

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ ННЦЗФН _____
(повна назва)

Кафедра _____ Штучного інтелекту _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

_____ Еволюційна нейро-фаззі система з online навчанням-самонавчанням _____
_____ (тема)

Виконав:
студент 2 курсу, групи _____ СШЗм-20-1 _____
_____ Сараєв О.В. _____
(прізвище, ініціали)

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

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

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

Керівник _____ к.т.н, доц. Дейнеко А.О. _____
(посада, прізвище, ініціали)

Допускається до захисту

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

2021 р.

Харківський національний університет радіоелектроніки

Факультет _____ **ННЦЗФН** _____
(повна назва)
Кафедра _____ **Штучного інтелекту** _____
(повна назва)
Рівень вищої освіти _____ **другий (магістерський)** _____
Спеціальність _____ **122 Комп'ютерні науки** _____
(код і повна назва)
Тип програми _____ **освітньо-професійна** _____
(освітньо-професійна або освітньо-наукова)
Освітня програма _____ **Системи штучного інтелекту (СШІ)** _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ **Сараєву Олександровичу Володимировичу** _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ **Еволюційна нейро-фаззі система з online навчанням-самонавчанням** _____

затверджена наказом університету від _____ 20 21 р. № _____

2. Термін подання студентом роботи до екзаменаційної комісії **9** грудня 20 21 р.

3. Вихідні дані до роботи **Науково-технічні публікації, дані Інтернет та відомих наукових проєктів, електронні документації** _____

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Огляд стану проблеми і галузі дослідження

2) Нейро-нечіткі мережі в задачі прогнозування

3) Імітаційне моделювання і рішення задач на тестових вибірках

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Отримання завдання на кваліфікаційну роботу	01.09.2021	виконано
2.	Аналіз завдання та пошук літератури за темою	02.09.2021–08.09.2021	виконано
3.	Опрацювання літератури та аналіз об'єкту	09.09.2021–15.09.2021	виконано
4.	Вибір програмних засобів для розробки системи	16.09.2021–18.09.2021	виконано
5.	Розробка програмного засобу	19.09.2021–05.11.2021	виконано
6.	Аналіз отриманих результатів	06.11.2021–16.11.2021	виконано
7.	Оформлювання пояснювальної записки	17.11.2021–24.11.2021	виконано
8.	Проходження нормоконтролю	25.11.2021–28.11.2021	виконано
9.	Оформлення презентаційних матеріалів	29.11.2021–01.12.2021	виконано
10.	Попередній захист	02.12.2021	виконано
11.	Представлення кваліфікаційної роботи	09.12.2021	

Дата видачі завдання 1 вересня 2021 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис) _____
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 63 с., 16 рис., 1 табл., 2 дод, 40 джерел.

ГЕНЕТИЧНИЙ АЛГОРИТМ, ЕВОЛЮЦІЙНЕ НАВЧАННЯ,
НЕЧІТКА СИТЕМА, ПРОГНОЗУВАННЯ ЧАСОВИХ
ПОСЛІДОВНОСТЕЙ, ФАЗЗИ НЕЙРОМЕРЕЖА, ANFIS.

Об'єкт дослідження – процес прогнозування даних, що надходять в online режимі за допомогою нечіткої нейромережі.

Предмет дослідження – використання нечіткої логіки у нейромережах та методи навчання нейромереж, засновані на еволюційних алгоритмах.

Метою даного дослідження є розробка нечіткої нейромережі, що предбачає хаотичну часову послідовність, яка навчається за допомогою еволюційних алгоритмів.

Методи дослідження – теорія штучних нейронних мереж; теорія нечіткої логіки; генетичні алгоритми; імітаційне моделювання.

Припускається, що задача розробки нейронної мережі, заснованої на базі правил нечіткого виводу, призначеної для передбачення часових послідовностей в online режимі, коли дані надходять на обробку послідовно, одне за одним, є актуальною і такий підхід може бути використано для вирішення широкого класу задач прогнозування часових процесів.

РЕФЕРАТ

Пояснительная записка: 63 с., 16 рис., 1 табл., 2 прил., 40 источников.

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ, НЕЧЕТКАЯ СИСТЕМА,
ПРОГНОЗИРОВАНИЕ ВРЕМЕННЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ,
ФАЗЗИ-НЕЙРОСЕТЬ, ЭВОЛЮЦИОННОЕ ОБУЧЕНИЕ, ANFIS.

Объект исследования – процесс прогнозирования данных, которые поступают в online режиме при помощи нечеткой нейросети.

Предмет исследования – использование нечеткой логики в нейросетях и методы обучения нейросетей, основанные на эволюционных алгоритмах.

Целью данного исследования является разработка нечеткой нейросети, предсказывающую хаотическую временную последовательность, которая обучается с помощью эволюционных алгоритмов.

Методы исследования – теория искусственных нейронных сетей, теория нечеткой логики, генетические алгоритмы, имитационное моделирование.

Допускается, что задача разработки нейронной сети, основанной на использовании правил нечеткого вывода, предназначенной для предсказания временных последовательностей в online режиме, когда данные поступают на обработку последовательно, одно за другим, является актуальной и такой подход может быть использован для решения широкого класса задач по прогнозированию временных процессов.

ABSTRACT

Explanatory note: 63 p., 16 fig., 1 tabl., 2 ann., 40 sources

ANFIS, EVOLUTIONARY LEARNING, FUZZY NEURAL NETWORK, FUZZY SYSTEM, GENETIC ALGORITHM, TIME SEQUENCE PREDICTION.

The object of the study is the process of predicting data that comes online using a fuzzy neural network.

Subject of research - the use of fuzzy logic in neural networks and methods of training of neural networks based on evolutionary algorithms.

The purpose of this study is to develop a fuzzy neural network that predicts a chaotic temporal sequence, which is trained using evolutionary algorithms.

Research methods are artificial neural network theory, fuzzy logic theory, genetic algorithms, simulation modeling.

It is assumed that the task of developing a neural network based on the use of fuzzy inference rules, designed to predict temporal sequences in the online mode, when the data arrive for processing sequentially, one after another, is relevant and this approach can be used to solve a wide class of problems in predicting temporal processes.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ.....	9
1 Огляд стану проблеми і галузі дослідження	11
1.1 Фаззі системи.....	11
1.2 Фаззіфікація та дефаззіфікація	18
1.3 Основні типи нечітких систем.....	19
1.3.1 Нечітка модель Мамдані-Ассіліана	20
1.3.2 Нечітка система Такагі-Сугено-Канга.....	22
1.3.3 Нечітка система Цукамото	25
1.4 Нейронні фаззі системи.....	27
1.5 Еволюційні алгоритми навчання нейронних мереж	28
2 Нейро-нечіткі мережі в задачі прогнозування	31
2.1 Архітектура гібридної нейронечіткої системи	31
2.2 Навчання нечіткої нейромережі з використанням генетичного алгоритму	37
2.3 Хаотичний часовий процес	40
3 Імітаційне моделювання і рішення задач на тестових вибірках	42
3.1 Імітаційне моделювання нечіткої нейронної мережі	43
3.2 Аналіз отриманих результатів	45
Висновки	49
Перелік джерел посилання	50
Додаток А Вихідний код програми	54
Додаток Б Відомість кваліфікаційної роботи.....	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ГА – Генетичний алгоритм;

ІНЛ – Інструментарій нечіткої логіки;

ФН – Функція належності;

ANFIS – Adaptive Neuro Fuzzy Inference System – Адаптивна нейронна система нечіткого висновку;

ANNBFIS – Artificial Neural Network Based Fuzzy Inference System – Система нечіткого висновку на основі штучної нейронної мережі;

COG – Center Of Gravity – Метод центру тяжіння;

EA – Evolutionary Algorithm – Еволюційний алгоритм;

EANN – Evolutionary Artificial Neural Network – Еволюційна штучна нейронна мережа;

FLS – Fuzzy Logic System – Система нечіткої логіки;

GA – Genetic Algorithm – Генетичний алгоритм;

GFNN – Generic Fuzzy Neural Network – Узагальнена нечітка нейронна мережа;

MAE – Mean Absolute Error – Середня абсолютна помилка;

MAFS – Mamdani-Assilan Fuzzy System – Нечітка система Мамдані-Ассілана;

NFS – Neuro Fuzzy System – Нейро-фаззі система;

NN – Neural Network – Нейронна мережа;

TFS – Tsukamoto Fuzzy System – Нечітка система Цукамото;

TSKFS – Takagi-Sugeno-Kang Fuzzy System – Нечітка система Такагі-Сугено-Канг.

ВСТУП

Глибоке навчання нейронних мереж ґрунтується на навчанні сприйняття, а не на спеціалізованих алгоритмах, розроблених для конкретних завдань. Багато методів глибокого навчання було відомо ще в 1980-х роках, але результати були невражаючими, в той час як успіхи були в теорії штучних нейронних мереж (попереднє навчання нейронних мереж з використанням особливого випадку неорієнтованої графової моделі – моделі «граф») неорієнтованої графової моделі, так званої обмеженої машини Больцмана.

Перш за все, графічні процесори Nvidia і тепер тензорні процесори Google не дозволяли створювати складні технологічні архітектури нейронних мереж з достатньою продуктивністю і дозволяють вирішувати широке коло завдань, не є ефективним рішенням раніше, наприклад, в комп'ютерному зорі, машинному перекладі, розпізнаванні мови, при цьому якісні рішення, у багатьох випадках, зараз можна порівняти, а в деяких випадках перевершують «білкових» експертів. На відміну від машинного навчання, глибоке навчання вимагає набагато більшого обсягу навчальної вибірки. Також, на відміну від машинного навчання, глибока нейронна мережа може мати тисячі шарів.

Все це допомагає глибоким нейронним мережам досягати досить високої точності у завданнях з аналізу, класифікації та розпізнавання образів. Проте, основним недоліком глибокого навчання є величезна ресурсомісткість.

Щоб навчити глибоку нейронну мережу, іноді необхідно зробити навчальну вибірку з мільйона зображень, і навіть більше, а процес навчання може зайняти кілька днів. Для таких завдань навіть розроблено окремі процесори GPU для прискорення навчання. На сьогоднішній день існує низка бібліотек для глибоких нейронних мереж. Найбільш популярними є TensorFlow і Keras, які можуть бути використані в задачах з прогнозування.

У роботі розглядається завдання побудови гібридної системи прогнозування часових рядів з урахуванням глибоких нейронних мереж і нечіткої логіки. Цей підхід дозволяє враховувати як кількісні, так і якісні характеристики часових рядів. Для повноти викладу особливості нечітких когнітивних карт та їх застосування в задачах прогнозування часових рядів прогнозування. Також наводиться розроблений генетичний алгоритм для навчання нечітких когнітивних карт, який дозволяє уникнути трудомісткого завдання ручного налаштування когнітивної карти. Для вирішення проблеми роботи з напівструктурованими даними, які часто мають місце у завданнях прогнозування часових рядів, пропонується використовувати глибокі нейронні мережі архітектури, оскільки такі мережі здатні працювати з даним типом даних і показують найбільш достовірні результати.

1 ОГЛЯД СТАНУ ПРОБЛЕМИ І ГАЛУЗІ ДОСЛІДЖЕННЯ

1.1 Фаззі системи

Нечіткі системи – це структури, які засновані на нечітких методах, орієнтовані на обробку інформації, де використання класичної теорії множин та двійкової логіки неможливе чи важко. У літературі такі терміни, як нечітка система, нечітка модель, система на основі нечітких правил, нечіткий контролер або нечітка асоціативна пам'ять, використовуються взаємозаміно в залежності від типу програми [1]. Їхня основна характеристика включає символічне уявлення знань у вигляді нечітких умовних правил (якщо-то).

Типова структура нечіткої системи (рисунок 1.1) складається з чотирьох функціональних блоків: фаззіфікатора, механізму нечіткого виведення, бази знань та дефаззіфікатора.

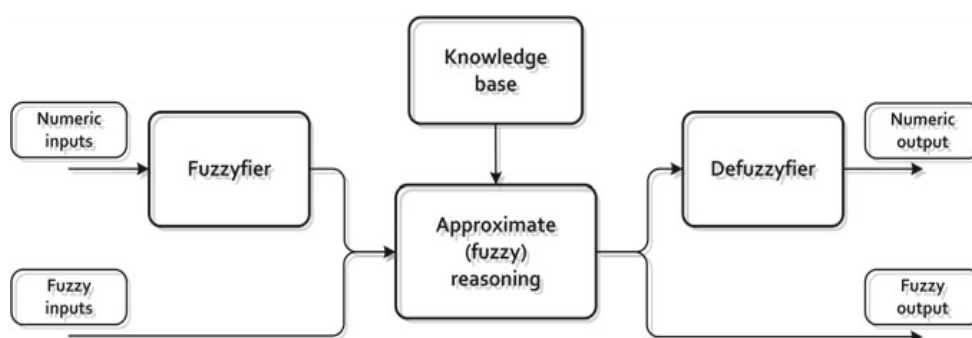


Рисунок 1.1 – Типова структура нечіткої системи

В якості вхідних даних для нечіткої системи можуть використовуватися як лінгвістичні значення (визначаються нечіткими множинами), так і чіткі (числові) дані. Якщо використовуються чіткі дані, процес виведення передусє фаззіфікація, яка присвоює нечіткому входу відповідну нечітку множину. Значення вхідних змінних відображаються у лінгвістичні значення вихідної змінної за допомогою відповідного методу

наближеного міркування (inference engine) з використанням експертних знань, які представлені у вигляді набору нечітких умовних правил (бази знань). Крім лінгвістичних значень, як вихідні дані нечіткої системи можуть знадобитися і числові дані.

