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)

Самонавчанна нейро-фаззі система
для кластерування великих масивів даних
(тема)

Виконав:
студент 2 курсу, групи СШІм-18-2
Іванова Є.В.
(прізвище, ініціали)

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

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

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

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

Допускається до захисту

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

2020 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-наукова
(освітньо-професійна або освітньо-наукова)

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

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2020 р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ
студентові Іванової Євгенії Владиславівни
(прізвище, ім'я, по батькові)

1. Тема роботи Самонавчанна нейро-фаззі система для кластерування великих масивів даних
затверджена наказом по університету від р. № Ст
2. Термін подання студентом роботи до екзаменаційної комісії ____ _____ 20__ р.
3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проектів щодо розробки систем нейро-фаззі кластерування в задачах динамічного аналізу даних

4. Перелік питань, що потрібно опрацювати в роботі 1. Аналіз предметної області і формалізація задачі, 2. Штучні нейронні мережі. Конкурентне навчання самоорганізованих мап, 3. Самоорганізована нейро-фазі система та її послідовне навчання, 4. Імітаційне моделювання і перевірка теоретичних досліджень

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1 – Архітектура SOINN, Рисунок 2 – Блок-схема алгоритму SOINN, Рисунок 3 – Порівняння функцій належності, Рисунок 4 – Візуалізація кластерування методом FSOINN, Рисунок 5 – Візуалізація кластерування методом SOINN, Рисунок 6 – Візуалізація кластерування методом FCM, Рисунок 7 – Кластерування розрідженої вибірки FSOINN, Рисунок 8 – Кластерування розрідженої вибірки SOINN, Рисунок 9 – Кластерування розрідженої вибірки FCM, Рисунок 10 – Кластерування вибірки з шумом методом FSOINN, Рисунок 11 – Кластерування вибірки з шумом методом SOINN, Рисунок 12 – Кластерування вибірки з шумом методом FCM, Рисунок 13 – Візуалізація аномалій у 3D, Рисунок 14 – Візуалізація матриці помилок

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
Аналіз предметної галузі	проф. Бодянський Є.В.		31.03.2020
Формування вимог до додатку	проф. Бодянський Є.В.		13.04.2020
Розробка додатку	проф. Бодянський Є.В.		22.04.2020
Аналіз готового продукту	проф. Бодянський Є.В.		23.04.2020

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	31.03.20 – 15.04.20	виконано
2	Дослідження існуючих методів кластерування	13.04.2020	виконано
3	Розробка нейро-фаззі системи для кластерування даних в	14.04.2020	виконано
4	Створення імітаційної моделі	15.04.20 – 22.04.20	виконано
5	Тестування і аналіз отриманих результатів	23.04.2020	виконано
6	Оформлення пояснювальної записки	24.04.20 – 08.05.20	виконано
7	Попередній захист	14.05.2020	виконано
8	Захист перед ЕК	19.05.2020	

Дата видачі завдання _____ 2020 р.

Студент _____
(підпис)

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

РЕФЕРАТ

Пояснювальна записка складається з 75 с., 22 рисунка, 2 таблиці, 34 формул, 24 джерел.

ІНКРЕМЕНТНА САМООРГАНІЗОВАНА МАПА, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, КЛАСТЕРУВАННЯ, НЕЙРОННА МЕРЕЖА, НЕЧІТКА СИСТЕМА, ПОСЛІДОВНЕ НАВЧАННЯ, ШТУЧНИЙ ІНТЕЛЕКТ

Об'єктом дослідження є процес опрацювання даних, що надходять одне за одним, в послідовному режимі за допомогою нечіткої інкрементної самоорганізованої багатошарової мапи.

Предметом дослідження є методи потокового кластерування з довільною кількістю кластерів в задачах інтелектуального аналізу даних.

Метою бакалаврської атестаційної роботи є створення нечіткої інкрементної самоорганізованої мапи та її послідовне навчання.

Методи дослідження базуються на теорії обчислювального інтелекту, а саме на методах теорії штучних нейронних мереж для побудови архітектури самоорганізованих мап, що складаються з нечіткого шару виводу і дозволяють проводити нечітке кластерування в послідовному режимі. Імітаційне моделювання застосовується для перевірки якості кластерування з використанням синтезованої архітектури самоорганізованої інкрементної нейронної мережі.

В атестаційній роботі розглядається задача нечіткого послідовного кластерування потоків даних в умові невідомої кількості класів.

РЕФЕРАТ

Пояснительная записка состоит из 75 с., 22 рисунка, 2 таблиці, 34 формул, 24 источников.

ИНКРЕМЕНТНАЯ САМООРГАНИЗУЮЩАЯСЯ КАРТА, ИНТЕЛЛЕКТУАЛЬННЫЙ АНАЛИЗ ДАННЫХ, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, КЛАСТЕРИЗАЦИЯ, НЕЙРОННЫЕ СЕТИ, НЕЧЁТКАЯ СИСТЕМА, ПОСЛЕДОВАТЕЛЬНОЕ ОБУЧЕНИЕ

Объектом исследования является процесс обработки данных, поступающих друг за другом, в последовательном режиме с помощью нечеткой инкрементной самоорганизующейся многослойной карты.

Предметом исследования являются методы потокового кластерування с произвольным количеством кластеров в задачах интеллектуального анализа данных.

Целью бакалаврской аттестационной работы является создание нечёткой инкрементной самоорганизующейся карты и её последовательное обучение.

Методы исследования базируются на теории вычислительного интеллекта, а именно на методах теории искусственных нейронных сетей для построения архитектуры самоорганизующихся карт, состоящие из нечёткого слоя вывода и позволяющие проводить нечёткую кластеризацию в последовательном режиме. Имитационное моделирование применяется для проверки качества кластеризации с использованием синтезированной архитектуры самоорганизующейся инкрементной нейронной сети.

В аттестационной работе рассматривается задача нечёткой последовательной кластеризации потоков данных в условии неизвестного количества классов.

ABSTRACT

Explanatory note: 75 pages, 22 figures, 2 tables, 34 formulas, 24 sources.

INCREMENTAL SELF-ORGANIZING MAP, INTELLIGENT DATA ANALYSIS, ARTIFICIAL INTELLIGENCE, CLUSTERING, FUZZY SYSTEM, NEURAL NETWORKS, SEQUENTIAL TRAINING

The object of the study is the process of processing data arriving one after another in a sequential mode using a fuzzy incremental self-organizing multilayer map.

The subject of the study are methods of streaming clustering with an arbitrary number of clusters in data mining tasks.

The aim of the bachelor's certification work is to create a fuzzy incremental self-organizing map and its consistent training.

The methods of investigation are based on the theory of computational intelligence, namely, on the methods of the theory of artificial neural networks for constructing an architecture of self-organizing maps, consisting of a fuzzy layer of output, which allows to conduct fuzzy clustering in a sequential mode. Simulation modeling is used to test the quality of a cluster using the synthesized architecture of a self-organizing incremental neural network.

In the certification work, the problem of fuzzy sequential clustering of data streams in the condition of an unknown number of classes is considered.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШНМ – штучні нейронні мережі;

FCM – Fuzzy C-means – нечіткий с-середніх;

FSOINN – Fuzzy Self-Organizing Incremental Neural Network – нечітка самоорганізована інкрементна нейронна мережа;

SOINN – Self-Organizing Incremental Neural Network – самоорганізована інкрементна нейронна мережа;

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

ВСТУП

Стрімкий розвиток інформаційних технологій змінює світ, штучні системи вже можуть робити те, що раніше могла робити лише людина. У деяких випадках ці системи можуть виконувати завдання навіть краще, ніж людина. Штучний інтелект тепер може імітувати людську мову, перекладати тексти, діагностувати ракові захворювання, розробляти юридичні документи та грати в ігри і навіть перемагати людських конкурентів. Але є проблема в тому, що немає повного уявлення механізму біологічного інтелекту, тому здійснити повний перехід від природнього інтелекту до штучного досі не вдалося. Проте під інтелектом в межах цієї науки розуміється тільки обчислювальна складова, котра забезпечує автоматизацію і швидкість обробки даних в релевантну інформацію.

Існує багато напрямків науки штучного інтелекту є машинне навчання та інтелектуальний аналіз даних. В рамках цих наук вирішується доволі багато завдань, а саме таких як регресія, класифікація, кластерування, прогнозування та інші. Найбільш ефективні методи реалізовані на основі штучних нейронних мереж.

Рішення задачі кластерування масивів даних здійснюється в режимі навчання без вчителя. Вхідна множина об’єктів розбивається на групи зі схожими властивостями. В сучасному світі ця задача використовується в різних галузях, наприклад, таких як медицина, геологія, біологія, антропологія, психологія та інших. Найбільш широкого поширення за свою простоту отримав алгоритм К-середніх. Головною проблемою алгоритмів заснованих на К-середніх є вибір кількості кластерів заздалегідь. Другий недолік полягає в тому, що ці алгоритми обробляють тільки ізотропні кластери, такі як коло, сфера. Всі алгоритми кластерування відрізняються за сферою застосування, швидкості, математичному обґрунтуванню і за вхідними параметрами.

По-перше, навчання нейронної мережі поділяють за режимом подачі

інформації на вхід: пакетний і послідовний. Перший починає навчання тільки після подачі всіх вхідних даних і при додаванні нових прикладів необхідно перенавчити мережу. Отже процес обробки даних займає багато часу, що в разі постійного їх оновлення не підходить. З точки зору процесів реального часу, послідовний режим є кращим, ніж пакетний, так як вимагає меншого обсягу внутрішнього сховища для кожного прикладу. Більш того, пред'являючи навчальні приклади по одному у випадковому порядку, вивчається нова інформація і не знищуються вже отримані знання. Отже, не потрібно перенавчати мережу, якщо в майбутньому будуть подані нові сигнали. Тому запропонований алгоритм є актуальним для вивчення, тому що використовує послідовний режим і може вирішувати реальні задачі.

По-друге, серед алгоритмів кластерування виділяється клас методів нечіткого кластерування. Застосування методів цього класу дозволяє формалізувати різного роду невизначеності, які завжди існують при вирішенні реальних завдань. Нечітке кластерування є синтезом ідей кластерного аналізу та теорії нечітких множин. На основі понять, що вводяться і вивчаються цією теорією, можна замість звичайного розбиття об'єктів на кластери розглянути їх нечіткі варіанти.

По-третє, у дипломній роботі реалізується алгоритм, який вирішує не тільки задачу кластерування, а також дозволяє відобразити топологічну структуру багатовимірного простору даних у вигляді двовимірного. Отриману мапу можна використовувати як засіб візуалізації при аналізі даних.

Мета дипломної роботи полягає в тому, що створюється новий метод нечіткого кластерування, в який дані надходять послідовно. Більш того є можливість візуалізувати багатовимірні вхідні дані для зручності їх аналізу.

Таким чином, актуальність полягає в тому, що пропонується алгоритм, яких може використовуватися для доволі швидкого вирішення реальних задач нечіткого кластерування аналізувати отримані результати.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ФОРМАЛІЗАЦІЯ ЗАДАЧІ

Кінець двадцятого століття характеризується інтенсивним вивченням штучних нейронних мереж (ШНМ). Поява ШНМ пов'язана з розумінням того, що мозок живого організму працює інакше, ніж комп'ютер. Людський мозок – це дуже складна нелінійна, паралельно інформаційно-керуюча система, яка здатна до мислення, накопичення і відновлення інформації, вирішення проблем. Ця система складається з великої кількості нервових клітин, або нейронів, що представляють собою прості елементи обробки сигналів, які отримують і комбінують інформацію від інших нейронів через входи – дендрити. Якщо об'єднаний з усіх входів сигнал досить сильний, нейрон переходить в збуджений стан («збуджується»), генеруючи сигнал на виході – аксоні, який пов'язаний з дендритами безлічі інших нейронів. Кожен сигнал, що надходить в нейрон, проходить через синаптичне з'єднання, де в результаті електрохімічних процесів потік електричних зарядів або прискорюється, або сповільнюється. Саме зміни провідності синаптичних зв'язків лежать в основі процесів навчання і запам'ятовування інформації. Але якщо початково дослідження базувалися на використанні моделей біологічних нейронів У. Маккалоха, У. Питтса и Ф. Розенблатта [4, 5], то сьогодні нейромережеві технології збагатилися новими моделями, заснованими на законах фізики, генетики, математики.

Поняття «навчання» є ключовим в теорії ШНМ. Тип і характер навчання визначаються на основі розв'язуваної задачі, властивостей даних, які надходять на вхід мережі із зовнішнього середовища у вигляді навчальної вибірки образів або прикладів. Відомо дві основні парадигми навчання: з учителем і без вчителя. Навчання з учителем є більш простим і очевидним. Завідомо відома інформація про зовнішнє середовище, яка задана у вигляді послідовності або пакета вхідних векторів x. Також відомий навчальний сигнал d. Реакція ненавченої мережі, відрізняється від «правильної» реакції вчителя, в результаті чого виникає помилка.

У процесі навчання необхідно так налаштувати параметри ШНМ, щоб деяка скалярна функція помилки(критерій якості) досягла свого мінімального значення. Навченою вважається мережа, яка в деякому, як правило, статистичному сенсі повторює реакцію вчителя. Оскільки інформація про зовнішнє середовище зазвичай має нестаціонарний характер, процес навчання йде безперервно, для чого використовуються ті чи інші рекурентні процедури. Відповідно оцінити обраний алгоритм і обчислити критерій якості неможливо, тому крім навчання мережі, необхідно визначати параметр по якому буде оцінено правило навчання. На рисунку 1.1 запропонована схема навчання з учителем.

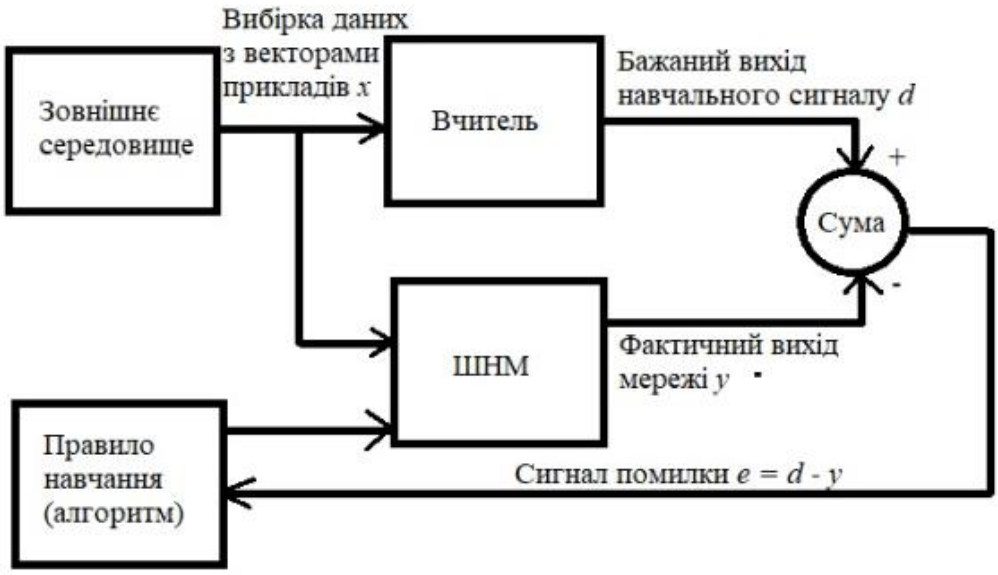


Рисунок 1.1 – Схема навчання з учителем

Парадигмою навчання без учителя, або самонавчання, називають навчання, коли правильна реакція на сигнали зовнішнього середовища невідома. Процес самонавчання схематично представлений на рисунку 1.2.

Мережі, які реалізують парадигму самонавчання, призначені, як правило, для аналізу внутрішньої латентної структури вхідної інформації і вирішують завдання автоматичної класифікації, кластеризації, факторного

аналізу, компресії даних. Ці задачі є більш складними виходячи з математичної постановки, але вирішують безліч реальних проблем.

