

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

СКЛАДАНЮК Владислав Миколайович

**«Алгоритми синтезу радіально-базисних нейронних
мереж / Synthesis Algorithms of radial-based neural
networks»**

спеціальність: 123 - Комп'ютерна інженерія
освітньо-професійна програма - Комп'ютерна інженерія
Кваліфікаційна робота

Виконав студент групи КІм-22
В.М. Складанюк

Науковий керівник:
к.т.н. Н.Я. Савка

Кваліфікаційну роботу допущено
до захисту:

" ___ " _____ 20__ р.

Завідувач кафедри
_____ О.Л. Дубчак

Тернопіль – 2022

ЗМІСТ

Вступ.....	7
1 Аналіз штучних нейронних мереж.....	11
1.1 Характеристика нейромереж.....	11
1.2 Аналіз архітектур нейромереж.....	14
1.3 Аналіз програмних засобів синтезу нейромереж.....	19
1.4 Постановка задачі на кваліфікаційну роботу.....	23
1.5 Висновки до розділу.....	24
2 Алгоритм синтезу радіально-базисних нейромереж на основі NNTool.....	26
2.1 Методи ідентифікації параметрів нейромереж.....	26
2.2 Програмне середовище Neural Network Toolbox.....	29
2.3 Алгоритм синтезу радіально-базисних нейромереж.....	38
2.4 Висновки до розділу.....	45
3 Реалізація алгоритму навчання радіально-базисних нейромереж.....	46
3.1 Структура програмної системи.....	46
3.2 Моделювання нейромереж із застосуванням NNTool.....	48
3.3 Експериментальні дослідження ефективності алгоритму.....	51
3.4 Висновки до розділу.....	57
Висновки.....	59
Список використаних джерел.....	60
Додаток А Світокопії публікацій.....	65
Додаток Б Довідка про використання.....	74

ВСТУП

У теперішній час нейромережі штучного типу є пріоритетним напрямком досліджень в області штучного інтелекту, методи якого працюють по аналогії із нервовою системою людини, зокрема: здатністю навчатися, виявляти та виправляти помилки, що, в свою чергу, дає можливість відтворити модель роботи людського мозку. Таким чином, нейромережі в штучному інтелекті – це спрощені моделі біологічних нейронів.

Штучні нейромережі характеризують унікальні властивості, проте здатність до навчання є ключовою, що вирізняє її з-поміж інших методів прогнозування, класифікації, розпізнавання. Тренування нейромережі перш за все полягає у підборі «сили» зв'язків між синапсами нейронів.

Нейромережі – це система з'єднаних та взаємодіючих простих процесорів, так званих нейронів штучного типу. Зазначені процесори, зазвичай, досить прості, мають справу лише з сигналами, що періодично отримують, і сигналами, що періодично посилає іншим таким процесорам. Варто зауважити, що з'єднання в досить потужну мережу із керованою взаємодією, такі процесори в сукупності уможливають виконання складних задач саме завдяки здатності до навчання в процесі роботи.

Інтелектуальні системи на основі нейромереж штучного типу дозволяють успішно розв'язувати задачі розпізнавання образів, прогнозування, оптимізації, асоціації та управління. Звісно, що для вирішення таких задач існують й інші, класичні методи, проте вони здебільшого не володіють достатньою гнучкістю, адаптованістю, в той же час, враховуючи задані умови та обмеження.

Сьогодні достатньо поширеними є різні архітектури нейромереж, проте за критерієм простоти розробки добре зарекомендували себе радіально-базисні структури. Не викака складність полягає у тому, що вони включають один радіально-базисний прихований шар. Таким чином, задача підбору кількості прихованих шарів не має змісту. Більше того, налаштування кількості нейронів

у вказаному прихованому шарі напряду залежить від кількості знайдених функцій радіального-базису.

Для моделювання архітектури нейромереж існує чимало програмних засобів, які здебільшого вимагають кваліфікації дослідника у розробці програм та вимагають суттєвих затрат часу на реалізацію функцій, процедур і об'єктів моделювання. Зважаючи на це, для розв'язування задач математичного моделювання з метою уникнення вищезазначених труднощів, ефективним є пакет прикладних програм Matlab, який побудовано на основі тулбоксів – функцій, що здатні вирішувати задачі певного класу. Зокрема, для моделювання нейромереж штучного типу застосовують Neural Network Toolbox, що характеризується зручним та зрозумілим інтерфейсом користувача, що дозволяє змодельовати архітектуру нейромережі із мінімальними затратами часу та обчислювальних ресурсів.

При дослідженні методів та алгоритмів ідентифікації радіально-базисних мереж штучного типу видно, впливає висновок, що немає чітко формалізованого методу одночасного налаштування параметрів прихованого шару та вихідного. Така умова забезпечує налаштування адекватної архітектури мережі шляхом переналаштування параметрів на основі заданої цільової функції.

Кваліфікаційну роботу виконано на основі вимог, що сформовано у працях [14, 15].

Мета роботи. Розробка алгоритму синтезу радіально-базисних нейромереж на основі об'єднання етапів налаштування параметрів прихованого та вихідного шару.

Задачі дослідження, необхідні до виконати, щоб досягти мету кваліфікаційної роботи:

- охарактеризувати штучні нейронні мережі;
- охарактеризувати особливості штучних нейронних мереж;
- проаналізувати відомі архітектури штучних нейронних мереж;
- проаналізувати програмні засоби моделювання архітектури

нейромереж;

- дослідити методи навчання нейронних мереж радіального типу;
- охарактеризувати параметри радіально-базисних нейромереж;
- розробити алгоритм синтезу нейронних мереж радіального типу;
- розробити програмний модуль для реалізації розробленого алгоритму

на основі NNTool;

– дослідити ефективність розробленого алгоритму на прикладі розв’язування прикладної задачі розпізнавання літер.

Об’єкт дослідження. Процеси навчання штучних нейромереж.

Предмет дослідження. Алгоритм розробки архітектури радіально-базисної нейромережі.

Методи дослідження. Аналізу та синтезу при дослідженні архітектур ата особливостей нейромереж штучного типу, кластеризації при обчисленні параметрів прихованого шару радіально-базисної мережі, зворотного поширення помилки при налаштуванні вагових коефіцієнтів, порівняльного аналізу при прийнятті рішення щодо вибору архітектури нейромережі, комп’ютерного моделювання та прогнозування при розробці архітектури радіально-базисної нейромережі.

Наукова новизна отриманих результатів дослідження:

- розроблено алгоритм синтезу архітектури радіально-базисних нейромереж, який на відміну від існуючих, поєднує процедури налаштування базисних функцій та вихідного шару;
- розроблено архітектуру нейромережі зазначеного типу для задачі розпізнавання літер.

Практичне значення отриманих результатів. На основі запропонованих методів та алгоритмів із застосуванням Neural Network Toolbox розроблено програмне забезпечення для розпізнавання літер англійського алфавіту.

Публікація та апробація результатів. Основні результати кваліфікаційної роботи опубліковано на IX Всеукраїнській науково-практичній конференції молодих науковців «Інформаційні технології – 2022» [28], VI науково-

практичній конференції «Інтелектуальні комп'ютерні системи та мережі» [29]. Копії публікацій поміщено у додатку А.

У першому розділі проаналізовано ознаки штучних нейромереж, як одного із провідних напрямів штучного інтелекту. Досліджено архітектури нейромереж, зазначено основні їх переваги та недоліки. Відзначено перспективність застосування радіально-базисних нейромереж штучного типу, як універсального апроксиматора із достатньо простою архітектурою. Проаналізовано методи налаштування архітектури таких нейромереж й на цій основі сформовано мету кваліфікаційної роботи та здійснено постановку задачі.

У другому розділі описано середовище моделювання MatLab, зокрема, Neural Network Toolbox, який призначений для розробки архітектур нейромереж. Розроблено алгоритм синтезу радіально-базисних нейромереж штучного типу на основі одночасного налаштування радіального базису та вагових коефіцієнтів. Описано процедуру моделювання нейромережі на основі NNTool.

У третьому розділі розроблено діаграму варіантів використання програмного забезпечення для моделювання архітектури нейромережі та на її основі описано структуру програмної системи. Реалізовано розроблений алгоритм синтезу на основі середовища MatLab й у результаті розроблено архітектуру нейромережі радіального базису для вирішення задач розпізнавання. Проведено експериментальні дослідження розробленої архітектури радіально-базисної нейромережі на основі розв'язування задачі розпізнавання літер англійського алфавіту.

У додатках представлено копії публікацій результатів кваліфікаційної роботи, довідка про використання.

1 АНАЛІЗ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

1.1 Характеристика нейромереж

Нейромережа штучного типу представляє зв'язані послідовно нейрони. Нейрони – це окремі об'єкти, які отримують і передають інформацію в певному напрямку. Сам нейрон, як окрема одиниця, не відіграє важливої ролі, однак у зв'язках їхня функція значима.

До нейрона надходять вхідні сигнали, кожному з яких присвоєно деяке значення. Сигнал множиться на свою вагу, значення підсумовуються, і виходить єдине число, яке отримує активаційна функція. На виході функція приймає рішення чи просувати сигнал далі.

Як зазначено вище, базовим елементом нейромережі є штучний нейрон. На рисунку 1.1 представлено будову штучного нейрона [13].

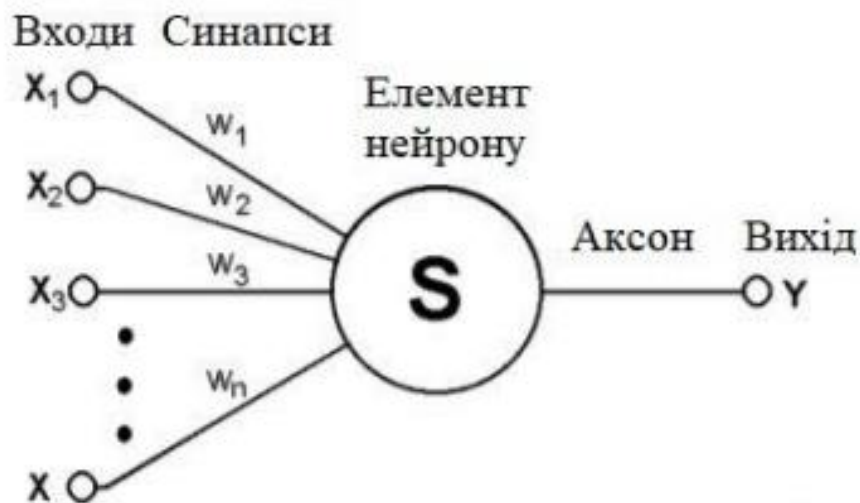


Рисунок 1.1 – Модель штучного нейрона

При цьому варто зазначити, що сумування зважених сигналів відбувається за формулою:

$$S = \sum_{i=1}^n x_i \cdot w_i . \quad (1.1)$$

Вихідний сигнал отримуємо із такого співвідношення:

$$Y = F(S). \quad (1.2)$$

Штучний нейрон, як видно на рисунку 1.1, володіє синапсами – однонапрямленими вхідними зв'язками, з'єднаними із виходами інших нейронів. Аксон – вихідний зв'язок штучного нейрона, з якого сигнал надходить на синапси наступних нейронів. Вхідні сигнали позначені на рисунку 1.1 (x_1, x_2, \dots, x_N) , надходять у штучний нейрон. Кожен синапс характеризується силою синаптичного зв'язку – ваговими коефіцієнтами w_i [13, 19, 20, 22].

Для реалізації нелінійності при активації нейрона його активність, окрім різних видів суматорів та систем вагових коефіцієнтів на входах, визначається функцією активації. Штучний нейрон, загалом, реалізує скалярну функцію векторного аргументу, а вихідний сигнал нейрона визначається видом функції активації і може бути дійсним або цілим.

Отже, із вищезазначеного випливає, що активність нейрона визначається параметрами – ваговими коефіцієнтами та функцією активації. Існує чимало функцій активації, що застосовують при розробці нейромереж, залежно від умов поставленої задачі та архітектури нейромережі штучного типу. Найпоширенішими є порогова функція активації, лінійна, сигмоїдальна, гіперболічного тангенсу, логістична [25].

Таким чином, архітектура нейромережі штучного типу складається із таких компонентів:

- вхідний шар;
- приховані (обчислювальні) шари;
- вихідний шар.

На рисунку 1.2 проілюстровано архітектуру простої одношарової нейромережі [27].

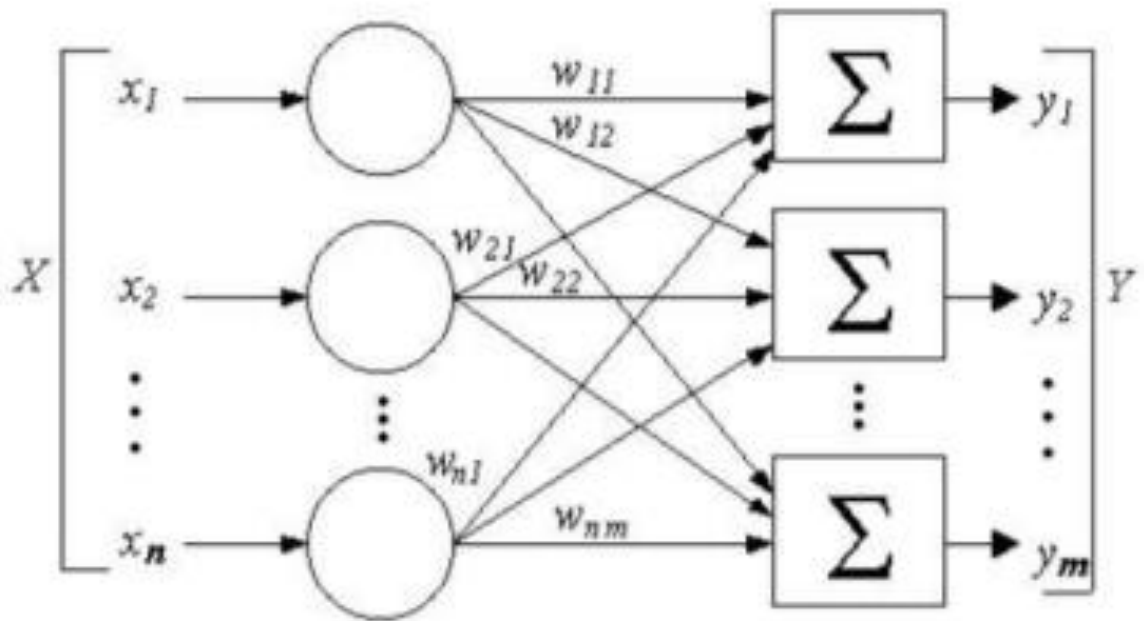


Рисунок 1.2 – Архітектура одношарової нейромережі

Відмінності обчислювальних процесів в нейромережах штучного типу часто обумовлені способом взаємозв'язків між нейронами. Таким чином, більш складними є багат шарові нейромережі.

За сукупністю критеріїв багат шарові архітектури розділяють на статичні та динамічні. До статичних архітектур відносять мережі прямого поширення, в яких реалізований однонапрямлений зв'язок між шарами, відсутні динамічні елементи і зворотній зв'язок, а вихід налаштованої нейромережі штучного типу визначається входом і не залежить від попередніх станів мережі. Динамічні архітектури реалізують рекурентну процедуру з використанням зворотніх зв'язків, зважаючи на що, стан нейромережі в кожен момент часу залежить від попереднього стану.

Зважаючи на вищеописане, у наступному підрозділі розглянемо детально існуючі архітектури нейромереж штучного типу та принципи їх роботи – специфіку передачі інформації, типи зв'язків між нейронами, кількість прихованих шарів та нейронів у них.

1.2 Аналіз архітектур нейромереж

За архітектурою нейромережі штучного типу поділяють на такі класи [1, 25, 38, 39]:

- багат шарові із прямою передачею інформації;
- рекурентні;
- клітинні;
- багат шарові із зворотною передачею інформації.

У багат шарових нейромережах із прямою передачею інформації, до яких відноситься багат шаровий перцептрон, нейрони з'єднані у шари, між якими інформація передається в одному напрямку із входу до виходу. Таку архітектуру зображено на рисунку 1.3. Результати опрацювання інформації із входу поступово передаються по шарах мережі до виходу, де отримуємо результуючий вихідний нейромережевий сигнал. Загалом, слід зауважити, що кількість прихованих шарів нейромережі може бути як завгодно багато. Це залежить від складності поставленої задачі [20].

