

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,  
управління та адміністрування

Кафедра автоматизованих систем  
моніторингу навколишнього  
середовища та інформатики

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

на тему : «Дослідження застосування лінійної штучної нейронної мережі для  
екстраполяції ряду спостережень температури повітря»

Виконав: студент 2 курсу групи МКМ-20  
спеціальності 122 «Комп'ютерні науки»  
Тимченко Владислав Сергійович

Керівник: Перелигін Борис Вікторович,  
к.техн.н., доцент

Консультант:

Рецензент: д.ф.-м.н., професор  
Ковальчук Володимир Володимирович

Одеса 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Комп'ютерних наук, управління та адміністрування  
Кафедра Автоматизованих систем моніторингу навколишнього середовища та інформатики  
Рівень вищої освіти магістр  
Спеціальність 122 Комп'ютерні науки  
(шифр і назва)

ЗАТВЕРДЖУЮ  
ТВО завідувача кафедри АСМНСІ  
Перелигін Б.В.  
“ 28 ” жовтня 2021 року

**З А В Д А Н Н Я**  
**НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

ТИМЧЕНКО ВЛАДИСЛАВ СЕРГІЙОВИЧ

(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження застосування лінійної штучної нейронної мережі для екстраполяції ряду спостережень температури повітря

керівник роботи: Перелигін Борис Вікторович, к.техн.н., доцент,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджений наказом вищого навчального закладу від 18 жовтня 2021 року № 216,,С”

2. Строк подання студентом роботи: 10. 12. 2021 р.

3. Вихідні дані до роботи: ряд спостережень за температурою повітря на метеорологічній станції Одеса, лінійна нейронна мережа системи комп'ютерної математики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Аналіз застосування штучних нейронних мереж для екстраполяції даних
2. Особливості побудови і навчання лінійної штучної нейронної мережі
3. Аналіз вихідного ряду спостережень температури повітря
4. Розробка алгоритму і програми екстраполяції значень температури повітря на основі лінійної штучної нейронної мережі. Аналіз результатів
5. Управління проектом

5. Перелік графічного матеріалу: 1. Презентація роботи

## 6. Консультанти розділів роботи:

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|--------|-------------------------------------------|----------------|------------------|
|        |                                           | завдання видав | завдання прийняв |
|        |                                           |                |                  |
|        |                                           |                |                  |

7. Дата видачі завдання: \_\_\_\_\_ „28” жовтня 2021 року \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів магістерської роботи                                                                                  | Термін виконання етапів роботи | Оцінка виконання етапу |                       |
|-------|--------------------------------------------------------------------------------------------------------------------|--------------------------------|------------------------|-----------------------|
|       |                                                                                                                    |                                | у %                    | за 4-х бальною шкалою |
| 1     | Одержання завдання на виконання магістерської роботи                                                               | 28.10.2021                     |                        |                       |
| 2     | Пошук та підбір літератури та інших джерел інформації                                                              | 02.11.2021                     |                        |                       |
| 3     | Проведення аналізу предметної області і написання першого розділу пояснювальної записки до магістерської роботи    | 09.11.2021                     |                        |                       |
| 4     | Проведення аналізу предметної області і написання другого розділу пояснювальної записки до магістерської роботи    | 15.11.2021                     |                        |                       |
| 5     | Проведення аналізу предметної області і написання третього розділу пояснювальної записки до магістерської роботи   | 16.11.2021                     |                        |                       |
| 6     | Проведення аналізу предметної області і написання четвертого розділу пояснювальної записки до магістерської роботи | 21.11.2021                     |                        |                       |
| 7     | Рубіжна атестація                                                                                                  | 22-26.11.2021                  |                        |                       |
| 8     | Проведення аналізу предметної області і написання п'ятого розділу пояснювальної записки до магістерської роботи    | 30.11.2021                     |                        |                       |
| 9     | Виготовлення презентації                                                                                           | 02.12.2021                     |                        |                       |
| 10    | Друківання пояснювальної записки                                                                                   | 03.12.2021                     |                        |                       |
| 11    | Одержання висновку керівника магістерської роботи                                                                  | 04.12.2021                     |                        |                       |
| 12    | Проходження нормативного контролю                                                                                  | 04.12.2021                     |                        |                       |
| 13    | Перевірка на оригінальність тексту                                                                                 | 05.12.2021                     |                        |                       |
| 14    | Переплетіння пояснювальної записки                                                                                 | 07.12.2021                     |                        |                       |
| 15    | Здача роботи на кафедру та одержання висновку кафедри про допуск магістерської роботи до захисту                   | 08.12.2021                     |                        |                       |
| 16    | Одержання рецензії                                                                                                 | 09.12.2021                     |                        |                       |
| 17    | Здача готової магістерської роботи і документів секретарю АК                                                       | 10.12.2021                     |                        |                       |
|       |                                                                                                                    |                                |                        |                       |
|       |                                                                                                                    |                                |                        |                       |
|       | <b>Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)</b>                               |                                |                        |                       |

Студент \_\_\_\_\_ Тимченко В.С.  
( підпис ) (прізвище та ініціали)

Керівник роботи \_\_\_\_\_ Перелигін Б.В.  
( підпис ) (прізвище та ініціали)

## АНОТАЦІЯ

Представлена магістерська робота Тимченка Владислава Сергійовича, на тему «Дослідження застосування лінійної штучної нейронної мережі для екстраполяції ряду спостережень температури повітря».

В сучасному прогнозуванні одним з найпопулярніших методів є синоптичний метод. Недоліком синоптичного методу прогнозування є людський фактор, тому для покращення прогнозів було вирішено використати лінійну нейронну мережу.

Мета роботи полягає в дослідженні застосування лінійної штучної нейронної мережі для екстраполяції ряду спостережень температури повітря.

Для досягнення зазначеної мети необхідно вирішення наступних завдань:

- створення та навчання нейронної мережі;
- створення програми;
- побудова графіків результату та їх дослідження.

Як результат було досліджено застосування лінійної штучної нейронної мережі для екстраполяції ряду спостережень температури повітря.

Результати дослідження можна використовувати в метеорології при прогнозуванні температури, використовуючи лінійну штучну нейронну мережу.

Такі знання можуть використовуватись у флоті чи авіації при прогнозуванні погоди на різні проміжки часу.

Магістерську роботу викладено на 76 сторінках, має 44 рисунка, 30 формул, 1 додаток та 29 літературних посилань.

**КЛЮЧОВІ СЛОВА:** спостереження, прогнозування, метеорологія, клімат, лінійна штучна нейронна мережа, температура, екстраполяція ряду.

## SUMMARY

The work of Vladyslav Serhiyovych Tymchenko master's on the topic "Study of the application of a linear artificial neural network for extrapolation of a number of air temperature observations" is presented.

In modern forecasting, one of the most popular methods is the synoptic method. The disadvantage of the synoptic forecasting method is the human factor, so it was decided to use a linear neural network to improve forecasts.

The aim of the work is to study the application of a linear artificial neural network for extrapolation of a number of observations of air temperature.

To achieve this goal, it is necessary to solve the following tasks:

- creation and training of a neural network;
- creating a program;
- construction of graphs of the result and their study.

As a result, the application of a linear artificial neural network to extrapolate a series of air temperature observations was investigated.

The results of the study can be used in meteorology to predict temperature using a linear artificial neural network.

Such knowledge can be used in the fleet or aviation in forecasting weather for different periods of time.

The master's work is presented on 76 pages, 44 figures, 30 formulas, 1 appendix and 29 literary references.

**KEY WORDS:** observation, forecasting, meteorology, climate, linear artificial neural network, temperature, series extrapolation.

## ЗМІСТ

|                                                                                                                                                 |    |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Вступ.....                                                                                                                                      | 6  |
| 1 Аналіз застосування штучних нейронних мереж для екстраполяції даних..                                                                         | 7  |
| 2 Особливості побудови і навчання лінійної штучної нейронної мережі.....                                                                        | 25 |
| 3 Аналіз вихідного ряду спостережень температури повітря .....                                                                                  | 44 |
| 4 Розробка алгоритму і програми екстраполяції значень температури повітря на основі лінійної штучної нейронної мережі. Аналіз результатів ..... | 53 |
| 5 Управління проектом.....                                                                                                                      | 69 |
| 5.1 Планування проекту .....                                                                                                                    | 69 |
| 5.2 Оцінка трудомісткості і термінів розробки програмного забезпечення                                                                          | 70 |
| 5.3 Організаційне забезпечення проекту .....                                                                                                    | 74 |
| Висновок .....                                                                                                                                  | 75 |
| Перелік джерел посилання .....                                                                                                                  | 76 |
| Додатки.....                                                                                                                                    | 79 |

## **ВСТУП**

Мета роботи полягає в дослідженні застосування лінійної штучної нейронної мережі для екстраполяції ряду спостережень температури повітря.

В першому розділі магістерської роботи розглянуті основи застосування штучних нейронних мереж для екстраполяції даних.

В другому розділі подана основа побудови і навчання лінійної штучної нейронної мережі.

В третьому розділі представлено опис вихідного ряду спостережень температури повітря.

В четвертому розділі представлена розробка, навчання та робота нейронної мережі та коду програми.

Заключна частина магістерської роботи містить інформацію про розробку проекту.

## **1 АНАЛІЗ ЗАСТОСУВАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ЕКСТРАПОЛЯЦІЇ ДАНИХ**

В роботі проаналізовано та описано можливе застосування штучних нейронних мереж в питаннях технічного прогнозування. Радіотехнічні науки з достатньою точністю можуть визначити розподіл усіх розвиток сигналу, використовуючи поняття екстраполяції. Доказовим і оптимальним є застосування нейронних мереж в екстраполяції радіотехнічних сигналів. Процес модернізації і прогрес в науці провокує людини спробувати передбачити розподіл усіх розвиток майбутнього.

Метою цієї роботи є необхідність розібратися в аспектах нейронних мереж для визначення успіху функціонування екстраполяції недетермінованих широкосмугових радіотехнічних сигналів на нейронних мережах.

Проведено безліч досліджень в даних областях, які свідчать, що, в цілому, штучні нейронні мережі добре справляються зі складною екстраполяцією, складають високу конкуренцію статистичної обробки на основі теорії ймовірності.

Впровадження штучних нейронних мереж на сьогоднішній день має інтенсивний характер розвитку. З кожним роком цей напрямок стає актуальнішою, необхідність використовувати штучний інтелект зростає, цього вимагає глобальна автоматизація процесів, використовуваних в суспільстві.

В даний час існує безліч методів прогнозування. Деякі проаналізовані та порівняні з штучними нейронними мережами в цій роботі. Так, наприклад, показано, що прогнозування з використанням методу статистичної обробки, вимагає великої затрати сил, часу і грошей.

На відміну від нейронних мереж, які виключають недоліки статистичного методу і можуть бути автоматизовані і інтегровані в будь-які системи обробки сигналів і емпіричних залежностей [1].



Штучні нейронні мережі широко використовуються в екстраполяції радіотехнічних сигналів. У даній роботі розглянуті застосовувані програмні засоби, алгоритми і методи штучних нейронних мереж.

Кожен з нас хоч раз замислювався про майбутнє, починаючи з припущення про те, яка погода буде наступним днем і закінчуючи динамікою поведінки складних залежностей на місяці вперед, не кажучи вже про можливі прогнозах модернізації технологій, заглядаючи далеко в майбутнє. Пророкуванням займалися споконвіку. Багато людей хотіло дізнатися, що з ними станеться, які події їх очікують. Таке бажання підштовхує людей аналізувати те, що відбувається і передбачити майбутні явища. Тяжіння людини до нового лежить в основі рушійною сили процесу модернізації та прогресу, а конкуренція між один одним висуває найкращий варіант з високою якістю використання. Таким чином, сама людина виявляється фундаментальним двигуном науково-технічного прогресу.

Розуміння прогнозів обмежується розумними рамками, в яких є місце для власного вкладення і інновацій. На підставі попередньої поведінки залежностей вибудовується найбільш оптимальна і розподіл усіх змін спостережуваної величини, це і є екстраполяція.

