

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

_____ Савченко А.С.

“ _____ ” _____ 2020 р.

ДИПЛОМНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
“МАГІСТРА”

ЗА СПЕЦІАЛІЗАЦІЮ “ІНФОРМАЦІЙНІ УПРАВЛЯЮЧІ СИСТЕМИ
ТА ТЕХНОЛОГІЇ (ЗА ГАЛУЗЯМИ)”

Тема: “Алгоритми дослідження нейронної мережі”

Виконавець: Паренюк Леонід Володимирович

Керівник: д.т.н., професор Воронін Альберт Миколайович

Нормоконтролер: _____ Райчев І.Е.

Київ 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, спеціалізація: 12 “Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології (за галузями)” .

ЗАТВЕРДЖУЮ

Завідувач кафедри

Савченко А.С.

“ _____ ” _____ 2020 р

ЗАВДАННЯ

на виконання дипломної роботи студента

Паренюка Леоніда Володимировича

(прізвище, ім'я, по батькові)

- 1. Тема проекту:** «Алгоритми дослідження нейронних мереж» затверджена наказом ректора №1891/ст від 02.10.2020
- 2. Термін виконання роботи:** з 05.10.2020 по 22.12.2020р.
- 3. Вихідні дані до роботи:** розробка нейронної мережі для прогнозування ризиків виходу з ладу компонентів системи.
- 4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):** вступ, аналітичний огляд і постановка завдання, розробка інформаційної системи, загальна характеристика системи, висновок.
- 5. Перелік обов'язкового графічного матеріалу:** таблиці, рисунки, результати роботи програми, а також слайди презентації доповіді у PowerPoint.

6. Календарний план-графік

<i>№ з/п</i>	<i>Завдання</i>	<i>Термін виконання</i>	<i>Підпис керівника</i>
1.	Аналіз літератури та джерел за темою дипломного проекту.	5.10.2020р – 7.10.2020р.	
2.	Розроблення та затвердження плану дипломного проекту.	8.10.2020р – 9.10.2020р.	
3.	Приведення консультації з науковим керівником щодо створення першого розділу.	10.10.2020р	
4.	Розробка розділу 1: АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11.10.2020р – 20.10.2020р.	
5.	Розробка розділу 2: РОЗРОБКА ІМІТАЦІЙНОЇ МОДЕЛІ ПАЛИВНОЇ СИСТЕМИ СУДНА	21.10.2020р – 31.10.2020р.	
6.	Розробка розділу 3: РОЗРОБКА НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ПРОГНОЗУВАННЯ РИЗИКІВ ВИХОДУ З ЛАДУ КОМПОНЕНТІВ СУДНОВИХ СИСТЕМ	1.11.2020р – 14.11.2020р.	
7.	Висновки та оформлення пояснювальної записки дипломного проекту.	15.11.2020р – 30.11. 2020р.	
8.	Підписання необхідних документів у встановленому порядку.	1.12.2020р – 9.12.2020р.	
9.	Підготовка до захисту та попередній захист дипломного проекту на випусковій кафедрі дипломного проекту	10.12.2020р – 22.12. 2020р.	

7. Дата видачі завдання: 05.10.2020 р.

Керівник дипломного проекту _____

(підпис керівника)

професор Воронін А. М.

(П.І.Б.)

Завдання прийняв до виконання _____

(підпис випускника)

Паренюк Л.В.

(П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Алгоритми дослідження нейронних мереж» складається із вступу, трьох розділів, висновків, списку використаних джерел та додатку. Робота містить 41 рисунок та 2 таблиці. Загальний обсяг роботи становить 70 сторінок.

Актуальність магістерської роботи полягає у використанні нейронних мереж для вирішення недосконало формалізованих завдань в інтелектуальному аналізі даних. У цій роботі розгляно технологію навчання нейронної мережі для прогнозування виходу з ладу компонентів системи, що є актуальним для безпеки екіпажу судна та запобіганню можливих екномічних витрат.

Штучні нейронні мережі дають багатообіцяючі перспективи в розвитку, а програмне забезпечення має величезну перевагу від їх використання.

Об'єктом дослідження є існуюче програмне забезпечення нейронних мереж, а також сучасні системи навчання їх та аналізу.

Предметом дослідження є проблема прогнозування ризиків виходу з ладу компонентів системи.

Мета дипломної роботи: розробка нейронної мережі для прогнозування ризиків виходу з ладу компонентів системи на базі паливної системи судна.

Для досягнення мети дипломної роботи поставлено такі завдання:

1. Проведення аналізу літературних джерел з тематики роботи.
2. Аналіз методів та моделей прогнозування ризиків.
3. Дослідження специфіки застосування штучних нейронних мереж.
4. Обґрунтування використаних засобів розробки.
5. Проектування моделі досліджуваної технічної системи судна.

6. Розробка діаграми зв'язків та алгоритму роботи нейронної мережі.
7. Розробка інтерфейсу головної форми системи створення штучної нейронної мережі.
8. Моделювання процесу тренування розробленої моделі штучної нейронної мережі.

Ключові слова : НЕЙРОНА МЕРЕЖА, ІМІТАЦІЙНА МОДЕЛЬ ДВИГУНА . НАВЧАННЯ НЕЙРОНОЇ МЕРЕЖІ, MATLAB.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Історія нейронної мережі.....	9
1.2 Поняття штучна нейрона мережа.....	11
1.3 Класифікації штучних інтелектуальних систем	16
1.4 Система Matlab.....	22
РОЗДІЛ 2 РОЗРОБКА ІМІТАЦІЙНОЇ МОДЕЛІ ПАЛИВНОЇ СИСТЕМИ СУДНА.....	25
2.1 Проектування паливної системи судна	25
2.2 Розробка графової моделі паливної системи судна.....	28
2.3 Побудова та дослідження моделі нейромережі.....	30
РОЗДІЛ 3 РОЗРОБКА НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ПРОГНОЗУВАННЯ РИЗИКІВ ВИХОДУ З ЛАДУ КОМПОНЕНТІВ СУДНОВИХ СИСТЕМ.....	40
3.1 Розробка алгоритму роботи нейронної мережі та діаграми зв'язків, ...	30
3.2 Розробка інтерфейсу головної форми та імпорту даних до системи	43
3.3 Моделювання процесу тренування створеної нейронної мережі за допомоги Neural Network Toolbox.....	48
3.4 Модифікація розробленої нейромережі за допомоги Neural Network Fitting Tool	61
ВИСНОВКИ.....	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	69

ВСТУП

У сучасному світі штучні інтелектуальні системи отримали велику популярність, завдяки багатим можливостям і ефективності використання.

Існує безліч завдань, де необхідно практична роздільна здатність з використанням інтелектуальних систем вже зараз: економіка і бізнес, робототехніка, геологорозвідка, математика, біофізика, авіоніка, безпека та охоронні системи, медицина та багато інші. Така велика кількість областей застосування, говорить про те, що інтелектуальні системи - це унікальний набір для вирішення питань аналізу та обробки великого обсягу даних, вирішення завдань різног рівня складності.

В даний час в системах штучного інтелекту активно використовуються штучні нейронні мережі.

Нейронна мережа – це обчислювальна структура, яка складається з безлічі елементів одного типу. Ці елементи виконують прості функції, а всі процеси, що відбуваються в штучній нейронній мережі можна асоціювати з процесами, що відбуваються в нервовій системі живих організмів.

Нейронні мережі нелінійні за своєю природою, в них відсутня явна залежність, що дозволяє відразу використовувати розроблену технологію (інформаційну модель нейромережі). Протягом багатьох років лінійне моделювання було основним методом моделювання, оскільки для нього добре розроблені процедури оптимізації.

Актуальність магістерської роботи полягає у використанні нейронних мереж для вирішення недосконало формалізованих завдань в інтелектуальному аналізі даних. У цій роботі розгляно технологію навчання нейронної мережі для прогнозування виходу з ладу компонентів системи, що є актуальним для безпеки екіпажу та запобіганню можливих економічних витрат.

Штучні нейронні мережі дають багатообіцяючі перспективи в розвитку, а програмне забезпечення має величезну перевагу від їх використання.

Об'єктом дослідження є існуюче програмне забезпечення нейронних мереж, а також сучасні системи навчання їх та аналізу.

Предметом дослідження є проблема прогнозування ризиків виходу з ладу компонентів системи.

Мета дипломної роботи: розробка нейронної мережі для прогнозування ризиків виходу з ладу компонентів системи на базі паливної системи судна.

Для досягнення мети дипломної роботи поставлено такі завдання:

1. Проведення аналізу літературних джерел з тематики роботи.
2. Аналіз методів та моделей прогнозування ризиків.
3. Дослідження специфіки застосування штучних нейронних мереж.
4. Обґрунтування використаних засобів розробки.
5. Проектування моделі досліджуваної технічної системи судна.
6. Розробка діаграми зв'язків та алгоритму роботи нейронної мережі.
7. Розробка інтерфейсу головної форми системи створення штучної нейронної мережі.
8. Моделювання процесу тренування розробленої моделі штучної нейронної мережі.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Історія розвитку нейронної мережі

Вперше про штучні нейронні мережі заговорили в 1940-х роках. Саме як науковий напрямок теорія нейронних мереж була відображена в роботі Мак Каллока і Пітса 1943 року. У цій роботі стверджувалося, що майже будь-яку логічну або арифметичну функцію можна реалізувати за допомогою найпростішої нейронної мережі.

Однією з основоположних робіт слід відзначити модель Д. Хебба, який у 1949 році сформулював закон, що став відправною точкою для навчання нейронних мереж. Хебб першим припустив, що навчання, перш за все, полягає в зміні сили синаптичних зв'язків. Теорія Хебба – типовий випадок самонавчання, де випробовувана система навчається виконувати необхідне завдання без втручання експериментатора. Свій внесок зробив і М. Мінський, завдяки дослідженню ряду завдань, в тому числі відомого завдання "Виключаючого" АБО "[2].

Ф. Розенблатт 1958 року запропонував нейронну мережу, названу перцептроном, який був призначений для класифікації об'єктів. При навчанні перцептрон отримував повідомлення від "вчителя". Завдяки повідомлення, можна визначити, до якого класу належить цей об'єкт. Крім того, навчений перцептрон був здатний самостійно класифікувати об'єкти, які не використовувалися раніше мале число помилок.

Кафедра КІТ (47)				НАУ 20 16 80 000 ПЗ			
<i>Виконав</i>	<i>Паренюк Л.В.</i>			АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	<i>Літера</i>	<i>аркуш</i>	<i>аркушів</i>
<i>Керівник</i>	<i>Воронін А.М.</i>					9	16
<i>Консульт.</i>					УС-211М 122		
<i>Н. контроль</i>	<i>Райчев І.Е.</i>						

Період затишшя в розвитку нейронних мереж припав на 1968-1985 роки. З появою високопродуктивних персональних комп'ютерів стало доступним моделювати нейронні мережі.

Провідною книгою для фахівців, які цікавляться теорією нейронних мереж, стала робота Ф. Уоссермена "Нейрокомп'ютерна техніка".

Після появи роботи Д. Хопфілда (1982 рік) інтерес до нейронних мереж різко зріс. Хопфілд, ґрунтуючись на правилах навчання Хебба, показав, що завдання з нейронами можуть бути зведені до узагальнення ряду моделей, розроблених до того моменту у фізиці неупорядкованих систем.

Потім у 1986 році з'явилася робота Вільямса, Румельхарта і Хінтона, яка відповідала на важливе питання щодо поведінки навчання верствоїстих нейронних мереж. Після цього, запропонований Хінтоном алгоритм зазнав безліч змін.[3]

У 80-ті роки поступово сформувався міцний теоретичний фундамент, на основі якого сьогодні створюється більшість мереж. Розроблена теорія широко стала застосовуватися в останні два десятиліття для вирішення прикладних завдань. Стали з'являтися фірми, розробники програмного забезпечення для конструювання штучних нейронних мереж. У 90-ті роки нейронні мережі стали використовуватися в бізнесі, де вони показали колосальну ефективність при вирішенні багатьох завдань - від передбачення попиту на продукцію до аналізу платоспроможності клієнтів банку.

У 2007 в університеті Торонто Джеффри Хінтон створив алгоритми глибокого вивчення нейронних мереж. При навчанні нижніх шарів мережі, Хінтон використовував обмежену машину Больцмана, яка представляє стохастичну рекуррентну нейронну мережу. Після навчання мережі отриманий додаток могло швидко вирішувати поставлену завдання (наприклад, пошук осіб на фотографії). Ця функція на сьогоднішній день

вбудована в усі цифрові фотоапарати. Подібна технологія використовується інтернет – пошуковиками під час класифікації картинок.[4]

За оцінками фахівців, в області проектування нейронних мереж і нейрокомп'ютерів очікується технологічне зростання. Чимало нових можливостей було відкрито за останні роки, а роботи в даній області стають важливим внеском у науку, технології та економіку.

Незважаючи на те, що вивчення нейронного моделювання ведеться вже більше шістдесяти років, немає жодної області мозку, де процес обробки інформації був би ясний до кінця. Також немає жодного нейрона, для якого можна було б визначити код передачі інформації у вигляді послідовності імпульсів.

Зараз існує велика кількість конфігурацій нейронних мереж, які відрізняються за принципами функціонування, і, отже, спрямовані на різні завдання.

Майбутнє нейрокомп'ютерних технологій буде пов'язано з новими відкриттями в області нейронного моделювання - як тільки вдасться розгадати таємницю функціонування хоча б однієї області мозку, відразу стане ясно багато про інші його області.

1.2 Поняття "штучна нейронна мережа"

Під штучною нейронною мережею розуміється математична модель, а також її програмна та апаратна реалізація, побудована за принципу біологічних нейронних мереж - нервових клітин живого організму. Це поняття виникло при спробі змоделювати процеси, люди, що протікають у мозку.[4]

Штучна нейронна мережа являє собою систему простих процесорів (штучних нейронів), з'єднаних і взаємодіючих між собою. Кожен з процесорів мережі має справу з сигналами, які періодично надходять або

передаються іншим процесорам. Велика мережа здатна вирішувати найскладніші завдання в найкоротші терміни.

З математичної точки зору нейронні мережі являють собою спосіб вирішення нелінійних завдань оптимізації. Кібернетика використовує теорію нейронних мереж у вирішенні завдань адаптивного управління, побудові алгоритмів для робототехніки.[2]

У програмуванні нейронна мережа один із способів вирішення проблеми ефективного паралелізму.

Програмування нейронних мереж передбачає саме навчання мережі, а не написання програмного коду. Саме завдяки навчанню мереж здатна виявляти залежність між даними (вхідними та вихідними) узагальнювати, спрощувати результати, використовувати знання для розбиття складних завдань на більш прості.

1.2.1 Біологічний нейрон і штучний нейрон

Мозок людини і її нервова система складаються з нейронів, з'єднаних нервовими волокнами. Між нейронами передаються електричні імпульси за допомогою нервових волокон. Всі дії, які відбуваються з живим організмом, всі подразнення шкіри, очей, біль, процеси мислення – є взаємодія між нейронами. Будова біологічного нейрона показано на рис 1.1.

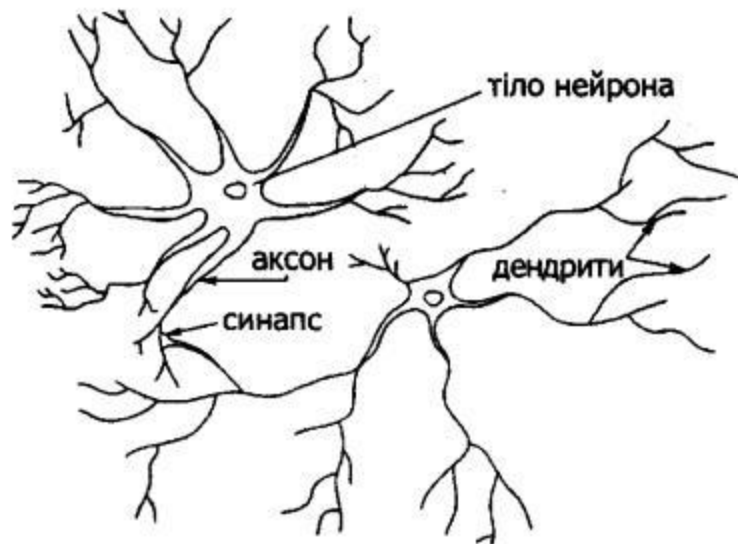


Рис.1.1. Біологічний нейрон

Дендрити – приймає імпульси нейрона;

Аксон – передає імпульс нейрона;

Синапси – утворення, що впливають на силу імпульсу, для контакту аксона і дендрита.

При проходженні синапсу сила імпульсу змінюється на певне число разів (вага синапсу). Коли до нейрона по декількох дендритах надходять імпульси, то вони підсумовуються. Якщо сумарний імпульс перевищено поріг, то нейрон переходить у стан збудження, формує власний імпульс і посилає його далі по аксону. Поведінка відповідного нейрона може змінюватися, оскільки ваги синапсів мають властивість змінюватися з часом. Математична модель описаного процесу представлено таким чином (рис. 1.2):

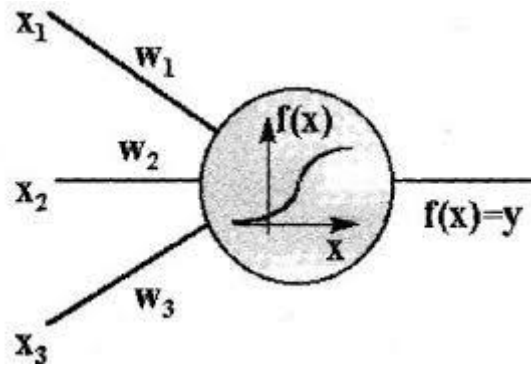


Рис.1.2. Математична модель нейрона

Ця модель описує нейрон з трьома входами (дендритами), де синапси мають ваги w_1 , w_2 , w_3 , до яких надходять сили x_1 , x_2 , x_3 відповідно. До нейрона надходять імпульси x_1w_1 , x_2w_2 , x_3w_3 після проходження синапсів і дендритів.

Отриманий сумарний імпульс $x = x_1w_1 + x_2w_2 + x_3w_3$ нейрон перетворює відповідно до передавальної функції $f(x)$.