Практичне застосування нечітких систем включає проблеми, для яких недоступний повний математичний опис, або де використання точної (не нечіткої) моделі є нерентабельним або вкрай незручним. Здатність обробляти нечітку інформацію робить нечітку систему відмінним інструментом, наприклад для процесів управління [2], [3], ідентифікації систем [4], [5], підтримки прийняття рішень [6], [7], та обробки сигналів і зображень [8], [9].

Одним із фундаментальних понять теорії нечітких множин є лінгвістична змінна [10]. Її значеннями є висловлювання природної мови (терміни), які є мітками (описами) нечітких множин, визначених на заданій множині. Формально лінгвістична змінна визначається як квінтел [11]:

$$X = (N, T, \mathbb{X}, G, M), \quad (1.1)$$

де X – лінгвістична змінна;

N – найменування лінгвістичної змінної;

T – множина значень (терм-множина), що є найменуванням нечітких змінних, областю визначення кожної з яких є множина \mathbb{X} ;

G – синтаксична процедура, що дозволяє оперувати елементами терм-множини T , зокрема, генерувати нові терми (значення);

M – семантична процедура, що дозволяє перетворити кожне нове значення лінгвістичної змінної, що утворене процедурою G , на нечітку змінну, тобто сформувати відповідну нечітку множину.

Наприклад, такі слова, як молодий, високий, хороший або високий, є нечіткими. Немає єдиного кількісного значення, що визначає термін

«молодий». Для одних людей 25 років – це молодість, а для інших – 35 років. Поняття «молодий» не має чітких меж. Вік 35 років має певну нагоду бути молодим і зазвичай залежить від контексту, в якому він розглядається.

Тоді лінгвістичну змінну, що описує вік людини, можна позначити так:

- N – «вік»;
- \mathbb{X} – множина цілих чисел з інтервалу $[0, 120]$;
- T – множина нечітких змінних, для кожного значення: «молодий», «зрілий», «старий», необхідно задати функцію належності, яка задає інформацію про те, людей якого віку вважати молодими, зрілими, старими;
- G – «дуже», «не дуже». Такі добавки дозволяють утворювати нові значення: «дуже молодий», «не дуже старий» та ін.;
- M – математичне правило, що визначає вид функції належності кожному за значення утвореного за допомогою правила G. Логічне значення виразу визначається на основі функції приналежності $\mu_A(x)$ нечіткої множини A. У попередньому прикладі елементарне твердження має вигляд, як показано на рисунку 1.2.

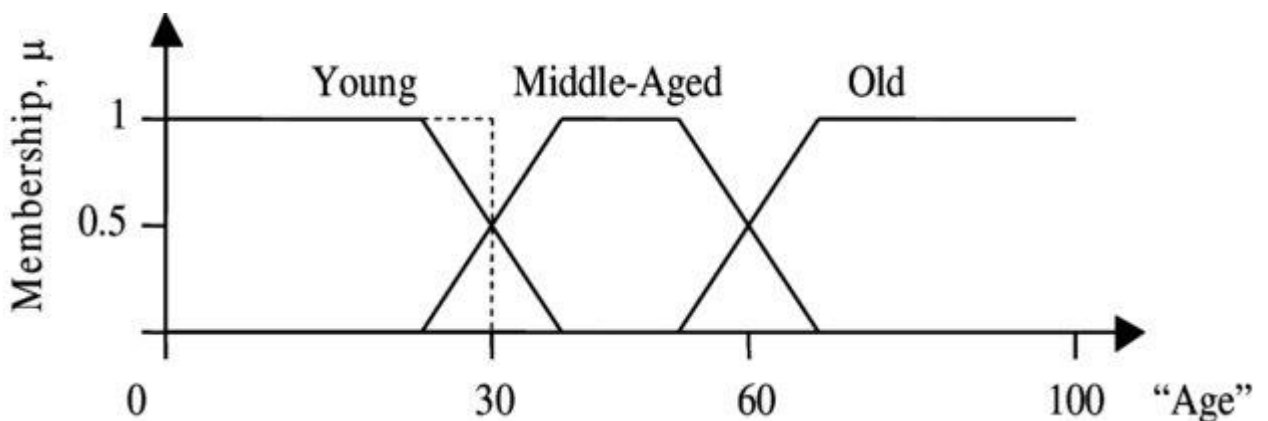


Рисунок 1.2 – Приклад функцій членства нечітких множин для лінгвістичних змінних

Більш складний нечіткий вираз може бути отриманий шляхом поєднання двох або більше елементарних виразів.

Такий вираз може мати як кон'юнктивну форму:

$$(X_1 \in L_{A_1}) \text{ та } (X_2 \in L_{A_2}), \quad (1.2)$$

так і диз'юнктивну форму:

$$(X_1 \in L_{A_1}) \text{ або } (X_2 \in L_{A_2}), \quad (1.3)$$

де X_1 , X_2 – лінгвістичні змінні з мітками L_{A_1} , L_{A_2} , що визначаються нечіткими множинами A_1 та A_2 відповідно на множинах \mathbb{X}_1 та \mathbb{X}_2

Значення складного нечіткого виразу для $x_1 \in \mathbb{X}_1$ та $x_2 \in \mathbb{X}_2$ визначається на основі функцій належності нечітких множин A_1 та A_2 [1] операцій кон'юнкції та диз'юнкції, відповідно:

$$T(x_1, x_2) = \mu_{A_1}(x_1) \wedge \mu_{A_2}(x_2), \quad (1.4)$$

$$S(x_1, x_2) = \mu_{A_1}(x_1) \vee \mu_{A_2}(x_2), \quad (1.5)$$

де – t- і s-норми відповідно $T(x_1, x_2)$, $S(x_1, x_2)$

Елементарне нечітке висловлювання також може бути виражене у формі імплікації, що утворює нечітке правило «якщо-то» (нечітке умовне висловлювання):

$$\text{якщо } (X \in L_A), \text{ то } (Y \in L_B) \quad (1.6)$$

Цей вираз визначає відношення між лінгвістичними змінними. Твердження $(X \in L_A)$ називається антецедентом (передумовою), а

твердження $(Y \in L_B)$ – консеквентом (висновком). Узагальнена форма нечіткого умовного висловлювання може бути визначена як імплікація складних нечітких виразів. Для кон'юнктивної форми вона може бути записана як:

$$\begin{aligned} \text{якщо } (X_1 \in L_{A_1}) \text{ та } (X_2 \in L_{A_2}) \text{ та } \dots \text{ та } (X_N \in L_{A_N}), \text{ то} \\ (Y_1 \in L_{B_1}), (Y_2 \in L_{B_2}), \dots, (Y_M \in L_{B_M}), \end{aligned} \quad (1.7)$$

для диз'юнктивної форми як:

$$\begin{aligned} \text{якщо } (X_1 \in L_{A_1}) \text{ або } (X_2 \in L_{A_2}) \text{ або } \dots \text{ або } (X_N \in L_{A_N}) \text{ то} \\ \text{то } (Y_1 \in L_{B_1}), (Y_2 \in L_{B_2}), \dots, (Y_M \in L_{B_M}), \end{aligned} \quad (1.8)$$

де X_1, X_2, \dots, X_N – вхідні лінгвістичні змінні;

Y_1, Y_2, \dots, Y_M – вихідні лінгвістичні змінні;

$L_{A_1}, L_{A_2}, \dots, L_{A_N}$ та $L_{B_1}, L_{B_2}, \dots, L_{B_M}$ – їх лінгвістичні значення, що визначаються нечіткими множинами A_1, A_2, \dots, A_N та B_1, B_2, \dots, B_M на множинах X_1, X_2, \dots, X_N та Y_1, Y_2, \dots, Y_M відповідно.

Канонічні нечіткі умовні твердження є основою представлення експертних знань у нечіткій системі. Використовуючи псевдовекторну нотацію, канонічне нечітке правило «якщо – то» можна записати як:

$$\text{якщо } (X \in L_A), \text{ то } (Y \in L_B), \quad (1.9)$$

що є $N + 1$ -нарним нечітким відношенням [8]:

$$R = ((A_1 \times A_2 \times \dots \times A_N) \Rightarrow B) = (A \Rightarrow B), \quad (1.10)$$

яке визначене на $X_1 \times X_2 \times \dots \times X_N \times Y$, з функцією належності:

$$\mu_R(x_1, x_2, \dots, x_N, y) = \Phi(\mu_A(x), \mu_B(y)), \quad (1.11)$$

де $x = [x_1, x_2, \dots, x_N] \in X_1 \times X_2 \times \dots \times X_N$, $y \in Y$ і в залежності від інтерпретації нечіткого правила «якщо-то»;

Φ – означає t-норму (кон'юнктивну інтерпретацію) [1], [12] або нечітку імплікацію (логічну інтерпретацію) [1], [12], [13].

Якщо зв'язка «та» у попередниках нечітких правил «якщо-то», представлена t-нормою, то:

$$\mu_A(x) = \mu_{A_1}(x_1) \wedge \mu_{A_2}(x_2) \wedge \dots \wedge \mu_{A_N}(x_N), \quad (1.12)$$

де A_1, A_2, \dots, A_N – нечіткі множини, що представляють значення лінгвістичних змінних в антецеденті канонічного нечіткого правила.

Отже, для кон'юнктивної інтерпретації ми маємо:

$$\begin{aligned} \mu_R(x, y) &= \mu_R(x_1, \dots, x_N, y) = \mu_A(x) \wedge \mu_B(y) = \\ &= \mu_{A_1}(x_1) \wedge \mu_{A_2}(x_2) \wedge \dots \wedge \mu_{A_N}(x_N) \wedge \mu_B(y) \end{aligned} \quad (1.13)$$

тоді як для логічної інтерпретації:

$$\begin{aligned} \mu_R(x, y) &= \mu_R(x_1, \dots, x_N, y) = \Psi(\mu_A(x), \mu_B(y)) = \\ &= \Psi(\mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_N}(x_N), \mu_B(y)), \end{aligned} \quad (1.14)$$

де Ψ – означає нечітку імплікацію.

Нечітка імплікація зазвичай вводиться за допомогою аксіоматичного підходу [13], де вона визначається як безперервна функція $\Psi: [0,1] \times [0,1] \rightarrow [0,1]$, яка для кожного $a, b, c \in [0,1]$ задовольняє восьми рекомендованим (специфічним) умовам та п'ятьом необхідним (загальним) умовам [8]:

– P1: якщо $a \leq c$, то $\Psi(a, b) \geq \Psi(c, b)$;

- P2: якщо $b \leq c$, то $\Psi(a, b) \leq \Psi(a, c)$;
- P3: $\Psi(0, b) = 1$;
- P4: $\Psi(a, 1) = 1$;
- P5: $\Psi(1, 0) = 0$.

Властивості P3, P4 і P5 називаються відповідно помилковістю, нейтральністю та булевістю [8], [14].

Одне нечітке правило визначає локальний зв'язок між вхідними та вихідними змінними нечіткої системи в межах, що визначаються областю нечітких множин в антецеденті правила. Повне відображення вхід-вихід представлено всією сукупністю нечітких правил «якщо» з бази знань (правил). Для подальших міркувань ми припускаємо, що база складається із I правил у формі:

$$\mathcal{R} = \{R^{(i)}\}_{i=1}^I = \left\{ \text{если и } \left(X_n \text{ является } L_{An}^{(i)} \right), \text{ тогда } Y \text{ является } L_B^{(i)} \right\}_{i=1}^I. \quad (1.15)$$

Добре визначена нечітка база правил має бути повною, послідовною та безперервною [15]. Повнота означає, що для кожного значення з вхідного простору активується хоча б одне правило, тобто $\exists_{i=1,2,\dots,I} \mu_{A^{(i)}}(x) \neq 0$. База знань є несуперечливою, якщо немає правил з однаковим антецедентом, але різними наслідками. І нарешті, база знань безперервна, якщо немає сусідніх правил, для яких результатом перетину нечітких множин в їхніх наслідках є порожня множина.

База знань будується спочатку шляхом здобуття знань про модальне явище, а потім шляхом подання їх у вигляді нечітких умовних правил. На практиці існує три основні методи створення бази нечітких правил [1]:

- використовуючи знання людини-експерта або ґрунтуючись на фізичних законах, що описують явище (моделювання «білої скриньки»);
- шляхом автоматичного вилучення правил на основі числових

даних, що представляють взаємозв'язок між входами та виходами явища (моделювання «чорної скриньки»);

– змішаний, коли частина знань отримана від людини-експерта, а частина – у результаті автоматичного вилучення (моделювання «сірої скриньки»).

Однак можливі застосування нечіткої системи залежать не тільки від правильно визначеної бази знань, а й від конструкції механізму виведення.

