

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Богатюк Володимир Васильович

**Програмний модуль синтезу радіально-базисної нейронної
мережі для аналізу релевантних даних /
Software module for synthesis radial-base neural network for
analysis of relevant data**

спеціальність: 123 – Комп'ютерна інженерія
освітньо-професійна програма – Комп'ютерна інженерія

Кваліфікаційна робота

Виконав: студент групи КІ-42
Богатюк Володимир Васильович

Науковий керівник
К.т.н., старший викладач Савка Н.Я.

Кваліфікаційну роботу
Допущено до захисту
«__» _____ 20__ р.

Завідувач кафедри
_____ О.М. Березький

ТЕРНОПІЛЬ - 2021

РЕЗЮМЕ

Кваліфікаційна робота на тему «Програмний модуль синтезу радіально-базисної нейронної мережі для аналізу релевантних даних» зі спеціальності 123 «Комп'ютерна інженерія» освітнього ступеня «бакалавр» містить 84 сторінки пояснюючої записки, 17 рисунків, 11 таблиць, 2 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою кваліфікаційної роботи є розробка програмного модуля для аналізу релевантних даних на основі радіальних штучних нейронних мереж.

Розглянуто задачу аналізу даних на відповідність потребам користувача із застосуванням штучних нейронних мереж із радіально-базисними функціями. Навчання штучних нейронних мереж радіального типу відбувається у два етапи: налаштування параметрів прихованого шару мережі та налаштування синаптичних ваг.

Охарактеризовано методи навчання штучних нейронних мереж із радіально-базисними функціями й показано, що для визначення центрів радіально-базисних функцій застосовуємо алгоритм субтрактивної кластеризації. Налаштування вагових коефіцієнтів синаптичних зв'язків відбувається на основі алгоритму, що ґрунтується на мінімізації середньоквадратичної похибки.

Розроблено алгоритм синтезу штучних нейронних мереж із радіально-базисними функціями для аналізу релевантних даних та їх прогнозування для прийняття управлінських рішень.

Розроблено структуру програмного модуля синтезу штучних нейронних мереж із радіально-базисними функціями для аналізу релевантних даних та їх прогнозування. Побудовано архітектуру штучної нейронної мережі радіального типу.

Ключові слова: РАДІАЛЬНО-БАЗИСНА ШТУЧНА НЕЙРОННА МЕРЕЖА, ВАГОВІ КОЕФІЦІЄНТИ, РЕЛЕВАНТНІ ДАНІ.

RESUME

Qualification thesis “Software Module for Synthesis Radial-Base Neural Network for Analysis of Relevant Data” in the specialty 123 "Computer Engineering" "Bachelor" education degree contains 84 pages of explanatory notes, 17 figures, 11 tables, 2 appendixes. The volume of graphic material is 2 sheets of A3 format.

The aim of the qualification project is to development of software for analysis relevant data based on artificial neural networks with radial-basic functions.

The task of data analysis to meet users' needs with use of artificial neural networks with radial-basic functions is considered. The training of artificial neural networks of the radial type takes place in two stages: adjustment of the parameters of the hidden network layer and adjustment of synaptic weights.

The methods of training of artificial neural networks with radial-basic functions are described and it is shown that for the determination of centers of radial-basic functions use subtractive clustering algorithm. Adjustment of weight coefficients of synaptic bonds is based on algorithms minimizing the mean square error.

Algorithm for training of artificial neural networks with radial-basic functions for analysis of relevant data and their forecasting for management decisions was developed.

The structure of the program module for training of artificial neural networks with radial-basic functions for analysis of relevant data and their was developed.

The architecture of artificial neural network of radial type was developed.

Key words: RADIAL-BASED ARTIFICIAL NEURAL NETWORK, WEIGHT COEFFICIENTS, RELEVANT DATA.

ЗМІСТ

Вступ.....	10
1 Управління технологічними процесами та засоби їх моделювання.....	12
1.1 Етапи розв’язку задач управління технологічними процесами.....	12
1.2 Аналіз штучних нейронних мереж	16
1.3 Аналіз програмних засобів моделювання штучних нейронних мереж.....	23
1.4 Постановка задачі кваліфікаційної роботи	30
2 Алгоритм аналізу релевантних даних на основі штучних нейронних мереж.....	32
2.1 Архітектура штучних нейронних мереж із радіально-базисними функціями	32
2.2 Алгоритми налаштування параметрів радіально-базисних нейронних мереж	35
2.3 Алгоритм синтезу архітектури радіально-базисної штучної нейронної мережі для аналізу релевантних даних	39
3. Програмна реалізація алгоритму синтезу радіально-базисної нейронної мережі	43
3.1 Структура програмного модуля аналізу релевантних даних	43
3.2 Реалізація алгоритму аналізу релевантних даних.....	46
3.3 Аналіз даних на відповідність на основі радіально-базисних нейронних мереж.....	52
4 Техніко-економічне обґрунтування розробки проекту.....	60
4.1 Визначення витрат на оплату праці та відрахувань у соціальні фонди	60
4.2 Розрахунок ціни проекту	67
4.3 Визначення економічної ефективності розробки проекту	70
Висновки.....	73

					КР.КІ.07171/17.00.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розробив		Богатюк В.В.			ПРОГРАМНИЙ МОДУЛЬ СИНТЕЗУ РАДІАЛЬНО- БАЗИСНИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ АНАЛІЗУ РЕЛЕВАНТНИХ ДАНИХ	Літ.	Арк.	Акрушів
Перевір.		Савка Н.Я.					8	84
Консульт.		Савка Н.Я.				ЗУНУ.ФКІТ. КІ-42		
Н. Контр.		Мельник Г.М.						
Затвердив		Березький О.М.						

Список використаних джерел.....	75
Додаток А Світлокопії публікації.....	80
Додаток Б Довідка про використання	84

					КР.КІ.07171/17.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Останнім часом спостерігається бурхливий розвиток методів машинного навчання, і, зокрема, штучних нейронних мереж (ШНМ). Така тенденція відбувається тому, що математичний апарат нейронних мереж є одним з найбільш перспективних напрямків машинного навчання.

Штучна нейромережа загалом є самонавчальною системою, яка в процесі навчання встановлює зв'язки між вхідними та вихідними даними. Однією із важливих властивостей таких мереж є гнучкість та масштабованість для великої кількості вхідних параметрів. В той же час, ШНМ є адаптивними системами, оскільки вони володіють властивістю перебудувати свою роботу для відповідності до поточних даних, не маючи при цьому будь-якої інформації щодо моделі. Крім цього, нейронні мережі здатні апроксимувати будь-яку функцію з певною точністю нелінійними моделями, що дозволяє моделювати різноманітні процеси навколишнього середовища.

Таким чином, застосування ШНМ уможливорює автоматизацію процесів вирішення задач прогнозування на основі аналізу даних, при цьому знизивши суттєво трудомісткість і час обробки даних. Важливим етапом розробки архітектури ШНМ перцептронного типу є обчислення кількості прихованих шарів та нейронів у них. Ця задача вимагає проведення низки експериментальних досліджень, а це потребує часових витрат, що неефективно при розв'язуванні задач управління, які включають певні процеси.

Сьогодні досить популярними при вирішенні вищезазначених труднощів є ШНМ із радіально-базисними функціям (РБФ). Такі мережі мають достатньо просту архітектуру, оскільки включають лише один прихований шар. Таким чином, задача налаштування архітектури зводиться до обчислення кількості нейронів у прихованому шарі, що значно знижує часові витрати. Також зазначені нейронні мережі здатні моделювати нестационарні процеси на основі неоднорідних вибірок даних.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Зважаючи на вищезазначене, метою кваліфікаційної роботи є розробка архітектури ШНМ із РБФ та дослідження її можливостей стосовно задач управління, які включають збір та аналіз релевантних даних.

Об'єктом дослідження є технологічні процеси у задачах управління.

Предметом дослідження метод ідентифікації штучних нейронних мереж із радіально-базисними функціями для аналізу релевантних даних.

Для досягнення мети кваліфікаційної роботи необхідно вирішити такі задачі:

- проаналізувати існуючі архітектури штучних нейронних мереж;
- охарактеризувати етапи розв'язку задач управління;
- дослідити існуючі програмні засоби реалізації штучних нейронних мереж радіального типу;
- спроектувати штучну нейронну мережу з радіально-базисними функціями для аналізу релевантних даних;
- дослідити ефективність розробленої архітектури штучної нейронної мережі на основі задачі аналізу даних на відповідність потребам користувача;
- охарактеризувати техніко-економічні показники розробки проекту.

За результатами роботи опубліковано тези доповіді на VIII Всеукраїнській науково-практичній конференції молодих науковців "ІТ-2021" [12]. Копії публікації наведено у додатку А.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

1 УПРАВЛІННЯ ТЕХНОЛОГІЧНИМИ ПРОЦЕСАМИ ТА ЗАСОБИ ЇХ МОДЕЛЮВАННЯ

1.1 Етапи розв'язку задач управління технологічними процесами

Сучасні технологічні процеси являють собою складні системи, що складаються з великої кількості різних технологічних агрегатів і установок. Системи управління такими процесами, в свою чергу – це складні ієрархічні системи, в яких кожен рівень ієрархії виконує свої задачі. Останнім часом виділяють три рівні управління технологічними процесами на підприємстві [1, 2]:

- верхній рівень керування підприємством – на якому вирішують стратегічні задачі, питання управління ресурсами в довгостроковій перспективі, та, крім того, виконують функції підтримки виробництва, наприклад, довгострокове планування в місячному або річному масштабі;

- інтеграційний рівень управління підприємством – на цьому рівні вирішують задачі оперативного управління технологічними процесами виробництва, а відповідна автоматизована система управління забезпечує ефективне використання ресурсів, таких як сировина, персонал, інші виробничі кошти. Крім цього, на зазначеному рівні відбувається забезпечення оптимального виконання планових задач на різних часових проміжках;

- нижній рівень вирішує класичні задачі управління технологічними процесами.

Таким чином, є кілька рівнів управління технологічними процесами, кожен із яких зі своєю специфікою. У зв'язку з постійним зростанням виробничих потужностей і сильно вираженим динамізмом виробничих процесів зростають і вимоги до ефективності управління технологічними процесами.

Основними цілями автоматизації технологічних процесів є [2]:

- підвищення загальної ефективності технологічного процесу;
- підвищення екологічності;
- підвищення безпеки;

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

- підвищення економічності.

Варто зазначити, що поряд із ускладненням виробничих технологічних процесів бурхливо розвиваються й інформаційні технології. Тому сучасному етапу управління технологічними процесами характерне широке застосування засобів обчислювальної техніки, в тому числі таких, як:

- реалізація алгоритмів оцінки якості управління;
- реалізація алгоритмів підвищення ефективності споживання виробничих ресурсів;
- збір і оцінка релевантних даних;
- реалізація алгоритмів підвищення ефективності проектування;
- реалізація діалогу «людина-система управління».

Автоматизація управління технологічними процесами є одним з основних факторів подальшого розвитку організації, підвищення її організаційно-технічного рівня і зростання продуктивності праці. Удосконалення структури та методів управління підприємствами за допомогою автоматизованих систем управління, з одного боку, і впровадження автоматизованих систем управління технологічними процесами – з іншого, є в даний час основними факторами підвищення ефективності суспільного виробництва.

На перших етапах для автоматизації управління технологічними процесами застосовувалися лише прості пристрої – регулятори (механічні, електромеханічні або електричні). Завдання автоматизації зводилася в основному до забезпечення стійкості регульованих процесів. Згодом з'явилися оптимальні регулятори, здатні при зміні зовнішніх умов змінювати значення регульованих параметрів для підтримування процесу в найбільш вигідному режимі.

Із впровадженням ЕОМ появилися автоматизовані системи управління технологічним процесом (АСУТП). Це – людино-машинні комплекси взаємопов'язаних і взаємодіючих елементів, де на основі централізованого отримання і комплексної обробки інформації, що надходить від зазначених елементів і зовнішнього середовища, виробляються керуючі впливи для

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

підтримки процесу в заданому режимі або для його зміни відповідно до заданого алгоритму управління [1, 23].

Відповідно до існуючої класифікації за рівнем автоматизації АСУТП можуть бути таких видів [23]:

– інформаційні АСУТП, в яких введення та обробка інформації від їх елементів і від зовнішнього середовища частково або повністю автоматизовані, а вироблення сигналів про вихід контрольованих параметрів технологічних процесів за задані межі здійснюється автоматично. Вплив на процеси за отриманими сигналами (зворотний зв'язок) здійснюється людиною, тобто людина приймає на основі отриманої інформації необхідні рішення для цілеспрямованої зміни технологічних процесів;

– інформаційно-порадні АСУТП, які, зберігаючи всі функції інформаційних АСУТП, відрізняються від останніх тим, що коригуючий вплив на процес відповідно до заданого алгоритму управління виробляється автоматично, зворотній зв'язок як і раніше здійснюється людиною;

– керуючі АСУТП – системи, в яких вироблення коригувальних впливів на керований технологічний процес відбувається автоматично, тобто це системи з автоматизованим зворотним зв'язком.

При обґрунтуванні вибору АСУТП необхідно визначити функції, які може виконувати система, та джерела її техніко-економічної ефективності стосовно технологічних процесів.

В АСУТП, зазвичай, використовуються спеціально розроблені керуючі обчислювальні машини. Вони оснащені пристроєм зв'язку із об'єктом управління, яке містить велику кількість каналів (до декількох тисяч), за якими можуть передаватися сигнали як в аналоговому, так і в цифровому вигляді. Ці пристрої оснащені аналогово-цифровими і цифро-аналоговими перетворювачами, що дозволяють управляти процесом обробки інформації на цифровій обчислювальній машині.

Основним завданням АСУТП, що використовують керуючі обчислювальні машини є проведення великої кількості складних математичних обчислень на

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

основі інформації, отриманої від об'єкта управління, і вироблення оптимального регулюючого впливу на процес або порад оператору в результаті зазначених обчислень.

Застосування обчислювальної техніки значно розширює можливості автоматичного управління, а саме:

- різке збільшення кількості регульованих параметрів;
- можливість реалізації складних алгоритмів управління;
- універсальність;
- здатність швидко перебудовуватися (за рахунок зміни програм) на нові алгоритми управління;
- наявність пам'яті, що дозволяє враховувати при управлінні не тільки поточний стан регульованого об'єкта, але і його передісторію.

Автоматизовані системи комплексного управління мають організаційно-технологічний характер, тобто крім основної функції можуть виконувати обробку певної планово-виробничої інформації, на базі якої проводиться оперативне управління роботою відповідного підрозділу (цеху, дільниці). Ці системи можуть бути автоматичними і автоматизованими. Характерною особливістю останніх є участь людини на різних етапах формування алгоритму управління технологічною операцією, стадією або процесом.

Інша важлива задача АСУТП – це робота із програмним управлінням. Перевага обладнання із програмним управлінням полягає у скороченні часу налагодження для обробки даних, підвищенні продуктивності праці шляхом більш раціонального використання його технологічних можливостей, стабільності протікання процесу та ін.

Для того, щоб повною мірою реалізувати переваги обладнання із програмним управлінням, необхідно автоматизувати процес підготовки керуючих програм. Це наступний крок у розвитку комплексної автоматизації виробництва – з'єднання програмно-керованої технології з автоматизованим проектуванням.

З усього перерахованого вище випливає, що сучасні системи управління технологічними процесами виконують роботу зі збору та аналізу зібраної

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

інформації, оцінці цієї інформації на відповідність, оптимізації виробничих процесів, виходячи з отриманих даних. Людина ж в такій системі виконує не більше, ніж функцію прийняття рішень, ґрунтуючись на представлених їй системою даних. При цьому найчастіше кількість людей, необхідних для прийняття рішень, знижується – частину рішень приймають алгоритмами.

Такий вид систем управління називається експертними системами. Однак виникає проблема – існує чимало досить складних нелінійних задач, які або перевищують можливості загальноприйнятих алгоритмічних методів, або вимагають великих матеріальних і часових витрат [2].

Для вирішення подібних задач використання методів машинного навчання підходить більшою мірою, ніж використання класичних методів алгоритмізації.

1.2 Аналіз штучних нейронних мереж

В останні роки спостерігається бурхливий розвиток технологій машинного навчання та розвиток нейромережових технологій, зокрема. Це пов'язано певними факторами. Технології нейромережового моделювання, мають такі переваги [3, 4-6, 19, 22, 30]:

- високу ефективність роботи з великими обсягами даних;
- здатність відтворювати складні нелінійні співвідношення;
- відсутність обмежень на функцію розподілу даних;
- можливість обробки різнотипних експериментальних даних;
- збереження працездатності при наявності пропусків в таблиці навчальної вибірки.

Основна ідея, що стоїть за нейронними мережами – людський головний мозок. Нейронні мережі, по суті своїй, є наближеною моделлю людського мозку і взаємодії між природними нейронами. Природний нейрон має відростки двох типів. Перший тип – дендрити. Дендритів може бути безліч, і вони виконують

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

функцію прийняття інформації, що надходить у вигляді електричних імпульсів, від інших нейронів. Також природний нейрон має аксон, в єдиному екземплярі. Аксон передає інформацію іншим нейронам. З'єднання між нейронами реалізується в синапсах. Прийняті синапсом сигнали підводяться до тіла нейрона. Тут вони підсумовуються, причому одні входи прагнуть порушити нейрон, інші перешкодити порушенню. Коли сума збудливих сигналів в тілі перевищує деякий поріг, нейрон збуджується, посылаючи по аксону сигнал збудження інших нейронів.

У цій функціональній схемі є багато узагальнень, але, тим не менше, більшість моделей штучного нейрона моделюють лише ці прості властивості. Штучним нейроном називається вузол штучної нейронної мережі, що є моделлю біологічного нейрона. Математично штучний нейрон представляють, як деяку нелінійну функцію від єдиного аргументу – лінійної комбінації всіх вхідних сигналів. Ця функція називається функцією активації, спрацьовування або передавальною функцією. Отриманий з цієї функції результат надсилається на єдиний вихід штучного нейрона.

