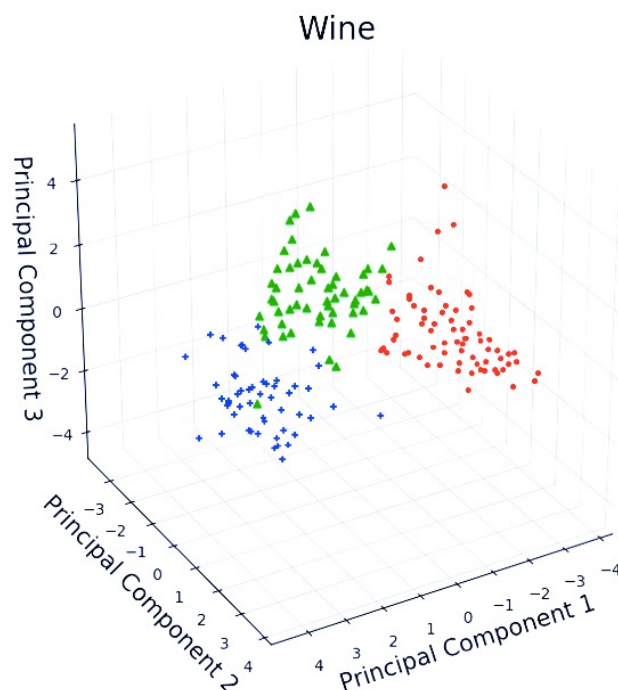


Рисунок 4.1 – Результат роботи системи на вибірці Wine

Після завдання всіх параметрів система починає процедуру кластеризації. Після закінчення процедури система видає показник кластерування – індекс Ксі-Бені, який зазначає чи було кластерування вдалим. А також система малює різні графіки, які можуть бути корисні кінцевому користувачеві.

Для того, щоб відобразити розбиття вибірки на кластери необхідно спочатку застосувати алгоритм PCA (principal component alogotihm), щоб виділити три головні компоненти, і вже по ним будувати графік. Результати розбиття по кластерам вибірки Wine представлено на рисунку 4.1.

Вочевидь, система розділила на 3 кластери обранну вибірку, що є коректним та вірним. Якщо порівняти графік побудований після кластеризації та правдиве розбиття (ground truth), алгоритм чудово впорався з кластерування, незважаючи на деякі відмінності. Показник індексу Ксі-Бені був 0.08287346.

#### 4.2 Імітаційне моделювання на виборці Dermatology

Вибірка Dermatology зберігає у собі дані, які були отримані в результаті дослідження гістопатологічних особливостей пацієнтів з дерматологічними захворюваннями. Метою є визначення типу дерматологічного захворювання.

Вибірка має 33 атрибута: еритема, масштабування, певні межі, свербіж, феномен Коебнера, полігональні папули, фолікулярного папули, ураження слизової оболонки рота, поразку колін і ліктів, ураження шкіри голови, історія сім'ї, вік, еозинофіли в інфільтраті, інфільтрація PNL, фіброз папілярної дерми, екзоцитоз, окантоз, акантом, гіперкератоз, паракератоз, спелетені епідерміальний гребенів, подовжені епідерміальний гребенів, витончення надпапілярного епідермісу, губчаста пустула, абсцес Мунро, фокальний гіпергранулез, зникнення гранульованого шару, вакуолізація і пошкодження базального шару, спонгиоз, пилкоподібний вид епідерміальний гребенів, вилка фолікулярного ріжка, періфолікулярний паракератоз, запальний

моноядерних інфлурат, смуговий інфільтрат.

Вибірка має один цільовий атрибут, який показує наявність певного дерматологічного захворювання, всього шість класів. Розбиття по класах: перший клас (псоріаз) – 112 прикладів, другий клас (себарейний дерматит) – 61 приклад, третій клас (лішай) – 72 прикладу, четвертий клас (рожевий лішай) – 49 прикладів, п'ятий клас (хронічний дерматит) – 52 прикладу, шостий клас (червоний плоский лішай) – 20 прикладів.