1.2 Фаззифікація та дефаззифікація

У багатьох додатках входи нечітких систем визначаються як чіткі числові дані. Однак наближені міркування вимагають, щоб входи були представлені у вигляді нечітких множин. Процес відображення реальних значень на N -мірну нечітку множину A , визначене на X , називається фаззифікацією. Фаззифікація може бути символічно виражена як перетворення N -вимірного простору на множину нечітких множин $x_0 = [x_{01}, x_{02}, \dots, x_{0N}] \in \underline{X} \in \mathbb{R}^N \underline{X}$ [1]

$$\underline{X} \Rightarrow \mathcal{F}(\underline{X}). \quad (1.16)$$

Використовуючи функції належності, ми можемо написати:

$$\underline{X} \Rightarrow \{\mu_{A'}(x) | x \in \underline{X}, \mu_{A'}(x) \in [0,1]\}. \quad (1.17)$$

Результатом наближених міркувань є нечітка множина $B'(y)$, яка може бути пов'язана з конкретною лінгвістичною міткою. Однак існують додатки, які потребують чіткого числового результату виведення. Процес обчислення представницького числового результату $y_0 \in Y$ з нечіткої множини $B'(y)$ на Y називається дефаззифікацією. Дефаззифікація – це перетворення множини нечітких множин, визначених на просторі, в одне

числове значення [1]:

$$\mathcal{F}(\mathbb{Y}) \rightarrow \mathbb{Y}. \quad (1.18)$$

Використовуючи функції належності, отримуємо:

$$\{\mu_{B'}(y) | y \in \mathbb{Y}, \mu_{B'}(y) \in [0,1]\} \rightarrow \mathbb{Y}. \quad (1.19)$$

Через різні критерії для визначення елемента y_0 з нечіткої множини $B'(y)$ який слід розглядати як найбільш представницький, існує безліч визначень процедури дефаззифікації [15], [16]. Одним з найбільш популярних є метод центру тяжіння (COG), який визначає результат як центр області під функцією $\mu_{B'}(y)$:

$$y_0 = \frac{\int_{\mathbb{Y}} y \mu_{B'}(y) dy}{\int_{\mathbb{Y}} \mu_{B'}(y) dy} \quad (1.20)$$

1.3 Основні типи нечітких систем

У зв'язку з широким спектром можливих застосувань існує безліч різних типів нечітких систем, які були запропоновані в літературі [1], [8], [14], [15], [17]. Однак нові рішення, що характеризуються зниженням складності обчислень, покращенням якості моделювання або більшою простотою лінгвістичної інтерпретації результатів виведення, все ще є предметом досліджень. Модель, запропонована Е.Х. Мамдані та С. Ассіланом [3], зазвичай вважається першою нечіткою системою, що є у літературі. В даний час її можна вважати основою сімейства нечітких моделей, заснованих на правилах «якщо-то» з нечіткими множинами як в антецедентах, так і в послідовності.

1.3.1 Нечітка модель Мамдані-Ассіліана

Нечітка система Мамдані-Ассіліана (MAFS) використовує набір умовних нечітких правил у канонічній формі (формула 2.16), які можуть бути визначені людиною-експертом. MAFS заснована на кон'юнктивній інтерпретації нечітких правил, де кон'юнкція «і» антецеденту правила визначається за допомогою t-норми мінімуму (\wedge). Результати висновків окремих правил агрегуються із застосуванням s-норми максимуму (\vee). Числові входи $x_0 = [x_{01}, x_{02}, \dots, x_{0N}]$ відображаються в нечіткі множини за допомогою сінгтонного фаззифікатора, який може бути визначений як:

$$\mu_{A'}(x) = \delta_{x,x_0} = \begin{cases} 1, & x = x_0, \\ 0, & x \neq x_0, \end{cases} \quad (1.21)$$

а числовий результат обчислюється методом COG. Схема наближених міркувань реалізується на основі рівняння, яке має вигляд:

$$\begin{aligned} \mu_{B'}(y) &= \vee_{i=1}^I [\mu_{A^{(i)}}(x_0) \wedge \mu_{B^{(i)}}(y)] = \\ &= \vee_{i=1}^I [\mu_{A^{(i)}}(x_0) \wedge \mu_{B^{(i)}}(y)] = \vee_{i=1}^I [F^{(i)}(x_0) \wedge \mu_{B^{(i)}}(y)], \end{aligned} \quad (1.22)$$

$$F^{(i)} = \mu_{A^{(i)}}(x_0) = \mu_{A_1^{(i)}}(x_{01}) \wedge \mu_{A_2^{(i)}}(x_{02}) \wedge \dots \wedge \mu_{A_N^{(i)}}(x_{0N}). \quad (1.23)$$

Наведене вище рівняння (формула 1.23) визначає так звану силу виконання і правила, що позначається як $F^{(i)}(x_0)$.

Використовуючи фаззифікацію за методом COG отримаємо:

$$y_0 = \frac{\int_Y y \mu_{B'}(y) dy}{\int_Y \mu_{B'}(y) dy}. \quad (1.27)$$

На рисунку 1.3 показано приклад нечіткого виведення з

використанням системи Мамдані-Ассіліана з двома входами та базою знань, що складається з двох умовних нечітких правил.

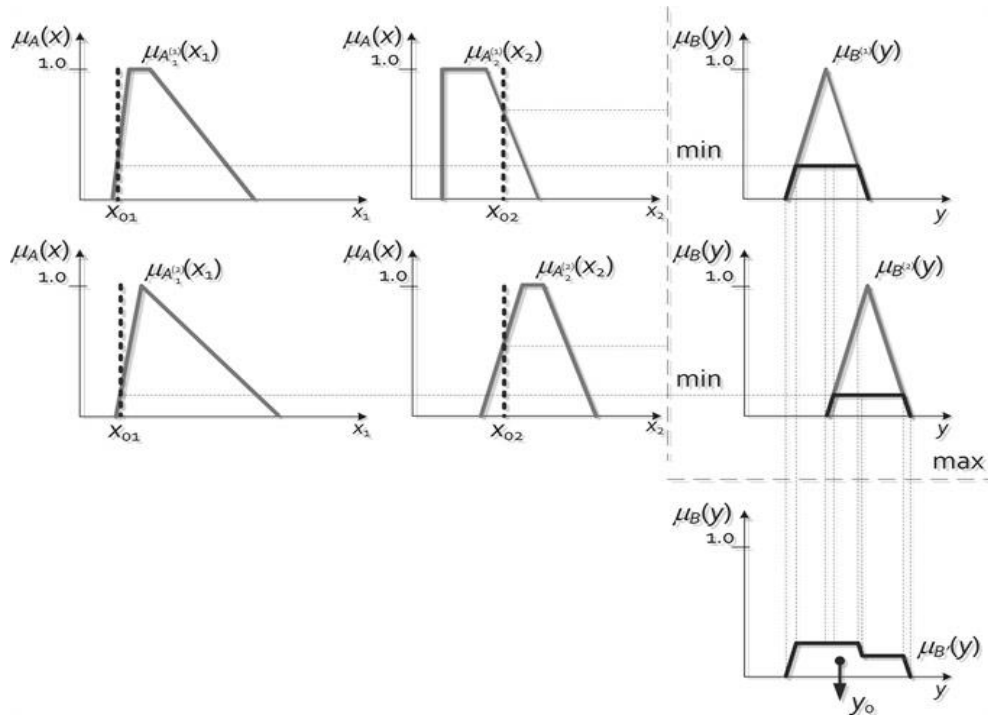


Рисунок 1.3 – Приклад нечіткого виведення з використанням нечіткої системи Мамдані-Ассіліана з двома входами та базою знань, що складається з двох умовних нечітких правил

Дефаззифікація потребує високої обчислювальної складності, проте можна застосувати деякі спрощення. Використовуючи алгебраїчний добуток t-норми та середнє арифметичне як оператор агрегування, ми отримуємо нечітку систему Ларсена, яка визначається як [1]:

$$\mu_{B'}(y) = \frac{1}{I} \sum_{i=1}^I F^{(i)}(x_0) \mu_{B^{(i)}}(y). \quad (1.24)$$

Підставивши (2.25) у (2.21), отримаємо:

$$y_0 = \frac{\sum_{i=1}^I F^{(i)}(x_0) \int_{\mathbb{Y}} y \mu_{B'}(y) dy}{\sum_{j=1}^I F^{(j)}(x_0) \int_{\mathbb{Y}} \mu_{B'}(y) dy}. \quad (1.25)$$

Позначимо область під функцією належності нечіткої множини $B'(y)$ як:

$$\mathcal{A}(\mu_{B'}(y)) = \int_{\mathbb{Y}} \mu_{B'}(y) dy, \quad (1.26)$$

та його центр тяжіння як $y^{(i)}$, ми можемо записати:

$$y_0 = \frac{\sum_{i=1}^I F^{(i)}(x_0) \mathcal{A}(\mu_{B^{(i)}}(y)) y^{(i)}}{\sum_{j=1}^I F^{(j)}(x_0) \mathcal{A}(\mu_{B^{(j)}}(y))}. \quad (1.27)$$

Наведене вище рішення вимагає лише одного обчислення площ під функціями належності та розташування центрів ваги для всіх нечітких правил. Припускаючи додатково, що $\mathcal{A}(\mu_{B^{(i)}}(y))$ однакові для всіх I наслідків, які ми отримуємо від нечіткої моделі Сугено-Ясукави [18].

Наближені міркування без необхідності дефаззифікації були представлені в роботах Такагі та Сугено [19] та Сугено та Канга [20]. Запропонована модель, названа нечіткою системою Такагі-Сугено-Канга (TSKFS), описана в наступному підрозділі.

1.3.2 Нечітка система Такагі-Сугено-Канга

База знань TSKFS складається з умовних нечітких правил із послідовностями у вигляді класичних функцій, аргументами яких є вхідні числові дані:

$$\begin{aligned} \mathcal{R} &= \{R^{(i)}\}_{i=1}^I = \\ &= \left\{ \text{якщо } \bigwedge_{n=1}^N (x_{0n} \in L_{An}^{(i)}), \text{ тоді } y = y^{(i)}(x_0) \right\}_{i=1}^I, \end{aligned} \quad (1.28)$$

де x_{0n} – вхідний синглтон, $x_0 = [x_{01}, x_{02}, \dots, x_{0N}]$;

$y^{(i)}(x)$ – функція в i -му консеквенті.

Виходом кожного нечіткого правила є чітке числове значення $y^{(i)}(x_0)$, а результат TSKFS розраховується як середньозважене значення окремих виходів:

$$y_0 = \frac{\sum_{i=1}^I F^{(i)}(x_0) y^{(i)}(x_0)}{\sum_{j=1}^I F^{(j)}(x_0)}, \quad (1.29)$$

де $F^{(i)}(x_0)$ – це сила спрацьовування правила.

Рівняння (1.29) можна інтерпретувати як сукупність експертів, кожен із яких моделюється одним нечітким правилом. Кожне правило визначає взаємозв'язок між виходами та входами системи у відповідному діапазоні входів. Середньозважене значення тверджень всіх локальних експертів (правил) визначає результат міркування. Вага, подана силою спрацьовування правила, визначає ступінь впливу окремого експерта на кінцевий результат виведення.

Консеквент i -го нечіткого правила TSKFS також можна розуміти як синглтон [8], місцезнаходження якого визначається функцією $y^{(i)}(x)$:

$$\mu_{B^{(i)}}(y) = \delta_{y, y^{(i)}} = \begin{cases} 1, & y = y^{(i)}(x_0), \\ 0, & y \neq y^{(i)}(x_0), \end{cases} \quad (1.30)$$

Отже, TSKFS зазвичай називають нечіткою системою з «рухомими» синглтонами. Термін «переміщення» відноситься до взаємозв'язку між розташуванням синглтона та вхідними числовими даними. Амплітуда (висота) синглтона після наближеного міркування визначається силою спрацьовування правила.

Консеквенти TSKFS часто визначаються як лінійні функції (поліноми першого порядку):

$$y^{(i)}(x_0) = p_0^{(i)} + p_1^{(i)}x_{01} + p_2^{(i)}x_{02} + \dots + p_N^{(i)}x_{0N} = p^{(i)}x'_0 \quad (1.31)$$

де $p^{(i)}$ – (N+1)-мірний вектор параметрів функції $y^{(i)}(x)$,

x'_0 – розширений вхідний вектор:

$$x'_0 = [1 \ x_0] \quad (1.32)$$

Набір простих лінійних функцій $y^{(i)}(x)$ дозволяє моделювати найскладніші стосунки «вхід-вихід». Области попередників у сусідніх правилах забезпечують плавне перемикавання між локальними моделями.

Приклад виведення TSKFS з двома входами та двома умовними нечіткими правилами показано на рис. 1.4. Основною перевагою TSKFS є низьке обчислювальне навантаження, необхідне визначення числового виходу системи, оскільки процес виведення не включає дефазифікацію. Проте вона дозволяє застосовувати різні інтерпретації нечітких правил і різні типи агрегуючих операторів. Це пов'язано із застосуванням сінглетонів у наслідках правил. Система нечітких висновків на основі штучних нейронних мереж (ANNBFIS) [23] позбавлена цих недоліків. ANNBFIS поєднує в собі переваги використання нечіткої множини внаслідок правила (як у MAFS) і залежність розташування слідства від входів системи (як у TSKFS) [1], [8], [22].

Інше розширення TSKFS – це нечітка система Цукамото (TFS) [23]. Основна відмінність між TSKFS і TFS полягає в методі позиціонування сінглтону в післядії нечіткого правила. У TFS воно визначається з допомогою монотонної функції, і навіть сили спрацювання правила.