Екстраполяція проводиться на штучних нейронних мережах (ШНМ). На сьогоднішній день напрямок «машинне навчання», до якого входять ШНМ, має інтенсивний характер розвитку. З кожним роком цей напрямок стає актуальнішою, необхідність використовувати штучний інтелект зростає, відбувається глобальна автоматизація процесів, де в спостереженні і контролюванні людиною ситуації немає необхідності.

В даний час існує безліч методів прогнозування. Деякі найбільш поширені способи далі будуть розцінені, проаналізовані і порівняні з ШНМ. Якщо розглядати прогнозування за допомогою статистичної обробки, то витрати сил, часу, грошей іде набагато більше, при цьому необхідно мати фахівця, що розбирається в цьому. З точки зору кореляційної складової,

розглядати вплив кожної тимчасової точки на інші і їх подальших комбінацій недоцільно.

Нейронні мережі виключають недоліки статистичного методу і можуть бути автоматизовані і інтегровані в будь-які системи обробки сигналів і емпіричних залежностей. З цього можна зробити висновок, що під впливом всіх мінусів, статистика поступається нейронним мережам.

Також використовують прогнозування на основі Фур'є-перетворення, ґрунтуючись на спектральних характеристиках, які можуть в достатній мірі описувати передісторію сигналу. «Склад» і принцип роботи штучних нейронних мереж також будуть описані [2].

Для реалізацій нейромережних обчислень існує безліч спеціалізованих програмних засобів.

Всі обчислення зручно проводити в пакеті прикладних програм Matlab, або в безкоштовному програмному забезпеченні Python, попередньо завантаживши всі необхідні бібліотеки. Крім того, використовують і інші програмні забезпечення, проте все впирається в наявність готових бібліотек з уже реалізованими функціями.

Актуальність теми дослідження безсумнівна, так як в даний час використовується в різних областях. Застосовується в прогнозах погоди, в бізнесі, особливо поширене в фінансових сферах, в радіолокаційних системах для супроводу маневро цілей, в електроніці для нелінійного моделювання та аналізу відмови мікросхем, автоматичних управління літальних об'єктів, і т. д.

Сучасні дослідження показують, що аналізують випадковий сигнал або якусь залежність переважно спектральним методом. Деякі дослідники наголошують на вузьку смугу низької частоти сигналу. Вони вважають, що глобальні перегини сигналу важливіші високочастотних флуктуацій. Дійсно, по спостереженню набору низькочастотних гармонійних складових можна відстежити динаміку зміни сигналу, але не для кожного завдання даний метод є правильним рішенням, так як може бути необхідність спостерігати найменші

зміни, які можуть в собі нести непрямі впливу зовнішніх або будь-яких інших факторів.

Розкладати початковий сигнал на гармоніки можна через фільтр низьких частот розклавши наявну залежність в спектр, набір більш простих функцій, їх необхідно скомбінувати і розділити весь спектр на кілька частин від 3 до 5. Кількість частин залежить від ширини спектра сигналу, якщо сигнал є широкосмуговим, то для точності весь спектр розбивається на більшу кількість частин. Ці частини поділяються на низькочастотні, середньої частоти, високочастотні.

У кожній з частин є набір функцій, які необхідно усереднити. Порахувати математичне очікування для кожної тимчасової точки. Далі всі розрахунки проводяться з декількома функціями, в залежності від того, на скільки груп був розбитий спектр [3].

Можливість екстраполювати далекі значення обмежена, так як буде зростати помилка, спрогнозовані точки будуть сильно відхилятися від справжніх. Тому дальність прогнозу у груп різна, відповідно низькочастотний сигнал можна прогнозувати на більшу кількість точок, ніж високочастотний. Як же після прогнозування відновити вихідний сигнал разом з екстраполяцією? Для цього необхідно вирахувати середньоквадратичне відхилення функцій середньої і високої частоти. Скласти все екстрапольовані точки і накласти середньоквадратичне відхилення на інтервали, де точки функцій середньої і високої частоти не прогнозувалися. Тим самим, буде спостерігатися інтервал, в якому з довірчою ймовірністю може знаходитися істинне значення функції.

Штучні нейронні мережі використовуються для різних завдань. Вони були засновані на припущенні про роботу нейронів нижчого рівня в мозку, але безпосереднього відношення до нього не мають, а запозичили тільки структуру і паралелізм. Нейронна мережа є нелінійною і являє собою алгоритмічний апарат, функціонуючий на квазінейронах, які реалізують

найпростіші алгебраїчні дії з подальшою їх підстроюванням.

Основний механізм роботи ІНС: під час навчання на вхід подається сигнал, який за допомогою збудження нейронів змінюється таким чином, щоб мінімально відрізнятись від цільової функції, яка є наступними значеннями вхідного сигналу. Під цільові значення або бажаний результат підганяються вхідні-це процедура називається навчання з учителем.

Вид нейронної мережі залежить від поставленого завдання, досвідченим шляхом виявлено, що для прогнозування вигідніше застосовувати двошарові мережі з затримками [4].

У процесі дослідження можна варіювати як кількість нейронів в першому шарі, так і затримку, починаючи з нульовою. Кількість нейронів у першому шарі підбирається вручну, але занадто велика кількість для даного завдання не прийнятне. Існують певні залежності між числом входів (кількість нейронів дорівнює числу входів) і розміром навчальної вибірки, проте ці відносини не використовувалися, але віддавалася перевага для використання експериментального методу з встановленням оптимального співвідношення «число входів - користь».

Розбиття навчальних даних також відіграє велику роль в продуктивності мережі і впливає на деякі аспекти, наприклад, побічно зачіпає алгоритм оцінки якості, а саме спосіб накопичення помилок мережі, має прямий вплив на кількість входів і виходів нейронної мережі. Формування навчальних вибірок ділиться на три типи: послідовне, групове, змішане.

У послідовному випадку входи і цільові вектор навчальних вибірок формується у вигляді масиву формату cell, це масив осередків з одним рядком, в якому кожен елемент зберігає реалізації для певного моменту часу. Кожен момент часу є динамічною величиною і містить в собі N реалізацій або вибірок, їх кількість задається вручну, наприклад, 10 значень функції, але дані величини не є послідовними в вихідній функції, а певними моментами часу в кожній з вибірок.

Припустимо, якщо є дві вибірки по два значення в кожній, то в послідовному вигляді в першій клітинці будуть лежати перші значення з кожної вибірки. У груповому випадку входи і цільовий вектор формуються у вигляді масиву формату `double`. Даний масив містить тимчасові послідовності для кожної реалізації, розділена функція на вибірки залишається незмінною. Розмірність реалізацій суто особистий вибір.

Найважливіша частина навчання нейронної мережі – це тренування, від неї безпосередньо залежить якість функціонування мережі. Тренування мережі містить в собі багато параметрів, що настроюються, кожен з яких необхідно вибрати і налаштувати, проаналізувавши всілякі варіанти.

Перший крок – ініціалізація мережі – це розстановка значень ваг синапсів і зсувів, відповідно це і задає початковий курс, правильність виконання завдання. Для кожного параметра, що набудовує існує значення за замовчуванням, в ініціалізації стандартний варіант - випадкова розстановка значень. Їх значення виставляються невеликими, щоб в процесі навчання не виникало паралічу мережі, до того ж запобігає ряд інших некоректних випадків.

Наприклад, якщо виставити всі значення однакові і великі, а функція має нерівномірність розподілу значень, то мережа не зможе правильно перебудувати вагові коефіцієнти. Також використовують інші функції ініціалізації, вони мають ряд переваг, але не завжди є найкращими в порівнянні зі стандартною, можливо тривалість навчання збільшиться на невелику кількість часу.

Другий крок – це вибір глобального алгоритму мінімізації помилки. Виходячи з виду навчальних даних накопичуються значення помилок мережі, помилки мережі характеризуються різницею цільових значень і отриманих на виході. Існує безліч алгоритмів, їх відносять до трьох груп, першого порядку з використанням градієнта, другого порядку - методи Ньютона і квазіньютонів.

У кожній з груп є набір алгоритмів, в яких присутні свої достоїнства і

недоліки. Розглянемо алгоритм градієнтного спуску першого порядку і квазіньютонів алгоритм Левенберга-Марквардта.

Обидва методи досліджувалися на двошаровій нейронної мережі. Алгоритм градієнтного спуску для поставленого завдання один з декількох з найшвидшою збіжністю, але він має маленьку стійкість під час навчання, проявляється це в відхиленні від зменшення помилки, кажучи іншими словами, різниця між вихідними і цільовими значеннями після кількох ітерацій зростає.

Тому параметр максимальної кількості помилок на перевірці безлічі необхідно виставляти вручну побільше, ніж за замовчуванням, щоб навчання не застрягло в локальному мінімумі. Також можуть виникати випадки довгої збіжності в кінці оптимізації - ця проблема повинна бути вирішена самої нейронною мережею, при виявленні «застрягання» навчання, мережа «струшує» свої значення, додаючи в них випадкову величину. Другий суто в рамках навчальної вибірки. Тому, якщо кількість вихідних даних мало, то не варто занадто точно їх описувати. Можливе рішення - це зменшення кількості нейронів в шарі.

При достатній розмірності навчальної вибірки варто використовувати алгоритм Левенберга-Марквардта, вихідні дані більш точно описують поведінку стрибків функції, ніж у градієнтного спуску. Але обидва варіанти відображають динаміку поведінки сигналу і амплітуду локальних мінімумів і максимумів. Переконатися в цьому можна буде трохи далі.

Постає дилема, які алгоритми використовувати в процесі навчання, безумовно конкретний алгоритм відсутній, необхідно досліджувати скомбіновані нутрощі мережі для кожного нового сигналу і визначити найкращий, єдиного стандарту для прогнозування в цілому не існує.

Постановка задачі та її рішення. В якості вихідних даних був представлений масив значень сигналу з тисячею моментами часу. Дану вибірку можна класифікувати як змінювані двовимірні координати крилатої

ракети. Завданням є передбачити сигнал в наступні моменти. Навчальна вибірка була стандартна з послідовним поданням даних, на кожні п'ятдесят значень доводилося одне еталонне, з подальшим перебором п'ятсот значень і кроком в одне значення [5].

На рис. 1.1 представлений прогнозований сигнал на сто точок разом з цільовим і довірчим інтервалом. За рис. 1 можна відстежити наростання відхилення прогнозованого сигналу (дана функція виділена маленькими колами) від істинних значень (функція з колами більшого радіусу) від істинних значень (функція з колами більшого радіусу).

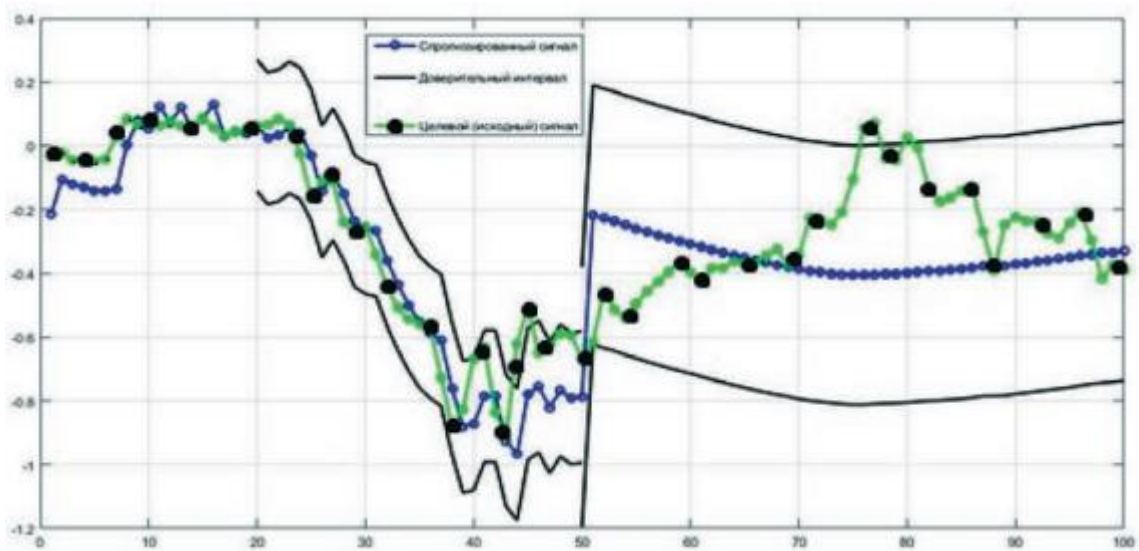


Рисунок 1.1 – Спрогнозований сигнал

Це обумовлюється неможливістю прогнозу далеких значень, тому зразкову і можливу зміну подальших точок показує довірчий інтервал (суцільні лінії), в якому з великою ймовірністю повинен коливатися «Хвіст» цільової функції.