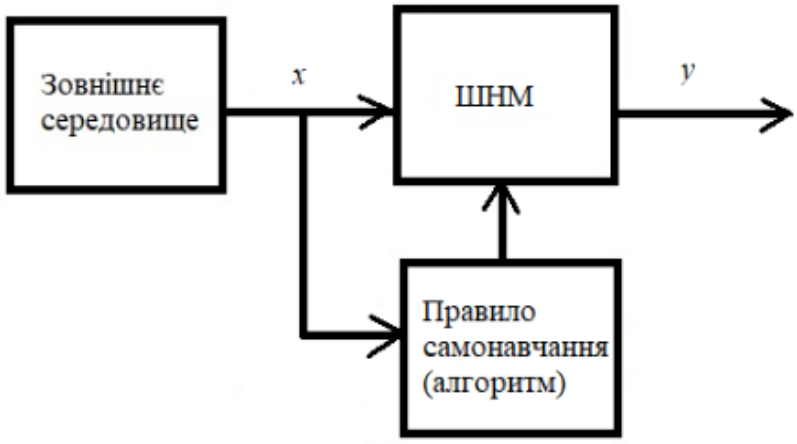


Рисунок 1.2 – Схема навчання без вчителя

Своєрідним компромісом між двома цими парадигмами є навчання з підкріпленням, при якому доступна лише непряма інформації про правильну реакції на вхідний сигнал. Нейронна мережа виробляє відображення вхідної інформації у вихідний вектор, проте, оскільки бажаний навчальний сигнал в явному вигляді не заданий, неможливо отримати помилку, на підставі якої відбувається навчання. Передбачається, що є деякі апріорні знання, що дозволяють зв'язати евристичний сигнал підкріплення з бажаним виходом за допомогою деякої функції. Зазвичай ця функція враховує зв'язок вихідних сигналів мережі з подіями у зовнішньому середовищі, для чого в схему навчання вводиться додатковий блок – «критик», що відображає поведінку мережі. Далі обчислюється евристична помилка, на основі якої і реалізується процес навчання. Досить широке поширення набула також парадигма змішаного навчання, коли частина параметрів мережі налаштовується за допомогою навчання з учителем, а інша частина або архітектура в цілому - за допомогою

самонавчання. Цей підхід набув найбільшого поширення при навчанні радіально-базисних ШНМ.

З введеними парадигмами тісно пов'язані правила навчання, що лежать в основі конкретних алгоритмів. У конкурентному навчанні, можуть бути реалізовані всі описані парадигми, при цьому його відмінною рисою є процес «змагання» нейронів вихідного шару. Існує два принципи змагання згідно з Когоненом «переможець отримує все» і «переможець отримує більше» [3]. У першому випадку збуджується тільки один вихідний нейрон – «переможець». У другому відбувається виправлення всіх синаптичних ваг і переможець отримає більше, ніж всі інші. Найбільш яскравими прикладами мереж, що використовують це правило, є мережі адаптивного резонансу і самоорганізовані мапи (SOM).

1.1 Постановка задачі кластерування

Однією з задач навчання без вчителя є кластерування масивів даних. У дипломній роботі запропоновано самонавчанну нейро-фаззі систему для вирішення цієї проблеми.

Кластерування – задача угруповання набору об'єктів таким чином, що об'єкти в одній і тій же групі (звані кластером) більш схожі (в тому чи іншому сенсі) один з одним, ніж з іншими групами (кластерами). Можна сказати, що задача полягає в пошуку взаємозв'язків в нерозміченому наборі даних.

Мета кластерування – пошук існуючих структур. Кластерування є описовою процедурою, вона не робить ніяких статистичних висновків, але дає можливість провести розвідувальний аналіз і вивчити структуру даних.

Поняття кластер визначено неоднозначно, але можна охарактеризувати як групу об’єктів, що мають спільні властивості. Характеристиками кластера є дві ознаки: внутрішня однорідність і зовнішня ізольованість. Тобто кожен класт приклади схожі один до одного

за певним набором характеристик. Зовнішня ізольованість описує відміну набору характеристик від інших кластерів. Також вони можуть бути непересічними і пересічними [5].

Важливо відзначити, що кожен метод кластерування має свої переваги і недоліки, тому вибір алгоритму завжди залежить від характеристик і властивостей набору даних, які були виділені на етапі попереднього аналізу даних. Також важливо розуміти розподіл даних для того, щоб отримати на виході очікуваний результат.

Найбільш поширені властивості методів кластерування є:

– алгоритм кластерування, який базується на центроїдах. У цьому методі угруповання даних посилається на вектор значень. Кожен об'єкт є частиною кластера і відстань якого мінімальна до центроїда в порівнянні з іншими центроїдами кластерів. Кількість кластерів має бути попередньо визначено, і це найбільша проблема такого роду алгоритмів, бо не завжди є інформація про те, на скільки кластерів необхідно розбити набір даних. Також необхідно розробити метод розташування центроїдів. Ця методологія найбільш близька до предмета класифікації і широко використовується для задач оптимізації;

– на основі розподілу. Пов'язана зі заздалегідь визначеними статистичними моделями розподілення об'єктів, значення яких залежать від одного і того ж розподілу. Через свою випадкову природу генерації значень цей процес потребує добре визначеної і складної моделі для взаємодії з реальними даними. Однак ці процеси можуть забезпечити оптимальне рішення і розрахувати кореляції і залежності;

− на основі зв’язків. У цьому типі алгоритму кожен об'єкт пов'язаний з його сусідами, в залежності від ступеня цього відношення на відстані між ними. Виходячи з цього припущення, кластери створюються поруч з об'єктами і можуть бути описані як максимальна межа відстані. При такому зв'язку між членами ці кластери мають ієрархічні уявлення. Функція відстані залежить від фокуса аналізу;

− на основі щільності. Ці алгоритми створюють кластери відповідно з високою щільністю елементів набору даних в певному місці. Він об'єднує деяку відстань до стандартного рівня щільності для угруповання членів в кластерах. Такі процеси можуть мати меншу продуктивність при виявленні граничних областей групи.

Розглядуваний в дипломній роботі алгоритм кластерування відноситься до методу на основі зв’язків кожного об'єкта з його сусідом.

1.2 Постановка задачі нечіткого кластерування

Алгоритми нечіткої кластеризації дозволяють отримати нечіткий розподіл даних по кластерам. Тобто кожен з прикладів не входить однозначно в будь-який кластер, а належить всім кластерам з різними ступенями належності. Чим більше значення належності, тим с більшою вірогідністю приклад належить до кластеру.

Нечітке кластерування дає переваги в випадках, коли кластери знаходяться близько один до одного, і велике число точок знаходиться на лінії перетину кластерів. Однак ціною такої нечіткості служать великі обчислювальні витрати, в порівнянні з такими алгоритмами як с-середніх і k-середніх.

Наприклад, нечіткий с-середніх базується на мінімізації наступної цільової функції:

$$F\left(x(k), c_j, u_j(k)\right) = \sum_k^N \sum_j^m u_j^2(k) \left\| x(k) - c_j \right\|^2, \qquad (1.1)$$

де $u_j(k)$ – визначає належність елемента j вихідної множини векторів до x(k) до кластеру j;

c_j - центр кластера, який розраховується як векторна норма.

При нечіткому розбитті ступінь належності об'єкта до кластеру приймає значення з інтервалу від нуля до одиниці $u_j(k) \in [0, 1]$, коли при

чіткому з двоелементної множини {0, 1}. Обмеження для матриці нечіткого розбиття записуються так:

$$\sum_{j=1}^{m} u_j(k)=1. \quad (1.2)$$

Нечітке розбиття дозволяє доволі легко вирішити проблему об'єктів, розташованих на кордоні двох кластерів і неможливо точно визначити до якого з кластерів наданий приклад відноситься. Наприклад, вибірка розбита на два класи і приклад, який розташований на їх кордоні належить з вірогідністю 0.4 до першого і 0.6 до другого кластера.

Але недолік нечіткого розбиття проявляється при роботі з об'єктами, віддаленими від центрів всіх кластерів. Віддалені об'єкти мають мало спільного з будь-яким з кластерів, тому інтуїтивно хочеться призначити для них малий ступень належності. Однак, за умовою (1.2) сума їх ступенів така ж, як і для об'єктів, близьких до центрів кластерів, тобто дорівнює одиниці.

1.3 Штучні нейронні мережі

Основною особливістю штучних нейронних мереж і, природно, що утворюють їх нейронів є здатність до навчання, в процесі якого синаптичні ваги налаштовуються за допомогою того чи іншого адаптивного алгоритму з метою найбільш ефективного вирішення поставленої проблеми.

Біологічний нейрон є особливою біологічною системою, призначеної для передачі і обробки інформації в живих організмах. Нейрон складається з тіла клітини, або соми, дендритів, аксона і синапсів.

Тіло клітини включає ядро, яке містить інформацію про спадкові властивості, і плазму, що володіє молекулярними засобами для створення необхідних нейрона матеріалів. Саме в сомі реалізуються основні функції, пов'язані з генетичними і метаболічними механізмами, необхідними для

життєдіяльності, а також інформаційні функції.

Аксон – це нервове волокно, поєднане з сомой і є провідником вихідного сигналу. Аксон має розгалуження – волокна, звані терміналами аксона, за якими нервові імпульси проходять до інших нейронів. Дендрити – дуже розгалужене дерево волокон, з'єднаних з сомой. Дендрити отримують сигнал від інших аксони терміналів через спеціальні контакти - синапси.

Синапс є функціональним інтерфейсом між двома нейронами (аксонний термінал одного нейрона і дендрит іншого) і здатний підсилювати або придушувати сигнал подібно електронного підсилювача, визначаючи характер обробки інформації в сомі. Характеристики синапсів можуть перебудовуватися проходять через них сигналами так, що синапси навчаються в залежності від типів протікають в них процесів.

Інтенсивність вихідного сигналу залежить як від рівня вхідних сигналів, так і провідності відповідних синаптичних зв'язків. Інформація між нейронами передається за допомогою короткої серії імпульсів. Повідомлення передається за допомогою частотно-імпульсної модуляції, при цьому частота може змінюватися від одиниць до тисяч імпульсів в секунду. Як видно, за швидкістю обробки інформації нейрон істотно поступається сучасним електронним схемам, однак, як ми вже відзначали, висока швидкість обробки інформації в мозку забезпечується розпаралелюванням протікають в ньому процесів.

Штучна нейронна мережа – це машина, яка спроектована для моделювання функцій мозку і подібно до нього має багатошарову ієрархічну структуру і здатність до навчання [4]. Вузли штучної нейронної мережі, іменовані також штучними нейронами (нейронними клітинами, формальними нейронами) представляють собою елементарні процесори і є спрощеними моделями біологічних нейронів.

Функціонування ШНМ відображає роботу мозку в двох аспектах [4]:

– всі знання накопичуються навколишнього серед процесі навчання;

– навчання відбувається шляхом зміни (цілеспрямованого або випадкового) сили зв'язку між нейронами (синаптичних ваг) або топології (архітектури) мережі.

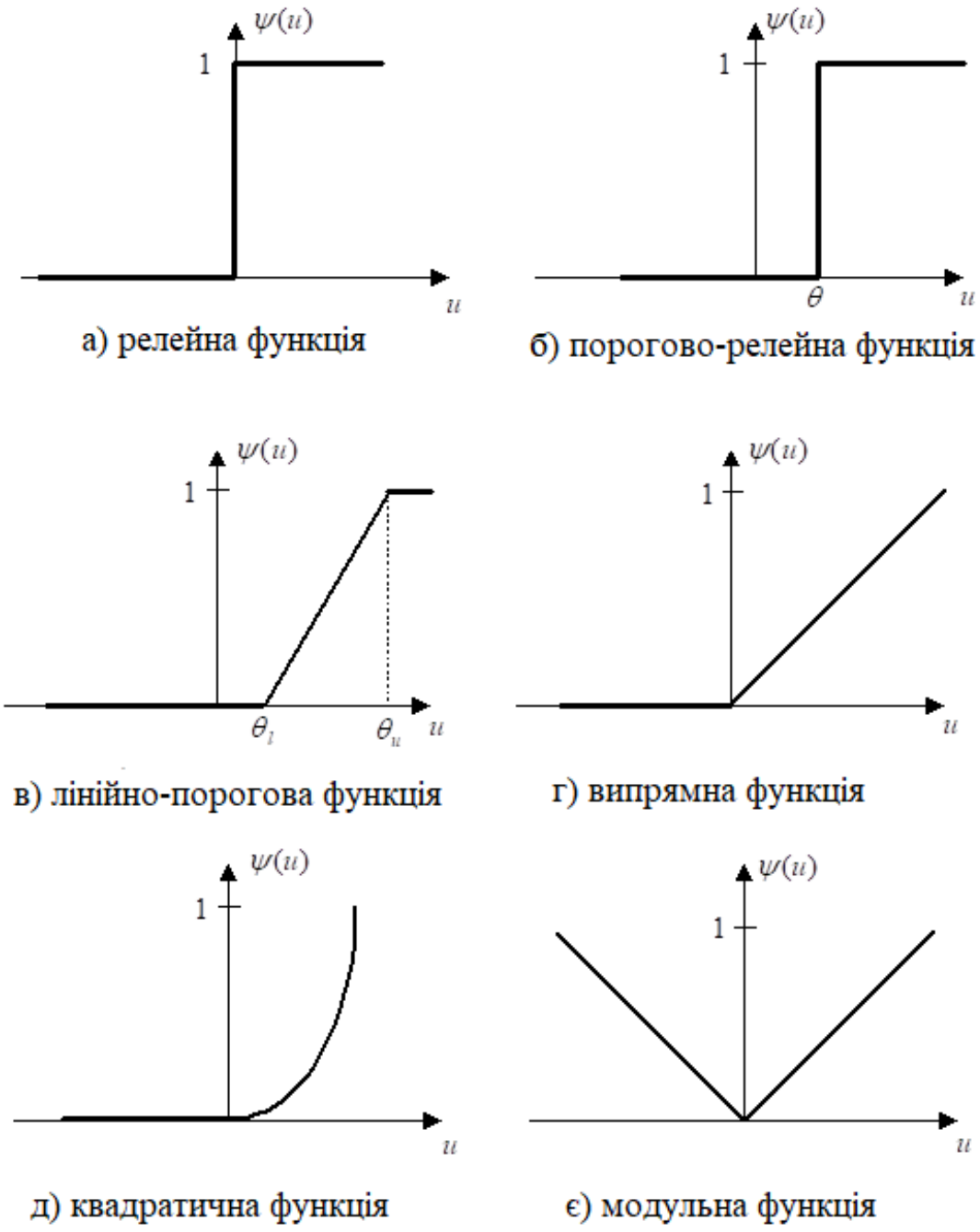
На кожен вхід нейрона подається сигнал, при цьому з кожним входом пов'язаний так звана синаптична вага. У тілі нейрона обчислюється примітивна функція суми добутку вхідних сигналів і синаптичних ваг, тобто фактично реалізується нелінійне відображення багатовимірного простору входів в скалярний вихід. Процедура, за допомогою якої відбувається навчання (настройка) окремого нейрона або нейромережі в цілому, називається алгоритмом навчання. У процесі навчання відбувається зміна (адаптація) синаптичних ваг, а можливо і топології ШНМ так, щоб вихідний сигнал був відповідним до деякого апріорі заданого критерію якості, що характеризує процес вирішення мережею конкретного завдання.

Важливою вимогою до процесу навчання є його адаптивність, яка забезпечує настройку ваг відповідно до характеристик навколишнього середовища. Якщо ці характеристики змінюються (при цьому не виключається можливість зміни і самої розв'язуваної задачі), мережа повинна в реальному часі «перенавчитися» відповідно до нових умов.

Більшість нейронних мереж утворено однотипними нейронами – це гомогенні (однорідні) мережі, хоча відомі гетерогенні мережі, сконструйовані з різних нейронів. Нейрони бувають аналоговими і бінарними, хоча цей поділ чисто умовно, оскільки один і той же формальний нейрон може функціонувати як в аналоговому, так і в цифровому режимах.

Функціональні характеристики окремих нейронів і мереж в цілому визначаються видом використовуваних активаційних функцій. Так, якщо первісна модель використовувала бінарний обмежувач (релейний функцію), то в даний час використовується безліч інших перетворень.

Найбільш розповсюджені і популярні на сьогодні функції активації наведено на рисунку 1.3 [4].



а) релейна функція

б) порогово-релейна функція

в) лінійно-порогова функція

г) випрямна функція

д) квадратична функція

є) модульна функція

Рисунок 1.3 – Приклади функцій активації

В нейронних мережах найбільшого поширення набула сигмоїдальна функція, яка зображена на рисунку 1.4. Характеристики цієї функції в значній мірі залежать від параметра крутизни (з ростом значення параметра функція наближається до релейної, не зазнаючи при цьому розриву в нульовій точці).