Формально штучний нейрон можна представити таким чином [30]:

$$y = f(u), \quad u = w_0 + w_1x_1 + w_ix_i, \quad i = 1, \dots, n, \quad (1.1)$$

де y – вихідний сигнал нейрона;

$f(u)$ – функція вихідного блоку нейрона;

w_i – синаптичний коефіцієнт, він же вага i -го входу;

x_i – i -й вхідний сигнал;

w_0 – початкове збудження або зміщення нейрона;

$i = 1, \dots, n$ – кількість входів.

Неформально штучний нейрон можна охарактеризувати як математичний суматор. Існує три типи нейронів – вхідні, проміжні (або приховані) і вихідні. Вхідні приймають вихідний вектор і передають його на вихід, можливо,

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

послабивши або посиливши його. Далі сигнал потрапляє в проміжні нейрони, які виконують основні обчислювальні операції, і звідти отримані обчислення йдуть на вихідні нейрони, що являють собою вихід мережі.

Можливі значення сигналів на входах нейрона вважають заданими в проміжку $[0; 1]$, при цьому вони можуть бути як аналоговими, так і дискретними. Передавальна функція найчастіше має область значень $[-1; 1]$ або $[0; 1]$, проте бувають і винятки, тому що для деяких нейронних мереж необхідна безперервна диференціація на всій числовій осі.

Існує кілька видів передавальної функції. Зупинимось лише на декількох з них. Порогова передавальна функція, по суті своїй, є перепадом. До тих пір, поки зважений сигнал не досягає деякого рівня T , вихідний сигнал дорівнює нулю. Після досягнення порогу вихідний сигнал дорівнює одиниці. Таким чином, порогову передавальну функцію можна формально описати як кусково-задану функцію [3]:

$$f(x) = \begin{cases} 0, & x < T; \\ 1, & x \geq T. \end{cases} \quad (1.2)$$

Радіально-базисна функція передачі приймає в якості аргументу відстань між деяким вхідним вектором і деяким заданим наперед центром функції. Чим ближче вхідний вектор до центру, тим вище значення активаційної функції.

Лінійна функція пов'язує лінійно сигнал вихідного сигналу зі зваженою сумою сигналів на вході:

$$f(x) = tx. \quad (1.3)$$

Так само архітектуру нейронних мереж прийнято розрізняти за методом поширення сигналу, або ж за характером зв'язків. Одним з методів поширення сигналу є метод прямого поширення, в якому всі зв'язки направлені строго від

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

вхідних нейронів до вихідних. Існують також рекурентні нейронні мережі, в яких сигнал із вихідних нейронів частково передається назад на вхідні нейрони, таким чином здійснюючи своєрідну фільтрацію даних.

Крім цього, нейронні мережі поділяють за методом навчання – існують нейронні мережі, які навчаються з «учителем» та самонавчаючі нейронні мережі.

Узагальнену схему штучної нейронної мережі представлено на рисунку 1.1.

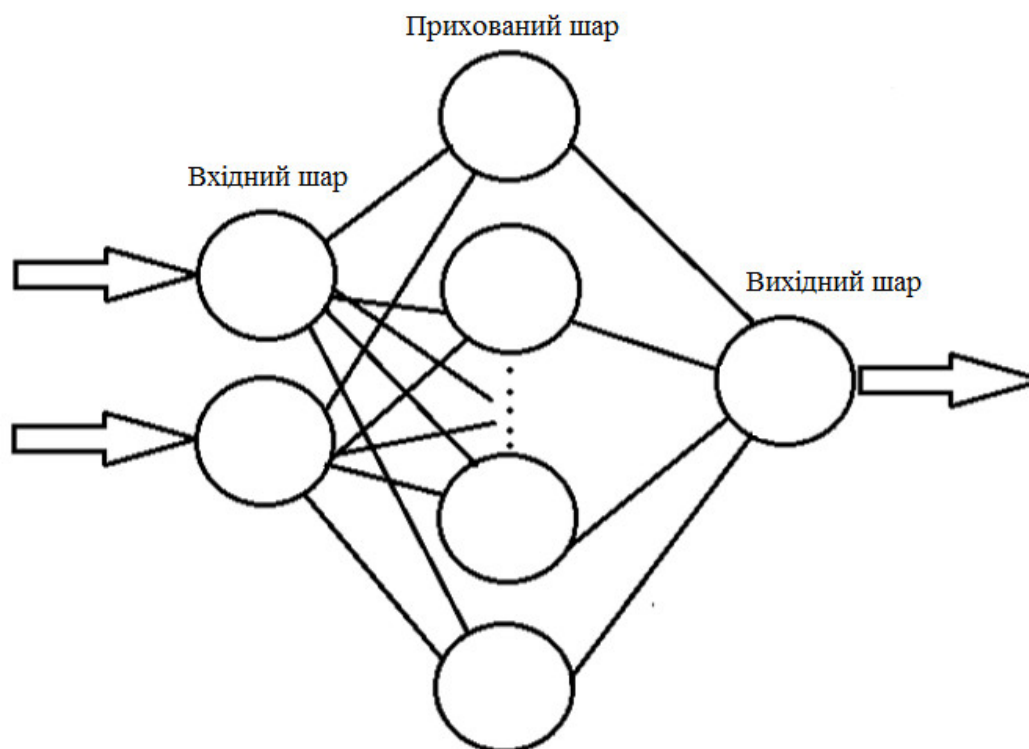


Рисунок 1.1 – Узагальнена схема одношарової штучної нейронної мережі

Для правильної роботи нейронної мережі необхідно її «навчити». При цьому потрібен масив промаркованих вхідних даних. В ході навчання нейронної мережі налаштовуються ваги нейронів прихованого шару. По закінченню навчання виходить нейронна мережа, яка, з певною ймовірністю помилки, отримавши вхідні дані видаватиме вихідні дані, які можна використовувати в подальшому для різних цілей.

Штучні нейронні мережі можуть використовуватися для вирішення найширшого кола задач, серед яких: класифікації, зприйняття рішень і управління, кластеризації, аналізу даних, прогнозування, апроксимації,

оптимізації. Саме для задач управління технологічними процесами підходять ті види нейронних мереж, які вирішують задачі прийняття рішень, прогнозування, апроксимації та оптимізації.

Такі задачі вирішує деякі види штучних нейронних мереж, зокрема, перцептрон, нейромережа із радіально-базисними функціями, нейронні мережі Кохонена, зокрема, самоорганізована карта Кохонена. Зупинимося детальніше на кожній із цих архітектур.

Розглянемо архітектуру перцептрона. За класифікацією нейронних мереж перцептрон є штучною нейронною мережею з одним (або декількома) прихованими шарами, пороговою функцією передачі і прямим поширенням сигналу. Перцептрон з одним прихованим шаром називається одношаровим, перцептрон з декількома прихованими шарами, відповідно, багатшаровим.

Функціонування перцептрона базується на принципі «стимул-реакція». У перцептрона наявні асоціативні і реагуючі елементи. Сигнал спочатку надходить на асоціативний елемент, а на реагуючий. Таким чином створюється набір «асоціацій» між вхідними стимулами і реакцією на виході. Найкраще нейронні мережі, побудовані за архітектурою перцептрона справляються із задачами класифікації та апроксимації. Однак перцептрони мають обмеження в задачах із інваріантним поданням образів, які не мають функціонально переваги над аналітичними методами, або можуть вимагати нереально великого часу і пам'яті для вирішення таких задач, як прогнозування.

Архітектура самоорганізованої карти Кохонена відрізняється тим, що ця нейронна мережа з навчанням без учителя. Найчастіше така нейронна мережа виконує задачі кластеризації або візуалізації. Крім цього часто застосовується для вирішення задач моделювання, прогнозування, пошуку закономірностей у великих масивах даних. За своєю суттю є методом проектування багатовимірного простору на простір з більш низькою розмірністю [4].

Мережа Кохонена досить проста за своєю структурою – включає усього два шари: вхідний і вихідний. Її називають картою. Елементи карти розташовуються в n -вимірному просторі, найчастіше двовимірному. Мережа Кохонена навчається

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

методом послідовних наближень. В процесі навчання на вхід подаються дані, але мережа при цьому підлаштовується не під еталонне значення виходу, а під закономірності у вхідних даних. Навчання починається з вибраного випадковим чином вихідного розташування центрів. В процесі послідовної подачі на вхід мережі навчальних прикладів визначається найбільш близький нейрон (той, у якого скалярний вихід ваги і поданого на вхід вектора мінімальний).

Такий нейрон оголошується переможцем і є центром при налаштуванні ваг у сусідніх нейронів. Інакше кажучи, таке правило навчання передбачає "змагальне" навчання з урахуванням відстані нейронів від "нейрона-переможця". Навчання при цьому полягає не в мінімізації помилки, як у випадку іншими штучними нейронними мережами, а в налаштуванні ваг (внутрішніх параметрів нейронної мережі) для найбільшого збігу із вхідними даними.

До переваг само організованих карт Кохонена можна віднести стійкість до зашумлених даних, можливість спрощення багатовимірних вхідних даних за допомогою візуалізації і швидке і некероване навчання.

До недоліків варто віднести сильну залежність від початкових налаштувань мережі – фахівцю, який працює з такою мережею, необхідно глибоко зануриться в об'єкт дослідження, і пройти серйозне навчання самому, перед тим як навчати безпосередньо самонавчальну карту Кохонена.

Розглянемо архітектуру радіально-базисної штучної нейронної мережі. Радіально-базисні нейронні мережі – такі нейромережі, які в якості функції активації використовують радіально-базисні функції. Виходом зазначених мереж є лінійна комбінація радіальних-базисних функцій і параметрів нейрона. Такі мережі мають безліч застосувань, в тому числі їх використовують для вирішення задач розпізнавання і класифікації, прийняття рішень і управління, прогнозування, апроксимації. Формально радіально-базисна функція може бути представлена таким чином [3]:

$$f = \varphi\left(\frac{x^2}{\sigma^2}\right), \quad (1.4)$$

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

де x – вектор вхідних сигналів нейрона,
 σ – ширина вікна функції,
 φ – якась спадна функція, найчастіше рівна нулю за межами певного відрізка.

Радіально-базисна мережа характеризується трьома особливостями:

- у ній є тільки один прихований шар.
- тільки нейрони прихованого шару мають нелінійну активаційну функцію.
- синаптичні ваги зв'язків вхідного і прихованого шарів дорівнюють одиниці.

Радіально-базисні штучні нейронні мережі мають суттєві переваги над іншими архітектурами. По-перше, як було згадано вище, вони мають всього лиш один прихований шар, що дозволяє звільнити розробника від вирішення питання про кількість прихованих шарів і полегшує реалізацію. По-друге, отримані лінійні комбінації можна оптимізувати за допомогою відомих методів лінійної оптимізації. У зв'язку з цим радіально-базисні ШНМ швидко навчаються.

Однак, у радіально-базисних мереж є і недоліки. Часто архітектури досить громіздкі при великій розмірності вектора входів, заздалегідь повинно бути відомо кількість еталонів.

Таким чином, для побудови мережі із радіально-базисних функцій необхідне виконання певних умов. В першу чергу, необхідна наявність еталонів, представлених у вигляді вагових векторів нейронів прихованого шару. Така задача на практиці може вирішуватися декількома способами, зокрема:

- взявши за еталонні вектори випадкові вектори із навчальної вибірки, однак в такому випадку збіжність нейронної мережі може виявитися досить довгою;
- попередньо обчисливши центри кластерів на основі одного із алгоритмів кластеризації. На практиці найчастіше використовується алгоритм *k-means* [24].

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

По-друге, необхідна наявність способу вимірювання відстані вхідного вектора від еталону. Зазвичай, це стандартна Евклідова відстань, проте може бути й будь-яка інша, наприклад, відстань Манхеттена.

По-третє, необхідно визначитися зі спеціальною функцією активації нейронів прихованого шару, яка задає обраний спосіб вимірювання відстані. Найчастіше використовується функція Гаусса, яка сіттево підсилює малу різницю між вхідним і еталонним векторами.

Проаналізувавши архітектури штучних нейронних мереж, можна зробити висновки про те, які з них найдоцільніше використати для задач управління технологічними процесами. Перцептрон має просту архітектуру, однак не має значних функціональних переваг над класичними алгоритмічними методами. Крім того, він мало пристосований для вирішення задач управління.

Самонавчальна карта Кохонена має суттєві переваги, але вимагає серйозної підготовки фахівця. Крім того, штучна нейронна мережа такої архітектури добре справляється із задачами кластеризації і візуалізації, але не така функціональна для задач оптимізації.

Оптимальним вибором для вирішення задачі управління технологічними процесами є радіально-базисна штучна нейронна мережа. Така нейромережа добре справляється із задачами управління, оптимізації та прогнозування, а також відносно проста у реалізації.

1.3 Аналіз програмних засобів моделювання штучних нейронних мереж

На даний момент існує кілька пакетів прикладних програм для розробки, навчання і експлуатації штучних нейронних мереж. З метою оцінки функціональності, а також виявлення переваг і недоліків для їх подальшого врахування під час розроблення власної системи зупинимося на деяких з них.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

MatLab ANFIS-редактор є абрєвіатурою Adaptive Neuro-Fuzzy Inference System – (адаптивна нейро-нечітка система). ANFIS-редактор дозволяє автоматично синтезувати з експериментальних даних нейро-нечіткої мережі [24]. При цьому функції приналежності синтезованих систем відразу налаштовані (навчені) так, щоб мінімізувати відхилення між результатами нечіткого моделювання та експериментальними даними. Головне вікно пакету ANFIS представлено на рисунок 1.2.

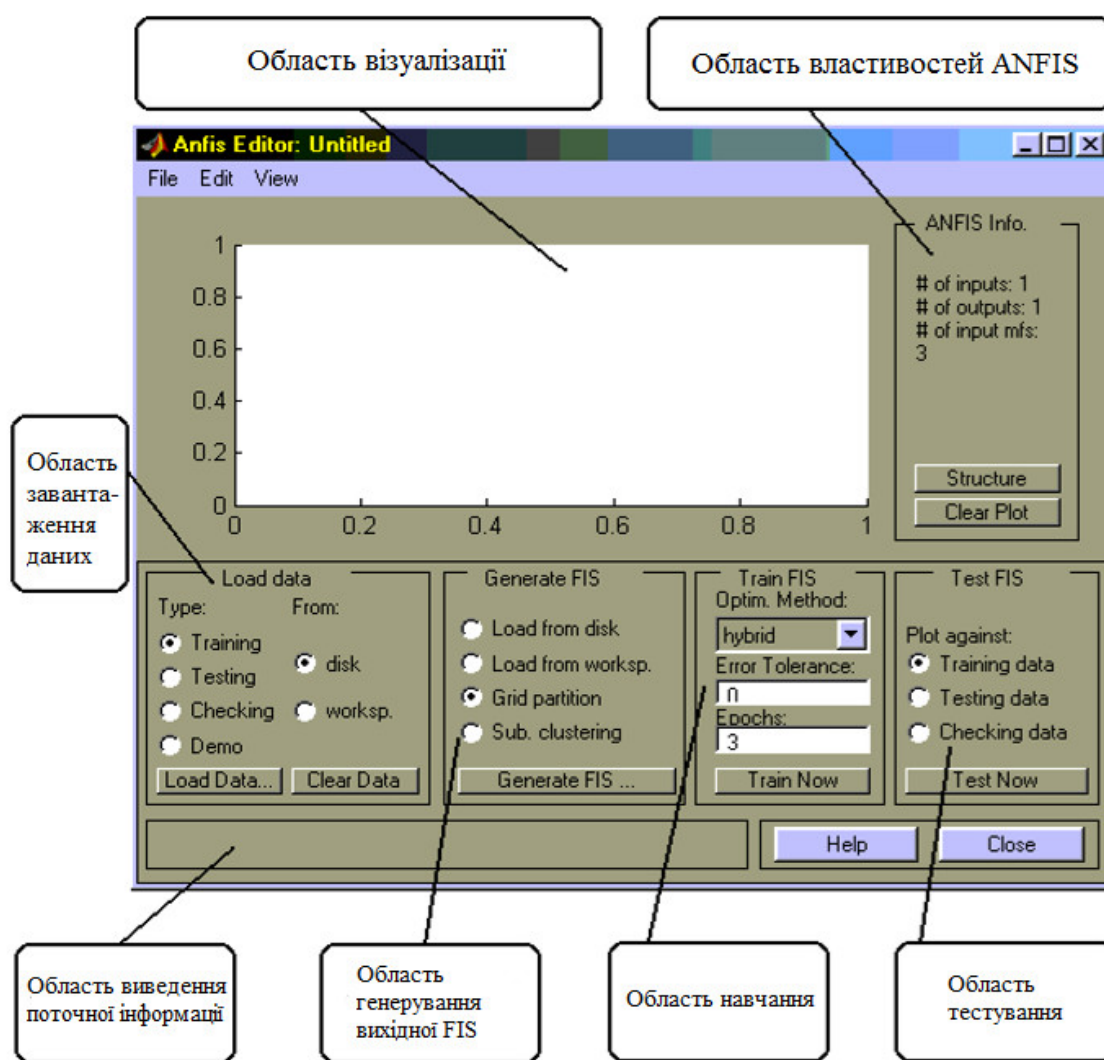


Рисунок 1.2 – Екранна форма головного вікна пакету ANFIS

STATISTICA Automated Neural Networks. Нейронну мережу для прогнозування даних надає пакет STATISTICA. В такому пакеті є можливість налаштувати параметри навчання: тип нейронної мережі (радіально-базисних

функцій або багатошаровий перцептрон), функцію активації, мінімальну та максимальну кількість нейронів в прихованому шарі, функцію помилки та інші [23].

STATISTICA Automated Neural Networks є одним із найбільш передових і ефективних продуктів на ринку для моделювання нейронних мереж. Пакет прикладних програм має унікальні переваги та можливості, зокрема автоматичного нейромережевого пошуку. Інструмент автоматизованої нейронної мережі дозволяє використовувати систему не тільки експертам з нейронних мереж, але і новачкам в області нейромережевих обчислень.

Крім того, існує можливість імпорту отриманої нейронної мережі в різні мови програмування, такі як Java або C ++. Однак, багатий функціонал зазначеного програмного забезпечення включає певні незручності. По-перше, трудомісткий процес початку роботи – для того, щоб повноцінно працювати із STATISTICA необхідно витратити певний час на те, щоб розібратися в досить заплутаному інтерфейсі, а також точно знати, яких результатів необхідно досягти. По-друге, рідко буває необхідність у застосуванні всього функціоналу, а опанувати потрібно усі інструменти. Інтерфейс пакету STATISTICA представлено на рисунку 1.3.