Таким чином, починаючи з першої точки і по сорокову, можна вважати, що візуальний збіг сигналів є прийнятним, а наступні значення з наростаючою помилкою компенсуються обчисленим довірчим інтервалом.

На рис. 1.2 проілюстровано прогнозування сигналу з двома методами

тренування мережі без використання спектрального методу.

Як приклад на рис. 1.2 приведені сигнали з використанням двох вище описаних методів навчання.

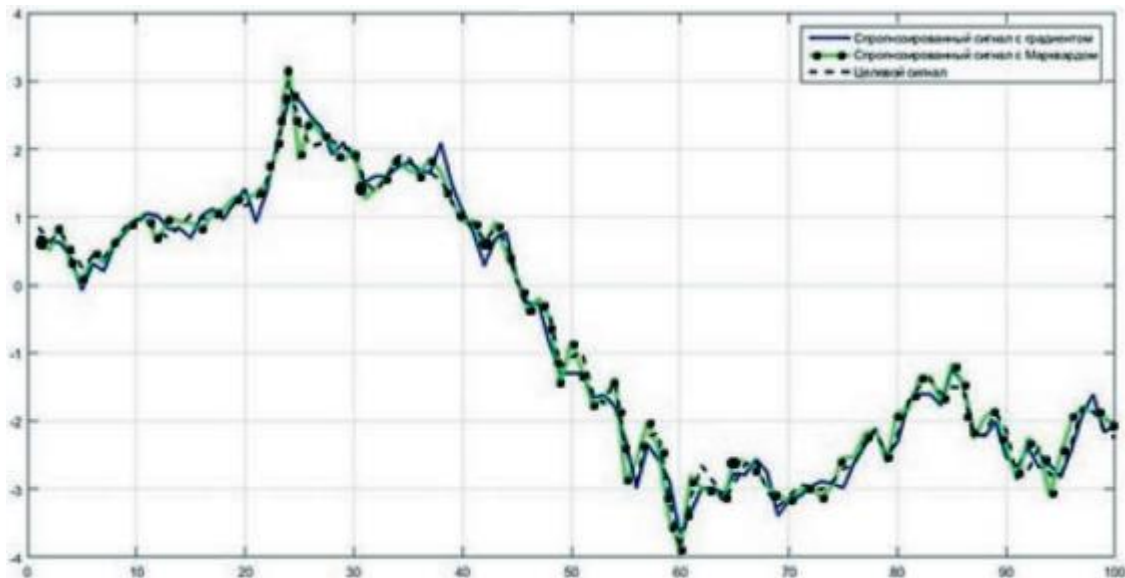


Рисунок 1.2 – Порівняння результатів прогнозування двома методами

З візуальної точки зору найменше відхилення від справжніх часових проміжків має залежність, навчається методом Левенберга-Марквардта, підтвердженням є чисельне значення стандартного відхилення, що менше величини відхилення методом градієнтного спуску.

Розглянуті моменти ШНС можна аналізувати і міняти в кращу сторону, можливо, для таких завдань потрібно унормувати дані в іншому діапазоні, використовувати іншу - власну функцію помилок, пробувати різні варіанти і прораховувати оптимальний. Ймовірно, ці зміни дозволять поліпшити результат прогнозування.

Тобто існує напрямок, в якому можна рухатися. Адже домогтися ідеальної архітектури і «нутроці» штучної нейронної мережі - це тільки перший крок, другий-це вдале навчання з підстроюванням всіх вагових значень, які дозволять і забезпечать миттєве прогнозування в режимі



реального часу.

Метою дослідження було розібратися які аспекти нейронної мережі розглянути, щоб переконатися в успіху функціонування екстраполяції недетермінованих широкосмугових радіотехнічних сигналів на нейронних мережах. Результати проведеного дослідження свідчать, що, в цілому, штучні нейронні мережі добре справляються зі складною екстраполяцією, складають високу конкуренцію статистичної обробки на основі теорії ймовірності [6].

Нейронні мережі мають великий недолік у порівнянні з лінійними методами наближення функції: вони не можуть екстраполювати. Це пов'язано з тим, що нейронна мережа може відобразити практично будь-яку функцію, регулюючи її параметри відповідно до представлених даних навчання (рис. 1.3). Для областей змінного простору, де немає даних про навчання, вихід нейронної мережі є ненадійним.

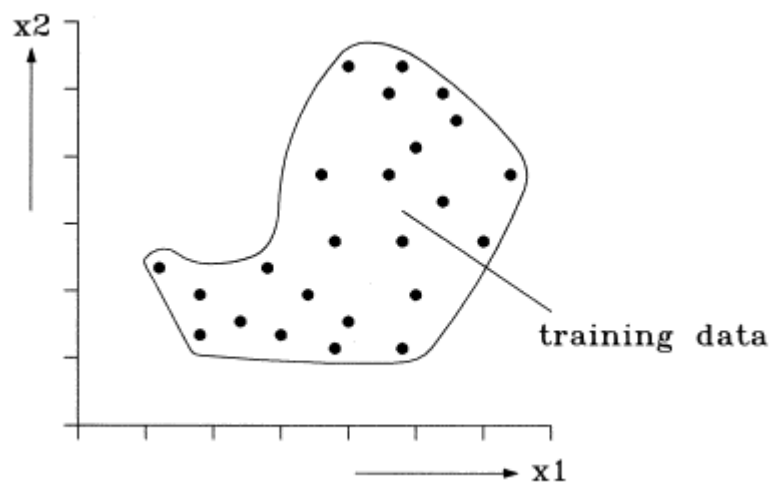


Рисунок 1.3 – Дані для навчання

В основному простір даних, який може бути оброблений навченою нейронною мережею, розділений на дві області:

- 1) область, де щільність даних навчального набору більше нуля,
- 2) будь-яка інша частина простору даних, де щільність навчальних даних

дорівнює нулю (або близько нуля).

Для невідомих точок даних, які потрапляють у першу область, ми використовуємо інтерполяцію, всі інші точки повинні бути оцінені екстраполяцією (рис. 1.4).

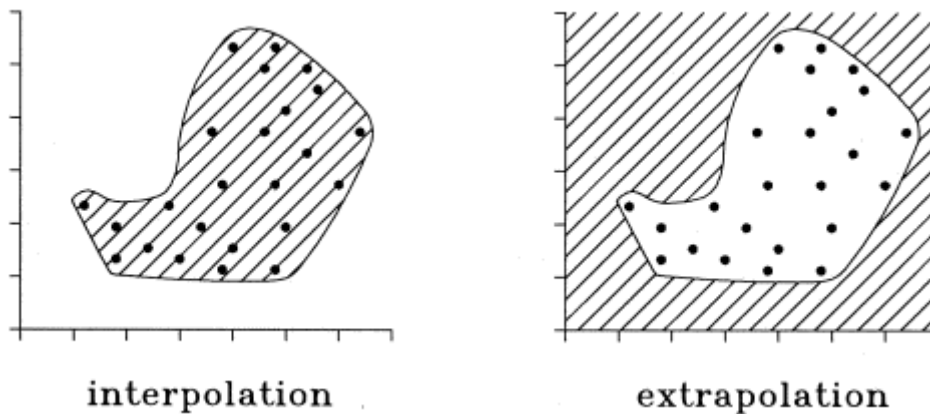


Рисунок 1.4 – Інтерполяція та екстраполяція

Для того, щоб подолати цю проблему, слід у якійсь формі записати діапазон змінного простору, де доступні дані навчання. В принципі, це можна зробити шляхом обчислення опуклої оболонки набору навчальних даних. Якщо невідомі дані, представлені мережі, знаходяться в цьому корпусі, вихід мережі можна вважати надійним. Однак концепція опуклої оболонки не є задовільною, оскільки ця оболонка є складною для розрахунку і не дає вирішення проблем, де простір вхідних даних увігнутий. Кращий метод, запропонований Леонардом, це оцінити локальну щільність навчальних даних за допомогою вікон Парзена. Це підходить для всіх типів мереж. Радіально-базисні мережеві функції забезпечують ще один елегантний, але простий метод виявлення областей екстраполяції.

Розглянемо приклад, що стосується продуктивності нейронних мереж в умовах екстраполяції. Щоб спростити налаштування, ми розглянемо одновимірні вхідні дані, які відносяться до однієї змінної відповіді. Дані

тренувань наведені нижче. Застосування 15 тренованих мереж до невідомих даних призводить до відповідей, які послідовні в тих областях, де були доступні дані навчання, і одночасно створюють довільні результати (рис. 1.5).

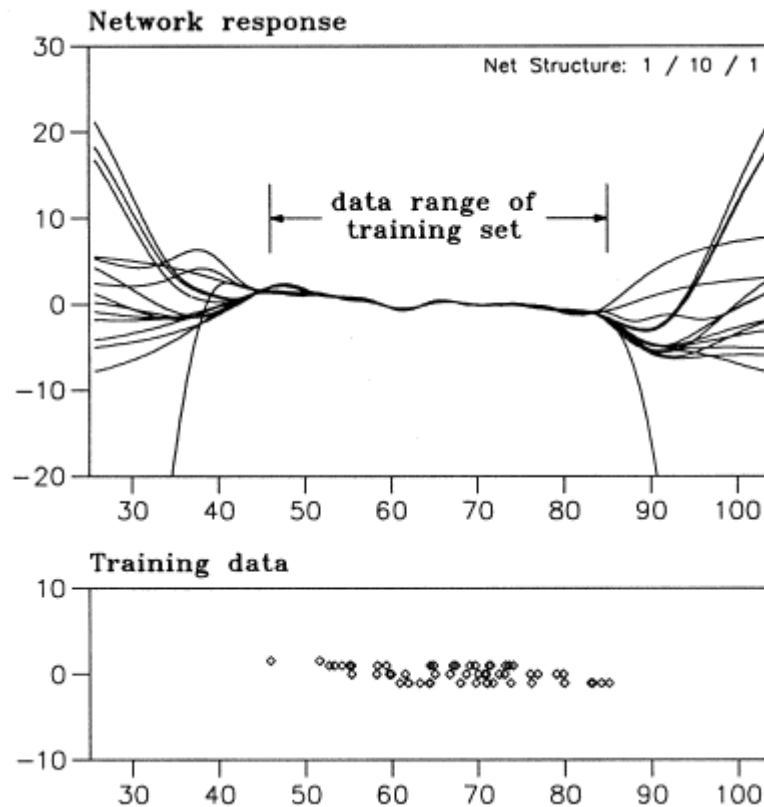


Рисунок 1.5 – Продуктивність нейронних мереж в умовах екстраполяції

Розглянемо екстраполяцію даних при застосуванні лінійної нейронної мережі на прикладі зондувального імпульсу.

Існує можливість передачі оцифрованого сигналу з витратоміра по каналу зв'язку до зовнішнього пристрій, що зчитує. В якості такого пристрою може виступати персональний комп'ютер з встановленим на ньому відповідним програмним забезпеченням. Прийнятий з пам'яті витратоміра масив оцифрованих даних при закритій засувці (рис. 1.6). Масив даних, прийнятий по цьому ж каналу, але при відкритій засувці (рис. 1.7).

На цих рисунках амплітуда вихідного зондувального імпульсу при закритій засувці прийнята рівною одиниці, а зашумлений сигнал при відкритій

засувці нормований до амплітуди вихідного зондувального сигналу (рис 1.6). По осі ординат відкладена тривалість часового інтервалу в мікросекундах (рис 1.7).