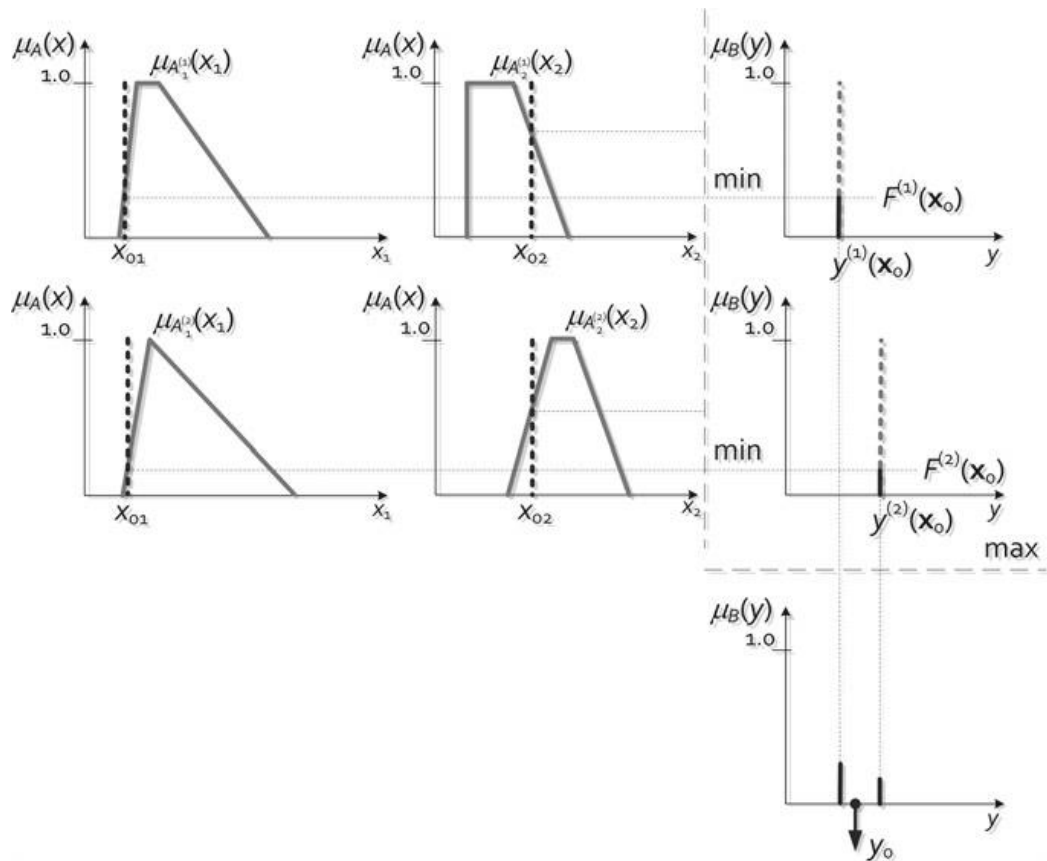


Рисунок 1.4 – Приклад наближеного міркування з нечіткою системою Такагі-Сугено-Канга з двома входами та двома нечіткими правилами «ЯКЩО-ТО»

1.3.3 Нечітка система Цукамото

База знань TFS є набір нечітких умовних тверджень у формі:

$$\mathcal{R}^{(i)} = \text{якщо } \bigwedge_{n=1}^N (X_{0n} \in L_{A_n}^{(i)}), \text{ тоді } y = f_i^{-1}(F^{(i)}(\mathbf{x}_0)), \quad (1.33)$$

де $f_i(y)$ – монотонна функція в i -му консеквенті.

Для сили спрацьовування правила рівною $F^{(i)}(\mathbf{x}_0)$ консеквент є сінглетоном з амплітудою $F^{(i)}(\mathbf{x}_0)$ та місцезнаходженням, таким, що:

$$F^{(i)}(x_0) = f_i^{-1}(y^{(i)}), \quad (1.34)$$

$$\mu_{B^{(i)}}(y) = F^{(i)}(x_0) \delta_{y, y^{(i)}} = \begin{cases} F^{(i)}(x_0), & y = y^{(i)}, \\ 0, & y \neq y^{(i)}. \end{cases} \quad (1.35)$$

Результат виведення TFS розраховується як середньозважене значення розташування сінгтонів з усіх правил, при цьому ваги визначаються як інтенсивність спрацьовування правил:

$$y_0 = \frac{\sum_{i=1}^I F^{(i)}(x_0) y^{(i)}}{\sum_{j=1}^I F^{(j)}(x_0)} = \frac{\sum_{i=1}^I F^{(i)}(x_0) f_i^{-1}(y^{(i)})}{\sum_{j=1}^I F^{(j)}(x_0)}. \quad (1.36)$$

Приклад наближеного міркування Цукamoto з двома входами та двома нечіткими правилами «якщо – то» наведено на рисунку 1.5.

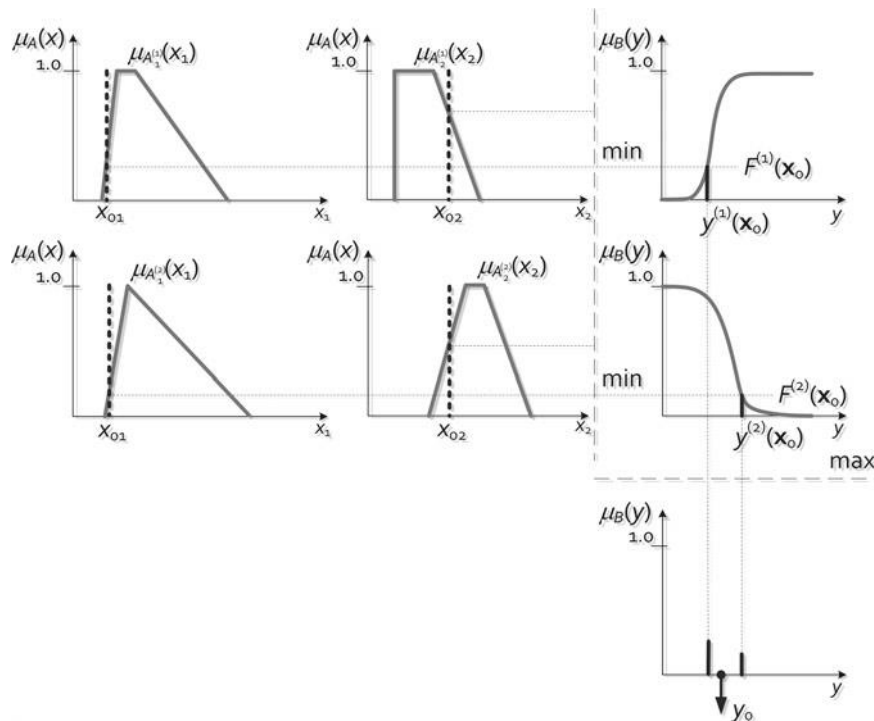


Рисунок 1.5 – Приклад наближеного міркування Цукamoto з двома входами та двома нечіткими правилами «якщо-то».

TFS рідко використовується через складність отримання умовних нечітких правил від людини-експерта у формі (1.33).

TFS є інший підхід до нечіткого моделювання, який засновано не на композиційному правилі виведення Заде, а на міркуваннях з використанням нечітких обмежень на істинні значення. У літературі описано багато інших цікавих пропозицій нечітких моделей, у тому числі заснованих на інтервально-значних нечітких множинах і нечітких множинах другого типу. Докладний огляд можна знайти, наприклад, у [24], [25], [26], [27].

1.4 Нейронні фаззі системи

Штучна нейронна мережа та метод нечіткої логіки є двома важливими методами для моделювання та управління нелінійною системою. З того часу, як було запропоновано модель МР, було зроблено багато проривів у нейронних мережах, особливо теоретичне досягнення що тришарова МР мережа може апроксимувати будь-яке безперервне нелінійне відображення [28]. У 1960-х роках Л. А. Заде запропонував теорію нечіткої логіки. До цього часу було отримано багато визначних досягнень, таких як універсальна апроксимація, теорема про те, що FLS може апроксимувати будь-яке безперервне нелінійне відображення. В останні роки нейронні мережі (NN) і FLS були більш тісно інтегровані та досягли багатьох значних моделей, таких як нейронна мережа нечітких висновків, тощо.

Для досягнення мети інтеграції NN та FLS пропонується загальна нечітка нейронна мережа (GFNN). Її основна ідея у тому, що практична фізична основа FLS відображається в тришарову мережу прямого розповсюдження. У поєднанні з фізичним тлом, проста структура нейрона змінюється на загальну структуру, і ця структура ближча до біологічного нейрона, тобто, вона має функції зберігання інформації та виведення функціями. Таким чином, нечіткі правила FLS відповідають прихованим нейронам GFNN, а механізм нечіткого виведення включається у функцію виведення і прихованих нейронів. Ключовим моментом є запровадження двох явних параметрів, що мають фізичний зміст; критерій ступеня

збігу (MDC) λ^* , який використовується для оцінки того, чи існуючий рівень відповідності між вхідним та прихованими нейронами є розумним, та мінімальний правдоподібний рівень (MBL) Θ^* , який відповідає порогу кожного прихованого нейрона. λ^* та Θ^* має явні фізичні значення та легко ініціалізується. Якщо ступені збігу між входом і всіма прихованими нейронами менше, ніж λ^* , то приховані нейрони вважаються неадекватними і мають бути створені нові. Одночасно, якщо ступінь збігу між входом і прихованим нейроном менший, ніж Θ^* , то нейрон не використовуватиметься для виведення.

Таким чином, відносна частота невикористання прихованого нейрона може бути використана для реалізації автоматичної експлуатації та мінімальний правдоподібний рівень (MBL) Θ^* , який відповідає порогу кожного прихованого нейрону λ^* та Θ^* мають явні фізичні значення та легко ініціалізуються. Таким чином, відносна частота невикористання прихованого нейрона може бути використана для реалізації автоматичної експлуатації.

1.5 Еволюційні алгоритми навчання нейронних мереж

Еволюційні штучні нейронні мережі (EANN) можна розглядати як комбінацію штучних нейронних мереж (ANN) та процедур еволюційного пошуку, таких як генетичні алгоритми (GA). Можна розглянути три рівні еволюції ANN, тобто еволюція терезів зв'язків, архітектури та правил навчання.

Еволюційні алгоритми (EA) [29], відносяться до набору біоінспірованих алгоритмів стохастичної оптимізації, які використовують еволюційні засади для побудови надійних адаптивних систем. Ця область бере свій початок у чотирьох основних еволюційних методах: Генетичні алгоритми [30], Еволюційні стратегії [31], Еволюційне програмування [32] та Генетичне програмування [33]. Ключовим елементом цих алгоритмів,

безсумнівно, є гнучкість, що дозволяє практикуючому спеціалісту використовувати елементи двох або більше методів ЕА. Саме тому межі між цими підходами перестали бути чіткими, що дозволило створити більш цілісну структуру ЕА [34]. Експерти працюють із популяцією μ -кодованих потенційних рішень конкретної проблеми. Кожне потенційне рішення, відоме як індивід, є точкою в просторі пошуку, де знаходиться оптимальне рішення. Популяція еволюціонує за допомогою генетичних операторів протягом кількох поколінь для отримання найкращих результатів вирішення проблеми. Кожен член популяції оцінюється за допомогою функції пристосованості, щоб визначити, наскільки добре чи погано потенційне рішення для цієї проблеми. Значення придатності, присвоєне кожній особини у популяції, ймовірно визначає, наскільки успішно ця особина поширюватиме (частину) свого коду в наступних поколіннях. Більш ефективним рішенням надаються більш високі значення (для проблем максимізації) або нижчі значення (для проблем мінімізації). Популяція еволюціонує за допомогою генетичних операторів протягом кількох поколінь для отримання найкращих результатів вирішення проблеми. Кожен член популяції оцінюється за допомогою функції пристосованості, щоб визначити, наскільки добре чи погано потенційне рішення для цієї проблеми.

Еволюційний процес здійснюється за допомогою генетичних операторів. Більшість експертів включають оператори, які відбирають особин для відтворення, генерують нові особини на основі відібраних особин та зрештою визначають склад особин у популяції у наступному поколінні. Відбір, кросинговер та мутація – ключові генетичні оператори, які використовуються в більшості парадигм ЕА. Оператор відбору відповідає за вибір однієї або кількох особин із популяції на основі їх значень придатності. Було запропоновано багато операторів відбору. Одним із найпопулярніших операторів відбору є турнірний відбір через його простоту. Ідея полягає в тому, щоб вибрати з популяції найкращу особину,

зазвичай, розміром = [2 – 7]. Стохастичний кросовер, також відомий як рекомбінація, оператор обмінюється матеріалом зазвичай від двох відібраних особин. Цей оператор відповідає за освоєння простору пошуку. Стохастичний оператор мутації вносить випадкові зміни до генів особин. Цей оператор відповідає за дослідження простору пошуку. Оператор мутації важливий для забезпечення різноманітності популяції, а також для відновлення генетичного матеріалу, втраченого в ході еволюції.

Еволюційний процес, описаний раніше, повторюється до того часу, доки буде виконана певна умова. Як правило, доти, доки не буде виконано максимальну кількість генерацій. Популяція в останньому поколінні є результатом вивчення та освоєння простору пошуку протягом кількох поколінь. Вона містить найкращі потенційні рішення проблеми, що еволюціонували і може також являти собою глобальне оптимальне рішення.

2 НЕЙРО-НЕЧІТКІ МЕРЕЖІ В ЗАДАЧІ ПРОГНОЗУВАННЯ

2.1 Архітектура гібридної нейронечіткої системи

Гібридні нейрон-нечіткі системи однорідні і зазвичай нагадують нейронні мережі. Тут нечітка система інтерпретується як особливий вид нейронної мережі. Перевага такої гібридної NFS полягає у її архітектурі, оскільки нечітка система та нейронна мережа більше не повинні взаємодіяти одна з одною. Вони є одне повністю злите ціле. Ці системи можуть навчатися онлайн та офлайн. На рисунку 2.1 показано структуру гібридної нейрон-нечіткої системи.