Нейропакет SNNS (Stuttgart Neural Network Simulator) [25] розроблено в Інституті паралельних і розподілених високопродуктивних систем (IPVR) Університету Штутгарта (University of Stuttgart) і підтримується університетом Тюбінгена (University of Tübingen). Спочатку це був програмний імітатор для нейронних мереж на робочих станціях Unix. В даний момент доступні версії під MS Windows. Метою проекту SNNS є створення ефективного і гнучкого середовища моделювання для дослідження і застосування нейронних мереж.

SNNS включає два основних компоненти: емулятор ядра, написаний на C, і графічний інтерфейс користувача. Емулятор ядра працює з внутрішніми мережами структур даних нейронних мереж і виконує операції навчання і перенавчання.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

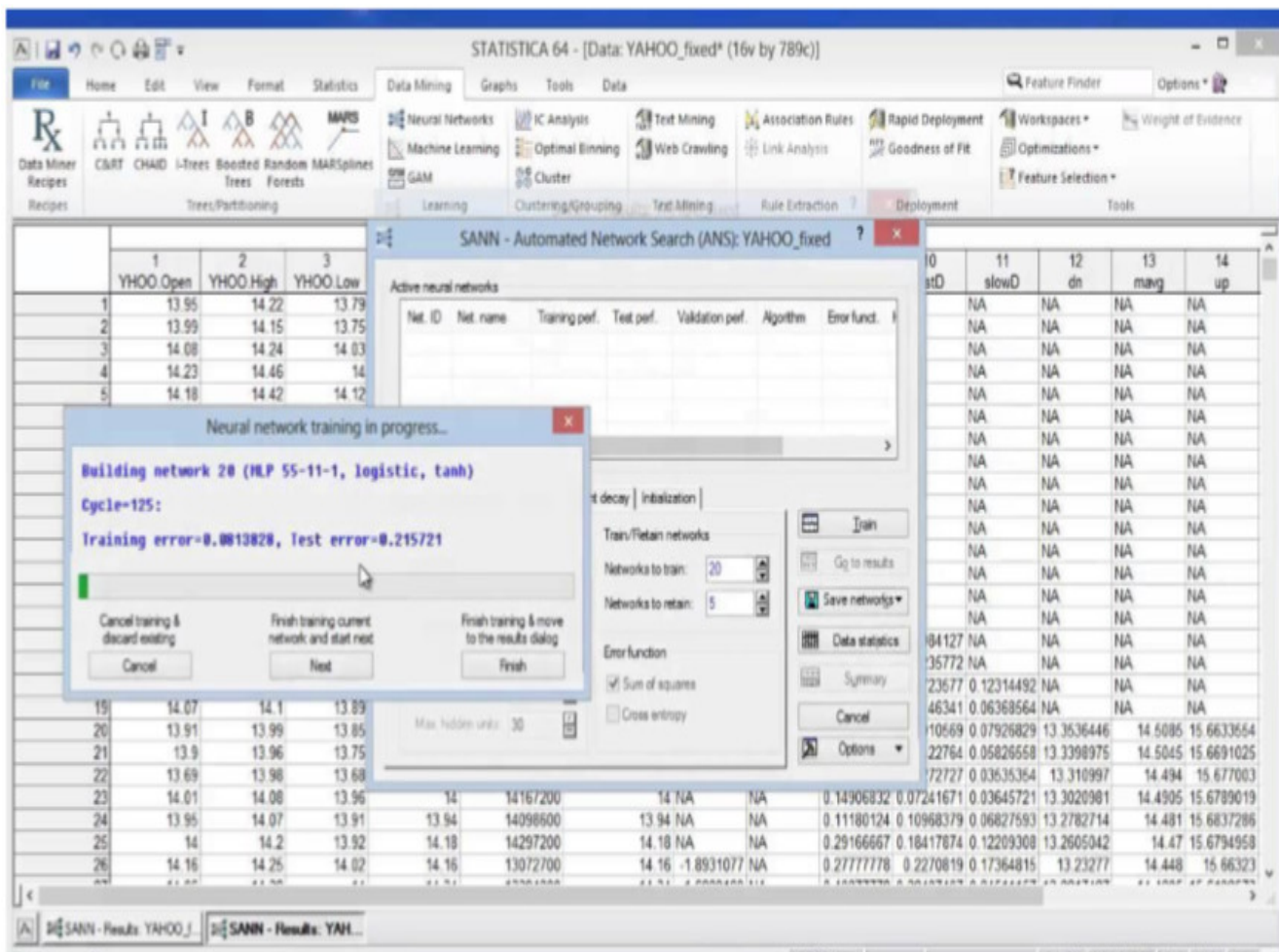


Рисунок 1.3 – Екранна форма інтерфейсу пакету прикладних програм STATISTICA

Він може бути використаний як програма на С, вбудована в призначені для користувача програми. SNNS може розширюватися за допомогою визначених користувачем функцій активації, вихідних функцій, функцій сайту і процедур навчання, які записуються у вигляді простих програм на С і зв'язуються з емулятором ядра.

Нейропакет JavaNNS (Java Neural Network Simulator) [25], розроблений в університеті Тюбінгена (University of Tübingen) та є наступником SNNS. Пакет заснований на власному обчислювальному ядрі, володіє розробленим графічним інтерфейсом користувача на Java, який сумісний з нейропакетом SNNS і дозволяє збільшити платформу незалежність.

Програмний інструмент Emergent застосовується для моделювання і призначений в першу чергу для створення складних моделей мозку та імітації когнітивних процесів, але також може бути використаний для вирішення будь-якої задачі на основі теорії нейронних мереж. Emergent має модульну конструкцію, засновану на принципах об'єктно-орієнтованого програмування. Пакет функціонує на платформах Microsoft Windows, Darwin / Mac OS X та Linux.

Пакет прикладних програм Neural Lab надає візуальне середовище для розробки і тестування штучних нейронних мереж, інтегрується з Microsoft Visual Studio, з C ++, включає штучні нейронні мережі в користувальницькі додатки, науково-дослідне моделювання, інтерфейси кінцевих користувачів. Neural Lab виконує перевірку на наявність помилок наборів даних, перш ніж використовувати їх для навчання, економлячи при цьому значну кількість часу. Інтерфейс активації дозволяє тестувати ШНМ в режимі реального часу, що уможливорює порівняти візуально фактичний вихід нейронної мережі з бажаним [14].

Нейропакет NEURON є середовищем моделювання окремих нейронів і мереж нейронів [44]. NEURON дозволяє моделювати окремі нейрони через використання секцій, які автоматично поділяються програмою замість їх ручного створення користувачем. Основна мова сценаріїв, який використовується для взаємодії зовнішнього світу з NEURON свій власний, але інтерфейс Python також доступний за потреби. Дані в програму записуються в інтерактивному режимі або завантажуються з файлу. NEURON підтримує розпаралелювання по протоколу MPI через внутрішні багатопотокові процедури для використання на комп'ютерах з декількома ядрами.

Програмний засіб GENESIS (The GEneral NEural SIMulation System – загальна нейронна система моделювання) є середовищем моделювання для побудови реальних моделей нейробіологічних систем різних рівнів наближення, в тому числі субклітинних процесів, окремих нейронів, мережі нейронів і нейронних систем. Основною метою комп'ютерних реалізацій моделей GENESIS є відповідь на питання: «що відомо про анатомічну будову і фізіологічні

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

особливості нейронної системи?». GENESIS призначений для кількісної оцінки нейронної системи таким чином, щоб було легко зрозуміти фізичну структуру нейрона [20].

Нейропакет Braincel від Promised Land Technologies Inc представляє програмну надбудову для табличного середовища MS Excel. Він дозволяє реалізовувати ШНМ прямого поширення з одним-двома прихованими шарами. Braincel використовує графічні можливості табличного процесора MS Excel та алгоритм навчання зворотного поширення помилки. Він дозволяє обробляти числові та символічні дані. Braincel надає можливості по визначенню значущості входів, вибору ШНМ із кращою структурою і використання для прогнозу одночасно декількох ШНМ.

Нейропакет Excel Neural Package від НейрОК доповнює можливості MS Excel щодо роботи з нейромережевими алгоритмами. Пакет складається з 2-х компонентів: Winnet, призначений для пошуку і моделювання прихованих залежностей в даних, реалізує функції багат шарового персептрона, має зручний графічний інтерфейс, візуальні засоби контролю, навчання, прогнозування, тестування, зупинку навчання при досягненні різних критеріїв; Kohonen Map, призначений для аналізу та побудови систем, що само організуються на основі карт Кохонена, які вирішують задачі кластеризації, візуалізації [20].

Пакет прикладних програм Neural Planner від Steve Wolstenholme – оболонка моделювання ШНМ різної конфігурації в середовищі MS Windows, яка використовує 2 алгоритми навчання з учителем: On-Line Back Propagation, і Batch Back Propagation. Neural Planner дозволяє вирішувати задачі класифікації, обробки даних, створювати експертні системи. В Neural Planner зручно створювати складно топологічні (наприклад, рекурентні) ШНМ при обмеженій кількості алгоритмів навчання.

Вільно розповсюджуваний нейропакет NeuroPro призначений для класифікації, прогнозування, вилучення знань з даних за допомогою ШНМ, в середовищі MS Windows. Середовище дозволяє працювати з даними в форматах: Clipper, dBase, FoxBase, FoxPro, Paradox; розробляти шари ШНМ, одночасно

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

вирішувати ряд задач прогнозування. Навчання ШНМ проводиться із застосуванням одного із декількох методів оптимізації. Середовище дозволяє виконувати тестування ШНМ, обчислення значущості вхідних сигналів, випадкові зміни ваг синапсів [18].

Нейропакет Partek Discovery створений для нейромережевого аналізу даних і моделювання ШНМ. Компонент Partek Discovery забезпечує візуальне середовище і реалізацію алгоритмів числового аналізу кластерів даних, що корисно при стисканні багатовимірних даних для подальшої візуалізації, аналізу або моделювання. Компоненти Partek Discovery дозволяють проводити прогнозує моделювання визначення оптимального набору змінних, наприклад, використовує статистичні методи, ШНМ, генетичні алгоритми.

Програмний інструмент QwikNet32 призначений для роботи в середовищі MS Windows, реалізує багат шарову мережу прямого поширення (до 5 прихованих шарів, з набором із 6 модифікацій алгоритму зворотного поширення помилки). QwikNet32 компактний, зручний у використанні за рахунок можливостей: графіки, збереження навченої ШНМ на мові C, моделювання, навчання ШНМ, ручного масштабування.

Нейропакети THINKS і ThinksPro є комплексом інструментальних засобів для розробки нейромережевих систем. Компонент THINKS – нейромережеве середовище розробки додатків, інструмент навчання. За рахунок наявності опцій вибору нейромережевої архітектури і визначення блоку обробки, оператор може працювати з новими мережевими конфігураціями. ThinksPro – нейромережеве середовище розробки для динамічної візуалізації в різних форматах значень входів, станів, ваг, виходів нейромережі.

Пакет прикладних програм MATLAB із вбудованим Neural Network Toolbox має модульну, відкриту і розширену архітектуру, що забезпечує підтримку багатьох нейромережевих парадигм. Пакет включає багат шаровий перцептрон з методом зворотного поширення помилки, рекурентні мережі зі змагальними шарами, самоорганізовані карти, радіально-базисні штучні нейромережі. Середовище Neural Network Toolbox призначене для моделювання ШНМ.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

перцептронного типу, має зручний графічний інтерфейс, призначений розробки, дослідження та управління процесами в режимах навчання і функціонування ШНМ [28].

Аналіз інструментальних засобів показав, що кожен спеціалізований засіб унікальний для вирішення певного вузького кола задач на основі деякої архітектури нейронної мережі. Пакет прикладних програм MATLAB має великий набір вбудованих функцій і можливості з технічної підтримки, що уможливорює реалізацію будь-якої архітектури ШНМ та розв'язування на її основі широкого спектру задач, зокрема, прогнозування, аналізу та прийняття рішення.

1.4 Постановка задачі кваліфікаційної роботи

Розглянуто задачі управління технологічними процесами й описано класи задач, що потребують розв'язання при управлінні технологічними процесами, зокрема, одними із важливих є збір та аналіз релевантних даних та прийняття на їх основі рішень. Охарактеризовано переваги методів машинного навчання, і зокрема, штучних нейронних мереж при вирішенні таких. Описано існуючі архітектури штучних нейронних мереж, які використовуються в задачах управління технологічними процесами, проаналізовано недоліки та переваги розглянутих архітектур. Зазначено, що для забезпечення ефективного розв'язку задач управління доцільно застосувати штучні нейронні мережі на основі радіально-базисних функцій, оскільки вони мають просту архітектуру та за рахунок радіально-базисних функцій здатні навчатися на неоднорідних даних й при цьому забезпечувати прогноз.

Існуючі програмні засоби моделювання штучних нейронних мереж здебільшого зорієнтовані на розв'язування задач певного класу із застосуванням конкретної архітектури нейронної мережі. Пакет прикладних програм STATISTICA Automated Neural Networks є досить ефективним для проектування

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

штучних нейронних мереж персептронного типу, проте є досить складним у користуванні. Зазначено, що для моделювання та дослідження архітектур ШНМ, зокрема архітектури на основі радіально-базисних функцій доцільно застосувати програмне середовище MatLAB із вбудованими тулбоксами.

Таким чином, зважаючи на вищезазначене, метою роботи є розробка програмного модуля аналізу релевантних даних на основі апарату штучних нейронних мереж із радіально-базисними функціями. Для досягнення мети бакалаврської роботи потрібно виконати такі завдання:

- описати особливості архітектури та навчання ШНМ із РБФ;
- охарактеризувати існуючі методи та алгоритми навчання ШНМ РБФ;
- розробити алгоритм навчання ШНМ РБФ для задач аналізу даних на відповідність потребам користувача;
- розробити програмний модуль для реалізації алгоритму на основі пакету прикладних програм MatLAB;
- розробити архітектуру штучної нейронної мережі з радіально-базисними функціями для аналізу релевантних даних;
- дослідити ефективність розробленої архітектури штучної нейронної мережі на основі задачі аналізу даних на відповідність потребам користувача;
- охарактеризувати техніко-економічні показники розробки проекту.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

2 АЛГОРИТМ АНАЛІЗУ РЕЛЕВАНТНИХ ДАНИХ НА ОСНОВІ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

2.1 Архітектура штучних нейронних мереж із радіально-базисними функціями

Радіально-базисна нейронна мережа складається з лише одного прихованого шару і одного вихідного шару. У деяких джерелах вхідний шар мережі радіально-базисних функцій може як враховуватися, так і опускатися. Це пов'язано з тим, що вхідний шар такої мережі, по суті, є вектором. Іншими словами, вхідний шар можна розглядати як набір нейронів, кожен з яких є однією з координат такого вектора, але точно так само можна опускати таке припущення [3, 14, 17, 27, 30, 32, 33-42].

Загальну архітектуру радіально-базисної мережі представлено на рисунку 2.1 [3].

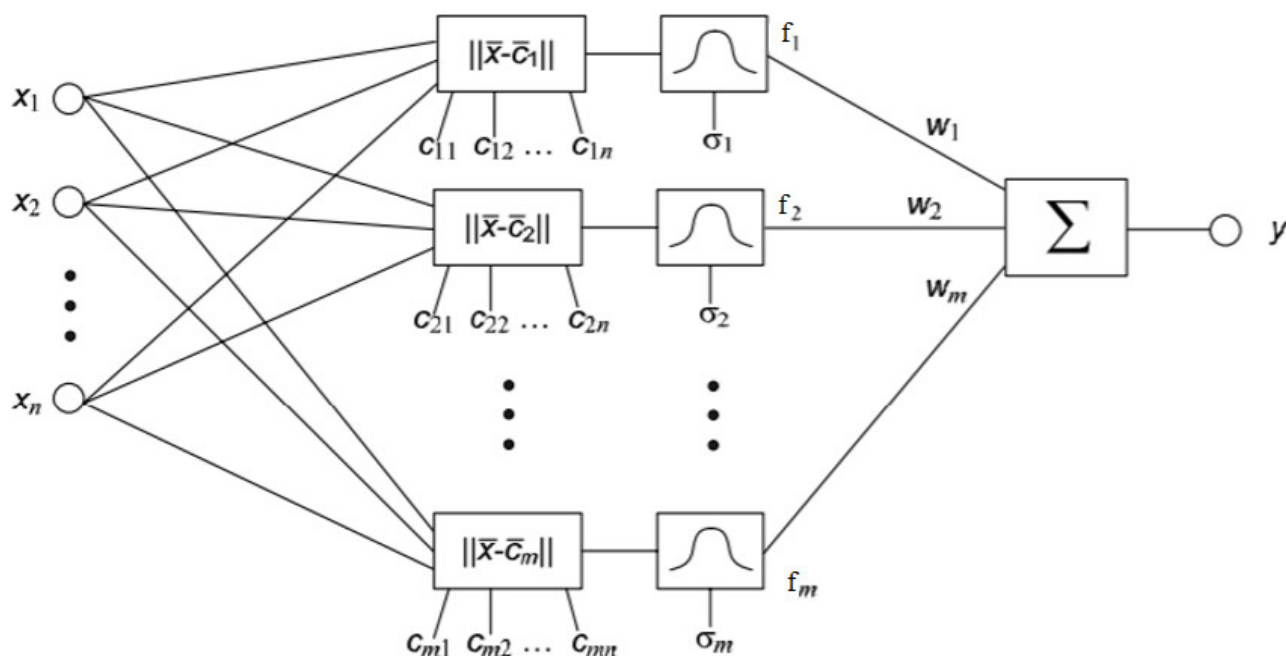


Рисунок 2.1. – Узагальнена архітектура радіально-базисної нейронної мережі

При чому, як бачимо, на рисунку використано такі позначення:

n – кількість елементів в першому шарі;

$\mathcal{X} = (x_1, x_2, \dots, x_b)$ – координати вхідного вектора;

b – кількість нейронів у другому шарі;

$\mathcal{C}_i = (c_{i1}, c_{i2}, \dots, c_{ib})$ – координати центру i -го елемента;

σ_{ij} – ширина радіальної функції i -го елемента;

f_i – вихідний сигнал i -го елемента;

w_i – ваговий коефіцієнт вихідного зв'язку i -го елемента;

w_0 – вага нейрона зміщення;

y – вихідний сигнал мережі.