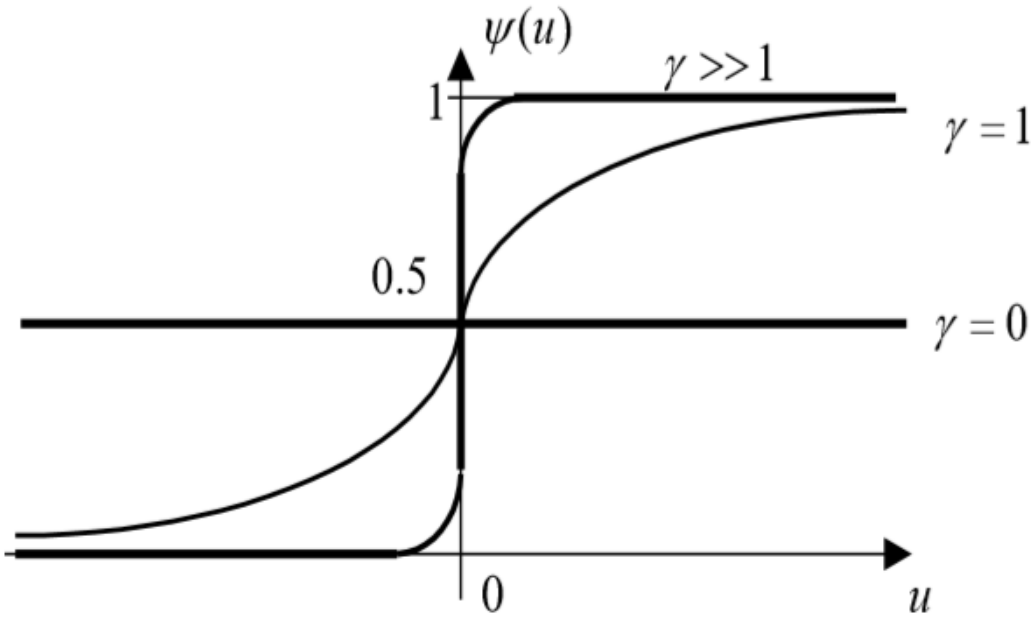


Рисунок 1.4 – Сигмоїдальна функція активації

1.4 Рішення задачі кластерування і топологічної структури на основі нейронних мереж

Класичні методи кластерування можливо реалізовувати на основі нейронних мереж. У дипломній роботі запропоновано самонавчанну нейро-фаззі систему для кластерування великих масивів даних. Також вона застосовується для відображення топологічної структури вхідних даних – перетворення вхідного багатовимірного вектора даних у двовимірний чи тривимірний для зручності подання та аналізу даних.

Існує декілька підходів для вирішення запропонованих завдань. Алгоритм на основі самоорганізованої мапи Когонена. SOM (Kohonen’s Self-Organizing Maps) є методом проектування багатовимірного простору в простір з більш низькою розмірністю з визначеною структурою. Суттєвим недоліком є те, що зменшується розмірність вихідної задачі і остаточний результат роботи нейронних мереж залежить від початкових установок мережі. Наприклад, необхідно заздалегідь визначити кількість кластерів. У зв’язку з цим, виникають дефекти проектування, аналіз яких є досить

складною задачею.

Альтернативою цьому підходу є поєднання конкурентного навчання Хебба і нейронного газу [6]. Це поєднання більш ефективно в побудові топологічної структури, але практичному застосуванню цього підходу перешкоджає ряд проблем: необхідні апріорні знання про розмір мережі і складність застосування методів адаптації швидкості навчання мережі, зайва адаптація призводить до зниження ефективності при навчанні новими даними, а надто повільна швидкість адаптації викликає високу чутливість до збурених даних. Для задач послідовного навчання, перераховані вище методи не підходять.

Таким чином фундаментальна проблема для таких завдань – це адаптація мережі до нової інформації без пошкодження або знищення вже відомої. Тобто неможливо навчати нейронну мережу в режимі послідовного подання даних.

Метою магістерської атестаційної роботи є створення самоорганізованої нейро-фаззі системи, яка навчається послідовно, на основі існуючого алгоритму SOINN (Self-Organizing Incremental Neural Network). По суті запропонований алгоритм є нечіткою версією SOINN.

Об'єктом дослідження є процес обробки даних, що надходять одне за одним, в послідовному режимі за допомогою FSOINN (Fuzzy Self-Organizing Incremental Neural Network).

Предметом дослідження є методи нечіткого кластерування та топологічної структури з послідовною обробкою в задачах інтелектуального аналізу даних.

Для побудови штучних нейронних мереж використовуються методи, які базуються на теорії обчислювального штучного інтелекту. Архітектура нечіткої самоорганізованої мапи заснована на основі SOINN і дозволяє в онлайн режимі проводити кластерування і перетворювати багатовимірні дані в двовимірні. Імітаційне моделювання застосовується для перевірки якості кластерування з використанням архітектури FSOINN. Також

необхідно порівняти якість розробленого метода кластерування з іншими алгоритмами на синтетичних та реальних наборів даних.

1.5 Постановка задачі

На основі проведеного аналізу предметної області можна сформулювати завдання, які необхідно вирішити:

− провести аналіз існуючих методів інтелектуального аналізу даних для розв'язання задачі послідовного кластерування та подання топологічної структури потоків даних;

− розробити метод і архітектуру інкрементної самоорганізованої багатошарової нейронної мережі для потокового кластерування та подання топології даних;

− на основі SOINN розробити нечітку нейронну мережу для визначення для кожного прикладу ступеню належності до кожного кластеру;

− провести імітаційне моделювання та порівняльний аналіз розробленої нейронної системи з існуючими на даний момент методами.

2 ШТУЧНІ НЕЙРОННІ МЕРЕЖІ. КОНКУРЕНТНЕ НАВЧАННЯ САМООРГАНІЗОВАНИХ МАП

Процес навчання штучних нейронних мереж розглядається як адаптація параметрів, а можливо і архітектури мережі для вирішення поставленого завдання шляхом оптимізації прийнятого критерію якості. На сьогодні ШНМ прийнято класифікувати за такими ознаками [9]:

− по типу нейронів-вузлів, що утворюють мережу;
− за способом навчання;
− по архітектурі (топології) мережі;
− по функціям, які реалізовані мережею.

Досить широкого поширення набула парадигма конкурентного навчання. Окремі нейрони вихідного шару такої мережі змагаються за право активації, в результаті чого активним виявляється нейрон в мережі або в групі. Вихідний нейрон, який виграв це змагання, називається переможцем. Такий принцип навчання називають «переможець отримує все».

Описана парадигма навчання є основою особливого класу штучних нейронних мереж, який носить назву самоорганізовані мапи. Розробка цієї моделі нейронної мережі обумовлена відмінною властивістю людського мозку. Він організований таким чином, що окремі сенсорні входи подаються впорядкованими обчислювальними мапами. Зокрема, такі сенсорні входи, як нервові закінчення тактильної системи, слуху та зору, топологічно впорядковано відображаються на різні контури церебральної кори мозку. Таким чином, обчислювальне відображення є цеглинкою в інфраструктурі обробки інформації нервової системи. Мапа обчислень, в свою чергу, утворена масивом нейронів, які перетворюють вхідні сигнали в просторовокодовані розподілення ймовірності, що представляють обчислювальні значення параметрів вузлів. В результаті, отримана інформація, представляється у вигляді вузлів з відносно простою схемою з'єднання.

2.1 Самоорганізовані мапи

Самоорганізовані мапи характеризуються формуванням топографічних мап вхідних образів, в яких просторове розташування нейронів решітки є індикатором вбудованих статистичних ознак, що містяться у вхідних прикладах.

Розглянутий спеціальний клас нейронних мереж в основному використовується для вирішення завдань кластерування і гетероасоціації. Найбільш відомою з цих мереж є самоорганізована карта Т. Когонена (SOM), що реалізує відображення вхідного простору X за допомогою деякого оператора F в вихідний простір Y[9].

Самоорганізована карта має вкрай просту архітектуру з прямою передачею інформації і крім нульового рецепторного шару містить єдиний шар нейронів, іменований іноді шаром Когонена [6]. Кожен нейрон шару Когонена пов'язаний з кожним рецептором нульового шару прямими зв'язками і з усіма іншими нейронами поперечними латеральними зв'язками. Саме латеральні зв'язки забезпечують збудження одних нейронів і гальмування інших.

Завдяки такій організації мережі, кожен нейрон отримує всю інформацію про аналізовані образи і генерує на своєму виході відповідний відгук, після чого в шарі Когонена виникає режим конкуренції, в результаті якого визначається єдиний нейрон-переможець з максимальним вихідним сигналом. Цей сигнал по латеральним зв’язків забезпечує збудження найближчих «сусідів» переможця і придушення реакції далеко віддалених вузлів. Таким чином в процесі конкурентного самонавчання формуються групи нейронів, кожен з яких максимальним відгуком реагує на образи з відповідних підобластей вхідного простору, що дозволяє мапі Когонена крім уже зазначених проблем кластеризації справлятися з компресією великих обсягів інформації.

Самоорганізована карти може мати різну топологію, хоча найбільш

часто рецептори і нейрони розташовуються у вузлах одне (1D) - або двовимірної (2D) решітки.

Можливі, звичайно, решітки і більшої розмірності, проте в основному на практиці використовуються структури, які виконують перетворення вхідного сигналу-образу довільної розмірності в одно- або двовимірну карту, що зберігає деяким чином топологічну впорядкованість вхідного простору.

Як видно з рисунка 2.1, вхідний простір X розбито на підобласті X_i, які накриваються «мапою» для візуалізації отриманих результатів. При цьому будь-який образ, що належить підобласті X_i, збуджує тільки один нейрон w_j^* самоорганізованної мережі. Після цього розраховується вихідний сигнал Y.

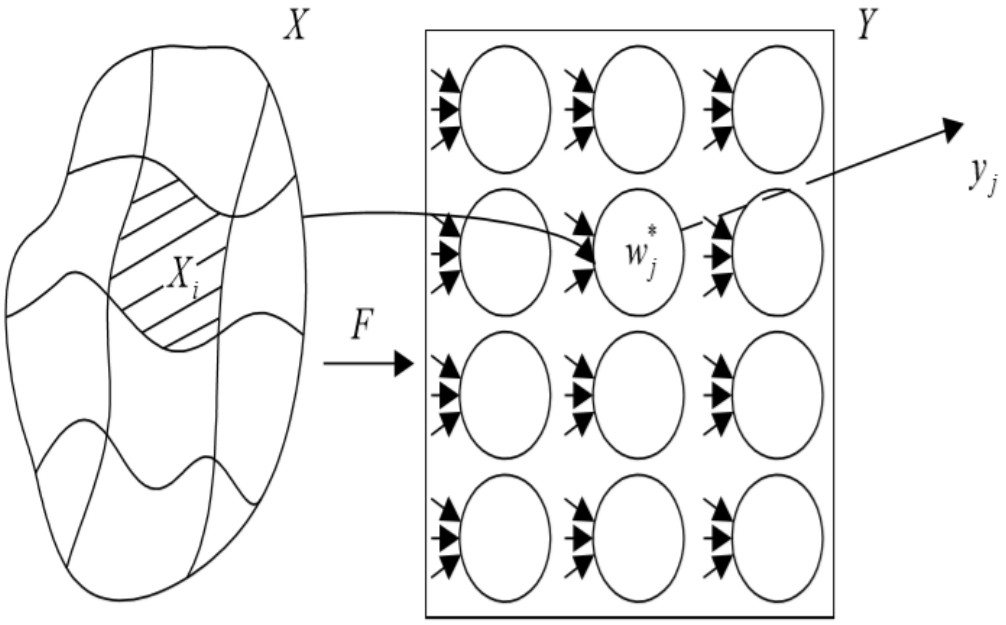


Рисунок 2.1 – Відображення вхідного простору в вихідний

Кожен нейрон мережі отримує n-мірний вхідний вектор x і генерує на своєму виході сигнал y_i, який залежить від вектора синаптичних ваг w_j, налаштованих за допомогою алгоритму самонавчання на певну область

вхідного простору X_i.

Найближчі в сенсі використовуваної метрики вхідні вектори x(k) і x(p) можуть порушувати або один і той же нейрон w_j, або два нейрона-сусіда, наприклад, w_j і w_{j+1} чи w_j і w_{j-1}. У деяких випадках нейронпереможець w_j^* з максимальним вихідним сигналом y_i може порушувати і своїх найближчих сусідів так, що $y_i > y_{i+1}$, $y_{i-1} > y_{i+2}$, $y_{i-2} > \ldots$.

Самоорганізовані мапи можуть мати різну топологію, хоча найбільш часто рецептори і нейрони розташовуються у вузлах одно (1D) – або двовимірної (2D) решітки. Найпростіша мапа Когонена, наведена на рисунку 2.2 і має 1D топологію, рецепторів і нейронів в шарі Когонена, кожен з яких характеризується власним вектором синаптичних ваг, при цьому сам нейрон – це, як правило, або адаптивний лінійний асоціатор але можливі і інші.

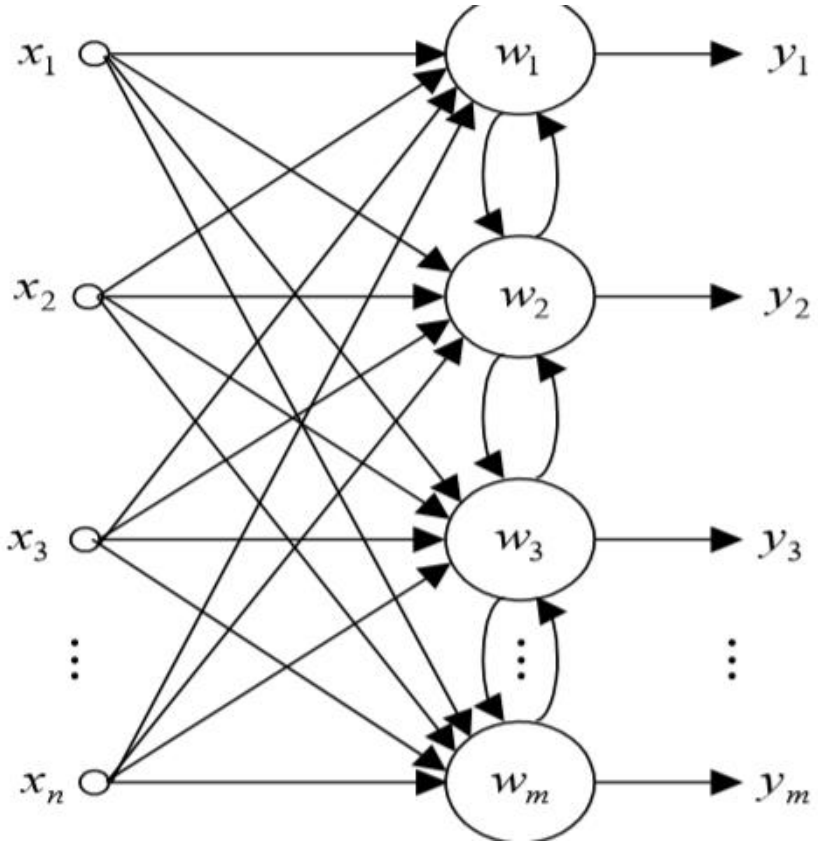


Рисунок 2.2 – 1D-мапа Когонена

Як і будь-яка інша процедура навчання, робота алгоритму починається з ініціалізації синаптичних ваг мережі, які зазвичай вибираються за допомогою генератора випадкових чисел і при цьому бажано для коректної роботи алгоритму, щоб для кожного з нейронів виконувалася умова $\|w_j(0)=1\|$.

Процедура самоорганізації реалізується в три основних етапи: конкуренції, кооперації і синаптичної адаптації і починається з аналізу образу x(k), що надходить з рецепторного шару на всі нейрони прошарку Когонена.

Для кожного з нейронів обчислюється відстань

$$D\left(x(k), w_j(k)\right)=\|x(k)- w_j(k)\|, \qquad (2.1)$$

причому, якщо входи попередньо нормовані так, що $\|x(k)\|=1$, а в якості відстані (2.1) використовується евклідова метрика.

Мірою належності між векторами і може служити їх скалярний добуток

$$D\left(x(k), w_j(k)\right)= x^T(k)w_j(k)= \cos\left(x(k), w_j(k)\right)= \cos\theta_j. \qquad (2.2)$$

Далі визначається нейрон-переможець, «найближчий» до вхідного образу.

Він буде вважатися активованим, якщо в мережі від вхідного образу до нього мінімальна відстань, тобто ребро між двома образами має найменшу довжину

$$D\left(x(k), w^*(k)\right)= \min_j D(x(k), w_j(k)). \qquad (2.3)$$

Після чого в найпростішому випадку, «перестрибуючи» через етап

кооперації, можна підлаштувати синаптичні ваги мережі за допомогою наступного правила навчання

$$w_j(k+1)=\begin{cases} w_j(k)+\eta(k)(x(k)+w_j(k) \\ w_j(k) \end{cases}, \qquad (2.4)$$

де перше виконується у випадку, якщо j-тий нейрон переміг, а друге в іншому випадку.