Дана вибірка не є досить популярною для тестування різного роду алгоритмів машинного навчання, оскільки вона має кілька лінійно нероздільних кластерів, які важко обробляються стандартними алгоритмами кластеризації. Однак буде цікаво перевірити чи впровадиться розроблена система з цим завданням.

Для початку потрібно завантажити вибірку з UCІ репозиторію, після цього задати параметри для системи. При бажанні можна також задати такі гіперпараметри як, кількість ітерацій, початковий коефіцієнт оновлення ваг (швидкість навчання), а також правило, за яким буде змінюватися швидкість навчання. Однак, цей крок можна і пропустити, тоді ці гіперпараметри матимуть значення задані в системі за замовчуванням.

Після завдання всіх параметрів система починає процедуру кластеризації. Після закінчення процедури система видає показник кластерування – індекс Ксі-Бені, який зазначає чи було кластерування вдалим. А також система малює різні графіки, які можуть бути корисні кінцевому користувачеві.

Для того, щоб відобразити розбиття вибірки на кластери необхідно спочатку застосувати алгоритм PCA (principal component alogotihm), щоб виділити три головні компоненти, і вже по ним будувати графік. Результати розбиття по кластерам вибірки Wine представлено на рисунку 4.2.

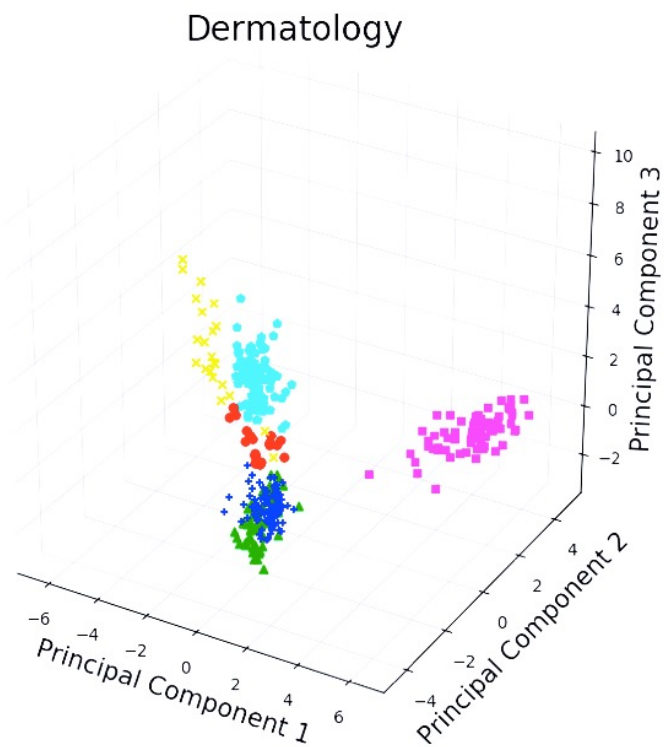


Рисунок 4.2 – Результат роботи системи на вибірці Dermatology

Вочевидь, система розділила на 6 кластери обрану вибірку, що є коректним та вірним. Якщо порівняти графік побудований після кластеризації та правдиве розбиття (ground truth), алгоритм чудово впорався з кластерування, незважаючи на деякі відмінності. Показник індексу Ксі-Бені був 0.08287346.

#### 4.3 Імітаційне моделювання на виборці Ionosphere

Вибірка Ionosphere має дані, які були отримані в результаті дослідження випромінювання високочастотних антен через іоносферу. Метою є опеределения присутствия іоносфери.

Вибірка має 33 атрибута, які показують різні властивості випромінюваного антеною сигналу. А також, вибірка має один цільової атрибут, який показує присутнє чи іоносфера на шляху сигналу.

Ця вибірка не є досить популярною для тестування різного роду

алгоритмів машинного навчання, поскільки вона має кілька лінійно нероздільних кластерів, які важко обробляються стандартними алгоритмами кластеризації. Однак буде цікаво перевірити чи впровадиться розроблена система з цим завданням.