Суть роботи радіально-базисної штучної нейронної мережі полягає в наступному. Вхідний вектор по черзі «показується» кожному з нейронів прихованого шару, в яких заздалегідь встановлені еталонні вектори. Далі вхідний вектор порівнюється з еталонним у вигляді [50]:

$$f_i(x) = f(\|x - c_b\|). \quad (2.1)$$

Для порівняння можна використовувати будь-яку метрику, наприклад, відстань Манхетана, але частіше використовують Евклідову відстань:

$$\|\mathcal{X} - \mathcal{C}_i\| = \left(\sum_{j=1}^N (x_j - c_{ij})^2 \right)^{\frac{1}{2}}. \quad (2.2)$$

Отриманий результат порівняння обробляється активаційною радіально-базисною функцією. Далі отриманий результат множиться на вагу синапсу нейрона і передається вихідному шару. Нейрони вихідного шару підсумовують отримані сигнали і видають відповідь. На цьому роботу мережі можна вважати закінченою.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

Вихідний сигнал кожного елемента визначається функцією Гауса. Однак, в залежності від розв'язуваних задач на основі штучної нейронної мережі радіального типу, функція Гауса може дещо змінюватися. При вирішенні завдання прогнозування функція Гауса буде виглядати наступним чином:

$$f_i(\mathbf{x}) = \exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{c}_i\|^2 \sigma^{-2}\right). \quad (2.3)$$

У разі вирішення задач прогнозування кількість нейронів у вихідному шарі має дорівнювати кількості класів й у кожного нейрона виходу буде своє значення ваги нейрона зміщення.

Вихідним сигналом радіально-базисної нейронної мережі буде зважена сума сигналів елементів:

$$\hat{y} = \sum_{i=0}^b w_i^{\text{out}} f_i(\|\mathbf{x} - \mathbf{c}_i\| \sigma_i). \quad (2.4)$$

Окрему увагу варто приділити радіальному нейрону. Нейрони радіального типу являють собою природне доповнення сигмоїдальних нейронів. Сигмоїдальний нейрон представляється в багатовимірному просторі гіперплощиною, яка розділяє цей простір на два класи, в яких виконується одна з двох умов.

Радіальний нейрон ж являє собою гіперсферу, яка здійснює кульовий поділ простору навколо центральної точки. У разі кругової симетрії даних, така структура дозволяє помітно зменшити кількість нейронів, необхідних для поділу різних класів. Схему нейрона прихованого шару радіально-базисної нейронної мережі наведено на рисунку 2.2 [30].

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

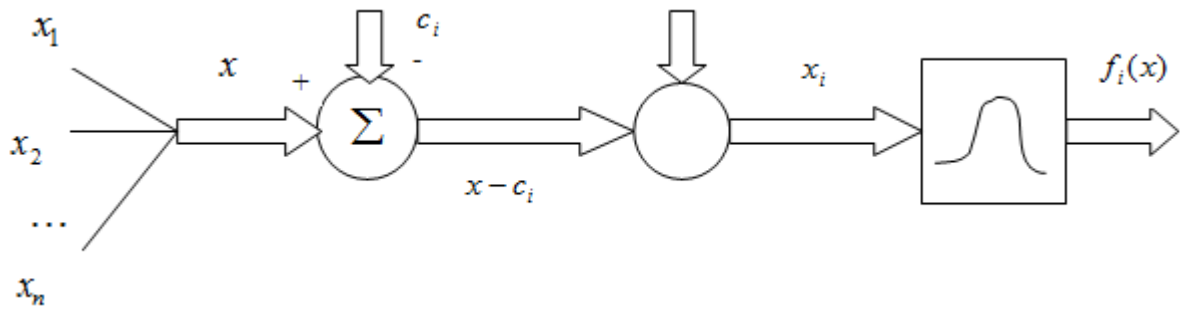


Рисунок 2.2 – Модель нейрона прихованого шару радіально-базисної нейронної мережі

Отже, радіально-базисні штучні нейронні мережі, як видно із аналізу, характеризують два типи параметрів:

- параметри прихованого шару – центри c_i та ширина базисної функції σ ;
- вагові коефіцієнти синаптичних зв'язків w_i^j , які є вихідними лінійними параметрами мережі.

Зважаючи на вищезазначене, у наступному підрозділі розглянемо детально існуючі алгоритми налаштування штучних нейронних мереж на основі радіально-базисних функцій.

2.2 Алгоритми налаштування параметрів радіально-базисних нейронних мереж

У випадках, коли є можливість оцінити ефективність роботи штучної нейронної мережі, навчання такої мережі можна уявити як задачу оптимізації. Одним із способів такої оптимізації є алгоритм зворотного поширення помилки [3, 30, 47, 50]. Навчання за допомогою такого алгоритму відбувається в два проходи по мережі – прямого і зворотного. При прямому проході вхідний вектор подається на кожен із нейронів мережі, після чого генерується набір вихідних сигналів, тобто – виходить реакція нейронної мережі на даний вхідний вектор.

Під час зворотного проходу фактичний результат нейронної мережі, отриманий під час прямого проходу, віднімається від бажаного і таким чином вираховується помилка нейронної мережі. Далі сигнал цієї помилки поширюється по мережі в напрямку зворотному напрямку синаптичних зв'язків і за певними правилами перераховуються параметри нейронної мережі.

В процесі оптимізації роботи штучної нейронної мережі за допомогою алгоритму зворотного поширення помилки можна модифікувати ваги нейронів прихованого шару, координати еталонних векторів кожного з нейронів прихованого шару і ширину радіальної функції σ . Окремим класом алгоритмів навчання радіально-базисних мереж є градієнтні алгоритми. Їх основу складає цільова функція, яка для одного навчального прикладу має вигляд [30, 49]:

$$E = \frac{1}{2} [w_i f_i x - d]^2, \quad i = 1, \dots, N. \quad (2.5)$$

Припустимо, що застосовують гаусівську радіальну функцію виду (2.3), де:

$$u_i(t) = \frac{x_j(t) - c_{ij}(t)^2}{\sigma_{ij}^2(t)}, \quad j = 1, \dots, N, \quad (2.6)$$

де i – індекс нейрона прихованого шару,

j – індекс компонента вхідного вектора,

t – індекс навчального прикладу у вибірці даних.

Існує два способи градієнтного спуску в реалізації алгоритму зворотного поширення помилки [2]:

- стохастична реалізація – в разі цього варіанту реалізації елементи, що оптимізуються змінюються відразу ж після розрахунку помилки, «на місці».
- пакетна реалізація – для цього варіанту реалізації функція втрат обчислюється по закінченню епохи (однократному проходу по навчальному

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

набору даних) для всіх елементів разом узятих, і тільки після цього вносяться поправки до елементів, які оптимізуються.

У кожної з цих реалізацій існують як переваги, так і недоліки. Для стохастичної реалізації характерна присутність внесених шумів в процесі навчання. Крім того, цей метод прийнято вважати більш повільним. Однак, стохастична реалізація має менше ускладнень за умов потрапляння в локальні мінімуми функції.

Пакетна реалізація більш стабільна, не вносить шумів і швидша, але при цьому має певні труднощі за умов зіткнення із локальними мінімумами. Окрім того, така реалізація вважається складнішою при впровадженні. Зважаючи на вищезазначене, для навчання радіально-базисних нейронних мереж варто застосувати стохастичну реалізацію алгоритму зворотного поширення помилки.

Навчання мережі на основі алгоритму зворотного поширення помилки проводиться в два етапи. На першому етапі представляється навчальний приклад, а потім розраховуються значення сигналів вихідних нейронів мережі і значення цільової функції, заданої виразом (2.5). На другому етапі мінімізується значення цієї функції. Підбір значень параметрів можна здійснювати, використовуючи градієнтні методи оптимізації незалежно від об'єкта навчання – чи це вага, чи центр.

Незалежно від обраного методу градієнтної оптимізації, необхідно, перш за все, отримати вектор градієнта цільової функції щодо всіх параметрів мережі. Очевидно, що для градієнтного спуску по функції, ця функція повинна бути диференційована. Продиференціювавши цільову функцію отримуємо [47]:

$$\frac{\partial E(t)}{\partial w_0(t)} = y(t) - d(t); \quad (2.7)$$

$$\frac{\partial E(t)}{\partial w_i(t)} = \exp\left(-\frac{1}{2} u_i(t) y(t) - d(t)\right); \quad (2.8)$$

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

$$\frac{\partial E(t)}{\partial c_{ij}(t)} = y(t) - d(t)w_i(t) \exp\left[-\frac{1}{2}u_i \frac{x_j(t) - c_{ij}(t)}{\sigma_{ij}^2(t)}\right]; \quad (2.9)$$

$$\frac{\partial E(t)}{\partial \sigma_{ij}(t)} = y(t) - d(t)w_i(t) \exp\left[-\frac{1}{2}u_i \frac{x_j(t) - c_{ij}(t)}{\sigma_{ij}^2(t)}\right]. \quad (2.10)$$

Формули (2.7) - (2.10) називати дельтами.

При використанні методу найшвидшого спуску формули для коригування параметрів радіально-базисної штучної нейронної мережі приймуть такий вигляд для ваг:

$$w_i(t+1) = w_i(t) - \eta \frac{\partial E(t)}{\partial w_i(t)}. \quad (2.11)$$

Для координат еталонного вектора, тобто для обчислення центрів радіально-базисних функцій:

$$c_{ij}(t+1) = c_{ij}(t) - \eta \frac{\partial E(t)}{\partial c_{ij}(t)}. \quad (2.12)$$

Для ширини рецепторного поля радіально-базисної функції:

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) - \eta \frac{\partial E(t)}{\partial \sigma_{ij}(t)} \quad (2.13)$$

Як можна бачити, формули (2.11)-(2.13) представляють різницю між поточним значенням оптимізаційного значення і дельтами, тобто формулами (2.8)-(2.10), помноженим на коефіцієнт навчання.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

Таким чином, визначившись із математичною складовою зазначеної нейронної мережі і її узагальненою структурою, варто приступати до розробки алгоритму синтезу архітектури радіально-базисної нейронної мережі для аналізу даних на відповідність потребам користувача.

2.3 Алгоритм синтезу архітектури радіально-базисної штучної нейронної мережі для аналізу релевантних даних

Як правило, навчання радіально-базисних штучних нейронних мереж відбувається у два етапи:

- налаштовуються параметри рецепторного поля радіально-базисних функцій;
- обчислюються параметри вихідного шару – вагові коефіцієнти.

Для налаштування центрів радіально-базисних функцій застосуємо метод «гірської» кластеризації, в основі якого алгоритм субтрактивної кластеризації, оскільки він простий у реалізації, а кількість кластерів регулюється лише на основі єдиного параметра – радіуса кластера [24, 45-48]. Слід зазначити, що обчислення центрів відбувається на основі лише вхідних даних.

На першому кроці алгоритму субтрактивної кластеризації визначаємо потенційні центри кластерів на основі рядків матриці X вхідних даних. Таким чином, розрахунок потенціалів визначених центрів кластерів дійсною за формулою [24]:

$$P(c_b) = \sum_{k=1}^N \exp(-\alpha \cdot \|c_b^p - x_k\|) = \sum_{k=1}^N \exp\left(-\alpha \cdot \sqrt{(c_{b1} - x_{k1})^2 + (c_{b2} - x_{k2})^2}\right), \quad (2.14)$$

де $c_b^p = (c_{b1}, c_{b2}, \dots, c_{bN})$ – потенційний центр b – го кластера;

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

a – додатна константа;

$\|\mathcal{E}_b^U - \mathcal{X}_k^U\|$ – відстань між потенційним центром b – го кластера \mathcal{E}_b^U та вектором вхідних експериментальних даних \mathcal{X}_k^U ;

N – кількість можливих кластерів, $h = \overline{1, N}$.

За центри сформованих кластерів обираємо координати "гірських" вершин. При цьому, центром першого кластера виступає точка із найбільшим потенціалом:

$$\mathcal{E}_1^U = \arg \max_{c_1, c_2, \dots, c_N} (P_1(\mathcal{E}_1^U), P_1(\mathcal{E}_2^U), \dots, P_1(\mathcal{E}_N^U)). \quad (2.15)$$

Зазвичай, найвища вершина оточена кількома високими точками. Враховуючи вищезначене, перед вибором наступного центру кластера потрібно знехтувати вплив щойно знайденого кластера. Для цього потенціал тих можливих центрів кластерів, що залишилися, перераховуємо за формулою:

$$\begin{aligned} P_2(\mathcal{E}_b^U) &= P_1(\mathcal{E}_b^U) - P_1(\mathcal{E}_1^U) \cdot \exp(-\beta \cdot \|\mathcal{E}_b^U - \mathcal{E}_1^U\|) = \\ &= P_1(\mathcal{E}_b^U) - P_1(\mathcal{E}_1^U) \cdot \exp\left(-\beta \cdot \sqrt{(c_{b1} - c_{11})^2 + (c_{b2} - c_{12})^2}\right) \end{aligned} \quad (2.16)$$

де $P_1(\mathcal{E}_b^U)$ – потенціал центру b -го кластера на першій ітерації;

$P_2(\mathcal{E}_b^U)$ – потенціал центру b -го кластера на другій ітерації;

\mathcal{E}_1^U – центр першого знайденого кластера;

β – додатна константа;

$\|\mathcal{E}_b^U - \mathcal{E}_1^U\|$ – відстань між потенційним центром b – го кластера та центром першого знайденого кластера \mathcal{E}_1^U .

Центр другого кластера визначаємо на основі максимального значення перерахованого потенціалу:

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

$$\mathcal{C}_2 = \arg \max_{c_1, c_2, \dots, c_N} (P_2(\mathcal{C}_1), P_2(\mathcal{C}_2), \dots, P_2(\mathcal{C}_N)). \quad (2.17)$$

Після цього знову відбувається перерахунок значень потенціалу для усіх можливих центрів кластерів, що оточують знайдену вершину:

$$\begin{aligned} P_3(\mathcal{C}_b) &= P_2(\mathcal{C}_b) - P_2(\mathcal{C}_2) \cdot \exp(-\beta \cdot \|\mathcal{C}_b - \mathcal{C}_2\|) = \\ &= P_2(\mathcal{C}_b) - P_2(\mathcal{C}_2) \cdot \exp\left(-\beta \cdot \sqrt{(c_{b1} - c_{21})^2 + (c_{b2} - c_{22})^2}\right). \end{aligned} \quad (2.18)$$

Процедура вибору центрів кластерів винується доти, поки не буде виключено усі рядки матриці вхідних даних X . Блок-схему алгоритму субтрактивної кластеризації наведено на КР.КІ.110418/15.00.00.000 А1

Параметр ширини базисної функції визначаємо експеименальним чином. При цьому він повинен бути не дуже малого розміру, щоб радіально-базисні функції накривали всю область вхідних даних та не надто великого, оскільки в цьому випадку значення радіально-базисних функцій будуть прямувати до 1.

Таким чином, а основі вищезазначеного можна сформувати алгоритм синтезу архітектури радіально-базисних нейронних мереж для аналізу релевантних даних. Блок-схему алгоритму зображено на рисунку 2.3. Для обчислення вагоих коефіцієнтів синаптичних зв'язків застосуємо алгоритм зворотного поширення помилки, що детально писаний у попередньому підрозділі. Коригування ваг відбувається на основі процер методів грдієнтного спуску, зокрема (2.11).

Слід зауважити, що такий алгоритм не вимагає складних очислювальних процедур, а обчислення параметрів не потребує суттєвих затрат часу. При розв'язуванні задач управління технологічними процесами застосування такого апарату при аналізі даних доцільно, оскільки це суттєво економить час при прийнятті відповідних рішень.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

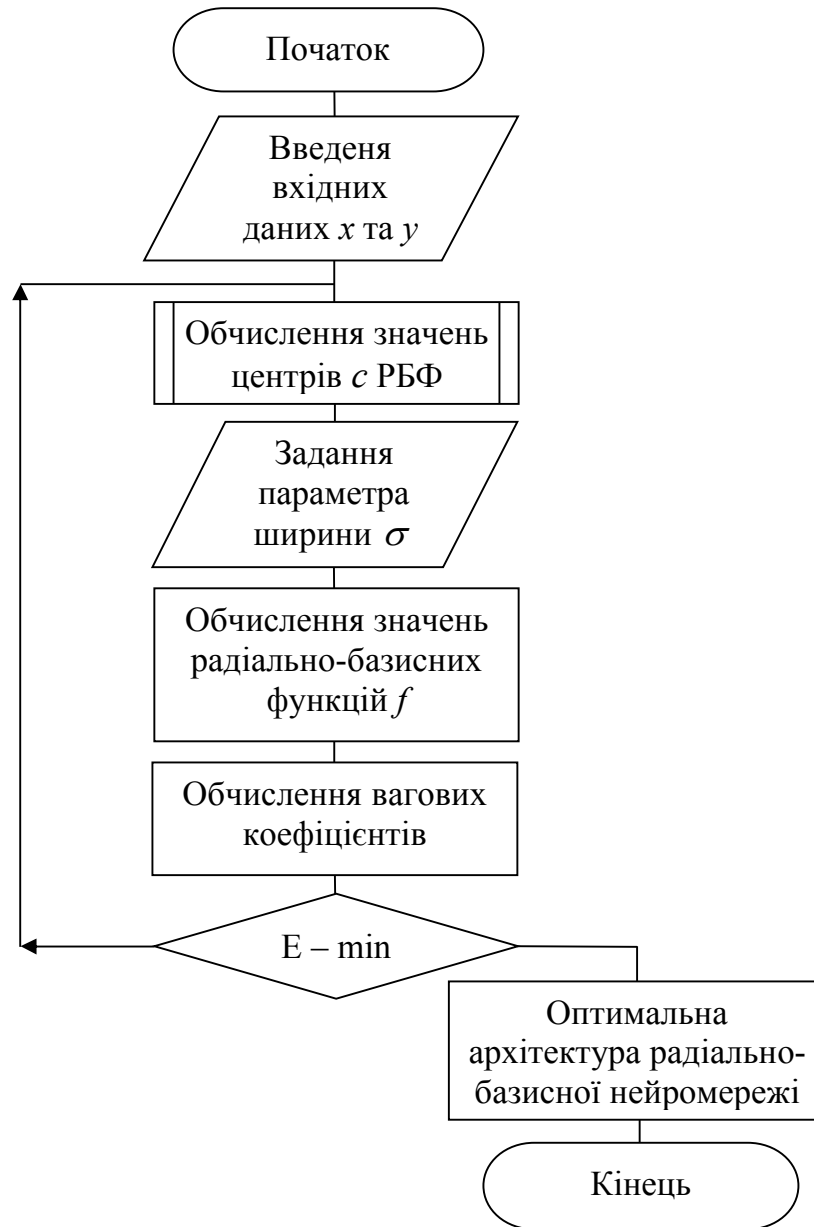


Рисунок 2.3 – Блок-схема алгоритму синтезу архітектури радіально-базисної нейронної мережі

У алгоритмі поєднано процедури налаштування центрів та задання параметра ширини базисної функції із процедурою обчислення вагових коефіцієнтів, яка реалізує алгоритм зворотного поширення помилки. Якщо значення функції похибки мінімальне, то отримуємо оптимальну архітектуру радіально-базисної нейронної мережі.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ СИНТЕЗУ РАДІАЛЬНО-БАЗИСНОЇ НЕЙРОННОЇ МЕРЕЖІ

3.1 Структура програмного модуля аналізу релевантних даних

Для проектування структури програмного модуля використаємо мову візуального моделювання UML. Мова UML (Unified Modeling Language) – уніфікована мова візуального моделювання, розроблена для візуалізації, створення специфікації, документування та конструювання бізнес-процесів і компонентів програмного забезпечення. UML можна застосовувати для побудови логічних і концептуальних моделей складних систем різного цільового призначення. [20]. В межах зазначеної мови всі уявлення про моделі складних систем представляються у вигляді спеціальних графічних конструкцій, які називаються діаграмами. У сукупності побудованих діаграм міститься вся інформація, необхідна для реалізації проекту складної системи.

