

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Штучного інтелекту  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

Каскадна нечітка система для вирішення задач  
динамічного аналізу даних  
(тема)

Виконав:  
студент 2 курсу, групи СШМ-19-2  
Ціунчик Л. М.  
(прізвище, ініціали)

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

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

Освітня програма Системи штучного інтелекту  
(повна назва спеціалізації)

Керівник к.т.н., доцент, Чала Л. Е.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

В.О. Філатов  
(прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

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

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Ціунчика Лева Михайловича \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ Каскадна нечітка система для вирішення задач \_\_\_\_\_  
динамічного аналізу даних \_\_\_\_\_

затверджена наказом університету від 29 березня 2021 р. № 390Ст

2. Термін подання студентом роботи до екзаменаційної комісії \_\_\_\_\_ 2021р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет та відомих наукових \_\_\_\_\_  
проектів, електронні документації, тестові набори даних \_\_\_\_\_

4. Перелік питань, що потрібно опрацювати в роботі аналіз предметної області та постановка \_\_\_\_\_  
задачі дослідження, основні методи навчання та самонавчання нейронних мереж, системи \_\_\_\_\_  
нечіткого висновування, каскадні нейронні мережі, навчання нео-фаззі нейронів середовище \_\_\_\_\_  
реалізації, опис алгоритму, аналіз отриманих результатів \_\_\_\_\_

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1 – Архітектура каскадної системи за Фальманом та Леб'єром після додавання двох вузлів, Рисунок 2 – Каскадно-кореляційна нейронна мережа, Рисунок 3 – Архітектура гібридної системи з оптимізованим пулом нейронів, Рисунок 4 – Архітектура нео-фаззі нейрону, Рисунок 5 – Синапс розширеного нео-фаззі нейрону, Рисунок 6 – Розширений нео-фаззі нейрон, Рисунок 7 – Схема системи ідентифікації, Рисунок 8 – Схема зворотного моделювання, Рисунок 9 – Схема сліпої сепарації, Рисунок 10 – Схема нейромережевого прогнозування, Рисунок 11 – Фазовий портрет часового ряду «Споживання електроенергії у Польщі за період з 1990-х років», Рисунок 12 – Прогнозування часового ряду «Споживання електроенергії у м. Сімферополь за 2007 рік» оптимізованою каскадною нейро-фаззі мережею, Рисунок 13 – Фазовий портрет часового ряду «Коливання рівню прилогоотливної зони», Рисунок 14 – Прогнозування часового ряду «Коливання рівню прилогоотливної зони» гібридної каскадної нейро-фаззі мережі з оптимізацією пулу нейронів, Рисунок 15 – Прогнозування часового ряду «ESTP Competition Time Series» гібридної каскадної нейро-фаззі мережі з оптимізацією пулу нейронів

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
Основна частина	доц. каф. ШІ Чала Л.Е.		

#### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Отримання завдання на атестаційне проектування	29.03.2021	виконано
2.	Аналіз завдання та пошук літератури за темою	30.03-05.04	виконано
3.	Опрацювання літератури та аналіз об'єкту	06.04-12.04	виконано
4.	Вибір програмних засобів для розробки системи	13.04-19.04	виконано
5.	Розробка програмного засобу	20.04-03.05	виконано
6.	Аналіз отриманих результатів	04.04-06.05	виконано
7.	Оформлювання пояснювальної записки	06.05-11.05	виконано
8.	Оформлення презентаційних матеріалів	12.05.2021	виконано
9.	Представлення на рецензування	13.05.2021	виконано
11.	Представлення кваліфікаційної роботи	20.05.2021	

Дата видачі завдання 29 березня 2021 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

доц. Чала Л. Е.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 84 с., 16 рис., 3 табл., 99 формул, 2 дод, 32 джерела

КАСКАДНА НЕЙРОННА МЕРЕЖА, НЕЙРОННА МЕРЕЖА, НЕЙРО-ФАЗЗИ СИСТЕМА, НЕО-ФАЗЗИ НЕЙРОН, СИСТЕМА НЕЧІТКОГО ВИСНОВУВАННЯ, ЧАСОВИЙ РЯД.

Об'єктом дослідження є процес прогнозування часових рядів за умов послідовного надходження даних на опрацювання.

Предметом дослідження є каскадна нечітка система, що еволюціонує, призначена для вирішення задач динамічного інтелектуального аналізу даних.

Метою дослідження є розробка каскадної нечіткої системи і методу її навчання з підвищеною швидкістю і можливостями інтерпретовності вихідного сигналу, а також параметричного та структурного налаштування в режимі послідовного опрацювання інформації.

Методи дослідження. Теорія еволюційних систем обчислювального інтелекту, що включає різні гібридні системи побудовані на основі нейронних мереж та нечіткої логіки, методи оптимізації другого порядку, використання Гавсівсько-Ньютоновських методів оптимізації дозволяє отримати високу збіжність методів навчання нейро-фаззи систем, методи EDA аналізу дозволили провести первинний аналіз часових рядів та побудувати вихідну аналітику по побудованим прогнозуючим моделям.

Завдяки стрімкому розвитку інформаційних технологій можна зробити припущення, що розробка каскадної нейро-фаззи архітектури, а також методу її налаштування і навчання, що дозволить подолати обмеження існуючих нейро-фаззи мереж та вирішувати задачі динамічного аналізу даних є актуальною та необхідною.

## РЕФЕРАТ

Пояснительная записка: 84 с., 16 рис., 3 табл, 99 формул, 2 прил.,  
32 источника

ВРЕМЕННОЙ РЯД, КАСКАДНАЯ НЕЙРОННАЯ СЕТЬ, НЕЙРОННАЯ СЕТЬ, НЕЙРО-ФАЗЗИ СИСТЕМА, НЕО-ФАЗЗИ НЕЙРОН, СИСТЕМА НЕЧЕТКОГО ВЫВОДА.

Объектом исследования является процесс прогнозирования временных рядов в условиях послидвного поступления данных на обработку.

Предметом исследования является каскадная нечеткая система, эволюционирует, предназначенная для решения задач динамического интеллектуального анализа данных.

Целью исследования является разработка каскадной нечеткой системы и метода ее обучение с повышенным быстродействием и возможностями интерпретовности выходного сигнала, а также параметрического и структурного настройки в режиме последовательного обработки информации.

Методы исследования. Теория эволюционных систем вычислительного интеллекта, включая различные гибридные системы построены на основе нейронных сетей и нечеткой логики, методы оптимизации второго порядка, использование Гауссовско-Ньютоновских методов оптимизации позволяет получить высокую сходимость методов обучения нейро-фаззи систем, методы EDA анализа позволили провести первичный анализ временных рядов и построить исходную аналитику по построенным прогнозируемым моделям.

Благодаря стремительному развитию информационных технологий можно предположить, что разработка каскадной нейро-фаззи архитектуры, а также метода ее настройки и обучения, позволит преодолеть ограничения существующих нейро-фаззи сетей и решать задачи динамического анализа данных является актуальной и необходимой.

## ABSTRACT

Explanatory note: 84 p., 16 fig., 3 tabl., 99 formulas, 2 ann., 32 sources

CASCADE NEURAL NETWORK, FUZZY CONCLUSION SYSTEM, NEURAL NETWORK, NEURO-FUZZY SYSTEM, NEO-FUZZY NEURON, TIME SERIES.

The object of research is the process of forecasting time series under the conditions of consistent receipt of data for processing.

The subject of research is a cascading fuzzy evolving system designed to solve problems of dynamic data mining.

The aim of the research is to develop a cascading fuzzy system and a method of its training with increased speed and interpretability of the output signal, as well as parametric and structural settings in the mode of sequential information processing.

Research methods. The theory of evolutionary systems of computational intelligence, including various hybrid systems based on neural networks and fuzzy logic, second-order optimization methods, the use of Gaussian-Newtonian optimization methods allows to obtain high convergence of teaching methods of neuro-phase systems, EDA analysis methods allowed to conduct primary analysis series and build the original analytics on the constructed forecasting models.

Due to the rapid development of information technology, it can be assumed that the development of cascading neuro-phase architecture, as well as the method of its configuration and training, which will overcome the limitations of existing neuro-phase networks and solve dynamic data analysis is relevant and necessary.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів..	9
Вступ.....	10
1 Мета та основні задачі дослідження.....	12
2 Огляд предметної галузі та постановка задачі дослідження.....	14
2.1 Навчання та самонавчання штучних нейронних мереж.....	14
2.2 Лінійні алгоритми навчання.....	15
2.3. Нечітка інформація та висновування.....	22
2.4. Гібридні системи обчислювального інтелекту та питання структурної адаптації.....	24
2.5. Постановка завдання дослідження.....	36
3 Гібридна каскадна нейро-фаззі мережа з оптимізацією пулу нейронів.....	38
3.1 Каскадно-кореляційні нейронні мережі.....	39
3.2 Архітектура оптимізованої каскадної нейронної мережі.....	42
3.3 Навчання елементарних персептронів Розенблатта у каскадній оптимізованій системі.....	43
3.4 Навчання нео-фаззі нейронів у оптимізованій каскадній нейронній мережі.....	48
3.5 Розширені нео-фаззі нейрони в якості елементів гібридної каскадної мережі, що еволюціонує.....	55
3.6 Оптимізація пулу нео-фаззі нейронів.....	58
4 Імітаційне моделювання розробленої каскадної нейро-фаззі мережі	62
4.1 Фільтрація, згладжування, прогнозування.....	64
4.2 Моделювання гібридної каскадної нейро-фаззі мережі з розширеними нео-фаззі нейронами та оптимізацією пулу нейронів.....	66
Висновки.....	76

Перелік джерел посилання.....	77
Додаток А Вихідний код програми.....	79
Додаток Б Відомість кваліфікаційної роботи.....	84

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

ШНМ – штучна нейрона мережа;

EDA – exploratory data analysis – розвідувальний аналіз даних;

ENS – extended nonlinear synapse – розширений нелінійний синапс;

ENFN – extended neo-fuzzy neuron – розширений нео-фаззі нейрон;

NFN – neo-fuzzy neuron – нео-фаззі нейрон;

OBD – optimal brain damage – оптимальна «мозкова небезпека»;

RMSE – root mean squared error – середньоквадратична похибка;

RAN – resource-allocating network – мережа динамічного регулювання  
забування;

SMAPE – symmetric mean absolute percentage error – симетрична  
абсолютна процентна похибка.

## ВСТУП

Кінець ХХ століття характеризується помітним сплеском досліджень в області штучних нейронних мереж завдяки тому, що, з одного боку, у другій половині 1980-их років був відкритий алгоритм зворотного поширення похибки, внаслідок чого вдалося подолати критичні зауваження Мінського і Пайперта, а з іншого – через те, що з року у рік справджувався закон Муру, дозволяючи персональним комп'ютерам проводити дедалі складніші обчислення [1]. У 1990-ті роки теорія штучних нейронних мереж стрімко розвивається, а отримані результати успішно застосовуються для вирішення широкого кола завдань ідентифікації, прогнозування, управління, кластерування та класифікування. Однак, в той же час, стають чітко зрозумілими недоліки традиційних нейромережових архітектур: велика обчислювальна складність, абсолютна неінтерпретовність результатів, емпіричний характер вибору архітектури мережі для вирішення будь-якої задачі. У зв'язку з цим застосування нейронних мереж у певному ряді випадків є недоцільним. З середини 1990-их років у світі активно проводяться дослідження з розробки методів, що дозволяють подолати зазначені недоліки. Останнім часом все більшої популярності набувають так звані гібридні нейро-фаззі мережі, що об'єднують в собі переваги нейромережового підходу і систем нечіткого висновування.

Можливості сучасної обчислювальної техніки дозволяють обробляти великі масиви інформації в послідовному режимі. Часто в таких випадках відсутня інформація про функціональні залежності всередині даних, про зміну цих залежностей і про обсяг всієї вибірки. Такі умови прийнято називати невизначеністю. За послідовної обробки гостро постає проблема підвищення швидкості опрацювання даних з мінімальною похибкою. У таких випадках доцільно використання штучних нейронних мереж, які володіють універсальними апроксимуючими властивостями і можливостями навчання.

Основним недоліком нейронних мереж є відсутність інформації про те, як налаштування параметрів самої мережі і алгоритмів її навчання впливають на рішення, так звана, проблема «темної скрині». У цьому випадку системи, побудовані шляхом поєднання нейромережевого підходу і нечіткої логіки, звані нейро-фаззі системами, дозволяють істотно розширити функціональні можливості і коло вирішуваних завдань. Опис процедур за допомогою нечітких правил забезпечує розуміння роботи нейронної мережі, що, у свою чергу, допомагає створювати більш ефективні програми.

Однією з важливих характеристик нечітких систем є можливість отримання нечіткої оцінки належності на виході, що на практиці часто дозволяє уникнути прийняття неправильного рішення, наприклад, у випадках, коли дані містять перетинні класи. Штучні нейронні мережі і нейро-фаззі системи, а також ефективні алгоритми їх навчання і самонавчання, дозволяють істотно прискорити і спростити цей процес.

## 1 МЕТА ТА ОСНОВНІ ЗАДАЧІ ДОСЛІДЖЕННЯ

З точки зору обчислювального інтелекту під гібридними нейро-фаззі мережами розуміють штучні нейронні мережі з можливістю у той чи інший спосіб отримувати знання про те, за якими правилами проводиться генерація вихідного сигналу.

Традиційні гібридні нейро-фаззі мережі є потужним інструментом для вирішення проблеми неінтерпретовності результатів, однак, слід зазначити, що вони не здатні працювати у послідовному режимі опрацювання даних, а крім того часто є адаптивними лише в тому випадку, коли можна налаштовувати їх синаптичні вагові коефіцієнти в процесі навчання, не маючи при цьому механізмів структурної адаптації. Тому на сьогоднішній день актуальною є розробка каскадної нейро-фаззі архітектури, а також методу її налаштування і навчання, що дозволить подолати обмеження існуючих нейро-фаззі мереж та вирішувати задачі динамічного аналізу даних.

Метою дослідження є розробка каскадної нечіткої системи і методу її навчання з підвищеною швидкістю і можливостями інтерпретовності вихідного сигналу, а також параметричного та структурного налаштування в режимі послідовного опрацювання інформації.

Для досягнення поставленої мети необхідно розв'язати такі наукові задачі:

- проаналізувати відомі архітектури нечітких систем та методи їх навчання;
- проаналізувати існуючі підходи до створення каскадних мереж;
- зсинтезувати архітектуру і методи навчання каскадної нейро-фаззі мережі, що еволюціонує, для опрацювання даних у послідовному режимі;
- виконати імітаційне моделювання розробленої каскадної нечіткої системи та розв'язати за її допомогою завдання прогнозування часових рядів.

Об'єктом дослідження є процес прогнозування часових рядів за умов послідовного надходження даних на опрацювання.

Предметом дослідження є каскадна нечітка система, що еволюціонує, призначена для вирішення задач динамічного інтелектуального аналізу даних.

Методи дослідження. Теорія еволюційних систем обчислювального інтелекту, що включає різні гібридні системи побудовані на основі нейронних мереж та нечіткої логіки, методи оптимізації другого порядку, використання Гавсівсько-Ньютоновських методів оптимізації дозволяє отримати високу збіжність методів навчання нейро-фаззі систем, методи EDA-аналізу (Exploratory data analysis) дозволили провести первинний аналіз часових рядів та побудувати вихідну аналітику по побудованим прогнозуючим моделям.

## 2 ОГЛЯД ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

### 2.1 Навчання та самонавчання штучних нейронних мереж

Здатність навчатися є основною властивістю біологічного мозку, а штучна нейронна мережа в деякому сенсі моделює мозок, поняття «навчання» посідає щонайперше місце в теорії штучних нейронних мереж [2]. Математичні проблеми, що пов'язані з навчанням, вивчають у напрямі загальної теорії штучних нейронних мереж, який дістав назву «нейроматематика» [3]. Із точки зору нейроматематики, навчання тлумачать як завдання адаптувати параметри, а можливо, й архітектуру мережі, щоби, оптимізуючи прийнятий критерій якості, розв'язати поставлену задачу. Таке визначення є узвичаєним та неявно припускає, що нейроматематика ґрунтується на методах оптимізації та ідентифікації [4].

Зазвичай припускають, що навчання має перманентний характер та з часом мережа покращує свої характеристики, постійно «наближаючись» до оптимального розв'язку поставленої задачі. Тип та характер навчання обумовлені, насамперед, обсягом попередньої та поточної інформації про довкілля, в яке «занурили» мережу, а також критерієм якості (цільовою функцією), що характеризує рівень відповідності нейронної мережі до розв'язуваної нею задачі. Інформацію про довкілля здебільшого задають у вигляді навчальної вибірки образів або зразків, оброблюючи їх мережа дістає відомості, необхідні для отримання шуканого розв'язку. Саме характер та обсяг цієї інформації визначають тип і метод навчання [2], [3], [4].

З погляду математики, навчання нейронних мереж – це багато-параметрична задача нелінійної оптимізації. Більшість методів навчання можна розділити на два класи: навчання з учителем (із заохоченням) та навчання без учителя (без заохочення, або самонавчання). Методи навчання

з учителем застосовують у випадках, коли відома бажана реакція системи в кожному мить часу, себто відомий навчальний сигнал, який впливає на налаштування параметрів системи, що навчається. Рівень «навченості» системи формально визначають за значенням цільової функції, тобто за тим станом, якого має набути система в результаті навчання.

## 2.2 Лінійні алгоритми навчання

В цьому підрозділі будуть наведені алгоритми, основані на парадигмі навчання зі вчителем та які реалізують правило корекції по похибці, при цьому покладається, що сама похибка є лінійною функцією синаптичних ваг. Такі алгоритми мають досить багато додатків, найпростішим з яких є навчання такої відомої нейронної мережі, як АДАЛІНА.

З математичної точки зору процес навчання в цьому випадку зводиться до мінімізації критерію навчання (цільової функції) по налаштованим синаптичним вагам  $w_{ji}$  ( $i = 0, 1, \dots, n$ ) і може протікати як в безперервному  $t$ , так і дискретно  $k = 0, 1, 2, \dots$  часі.

В якості цільової функції найбільш часто приймається квадрат поточного значення помилки навчання, тобто

$$E_j(t) = \frac{1}{2} e_j^2(t) = \frac{1}{2} (d_j(t) - u_j(t))^2 \quad (2.1)$$

або, що по факту є тим самим,

$$E_j(k) = \frac{1}{2} e_j^2(k) = \frac{1}{2} (d_j(k) - u_j(k))^2. \quad (2.2)$$

Градiєнтна оптимізація (формула 2.1) в неперервному часі приводить до системи диференційних рівнянь

$$\frac{dw_{ji}}{dt} = -\eta \frac{\partial E_j(t)}{\partial w_{ji}} = -\eta \frac{\partial E_j(t)}{\partial u_j} \cdot \frac{\partial u_j}{\partial w_{ji}} \quad (2.3)$$

або з урахуванням того, що

$$u_j = \sum_{i=0}^n w_{ji} x_i, \quad (2.4)$$

$$\frac{dw_{ji}}{dt} = \eta e_j(t) x_i(t) = \eta (d_j(t) - \sum_{l=0}^n w_{jl} x_l) x_i, \quad i = 0, 1, \dots, n, \quad (2.5)$$

де  $\eta > 0$  – скалярний параметер, який позначає швидкість навчання.