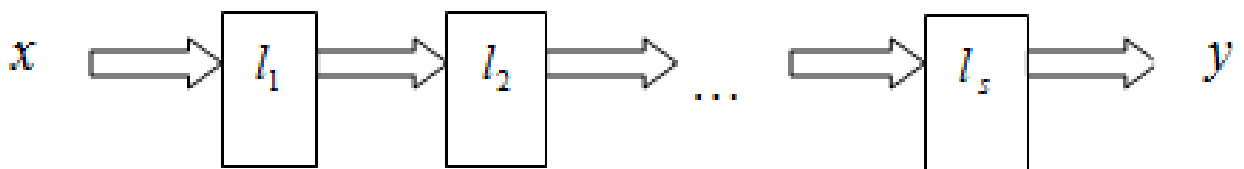


Рисунок 1.3 – Схема багат шарової нейромережі з прямим поширенням інформації

Архітектуру багат шарового перцептрона наведено на рисунку 1.4., де трьома крапками зазначено, що кількість прихованих шарів та нейронів у них може бути різною для кожної конкретної задачі. Процедура навчання якраз полягає у визначенні цих параметрів та інколи є достатньо трудомістким процесом.

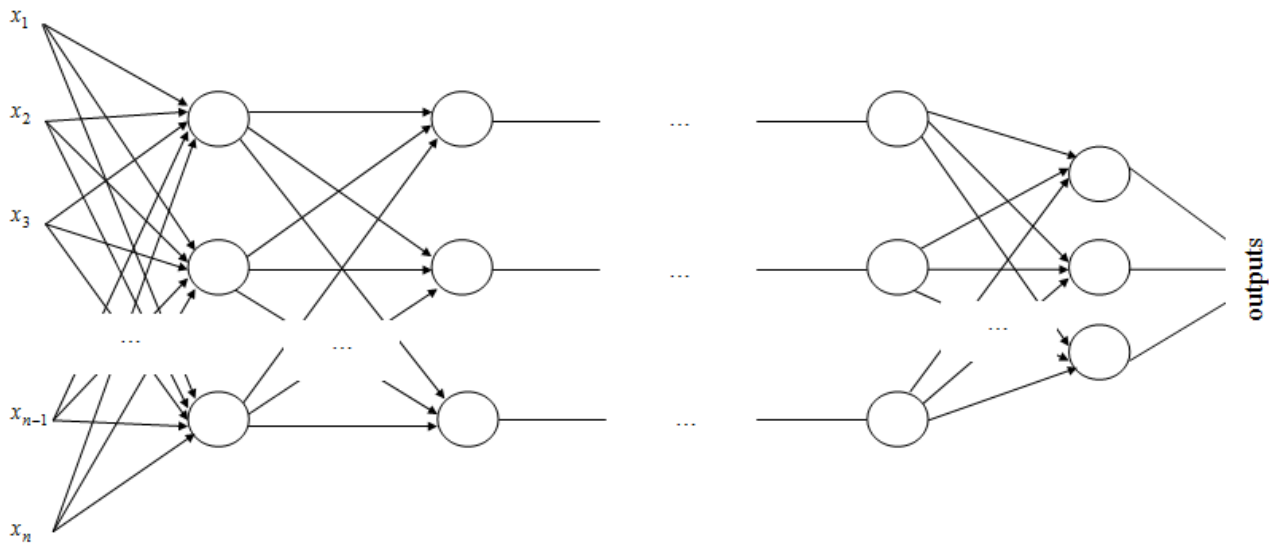


Рисунок 1.4 – Загальна архітектура багатошарового перцептрона

У рекурентних нейромережах штучного типу із виходу сигнали можуть подаватися на вхід чи приховані шари, тобто це мережі із зворотними зв'язками. Архітектуру зазначеного типу нейромереж проілюстровано на рисунку 1.5.

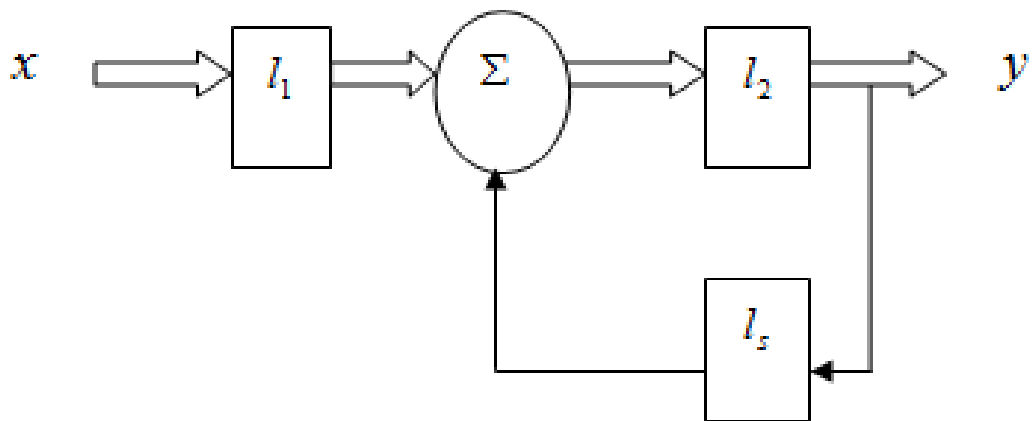


Рисунок 1.5 – Архітектура рекурентних нейронних мереж

У архітектурах клітинних нейромереж кожен нейрон зв'язаний із сусіднім нейроном. Тані структури є значно складніші, оскільки інформація від одного нейрона через синаптичні зв'язки передається до всіх його сусідніх нейронів. Зазначені архітектури потребують великих часових затрат на навчання. Структуру клітинних нейромереж проілюстровано на рисунку 1.6.

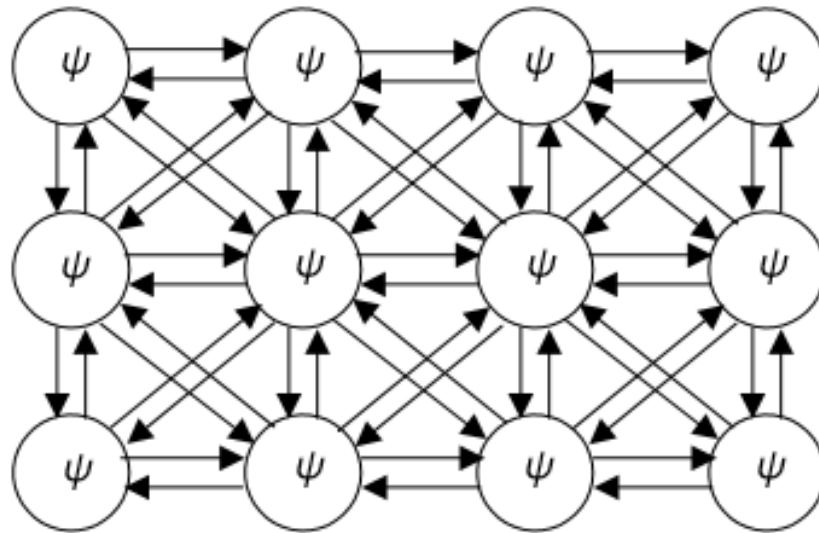


Рисунок 1.6 – Архітектура клітинних нейромереж

Зважаючи на вищенаписане, потік сигналів в нейромережах може бути або в одному напрямку або повторюватися. У першому випадку йде мова про динамічну структуру нейромережі, оскільки вхідні сигнали подаються у вхідній шар, а потім після опрацювання перенаправляються на наступний прихований шар, як показано на рисунку 1.7.

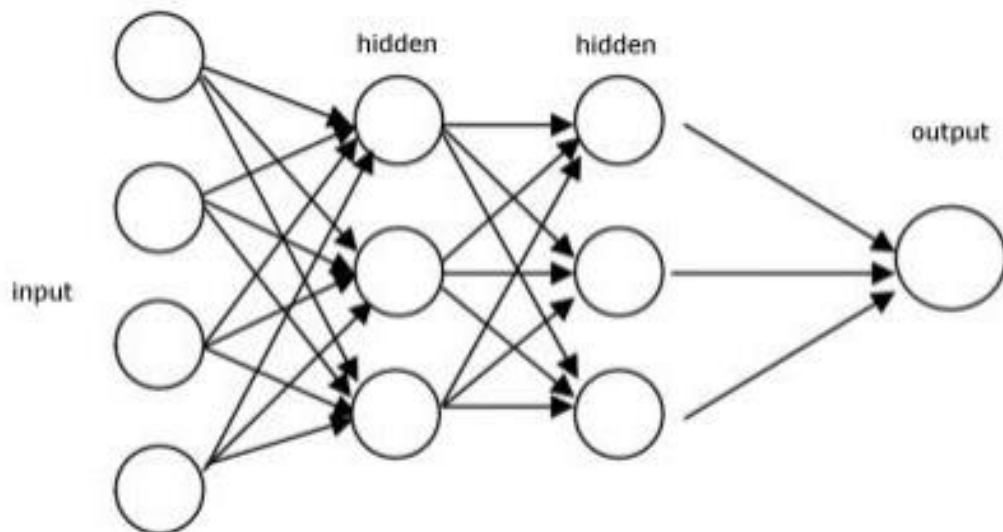


Рисунок 1.7 – Нейромережа прямого поширення інформації

Якщо нейромережа має певний внутрішній повтор, то це означає, що сигнали повертаються до нейрона або до шару, який вже отримував і

опрацьовував цей сигнал. У такому випадку неймережа штучного типу має зворотний зв'язок, що показано на рисунку 1.8.

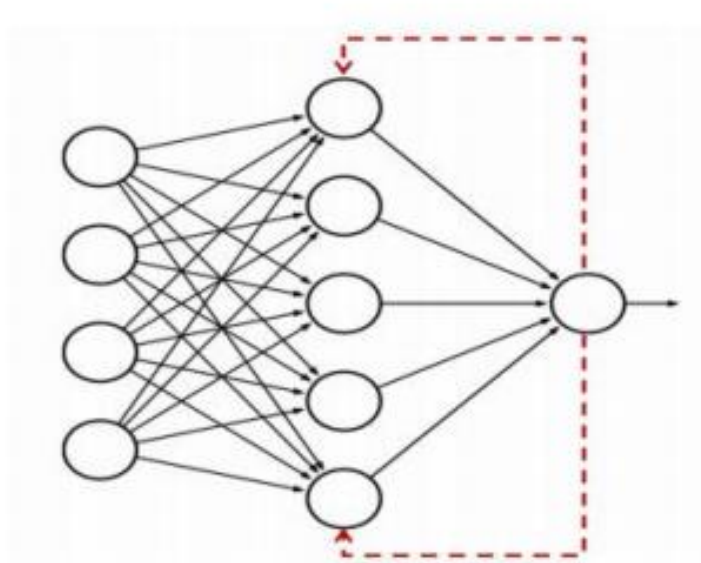


Рисунок 1.8 – Неймережа зворотного поширення інформації

Основною причиною повторення в мережі є створення динамічної поведінки, особливо коли вона вирішує задачі, пов'язані з тимчасовими рядами або розпізнаванням образів, для яких потрібна внутрішня пам'ять, щоб зміцнити процес навчання. Однак такі неймережі особливо важкі для навчання, а в кінцевому результаті часто не здатні навчитися. Більшість неймереж із зворотним зв'язком є одношаровими, зокрема, як мережі Ельмана і Хопфілда, проте можна побудувати і повторювану багатшарову мережу, наприклад, багатшаровий перцептрон, однак це вимагає суттєвих обчислювальних та часових ресурсів.

При використанні неймережевих методів у задачах часто виникає необхідність побудови неймережевої моделі об'єкта управління на основі отриманих вхідних та вихідних сигналів у реальному часі. Використання багатшарових перцептронів для побудови неймережевої моделі є складним у зв'язку з тим, що додаткове навчання багатшарового перцептрону в деякій ділянці робочої області призводить до втрати навченого стану у всій робочій

області нейромережі, що не дозволяє використовувати цей тип нейронних мереж для задач реального часу.

Зазначений недолік відсутній у радіально-базисних нейромережах (РБНМ), тому що кожен їх елемент впливає на значення вихідного сигналу переважно тільки в обмеженій ділянці робочої області, який характеризується положенням центру елемента і параметром σ , що називається шириною радіальної функції. Чим більше значення параметра σ , тим більший розмір області, на яку впливає даний елемент [1].

Радіально-базисна нейромережа – це архітектура, яка складається із трьох шарів: вхідного шару, на який подається вхідний сигнал-вектор, прихований шар, що складається з нейронів радіального типу та вихідний шар, який здійснює зважене підсумовування результату роботи прихованого шару. Структура радіальної нейронної мережі представлена рисунку 1. 9 [46].

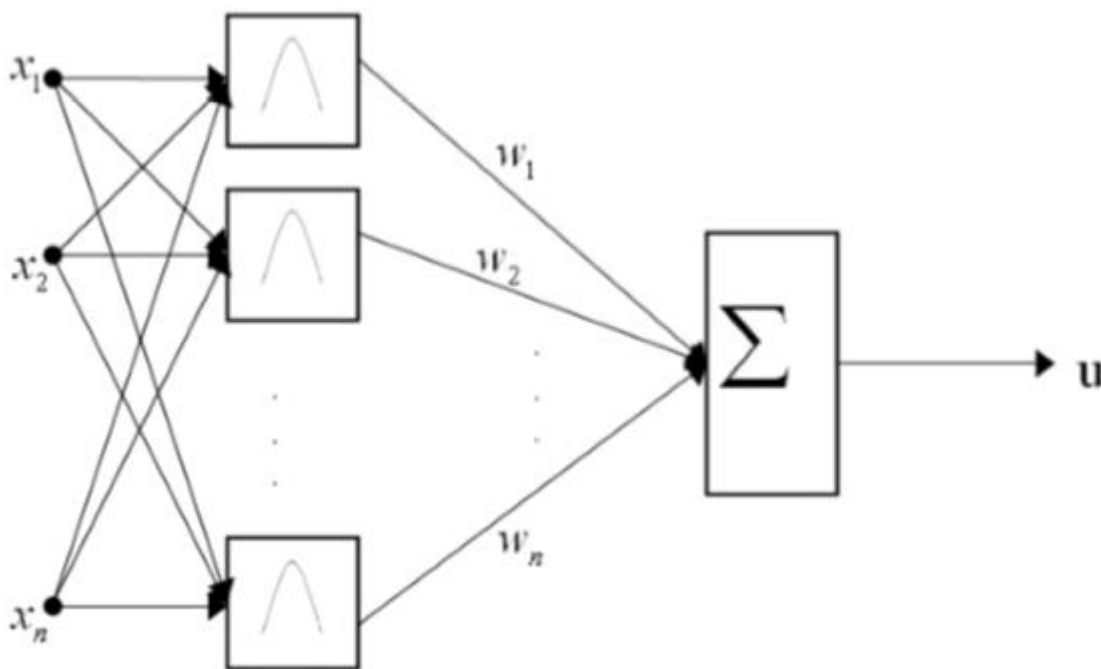


Рисунок 1.9 – Радіально-базисна архітектура нейромережі

Прихований шар перетворює вхідний вектор X , використовуючи різні функції радіального базису. Для створення ефективно функціонуючої мережі використовують різні види базисних функцій, зокрема, мультікватричну та функцію Гауса.

Методи навчання таких нейромереж варіюють від методу найменших квадратів до складних процедур на основі градієнтів. Проте такі методи вимагають попереднього задання кількості нейронів у прихованому шарі. Для налаштування параметрів прихованого шару, зазвичай, використовують методи кластеризації.

Для спрощення процедури моделювання вищеписаних архітектур нейромереж штучного типу застосовують різноманітні програмні засоби, які детально розглянемо у наступному підрозділі.

1.3. Аналіз програмних засобів синтезу нейромереж

На ринку нейромережових інструментальних засобів створення інтелектуальних систем представлена велика кількість різних програмних засобів, що пояснюється багатоплановістю задач інтелектуальної обробки інформації в різних сферах діяльності [12]. Проблема вибору відповідних програмних засобів полягає у виборі різноманітності функцій, можливостей налаштування, характеристик і властивостей, динамікою їх розвитку, розподіленою і різноманітною структурою систем захисту інформації.

До спеціалізованих інструментальних засобів створення нейромереж відносять нейропакет SNNS (Stuttgart Neural Network Simulator) [12]. Він спочатку був, як програмний імітатор для нейромереж 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 і графічний інтерфейс користувача. Емулятор ядра працює з внутрішніми мережами структур даних нейромереж і виконує операції навчання і перенавчання. SNNS може розширюватися за допомогою певних користувацьких функцій активації, вихідних функцій, функцій сайту і процедур

навчання, які записуються у вигляді простих програм на C і зв'язуються з емулятором ядра.

Нейропакет JavaNNS: Java Neural Network Simulator [19] є наступником SNNS. Він заснований на власному обчислювальному ядрі, володіє розробленим графічним інтерфейсом користувача на Java, який сумісний з нейропакетом SNNS і дозволяє збільшити платформенну незалежність.

Нейропакет Neural Lab [12, 19] надає візуальне середовище для розробки і тестування нейромереж штучного типу, інтегрується з Microsoft Visual Studio, з C ++, включає нейромережі в користувацькі додатки, науково-дослідне моделювання, інтерфейси кінцевих користувачів. Neural Lab виконує перевірку на наявність помилок наборів даних, перш ніж використовувати їх для навчання, зберігаючи значну кількість часу. Інтерфейс активації дозволяє тестувати ШНМ в режимі реального часу, що робить можливим порівняти візуально фактичний вихід нейромережі з бажаним.