$y = f(x) = f(x_1w_1 + x_2w_2 + x_3w_3)$ - сила вихідного імпульсу. Підводячи підсумок, отримуємо набір чисел x_k (вектор) у вигляді входів. Далі нейрон видає деяке число на виході .[5]

Штучний нейрон виглядає наступним чином: на його вхід надходить безліч сигналів, кожен з яких одночасно є виходом іншого нейрона. Такий вхід множиться на відповідну вагу, потім твори підсумовуються, визначаючи рівень активації нейрона.

Модель, що реалізує цю ідею, представлена на рис 1.3:

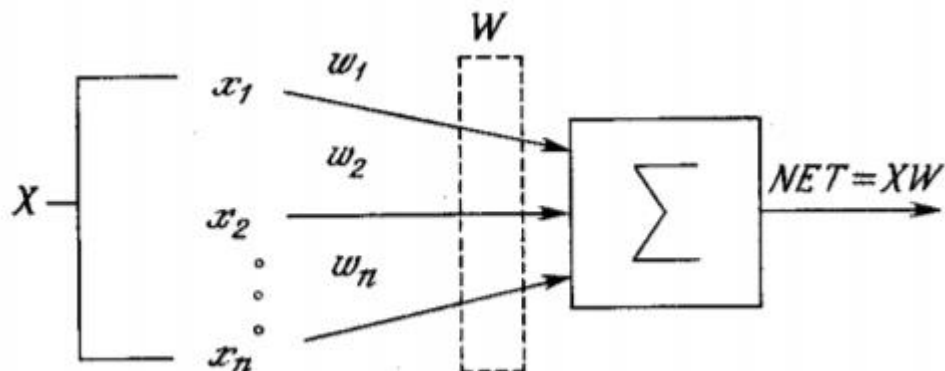


Рис.1.3. Модель активації нейрона

Тут безліч сигналів x_1, x_2, \dots, x_n , надходить на вхід штучного нейрона, у сукупності вони позначаються вектором X . Ці сигнали, аналогічні тим, що отримує на вході біологічний нейрон. Далі сигнал також множиться на відповідну вагу w_1, w_2, \dots, w_n , потім підсумовується в підсумовуючому блоці Σ . Кожна вага дорівнює силі одного синаптичного зв'язку в біологічному нейроні. Вихід, який зазвичай називається NET, створюється з підсумовуючого блоку, де зважені елементи складаються алгебраїчно.

Сигнал NET перетворюється звичайною лінійною функцією, званою активаційною. Вона позначається F і дає вихідний сигнал OUT.

$$OUT = K(NET),$$

де K - постійна, порогової функції

$$OUT = 1, \text{ якщо } NET > T$$

$$OUT = 0 \text{ в інших випадках,}$$

T – постійна порогова величина, яка більш точно моделює нейронну мережу.

Штучний нейрон з активаційною функцією представлений на рис. 1.4:

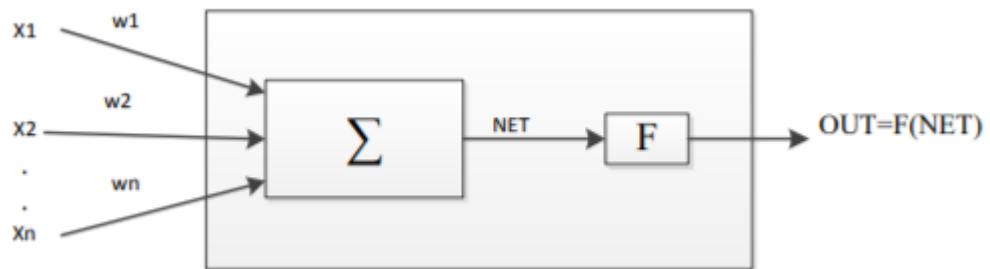


Рис.1.4. Активаторська функція нейрону

Блок, позначений F, приймає сигнал NET і видає сигнал OUT.

F— називається стискаючою функцією, якщо при будь-яких значеннях NET значення OUT належать певному кінцевому інтервалу.

Модель штучного нейрона ігнорує більшість властивостей біологічного нейрона. Наприклад, затримки в часі, які впливають на динаміку системи.

Вхідні сигнали відразу ж породжують вихідний сигнал. Крім того, штучний нейрон не враховує вплив синхронізуючої функції біологічного нейрона.

Однак слід відзначити виняткову схожість живого нейрона і штучного.

Для визначення місця нейронних мереж в області інформаційних технологій необхідно звернутися до класифікації штучних інтелектуальних систем.

1.3 Класифікації штучних інтелектуальних систем

Штучні інтелектуальні системи мають характерні особливості:

По – перше, розвинені комунікативні здібності, які характеризують спосіб взаємодії між комп'ютером користувача з системою. Не виключена можливість звернення до системи з довільним запитом у діалоговому вікні інтелектуальної системи. При цьому мова інтелектуальній системі повинен бути максимально наближений до природній мові.

По – друге, вирішення поганих завдань, тобто таких завдань, які не мають конкретного рішення, а потребують нестандартного підходу, залежно від ситуації, існуючих даних і кінцевого результату.

Поганоформульовані завдання ефективно вирішуються за допомогою штучних нейронних мереж.

По – третє, здатність до самонавчання - тобто можливості вилучення знань інтелектуальною системою з накопиченого досвіду конкретних ситуацій. Для попереднього навчання системи необхідні оброблені початкові дані.

Відповідно до представлених ознак, інтелектуальні системи можна розділити на наступні (таблиця 1.1):

Таблиця 1.1 Види штучних інтелектуальних систем

Вид штучної інтелектуальної системи	Тип штучної інтелектуальної системи
Системи з комутативними здібностями	-інтелектуальні бази даних; -природно-мовні інтерфейси; - гіпертекстові системи; - контекстні довідкові системи; -когнітивна графіка.
Експертні системи	- класифікуючі системи; - до визначальні системи; - трансформуючі системи;

	- багатоантенні системи
Самонавчальні системи	- індуктивні системи; - нейронні мережі; - системи на прецедентах; - інформаційні сховища
Адаптивні системи	- CASE - технології; - компонентна технологія.

Інтелектуальні бази даних відрізняються від звичайних баз даних можливістю вибірки за запитом необхідної інформації, яка може не зберігатися, а виводитися з наявної в базі даних.

Природно, мовний інтерфейс перетворює конструкції природної мови у внутрішньомашинний рівень представлення знань. Використовується для доступу до інтелектуальних баз даних, контекстного пошуку документальної текстової інформації, команд голосового вводу в системах управління, машинного перекладу з іноземних мов.

Гіпертекстові системи застосовуються в базах текстової інформації, де необхідний пошук за ключовими словами, мають більше складну семантичну організацію ключових слів.

У контекстних довідкових системах користувач описує проблему (ситуацію), а система за допомогою додаткового діалогу її конкретизує і виконує пошук рекомендацій, що підходять для даної ситуації. Такі системи створюються як програма до систем документації і належать до класу систем поширення знань.

Системи когнітивної графіки використовуються в моніторингу і управління оперативними процесами. Графічні образи в наглядному і інтегрованому вигляді описують безліч параметрів вивчення ситуації.

Експертні системи призначені для вирішення завдань на основі накопичуваної бази знань, що відображає досвід роботи експертів у розглянутої проблемної області.

Багатоагентні системи - це динамічні системи, для яких характерна інтеграція в базі знань кількох різнорідних джерел знань, що обмінюються між собою отримуваними результатами на динамічній основі.

Самонавчальні системи засновані на методах автоматичної класифікації прикладів ситуацій реальної практики.

Характерними ознаками самонавчальних систем є:

- самонавчальні системи "з учителем", коли для кожного прикладу задається в явному вигляді значення ознаки його належності деяком класу ситуацій (класоутворюючої ознаки);

- самонавчальні системи "без вчителя", коли за ступенем близькості значень ознак класифікації система сама виділяє класи ситуацій.

Індуктивні системи за принципом від приватного до загального узагальнюють приклади, а процес узагальнення здійснюється таким чином:

1) Вибирається ознака класифікації з багатьох вказаних (послідовно або за правилом).

2) За значенням вибраної ознаки багато прикладів розбивається на підмножини.

3) Виконується перевірка про належність прикладу до одного класу.

4) Якщо якась підмножина прикладів належить одному підкласу, тобто у всіх прикладів підмножини збігається значення класоутворюючої ознаки, процес класифікації закінчується (інші ознаки класифікації не розглядаються).

5) Для підмножин прикладів з неспівпадаючим значенням класоутворюючої ознаки процес класифікації триває, починаючи з пункту 1.

Нейронні мережі - це такі інструменти для паралельних обчислень, які складаються з безлічі простих процесорів, які періодично отримують і відправляють сигнали іншим процесорам.

Далі слід звернутися до інтелектуальних інформаційних системам з точки зору вирішуваних завдань. Тут ви можете вибрати наступні системи:

- системи управління та довідкові системи;
- системи комп'ютерної лінгвістики;
- системи розпізнавання;
- ігрові системи;
- системи створення інтелектуальних інформаційних систем.

На рис. 1.5 представлена класифікація інтелектуальних інформаційних систем за вирішуваними завданнями:

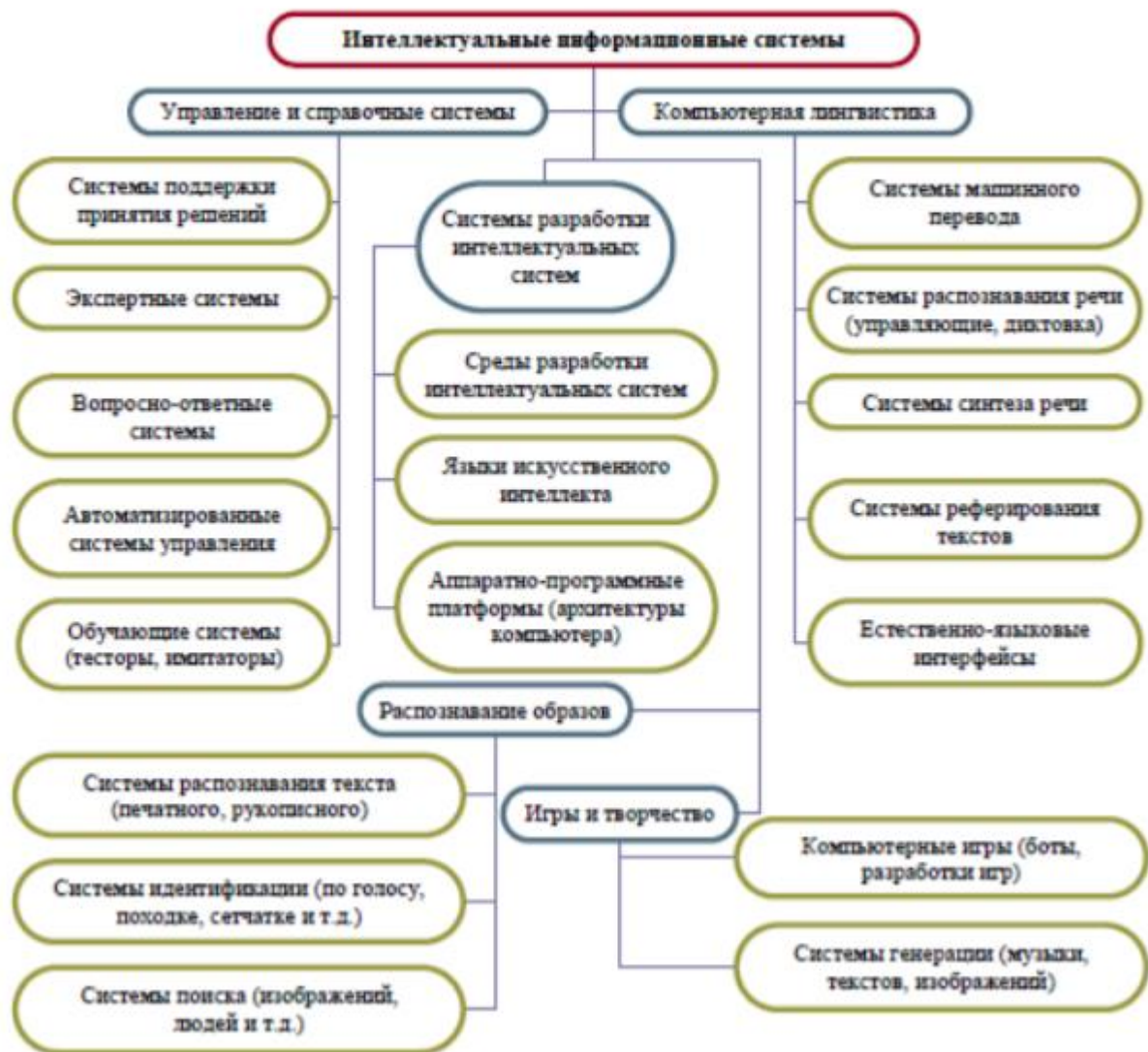


Рис.1.5 Класифікація ІС за вирішеними завданнями

Системи можуть вирішувати не одне, а кілька завдань або в процесі вирішення одного завдання вирішувати і ряд інших завдань.

Інтелектуальні системи також класифікуються за таким критерієм як "використовувані методи". Тут можна виділити м'які, жорсткі та гібридні методи (рис. 1.6):

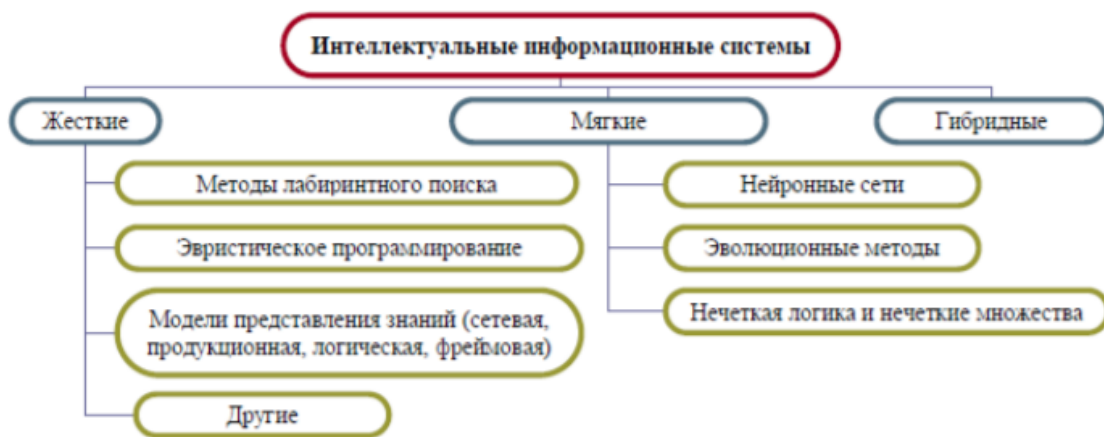


Рис.1.6 - Класифікація ІС за методами

М'які обчислення (Soft Computing) - це складна комп'ютерна методологія, заснована на нечіткій логіці, генетичних обчисленнях, нейрокомп'ютингу і ймовірнісних обчисленнях.

Жорсткі обчислення - стандартні комп'ютерні обчислення (деякі належать до м'яких).

Гібридні системи - системи, що використовують більш ніж одну комп'ютерну технологію (у разі інтелектуальних систем – технології штучного інтелекту).

Класифікація штучних інтелектуальних систем охоплює величезну теоретичну базу знань у різних галузях науки. Створення і реалізація ІС - складний процес від початкового етапу до кінцевого.

1.4 Система MATLAB

MATLAB - матрична лабораторія - найбільш розвинена система програмування для науково-технічних розрахунків, доповнена до теперішнього часу кількома десятками більш приватних застосувань, що відносяться до обчислювальної математики, обробці інформації, конструюванню електронних приладів, економіці та ряду інших розділів прикладної науки. Вивчення MATLAB'a по фірмовій документації, яка тепер

додається на інсталяційному компакт-диску, займає у початківців користувачів занадто багато часу не тільки через необхідність читати її англійською мовою зі специфічним сленгом, але, головним чином, через неминуче для таких керівництв послідовного і досить формального викладу великого обсягу інформації, а наявні російською мовою посібники в основному слідують цьому стереотипу. Навіть для досвідченого фахівця з розрахунками на комп'ютері таке вивчення пов'язане з невиправдано великими витратами праці.

MATLAB призначений насамперед для програмування чисельних алгоритмів. Він розробляється вже більше 15 років і виник на основі більш ранніх прикладних пакетів LINPACK і EIGPACK, створених в 1970-і рр. в США, і в свою чергу вплинув на появу таких систем, як MathCad, MAPLE і Mathematica. Удосконалення системи MATLAB відбувалося як у зв'язку з досягненнями в обчислювальній математиці, так і у зв'язку зі змінами в архітектурі персональних комп'ютерів і розвитком загальносистемних засобів. З часом MATLAB був доповнений цілим рядом вже згадуваних додатків (toolboxes), далеко розвинувши межі його застосовності. Далі мова піде лише про ядрі MATLAB'a, яке ми будемо називати системою, і конкретно про її версії 5.2, випущеної фірмою MathWorks в січні 1998 р.

MATLAB - система програмування високого рівня, що працює як інтерпретатор і включає великий набір інструкцій (команд) для виконання найрізноманітніших обчислень, завдання структур даних і графічного представлення інформації. Команди ці розбиті на тематичні групи, розташовані в різних директоріях системи. Тепер в системі близько 800 команд, і приблизно половина з них цілком доступна починаючому користувачеві. Команди з великим можливим обсягом обчислень написані на C, але багато і таких команд, які представлені в термінах цих перших. Тому система виявляється майже відкритою для користувача. Є великі можливості для виведення двовимірної і тривимірної графіки і засоби управління нею.

Користувач може без особливих труднощів додавати свої команди і писати програми в термінах вже існуючих команд; дещо складніше робити це в рамках Фортрана і С. Можна обмінюватися даними з програмами на цих мовах, а з них звертатися до системи. Стислість і наочність програмування і виняткові можливості візуалізації результатів роблять систему дуже ефективною при пошуках і апробації нових алгоритмів, при проведенні разових розрахунків і в навчальному процесі, оскільки її можна освоювати без попереднього знайомства з основами програмування і виконувати такі складні приклади, які неможливо робити з використанням інших систем.

У системи MATLAB є схожі риси з програмами MathCad і Electronics Workbench.