На практиці найбільшого поширення набули дискретні алгоритми навчання виду

$$w_{ji}(k+1) = w_{ji}(k) + \eta(k) e_j(k) x_i(k), \quad (2.6)$$

або у векторному вигляді

$$w_j(k+1) = w_j(k) - \eta(k) \nabla_{w_j} E_j(k) = w_j(k) + \eta(k) e_j(k) x(k), \quad (2.7)$$

де  $\nabla_{w_j} E_j(k) = -e_j(k) x(k)$  – вектор-градієнт цільової функції по синаптичним вагам.

Швидкість процесу навчання за допомогою алгоритму (формула 2.6) та (формула 2.7) повністю визначається параметрами  $\eta(k)$ , який визначає крок зсуву в просторі параметрів, що настроюються. Природно вибрати цей параметр так, щоб швидкість збіжності поточних значень  $w_j(k)$  до оптиманих гіпотетичним вагам  $w_j$  була максимальною. Вводячи, в розгляд вектор ухилень поточних значень  $w_j(k)$  від оптимальних  $w_j$  у вигляді

$$\tilde{w}_j(k) = w_j - w_j(k) \quad (2.8)$$

та розв'язуючи диференційне рівняння

$$\frac{\partial \|\tilde{w}_j(k)\|^2}{\partial \eta} = 0, \quad (2.9)$$

нескладно отримати значення параметра кроку навчання у вигляді

$$\eta(k) = \|x(k)\|^{-2}, \quad (2.10)$$

що веде до алгоритму навчання

$$w_j(k+1) = w_j(k) + \frac{e_j(k)x(k)}{\|x(k)\|^2}, \quad (2.11)$$

відомому в теорії штучних нейронних мереж як алгоритм Уідрой-Ґоффа.

Не можна не відзначити, що вперше цей алгоритм був запропонований С. Качмажем задовго до появи нейроматематики та використовувався для вирішення завдань адаптивної ідентифікації об'єктів управління в так званій мультиплікативній формі [4]

$$w_j(k+1) = w_j(k) + \frac{\eta e_j(k)x(k)}{\|x(k)\|^2}, \quad 0 < \eta < 2, \quad (2.13)$$

забезпечує при відповідному виборі параметра  $\eta$  стійкість до збурень процесу навчання. Процес налаштування одиничного елементарного нейрона нескладно поширити на деякі види неймереж в цілому. Записавши перетворення в формі

$$y = F(x) = w_0 + \sum_{i=1}^h w_i \varphi_i(x) = w^T \varphi(x), \quad (2.14)$$

де  $w = (w_0, w_1, w_2, \dots, w_h)^T$ ,  $\varphi(x) = (1, \varphi_1(x), \dots, \varphi_h(x))^T$ , приходимо до градієнтного алгоритму навчання таких відомих нейронних мереж як радіально-базисні нейронні мережі.

$$w(k+1) = w(k) + \eta \frac{d(k) - w^T(k) \varphi(x(k))}{\|\varphi(x(k))\|^2} \varphi(x(k)), \quad (2.15)$$

цей алгоритм забезпечує збіжність синаптичних ваг до своїх оптимальних значень для будь-якої послідовності лінійно-незалежних векторів  $\varphi(x(1)), \varphi(x(2)), \dots, \varphi(x(k)), \dots$

Для елементарної мережі з множиною вихідних сигналів  $y_j(k)$  отримуємо

$$\begin{cases} w_j(k+1) = w_j(k) + \eta \frac{d_j(k) - y_j(k)}{\|\varphi(x(k))\|^2} \varphi(x(k)), \\ y_j(k) = w_j^T(k) \varphi(x(k)); \quad j = 1, 2, \dots, m. \end{cases} \quad (2.16)$$

Поряд з квадратичними критеріями якості навчання (формула 2.1), (формула 2.2) набули поширення і інші форми цільових функцій, вибір яких значною мірою визначається апріорною інформацією про характер розподілу вхідних сигналів і діючих перешкод. Так найбільшу стійкість перед перешкодами (робастної) забезпечує використання модульного критерію

$$E_j(k) = |d_j(k) - y_j(k)| = |e_j(k)|, \quad (2.17)$$

що веде до алгоритму навчання

$$w_j(k+1) = w_j(k) - \eta(k) \text{sign}(y_j(k) - w_j^T(k)\varphi(x(k)))\varphi(x(k)). \quad (2.18)$$

Якщо замість звичайної сигнум-функції використовується релійна функція з зоною нечутливості, то процедура (формула 2.18) набуває вигляду [8]

$$w_j(k+1) = w_j(k) - \frac{\eta(k)}{2} (\text{sign}(y_j(k) - w_j^T(k)\varphi(x(k))) + \Delta) + \text{sign}(y_j(k) - w_j^T(k)\varphi(x(k)) - \Delta)\varphi(x(k)). \quad (2.19)$$

Також була запропонована методика побудови алгоритмів навчання, яка складається в тому, що при відомій оцінці  $w_j(k)$  наступне значення  $w_j(k+1)$  знаходиться із умов мінімуму норми  $\left| \sum_{i=0}^h |w_{ji}(k+1) - w_{ji}(k)|^q \right|^{\frac{1}{q}}$  при обмеженнях

$$y_j(k) - w_j^T(k+1)\varphi(x(k)) = 0, \quad (2.20)$$

тобто уточнений вектор синаптичних ваг  $w_j(k+1)$  повертає в нуль апостеріорну похибку навчання. Із цих міркувань можна зробити висновки, що алгоритм Кажмажа-Уїдроу-Гоффа мінімізує евклідову норму ( $q = 2$ ). Мінімум кубичної норми надає алгоритм Нагумо-Ноди [3], [4], [5]

$$w_j(k+1) = w_j(k) + \frac{y_j(k) - w_j^T(k)\varphi(x(k))}{\varphi^T(x(k)) \text{sign}\varphi(x(k))} \text{sign}\varphi(x(k)), \quad (2.21)$$

а алгоритм Некрасова [5]

$$\begin{cases} w_{ji}(k+1) = w_{ji}(k) + \frac{y_j(k) - w_j^T(k)\varphi(x(k))}{\max_{i=0,1,\dots,h} |\varphi_i(x(k))|}, \\ w_{jp}(k+1) = w_{jp}(k) \end{cases} \quad (2.22)$$

мінімізує октаендричну норму.

Заслуговує уваги алгоритм навчання, який є розширенням наступного алгоритму (формула 2.23)

$$w_j(k+1) = w_j(k) + \frac{y_j(k) - w_j^T(k)\varphi(x(k))}{\psi(\varphi^T(x(k)))\varphi(x(k))} \psi(\varphi(x(k))), \quad (2.23)$$

(тут  $\psi(\varphi(x(k))) = (\psi_0(\varphi(x(k))), \psi_1(\varphi(x(k))), \dots, \psi_h(\varphi(x(k))))^T$ ) і тісно пов'язаний з методом інструментальних змінних.

Ціла низка алгоритмів може бути отримана при використанні в якості цільової функції співвідношення

$$E_j(k) = e_j(k)\Delta e_j(k) \leq 0, \quad (2.24)$$

де  $\Delta e_j(k) = e_j(k+1) - e_j(k)$  – перша різниця послідовності похибок навчання.

Нескладно бачити, що всі розглянуті вище алгоритми задовольняють цьому виразу. Крім того, легко можуть бути отримані алгоритми, що використовують нелінійності типу сігнум-функції, наприклад (2.25) та (2.27), та багато інших.

$$w_j(k+1) = w_j(k) + \eta(k)(y_j(k) - w_j^T(k)\varphi(x(k))) \text{sign} \varphi(x(k)), \quad (2.25)$$

$$w_j(k+1) = w_j(k) + \eta(k) \frac{\text{sign}(y_j(k) - w_j^T(k)\varphi(x(k)))}{\|\varphi(x(k))\|^2} \varphi(x(k)) \quad (2.26)$$

Розглянуті процедури відносяться до так званих однокрокових алгоритмів навчання, оскільки при кожному уточненні синаптичних ваг використовується тільки одне останнє значення похибки  $e_j(k)$ .

Застосовуючи алгоритми більш складної структури, що враховують інформацію про передісторію процесу навчання, можна домогтися істотного скорочення часу налаштування і забезпечити можливість стійкої роботи як в умовах перешкод, так і нестационарності зовнішнього середовища.

На практиці найбільш поширеними є алгоритми, які пов'язані з критерієм мінімізації суми квадратів похибки

$$E_j^k = \sum_{p=0}^k \alpha(p) e_j^2(p) = \sum_{p=0}^k \alpha(p) E_j(p) \quad (2.27)$$

та їх модифікації, які визначаються системою вагових коефіцієнтів  $\alpha(p)$ ,  $p = 0, 1, 2, \dots, k$ .

При цьому дуже важливим є той факт, що процедури навчання уніфіцирують форму

$$w_j(k+1) = w_j(k) + \eta(k)(d_j(k) - w_j^T(k)\varphi(x(k)))\varphi(x(k)), \quad (2.28)$$

а різниця між ними визначається тільки коефіцієнтом  $\eta(k)$ , який може бути не тільки скаляром, але і матрицею.

Наприклад, алгоритму Качмажа, як вже було наведено, відповідає  $\eta(k) = \|\varphi(x(k))\|^{-2}$  до умов стохастичної апроксимації –  $\eta(k) = \left(\sum_{p=0}^k \varphi^T(x(p))\varphi(x(p))\right)^{-1}$ , методу найменших квадратів – матриця  $\eta(k) = \left(\sum_{p=0}^k \varphi(x(p))\varphi^T(x(p))\right)^{-1}$ .

### 2.3. Нечітка інформація та висновування

Мабуть найбільш вражаюча властивість людського інтелекту – це здатність приймати правильні рішення за умов неповної та нечіткої інформації. Побудова моделей наближених міркувань людини і використання їх у системах обчислювального інтелекту стрімко стає дедалі актуальнішим завданням. Значний крок у цьому напрямку зробив професор Каліфорнійського університету Берклі Л. Заде. Його робота «Fuzzy Sets» [6], надрукована у 1965 році, поклала основи моделювання інтелектуальної діяльності людини і стала початковим поштовхом до розвитку нової математичної теорії. Він розширив класичне канторовське тлумачення поняття множини, припустивши, що характеристична функція (функція приналежності елемента множині) може приймати будь-які значення в інтервалі  $[0, 1]$ , а не тільки значення 0 або 1. Така множина отримала назву нечіткої, і були и запропоновані узагальнення відомих методів логічного висновування. Таким чином було запропоновано апарат для опису процесів інтелектуальної діяльності, зокрема для нечіткості і невизначеності виразів.

Подальші роботи професора Л. Заде і його послідовників заклали міцний фундамент нової теорії і створили передумови для впровадження методів нечіткого управління до інженерну практики. Сьогодні спектр їх використання дедалі ширшає: від управління процесом відправлення та зупинки поїздів метрополітену, управління вантажними ліфтами і доменної піччю до моделювання роботи пральних машин. При цьому нечіткі системи дозволяють підвищити якість продукції при зменшенні ресурсо- та енерговитрат і забезпечують більш високу стійкість до впливу чинників, що заважають, в порівнянні з традиційними системами автоматичного керування. Іншими словами, нові підходи дозволяють поширити сферу застосування систем автоматизації за межі класичної теорії. З цього приводу цікава думка професору Заде: «Я вважаю, що надлишкове прагнення до точності зводить нанівець теорію управління і теорію систем, позаяк

дослідження в цій області зосереджуються на тих і тільки тих проблемах, які піддаються точному рішенню. В результаті багато класів важливих проблем, в яких дані, цілі та обмеження є занадто складними або погано визначеними для точного математичного аналізу, залишалися і залишаються осторонь з тієї лише причини, що вони не піддаються математичному трактуванню. Задля розв'язування таких проблем нам варто поступитися вимогами щодо точності і допустити результати, що є дещо розмитими або невизначеними».

Математична теорія нечітких множин дозволяє описувати нечіткі поняття і знання, оперувати цими знаннями і робити нечіткі висновки. Нечітке управління виявляється особливо корисним, коли технологічні процеси є занадто складними для аналізу за допомогою загальноприйнятих кількісних методів або коли доступні джерела інформації інтерпретуються неточно або невизначено. Нечітка логіка, на якій засновано нечітке управління, ближче за духом до людського мислення і природним мовам, ніж традиційні логічні системи. Нечітка логіка в основному забезпечує ефективні засоби відображення невизначеностей і неточностей реального світу. Тому апарат нечітких множин і нечіткої логіки з успіхом застосовується для вирішення завдань, де вихідні дані є ненадійними і слабо формалізованими. Проте варто звернути увагу й на недоліки цього підходу: початковий набір нечітких правил формується експертом і може виявитися неповним чи суперечливим, вид і параметри функцій приналежності, що описують вхідні і вихідні змінні системи, вибираються суб'єктивно і можуть не цілком відображати дійсність. Для усунення зазначених недоліків були запропоновані нечіткі адаптивні системи, що корегують і правила і параметри функцій належності. Серед кількох варіантів такої адаптації особливе місце, на нашу думку, посідає метод так званих гібридних нейронних мереж.

## 2.4. Гібридні системи обчислювального інтелекту та питання структурної адаптації

В обчислювальному інтелекті вельми поширеним є підхід створювати систем обробляння даних на основі кількох наукових напрямів. Як засвідчують теоретичні та практичні результати, таким гібридним системам властивий синергетичний ефект, тобто вони виявляють такі властивості, яких не мають системи, що їх утворюють .

Для розв'язання задач, які пов'язані з інтелектуальною обробкою даних в умовах апріорної та поточної невизначеності дослідники часто не обмежуються використанням якогось одного підходу (нейронні мережі, нечітка логіка, генетичні алгоритми тощо), а зв'язують групу методів в одну гібридну систему . Такі системи відповідають усім вимогам до інтелектуальних систем і отримали назву гібридні системи обчислювального інтелекту. Нейро-фаззі системи та м'які обчислювання є напрямками дисципліни обчислювального інтелекту, які й займаються проблемами таких систем.

Серед основних характеристик систем, що розробляються в рамках нейро-фаззі систем та м'яких обчислень, можна виділити такі:

- обчислювальні моделі, що засновані на біологічних прототипах;
- паралельна обробка даних у послідовному режимі;
- в основі системи лежать експертні знання;
- стійкість системи до зашумленості.

Варто зазначити, що однією із головних умов до такого типу систем, є їх орієнтованість на розв'язання практичних завдань, що означає здатність оброблювати великі масиви даних великої розмірності, які можуть мати пропущені та зашумлені значення. Однак навчання таких систем зводиться до налаштування синаптичних коефіцієнтів та/або адаптації бази нечітких правил. Тобто архітектура такої системи не зазнає жодних змін, що може в деяких випадках призвести до погіршення точності результатів. В зв'язку з

цим видається очевидно корисним зсинтезувати таку гібридну нейро-фаззі архітектуру та такі алгоритми її навчання, що здатні змінювати не тільки параметри системи, а й її архітектуру.

Як вже зазначалось, основною метою навчання є отримання нейронної мережі, яка здатна у найкращий спосіб відтворювати попередньо невідоме відображення  $R^n \rightarrow R^m$ . В якості такого відображення може виступати залежність вихідних параметрів процесу від вхідних, прогнозування від передісторії, класу об'єкту від набору його властивостей, управляючої дії від поточного стану об'єкта тощо. Коректне налаштування не тільки синаптичних коефіцієнтів, а й архітектури нейронної мережі, зокрема налаштування кількості шарів та кількості нейронів у кожному шарі, дозволяю суттєво покращити показники такої мережі. Серед підходів до налаштування архітектури нейронної мережі виділяють

– деструктивний підхід: за основу береться заздалегідь надлишкова модель, до неї застосовуються різні процедури, що видаляють із початкової мережі елементи, які надають негативний або незначний позитивний ефект на кінцевий результат;

– конструктивний підхід: за основу береться максимально проста модель (складається із одного або декількох нейронів), до неї застосовуються процедури, що додають початковій мережі нові елементи до певного моменту, в залежності від методу, що використовується. Як варіант, конструктивний алгоритм може стартувати з цілком нульової архітектури та самостійно генерувати шари мережі в процесі своєї роботи.

Деструктивний підхід до налаштування архітектури нейронної мережі. Основна ідея деструктивних алгоритмів полягає у видаленні параметрів, що мають найменший вплив на вихідний сигнал мережі. У ряді публікацій було сформульовано та підтверджено припущення, що використання деструктивних алгоритмів нерідко призводить до покращення узагальнюючих властивостей мережі, допомагає нейтралізувати появу так званого ефекту перенавчання, а, крім того, після закінчення роботи такого

алгоритму, архітектура мережі набуває меншого та простішого вигляду, що, вочевидь, позитивно відбивається на її обчислювальній складності.

У процесі функціонування деструктивних алгоритмів із мережі можуть бути цілком видалені як деякі вхідні параметри або вузли у прихованих шарах, так і лише деякі синаптичні зв'язки між нейронами, що мають лише один параметр – ваговий коефіцієнт. В основі деструктивного підходу до структурної адаптації нейронної мережі має бути закладено обчислення певної міри значущості, яка буде характеризувати ступінь впливу кожного конкретного параметру на вихідний сигнал.

В одній із перших публікацій, що розглядає цю проблему, автори запропонували деструктивний алгоритм структурної оптимізації під назвою Optimal Brain Damage (OBD), який складається із наступних етапів:

1. обрати архітектуру нейронної мережі;
2. використовуючи один із методів мінімізації цільової функції якості, провести навчання мережі;
3. для кожного елемента мережі обчислити міру значущості за формулою (2.28):

$$s_q = h_q u_q^2 / 2, \quad (2.28)$$

де  $u_q$  – вихідний сигнал  $q$ -го елемента мережі;

$h_q$  обчислюється за формулою (2.29):

$$h_q = \sum_{(i,j) \in V_q} \frac{\partial^2 E}{\partial w_{ij}^2}, \quad (2.19)$$

де  $V_q$  – множина пар коефіцієнтів  $i$  та  $j$  для  $q$ -го елемента мережі;

$E$  – помилка на виході нейронної мережі;

$w_{ij}$  – синаптичний ваговий коефіцієнт в  $j$ -му шарі.

4. видалити із мережі деяку кількість елементів, для яких міра значущості  $s_q$  найменша. У цьому контексті під видаленням елемента мається на увазі зміна вихідного значення елемента на 0 та замороження його в такому стані;

5. повернутися до кроку номер 2 та повторити процедуру.