Нейропакет NEURON є середовищем моделювання окремих нейронів і мереж нейронів. NEURON дозволяє моделювати окремі нейрони через використання секцій, які автоматично поділяються програмою замість їх ручного створення користувачем. Дані в програму записуються в інтерактивному режимі або завантажуються з файлу. Властивості мембрани каналів нейрона моделюються з використанням зібраних механізмів, написаних з використанням мови NMODL або скомпільовані процедурами, які працюють з внутрішніми структурами даних, створених за допомогою інструменту GUI (Channel Builder).

Нейропакет GENESIS [12] є середовищем моделювання для побудови реалістичних моделей нейробиологічних систем різних рівнів наближення, в тому числі субклітинних процесів, окремих нейронів, мережі нейронів і нейронних систем. Головною метою комп'ютерних реалізацій моделей GENESIS є відповідь на питання: «що відомо про анатомічну будову і фізіологічні особливості нейронної системи? ». GENESIS призначений для

кількісної оцінки нейронної системи таким чином, щоб було легко зрозуміти фізичну структуру нейрона.

Нейропакет XNBC є інструментом моделювання біологічних нейромереж з відкритим кодом.

Нейропакет Braincel [21] представляє програмну надбудову для табличного середовища MS Excel. Він дозволяє реалізовувати нейромережі прямого поширення з одним-двома прихованими шарами. Braincel використовує графічні можливості табличного процесора MS Excel та алгоритм навчання BackPercolation. Він дозволяє обробляти числові та символічні дані.

Нейропакет Excel Neural Package доповнює можливості MS Excel, по роботі з нейромережевими алгоритмами. Пакет складається із 2-х компонент.

1. Winnet призначений для пошуку і моделювання прихованих залежностей в даних, реалізує функції багат шарового персептрона, має графічний інтерфейс, візуальні засоби контролю, навчання, прогнозування, тестування, зупинку навчання при досягненні різних критеріїв.

2. Kohonen Map призначений для аналізу та побудови систем, що самоорганізують карти Кохонена, що вирішують завдання кластеризації, візуалізації.

Пакет Excel Neural Package зручний при візуалізації результатів у вигляді двовимірної карти, але має обмежений функціонал. Нейропакет Neural Planner – оболонка моделювання ШНМ різної конфігурації в середовищі MS Windows використовує 2 алгоритми навчання ШНМ з учителем: On-Line Back Propagation і Batch Back Propagation.

Neural Planner дозволяє вирішувати завдання класифікації, обробки даних, деякі математичні завдання, створювати експертні системи. Основними компонентами є: графічний і табличний редактори. В Neural Planner зручно створювати складно топологічні (наприклад, рекурентні) ШНМ при обмеженій кількості алгоритмів навчання.

Вільно розповсюджуваний нейропакет NeuroPro [12, 21], 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; створювати шари ШНМ, одночасно вирішувати ряд завдань прогнозування.

Нейропакет Partek Discovery [21], 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.

Нейропакет QwikNet32 [12, 21] призначений для роботи в середовищі MS Windows, реалізує багат шарову мережу прямого поширення (до 5 прихованих шарів, з набором з 6 модифікацій алгоритму зворотного поширення помилки). За замовчуванням працює алгоритм Online Backprop, при якому ваги і зміщення ШНМ коригуються після пред'явлення кожного вектора навчальної вибірки.

Нейропакет STATISTICA Automated Neural Networks є технологічно розвинутим програмним засобом для розробки нейромережевих додатків, призначеним для фахівців і непідготовлених користувачів в області нейромережевих обчислень. Нейропакет THINKS і ThinksPro є комплексом інструментальних засобів для розробки нейромережевих систем.

Компонент THINKS – нейромережеве середовище розробки додатків, як 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 формах значень входів, станів, ваг, виходів.

Із аналізу вищеописаних програмних інструментів призначених для моделювання ШНМ видно, що вони вимагають досвіду розробки у написанні програмних кодів на різних мовах програмування, а також уможливають розв'язання вузького кола задач, тобто є вузьконаправленими.

Зважаючи на вищезазначене, варто розглянути програмний пакет для моделювання систем – MatLab, який містить чимало вбудованих бібліотек для

розв'язування задач різних класів. Same Neural Network Toolbox призначений для моделювання архітектур нейромереж штучного типу.

1.4 Постановка задачі на кваліфікаційну роботу

Нейромережі це потужний математичний апарат для моделювання, побудований за принципом біологічних нейронів, які здатні накопичувати, зберігати та передавати інформацію у такому вигляді, який потрібний для прийняття рішення при розв'язуванні задач. Ключовим будівельним елементом нейромереж є штучний нейрон.

Архітектура нейромереж штучного типу складається із вхідного, прихованого та вихідного шарів, нейрони яких зв'язуються на основі вагових коефіцієнтів синаптичних зв'язків. У нейромереж перцептронного типу прихованих шарів може бути дуже багато, що залежить від складності поставленої задачі. Проте велика кількість прихованих шарів і нейронів у них призводить до ускладнення архітектури та до збільшення часових та обчислювальних витрат. Виграшними в цьому сенсі є архітектури радіально-базисних нейромереж, які містять лише один прихований шар. При цьому задача визначення кількості прихованих шарів виключена й потрібно обчислити лише кількість нейронів у прихованому шарі.

Існує чимало архітектур нейромереж із прямим та зворотним поширенням інформації, які уможливають розв'язати широкий коло задач. Саме розробка адекватної архітектури нейромережі шляхом моделювання є запорукою успішного досягнення потрібного результату. Наявні програмні засоби моделювання нейромереж унеможливають в комплексі дослідити усі властивості та характеристики потрібної архітектури мережі. В той же час для розробки моделей все частіше застосовують програмне середовище Matlab, яке містить велику кількість вбудованих інструментів – тулбоксів.

Важливою функцією нейромереж є здатність до навчання на основі методів та алгоритмів, яка полягає у налаштуванні вагових коефіцієнтів, що визначають вагу синаптичних зв'язків. Здебільшого алгоритми навчання нейромереж ґрунтуються на мінімізації похибки навчання.

Специфіка налаштування архітектури радіально-базисної нейромережі полягає у ідентифікації параметрів прихованого шару та параметрів вихідного шару. Досить часто при розробці архітектур доводиться корегувати знайдені параметри, а це вимагає затрат часу. Проведені дослідження підтверджують відсутність алгоритму навчання нейромереж такого типу, який би об'єднував процедури налаштування параметрів обох рівнів.

Зважаючи на це, метою кваліфікаційної роботи є розробка алгоритму синтезу радіально-базисних нейромереж штучного типу, який включає процедуру налаштування базисних функцій та вагових коефіцієнтів.

Для досягнення цієї мети у роботі потрібно виконати такі задачі:

- дослідити методи навчання нейронних мереж радіального типу;
- охарактеризувати параметри радіально-базисних нейромереж;
- розробити алгоритм синтезу нейронних мереж радіального типу;
- розробити програмний модуль для реалізації розробленого алгоритму на основі NNTool;
- дослідити ефективність розробленого алгоритму на прикладі розв'язування прикладної задачі розпізнавання літер.

1.5 Висновки до розділу

Описано основні характеристики нейромереж штучного типу, зазначено, що основним будівельним елементом при цьому є штучний нейрон. Розглянуто відомі архітектури нейромереж із прямим та зворотним поширенням сигналу. Проаналізовано особливості архітектури багатошарового персептрона й

зазначено, що її налаштування є достатньо громіздкою та затратною задачею. Відзначено простоту радіально-базисних архітектур нейромереж та перспективність їх застосування для розв'язування прикладних задач. Описано основні програмні засоби моделювання нейромереж штучного типу, виокремлено їх переваги та недоліки та доцільність застосування середовища MatLab для розробки та дослідження таких структур.

2 АЛГОРИТМ СИНТЕЗУ РАДІАЛЬНО-БАЗИСНИХ НЕЙРОМЕРЕЖ НА ОСНОВІ NNTOOL

2.1 Методи ідентифікації параметрів нейромереж

Засоби розробки нейромереж, як правило, включають засоби моделювання, так звані імітатори або нейропакети, засоби навчання і графічний інтерфейс користувача. Такі програмні засоби реалізують алгоритм синтезу моделі мережі, навчання та тестування [12].

Моделі нейромереж визначаються моделями формальних нейронів і структурою міжнейронних зв'язків, види яких представлено на рисунку 2.1.

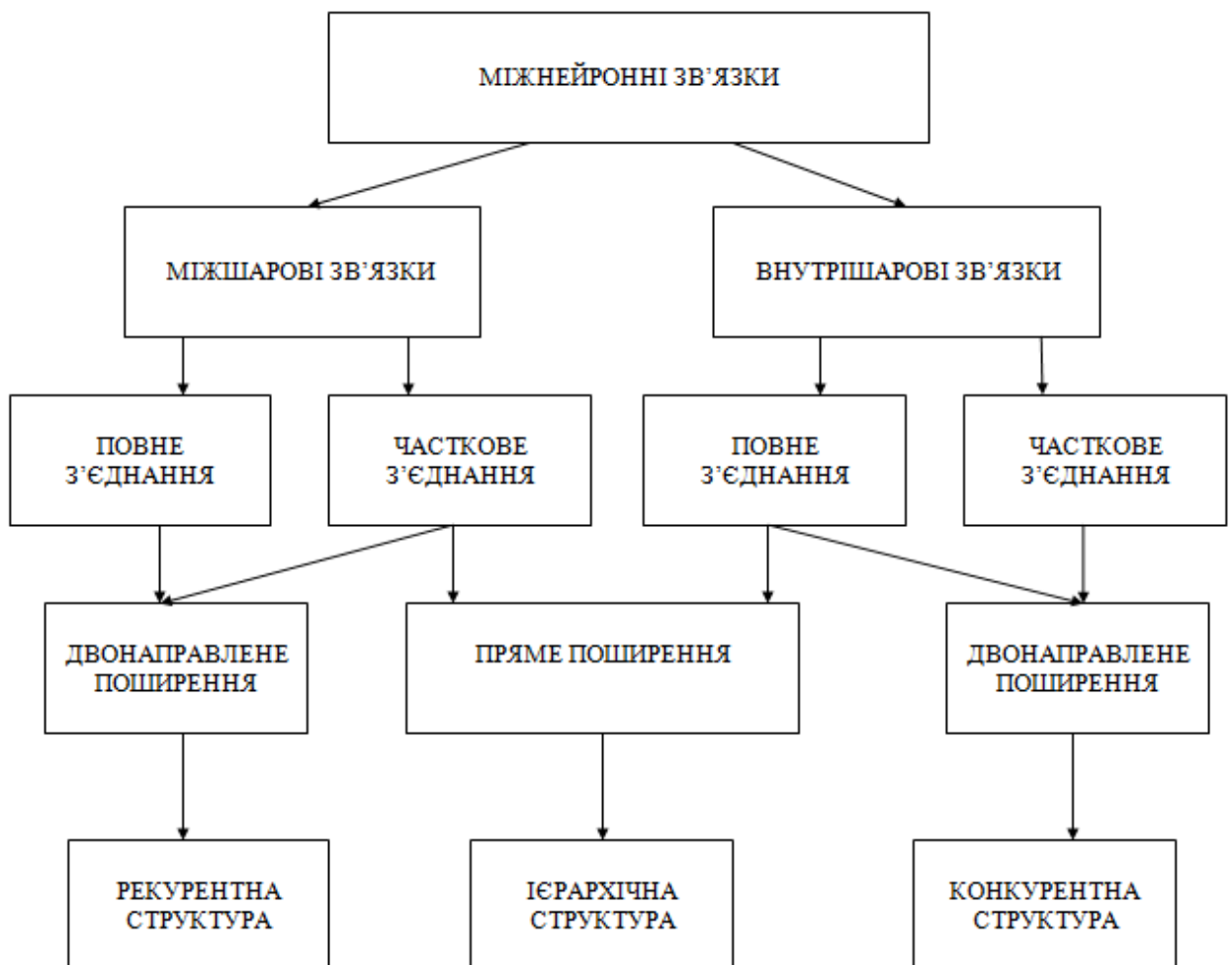


Рисунок 2.1 – Типи міжнейронних зв'язків

Залежно від структури зв'язків у нейропакетах, зазвичай, реалізуються такі топології нейромереж штучного типу [22]:

- багат шарова;
- повнозв'язна (кожен формальний нейрон зв'язаний з усіма іншими нейронами мережі;
- з локальними зв'язками (нейрони розташовуються у вузлах прямокутової або гексагональної решітки мережі, кожен нейрон зв'язаний із найближчими сусідніми);
- неструктурована (модель мережі, яка не має структури).

У більшості нейропакети використовують стандартні процедури навчання нейромереж штучного типу, орієнтовані на конкретні нейропарадигми. Зазвичай, в нейропакет можливе задання різних типів даних і розмірності вхідних і вихідних сигналів в залежності від поставленої задачі. Вхідними даними можуть бути: зображення, числа, розподіл значень, а типом вхідних даних – числа на заданому діапазоні: бінарні, біполярні, цілі чи дійсні. Вихідними сигналами нейромереж є цілі або дійсні числа, або їх вектори.

При вирішенні практичних задач, зазвичай, використовуються навчальні вибірки, що передбачено у нейропакетах у вигляді засобів їх формування. Середньоквадратичне відхилення використовується, в більшості випадків, в якості помилки і дозволяє встановити відповідність вихідних сигналів нейромережі бажаним вихідним сигналам навчальної вибірки. У багатьох нейропакетах є можливість вибору або задання функції помилки.

Реалізовані в нейропакет алгоритми навчання нейромереж штучного типу поділяють на групи [21]:

- градієнтні;
- стохастичні;
- генетичні.

В градієнтних алгоритмах обчислення частинних похідних функції помилки відбувається за параметрами нейромереж. В стохастичних алгоритмах мінімум функції похибки шукають випадковим чином. У генетичних

алгоритмах поєднуються властивості градієнтних і стохастичних алгоритмів (перебір варіантів реалізують на основі операції успадкування, градієнтний спуск реалізують на основі операції відбору).

Наявність процедур ініціалізації нейромереж перед навчанням також передбачено в нейроімітатора (присвоєння параметрам нейромереж початкових значень). Критеріями зупинки при навчанні нейромереж штучного типу є:

- коли значення функції похибки досягає заданого значення;
- коли успішно вирішуються всі приклади навчальної вибірки на кожній ітерації.

Однією із найважливіших характеристик програмних засобів для моделювання нейромереж – швидкість збіжності алгоритму навчання.

Тестування навчених мереж штучного типу є однією із важливих задач У нейроімітатора передбачено засоби тестування для верифікації навченої нейромережі. Для цього подають вхідні сигнали, які не брали участі у навчанні мережі і перевіряють правильність результату, який формується на вході нейромережі. Тестування проводять на одиничних вхідних сигналах або на тестовій вибірці, що складається із пар бажаних вхідних сигналів. Тестову вибірку будують індивідуально, залежно від поставленої задачі.

Оцінку інструментальних програмних засобів моделювання зручно проводити, виходячи з можливості використання різних нейромережевих структур, критеріїв оптимізації та алгоритмів навчання, а також простоти використання нейропакета і наочності подання інформації. При тестуванні нейромережі штучного типу варто враховувати [21]:

- сферу застосування, функціонал використання програмного забезпечення для розробки нейромережевих систем;
- можливості реалізації: розробки, навчання, наявність графічного інтерфейсу;
- можливість підготовки навчальних вибірок;
- цілісності, конфіденційності, доступності представленої інформації при розробці, навчанні нейромереж;

- застосування нейромережових парадигм, критеріїв, алгоритмів навчання;
- створення користувацьких нейромереж, використання призначених для користувача критеріїв оптимізації та алгоритмів навчання;
- обміну даними між нейропакетами і додатками операційної системи; розширення нейропакета за рахунок своїх програмних модулів;
- генерації вихідних кодів; можливості інтеграції з іншим програмними засобами.

Також не менш важливим при виборі програмного засобу для моделювання нейрорежі є наявності чи перевага будь-якої функції (наприклад, виду навчання, тестування, перенавчання нейронної мережі) або можливості задання певних параметрів (наприклад, видів архітектур нейромереж, їх зв'язків, кількості входів, виходів, вагових коефіцієнтів, кількості шарів, кількості епох навчання мережі).

Розв'язок задачі може полягати у необхідності проведення моделювання тільки на одній архітектурі нейромережі, тоді достатньо використати конкретний вузькоспеціалізований програмний інструмент. У випадку, коли необхідно дослідити модель на різних нейромережах із різними параметрами, потрібно використовувати універсальний програмний комплекс.