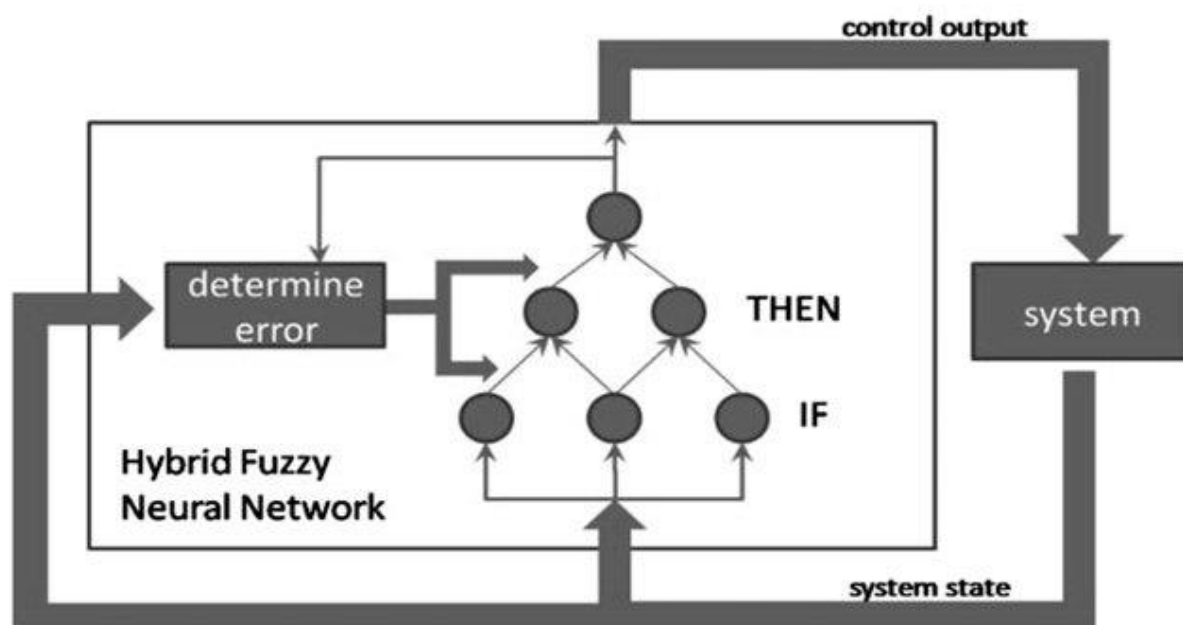


Рисунок 2.1 – Структура гібридної нейронечіткої системи

База правил нечіткої системи інтерпретується як нейронна мережа. Нечіткі множини можна розглядати як ваги, тоді як вхідні та вихідні змінні та правила моделюються як нейрони. Нейрони можуть бути увімкнені або видалені на етапі навчання. Нарешті, нейроні мережі є нечіткою основою

знань. Очевидно, що в такий спосіб долаються основні недоліки обох базових систем.

Щоб побудувати нечіткий контролер, необхідно визначити функції належності, що виражають лінгвістичні терміни правил виведення. Теоретично в нечітких множинах немає формального підходу до визначення цих функцій. Будь-яку форму (наприклад, трикутну, гауссівську тощо) можна розглядати як функцію належності з довільним набором параметрів. Таким чином, оптимізація цих функцій з точки зору узагальнення даних є дуже важливою для нечітких систем. Для вирішення цієї проблеми можна використовувати нейронні мережі.

ANFIS – це аббревіатура Adaptive Neuro-Fuzzy Inference System – адаптивна нейронна мережа нечіткого виведення. Запропонована на початку дев'яностих років [35], ANFIS є одним з перших варіантів гібридних нейрон-нечітких мереж – нейронна мережа прямого поширення сигналу особливого типу. Архітектура нейро-нечіткої мережі ізоморфна нечіткої основі знань. Нейро-нечіткі мережі використовують диференційовані реалізації трикутних норм (множення та ймовірнісне АБО), а також гладкі функції. Це дозволяє використовувати кросс-фаззи нейронні мережі, швидкі алгоритми навчання нейронних мереж, які засновані на методі зворотного поширення помилки. Архітектура та правила для кожного шару ANFIS описані нижче. ANFIS реалізує систему нечітких висновків Сугено у вигляді п'ятишарової нейронної мережі прямого поширення сигналу поширення [36]. Входи мережі в окремому шарі не виділено. На рисунку 2.2. показано приклад ANFIS мережі з двома вхідними змінними (x та y) та чотирма нечіткими правилами.

Типові процедури навчання нейронних мереж можна використовувати лише для налаштування мережі ANFIS. Зазвичай використовується комбінація градієнтного спуску як алгоритму зворотного поширення помилок і параметрів найменших квадратів.

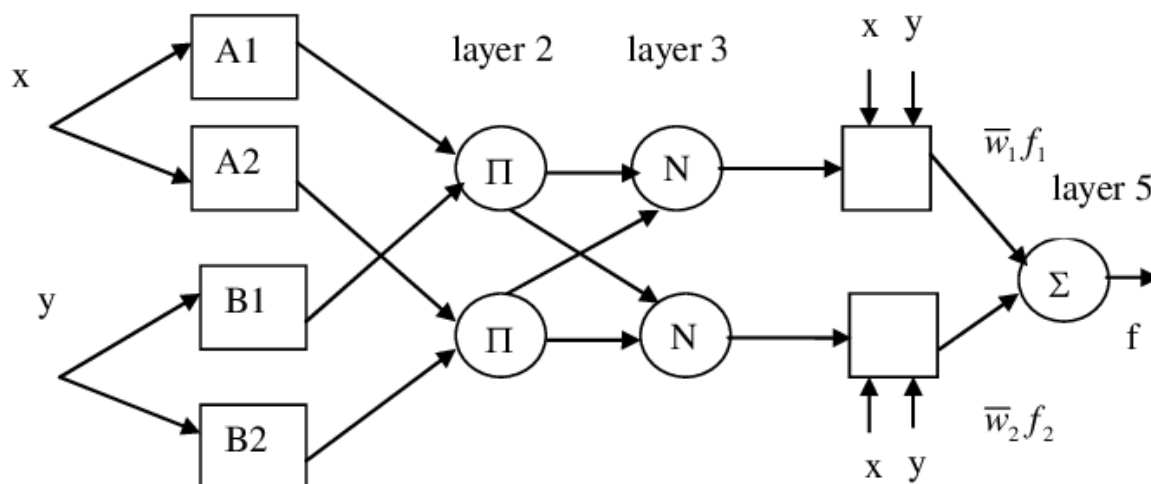


Рисунок 2.2 – Структура ANFIS

Алгоритм зворотного поширення помилок настроює параметри попередників правил, тобто. функції власності. Метод найменших квадратів оцінює коефіцієнти ув'язнених правил, оскільки вони лінійно пов'язані з виходом мережі. Кожна процедура виконується у два етапи. На першому етапі вводиться вибір навчання і знаходиться зв'язок між бажаною та фактичною поведінкою мережі; ітераційним методом найменших квадратів є оптимальні параметри вузлів четвертого шару. Метод зворотного розповсюдження помилок змінює параметри вузлів першого шару. Він узгоджений із першим етапом коефіцієнтів. Ітераційна процедура налаштування продовжується доти, доки не буде встановлено наперед задане значення. Наприклад, метод Левенберга-Марквардта.

На сьогоднішній день існує велика кількість різних підходів до нейрон-нечіткого прогнозування часових рядів, зокрема, для дослідження мереж використовуються генетичні алгоритми, методи роєвої оптимізації, а також різні гібридні методи.

Модель нечіткої нейронної мережі TS зазвичай включає дві частини: антецедентну мережу та посткомпонентну мережу. Антецедентну мережу використовують для обчислення придатності кожного правила;

посткомпонентна мережа є функцією вхідних лінгвістичних змінних, яка є лінійною комбінацією вхідних змінних [37].

Пропонована архітектура нечіткої нейромережі для передбачення тимчасової послідовності наведена на рисунку 2.3, і є багатошаровою мережею прямого поширення. Квадратна рамка представляє параметри, які необхідні для навчання. Функції кожного шару та відповідні формули виглядають наступним чином [36].

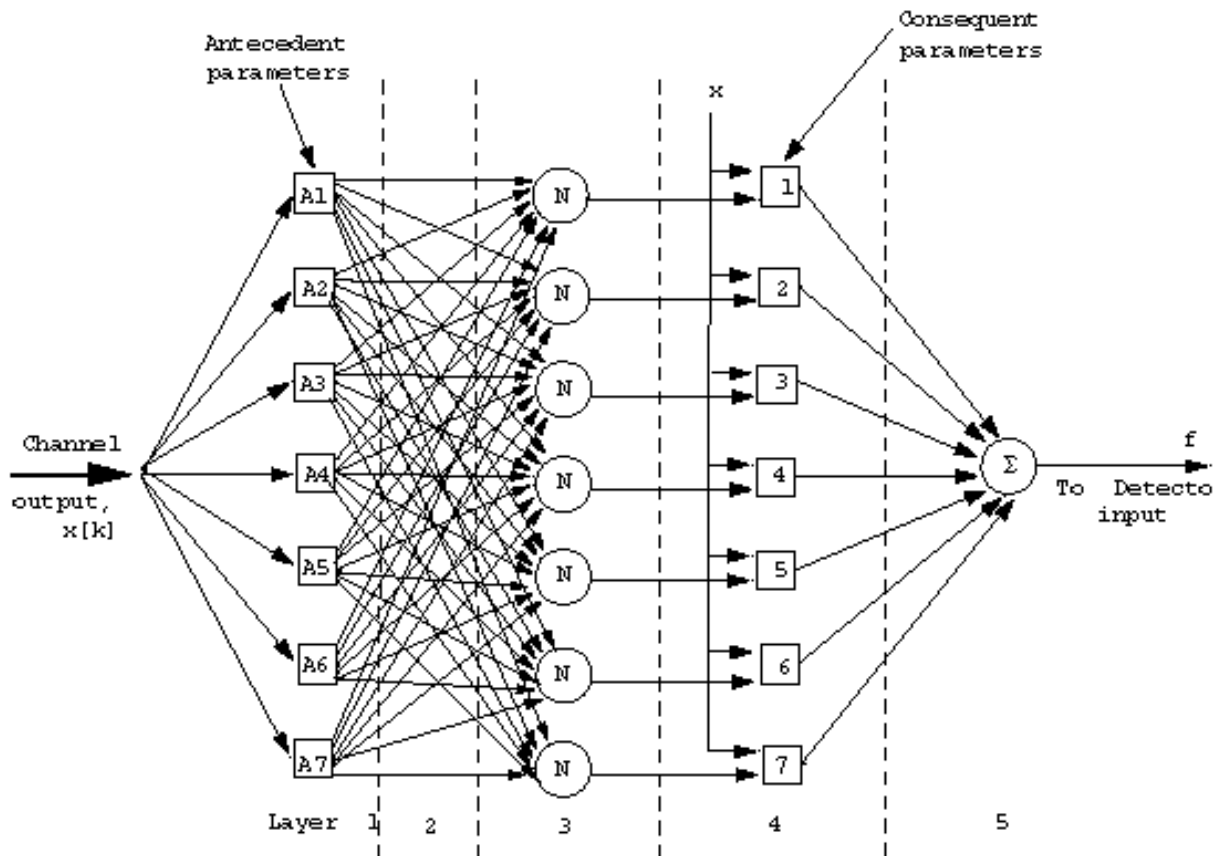


Рисунок 2.3 – Модель ANFIS

Перший шар відповідає за фаззифікацію вхідного сигналу відповідно до обраної функції належності, а вихідна функція вузла i має вигляд:

$$O_i^1 = \mu_{A_i}(x), \quad (2.1)$$

де O_i^1 – вихідний сигнал першого шару;

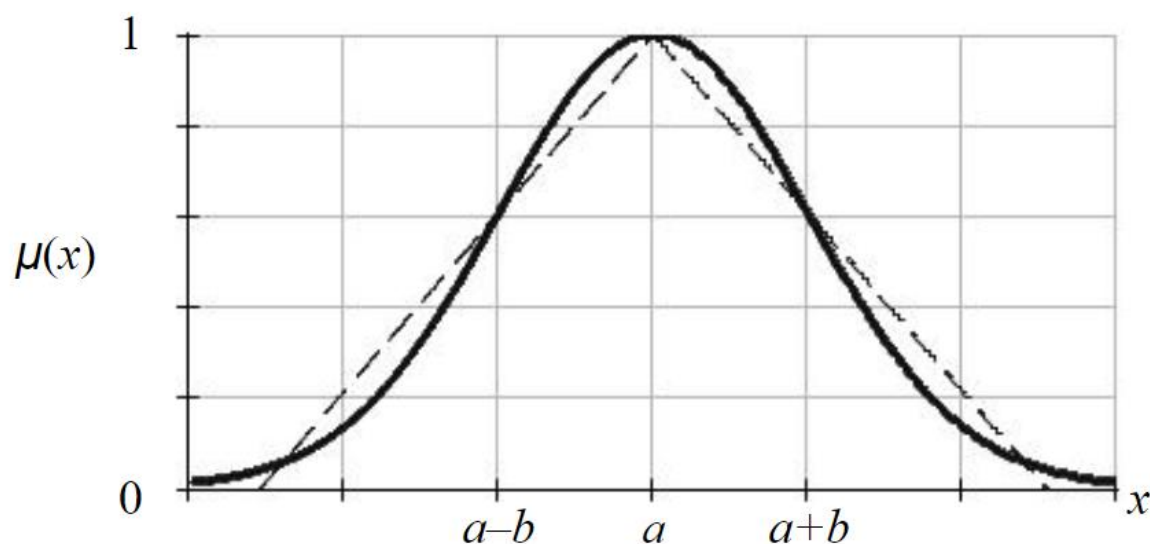
$\mu A_i(x)$ – відповідна параметризована функція власності нечіткої множини А.