Для реалізації розробленого алгоритму, який ґрунтується на алгоритмі субтративної кластерзації та мінімізації середньоквадратичної помилки навчання розроблено програмний модуль, діаграму варіантів використання якого проілюстровано на рисунку 3.1. У ролі адміністратора системи може виступати будь-який користувач із необмеженим доступом, що зуміє ввести вхідні та вихідні експериментальні дані й на їх основі побудувати архітектуру радіально-базисної нейронної мережі.

Функція “Побудова архітектури радіальної нейромережі” містить підфункції “Обчислення параметрів прихованого шару мережі”, “Обчислення значень радіально-базисних функцій”, “Налаштування вагових коефіцієнтів”, “Запис проміжних результатів у файл”, “Збереження архітектури радіально-базисної нейронної мережі у файл”.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

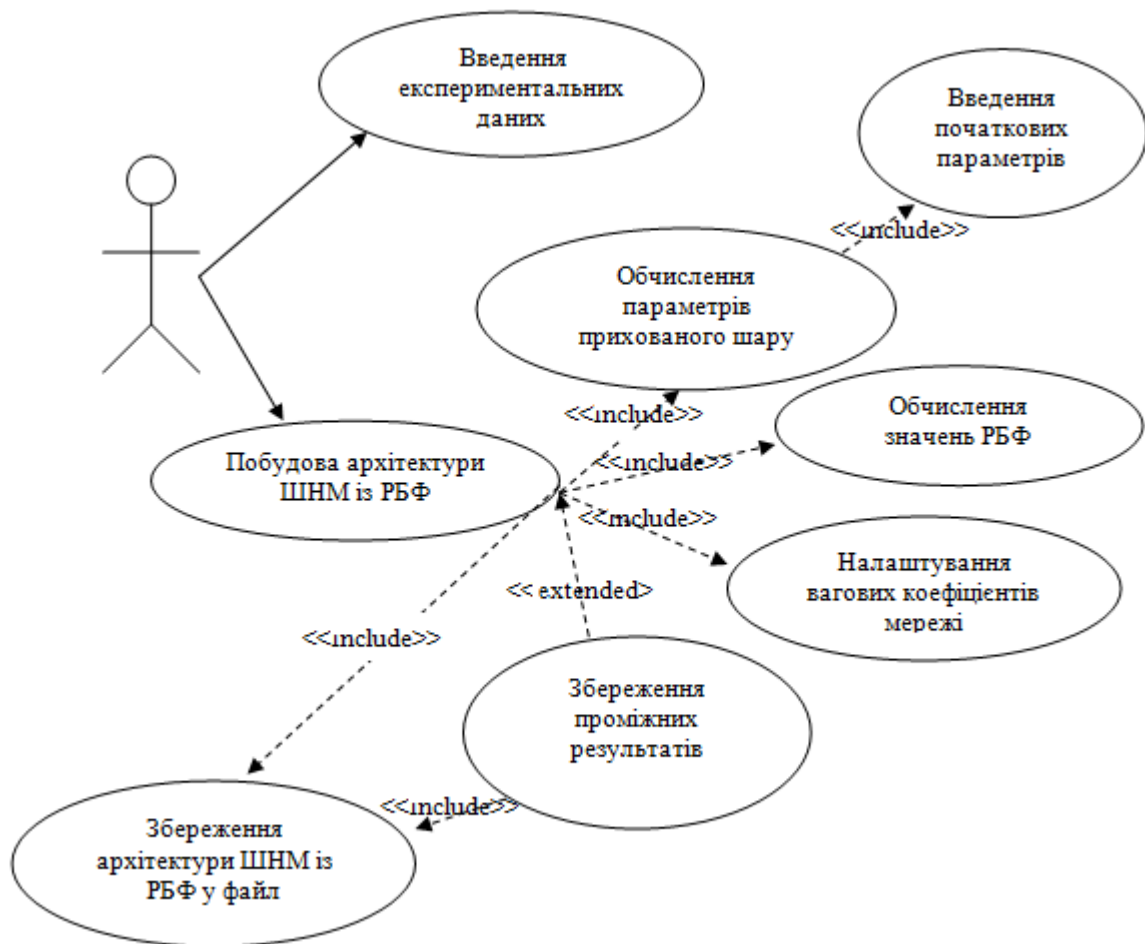


Рисунок 3.1 – Діаграма варіантів використання програмного модуля

Підфункція “Обчислення параметрів прихованого шару мережі”, реалізує алгоритм субтрактивної кластеризації. У результаті зазначеної підфункції отримуємо вектори центрів – \mathcal{C}_b , параметр ширини базисної функції σ_b та кількість радіально-базисних функцій – b .

Підфункція “Обчислення значень радіальних функцій” на основі вектора входів \mathcal{X}_n , $n=1, \dots, N$, вектора \mathcal{C}_b , $b=1, \dots, K$ та параметра впливу σ_b реалізує формулу обчислення функції Гауса f_b^v , $b=1, \dots, K$.

Підфункція “Налаштування вагових коефіцієнтів нейромережі” реалізує алгоритм зворотного поширення помилки. У результаті отримуємо вагові коефіцієнти синаптичних зв’язків w_b . Корекція значень вагових коефіцієнтів відбувається на основі механізму градієнтного спуску.

Підфункція “Запис проміжних результатів у файл” враховує можливість

зупинки програми у процесі реалізації функції розробки архітектури радіально-базисної нейромережі та запис отриманих на конкретний момент часу результатів у файл. При запуску програми забезпечується продовження процедури синтезу архітектури, починаючи із місця зупинки.

Підфункція “Запис архітектури нейромережі у файл” уможливорює збереження отриманої оптимальної архітектури мережі із побільшим її використанням для розв’язування задачі аналізу релевантних даних.

Програмний модуль автоматизації синтезу радіально-базисної нейронної мережі для аналізу отриманих даних на відповідність потребам користувача, розроблено із застосуванням структурно-функціонального підходу. Діаграму функціонування модулів представлено на рисунку 3.2.

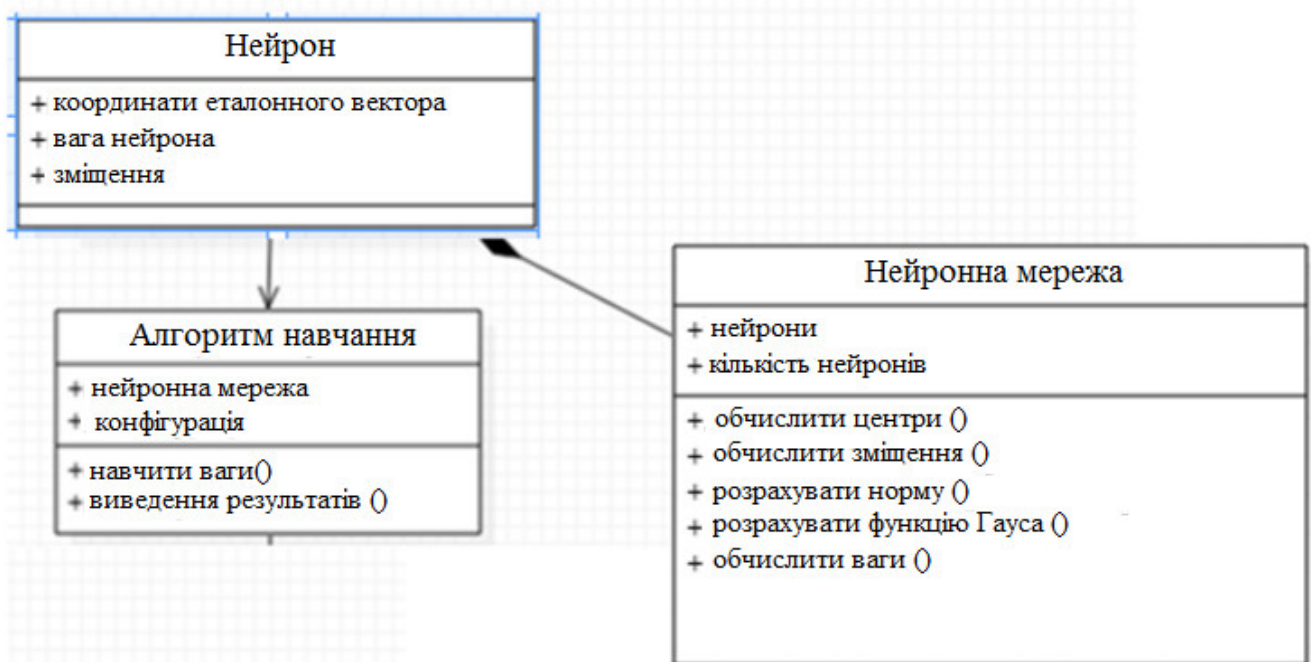


Рисунок 3.2 – Діаграма функціонування модулів програмного забезпечення

Як видно на рисунку 3.2, модульний принцип функціонування програми синтезу архітектури радіально-базисної штучної нейронної мережі включає почергову реалізацію модулів. Спочатку формуємо штучний радіально-базисний нейрон, тоді підключаємо алгоритм навчання нейронної мережі, а після цього моделюємо архітектуру нейромережі на основі радіально-базисних функцій.

3.2 Реалізація алгоритму аналізу релевантних даних

Для реалізації алгоритму аналізу релевантних даних на основі радіально-базисної нейромережі застосовано середовище пакету прикладних програм Matlab. Перевагами такого середовища моделювання є наявність вбудованих бібліотек – тулбоксів, які суттєво спрощують процедуру розробки моделей, а також економлять часові ресурси, що необхідні для розробки програмного коду, що реалізує певну функцію [18, 28].

Інтерфейс пакету прикладних програм Matlab досить простий, зручний та зрозумілий. Для виклику конкретної функції потрібно у командному рядку написати лише її назву. У результаті появляється зручний для користувача GUI – інтерфейс, який призначено для керування роботою алгоритму, а не його розробки. Екранну форму вікна інтерфейсу пакету прикладних програм Matlab зображено на рисунку 3.3.

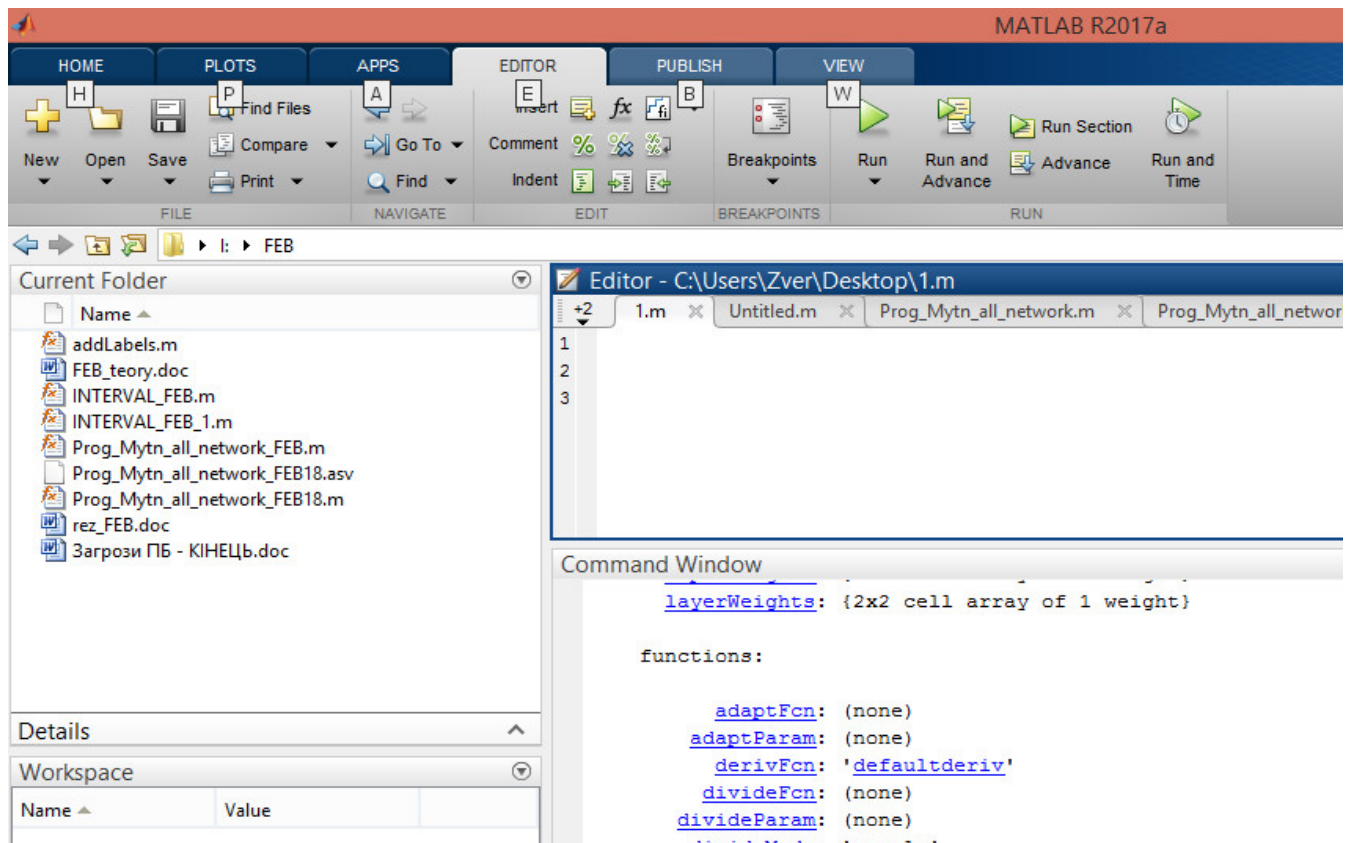


Рисунок 3.3 – Екранна форма інтерфейсу середовища Matlab

У загальному, реалізація алгоритму синтезу архітектури штучної нейронної мережі на основі радіально-базисних функцій відбувається таким чином. При запуску програми користувач вводить параметри конфігурації радіально-базисної штучної мережі, а також назву файлу, в якому знаходиться навчальна вибірка. Навчальна вибірка представляє собою набір рядків, кожна з яких, в свою чергу, являє собою набір розділених табуляціями числових значень. Крім перерахованого вище, користувач вводить кількість циклів навчання нейронної мережі і бажане максимальне значення середньоквадратичної помилки. Кількість циклів навчання необхідно в тому випадку, якщо середньоквадратична помилка нейронної мережі з якихось причин не сходиться до бажаного значення. В такому випадку нейронна мережа може навчатися необмежену кількість часу і назавжди «застрягти» всередині циклу навчання. Для того, щоб уникнути такої ситуації, потрібна якомога більша кількість циклів навчання – навіть якщо середньоквадратична помилка мережі не зійшлася до бажаної, то програма все одно завершить роботу.

Після цього відбувається ініціалізація радіально-базисної нейронної мережі, зокрема початкові нейрони прихованого шару, тобто, присвоюються випадкові значення сігм і ваг. Крім цього, кожному нейрону присвоюється випадковий вектор із навчальної вибірки.

Так само ініціалізується модуль BackPropagation, тобто ідентифікується об'єкт, який відповідає за навчання нейронної мережі. В процесі ініціалізації такого об'єкта завантажується файл навчальної вибірки. Після цього запускається безпосередньо алгоритм навчання штучної нейронної мережі, а саме алгоритм зворотного поширення помилки. Як було описано вище, у процесі цього алгоритму спочатку вираховується вихідне значення мережі на навчальній вибірці, отримане значення порівнюється з бажаним, після чого відбувається перенавчання мережі, а саме зміна значень ваг нейронів прихованого шару, параметрів ширини і координат еталонних векторів.

Для того, щоб знайти вихідне значення мережі, необхідно скористатися формулою (2.4). Іншими словами, необхідно розрахувати значення функції Гаусса

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

для кожного із нейронів, після чого помножити на значення ваги відповідного нейрона і підсумувати отримані від всіх нейронів результати.