Нескладно побачити, що процедура (2.4) реалізує принцип «переможець отримує все», при цьому вектор синаптичних ваг нейронапереможця «підтягується» до вхідного вектору на відстань, що визначається кроком пошуку $\eta(k)$. Регулювання величини кроку зазвичай проводиться, виходячи з емпіричних міркувань [9], а загальна рекомендація зводиться до того, що він повинен монотонно зменшуватися в процесі самонавчання.

Однією з особливостей мапи Когонена є наявність етапу кооперації в процесі самоорганізації, коли нейрон-переможець визначає так звану локальну область топологічного сусідства, в якій порушується не тільки він сам, але і його найближче оточення, при цьому більш близькі до переможця нейрони збуджуються сильніше, ніж віддалені.

Ця топологічна область визначається функцією належності $u(j, 1)$, яка залежить від відстані $D\left(w_j(k), w_1(k)\right)$ між переможцем $w_j^*(k)$ і будь-яким з нейронів шару Когонена m і деякого параметра, що задає її «ширину». Як правило $u(j, 1)$ – це потенційна (ядерна) функція, симетрична щодо максимуму в точці $D\left(w_j(k), w_j^*(k)\right)=0$ і приймаюча в ній середнє одиничне значення, монотонно спадаюча зі зростанням відстані і прагне до нуля при відстані $\left(w_j(k), w_1(k)\right)\to\infty$. Найчастіше в якості функції належності використовується функція Гауса, конус (трикутник), параболоїд (перевернута квадратична функція), «мексиканський капелюх»

і цілий ряд інших[9]. Використання функцій сусідства призводить до модифікованого правилом навчання Когонен

$$w_1(k+1)=w_1(k)+\eta(k)u(j,1,k)(x(k)-w_1(k)), \qquad (2.5)$$

реалізує принцип «переможець отримує більше» замість традиційного «переможець отримує все».

Звісно, що при u(j,1,k)=1 приходимо до стандартного алгоритму (2.4), що забезпечує на кожному такті настройку тільки одного нейрона $w_j^*(k)$.

Використання ж в якості ядерних функцій u(j,1) веде до того, що усі нейрони мережі в більшій чи меншій мірі підтягують вектори своїх синаптичних ваг до поточного образу. Алгоритм нейронного газу – підхід до самоорганізації мережі Когонена заснований на використанні порядкових статистик[1]. При цьому підході всі нейрони ранжуються в порядку зростання відстаней $D\left(w_j(k), w_1(k)\right)$ так, що

$$D\left(x(k), w^*(k)\right) < D\left(x(k), w^1(k)\right) M<\ldots<D\left(x(k), w^{m-1}(k)\right), \qquad (2.6)$$

де верхній індекс позначає ранг $R(w_1(k))$ кожного нейрона в шарі після пред'явлення образу x(k), тобто

$$R\left(w^*(k)\right)=0<R\left(w^1(k)\right)=1<\ldots<R\left(w^{m-1}(k)\right)=m-1. \qquad (2.7)$$

Нескладно побачити, що при нормованих входах зручно використовувати ранжування

$$1 \geq \cos\theta_1(k) >\cos\theta_2(k) >\ldots> \cos\theta^{m-1}(k) \geq -1 \qquad (2.8)$$

або

$$y^{*}(k) > y^{1}(k) > y^{2}(k) > \ldots > y^{m-1}(k). \qquad (2.9)$$

Для кожного з нейронів визначається значення функції сусідства

$$u\big(x(k), w_1(k)\big) = \exp\left(\frac{R(w_1(k))}{\lambda(k)}\right), \qquad (2.10)$$

де $\lambda(k)$ – параметр ширини.

Уточнення синаптичних ваг здійснюється згідно з формулою

$$w_1(k+1) = w_1(k) + \eta(k) u(x(k), w_1(k))(x(k) - w_1(k)). \qquad (2.11)$$

Нескладно побачити, що параметри нейрона з нульовим рангом (фактично нейрона-переможця) при цьому уточняються за допомогою процедури (2.4).

Процес самоорганізації має дві тимчасові фази: початкова фаза упорядкування, в якій відбувається топологічний розбиття вхідного простору, і подальша фаза збіжності, в якій здійснюється точна настройка синаптичних ваг.

Після закінчення цього процесу нейронна мережа в принципі може вирішувати поставлені завдання без уточнення ваг, однак, якщо з'явиться вхідний образ, який не буде віднесений до жодного з сформованих кластерів, картою повинен бути утворений додатковий нейрон в прошарку Когонена, що несе інформацію про цей образ, при цьому дуже бажано, щоб знову включився процес самонавчання.

Оскільки процес навчання карт Когонена відбувається значно швидше, ніж настройка багатошарової мережі за допомогою зворотного поширення помилок [6], ці штучні нейронні мережі найбільш ефективні для роботи в реальному часі, коли настройка синаптичних ваг і обробка вхідних сигналів протікають паралельно. Тобто всі дані поступаюсь по черзі у вигляді сигналів і немає необхідності перенавчати мережу, вибірка

доповнюється новими прикладами. Це є суттєвою перевагою при роботі з великими наборами даних.

2.2 Вирішення завдання кластерування на основі самоорганізованих мап

Кластерування (або кластерний аналіз) – це задача розбиття множини об'єктів на групи, які називаються кластерами. Усередині кожної групи повинні виявитися «схожі» об'єкти, а об'єкти різних груп повинні бути якомога більш відмінні. На відміну від звичайної класифікації, де кількість груп об'єктів фіксоване і заздалегідь визначено в наборі даних, тут немає інформації про групи і їх кількість, тобто заздалегідь нічого не визначено і групи формуються в процесі роботи системи виходячи з певної міри близькості об'єктів.

Дане завдання знаходить широке практичне застосування. Наприклад, в області медицини завдання кластерування може допомогти ідентифікувати центри клітин на зображенні групи клітин. Використовуючи GPS-дані мобільного пристрою, можна визначити найбільш відвідувані користувачем локації в межах певної території. Також завдяки кластеруванню можна знаходити аномалії.

Для будь-якого набору даних, яким не присвоєно мітки, кластерування допомагає виявити наявність певної структури, що вказує на те, що дані піддаються угрупованню. Існує кілька основних методів розбиття груп об'єктів на кластери. У даній роботі описаний FSOINN, який відноситься до особливого класу нейронних мереж – самоорганізовані мапи, які не тільки знаходять особливості структури і взаємозв’язки в даних, а також здатні проводити компресію для відображення в двомірній площині без втрати важливої інформації.

На рисунку 2.3 мірою подібності – є відстань, де два або більше об'єктів належать до одного кластеру, якщо вон розташовані близько один

до одного відповідно до заданої метрики. Відстань рахується від об’єкту да центроїду кластеру. Кількість центроїдів може рахуватися автоматично або вибиратися вручну. Це кластерний аналіз на підставі відстані.

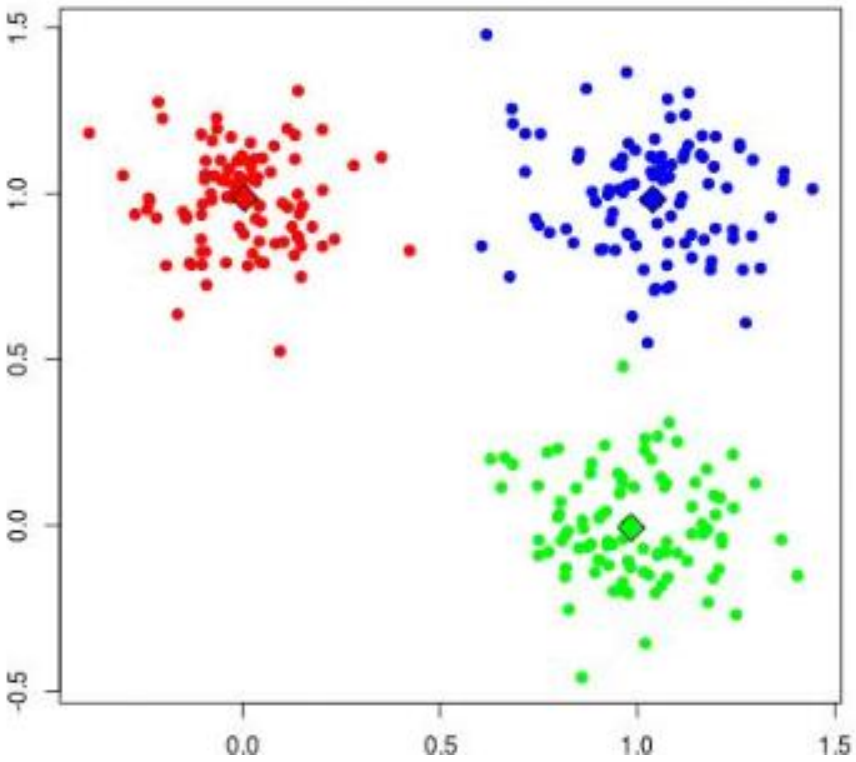


Рисунок 2.3 – Приклад результатів кластерування

Об'єднання або метод деревовидного кластерування використовується при формуванні кластерів або відстані між об'єктами. Ці відстані можуть визначатися в одновимірному або багатовимірному просторі.

Найбільш прямий шлях обчислення відстаней між об'єктами в багатовимірному просторі полягає в обчисленні евклідових відстаней. Якщо є дво- або тривимірний простір, то це є реальною геометричною відстанню між об'єктами в просторі. Метрика – функція, що визначає відстань в метричному просторі. В даному просторі визначено відстань між будь-якою парою елементів. Виникає проблема вибору метрики. Вибір метрики – найбільш важливий фактор, що впливає на результати

кластерного аналізу. Залежно від типу ознак використовуються різні міри близькості (метрики).

Нехай початкова інформація задається у вигляді матриці X і є образи X_i і X_k в N-мірному просторі ознак. Нижче наведені основні метрики, які використовуються при кластерному аналізі.

Евклідова відстань використовується для кількісних ознак. Ця метрика є найбільш поширеною. Це геометрична відстань в багатовимірному просторі і обчислюється таким чином:

$$d_{ik}=\left(\sum_{j=1}^{N}(x_{ij}-x_{kj})^2\right)^{\frac{1}{2}}. \quad (2.12)$$

Відстань міських кварталів (Манхеттенський відстань). Це відстань є середнім різниць по координатах. У більшості випадків ця міра відстані приводить до таких же результатів, як і відстань Евкліда. Однак, для цього заходу вплив окремих великих різниць (викидів) зменшується. Манхеттенський відстань обчислюється за формулою:

$$d_{ik}=\sum_{j=1}^{N}\left|x_{ij}-x_{kj}\right|. \quad (2.13)$$

Міра подібності Хеммінга використовується тільки для якісних ознак. У формулі 2.14 аргумент n_{ik} −це кількість співпадаючих ознак в X_i і X_k образах:

$$\mu_{ji}^{H}=\frac{n_{ik}}{N}. \quad (2.14)$$

Відстань Махаланобіса визначає відмінність між векторами і не залежить від масштабу. Вона широко застосовується в задачах кластерування та класифікації: для того, щоб визначити, до якого з відомих класів відноситься точка, необхідно знайти коваріаційні матриці w для всіх класів і взяти клас з найменшою відстанню до точки:

$$d_{ik}^{M}=(x_{ik}-x_{ki})^{T}W^{-1}(x_{ik}-x_{ki}). \quad (2.15)$$

Внутрікласові відстані, як правило, менше ніж класові. Результат кластерування істотно залежить від метрики, яку експерт задає суб'єктивно.

Існує велика кількість типів кластерних структур: стрічкові кластери, кластери з центром, кластери можуть з'єднуватися перемичкою, що перекриваються кластери, розріджені кластери. Кожен метод кластеризації має свої обмеження і виділяє кластери лише деяких типів. Поняття «тип кластерної структури» залежить від методу і не має формального визначення.

Кластерні алгоритми можна класифікувати на масштабовані і немасштабовані. Масштабованість – одна із найважливіших властивостей алгоритму, залежне від його обчислювальної складності та програмної реалізації. Тобто алгоритми даного класу Робота в обмеженому обсязі оперативної пам'яті комп’ютера. Існує і більш ємне визначення. Алгоритм називають масштабується, якщо при незмінній місткості оперативної пам'яті зі збільшенням числа записів в базі даних час його роботи зростає лінійно.

Але далеко не завжди потрібно обробляти надвеликі масиви даних. Тому в теорії кластерного аналізу питань масштабованості алгоритмів уваги практично не приділялося. Передбачалося, що всі оброблювані дані будуть уміщатися в оперативній пам'яті, головний упор завжди робився на поліпшення якості кластерування.

Але завжди повинен бути баланс між високою якістю кластерування і масштабованість. Тому в ідеалі в арсеналі Data Mining повинні бути присутніми як ефективні алгоритми кластерування мікромассивів (microarrays), так і масштабовані для обробки надвеликих баз даних (large databases).

За способом розбиття на кластери алгоритми бувають двох типів: ієрархічні і неієрархічні. Класичні ієрархічні алгоритми працюють тільки з категорійними атрибутами, коли будується повне дерево вкладених кластерів. Тут поширені агломеративні методи побудови ієрархій кластерів – в них проводиться послідовне об'єднання вихідних об'єктів і відповідне зменшення числа кластерів. Ієрархічні алгоритми забезпечують порівняно високу якість кластерування і не вимагають попереднього завдання кількості кластерів.

Неієрархічні алгоритми засновані на оптимізації деякої цільової функції, яка визначає оптимальний в певному сенсі розбиття множини об'єктів на кластери. У цій групі популярні алгоритми сімейства k-середніх (k-середніх, нечіткий с-середніх, алгоритм Густафсона-Кесселя), які в якості цільової функції використовують суму квадратів зважених відхилень координат об'єктів від центрів шуканих кластерів. Кластери шукаються сферичної або еліпсоїдної форми.

3 САМООРГАНІЗОВАНА НЕЙРО-ФАЗЗІ СИСТЕМА ТІ ЇЇ ПОСЛІДОВНЕ НАВЧАННЯ

3.1 Самоорганізована нейрона мережа

Самоорганізована нейрона мережа використовується для навчання без вчителя в онлайн режимі, а це означає, що вона здатна обробляти великі масиви даних. Ключове завдання полягає в розділенні немічених нестаціонарних вхідних даних в різні класи без попереднього знання о кількості існуючих класів і основних характеристиках вибірки даних. Також завдяки використанню запропонованого алгоритму є можливість вивчити топологію вхідних даних. Цілі запропонованого алгоритму:

− обробити нестаціонарні дані в режимі онлайн;

− не використовуючи жодних попередніх умов, таких як відповідна кількість прикладів у вибірці або скільки класів існує, проводити навчання без учителя, повідомляти про відповідну кількість класів і представляти топологічну структуру щільності імовірнісних значень вхідних даних;

− для того, щоб відокремити класи з перекриттям низької щільності і виявити основну структуру кластерів.

Для реалізації цих цілей потрібно визначити ключові аспекти, а саме концепція введення нових вузлів, адаптивна швидкість навчання та видалення вузлів з низькою щільністю. SOINN (self organizing incremental neural network) [1] використовує двошарову конкурентну мережу (рисунок 3.1). Перший шар аналізує щільність розподілу вхідних даних і використовує вузли та ребра для представлення цього розподілу[12]. Другий шар розділяє дані на кластери шляхом розташування області вхідних даних з низькою щільністю і використовує менше вузлів ніж перший шар. Коли навчання другого шару завершено, SOINN підлаштовує кількість кластерів і видає прототипи вузлів. Однаковий алгоритм використовується навчання для першого та другого шарів.

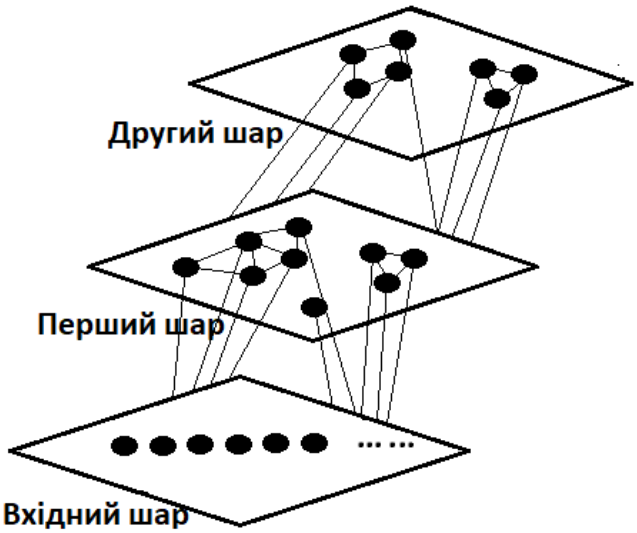


Рисунок 3.1 – Архітектура SOINN

Для завдання класифікації без учителя необхідно визначити, чи відноситься вхідний приклад до раніше вивчених кластерів або до нового кластеру. Припустімо, що два приклади належать до одного кластеру, якщо евклідова відстань між ними менша за порогову відстань T. Якщо T занадто велике, всі зразки будуть віднесені до одного кластеру. Якщо T занадто мале, кожен зразок буде утворювати ізольований одиночний кластер. Щоб отримати «природні» кластери, T має бути більшим ніж відстань всередині кластера і менша, ніж відстань між кластерами.