Так з MathCad її ріднить те, що в шпалери системах є широкі можливості по виконанню обчислень, вироблених з матрицями, векторами і комплексними числами, а також графічне представлення отриманих результатів. Відмінною рисою є вхідний мову, максимально наближений до звичайного математичному мови.

А з Electronics Workbench загальним є можливість створення моделей як окремих об'єктів так і систем, шляхом поблочного моделювання та за допомоги спеціальних блоків спостерігати протікають процеси в моделі.

Основний об'єкт системи MATLAB - прямокутний числовий масив (матриця), в якому допускається застосування комплексних елементів. Використання матриць не вимагає явної вказівки їх розмірів.

Система MATLAB забезпечує виконання операцій з векторами і матрицями навіть в режимі безпосередніх обчислень. Нею можна користуватися як найпотужнішим калькулятором, в якому поряд зі звичайними арифметичними і алгебраїчними діями можуть використовуватися такі складні операції, як звернення матриці, обчислення її власних значень і векторів, рішення систем лінійних алгебраїчних рівнянь і

багато інших. Характерною особливістю системи є її відкритість, тобто можливість її модифікації та адаптації до конкретних завдань користувача.

РОЗДІЛ 2

РОЗРОБКА ІМІТАЦІЙНОЇ МОДЕЛІ ПАЛИВНОЇ СИСТЕМИ СУДНА

2.1 Проектування структури паливної системи судна

Зовнішня паливна система (рис. 2.1) призначена для подачі палива з витратних баків до паливних насосів високого тиску дизеля. Встановлений на дизелі паливний насос 7, що приводиться від колінчастого вала, засмоктує паливо з паливного бака 12 (2) через фільтр грубої очистки 17 і зворотний клапан 16 (1) і нагнітає його через фільтр тонкого очищення 6 в трубу, що має відводи до паливних насосів високого тиску 3 кожного циліндра. Підпірний клапан 19 призначений для підтримки певного тиску в цій трубі, необхідного для надійної роботи насосів 3. Невикористаний в паливних насосах паливо (надлишкове), пройшовши клапан 19, надходить в топлівопідігрівач 18, а потім повертається в бак. Вхідні в паливний бак трубки (зливна і огорожі) утворюють топлівозаборну пристрій.

Кафедра КІТ (47)				НАУ 20 16 80 000 ПЗ			
<i>Виконав</i>	<i>Паренюк Л.В.</i>			РОЗРОБКА ІМІТАЦІЙНОЇ МОДЕЛІ ПАЛИВНОЇ СИСТЕМИ СУДНА	<i>Літера</i>	<i>аркуш</i>	<i>аркушів</i>
<i>Керівник</i>	<i>Воронін Л.В.</i>					25	15
<i>Консульт.</i>					УС-211М 122		
<i>Н. контроль</i>	<i>Райчев І.Е.</i>						

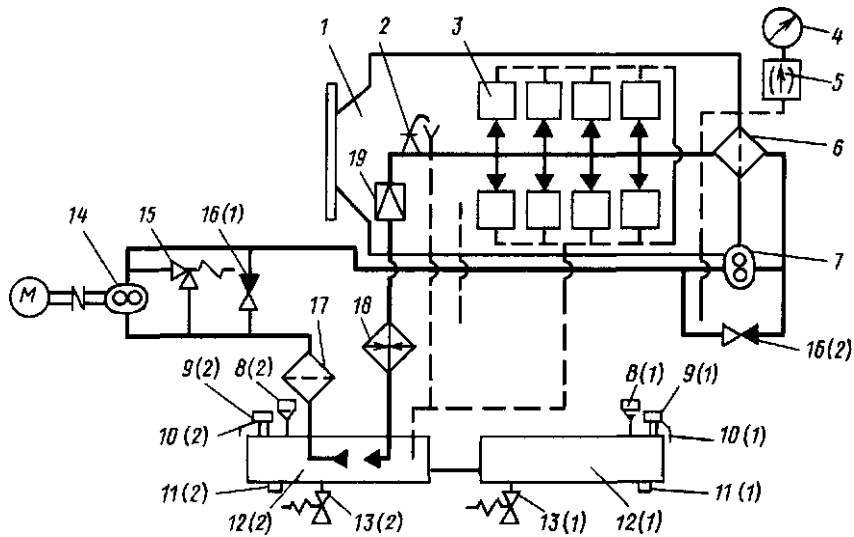


Рис. 2.1 Схема суднової паливної системи [1]

Структура топливної системи містить: 1-дизель, 2-край для випуску повітря, 3-паливний насос високого тиску; 4-дистанційний електроманометр; 5-демпфер; 6-фільтр тонкого очищення палива; 7-паливний насос, 8 (1), 8 (2) -залівочні горловини; 9 (1), 9 (2) - труби паливоміри; 10 (1), 10 (2) - вентиляційні труби, 11 (1), 11 (2) - сливи пробки; 12 (1), 12 (2) -паливний баки, 13 (1), 13 (2) -клапани зливу палива; 14-топливоподкачивающий насос; 15-запобіжний клапан; 16 (1), 16 (2) -зворотний клапани; 17-фільтр глибокой очищення палива, 18-паливо-підігрівач, 19-підпірних клапан.

Перед пуском дизеля паливо в паливній системі прокачують за допомогою топливоподкачивающего насоса 14, що має привід від електродвигуна. Це необхідно для заповнення всіх трубопроводів і порожнин системи паливом і видалення з неї повітря, який виходить через кран 2 і краники на фільтрі тонкого очищення 6. Клапан 15 оберігає т топливоподкачивающий агрегат від перевантаження, перепускаючи частина палива з нагнітальної у всмоктувальну магістраль. Зворотний клапан 16 (1) при роботі топливоподкачивающего агрегату закритий під тиском палива і розділяє всмоктувальну і нагнітаючу магістралі, а зворотний клапан 16 (2) під дією цього ж тиску відкритий і пропускає паливо до фільтру 6 в обхід насоса 7.

Після початку роботи дизеля топливопідкачуючий агрегат відключають, так як створюваний ним підпір палива у всмоктувальній магістралі паливного насоса 7 знижує довговічність ущільнювач-них манжет цього насоса. При роботі тільки одного насоса 7 зворотний клапан 16 (1) відкривається під дією розрідження у всмоктувальній магістралі, а клапан 16 (2) закривається під тиском палива, розділяючи нагнітаючу і всмоктувальну магістралі.

Тиск палива, що підводиться до насосів 3, вимірюється дистанційним електроманометром 4. Демпфер 5, що складається з пластин з отворами малого діаметру, призначений для захисту чутливого датчика манометра від руйнування при пульсаціях тиску, що виникають в системі.

Паливо на тепловозі розміщено в баках 12 (1) і 12 (2) загальною місткістю 5400 л. Баки підвішені під рамою тепловоза на кронштейнах і закріплені болтами. Вони з'єднані між собою трубою і є сполученими посудинами. Наявність двох баків викликано конструктивною особливістю тепловоза з гідропередачею, яка разом з карданним приводом осьових редукторів візків займає середню частину тепловоза. Паливні баки зварюють із сталевих листів товщиною 4 мм. Для захисту від корозії внутрішні і зовнішні поверхні бака фарбують. Для забезпечення роботи тепловоза в умовах гарячих цехів металургійних підприємств на вимогу замовника баки обладнають екранами теплового захисту, виготовляються зі сталевих або алюмінієвих листів і розташовані на відстані 40-50 мм від поверхні бака по всьому його периметру. Баки заповнюють паливом через заливні горловини 8 (1) і 8 (2), розташовані по обидва боки тепловоза. У горловини вставлені сітчасті фільтри, аналогічні вживаним на тепловозах ТГМЗ. У днище кожного бака є відстійник із зливною пробкою 11, яка відкривається при промиванні бака, і клапаном зливу палива 13, призначеним для зливу відстою через спеціальний шланг. На бічній поверхні кожного бака є чотири лочка (на конічній різьби) для огляду і промивання внутрішньої порожнини. [1]

Обсяг палива в баках заміряють топлівомірними рейками, що встановлюються в трубах 9 (1) до 9 (2). Через ці труби і труби 10 (1) і 10 (2) баки повідомляються з атмосферою, що забезпечує вихід повітря з них при заповненні і виключає появу розрідження в баках при витраті палива дизелем.

2.2 Розробка графової моделі паливної системи судна

Для дослідження специфіки функціонування паливної системи судна слід побудувати модель цієї системи у вигляді орієнтованого графу, у якому вузли – це назви компонентів системи, а ребра графа – зв'язки між її компонентами.

Результат побудови графової моделі паливної системи судна наведено на рис.2.2.

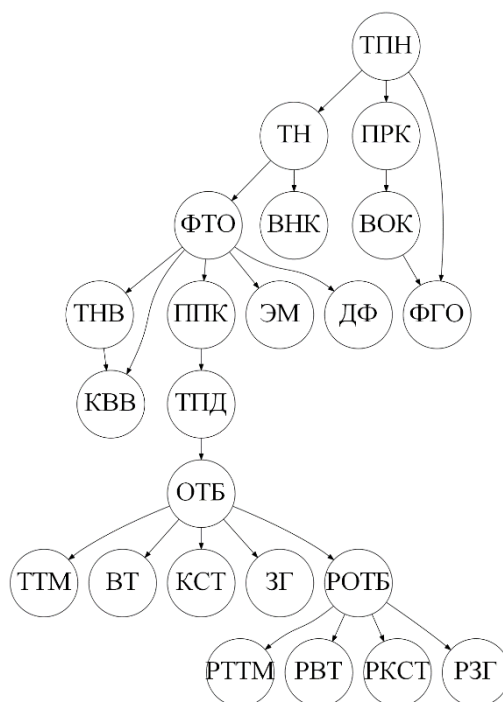


Рис. 2.2. Графова модель топливної системи судна

Перелік назв та їх скорочень для компонентів паливної системи судна наведено у таблиці 2.1.

Таблиця 2.1. Перелік назв та скорочень компонентів паливної системи судна

Назва	Скорочення
Топливоподкачивающий Насос	ТПН
Запобіжні клапани	ПРК
Паливний насос	ТН
Фільтр глибокого очищення палива	ФГО
Зовнішній зворотний клапан	ВОК
Внутрішній зворотний клапан	ВНК
Фільтр тонкою очищення палива	ФТО
Паливний насос високого тиску	ТНВ
Підпірний клапан	ППК
Край випуску повітря	КВВ
Паливо-підігрівач	ТПД
Дистанційний електромаіометр	ЭМ
Демпфер	ДФ
Основний паливний бак	ОТБ
Основна труба паливоміри	ТТМ
Основна вентиляційна труба	ВТ
Основний клапан слива палива	КСТ
Основна заливальне горловина	ЗГ

Резервний паливний бак	РОТБ
Резервна труба паливоміри	РТТМ
Резервна вентиляційна труба	РВТ
Додатковий клапан слива палива	РКСТ
Додаткова заливальне горловина	РЗГ

2.3 Побудова та дослідження моделі нейромережі

У режимі конфігурації нейромережі майстер обробки дозволяє сконструювати нейронну мережу із заданою структурою, визначити її параметри і навчити за допомогою одного з доступних в системі алгоритмів навчання. В результаті буде отримано емулятор нейронної мережі, який може бути використаний для вирішення завдань прогнозування, класифікації, пошуку прихованих закономірностей, стиснення даних і багатьох інших додатків.

Налагодження та навчання нейронної мережі складається з наступних кроків:

1. Налаштування призначень полів.
2. Налаштування нормалізації полів.
3. Налаштування навчальної вибірки.
4. Налаштування структури нейронної мережі.
5. Вибір алгоритму і настройка параметрів навчання.
6. Встановлення умов зупинки навчання.
7. Запуск процесу навчання.

8. Вибір способу відображення даних.

9. Відомості про вузол.

На першому етапі здійснюється формування списку всіх полів вихідного набору даних. Для настройки поля слід виділити його в списку, при цьому в правій частині вікна будуть відображені його параметри:

– Ім'я поля - ідентифікатор поля, визначений для нього в джерелі даних. Змінити його тут не можна.

– Тип даних - тип даних, що містяться в поле. тут його змінити не можна.

○ Логічний - дані в поле можуть приймати тільки два значення: Істина або Брехня.

○ Дата / час - поле містить дані типу дата / час.

○ Речовий - дані в поле є числа з плаваючою крапкою.

○ Цілий - дані в поле є цілі числа.

○ Строковий - дані в стовпці представляють собою рядки символів.

– Призначення - тут необхідно вибрати порядок використання даного поля при навчанні і роботі нейронної мережі. Вибір проводиться за допомогою списку, що відкривається кнопкою і містить наступні варіанти:

○ Вхідний - значення поля будуть подаватися на відповідні входи нейронної мережі, тобто бути вихідними даними для навчання та подальшої практичної роботи навченої нейронної мережі.

○ Вихідна - значення даного поля будуть еталонними при навчанні нейронної мережі, а при практичній роботі з нею будуть містити результати обробки вхідних полів.

○ Інформаційне - поле не використовуватиметься при навчанні мережі, але буде поміщено в результуючий набір в початковому стані.

○ невикористовуваних - поле не використовуватиметься при навчанні і роботі нейронної мережі і буде виключено з результуючої вибірки. На

відміну від непридатного таке поле може бути використано, якщо в цьому виникне необхідність.

- Непридатне - поле не може бути використано при побудові і роботі алгоритму, але буде поміщено в результуючий набір в початковому стані.

- Вид даних - вказує на характер даних, що містяться в поле (безперервний або дискретний). Змінити цю властивість тут не можна.

Статус непридатного поля встановлюється тільки автоматично і в подальшому може бути змінений тільки на невикористовуване або інформаційне. Поле буде заборонено до використання, якщо:

- поле є дискретним і містить всього одне унікальне значення;
- безперервне поле з нульовою дисперсією;
- поле містить пропущені значення.

До вибору вхідних і вихідних полів слід ставитися особливо уважно, беручи до уваги наступні міркування:

- Вхідні і вихідні поля визначаються характером розв'язуваної задачі: вхідні поля містять вихідні дані, що підлягають обробці, а Вихідні - результати обробки.

- Використання полів, дані в яких не мають відношення до розв'язуваної задачі, вкрай небажано, оскільки вони тільки захаращують вихідну і результуючу вибірку і уповільнюють роботу мережі.

У разі, якщо поточне поле містить безперервні (числові) дані, в правій нижній частині вікна з'являється секція "Статистика", в якій відображаються максимальне і мінімальне значення поля, його середнє значення та стандартне відхилення. Якщо виділене поле містить дискретні (строкові) дані, то для нього відкривається секція "Унікальні значення", в якій відображається загальна кількість унікальних значень в поле, а також список самих унікальних значень.

На етапі нормалізації здійснюється перетворення даних до виду, найбільш підходящому для обробки засобами пакету Deductor. Для таких обробників, як нейронна мережа, лінійна модель прогнозування, самоорганізуються карти, дані, що надходять на вхід, повинні мати числовий тип (тобто мати безперервний характер), а їх значення повинні бути розподілені в певному діапазоні. В цьому випадку нормалізатор може перетворити дискретні дані до набору унікальних індексів або безперервним значенням.

Вікно налаштування нормалізації полів викликається за допомогою кнопки "Налаштування нормалізації". У вікні зліва наведено повний список вхідних і вихідних полів. При цьому кожне поле позначено значком, що позначає вид нормалізації поля:

- Бітова маска - перетворення дискретних значень в бітову маску.
- Лінійна - лінійна нормалізація вихідних значень.
- Унікальні значення - перетворення унікальних значень в їх індекси.

У правій частині вікна для виділеного поля відображаються параметри нормалізації.

Для числових (безперервних) полів з лінійної нормалізацією на закладці "Параметри нормалізації" доступні додаткові налаштування.

Привести до діапазону - дозволяє розширити діапазон значень, які може приймати виділений вхід або вихід мережі. Це використовується, коли заздалегідь відомо, що навчальна вибірка покриває не весь діапазон реальних значень, які буде приймати це поле при використанні моделі. За замовчуванням значення мінімальної і максимальної меж діапазону обчислюються за навчальною вибіркою. Їх можна змінити, безпосередньо вводячи нові обчислення у відповідних полях.

У секції Розширити діапазон % можна вказати, на скільки відсотків розширити діапазон. При встановленому прапорі "Синхронно" верхня і

нижня межі розширюються на однакову величину, при знятому прапорці можна вказати ці значення індивідуально. У будь-якому випадку поля "Мінімум" і "Максимум" секції "Діапазон" будуть розраховані автоматично.

На наступному етапі здійснюється розбиття вихідного набору даних на підмножини - навчальне і тестове.

1. Навчальна множина - включає записи (приклади), які будуть використовуватися в якості вхідних даних, а також відповідні бажані вихідні значення.

2. Тестове безліч - також включає записи, що містять вхідні і бажані вихідні значення, але використовується не для навчання моделі, а для перевірки його результатів.

Для розбиття вихідної безлічі на навчальну і тестову необхідно налаштувати декілька параметрів:

– Зі списку "Спосіб розділення вихідного безлічі" вибирається порядок відбору записів в усі три безлічі:

○ По порядку - порядок проходження записів при їх поділі не змінюється. Безлічі послідовно формуються відповідно до визначеного для них числом записів.

○ Випадково - відбір записів відбувається випадковим чином.

○ За одну - при виборі даного способу, необхідно вказати стовпець для розділення вихідного безлічі. Можна використовувати лише стовпці логічного типу. Поля зі значенням "істина" потрапляють в тестове безліч, зі значенням "брехня" в навчальне.

– Необхідно вказати, які безлічі будуть використовуватися. Для того щоб безліч було сформовано, потрібно встановити прапорець зліва від його назви. Якщо прапорець скинутий, то безліч використано не буде. Навчальна множина використовується завжди, тому скинути прапорець для нього не можна.

– Для кожного з використовуваних множин необхідно поставити його розмір. Розмір може бути заданий безпосередньо кількістю записів або у відсотках від обсягу вихідної вибірки. Для цього досить двічі клацнути мишею у відповідній клітці і ввести потрібне значення з клавіатури. При цьому розмір, введений в процентах, автоматично перераховується в кількість рядків і навпаки.

В полі "Кількість рядків (всього)" відображається загальна кількість записів у вихідній вибірці даних, яке може бути задіяно для формування множин.