2.2 Програмне середовище Neural Network Toolbox

Нейропакет Neural Network Toolbox [6, 7, 11, 37, 44] є компонентом пакету прикладних програм Matlab і має модульну, відкриту і розширену архітектуру, що забезпечує підтримку широкого класу нейромережових парадигм. Він включає багат шаровий персептрон з методом зворотного поширення помилки, радіально-базисні, рекурентні мережі, нейромережі із змагальними шарами і самоорганізуючі карти. Середовище Neural Network

Toolbox має зручний графічний інтерфейс користувача для розробки, дослідження і управління процесами в режимах навчання і роботи нейромережі штучного типу.

У пакет входить більше 15 типів нейромереж і навчальних правил, які дозволяють користувачу вибрати найбільш підходящу для конкретної задачі. Для кожного типу архітектури та навчального алгоритму нейромереж штучного типує функції ініціалізації, навчання, створення, моделювання, демонстрації, а також приклади застосування.

За допомогою Neural Network Toolbox можна змоделювати найбільш поширені види нейромереж, а також вибрати спосіб навчання, який найкраще підходить для вирішення поставленої задачі. Neural Network Toolbox так само підтримує архітектури контрольованих і неконтрольованих мереж.

У Neural Network Toolbox передбачений, як зазначено вище, графічний інтерфейс користувача, який викликається за допомогою команди `nntool` і на екрані з'явиться вікно `Network / Data manager`, екранну форму якого продемонстровано на рисунку 2.2, де:

- `Input data` – задаються входи нейромережі;
- `Target data` – задається послідовність цілей;
- `Networks` – список створених нейромереж;
- `Outputs data` – відображаються послідовності виходів нейромережі;
- `Error data` – показує помилки;
- `Input delay state` – початкові умови лінії затримки входу;
- `Layer delay state` – початкові умови лінії затримки шару;
- `Import` – кнопка для завантаження вже існуючих даних;
- `New` – кнопка, за допомогою якої можна створити нейромережу;
- `Open` – служить для перегляду даних, які знаходяться в змінних або ж для налаштування нейромережі;
- `Export` – служить для експорту даних в робочу область Matlab або жля збереження даних на диск;

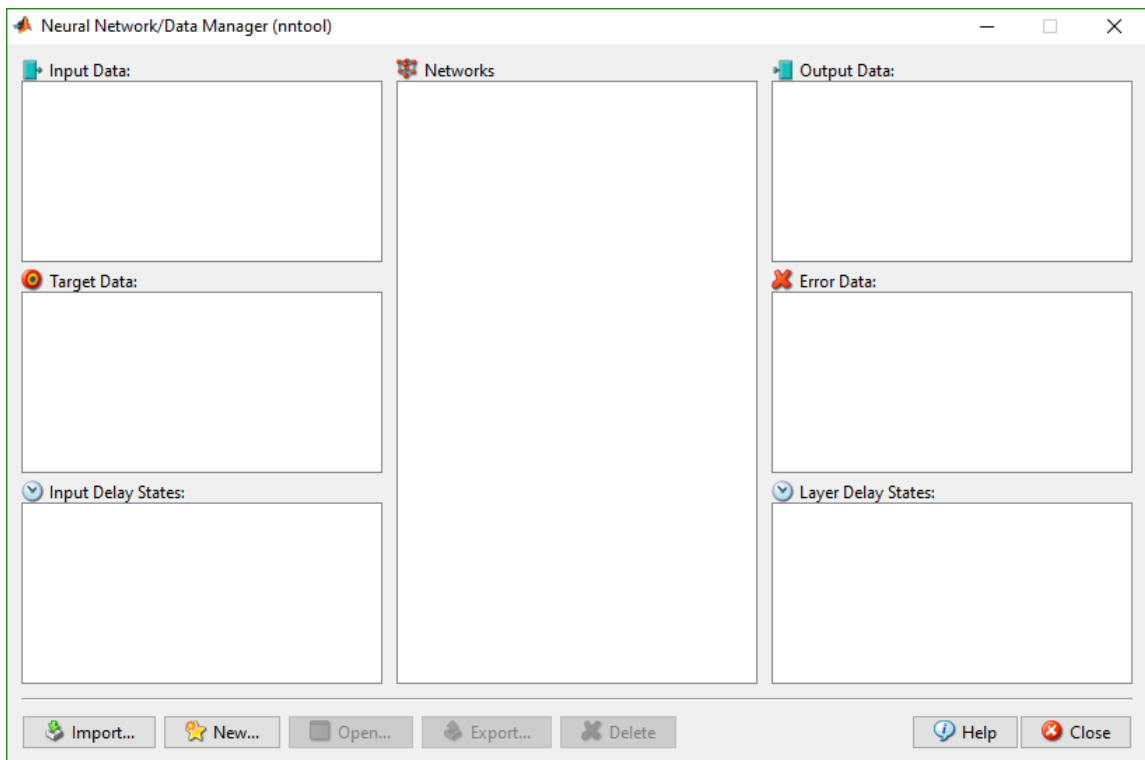


Рисунок 2.2 – Екранна форма головного діалогового вікна функції nntool

- Delete – використовується для видалення обраних змінних;
- Help – застосовують для виклику вікна довідки, в якому описуються правила роботи з диспетчером nntool.

При натисканні на кнопку Import з'являється діалогове вікно із вибором, звідки потрібно завантажувати файли – з диска чи з робочої області Matlab. Екранну форму зазначеного діалогового вікна представлено на рисунку 2.3. Аналогічне вікно, що зображене на рисунку 2.4, з'являється при нажатті кнопки Export, проте тут потрібно вказувати, куди збережені дані експортувати.

На рисунку 2.5 показано діалогове вікно, яке призначене для розробки архітектурно-структурної нейромережі. У вікні можна вибрати тип нейромережі, кількість нейронів в прихованому шарі і кількість прихованих шарів, функцію активації, алгоритм навчання, а також побачити вигляд архітектури розробленої нейромережі.

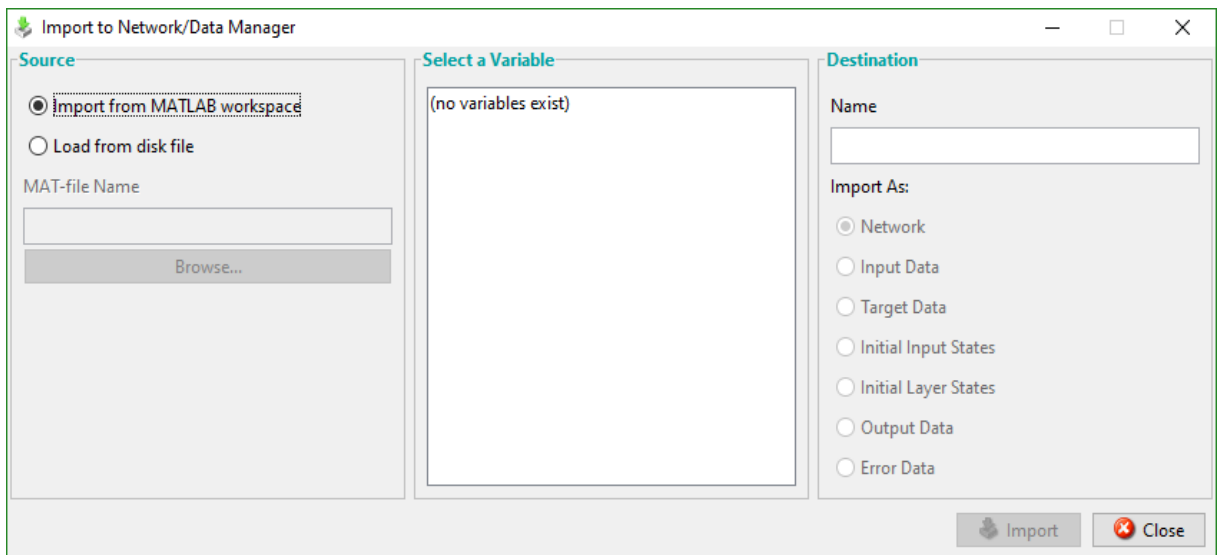


Рисунок 2.3 – Екранна форма діалогового вікна імпорту даних

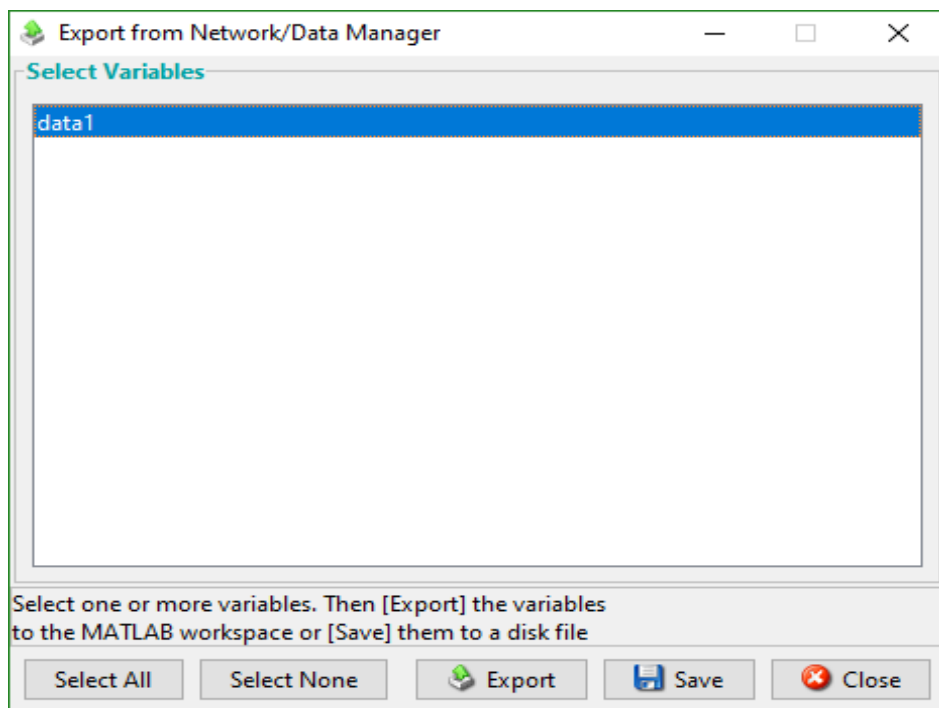


Рисунок 2.4 – Екранна форма діалогового вікна експорту даних

Моделювання штучної нейронної мережі. У вікні Network / Data Manager обираємо New Network, Create New Network, обираємо тип нейромережі (по замовчуванню – Feed-forward backprop з прямою передачею сигналу і зворотнім поширенням помилки), функцію активації у другому шарі – за замовчуванням – гіперболічний тангенс (TANSIG).

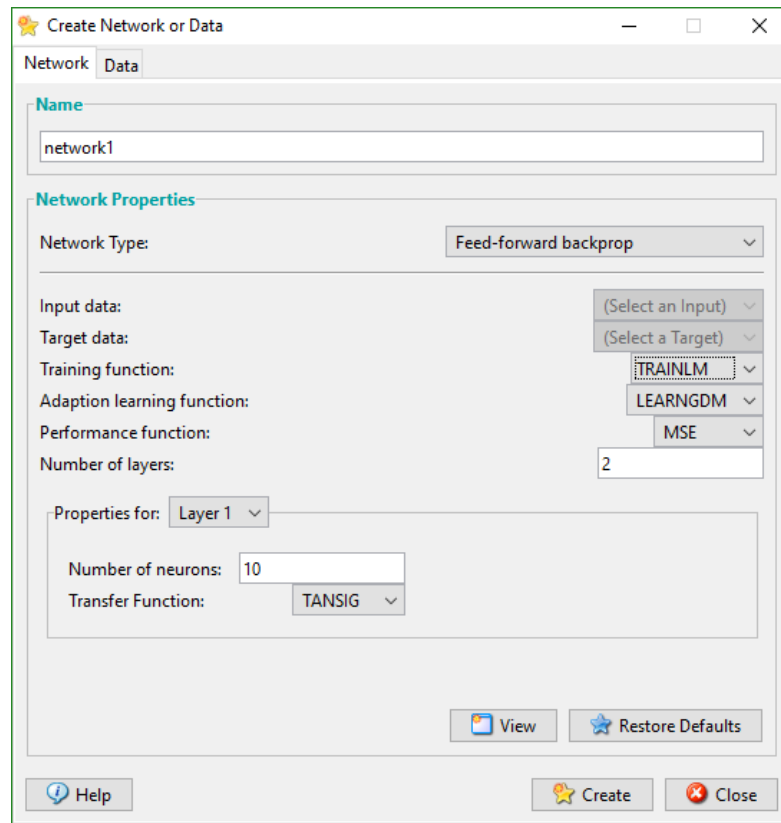


Рисунок 2.5 – Екранна форма діалогового вікна розробки архітектури нейромережі

Слід зазначити, що при моделюванні нейромережі штучного типу насамперед необхідно задати їй унікальне ім'я, яке буде вирізняти її від уже створених.

Екранну форму діалогового вікна, яке відповідає за створення нейромережі зображено на рисунку 2.6. Як бачимо на рисунку, графічний інтерфейс дуже зручний, усі необхідні функції та алгоритми навчання є вбудованими. Кількість шарів та нейронів у них дослідник задає експериментальним чином, спираючись на вхідні експериментальні дані. Досить часто через не достатній досвід розробника архітектура нейромережі є непридатною для моделювання, зважаючи на це – вибір кількості шарів у прихованому шарі та нейронів у них є важливою задачею. Слід зауважити що для радіально-базисної архітектури задача спрощується шляхом підбору кількості нейронів у прихованому шарі. Це вирішується на основі вхідної вибірки даних.

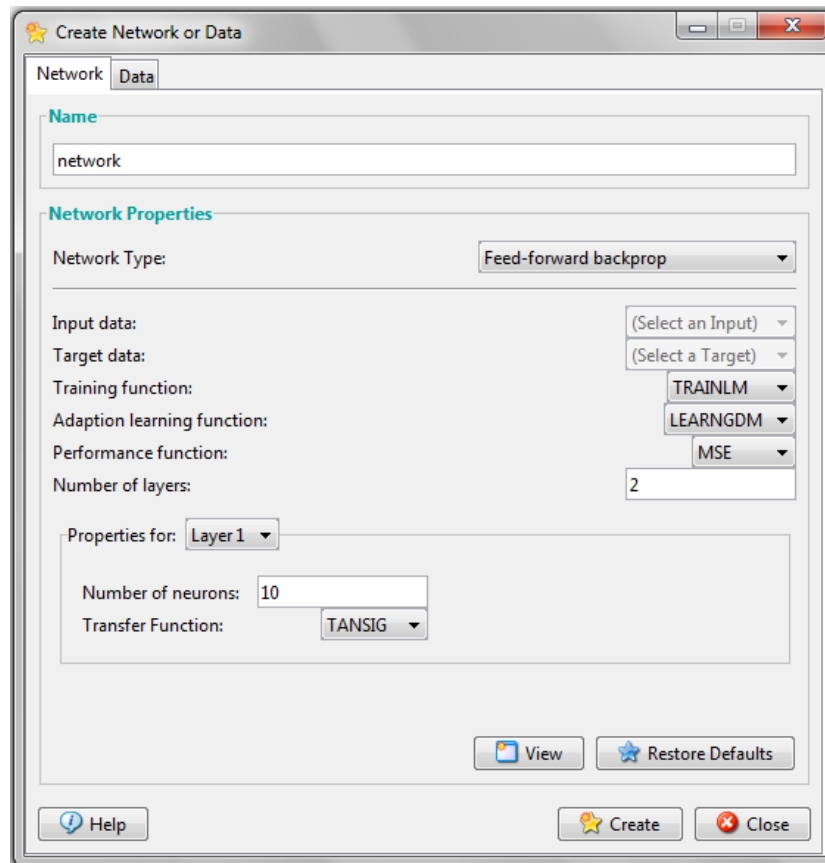


Рисунок 2.6 – Екранна форма вікна розробки штучної неймережі

Після розробки нової архітектури неймережі у вікні Network / Data Manager, в області Networks з'явиться її ім'я. Навчаємо мережу, вибравши вкладку Train. При цьому відкривається діалогове вікно, яке показано на рисунку 2.7.

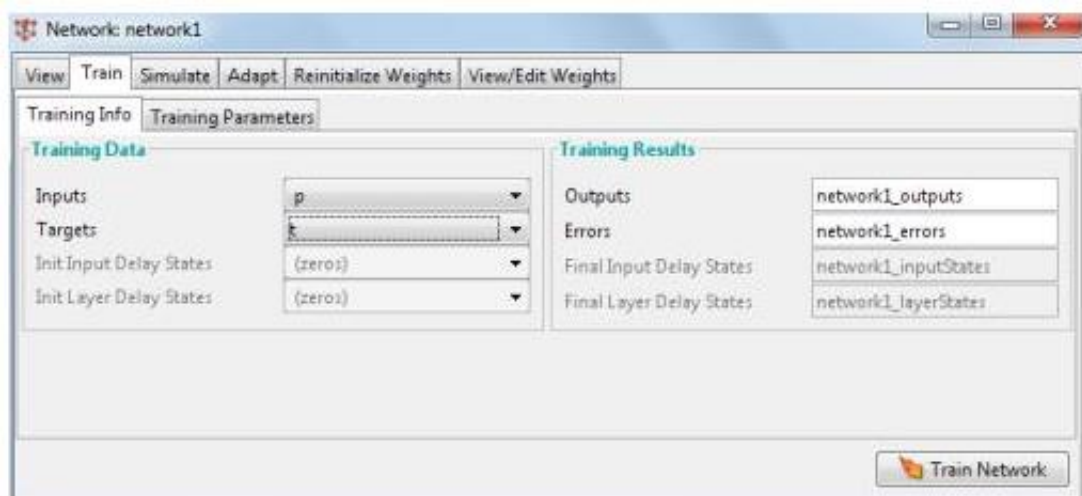


Рисунок 2.7 – Екранна форма діалогового вікна навчання неймережі

Параметри навчання нейромережі штучного типу можна відслідкувати у діалоговому вікні, екранну форму якого проілюстровано на рисунку 2.8.