Інструментарій нечіткої логіки (ІНЛ) містить безліч типів функцій належності (ФН), що формуються на основі шматково-лінійних функцій, розподілу Гауса, сигмоїдної кривої, квадратичних та кубічних поліноміальних кривих. До найпростіших ФП можна віднести трикутну та трапецієподібну.

Традиційно використовується Гаусова функція належності, загальний вигляд якої є:

$$\mu(x) = e^{-\frac{(x-a)^2}{2b^2}}. \quad (2.2)$$

Графік функції (рисунок 2.4) за своєю формою збігається з графіком щільності нормального ймовірнісного розподілу.



Рисунку 2.4 – Загальний вигляд Гаусової функції належності

Експонентна функція належності характеризується двома параметрами: a та b , ($b > a$), при цьому в точці c (і тільки в ній) функція набуває значення 1, а точки $x = a \pm b$, є точками перегину функції.

Другий шар відповідає за множення вхідного сигналу. Вихід кожного вузла другого шару є достовірністю даного правила, і є оператором AND, який задовольняє будь-якій Т-нормальній формі.

$$O_i^2 = w_i = \mu A_i(x) \times \mu B_i(y). \quad (2.3)$$

Третій рівень нормалізації відповідає за обчислення нормалізаційної довіри правила і на вузлі і:

$$O_i^3 = \bar{w} = \frac{w_i}{w_1 + w_2 + \dots + w_n}. \quad (2.4)$$

Четвертий шар дефазифікації відповідає за вихід вузла і:

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i x + r_i), \quad (2.5)$$

де p , q , r – параметр посткомпоненти.

П'ятий шар, що підсумовує, обчислює загальний вихід мережі як суму всіх вхідних сигналів:

$$O_i^5 = \sum_i \bar{w}_i f_i. \quad (2.6)$$

Можна відзначити, що оскільки вибір параметрів нечітких нейросистем є основною проблемою в нечіткому моделюванні, і оскільки більшість цих параметрів вибираються на основі досвіду користувачів та/або методом спроб і помилок, то для того, щоб автоматизувати цей процес та мінімізувати помилку, можна використовувати алгоритми

машинного навчання. Метою даної роботи є використання нейромережі з представленою вище архітектурою для побудови системи нечітких висновків (FIS). Параметри якої можна налаштовувати за допомогою алгоритму зворотного розповсюдження помилки, що дозволяє нечіткій системі вловлювати просторові відносини між даними та ефективно надавати оцінку.

Слід зазначити, деякі параметри все ще залишаються неоптимальними. Фактично алгоритм навчання намагається оптимізувати параметри нечіткої логіки, тобто параметри функцій належності нечітких множин, у той час як він може застрягти в локальному мінімумі. Щоб подолати цю проблему, ми спробуємо використати генетичний алгоритм навчання як потужний інструмент для пошуку оптимальних параметрів функцій належності для кожного входу та найкращих значень швидкості навчання.

2.2 Навчання нечіткої нейромережі з використанням генетичного алгоритму

Вперше генетичний алгоритм (ГА) було представлено Холландом у 1975 р. Це універсальний метод вирішення різноманітних завдань оптимізації з обмеженнями і без обмежень [38].

ГА також може бути використаний для вирішення різноманітних задач оптимізації, які погано піддаються стандартним алгоритмам оптимізації, включаючи завдання, в яких об'єктивна функція є перервною, недиференційованою, стохастичною або нелінійною [39]. Деякі дослідники також припустили, що методи глобального пошуку, включаючи ГА, можуть запобігти попаданню нейромережі в локальний мінімум.

Початкова популяція буде модифікована задля досягнення кращої відповіді. На кожному кроці ГА вибирає особин (хромосоми) з поточної популяції (батьків) випадковим чином і використовує їх для нащадків для

наступного покоління. Після кількох поколінь, відповідно до сутності ГА, він намагається перейти до найкращого рішення (рисунок 2.5).

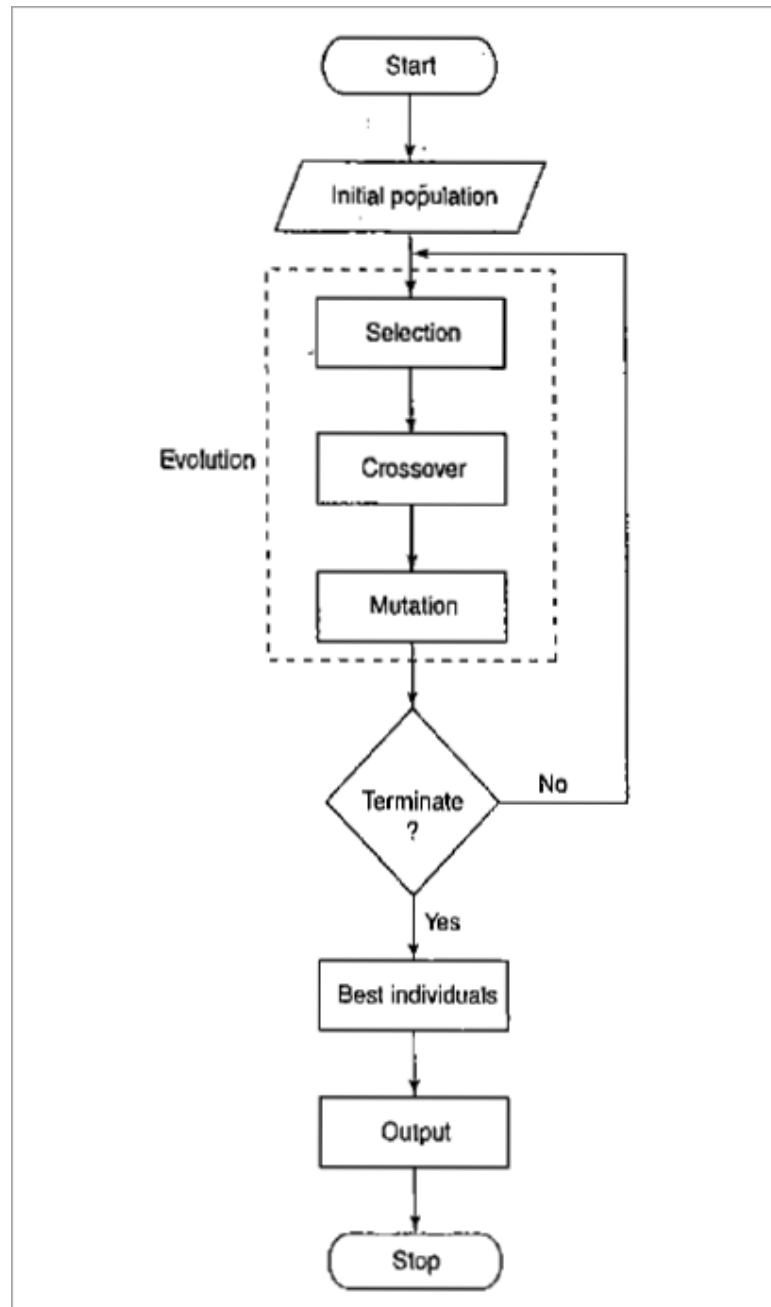


Рисунок 2.5 – Схема генетичного алгоритму

На кожному кроці ГА використовує три основні типи правил створення наступного покоління з поточної популяції.

Ці типи правил розглядаються наступним чином [40]. Правила відбору вибирають особин, званих батьками, які роблять свій внесок у популяцію в наступному поколінні. Правила кросинговеру поєднують хромосоми для створення наступного покоління.

Правила мутації (рисунок 2.6) приводять хромосоми до зміни та змінюють їх значення. Спочатку змінні мають бути представлені двійковим рядком, який кодує параметри нечіткої нейромережі, а кожна хромосома складається з кількох генів, які представляють параметри мережі.

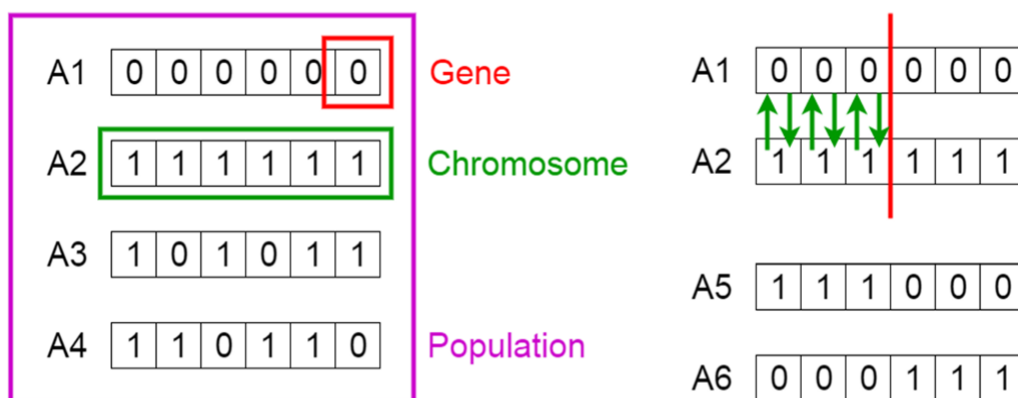


Рисунок 2.6 – Мутації хромосом у популяції

Потім створюється популяція рядків з початковими випадковими параметрами як кандидати на краще рішення (тобто гени-хромосоми-популяція-покоління). Після цього значення фітнес-функції (яка в основному пов'язана з різницею між вихідним результатом нейромережі та реальною оцінкою) сортуються, а потім визначаються найкращі та найгірші хромосоми (зверніть увагу, що лише найкращі хромосоми можуть схрещуватися або мутувати за рейтингом). Таким чином, ГА із зазначеними параметрами використовується для отримання оптимальних параметрів нейромережі (наприклад, кількість вхідних нейронів, функції належності, швидкість навчання).

2.3 Хаотичний часовий процес

Прогнозування часових рядів – актуальна тема, яка має безліч можливих застосувань, таких як прогнозування цін на акції, прогнозування погоди, бізнес-планування, розподіл ресурсів та багато інших. Попри те що, що прогнозування можна як підмножина завдань регресії з спостереженням, через тимчасової природи спостережень необхідні деякі специфічні інструменти.

Тимчасовий ряд зазвичай моделюється через стохастичний процес $Y(t)$, тобто. послідовність випадкових змінних. В умовах прогнозування ми знаходимося в момент часу t та зацікавлені в оцінці $Y(t + h)$, використовуючи лише інформацію, доступну в момент часу t .

Хаотична система – це система, у якій дуже невелика зміна початкових умов може призвести до іншого результату. У таких системах неможливо точно передбачити майбутнє навіть якщо ви знаєте детерміновані рівняння, що описують майбутнє. Це з тим, що для обчислення знадобиться нескінченно точний комп'ютер і точні початкові умови. Насправді це принципово неможливо. У зв'язку з цим, як прогноз з реального життя, прогнози погоди точні тільки на короткому проміжку часу. Вони покращуються з часом у міру вдосконалення динамічних та статистичних моделей та збільшення обчислювальної потужності. Проте неможливо точно передбачити щоденну погоду у довгостроковій перспективі, оскільки це є хаотичною системою.

Для створення хаотичного часового ряду використовуватимемо рівняння Маккі-Гласа.

Метою даної роботи є створення нечіткої системи для прогнозування тимчасових рядів для диференціального рівняння із затримкою Маккі-Гласа:

$$\frac{dx(t)}{dt} = \beta \frac{x(t-\tau)}{1+x^n(t-\tau)} - \gamma x(t). \quad (2.7)$$

Мета – у час t_1 передбачити $x(t_1+1)$, вибираємо $\tau=30$. Нам потрібно встановити $n \sim 10$, тому що саме за такого значення з'являється хаос. Інші два параметри встановлюються наступним чином: $\beta = 0,2$ та $\gamma = 0,1$.

Ми можемо апроксимувати функцію (1) таким чином:

$$x(t + 1) = x(t) + t_1 \frac{dx(t)}{dt}. \quad (2.8)$$

Складемо рівняння (2.7) та (2.8) разом і $t_1 = 1$, тому що ми йдемо крок за кроком за зразками.

Наступне рівняння є поточною апроксимацією:

$$x(t + 1) = x(t) + \beta \frac{x(t-\tau)}{1+x^n(t-\tau)} - \gamma x(t). \quad (2.9)$$

Підставивши певні константи в рівняння (2.9), отримаємо:

$$x(t + 1) = x(t) + 0.2 \frac{x(t-30)}{1+x^{10}(t-30)} - 0.1x(t). \quad (2.10)$$

Це і є остаточна форма рівняння.

Для побудови графіка даного тимчасового процесу як вхідний набір ми використовуємо 30 випадкових значень з діапазону (0, 1) (рисунок 2.7).