Якщо сумарне число рядків для всіх використовуваних множин менше повного числа рядків вихідної вибірки, то розміри множин можна задавати довільно. Можна, наприклад, використовувати не всі записи, а тільки частина з них. Якщо ж сумарне вказане число рядків перевищує максимальну для даної вихідної вибірки, то автоматично включається баланс множин, тобто при вказівці для одного з множин розміру, в результаті якого буде перевищено максимальну кількість записів у вихідній вибірці, розмір інших множин буде автоматично зменшено таким чином, щоб сумарний розмір множин не перевищував доступного числа записів.

На цьому кроці задаються параметри, що визначають структуру нейронної мережі, - кількість прихованих шарів і нейронів у них, а також активаційна функція нейронів.

У секції "Нейрони в шарах" необхідно вказати кількість прихованих шарів, тобто шарів нейронної мережі, розташованих між вхідним і вихідним шарами. Для цього в поле "прихованих шарів" вказати відповідне значення. Це можна зробити з клавіатури, попередньо викликавши клацанням миші курсор, або за допомогою стрілок в правій частині поля. Число нейронів у вхідному і вихідному шарах автоматично встановлюється відповідно до числа вхідних і вихідних полів навчальної вибірки і тут змінити його не можна.

До вибору кількості прихованих шарів і кількості нейронів для кожного прихованого шару потрібно підходити обережно. Хоча до цих пір не вироблені чіткі критерії вибору, дати деякі загальні рекомендації все ж можливо. Вважається, що завдання будь-якої складності можна вирішити за допомогою двошарової нейромережі, тому конфігурація з кількістю прихованих шарів, що перевищують 2, навряд чи виправдана. Для вирішення багатьох завдань цілком підійде одношарова нейронна мережа. При виборі кількості нейронів слід керуватися таким правилом: "кількість зв'язків між нейронами має бути приблизно на порядок менше кількості прикладів в навчальній множині". Кількість зв'язків розраховується як зв'язок кожного нейрона з усіма нейронами сусідніх шарів, включаючи зв'язку на вхідному і вихідному шарах. Занадто велика кількість нейронів може привести до так званого "перенавчання" мережі, коли вона видає хороші результати на прикладах, що входять в навчальну вибірку, але практично не працює на інших прикладах.

У секції "Активаційна функція" необхідно визначити тип функції активації нейронів і її крутизну. Для цього в списку "Тип функції" слід вибрати потрібну функцію активації, а в поле "Крутизна" - задати її крутизну (також крутизну можна задати за допомогою повзунка, розташованого нижче). У нижній частині вікна відображається графік обраної функції відповідно до встановленої крутизною.

Візуалізатор графа нейромережі буде доступний тільки для тих гілок сценарію, які були створені за допомогою нейронної мережі. Він дозволяє графічно представити нейронну мережу з усіма її нейронами і синаптичeskими зв'язками. При цьому користувач може побачити не тільки структуру нейронної мережі, але і значення ваг, які приймають ті чи інші нейрони. Залежно від ваги нейрона він відображається певним кольором, а відповідне значення можна визначити за колірною шкалою, розташованою внизу вікна.

Граф змодельованої нейромережі наведено на рис.2.3.

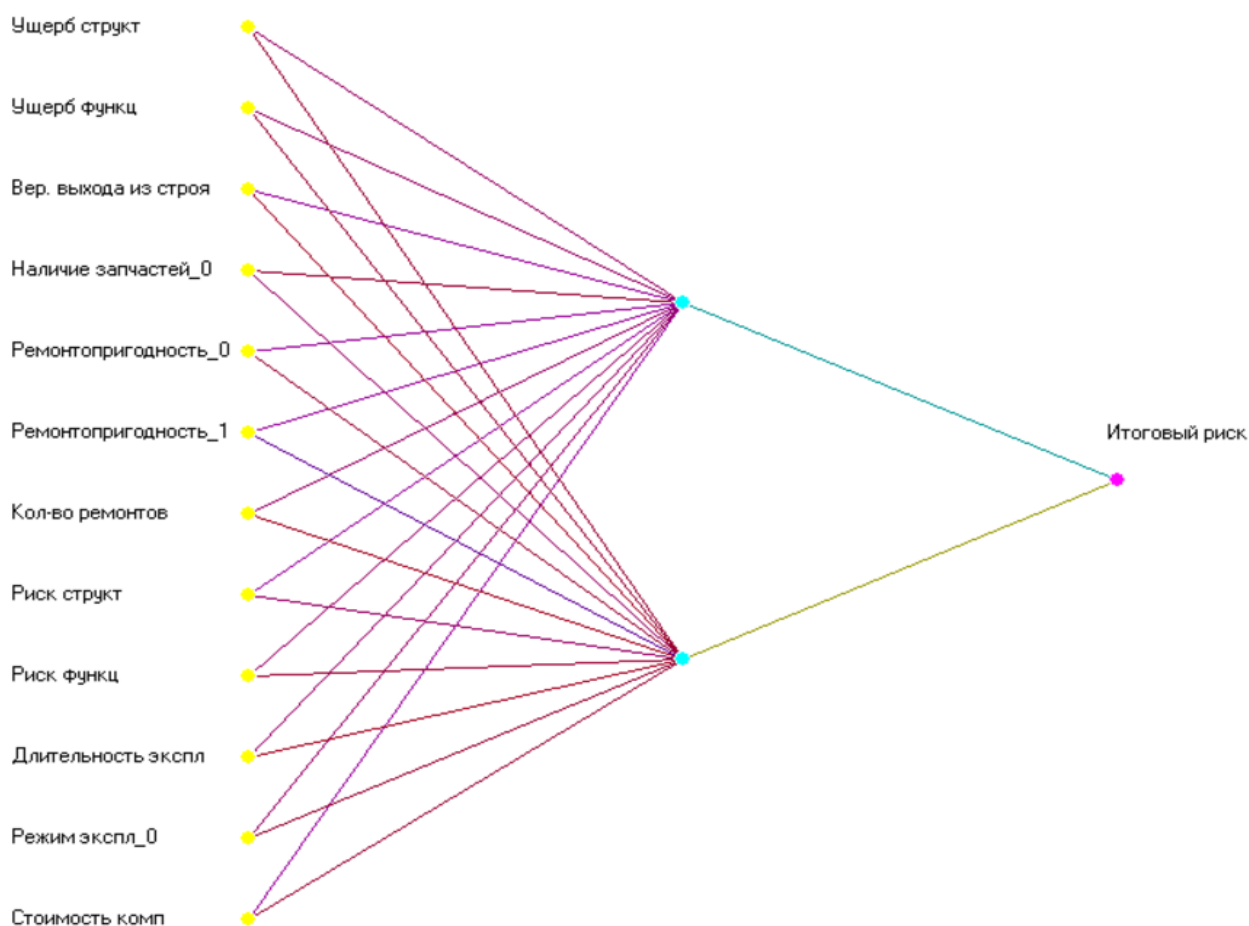


Рис. 2.3. Граф нейромережі

На діаграмі розсіювання (рис.2.4.) кожному спостереженню (або елементарної одиниці набору даних) відповідає точка, координати якої (в декартовій системі координат) дорівнюють значенням двох якихось параметрів цього спостереження. Якщо передбачається, що один з параметрів залежить від іншого, то зазвичай значення незалежного параметра відкладається по горизонтальній осі, а значення залежного - по вертикальній. Діаграми розсіювання використовуються для демонстрації наявності або відсутності кореляції між двома змінними

При необхідності можна збільшувати масштаб перегляду тих областей діаграми розсіювання, які потребують більш детального дослідження. Для цього досить, утримуючи ліву кнопку миші, виділити на діаграмі відповідну область. Після того, як миша буде відпущена, масштаб перегляду виділеної області буде збільшено. При цьому рух покажчика миші має здійснюватися зліва направо і зверху вниз. Аналогічну дію з рухом миші справа наліво, незалежно від розміру виділеної області, поверне діаграмі вихідний розмір. Утримуючи праву кнопку миші, можна переміщати діаграму, вивчаючи різні її області.

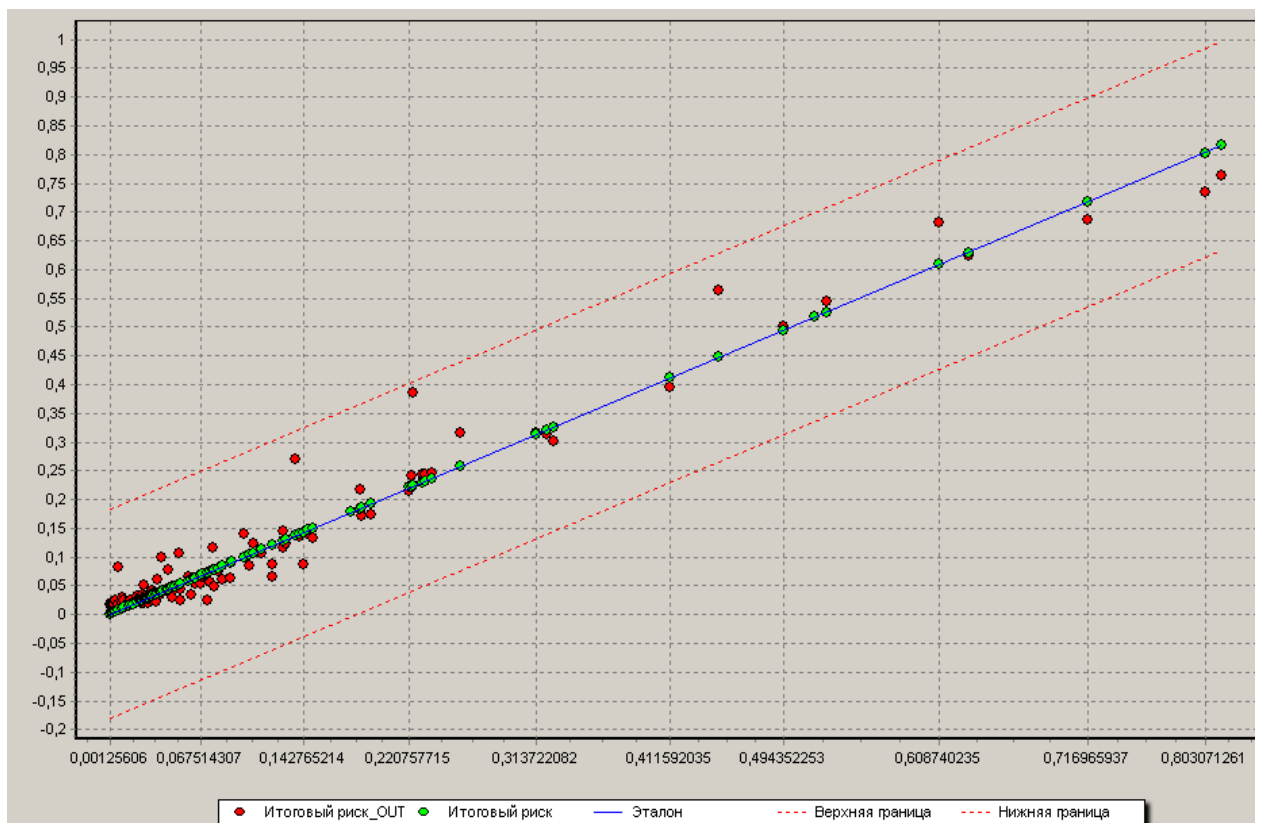


Рис. 2.4. Діаграма розсіювання

За графіками, які з'являтимуться у вікні оцінки якості, можна оцінити якість побудованої моделі і порівняти різні моделі між собою. Червоним

кольором тут показана гістограма щільності розподілу помилок, синім кольором - емпірична функція розподілу помилок.

По вертикальній осі показано кількість записів, у яких помилка потрапляє до відповідного інтервал.

На горизонтальній осі гістограми відкладаються інтервали помилок, як підписів по осі вказано значення верхньої межі інтервалу. Колонка відповідає інтервалу $[0. \langle \text{Значення1 по X} \rangle)$, наступний стовпець - $(\langle \text{Значення1 по X} \rangle. \langle \text{Значення2 по X} \rangle]$ і так далі.

У дужках відображається відсоток, який складають ці записи у всьому наборі. Функція розподілу помилок визначає відсоток записів, у яких помилка менше або дорівнює значенню по осі X.

Порівнюючи діаграми розподілу помилок для різних моделей, можна вибрати кращу модель, тобто модель з найкращим видом функцій розподілу (щільності розподілу) помилок.

РОЗДІЛ 3

РОЗРОБКА НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ПРОГНОЗУВАННЯ РИЗИКІВ ВИХОДУ З ЛАДУ КОМПОНЕНТІВ СУДНОВИХ СИСТЕМ

3.1 Розробка алгоритму роботи нейронної мережі та діаграми зв'язків

Діаграма зв'язків в рамках даної роботи реалізується у вигляді дерева схеми, на якій зображені слова, ідеї, завдання або інші поняття, пов'язані гілками, що відходять від центрального поняття або ідеї. Перевагою є легкість отримання загального уявлення про закладені в діаграму знання та структурованість інформації. До недоліків відносять сфокусованість на єдиному центральному понятті і обмежену масштабованість. Розроблена діаграма зв'язків наведена на рис.3.1. Розроблений алгоритм побудови нейронної мережі наведено на рис.3.2.

Кафедра КІТ (47)				НАУ 20 16 80 000 ПЗ			
<i>Виконав</i>	<i>Паренюк Л.В.</i>			РОЗРОБКА НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ПРОГНОЗУВАННЯ РИЗИКІВ ВИХОДУ З ЛАДУ КОМПОНЕНТІВ СУДНОВИХ СИСТЕМ	<i>Літера</i>	<i>аркуш</i>	<i>аркушів</i>
<i>Керівник</i>	<i>Воронін А.М.</i>					40	28
<i>Консульт.</i>					УС-111М 122		
<i>Н. контроль</i>	<i>Райчев І.Е.</i>						

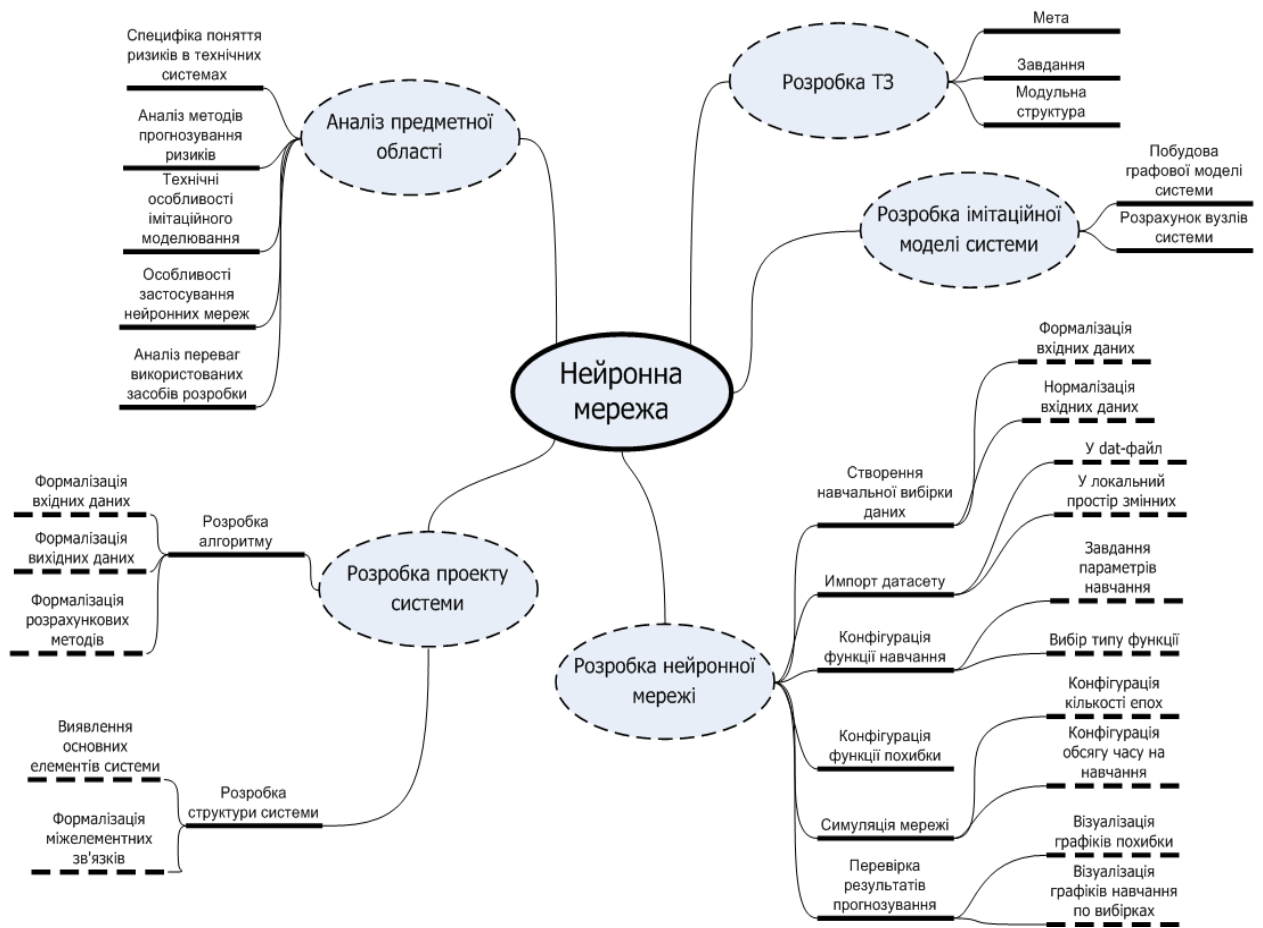


Рис. 3.1. Розроблена структура



Рис. 3.2. Розроблений алгоритм побудови нейронної мережі

Спочатку користувач запускає виконуваний файл розробленої програми, після чого здійснюється завантаження усіх компонентів системи та візуалізація головної форми. Після цього користувач повинен здійснити імпорт відповідних вхідних даних (збитки структурний і функціональний, ймовірність виходу з ладу, ризики структурний і функціональний, тривалість експлуатації, режим експлуатації, вартість) та вектору вихідної змінної

(підсумковий ризик). На базі цього відбувається створення моделі нейромережі, завдання параметрів її тренування, конфігурування налаштувань тренування моделі, безпосереднє тренування нейронної мережі, оцінка результатів тренування (у вигляді розмірів отриманих похибок та оцінок) та експорт даних до mat-файлу.

