

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

РОЗРОБКА ТА ДОСЛІДЖЕННЯ АДАПТИВНОГО МЕТОДУ
ЙМОВІРНІСНОЇ НЕЧІТКОЇ КЛАСТЕРИЗАЦІЇ ДАНИХ
(тема)

Виконав:
студент 2 курсу, групи ІНФМ-18-1
Харченко В.В.
(прізвище, ініціали)

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

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

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доцент., Шафроненко А.Ю.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Путятін Є.П.
(прізвище, ініціали)

2019 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 20 ____ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУстудентові Харченко Владислав Віталіївни
(прізвище, ім'я, по батькові)1. Тема роботи «Розробка та дослідження адаптивного методу ймовірнісної нечіткої кластеризації даних»

затверджена наказом по університету від « 21 » жовтня 2019 року № 1506Ст

2. Термін подання студентом роботи до екзаменаційної комісії 12 грудня 2019 р.3. Вихідні дані до роботи карти Кохонена, адаптивний метод ймовірнісної нечіткої кластеризації даних, перелік використовуваних програмних засобів: програмна платформа MATLAB, UCI Iris repository, UCI Wine repository

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Огляд нечіткої логіки та систем нечіткого виводу _____

2. Карти Кохонена для нечіткої кластеризації даних _____

3. Нейро-фаззи методи для кластеризації викривлених даних _____

4. Адаптивний метод ймовірнісної нечіткої кластеризації даних _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

схема радіально-базисної нейронної мережі, топологія карт Кохонена, ID карта Кохонена, таблиця з результатами експериментальних досліджень порівняльного аналізу кластеризації даних вибірки Iris UCI repository, таблиця результатів експериментальних досліджень порівняльного аналізу кластеризації даних вибірки Wine UCI repository

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	21.10.2019	
2	Аналіз завдання, підбір літератури	05.11.19-10.11.19	
3	Аналіз літератури з досліджуваної проблеми	11.11.19-15.11.19	
4	Аналіз технічних засобів	16.11.19-18.11.19	
5	Розробка методу	19.11.19-22.11.19	
6	Програмна реалізація	22.11.19-24.11.19	
7	Оформлення пояснювальної записки	25.11.19-01.12.19	
8	Перевірка на плагіат	02.12.19	
9	Рецензування	03.12.19	
10	Підготовка презентації та доповіді	03.12.19	
11	Занесення роботи в електронний архів	04.12.19	
12	Попередній захист атестаційної роботи	05.12.19	

Дата видачі завдання 21 жовтня 2019 р.

Студент _____
(підпис)

Керівник роботи _____ доцент. Шафроненко А. Ю.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до атестаційної роботи: 62 с., 2 табл., 3 рис., 2 дод., 117 джерело.

НЕЧІТКА КЛАСТЕРИЗАЦІЯ, САМООРГАНІЗОВНА МЕРЕЖА КОХОНЕНА, ПРАВИЛА НАВЧАННЯ

Метою роботи є розробка ймовірнісного методу нечіткої on-line кластеризації даних що може бути вирішена на основі самоорганізованої нейро-фаззі мережі Кохонена.

Об'єктом дослідження є послідовність вибірки даних UCI репозиторія, а саме такі як Iris та Wine.

Використано методи нечіткої кластеризації даних та само організовані карти Кохонена. Проведено дослідження методів кластеризації даних, а також порівняльний аналіз роботи відомих методів кластеризації та запропонованого методу по критеріям якості кластеризації.

У результаті роботи розглянуто задачу ймовірнісної нечіткої on-line кластеризації даних і показано, що вона може бути вирішена на основі само організованої нейро-фаззі мережі Кохонена та адаптивного алгоритму самонавчання, який було запропоновано, що є гібридом правила «переможець отримує більше» і рекурентних алгоритмів нечіткої кластеризації.

FUZZY CLUSTERING, SELF-ORGANIZING KOHONEN NETWORK, TRAINING RULES

The purpose of the paper is to develop a probabilistic method of fuzzy on-line data clustering that can be solved on the basis of the Kohonen network self-organizing neuro-phase.

The object of the study is a sample sequence of UCI repository data, such as Iris and Wine.

Fuzzy data clustering methods and self-organized Kohonen maps are used. The methods of data clustering were investigated, as well as a comparative analysis of the work of known clustering methods and the proposed method on the criteria of clustering quality.

As a result, the problem of probabilistic fuzzy on-line data clustering is considered, and it is shown that it can be solved on the basis of Kohonen's self-organized neuro-phase network and the proposed adaptive self-learning algorithm, which is a hybrid of the winner-take-more rule and recurrent fuzzy algorithms.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ	7
1 Огляд проблеми та постановка задачі дослідження	8
1.2 Нечітка логіка і системи нечіткого виведення	11
1.3 Нейромережеві і нейро-фаззі моделі для класифікації даних і їх навчання	12
1.4 Самоорганізовані карти Кохонена	14
1.5 Нейро-фаззі системи	21
1.6 Постановка задачі дослідження	25
2 Адаптивні нейро-фаззі методи для кластеризації викривлених даних	27
2.1 Формальна постановка задачі кластеризації	27
2.2.1 Евклідова відстань	29
2.2.2 Відстань міських кварталів (Манхеттенська відстань)	29
2.2.3 Відстань Чебишева	30
2.2.4 Часткова відстань	30
2.2.5 Методи кластеризації даних	31
2.2.6 Адаптивний метод ймовірнісної нечіткої кластеризації даних	31
3 Програмна реалізація	35
3.1 MATLAB	35
3.2 DATA BASE (ВИБОРКА)	39
3.1.1 Iris	39
3.1.2 WINE	39
3.2 Експериментальні дослідження	40
Висновки	Ошибка! Закладка не определена.
Перелік джерел посилання	44
Додаток А	55

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

НМ – нейронні мережі

ШНМ - штучні нейронні мережі

РБНМ – радіально-базисні нейронні мережі

НРБНМ – нормовані радіально-базисні нейронні мережі

УРНС – узагальнені регресійні нейронні мережі

ННМ – нейронні нечіткі мережі

РБФ – радіально-базисні функції

NEFCLASS – Neuro Fuzzy CLASSifier

ВСТУП

Для вирішення складних завдань, що часто не мають аналітичного вирішення, на сьогоднішній день нерідко і вдало використовують технології обчислювального інтелекту. Ці технології і, особливо, штучні нейронні мережі (ШНМ) стають все поширенішими для вирішення різних завдань обробки сигналів, оптимізації, оптимального і адаптивного управління, розпізнавання образів, ідентифікації, прогнозування в реальному часі і т.д. Створено реальні системи обробки зображень та комп'ютерного зору, управління аерокосмічними об'єктами, діагностики технічної та медичної, у фінансах та економіці, у військовій справі, управління дорожнім рухом, в енергетичній промисловості, криміналістиці, аналізу сигналів різноманітної природи походження та інше, причому цей список стає довшим кожного дня.

Сьогодні є достатня кількість інформації про підприємницьку діяльність підприємств, лікарень, фірм, яка показує діяльність цих закладів.

В наслідок розгляду інформації стає можливим побачити закономірності при умовах того, що зроблена таблиця показує фактичні дані, що демонструють причинно-наслідкові залежності.

Ця інформація збирається протягом дового часу (темпи інфляції, рівень прибутку населення, витрати народу, ціну послуг житлово-комунального господарства, стан промислового і сільськогосподарського виробництва, своєчасність виплати заробітної плати та пенсій, тощо). Зрозуміло, такі дані складно обробити вручну через великий обсяг інформації і складних нелінійних причинно-наслідкових зв'язків. Тому і виникла потреба у створенні новіших методів аналізу, кластеризації і прогнозування, до яких відносять машинні методи виявлення послідовних зв'язків.

1 ОГЛЯД ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Однією в головних властивостей біологічного мозку є його здібність до навчання, і через те, що штучна нейронна мережа моделює роботу мозку «навчання» є ключовим поняттям в теорії штучних нейронних мереж. Процес навчання має сталий характер. З плином часу мережа робить краще свої характеристики, поступово приходячи до найбільш ефективного рішення поставленої задачі.

Нейронні мережі використовують для вирішення складних задач, які потребують аналітичних обчислень схожих тим, що робить справний мозок. Найчасішими вживання нейронних мереж:

1) класифікація – розподіл даних по параметрам. Наприклад, на вхід подається список людей і потрібно прийняти рішення, кому з них одобрити кредит, а кому ні. Цю роботу може виконати нейронна мережа, обробляючи таку інформацію як: вік, платоспроможність, кредитна історія і тд,

2) передбачення – можливість передбачити наступний крок. Наприклад, зростання або падіння акцій, ґрунтуючись на ситуацію фондового ринку,

3) розпізнавання – на даний момент, саме популярне використання нейронних мереж. Застосовується в Google, коли ви шукаєте фотографію або в камерах телефонів, коли воно визначає положення вашого обличчя і виділяє його і багато іншого.

1.1 Тип і характер навчання

Тип і характер навчання визначаються передусім об'ємом апріорної і поточної інформації про середовище, в якій знаходиться мережа, а також цільовою функцією (критерієм якості), що характеризує ступінь доречності

нейронної мережі до розв'язуваної нею задачі. Інформація про зовнішнє середовище задана, як правило, у виді навчальної вибірки образів або прикладів.

На даний момент існує кілька головних парадигм навчання штучних нейронних мереж таких, як навчання «з учителем», «без учителя», або «самонавчання», а так само навчання «з підкріпленням».

Найбільш поширеною і очевидною до тепер є парадигма навчання «з учителем». У даній парадигмі «вчителю» відома інформація про зовнішнє середовище. Вона подана у вигляді деякої послідовності даних або в пакетах вхідних векторів x , а також реакції на ці сигнали, показана у вигляді навчального сигналу d .

Альтернативою цій парадигмі є навчання «без вчителя», або самонавчання, коли правильна реакція на сигнали зовнішнього середовища невідома. В даний час існує декілька основних парадигм навчання штучних нейронних мереж таких, як навчання «з учителем», «без вчителя», або «самонавчання», а так само навчання «з підкріпленням». Найбільш поширеною і очевидною до тепер є парадигма навчання «з учителем». У даній парадигмі «вчителю» відома інформація про зовнішнє середовище. Вона задана у вигляді деякої послідовності даних або в пакетах вхідних векторів x , а також реакція на ці сигнали, показана у вигляді навчального сигналу d . Альтернативою цій парадигмі є навчання «без вчителя», або самонавчання, коли правильна реакція на сигнали зовнішнього середовища невідома.