Для обох шарів обчислюється порогова відстань T [2]. У першому шарі встановлюється вхідний сигнал як новий вузол (перший вузол нового кластеру), коли відстань між сигналом і найближчим вузлом (або другим найближчим вузлом) перевищує порогове значення T, яке постійно адаптується до поточної інформації. У другому шарі розраховується відстань всередині кожного кластера і відстані між кластерами на основі тих вузлів, що були згенеровані в першому шарі. В результаті отримується постійна порогова відстань T_c відповідно до відстаней всередині кластера та відстаней між кластерами.

На рисунку 3.2 представлена блок-схема алгоритму SOINN.

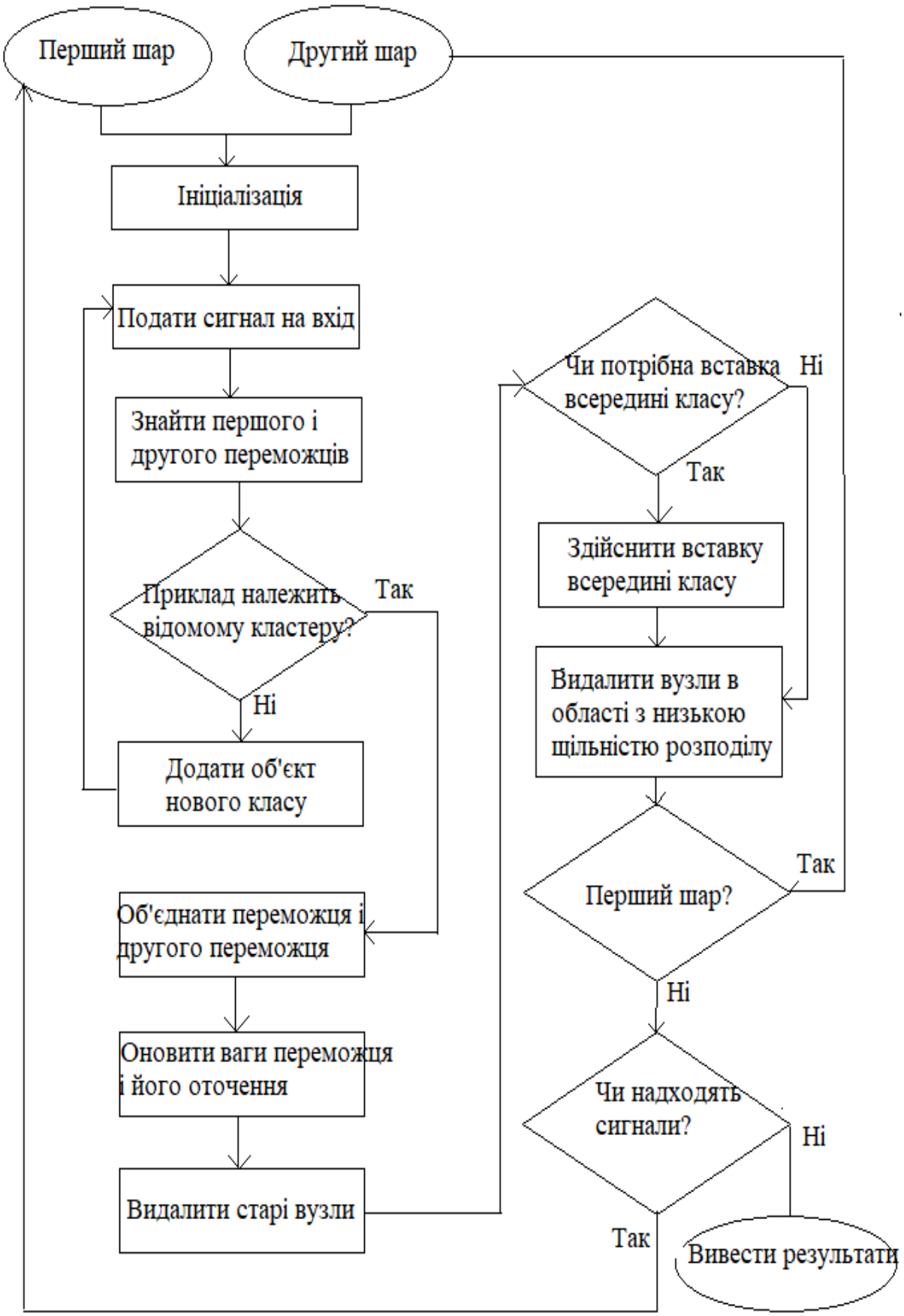


Рисунок 3.2 – Блок-схема алгоритму SOINN

Для того, щоб представляти топологічну структуру для завдання онлайн навчання, адаптація до зростання вхідних даних при збереженні старих прототипів є важливою особливістю. Тому вставка нових вузлів є дуже корисним внеском у пластичність і стабільність метода. Вставка повинна бути зупинена, щоб заборонити постійне збільшення кількості вузлів та уникнути перенавчання нейронної мережі. З цієї причини необхідно вирішити, коли і як вставити новий вузол в межах одного кластера, і коли вставка буде зупинена.

Для вставки в межах кластера використовується така схема: вставляється вузол між вузлом q з максимальною накопиченою помилкою та вузлом f, який є серед сусідів q з максимальною накопиченою помилкою. Коли вставляється новий вузол, оцінюється вставка за параметром корисності, тобто чи зменшився радіус накопиченої помилки, щоб визначити, чи потрібна проведена вставка. Ця оцінка гарантує, що вставка нового вузла призведе до зменшення помилки та контролює зростання вузлів, зрештою стабілізуючи їх кількість.

Конкурентне навчання Хебба представляє парадигму для побудови зв'язків між вузлами нейронної мережі [1]. Конкурентне правило Хебба можна описати таким чином: для кожного вхідного сигналу з’єднається два найближчі вузли, вимірювані на основі евклідової відстані. Граф є оптимальною топологією в режимі послідовного навчання та протягом усього процесу навчання мережі вузли, сусідні на ранній стадії, можуть стати не сусідніми на більш розвиненій стадії. Тим самим стає необхідним видаляти з'єднання, які були оновлені дуже давно.

Загалом, кластери можуть перетинатися. Щоб точно визначити число кластерів припускається, що вихідні дані можна розділити: щільність ймовірності в центральній частині кожного кластеру вище, ніж щільність в між кластерами; і перекриття між кластерами має низьку ймовірність. Виділяються кластери, видаливши ті вузли, положення яких знаходиться в регіоні з дуже низькою щільністю ймовірності. Щоб оцінити щільність

ймовірності і визначити регіон з низькою ймовірністю щільності, якщо щільність нижче порога η, вузол видаляється. Тут пропонується стратегія: якщо кількість вхідних сигналів, є цілим числом, то видаляються вузли тільки з одним або без топологічних сусідів. Звідси випливає, що якщо у вузла є лише один сусід чи нема взагалі, то протягом періоду навчання накопичена похибка цього вузла має дуже низьку ймовірність, а вставка нових вузлів поблизу цього вузла є деструктивним: щільність ймовірності регіону, що містить вузол дуже низький.

На основі вищесказаного, можно дати формальний опис алгоритму для навчання мережі SOINN. Цей алгоритм використовується для навчання і першого і другого шарів мережі. Різниця полягає лише в тому, що вхідні дані для навчання другого шару породжуються першим шаром і при навчанні другого шару, вже є знання про топологічної структурі першого шару, для обчислення постійного порога подібності T.

Напочатку необхідно ініціалізувати множину вузлів A, яка буде використовуватися для збереження всіх вузлів мережі, двома вузлами c_1 і c_2, де $A=\{c_1, c_2\}$ та множину ребр C порожньою множиною. Перші два вузла обираються випадково з поданих на вхід прикладів. Подати на вхід новий сигнал $x \in R^n$. R служить пам'яттю для радіуса помилки вузла і в момент вставки. Знайти в множині A вузли переможця s_1 і другого переможця s_2, як вузли з найближчим і наступним за ним векторами ваг W_c, за формулою

$$s_{1,2}= \arg\min_{c \in A} \|x- W_c\|. \quad (3.1)$$

Якщо відстань між x, s_1 та s_2 більше порогів подібності T_{s1} або T_{s2}, то створити новий вузол, додати до множини A і подати на вхід новий сигнал. В іншому випадку створити ребро між s_1 та s_2, якщо воно не існує. Додати нове ребро до множини ребер C та встановити вік ребра між ними рівним 0. Збільшити вік всіх дуг, що виходять з s_1 на 1. Додати відстань між вхідним сигналом і переможцем до сумарної помилки E_{s1}. Збільшити

локальну кількість сигналів вузла s_1:

$$M_{s1}=M_{s1}+1. \tag{3.2}$$

Адаптувати вектора ваг переможця

$$\Delta W_{s1}=\eta_1(t)(x-W_{s1}) \tag{3.3}$$

та його прямих топологічних сусідів

$$\Delta W_i=\eta_2(t)(x-W_i), \tag{3.4}$$

де $\eta_1(t)$ і $\eta_2(t)$ – крок навчання переможця і його сусідів.

Видалити ребра з віком, вище заданого порогового значення. Якщо генеруються досі вхідні сигнали і їх число кратно параметру λ, вставити новий вузол і видалити вузли в областях низької щільності. Спочатку обирається вузол q з максимальною локальною помилкою $q=\arg\max E_c$. Потім серед сусідів q знайти вузол f з максимальною локальною помилкою. Додати до мережі новий вузол r та інтерполювати його у вектор ваг із q і f:

$$W_r=\frac{W_q+W_f}{2.0}. \tag{3.5}$$

Теж саме необхідно зробити з локальною кількістю сигналів, радіусом помилки та локальною помилкою вузла.

Якщо вставка не може зменшити середню помилку цієї локальної області, вставлення не вдалося. Новий вузол r видаляється з набору A і всі параметри відновлюються.

Після λ ітерацій (λ є таймером) SOINN [2] вставляє нові вузли в положення, де накопичена помилка є великою. Скасовується вставка, якщо вона не може зменшити помилку. Вставка називається внутрікласовою

вставкою, тому що новий вставлений вузол знаходиться в межах існуючого класу. Крім того, під час вставки новий клас не буде створено.

Потім SOINN знаходить вузли, кількість сусідів яких менша або дорівнює одиниці і видаляє ці вузли. Після того, як поріг навчання [3] аналізує ітерації першого шару, результати навчання використовуються як вхід для другого шару. Другий шар SOINN використовує той же алгоритм навчання, що і перший.

Для другого шару поріг подібності є постійним. Він розраховується з використанням відстані всередині кластера і на основі відстані між кластерами. При великому постійному порозі подібності помилка накопичення для вузлів другого шару буде дуже високою, а всередині класова вставка грає важливу роль в процесі навчання. При великому постійному порозі подібності другий шар також може видалити деякі «вузли шуму», які залишаються під час навчання на першому рівні.

Щоб видалити вузли, викликані шумом, SOINN видаляє вузли в областях з дуже низькою щільністю ймовірності. Якщо кількість вхідних сигналів, що генеруються досі, є цілим числом, кратним параметру λ, видаліть ці вузли з одним або без топологічного сусіда. Для одновимірних вхідних даних і наборів даних з невеликим шумом SOINN використовує локально накопичену кількість сигналів вузлів-кандидатів для управління поведінкою видалення.

Крім того, двошарова мережева структура допомагає SOINN видаляти вузли викликані шумом, бо перевірка виконується декілька разів з різним порогом подібності.

Швидкість навчання η(t) визначає ступінь, з якою переможець і сусіди переможця адаптуються до вхідного сигналу і, де t – це загальна кількість кроків адаптації.

Для завдань в режимі онлайн навчання загальна кількість кроків адаптації t недоступна, бо данні подаються на вхід по мірі появи. Тому SOINN використовує схему, подібну методу k-середніх, щоб з часом

адаптувати швидкість навчання:

$$\eta_1=\frac{1}{t},\ \eta_2=\frac{1}{100t}. \tag{3.6}$$

Тут параметр t представляє кількість вхідних сигналів, для яких даний конкретний вузол був переможцем, тобто $t=M_i$. Цей алгоритм відомий як k-середніх. Ця схема використовується, тому що вузол стає більш стабільним шляхом зниження швидкості навчання, якщо вузол стає переможцем практично на кожному етапі подачі сигналу в рамках одного кластера.

Коли вхідний вектор подається на вхід SOINN [1], він знаходить найближчий вузол (переможець) і другий найближчий вузол (другий переможець) вхідного вектору. Потім використовуючи порогові критерії подібності мережа визначає, чи належить вхідний вектор кластеру першого переможця або другого.

Перший рівень SOINN адаптивно оновлює поріг подібності кожного вузла, оскільки розподіл вхідних даних невідомий. Якщо вузол і-тий має сусідні вузли, поріг подібності T_i обчислюється з використанням максимальної відстані між вузлом і-тим і сусідніми його вузлами

$$T_i=\max_{j\in N_i}\left\|w_i-w_j\right\|. \tag{3.7}$$

У цьому випадку N_i – множина сусідніх вузлів, а w_i – векторні ваги вузла і.

Поріг схожості T_i визначається як мінімальна відстань між вузлом та іншими вузлами в мережі, якщо вузол і не має сусідніх вузлів

$$T_i=\min_{j\in N\{i\}}\left\|w_i-w_j\right\|, \tag{3.8}$$

де N – множина всіх вузлів.

Вхідний вектор визначається в мережі як новий вузол для

представлення першого вузла нового класу, якщо відстань між вхідним вектором і переможцем або другим переможцем більше, ніж поріг схожості переможця або другого переможця. Якщо вхідний вектор визначений як належний до одного кластеру як у переможця або другого переможця і якщо немає ребра, що з'єднує переможця і другого переможця. Тоді з'єднуємо переможця і другого переможця за допомогою ребра і встановлюємо вік ребра рівним нулю. Надалі збільшуємо вік всіх ребр, пов'язаних з переможцем, на одиницю. Після цього оновлюється вектор ваги переможця та його сусідні вузли [4]. Використаємо і для позначення вузла переможця і M_i, щоб показати час, коли він став переможцем. Зміна ваги переможця Δw_i і зміна ваги Δw_j сусіднього вузла $j \in N_i$ визначаються як:

$$\Delta w_i = \frac{1}{M_i}(w_s - w_i) \tag{3.9}$$

та

$$\Delta w_j = \frac{1}{M_i}(w_s - w_j), \tag{3.10}$$

де w_s – вага вхідного вектору.

3.2 Нечітка інкрементна самоорганізована мапа

Відмінною особливістю нечіткого кластерування є той факт, що кожен об'єкт може відноситися до кожного кластеру з певним ступенем належності. Правильний вибір функції належності є важливим для коректної роботи алгоритму. Аналіз збіжності процесів конкурентного самонавчання, проведений М. Котреля і Дж. Фортом [7], показав, що в процесі настройки синаптичних ваг, повинен зменшуватися не тільки крок пошуку, але і параметр ширини функції сусідства. Чим більша відстань переможця від об’єкта, тим менше корегуєтеся його синаптична вага.

Функція сусідства повинна бути ширше на початку процесу навчання, і її ширина повинна зменшуватися з часом таким чином, щоб до кінця цього процесу здійснювалася підгонка тільки безпосередніх сусідів переможця і другого переможця. В якості функції сусідства найчастіше використовують функцію Гауса [10], параметри якого змінюються від часу, найчастіше лінійно. Ця функція має наступну формулу:

$$(u,1,k)= \exp\left(- \frac{d^2(w_j^*(k),\, w_l(k))}{\sigma^2}\right), \qquad (3.11)$$

де σ – параметр ширини функції належності;

$w_j^*(k)$ – синаптична вага нейрона переможця;

$w_l(k)$ – синаптична вага мережі l=1, 2,…, m.