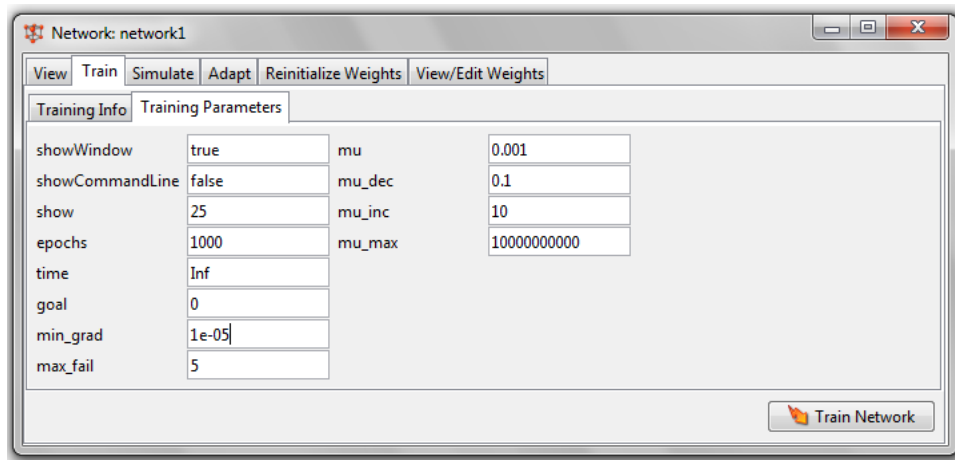


Рисунок 2.8 – Екранна форма діалогового вікна візуалізації параметрів навчання

Процес навчання нейромереж відображено на рисунку 2.9. Тут можна поспостерігати, як навчалася нейромережа, видно кількість епох навчання, витрачений час, продуктивність, архітектуру.

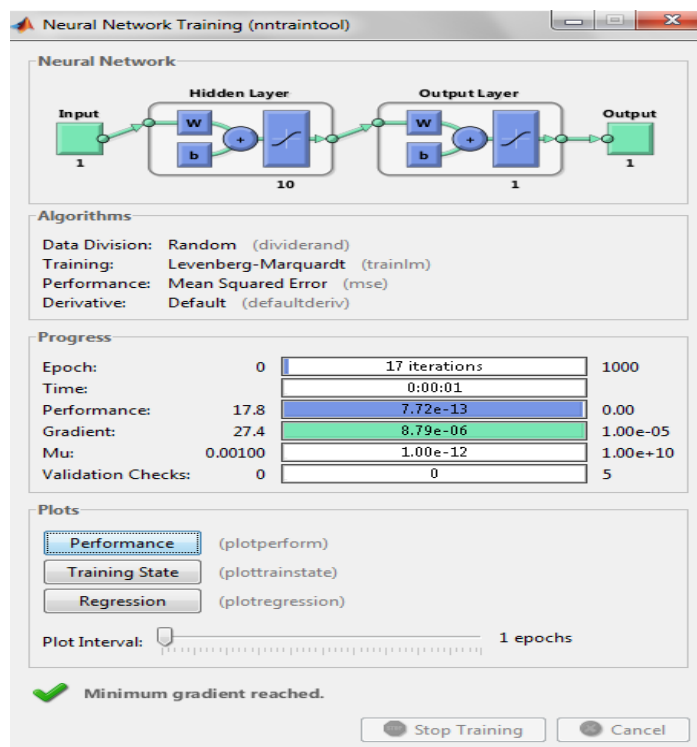


Рисунок 2.9 – Екранна форма вікна візуалізації процесу навчання нейромережі

Якість навчання нейромережі для обраної навчальної вибірки відображає похибка навчання, яка повинна зменшуватися у процесі навчання. Результат зміни похибки у процесі навчання нейромережі на основі алгоритму зворотного поширення похибки продемонстровано на рисунку 2.10.

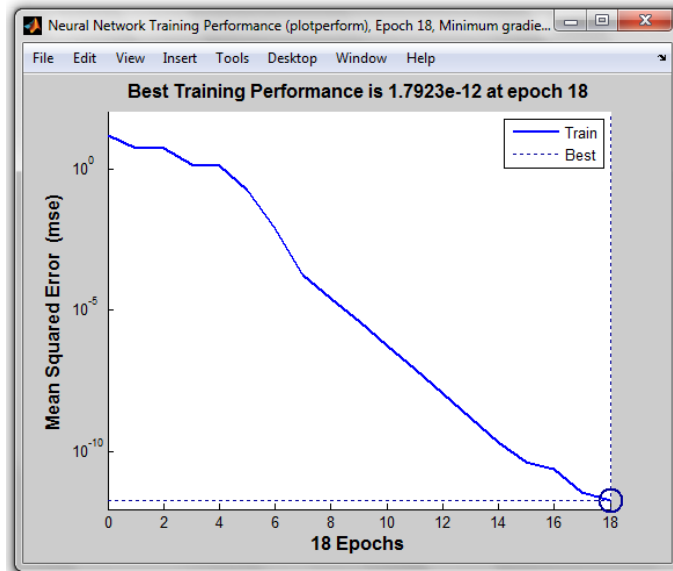


Рисунок 2.10 – Зміни похибки навчання нейромережі

Зміну станів навчання нейромережі штучного типу проілюстровано на рисунку 2.11.

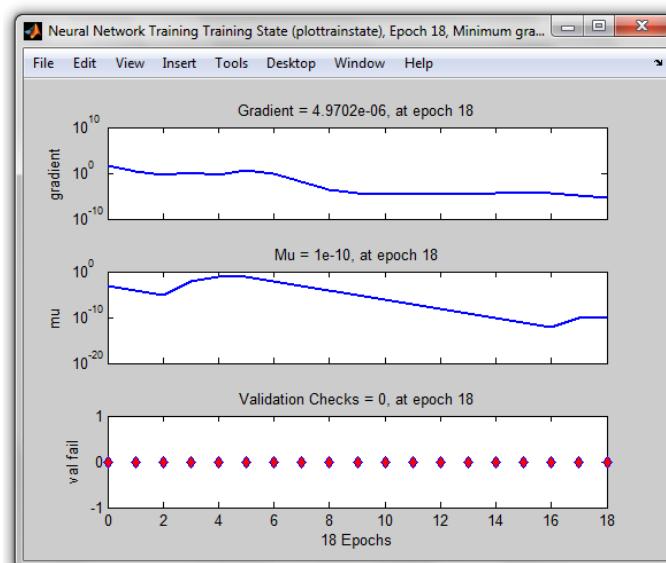


Рисунок 2.11 – Зміна стану навчання нейромережі

Значення налаштованих вагових коефіцієнтів сили синаптичних зв'язків для всіх шарів розробленої архітектури нейромережі штучного типу можна проглянути у вікні, яке зображено на рисунку 2.12.

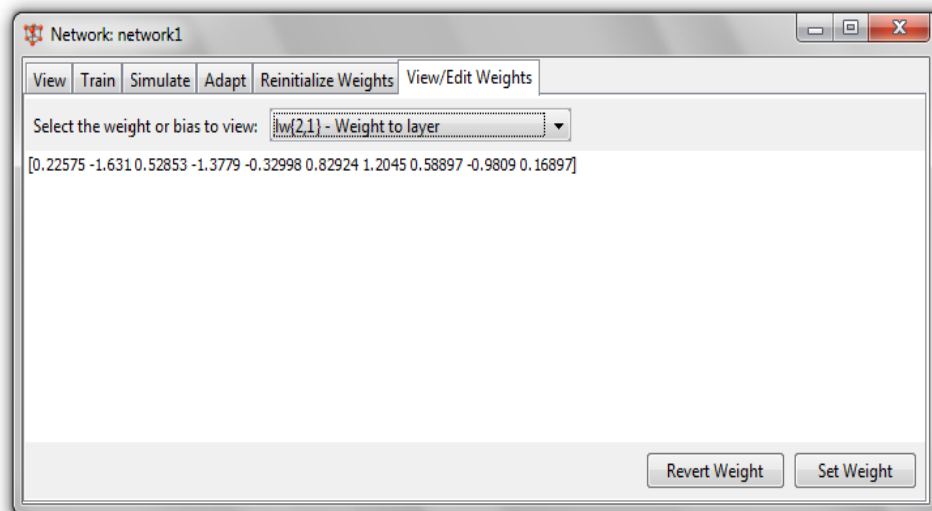
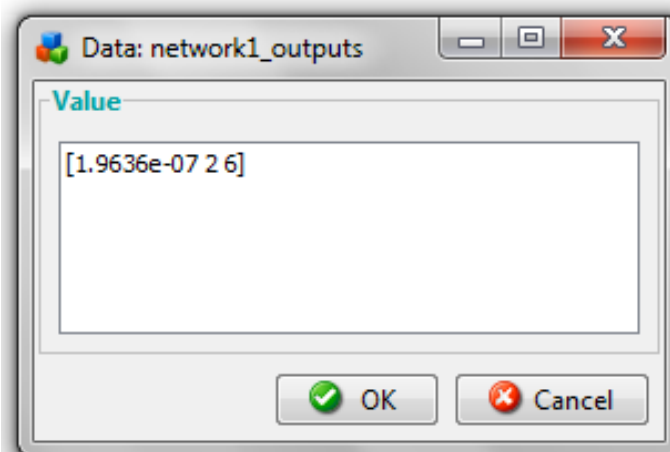
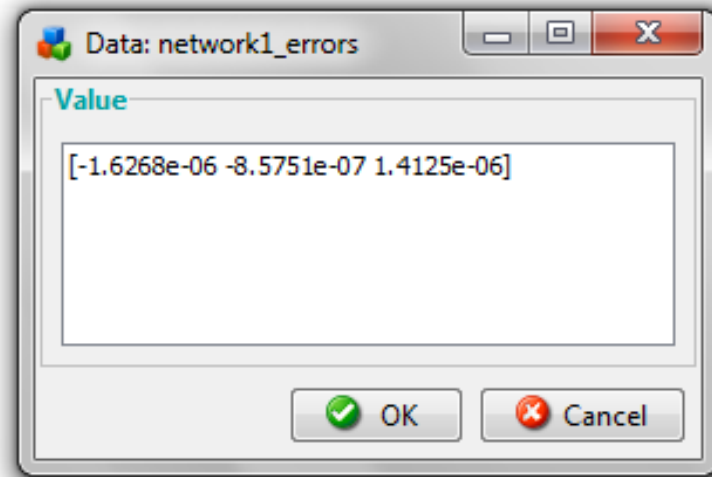


Рисунок 2.12 – Екранна форма вікна візуалізації значень вагових коефіцієнтів

Останнім кроком моделювання архітектури нейромережі штучного типу є тестування роботи нейромережі на тестовому наборі даних, використовуючи вкладку Simulate. Результати тестування розробленої архітектури проілюстровано на рисунку 2.13 а) – значення вихідного сигналу, б) – значення похибки мережі.



а)



б)

Рисунок 2.13 – Результати тестування штучної нейромережі

За результатами тестування розробленої архітектури нейромережі штучного типу приймають рішення щодо працездатності нейромережі та її адекватності до розв'язування поставленої задачі.

2.3 Алгоритм синтезу радіально-базисних нейромереж

Структуру РБНС показано на рисунку 1 9. Мережа складається із двох шарів. Вхідні сигнали надходять на нейрони першого шару без змін, при чому [1]:

n – кількість нейронів у першому шарі;

x_1, x_2, \dots, x_n – вхідні сигнали;

h – кількість нейронів у другому шарі;

$c_{i1}, c_{i2}, \dots, c_{in}$ – координати центру i -го нейрона;

i – ширина радіальної функції i -го нейрона;

θ_i – вихідний сигнал i -го нейрона;

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

y – вихідний сигнал РБНС. Вихідний сигнал кожного елемента визначається функцією Гауса [2]:

$$\theta_i = \exp\left(-\frac{\sum_{j=1}^n (x_j - c_{ij})^2}{2\sigma_i^2}\right). \quad (2.1)$$

Вихідний сигнал РБНС обчислюється як зважена сума сигналів нейронів:

$$y = \sum_{i=1}^h w_i \cdot \theta_i. \quad (2.2)$$

Для навчання РБНС використовують, зазвичай, градієнтний алгоритм, що ґрунтується на мінімізації цільової функції помилки мережі. Відповідно до цього алгоритму для кожного нейрона обчислюються величини змін вагового коефіцієнта Δw_i , ширини базисної функції $\Delta \sigma_i$ та координат центру нейронів Δc_{ij} .

Оскільки більшість алгоритмів ідентифікації параметрів вихідного шару нейромережі ґрунтується на мінімізації функції похибки [1, 45], то при $k=1, \dots, N$ цільову функцію можна записати у вигляді:

$$E = \frac{1}{2} \sum_{k=1}^N [y_k - d_k]^2 = \frac{1}{2} \sum_{k=1}^N \left[\sum_{i=0}^h w_i f(x_k) - d_k \right]^2, \quad (2.3)$$

де d_k – очікуваний вихідний сигнал нейромережі.

В результаті проведених досліджень виявлено деякі недоліки класичного градієнтного алгоритму навчання РБНС:

– в алгоритмі навчання РБНС немає правил для початкового завдання кількості нейронів мережі та їх параметрів, а також немає правил зміни кількості нейронів у процесі навчання. Рівномірний розподіл нейронів у робочій області не завжди є оптимальним. Також може виникнути ситуація, коли кількість нейронів, задана спочатку, є недостатньою для досягнення необхідної якості навчання.

– у процесі навчання змінюються параметри всіх нейронів мережі. У результаті при збільшенні кількості нейронів обчислювальні витрати на навчання також збільшуються.

– РБНС не може досягти сталого стану в процесі навчання у випадках, коли існують нейрони з близькими значеннями параметрів центрів та ширини базисної функції. Поява подібних ситуацій багато в чому залежить від обраної кількості нейронів та їх початкових параметрів. Причина погіршення якості навчання полягає в тому, що в градієнтному алгоритмі передбачається, що на вихідне значення РБНС у кожній точці робочої області в основному впливає лише один нейрон. За наявності кількох нейронів в одній ділянці робочої області зміна їх параметрів відповідно до градієнтного алгоритму не завжди призводить до зменшення помилки навчання.

З метою визначення ситуацій, коли кілька деяких нейронів стають близькими один до одного, було введено поняття коефіцієнта взаємного перетину нейронів. Для обчислення цього коефіцієнта для деякого нейрона РБНС необхідно знайти другий нейрон, центр якого розташований найближче до центру нейрона, що розглядається. Значення коефіцієнта взаємного перетину визначається як сума вихідного значення поточного нейрона в центрі другого нейрона та вихідного значення другого нейрона в центрі поточного нейрона:

$$p_i = \exp\left(-\frac{\sum_{j=1}^n (c_{ij} - c_{dj})^2}{2\sigma_i^2}\right) + \exp\left(-\frac{\sum_{j=1}^n (x_j - c_{ij})^2}{2\sigma_d^2}\right), \quad (2.4)$$

де i – номер нейрона, для котрого обчислюється значення коефіцієнта взаємного перетину;

d – номер нейрона, центр якого розташований ближче до центру нейрона з номером i .

Номер нейрона d визначається за формулою:

$$d = \arg \min_k \sqrt{\sum_{j=1}^n (c_{ij} - c_{kj})^2}.$$

Значення коефіцієнта взаємного перетину знаходиться в інтервалі $(0; 2]$. Коефіцієнт приймає максимальне значення в тому випадку, коли центри нейронів, що розглядаються, збігаються. В ході експериментів з апроксимації різних функцій за допомогою РБНС було визначено, що помилка РБНС починає збільшуватися у випадку, якщо значення коефіцієнта взаємного перетину перевищує 1,9, тому для досягнення максимальної якості навчання РБНС необхідно обмежити максимальне значення коефіцієнта взаємного перетину величиною 1,9.

З метою виключення недоліків класичного градієнтного алгоритму навчання РБНС розроблено алгоритм синтезу радіально-базисних нейромереж, блок-схему якого показано на рисунку 2.14. Варто зауважити, що такий алгоритм навчання об'єднує процедури налаштування параметрів прихованого та вихідного шару.