Варто зазначити, що при використанні такого методу значно збільшується обчислювальна складність методу навчання і через необхідність розрахувати міру значущості для кожного нейрону, і через додаткові ітерації перенавчання, які необхідно виконати після видалення кожного нейрона із мережі. Також суттєвим недоліком є те, що разом з видаленням нейрона ми видаляємо відразу декілька синаптичних зв'язків, хоча деякі із них можуть бути корисними. У [8] було запропоновано спосіб обійти цей недолік, а разом із тим збільшити швидкість процесу навчання. Цей підхід отримав назву Optimal Brain Surgeon (OBS), і складається він із наступних етапів:

1. обрати мережу із достатньо надлишковою архітектурою та провести її навчання;

2. обчислити  $H^{-1}$  – матрицю, зворотною до гессіану:  $H = \frac{\partial^2 E}{\partial w^2}$ ;

3. обчислити міру значущості для кожного елемента:

$$L_q = w_q^2 / (2[H^{-1}]_{qq}), \quad (2.20)$$

де  $w_q$  –  $q$ -ий ваговий коефіцієнт в мережі;

4. якщо мінімальний  $s_q$  значно менший за поточну помилку, то  $w_q$  має бути видалений, після чого перейти до кроку номер 5, в іншому разі перейти до кроку номер 6;

5. оновити вектор вагових коефіцієнтів мережі, використовуючи наступний вираз:

$$\delta W = -\frac{w_q}{[H^{-1}]_{qq}} H^{-1} \zeta_q, \quad (2.31)$$

де  $\zeta_q$  – орт-вектор у площині вагових коефіцієнтів мережі, який відповідає  $q$ -й синаптичній вазі;

б. усі незначущі вагові коефіцієнти видаляються, після чого бажано перенавчити мережу.

Слід зауважити, що перший же крок цього методу, а саме вибір критерію надмірності архітектури, може викликати багато запитань. В цілому ж, запропонований в метод також характеризується значною обчислювальною складністю, хоч і меншою, в порівнянні з OBD, оскільки він не потребує перенавчання всієї мережі після видалення кожного вагового коефіцієнта. Але методам, що описані вище, притаманний ще один істотний недолік – необхідність мати вже навчену нейронну мережу до початку роботи алгоритму [8]. Це обмеження вдалося обійти завдяки методу, що був запропонований у, де міра значущості для вагових коефіцієнтів визначається через тестову статистику  $T$ , спираючись на те, що ваговий коефіцієнт обнуляється у процесі навчання мережі:

$$T_{(wq)} = \log \left( \frac{\left| \sum_{k=1}^N \left( w_q - \eta \left( \frac{\partial E}{\partial w_q} \right)_k \right) \right|}{\eta \sqrt{\sum_{k=1}^N \left( \left( \frac{\partial E}{\partial w_q} \right)_k - \left( \frac{\partial E}{\partial w_q} \right) \right)^2}} \right), \quad (2.32)$$

де  $K$  – кількість прикладів у виборці.

У рамках деструктивного підходу відомі також і багато інших методів оптимізації архітектури нейронної мережі, проте в силу специфіки цього підходу всім їм в тій чи іншій мірі властива додаткова обчислювальна складність, а також орієнтація на нейронні мережі з архітектурою типу

багатошарового персептрон. Використання таких алгоритмів для налаштування інших архітектур, зокрема для нейро-фаззі систем, неможливо, а розробка видається недоцільною, оскільки в будь-якому разі використання деструктивного алгоритму матиме прямий негативний вплив як на час навчання, так і на час функціонування мережі. Для систем, що розробляються в рамках напряму нейро-фаззі і м'яких обчислень, час є досить критичним параметром, оскільки ці системи орієнтуються на рішення практичних задач. У зв'язку з цим вельми привабливо виглядає використання конструктивного підходу для синтезу архітектури нейронної мережі, про що йтиметься далі.

Конструктивний підхід до налаштування архітектури нейронної мережі. Суть конструктивного підходу полягає в нарощуванні архітектури нейронної мережі і налаштування її вагових коефіцієнтів паралельно, доки не будуть задоволені вимоги критерію зупинки. Тобто цей підхід дозволяє не тільки уникнути перенавчання нейронної мережі, а й оптимізує структуру й архітектуру мережі на етапі навчання.

Таким чином, використовуючи конструктивний підхід, можна повністю вирішити питання про вибір початкової архітектури мережі: в загальному випадку вона повинна бути максимально простою, складатися з одного або декількох нейронів (залежить від методу). Слід зауважити, що, як правило, в результаті роботи конкретного конструктивного алгоритму на виході отримуємо нейронну мережу нетрадиційної архітектури.

У Джон Платт описує нейронну мережу, що дістала назву Resource-Allocating Network (RAN), яка в процесі навчання додає в свою архітектуру нові обчислювальні елементи (штучні нейрони) кожен раз, коли до входу мережі подається новий навчальний приклад, який в RAN має двошарову архітектуру [9], [10]. Перший шар складається з нейронів, що відповідають за локальну область з простору вхідних сигналів. У випадку, коли вхідний сигнал віддаляється від області конкретного нейрона, то значення сигналу на його виході зменшуватиметься відповідно співвідношенню:

$$\begin{cases} x_j = \begin{cases} \left(1 - \left(\frac{z_j}{\chi w_j^2}\right)\right)^2, & \text{якщо } z_j < \chi w_j^2 \\ 0 & \text{у протилежному випадку} \end{cases} \\ z_j = \sum_j (c_{ij} - X_i)^2, \end{cases} \quad (2.33)$$

де  $X_i$  –  $i$ -ий вхід нейронної мережі;

$c_{ij}$  –  $i$ -ий ваговий коефіцієнт  $j$ -го нейрона першого шару;

$\chi$  – параметр налаштування, який підбирається емпіричним шляхом.

Виходи першого шару  $x_j$  надходять до другого шару, який агрегує ці значення і генерує вихідний сигнал. Метою кожного синапсу другого шару є визначити, який вплив надає кожен нейрон першого шару на формування конкретного цільового вектору  $\vec{y}$ . Вихідним сигналом мережі  $\vec{y}$  є зважена сума виходів першого шару плюс незалежний вектор  $\vec{\gamma}$ , що містить постійні елементи:

$$\vec{y} = \sum_j \vec{w}^{[o]} x_j + \vec{\gamma}, \quad (2.34)$$

де  $\vec{w}^{[o]}$  – вектор синаптичних вагових коефіцієнтів вихідного шару, або в скалярній формі (2.35):

$$y_i = \sum_j w_{ji}^{[o]} x_j + \gamma. \quad (2.35)$$

Також  $\vec{y}$  є виходом нейронної мережі у випадку, якщо не активувався жоден з нейронів першого шару. У певному сенсі вираз  $\vec{w}_j^{[o]} x_j$  може

розглядатися як адитивний елемент, який може бути використаний для того, щоб отримати бажаний вихідний сигнал.

Навчання RAN починається з нульового стану, тобто в першому шарі не міститься жодного нейрона, а в другому кількість нейронів дорівнює розмірності завдання, проте, на цьому етапі всі вони не мають вхідних параметрів (за винятком зсуву  $\gamma$ ). Після подання на вхід першого навчального прикладу у вхідній шар додається перший нейрон, центр функції активації (2.33) якого встановлено наступним чином:

$$\vec{c}_i = \vec{X}. \quad (2.36)$$

Вихідний сигнал першого шару автоматично передається до всіх нейронів другого шару, а його лінійні синапси налаштовуються таким чином, щоб різниця між виходом мережі і навчальним сигналом була мінімальна:

$$w_j^{[0]} = \vec{Y} - \vec{y}, \quad (2.37)$$

де  $\vec{Y}$  – бажаний вихідний сигнал мережі.

Нейрон, що його було додано до мережі, реагуватиме на нові вхідні сигнали, якщо вони належать до певного інтервалу, що визначається відстанню між найближчим вектором і новим вхідним образом

$$w_i = \omega \|\vec{X} - \vec{c}_{nearest}\|, \quad (2.38)$$

де  $\omega$  – параметр покриття, підібраний емпіричним шляхом.

Щобільше значення цього параметра, то на більшу кількість вхідних сигналів будуть реагувати вже існуючі нейрони першого шару.

У RAN для додавання нового нейрона до перший шару мережі мають бути задоволені одна з двох умов: по-перше, це відбувається в тому випадку,

якщо вхідний сигнал знаходиться далеко від вже існуючих центрів функцій активації нейронів першого шару:

$$\|\vec{X} - \vec{c}_{nearest}\| > \delta(t), \quad (2.39)$$

а також у випадку, коли із допомогою поточного набору елементів не вдається забезпечити необхідну точність вихідного сигналу:

$$\|\vec{Y} - \vec{y}(\vec{X})\| > \varepsilon, \quad (2.40)$$

де  $\varepsilon$  – необхідна точність вихідного сигналу.

У випадку, коли при поданні до входу мережі нового вектору, на виході отримуємо помилку більшу, ніж  $\varepsilon$ , то у вхідний шар мережі додається новий нейрон з центрами активаційних функцій, налаштованими на поточний вхідний образ [10].

Відстань  $\delta(k)$  – динамічний параметр, який змінює своє значення протягом процесу навчання. Для його обчислення використовується наступний вираз:

$$\delta(t) = \max\left(\delta_{max}e^{-t/\tau}, \delta_{min}\right), \quad (2.31)$$

де  $\delta_{max}$ ,  $\delta_{min}$ ,  $\tau$  – параметри, що вибираються емпірично.

Якщо згідно умов (2.39) і (2.40) не потребується додавання нового нейрона у вхідний шар, то проводиться налаштування вагових коефіцієнтів вихідного шару. Для цього можуть використовуватися градієнтні методи мінімізації або ж метод найменших квадратів.

На перших етапах навчання до мережі переважно додаються нові елементи, проте через деякий час цей процес сповільнюється і замість

додавання нових нейронів у вхідному шар відбувається налаштування синаптичних вагових коефіцієнтів вихідного шару. Такий порядок роботи конструктивного алгоритму стає можливим завдяки використанню двох умов додавання нового нейрона (2.39) і (2.40), забезпечує оптимальну складність моделі нейронної мережі поряд з хорошим рівнем узагальнюючих здібностей. У разі використання виключно (2.39) найбільш ймовірно, що це призведе до перенавчання, а в разі – (2.40) можуть бути пропущені деякі нейрони вхідного шару, що вплине на точність вихідного сигналу мережі.

Серед недоліків запропонованого Джоном Платтом методу має сенс назвати досить велику кількість параметрів, що підбираються емпірично, від яких безпосередньо залежить якість роботи RAN.

Надалі підхід до конструктивної організації архітектури нейронної мережі, відомий в англомовній літературі під назвою Resource Allocation, породив безліч різних модифікацій, які спрямовані на оптимізацію швидкості навчання і точності вихідного сигналу при вирішенні певного кола завдань.

В рамках конструктивного підходу можна виділити такий напрямок, як каскадні нейронні мережі, найбільш характерним і ефективним представником яких є каскадно-кореляційний архітектура, запропонована Фальманом і Леб'єром в роботі. Основна особливість мережі цього типу полягає в можливості додавання нових вузлів у процесі навчання. На рисунку 2.1 наведена схема подібної мережі, після додання другого каскаду.

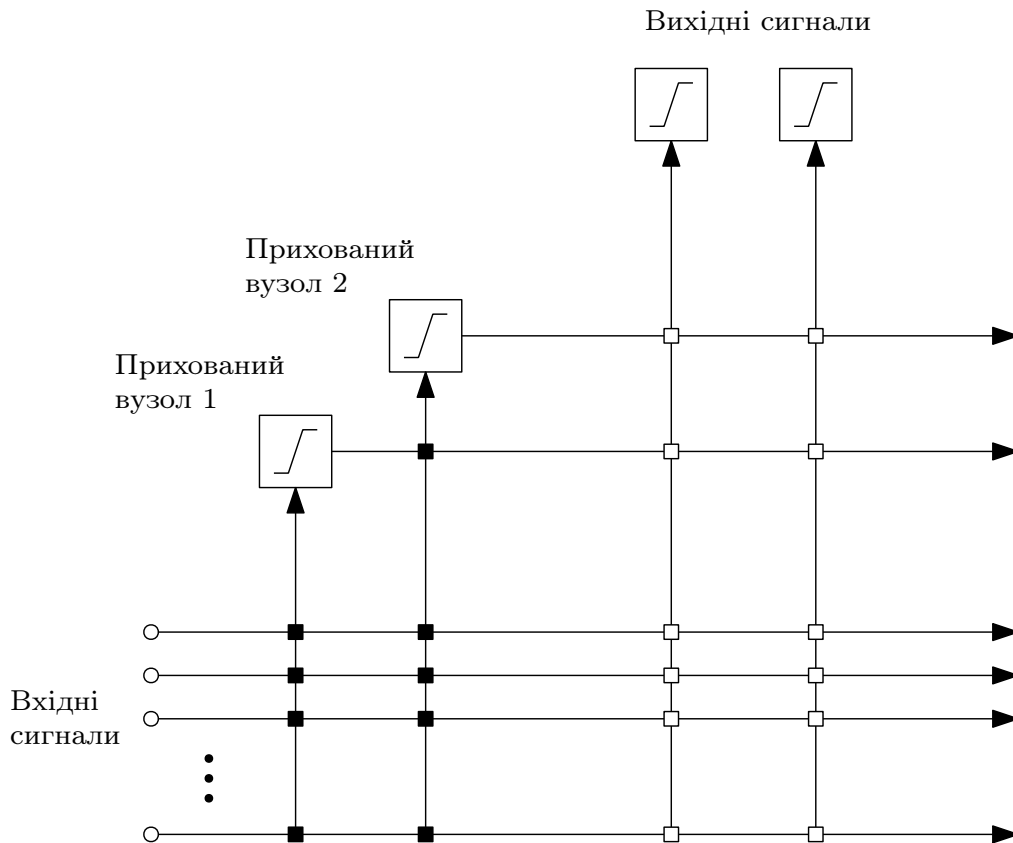


Рисунок 2.1 – Архітектура каскадної системи за Фальманом та Леб'єром після додавання двох вузлів

На початку процесу навчання формується стандартна одношарова структура з декількома входами і єдиним виходом, яка навчається за допомогою того чи іншого нелінійного методу навчання. Після проходження всієї навчальної вибірки  $x(1), x(2), \dots, x(N)$  оцінюється точність апроксимації і в тому випадку, якщо помилка занадто велика, формується каскад з  $n_2$  нейронів-кандидатів, паралельно підключених до входів мережі  $x_1, x_2, \dots, x_n$  і виходу першого каскаду  $o^{[1]}$ . Нейрони-кандидати, як правило, відрізняються один від одного початковими значеннями синаптичних вагових коефіцієнтів  $W^{[2]}(0)$ , видом функцій активації та методами навчання. На наступному етапі проводиться навчання нейронів другого каскаду при «заморожених» синаптичних коефіцієнтах

$W^{[1]}(N)$  першого каскаду. Серед  $n^2$  нейронів-кандидатів обирається один нейрон-переможець з найвищим параметром кореляції

$$R_{[q]}^2 = \left| \sum_{k=1}^N (o_q^{[2]}(k) - \bar{o}_q^{[2]}) (e_q^{[2]}(k) - \bar{e}_q^{[2]}) \right|, q = 1, 2, \dots, n_2, \quad (2.42)$$

Вхідні сигнали, що надходять до вертикальних ліній, сумуються; вагові коефіцієнти, позначені ■, зафіксовані, позначені □, – налаштовуються. Саме цей нейрон з «замороженими» вагами  $W^{[2]}(N)$  утворює другий каскад, в той час як нейрони, які програли, видаляються з мережі.

Далі надається оцінка точності апроксимації, що забезпечується другим каскадом, і в разі потреби формується набір з  $n_3$  кандидатів третього каскаду, серед яких вибирається переможець з максимальним значенням

$$R_{[q]}^3 = \left| \sum_{k=1}^N (o_q^{[3]}(k) - \bar{o}_q^{[3]}) (e_q^{[3]}(k) - \bar{e}_q^{[3]}) \right|, \quad (2.43)$$

де  $q = 1, 2, \dots, n_3$ .

У разі досягнення необхідної точності процес нарощування каскадів завершується і вихідний сигнал останнього каскаду приймається в якості вихідного сигналу мережі в цілому.

В якості основних відмінних рис каскадно-кореляційних мереж слід зазначити такі:

- ці мережі не вимагають попереднього завдання ні архітектури, ні кількості нейронів в каскадах;

- нейрони в мережу додаються по мірі необхідності, утворюючи не приховані шари, а каскади, кожен з яких в якості вхідних сигналів використовує вхідні сигнали мережі та вихідні сигнали попереднього каскаду;

– навчання не пов'язане з концепцією зворотнього поширення похибки, що дозволяє істотно скоротити час налаштування вагових коефіцієнтів;

– за рахунок “заморожування” синаптичних вагових коефіцієнтів каскадів, що були сформовані, скорочуються обчислювальні витрати на навчання.

Головним недоліком даних мереж прийнято вважати неможливість їх навчання в режимі послідовної обробки інформації. Далі буде показано, як можна подолати це обмеження, синтезувавши на основі мережі, запропонованої в архітектуру, яка ладна відповідати критеріям, що висуваються до нейро-фаззі систем.

## 2.5. Постановка завдання дослідження

Оскільки, як зазначалося раніше, сучасні обчислювальні технології дозволяють накопичувати і обробляти досить великі масиви інформації, то на перший план виходить швидкість оброблення даних, а також можливість роботи з ними в послідовному режимі. Крім того варто зазначити, що інформація, яка обробляється, може характеризуватися нелінійним і нестационарним характером даних. У таких випадках доцільно використання штучних нейронних мереж, які володіють універсальними апроксимуючими властивостями.

Застосування апарату нечіткої логіки дозволяє розширити функціональні можливості штучних нейронних мереж і коло вирішуваних завдань. Завдання дослідження полягає в розробці каскадних еволюційних штучних нейро-фаззі мереж і методів їх навчання з підвищеною швидкістю і можливостями інтерпретовності вихідного сигналу, а також параметричного та структурного налаштування в режимі послідовного опрацювання інформації. Для досягнення поставленої мети необхідно розглянути наступні питання:

- проаналізувати відомі архітектури гібридних систем обчислювального інтелекту для опрацювання потоків даних в послідовному режимі надходження спостережень;
- виявити недоліки та переваги існуючих гібридних систем, зокрема каскадних мереж, при вирішенні завдання прогнозування часових рядів;
- розробити архітектуру і метод навчання каскадної нейро-фаззі мережі, що еволюціонує, для опрацювання даних у послідовному режимі;
- провести первинний аналіз часових рядів, які будуть використані для будови моделі синтезованої гібридної архітектури каскадної мережі;
- провести навчання розробленої моделі каскадної нейро-фаззі мережі для прогнозування;
- провести порівняльний аналіз результатів прогнозування синтезованої каскадної нейро-фаззі мережі та існуючих відомих мереж.