Для початку потрібно завантажити вибірку з UCІ репозиторію, після цього задати параметри для системи. При бажанні можна також задати такі гіперпараметри як, кількість ітерацій, початковий коефіцієнт оновлення ваг (швидкість навчання), а також правило, за яким буде змінюватися швидкість навчання. Однак, цей крок можна і пропустити, тоді ці гіперпараметри матимуть значення задані в системі за замовчуванням.

Після завдання всіх параметрів система починає процедуру кластеризації. Після закінчення процедури система видає показник кластерування – індекс Ксі-Бені, який зазначає чи було кластерування вдалим. А також система малює різні графіки, які можуть бути корисні кінцевому користувачеві.

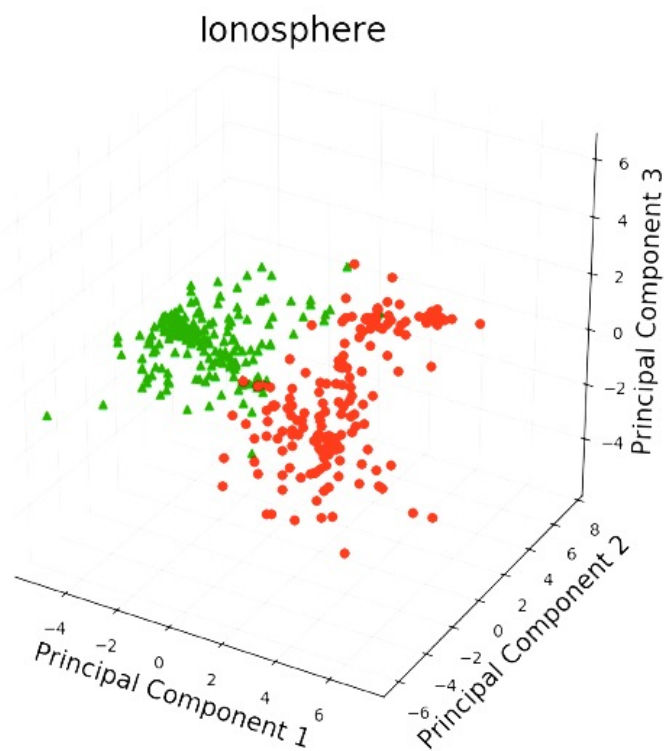


Рисунок 4.2 – Результат роботи системи на вибірці Ionosphere

Для того, щоб відобразити розбиття вибірки на кластери необхідно спочатку застосувати алгоритм PCA (principal component alogotihm), щоб виділити три головні компоненти, і вже по ним будувати графік. Результати розбиття по кластерам вибірки Wine представлено на рисунку 4.2.

Вочевидь, система розділила на 2 кластери обранну вибірку, що є коректним та вірним. Якщо порівняти графік побудований після кластеризації та правдиве розбиття (ground truth), алгоритм чудово впорався з кластерування, незважаючи на деякі відмінності. Показник індексу Ксі-Бені був 0.05034857.

#### 4.4 Висновки імітаційного моделювання

Під час імітаційного моделювання була розроблена програмная система, яка працює у послідовному (online) режимі, може подолати локальні екстремуми, та базується на самоорганізованій мапі Когонена. Система була реалізована на скриптовій мові програмування Python 3.6, крім того, в самій системі були використані бібліотеки scikit і matplotlib. Система була протестована на трьох різних вибірках з UCI репозиторія: Wine, Dermatology, Ionosphere. Розроблена система показала себе дуже добре на усіх трьох вибірках й розподілила на вірну кількість кластерів. Алгоритм працює потимально на довільній кількості кластерів, а також незалежно від того чи лінійно-подільні класи, чи ні.

У таблицях 4.1-4.3 наведено показний якості кластерування – індекс Ксі-Бені. Вочевидь, розроблений алгоритм краще вторався з випробуваннями на усіх трьох вибірках з репозиторію. Також непогані результати, а також схожі показники виявилися у алгоритма Fuzzy J-means Fuzzy Self-Organizing Maps, тому, що мають великі переваги перед Fuzzy C-means алгоритмом, а саме – можуть працювати у онлайн режимі, вістрибувати з локальних екстремумів, для пошуку рішення. Експеримент вважається вдалим, та усі теоретичні данні підкріплені на практиці.