Основні відмінності розробленого алгоритму від класичного алгоритму полягають у:

- додано правила зміни структури РБНС у процесі навчання (блок 2). На початку навчання РБНС не містить нейронів. По мірі необхідності нові нейрони додаються, а нейрони, що не використовуються, видаляються;
- зменшено обчислювальні витрати, які затрачаються для кожного циклу навчання. Це досягається зміною параметрів не всіх нейронів, як у

класичному алгоритмі, а тільки нейронів, вихідний розмір яких у розглянутій точці більший, ніж величина $\Theta_{зм}$ (блоки 4 і 5).

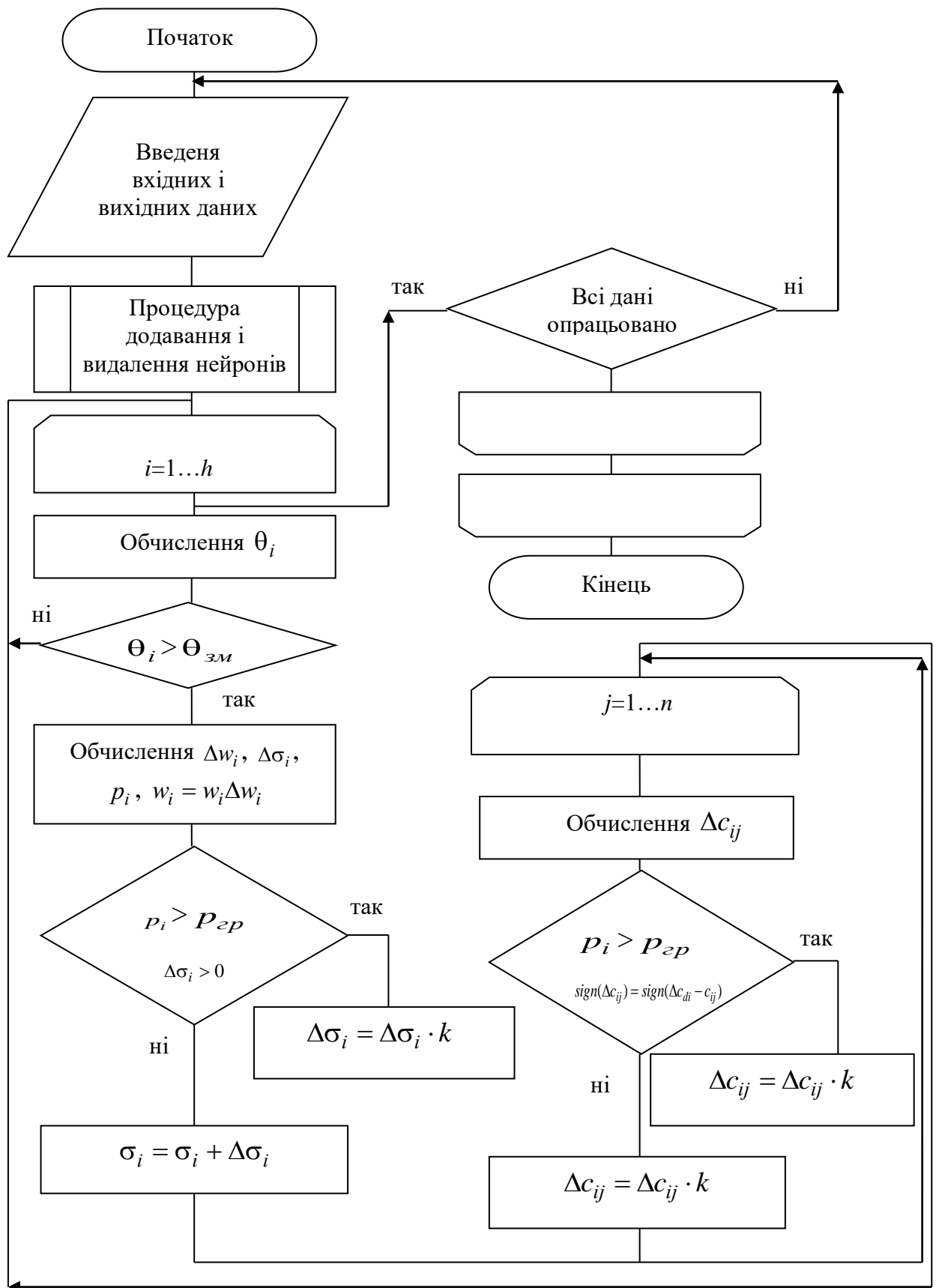


Рисунок 2.14 – Блок-схема алгоритму синтезу радіально-базисних нейромереж

– виключено можливість виникнення ситуації, коли параметри деяких нейронів практично збігаються. Для цього обчислені величини Δc_{ij} і $\Delta \sigma_j$ зменшуються, якщо коефіцієнт взаємного перетину нейронів перевищує граничну величину P_{ep} , рівну 1,9 (блоки 7, 8, 12, 13).

Також на основі результатів аналізу діалогових вікон пакету Neural Network Toolbox, опишемо алгоритм моделювання нейромереж штучного типу із застосуванням цього інструменту.

Крок 1. Аналіз поставленої задачі, її вхідних та вихідних характеристик.

Крок 2. Розробка проекту визначеної архітектури нейромережі із зазначенням кількості нейронів у прихованому шарі.

Крок 3. Збереження вхідних та вихідних даних, необхідних для побудови нейромережі у файл.

Крок 4. Задання векторів вхідних та вихідних даних у діалоговому вікні Neural Network Toolbox або імпортування їх із файлу.

Крок 5. Розробка архітектури нейромережі із заданням кількості нейронів у вхідному, прихованому та вихідному шарі, функції активації, алгоритму навчання.

Крок 6. Навчання розроблену архітектуру нейромережі.

Крок 7. Якщо похибка навчання нейромережі сягає деякого наперед заданого мінімального значення, то переходимо на крок 7. У протилежному випадку – переходимо на крок 5.

Крок 7. Моделюємо розроблену структуру нейромережі.

Крок 8. Тестуємо архітектуру нейромережі на тестовій вибірці експериментальних даних.

Крок 9. Якщо на тестовій вибірці даних мережа продуктивна, то завершуємо алгоритм. У протилежному випадку повертаємося на крок 7.

Із описаних кроків алгоритму моделювання нейромереж із застосуванням середовища NNTool видно, що алгоритм є доволі простий у реалізації та

фактично зводиться до правильного підбору кількості нейронів у прихованому шарі. Блок-схему алгоритму моделювання зображено на рисунку 2.15.



Рисунок 2.15 – Блок схема алгоритму моделювання архітектури нейромережі

Розроблена блок-схема алгоритму демонструє усі кроки, які необхідно виконати для реалізації алгоритму із застосуванням середовища для моделювання архітектур нейромереж штучного типу.

2.4 Висновки до розділу

Описано алгоритми та засоби ідентифікації параметрів нейромереж штучного типу. Зазначено, що вони ґрунтуються на типі між нейронних зв'язків. Охарактеризовано середовище для моделювання архітектур нейромереж штучного типу Neural Network Toolbox, зокрема, основні задачі моделювання нейромережі, вікна аналізу параметрів, розробки на симуляції архітектури, тестування та аналіз показників продуктивності. Розроблено алгоритм синтезу радіально-базисних нейромереж, який включає процедури налаштування параметрів радіального та вихідного шару. Побудовано алгоритм реалізації методу ідентифікації структури мережі у середовищі NNTool.

3 РЕАЛІЗАЦІЯ АЛГОРИТМУ НАВЧАННЯ РАДІАЛЬНО-БАЗИСНИХ НЕЙРОМЕРЕЖ

3.1 Структура програмної системи

Для реалізації розробленого алгоритму синтезу нейромереж штучного типу розроблено програмне забезпечення. На основі аналізу вимог побудовано діаграму використання розробленого програмного забезпечення, яку проілюстровано на рисунку 3.1.

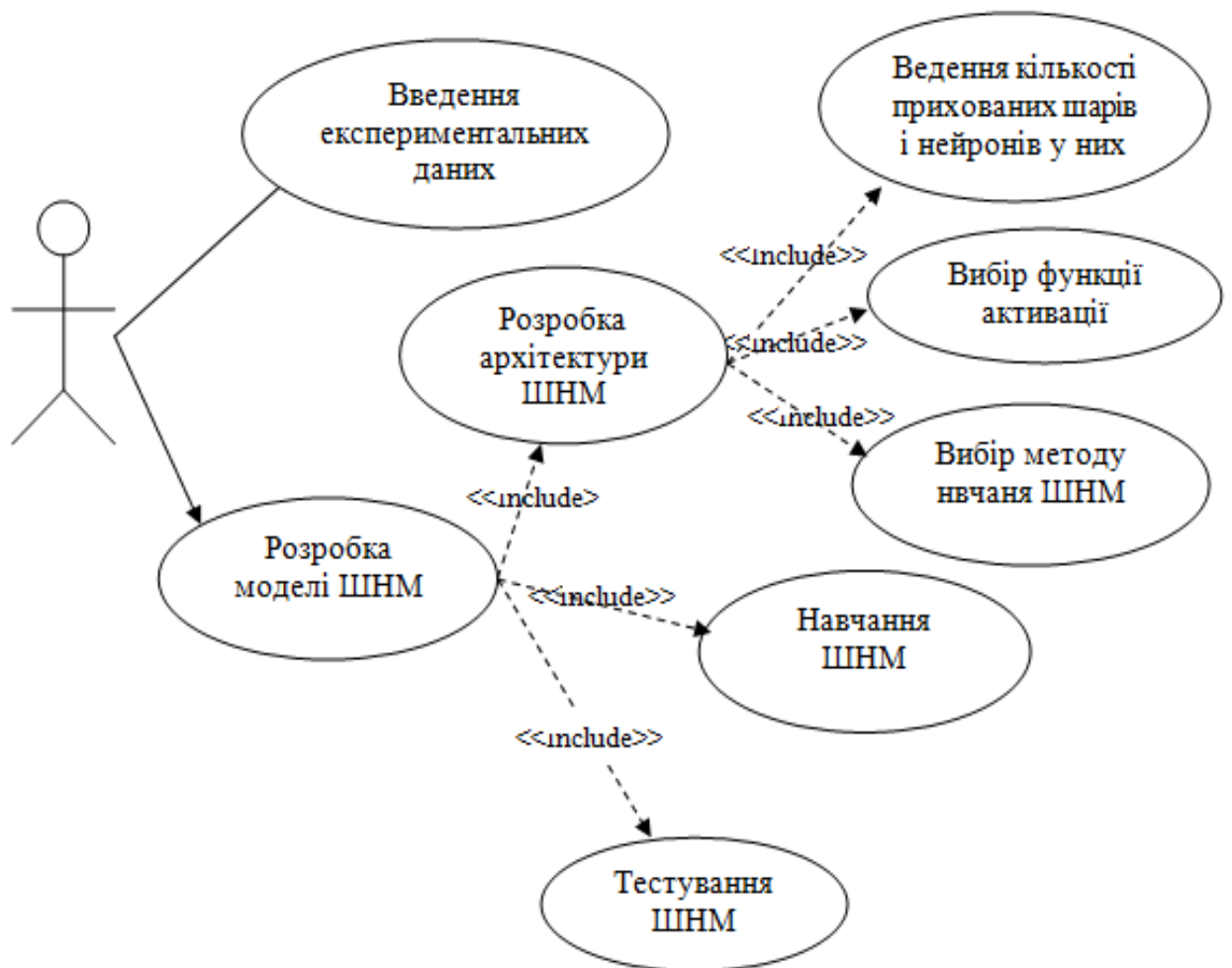


Рисунок 3.1 – Діаграма варіантів використання програмного забезпечення

Адміністратор системи, у ролі якого може бути будь-яка особа із обмеженим доступом до самої системи, як бачимо на рисунку 3.1, має

можливість вводити експериментальні дані та розробляти моделі нейромереж штучного типу.

Функція «Розробка моделі ШНМ» включає три підфункції «Розробка архітектури ШНМ», «Навчання ШНМ» та «Тестування ШНМ». У той же час підфункція «Розробка архітектури ШНМ» вимагає введення кількості прихованих шарів нейромережі та кількості нейронів у кожному прихованому шарі, вибір функції активації та вибір методу навчання нейромережі.

Результатом реалізації підфункції «Розробка архітектури ШНМ» є архітектура нейромережі із зазначенням кількості нейронів вхідного рівня, структури прихованого рівня та кількості нейронів вихідного рівня.

Підфункція «Навчання ШНМ» забезпечує навчання архітектури нейромережі на основі методу навчання, який ґрунтується на мінімізації похибки навчання. У результаті реалізації зазначеної підфункції отримуємо графік навчання нейромережі із зазначенням того, як змінюється похибка та стани навчання. За цими графіками можна робити висновки щодо продуктивності нейромережі. Також у результаті нейромережа штучного типу налаштовує синаптичні ваги, значення яких можна побачити у діалоговому вікні.

Підфункція «Тестування ШНМ» уможливорює проведення тестування вже навченої архітектури нейромережі на тестовій вибірці даних. Перевірочна вибірка даних – це експериментальні дані, які не приймали участі у навчанні. Результатом реалізації зазначеної підфункції програмної системи є графіки отриманих вихідних сигналів на основі нейромережі та бажані виходи, які потрібно отримати у результаті розв'язування поставленої задачі. Якщо відносна похибка відхилення прогнозного значення нейромережі від експериментального перевищує допустиме значення, то архітектура розробленої нейромережі штучного типу є не придатною для розв'язування поставленої задачі.

Також варто зазначити, що розроблене програмне забезпечення включає можливість імпортування експериментальних даних у файл із будь-якої області

робочого стола, а також експортування даних для збереження у потрібному місці.

Можливість збереження розробленої архітектури передбачено самим пакетом Neural Network Toolbox, тому не доцільно створювати функцію, яка б це забезпечувала.

У наступному підрозділі розглянемо детально особливості пакету прикладних програм Matlab, оскільки саме це середовище розробки математичних моделей використано для реалізації розробленого алгоритму.

3.2 Моделювання нейромереж із застосуванням NNTool

Matlab розшифровується як матрична лабораторія (matrix laboratory), яка спеціально розроблена для доступу до LINPACK і EISPACK, які призначені для матричних обчислень. Сьогодні середовище Matlab вийшло за межі спеціалізованої матричної системи і стало однією із найбільш потужних універсальних інтегрованих систем комп'ютерного моделювання.

Matlab являє стандартний інструмент для розробки моделей в різних областях. Можливості середовища Matlab досить широкі, по швидкості виконання задач суттєво перевершує конкурентів.

Важливу роль в середовищі Matlab відводиться спеціальним групам програм – тулбоксам, що мають суттєву значимість для більшості користувачів. Toolboxes – це набір функцій Matlab, які дозволяють вирішувати різні класи задач. Matlab включає п'ять частин [6, 11].

1. Мова Matlab – це мова масивів і матриць високого рівня з керованими потоками, функціями і особливостями об'єктно-орієнтованого програмування, що уможливорює створювати як прості програми, так і програми для вирішення складніших задач.

2. Середовище Matlab – це набір пристроїв та інструментів, які використовує користувач. Він включає засоби для управління змінними в робочій області Matlab, введенням і виведенням даних, а також створення і налагодження m-файлів.

3. Керована графіка – це графічна система Matlab, яка включає команди для візуалізації дво- і тривимірної графіки. Варто зазначити, що наявні команди, за допомогою яких можна редагувати зовнішній вигляд графіки і створювати графічний інтерфейс користувача для додатків Matlab.

4. Бібліотека математичних функцій – це набір обчислювальних елементів від елементарних функцій до якзавгодно складних.

5. Програмний інтерфейс – це бібліотека, за допомогою якої можна писати програми на Сі, які можуть взаємодіяти з Matlab. Вона включає засоби для виклику програм з Matlab та роботи із m-файлами.

Головне вікно середовища для математичного моделювання представлено на рисунку 3.2.