### 3 ГІБРИДНА КАСКАДНА НЕЙРО-ФАЗЗИ МЕРЕЖА З ОПТИМІЗАЦІЄЮ ПУЛУ НЕЙРОНІВ

Зазвичай під «навчанням» розуміють процес коригування синаптичних вагових коефіцієнтів за допомогою певної процедури оптимізації, що ґрунтується на пошуку екстремуму заданого критерію навчання. Якість процесу навчання може бути поліпшена шляхом коригування топології мережі поспіль з синаптичними вагами [11]. Ця ідея полягає в основі систем обчислювального інтелекту, що еволюціонують [12].

Мабуть, найбільш відомою реалізацією цього підходу є каскадно-кореляційні нейронні мережі [12], [13], привабливі високою ефективністю та простотою налаштування як синаптичних вагових коефіцієнтів, так і топології мережі. Така мережа на початку містить лише один пул (ансамбль) нейронів, які навчаються незалежно один від іншого (перший каскад). Кожен нейрон у пулі може мати відмінні функції активації та метод навчання. Доки навчання триває, нейрони у пулі не взаємодіють один з одним. Після того, як процес налаштування вагових коефіцієнтів завершився для всіх нейронів пулу першого каскаду, кращий нейрон відповідно до обраного критерію навчання формує перший каскад і коефіцієнти його синаптичних ваг більше не коригуються.

Далі формується другий каскад зазвичай з нейронів, подібних до нейронів першого каскаду. Різниця лише в тому, що нейрони, які навчаються в пулі другого каскаду, мають додатковий вхід ( $i$ , отже, додатковий синаптичний ваговий коефіцієнт) – вихід першого каскаду. Подібно до першого каскаду, у другому каскаді залишається лише один найбільш продуктивний нейрон і його синаптичні вагові коефіцієнти фіксуються.

Аналогічним чином нейрони третього каскаду матимуть два додаткових входи, а саме виходи першого та другого каскадів. Еволюційна

мережа продовжуватиме розширяти свою архітектуру новими каскадами, доки вона не досягне бажаної якості вирішення завдання для заданого набору даних.

Автори найбільш поширеної каскадної нейронної мережі, що еволюціонує, CasCorLA, Фальман та Леб'єр, використовували елементарні перцептрони Розенблатта з традиційними сигмоїдальними функціями активації і коригували синаптичні вагові коефіцієнти за допомогою QuickProp-алгоритму, що є модифікацією  $\delta$ -правила. Оскільки вихідний сигнал таких нейронів нелінійно залежить від синаптичних ваг, швидкість навчання не може бути суттєво збільшена для таких нейронів [14].

Для уникнення багатоепошного навчання доцільно в якості вузлів системи використовувати такі типи нейронів, що їх виходи лінійно залежать від синаптичних ваг, що дозволить використовувати оптимальні за швидкістю методи навчання та обробляти дані в онлайн режимі [15].

Проте варто зазначити, що у випадку послідовного навчання системи, неможливо визначити найкращий нейрон у пулі, адже при оброблянні нестационарних об'єктів певний нейрон може бути кращим для однієї частини тренувальної вибірки, проте поступатися у точності іншому нейрону на іншій частині вибірки. Отже доцільно зберегти усі нейрони пулу та використовувати певну оптимізуючу процедуру (відповідно обраному критерію якості) задля визначення нейрона-переможця на кожному кроці оброблення даних.

### 3.1 Каскадно-кореляційні нейронні мережі

До нейронних мереж з прямою передачею інформації відносяться і так звані каскадно-кореляційні мережі, запропоновані в роботі С. Фальман і К. Леб'єра, основна особливість яких полягає в можливості додавання нових вузлів в процесі навчання [13], [14], [15]. На рисунку 3.1 приведена схема подібної мережі, яка містить три каскаду, входів і один вихід.

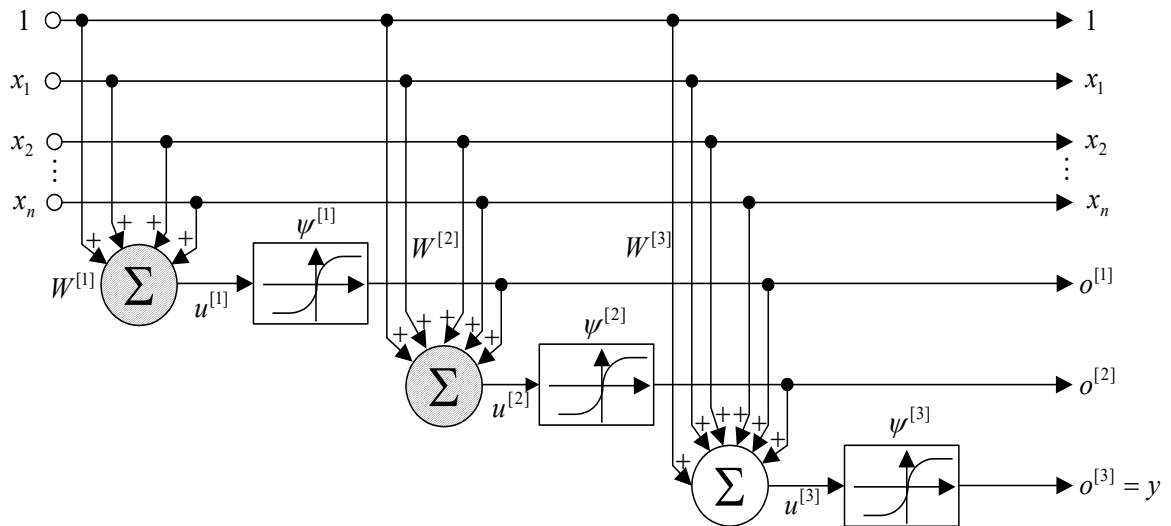


Рисунок 3.1 – Каскадно-кореляційна нейронна мережа

На початку процесу навчання формується стандартна одношарова структура з  $n$  входами і  $m$  виходами (для  $m = 1$  це єдиний нейрон так, як показано на рисунку 3.1), яка навчається за допомогою будь-якого з нелінійних алгоритмів навчання, але в класичній каскадно-кореляційній нейронній мережі використовувався алгоритм «Quickprop».

Після завантаження всієї вибірки до мережі  $x(1), x(2), \dots, x(N)$  оцінюється точність апроксимації та у випадку, якщо похибка дуже велика, формується каскад зі  $n_2$  нейронів-кандидатів, паралельно підключених до входів мережі  $1, x_1, x_2, \dots, x_n$  та виходу першого каскаду  $o^{[2]}$ . Нецроні-кандидати, як правило, відрізняються один від одного початковими значеннями синаптичних ваг  $W^{[2]}(0)$ , виглядом активаційної функції та алгоритмом навчання. На рисунку 3.1 «заморожені» ваги наведені у вигляді заштрихованих саматорів. Посеред  $n_2$  нейронів-кандидатів обирається один нейрон-переможець, в якого параметр кореляції

$$r_q^{[2]} = \left| \sum_{k=1}^N (o_q^{[2]}(k) - \bar{o}_q^{[2]}) (e_q^{[2]}(k) - \bar{e}_q^{[2]}) \right|, \quad q = 1, 2, \dots, n_2 \quad (3.1)$$

(тут  $\bar{o}_q^{[2]}$  и  $\bar{e}_q^{[2]}$  – середнє значення вихідного сигналу та похибки) є максимальним.

Саме цей нейрон з «замороженими» вагами  $W^{[2]}(N)$ , складає другий каскад, в той час як «програвці» нейрони видаляються з мережі. Далі оцінюється точність апроксимації, яка забезпечується другим каскадом, та у випадку необхідності формується система з  $n_3$  кандидатів третього каскаду, серед яких обирається переможець з максимальним значенням

$$r_q^{[3]} = \left| \sum_{k=1}^N (o_q^{[3]}(k) - \bar{o}_q^{[3]}) (e_q^{[3]}(k) - \bar{e}_q^{[3]}) \right|, \quad q = 1, 2, \dots, n_3. \quad (3.2)$$

У разі досягнення необхідної точності процес нарощування каскадів завершується і вихідний сигнал останнього каскаду (на рисунку 3.1. –  $o^{[3]}$ ) приймається в якості вихідного сигналу мережі в цілому.

В якості основних відмінних властивостей каскадно-кореляційні мережі слід відмітити наступне:

- ці мережі не потребують попереднього завдання ні архітектури, ні кількості нейронів в каскаді;
- нейрони в мережу додаються по мірі необхідності, утворюючи не приховані шари, а каскади, кожен з яких в якості вхідних сигналів використовує входи мережі і виходи попереднього каскаду;
- навчання не пов'язане з концепцією зворотного поширення похибок, що дозволяє істотно скоротити час налаштування;
- за рахунок «заморожування» синаптичних ваг сформованих раніше каскадів скорочуються обчислювальні витрати на навчання.

Процедуру навчання каскадно-кореляційних мереж можна істотно спростити, додаючи до попереднього каскаду не групою нейронів-кандидатів, а один єдиний узагальнений формальний нейрон, налаштовуючи не тільки його синаптичні ваги, але і характеристики

активаційної функції. До недоліків даних мереж слід віднести можливість навчання тільки в пакетному режимі при наявності заздалегідь заданій навчальної вибірки.

### 3.2 Архітектура оптимізованої каскадної нейронної мережі

Архітектура запропонованої гібридної системи з оптимізованим пулом нейронів у кожному каскаді наведена на рисунку 3.2.

На вхід такої системи (так званий «рецептивний» шар) подається векторний сигнал

$$x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T, \quad (3.3)$$

де  $k = 1, 2, \dots$  – кількість образів у таблиці «об'єкт – властивість» або поточний дискретний час.

Ці сигнали подаються на входи кожного нейрона в мережі  $N_j^{[m]}$   $j = 1, 2, \dots, q$  – кількість нейронів у тренувальному пулі,  $m = 1, 2, \dots$  – номер каскаду) з вихідним сигналом  $\hat{y}_j^{[m]}(k)$ . Далі вихідні сигнали кожного каскаду надходять до «узагальнюючого» вузлу  $GN^{[m]}$ , який генерує поточно-оптимальний вихідний сигнал відповідного каскаду  $\hat{y}^{*[m]}$ .

Слід зауважити, що вхідними сигналами першого каскаду є вектор  $x(k)$  (що може містити опціональне порогове значення  $x_0(k) \equiv 1$ ), другий каскад має додатковий вхід для сгенерованого першим каскадом вихідного сигналу  $\hat{y}^{*[1]}(k)$ , нейрони третього каскаду оброблятимуть два додаткових сигнали  $\hat{y}^{*[1]}(k)$ ,  $\hat{y}^{*[2]}(k)$ , нейрони  $m$ -ого каскаду матимуть  $(m - 1)$  додаткових вхідних сигналів:  $\hat{y}^{*[1]}(k), \hat{y}^{*[2]}(k), \dots, \hat{y}^{*[m-1]}(k)$ . Під час тренування системи нові каскади додаються доки не буде досягнута бажана точність.

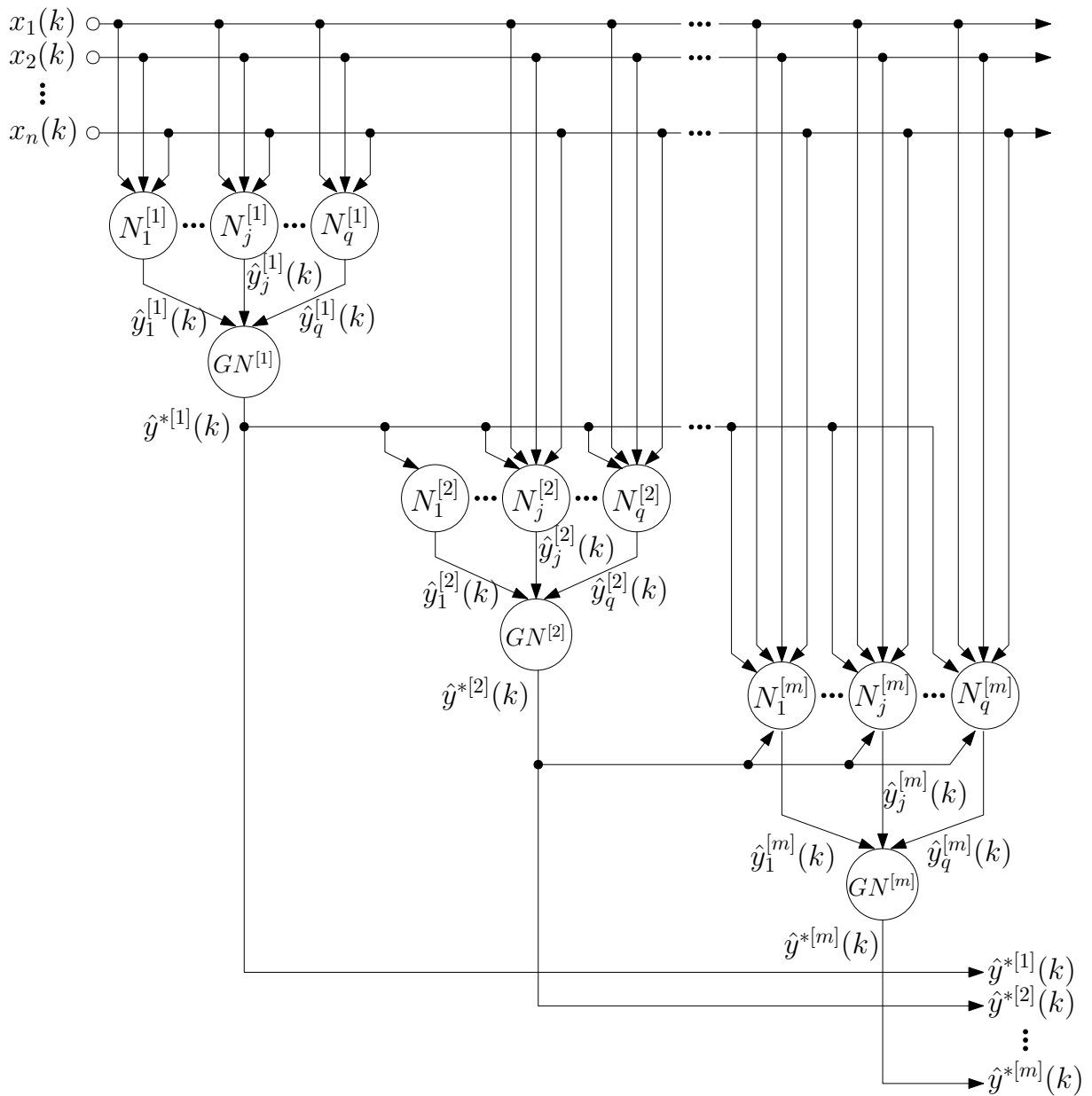


Рисунок 3.2 – Архітектура гібридної системи з оптимізованим пулом нейронів

### 3.3 Навчання елементарних перцептронів Розенблатта у каскадній оптимізованій системі

Наразі вважатимемо  $j$ -й вузол  $m$ -ого каскаду елементарним перцептроном Розенблатта з активаційною функцією

$$0 < \sigma_j^{[m]} \left( \gamma_j^{[m]} u_j^{[m]} \right) = \frac{1}{1 + e^{-\gamma_j^{[m]} u_j^{[m]}}} < 1, \quad (3.4)$$

де  $u_j^{[m]}$  – внутрішній активаційний сигнал  $j$ -ого нейрону  $m$ -ого каскаду;

$\gamma_j^{[m]}$  – параметр посилення.

У такому випадку вихідні сигнали нейронів тренувального пулу першого каскаду матимуть вигляд

$$\hat{y}_j^{[1]} = \sigma_j^{[1]} \left( \gamma_j^{[1]} \sum_{i=0}^n w_{ji}^{[1]} x_i \right) = \sigma_j^{[1]} \left( \gamma_j^{[1]} w_j^{[1]T} x \right), \quad (3.5)$$

де  $w_{ji}^{[1]}$  –  $i$ -й ваговий коефіцієнт  $j$ -ого нейрону першого каскаду.

Вихідні сигнали другого каскаду дорівнюватимуть

$$\hat{y}_j^{[2]} = \sigma_j^{[2]} \left( \gamma_j^{[2]} \left( \sum_{i=0}^n w_{ji}^{[2]} x_i + w_{j,n+1}^{[2]} \hat{y}^{*[1]} \right) \right), \quad (3.6)$$

вихідні сигнали  $m$ -ого каскаду матимуть вигляд

$$\begin{aligned} \hat{y}_j^{[m]} &= \sigma_j^{[m]} \left( \gamma_j^{[m]} \left( \sum_{i=0}^n w_{ji}^{[m]} x_i + w_{j,n+1}^{[m]} \hat{y}^{*[1]} + w_{j,n+2}^{[m]} \hat{y}^{*[2]} + \dots + \right. \right. \\ &\quad \left. \left. + w_{j,n+m-1}^{[m]} \hat{y}^{*[m-1]} \right) \right) = \sigma_j^{[m]} \left( \gamma_j^{[m]} \sum_{i=0}^{n+m-1} w_{ji}^{[m]} x_i^{[m]} \right) = \\ &= \sigma_j^{[m]} \left( w_j^{[m]T} x^{[m]} \right), \end{aligned} \quad (3.7)$$

де  $x^{[m]} = (x^T, \hat{y}^{*[1]}, \dots, \hat{y}^{*[m-1]})^T$ .

Таким чином, нейронна мережа з персептронами Розенблатта у якості вузлів, що містить  $m$  каскадів, залежить від  $(m(n + 2) + \sum_{p=1}^{m-1} p)$  параметрів, у тому числі від параметрів посилення  $\gamma_j^{[p]}$ ,  $p = 1, 2, \dots, m$ .

У якості критерію навчання можна використовувати загальноприйнятту квадратичну функцію, хоча її використання обумовлено виключно тим, що похідні від квадратичної форми – лінійні, тим самим дозволяють використовувати лінійні методи. В умовах забруднених даних в якості критерію навчання необхідно використовувати робастні конструкції або валідаційний (кросс-валідаційний) критерій навчання [16].

$$\begin{aligned}
 E_j^{[m]} &= \frac{1}{2} \left( e_j^{[m]}(k) \right)^2 = \\
 &= \frac{1}{2} \left( y(k) - \hat{y}_j^{[m]}(k) \right)^2 = \\
 &= \frac{1}{2} \left( y(k) - \sigma_j^{[m]} \left( \gamma_j^{[m]} w_j^{[m]T} x^{[m]}(k) \right) \right)^2,
 \end{aligned} \tag{3.8}$$

де  $y(k)$  – бажане значення вихідного сигналу.