Можливість самонавчання являється необхідним атрибутом нейронних мереж (НМ) для аналізу і знаходження скритих закономірностей в багатовимірній інформації, що властиві більшості завдань інтелектуального аналізу даних, зокрема, задачам класифікації та кластеризації.

Своєрідним компромісом між парадигмами «самонавчання» і «навчання з учителем» є навчання з підкріпленням, при якому доступна лише непряма інформація про правильні реакції на вхідний сигнал x .

Процес навчання з підкріпленням розбділяється на два етапи:

1) Навчання тому, як вихідний сигнал мережі впливає на спостережувані середовища, що змінюються, тобто відновлення зображення \tilde{F} .

2) Навчання мережі на базі мінімізації прийнятого критерію.

У великій кількості додатків, пов'язаних з управлінням технологічними виробничими процесами, функціонально-наближена архітектура мережі достатньо проста, і вона успішно справляється з моделюванням виробничих процесів. Нейронні мережі повинні бути привчені маючи набори даних, отриманих за допомогою найкращого з можливих варіантів параметричного тестування. Щоб зімітувати управління процесом відповідним чином, користувач обирає змінну процесу і робить набір даних для навчання. Цей набір повинен закрити весь можливий діапазон навчання і відповідати його функціональному призначенню. Набір даних містить дані, пов'язані з входами, і корелює їх до цільового призначення. Входи, які використовуються у навчанні, демонструють простір управління нейронної мережевої моделі.

Всі записи набору даних виконуються з тимчасовими мітками, що визначають їх цільовий відлік до часу захоплення і значення параметра входу. Набори даних мають бути прив'язані до конфігурації обладнання, яка мала місце в момент покупки системи. Для покращення надійності моделі, ліквідації перешкод і зниження вартості критично велике значення має правильний вибір моделі входів. При цьому калібрування повинне бути завершено для всіх входів ще до поступання даних. Це потрібно для того, щоб встановити їх базовий опорний рівень.

Після того як набір даних був отриманий, він повинен бути розібраний на три підмножини: навчання, випробування і валідацію. Навчання нейронної мережі передбачає зворотне поширення помилки між виходом моделі і цільовим значенням. Навчання є ітеративним процесом зворотнього поширення, який вимагає стратегії, що робить упор на надійність і

підтвердження достовірності. Значення (ваги) і їх зміщення для кожного нейрона надаються в процесі навчання. Звіт про валідацію має бути розглянутий ще до імплементації системи на платформі управління. Цей звіт повинен показати збіг цільової функції управління у фазі з моделлю при мінімальній похибці.

Більшість підходів навчання нейронної мережі працюють на настільних комп'ютерах, а основні платформи управління вже мають весь необхідний набір команд для налаштування моделі нейронної мережі. Там, де це можливо, кодування нейронів на платформі управління повинно бути виконано в підпрограмі, що дозволяє повторне застосування коду на майбутніх моделях. У працюючих моделях всі входи повинні бути перевірені на відповідні якісні показники і допустимий простір контролю. Якщо входи знаходяться не в колі допустимого простору контролю, то повинна передбачатись стратегія альтернативного управління.

НМІ повинен показувати вихід моделі, входи, сигнали тривоги, технологічні уставки процесу, давати можливість вибору режиму (автоматичний / ручний) і стратегії управління (звичайна / альтернативна). Тенденції і логіка з обслуговування в порядку поступання є достатньо важливим рішенням для оперативного інформування та діагностики. Для введення необхідних констант, сигналізації і обмежень краще використовувати окремий екран налаштування.

1.2 Нечітка логіка і системи нечіткого виведення

Нечітка логіка вперше була запропонована Л. Заде для подання нечітких і неточних знань. Вона надає ефективний спосіб опису поведінки дуже складних, погано визначених або погано формалізованих математичних систем. Маніпуляції з нечіткими змінними утворюють формальну систему, яку називають нечіткою логікою. Нечітка логіка має у своєму складі:

фазифікації - переклад чіткого вхідного значення в нечітке, нечіткий висновок вилучення даних з бази знань, а також дефазифікація - переводить нечіткий результат в чіткий.

Головною перевагою нечітких множин можна вважати можливість лінгвістичної інтерпретації. А саме, використовуючи нечіткі множини, вдається легко визначати різні лінгвістичні терміни, які можуть мати нечіткі значення.

1.3 Нейромережеві і нейро-фаззі моделі для класифікації даних і їх навчання

На сьогоднішній день ШНМ користуються попитом для рішення задач ідентифікації, прогнозування та управління нелінійними об'єктами.

Більшість реальних завдань описуються тим, що дані надходять на обробку послідовно. В даному випадку альтернативою може бути радіально-базисна нейронна мережа (РБНМ), тому що класичний багат шаровий персептрон не підходить. Ці мережі ще є універсальними апроксиматорами. Для їх навчання в реальному часі може бути використано рекурентний метод найменших квадратів або його модифікації, які по суті є алгоритмами оптимізації другого порядку, які забезпечують квадратичну збіжність до оптимального рішення. Практичне використання РБНМ обмежується, так званим, прокляттям розмірності, а також появою пропусків в просторі радіально-базисних функцій, що призводять до появи областей, де всі нейрони мережі не являються активними. Уникнути схожих ситуацій можна, скориставшись одиничною розбивкою простору, що реалізовується за допомогою нормалізованих радіально-базисних нейронних мереж (НРБНМ), в яких вихідний сигнал нормується на суму виходів всіх нейронів. Ці мережі навчаються за допомогою рекурентних градієнтних алгоритмів, загальним недоліком яких є не велика швидкість збіжності і можливість попадання в

локальні мінімуми прийнятого критерію навчання. На сьогоднішній день ШНМ користуються попитом для рішення задач ідентифікації, прогнозування та управління нелінійними об'єктами. Більшість реальних завдань описуються тим, що дані надходять на обробку послідовно. В даному випадку альтернативою може бути радіально-базисна нейронна мережа (РБНМ), що показана на рис. 1.1, тому що класичний багатoshаровий перцептрон не підходить. Ці мережі ще є універсальними апроксиматорами. Для їх навчання в реальному часі може бути використано рекурентний метод найменших квадратів або його модифікації, які по суті є алгоритмами оптимізації другого порядку, які забезпечують квадратичну збіжність до оптимального рішення. Практичне використання РБНМ обмежується, так званим, прокляттям розмірності, а також появою пропусків в просторі радіально-базисних функцій, що призводять до появи областей, де всі нейрони мережі не являються активними. Уникнути схожих ситуацій можна, скориставшись одиничною розбивкою простору, що реалізовується за допомогою нормалізованих радіально-базисних нейронних мереж (НРБНМ), в яких вихідний сигнал нормується на суму виходів всіх нейронів. Ці мережі навчаються за допомогою рекурентних градієнтних алгоритмів, загальним недоліком яких є не велика швидкість збіжності і можливість попадання в локальні мінімуми прийнятого критерію навчання.

Вхідні шари такої мережі – це сенсори, які пов'язують ІНМ з навколишнім середовищем. Єдиний прихований шар, створений нейронами, робить нелінійне перетворення вхідного простору R^n розкритого простору R^h , як правило, більш високої розмірності ($h \gg n$).

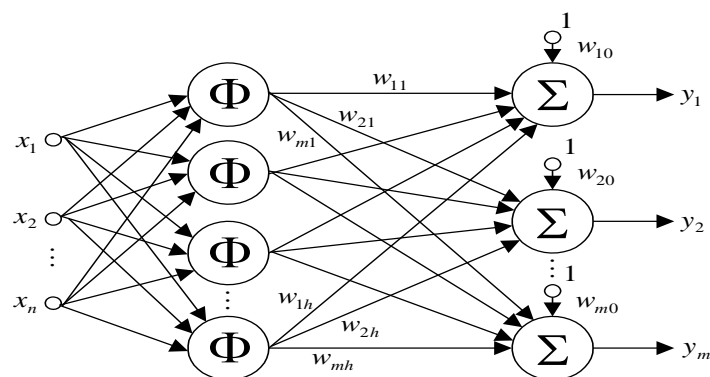


Рисунок 1.1 – Радіально-базисна нейронна мережа

I, нарешті, вихідний шар, утворений адаптивними лінійними асоціатором, формує реакцію мережі $y = (y_1, y_2, \dots, y_m)^T$ на вхідний сигнал $x = (x_1, x_2, \dots, x_n)^T$.

Альтернативами мережам, заснованим на оптимізації, являються мережі засновані на пам'яті, найяскравішим представником яких являється узагальнена регресійна нейронна мережа (УРНС), запропонована Д. Шпехтом [38]. Ці мережі можуть бути віднесені до миттєво навчених моделей [39], налаштованим завдяки одному проходу алгоритму навчання. Збігаючись з архітектурою з нормалізованими радіально-базисними нейронними мережами (НРБНС), УРНМ вчиться на багато швидше, встановлюючи центри РБФ в точках з координатами, визначеними вхідними сигналами об'єкта за принципом «нейрони в точках даних» [40], і з висотою РБФ, що збігається з відповідними значеннями вихідного сигналу об'єкта. Саме висока швидкість навчання ОРНС забезпечила їх ефективне використання в задачах реального часу [7, 10].

1.4 Самоорганізовані карти Кохонена

В основі карти Кохонена лежить нейробиологічна модель Д. Віллшоу - К. фон дер Мальсбурга. Здатність до самоорганізації мережі пов'язані з тим, що налаштування синаптичних ваг відбувається без наявності зовнішнього навчального сигналу, тобто в режимі самонавчання, при цьому кожен наступний сигнал викликає адаптацію тих чи інших параметрів ШНМ. Даний процес може бути безперервним, забезпечуючи можливість вирішення завдань послідовно.

Самоорганізована карта має просту архітектуру з прямою передачею інформації, а також крім нульового рецепторного шару містить єдиний шар

нейронів, який іноді називають шаром Кохонена [41]. Кожен нейрон шару Кохонена пов'язаний з кожним рецептором нульового шару прямими зв'язками і з усіма іншими нейронами поперечними всерединішаровими (латеральний) зв'язками. Саме латеральні зв'язки забезпечують збудження одних нейронів і гальмування інших.

Завдяки такій організації мережі, кожен нейрон отримує всю інформацію про аналізований образ і генерує на своєму виході відповідну реакцію після чого в шарі Кохонена виникає режим конкуренції, в результаті якого визначається єдиний нейрон-переможець з максимальним вихідним сигналом. Цей сигнал забезпечує збудження найближчих «сусідів» переможця і придушення реакції вузлів, що далеко знаходяться.