Цей параметр необхідно налаштовувати і вводити формулу для його розрахунку. Тому в атестаційної роботі запропоновано функцію належності, параметри якої налаштовувалися без введення нових. Використання функцій сусідства призводить до модифікованого правила навчання інкрементної самоорганізованої мапи і реалізує принцип «переможець отримує більше» [7]. Для спостереження x(k), яке поступає на вхід, визначається перший і другий вузол переможець $w_j^*(k)$ і наближається до спостереження за формулою

$$w_j^*(k+1)=w_j^*(k)+\eta(k)u^2(j,l,k)(x(k)-w_j^*(k)). \qquad (3.12)$$

Всі інші вузли (для спрощення обчислень це може бути фіксована кількість вузлів або всі вузли ближче деякої порогової відстані) перераховуються за формулою

$$w_l(k+1)=w_l(k)+\eta(k)u^2(j,l,k)\left(x(k)-w_l(k)\right). \qquad (3.13)$$

Змінна u(j,l,k) – функція належності. На прикладі алгоритму FCM

визначається функція сусідства і розраховується вона таким чином[10]:

$$u_j(k)= \frac{d^{-2}(x(k),c_j)}{\sum_{l=1}^{m} d^{-2}(x(k),c_l)} = \frac{\|x(k)-c_j\|^{-2}}{\sum_{l=1}^{m}\|x(k)-c_l\|^{-2}}. \quad (3.14)$$

де c_l – будь-який з центрів кластерів l=1, 2,…, m.

Після розрахунку функції належності, перевизначаються координати центроїдів кожного кластера:

$$c_j=\frac{\sum_{k=1}^{N} u_j^2(k)x(k)}{\sum_{k=1}^{N} u_j^2(k)}. \quad (3.15)$$

На основі формули (3.14) ділиться чисельник і знаменник на $\sum_{l=1}^{m} d^{-2}(x(k),c_l)$. Також так як запропонована інкрементна самоорганізована мапа не базується на центроїдах, то функція належності залежить від синаптичних ваг $w_l(k)$ і синаптичних ваг переможців мережі $w_j^*(k)$. В результаті перетворень виводиться наступна формула:

$$u_j(k)=\frac{1}{1+\frac{d^2(w_j^*(k),\, w_l(k))}{\gamma_j}}, \quad (3.16)$$

де γ – параметр ширини функції.

Параметр ширини функції розраховується автоматично, тобто непотрібно вводити нові параметри. Він розраховується як величина обернено пропорційна до відстані в квадраті:

$$\gamma_j=\Big(\sum_{\substack{l=1 \\ l\neq j}}^{m} d^2(w_j^*(k),\, w_l(k))\Big)^{-1}. \quad (3.17)$$

Формула (3.16) – це математичне трактування функції належності Коші. Крім того, що не потрібно вводити нових параметрів, ще однією перевагою запропонованої функції належності є те, що на відміну від

функції Гауса значення функції на нескінченності не близько до нуля. Що сприяє тому, що в навчанні беруть участь всі вузли. Запропонована функція належності Коші забезпечує більш широкі можливості процесу самоорганізації.

На рисунку 3.3 зображено порівняння функцій належності Гауса і Коші.

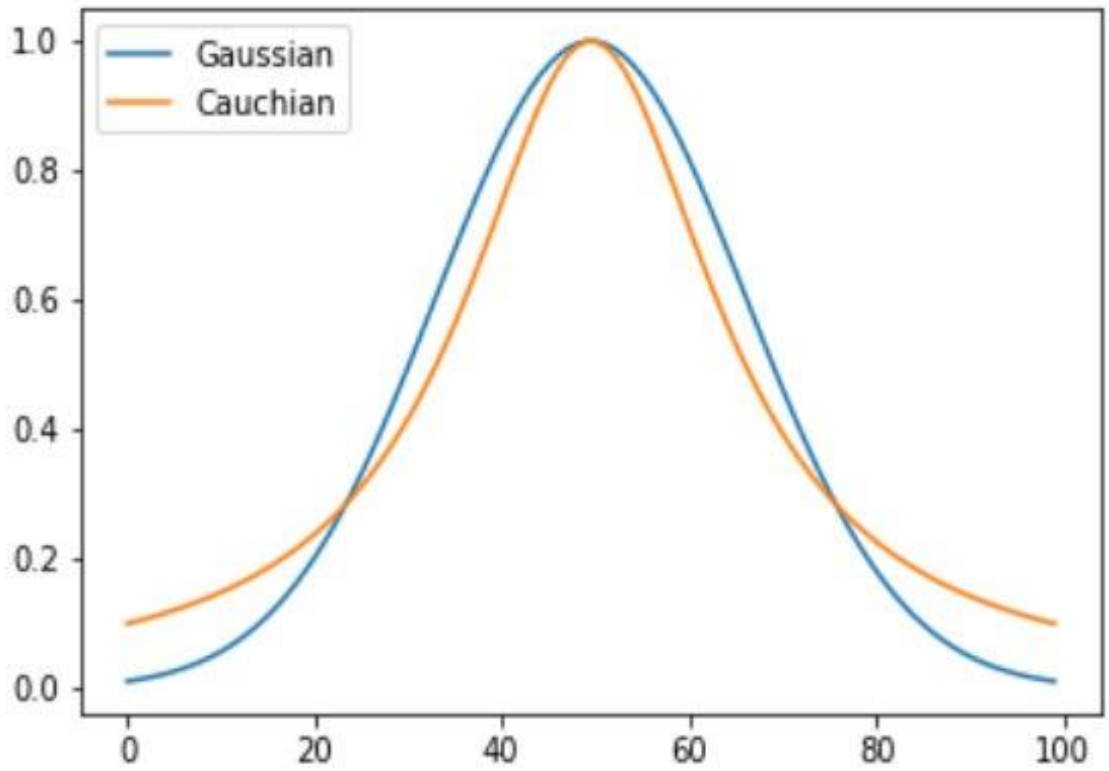


Рисунок 3.3 – Порівняння функцій належності

В процесі роботи нечітка інкрементна самоорганізована багатошарова мапа реагує на отримання спостереження x(k), наближаючи до нього вузол переможець і другого переможця $w_j^*(k)$ відповідно до принципів конкурентного навчання, описаними в підрозділі 2.1. При цьому інші вузли $w_l(k)$, при чому $1 \le l \le m$, $l \ne j$, наближаються до вузла-переможця і другого переможця пропорційно ступеня сусідства між ними. В результаті в навчанні нейронної мережі беруть участь всі поступаючи сигнали на вхід

на кожному кроці змінюється синаптична вага кожного.

Що стосується завдання чіткого кластерування, вихід $y_j(k)=1$, якщо спостереження x(k) належить j-му кластеру, але ясно, що і решта вузлів можуть бути активовані пропорційно до належності до вузла-переможця, забезпечуючи таким чином нечітке кластерування, яке дозволяє вирішувати проблему неоднозначності прикладів на границях кластерів.

4 ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ І ПЕРЕВІРКА ТЕОРЕТИЧНИХ ДОСЛІДЖЕНЬ

Для реалізації нечіткого методу послідовного кластерування на основі інкрементної самоорганізованої нейронної мережі обрано середовище програмування PyCharm і мова програмування Python.

Нижче описуються базові бібліотеки, які були використані в ході написання програми FSOINN. Ці бібліотеки роблять з мови програмування Python потужний і надійний інструмент для аналізу і візуалізації даних. На них ґрунтуються більш спеціалізовані бібліотеки. Іноді їх називають набором SciPy.

NumPy – основна бібліотека Python, яка спрощує роботу з векторами і матрицями. Містить готові методи для самих різних операцій: від створення, зміни форми, множення і розрахунку детермінанта матриць до рішення лінійних рівнянь і сингулярного розкладання. В реалізації FSOINN використовувалась ця бібліотека для створення масивів, в яких зберігався поданий на вхід сигнал і в цьому виді передавався на обробку. Також застосовувалися функції знаходження суми, максимального і мінімального значення, наприклад, для формул (3.1) і (3.17).

Pandas – потужним інструментом для аналізу набору даних. Пакет дає можливість будувати зведені таблиці, виконувати угруповання, надає зручний доступ до табличних даних, а при наявності пакета matplotlib дає можливість малювати графіки на отриманих наборах даних.

SciPy – бібліотека ґрунтується на NumPy і розширює її можливості. Включає методи лінійної алгебри і методи для роботи з ймовірними розподілами, інтегральним обчисленням і перетвореннями Фур'є. Також містить підпрограми, такі як чисельна інтеграція і оптимізація, для підвищення продуктивності. Як і всі бібліотеки Python описана в документації. Використовується бібліотека scipy.sparse.dok_matrix – словник ключів на основі розрідженої матриці. Це ефективн структура для

побудови розріджених матриць. Розріджені матриці можуть бути використані в арифметичних операціях: вони підтримують додавання, віднімання, множення, ділення.

Бібліотека Matplotlib для створення двовимірних діаграм і графіків. З її допомогою можна побудувати будь-який графік. Однак для складної візуалізації потрібно більше коду, ніж в розвинених бібліотеках. Для перевірки роботи розробленого алгоритму кластерування, були згенеровані вибірки різних форм. Ці вибірки були обрані, бо головна мета – показати характеристики різних алгоритмів кластерування на наборах даних, які є «цікавими», але все ще в 2D. Останній набір даних є прикладом «нульової» ситуації для кластеризації: дані є однорідними, і немає хорошого кластеризації. Останній набір даних, який буде представлено, є прикладом «нульової» ситуації для кластерування: дані є однорідними. Також можна задавати їх розмір і рахувати швидкість обробки даних.

Також в роботі проводиться порівняння розробленого алгоритму нечіткого кластерування з існуючими методами. Для того, щоб не реалізовувати методи самостійно, використовується бібліотека scikit-learn – найпоширеніший вибір для вирішення завдань класичного машинного навчання. Вона надає широкий вибір алгоритмів навчання з учителем і без вчителя. В рамках котрої реалізовано алгоритм кластерування с-середніх, який використовується для порівняння.

Для порівняння FSOINN, всі протестовані вибірки були оброблені з використанням методу кластерування: самоорганізованої інкрементної нейронної мережі. Найбільш вагомим є останній експеримент, бо для нього обрано набір даних з реального життя. Тобто стає можливим оцінити чи можна використовувати запропоновану нейро-фаззі систему для кластерування великих наборів даних.

Процес обробки вхідних даних включає в себе такі етапи:

− користувач задає параметри системі;

− завантажує в систему дані;

− система задає початкові параметри мережі;

− система проводить нечітке кластерування;

− система підраховує функції належності;

− система розраховує час навчання нейронної мережі;

− дані відображаються в 2D (елементи фарбуються згідно з результатами кластерування).

Для першого експерименту взяті штучно згенеровані вибірки складної неопуклого форми, а саме кластер у кластері і дві спіралі. На рисунку 4.1 показано дві вибірки і час навчання нейронної мережі. Причому в першій вибірці один клас знаходиться всередині іншого, а в другій вони лінійно неподільні. Але запропонований нечіткий метод кластерного аналізу FSOINN точно відокремлює одне кільце від іншого. Вибірка складалася з 2000 прикладів і нейронна мережа навчилася за 30 секунд для першої вибірки і за 28 секунд для другої.

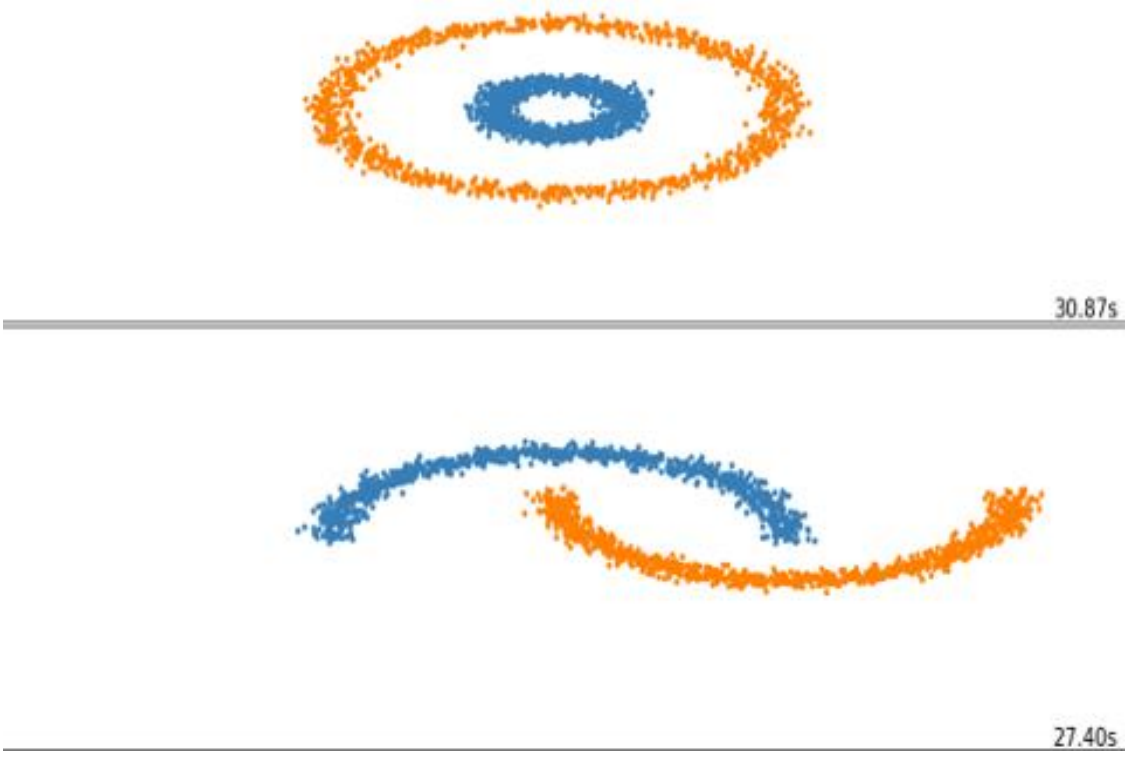


Рисунок 4.1 – Результати навчання методу FSOINN

Наступним кроком необхідно порівняти запропонований метод FSOINN з чітким кластерним методом SOINN. На рисунку 4.2 зображено результат роботи SOINN з такими же початковими параметрами системи. Цей метод працює дещо швидше, бо він не використовує функцію належності і саме тому обчислювальний процес дещо швидше. Процес навчання для першої і другої вибірки склав 22 секунди, що 1,4 рази швидше за FSOINN.

Однак, він гірше показав себе при навчанні з вибіркою з двома спіралями і визначив їх як один клас. Тому при виборі алгоритму дуже важливим є співвідношення часу навчання до точності, яку необхідно отримати на виході.

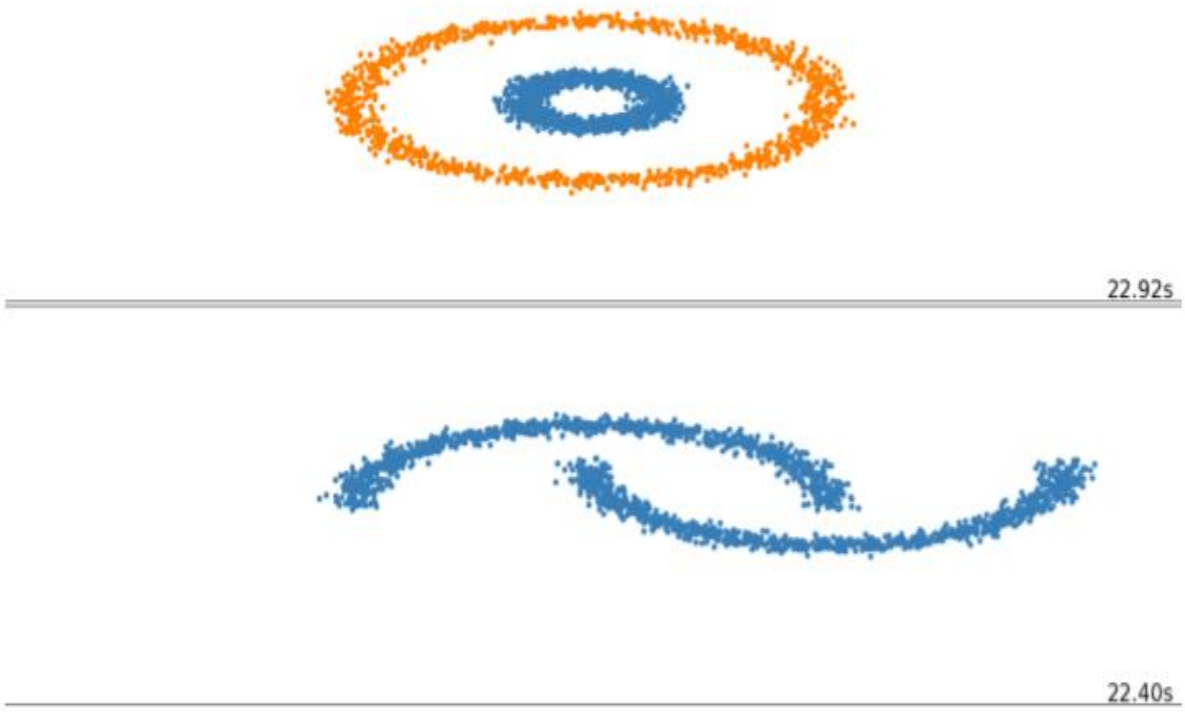


Рисунок 4.2 – Результати навчання методом SOINN

В таблиці 4.1 відображені функції належності для прикладів із набору даних з двома спіралями, які розташовані в спірній області і значення функції належності близькі.