Градiєнтну оптимізацію критерію (3.8) відносно  $\gamma_j^{[m]}$  можна записати у вигляді

$$\begin{aligned}
 w_j^{[m]}(k + 1) &= w_j^{[m]}(k) + \eta_j^{[m]}(k + 1) e_j^{[m]}(k + 1) \gamma_j^{[m]} \hat{y}_j^{[m]}(k + 1) \times \\
 &\times \left( 1 - \hat{y}_j^{[m]}(k + 1) \right) x^{[m]}(k + 1) = \\
 &= w_j^{[m]}(k) + \eta_j^{[m]}(k + 1) e_j^{[m]}(k + 1) \gamma_j^{[m]} J_j^{[m]}(k + 1),
 \end{aligned} \tag{3.9}$$

де  $\eta_j^{[m]}(k + 1)$  – параметр кроку навчання.

Мінімізувати критерій (3.8) відносно  $\gamma_j^{[m]}$  можна за допомогою алгоритму Крушке-Мовеланна [17]

$$\begin{aligned} \gamma_j^{[m]}(k+1) &= \gamma_j^{[m]}(k) + \eta_j^{[m]}(k+1)e_j^{[m]}(k+1)\hat{y}_j^{[m]}(k+1) \times \\ &\times \left(1 - \hat{y}_j^{[m]}(k+1)\right) u_j^{[m]}(k+1). \end{aligned} \quad (3.10)$$

Поєднуючи (3.8) та (3.9), отримаємо алгоритм навчання для  $j$ -ого нейрону  $m$ -ого каскаду

$$\begin{aligned} \begin{pmatrix} w_j^{[m]}(k+1) \\ \dots \\ \gamma_j^{[m]}(k+1) \end{pmatrix} &= \begin{pmatrix} w_j^{[m]}(k) \\ \dots \\ \gamma_j^{[m]}(k) \end{pmatrix} + \eta_j^{[m]} \\ &\times (k+1)e_j^{[m]}(k+1)\hat{y}_j^{[m]}(k+1) \times \\ &\times \left(1 - \hat{y}_j^{[m]}(k+1)\right) \begin{pmatrix} \gamma_j^{[m]}x^{[m]}(k+1) \\ \dots \\ u_j^{[m]}(k+1) \end{pmatrix}, \end{aligned} \quad (3.11)$$

або, вводячи нові змінні, у більш компактній формі (формула 3.12)

$$\begin{aligned} \tilde{w}_j^{[m]}(k+1) &= \tilde{w}_j^{[m]}(k) \\ &+ \eta_j^{[m]}(k+1)e_j^{[m]}(k+1)\hat{y}_j^{[m]}(k+1)\tilde{x}^{[m]}(k+1) = \\ &= \tilde{w}_j^{[m]}(k) + \eta_j^{[m]}(k+1)e_j^{[m]}(k+1)\tilde{f}_j^{[m]}(k+1). \end{aligned} \quad (3.12)$$

Використовуючи регуляризуючий параметр (momentum term), можна удосконалити процес корегування синаптичних вагових коефіцієнтів під час навчання, але з точки зору розробки цієї системи, додавання цього параметру ускладнить цей процес [18], [19].

Тоді, замість критерію (3.8) слід використовувати функцію (формула 3.13)

$$E_j^{[m]}(k) = \frac{\eta}{2} \left( e_j^{[m]}(k) \right)^2 + \frac{1-\eta}{2} \left\| \tilde{w}_j^{[m]}(k) - \tilde{w}_j^{[m]}(k-1) \right\|^2, \quad (3.13)$$

$$0 < \eta < 1.$$

Тоді метод навчання приймає вигляд (формула 3.14)

$$\tilde{w}_j^{[m]}(k+1) = \tilde{w}_j^{[m]}(k) + \eta_j^{[m]}(k+1) \left( \eta e_j^{[m]}(k+1) \tilde{f}_j^{[m]}(k+1) + \right. \\ \left. (1-\eta) \left( \tilde{w}_j^{[m]}(k) - \tilde{w}_j^{[m]}(k+1) \right) \right), \quad (3.14)$$

що є модифікацією процедури Сільви-Альмейди [19].

Доцільно вдосконалити алгоритм, використовуючи підхід, запропонований у [19], тоді алгоритм (3.14) набуває слідкуючих та фільтруючих властивостей. Таким чином, кінцева модифікація методу набуває вигляду (формула 3.15)

$$\left\{ \begin{array}{l} \tilde{w}_j^{[m]}(k+1) = \tilde{w}_j^{[m]}(k) + \frac{\eta e_j^{[m]}(k+1) \tilde{f}_j^{[m]}(k+1)}{r_j^{[m]}(k+1)} + \\ \quad + \frac{(1-\eta) \left( \tilde{w}_j^{[m]}(k) - \tilde{w}_j^{[m]}(k-1) \right)}{r_j^{[m]}(k+1)}, \\ r_j^{[m]}(k+1) = r_j^{[m]}(k) + \left\| \tilde{f}_j^{[m]}(k+1) \right\|^2 - \left\| \tilde{f}_j^{[m]}(k-s) \right\|^2, \end{array} \right. \quad (3.15)$$

де  $s$  – розмір ковзного вікна.

Цікаво, що при  $s = 1$  та  $\eta = 1$  отримуємо нелінійну версію загально-відомого алгоритму Качмажа-Уїдрон-Хоффа [20]:

$$\tilde{w}_j^{[m]}(k+1) = \tilde{w}_j^{[m]}(k) + \frac{e_j^{[m]}(k+1)\tilde{f}_j^{[m]}(k+1)}{\|\tilde{f}_j^{[m]}(k+1)\|^2}, \quad (3.16)$$

який широко використовується для навчання штучних нейронних мереж і відомий високою швидкістю збіжності.

### 3.4 Навчання нео-фаззі нейронів у оптимізованій каскадній нейронній мережі

Низька швидкість навчання перцептронів Розенблатта у поєднанні з труднощами інтерпретації результатів (властиві всім штучним нейронним мережам в цілому) спонукає шукати альтернативні підходи до синтезу еволюційних нейронних мереж. Як зазначається у [20], нейро-фаззі системи відомі високою інтерпретовністю і прозорістю, а також високими апроксимаційними властивостями, та є основою гібридних систем штучного інтелекту. У [20] розглядаються гібридні каскадні системи штучного інтелекту побудовані на нео-фаззі нейронах, що дозволяє їм суттєво підвищити швидкість корегування синаптичних вагових коефіцієнтів.

Нео-фаззі нейрон (NFN), що його архітектуру наведено на рисунку 3.3 – це нелінійна система, що реалізує нечітке висновування

$$\hat{y} = \sum_{i=1}^n f_i(x_i), \quad (3.17)$$

де  $x_i$  –  $i$ -й вхідний сигнал  $i = 1, 2, \dots$ ;

$\hat{y}$  – вихідний сигнал нео-фаззі нейрону.

Структурними елементами нео-фаззі нейрона є нелінійні синапси  $NS_i$  які трансформують вхідні сигнали в такий спосіб:

$$f_i(x_i) = \sum_{l=1}^h w_{li} \mu_{li}(x_i), \quad (3.18)$$

де  $w_{li}$  –  $l$ -й ваговий коефіцієнт  $i$ -ого нелінійного синапсу,

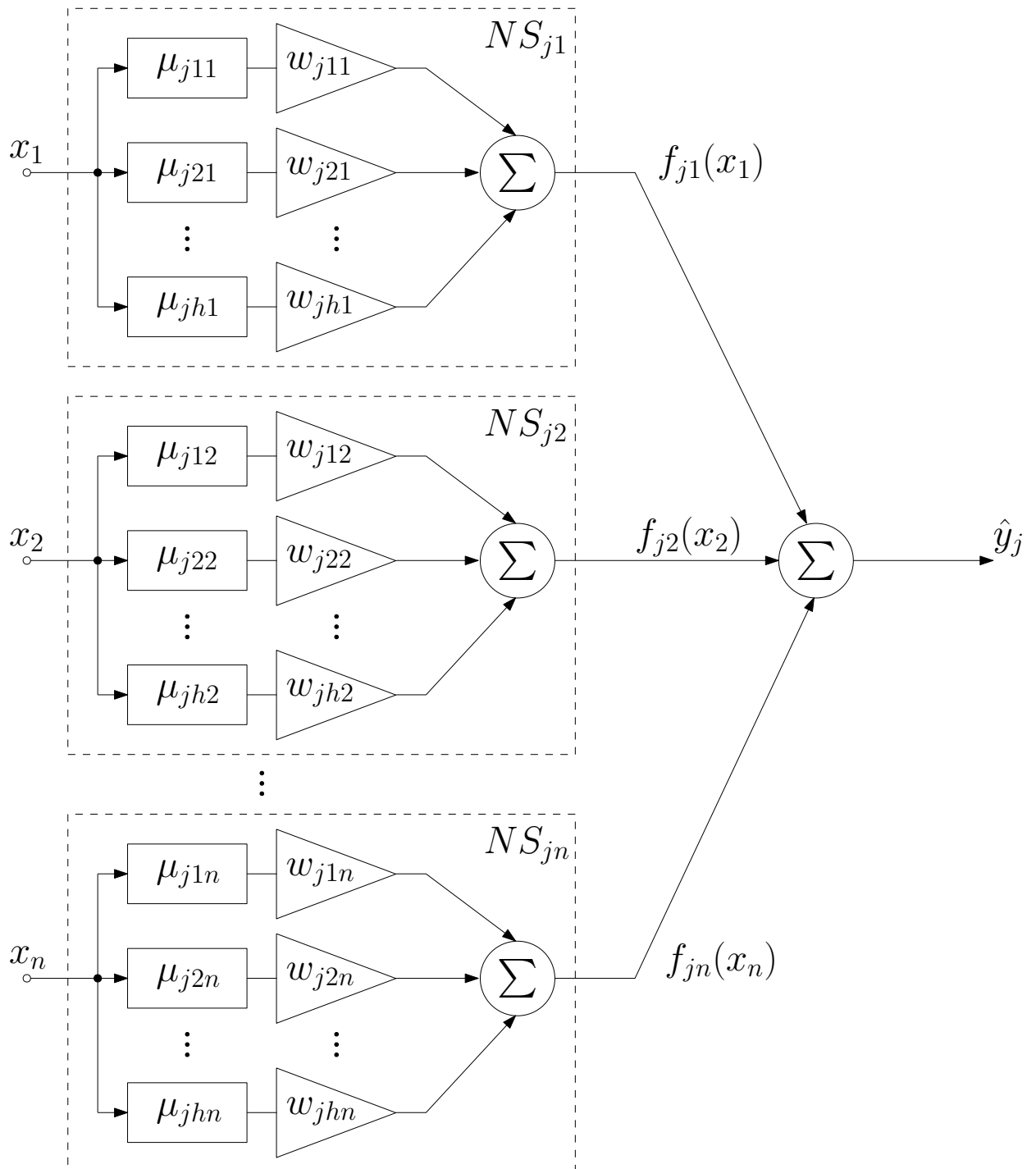


Рисунок 3.3 – Архітектура нео-фаззі нейрону

де  $l = 1, 2, \dots, h$  – кількість синаптичних вагових коефіцієнтів, а отже і функцій належності у  $\mu_{li}(x_i)$  синапсі.

Таким чином, нелінійний синапс  $NS_i$  реалізує нечітке висновування

$$\text{Якщо } x_i \text{ – це } X_{li}, \text{ тоді вихід – } w_{li}, \quad (3.19)$$

де  $X_{li}$  – нечітка множина з функцією належності  $\mu_{li}$ ;

$w_{li}$  – сінглтон (синаптичний ваговий коефіцієнт у консеквенті).

Тобто нелінійний синапс фактично є системою висновування Такагі-Сугено нульового порядку [21].

Запишемо вихідні сигнали для нейронів першого каскаду у такому вигляді (формула 3.20):

$$\left\{ \begin{array}{l} \hat{y}_j^{[1]}(k) = \sum_{i=1}^n f_{ji}^{[1]}(x(k)) = \sum_{i=1}^n \sum_{l=1}^h w_{ji}^{[1]} \mu_{ji}^{[1]}(x_i(k)), \\ \text{якщо } x_i \text{ – це } X_{li}, \text{ тоді вихід – } w_{li}. \end{array} \right. \quad (3.20)$$

Автори нео-фаззі нейрона в якості функцій належності використовували традиційні трикутні структури, які задовольняють умові розбиття Руспіні [21]:

$$\mu_{jli}^{[1]}(x_i) = \begin{cases} \frac{x_i - c_{j,l-1,i}^{[1]}}{c_{jli}^{[1]} - c_{j,l-1,i}^{[1]}}, & \text{якщо } x_i \in [c_{j,l-1,i}^{[1]}, c_{jli}^{[1]}], \\ \frac{c_{j,l+1,i}^{[1]} - x_i}{c_{j,l+1,i}^{[1]} - c_{jli}^{[1]}}, & \text{якщо } x_i \in [c_{jli}^{[1]}, c_{j,l+1,i}^{[1]}], \\ 0 & \text{у протилежному випадку,} \end{cases} \quad (3.21)$$

де  $c_{jli}^{[1]}$  – довільно обрані центри параметрів функцій належності на інтервалі  $[0,1]$ , зазвичай рівномірно розподілені.

Такий вибір функцій належності гарантує, що вхідний сигнал  $x_i$  активує лише два сусідні функції, а сума їх значень завжди дорівнюватиме 1:

$$\mu_{jli}^{[1]}(x_i) + \mu_{j,l+1,i}^{[1]}(x_i) = 1, \quad (3.22)$$

$$f_{jl}^{[1]}(x_i) = w_{jli}^{[1]} \mu_{jli}^{[1]}(x_i) + w_{j,l+1,i}^{[1]} \mu_{j,l+1,i}^{[1]}(x_i). \quad (3.23)$$

Апроксимуючі властивості системи можна поліпшити використовуючи кубічні сплайни замість трикутних функцій належності [22]:

$$\mu_{jli}^{[1]}(x_i) = \begin{cases} \frac{1}{4} \left( 2 + 3 \frac{2x_i - c_{jli}^{[1]} - c_{j,l-1,i}^{[1]}}{c_{jli}^{[1]} - c_{j,l-1,i}^{[1]}} - \left( \frac{2x_i - c_{jli}^{[1]} - c_{j,l-1,i}^{[1]}}{c_{jli}^{[1]} - c_{j,l-1,i}^{[1]}} \right)^3 \right), \\ \text{якщо } x_i \in [c_{j,l-1,i}^{[1]}, c_{jli}^{[1]}], \\ \frac{1}{4} \left( 2 - 3 \frac{2x_i - c_{j,l+1,i}^{[1]} - c_{jli}^{[1]}}{c_{j,l+1,i}^{[1]} - c_{jli}^{[1]}} + \left( \frac{2x_i - c_{j,l+1,i}^{[1]} - c_{jli}^{[1]}}{c_{j,l+1,i}^{[1]} - c_{jli}^{[1]}} \right)^3 \right), \\ \text{якщо } x_i \in [c_{jli}^{[1]}, c_{j,l+1,i}^{[1]}], \\ 0 \text{ у протилежному випадку} \end{cases} \quad (3.24)$$

або  $B$ -сплайни [22]:

$$\mu_{jli}^{g[1]}(x_i) = \begin{cases} 1, \text{ якщо } x_i \in [c_{jli}^{[1]}, c_{j,l+1,i}^{[1]}] \\ 0 \text{ у протилежному випадку} \end{cases} \text{ для } g = 1$$

$$\left( \frac{x_i - c_{jli}^{[1]}}{c_{j,l+g-1,i}^{[1]} - c_{jli}^{[1]}} \mu_{jli}^{g-1,[1]}(x_i) + \frac{c_{j,l+g,i}^{[1]} - x_i}{c_{j,l+g,i}^{[1]} - c_{j,l+1,i}^{[1]}} \mu_{j,l+1,i}^{g-1,[1]} \right) \text{ для } g > 1 \quad (3.25)$$

де  $\mu_{jli}^{g[1]}(x_i)$  –  $l$ -й сплайн  $g$ -ого порядку.

Нескладно помітити, що при  $g = 2$  отримуємо трикутні функції належності (3.21).

$B$ -сплайни, як і трикутні функції належності, забезпечують розбиття Руспіні, але в загальному випадку вони можуть активувати довільне число функцій належності за межами інтервалу  $[0, 1]$ , що може стати у нагоді для подальших каскадів.

Також у якості функцій належності нелінійних синапсів можна використовувати інші структури, такі, як поліноміальні, гармонійні функції, вейвлети, ортогональні функції, тощо. Проте не можна сказати наперед, які з функцій забезпечать кращі результати, тому ідея використання не одного нейрона, а пулу нейронів з різними функціями належності та активації виглядає доречною та перспективною.

За аналогією до (3.20) визначаємо вихідні сигнали інших каскадів. Так, для другого каскаду можемо записати вихідні сигнали у формі (3.26)

$$\hat{y}_j^{[2]} = \sum_{i=1}^n \sum_{l=1}^h w_{jli}^{[2]} \mu_{jli}^{[2]}(x_i) + \sum_{l=1}^h w_{j,l,n+1}^{[2]} \mu_{j,l,n+1}^{[2]}(\hat{y}^{*[1]}), \quad (3.26)$$

вихідні сигнали для нейронів  $m$ -ого каскаду

$$\hat{y}_j^{[m]} = \sum_{i=1}^n \sum_{l=1}^h w_{jli}^{[m]} \mu_{jli}^{[m]}(x_i) + \sum_{p=n+1}^{n+m-1} \sum_{l=1}^h w_{jlp}^{[m]} \mu_{jlp}^{[m]}(\hat{y}^{*[p-n]}) \quad (3.27)$$

Таким чином, каскадна нейронна мережа з нео-фаззі нейронів, що сформована  $m$  каскадами, містить  $h(\sum_{p=1}^{m-1} p)$  параметрів.