Таким чином, в процесі конкурентного самонавчання формуються групи нейронів, кожен з яких максимальним відгуком реагує на образи з відповідних підобластей вхідного простору, що дозволяє карті Кохонена успішно справлятися з компресією великих обсягів інформації. Самоорганізовані карти можуть мати різну топологію, як це показано на рис. 1.2.

Найпростіша карта Кохонена, наведена на рис. 1.2, має 1D топологію, n рецепторів і $m (m < n)$ нейронів в шарі Кохонена, кожен з яких характеризується власним вектором синаптичних ваг $w_q, q = 1, 2, \dots, m$.

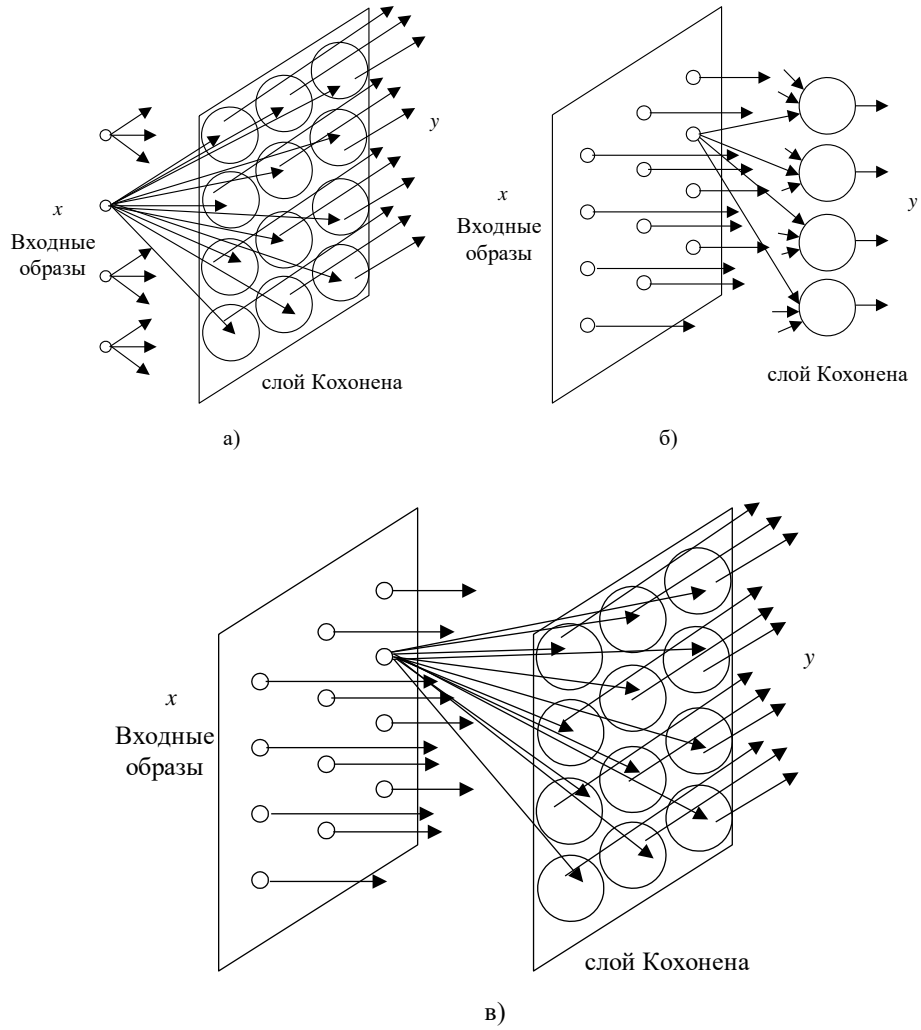


Рисунок 1.2 – Топологія карт Кохонена

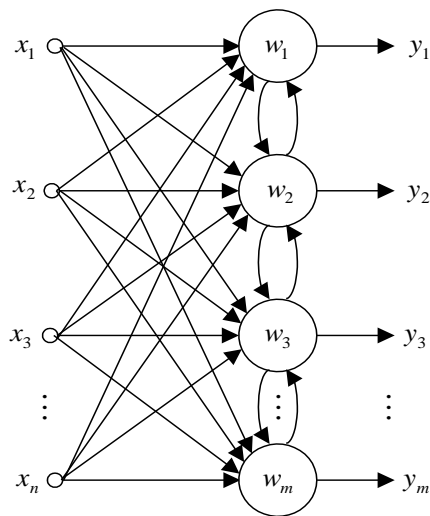


Рисунок 1.3 – 1D-карта Кохонена

В основі алгоритму самоорганізації карти Кохонена було покладено принципи конкурентного самонавчання. Як і будь-яка інша процедура навчання, робота алгоритму починається з ініціалізації синаптичних ваг мережі, які зазвичай обираються за допомогою застосування генератора випадкових чисел, при цьому бажано, щоб для кожного з нейронів виконувалася умова $\|w_q(0)\| = 1$.

Процедура самоорганізації реалізується в три основних етапи: конкуренції, кооперації і синаптичної адаптації. Самоорганізація починається з аналізу образу, що надходить з рецепторного шару на всі нейрони прошарку Кохонена. Для кожного з нейронів обчислюється відстань

$$D(\tilde{x}_k, w_q(k)) = \|\tilde{x}_k - w_q(k)\|, \quad (1.1)$$

причому, входи слід попередньо пронумерувати так, щоб $\|\tilde{x}_k\| = 1$. Як відстань (1.1) використовується евклідова метрика, а мірою близькості між векторами \tilde{x}_k і $w_q(k)$ може служити скалярний добуток

$$D(\tilde{x}_k, w_q(k)) = \tilde{x}_k^T w_q(k) = \cos(\tilde{x}_k, w_q(k)) = \cos \theta_q. \quad (1.2)$$

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

$$D(\tilde{x}_k, w^*(k)) = \min_q D(\tilde{x}_k, w_q(k)), \quad (1.3)$$

після чого в найпростішому випадку, «перестрибуючи» через етап кооперації, можна підлаштувати синаптичні ваги мережі

$$w_q(k+1) = \begin{cases} w_q(k) + \eta(k)(\tilde{x}_k - w_q(k)), & \text{якщо } q \text{ – тій нейрон виграв,} \\ w_q(k) & \text{в іншому випадку.} \end{cases} \quad (1.4)$$

Процедура (1.4) реалізує принцип «переможець отримує все», при цьому вектор синаптичних ваг нейрона-переможця «підтягується» до відповідного вектору на відстань, що дорівнює кроку пошуку.

Величина кроку має монотонно зменшуватися в процесі навчання, наприклад за правилом

$$\eta(k) = r^{-1}(k), r(k) = \alpha r(k-1) + \|\tilde{x}_k\|^2 = \alpha r(k-1) + 1, 0 < \alpha \leq 1, \quad (1.5)$$

При цьому при $\alpha = 1$ параметр кроку $\eta(k) = \frac{1}{k}$, тобто задовольняє умовам Дворецького. Варіюючи фактором забування α , нескладно забезпечити достатньо широкий інтервал зміни кроку пошуку

$$\frac{1}{k} \leq \eta(k) \leq 1. \quad (1.6)$$

Однією з особливостей карти Кохонена є наявність етапу кооперації в процесі самоорганізації, коли нейрон-переможець визначає так звану локальну область топологічного сусідства, в якій псується не тільки він сам, але і його найближче оточення, при цьому ближчі до переможця нейрони збуджуються сильніше, ніж віддалені. Ця топологічна область визначається функцією сусідства $\phi(q, l)$, яка залежить від відстані $D(w_q(k), w_l(k))$ між переможцем $w_q^*(k)$ і будь-яким з нейронів шару Кохонена $l = 1, 2, \dots, m$ і деякого параметра, який задає її «ширину».

Використання функцій сусідства призводить до модифікованого правила навчання Кохонена

$$w_l(k+1) = w_l(k) + \eta(k)\phi(q, l, k)(\tilde{x}_k - w_l(k)), l = 1, 2, \dots, m, \quad (1.7)$$

реалізує принцип «Переможець отримує більше» (Winner takes most) замість традиційного «Переможець отримує все».

Аналіз збіжності процесів конкурентного самонавчання, показав, що в процесі налаштування синаптичних ваг, повинен ставати менше не тільки крок пошуку, але і параметр ширини функції сусідства, яка таким чином стає залежною від моменту у часі.

Для гауссовської функції

$$\phi(q, l) = \exp\left(-\frac{\|w_q^*(k) - w_l(k)\|^2}{\sigma^2}\right). \quad (1.8)$$

Г. Рітгер і К. Шультен запропонували для налаштування параметра ширини застосовувати процедуру

$$\sigma(k) = \sigma(0) \exp\left(-\frac{k}{\beta}\right), \quad (1.9)$$

де $\beta > 0$ - скалярний параметр, що визначає швидкість зменшення сили впливу нейрона переможця на своє оточення. Звичайно, що при цьому змінюється і форма області топологічного сусідства, що набуває вид

$$\phi(q, l, k) = \exp\left(-\frac{\|w_q^*(k) - w_l(k)\|^2}{\sigma^2(k)}\right). \quad (1.10)$$

Треба звернути увагу на те, що експоненціальне убування параметра ширини може бути забезпечено і за допомогою більш простого, ніж (1.9) вираження

$$\sigma(k) = \beta\sigma(k-1), 0 < \beta < 1. \quad (1.11)$$

Для навчання самоорганізованої карти пропонується взагалі не обирати переможця як такого, а в якості найбільш універсальної функції сусідства використовувати вихідний сигнал кожного нейрона $y_l(k)$ так, що

$$\begin{aligned} w_l(k+1) &= w_l(k) + \eta(k)\phi(q, l, k)(\tilde{x}_k - w_l(k)) = \\ &= w_l(k) + \eta(k)y_l(k)(\tilde{x}_k - w_l(k)), l = 1, 2, \dots, m. \end{aligned} \quad (1.12)$$

Якщо в якості нейронів шару Кохонена використовується лінійний асоціатор, то (1.12) можна переписати у формі

$$w_l(k+1) = w_l(k) + \eta(k)w_l^T(k)\tilde{x}_k(\tilde{x}_k - w_l(k)), \quad (1.13)$$

а якщо крім того $\|w_l(k)\| = \|\tilde{x}_k\| = 1$, то і в більш простому вигляді

$$\begin{aligned} w_l(k+1) &= w_l(k) + \eta(k)\cos(\tilde{x}_k, w_l(k))(\tilde{x}_k - w_l(k)) = \\ &= w_l(k) + \eta(k)\cos\theta_l(k)(\tilde{x}_k - w_l(k)). \end{aligned} \quad (1.21)$$

Увесь процес самоорганізації має дві тимчасові фази: початкова фаза упорядкування, в якій відбувається топологічне розбиття вхідного простору, і подальша фаза збіжності, в якій здійснюється точне налаштування синаптичних ваг. Після закінчення цього процесу нейронна мережа може вирішувати поставлені задачі без уточнення ваг. У разі появи вхідного образу, який не може бути віднесеним до жодного з сформованих кластерів, картою повинен бути утворений додатковий нейрон в прошарку Кохонена, що несе інформацію про цей образ. Ці ШНМ найбільш ефективні для роботи в реальному часі, коли налаштування синаптичних ваг і обробка вхідних сигналів виконуються паралельно

1.5 Нейро-фазі системи

Завдання комплексного забезпечення адекватного уявлення взаємодіючих динамічних нечітких процесів, а також оптимізації ресурсів і вибору альтернатив розвитку нечітких процесів на безлічі обмежень є важливими і на даний момент не мають рішень, які б знайшли ефективне використання більшості практичних реалізацій і часто носять емпіричний, вузькоспеціалізований характер.