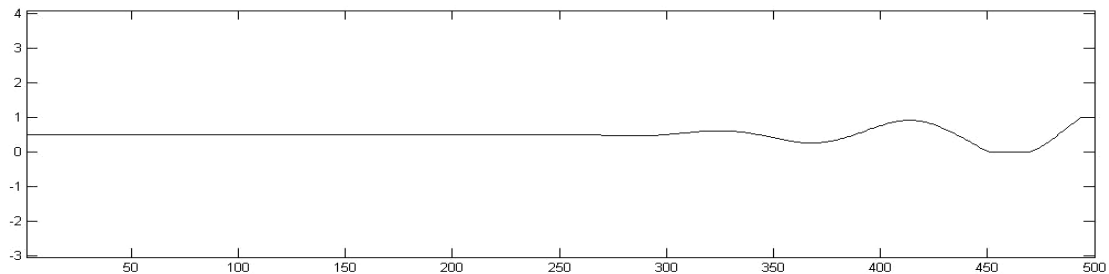


Рисунок 1.6 – Прийнятий зондувальний імпульс при закритій засувці

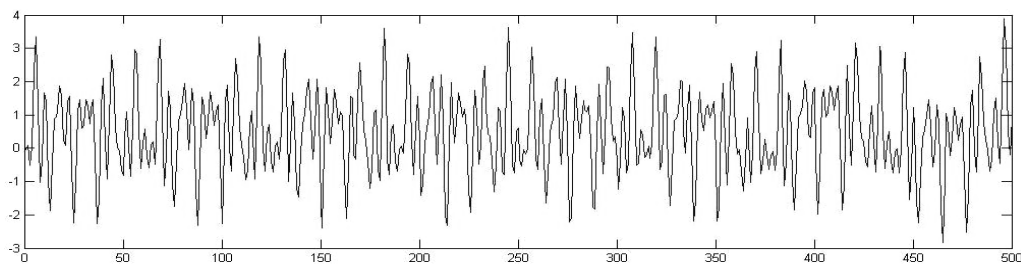


Рисунок 1.7 – Прийнятий зондувальний імпульс при відкритій засувці

Саме цей тип цифрових масивів з пам'яті витратоміра використовувався в якості даних, необхідних для процесу навчання і тестування штучних нейронних мереж.

Для моделювання процесу навчання і тестування нейронних мереж використовувалася система комп'ютерної математики.

Робота штучної нейронної мережі в якості адаптивного фільтра відбувається наступним чином. Вихідний масив даних за часом можна розбити на два підмасиви – підмасив, в якому міститься виключно шумовий фон (масив N), і підмасив, в якому присутні як шумовий фон, так і сигнал (масив SN). Процес роботи адаптивного фільтра складається з двох основних етапів –

етапу навчання і етапи безпосередньої роботи [7].

На етапі навчання використовується тільки підмасив, що містить виключно шумовий фон. Для підмасива  $N$  вводиться два вікна – вікно з вхідними даними  $W_i$  і вікно з цільовими даними  $W_o$ .

Нехай  $n$  – кількість точок всередині вікна вхідних даних  $W_i$  (довжина вікна),  $m$  – кількість прогнозованих точок в цільовому вікні  $W_o$ ,  $s$  – кількість точок, на яке виконується зрушення вікна. Таким чином, наприклад, для ряду виду  $\{3\ 1\ 2\ 4\ 1\ 1\ 2\ 4\ 1\ 3\ 4\ 1\ 4\ 2\ 1\}$  і  $n=4$ ,  $m=1$ ,  $s=5$  матимемо такі вектори:

$$3\ 1\ 2\ 4 \rightarrow 1$$

$$1\ 2\ 4\ 1 \rightarrow 3$$

$$4\ 1\ 4\ 2 \rightarrow 1$$

Дані з масиву  $W_i$  подаються на вхід нейронної мережі. У свою чергу, дані  $W_o$  є цільовими для алгоритму навчання. На виході нейронної мережі при подачі на її вхід шумових даних формується сигнал  $\hat{N}$ , який характеризує вибірку шуму  $W_i$  на її вході. введемо величину  $\hat{S}_l$  :

$$\hat{S}_l = N - \hat{N}, \quad (1.1)$$

де  $N$  – елемент з  $W_o$ , що характеризує реальний шум;  $\hat{N}$  – вихідний сигнал мережі.

Якщо нейронна мережа навчилася точно передбачати поведінку шумового сигналу, то  $N = \hat{N}$  і при цьому  $\hat{S}_l = 0$ . Таким чином, на етапі навчання величина  $\hat{S}_l$  використовувалася нами для оцінки того, наскільки добре нейронна мережа навчилася моделювати подається на її вхід шум  $N$ .

На етапі роботи використовується підмасив, що містить як шумовий фон, так і корисний сигнал. Як і на попередньому етапі, для підмасива  $SN$  також вводиться два вікна – з вхідними даними  $W_i$  і з цільовими даними  $W_o$ .

Однак тепер  $W_o$ , що містить як корисний сигнал, так і шум ( $S + N$ ), використовуються для виділення корисного сигналу. При цьому вихідний

сигнал нейронної мережі  $\hat{S}_w$  розраховується як:

$$\hat{S}_w = S + N - \hat{N}, \quad (1.2)$$

де  $S$  – корисний сигнал;  $N$  – реальний шумовий фон;  $\hat{N}$  – шумовий фон, змодельований нейронною мережею.

У разі, якщо на етапі навчання було досягнуто умова  $\hat{S}_l = 0$ , то  $N = \hat{N}$ ,  
Отже,

$$\hat{S}_w = S. \quad (1.3)$$

Таким чином, на етапі роботи величина  $\hat{S}_w$  безпосередньо використовується в якості результату фільтрації. величина  $\hat{S}_w$  тим точніше наближається до вихідного сигналу  $S$ , чим точніше нейронна мережа навчилася вирішувати завдання прогнозування шумового фону ( $N = \hat{N}$ ). У свою чергу ступінь корельованості відліків шуму може змінюватися в значних межах залежно від збудливою його причини, що також позначається на точність передбачення. Крім співвідношення сигнал-шум, нас цікавить точність виділення сигналу з шуму. Для оцінки ступеня подібності отриманого після обробки і вихідного сигналу використовували обчислення коефіцієнта взаємкореляції [8].

Мережа довготривалої короткостроковій пам'яті (Long shortterm memory - LSTM) - мережа, яка з'явилася в результаті розвитку RNN-мереж. Це цікава модифікація RNN-мереж, що дозволяє мережі не просто «тримати контекст», а й вмiє «бачити» довготривалі залежності. Тому LSTM підходить для прогнозування різних змін за допомогою екстраполяції (виявлення тенденції на основі даних), а також в будь-яких завданнях, де важливим є вмiння «тримати контекст», особливо добре підвладне для даної нейромережі.

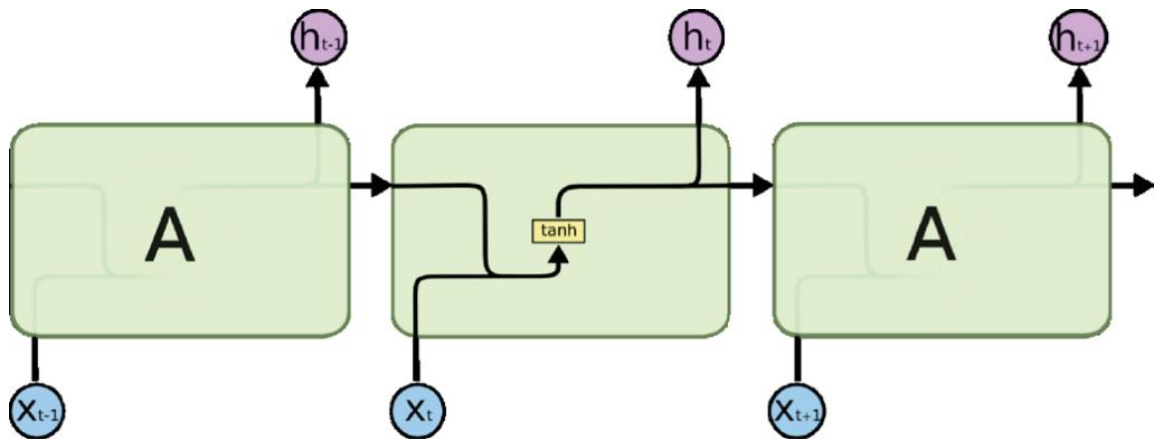


Рисунок 1.8 – Модулі рекурентної нейронної мережі

Форма рекурентних нейронних мереж є кілька повторюваних модулів. У стандартному вигляді ці модулі мають просту структуру. На рис. 1.8 зображена мережа, яка має в одному з шарів гіперболічний тангенс як функції активації.

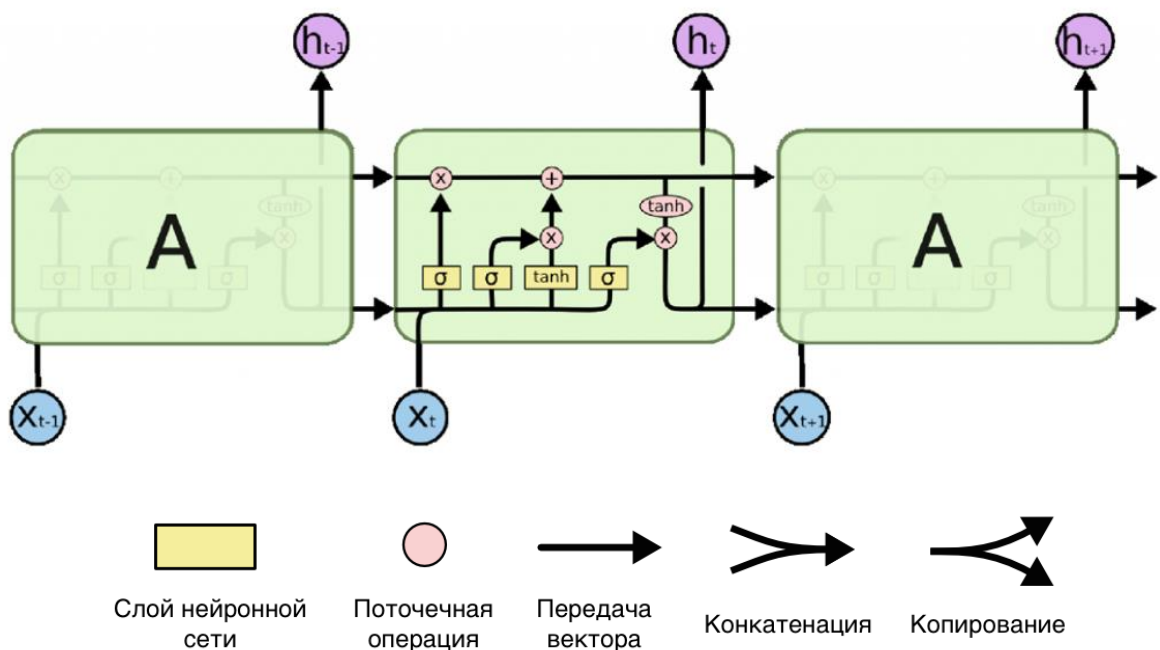


Рисунок 1.9 – Модуль LSTM-мережі

LSTM також має ланцюгову структуру. Відмінність полягає в тому, що мережа має чотири шари, а не один (рис. 1.9). Головною відмінністю LSTM-мережі є її клітинне стан (або стан осередку), яке позначається горизонтальною лінією у верхній частині діаграми і по якій проходить інформація (рис. 1.10). Це саме стан осередку нагадує стрічку конвеєра: вона проходить безпосередньо через весь ланцюжок, беручи участь в деяких перетвореннях. Інформація може як «текти» по ній, не наражаючись змін, так і бути піддана перетворенням з боку нейросеті. (Уявіть собі лінію контролю на конвеєрі: вироби просуваються на стрічці перед співробітником контролю, який лише спостерігає за ними, але якщо йому щось здається важливим або підозрілим, він може втрутитися).