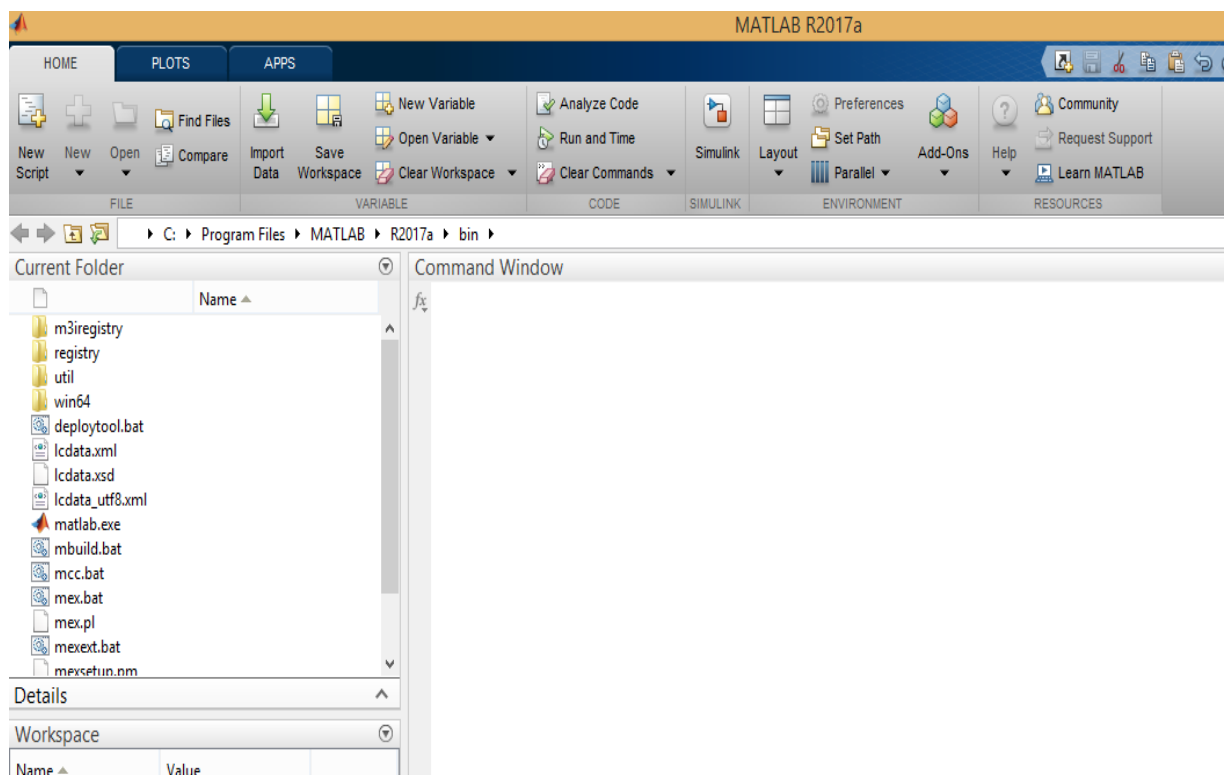


Рисунок 3.2 – Екранна форма частини головного вікна середовища Matlab

Для моделювання нейромереж із застосуванням середовища Matlab існує тулбокс Neural Network Toolbox, який викликається із командного рядка функцією `nntool`. Розглянемо основні команди, які використовуються в пакеті Neural Network Toolbox для моделювання нейронних мереж, а також адаптації та навчання.

Основні команди для розробки архітектур ШНМ:

- `network` – створення шаблону нейромережі;
- `newp` – розробка персептрона;
- `newlin` – розробка лінійного шару;
- `newlind` – розробка лінійного шару шляхом розширення лінійних рівнянь;
- `newff` – розробка мережі прямої передачі інформації;
- `newfftd` – розробка мережі прямої передачі із затримками;
- `newscf` – створення каскадні мережі прямої передачі інформації;
- `newrb` – розробка радіально-базисної штуної нейронної мережі;
- `newrbe` – розробка радіально-базисної нейромережі із нульовою похибкою;
- `newrpn` – розробка імовірнісної нейромережі;
- `newgrnn` – розробка узагальненої регресійної нейромережі;
- `newlvq` – розробка архітектури для задач класифікації;
- `newsom` – розробка самоорганізуючих мереж Кахонена;
- `newhop` – розробка мережі Хопфілда;
- `newelm` – розробка мережі Елмана.

Функції, що використовуються для навчання та адаптації нейромереж штучного типу:

- `adapt` – адаптація параметрів мережі;
- `adaptwb` – встановлення режиму адаптації;
- `train` – навчання мережі;
- `trainb` – групове навчання;

- `trainbfg` – алгоритм навчання Бройтона, Флетчера;
- `trainbr` – алгоритм навчання Бейєса;
- `traincgb` – алгоритм навчання на основі спряженого градієнта;
- `trainc` – навчання із циклічним представленням вхідних сигналів;
- `traingd` – метод градієнтного спуску;
- `trainr` – навчання у режимі випадкового представлення вхідних сигналів;
- `trainrp` – пороговий алгоритм зворотного поширення помилки;
- `trains` – адаптивне навчання із послідовним представленням вхідних сигналів;
- `trainscg` – алгоритм навчання SCG.

Для моделювання радіально-базисної нейромережі у Neural Network Toolbox наявний зручний та зрозумілий графічний інтерфейс, який включає усі кроки алгоритму.

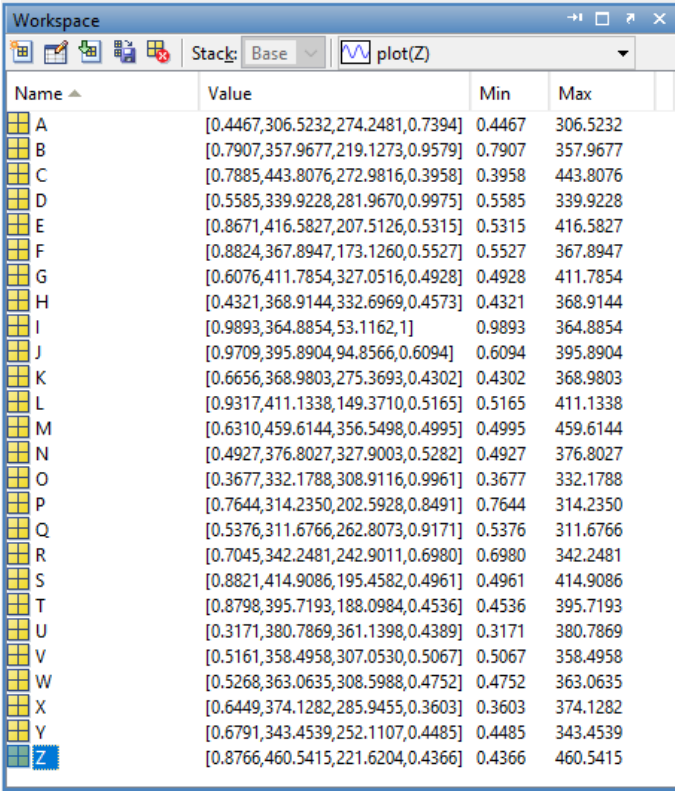
3.3 Експериментальні дослідження ефективності алгоритму

Для дослідження ефективності розробленого алгоритму моделювання архітектури нейромережі на основі Neural Network Toolbox розв'яжемо прикладну задачу розпізнавання літер англійського алфавіту [3, 8, 30, 40], які мають кут повороту або нахил.

Перш, ніж розпочати моделювання радіально-базисної нейромережі для розпізнавання літер, необхідно створити скрипти для кожної букви з метою представлення їх як характерні ознаки, які притаманні кожній літері [49, 50]. Для цього використовуємо функцію `regionprops` Image Processing Toolbox середовища Matlab [23, 49].

Для кожної літери створюємо `m`-файл, за допомогою якого розробляються скрипти. Отримуємо набір змінних, які характеризують кожну літеру. Оскільки

у англійському алфавіті 26 літер, то й у результаті отримуємо набір із 26 змінних, що проілюстровано на рисунку 3.3.



Name	Value	Min	Max
A	[0.4467,306.5232,274.2481,0.7394]	0.4467	306.5232
B	[0.7907,357.9677,219.1273,0.9579]	0.7907	357.9677
C	[0.7885,443.8076,272.9816,0.3958]	0.3958	443.8076
D	[0.5585,339.9228,281.9670,0.9975]	0.5585	339.9228
E	[0.8671,416.5827,207.5126,0.5315]	0.5315	416.5827
F	[0.8824,367.8947,173.1260,0.5527]	0.5527	367.8947
G	[0.6076,411.7854,327.0516,0.4928]	0.4928	411.7854
H	[0.4321,368.9144,332.6969,0.4573]	0.4321	368.9144
I	[0.9893,364.8854,53.1162,1]	0.9893	364.8854
J	[0.9709,395.8904,94.8566,0.6094]	0.6094	395.8904
K	[0.6656,368.9803,275.3693,0.4302]	0.4302	368.9803
L	[0.9317,411.1338,149.3710,0.5165]	0.5165	411.1338
M	[0.6310,459.6144,356.5498,0.4995]	0.4995	459.6144
N	[0.4927,376.8027,327.9003,0.5282]	0.4927	376.8027
O	[0.3677,332.1788,308.9116,0.9961]	0.3677	332.1788
P	[0.7644,314.2350,202.5928,0.8491]	0.7644	314.2350
Q	[0.5376,311.6766,262.8073,0.9171]	0.5376	311.6766
R	[0.7045,342.2481,242.9011,0.6980]	0.6980	342.2481
S	[0.8821,414.9086,195.4582,0.4961]	0.4961	414.9086
T	[0.8798,395.7193,188.0984,0.4536]	0.4536	395.7193
U	[0.3171,380.7869,361.1398,0.4389]	0.3171	380.7869
V	[0.5161,358.4958,307.0530,0.5067]	0.5067	358.4958
W	[0.5268,363.0635,308.5988,0.4752]	0.4752	363.0635
X	[0.6449,374.1282,285.9455,0.3603]	0.3603	374.1282
Y	[0.6791,343.4539,252.1107,0.4485]	0.4485	343.4539
Z	[0.8766,460.5415,221.6204,0.4366]	0.4366	460.5415

Рисунок 3.3. – Екранна форма діалогового вікна із набором змінних

Тепер переходимо до моделювання архітектури радіально-базисної нейромережі для розпізнавання літер. На початковому етапі задаємо кількість вхідних та вихідних змінних. Екранну форму діалогового вікна заданих експериментальних даних задачі проілюстровано на рисунку 3.4.

На наступному кроці необхідно порівняємо результати роботи нейромережі перцептронного типу та радіально-базисної нейромережі. На виході архітектура нейромережі має 26 вихідних сигнали. Визначаємо кількість прихованих шарів нейромережі та нейронів у ній.

У результаті проведених експериментів для розпізнавання літер обираємо архітектуру радіально-базисної нейромережі із 15 нейронами у прихованому шарі (див. рис. 3.5).

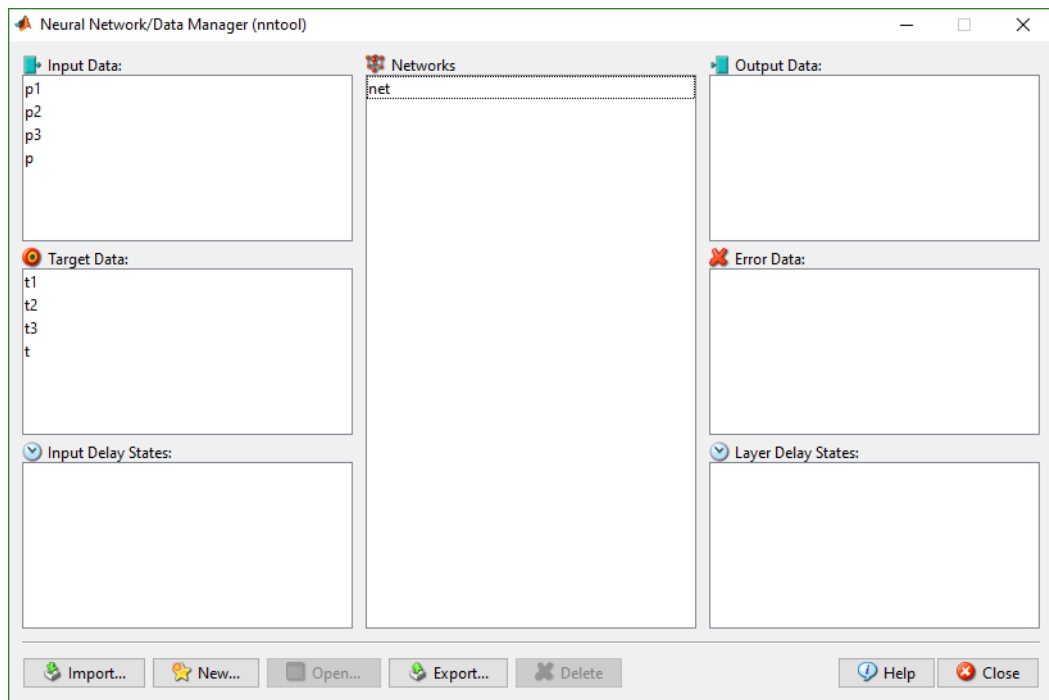


Рисунок 3.4 – Екранна форма демонстрації заданих вхідних та вихідних даних

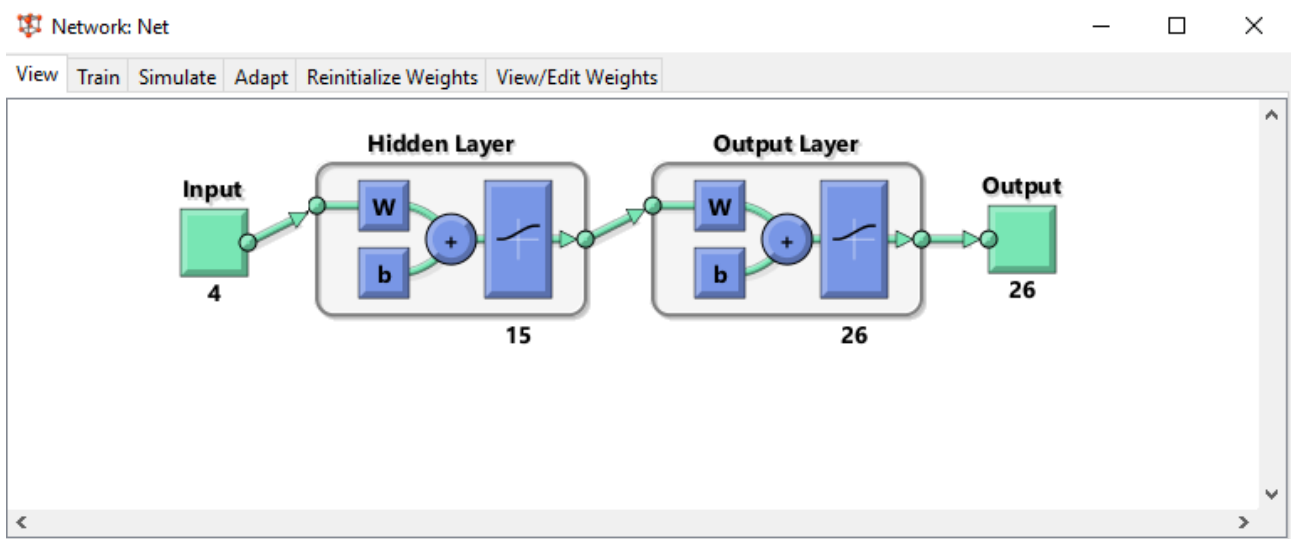


Рисунок 3.5 – Архітектура розробленої радіально-базисної нейромережі

Графік залежності похибки навчання радіально-базисної нейромережі від кількості нейронів у прихованому шарі наведено на рисунку 3.6. Як бачимо, нульова похибка навчання досягається при 15 та 25 нейронів у прихованому шарі. Найбільша похибка навчання мережі тоді, коли у прихованому шарі 3 нейрони.

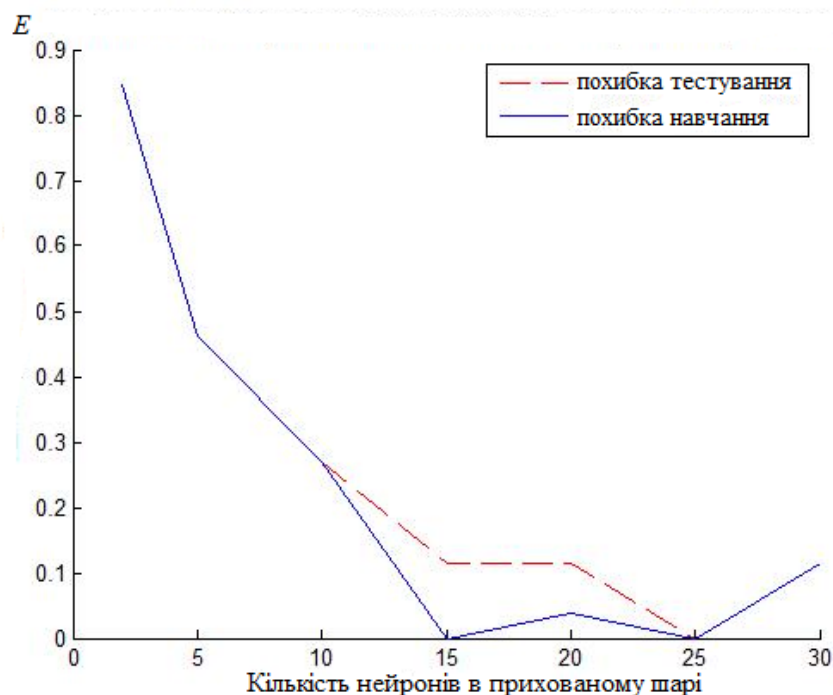


Рисунок 3.6 – Залежність похибки навчання нейромережі від кількості нейронів

На кроці навчання нейромережі обираємо параметри для навчання та вчимо нейромережу із застосуванням розробленого алгоритму. Тестування навченої архітектури радіально-базисної нейромережі проводимо на основі контрольної вибірки експериментальних даних. Графік продуктивності розробленої нейромережі із зазначенням графіків навчання, тестування та валідації мережі проілюстровано на рисунку 3.7. Як бачимо, радіально-базисна нейромережа вчиться 279 епох для досягнення найкращого результату моделювання.