3.2 Розробка інтерфейсу головної форми та імпорту даних до системи

Інтерфейс головної форми розробленої системи на базі використання фреймворку Guide наведено на рис.3.3.

Для гнучкості користувальницького інтерфейсу та швидкості роботи с системою реаліовано 4 групи кнопок: «Імпорт», «Перегляд», «Запуск» та «Інформація».

Загалом, на головній формі системи розміщено наступні програмні управляючі кнопки:

1. Імпорт вхідних даних (input data) – забезпечує завантаження вхідних векторів даних до робочого простору системи.
2. Імпорт вихідних даних (target data) – забезпечує завантаження вихідного вектора даних до робочого простору системи.
3. Перегляд mind-map – забезпечує візуалізацію розробленої діаграми зв'язків (рис.3.1).
4. Перегляд алгоритму – забезпечує візуалізацію розробленого алгоритму роботи системи (рис.3.2).
5. Перегляд моделі нейромережі – забезпечує візуалізацію структурних схем розроблених моделей нейромережі.

6. Перегляд графіків – відкриває графіки похибок та помилок по результатам проведеного тренування розробленої нейронної мережі.

7. Налаштування мережі – відкриває форму Network Data Manager, що дозволяє обрати імпортовані вхідні та вихідні дані системи для побудови нейромережі.

8. Навчання мережі – відкриває модуль Neural Network Training для ініціалізації процесу навчання.

9. Створення мережі – відкриває модуль Neural Network Fitting Tool для виконання управляючих дій по створенню нейромережі.

10. Перевірка мережі – завантажує тестову вибірку та тестує роботу розробленої нейронної мережі.

11. Про програму – відкриває вікно інформації про розробника даної системи.

12. Вихід – припиняє роботу програми.

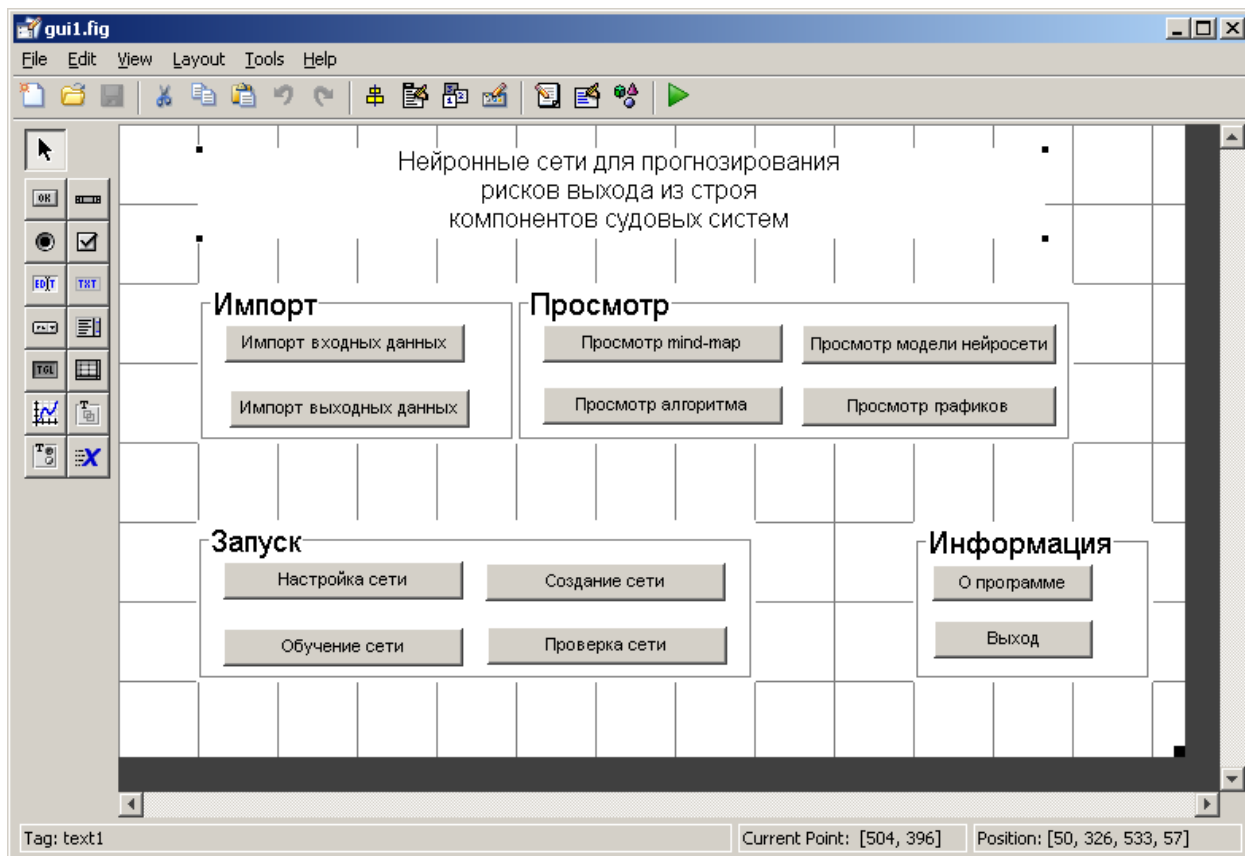


Рис. 3.3. Інтерфейс головної форми розробленої системи

Інтерфейс вікна імпорту даних вхідних сигналів у розроблену систему наведено на рис.3.4.

Дане вікно містить можливості:

- імпорту вхідних даних з робочого простору системи Matlab;
- імпорту вхідних даних з окремого .mat файлу, що розташовано на системному носії інформації;
- обиранню створеної нейромережі.

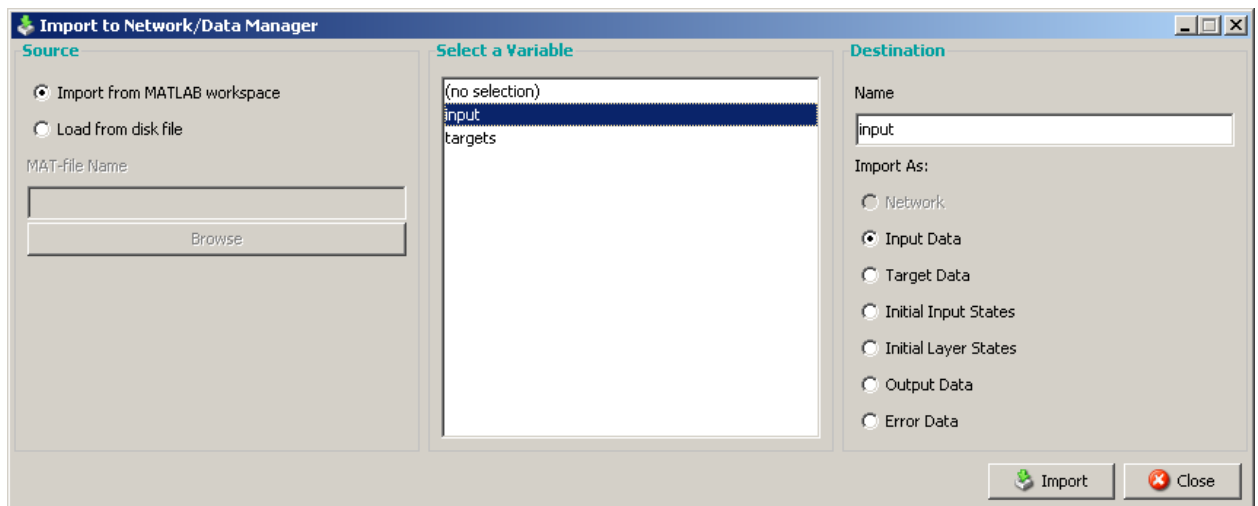


Рис. 3.4. Інтерфейс вікна імпорту даних вхідних сигналів у розроблену систему

Результат успішного імпорту даних по вхідних сигналах до розробленої системи наведено на рис.3.5. Для підтвердження створення вибірки необхідно натиснути кнопку ОК.

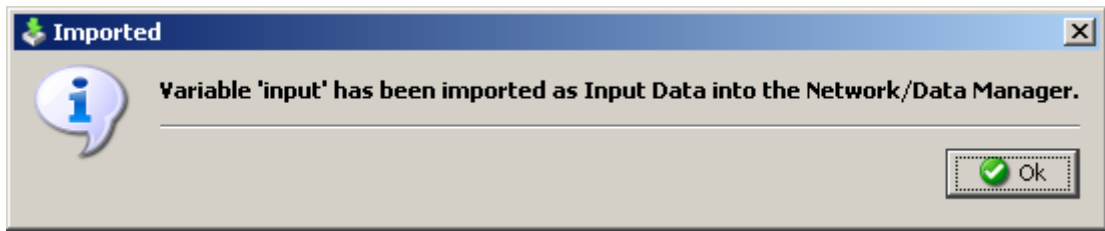


Рис. 3.5. Результат імпорту даних по вхідних сигналах до розробленої системи

Інтерфейс вікна імпорту даних вихідних сигналів у розроблену систему наведено на рис.3.6.

Дане вікно містить можливості:

- імпорту вихідних даних з робочого простору системи Matlab;
- імпорту вихідних даних з окремого .mat файлу, що розташовано на системному носії інформації;
- обираю створеної нейромережі.

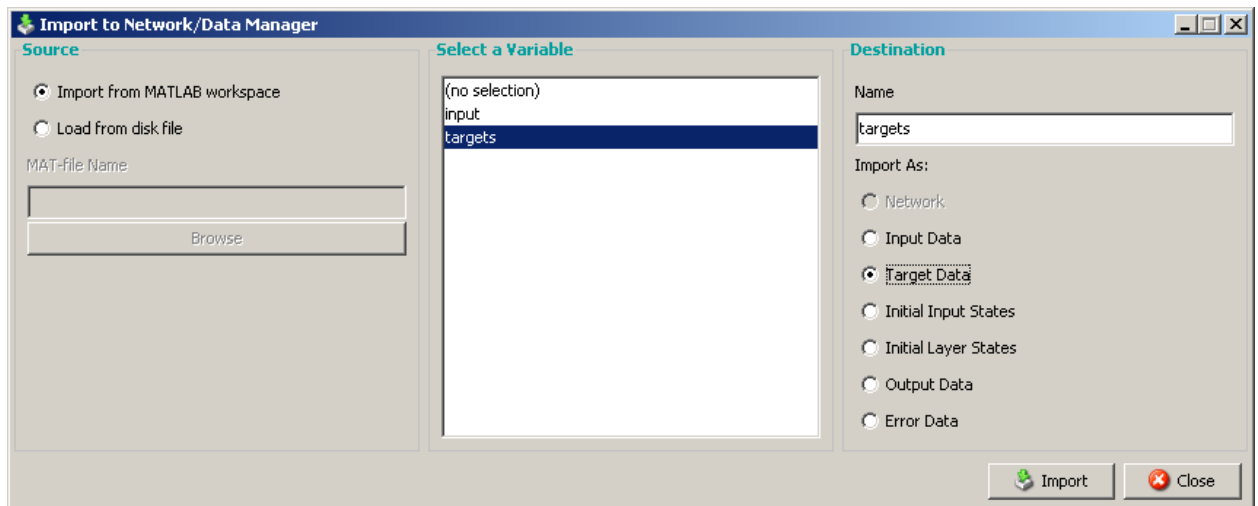


Рис. 3.6. Інтерфейс вікна імпорту даних вихідних сигналів у розроблену систему

Результат успішного імпорту даних по вихідних сигналах до розробленої системи наведено на рис.3.7. Для підтвердження створення вибірки необхідно натиснути кнопку ОК.

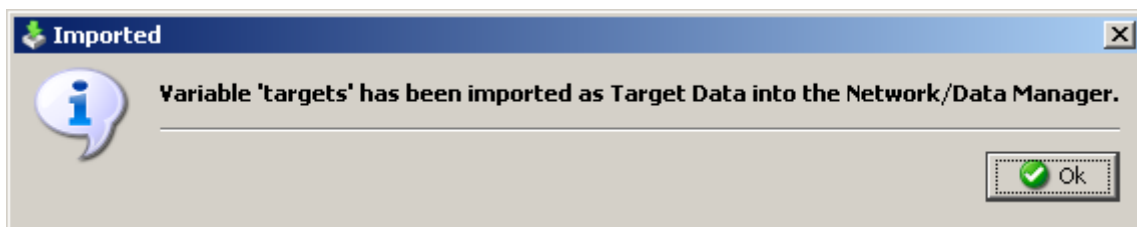


Рис. 3.7. Результат імпорту даних по вихідних сигналах до розробленої системи

Після імпорту даних до системи у користувача є можливість переглянути структуру вхідних та вихідних даних за допомоги спеціального модульного вікна, що містить текстове поле.

Вікно візуалізації структури імпортованих даних для побудови нейромережі наведено на рис.3.8.

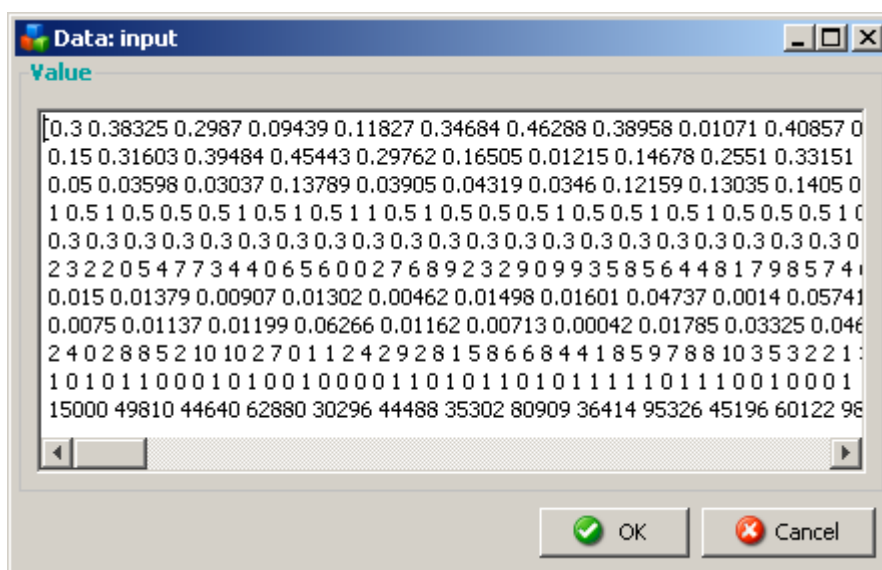


Рис. 3.8. Вікно візуалізації структури імпортованих даних для побудови нейромережі

3.3 Моделювання процесу тренування створеної нейронної мережі за допомоги Neural Network Toolbox

Для моделювання створеної нейронної мережі необхідно використати функцію тренування TRAINLM.

TRAINLM - функція тренування мережі, яка модифікує значення ваг і зсувів відповідно до методу оптимізації Левенберга-Маркара.

Вхідні параметри:

- NET - нейронна мережа;
- Pd - вектори вхідних затримок;
- Tl - вектори еталонів шару;
- Ai - початкові умови вхідних затримок;
- Q - розмір пакета;
- TS - тимчасові кроки;
- VV - порожня матриця або структура контрольних векторів;
- TV - порожня матриця або структура тестових векторів;

Вихідні параметри:

- NET - тренована мережу;
- TR - запис, що включає параметри тренування,
- TR.epoch - кількість епох тренування;
- TR.perf - характеристика тренування;
- TR.vperf - характеристика контрольної перевірки;
- TR.tperf - характеристика тестування;
- TR.mu - значення μ .

Тренування реалізується відповідно до параметрів функції навчання TRAINLM. Ці параметри разом з їх значеннями за замовчуванням перераховані нижче:

- net.trainParam.epochs - (100) - Максимальна кількість епох тренування;
- net.trainParam.show - (25) - Кількість епох між графіками;
- net.trainParam.goal - (0) - Умова зупинки по відхиленню від еталону;
- net.trainParam.time - (inf) - Максимальний час тренування в секундах;
- net.trainParam.min_grad - (1e-10) - Мінімальний градієнт;
- net.trainParam.mu - (0.001) - початкове значення μ ;
- net.trainParam.mu_dec - (0.1) - коефіцієнт зменшення μ ;
- net.trainParam.mu_inc - (10) - коефіцієнт зростання μ ;
- net.trainParam.mu_max - (1e10) - максимальне значення μ ;
- net.trainParam.mem_reduc - 1 - коефіцієнт використання пам'яті;
- net.trainParam.max_fail - (5) - Максимальна кількість помилок на контрольному масиві.

Розмірності змінних:

1. Pd - NoxNixTS масив осередків, кожен елемент якого P {i, j, ts} - матриця D_{ijxQ},
2. Tl - NlxTS масив осередків, кожен елемент якого P {i, ts} - матриця V_{ixQ},
3. Ai - NlxLD масив осередків, кожен елемент якого Ai {i, k} - матриця S_{ixQ},

де

- Ni = net.numInputs
- Nl = net.numLayers
- LD = net.numLayerDelays
- Ri = net.inputs {i} .size

- $S_i = \text{net.layers } \{i\} .\text{size}$
- $V_i = \text{net.targets } \{i\} .\text{size}$
- $D_{ij} = R_i * \text{length}(\text{net.inputWeights } \{i, j\} .\text{delays})$

Якщо VV або TV не [], то є структуру контрольних / тестових векторів:

- VV.PD, TV.PD - Затримки контрольних / тестових входів;
- VV.T1, TV.T1 - Контрольні / тестові зразки шару;
- VV.Ai, TV.Ai - Вхідні початкові умови контрольного / тестового

масиву;

- VV.Q, TV.Q - Розмір пакету контрольних / тестових масивів;
- VV.TS, TV.TS - Тимчасові кроки контрольного / тестового масиву.

Контрольні вектори використовуються для дострокової зупинки тренування якщо показник функціонування мережі на контрольному масиві векторів перестане поліпшуватися або залишатиметься на одному рівні для MAX_FAIL епох поспіль.

TRAINLM (CODE) повертає корисну інформацію для кожної з рядків CODE:

- 'rnames' - Імена параметрів тренування;
- 'rdefaults' - Параметри тренування за замовчуванням.

Стандартну мережу, яка використовує TRAINLM можна створити за допомогою функцій NEWFF, NEWCF або NEWELM.

Вікно конфігурації параметрів нейромережі наведено на рис.3.9.