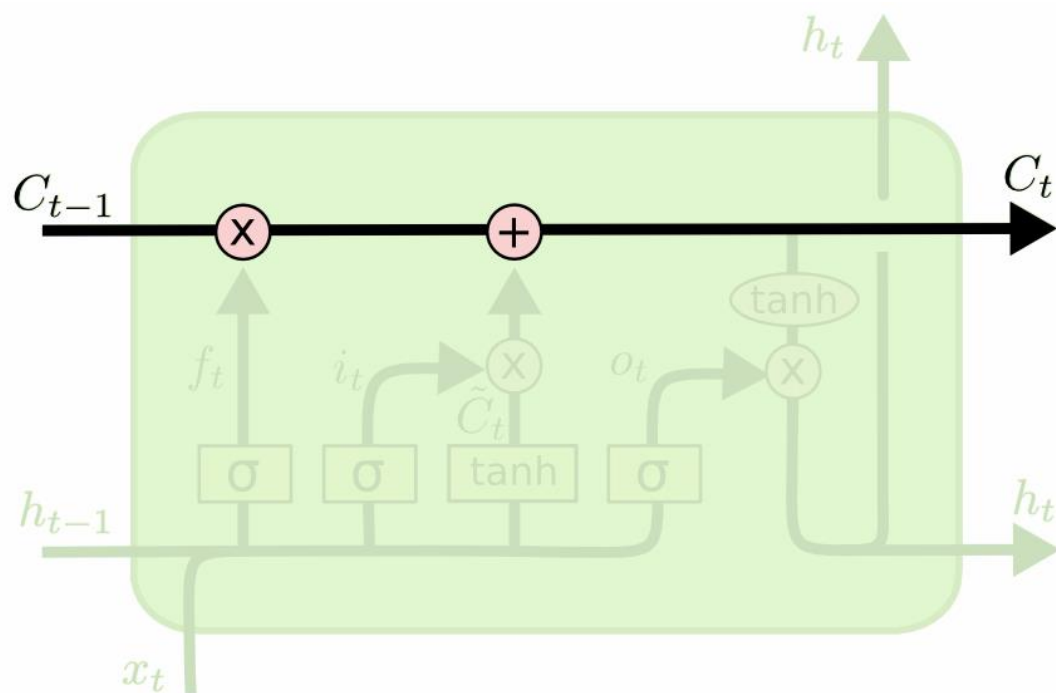


Рисунок 1.10 – Клітинний стан LSTM-модуля

LSTM має здатність видаляти або додавати інформацію до клітинного стану. Дана здатність здійснюється за допомогою вентилів, які регулюють процеси взаємодії з інформацією. Вентиль - можливість вибірково пропускати



інформацію. Він складається з сигмоїдного шару нейронної мережі і операції поточного множення (рис. 1.11).

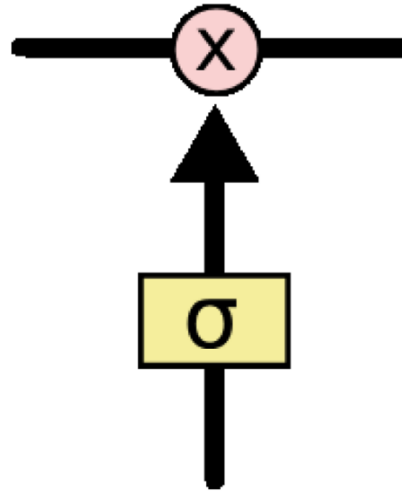


Рисунок 1.11 – Вентиль

Сигмоїдний шар (нейрони з сигмоїдальною функцією активації) подає на вихід числа між нулем і одиницею, описуючи таким чином, наскільки кожен компонент повинен бути пропущений крізь вентиль. Нуль - «не пропускати зовсім», один - «пропустити все». Таких «фільтрів» в LSTM-мережі кілька. Для чого мережі потрібна ця здатність: вирішувати, що важливо, а що ні? Уявіть собі таке: мережа переводить пропозицію і в якийсь момент їй належить перекласти з англійської мови на російську, скажімо, займенник, прикметник або дієприкметник. Для цього мережі необхідно, як мінімум, пам'ятати рід і / або число попереднього іменника, до якого перекладається слово відноситься. Однак як тільки ми зустрінемо нове іменник, рід попереднього можна забути. Звичайно, це дуже спрощений приклад, тим, він дозволяє зрозуміти труднощі прийняття рішення: яка інформація ще важлива, а яку вже можна забути. Інакше кажучи, мережі потрібно вирішити, яка інформація буде зберігатися в стані осередки, а потім - що на виході з неї буде виводити [9].

## **2 ОСОБЛИВОСТІ ПОБУДОВИ І НАВЧАННЯ ЛІНІЙНОЇ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ**

Здатність до навчання є фундаментальною властивістю мозку. Процес навчання може розглядатися як визначення архітектури мережі і налаштування ваг зв'язків для ефективного виконання спеціальної задачі. Нейромережа налаштовує ваги зв'язків по наявній навчальній множині. Властивість мережі навчатися на прикладах робить їх більш привабливими в порівнянні із системами, які функціонують згідно визначеній системі правил, сформульованої експертами.

Для процесу навчання необхідно мати модель зовнішнього середовища, у якій функціонує нейронна мережа – потрібну для вирішення задачі інформацію. По-друге, необхідно визначити, як модифікувати вагові параметри мережі. Алгоритм навчання означає процедуру, в якій використовуються правила навчання для налаштування ваг.

Існують три загальні парадигми навчання: "з вчителем", "без вчителя" (самонавчання) і змішана.

У першому випадку нейромережа має у своєму розпорядженні правильні відповіді (виходи мережі) на кожен вхідний приклад. Ваги налаштовуються так, щоб мережа виробляла відповіді як можна більш близькі до відомих правильних відповідей.

Навчання "без вчителя" не вимагає знання правильних відповідей на кожен приклад навчальної вибірки. У цьому випадку розкривається внутрішня структура даних та кореляція між зразками в навчальній множині, що дозволяє розподілити зразки по категоріях.

При змішаному навчанні частина ваг визначається за допомогою навчання зі вчителем, у той час як інша визначається за допомогою самонавчання.

У загальному використанні є багато правил навчання, але більшість з

цих правил є деякою зміною відомого та найстаршого правила навчання, правила Хеба. Дослідження різних правил навчання триває, і нові ідеї регулярно публікуються в наукових та комерційних виданнях. Представимо декілька основних правил навчання [10].

Правило Хеба. Опис правила з'явився у його книзі "Організація поведінки" у 1949 р. "Якщо нейрон отримує вхідний сигнал від іншого нейрону і обидва є високо активними (математично мають такий самий знак), вага між нейронами повинна бути підсилена". При збудженні одночасно двох нейронів з виходами ( $x_j$ ,  $y_i$ ) на  $t$ -тому кроці навчання вага синаптичного з'єднання між ними зростає, в інакшому випадку - зменшується, тобто

$$W_{ij}(k) = r x_j(k) y_i(k),$$

де  $r$  - коефіцієнт швидкості навчання.

Може застосовуватись при навчанні "з вчителем" і "без вчителя".

Правило Хопфілда. Є подібним до правила Хеба за винятком того, що воно визначає величину підсилення або послаблення. "Якщо одночасно вихідний та вхідний сигнал нейрона є активними або неактивними, збільшуємо вагу з'єднання оцінкою навчання, інакше зменшуємо вагу оцінкою навчання".

Правило "дельта". Це правило є подальшою зміною правила Хеба і є одним із найбільш загально використовуваних. Це правило базується на простій ідеї неперервної зміни синаптичних ваг для зменшення різниці ("дельта") між значенням бажаного та біжучого вихідного сигналу нейрона.

$$W_{ij} = x_j(d_i - y_i).$$

За цим правилом мінімізується середньоквадратична похибка мережі. Це правило також згадується як правило навчання Відрова-Хофа та правило

навчання найменших середніх квадратів.

У правилі "дельта" похибка отримана у вихідному прошарку перетворюється похідною передатної функції і послідовно пошарово поширюється назад на попередні прошарки для корекції синаптичних ваг. Процес зворотного поширення похибок мережі триває до досягнення першого прошарку.

При використанні правила "дельта" важливим є невпорядкованість множини вхідних даних. При добре впорядкованому або структурованому представленні навчальної множини результат мережі може не збігтися до бажаної точності і мережа буде вважатись нездатною до навчання.

Правило градієнтного спуску. Це правило подібне до правила "дельта" використанням похідної від передатної функції для змінювання похибки "дельта" перед тим, як застосувати її до ваг з'єднань. До кінцевого коефіцієнта зміни, що діє на вагу, додається пропорційна константа, яка пов'язана з оцінкою навчання. І хоча процес навчання збігається до точки стабільності дуже повільно, це правило поширене і є загально використовуване.

Доведено, що різні оцінки навчання для різних прошарків мережі допомагає процесу навчання збігатись швидше. Оцінки навчання для прошарків, близьких до виходу, встановлюються меншими, ніж для рівнів, ближчих до входу.

Навчання методом змагання. На відміну від навчання Хеба, у якому множина вихідних нейронів може збуджуватись одночасно, при навчанні методом змагання вихідні нейрони змагаються між собою за активізацію. Це явище, відоме як правило "переможець отримує все". Подібне навчання має місце в біологічних нейронних мережах. Навчання за допомогою змагання дозволяє кластеризувати вхідні дані: подібні приклади групуються мережею відповідно до кореляцій і представляються одним елементом.

При навчанні модифікуються синаптичні ваги нейрона-переможця. Ефект цього правила досягається за рахунок такої зміни збереженого в мережі

зразка (вектора синаптичних ваг нейрона-переможця), при якому він стає подібним до вхідного приклада. Нейрон з найбільшим вихідним сигналом оголошується переможцем і має можливість гальмувати своїх конкурентів і збуджувати сусідів. Використовується вихідний сигнал нейрона-переможця і тільки йому та його сусідам дозволяється коректувати свої ваги з'єднань:

$$W_{ij}(k+1)=W_{ij}(k)+r[x_j-W_{ij}(k)]$$

Розмір області сусідства може змінюватись під час періоду навчання. Звичайна парадигма повинна починатись з великої області визначення сусідства і зменшуватись під час процесу навчання. Оскільки елемент-переможець визначається по найвищій відповідності до вхідного зразку, мережі Кохонена моделюють розподіл входів. Це правило використовується в самоорганізованих картах.

Розвиток штучних нейронних мереж має прямий зв'язок з біологією.

Детальна структура мозку стала відома близько ста двадцяти років тому завдяки доктору Рамоні Каялу. Емпірично було показано, що тканини людського мозку побудовані з великої кількості вузлів одного типу - нейронів-з'єднаних з медом. Вони з'єднані між собою нервовими волокнами. Через них сигнали передаються хімічно і електрично між нейронами. Приблизно 1011 нейронів беруть участь приблизно в 1015 зв'язках, які передають імпульси. Ці з'єднання можуть бути довжиною більше метра.

Біологічний нейрон характеризується дуже складною структурною схемою.

Центральна частина нейрона (сома) з'єднується з центральною частиною іншого нейрона за допомогою деревоподібних процесів (дендриту). Аксон є найбільшим процесом, його завдання - забезпечити передачу сигналу активності від соми до інших нейронів.

Сигнали надходять в точках з'єднання, званих синапсами. Це невеликий

зазор, близько 200 нанометрів, на стику аксона і дендриту іншого нейрона.

Протоплазма нейрона негативно заряджена в спокої, з потенціалом близько 70 мВ. Заряд за рахунок отримання збудливих або гальмівних сигналів від дендритів деполяризований до значення 60 мВ, в результаті дифузії позитивно заряджених іонів натрію  $\text{Na}^+$  відбувається в сомі. Заряд нейрона різко піднімається до позитивного, а потім через аксон поширюється збудження інших нейронів. Після того, як ядро повертається до початкового стану [11].

Поширення збудження через нейрони мозку відбувається не одночасно. Частота роботи коливається від 1 до 100 Гц. Сигнал здатний поширюватися зі швидкістю від 0,5 до 2 м/с.