Введемо вектор функцій належності для  $j$ -ого нео-фаззі нейрона  $m$ -ого каскаду

$$\begin{aligned} & \mu_j^{[m]}(k) \\ = & \left( \mu_{j11}^{[m]}(x_1(k)), \dots, \mu_{jh1}^{[m]}(x_1(k)), \mu_{j12}^{[m]}(x_2(k)), \dots, \mu_{jh2}^{[m]}(x_2(k)), \dots, \right. \\ & \left. \mu_{jhn}^{[m]}(x_i(k)), \dots, \mu_{j1,n+1}^{[m]}(\hat{y}^{*[1]}(k)), \dots, \mu_{jh,n+m-1}^{[m]}(\hat{y}^{*[m-1]}(k)) \right)^T \end{aligned} \quad (3.28)$$

та відповідний вектор синаптичних вагових коефіцієнтів

$$\begin{aligned} w_j^{[m]} = & \left( w_{j11}^{[m]}, \dots, w_{jh1}^{[m]}, \dots, w_{j12}^{[m]}, \dots, w_{jh2}^{[m]}, \dots, \right. \\ & \left. \dots, w_{jli}^{[m]}, \dots, w_{jhn}^{[m]}, \dots, w_{j1,n+1}^{[m]}, \dots, w_{jh,n+m-1}^{[m]} \right)^T. \end{aligned} \quad (3.29)$$

Тоді можемо компактно записати вихідні сигнали для  $j$ -ого нейрону  $m$ -ого каскаду

$$\hat{y}_j^{[m]}(k) = w_k^{[m]T} \mu_j^{[m]}(k). \quad (3.30)$$

У такому разі критерій навчання (3.) приймає вигляд (формула 3.31)

$$E_j^{[m]}(k) = \frac{1}{2} \left( e_j^{[m]}(k) \right)^2 = \frac{1}{2} \left( y(k) - w_j^{[m]T} \mu_j^{[m]}(k) \right)^2, \quad (3.31)$$

а мінімізувати його можна використавши модифікацію процедури для ковзного вікна [23]

$$\begin{cases} w_j^{[m]}(k+1) = w_j^{[m]}(k) + \frac{e_j^{[m]}(k+1)\mu_j^{[m]}(k+1)}{r_j^{[m]}(k+1)} \\ r_j^{[m]}(k+1) = r_j^{[m]}(k) + \left\| \mu_j^{[m]}(k+1) \right\|^2 - \left\| \mu_j^{[m]}(k-s) \right\|^2 \end{cases} \quad (3.32)$$

або для випадку, коли  $s = 1$ ,

$$w_j^{[m]}(k+1) = w_j^{[m]}(k) + \frac{e_j^{[m]}(k+1)\mu_j^{[m]}(k+1)}{\left\| \mu_j^{[m]}(k+1) \right\|^2}, \quad (3.33)$$

що збігається з одношаговим оптимальним алгоритмом Качмажа-Уїдроу-Хоффа.

Вочевидь, замість (3.33) можна скористатися іншими алгоритмами, як-от експоненційно зважений рекурентний метод найменших квадратів (EWRLSM), що використовується у DENFIS, ETS, та FLEXFIS [20-23]. Та варто зауважити, що EWRLSM може бути нестійким при малому коефіцієнті забування.

При використанні критерія навчання з регуляризуючим параметром (3.8) замість (3.31) отримуємо остаточний метод навчання неофаззі нейрона

$$\left\{ \begin{array}{l} w_j^{[m]}(k+1) = w_j^{[m]}(k) + \frac{\eta e_j^{[m]}(k+1) \mu_j^{[m]}(k+1)}{r_j^{[m]}(k+1)} + \\ (1-\eta) \left( w_j^{[m]}(k) - w_j^{[m]}(k-1) \right) \\ + \frac{\quad}{r_j^{[m]}(k+1)}, \\ r_j^{[m]}(k+1) = r_j^{[m]}(k) + \left\| \mu_j^{[m]}(k+1) \right\|^2 - \left\| \mu_j^{[m]}(k-s) \right\|^2. \end{array} \right. \quad (3.34)$$

Варто наголосити, що оскільки вихідні сигнали нео-фаззі нейрона лінійно залежать від його синаптичних вагових коефіцієнтів, можна використовувати будь-які методи адаптивної лінійної ідентифікації (наприклад, рекурентний метод найменших квадратів, робасні методи, методи, що ігнорують застарілі данні, тощо), що дозволяє обробляти нестационарні сигнали в онлайн режимі.

3.5 Розширені нео-фаззі нейрони в якості елементів гібридної каскадної мережі, що еволюціонує

Як зазначалося вище, розглядаючи нелінійний синапс нео-фаззі нейрону з позицій нечіткої логіки, нескладно побачити, що він є вельми схожим на шар фаззіфікування таких нейро-фаззі систем як мережі Такагі-Сугено-Канга, Дженга, Ванга-Менделя, і, фактично реалізує нечітке висновування Такагі-Сугено нульового порядку [24].

Та задля поліпшення апроксимуючих властивостей таких систем видається доцільним запропонувати удосконалений нелінійний синапс такий, що реалізує нечітке висновування довільного порядку, далі «розширений нелінійний синапс» (ENS), та зсинтезувати «розширений нео-фаззі нейрон» (ENFN), що містить такі структури замість традиційних нелінійних синапсів  $NS_i$ . Архітектури розширеного нелінійного синапсу та розширеного нео-фаззі нейрону наведено на рисунку 3.4 та рисунку 3.5 відповідно.

Вводять нові змінні

$$\phi_{li}(x_i) = \mu_{li}(x_i)(w_{li}^0 + w_{li}^1 x_i + w_{li}^2 x_i^2 + \dots + w_{li}^p x_i^p), \quad (3.35)$$

$$\begin{aligned} f_i(x_i) &= \sum_{l=1}^h \mu_{li}(x_i)(w_{li}^0 + w_{li}^1 x_i + w_{li}^2 x_i^2 + \dots + w_{li}^p x_i^p) = \\ &= w_{1i}^0 \mu_{1i}(x_i) + w_{1i}^1 x_i \mu_{1i}(x_i) + \dots + w_{1i}^p x_i^p \mu_{1i}(x_i) + \\ &\quad + w_{2i}^0 \mu_{2i}(x_i) + w_{2i}^1 x_i \mu_{2i}(x_i) + \dots + w_{hi}^p x_i^p \mu_{hi}(x_i) \end{aligned} \quad (3.36)$$

$$\begin{aligned} \tilde{\mu}_i(x_i) &= (\mu_{1i}(x_i), x_i(\mu_{1i}(x_i)), \dots, x_i^p(\mu_{1i}(x_i)), \\ &\quad \mu_{2i}(x_i), \dots, x_i^p(\mu_{2i}(x_i)), \dots, x_i^p(\mu_{hi}(x_i)))^T, \end{aligned} \quad (3.37)$$

можна представити вихідні сигнали розширеного нео-фаззі нейрона у вигляді (3.38) та (3.39)

$$f_i(x_i) = w_i^T \tilde{\mu}_i(x_i), \quad (3.38)$$

$$\hat{y} = \sum_{i=1}^n f_i(x_i) = \sum_{i=1}^n w_i^T \tilde{\mu}_i(x_i) = \tilde{w}^T \tilde{\mu}(x), \quad (3.39)$$

де  $\tilde{w}^T = (w_1^T, \dots, w_i^T, \dots, w_n^T)^T$  – вектор синоптичних ваг;

$\tilde{\mu}^T = (\tilde{\mu}_1^T(x_1), \dots, \tilde{\mu}_i^T(x_i), \dots, \tilde{\mu}_n^T(x_n))^T$  – вектор функцій належності.

Таким чином, *ENFN* містить  $(p+1)hn$  вагових коефіцієнтів та реалізує нечітке висновування Такагі-Сугено  $p$ -ого порядку, а висновування, що його реалізує кожний розширений нелінійний синапс *ENS<sub>i</sub>* можна записати у формі (3.40)

Якщо  $x_i$  це  $X_{li}$  тоді вихід –  $w_{li}^0 + w_{li}^1 x_i + w_{li}^p x_i^p, l = 1, 2, \dots, h,$  (3.40)

що збігається з нечітким висновуванням Такагі-Сугено  $p$ -ого порядку.

Коли подати векторний сигнал  $x(k)$  на вхід ENFN першого каскаду, на виході отримуємо скалярне значення

$$\hat{y}^{[1]}(k) = \tilde{w}^{[1]T}(k-1) \tilde{\mu}^{[1]}(x(k)), \quad (3.41)$$

що відрізняється від виразу (3.30) для звичайних NFN тим, що містить у  $p + 1$  більше параметрів, що корегуються.

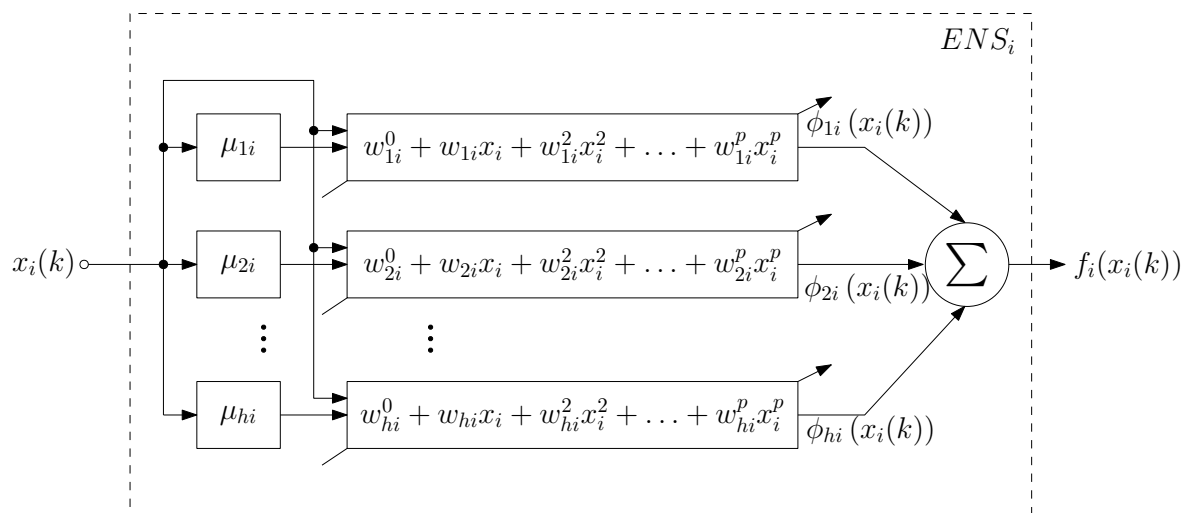


Рисунок 3.4 – Синапс розширеного нео-фаззі нейрону

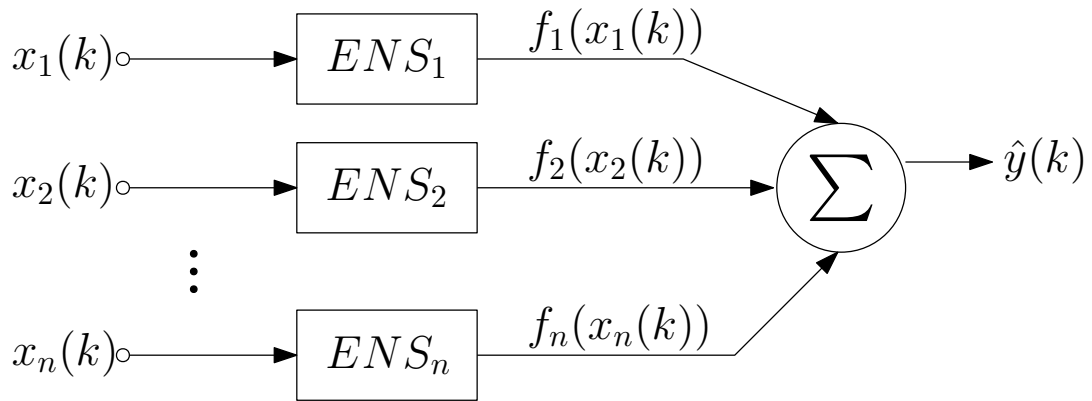


Рисунок 3.5 – Розширений нео-фаззі нейрон

Вочевидь, будь-які методи навчання нео-фаззі нейронів підійдуть і для розширених нео-фаззі нейронів. Так, вирази (3.32) та (3.33) для  $j$ -ого нейрону  $m$ -ого каскаду приймають вигляд

$$\begin{cases} \tilde{w}_j^{[m]}(k+1) = \tilde{w}_j^{[m]}(k) + \frac{e_j^{[m]}(k+1)\tilde{\mu}_j^{[m]}(k+1)}{\tilde{r}_j^{[m]}(k+1)}, \\ \tilde{r}_j^{[m]}(k+1) = \tilde{r}_j^{[m]}(k) + \left\| \tilde{\mu}_j^{[m]}(k+1) \right\|^2 - \left\| \tilde{\mu}_j^{[m]}(k-s) \right\|^2 \end{cases} \quad (3.42)$$

та

$$\tilde{w}_j^{[m]}(k+1) = \tilde{w}_j^{[m]}(k) + \frac{e_j^{[m]}(k+1)\tilde{\mu}_j^{[m]}(k+1)}{\left\| \tilde{\mu}_j^{[m]}(k+1) \right\|^2} \quad (3.43)$$

відповідно.

### 3.6 Оптимізація пулу нео-фаззі нейронів

Вихідні сигнали, згенеровані нейронами пулу кожного з каскадів,

можна об'єднати у окремому вузлі-нейроні  $GN^{[m]}$ , з точністю  $\hat{y}_j^{*[m]}(k)$ , не меншою від точності будь-якого нейрону пулу  $\hat{y}_j^{[m]}(k)$ . Це завдання можна вирішити за допомогою підходу, пов'язаного з ансамблями нейронних мереж.

Хоча відомі алгоритми не призначені для роботи в онлайн-режимі, варто розглянути методи адаптивного узагальнюючого прогнозування [25].

Введемо вектор вхідних сигналів для  $m$ -ого каскаду:

$$\hat{y}_j^{[m]}(k) = \left( \hat{y}_1^{[m]}(k), \hat{y}_2^{[m]}(k), \dots, \hat{y}_q^{[m]}(k) \right)^T; \quad (3.44)$$

тоді оптимальний вихідний сигнал, що його генерує нейрон  $GN^{[m]}$  (що, власне, є адаптивним лінійним асоціатором), можна записати у формі [16], [26]

$$\hat{y}_j^{*[m]}(k) = \sum_{j=1}^q c_j^{[m]} \hat{y}_j^{[m]}(k) = c^{[m]T} \hat{y}^{[m]}(k) \quad (3.45)$$

з обмеженнями на незміщенність

$$\sum_{j=1}^q c_j^{[m]} = E^T c^{[m]} = 1, \quad (3.46)$$

де  $c^{[m]} = (c_1^{[m]}, c_2^{[m]}, \dots, c_q^{[m]})^T$  та  $E = (1, 1, \dots, 1)^T$  –  $(q \times 1)$ -вектори.

Введемо критерій навчання на ковзному вікні

$$\begin{aligned}
E^{[m]}(k) &= \frac{1}{2} \sum_{\tau=k-s+1}^k \left( y(\tau) - \hat{y}^{*[m]}(\tau) \right)^2 = \\
&= \frac{1}{2} \sum_{\tau=k-s+1}^k \left( y(\tau) - c^{[m]T} \hat{y}^{[m]}(\tau) \right)^2,
\end{aligned} \tag{3.47}$$

зважаючи на обмеження (3.), функція Лагранжа матиме вигляд (3.48)

$$L^{[m]}(k) = E^{[m]}(k) - \lambda(1 - E^T c^{[m]}), \tag{3.48}$$

де  $\lambda$  – невизначений Лагранжів множник. Мінімізуючи (3.) відносно  $c^{[m]}$ , отримуємо

$$\begin{cases} \hat{y}^{*[m]}(k+1) = \frac{\hat{y}^{[m]T}(k+1)P^{[m]}(k+1)E}{E^T P^{[m]}(k+1)E}, \\ P^{[m]}(k+1) = \left( \sum_{\tau=k-s+2}^{k+1} \hat{y}^{[m]}(\tau)\hat{y}^{[m]T}(\tau) \right)^{-1} \end{cases} \tag{3.49}$$

або у рекурентній формі (3.50)

$$\begin{cases} \tilde{P}^{[m]}(k+1) = P^{[m]}(k) - \frac{P^{[m]}(k)\hat{y}^{[m]}(k+1)\hat{y}^{[m]T}(k+1)P^{[m]}(k)}{1 + \hat{y}^{[m]T}(k+1)P^{[m]}(k)\hat{y}^{[m]}(k+1)}, \\ P^{[m]}(k+1) = \tilde{P}^{[m]}(k+1) + \\ \quad + \frac{\tilde{P}^{[m]}(k+1)\hat{y}^{[m]}(k-s+1)\hat{y}^{[m]T}(k-s+1)\tilde{P}^{[m]}(k+1)}{1 - \hat{y}^{[m]T}(k-s+1)\tilde{P}^{[m]}(k+1)\hat{y}^{[m]}(k-s+1)} \\ \hat{y}^{*[m]}(k+1) = \frac{\hat{y}^{[m]T}(k+1)P^{[m]}(k+1)E}{E^T P^{[m]}(k+1)E}. \end{cases} \tag{3.50}$$

У випадку одиничного ковзного вікна  $s = 1$ , (3.) та (3.) приймають доволі простий вигляд (3.51):

$$\begin{aligned}\hat{y}^{*[m]}(k+1) &= \frac{\hat{y}^{[m]T}(k+1)\hat{y}^{[m]}(k+1)}{E^T\hat{y}^{[m]}(k+1)} = \\ &= \frac{\|\hat{y}^{[m]}(k+1)\|^2}{E^T\hat{y}^{[m]}(k+1)} = \\ &= \frac{\sum_{j=1}^q \left(\hat{y}^{[m]}(k+1)\right)^2}{\sum_{j=1}^q \hat{y}^{[m]}(k+1)}.\end{aligned}\tag{3.51}$$

Важливо зазначити, що навчання як нео-фаззі нейронів, так і нейронів-узагальнювачів можна організувати в онлайн-режимі. Таким чином, вагові коефіцієнти нейронів попередніх каскадів (на відміну від CasCorLA) можна не заморожувати, а постійно корегувати. Так само, число каскадів не має бути фіксованим і може змінюватись у часі, що відрізняє запропоновану нейронну мережу від інших відомих каскадних систем.

В даному розділі було розглянуто існуючі гібридні системи обчислювального інтелекту, що еволюціонують, та визначені потенційні модифікації, що їх варто привнести аби такі системи можна було застосувати у режимі послідовного надходження даних на опрацювання.

Запропоновано архітектуру та метод навчання гібридної каскадної системи, що еволюціонує, засновану на нео-фаззі нейронах. Пропонованій системі притаманні усі переваги нео-фаззі нейронів (інтерпритовність та прозорість одночасно з високими апроксимаційними властивостями), а також, зрештою, вона забезпечує модель адекватної складності для кожного поставленого завдання.

## 4 ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ РОЗРОБЛЕНОЇ КАСКАДНОЇ НЕЙРО-ФАЗЗИ МЕРЕЖІ

З задачею навчання тісно пов'язана достатньо часто виникаюча на практиці задача апроксимації функцій, які задані на деякому наборі точок. Розглянемо нелінійне відображення «вхід-вихід», яке описується функціональним співвідношенням (формула 4.1)

$$d = f(x), \quad (4.1)$$

де  $d$  та  $x$  –  $(m \times 1)$  и  $(n \times 1)$  – вектори виходів та входів відповідно;

$f(\bullet)$  – невідома вектор-функція, яку необхідно оцінити за допомогою заданої навчальної вибірки;

$\{x(k), d(k)\}, k = 1, 2, \dots, N$  – навчальна вибірка даних.