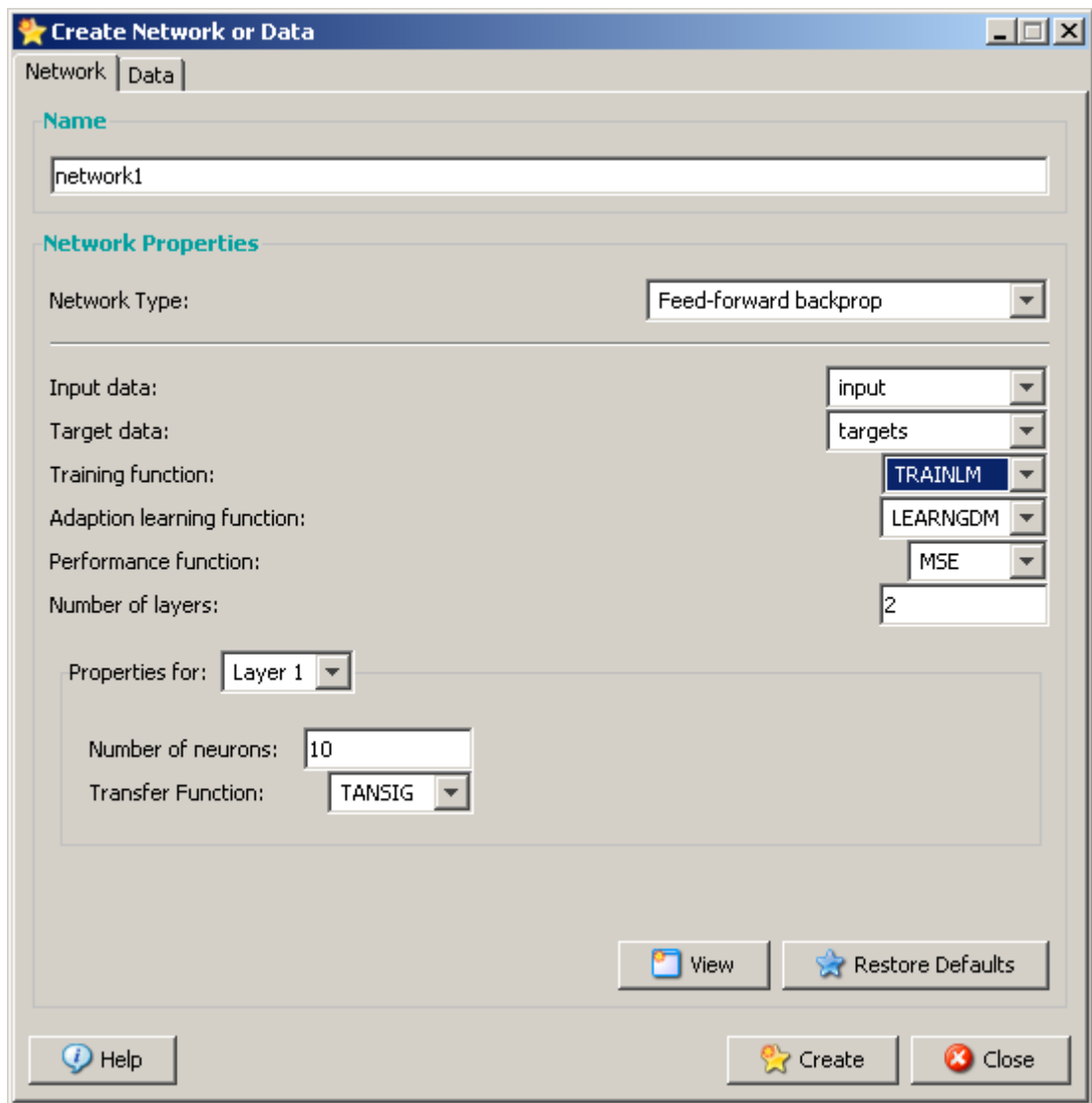


Рис. 3.9. Вікно конфігурації параметрів нейромережі

При завданні усіх необхідних параметрів у даному вікні можна натиснути кнопку «View», яка дозволяє візуалізувати структурний вигляд розробленої нейронної мережі. Результат наведено на рис.3.10.

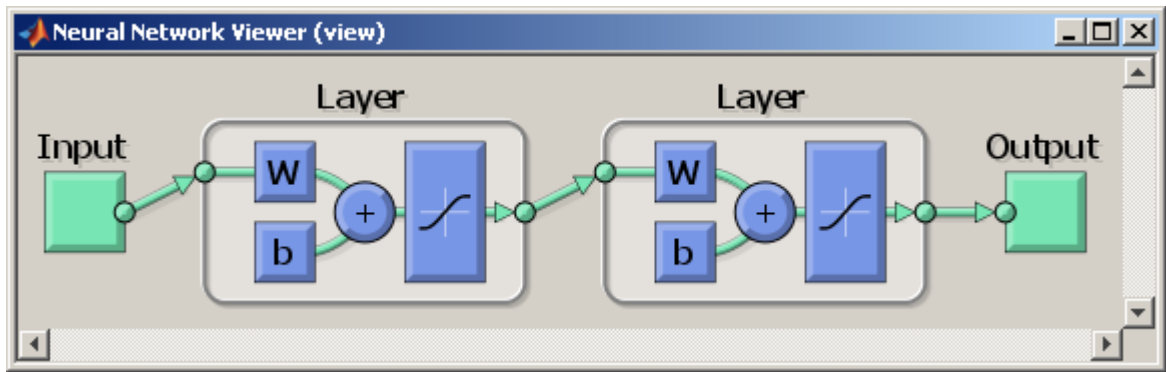


Рис. 3.10. Схематичний вигляд структури розробленої нейромережі

При натисканні кнопки «Create» створюється нове налаштування мережі. Вікно інформування про успішне створення нейромережі наведено на рис.3.11.

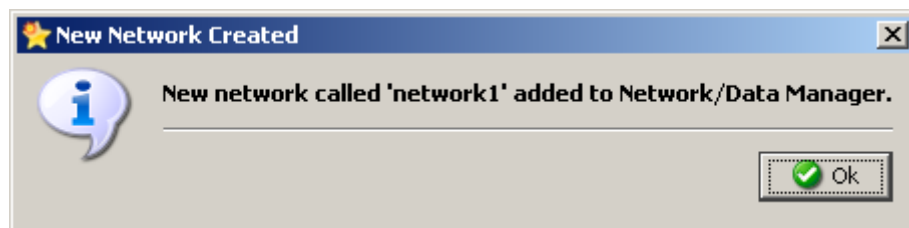


Рис. 3.11. Вікно інформування про успішне створення нейромережі

Для того, щоб створити для користувача мережу, яку необхідно тренувати за допомогою TRAINLM було:

1) Встановлено NET.trainFcn як 'trainlm'. Тим самим NET.trainParam буде за замовчуванням TRAINLM.

2) Встановлено необхідні значення для властивостей NET.trainParam.

Виклик TRAIN з готовою мережею забезпечить тренування мережі за допомогою TRAINLM.

TRAINLM може навчати будь-яку мережу, якщо її вагові, вхідні і активаційні функції мають похідні.

Параметр MEM_REDUС визначає кількість пам'яті, необхідне для обчислення якобіана JX , а також швидкість обчислень. При MEM_REDUС = 1 швидкість тренування максимальна, але при цьому потрібно найбільшу кількість пам'яті. При збільшенні MEM_REDUС швидкість тренування знижується, але одночасно зменшується обсяг використовуваної пам'яті.

Тренування зупиняється, якщо виконується одна з наступних умов:

- 1) Досягнуто максимальне значення кількості епох "EPOCHS".
- 2) Ви перевищили значення максимального часу тренування "TIME".
- 3) Ефективність функціонування досягне значення "GOAL".
- 4) Градієнт ефективності функціонування зменшиться нижче "MINGRAD".
- 5) Значення MU досягає MU_MAX
- 6) Контрольне значення функціонування збільшилася більш ніж "MAX_FAIL" раз поспіль після того, як воно в останній раз зменшилася (при використанні контрольного масиву)..

У разі створення нейронної мережі слід завдати відповідні параметри її роботи у вкладках «Train», «Simulate», «Adapt». Вікно завдання вхідних даних для тренування розробленої нейромережі наведено на рис.3.12.

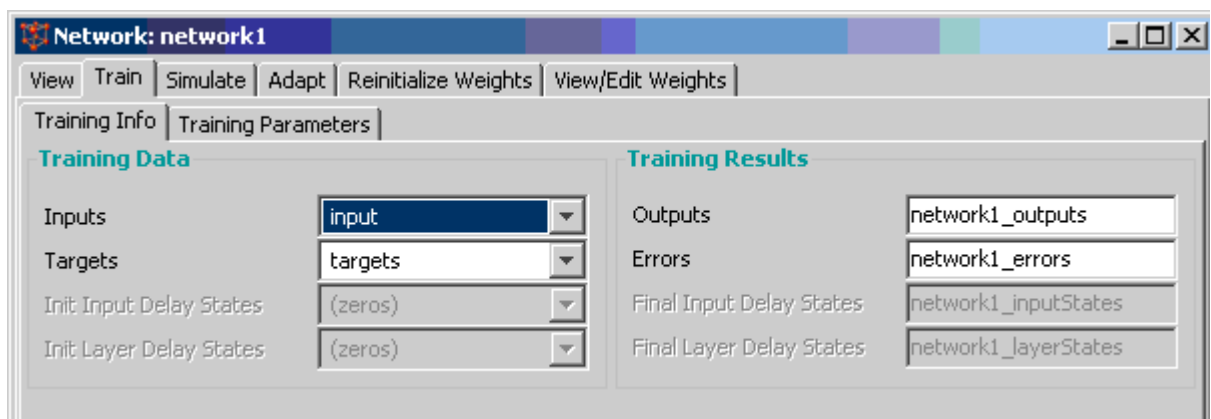


Рис. 3.12. Вікно завдання вхідних даних для тренування розробленої нейромережі

Вікно завдання параметрів тренування розробленої нейромережі наведено на рис.3.13.

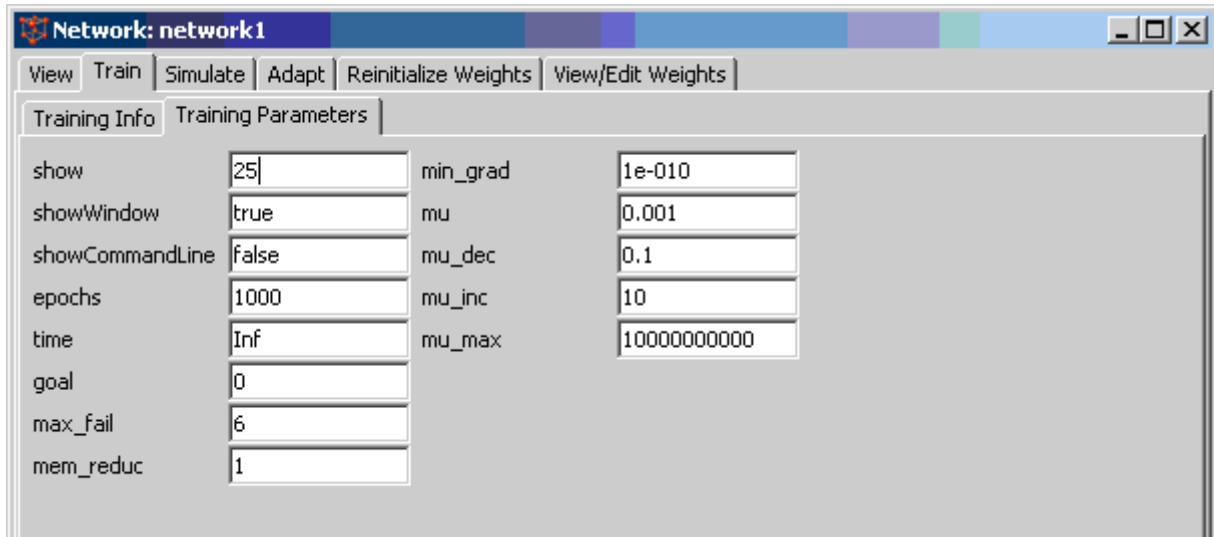


Рис. 3.13. Вікно завдання параметрів тренування розробленої нейромережі

Серед завдаваних параметрів у вкладці «Training Parameters» є наступні: show, showWindow, showCommandLine, epochs, time, goal, max_fail, mem_reduc, min_grad, mu, mu_dec, mu_inc, mu_max.

Вікно завдання параметрів симуляції роботи розробленої нейромережі наведено на рис.3.14. Вікно завдання параметрів адаптації розробленої нейромережі наведено на рис.3.15. В цих вікнах необхідно завдати вхідні та цільові вектори даних для роботи нейромережі.

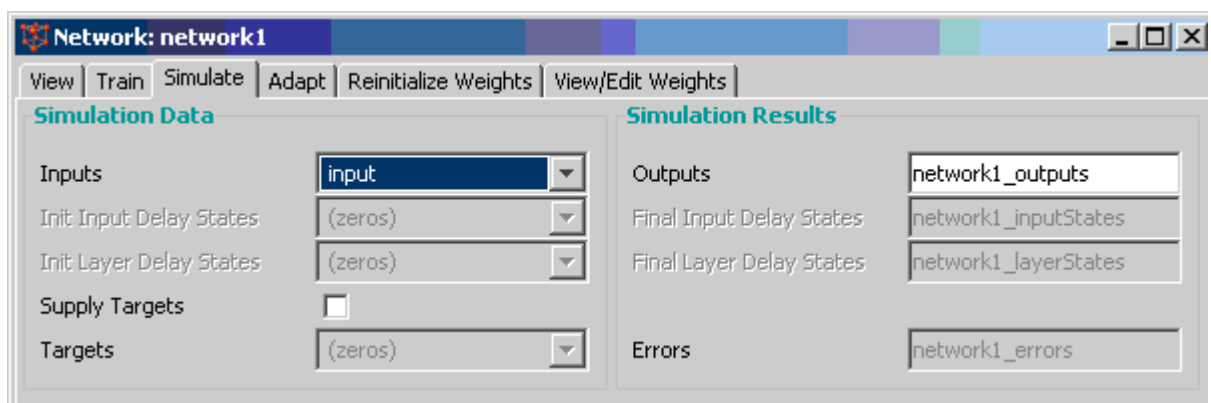


Рис. 3.14. Вікно завдання параметрів симуляції роботи розробленої неймережі

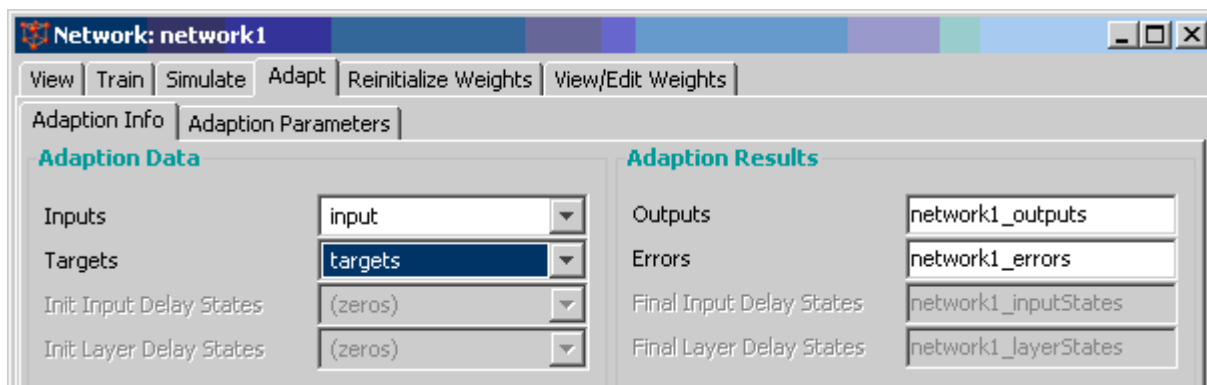


Рис. 3.15. Вікно завдання параметрів адаптації розробленої неймережі

Далі, для візуалізації використовуємо nntraintool, який здійснює відкриття нейронної мережі для моніторингу процесу її навчання.

Це вікно може бути визвало для того, щоб зробити видимим процес навчання, від початку тренування до її завершення

Для того, щоб отримати доступ до додаткових статистичних даних, що відображає процес навчання, під час або після тренування, можна вибрати відповідні кнопки: Perfomance, Training State, Regression.

Вигляд форми візуалізації алгоритму та процесу навчання розробленої неймережі наведено на рис.16.

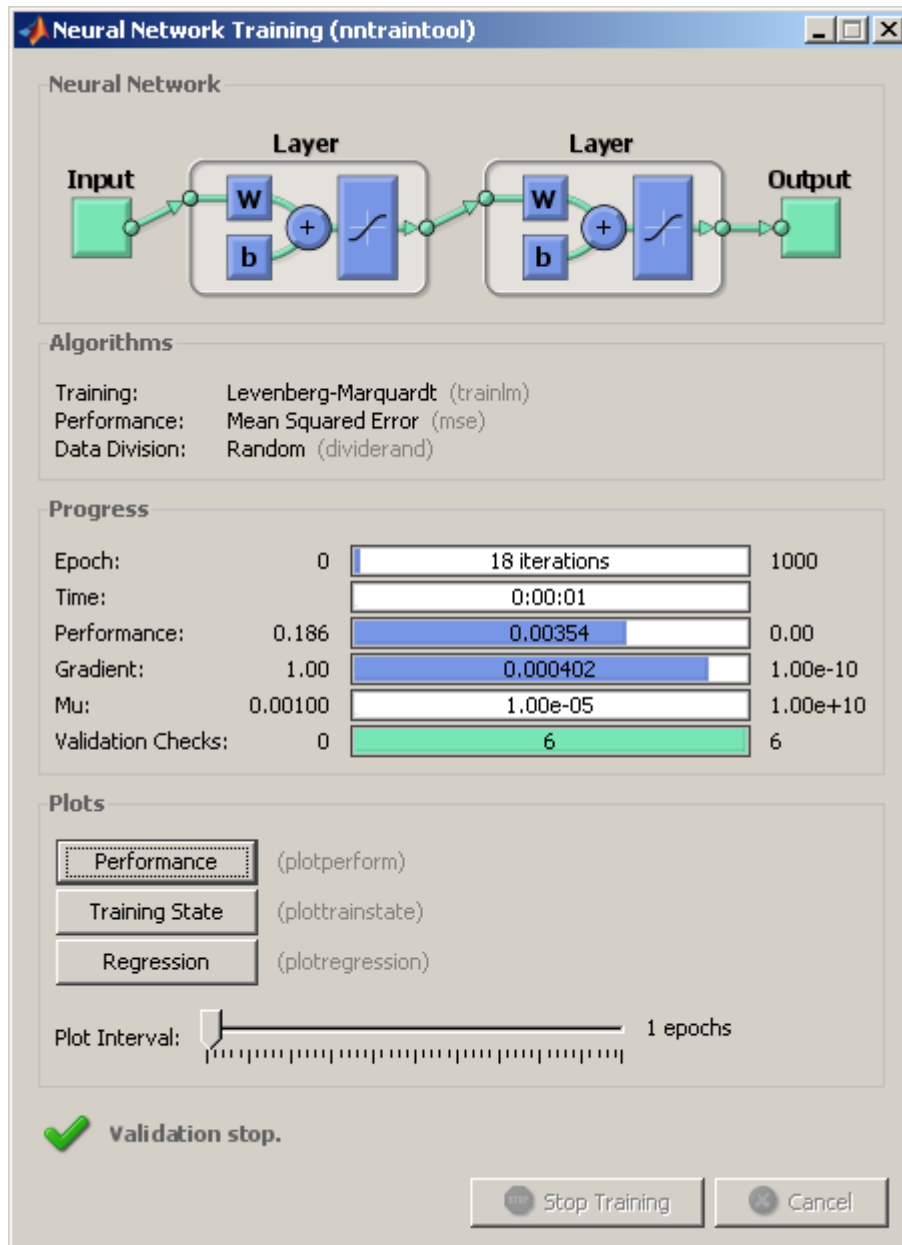


Рис. 3.16. Вигляд форми візуалізації алгоритму та процесу навчання розробленої нейромережі

Процес навчання ілюструється графіком залежності оцінки функціонування від номера циклу навчання.