Таблиця 4.1 – Відображення функції належності

Функція належності	
0 клас	1 клас
0.455	0.545
0.443	0.557
0.406	0.594
0.355	0.645

Ймовірності близькі і в деяких випадках відрізняються на соті. Тому для такої вибірки алгоритм FSOINN працює, краще ніж SOINN. Наявність функції належності для настроювання синоптичних ваг дозволяє розділити дві спіралі на кластери. Також суттєво впливає те, що функція належності корегує всі вузли.

В підрозділі 3.2 був згаданий алгоритм кластерування FCM – нечіткий с-середніх. Обчислювальний ядро алгоритму складають кроки обчислення центрів кластерів, відстаней між ними і точками даних і особливо перерахунку матриці ступенів належності точок даних. Тобто FCM включає в себе три основних етапи - обчислення центрів кластерів, обчислення відстаней між центрами кластерів і точками даних. Але необхідно заздалегідь знати кількість кластерів, що не завжди відомо заздалегідь і алгоритм дуже чутливий до вибору початкових центрів кластерів. Також запропонований метод кластерування відрізняється від FCM функцією належності. На відміну від функції Коші, ядро Гауса на нескінченності не близько нуля.

В атестаційній роботі проведено порівняння розробленого алгоритму FCM та FSOINN. На рисунку 4.3 зображено результат кластерування методом кластерування FCM. Отримано, що запропонований алгоритм краще навчається на складних фігурах, бо нечіткий с-середніх поділив класи лінійно. За параметром часу порівнювані методи кластерного аналізу відрізняються.

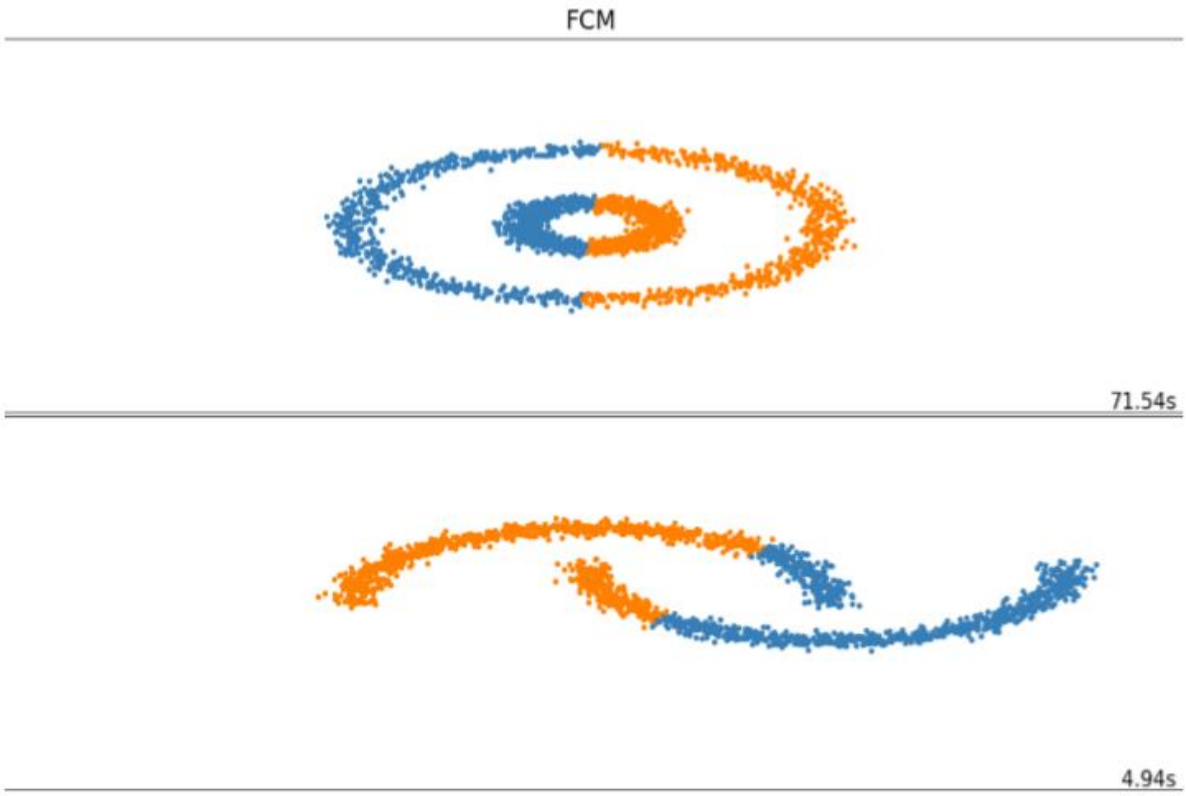


Рисунок 4.3 – Візуалізація кластерування методом FCM

Для другого експерименту була обрана вибірка з областями, в яких щільність розподілу даних мала, для того, щоб визначити як FSOINN обробляє проблему неоднозначності об'єктів, які віддалені від центрів всіх кластерів. Тобто закон розподілу, якому відповідає вибірка невідомий, тому виникають труднощі з відновленням теоретичної моделі розподілу ймовірностей. Така ситуація поширена в реальних задачах, пов'язаних з аналізом даних. Разом з тим, ситуації, коли закон розподілів відомий лише частково, наприклад, виходячи з будь-яких апріорних відомостей про природу даних теж цілком поширені. Тому перед тим як почати навчання нейронної мережі потрібно налаштувати системні параметри. У разі, якщо вони будуть спочатку задані неправильно, потрібно перенавчити нейронну мережу. А якщо вибірка даних велика, то навчання в сумі займе дуже багато часу. На рисунках 4.4, 4.5 і 4.6 показано результат роботи алгоритмів FSOINN, SOINN і FCM.

Рисунок 4.4 – Кластерування розрідженої вибірки FSOINN

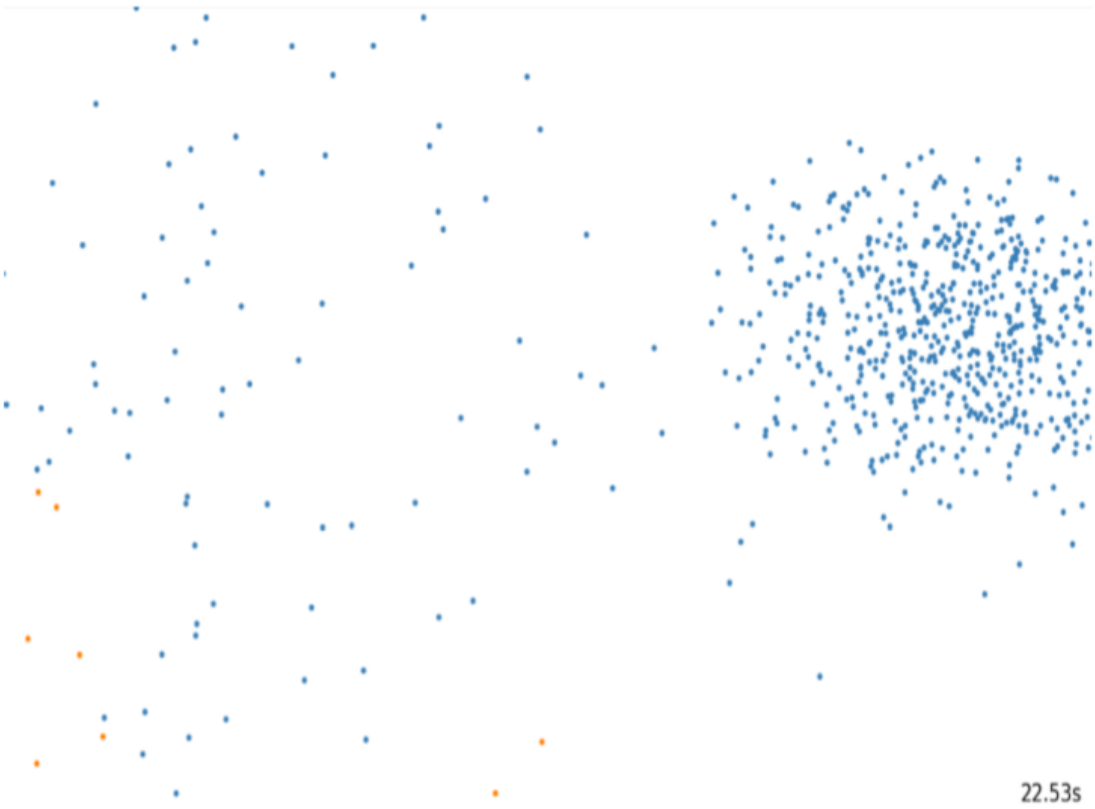


Рисунок 4.5 – Кластерування розрідженої вибірки SOINN

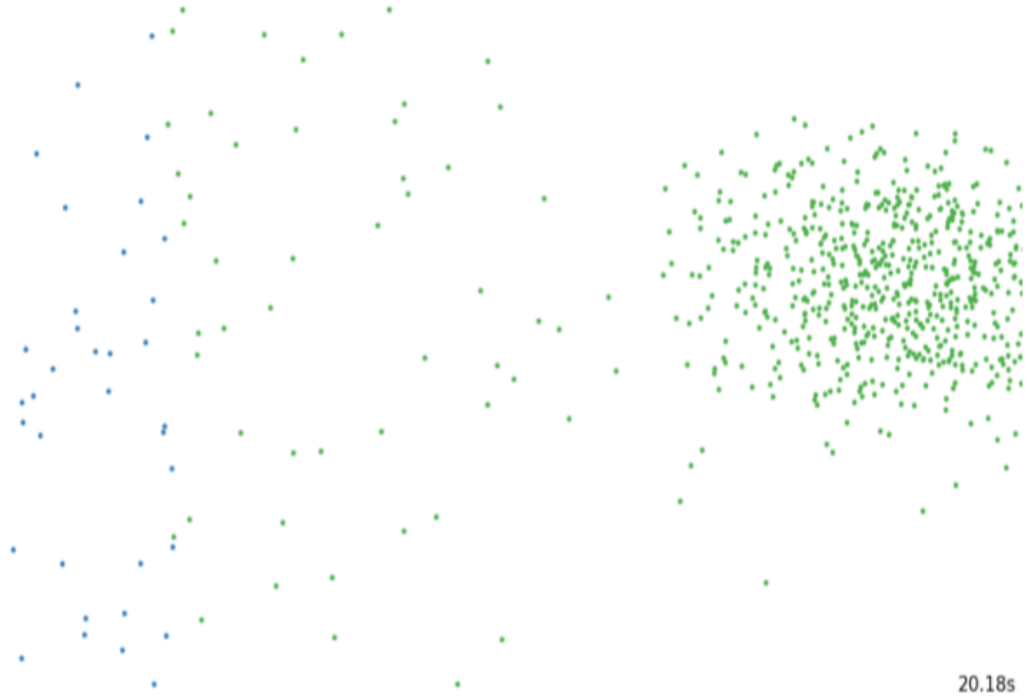


Рисунок 4.6 – Кластерування розрідженої вибірки FCM

В результаті проведеного експерименту методи FSOINN і SOINN розділили вибірку на два класи, а тому для порівняння цих методів з FCM, параметр кількості класів теж дорівнює двом. Нечітка самоорганізована інкрементна мапа віднесла до першого класу приклади, які знаходяться в області з великою щільністю розподілу. Для цього набору даних час навчання практично не відрізняється для всіх методів. А до другого належать об’єкти, які розташовані на великій відстані. Тобто, можна зробити висновок, що розроблений метод кластерування вирішує проблему неоднозначності даних.

Експеримент проведений на вибірці, в якій приклади згруповано та розподілені в рамках однієї області. Але є п'ять відсотків об'єктів, розташованих на невеликій відстані від основного згущення. Такі приклади називають «шумом».

На рисунках 4.7, 4.8 і 4.9 зображено результат навчання порівнюваних алгоритмів кластерування.

Рисунок 4.7 – Кластерування вибірки з шумом методом FSOINN

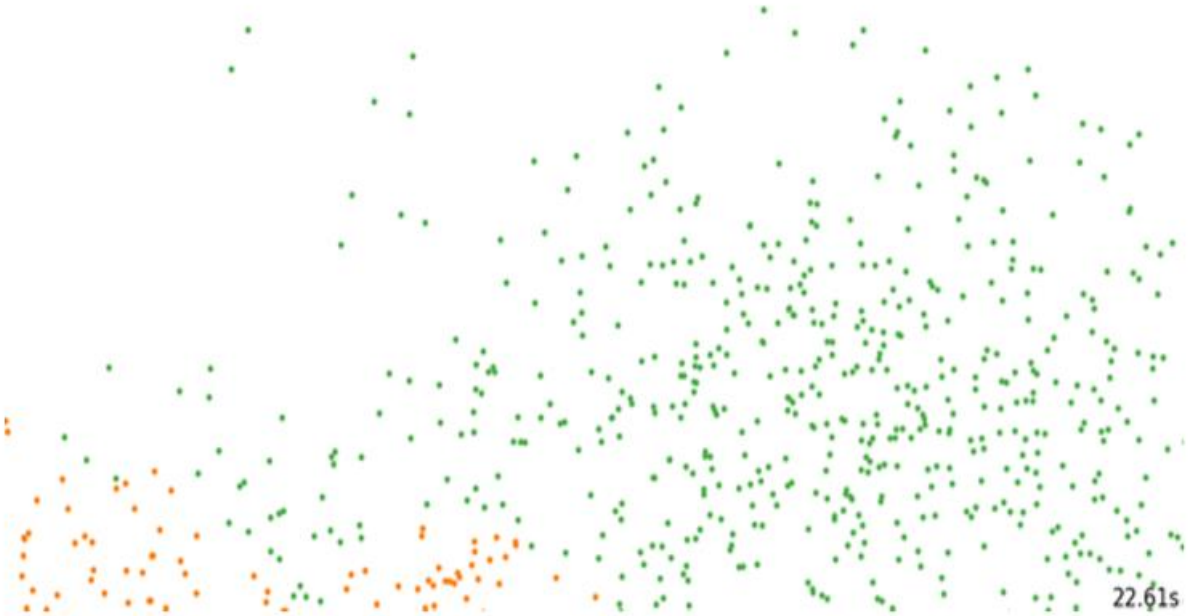


Рисунок 4.8 – Кластерування вибірки з шумом методом SOINN

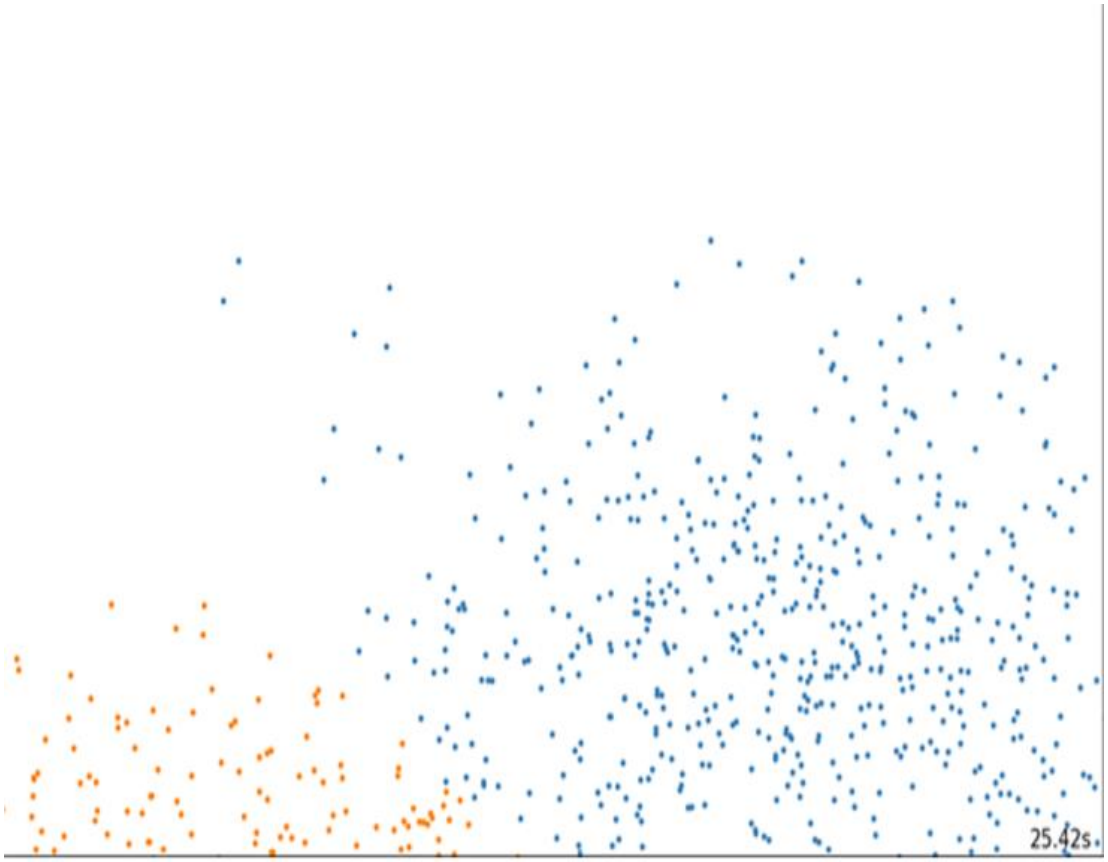


Рисунок 4.10 – Кластерування вибірки з шумом методом FCM

Експеримент показав, що нечітка самоорганізована інкрементна мапа розділяє вибірку на три класи і дані з шумом виділяються в окремий клас. Алгоритми SOINN і FCM не справляються з шумом і ці об'єкти тільки заважають адекватно проводити навчання.