Задача навчання апроксимуючої нейронної мережі складається з знаходження функції  $F(x)$  в деякому сенсі достатньо близької до  $f(x)$  такої, що

$$\|F(x) - f(x)\| \leq \varepsilon \quad \text{для всіх } x(k), k = 1, 2, \dots, N, \quad (4.2)$$

де  $F(x)$  – відображення, яке реалізує мережа;

$\varepsilon$  – мале позитивне число.

Якщо об'єм вибірки  $N$  достатньо великий, а мережа має достатню кількість синаптичних ваг, які треба налаштувати, похибка апроксимації  $\varepsilon$  може бути зроблена достатньо малою, хоча тут є можливість перетворення мережі із апроксимуючої в інтерполяційну.

Нескладно помітити, що проблема апроксимації в цьому контексті повністю співпадає з завданням навчання зі вчителем, де послідовність  $x(k)$  грає роль вхідного сигналу нейронної мережі, а  $d(k)$  – це

навчальний сигнал.

Здатність нейромереж апроксимувати невідомі відображення «вхід-вихід» знаходить два найважливіших додатки в задачах інтелектуального управління [27]. Перше з них – ідентифікація об'єктів управління, або емуляція – в термінах нейроуправління [28]. Схема системи ідентифікації (емуляції) приведена на рисунку 4.1, при цьому передбачається, що багатовимірний статичний об'єкт описується співвідношенням (формула 4.1), а нейронна мережа, включена паралельно об'єкту, навчається в реальному часі, «підганяючи» свої вихідні сигнали до виходів реального об'єкта.

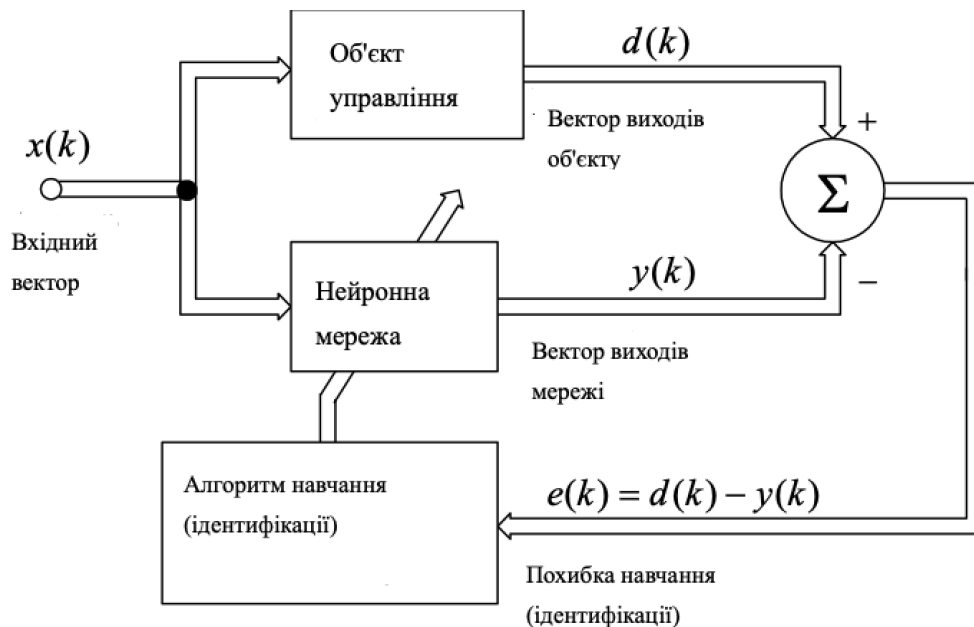


Рисунок 4.1 – Схема системи ідентифікації

Другий додаток – це зворотне моделювання, яке використовується в деяких адаптивних системах управління і полягає в тому, що для об'єкта управління (формула 4.1) потрібно побудувати «зворотню систему», яка генерує вектор  $x(k)$  як відгук на вхідний сигнал  $d(k)$  [28]. У загальному вигляді зворотня система має форму (4.3)

$$x = f^{-1}(d), \quad (4.3)$$

однак, оскільки функція  $f$  або не відома, або занадто складна, розумним виходом є використання ІНС в якості зворотнього моделі так, як це показано на рисунку 4.2.

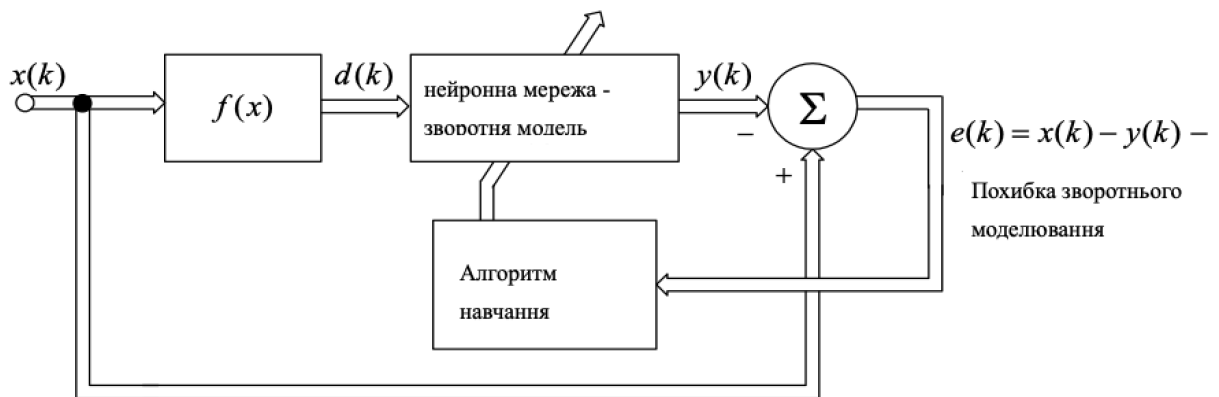


Рисунок 4.2 – Схема зворотнього моделювання

У цій схемі ролі сигналів  $x(k)$  і  $d(k)$  помінялися: вектор  $d(k)$  використовується як вхід мережі, а  $x(k)$  – як бажаний відгук (навчальний сигнал). Подібно системі ідентифікації сигнал похибки  $e(k) = x(k) - y(k)$  використовується для навчання ШНМ.

#### 4.1 Фільтрація, згладжування, прогнозування

Системи обробки «зашумлених» сигналів в умовах невизначеності в даний час знаходять широке застосування в найрізноманітніших додатках [28]. Власне поняття «обробка сигналів» традиційно включає в себе три завдання: фільтрація, згладжування і прогнозування. Якщо в розпорядженні дослідника є вибірка «забруднених» спостережень  $x(1), x(2), \dots, x(k), \dots, x(N)$ , то задача фільтрації зводиться до знаходження найкращої оцінки процесу  $\hat{x}(N | N)$  в момент часу  $N$  за інформацією про  $N$

спостереженнях, згладжування – оцінки  $\hat{x}(k|N)$  при  $\hat{x}(k|N)$  і прогнозування – при  $N+l > N$ , де  $l$  – горизонт попередження.

Останні роки увагу дослідників притягнуто ще до однієї нетрадиційної задачі обробки – «сліпий» сепарації та ідентифікації сигналів [29]. Передбачається, що є множина невідомих джерел сигналів  $\{u_i(k)\}_{i=1}^n$ , які не залежать одне від одного. Сенсори сприймають ці сигнали покомпонентно, а в суміші, що представляє собою невідому лінійну комбінацію  $\{u_i(k)\}_{i=1}^n$  так, як це показано на рисунк 4.3.

Ця задача зводиться до відновлення вектора  $y(k) \approx u(k)$  за даними спостережень вектора  $x(k)$  при невідомої  $(n \times n)$  – матриці.

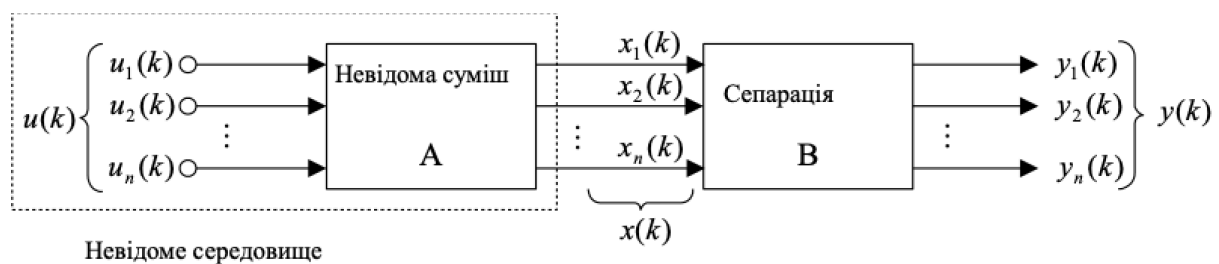


Рисунок 4.3 – Схема сліпої сепарації

Нескладно бачити, що перші три задачі дуже близькі до проблеми ідентифікації, а завдання сліпої сепарації практично збігається із завданням зворотного моделювання і зводиться до знаходження оператора сепарації  $B = A^{-1}$ . Природньо, що застосування ШНМ для вирішення цих завдань принципових труднощів не викликає. Зупинимося коротко на задачі поточного прогнозування стохастичної послідовності  $x(k)$  за даними про її передісторії  $x(k-1), x(k-2), \dots$

Проблема зводиться до знаходження оцінки  $\hat{x}(k) = F(x(k-1), x(k-2), \dots, x(k-p))$  в реальному часі в темпі з надходженням даних. У лінійному випадку ця задача добре досліджена і успішно може бути вирішена за допомогою адаптивних прогнозуючих авторегресійних

моделей [29]. Для побудови ж нелінійних прогнозів найбільш доцільним видається застосування ІНС, наприклад так, як це показано на рисунк 4.4.

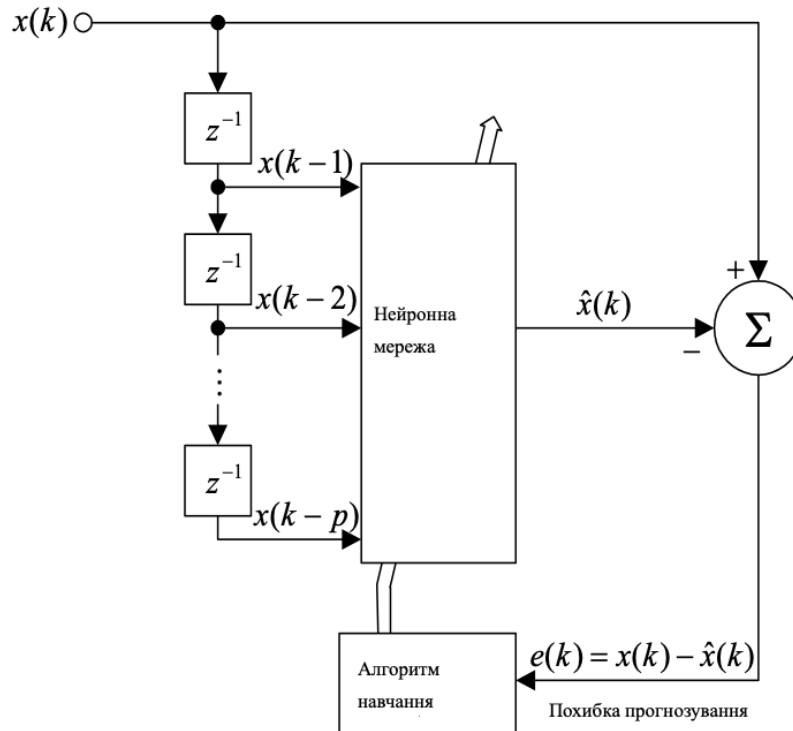


Рисунок 4.4 – Схема нейромережевого прогнозування

В даний час нейромережеві прогнозуючі моделі успішно використовуються для вирішення широкого кола завдань науки, техніки, економіки [30].

#### 4.2 Моделювання гібридної каскадної нейро-фаззі мережі з розширеними нео-фаззі нейронами та оптимізацією пулу нейронів

Для проведення чисельних експериментів, було обрано такі критерії оцінки: RMSE (Root Mean Squared Error, середньоквадратична похибка), SMAPE (Symmetric Mean Absolute Percentage Error, симетрична абсолютна процентна похибка), що обчислюються за формулами (4.4) та (4.5) відповідно,

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (y(k) - \hat{y}(k))^2}, \quad (4.4)$$

$$SMAPE = \frac{1}{N} \sum_{k=1}^N \frac{|y(k) - \hat{y}(k)|}{y(k) + \hat{y}(k)} \quad (4.5)$$

де  $y(k)$  – шуканий сигнал;

$\hat{y}(k)$  – вихідний сигнал системи.

Низку експериментів для апробації гібридної каскадної нейро-фаззі мережі на розширених нео-фаззі нейронах з оптимізацією пулу нейронів було проведено на датасетах, що їх надала дослідницька група «The Applications of Machine Learning (AML)» з Університету Аалто, що у Фінляндії (Aalto University School of Science, Espoo, Finland). Одним з таких датасетів є часовий ряд «Споживання електроенергії у Польщі за період з 1990-х років» (фазовий портрет наведено на рисунку 4.5).

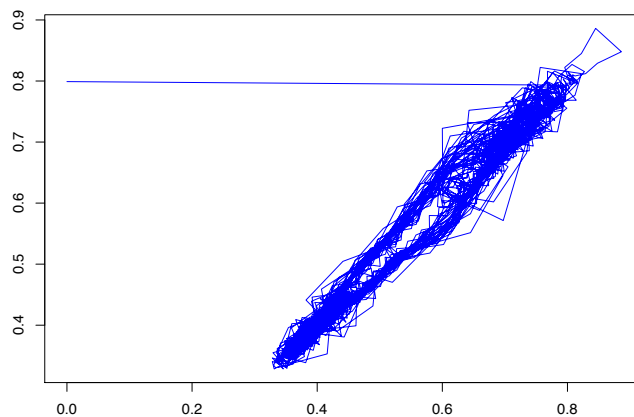


Рисунок 4.5 – Фазовий портрет часового ряду «Споживання електроенергії у Польщі за період з 1990-х років»

Таблиця 4.1 – Результати прогнозування часового ряду «Споживання електроенергії у Польщі за період з 1990-х років» нейронів каскадної нейрофаззи системи, що еволюціонує

Каскад I	SMAPE	RMSE
Нейрон I (3 функції належності, нечітке висновування III порядку)	0.0808	0.393
Нейрон II (5 функцій належності, нечітке висновування IV порядку)	0.075	0.037
Нейрон III (4 функції належності, нечітке висновування V порядку)	0.075	0.038
Нейрон-узагальнювач I каскаду	0.059	0.030
Каскад II	SMAPE	RMSE
Нейрон I (3 функції належності, нечітке висновування III порядку)	0.080	0.039
Нейрон II (5 функцій належності, нечітке висновування IV порядку)	0.059	0.037
Нейрон III (4 функції належності, нечітке висновування V порядку)	0.075	0.038
Нейрон-узагальнювач II каскаду	0.059	0.037
Каскад III	SMAPE	RMSE
Нейрон I (3 функції належності, нечітке висновування III порядку)	0.081	0.039
Нейрон II (5 функцій належності, нечітке висновування IV порядку)	0.059	0.037
Каскад III	SMAPE	RMSE
Нейрон III (4 функції належності, нечітке висновування V порядку)	0.750	0.038

Продовження таблиці 4.1

Каскад III	SMAPE	RMSE
Нейрон-узагальнювач III каскаду	0.059	0.030
Каскад IV	SMAPE	RMSE
Нейрон I (3 функції належності, нечітке висновування III порядку)	0.080	0.039
Нейрон II (5 функцій належності, нечітке висновування IV порядку)	0.059	0.030
Нейрон III (4 функції належності, нечітке висновування V порядку)	0.075	0.038
Нейрон-узагальнювач IV каскаду	0.059	0.030
Нейрон-узагальнювач системи	0.059	0.030

Результати прогнозування вищезазначеного ряду пропонованою каскадною нейро-мережею проілюстровано на рисунку 4.6.

Також зсинтезовану каскадну нейро-фаззі систему, що еволюціонує, було апробовано на ще одному реальному часовому ряді – «Коливання рівню приловоотливної зони» (Subtidal coastal level of fluctuations).

Таблиця 4.2, де наведено похибки (SMAPE та RMSE) усіх нейронів для кожного каскаду, демонструє значимість узагальнюючих елементів каскадів та системи вцілому.

Фазовий портрет датасету «Коливання рівню приловоотливної зони» наведено на рисунку 4.7.