Після проведення перерахунку ваг, координат еталонних векторів, та параметрів ширини розраховується новий вихідний сигнал нейронної мережі і середньоквадратична помилка. Якщо отримана середньоквадратична помилка менша за бажану, задану користувачем, навчання можна вважати закінченим. В іншому випадку навчання триває до тих пір, поки не виконається одне з двох умов: значення середньоквадратичної помилки менше заданого значення користувачем; не пройдено всі цикли навчання.

По завершенню роботи програма виводить значення середньоквадратичної помилки та графік навчання мережі, а також отримані у процесі навчання значення параметрів радіально-баисної нейронної мережі. До недоліків такої реалізації можна віднести випадковий вибір еталонних векторів, а не попередня кластеризація на основі алгоритму субтрактивної кластеризації, що описано у попередньому підрозділі.

Зважаючи на вищезазначене, опишемо реалізацію алгоритму аналізу даних на відповідність потребам користувача на основі пакету прикладних програм Matlab. На основі матриці вхідних даних реалізуємо алгоритм субтративної кластеризації із застосуванням функції subclust. Оскільки дуже часто дані бувають різної розмірності, то на початку роботи алгоритму їх доцільно пронормувати:

```
Xmin = min(X);
Xmax = max(X);
for i=1:size(X,1)
    X(i,:) = (X(i,:) - Xmin)./(Xmax-Xmin)
    fprintf('%0.3f\t',X(i,:))
end.
```

Кластеризацію даних проводимо на основі єдиного параметра – радіуса кластера. Найкращі результати кластеризації досягають, якщо параметр radii в межах [0,2;05]:

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

```

radii = [0.2];
xBounds = [];
options = [];
[C,S] = subclust(X,radii,xBounds);
clc
fprintf('Centry:\n')
prt(C)
fprintf('Sigma:\n');
prt(S).

```

У результаті отримуємо кількість кластерів із відповідними центрами та параметр ширини радіально-базисної функції – Sigma, що проілюстровано на рисунку 3.4.

```

Command Window
Centry:
0.147    0.418    0.352    0.319    0.026
0.899    0.087    0.085    0.912    0.434
0.374    0.267    0.922    0.427    0.781
0.337    0.922    0.711    0.065    0.123
Sigma:
0.177    0.177    0.177    0.177    0.177

```

Рисунок 3.4 – Екранна форма результатів реалізації функції subclust

Налаштування архітектури радіально-базисної нейромережі здійснюємо на основі функції Neural Network Toolbox newrb. Функція newrb використовує ітераційну процедуру, що полягає у додаванні по одному нейрону до прихованого шару на кожному кроці. Процедура додавання нейронів триває доти, поки сума квадратів похибок не стане меншою, ніж значення, задане користувачем або ж не буде використано максимальну кількість нейронів. Кількість нейронів прихованого шару відповідає кількості радіально-базисних функцій, яку обчислюють на попередньому етапі налаштування архітектури штучної нейронної мережі радального типу.

Для реалізації алгоритму синтезу архітектури штучної неронної мережі для аналізу релевантних даних застосовуємо функцію *net*, що реалізує алгоритм зворотного поширення помилки, та має такі аргументи:

- *net* – нейронної мережа, що буде налаштовуватися;
- параметри навчання нейронної мережі (вхідні та вихідні дані, початкові умови, часові значення).

Синтаксис функції *net* залежить від типу архітектури ШНМ. Для навчання радіально-базисної нейромережі записуємо:

$$\text{net} = \text{newrb} (P, T, \text{goal}, \text{spread}),$$

де *P* і *T* – масиви вхідних і вихідних даних,

goal – допустима середньоквадратична похибка навчання мережі,

spread – параметр ширини.

Допустиму середньоквадратичну похибку навчання та параметр ширини задаємо експериментально, залежно від поставленої задачі та її характеристик. У результаті реалізації зазначеної функції отримуємо детальний опис архітектури радіально-базисної нейронної мережі із зазначенням процедури формування шарів, вагових коефіцієнтів синаптичних зв'язків та зміщень. Екранну форму частини командного вікна, що відображає опис архітектури мережі зображено на рисунку 3.5.

Моделювання виходу мережі здійснюємо із застосуванням функції *sim*, яка має такі аргументи:

$$[Y, Pf, Af, E, perf] = \text{SIM} (\text{net}, P, Pi, Ai, T),$$

де *net* – назва штучної нейронної мережі;

P – входи;

Pi – початкові затримки входів;

Ai – початкові затримки шарів;

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

T – цільові значення;
 Y – вихідні сигнали;
 P_f – кінцеві вхідні затримки;
 A_f – кінцеві затримки шарів;
 $perf$ – параметр функціонування;
 E – похибка.

```

Command Window
NEWRB, neurons = 0, MSE = 2.59192e+08
NEWRB, neurons = 50, MSE = 3.70469e+07
HHHH

net_2 =

    Neural Network

        name: 'Radial Basis Network'
        userdata: (your custom info)

    dimensions:

        numInputs: 1
        numLayers: 2
        numOutputs: 1
        numInputDelays: 0
        numLayerDelays: 0
        numFeedbackDelays: 0
        numWeightElements: 1908
        sampleTime: 1

    connections:

        biasConnect: [1; 1]
  
```

Рисунок 3.5 – Екранна форма процесу синтезу архітектури радіально-базисної нейронної мережі

Слід зазначити, що аргументи P_i , A_i , P_f , A_f – не є обов'язковими. За умов, якщо ШНМ містить входи і шари із затримками їх потрібно зазначати:

$$Y = \text{sim}(\text{net_2}, \text{Input}(\text{vybirka},:))$$

Таким чином, на основі спроектованої діаграми варіантів використання програмного забезпечення описано процедуру реалізації архітектури штучної

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

нейронної мережі на основі радіально-базисних функцій із застосуванням програмних засобів пакету прикладних програм Matlab.

3.3 Аналіз даних на відповідність на основі радіально-базисних нейронних мереж

Важливим елементом конфігурації радіально-базисної штучної нейронної мережі є підбір так званих гіперпараметрів, які можна розділити на дві групи: глобальні та локальні (вузлові) [11, 18]. До глобальних гіперпараметрів відносять, кількість нейронів в кожному шарі, рівень навчання і момент, ініціалізація ваг нейронів. Локальні гіперпараметри – тип шару, функція активації та інші параметри регуляризації.

Оскільки більшість локальних гіперпараметрів, по суті, визначено обраною структурою нейронної мережі, то варто розглянути глобальні гіперпараметри, а саме: коефіцієнт навчання; момент навчання m ; кількість нейронів у прихованому шарі. Для того, щоб підібрати перераховані вище гіперпараметри, необхідно провести дослідження їх впливу на прогнозовану похибку радіально-базисної ШНМ.

Для того, щоб оцінити ефективність навчання використовують багато різних методів. Одним із найпоширеніших є RMSE – корінь середньоквадратичного відхилення, що є показником розсіювання значень випадкової величини щодо її математичного сподівання. Варто відзначити, що оцінка відхилення, отримана при розрахунку RMSE, є зміщеною, однак її все-вважають здійсненою [47].

Радіально-базисна нейронна мережа навчалася на вибірці такого вигляду: набір рядків із довільної кількості значень, розділених табуляцією. Останнє значення кожного рядка – очікуване значення. Експериментальні дані для навчання отримано із ТОВ «ТСК-Штамп», що є показниками різних лічильників,

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

які використовуються при контактному зварюванні. Експериментальні дані представлено на рисунку 3.6.

Файл	Правка	Формат	Вид	Справка								
61,83917	85,40581	100	100	32,53909	16,67908	0	0	0	5,68	4267		
100	74,67025	65,60821	65,60821	65,60821	65,60821	21,34831	10,94284	0				
83,82353	17,59962	83,82353	100	100	100	17,59962	0	5,71	4238			
87,5063	66,32242	100	66,32242	59,01763	59,01763	17,0529	8,740554	0	6,15	517		
62,17237	93,74262	100	68,66588	93,74262	62,17237	15,98583	8,193625	0				
93,18416	58,7963	93,18416	100	65,86934	51,33745	51,33745	8,487654	0	6,2			
76,26662	71,37981	100	80,9558	76,26662	42,13079	36,56126	30,72224	0	6,5			
100	84,51589	80,30303	80,30303	51,66297	46,84035	11,73319	6,005174	0				
94,50069	100	100	100	64,33502	58,3295	14,61114	7,478141	0	6,59	6347		
74,75682	94,61447	100	100	63,84342	58,03084	14,13998	7,236062	0	7,0			
88,9917	100	100	63,84342	20,71174	7,236062	0	7,04	6661				
94,85722	83,9904	100	83,9904	62,35785	36,40384	29,22669	4,334091	0	7,09	667		
100	100	100	75,18224	45,80848	36,945	5,301524	0	7,44	7014			
80,55303	97,20888	100	60,78151	50,59068	25,79514	21,64092	0	12,7353	7,3			
93,74789	100	100	100	75,74992	47,47219	13,83552	4,819683	0	7,43	701		
93,19464	100	82,19075	79,8471	58,92969	37,22271	11,27961	3,922797	0	7,9			
84,35646	100	88,48851	70,73039	49,77883	27,44633	21,11339	17,73654	0				
68,23958	100	76,36203	54,28224	32,82131	27,25163	6,774229	3,458946	0				
100	95,29426	95,29426	78,55338	39,55951	28,5955	6,908172	3,526297	0	8,3			
93,33677	100	100	63,57778	36,73154	30,6225	7,397861	3,77626	0	8,34	7534		
84,14986	100	88,18444	70,29483	49,32387	27,08934	20,98204	17,73443	0				
94,93372	100	85,75686	54,7734	30,53129	25,21837	22,39235	6,381667	0	8,6			
93,77787	100	86,8922	53,71478	29,79441	24,92114	5,863157	0	0	8,81	744		
83,87897	100	76,76816	17,28548	14,87067	14,87067	14,87067	14,87067	0				
89,31014	100	22,02759	17,78044	17,78044	17,78044	17,78044	17,78044	0				
93,96382	100	83,67997	60,5921	55,09112	13,08854	10,82582	10,82582	0	9,2			
100	95,51199	81,99821	50,9802	42,89979	25,92298	20,8976	5,761767	0	9,08	738		

Рисунок 3.6 – Експериментальні дані

Після завершення навчання реалізована програма виводить у командному вікні значення середньоквадратичної похибки а у файл – спрогнозовані значення, по одному значенню на рядок. У зв'язку зі стохастичністю кожного перенавчання радіально-базисної нейронної мережі, за результуюче значення RMSE обрано середнє за 20 етапів перенавчання. Результати виведення похибки навчання нейронної мережі та спрогнозованих значень даних при аналізі їх на відповідність зображено на рисунках 3.7-3.8.

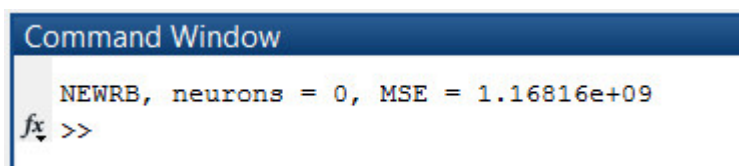
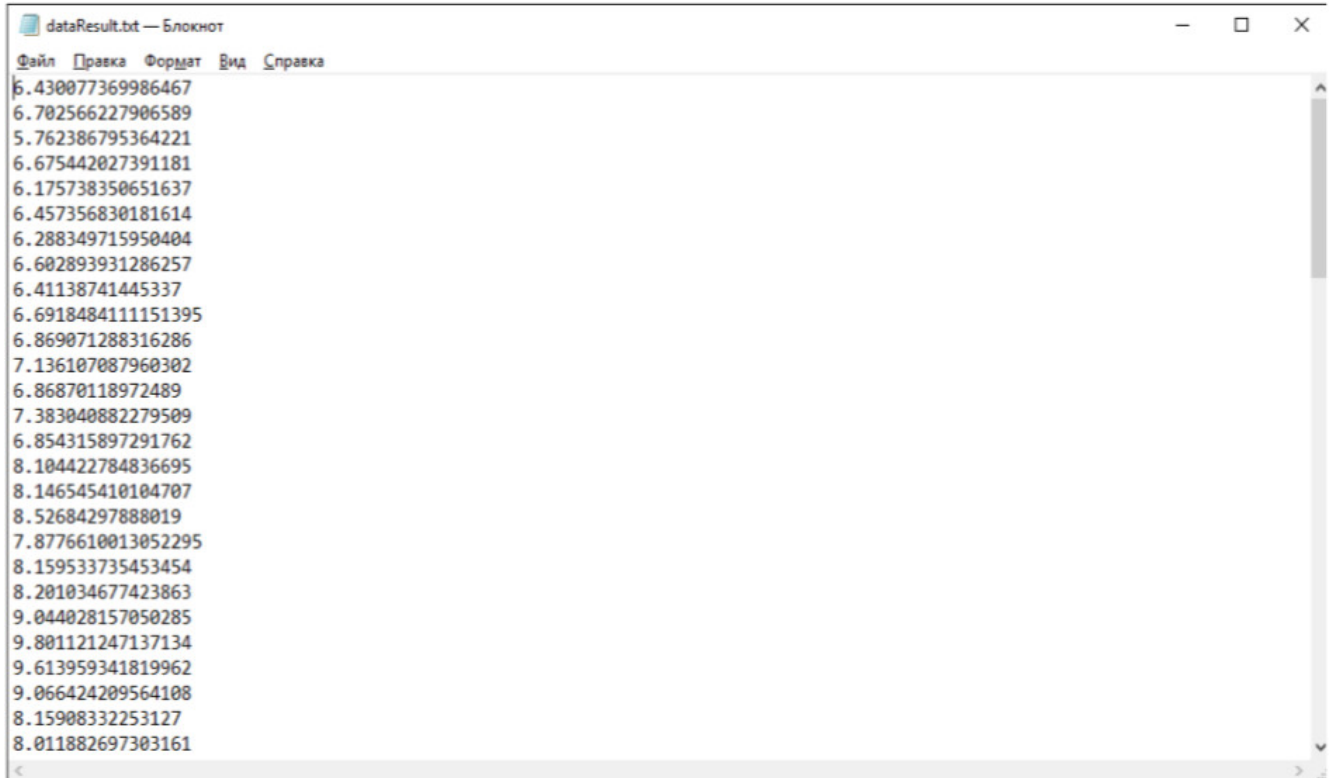


Рисунок 3.7 – Екранна форма командного вікна з похибкою навчання мережі



```
dataResult.txt — Блокнот
Файл Правка Формат Вид Справка
6.430077369986467
6.702566227906589
5.762386795364221
6.675442027391181
6.175738350651637
6.457356830181614
6.288349715950404
6.602893931286257
6.41138741445337
6.6918484111151395
6.869071288316286
7.136107087960302
6.86870118972489
7.383040882279509
6.854315897291762
8.104422784836695
8.146545410104707
8.52684297888019
7.8776610013052295
8.159533735453454
8.201034677423863
9.044028157050285
9.801121247137134
9.613959341819962
9.066424209564108
8.15908332253127
8.011882697303161
```

Рисунок 3.8 – Екранна форма файлу прогнозних значень даних

В процесі навчання нейронної мережі коефіцієнт навчання враховується при оновленні ваг нейрона, координат еталонного вектора – центрів і ширини радіальної функції під час роботи алгоритму зворотного поширення похибки із урахуванням моментів. Значення коефіцієнта навчання η емпірично вибирають із інтервалу (0; 1). Оновлення ваг відбувається за формулою:

$$\Delta w_{ij}(t+1) = -\eta \delta_i x_i + \alpha \Delta w_{ij}(t),$$

де $\delta_i x_i$ – i -та складова градієнта цільової функції,

$w_{ij}(t+1)$ і $w_{ij}(t)$ – ваги на наступному і попередніх етапах навчання відповідно.

Дослідимо залежність RMSE від коефіцієнта навчання. Параметри навчання радіально-базисної нейромережі задамо таким чином:

- тестова вибірка – 200 векторів;
- кількість ітерацій навчання: 15000;
- момент $m = 0.1$;

– кількість нейронів в прихованому шарі $b = 7$.

Результати дослідження представлено у таблиці 3.1.

Таблиця 3.1 – Залежність RMSE від коефіцієнта навчання

Коефіцієнт навчання	RMSE
0.005	0.7264
0.01	0.7036
0.02	0.7226
0.03	0.6822
0.04	0.7071
0.05	0.7231
0.1	0.7636
0.2	0.8364
0.3	0.8647
0.4	0.8897
0.6	1.0384
0.8	1.0752

Діаграму залежності RMSE від коефіцієнта навчання представлено на рисунку 3.9.

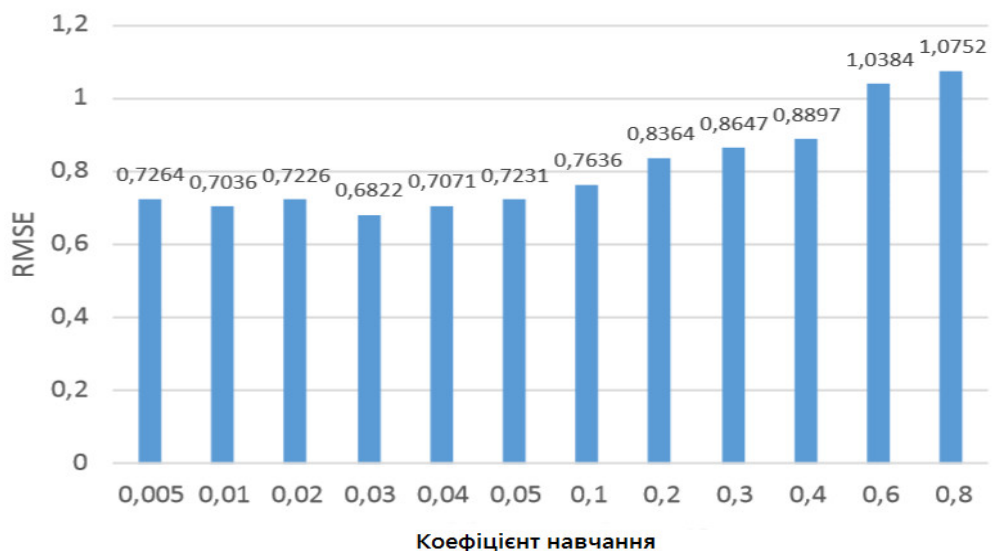


Рисунок 3.9 – Залежність похибки прогнозування від коефіцієнта навчання

Як бачимо на рисунку 3.9 на основі експериментальних даних середньоквадратична похибка прогнозування зростає із збільшенням коефіцієнта навчання, при цьому оптимальне значення коефіцієнта навчання $\eta = 0.03$.