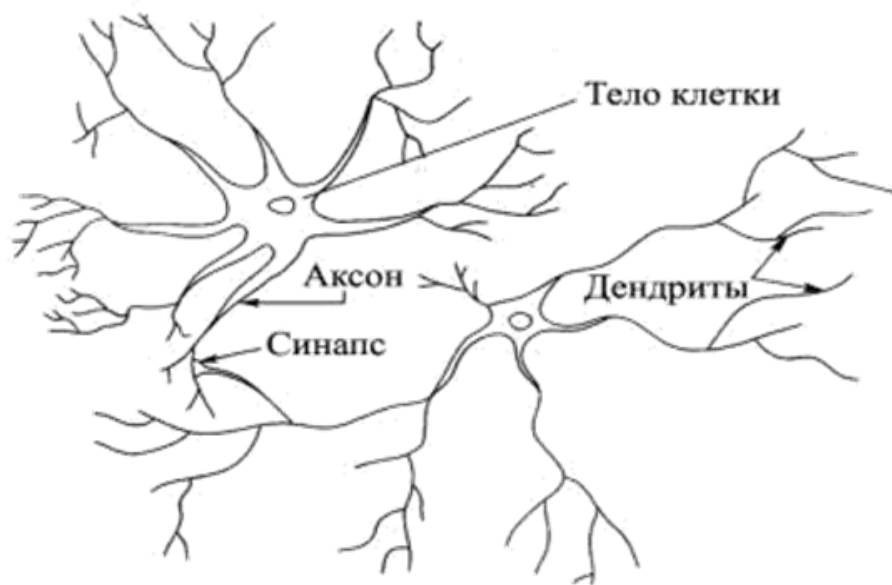


Рисунок 2.1 – Нейрон

Насправді функціональний ланцюг нейрона має безліч ускладнень і винятків. Більшість штучних нейронних мереж імітують тільки свої властивості, описані вище, так як знання про будову мозкової діяльності дуже обмежені.

Розробники, що займаються будівництвом нейронних мереж, повинні

використовувати структури, що відрізняються від біологічних знань. В результаті штучні нейронні мережі відмовляються від біологічної правдоподібності.

У середині XX століття з'явився термін «нейронна мережа». Одними з перших робіт в цій області були роботи Маккаллока і Піттса. У 1943 році була розроблена комп'ютерна модель нейронної мережі, в основі якої лягнули математичні алгоритми і теорії мозкової діяльності. Було висловлено припущення, що була висунута спрощена модель нейрона як пристрою, що працює з бінарними числами. Кожен елемент  $i = 1, 2, \dots, n$  має входи і виходи, які приймають значення 0 або 1. Ця модель називається "порогова логіка" [12].

Як і біологічний прототип, нейрони Маккаллока-Піттса змогли навчитися, регулюючи параметри, відповідальні за синаптичну провідність.

Ці системи (і багато інших подібних їм) називаються *перцептронами*.

Вплив інших нейронів визначає стан окремого нейрона як зваженої лінійної комбінації:

$$\sum w_{ij}n_j$$

де  $n_j$  є входом нейрона, а  $w_{ij}$  є відповідною вагою цього входу.

Потім до суми застосовується порогова функція форми:

$$g(\Sigma < 0) = 0 ; g(\Sigma \geq 0) = 1 .$$

Маккаллох і Піттс показали, що такі мережі можуть проводити довільні розрахунки. Єдиною невирішеною проблемою був підбір всіх шкал.

У 1949 році Д. Хебб припустив, що сила зміни синаптичних зв'язків впливає на здатність до навчання нейронної мережі. Згідно з алгоритмом, запропонованим Д. Хеббом, випробувана мережа спонтанно вчиться виконувати завдання без втручання, відбувається процес самонавчання.

У 1986 році Девід І. Румельхарт, Дж. Е. Хінтон і Рональд Дж. Вільямс

знову відкрив і значно розробив метод зворотного поширення помилки. Запропоновано ефективний алгоритм навчання більш просунутого багатошарового перцептрона (рис. 2.2).

Штучний нейрон імітує, до першого наближення, властивості біологічного нейрона.

Складовими є три типи елементів: множники (синапси), гадюки і нелінійні перетворювачі. Синапси забезпечують зв'язок між нейронами, множать вхідний сигнал на число, що характеризує силу з'єднання (вага синапсу). Гадюка додає сигнали, що надходять через синаптичні зв'язки з інших нейронів і вхідні сигнали ззовні. Функція активації або функція передачі нейрона - це вихід гадюки, який реалізує нелінійну функцію одного аргументу (рис. 2.3).

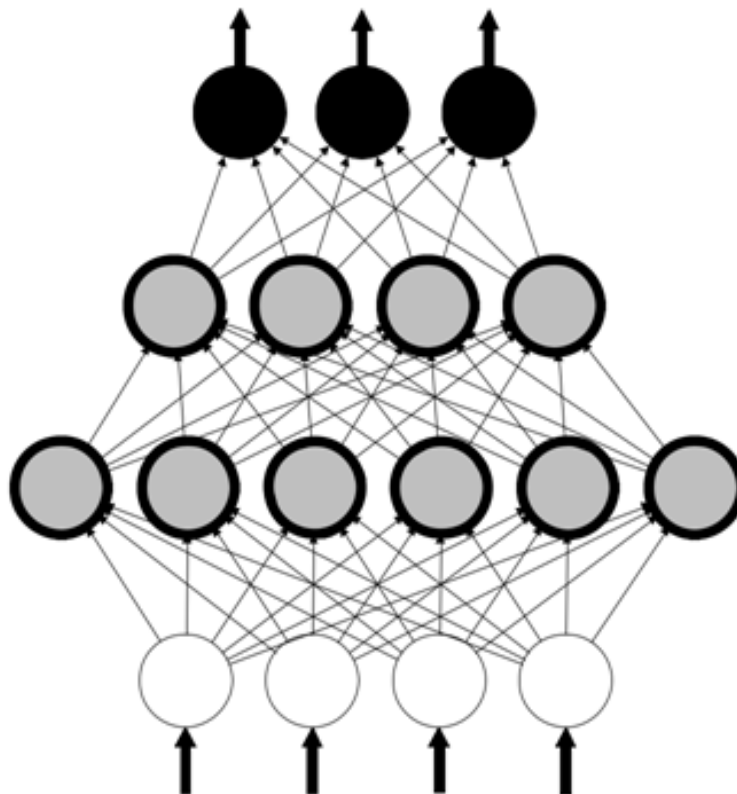


Рисунок 2.2 – Схема багатошарового перцептрона



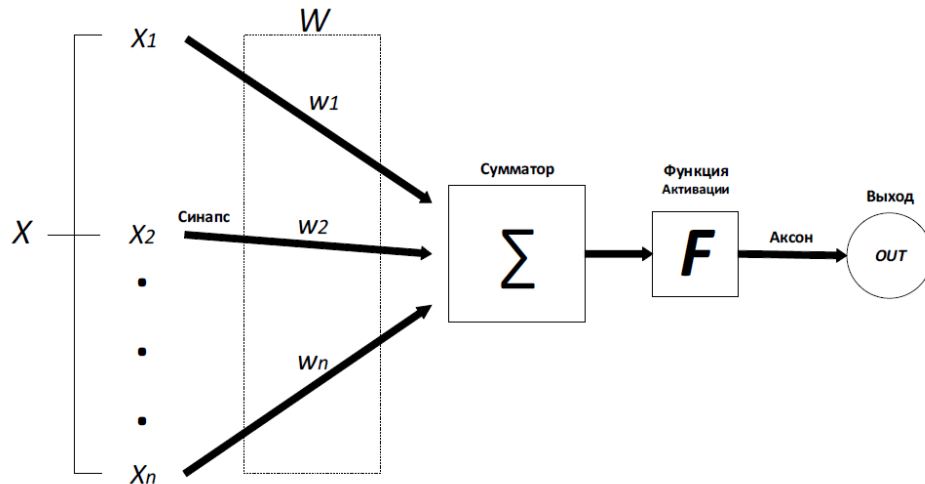


Рисунок 2.3 – Будова штучного нейрона

Математична модель нейрона:

$$s = \sum_{i=1}^n w_i x_i + b ; \quad y = f(s),$$

де  $w_i$  - вага синапсу;  $b$  - значення зсуву;  $s$  є результатом підсумовування;  $x_i$  є складовою вхідного вектора (вхідного сигналу);  $y$  - вихідний сигнал нейрона;  $n$  - кількість входів нейронів;  $f$  - нелінійне перетворення (функція активації) [14].

Позитивні ваги в синаптичних зв'язках називаються збудливими, а зв'язки з негативними вагами - інгібіторними. В цілому коефіцієнти вагових коефіцієнтів, а також вхідний сигнал і зміщення можуть приймати реальні або фіксовані значення.

Найбільш часто використовуваними функціями активації є:

- Один стрибок або жорстка порогова функція.
- Проста кумна лінійна функція. Якщо вхідне значення менше порогового значення, значення функції активації є мінімальним дозволеним, інакше

це максимально дозволено.

- Лінійний поріг або гістерезис.
- Проста кумна лінійна функція. Вона має дві лінійні області, де функція активації однаково дорівнює мінімально допустимому і максимально допустимому значенню і є розділ, в якому функція зростає строго монотонно;
- Сигмоїдальна функція або сигмоїд. Однією з найпоширеніших функцій активації є нелінійна логістична функція або сигмоїд (S-подібна

функція): 
$$f(s) = \frac{1}{1 + e^{-as}}.$$

- Гіперболічний тангенс. Гіперболічний тангенс є ще одним прикладом функції сигмоїдальної активації, визначеної наступним виразом: 
$$f(s) = th\left(\frac{s}{a}\right),$$
 де  $a$  - параметр, що впливає на нахил сигмоїдальної функції.

Розрізняють три типи нейронних мереж, в залежності від функцій:

- вхідні нейрони, до яких подається вектор, що кодує вхідний ефект або зображення зовнішнього середовища;
- вихідні нейрони, вихідні значення яких представляють виходи нейронної мережі;
- проміжні нейрони, які складають основу нейронних мереж.

Тип моделі нейронів в мережі залежить від її розташування. Відбувається деяка обробка інформації за рахунок перетворення вхідного вектору під час роботи мережі. Тип виконуваного перетворення даних також залежить від специфіки архітектури мережі.

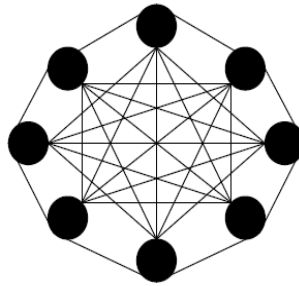
З точки зору топології можна виділити три основних типи нейронних мереж:

- 1) Повністю підключений - кожен нейрон передає вихідний сигнал собі та іншим нейронам;
- 2) Багатошаровий — набір нейронних мереж утворює шар з єдиним

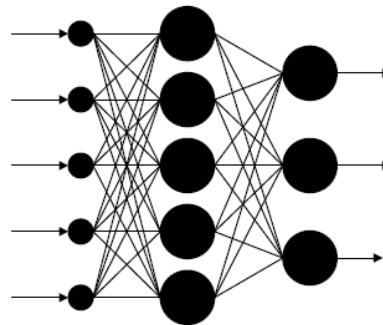
вхідним сигналом;

- 3) Слабо пов'язані - нейрони розташовуються в вузлах прямокутної або шестикутної решітки.