Підходи на основі штучних нейронних мережах (ШНМ), нейронних нечітких мережах (ННМ) є універсальним засобом моделювання складних процесів великої розмірності, але в меншій мірі ефективні при моделюванні процесів і процедур малої і середньої розмірності. Вони також потребують додаткових ресурсів на машинне навчання моделям.

Динамічні процеси в реальних системах характеризуються складними багатофункціональними залежностями і істотною нелінійністю. Моделювання таких процесів в задачах прогнозування, діагностики, ідентифікації, класифікації на основі існуючих класичних підходів спричиняє певні труднощі. Теоретико-системний аналіз на основі подання процесів математичними рівняннями дозволяє створювати досить точні моделі, але досволі трудомісткий і складний в реальних розробках. Експериментальний системний аналіз або ідентифікація базуються на моделях, параметри яких ґрунтуються на даних вимірювань. Перевагами таких моделей є більш короткі терміни розробки, але їх якість істотно залежить від коректного вибору структури та інтерпретації результатів вимірювань.

Особливістю ННМ є те, що вони відносяться до не параметричних моделей, а це викликає складності у визначенні відповідності (інтерпретації) її параметрів в термінах реальних процесів. В задачах ідентифікації на основі апроксимуючих властивостей ШНМ найбільш ефективними представляються моделі на основі багатошарового персептрона і мереж з радіально-базисними функціями (РБФ). Архітектури моделей на основі

мереж Кохонена або Хопфілда і деяких інших загалом використовуються в задачах угруповання і класифікації.

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

Нейронна мережа може вчитися на вхідних і вихідних даних для визначення поведінки системи. Ці знання можуть бути використані для створення нечітких правил і функцій приналежності, що істотно зменшує час, необхідний на розробку. Таке об'єднання також допомагає вирішити проблему неінтерпретованості результатів, отриманих за допомогою нейронних мереж. Подання ваг нейронної мережі за допомогою нечітких правил забезпечує розуміння роботи нейронної мережі, що, у свою чергу, допомагає створювати більш ефективні програми.

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

Дані ШНМ використовують нелінійні функції приналежності. Перевагою таких функцій є рівномірний розподіл знань між базою правил і базою функцій приналежності, що демонструється в скороченні обсягу бази правил. Важливим є той факт, що властивості узагальнення і навченості нейро-фаззі систем дозволяють генерувати правила і функції приналежності, що забезпечують більш достовірне і точне рішення в порівнянні з альтернативними методами. За допомогою коректної комбінації нечіткої логіки і нейронних мереж представляється можливим повне відображення знань нейронної мережі нечіткою логікою, у випадку, якщо алгоритми нечіткої логіки повністю засновані на архітектурі нейронної мережі. Це дозволяє генерувати нечіткі логічні рішення з заздалегідь заданою точністю виходу.

Алгоритми, що дозволяють на основі даних спостережень будувати в кінцевому підсумку нечітку систему, на початкових етапах використовують або нечітку кластеризацію, яке покриття простору багатовимірними гіперкубами, або оптимізацію багатовимірних ґратчастих структур, що виникають при розбитті координатних осей n -мірного простору на нечіткі підмножини. Основними проблемами в цьому випадку є забезпечення єдності подання одного і того ж лінгвістичного значення і інтерпретується індукованого набору правил приведення одержаних дискретних функцій приналежності до однієї зі стандартних форм, а також уникнути серйозних інформаційних втрат після проектування одержаних в результаті кластеризації областей.

Один з напрямків розвитку систем нечіткого виводу було запропоновано в [42], де розглядаються адаптивні мережеві системи нечіткого виведення (Adaptive-Network-Based Fuzzy Inference Systems, ANFIS), побудовані на базі архітектури Такагі-Сугено, а також гібридний метод настройки параметрів антецедента (тобто параметрів функції належності). В [43], доведена еквівалентність ANFIS і радіально-базисних нейронних мереж. Головною властивістю ANFIS є те, що апріорно задані правила налаштовуються в процесі навчання для отримання більш точних результатів.

Підходи на основі штучних нейронних мережах (ШНМ), нейронних нечітких мереж (ННМ) є універсальним засобом моделювання складних процесів великої розмірності, але в меншій мірі ефективні при моделюванні процесів і процедур малої і середньої розмірності. Вони також потребують додаткових ресурсів на машинне навчання моделей.

Динамічні процеси в реальних системах характеризуються складними багатофункціональними залежностями і істотною нелінійністю. Моделювання таких процесів в задачах прогнозування, діагностики, ідентифікації, класифікації на основі існуючих класичних підходів викликає певні труднощі. Теоретико – системний аналіз на основі представлення

процесів математичними рівняннями дозволяє створювати досить точні моделі, але досить трудомісткий і складний в реальних розробках. Експериментальний системний аналіз або ідентифікація базуються на моделях, параметри яких ґрунтуються на даних вимірювань. Перевагами таких моделей є більш короткі терміни розробки, але їх якість істотно залежить від коректного вибору структури та інтерпретації результатів вимірювань.

Особливістю ННМ є те, що вони відносяться до непараметрическим моделям, а це викликає складності у визначенні відповідності (інтерпретації) її параметрів в термінах реальних процесів. В задачах ідентифікації на основі апроксимуючих властивостей ШНМ найбільш ефективними представляються моделі на основі багат шарового персептрона і мереж з радіально - базисними функціями (РБФ). Архітектури моделей на основі мереж Кохонена або Хопфілда і деяких інших більшою мірою використовуються в задачах угруповання і класифікації.

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

Нейронна мережа може вчитися на вхідних і вихідних даних для визначення поведінки системи. Ці знання можуть бути використані для створення нечітких правил і функцій приналежності, що істотно зменшує час, необхідний на розробку. Таке об'єднання також допомагає вирішити проблему неінтерпритуємості результатів, одержуваних за допомогою нейронних мереж. Вираз ваг нейронної мережі за допомогою нечітких правил забезпечує розуміння роботи нейронної мережі, що, у свою чергу, допомагає створювати більш ефективні програми.

Нейро-фаззі системи можуть генерувати нечіткі логічні правила і функції приналежності для складних систем, яким не досить стандартного нечіткого підходу. Для таких систем стандартна нечітка логіка використовує

складні ієрархічні правила, для зменшення кількості самих правил, що відповідно знижує ефективність і точність рішення.

Дані ШНМ використовують нелінійні функції приналежності. Перевагою таких функцій є рівномірний розподіл знань між базою правил і базою функцій приналежності, що демонструється в скороченні обсягу бази правил. Важливим є той факт, що властивості узагальнення і навченості нейро-фаззі систем дозволяють генерувати правила і функції приналежності, що забезпечують більш достовірне і точне рішення в порівнянні з альтернативними методами. За допомогою коректної комбінації нечіткої логіки і нейронних мереж представляється можливим повне відображення знань нейронної мережі нечіткою логікою, у випадку, якщо алгоритми нечіткої логіки повністю засновані на архітектурі нейронної мережі. Це дозволяє генерувати нечіткі логічні рішення з заздалегідь заданою точністю виходу.

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

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

1.6 Постановка задачі дослідження

У багатьох задачах інтелектуального аналізу даних, пов'язаних з обробкою емпіричних кількісних спостережень, дані можуть бути спотворені

пропусками. Задачі відновлення таких спостережень приділялася достатня увага, при цьому вельми ефективними в даній ситуації опинилися підходи, засновані на математичному апараті обчислювального інтелекту і, перш за все, штучних нейронних мереж, які вирішують задачу відновлення загублених спостережень.

Разом з тим, описані підходи до відновлення даних працездатні лише у випадках, коли вихідні дані задані апріорно, а сама таблиця «об'єкт-властивість» або тимчасової ряд мають фіксовану кількість спостережень, тобто не змінюються в процесі обробки. У той же час існує широкий клас задач, коли дані надходять на обробку послідовно так, як це відбувається при навчанні самоорганізованих карт Кохонена [9] або їх модифікацій [10].

Для роботи з такими даними найбільш ефективними представляються штучні нейронні мережі, нейро-фаззи системи, гібридні системи, відомі своїми універсальними апроксимуючими властивості 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.

В даний час існує ряд підходів, хоча найбільшого поширення набув, так званий, конструктивний підхід, при якому система обчислювального інтелекту, стартуючи з найпростішої архітектури, поступово нарощує свою складність, одночасно налаштовуючи і свої параметри, до досягнення необхідної якості виконання завдання. У той же час більшість таких систем обробляє інформацію в пакетному режимі, що ускладнює їх використання в випадках, коли дані на обробку надходять послідовно в online режимі.

2 АДАПТИВНІ НЕЙРО-ФАЗЗИ МЕТОДИ ДЛЯ КЛАСТЕРИЗАЦІЇ ВИКРИВЛЕНИХ ДАНИХ

Завдання кластеризації масивів даних, що описуються наборами векторів-образів, досить часто зустрічається в багатьох додатках, пов'язаних з інтелектуальним аналізом даних, при цьому останнім часом особлива увага приділяється, так званій, нечіткій кластеризації [70-71], коли обробляється вектор-образ ознак з різними рівнями ймовірностей, можливостей або приладдя може належати одночасно до кількох класів.