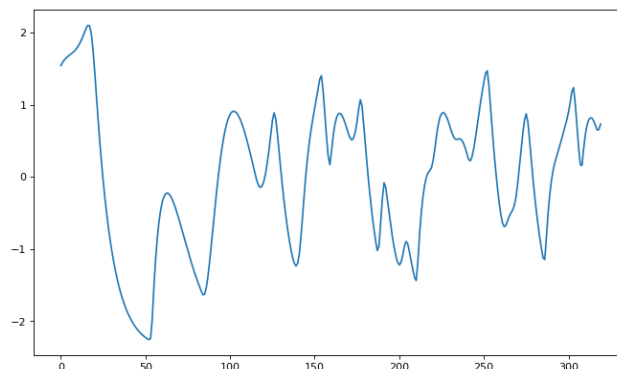


Рисунок 2.7 – Тимчасовий ряд за рівнянням Маккі-Гласса.

3 ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ І РІШЕННЯ ЗАДАЧ НА ТЕСТОВИХ ВИБІРКАХ

У цьому розділі наведено результати моделювання розробленої моделі нечіткої нейронної мережі. Навчання моделі та перевірка вихідних даних виконувалася за допомогою згенерованої часової послідовності відповідно до рівняння Макі-Гласу. Результати моделювання запропонованого методу були порівняні зі стандартними методами кластеризації з метою оцінки порівняльної якості вирішення розглянутих завдань. В якості основних засобів для реалізації синтезованої в попередньому розділі архітектури гібридної нечіткої нейронної мережі було використано мову програмування Python 3, яка на даний час є найпопулярнішою мовою для розробок систем, пов'язаних з обчислювальним інтелектом. Також було використано ряд бібліотек та програмних пакетів, а саме:

- TensorFlow 2.0 – бібліотека з відкритим кодом, надає велику екосистему інструментів для розробників, підприємств та дослідників, надає можливості просувати передові досягнення в галузі машинного навчання та створювати масштабовані додатки на базі ML;

- Keras – це API глибокого навчання, написаний мовою Python і працює поверх платформи машинного навчання TensorFlow. Він був розроблений з упором забезпечення можливості швидкого проведення експериментів. Надає важливі абстракції та будівельні блоки для розробки та впровадження рішень машинного навчання із високою швидкістю ітерацій;

- PyGAD – це бібліотека Python з відкритим вихідним кодом для побудови генетичного алгоритму та оптимізації алгоритмів машинного навчання. Вона працює з Keras та PyTorch. PyGAD підтримує різні типи операторів кросинговеру, мутації та вибору батьків. PyGAD дозволяє

оптимізувати різні типи завдань за допомогою генетичного алгоритму, настроюючи фітнес-функцію;

– SciKit – це безкоштовна програмна бібліотека машинного навчання для мови програмування Python, яка надає функціональність для створення та тренування різноманітних алгоритмів класифікації, регресії та кластеризації, таких як лінійна регресія, random forest, градієнтний бустинг, та працює у зв'язці з бібліотеками NumPy та SciPy. Scikit-learn є однією з найпопулярніших бібліотек машинного навчання;

– NumPy (Numeric Python) – це open-source модуль для python, який надає загальні математичні та числові операції у вигляді прескомпільованих, швидких функцій. Вони поєднуються у високорівневі пакети. Вони забезпечують функціонал, який можна порівняти із функціоналом MatLab. NumPy надає базові методи для маніпуляції з великими масивами та матрицями;

– Matplotlib – бібліотека мовою програмування Python для візуалізації даних двовимірною (2D) графікою (3D графіка також підтримується);

– програмування та виконання програм було виконано у середовищі Google Colaboratory (або Colab), який дає змогу писати та виконувати код Python у веб-переглядачі.

3.1 Імітаційне моделювання нечіткої нейронної мережі

За допомогою реалізації рівняння Макі-Гласса (2.10) було згенеровано часову послідовність, яка складається з 1600 значень. На рисунку 3.1 приведено графік цієї часової послідовності.

За допомогою бібліотеки SciKit вхідні дані часової послідовності було нормалізовано. З метою навчання створеної моделі нечіткої нейромережі згенерована послідовність була опрацьована певним чином, а саме розділена на навчальну та тестову послідовність у співвідношенні 6:4.

Також вся послідовність була розділена на пакети розміром 16 значень для послідовного пакетного навчання.

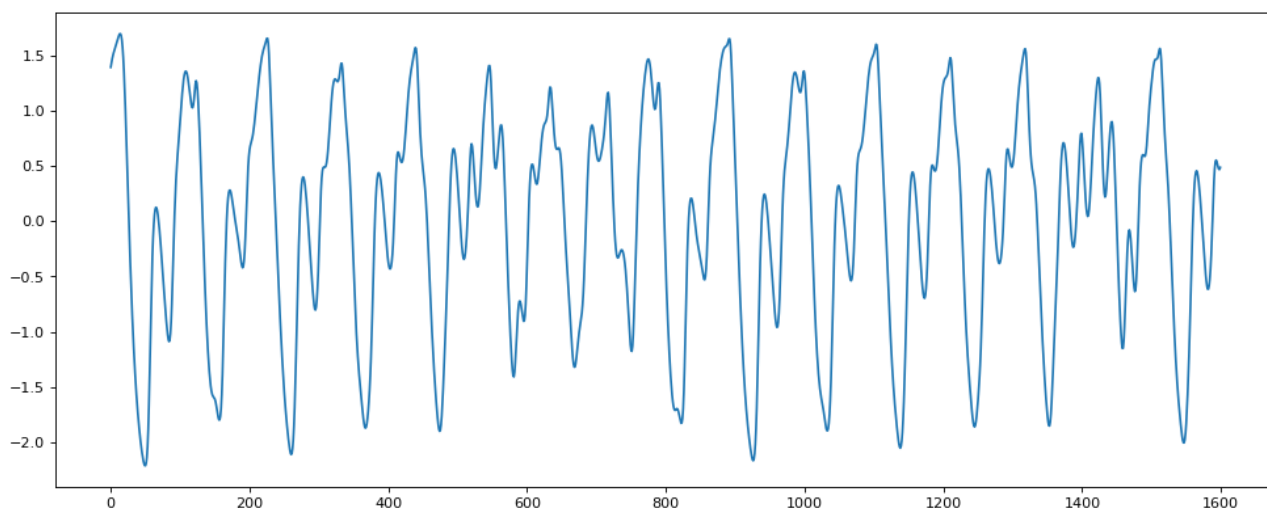


Рисунок 3.1 – Часова послідовність, яка використовується для навчання нейронної мережі.

Для створення моделі нейромережі згідно з архітектурою, розглянутою у попередньому розділі, за допомогою бібліотеки Керас було створено кілька додаткових шарів нейромережі, які відсутні в стандартному наборі шарів, які пропонує бібліотека, а саме шари фаззифікації, дефаззифікації, нормалізації та складання вихідного сигналу. Використовуючи ці шари а також шар вхідних значень створюємо послідовну модель та поєднуємо шари згідно структури, що представлена на рисунку 2.2.

Далі виконаємо навчання створеної моделі за допомогою генетичного алгоритму, використовуючи бібліотеку PyGAD. Оберемо наступні параметри генетичного алгоритму для навчання: кількість поколінь – 70, кількість особин для схрещування – 2. Також реалізуємо користувальницьку функцію для використання в якості фітнес-функції. Ця функція повинна бути функцією максимізації, тобто обраним рішенням буде обрано те, що має більш високе значення. Ця функція реалізована за допомогою алгоритму

середньої абсолютної похибки. Середня абсолютна помилка (MAE) – це міра помилок між парними спостереженнями, що виражають те саме явище. Приклади Y проти X включають порівняння передбаченого зі спостережуваним, наступного часу з початковим та одного методу вимірювання з альтернативним методом вимірювання.

Програмний код моделі нейромережі, наведених вище шарів, функцій що будують часові послідовності а також алгоритм навчання та тестування моделі нечіткої нейромережі наведені у додатку А.

3.2 Аналіз отриманих результатів

Результати, отримані під час навчання нейронної мережі відображені на графіку залежності значень функції придатності від покоління (рисунок 3.2).

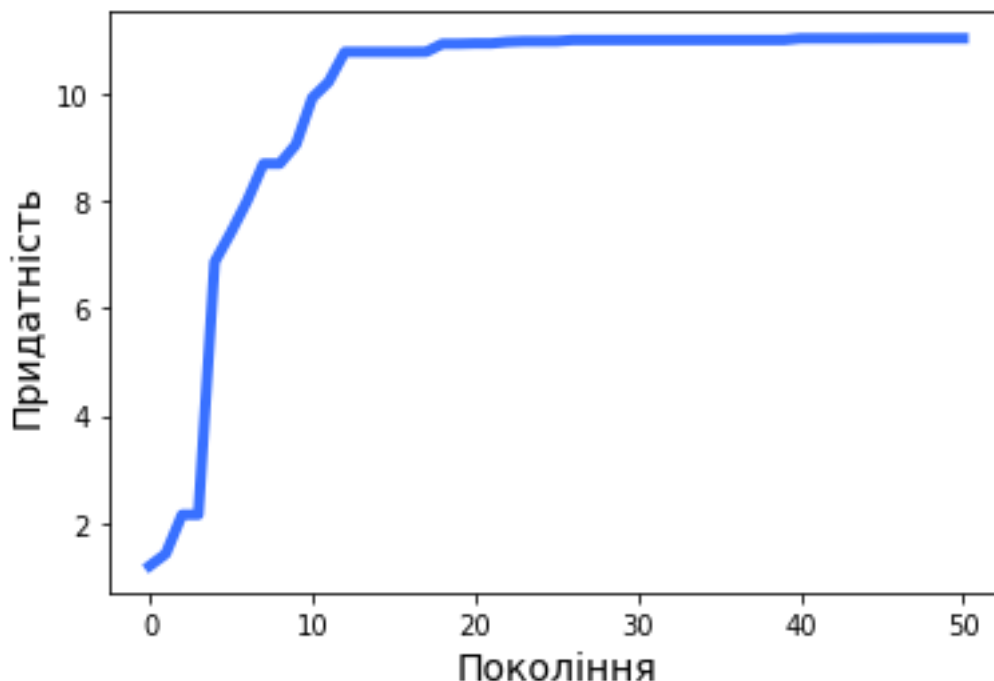


Рисунок 3.2 – Графік зміни значення функції придатності від покоління під час навчання нейронної мережі

Також можна порівняти параметри функцій належності, що змінилися в результаті навчання шару фаззифікації в нейронній мережі (рисунок 3.3). Пунктирною лінією на графіку відображені початкові значення функцій належності присвоєні під час створення ненавчаної моделі нейромережі.

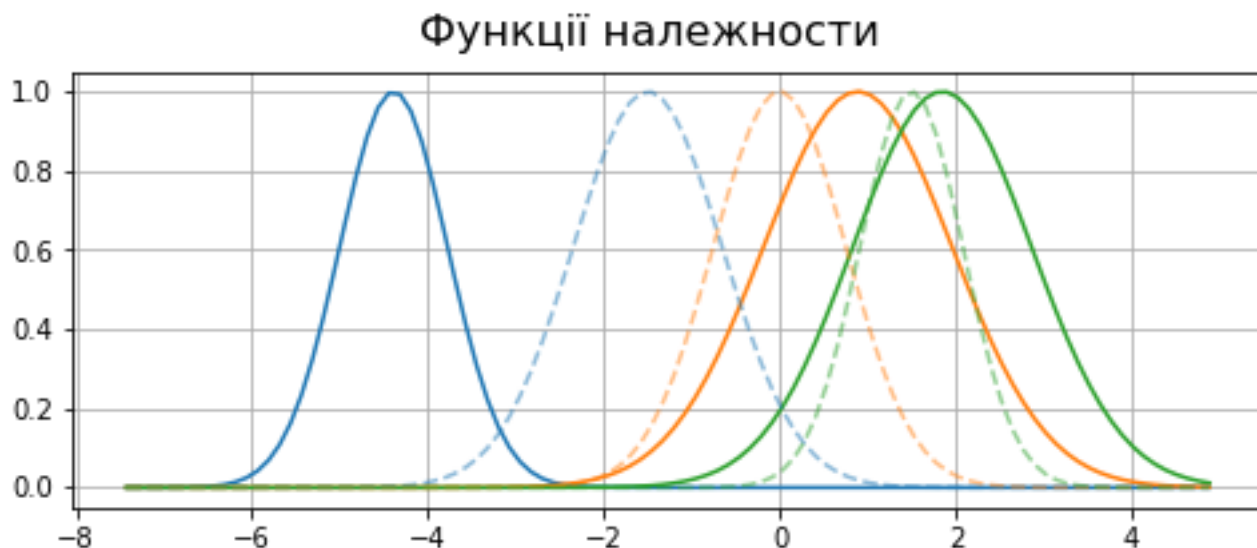


Рисунок 3.3 – Графік зміни параметрів функцій належності шару фаззифікації в результаті навчання нейронної мережі

Для перевірки результатів навчання нейронної мережі використаємо дані часової послідовності Макі-Гласса, але вже кількістю, більшою ніж тою, що використовувалась для навчання нейромережі, а саме 3200. Передавши цю послідовність на вхід нейромережі отримаємо часову послідовність, передбачену нейромережею. На рисунку 3.4 відображено графіки послідовностей, де синім кольором відображено графік вхідної послідовності, а жовтим – графік передбаченої послідовності.

Порівнявши їх можемо визначити розмір похибки передбачення як різницю між вихідною та вхідною послідовностями. Графік різниці між реальною та передбаченою послідовностями відображено на рисунку 3.5.