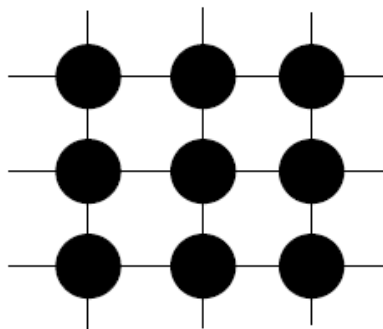
Архітектури нейронних мереж:



(а) Мережа з повним зв'язком



б) багат шарова мережа з послідовними з'єднаннями



в) мережа з низьким зв'язком

Багатошарові нейронні мережі можна розділити на наступні типи:

1. Монотонний. У одноманітних типах мережі кожен шар, за винятком вихідного шару, ділиться на два блоки. Збуджуючі блоки і блоки, які пригнічують, також з'єднуються один з одним або гальмівними, або збудливими зв'язками. Якщо тільки збуджуючі з'єднання ведуть від блоків одного типу до блоків іншого типу, то це означає, що вихідний сигнал є монотонно не спадною функцією, а якщо сигнал інгібіторний, то монотонно не зменшується. Монотонні мережі вимагають монотонного зв'язку між вхідним і вихідним сигналом;

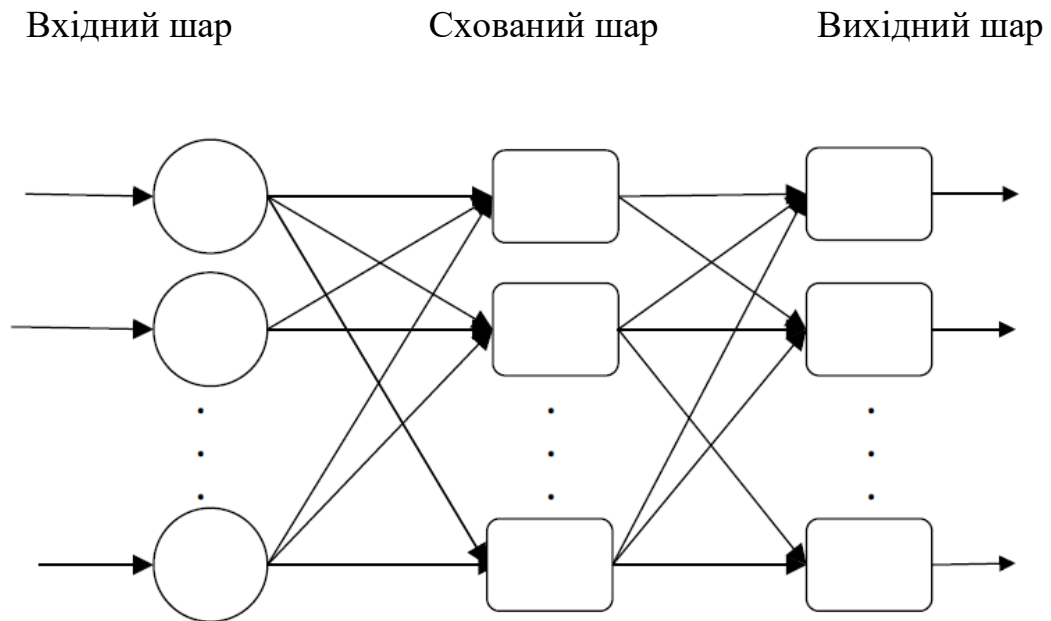


Рисунок 2.4 – Багатошарова (двошарова) пряма розподільна мережа

2. Мережі без зворотного зв'язку. Цей тип характеризується тим, що з першого прихованого шару сигнал передається в наступні шари, з подальшою передачею сигналу перекладачу;

3. Мережі зі зворотним зв'язком. У цьому типі мережі інформація з наступних шарів передається попереднім.

Існує класифікація нейронних мереж на синхронні і асинхронні. У синхронних мережах тільки один нейрон змінює стан в даний момент часу. У асинхронних мережах стан змінюється в колекції нейронів [13].

Підготовка вчителя-викладача характеризується тим, що для кожного вхідного вектора є цільовий вектор, який представляє необхідний вихід. Разом вони утворюють тренувальну пару. Зазвичай мережа тренується на ряді таких тренувальних пар.

Представлений вихідний вектор, вивід мережі розраховується і порівнюється з відповідним цільовим вектором, різниця (похибка) подається в мережу за допомогою зворотного зв'язку і ваги змінюються відповідно до алгоритму, спрямованого на мінімізацію похибки. Вектори тренувального набору представлені послідовно, помилки розраховуються, а ваги коригуються для кожного вектора, поки похибка по всьому тренувальному масиву не досягне прийнятно низького рівня.

Грунтуючись на вбудованих знаннях, викладач може формувати і передавати в навчену нейронну мережу потрібну відповідь, відповідну цьому вхідному вектору. Параметри мережі коригуються з урахуванням навчального століття і сигналу про помилку.

Сигнал про помилку - це різниця між бажаним сигналом і поточною реакцією нейронної мережі. Після завершення навчання викладач може бути вимкнений і дозволити нейронній мережі працювати з навколишнім середовищем самостійно.

Система представлена еталонним зображенням. Якщо результат розпізнавання відповідає вказаному, вага зв'язків не змінюється. Якщо штучна нейронна мережа неправильно розпізнає результат, то коефіцієнтам зважування дається приріст в напрямку підвищення якості розпізнавання.

Існує кілька підходів, одним з них є розробка наборів вихідних сигналів, що відповідають вхідним сигналам для кожного шару нейронної мережі. Другий варіант - динамічне регулювання вагових коефіцієнтів синапсів, під

час яких, як правило, найслабші з'єднання вибираються і змінюються невеликими обсягами, і зберігаються тільки ті зміни, які призвели до зниження похибки на виході всієї мережі. Третій варіант - поширення сигналів помилок від виходів нейронної мережі до її входів, в напрямку, протилежному прямому поширенню сигналів в нормальній роботі. цей алгоритм навчання називається процедурою зворотного розповсюдження помилок (error back propagation).

Алгоритм розповсюдження помилок. У 1986 році Румельхарт, Хінтон і Вільям запропонували алгоритм зворотного поширення помилки. Багатошарове навчання перцептронів базується на мінімізації функції помилки мережі  $E(w)$ . Похибка визначає відхилення бажаних виходів мережі  $t^n$  від отриманої  $y^n$ . Зазвичай функція  $E(w)$  визначається методом найменших квадратів:

$$E(w) = \frac{1}{2} \sum_n (t^n - y^n)^2 .$$

Суть методу полягає в тому, що рухаючись у напрямку, протилежному градієнту функції  $\Delta w_{ji}$ , ми наближаємося до мінімальної, можливо локальної, функції

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} ,$$

де  $0 < \eta < 1$  - встановлює швидкість навчання;  $w_{ji}$  — коефіцієнт зв'язування  $i$ -го нейрона шару  $n-1$  з нейроном  $j$   $n$  шару  $n$ .

Кожен нейрон обчислює зважену суму своїх входів:

$$a_i = \sum_n w_{ij} z_i$$

де  $z_i$  є входами нейрона і виходами попереднього шару нейронів відповідно до

рис. 2.5.

Вихід нейрона  $j$  - це перетворення суми  $a_j$  на порогову функцію  $g$ :

$$z_j = g(a_j).$$

Схема нейронної мережі алгоритму зворотного поширення:

$E(w)$  - це комплексна функція, для якої  $w$ :

$$w = w(a_j) .$$

Отже, за правилом розрахунку похідної комплексної функції:

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}}, \quad \frac{\partial a_j}{\partial w_{ji}} = \frac{\partial \sum w_{ji} z_i}{\partial w_{ji}} = z_i$$

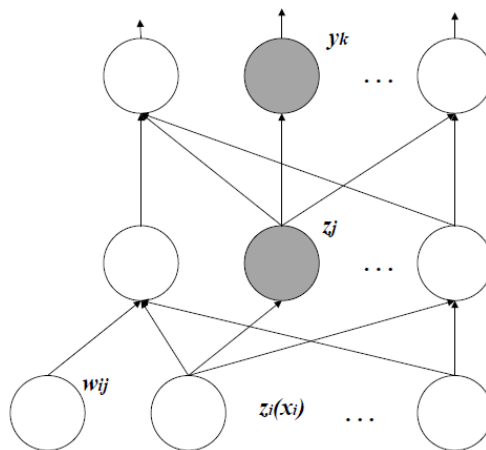


Рисунок 2.5 – Обчислення суми входів нейронів

Для обчислення градієнта на кожному шарі і добуток значення помилки попереднього  $i$ , розташованого поверх шару  $j$ , обчислюється за вхідним значенням шару:

$$\frac{\partial E}{\partial w_{ji}} = \delta_j z_i$$

Для вихідного шару  $\delta_k$  можна отримати:

$$\delta_k = \frac{\partial E}{\partial a_k} = \left| \begin{array}{l} a_k = \sum_j w_{jk} z_j \\ z_k = g(a_k) \end{array} \right| = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial a_k}.$$

Для логічного сигмоїда:

$$g(a) = \frac{1}{1 + e^{-a}},$$

така похідна буде мати досить просту форму:

$$g'(a) = g(a)(1 - g(a))$$

Другий множник  $\frac{\partial E}{\partial y_k}$  дається виразом:

$$\frac{\partial E}{\partial y_k} = y_k - t_k.$$

В результаті формула розрахунку похибки на вихідному шарі становить:

$$\delta_k = g'(y_k - t_k).$$

Для проміжного шару  $\delta_j$  виходить аналогічним чином:

$$\delta_j = \frac{\partial E}{\partial a_j} = \left| \begin{array}{l} a_k = \sum_j w_{jk} z_j \\ z_j = g(a_j) \end{array} \right| = \sum_k \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial z_j} \frac{\partial z_j}{\partial a_j}$$



Підсумовування відбувається над усіма  $k$ , до яких нейрон  $j$  посиляє сигнал відповідно до рис. 2.6.

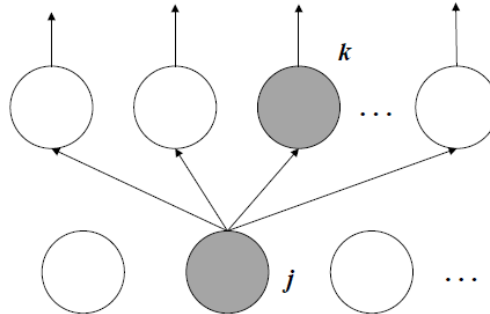


Рисунок 2.6 – Обчислення суми входів нейронів, до яких нейрон  $j$  посиляє сигнал

В результаті:

$$\delta_j = g'(a_j) \sum_k \delta_k w_{jk}$$

Помилка на кожному шарі обчислюється рекурсивно через значення помилок на попередніх шарах [15].

Алгоритм поширення помилок:

1. Перед запуском алгоритму ваги  $w_{ij}$  ініціалізуйте випадковими значеннями;
2. Постачаємо вхід перцептрона з вектором  $x_p$  з навчального зразка і отримуємо значення  $u_p$  на виході за відповідними формулами;
3. Обчисліть помилки  $\delta_k$  для виходів мережі за відповідною формулою;
4. Обчисліть помилки  $\delta_j$  для всіх прихованих шарів за відповідною формулою;
5. Знайдіть значення градієнта за відповідною формулою;
6. Відрегулюйте значення синаптичних ваг за відповідною формулою;
7. Обчисліть значення помилки. Якщо значення помилки не підходить, то

повторіть алгоритм.

Навчання без вчителів . Головною особливістю, яка робить навчання без вчителя привабливим, є його самостійність. Процес навчання, як і у випадку з підготовкою вчителів, полягає в коригці масштабів синапсів. Деякі алгоритми, однак, також змінюють структуру мережі, тобто кількість нейронів і їх взаємозв'язки, але такі перетворення більш правильно називаються більш широким терміном - самоорганізацією. Очевидно, що регулювання ваги синапсів може здійснюватися тільки на основі інформації про стан нейронів і вже наявних ваг). На цьому, зокрема, за аналогією з відомими принципами самоорганізації нервових клітин побудовані алгоритми навчання Хебба.

Найпростіша реалізація правила Хебба називається сигнальний метод:

$$\Delta w_{ij} = \eta z_i^{n-1} z_j^n$$

де  $\Delta w_{ij}$  - приріст ваги;  $z_i^{n-1}, z_j^n$  - вихідні значення нейронів  $i$  і  $j$ ;  $\eta > 0$  - встановлює швидкість навчання.

Більш досконалою варіацією правила Хебба є диференціальний метод:

$$\Delta w_{ij} = \eta (z_i^{n-1} - z_i^{*n-1})(z_j^n - z_j^{*n})$$

Тут додається  $z_i^{*n-1}$   $z_j^{*n}$  - вихідні значення  $i$  і  $j$  нейронів на попередній ітерації тренування.

Існує третій метод навчання – навчання з забуттям. Введемо поняття коефіцієнта забуття  $\gamma$  – тієї частини синаптичної ваги, яка «забувається» кожним нейроном в кожній ітерації [16]. Потім навчання виконується відповідно до виразу

$$\Delta w_{ij}(t+1) = (1 - \gamma)w_{ij}(t) + \eta \cdot z_i^{n-1} z_j^n$$

Алгоритм навчання нейронної мережі за правилом Хебба:

0. Ініціювати ваги  $w_{ij}$  випадковими змінними невеликої величини;
1. По черзі додаємо всі вектори від  $\{x^n\}$  до мережових входів і отримуємо вихідні значення  $z$  для всіх нейронів мережі;
2. Виходячи з отриманих значень  $z$ , ваги  $w_{ij}$  всієї мережі змінюються за формулами;
3. Приступайте до початку алгоритму, виключаючи нульовий крок, поки вихідні значення мережі не стабілізуються з заданою точністю.