При навчанні штучної нейронної мережі методом градієнтного спуску може виникнути ситуація, в якій навчання зупиняється, знаходячи локальний мінімум цільової функції. Для того, щоб уникнути такої ситуації, в формулу корекції ваг нейрона слід ввести додатковий параметр – момент m . Так само, як і у випадку із коефіцієнтом навчання момент підбираємо емпірично із діапазону (0; 1).

Дослідимо залежність середньоквадратичної похибки прогнозування радіально-базисної нейронної мережі від моменту m . Параметри навчання нейромережі задамо таким чином:

- тестова вибірка –200 векторів;
- кількість ітерацій навчання: 15000;
- коефіцієнт навчання = 0.03;
- кількість нейронів в прихованому шарі $b = 7$.

Результати дослідження представлено у таблиці 3.2.

Таблиця 3.2 – Залежність RMSE від моменту навчання нейромережі

Момент навчання	RMSE
0.005	0.6906
0.01	0.6847
0.05	0.6604
0.1	0.7636
0.3	0.6582
0.5	0.6142
0.7	0.6768
0.9	0.8872

Графік залежності середньоквадратичного відхилення прогнозного значення від моменту навчання мережі представлено на рисунку 3.10.

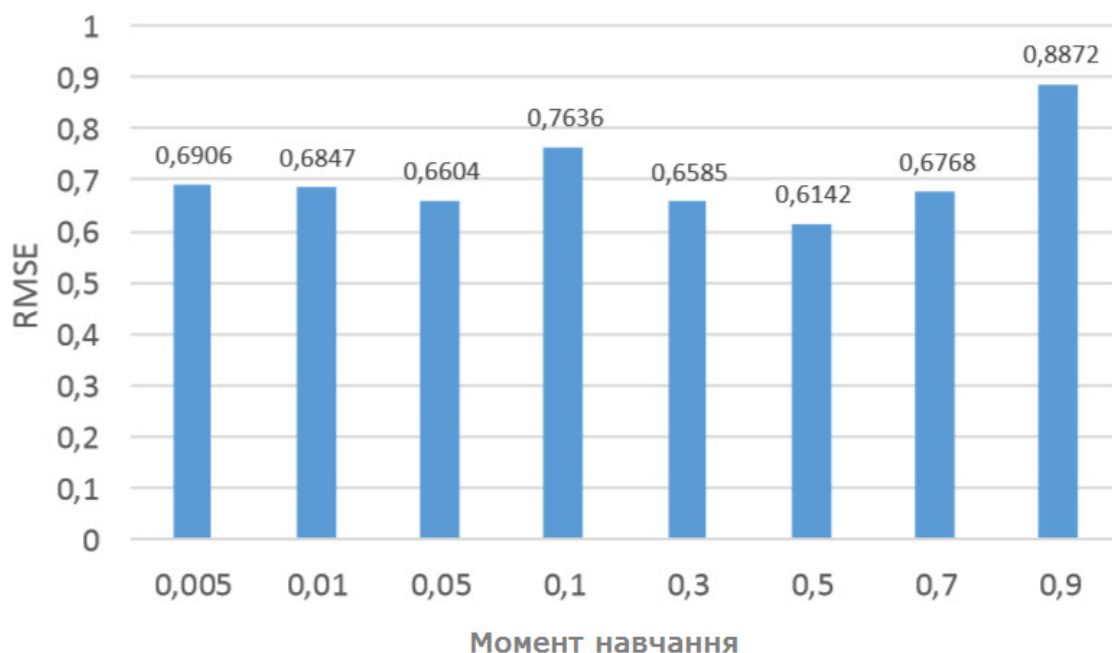


Рисунок 3.10 – Залежність похибки прогнозування від моменту навчання мережі

Кількість нейронів у прихованому шарі радіально-базисної нейронної мережі також є важливим гіперпараметром при налаштуванні конфігурації нейронної мережі. Так, при занадто малій кількості нейронів в прихованому шарі нейронна мережа може бути піддана великому розкиду прогнозованих даних, а при занадто великій кількості нейронів можуть виникати шуми.

Дослідимо залежність RMSE прогнозування від кількості нейронів у прихованому шарі при таких параметрах навчання радіально-базисної нейронної мережі:

- тестова вибірка – 200 векторів;
- кількість ітерацій – 15000;
- коефіцієнт навчання = 0.03;
- момент навчання $m = 0.5$.

Результати дослідження представлено в таблиці 3.3.

Таблиця 3.3 – Залежність RMSE від кількості нейронів у прихованому шарі

Кількість нейронів	RMSE
3	1.200
5	0.8633
7	0.8132
9	0.7636
11	0.9132
13	0.9399
15	0.9811
20	1.003
40	1.26

Діаграму залежності RMSE від кількості нейронів у прихованому шарі представлено на рисунку 3.11.

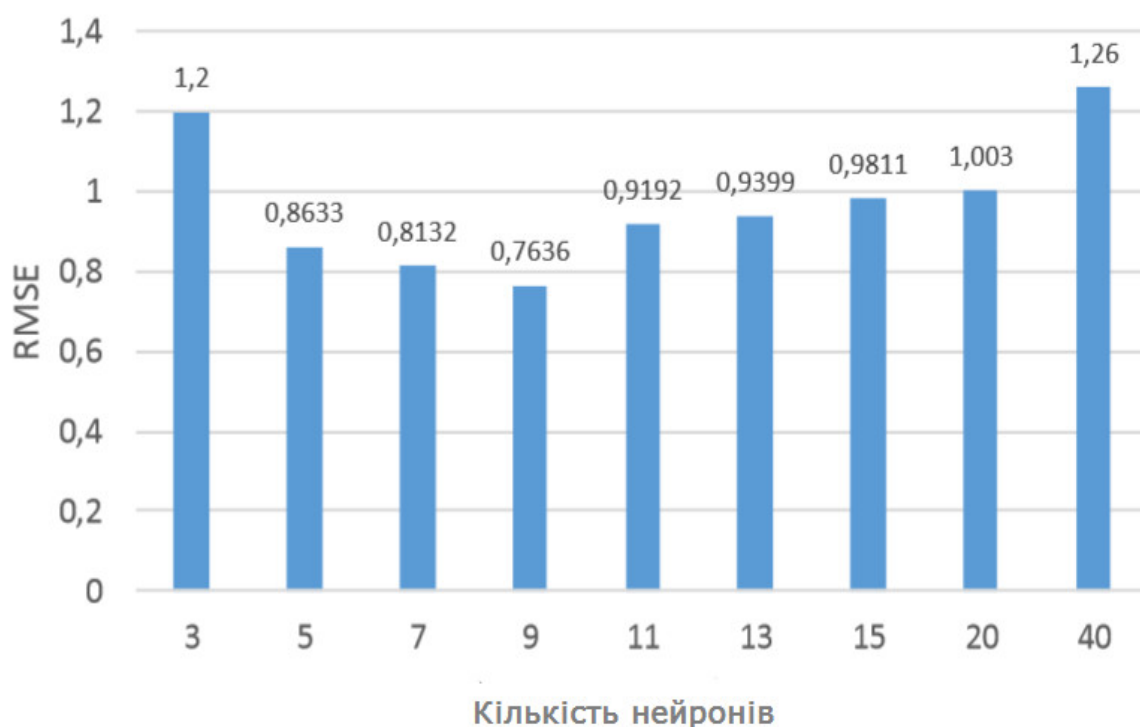


Рисунок 3.11 – Залежність RMSE від кількості нейронів у прихованому шарі нейромережі

Із рисунка 3.11 та результатів таблиці 3.3 випливає висновок, що нейронна мережа при занадто малій кількості нейронів у прихованому шарі допускає значні помилки прогнозування. При збільшенні кількості нейронів у прихованому шарі помилка знижується, однак, при суттєвому збільшенні кількості нейронів помилка знову починає зростати. Таким чином, оптимальною кількістю нейронів прихованого шару радіально-базисної штучної нейронної мережі для аналізу релевантних даних є $b = 9$.

Для розв'язування задачі аналізу даних на відповідність на основі експериментальних даних оптимальними значеннями гіперпараметрів для синтезу архітектури радіально-базисної нейронної мережі є:

- коефіцієнт навчання = 0.03;
- кількість нейронів в прихованому шарі $b = 7$;
- момент навчання $m = 0.5$.

Архітектуру розробленої радіально-базисної нейронної мережі представлено на КР.КІ.110418/15.00.00.000 С1. Зважаючи на експериментальні дані та проведені дослідження, радіально-базисна штучна нейронна мережа для аналізу раних на відповідність потребам користувача має 27 нейронів у вхідному шарі, 7 – у прихованому та 27 у вихідному шарі – 27:7:27.

Отже, із результатів експериментальних досліджень випливає, що розроблений алгоритм побудови архітектури штучної нейронної мережі радіального типу, що базується на алгоритмі субтрактивної кластеризації та зворотного поширення помилки, розроблене на його основі програмне забезпечення уможлиблюють аналіз даних на відповідність та їх прогнозування при розв'язуванні задач управління технологічними процесами.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ТЕХНІКО – ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ ПРОЕКТУ

4.1 Визначення витрат на оплату праці та відрахувань у соціальні фонди

Розроблене програмне забезпечення призначене для аналізу релевантних даних на основі ШНМ із РБФ. Алгоритм навчання ШНМ із РБФ, на якому базується розроблене програмне забезпечення не вимагає значних обчислювальних процедур та характеризується високою швидкістю реалізації, що уможливорює одержати модельоване значення відповідних даних із мінімальними затратами часу.

Витрати на розробку і впровадження програмних засобів (K) включають [5]:

$$K = K_1 + K_2, \quad (4.1)$$

де K_1 – витрати на розробку програмних засобів, грн.;

K_2 – витрати на відлагодження і дослідну експлуатацію програми вирішення задачі на комп'ютері, грн.

Витрати на розробку програмних засобів включають:

- витрати на оплату праці розробників;
- витрати на відрахування у спеціальні державні фонди;
- витрати на покупні вироби;
- витрати на придбання спецобладнання для проведення експериментальних робіт;
- накладні витрати;
- інші витрати.

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування програмного засобу. Перелік необхідної програмної документації визначено відповідно до ДСТУ 3008-95 та включає:

- текст програми;

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

- керівництво користувача, яке включає інструкцію користувача;
- опис програми – відомості про логічну і фізичну модель, відомості щодо функціонування програми;
- пояснювальна записка – схема алгоритму, загальний опис алгоритму або функціонування програми, а також обґрунтування прийнятих технічних і технічно-економічних рішень.

Перш за все визначаємо стадії розробки програмного засобу. У таблиці 4.1 відображено інформацію щодо етапів технологічного процесу розробки проекту

Таблиця 4.1 – Стадії розробки програмного засобу

№ п/п	Назва операції (стадії)	Виконавець, посада	Середній час виконання операції, год.
1	Підготовка, складання ТЗ	Менеджер проекту	3
2	Розробка макету системи	Team lead, Дизайнер (2)	2
3	Створення алгоритму системи	Архітектор (1)	3
4	Розробка системи	Програміст (6), дизайнер (2)	140
5	Тестування продукту	Тестувальник (3)	10
6	Попереднє представлення для замовника	Менеджер проекту	4
7	Представлення реалізованої системи	Менеджер проекту	1
Всього		14	163

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{оп} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij}, \quad (4.2)$$

де n_{ij} – чисельність розробників i -ої спеціальності j -го тарифного розряду, осіб;

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

t_{ij} – затрачений час на розробку проекту співробітником i -ої спеціальності j -го тарифного розряду, год.;

C_{ij} – годинна ставка працівника i -ої спеціальності j -го тарифного розряду, грн.

За умов, якщо середньогодинну ставку розробника не відомо, її можна розрахувати за формулою:

$$C_{ij} = \frac{C_{ij}^0(1+h)}{PЧ_i}, \quad (4.3)$$

де C_{ij}^0 – основна місячна заробітна плата розробника i -ої спеціальності j -го тарифного розряду, грн.;

h – коефіцієнт, що визначає розмір додаткової заробітної плати;

$PЧ_i$ – місячний фонд робочого часу працівника i -ої спеціальності j -го розряду, год.

Результати розрахунків записуємо у таблицю 4.2.

Таблиця 4.2 – Розрахунок витрат на оплату праці при розробці проекту розглянутого типу

№ п/п	Посада виконавця	Час розробки, год.	Погодинна заробітна плата, грн	Витрати на оплату праці, грн
1	Back-end розробник (2)	25	350	17 500
2	Front-end розробник (2)	25	300	15 000
4	Team lead(1)	15	400	6 000
5	Python розробник(1)	35	350	12 250
6	Менеджер(1)	10	450	4 500
7	Тестувальник(3)	10	250	2 500
8	Дизайнер(2)	20	300	12 000
9	Архітектор(1)	3	225	675
Всього		143		70425

Оскільки виконавцем кваліфікаційної роботи є студент, то він є і розробником і тестувальником і дизайнером. Таким чином, оплата його праці – це стипендія без додаткових нарахувань, надбавок та премій. Стипендія студента становить 1400 грн. Зважаючи на це, вартість проекту включає стипендію студента, а також витрати керівника на керівництво розробкою проекту та консультанта із написання техніко-економічного розділу. У таблицю 4.3 записуємо витрати на розробку програмного засобу у вигляді написання кваліфікаційної роботи.

Таблиця 4.3 – Розрахунок витрат на оплату праці

№ п/п	Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год (за весь рік)	Витрати на розробку, грн.
1	Керівник КР, старший викладач	8	409	3272
2	Консультант з техніко-економічного розділу, старший викладач	1	409	409
3	Студент	150	22	3300
Разом				6981

У таблиці 4.4 наведено витрати на матеріали та комплектуючі вироби при виконанні кваліфікаційної роботи.

Загальна сума витрат на матеріальні ресурси B_M визначається за формулою:

$$B_M = \sum_{i=1}^n K_i \cdot C_i, \quad (4.4)$$

де K_i - витрата i -го типу матеріалу, натуральні одиниці вимірювання;

C_i - ціна за одиницю i -го типу матеріалу, грн.;

i - тип матеріального ресурсу;

n - кількість типів матеріальних ресурсів.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.4 – Розрахунок витрат на матеріали та комплектуючі

№ п/п	Найменування купованих виробів	Од. виміру	Ціна, грн	К-сть купованих виробів	Сума, грн	Транспортні витрати	Заг. сума, грн
1	Папір (формат А4)	уп	100,0	2	200,00	20,0	220,0
2	Ручка кулькова	шт	10,0	1	10,00	1,0	11,0
3	Олівець простий	шт	8	2	16,00	1,6	17,6
4	Диски CD-R	шт	9	2	18,00	1,8	19,8
5	Зошит, 24 арк.	шт	10	1	10	1	11
6	Тонер для принтера	уп	80	1	80	8,0	88
Разом							367,40

Накладні витрати включають три групи витрат: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками від витрат на оплату праці. При цьому накладні витрати складають 70% від заробітної плати:

$$H = 70425 \cdot 0,75 = 49297,5 \text{ грн.}$$

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I = 6981 \cdot 0,1 = 698,1 \text{ грн.}$$

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{OP} + B_{\Phi} + B_{KB} + I. \quad (4.4)$$

$$K_1 = 6981 + 754,6 + 367,40 + 698,1 = 8801,1 \text{ грн.}$$

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою:

$$K_2 = S_{м.г.} \cdot t_{від}, \quad (4.5)$$

де $S_{м.г.}$ – вартість однієї машино-години роботи ПК, грн./год (приймаємо 6);

$t_{від}$ – час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 25 днів, середній щоденний час роботи на комп'ютері – 1 год., вартість години роботи комп'ютера дорівнює 6 грн. Звідси витрати на відлагодження та експлуатацію розраховуємо:

$$K_2 = 6 \cdot 25 = 150 \text{ грн.}$$

Оскільки розробка проекту включає застосування засобів обчислювальної техніки, то розрахуємо витрати на електроенергію, а результати розрахунків занесемо у таблицю 4.5.

Загальну суму витрат на електроенергію розраховуємо за формулою:

$$B_E = \sum_{i=1}^n P_i \cdot k_i \cdot T_i \cdot Ц, \quad (4.6)$$

де P_i – паспортна потужність i -го електрообладнання, кВт;

k_i – коефіцієнт використання потужності i -го електрообладнання (приймається 0.7...0.9);

T_i – час роботи i -го устаткування за весь період розробки, год;

$Ц$ – ціна електроенергії, грн / кВт*год;

i – тип електрообладнання;

n – кількість електрообладнання.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.5 – Витрати на електроенергію

Назва устаткування	Паспортна потужність, кВт	Коефіцієнт використання потужності	Час роботи обладнання, год	Ціна електроенергії, грн, кВт/год	Сума, грн
Комп'ютер	0.5	0.7	163	1,68	95,85

До амортизації основних фондів включається сума амортизаційних відрахувань від вартості обладнання і приладів, що використовуються при розробці програмного продукту. Амортизаційні відрахування розраховуємо за формулою:

$$B_{AM} = \sum_{i=1}^n \frac{B_i \cdot H_i \cdot T_i}{100 \cdot T_{E\Phi i}}, \quad (4.7)$$

де B_i – вартість i -го устаткування, грн.;

H_i – річна норма амортизації i -го устаткування, %;

T_i – час роботи i -го устаткування за весь період розробки, год.;

$T_{E\Phi i}$ – ефективний фонд часу роботи i -го устаткування за рік, год / рік;

i – тип устаткування;

n – кількість устаткування.

Усі розрахунки заносимо у таблицю 4.6.

Таблиця 4.6 – Амортизація основних фондів

Найменування устаткування	Вартість устаткування, грн	Річна норма амортизації, %	Ефективний фонд часу роботи обладнання, год / рік	Час роботи обладнання для розробки системи, год	Сума, грн.
Комп'ютер	20000	60	1700	163	1150,6
Разом амортизаційні відрахування					1150,6

На основі отриманих даних в результаті обчислень складаємо кошторис

									Арк.
									66
Змн.	Арк.	№ докум.	Підпис	Дата	КР.КІ.07171/17.00.00.000.ПЗ				

витрат на розробку програмного забезпечення і заносимо їх у таблицю 4.7.