Разом з тим, у багатьох задачах DataMining, включаючи, звичайно, кластеризації, вихідні масиви даних можуть містити пропуски, інформація в яких з тих чи інших причин відсутня. Для роботи в подібних умовах досить ефективним може виявитися використання математичного апарату обчислювального інтелекту [73] і, перш за все, штучних нейронних мереж [44], що вирішують завдання відновлення втрачених спостережень, і модифікацій популярного методу нечітких k -середніх [5], що вирішують завдання кластеризації без такого відновлення.

Крім зазначених, існує ще цілий ряд підходів до обробки даних, що містять пропуски, однак, всі вони працездатні тільки в випадках, коли масив вихідних спостережень заданий заздалегідь в повному обсязі і не змінюється в процесі обробки. У той же час існує широкий клас задач, коли дані надходять на обробку послідовно в режимі реального часу так, як це відбувається при навчанні самоорганізованих карт Кохонена [1] або їх модифікацій [6].

2.1 Формальна постановка задачі кластеризації

Вихідною інформацією для стандартної задачі кластеризації є вибірка даних, сформована з N n -мірних векторів (образів) ознак

$X = \{x_1, x_2, \dots, x_N\} \subset R^n$, $x_k \in X$, $k = 1, 2, \dots, N$ - або номер спостереження в таблиці «об'єкт-властивість», або поточний дискретний час в задачах on-line обробки.

Результатом кластеризації є розбиття X на m -пересічних класів з деяким рівнем приналежності $U_q(k)$ -го вектора x_k ознак до q -му кластеру.

Вихідні дані в процесі обробки центруються і стандартизовані за всіма компонентами так, щоб всі спостереження належали гіперкубу $[-1, 1]^n$.

Таким чином вихідні дані набувають вигляду $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N\} \subset R^n$,

$$\tilde{x}_k = \{\tilde{x}_{k1}, \tilde{x}_{k2}, \dots, \tilde{x}_{kn}\}^T, \quad -1 \leq \tilde{x}_{ki} \leq 1, \quad 1 \leq m \leq N, \quad 1 < q < m, \quad 1 \leq i \leq n,$$

$1 \leq k \leq N$, при цьому N в загальному випадку не обмежена і може

збільшуватися з плином часу. Відзначимо, що традиційно прийняте в картах

Кохонена (SOM) перетворення даних до виду $\|\tilde{x}_k\| = 1$ в даному випадку не

має сенсу, оскільки якщо x_k містить пропуск - обчислення норми такого

вектора неможливо, а якщо x_k містить викидень 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 в околиці

нуля, а дані з викидами - близько -1 і $+1$.

2.2 Функції відстані

Вибір відстані між об'єктами є вирішуючим моментом дослідження, від нього багато в чому залежить остаточний варіант розбиття об'єктів на класи при даному алгоритмі розбиття.

2.2.1 Евклідова відстань

Найбільш простий шлях обчислення відстаней між об'єктами в багатовимірному просторі полягає в обчисленні евклідових відстаней. Якщо є дво- або тривимірний простір, то цей захід є реальною геометричною відстанню між об'єктами в просторі.

$$D^2(\tilde{x}_k, w_q) = \|\tilde{x}_k - w_q\|_2^2 \quad (2.1)$$

Зауважимо, що евклідова відстань (і його квадрат) обчислюється по вихідним, а не за стандартизованим даним. Це звичайний спосіб його обчислення, який має певні переваги. Проте, на відстані можуть сильно впливати відмінності між осями, за координатами яких обчислюються ці відстані.

2.2.2 Відстань міських кварталів (Манхеттенська відстань)

Ця відстань є просто сумою модулів різниць по координатах. У більшості випадків ця міра відстані призводить до таких же результатів, як і для звичайна відстань Евкліда. Однак зазначимо, що для цього заходу вплив окремих великих різниць (викидів) зменшується (так як вони не зводяться в квадрат). Манхеттенська відстань обчислюється за формулою:

$$D(\tilde{x}_k, w_q) = \|\tilde{x}_k - w_q\|_1 = \sum_{i=1}^n |\tilde{x}_{ki} - w_{qi}|. \quad (2.2)$$

2.2.3 Відстань Чебишева

Це відстань може виявитися корисною при визначенні двох об'єктів як "різні", якщо вони розрізняються за якоюсь однією координатою (будь-яким одним виміром). Відстань Чебишева обчислюється за формулою:

$$D(\tilde{x}_k, w_q) = \max |\tilde{x}_{ki} - w_{qi}| \quad (2.3)$$

2.2.4 Часткова відстань

Дана міра відстані обчислюється за формулою (2.4)

$$D_P^2(\tilde{x}_k, w_q) = \frac{n}{\delta_{k\Sigma}} \sum_{i=1}^n (\tilde{x}_{ki} - w_{qi})^2 \delta_{ki}, \quad (2.4)$$

де w_{qi} - а компонента q -го прототипу (центроїда) відповідного кластера ($q = 1, 2, \dots, m$),

$$\delta_{ki} = \begin{cases} 0 & | \tilde{x}_{ki} \in X_G, \\ 1 & | \tilde{x}_{ki} \in X_F, \end{cases}$$

$$\delta_{k\Sigma} = \sum_{i=1}^n \delta_{ki}.$$

Часткова відстань (PD) дозволяє врахувати пропущені значення, що дозволяє кластеризувати зіпсовані дані. Жоден з наведених вище методів розрахунку відстаней не надає такої можливості.

2.2.5 Методи кластеризації даних

При великій кількості спостережень ієрархічні методи кластерного аналізу не придатні. У таких випадках використовують неієрархічні методи, засновані на поділі, які представляють собою ітеративні методи подрібнення вихідної сукупності. У процесі поділу нові кластери формуються до тих пір, поки не буде виконано правило зупинки.

Така неієрархічна кластеризація полягає в поділі набору даних на певну кількість окремих кластерів. Існує два підходи. Перший полягає у визначенні меж кластерів як найбільш щільних ділянок в багатовимірному просторі вихідних даних, тобто визначення кластера там, де є велике "зкупчення точок". Другий підхід полягає в мінімізації заходів відмінностей об'єктів.

2.2.6 Адаптивний метод ймовірнісної нечіткої кластеризації даних.

Вводячи цільову функцію кластеризації

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

при обмеженнях $\sum_{q=1}^m U_q(k) = 1$, $0 < \sum_{k=1}^N U_q(k) < N$ і вирішуючи завдання нелінійного програмування, приходимо до алгоритму ймовірнісної нечіткої кластеризації [73,97]

$$\begin{cases} U_q^{(\tau+1)}(k) = \frac{(D^2(\tilde{x}_k, w_q^{(\tau)}))^{-\frac{1}{1-\beta}}}{\sum_{l=1}^m (D^2(\tilde{x}_k, w_l^{(\tau)}))^{-\frac{1}{1-\beta}}}, \\ w_q^{(\tau+1)} = \frac{\sum_{k=1}^N (U_q^{(\tau+1)})^\beta \tilde{x}_k}{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta}, \end{cases} \quad (2.5)$$

де w_q - прототип (центр ваги) -го кластера,

$\beta > 1$ - параметр, іменований фаззифікатором і визначає «розмитість» меж між класами,

$D^2(\tilde{x}_k, w_q)$ - відстань між \tilde{x}_k і w_q в прийнятій метриці,

$\tau = 0, 1, 2, \dots$ - індекс епохи обробки інформації, яка організовується у вигляді послідовності $w_q^{(0)} \rightarrow U_q^{(1)} \rightarrow w_q^{(1)} \rightarrow U_q^{(2)} \rightarrow \dots$

Процес обчислень триває до виконання умови

$$\|w_q^{(\tau+1)} - w_q^{(\tau)}\| \leq \varepsilon \quad \forall 1 \leq q \leq m,$$

де ε - наперед задається поріг точності обчислень.

Вибираючи $\beta = 2$ і приймаючи евклідову відстань $D^2(\tilde{x}_k, w_q) = \|\tilde{x}_k - w_q\|^2$, отримуємо популярний алгоритм нечітких с-середніх (FCM) Дж. Бездека:

$$\begin{cases} U_q^{(\tau+1)}(k) = \frac{\|\tilde{x}_k - w_q^{(\tau)}\|^{-2}}{\sum_{l=1}^m \|\tilde{x}_k - w_l^{(\tau)}\|^{-2}}, \\ w_q^{(\tau+1)} = \frac{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^2 \tilde{x}_k}{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^2}. \end{cases}$$

Процес нечіткої кластеризації може бути організований в on-line режимі по мірі послідовного надходження даних на обробку. При цьому пакетний алгоритм може бути переписаний в рекуррентній формі

$$\begin{cases} U_q(k+1) = \frac{(D^2(\tilde{x}_{k+1}, w_q^{(k)}))^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (D^2(\tilde{x}_{k+1}, w_l^{(k)}))^{\frac{1}{1-\beta}}}, \\ w_q(k+1) = w_q(k) + \eta(k+1) U_q^\beta(k+1) (\tilde{x}_{k+1} - w_q(k)), \end{cases} \quad (2.6)$$

(тут $\eta(k+1)$ - параметр кроку навчання), що є узагальненням градієнтної процедури кластеризації Парка-Деггера [79] і алгоритму навчання Чанга-Лі.

Якщо дані надходять на обробку з високою частотою, перерахунок по епохах не проводиться, якщо ж ця частота невелика, то між моментами часу k і $k+1$ можна організувати кілька епох в прискореному часі, при цьому моменту k відповідає $\tau = 0$, а в $(k+1)$ -й момент цей перерахунок закінчується.