Графік залежності середньоквадратичної помилки навченої нейромережі від кількості пройдених епох наведено на рис.3.17.

Синім кольором відображена функціонування тренувального процесу, зеленим – функціонування валідаційного процесу, червоним – тестового процесу, а крапка перетину пунктирних ліній – місце найкращого функціонування нейромережі, після якої мережа починає функціонувати з більшими похибками, тобто починається процес перенавчання.

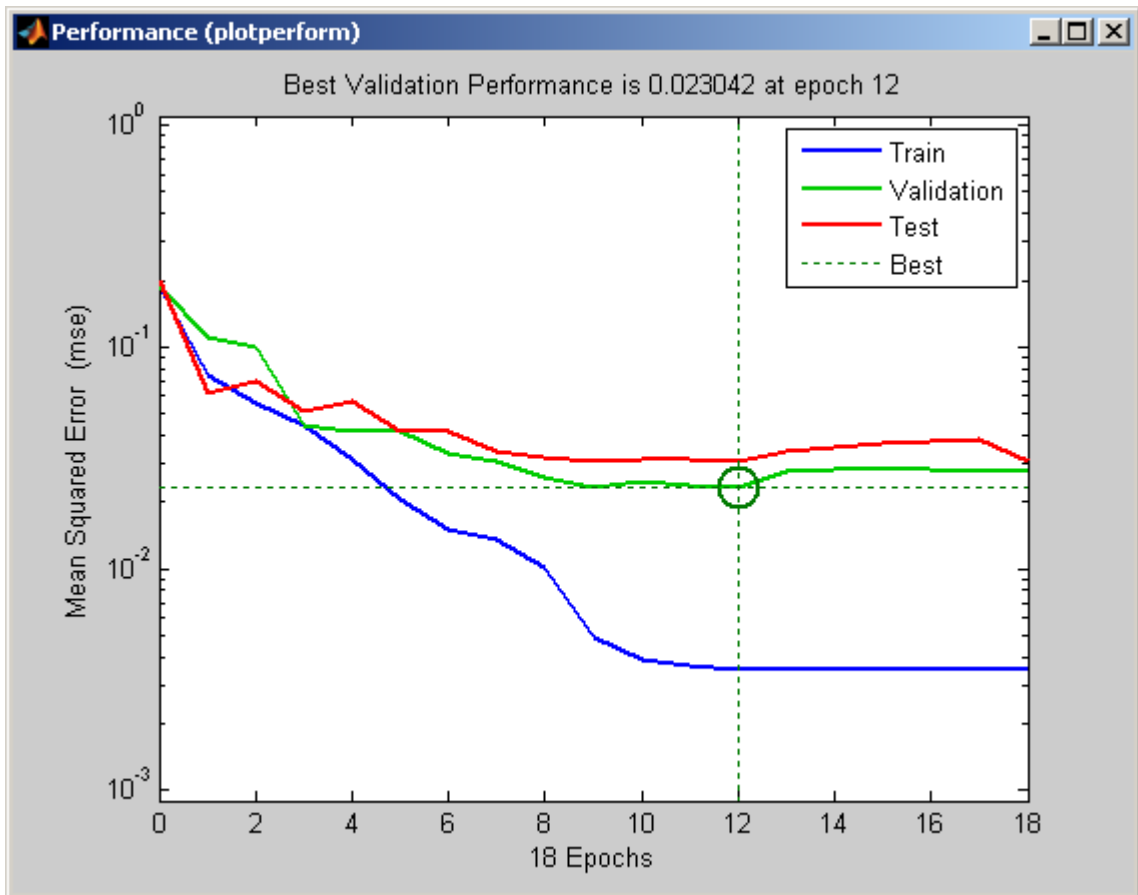


Рис. 3.17. Графік залежності середньоквадратичної помилки навченої нейромережі від кількості пройдених епох

Графіки відображення стану тренування нейромережі наведено на рис.3.18.

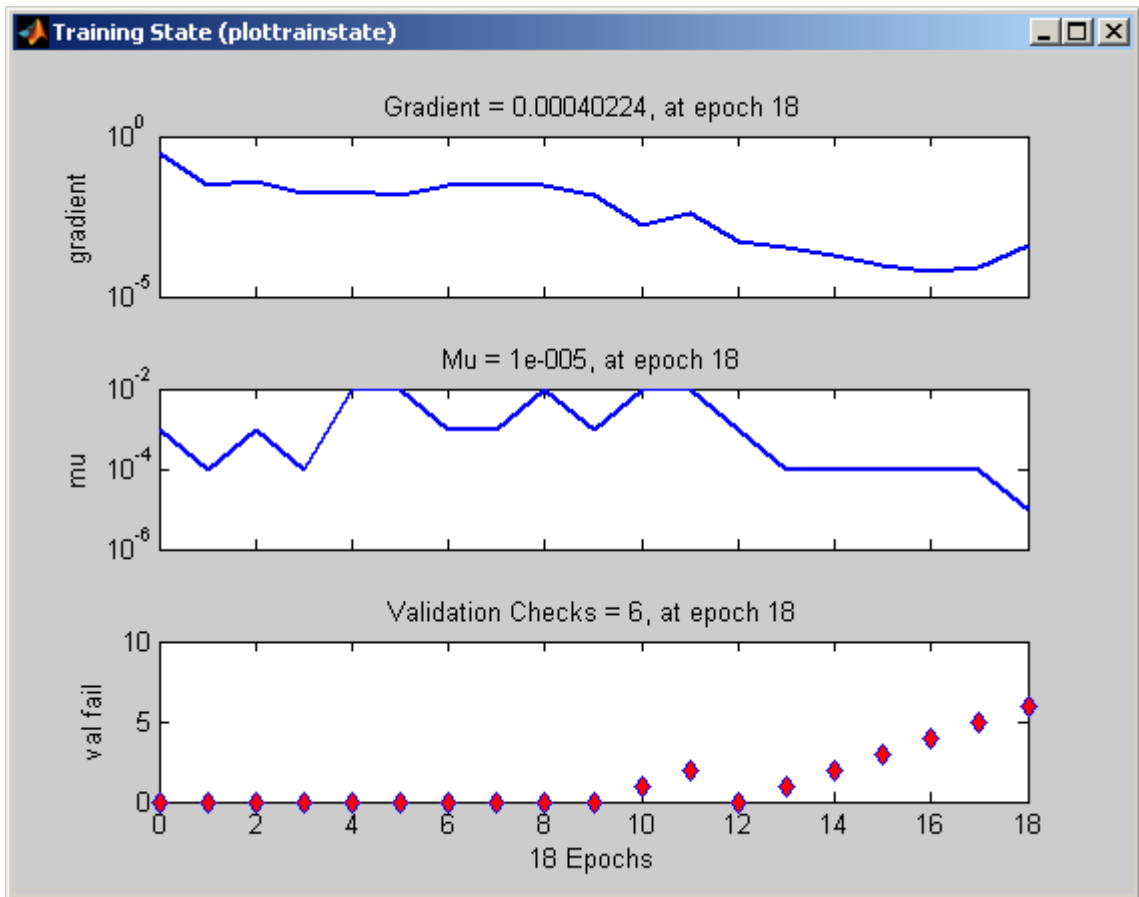


Рис. 3.18. Графіки відображення стану тренування нейромережі

Перед тим, як скористатися нейронною мережею, необхідно досліджувати ступінь достовірності результатів обчислень мережі на тестовому масиві вхідних векторів. В якості тестового масиву необхідно використовувати масив, компоненти якого відрізняються від компонентів масиву, використаного для навчання. У нашому випадку для отримання тестового масиву досить скористатися ще раз програмою `mas1`.

Для оцінки достовірності результатів роботи мережі можна скористатися результатами регресійного аналізу, отриманими при порівнянні еталонних значень зі значеннями, отриманими на виході мережі коли на вхід подано вхідні вектори тестового масиву. У середовищі MATLAB для цього можна скористатися функцією `postreg`. Результати використання цієї функції наведено на рис.3.19-3.22.

Графік відображення результатів навчання нейромережі на вхідній виборці даних наведено на рис.3.19. Еталонні значення векторів відкладено у вигляді пунктирної лінії, окремі значення даних позначено маркерами типу «коло», кольорова лінія показує кореляцію по всім досліджуваним значенням моделі.

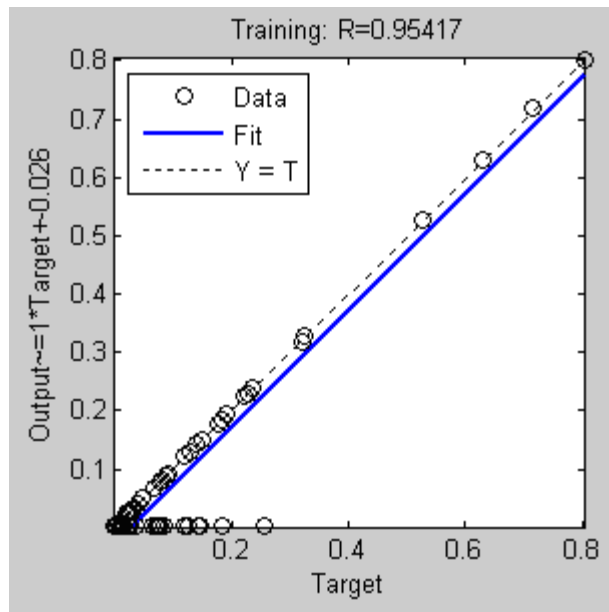


Рис. 3.19. Графік відображення результатів навчання нейромережі на вхідній виборці даних

Графік відображення результатів тренування нейромережі наведено на рис.3.20.

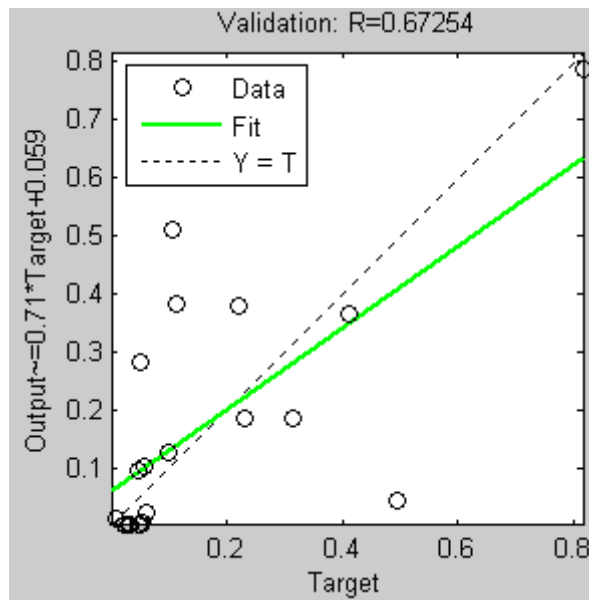


Рис. 3.20. Графік відображення результатів валідації нейромережі

Еталонні значення векторів відкладено у вигляді пунктирної лінії, окремі значення даних позначено маркерами типу «коло», кольорова лінія показує кореляцію по всім досліджуваним значенням моделі.

Графік відображення результатів тестування нейромережі на перевіірочній вибірці даних наведено на рис.3.21. Графік відображення загального розподілу даних нейромережі наведено на рис.3.22.

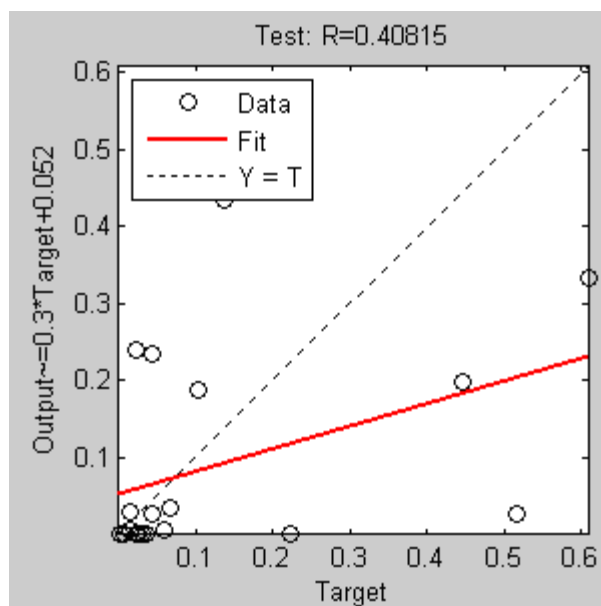


Рис. 3.21. Графік відображення результатів тестування нейромережі на перевіірчній вибірці даних

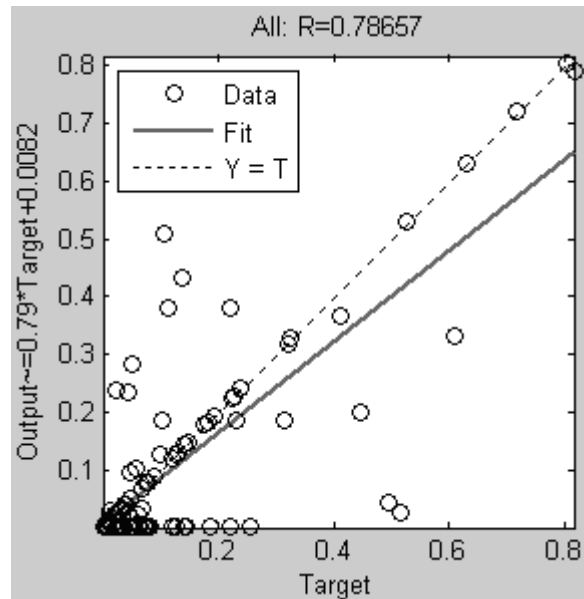


Рис. 3.22. Графік відображення загального розподілу даних нейромережі

3.4 Модифікація розробленої нейромережі за допомоги Neural Network Fitting Tool

Отримані результати тестування створеної нейронної мережі свідчать про необхідність додаткового коригування вагових коефіцієнтів для досягнення більшого рівня достовірності використання та зниження можливої похибки. Для цього скористаємося модулем Neural Network Fitting Tool. Вікно завдання вхідних вибірок для модифікації роботи нейромережі наведено на рис.3.23. В даному окні ми обираємо необхідні масиви вхідних та вихідних даних. Вікно розподілу вхідних даних на категорії валідації, верифікації та тестування нейромережі наведено на рис.3.24.

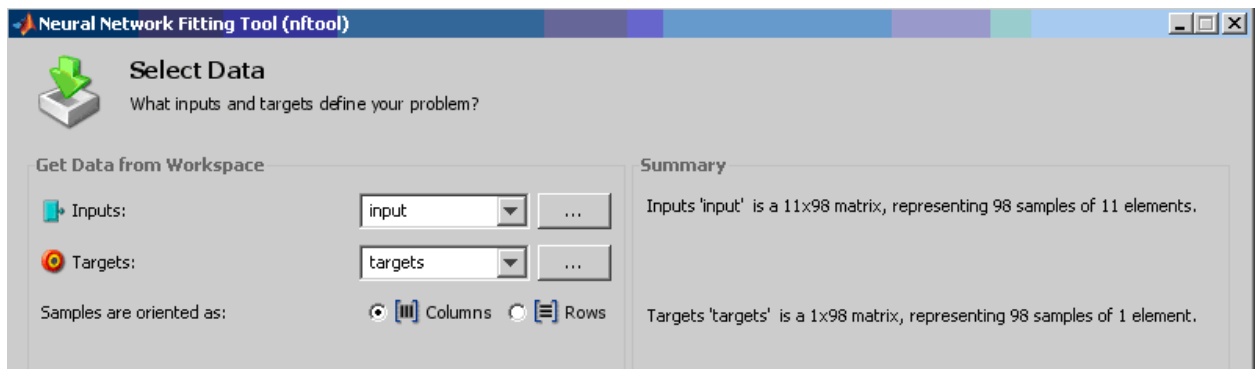


Рис. 3.23. Вікно завдання вхідних вибірок для модифікації роботи нейромережі

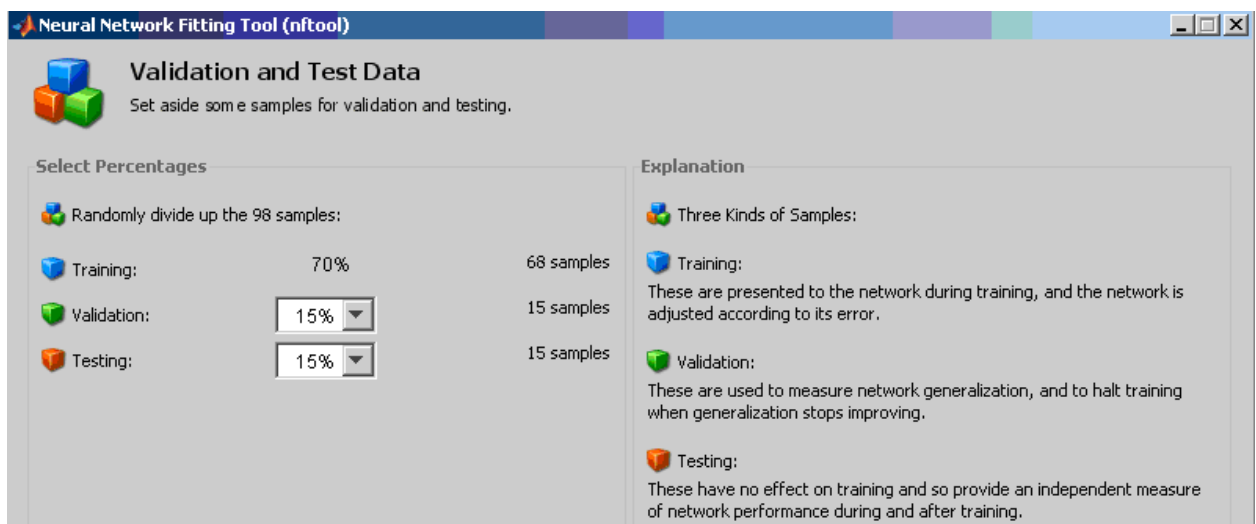


Рис. 3.24. Вікно розподілу вхідних даних на категорії валідації, верифікації та тестування нейромережі

В даному вікні ми можемо розподілити загальну вибірку значень на 3 не повторювані частини:

1. Для тренування нейромережі на базі обраного алгоритму та з урахуванням сконфігурованих параметрів.
2. Для валідації якості тренування та ступеню навчання розробленої нейромережі.