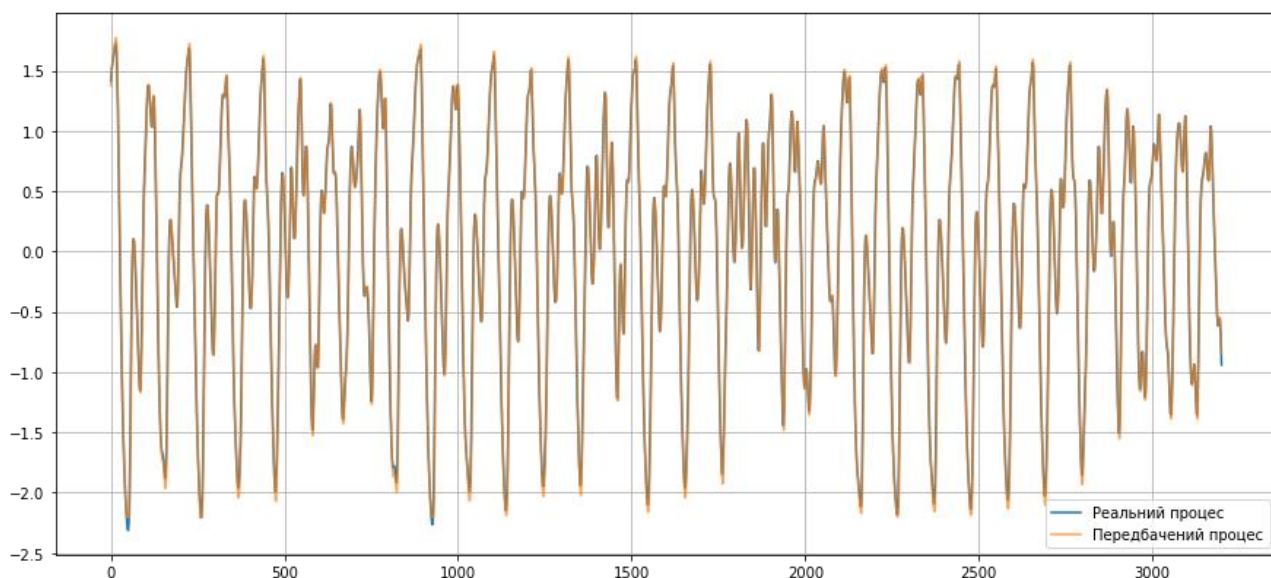


Рисунок 3.4 – Графіки реальної часової послідовності та послідовності, передбаченої навченою нейромережею

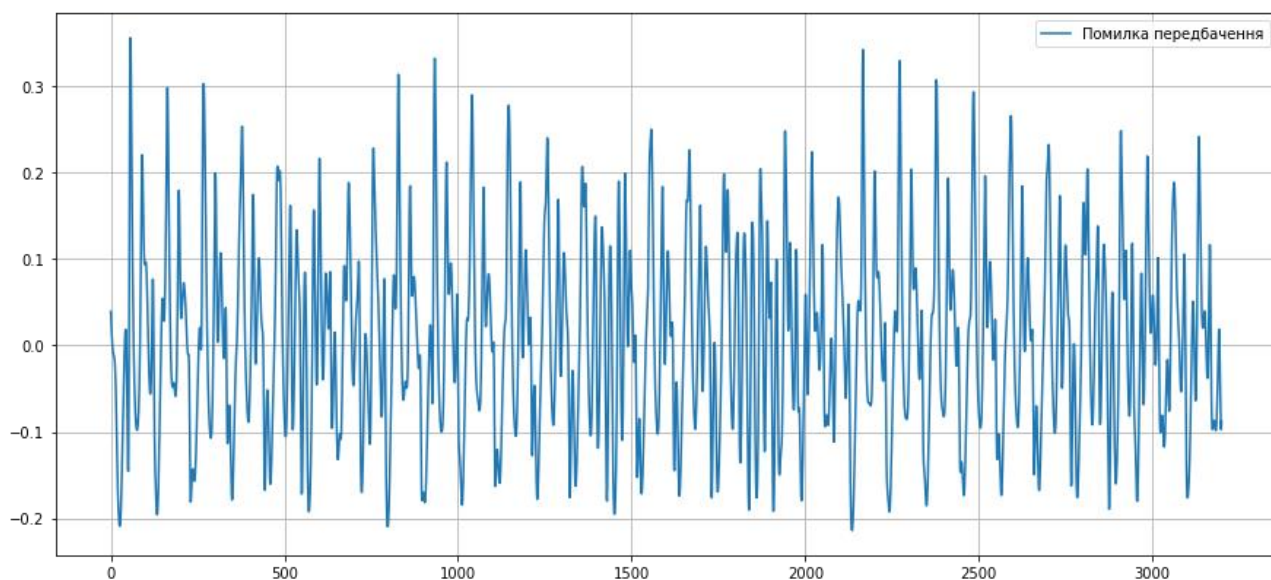


Рисунок 3.5 – Графіки помилки передбачення часової послідовності нейромережею

Для оцінки якості роботи розробленої моделі та впливу гіперпараметрів нейронної мережі на якість порівняємо показники точності

передбачення для кількох моделей з різною кількістю функцій належності в шарі фаззифікації.

Показник точності передбачення відображає певну характеристику нейронної мережі, а саме, наскільки точно навчана модель відображає дані. Виходячи з цього, чим точніше модель відображає дані, тим точніше вона зробить передбачення.

Для розрахунку показника точності використаємо наступну формулу

$$A = (1 - \sigma), \quad (3.1)$$

де σ – середньоквадратичне відхилення похибки передбачення.

Результати розрахунків показників точності наведено у таблиці 3.1

Таблиця 3.1 – Результати розрахунків показників точності нечіткої нейромережі в залежності від кількості функцій належності в шарі фаззифікації

Кількість функцій належності	Максимальна абсолютна похибка предбачення	Середньоквадратичне відхилення похибки предбачення	Показник точності предбачення, %
2	0,4434	0,1368	86,3
3	0,4	0,1229	87,7
4	0,3	0,1186	88,2
5	0,378	0,1118	88,8
10	0,347	0,1120	88,79

ВИСНОВКИ

В роботі представлено результати предбачення часової послідовності на прикладі хаотичного процесу Макі-Гласса за допомогою нечіткої нейронної мережі відповідно до поставленої мети. Проведені дослідження дозволили зробити наступні висновки:

1) проаналізовано теоретичні засади теорії нечіткої логіки та існуючі принципи щодо реалізації нечітких систем. Розглянуто існуючі нечіткі системи. Проаналізовано алгоритми навчання нейронних мереж за допомогою еволюційних алгоритмів, використання цих інструментів в задачах прогнозування часових послідовностей;

2) відповідно до поставленої мети було визначено схему та запропоновано структуру нечіткої системи;

3) запропоновано архітектуру гібридної нечіткої нейронної мережі призначеної для передбачення часових послідовностей, які послідовно в online режимі надходять на обробку та метод її самонавчання, заснований на еволюційному алгоритмі;

4) за допомогою програмних засобів було створено модель нечіткої нейронної мережі, її навчання за допомогою еволюційного алгоритму;

5) запропонована система та модель дозволяє виконувати передбачення часових послідовностей, що було підтверджено при тестуванні на часовій послідовності Макі-Гласса. Виходячи з показників точності передбачення створеної моделі на точність передбачення впливають параметри слою фазифікації, а саме кількість функцій належності. Тобто зі збільшенням кількості функцій належності у шарі фазифікації збільшується точність передбачення моделі, хоча й збільшується час навчання нейромережі. Треба помітити, що зі збільшенням кількості функцій фазифікації точність прогнозу не збільшується пропорційно, тому що через велику кількість функцій вони перетинаються перекриваючи одна одну, що є збитковим фактором.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Leski J. *Neuro-Fuzzy Systems*. Warsaw : WNT, 2008.
2. Kacprzyk J. *Multistage Decision-making Under Fuzziness: Theory and Applications*. Rheinland, Cologne : Verlag TV, 1983.
3. Mamdani E., Assilian S. An experiment in linguistic synthesis with fuzzy logic controller. *International Journal of Man-Machine Studies*. 1975. Vol. 20, No. 2, P. 1–13.
4. Hellendoorn H., Driankov D. *Fuzzy Model Identification. Selected Approaches*. Berlin : Springer, 1997.
5. Marszalek A., Burczynski T. Modeling and forecasting finance time series with ordered fuzzy candlesticks. *Information Sciences*. 2014. No. 273, P. 144-155.
6. Stachowiak A., Dyczkowski K., Wojtowicz A., Zywica P., Wygralak M. A Bipolar View on Medical Diagnosis in OvaExpert System. Berlin: Springer 2016. 400 p.
7. Yu X., Kacprzyk J. *Applied Decision Support with Soft Computing*. Heidelberg: Springer. 2003.
8. Czogala E., Leski J. *Fuzzy and Neuro-Fuzzy Intelligent Systems*. Heidelberg: Springer. 2000.
9. Rutkowski L. *New Soft Computing Techniques for System Modeling, Pattern Classification and Image Processing*. Heidelberg: Springer. 2004.
10. Zadeh L. Outline of new approach to analysis of complex systems and decision processes. *IEEE Trans. Syst. Man Cybern.* 1973. Vol 3. No 1. P. 28–44.
11. Zadeh L. Концепція лінгвістичної варіації та його застосування до прикладу reasoning-I. *Inf. Sci.* 1975. № 8. P. 199–249.
12. Dubois D., Prade H. What are fuzzy rules and how to use them. *Fuzzy Sets Syst.* 1996. № 84. P. 169–185.
13. Fodor J. On fuzzy implication operators. *Fuzzy Sets Syst.* 1991. № 42. P. 293–300.

14. Rutkowska D. Neuro-Fuzzy Architectures and Hybrid Learning. Physica-Verlag, Springer Comp., Heidelberg. 2002.
15. Wang L.X. A Course in Fuzzy Systems and Control. Prentice-Hall, Upper Saddle River. 1997.
16. Leekwijck W., Kerre E. Defuzzification: criteria and classification. *Fuzzy Sets Syst.* 1999. №108. P. 159–178.
17. Rutkowski L. New Soft Computing Techniques for System Modeling, Pattern Classification and Image Processing. Physica-Verlag, Springer Comp., Heidelberg. 2004.
18. Sugeno M., Yasukawa T. A fuzzy-logic-based approach до qualitative modeling. *IEEE Trans. Fuzzy Syst.* 1993. № 1(1). P. 7–31.
19. Takagi T., Sugeno M. Fuzzy identification of systems and its application to modeling and control. *IEEE Trans. Syst. Man Cybern.* 1985. № 15(1). P. 116–132.
20. Sugeno M., Kang G. Structure identification of fuzzy model. *Fuzzy Sets Syst.* 1988. № 28. P. 15–33.
21. Leski J., Czogala E. New artificial neural network заснована на fuzzy inference system with moving consequents in if-then rules and its applications. *Fuzzy Sets Syst.* 1999. № 108. P. 289–297.
22. Leski J. ε -insensitive fuzzy c-regression models: introduction to ε -insensitive fuzzy modeling. *IEEE Trans. Syst. Man Cybern.* 2004. Part B Cybern. № 34(1). P. 4–15.
23. Tsukamoto Y. An approach до fuzzy reasoning method. *Advances in Fuzzy Set Theory and Applications.* North-Holland, Amsterdam. 1979. P. 137–149.
24. Karnik N., Mendel J., Liang Q. Type-2 fuzzy логічні системи. *IEEE Trans. Fuzzy Syst.* 1999. № 7(6). P. 643–658.
25. Liang Q., Mendel J. Interval type-2 fuzzy логічні системи: theory and design. *IEEE Trans. Fuzzy Syst.* 2000. № 8(5). P. 535–550.

26. Türksen I. Four methods for approximate reasoning with interval-valued fuzzy sets. *Int. J. Approx. Reason.* 1989. № 3. P. 121–142.
27. Türksen I. Type I i type II fuzzy system modeling. *Fuzzy Sets Syst.* 1999. № 106. P. 11–34.
28. David E. Rumelhart, Geoffrey E. Hinton & Ronald J. Williams. Learning representations by back-propagating errors. *Nature.* 1986. Vol 323. P. 533–536.
29. T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* Oxford, UK: Oxford University Press. 1996
30. Goldberg D. E. *Genetic Algorithms in Search Optimization and Machine Learning.* Addison-Wesley. 1989.
31. Rechenberg I., Bergmann H. W. Evolution strategy: Nature's way of optimization. In, editor, *Optimization: Methods and Applications, Possibilities and Limitations.* Berlin: Heidelberg, 1989. P. 106–126.
32. Fogel L., Owens A., Walsh M. *Artificial intelligence through simulated evolution.* Chichester, WS, UK: Wiley. 1966.
33. Koza J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge, MA, USA: MIT Press. 1992.
34. Fleming P., Purshouse R. Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice.* 2002. Vol 10. No 11. P. 1223–1241.
35. Jang J.-S.R. Anfis: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics.* 1993. Vol. 23. No. 3. P. 665–685.
36. Kong W. Short-Term Residential Load Forecasting based on Resident Behaviour Learning. *IEEE Transactions on Power Systems.* 2018. Vol 33. No 1. P. 1087–1088.

37. Yarushev S. A., Averkin A. N. Review of studies on time series forecasting based on hybrid methods, neural networks and multiple regression. Программные продукты и системы. 2016. No. 1. P. 113.
38. Holland J. H. Genetic Algorithms and Adaptation. Boston. MA: Springer, 1984. Vol.16. P. 317–333.
- 39 Goldberg D. E., Holland J. H. Genetic Algorithms and Machine Learning. Machine Learning. 1988. Vol.3. P. 95–99.
40. Deb K. An introduction to genetic algorithms. Sadhana. 1999. Vol.24. P. 293–315.