Цікаво також зауважити, що перший вираз може бути переписано в формі

$$\begin{aligned} U_q(k+1) &= \frac{(D^2(\tilde{x}_k, w_q(k)))^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (D^2(\tilde{x}_k, w_l(k)))^{\frac{1}{1-\beta}}} = \\ &= \frac{(D^2(\tilde{x}_k, w_q(k)))^{\frac{1}{1-\beta}}}{(D^2(\tilde{x}_k, w_q(k)))^{\frac{1}{1-\beta}} + \sum_{\substack{l=1 \\ l \neq q}}^m (D^2(\tilde{x}_k, w_l(k)))^{\frac{1}{1-\beta}}} = \\ &= \frac{1}{1 + (D^2(\tilde{x}_k, w_q(k)))^{\frac{1}{\beta-1}} \sum_{\substack{l=1 \\ l \neq q}}^m (D^2(\tilde{x}_k, w_l(k)))^{\frac{1}{1-\beta}}}, \end{aligned}$$

для евклідової метрики і $\beta = 2$ набуває вигляду функції Коші з параметром ширини σ^2 :

$$U_q(k+1) = \frac{1}{1 + \frac{\|\tilde{x}_k - w_q(k)\|^2}{\sigma^2}},$$

$$\sigma^2 = \left(\sum_{\substack{l=1 \\ l \neq q}}^m \|\tilde{x}_k - w_l(k)\|^{-2} \right)^{-1}.$$

Дана обставина дозволяє переписати другий вираз (2.6) при $\beta = 2$ у формі

$$\begin{aligned}w_q(k+1) &= w_q(k) + \eta(k+1)U_q^2(k+1)(\tilde{x}_{k+1} - w_q(k)) = \\ &= w_q(k) + \eta(k+1)\varphi_q(k+1)(\tilde{x}_{k+1} - w_q(k)),\end{aligned}$$

де $U_q^2(k+1) = \varphi_q(k+1)$ дзвінкоподібна функція сусідства нейро-фаззі мережі Кохонена, призначеної для вирішення завдань нечіткої кластеризації [80] на основі принципу «переможець отримує більше» (WТМ).

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

Для перевірки коректної роботи програми та запропонованого алгоритму було проведено імітаційне моделювання з використанням UCI репозиторію Iris і Wine. Імітаційне моделювання проводилося в середовищі MATLAB.

3.1 MATLAB

MATLAB - це програмна платформа, розроблена спеціально для інженерів і вчених. Серцем MATLAB є мова MATLAB, матричний мову, що дозволяє найбільш природне вираження обчислювальної математики.

MATLAB поєднує в собі настільну середу, налаштовану для ітеративного аналізу і процесів проектування з мовою програмування, який безпосередньо виражає математику матриць і масивів.

Використовуючи MATLAB, ви можете:

- 1) Аналізувати дані;
- 2) Розроблений алгоритм;
- 3) Створення моделей і додатків.

Мова, Додатки і вбудовані математичні функції дозволяють швидко вивчити кілька підходів, щоб прийти до вирішення. MATLAB дозволяє вам переходити від досліджень до виробництва, розгортаючи корпоративні додатки і вбудовані пристрої, а також інтегруючись з моделями і додатками Simulink і Model-Based Design.

Мова, інструментарій та вбудовані математичні функції якого дозволяють вам досліджувати різні підходи і отримувати рішення швидше, ніж з використанням електронних таблиць або традиційних мов програмування, таких як C / C ++ або Java.

MATLAB широко використовується в таких областях, як:

- 1) обробка сигналів і зв'язок
- 2) обробка зображень і відео
- 3) системи управління
- 4) автоматизація тестування і вимірювань
- 5) фінансовий інжиніринг
- 6) обчислювальна біологія і т.п.

Понад мільйон інженерів і вчених по всьому світу використовують MATLAB в якості мови технічних обчислень.

MATLAB в порівнянні з традиційними мовами програмування (C / C ++, Java, Pascal, FORTRAN) дозволяє на порядок скоротити час вирішення типових завдань і значно спрощує розробку нових алгоритмів. MATLAB являє собою основу всього сімейства продуктів MathWorks і є головним інструментом для вирішення широкого спектра наукових і прикладних задач, в таких областях як: моделювання об'єктів і розробка систем управління, проектування комунікаційних систем, обробка сигналів і зображень, вимірювання сигналів і тестування, фінансове моделювання, обчислювальна біологія та ін.

Ядро MATLAB дозволяє максимально просто працювати з матрицями реальних, комплексних і аналітичних типів даних і зі структурами даних і таблицями пошуку. MATLAB містять задану вбудовані функції лінійної алгебри (LAPACK, BLAS), швидкого перетворення Фур'є (FFTW), функції для роботи з поліномами, функції базової статистики та чисельного рішення диференціальних рівнянь; розширені математичні бібліотеки для Intel MKL. Всі вбудовані функції ядра MATLAB розроблені й оптимізовані фахівцями і працюють швидше або так само, як їх еквівалент на C / C ++.

MATLAB надає безліч методів для аналізу даних, розробки алгоритмів і створення моделей. Мова MATLAB включає в себе математичні функції для інженерних і наукових операцій. Вбудовані математичні функції використовують процесор-оптимізовані бібліотеки, призначені для прискорення векторних і матричних обчислень. Доступні наступні операції:

- 1) Інтерполяція і регресія
- 2) Диференціювання та інтегрування
- 3) Системи лінійних рівнянь
- 4) Фур'є аналіз
- 5) Власні значення і сингулярні числа матриць
- 6) Звичайні диференціальні рівняння
- 7) Розріжені матриці

Розширення MATLAB надають спеціалізований функціонал в таких областях як статистика, оптимізація, обробка сигналів, машинне навчання.

MATLAB надає інструменти для отримання, аналізу та візуалізації даних, що дозволяють досліджувати проблему швидше, ніж це можливо за допомогою електронних таблиць або традиційних мов програмування. Також ви можете документувати результати у вигляді графіків, звітів або публікації коду MATLAB.

MATLAB дозволяє вам отримувати доступ до даних з файлів, інших додатків, баз даних, зовнішніх пристроїв. Ви можете читати дані з файлів таких популярних форматів як Microsoft Excel, текстових або довічних файлів, зображень, аудіо та відео файлів, наукових форматів (netCDF і HDF). Функції вводу-виводу дозволяють працювати з файлами даних будь-яких форматів. Використовуючи розширення MATLAB можна отримувати дані з різних пристроїв, таких як послідовний порт комп'ютера або звукова карта, а також потокові дані в реальному часі з вимірювальних пристроїв безпосередньо в MATLAB для аналізу і візуалізації. Крім того, ви можете управляти такими приладами, як осцилографи, аналізатори сигналів і генератори коливань спеціальної форми.

MATLAB дозволяє управляти, фільтрувати і здійснювати попередню обробку даних. Ви можете досліджувати дані для знаходження трендів, перевірки гіпотез, побудови описових моделей. В MATLAB включені функції для фільтрації, згладжування, згортки і швидкого перетворення Фур'є

(FFT). Продукти-розширення включають можливості підбору кривих і поверхонь, багатовимірної статистики, спектрального аналізу, аналізу зображень, ідентифікації систем та інші інструменти аналізу.

Системні вимоги для MATLAB:

1. Operating Systems:

- Windows 10;
- Windows 7 Service Pack 1;
- Windows Server 2019;
- Windows Server 2016.

2. RAM:

- Minimum: 4 GB;
- Recommended: 8 GB.

3. Graphics:

- No specific graphics card is required;
- Hardware accelerated graphics card supporting OpenGL 3.3 with 1GB GPU memory is recommended.

4. Processors:

- Minimum: Any Intel or AMD x86-64 processor;
- Recommended: Any Intel or AMD x86-64 processor with four logical cores and AVX2 instruction set support.

5. Disk:

- Minimum: 2.9 GB of HDD space for MATLAB only, 5-8 GB for a typical installation;
- Recommended: An SSD is recommended;
- A full installation of all MathWorks products may take up to 29 GB of disk space.

3.2 DATA BASE (Виборка)

3.2.1 Iris

Iris – об'єктно-орієнтована система управління базами даних, що розробляється в лабораторіях Hewlett-Packard. Iris призначена для задоволення потреб нових і нових додатків баз даних, таких як офісні та інженерні інформаційні системи, системи, засновані на знаннях, виробничі програми та проектування апаратного та програмного забезпечення. Ці програми потребують багатого набору можливостей, які не підтримуються поточним поколінням (тобто реляційними) СУБД. Модель даних Iris-це модель об'єкта та функції. Він надає три основні конструкції об'єктів, типів і функцій. Як і в інших об'єктних системах, об'єкти Iris мають унікальний ідентифікатор й можуть бути доступні і управлятися тільки за допомогою функцій. Об'єкти класифікуються за типом. Об'єкти, що належать до одного типу, спільно використовують загальні функції. Типи організовані в ієрархію з успадкованими функціями.

3.2.2 Wine

Інформація про набір даних:

Ці дані є результатами хімічного аналізу вин, вирощених в одному регіоні Італії, але отриманих з трьох різних сортів. Аналіз визначається кількістю 13 виборців у кожному з трьох типів вин.

Атрибути :

- 1) Alcohol;
- 2) Malic acid;
- 3) Ash;
- 4) Alcalinity of ash;
- 5) Magnesium;

- 6) Total phenols;
- 7) Flavanoids;
- 8) Nonflavanoid phenols;
- 9) Proanthocyanins;
- 10) Color intensity;
- 11) Hue;
- 12) OD280/OD315 of diluted wines;
- 13) Proline.

У контексті класифікації це добре поставлена проблема з "добре себе ведуть" структурами класів. Гарний набір даних для першого тестування нового класифікатора, але не дуже складний.

Всі атрибути є безперервними.

Статистики немає, але пропонується стандартизувати змінні для певних цілей (наприклад, для нас з класифікаторами, які не є масштабно-інваріантними)

Примітка: 1-й атрибут-ідентифікатор класу (1-3)

3.3 Експериментальні дослідження

В експериментальних дослідженнях порівнювалися з FCM, за основними параметрами кластеризації: коефіцієнту розбиття (PC), класифікаційної ентропії (CE), індексом розділів (SC), індексу розподілу (S), індексу CE і Бені (XB), індексу даних (DI).

Коефіцієнт розбиття (PC): вимірює кількість "перекриття" між кластерами.

Класифікаційна ентропія (CE): вона вимірює нечіткість тільки розділу кластера, що схоже на коефіцієнт розподілу.

Індекс розділів (SC): це відношення суми компактності та розділеності кластерів. Це сума окремих дій щодо дій кластеру, нормалізованих за

допомогою поділу нечіткою кардинальністю кожного кластеру. SC корисний при порівнянні різних розділів, що мають однакову кількість кластерів. Менше значення SC вказує на кращий розділ.

Індекс поділу (S): навпаки індексу розділів (SC), поділ in-dex використовує поділ мінімальної відстані для дійсності розділу.

Індекс Сі і Бені (XB): він спрямований на кількісне визначення співвідношення загальної зміни в кластерах та поділу кластерів. Оптимальна кількість кластерів повинна мінімізувати значення індексу.

Індекс даних (DI): цей індекс спочатку пропонується використовувати для ідентифікації "компактних і добре відокремлених кластерів". Таким чином, результат кластеризації потрібно переглядати, оскільки це був алгоритм жорсткого розподілу.