Таблиця 4.7 – Кошторис витрат на розробку програмного забезпечення

№ п/п	Найменування витрат	Сума витрат, грн.
1	Витрати на оплату праці	6981
2	Витрати на електроенергію	95,85
3	Витрати на куповані вироби	367,40
4	Амортизаційні відрахування	1150,6
5	Інші витрати	698,1
6	Витрати на відлагодження і дослідну експлуатацію програмного продукту	150
Разом		9442,95

4.2 Розрахунок ціни проекту

Для оцінки економічної ефективності розробленого програмного продукту слід порівняти його з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення. Для цього визначимо експлуатаційні витрати на розробку проекту.

Експлуатаційні одноразові витрати по розробці програмного забезпечення і його аналогу включають вартість підготовки даних і вартість роботи комп'ютера (за час дії програми):

$$E_{\Pi} = E_{1\Pi} + E_{2\Pi}, \quad (4.8)$$

де E_{Π} – одноразові експлуатаційні витрати на ПЗ (аналог), грн.;

$E_{1\Pi}$ – вартість підготовки даних для експлуатації ПЗ (аналогу), грн.;

$E_{2\Pi}$ – вартість роботи комп'ютера для розробки програмного продукту (аналогу), грн.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

Річні експлуатаційні витрати B_{EP} визначаються за формулою:

$$B_{EP} = E_{EP} * N_{EP}, \quad (4.9)$$

де N_{EP} – періодичність експлуатації ПЗ (аналогу), раз/рік.

Вартість підготовки даних для роботи на комп'ютері визначається за формулою:

$$E_{1EP} = \sum_{l=1}^n n_i t_i c_i, \quad (4.8)$$

де i – категорії працівників, які приймають участь у підготовці даних ($i=1,2,\dots,n$);

n_i – кількість працівників i -ої категорії, осіб;

t_i – трудомісткість роботи співробітників i -ої категорії по підготовці даних, год.;

c_i – середньогодинна ставка працівника i -ої категорії з врахуванням додаткової заробітної плати, що знаходиться із співвідношення (4.3).

Трудомісткість підготовки даних для проектного рішення складає 4 год., для аналога 2 год.

Таблиця 4.7 – Розрахунок витрат на реалізацію програмного забезпечення

№	Час роботи співробітників, год.	Середньогодинна заробітна плата, грн./год.	Витрати, грн.
Проектне рішення			
1	4	8	32
Аналог			
1	2	32	64

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2\Pi} = t * S_{MG},$$

де t – витрати машинного часу для реалізації проектного рішення (аналогу), год.;

S_{MG} – вартість однієї години роботи комп'ютера, грн./год.

Зважаючи на вищенаписане, перевіримо розрахунки:

$$E_{2\Pi} = 4 \cdot 6 = 24 \text{ грн.}, E_{2\Pi_a} = 2 \cdot 6 = 12 \text{ грн.};$$

$$E_{\Pi} = 32 + 24 = 56 \text{ грн.}, E_{\Pi_a} = 64 + 12 = 78 \text{ грн.};$$

$$B_{E\Pi} = 56 \cdot 252 = 14112 \text{ грн.}, B_{E\Pi_a} = 78 \cdot 252 = 19656 \text{ грн.}$$

Ціна програмного продукту – це витрати на придбання і експлуатацію програмного засобу за весь період його служби:

$$C_{\Pi} = K \cdot \left(1 + \frac{P_p}{100}\right) + K_0 + K_k, \quad (4.9)$$

де K – кошторисна вартість;

P_p – рентабельність;

K_0 – витрати на встановлення та освоєння програмного засобу на конкретному об'єкті, грн.;

K_k – витрати на доукомплектування технічних засобів на об'єкті, грн.

Зважаючи на вищеописане, розрахуємо ціну програмного засобу

$$C_{\Pi_a} = 141719,525 \cdot (1 + 0,3) = 184235,4 \text{ грн.}$$

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

$$Ц_{П} = 9442,95 \cdot (1 + 0,3) = 12275,8 \text{ грн.}$$

У наступному підрозділі проведемо аналіз економічної ефективності розробки програмного продукту.

4.3 Визначення економічної ефективності розробки проекту

Для того, щоб побудувати таблицю показників економічної ефективності розробки програмного продукту, проведемо розрахунки необхідних показників.

Економічний ефект в сфері проектування проектного рішення розраховуємо за формулою:

$$E_{ПР} = Ц_{П} - Ц_{А}, \quad (4.10)$$

$$E_{ПР} = 184235,4 - 12275,8 = 171959,6 \text{ грн.}$$

Річний економічний ефект від експлуатації програмного продукту:

$$E_{КС} = B_{ЕА} - B_{ЕП}, \quad E_{КС} = 19656 - 14112 = 5544 \text{ грн.} \quad (4.11)$$

Дохід від розробки ПЗ у i -му періоді розраховуємо за формулою:

$$D_i = J_i(B_i - C_i), \quad (4.12)$$

де B_i – ціна продажу програмного продукту в i -му періоді;

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

C_i – собівартість програмного продукту (фактично дорівнює сумі витрат на розробку ПЗ);

J_i – кількість ПЗ.

$$D_i = 1 \cdot (12275,8 - 9442,95) = 2832,85 \text{ грн.}$$

Економічний ефект полягає у відношенні результату від розробленого програмного продукту до затрачених ресурсів та розраховується за формулою:

$$E = \frac{D_i}{B_{заг}}. \quad (4.13)$$

$$E = \frac{2832,85}{9442,95} = 0,3.$$

Тоді термін окупності обчислюємо за такою формулою:

$$T = \frac{1}{E} \quad (4.14)$$

$$T = \frac{1}{0,3} = 3,3 \text{ р.}$$

Зважаючи на проведені розрахунки ефективності розробки програмного продукту, обчислимо сумарний ефект від розробки програмного продукту за формулою:

$$E = E_{IP} + E_{KC}$$

$$E = 171959,6 + 5544 = 177503,6 \text{ грн.}$$

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.8 – Показники економічної ефективності проектного рішення

№	Найменування	Значення показників	
		Аналог	Новий варіант
1	Капітальні вкладення	70425	9442,95
2	Ціна придбання	184235,4	12275,8
3	Економічний ефект в сфері проектування	-	171959,6
4	Економічний ефект в сфері експлуатації	-	5544
5	Дохід від розробки		2832,85
6	Сумарний ефект	177503,6	
7	Термін окупності проекту	3,3	

Отже, у цьому розділі проведено розрахунок витрат на розробку програмного забезпечення. Показники, що характеризують витрати на розробку програмного продукту порівняно із показниками, які характеризують програмний продукт із аналогічним функціональним призначенням.

Розроблене програмне забезпечення має суттєві переваги у порівнянні із аналогами, зокрема простота використання, швидкість проведення розрахунків, стійкість до неоднорідних даних, зручність.

Згідно із проведеними розрахунками, що обґрунтовують економічну ефективність розробки програмного продукту, можна зробити висновок, що розроблене програмне забезпечення є суттєво дешевшим, оскільки у ролі розробника виступає студент. Отримано економічний ефект у розмірі 177503,6 грн., що свідчить про економічну доцільність розробки і впровадження програмного забезпечення для аналізу релевантних даних на основі радіально-базисних нейронних мереж.

ВИСНОВКИ

У кваліфікаційній роботі розв'язано практичну задачу аналізу релевантних даних та їх прогнозування у задачах управління технологічними процесами на основі розробленого програмного засобу, що базується на радіально-базисних штучних нейронних мережах. При цьому отримано такі практичні результати.

1. Проведено аналітичний огляд найпопулярніших архітектур штучних нейронних мереж й зазначено, що здебільшого у їх основі багатoshаровий перцептрон. Відзначено переваги радіально-базисних штучних нейронних мереж для задач аналізу та прогнозування, зокрема, можливість їх навчатися на неоднорідній вибірці даних, висока швидкість навчання.

2. Описано етапи розв'язування задач управління технологічними процесами й зазначено, що одним із важливих процедур є збір та аналіз релевантних даних та прийняття рішення на їх основі. Таким чином, правильний аналіз даних впливає на ефективні рішення щодо задач, що потребує застосування відповідних програмних інструментів.

3. Досліджено особливості існуючих програмних засобів реалізації штучних нейронних мереж й зазначено, що вони в основному спрямовані на вирішення задач певного класу, потребують суттєвих часових витрат для реалізації архітектур нейронних мереж, часто мають складний користувацький інтерфейс, що вимагає досвіду дослідника. Виокремлено особливості застосування пакету прикладних програм Matlab для моделювання штучних нейронних мереж, зокрема, наявність вбудованих тулбоксів із готовими реалізованими функціями та алгоритмами.

4. У результаті аналізу гіперпараметрів штучних нейронних мереж радіального типу та їх впливу на похибку прогнозування мережі, розроблено архітектуру радіально-базисної ШНМ для аналізу та прогнозування релевантних даних.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

5. На основі проведених експериментів досліджено ефективність розробленої архітектури нейронної мережі при аналізі даних на відповідність потребам користувача та їх прогнозування із мінімальною середньоквадратичною похибкою.

6. Обґрунтовано техніко-економічні показники ефективності розробки програмного модуля.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Автоматизована система управління підприємством (АСУП): веб-сайт. URL: <http://asutp.ru/?p=600157> (дата звернення: 17.04.2021).
2. Автоматизована система управління: веб-сайт. URL: https://leksika.com.ua/16010716/ure/avtomatizovana_sistema_upravlinnya (дата звернення: 15.04.2021).
3. Бодянский Е. В., О. Г. Руденко Искусственные нейронные сети: архитектуры, обучение, применения: Монография. Харьков: ТЕЛЕТЕХ, 2004. 372 с.
4. Галушкин А.И., Иошуа Б. Нейронные сети: история развития теории: учебное пособие. М.: Альянс, 2015. 840с.
5. Искусственный интеллект: принятие решения в сложных системах управления: веб-сайт. URL: <http://asutp.ru/?p=600157> (дата звернення: 17.04.2021).
6. Кохонен, Т. Самоорганизующаяся карты. М.: Бином. Лаборатория знания, 2017. 656с.
7. Люггер Джордж Ф. Искусственный интеллект: стратегии и методы решения сложных проблем М. : Издательский дом «Вильямс», 2004. 864 с.
8. Методичні вказівки до виконання практичних робіт з дисципліни «Техніко-економічне обґрунтування розробки комп'ютерних систем»/ Н.Я. Савка, І.Р. Паздрій / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 40 с.
9. Методичні вказівки до оформлення курсових, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія» / І.В. Гураль, Л.О.Дубчак / під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 34 с.
10. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня «Бакалавр» спеціальності 123 «Комп'ютерна інженерія» галузі знань 12 Інформаційні технології / О.М. Березький, Л.О.Дубчак, Г.М. Мельник, Ю.М. Батько / Під ред. О.М. Березького. Тернопіль: ЗУНУ, 2020. 60с.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		

11. Осовский С., Нейронные сети для обработки информации М.: Финансы и статистика, 2002. 344 с.
12. Пекельна В.В., Богатюк В.В., Боднарчук С.С., Рудик О.А. Метод синтеза радиальных штучних нейронних мереж для аналізу релевантних даних //
13. Применение нейронных сетей для задач классификации: веб-сайт. URL: <https://basegroup.ru/> (дата звернення 17. 05. 2021).
14. Савка Н.Я. Етапи ідентифікації штучних нейронних мереж з радіально-базисними функціями : зб. матеріалів Всеукраїнської школи-семінару молодих вчених і студентів “Сучасні комп’ютерні інформаційні технології”. Тернопіль: ТНЕУ, 2011. С. 87.
15. Савка, Н.Я., Козак О.Л., Струбицька І.П. Структурно-параметрична ідентифікація інтервальних моделей з радіально- базисними функціями // Proceedings of the international scientific and practical conference "The Top Actual Researches in Modern Science". - Dubai: Roast Publishing, 2015. Vol I. - P. 17-22.
16. Савка Н.Я. Методи ідентифікації штучних нейронних мереж з радіально-базисними функціями: зб. матеріалів першої Міжнародної науково-технічної конф. “Обчислювальний інтелект 2011”. Черкаси, 2011. С. 120-121.
17. Савка Н.Я., Спільчук В.М, Співак І.Я Проблеми ідентифікації штучних нейронних мереж з радіально-базисними функціями та можливі напрямки їх розв’язання. Індуктивне моделювання складних систем: зб. наук. праць. 2012. Вип. 2. С. 181-193.
18. Сирота А. А. Методы и алгоритмы анализа данных и их моделирование в MATLAB. Учебное пособие. СПб.: БХВ- Петербург, 2016. 384с.
19. Рассел С. Искусственный интеллект. Современный подход. М.: Вильямс, 2018. 1408 с.
20. Рашид Т. Создаем Нейронную сеть. М.: Вильямс, 2017. 272 с.
21. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. Учебник. М.:ДМК-Пресс, 2015. 400 с.
22. Хайкин, С. Нейронные сети. Полный курс. М.: Вильямс, 2016. 1104 с.

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

23. Шишов О.В. Элементы систем автоматизации: предприятие как целостный объект автоматизации: пособие. М.: Directmedia, 2015. 41с.
24. Штовба С. Д. Введение в теорию нечетких множеств и нечеткую логику: веб-сайт. URL: <http://matlab.exponenta.ru/fuzzylogic/book1/index.php> (25.01.2021).
25. Ян Г., Иошуа Б., Аарон К. Глубокое обучение. М.: ДМК- Пресс, 2018. 652 с.
26. Dyvak M., Maslyiak Y., Papa O., Savka N. Clustering and interval analysis of heterogeneous data sample // The Proc. of 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). Conf. Proc., 2017, с. 528-532.
27. Hamad A. Dingli Yu, J. B. Gomm, Mahavir S. Sangha Radial basis function neural network in fault detection of automotive engines. International Journal of Engineering, Science and Technology. 2010. Vol. 2., No 10. P.1-8.
28. MATLAB&Toolboxes: веб-сайт. URL: <http://matlab.exponenta.ru> (дата звернення 20.02.2021).
29. Markopoulos, A.P., Georgiopoulos, S. and Manolakos, D.E. On the use of back propagation and radial basis function neural networks in surface roughness prediction // Journal of Industrial Engineering International. 2016. Vol. 12. С. 389–400. <https://doi.org/10.1007/s40092-016-0146-x>.
30. Nelles O. Nonlinear Systems Identification. Berlin: Springer, 2001. 785 p.
31. Que Q., Belkin M. Back to the future: Radial basis function networks revisited. In AISTATS, 2016. P/ 1375–1383.
32. Basis Function Networks and Complexity in Function Learning: веб-сайт. URL: <https://proceedings.neurips.cc/paper/1996/file/97af4fb322bb5c8973ade16764156bed-Paper.pdf> (дата звернення 19.01.2021).
33. Radial basis function: веб-сайт. URL: <https://www.ibm.com/docs/ko/spss-statistics/23.0.0?topic=option-radial-basis-function> (дата звернення 15.01.2021).

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

34. Radial Basis Functions Neural Networks – All We Need to Know: веб-сайт. URL: <https://towardsdatascience.com/radial-basis-functions-neural-networks-all-we-need-to-know-9a88cc053448> (дата звернення 15.01.2021).

35. Basis Function Networks: веб-сайт. URL: <https://www.mathworks.com/help/deeplearning/ug/radial-basis-neural-networks.html> (дата звернення 20.01.2021).

36. Basis Function Networks: веб-сайт. URL: <https://deeprai.org/machine-learning-glossary-and-terms/radial-basis-function-network> (дата звернення 25.04.2021).

37. Radial Basis Function Networks: Introduction: веб-сайт. URL: <https://www.cs.bham.ac.uk/~jxb/NN/I12.pdf> (дата звернення 25.04.2021).

38. Radial basis function networks for convolutional neural networks to learn similarity distance metric and improve interpretability: веб-сайт. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9133368> (дата звернення 20.01.2021).

39. Radial-basis function networks: веб-сайт. URL: https://link.springer.com/chapter/10.1007/978-94-015-9341-0_6 (дата звернення 16.02.2021).

40. Radial Basis Function Network (RBFN) Tutorial: веб-сайт. URL: <https://mccormickml.com/2013/08/15/radial-basis-function-network-rbfn-tutorial/> (дата звернення 20.01.2021).

41. Radial basis function networks for convolutional neural networks to learn similarity distance metric and improve interpretability: веб-сайт. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9133368> (дата звернення 20.01.2021).

42. RBF Neural Networks: веб-сайт. URL: <https://www.dtreg.com/solution/view/25> (дата звернення 25.01.2021).

43. Robust radial basis function neural networks: веб-сайт. URL: <https://pubmed.ncbi.nlm.nih.gov/18252348/> (дата звернення 24.04.2021).

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

44. Schmidhuber J. Deep learning in neural networks: An overview // Neural Networks. 2015. No 61. P.85-117.
45. Shtovba S. Fuzzy model tuning based on a training set with fuzzy model output values // Cybernetics and Systems Analysis. 2007. Vol. 43, №3. P. 334–340.
46. Shtovba S D., Galushchak A. V. Fuzzy classifier learning by main competitors method // Матеріали II Міжнародного форуму «IT-Trends». м. Кременчук, 2015. С. 56-58.
47. Shtovba Serhiy, Mashnitskiy Yakiv The Backpropagation Multilayer Feedforward Neural Network Based Competition Task Solution // Vinnitsa State Technical University. 2000. № 3. С. 18-21.
48. Shtovba S., Pankevich O., Nagorna A. Analyzing the criteria for fuzzy classifier learning // Automatic Control and Computer Sciences. 2015. Vol. 49, №3. P. 123–132.
49. Schmidhuber, J., Deep learning in neural networks: An overview// Neural Networks. 2015. №61. С. 58-69.
50. Using neural networks for regression: radial basis function networks: веб-сайт. URL: <https://pythonmachinelearning.pro/using-neural-networks-for-regression-radial-basis-function-networks/> (дата звернення 24.04.2021).

					КР.КІ.07171/17.00.00.000.ПЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		