Таблиця 4.1 – Показники якості кластерування на вибірці Wine

Досліджувані методи	Wine
	Index Xie-Beni
Fuzzy C-means	0.93542919
Fuzzy Self-Organizing Maps	0.07935471
Fuzzy J-means	0.06049569
Online Fuzzy SOM+jump	0.04979868

Таблиця 4.2 – Показники якості кластерування на вибірці Dermatology

Досліджувані методи	Dermatology
	Index Xie-Beni
Fuzzy C-means	0.69646095
Fuzzy Self-Organizing Maps	0.32048576
Fuzzy J-means	0.12094750
Online Fuzzy SOM+jump	0.08287346

Таблиця 4.3 – Показники якості кластерування на вибірці Ionosphere

Досліджувані методи	Ionosphere
	Index Xie-Beni
Fuzzy C-means	0.3621412
Fuzzy Self-Organizing Maps	0.07204985
Fuzzy J-means	0.12085710
Online Fuzzy SOM+jump	0.05034857

## ВИСНОВКИ

У дипломній роботі був проведений аналіз поточної ситуації в області штучного інтелекту. Розглянуті різні розділи машинного навчання, такі як: навчання без учителя, навчання з учителем. Крім того, були вивчені основні алгоритми кластеризації, були виділені з гідності і недоліки. Також були розглянуті нейронні мережі, історія їх розвитку, їх структура і типи.

У цій дослідницькій роботі був запропонован й опробуван новий метод послідовного кластерування даних для пошуку глобального екстремуму, шляхи оминання локального екстремуму. Коли кластери роздільні або перетинаються, застосування запропонованого методу дає можливість подолати цю проблему.

Програмна реалізація методу пошуку глобального екстремуму у послідовному режимі проста й полягає у ансамблі можливостей нейромереж, методів чіткого і нечеткого кластерування, а також методів, які працюють у online режимі.

В атестаційній роботі магістра представлені результати, які є рішеннями актуального завдання кластерування відповідно до поставленої мети. Метод полягає у кластеруванні багатовимірних потоків даних, завдяди якому, можна досягти рішення, тобто глобального екстремуму, при потраплянні у локальні, здійснювати «стрибок». Основую є нейро-фаззі система та її послідовне самонавчання.

В якості імітаційного моделювання розробленого алгоритму була розроблена програмна система на мові програмування python. За допомогою цієї програмної системи, розроблений алгоритм був практично перевірений на наступних вибірках з UCI репозиторія : Wine, Dermatology, Ionosphere.

Візуалізація результатів роботи алгоритму була проведена за допомогою python бібліотеки matplotlib. А також були реалізовані модулі інтеграції з UCI репозиторієм, налаштуванням початкових параметрів SOM.

Таким чином, якщо підсумувати усе вище сказане й озвучити у пунктах,

то під часи розробки й написання роботи магістра:

- проведено аналіз існуючих методів кластерування, який показав, що основні методи, які повсюдно використовуються в рамках напрямку Data Mining, призначені тільки для роботи в пакетному режимі;

- в рамках атестаційної роботи був розроблений алгоритм нечіткого кластерування даних, який працює в online режимі;

- в рамках атестаційної роботи був розроблений алгоритм нечіткого кластерування даних, який працює в online режимі;

- розроблений алгоритм дозволяє оминати та «вистрибувати» з локальних екстремумів для пошуку глобального екстремума;

- проведено імітаційне моделювання на даних, згенерованих за допомогою штучної вибірки, в якій кластери лінійно нероздільні і так само на декількох тестових вибірках з UCI-репозиторія, розроблений алгоритм кластерування показав високу точність кластерування як на штучних, так і на даних тестових вибірок;

- імітаційно проведено порівняння існуючих методів кластерування з запропонованим;