Результати експериментів представлені в Таблиці 1 та Таблиці 2. Результати запропонованого нами алгоритму ми також порівняли з іншими більш відомими, такими як алгоритм кластеризації нечітких С-засобів (FCM).

Порівняльний результат приведено в Таблицях 3.1 та 3.2.

Таблиця 3.1 – Результати експериментальних досліджень, порівняльний аналіз кластеризації даних вибірки Iris UCI repository

Алгоритми	Iris UCI repository					
	PC	CE	SC	S	XB	DI
Адаптивний ймовірнісний нечіткий метод кластеризації даних	0,0199	0,0122	-0,2439	0,0022	0,0015	1
FCM	0,8011	0,3410	0,2567	0,0030	7,1965	0,0080

Таблиця 3.2 – Результати експериментальних досліджень, порівняльний аналіз кластеризації даних вибірки Wine UCI repository

Алгоритми	Wine UCI repository					
	PC	CE	SC	S	XB	DI
Адаптивний ймовірнісний нечіткий метод кластеризації даних	0,0230	0,0219	-0,3007	0,0032	0,0065	0,9999
FCM	0,7411	0,2389	0,3112	0,0040	6,9945	0,0078

ВИСНОВКИ

Задача кластеризації масивів даних часто зустрічається в багатьох додатках, пов'язаних з інтелектуальним аналізом даних. Традиційний підхід до вирішення цих завдань передбачає, що кожне спостереження може відноситися тільки до одного кластеру, хоча більш природною видається ситуація, коли обробляється вектор ознак з різними рівнями ймовірностей або можливостей може належати відразу декільком класів. Дана ситуація є предметом розгляду нечіткого (фаззі) кластерного аналізу, що інтенсивно розвивається в даний час.

Відомі підходи і рішення працездатні лише у випадках, коли вихідний масив даних заданий заздалегідь в повному обсязі і не змінюється в процесі аналізу. У той же час існує досить широкий клас задач, коли дані надходять на обробку послідовно в on-line режимі так, як це відбувається при навчанні самоорганізованих карт Кохонена.

Розглянуто задачу ймовірнісної нечіткої on-line кластеризації даних з пропусками і показано, що вона може бути вирішена на основі самоорганізованої нейро-фаззі мережі Кохонена та запропонованого адаптивного алгоритму самонавчання, що є гібридом правила «переможець отримує більше» і рекурентних алгоритмів нечіткої кластеризації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ritter, H., & Schulten, K. (1988, July). Kohonen's self-organizing maps: exploring their computational capabilities. In Proceedings of IEEE international conference on neural networks (Vol. 1, pp. 109-116).
2. Gorshkov, Y., Kolodyazhniy, V., & Bodyanskiy, Y. (2009, June). New recursive learning algorithms for fuzzy Kohonen clustering network. In Proc. 17th Int. Workshop on Nonlinear Dynamics of Electronic Systems (pp. 58-61).
3. Kasabov, N. K. (2007). Evolving connectionist systems: the knowledge engineering approach. Springer Science & Business Media.
4. Lughofer, E. (2011). Evolving fuzzy systems-methodologies, advanced concepts and applications (Vol. 53). Berlin: Springer.
5. Gan, G., Ma, C., & Wu, J. (2007). Data clustering: theory, algorithms, and applications (Vol. 20). Siam.
6. Загоруйко, Н. Г. (1979). Емпирическое предсказание. Наука, Сиб. отд-ние.
7. Gorban, A. N., Kégl, B., Wunsch, D. C., & Zinovyev, A. Y. (Eds.). (2008). Principal manifolds for data visualization and dimension reduction (Vol. 58, pp. 96-130). Berlin: Springer.
8. Marwala, T. (Ed.). (2009). Computational Intelligence for Missing Data Imputation, Estimation, and Management: Knowledge Optimization Techniques: Knowledge Optimization Techniques. IGI Global.
9. Bishop, C. M. (1995). Neural networks for pattern recognition. Oxford university press.
10. Gorban, A. N., Rossiev, A. A., & Wunsch, D. C. (2000). Neural network modeling of data with gaps. *Радіоелектроніка, інформатика, управління*, (1 (3)).
11. Tkacz, M. (2005). Artificial neural networks in incomplete data sets processing. In *Intelligent Information Processing and Web Mining* (pp. 577-583). Springer, Berlin, Heidelberg.

12. Golden, R. M., & Golden, R. (1996). *Mathematical methods for neural network analysis and design*. MIT Press.
13. Braun, H. (2013). *Neuronale Netze: Optimierung durch Lernen und Evolution*. Springer-Verlag.
14. Dracopoulos, D. C. (2013). *Evolutionary learning algorithms for neural adaptive control*. Springer.
15. Shepherd, A. J. (2012). *Second-order methods for neural networks: Fast and reliable training methods for multi-layer perceptrons*. Springer Science & Business Media.
16. Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.
17. Bodyanskiy, Y., Shafronenko, A., & Volkova, V. (2012). Adaptive clustering of incomplete data using neuro-fuzzy Kohonen network. *Artificial Intelligence Methods and Techniques for Business and Engineering Applications*—Rzeszow-Sofia: ITHEA, 287-296.
18. Плісс, І. П., Шевякова, А. Ю., & Шевякова, Ю. Ю. (2011). Нейромережеве відновлення пропусків у таблицях даних. *Наукові праці [Чорноморського державного університету імені Петра Могили]*. Сер.: Комп'ютерні технології, (160, Вип. 148), 59-61.
19. Шафроненко, А. Ю., Волкова, В. В., & Бодянский, Е. В. (2011). Адаптивная кластеризация данных с пропущенными значениями. *Радиоелектроніка, інформатика, управління*, (2 (25)).
20. Bodyanskiy, Y., Shafronenko, A., & Volkova, V. (2012). Adaptive fuzzy probabilistic clustering of incomplete data. *INFORMATION MODELS & ANALYSES*, 112.
21. Shafronenko, A., Pliss, I., & Ye, B. (2013). The evolving adaptive neural network for data processing with missing observations. *Радиоелектроніка, інформатика, управління*, (2 (29)).
22. Bodyanskiy, Y., & Shafronenko, A. (2013). Adaptive fuzzy clustering for data with missing values based on the nearest prototype-centroid strategy.

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

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

25. Шафроненко, А. Ю., Волкова, В. В., & Бодянский, Е. В. (2011). Адаптивная кластеризация данных с пропущенными значениями. Радиоелектроніка, інформатика, управління, (2 (25)).

26. Bodyanskiy, Y., Shafronenko, A., & Volkova, V. (2013, September). Neuro fuzzy Kohonen network for incomplete data clustering using optimal completion strategy. In Proceedings 20 th East West Fuzzy Colloquium (pp. 25-27).

27. McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), 115-133.

28. Braun, H., Feulner, J., & Malaka, R. (2013). *Praktikum Neuronale Netze*. Springer-Verlag.

29. Beltratti, A., Margarita, S., & Terna, P. (1996). *Neural networks for economic and financial modelling*. London, UK: International Thomson Computer Press.

30. Hunt, K. J., Sbarbaro, D., Żbikowski, R., & Gawthrop, P. J. (1992). Neural networks for control systems—a survey. *Automatica*, 28(6), 1083-1112.

31. Leondes, C. T. (1998). *Neural network systems techniques and applications: Advances in theory and applications (Vol. 7)*. Academic Press.

32. Nelles, O., Ernst, S., & Isermann, R. (1997). Neuronale Netze zur Identifikation nichtlinearer, dynamischer Systeme: Ein Überblick. *at-Automatisierungstechnik*, 45(6), 251-262.

33. Ротштейн, А. П. (1999). Интеллектуальные технологии идентификации: нечеткие множества, генетические алгоритмы, нейронные сети. Винница: УНІВЕРСУМ-Вінниця, 320.
34. Tsoukalas, L. H., & Uhrig, R. E. (1996). Fuzzy and neural approaches in engineering. John Wiley & Sons, Inc.
35. Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3), 665-685.
36. Jang, J. S., & Sun, C. T. (1993). Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE transactions on Neural Networks*, 4(1), 156-159.
37. Хан, J., Пеи, J., & Камбер, М. (2011). Data mining: concepts and techniques. Elsevier.
38. Альберт, А., & Липцер, Р. Ш. (1977). Регрессия, псевдоинверсия и рекуррентное оценивание: Пер. с англ. Наука. Гл. ред. физ.-мат. лит..
39. Бодянский, Е. В., Плисс, И. П., & Соловьева, Т. В. (1986). Многошаговые оптимальные упредители многомерных нестационарных стохастических процессов. Доклады АН УССР.–1986.–Сер. А, (12), 47-49.
40. Bodyanskiy, Y., Kolodyazhniy, V., & Stephan, A. (2001, October). An adaptive learning algorithm for a neuro-fuzzy network. In *International Conference on Computational Intelligence* (pp. 68-75). Springer, Berlin, Heidelberg.
41. Yamakawa, T. (1992). A neo fuzzy neuron and its applications to system identification and prediction of the system behavior. In *Proc. of the 2nd Int. Conf. on Fuzzy Logic & Neural Networks* (pp. 477-483).
42. Uchino, E., & Yamakawa, T. (1997). Soft computing based signal prediction, restoration, and filtering. In *Intelligent Hybrid Systems* (pp. 331-351). Springer, Boston, MA.
43. Miki, T. S. U. T. O. M. U., & Yamakawa, T. A. K. E. S. H. I. (1999). Analog implementation of neo-fuzzy neuron and its on-board learning. *Computational Intelligence and Applications*, 144-149.

44. Hathaway, R. J., & Bezdek, J. C. (2001). Fuzzy c-means clustering of incomplete data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(5), 735-744.
45. Bezdek, J. C. (2013). *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media.
46. Суєтин, П. (2018). *Классические ортогональные многочлены*. Litres.
47. Karlin, S., & Studden, W. J. (1966). *Tchebyeff systems: with applications in analysis and statistics*.
48. Семесенко, М. П. (1983). *Методы обработки и анализа измерений в научных исследованиях*. К.: Донецк: Вища шк.
49. Бодянский, Е. В., Викторов, Е. А., & Слипченко, А. Н. (2007). Ортосинапс, ортонейроны и нейропредиктор на их основе. *Системы обробки інформації*, (4), 139-143.
50. Yang, S. S., & Tseng, C. S. (1996). An orthogonal neural network for function approximation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(5), 779-785.
51. Lee, T. T., & Jeng, J. T. (1998). The Chebyshev-polynomials-based unified model neural networks for function approximation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(6), 925-935.
52. András, P. (1999). Orthogonal RBF neural network approximation. *Neural Processing Letters*, 9(2), 141-151.
53. Sher, C. F., Tseng, C. S., & Chen, C. S. (2001). Properties and performance of orthogonal neural network in function approximation. *International Journal of intelligent systems*, 16(12), 1377-1392.
54. Patra, J. C., & Kot, A. C. (2002). Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 32(4), 505-511.
55. Ye, B., Kolodyazhniy, V., & Slipchenko, O. (2004). Artificial neural network with orthogonal activation functions for dynamic system identification.