3. Для тестування точності та ефективності роботи розробленої нейромережі.

Вікно завдання кількості прихованих нейронів у розробленій нейромережі наведено на рис.3.25. Збільшемо їх кількість до 20.

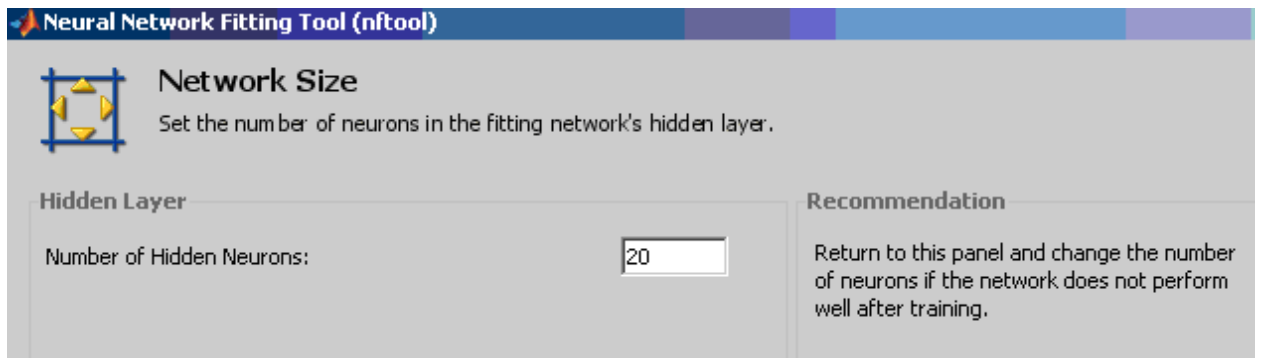


Рис. 3.25. Вікно завдання кількості прихованих нейронів у розробленій нейромережі

Кількість вхідних сигналів залишається такою самою, вихідна змінна – прогнозний ризик, також структурно не змінюється. Структура модифікованої нейромережі наведена на рис.3.26.

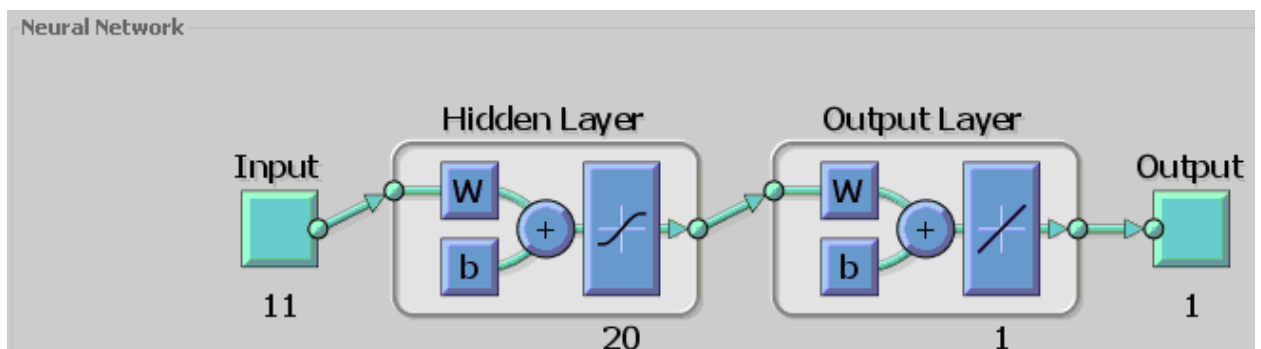


Рис. 3.26. Структура модифікованої нейромережі

Після формування усіх відповідних попередніх кроків відкриваємо форму перевірки та запуску моделювання тренування нейромережі. Форма запуску процесу навчання розробленої нейромережі наведена на рис.3.27.

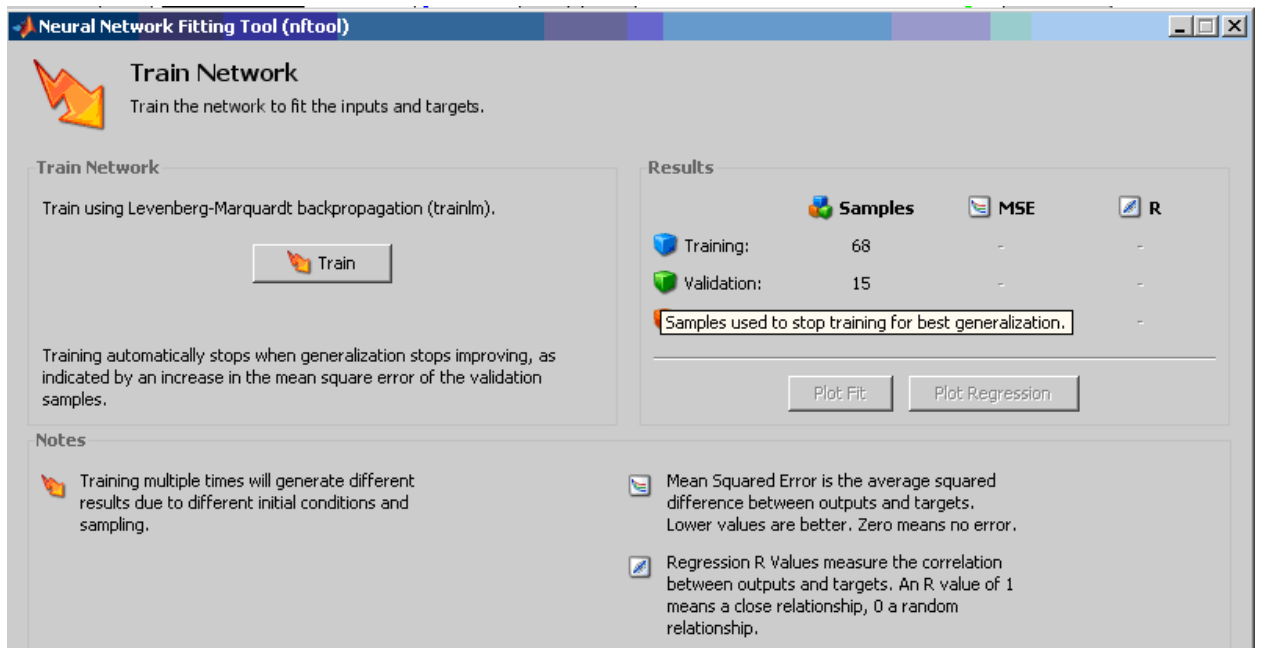


Рис. 3.27. Форма запуску процесу навчання розробленої нейромережі

В результаті модифікації створеної нейромережі шляхом зміни кількості нейронів та коригування вагових коефіцієнтів отримуємо нові графіки відображення відповідних значень R (рис.3.28-3.31).

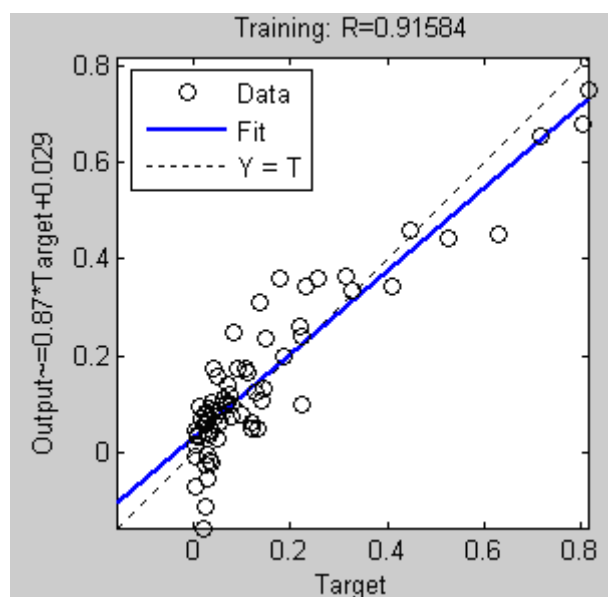


Рис. 3.28. Графік відображення результатів навчання модифікованої нейромережі на вхідній вибірці даних

Графік відображення результатів валідації модифікованої нейромережі наведено на рис.3.29.

Еталонні значення векторів відкладено у вигляді пунктирної лінії, окремі значення даних позначено маркерами типу «коло», кольорова лінія показує кореляцію по всім досліджуваним значенням моделі.

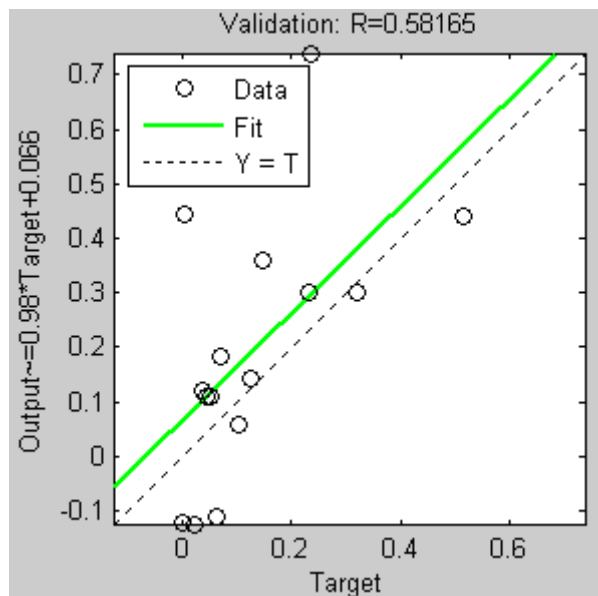


Рис. 3.29. Графік відображення результатів валідації модифікованої нейромережі

Графік відображення результатів тестування модифікованої нейромережі на перевіірчній вибірці даних наведено на рис.3.30.

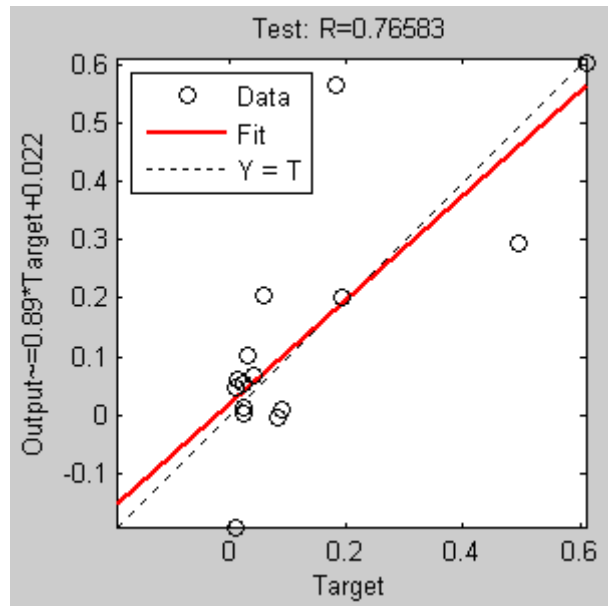


Рис. 3.30. Графік відображення результатів тестування модифікованої нейромережі на перевіірчній вибірці даних

Графік відображення загального розподілу даних модифікованої нейромережі наведено на рис.3.31. Еталонні значення векторів відкладено у вигляді пунктирної лінії, окремі значення даних позначено маркерами типу «коло», кольорова лінія показує кореляцію по всім досліджуваним значенням моделі. Загальна оцінка R стала більшою, ніж в попередньому варіанті моделювання, що свідчить про більшу точність та адекватність побудованої моделі.

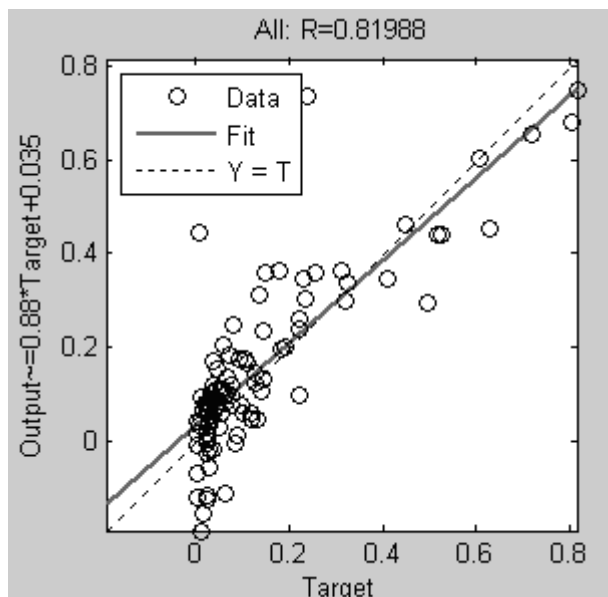


Рис. 3.31. Графік відображення загального розподілу даних модифікованої нейромережі

Форма збереження та експорту результатів розробленої нейромережі дозволяє задати ім'я та зберегти наступні дані, що були отримані при моделюванні нейромережі:

1. Структуру нейромережі до окремого файлу та до робочого простору Matlab.
2. Графіки похибок та набір даних до окремого файлу та до робочого простору Matlab.
3. Вектор прогнозних вихідних значень роботи нейромережі.
4. Вектор помилок та похибок до окремого файлу та до робочого простору Matlab.
5. Вектори вхідних та вихідних вибірок даних.

ВИСНОВКИ

В ході дипломної роботи була створена нейронна мережа для завдань прогнозування ризиків виходу з ладу компонентів.

Результати проведених досліджень дозволяють стверджувати, що поставлена мета роботи досягнута.

Для досягнення поставленої мети було виконано наступні завдання:

1. Проведено аналіз літературних джерел з досліджуваної тематики роботи.
2. Виконано аналіз методів та моделей прогнозування ризиків технічних систем.
3. Досліджено специфіку складу та застосування штучних нейронних мереж.
4. Обрано та обгрунтовано використані засоби розробки.
5. Спроектовано імітаційну модель паливної системи судна.
6. Розроблено діаграму зв'язків та алгоритм роботи штучної нейронної мережі.
7. Розроблено інтерфейс головної форми системи створення штучної нейронної мережі у середовищі Matlab.
8. Виконано моделювання та дослідження процесу тренування розробленої моделі штучної нейронної мережі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Фрейдзон И.Р. Судовые автоматизированные системы / И.Р. Фрейдзон. – Л.: Судостроение, 1988. – 365 с.
2. Крисилов В.А. Представление исходных данных в задачах нейросетевого программирования / Одесса: ОНПУ. 2003. С. 7.
3. Леван Д.Н., Феоктистов Н.А. Особенности использования многослойного персептрона/ Науковедени. вып. 2. 2014. С. 8
4. Шитиков В.К., Розенберг Г.С., Зинченко Т.Д. Методы системной идентификации / Тольятти: ИЭВБ РАН. 2003. 463 с..
5. Шахнов В.А., Власов А.И., Кузнецов А.С. Нейрокомпьютеры: архитектура и схемотехника / М.: Изд-во Машиностроение. 2000. 64 с.
6. McMillan C. The Connectionist Scientist Game: Rule Extraction and Refinement in a Neural Network / C. McMillan, M.C. Mozer, P. Smolensky // Proc. XIII Annual Conf of the Cognitive Science Society, Hillsdale, NJ, USA. – 2001.
7. Области применения искусственных нейронных сетей [Электроний ресурс//Основные направления использования/ URL: <http://www.neuropro.ru/> (дата звернення 20.11.2020)

Додаток А

Код програми

```
function net = create_fit_net(inputs,targets)
%CREATE_FIT_NET Creates and trains a fitting neural network.
%
% NET = CREATE_FIT_NET(INPUTS,TARGETS) takes these arguments:
% INPUTS - R×Q matrix of Q R-element input samples
% TARGETS - S×Q matrix of Q S-element associated target samples
% arranged as columns, and returns these results:
% NET - The trained neural network
%
% For example, to solve the Simple Fit dataset problem with this function:
%
% load simplefit_dataset
% net = create_fit_net(simplefitInputs,simplefitTargets);
% simplefitOutputs = sim(net,simplefitInputs);
%
% To reproduce the results you obtained in NFTOOL:
%
% net = create_fit_net(input',targets');

% Create Network
numHiddenNeurons = 20; % Adjust as desired
net = newfit(inputs,targets,numHiddenNeurons);
net.divideParam.trainRatio = 70/100; % Adjust as desired
net.divideParam.valRatio = 15/100; % Adjust as desired
net.divideParam.testRatio = 15/100; % Adjust as desired

% Train and Apply Network
[net,tr] = train(net,inputs,targets);
outputs = sim(net,inputs);

% Plot
plotperf(tr)
plotfit(net,inputs,targets)
plotregression(targets,outputs)
```