Для порівняння, розглянемо ілюстрацію непродуктивної структури нейромережі перцептронного типу для задачі розпізнавання літер англійського алфавіту. Зазначений випадок представлено на рисунку 3.8. На перший погляд така архітектура вимагає менше обчислювальних ресурсів, оскільки вчиться лише 7 епох, проте помилка розпізнавання становить 0,81 (81%), а графіки тренування, валідації та тестування мережі зовсім не збігаються. Зазначений результат підтверджує перспективність застосування радіально-базисних нейромереж для розв'язування задач такого типу.

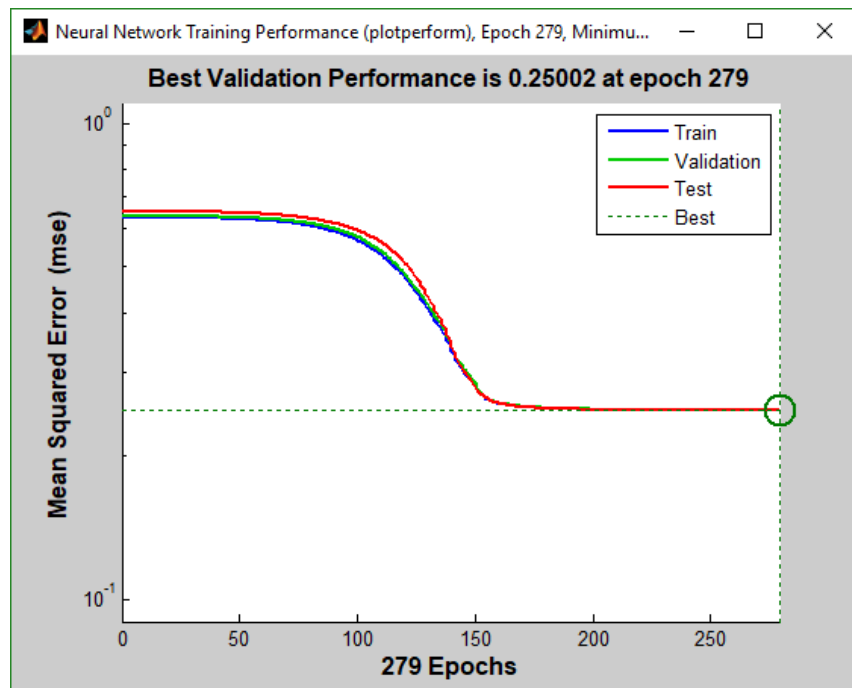


Рисунок 3.7 – Графіки продуктивності розробленої архітектури нейромережі

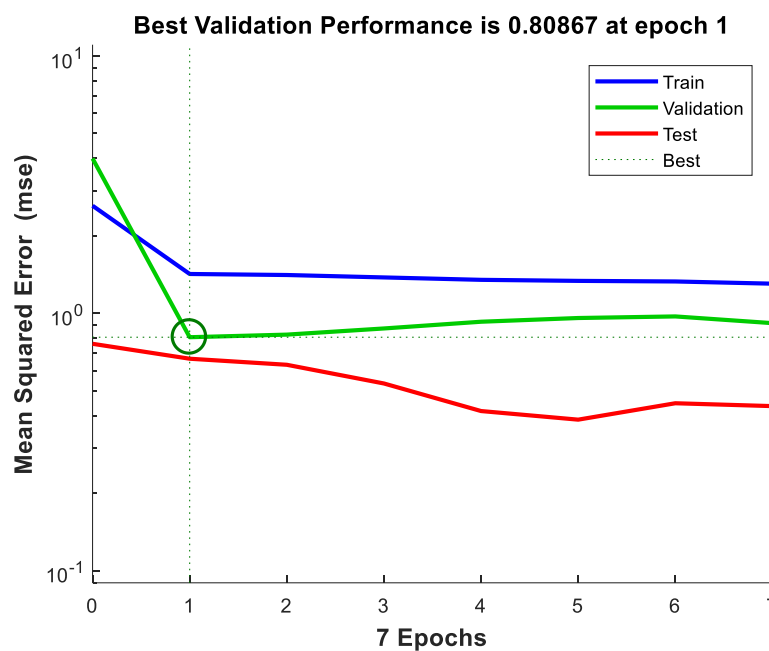


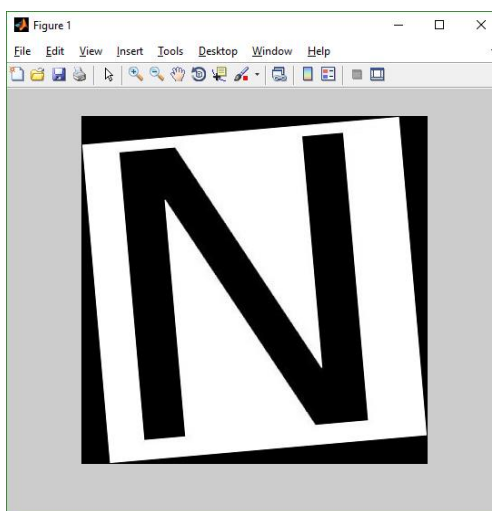
Рисунок 3.8 – Графіки продуктивності неадекватної архітектури нейромережі

Для демонстрації роботи мережі символи “А”, “Н” і “G” повернемо на +5, -10, +3 градусів. Для повороту застосуємо функцію `imrotate (g, 5, 'bicubic')`. Результати занесемо в таблицю 3.1.

Таблиця 3.1 – Вектори-позначення літер

Кут повороту літери	A	H	G
0	[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]	[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]	[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
+5	[1.35 0 0 0 0 0 0 0 0.51 0 0 0 0 0 0 0 0 0 0.57 0 0 0 0 0 0 0 0]	[0 0 0 0 0 0 0 1.36 0 0 0 0 0.31 0 0 0 0.21 0 0 0 0 0 0 0 0]	[0 0.32 0 0 0 0 1.25 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
-10	[1.82 0 0 0 0 0 0 0 0 0 0 0 0 0.45 0 0 0 0 0.33 0 0 0 0 0 0]	[0 0 0 0 0 0 0 1.68 0 0 0 0 0 0 0 0.77 0 0 0 0 0 0.39 0 0 0]	[0 0 0 0 0 0 1.11 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.69 0]
+3	[1.44 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.36 0 0 0.58 0 0 0 0 0 0 0]	[0.33 0 0 0 0 0 0 0 1.70 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]	[0 0 0 0 0 0 1.67 0 0 0 0 0 0 0.5 0 0 0 0 0 0 0 0 0 0 0 0]

Як приклад, розпізнаємо літеру англійського алфавіту N із кутом нахилу. Для моделювання розробленої архітектури радіально-базисної нейромережі розпізнавання літер використовуємо функцію *sim*. У результаті моделювання виводиться зображення цієї букви, що проілюстровано на рисунку 3.9 а) та номер, присвоєний букві у результаті навчання нейромережі, як показано на рисунку 3.9 б).



а)

```
Command Window
>> y=sim(Net, N5)

y =

    0
    0.5700
    0
    0
    0
    0
    0
    0
    0
    0
    0
    0
    0
    0
    0
    1.5500
    0
    0
    0
    0
    0
    0
    0
    0
    0
    0
    0.2300
    0
    0

fx >>
```

б)

Рисунок 3.9 – Результат розпізнавання літери англійського алфавіту

Із результатів розпізнавання видно, що побудована архітектура радіально-базисної нейромережі на основі розроблених у роботі алгоритмів синтезу є придатною для розпізнавання літер, при чому достовірність розпізнавання становить близько 95% на основі 26 прикладів.

3.4 Висновки до розділу

Розроблено програмну систему для ідентифікації радіально-базисних нейромереж на основі алгоритму синтезу параметрів прихованого та вихідного

шару. Побудовано діаграму варіантів використання програмної системи на основі аналізу вимог. Описано процедуру моделювання архітектури нейромереж штучного типу на основі NNTool, основні функції системи, можливості та параметри. Розроблено архітектуру радіально-базисної нейромережі для розпізнавання літер. Результатами експериментів підтверджено ефективність такої структури нейромережі.

ВИСНОВКИ

1. Проаналізовано особливості нейромереж й зазначено, що основним будівельним інструментом їх архітектури є штучний нейрон.
2. Проаналізовано відомі архітектури нейромереж й зазначено, що вони поділяються на топології із прямою передачею інформації та зворотною. Відзначено перспективність радіально-базисних нейромереж.
3. Проаналізовано програмні засоби моделювання штучних нейромереж й зазначено, що вони налаштовані в основному на розв'язування вузького кола задач.
4. Проаналізовано пакет для моделювання нейромереж середовища Matlab Neural Network Toolbox й зазначено, що він характеризується зручним графічним інтерфейсом користувача, великою кількістю вбудованих функцій та алгоритмів навчання ШНМ.
5. Розроблено алгоритм синтезу радіально-базисних нейромереж на основі поєднання процедур налаштування центрів базисних функцій та вагових коефіцієнтів.
6. Розроблено алгоритм моделювання структури нейромереж на основі Neural Network Toolbox.
7. Розроблено програмне забезпечення для реалізації алгоритмів та діаграму варіантів його використання.
8. На основі розроблених алгоритмів побудовано архітектуру радіально-базисної нейромережі для задачі розпізнавання літер англійського алфавіту. Результатами моделювання підтверджено ефективність розробленого алгоритму.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бодянский Е. В., О. Г. Руденко Искусственные нейронные сети: архитектуры, обучение, применения: Монография. Харьков: ТЕЛЕТЕХ, 2004. 372 с.
2. Бодянский Е. В., Горшков Е. В., Колодяжный В.В. и др. Рекуррентный алгоритм обучения радиально-базисных нейронных сетей, основанный на приближительных множествах. Радіоелектроніка. Інформатика. Управління. 2005. № 1. С.116-122.
3. Буханцева С. К. Разработка алгоритма распознавания рукописных символов на основе растровой графики: веб-сайт. URL: <https://cyberleninka.ru/article/n/razrabotka-algoritma-raspoznavaniya-rukopisnyh-simvolov-na-osnove-rastrovoy-grafiki> (дата звернення 22.10.2022).
4. Васенков Д.О. Методы обучений искусственных нейронных сетей: веб-сайт. URL: <http://www.ipospb.ru/journal/content/> (дата звернення 10.10.2022).
5. Гонсалес. Р., Вудс Р. Цифровая обработка изображений М.: Техносфера, 2006. 1072 с.
6. Дьяконов В. П. MATLAB. Полный самоучитель. М.: ДМК Пресс, 2012. 768 с.
7. Забара С. Моделювання систем у середовищі MATLAB. Университет "Україна", 2011. 137 с.
8. Заяць І.П., Крамар В.В. Методи розпізнавання медичних зображень // Матеріали V науково-практичної конференції молодих вчених та студентів «Інтелектуальні комп'ютерні системи та мережі». Тернопіль, 2021. С 15.
9. Колесников С. Распознавание образов. Общие сведения: веб-сайт. URL:<http://www.ci.ru/> (дата звернення 22.09.2022).

10. Куссуль М.Э., Сычев А.С. Нейросетевой классификатор для систем безопасности автомобиля // Математичні машини і системи. 2004. № 2. С. 15-21.
11. Лазарев Ю. Ф. Начала программирования в среде MatLAB: Учебное пособие. Киев: НТУУ "КПИ", 2003. 424 с.
12. Машинное обучение. нейронные сети (часть 1): процесс обучения перцептрона: веб-сайт. URL: <https://habr.com/ru/post/516458/> (дата звернення 24.11.2021).
13. Медведев В.С, Потемкин В.Г. Нейронные сети. MATLAB 6. М.: ДИАЛОГ-МИФИ, 2002. 496 с.
14. Методичні рекомендації до виконання дипломної роботи з освітньокваліфікаційного рівня «Магістр». Спеціальність «Комп'ютерна інженерія» / О.М. Березький, Л.О. Дубчак. Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2020. 47 с.
15. 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.
16. Методы распознавания образов. Часть 1: веб-сайт. URL: <https://oxozle.com/2015/03/29/metody-raspoznavaniya-obrazov-chast-1/> (дата звернення 23.10.2022).
17. Лазарев Ю. Моделирование процессов и систем в MATLAB: Учебный курс. СПб.: Питер; Киев: Изд. группа BHV, 2005. С. 474 с.
18. Лукин А. Введение в цифровую обработку сигналов. М. : Лаборатория компьютерной графики и мультимедиа, МГУ, 2002. 44 с.
19. Любунь З.З. Основы теорії нейромереж. львів: видавничий центр ЛНУ ім. Івана Франка, 2006. 140 с.

20. Тимошук П.В., Бренич Я.В. Нейромережеві методи розв'язання задачі класифікації. Науковий вісник НЛТУ України. 2012. Вип. 22.13. С. 343-349.
21. Нестерук Ф. Г., Котенко И. В. Инструментальные средства создания нейросетевых компонент интеллектуальных систем защиты информации. Тр.:СПИИРАН, 2013. Вып. 26. С. 7-25.
22. Новотарський М.А., Нестеренко Б.Б. Штучні нейронні мережі: обчислення. Праці Інституту математики НАН України. Київ: Ін-т математики НАН України, 2004. Т50. 408 с.
23. Проект ImageTextEditor: веб-сайт. URL: <http://imated.sourceforge.net> (дата звернення 18.08.2021).
24. Рудик О.А., Крамар В.В. Радіально-базисні нейронні мережі для прогнозування випадкових процесів // Матеріали V науково-практичної конференції молодих вчених та студентів «Інтелектуальні комп'ютерні системи та мережі». Тернопіль, 2021. С 30.
25. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. Москва: Телеком, 2006. 452 с.
26. Стокман Д., Шапиро Л. Компьютерное зрение. М.: Бином. Лаборатория знаний, 2006. 752 с.
27. Рутковская Д., Пилиньский М. Нейронные сети, генетические алгоритмы и нечеткие системы. Москва: Телеком, 2006. 652 с.
28. Савка Н.Я., Складанюк В.М. Метод синтезу радіально-базисних нейромереж // Матеріали ІХ Всеукраїнської науково-практичної конференції молодих науковців. Київ, 2022. С.158-159.
29. Стефаник В.А., Складанюк В.М., Климчук С. Б. Методи оцінки прямих та непрямих вигод від розробки програмного продукту // Матеріали VI науково-практичної конференції молодих вчених та студентів «Інтелектуальні комп'ютерні системи та мережі». Тернопіль, 2022. С. 16.
30. Фисенко В.Т., Фисенко Т.Ю. Компьютерная обработка и распознавание изображений : учеб. Пособие. СПб. : СПбГУ ИТМО, 2008. 192 с.

31. Хайкин С. Нейронные сети: полный курс. Москва: Вильямс, 2006. 356 с.
32. Шапиро Л., Стокман Дж. Компьютерное зрение. М.: Бином. Лаборатория знаний, 2006. 752 с.
33. Штовба С.Д. Введение в теорию нечетких множеств и нечеткую логику: веб-сайт. URL: <http://matlab.exponenta.ru/fuzzylogic/book1/index.php> (дата звернения 18.11.2021).
34. Ясницкий Л.Н. Введение в искусственный интеллект. Москва: «Академия», 2008. 176 с.
35. Яхьяева Г. Э. Нечеткие множества и нейронные сети. Москва: Бином, 2006. 316 с.
36. Datong Chen, Jean-Marc Odobez, Herve Boulard Text detection and recognition in images and video frames // Pattern Recognition journal, 2003. P. 595–608.
37. Doetsch P., Kozielski M., Ney H. Fast and robust training of recurrent neural networks for offline handwriting recognition // 14th International Conference on Frontiers in Handwriting Recognition. 2014. №14. P. 279–284.
38. Forsyth D. A., Ponce J. Computer Vision: A Modern Approach. Pearson Education, Inc., 2011. 792 p.
39. Jaderberg M., Simonyan M. Synthetic data and artificial neural networks for natural scene text recognition. NIPS Deep Learning Workshop, 2014. P. 1-10.
40. MATLAB&Toolboxes: веб-сайт. URL: <http://matlab.exponenta.ru> (дата звернения 02.11.2022).
41. Nelles O. Nonlinear Systems Identification. –Berlin: Springer, 2001. – 785p.
42. Paramud Y., Yarkun V. Algorithmic and software means of handwritten symbol recognition // Bulletin of the National University “Lviv Polytechnic” “Computer Systems and Networks” No 881. Lviv, Ukraine. 2017. P. 98–106.