Synergies between Information Processing and Automation. Ed. O. Sawodny and P. Scharff–Aachen: Shaker Verlag, 24-30.

56. Stasiak, B., & Yatsymirskyy, M. (2006, June). Fast orthogonal neural networks. In *International Conference on Artificial Intelligence and Soft Computing* (pp. 142-149). Springer, Berlin, Heidelberg.

57. Rodriguez, N., & Cubillos, C. (2007, September). Orthogonal neural network based predistortion for OFDM systems. In *Electronics, Robotics and Automotive Mechanics Conference (CERMA 2007)* (pp. 225-228). IEEE.

58. Wang, H., & Yu, S. (2010). Tracking control of robot manipulators based on orthogonal neural network. *International Journal of Modelling, Identification and Control*, 11(1-2), 130-135.

59. Ye, B., Kolodyazhniy, V., & Slipchenko, O. (2004). Structural and synaptic adaptation in the artificial neural networks with orthogonal activation functions. *Sci. Proc. of Riga Technical University. Comp. Sci., Inf. Technology and Management Sci*, 20, 69-76.

60. Bodyanskiy, Y., & Slipchenko, O. (2006). Ontogenic neural networks using orthogonal activation functions. *Prace Naukowe Akademii Ekonomicznej we Wrocławiu, (1121 Pozyskiwanie wiedzy i zarządzanie wiedzą)*, 13-20.

61. Bodyanskiy, Y., Pliss, I., & Slipchenko, O. (2007). Growing neural networks using nonconventional activation functions.

62. Bodyanskiy, Y., Dolotov, A., Pliss, I., & Viktorov, Y. (2008). The cascade orthogonal neural network.

63. Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3), 264-323.

64. Дюк, В., & Самойленко, А. (2001). *Data Mining: учебный курс*. СПб.: Питер, 368, 16.

65. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

66. Xu, R., & Wunsch, D. C. (2005). *Survey of clustering algorithms*.

67. Abonyi, J., & Feil, B. (2007). Cluster analysis for data mining and system identification. Springer Science & Business Media.
68. Olson, D. L., & Delen, D. (2008). Advanced data mining techniques. Springer Science & Business Media.
69. Бодянский, Е. В., & Руденко, О. Г. (2004). Искусственные нейронные сети: архитектуры, обучение, применения. Харьков: Телетех, 369.
70. Höppner, F., Klawonn, F., Kruse, R., & Runkler, T. (1999). Fuzzy cluster analysis: methods for classification, data analysis and image recognition. John Wiley & Sons.
71. Chung, F. L., & Lee, T. (1994). Fuzzy competitive learning. *Neural Networks*, 7(3), 539-551.
72. Park, D. C., & Dagher, I. (1994, June). Gradient based fuzzy c-means (GBFCM) algorithm. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)* (Vol. 3, pp. 1626-1631). IEEE.
73. Krishnapuram, R., & Keller, J. M. (1993). A possibilistic approach to clustering. *IEEE transactions on fuzzy systems*, 1(2), 98-110.
74. Bodyanskiy, Y. (2005). Computational Intelligence Techniques for Data Analysis. In *Leipziger Informatik-Tage* (pp. 15-36).
75. Davé, R. N., & Krishnapuram, R. (1997). Robust clustering methods: a unified view. *IEEE Transactions on fuzzy systems*, 5(2), 270-293.
76. Bodyanskiy, Y., Gorshkov, Y., Kokshenev, I., & Kolodyazhniy, V. (2005). Robust recursive fuzzy clustering algorithms. *Proceedings of East West Fuzzy Colloquim*, 301-308.
77. Bodyanskiy, Y., Kokshenev, I., Gorshkov, Y., & Kolodyazhniy, V. (2006). Outlier resistant recursive fuzzy clustering algorithms. In *Computational Intelligence, Theory and Applications* (pp. 647-652). Springer, Berlin, Heidelberg.
78. Bezdek, J. C., Keller, J., Krisnapuram, R., & Pal, N. (1999). Fuzzy models and algorithms for pattern recognition and image processing (Vol. 4). Springer Science & Business Media.

79. Klawonn, F. (2013, November). What can Fuzzy cluster analysis contribute to clustering of high-dimensional data?. In International Workshop on Fuzzy Logic and Applications (pp. 1-14). Springer, Cham.
80. Keller, A., & Klawonn, F. (2000). Fuzzy clustering with weighting of data variables. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 8(06), 735-746.
81. Maimon, O., & Rokach, L. (Eds.). (2007). *Soft computing for knowledge discovery and data mining*. Springer Science & Business Media.
82. Rutkowski, L. (2008). *Computational intelligence: methods and techniques*. Springer Science & Business Media.
83. Perfilieva, I. (2006). Fuzzy transforms: Theory and applications. *Fuzzy sets and systems*, 157(8), 993-1023.
84. Perfilieva, I. (2004). Fuzzy transforms. In *Transactions on rough sets II* (pp. 63-81). Springer, Berlin, Heidelberg.
85. Cochocki, A., & Unbehauen, R. (1993). *Neural networks for optimization and signal processing*. John Wiley & Sons, Inc..
86. Little, R. J., & Rubin, D. B. (2019). *Statistical analysis with missing data* (Vol. 793). John Wiley & Sons.
87. Злоба, Е., & Яцкив, И. (2002). Статистические методы восстановления пропущенных данных. *Computer Modelling & New Technologies*, 6(1), 51-61.
88. Dave, R. N., & Sen, S. (1997, June). On generalizing the noise clustering algorithms. In *Proc. 7th Int. Fuzzy Systems Association World Congress (IFSA'97)* (Vol. 3, pp. 205-210).
89. Höppner, F., Klawonn, F., Kruse, R., & Runkler, T. (1999). *Fuzzy cluster analysis: methods for classification, data analysis and image recognition*. John Wiley & Sons.
90. Zhang, J. S., & Leung, Y. W. (2003). Robust clustering by pruning outliers. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(6), 983-998.

91. Wu, C. W., Chen, J. L., & Wang, J. H. (2001, October). Self-organizing mountain method for clustering. In 2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat. No. 01CH37236) (Vol. 4, pp. 2434-2438). IEEE.

92. Yao, Y., Chen, Y. Q., & Chen, L. (2000). A novel similarity measure for data clustering. *Intelligent Data Analysis*, 4(5), 421-431.

93. Veenman, C. J., & Reinders, M. J. (2005). The nearest subclass classifier: A compromise between the nearest mean and nearest neighbor classifier. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9), 1417-1429.

94. Zhang, C., Qin, Y., Zhu, X., Zhang, J., & Zhang, S. (2006, August). Clustering-based missing value imputation for data preprocessing. In 2006 4th IEEE International Conference on Industrial Informatics (pp. 1081-1086). IEEE.

95. Yong, Z., Yue'e, L., & Shixiong, X. (2008, December). Robust fuzzy-possibilistic c-means algorithm. In 2008 Second International Symposium on Intelligent Information Technology Application (Vol. 1, pp. 669-673). IEEE.

96. Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

97. Bodyanskiy, Y., Gorshkov, Y., Kokshenev, I., Kolodyazhniy, V., & Shilo, O. (2006). Recursive fuzzy clustering algorithm for segmentation of biomedical time series. In *East West Fuzzy Colloquium* (pp. 130-139).

98. Hawkins, S., He, H., Williams, G., & Baxter, R. (2002, September). Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery* (pp. 170-180). Springer, Berlin, Heidelberg.

99. Babuka, R., Van der Veen, P. J., & Kaymak, U. (2002, May). Improved covariance estimation for Gustafson-Kessel clustering. In 2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291) (Vol. 2, pp. 1081-1085). IEEE.

100. Bodyanskiy, Y., Gorshkov, Y., Kokshenev, I., Kolodyazhniy, V., & Shilo, O. (2006). Recursive fuzzy clustering algorithm for segmentation of biomedical time series. In *East West Fuzzy Colloquium* (pp. 130-139).
101. Tipping, M. E., & Bishop, C. M. (1999). Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2), 443-482.
102. Cowell, R. G., Dawid, P., Lauritzen, S. L., & Spiegelhalter, D. J. (2006). *Probabilistic networks and expert systems: Exact computational methods for Bayesian networks*. Springer Science & Business Media.
103. Зайченко, Ю. П. (2008). *Нечеткие модели и методы в интеллектуальных системах*. К.: Слово, 344.
104. Круглов, В. В., & Борисов, В. В. (2002). *Искусственные нейронные сети. Теория и практика*. М.: Горячая линия-Телеком.
105. Згуровский, М. З., & Зайченко, Ю. П. (2013). *Основы вычислительного интеллекта*. К.: Наук. думка.
106. Yin, Y., Kaku, I., Tang, J., & Zhu, J. (2011). *Data mining: Concepts, methods and applications in management and engineering design*. Springer Science & Business Media.
107. Everitt, B. S., Landau, S., Leese, M., & Stahl, D. (2011). *Cluster Analysis*. –John Wiley & Sons. Ltd., New York, 330.
108. Miamoto, S., Ichihashi, H., & Honda, K. (2008). *Algorithms for Fuzzy Clustering: Methods in c-Means Clustering with Applications*.
109. De Oliveira, J. V., & Pedrycz, W. (Eds.). (2007). *Advances in fuzzy clustering and its applications*. John Wiley & Sons.
110. Mueller, E. T. (2011). *Neurocomputing: Learning, Architectures and Modeling*. Nova Science Publishers, Inc..
111. Suresh, S., Sundararajan, N., & Savitha, R. (2013). Supervised learning with complex-valued neural networks (pp. 125-132). Berlin: Springer.
112. Babinec, Š., & Pospíchal, J. (2008, November). Gating echo state neural networks for time series forecasting. In *International Conference on Neural Information Processing* (pp. 200-207). Springer, Berlin, Heidelberg.