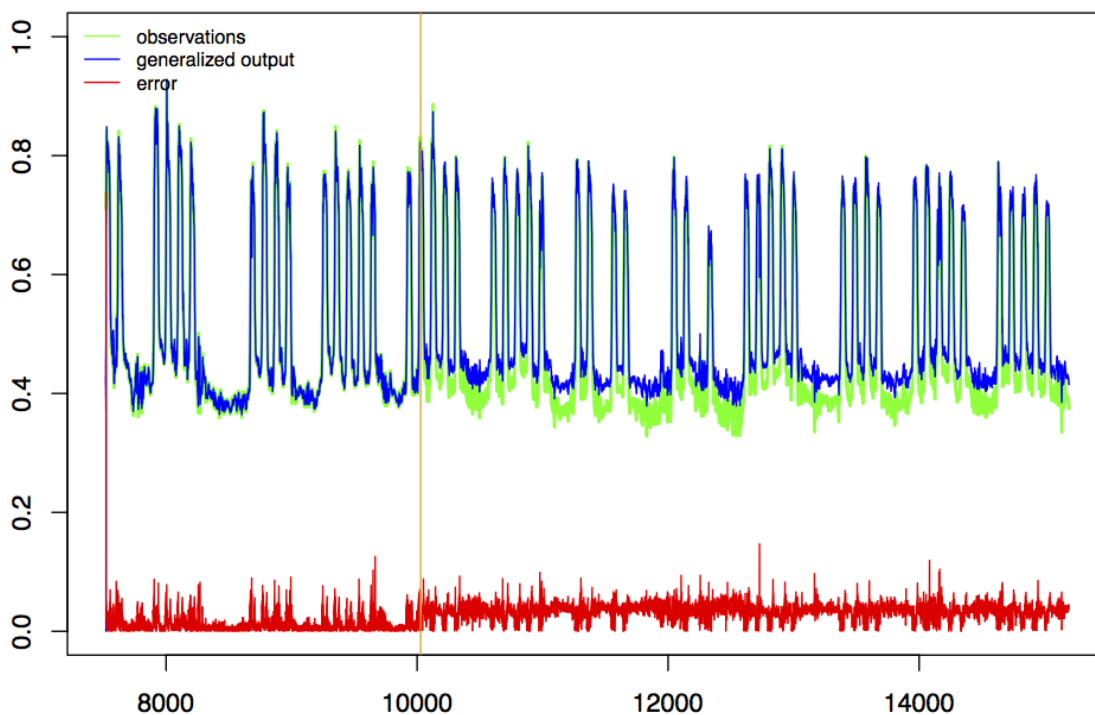


Рисунок 4.6 – Прогнозування часового ряду «Споживання електроенергії у Польщі за період з 1990-х років» оптимізованою каскадною нейро-фаззі мережею

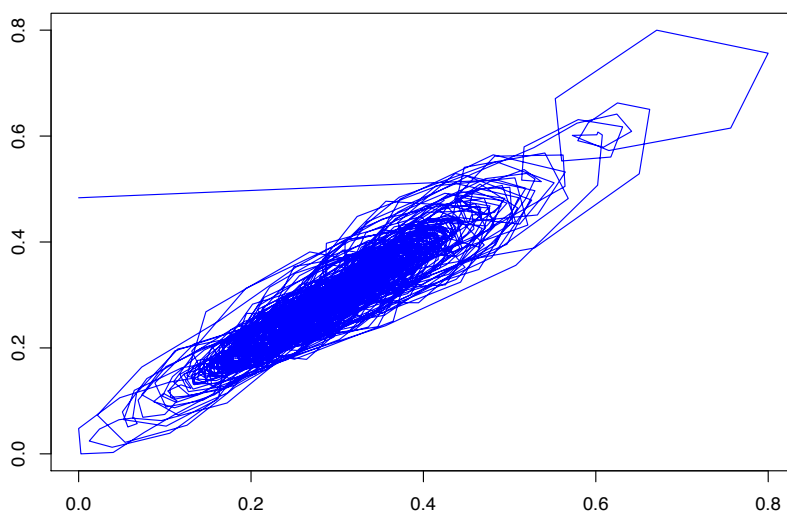


Рисунок 4.7 – Фазовий портрет часового ряду «Коливання рівню приливоотливної зони»

Таблиця 4.2 – Результати прогнозування часового ряду «Коливання рівню приловоотливної зони» нейронів каскадної нейро-фаззі системи

Каскад I	SMAPE	RMSE
Нейрон I (3 функції належності, нечітке висновування III порядку)	0.110067	0.036612
Нейрон II (5 функцій належності, нечітке висновування IV порядку)	0.105192	0.035474
Нейрон III (4 функції належності, нечітке висновування V порядку)	0.103129	0.034814
Нейрон-узагальнювач I каскаду	0.105598	0.035301
Каскад II	SMAPE	RMSE
Нейрон I (3 функції належності, нечітке висновування III порядку)	0.110023	0.035457
Нейрон II (5 функцій належності, нечітке висновування IV порядку)	0.105118	0.036683
Каскад II	SMAPE	RMSE
Нейрон III (4 функції належності, нечітке висновування V порядку)	0.103148	0.035818
Нейрон-узагальнювач II каскаду	0.10237	0.03469
Каскад III	SMAPE	RMSE
Нейрон I (3 функції належності, нечітке висновування III порядку)	0.110013	0.03659
Нейрон II (5 функцій належності, нечітке висновування IV порядку)	0.105126	0.035458
Нейрон III (4 функції належності, нечітке висновування V порядку)	0.103155	0.034820
Нейрон-узагальнювач III каскаду	0.103155	0.034820

Продовження таблиці 4.2

Каскад IV	SMAPE	RMSE
Нейрон I (3 функції належності, нечітке висновування III порядку)	0.110026	0.036594
Нейрон II (5 функцій належності, нечітке висновування IV порядку)	0.105153	0.035464
Нейрон III (4 функції належності, нечітке висновування V порядку)	0.103168	0.034822
Нейрон-узагальнювач IV каскаду	0.102545	0.034083
Нейрон-узагальнювач системи	0.102377	0.030302

На рисунку 4.8 видно, що похибка прогнозування такого часового ряду є невеликою, і під час налаштування системи, і після закінчення тренувальної частини вибірки, що позначено вертикальною жовтогарячою лінією.

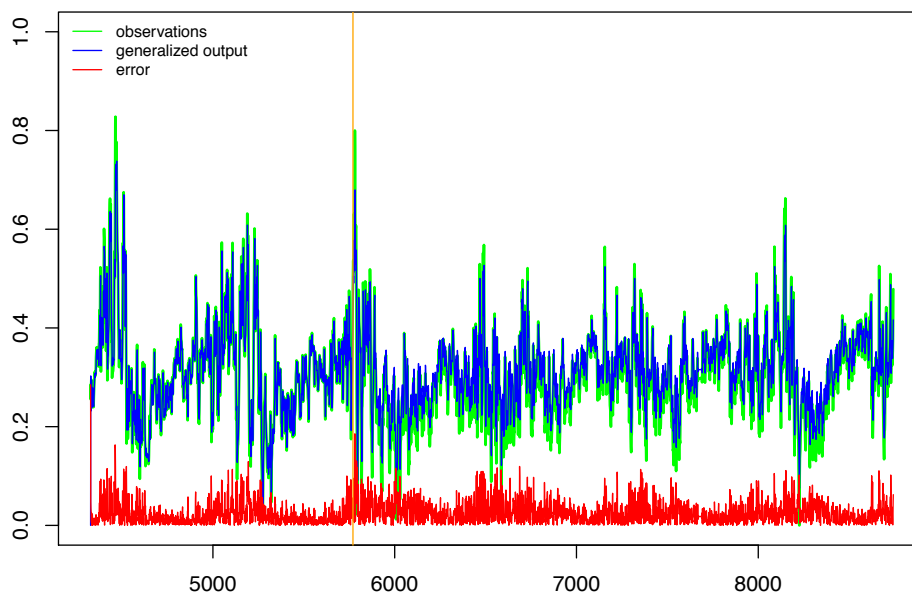


Рисунок 4.8 – Прогнозування часового ряду «Коливання рівню приловоотливної зони» гібридної каскадної нейро-фаззі мережі з оптимізацією пулу нейронів

Останній датасет для цієї низки експериментів було взято зі змагань у прогнозуванні часових рядів «European Symposium on Time Series Prediction 2008», фазовий портрет наведено на рисунку 4.9.

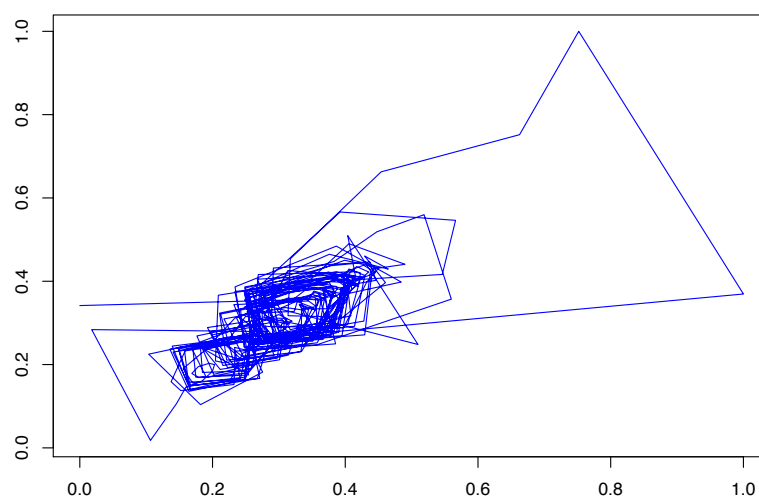


Рисунок 4.9 – Фазовий портрет часового ряду «ESTP Competition Time Series»

На рисунку 4.10 проілюстровано результати роботи запропонованої системи: зеленою лінією позначено шуканий сигнал, синьою – вихідний сигнал узагальнюючого елементу останнього каскаду, червоною лінією – абсолютне значення похибки, вертикальна жовтогаряча лінія позначає закінчення тренувальної частини вибірки.

Таблиця 4.3 – Результати прогнозування часового ряду «ESTP Competition Time Series» нейронів каскадної нейро-фаззі системи

Каскад I	SMAPE	RMSE
Нейрон I (3 функції належності, нечітке висновування III порядку)	0.175623	0.062985
Нейрон II (5 функцій належності, нечітке висновування IV порядку)	0.149139	0.056551

Продовження таблиці 4.3

Каскад I	SMAPE	RMSE
Нейрон III (4 функції належності, нечітке висновування V порядку)	0.146186	0.054414
Нейрон-узагальнювач I каскаду	0.159002	0.055274
Каскад II	SMAPE	RMSE
Нейрон I (3 функції належності, нечітке висновування III порядку)	0.175602	0.062942
Нейрон II (5 функцій належності, нечітке висновування IV порядку)	0.149092	0.056487
Нейрон III (4 функції належності, нечітке висновування V порядку)	0.146278	0.054415
Нейрон-узагальнювач II каскаду	0.158996	0.055264
Каскад III	SMAPE	RMSE
Нейрон I (3 функції належності, нечітке висновування III порядку)	0.175592	0.062933
Нейрон II (5 функцій належності, нечітке висновування IV порядку)	0.149090	0.056487
Нейрон III (4 функції належності, нечітке висновування V порядку)	0.146270	0.054415
Нейрон-узагальнювач III каскаду	0.158986	0.055262
Каскад IV	SMAPE	RMSE
Нейрон I (3 функції належності, нечітке висновування III порядку)	0.175605	0.062947
Нейрон II (5 функцій належності, нечітке висновування IV порядку)	0.149127	0.056487

Продовження таблиці 4.3

Каскад IV	SMAPE	RMSE
Нейрон III (4 функції належності, нечітке висновування V порядку)	0.146323	0.054419
Нейрон-узагальнювач IV каскаду	0.159015	0.055268
Нейрон-узагальнювач системи	0.102377	0.030302

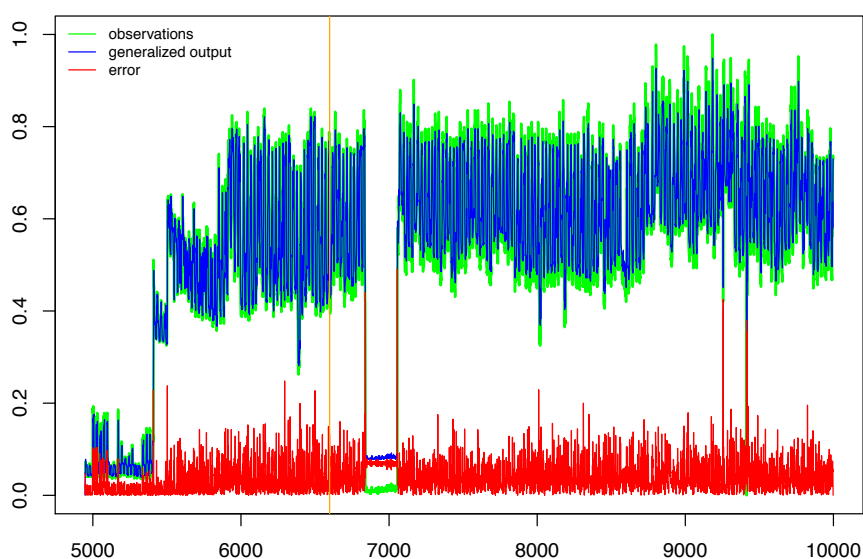


Рисунок 4.10 – Прогнозування часового ряду «ESTP Competition Time Series» гібридної каскадної нейро-фаззі мережі з оптимізацією пулу нейронів

Проведено імітаційне моделювання запропонованої у дипломній роботі магістра архітектури гібридної системи обчислювального інтелекту, що ґрунтується на розширених нео-фаззі нейронах та методу навчання гібридної каскадної нейронної мережі, що еволюціонує, з оптимізацією пулу нейронів у кожному каскаді. Проведені в цьому розділі експериментальні дослідження довели, що пропозиція нейронна мережа реалізує оптимальний за точністю прогноз нелінійних стохастичних і хаотичних сигналів у онлайн режимі.

## ВИСНОВКИ

В кваліфікаційній роботі представлено результати, що відповідають основній тематиці дослідження, а саме – розробці каскадної нечіткої системи і методу її навчання з підвищеною швидкістю і можливостями інтерпретовності вихідного сигналу, а також параметричного та структурного налаштування в режимі послідовного опрацювання інформації. Проведені дослідження дозволили зробити такі висновки:

1. Виконано огляд стану проблеми динамічного інтелектуального аналізу даних, що послідовно надходять на опрацювання. На базі аналізу переваг і недоліків відомих методів зроблено висновок, що найбільш ефективними і пристосованими для обробки нелінійних нестационарних даних в онлайн режимі є методи обчислювального інтелекту, і в першу чергу, нейро-фаззі системи. Виявлено недоліки існуючих гібридних нейро-фаззі систем: неможливість функціонувати в онлайн режимі і відсутність механізмів структурної адаптації.

2. Запропоновано архітектуру та метод навчання каскадної нейро-фаззі мережі, що еволюціонує, з оптимізацією пулу нейронів у кожному каскаді для інтелектуального опрацювання даних, що надходять у послідовному режимі. Варто зазначити, що оптимізація пулу нейронів дуже доречна саме у разі застосування системи для аналізу даних у послідовному режимі, адже використання узагальнюючих нейронів дозволяє визначати оптимальний нейрон на кожному етапі функціонування системи, який з високою вірогідністю може змінюватися у випадку опрацювання сигналів нестационарних об'єктів.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Бодянський Є.В., Тищенко О.К., Копаліані Д.С. Прогнозування багатовимірних нестационарних часових рядів на основі адаптивної неофаззі-моделі. *Вісник НУ «Львівська політехніка». «Комп'ютерні науки та інформаційні технології»*. 2012. № 744. С. 312–118.
2. Bodyanskiy Ye., Tyschenko O., Kopaliani D. A hybrid cascade neural network with an optimized pool in each cascade. *Soft Computing*. 2014. Vol. 19. No 12. P 3445–3454.
3. Аведьян Э. Д., Баркан Г. В., Левин И. К. Каскадные нейронные сети. *Автоматика и телемеханика*. 1999. №3. С. 38-55.
4. Mokris I., Skovajsova L. Proposal of cascade neural network model for text document space dimension reduction by latent semantic indexing. *Symposium on Applied Machine Intelligence and Informatics. Proc. of 6<sup>th</sup> Int. Conf. SAMI 2008*. P. 79-84.
5. Bodyanskiy Ye., Dolotov A., Pliss I., Viktorov Ye. The cascade orthogonal neural network. *Advanced Research in Artificial Intelligence*. Bulgaria, Sofia. Institute of Informational Theories and Applications FOI ITHEA. 2008. Vol. 2. P. 13-20.
6. Cichocki A., Unbehauen R. *Neural Networks for Optimization and Signal Processing*. Stuttgart: Teubner, 1993. 526 p.
7. Kasabov, N. *Evolving Connectionist Systems*. London: Springer-Verlag, 2003. 307p.
8. Bodyanskiy Ye., Viktorov Ye., Pliss I. The cascade growing neural network using quadratic neurons and its learning algorithms for on-line information processing. *Intelligent Information and Engineering Systems. Int. Book Series "Information Science & Computing"* / Eds. by G. Setlak, K. Markov. – Institute of Information Theories and Applications FOI ITHEA, Rzeszow, Poland, 2009. Vol. 13. P. 27-34.

9. Bodyanskiy Ye., Kolodyazhniy V., Slipchenko O. Structural and synaptic adaptation in the artificial neural networks with orthogonal activation functions. *Technology and Management Sci. Sci. Proc. of Riga Technical University. Comp. Sci., Int.* 2004. No20. P. 69-76.

10. Bodyanskiy Ye., Viktorov Ye. The cascade neo-fuzzy architecture and its online learning algorithm. *Int. Book Series "Information Science & Computing", Intelligent Processing.* Bulgaria, Sofia: Institute of Informational Theories and Applications FOI ITHEA, 2009. Vol. 9. P.110-116.

11. Kruschke J., Movellan. J. Benefits of gain: Speeded learning and minimal hidden layers in back-propagation networks. *IEEE*, 1991. Vol. 21. No 1. P. 273-280.

12. Han, J., Kamber M. *Data Mining: Concepts and Techniques.* Amsterdam: Morgan Kaufman Publ. 2006. 743p.

1. Wasan, M.T. *Stochastic Approximation.* Cambridge: The University Press. 2004. 216 p.

14. Arrow K. J., Hurwicz L., Uzava H. *Studies in Linear and Nonlinear Programming.* Stanford: Stanford University Press. 1958. 229 p.

15. Цыпкин, Я.З. *Основы теории обучающихся систем.* М.: Наука. 1970. 252 с.

16. Mackey M. C., Glass L. Oscillation and chaos in physiological control systems. *Science.* 1977. Vol. 197. P. 287-289.

2. Bishop C. M. *Neural Networks for Pattern Recognition.* Oxford: Clarendon Press, 1995. 482 p.

18. Guckenheimer J., Williams R. Structural stability of the Lorenz attractor. *Publ. Math. IHES.* Vol. 50. 1979. P. 55–72.

19. Schalkoff R. *Artificial Neural Networks.* N.Y.: The McGraw–Hill Comp., Inc., 1997. 422 p.

20. Moody J., Darken C.J. Fast learning in networks of locally-tuned processing units. *Neural Computation.* 1989. №1. P. 281–294

21. Айзерман М.А., Браверман Э.М., Розоноэр Л.И. Метод потенциальных функций в теории обучения машин. М.: Наука, 1970. 384 с.
22. Parzen E. On the estimation of a probability density function and the mode. *Ann. Math. Statist.* 1962. №38. P. 1065–1076.
23. Надарая Э.А. О непараметрических оценках плотности вероятности и регрессии. *Теория вероятностей и ее применение.* 1965. 10. № 1. С. 199–203.
24. Haykin S. *Neural Networks. A Comprehensive Foundation.* Upper Saddle River, N.J.: Prentice Hall, Inc., 1999. 842 p.
25. Живоглядов В.П., Медведев А.В. Непараметрические алгоритмы адаптации. Фрунзе: Илим, 1974. 214 с.
26. Медведев А.В. Адаптация в условиях непараметрической неопределенности. *Адаптивные системы и их приложения.* Новосибирск: Наука, 1978. С. 4–34.
27. Раудис Ш.Ю. Оптимизация непараметрического алгоритма классификации. *Адаптивные системы и их приложения.* Новосибирск: Наука, 1978. С. 57–61.
28. Медведев А.В. Непараметрические алгоритмы идентификации нелинейных динамических систем. *Стохастические системы управления.* Новосибирск: Наука, 1979. С. 15–22.
29. Варядченко Т.В., Катковник В.Я. Непараметрический метод обращения функций регрессии. *Стохастические системы управления.* Новосибирск: Наука, 1979. С. 4–14.
30. Hartman E.J., Keeler J.D., Kowalski J. Layered neural networks with Gaussian hidden units as universal approximations. *Neural Computation.* 1990. 2. P. 210–215.
31. Franke R. A critical comparison of some methods for interpolation of scattered data. 1979.
32. Poggio T., Girosi F., Networks for approximation and learning, *Proc. IEEE*, vol. 78, no. 9, pp. 1481-1497, Sep. 1990.