SOINN розділяє вибірку на два класи, але вони не несуть ніякого сенсу. Тобто шум призводить до неефективної роботи цього алгоритму кластерного аналізу. Початкові параметри нейронної мережі були встановлені такі самі, як і в FSOINN.

Змінюючи параметр кількості кластерів в методі FCM, все одно визначаються тільки два кластери, тому можна зробити висновок, що FCM не розпізнає шум у прикладах. Реальні дані, які підлягають обробці, в багатьох випадках включають ненормовані приклади. Саме тому, можна стверджувати, що FSOINN має перевагу і вміє працювати з шумом.

Останній експеримент проводився на реальних даних, які були взяті зі змагання на платформі Kaggle. Важливо, щоб компанії кредитних карток мали змогу визнавати шахрайські операції з кредитними картками, щоб клієнти не стягували плату за товари, які вони не купували. У цьому експерименті основна ціль показати як запропонована нейро-фаззі система працює з даними з реального світу.

Набір даних містить транзакції, здійснені кредитними картками у вересні 2013 року власниками європейських карт. У цьому наборі даних представлені операції, що відбулися за два дні, де є 492 приклади шахрайства з 284 807 транзакцій. Набір даних сильно незбалансований, позитивний клас (шахрайство) становить 0,172% усіх транзакцій. На рисунку 4.10 показано розподіл даних по класам.

Набір даних містить лише числові вхідні змінні, які є результатом перетворення PCA. На жаль, через проблеми конфіденційності немає змоги надати оригінальні функції та додаткову довідкову інформацію про дані.

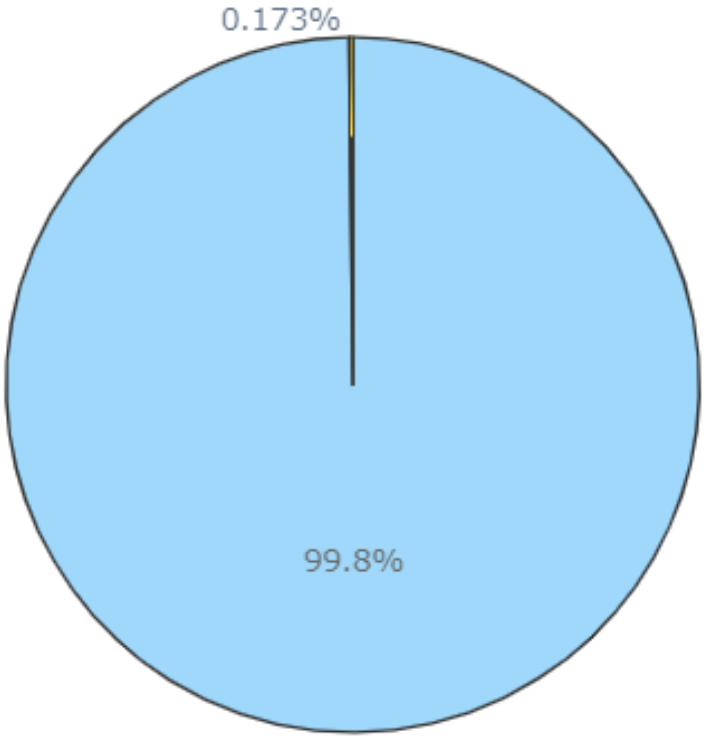


Рисунок 4.10 – Розподіл даних

Використовуючи цей набір даних може підрахувати точність кластерування, бо є розмічені класи. Але для навчання розмітка не використовується. Через те, шо набір даних незбалансований вирішено побудувати матрицю помилок. Наприклад, якщо в результаті експерименту буде тільки один кластер, то точність буде вище ніж 99%. Але це не є правильним, бо основна задача виділити шахрайські транзакції. Матриця помилок побудована за допомоги бібліотеки scikit-learn.

Є два класи і алгоритм, який пророкує належність кожного об'єкта одному з класів, тоді матриця помилок класифікації буде виглядати як в таблиці 4.2.

Таблиця 4.2 – Матриця помилок

	Розмітка – нормальна транзакція	Розмітка – спроба шахрайства
FSOINN – нормальна транзакція	істинно-позитивний	помилково-позитивний
FSOINN – спроба шахрайства	помилково-негативний	істинно-негативний

Результати кластерного аналізу для цієї вибірки наочніше будуть у вигляді 3D моделі. Тому інтерактивні графіки будуються в Jupyter Notebook за допомогою бібліотеки plotly.

Plotly – це online-платформа, де можна створювати та публікувати свої графіки. Однак, цю бібліотеку можна використовувати і просто в Jupyter Notebook. У бібліотеки є offline-mode режим, який і використовувався, бо він дозволяє використовувати її без реєстрації і публікації даних і графіків на сервер plotly.

На рисунку 4.11 кластеризації набору даних транзакцій з кредитних карток.

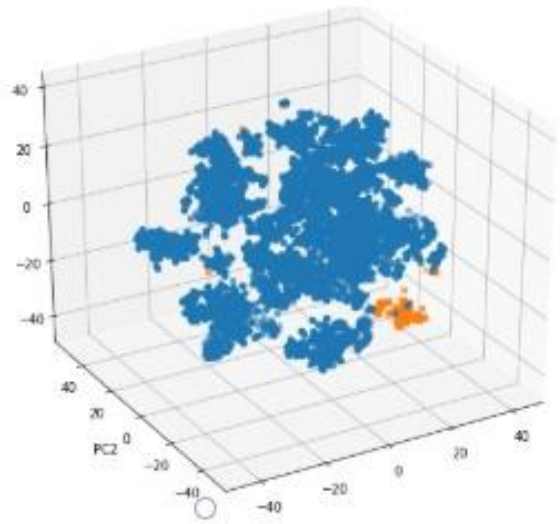


Рисунок 4.11 – Візуалізація кластерування транзакцій

На рисунку 4.12 показана матриця помилок.

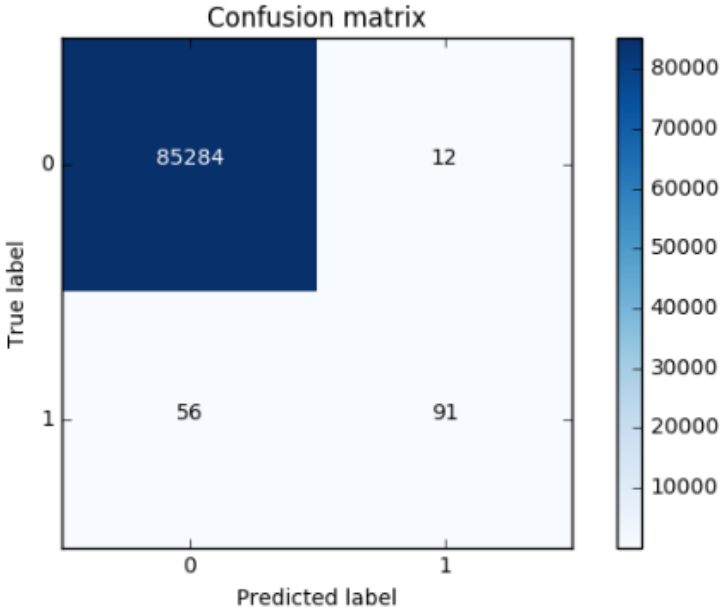


Рисунок 4.12 – Матриця помилок

Також як і зауважено вище, побудована матриця помилок. Вибірка була пропорційна зменшена в 4 рази для пришвидшення отримання результатів. Навчання нейронної мережі зайняло приблизно 3 години.

Отримана точність дорівнює 99% для класу нормальних транзакцій, а 0,11% це частка з шахрайських транзакцій класу алгоритм передбачив невірно. Тобто можна сказати, що запропонований алгоритм працює з високою точністю для пошуку аномальних транзакцій.

Отже, в даному розділі були доведені теоретичні дослідження на практиці. Наочно представлені результати кластерного аналізу трьох алгоритмів в двомірному просторі. Проведено три експерименти, які демонструють переваги і недоліки методів кластерування.

Запропонований алгоритм нечіткого кластерування порівняно з методом SOINN, на якому він заснований. Так само проведено порівняння з нечітким c-середніх алгоритмом кластерного аналізу. В результаті отримано, що алгоритми не поступається за часом роботи і ефективності. FSOINN працює з неопуклими і складними фігурами. Завдяки функції належності можливо визначити ймовірність відношення прикладу до того чи іншого кластеру. Особливо актуально це для прикладів, які знаходяться на границях між кластерами. Також він вирішує проблему неоднозначності даних, які знаходяться на великій відстані від основного розподілу прикладів. Мапа адекватно працює з шумом в даних. Однак, нечітку самоорганізовану інкрементну мапу складно налаштовувати початкові параметри, для кожної моделі вони відрізняються.

ВИСНОВКИ

В атестаційній роботі магістра було запропоновано метод кластерного аналізу нечітка інкрементна самоорганізована мапа в режимі послідовного навчання на основі самоорганізованої інкрементної нейронної мережі. Запропонований алгоритм дозволяє без апріорного знання кількості кластерів вирішувати задачу нечіткого послідовного кластерування і візуалізувати топологічну структуру даних в двомірному просторі для ефективного аналізу даних.

Розроблений алгоритм ефективно працює з неопуклими наборами даних і досить простий в реалізації, але для проведення навчання необхідно налаштовувати параметри. Він відноситься до особливого класу штучних нейронних мереж, а саме до самоорганізованих мап, які вирішують завдання кластерування і топологічної структури даних.

В роботі представлені результати, які відповідно до поставленої мети є рішенням актуального завдання нечіткого кластерування масивів даних в умовах довільної кількості кластерів. Теоретично була виведена, обґрунтована і запропонована функція належності, а також доведені її переваги.

Всі поставлені завдання були виконані і можна зробити такі висновки по проведеній роботі:

− проведений аналіз предметної області та існуючих методів кластеризації показав, що методів, які повсюдно використовуються в рамках напряму Data Mining, в основному призначені для роботи в пакетному режимі і вимагають заздалегідь встановлювати кількість класів. Що для реальних завдань є проблемою, так як найчастіше немає апріорної інформації про розподіл даних;

− в ході організації нечіткості метода кластерування теоретично виведена функція належності на основі методу FCM і ця функція використана для організації нечіткості запропонованого алгоритму;

− в рамках атестаційної роботи бакалавра була розроблена модель штучної нейронної мережі, яка дозволяє обробляти обсяги даних в послідовному режимі і обчислювати функцію належності при заздалегідь невідомої кількості кластерів;

− проведено імітаційне моделювання на даних для доказу теоретичних досліджень. В згенерованих вибірках приклади лінійно нероздільні и мають складну форму;

− проведено три експерименти для порівняння запропонованого методу кластерування з SOINN і FCM. Останній працює в пакетному режимі. Також оцінено швидкість алгоритмів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Shen F., Osamu H. An Algorithm for Incremental Unsupervised Learning and Topology Representation. *Computer Vision and Pattern Recognition, 2005*: IEEE Computer Society Conference, Tokyo, Japan, 20 July, 2005. IEEE Xplore Digital Library, 2005. Vol. 1. P. 651-656.

2. Shen F. An enhanced self-organizing incremental neural network for online unsupervised learning. *Neural Networks*. 2007. №20. P. 893–903.

3. Shen F., Osamu H. A fast nearest neighbor classifier based on self-organizing incremental neural network. *Neural Networks*. 2008. Vol. 21, No 10. P. 1537–1547.

4. Shen F. An incremental network for on-line unsupervised classification and topology learning. *Neural Networks*. 2006. Vol. 19. No 21. P. 90–106.

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

6. Kohonen T. Self-Organizing Maps. Berlin: SpringerVerlag, 1995. 362 p.

7. Bodyanskiy Ye., Deineko A., KutsenkoY., Zayika O. Data streams fast EM-fuzzy clustering based on Kohonen`s self-learning. *The 1th IEEE International Conference on Data Stream Mining & Processing (DSMP 2016):* proc. of int. conf. Lviv, August 23-27, 2016 p. Lviv, 2016. P. 309–313.

8. Hoeppner F., Klawonn F., Kruse R. Fuzzy-Clusteranalysen. Braunschweig:Vieweg, 1997. 280 p.

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

10. Бодянский Е. В., Самитова В.А. Нечёткая кластеризация данных в порядковой шкале на основе совместного использования функций принадлежности и правдоподобия. *Збірник наукових праць Харківського університету Повітряних сил.* 2010. No 3. С. 91-95.

11. Томашевский Ю. Б. Нечёткая кластеризация: Р СГТУ, 2009. 12 с.

12. Иванова Е.В. Самоорганизованная инкрементная нейронная сеть для кластеризации массивов. *22-й Международный молодежный форум «Радиоэлектроника и молодежь в 21 веке»*. Харьков, 25-27 апреля, 2018 г. Харьков: ХНУРЭ, 2018. Том 7. С. 2930.

13. Tsoukalas L. H., Uhrig R. E. Fuzzy and Neural Approaches in Engineering. N.Y.:John Wiley camp; Sons, Inc., 1997. 587 p.

14. Villmann, T., Schleif, F.-M., & Hammer, B. Supervised neural gas and relevance learning in learning vector quantization. *Proceedings of the Workshop on Self-Organizing Maps (WSOM)*. Japan, 2003.

15. King, B. Step-wise clustering procedures. *Journal of American Statistical Association*. 1967. No. 69, P. 86–101.

15. Kohonen, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*. 1982. No. 43, P. 59–69.

16. Likas, A., Vlassis, N., & Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern Recognition*. 2003. No. 36, P. 451–461.

17. Lim, C. P., Harrison, R. F. A incremental adaptive network for on-line supervised learning and probability estimation. *Neural Net-works*. 1997, No. 10, P. 925–939.

18. Carpenter, G. A., Grossberg, S. The art of adaptive pattern recognition by a self-organizing neural network. *IEEE Computer*. 1998, No 21, P. 77–88.

19. Fritzke, B. Growing cell structures – a self-organizing network for unsupervised and supervised learning. *Neural Networks*. 1994, No. 7, P. 1441–1460.

20. Круглов В. В., Борисов В. В. Искусственные нейронные сети. Теория и практика. М.: Горячая линия, Телеком, 2001. 382 с.

21. Горбань А. Н., Россиев Д. А. Нейронные сети на персональном компьютере. Новосибирск: Наука, 1996. 276 с.

22. Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches. Ed. by D. A. White, D. A. Sofge. N.Y.: Van Nostrand Reinhold, 1992. 568 p.

23. Розенблатт Ф. Обобщение восприятий по группам преобразований. В кн.: «Самоорганизующиеся системы». М.: Мир, 1964. 65-112 с.

24. Розенблатт Ф. Модель памяти на нейронных сетях. *Автоматика*. 1965, No 5, P. 4.