- якість кластерування розрахована за допомогою індексу Kсі-Бені.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Computational Intelligence. A Methodological Introduction / R. Kruse, C. Borgelt, F. Klawonn, C. Moewes, M. Steinbrecher, P. Held. Berlin : Springer, 2013. 488 p.
2. Aggarwal C.C. Data Mining. Cham: Springer, Int. Publ., Switzerland, 2015. 734 p.
3. The Elements of Statistical Learning. Data Mining, Inference and Prediction / J. Friedman, T. Hastie, B. Feil, R. Tibshirani. Berlin : Springer, 2003. 552 p.
4. Aggarwal C.C., Reddy C.K. Data Clustering. Algorithms and Application. Boca Raton : CRC Press, 2014. 648 p.
5. Data streams fast EM-fuzzy clustering based on Kohonen`s self-learning / Ye. Bodyanskiy, A. Deineko, Y. Kutsenko, O. Zayika // The 1th IEEE International Conference on Data Stream Mining & Processing (DSMP 2016): proc. of int. conf. (Lviv, August 23-27, 2016). Lviv. 2016. P. 309-313.
6. Бодянский Е.В., Руденко О.Г. Искусственные нейронные сети: архитектуры, обучение, применение. Харьков : ТЕЛЕТЕХ, 2004. 372 с.
7. Нечеткая кластеризация потоков данных с помощью EM-алгоритма на основе самообучения по Т. Кохонену / Е. В. Бодянский, А. А. Дейнеко, А. А. Заика, Я. В. Куценко // Прикладная радиоэлектроника. 2016. Т. 15, № 1. С. 80–83.
8. Du K.-L., Swamy M.N.S. Neural Networks and Statistical Learning. London : Springer-Verlag, 2014. 824 p.
9. Borgelt C. Prototype-based Classification and Clustering. Magdebur, 2005. 350 p.
10. Сербіна Д.В. Гибридні нейро-фаззі системи в задачах потокової обробки даних. XXIII Міжнародному молодіжному форуму – Радиоелектроніка та молодь у XXI столітті: матеріали конференції, м. Харків, 16-18 квітня, 2019 м. Харків: ХНУРЕ. 2019. С.45-46

11. Сербіна Д.В. Нечітка класифікація масивів даних, що надходять у послідовному online режимі. V Міжнародна науково-практична конференція – Обчислювальний інтелект (Результати, проблеми, перспективи): матеріали конференції, м. Ужгород, 15-20 квітня, 2019 м. Ужгород: УжНУ. 2019. С. 72-73
12. Нечеткая кластеризация потоков данных с помощью EM–алгоритма на основе самообучения по Т. Кохонену / Е. В. Бодянский, А. А. Дейнеко, А. А. Заика, Я. В. Куценко // Прикладная радиоэлектроника. 2016. Т. 15, № 1. С. 80–83.
13. Abonyi J., Feil B. Cluster Analysis for Data Mining and System Identification. Basel : Birkhauser, 2007. 303 p.
14. The Elements of Statistical Learning. Data Mining, Inference and Prediction / J. Friedman, T. Hastie, R. Tibshirani. Berlin : Springer, 2003. 552 p.
15. Bodyanskiy Ye. Computational intelligence techniques for data analysis // Lecture Notes in Informatics. Bonn. 2005. №72. P. 15–36.
16. Bezdek J.C., Hathaway R.J. Optimization of fuzzy clustering criteria using genetic algorithms. IEEE Conference on Evolutionary Computation, Vol. 2: IEEE Press, 2002. NJ : Picataway, 1994. P. 589–594.
17. Hansen P., Mladenovic N. J-Means: a new local search heuristic for minimum sum-of-squares clustering. Brussel : Pattern Recognition, 2001. P. 405 413.
18. Bezdek J.C. Pattern Recognition with Fuzzy Objective Function Algorithms. New York : Plenum Press, 1981. 257p.
19. Растрингин Л.А. Системы экстремального управления. Москва: Наука. 1974. 632с.
20. Lughofer E. Evolving Fuzzy Systems – Methodologies, Advanced Concept and Applications. Berlin-Heidelberg : Springer-Verlag, 2011. 454 p.
21. Є. В. Бодяньський, Д. Д. Пелешко, О. А. Винокурова, С. В. Машталір, Аналіз та обробка потоків даних засобами обчислювального інтелекту. Львів: Львів. політехніки, 2016, 235с.