Оскільки нейронна мережа - це мережа прямого поширення, рано чи пізно вона прийде в стабільний стан [17].

Загальна схема роботи струменних нейронних мереж. У періодичних нейронних мережах петлі зворотного зв'язку з'являються між нейронами від виходів назад до входів. Завдяки зворотному зв'язку виходи всієї мережі в різні моменти часу з однаковими мережевими входами можуть приймати довільні стани.

На першому кроці періодичної мережі, згідно з рис. 2.7, вхідні значення  $x_i$  подаються на вхід нейрона і розраховується вихід нейрона  $z$ . Отримане вихідне значення подається на вхід нейрона разом з іншими значеннями. Нове значення обчислюється на виході. Процес повторюється до тих пір, поки вхідне значення нейрона не має невеликої зміни.

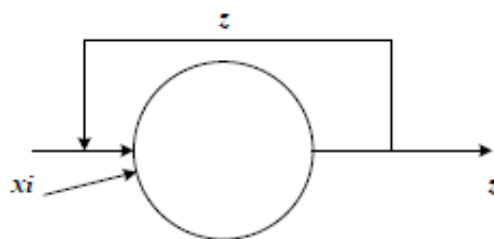


Рисунок 2.7 – Нейрон зворотного зв'язку

Періодичні нейронні мережі, для яких можна отримати виходи, які стабілізуються до певного значення, називаються стабільними, в іншому випадку - нестабільними. Нестабільні мережі мало використовуються для практичного застосування [18].

Хопфілда мережа. У 1982 році Д. Хопфілд запропонував стабільну періодичну нейронну мережу з ланцюгом відповідно до малюнка 7. Мережа працює тільки з кількостями  $\{-1; 1\}$ . Мережа має один шар налаштовуваних ваг  $W_{ij}$ .

Всі нейрони одного шару повертають свої виходи на вхід і входи всіх інших нейронів мережі через розподільників (а не нейрони) шару (0).

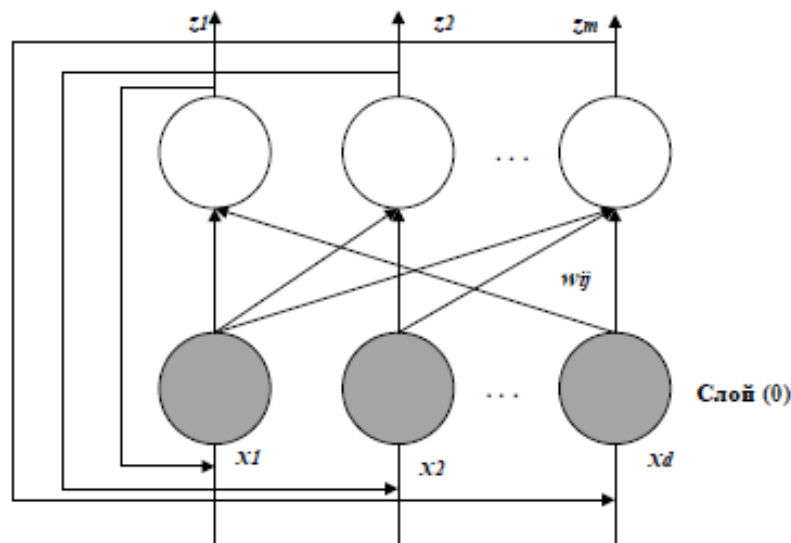


Рисунок 2.8 – Мережа Хопфілда

Кожен з нейронів виконує наступні дії:

- 1) обчислює зважену суму а її входів:

$$a_j = \sum_{i \neq j}^M (w_{ji} z_i) + x_j$$

- 2) до суми застосовується нелінійна порогова функція  $g(a)$  :

$$z_j = g(a) = \begin{cases} 1, & a_j > T_j \\ -1, & a_j < T_j \\ \text{не\,меняється} & a_j = T_j \end{cases}$$

Функціонування мережі відбувається таким чином, що вхідні сигнали подаються на вхід один раз, а подальша робота відбувається за рахунок повернених виходів. Зміна станів всіх нейронів відбувається одночасно.

Для  $n$  виходів, відповідно, є  $2n$  станів і  $n$ -мірний гіперкуб [19].

Стабільна вершина  $z$  для заданого вводу  $x$  подається матрицею вбудованих ваг  $w_{ji}$  і порогами  $T_j$ . Якщо вхідний вектор  $x$  частково спотворений, то мережа Хопфілда вступає в стан  $z$ .

### **3 АНАЛІЗ ВИХІДНОГО РЯДУ СПОСТЕРЕЖЕНЬ ТЕМПЕРАТУРИ ПОВІТРЯ**

Об'єктом дослідження став процес використання штучної нейронної мережі прямого поширення для прогнозування значень температури повітря. Процес застосування АНН для прогнозування включає в себе кілька етапів, а саме: 1) збір даних для навчання, 2) їх підготовка і нормалізація, 3) підбір мережевої топології, 4) експериментальний підбір мережевих характеристик, 5) експериментальний підбір параметрів навчання, 6) сам навчання, 7) перевірка адекватності навчання, 8) коригування параметрів, 9) підсумкове навчання, 10) вербалізація мережі для подальшого використання. У процесі дослідження не всі етапи були виконані, увага не приділялася останнім трьом. Це пов'язано з тим, що в ході дослідження доведення рішення проблеми до рівня розробки програмного пакета для прогнозування не було встановлено.

У процесі проведення досліджень необхідно було з'ясувати: 1) оптимальну довжину і кількість векторів навчання нейронної мережі, які забезпечують найкраще обґрунтування короткострокових прогнозів з авансом в 3 години, 1 день і 3 дні; 2) вплив на обґрунтування короткострокових

прогнозів вищезазначеного просування параметрів навчання даних нейронних мереж (довжина навчальних векторів, кількість навчальних векторів, типи вихідних даних); 3) вплив на обґрунтування прогнозів з вищенаведеним просуванням параметрів нейронної мережі (кількість її прихованих шарів, наявність обмеження в функціях активації нейронів прихованих шарів); 4) якість отриманих прогнозів.

Для досліджень значення температури повітря були обрані як дані через безперервність цих даних і видимість отриманих результатів. Дані - це довга 15-річна серія значень температури повітря, отримана під час регулярних восьмихвилинних спостережень на метеостанції 33837 Одеса з 1 лютого 2005 року по 31 грудня 2015 року (рис. 3.1, 3.2).

|    | A                                                      | B    | C | D | E |
|----|--------------------------------------------------------|------|---|---|---|
| 1  | # Метеостанция Одесса, Украина, WMO_ID=33837, выбо     |      |   |   |   |
| 2  | # Кодировка: UTF-8                                     |      |   |   |   |
| 3  | # Информация предоставлена сайтом "Расписание Погод    |      |   |   |   |
| 4  | # Пожалуйста, при использовании данных, любезно указ   |      |   |   |   |
| 5  | # Обозначения метеопараметров см. по адресу http://rp5 |      |   |   |   |
| 6  | #                                                      |      |   |   |   |
| 7  | Местное в                                              | Т    |   |   |   |
| 8  | 16.10.2014                                             | 15,6 |   |   |   |
| 9  | 16.10.2014                                             | 17,9 |   |   |   |
| 10 | 16.10.2014                                             | 20,5 |   |   |   |
| 11 | 16.10.2014                                             | 21,2 |   |   |   |
| 12 | 16.10.2014                                             | 18,0 |   |   |   |
| 13 | 16.10.2014                                             | 17,2 |   |   |   |
| 14 | 16.10.2014                                             | 16,8 |   |   |   |
| 15 | 15.10.2014                                             | 16,6 |   |   |   |
| 16 | 15.10.2014                                             | 16,6 |   |   |   |
| 17 | 15.10.2014                                             | 17,3 |   |   |   |
| 18 | 15.10.2014                                             | 19,3 |   |   |   |
| 19 | 15.10.2014                                             | 19,3 |   |   |   |
| 20 | 15.10.2014                                             | 16,4 |   |   |   |
| 21 | 15.10.2014                                             | 15,6 |   |   |   |
| 22 | 15.10.2014                                             | 15,5 |   |   |   |
| 23 | 14.10.2014                                             | 15,3 |   |   |   |
| 24 | 14.10.2014                                             | 15,2 |   |   |   |
| 25 | 14.10.2014                                             | 15,7 |   |   |   |
| 26 | 14.10.2014                                             | 16,7 |   |   |   |
| 27 | 14.10.2014                                             | 15,0 |   |   |   |

Рисунок 3.1 – Фрагмент даних про температуру повітря метеостанції 33837  
Одеса

При формуванні масиву вихідних даних відсутні значення температури повітря були інтерпольовані як середнє арифметичне сусідніх температурних значень. Таких прогалин в даних було всього 6, тому при довжині серії 31224 зразків їх редагування не мало істотного впливу на результат дослідження.

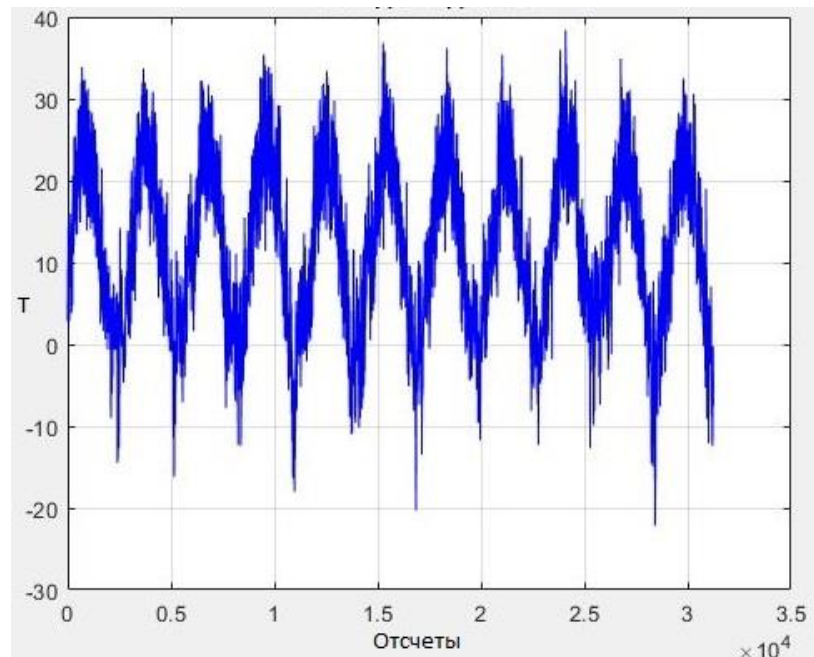


Рисунок 3.2 – Ряд значень температур, що використовуються для прогнозування значень температури повітря за допомогою штучної нейронної мережі (31224 зразків з 01.02.2005 по 31.12.2015)

У зв'язку з високою мінливістю метеорологічних величин в просторі і часі, конкретне значення будь-якого значення, зазначеного в прогнозі, слід розглядати як найбільш ймовірне значення, яке це значення буде мати протягом прогнозного періоду [20]. В кінці короткострокового прогнозу проводиться оцінка його успішності, яка ґрунтується на обґрунтуванні. Обґрунтування - це ступінь відповідності, з деякими встановленими допусками, прогностичних і фактичних метеорологічних величин і явищ. Обґрунтування прогнозу температури зроблено альтернативно. Якщо прогностична температура відрізнялася від фактичної не більше  $2,0^{\circ}\text{C}$ , то

обґрунтування прогнозу становить 100%, якщо різниця становить  $3,0^{\circ}\text{C}$  - 50%, якщо  $\geq 4,0^{\circ}\text{C}$  - 0% [Настанова, 2019] При проведенні досліджень обґрунтування вираховувалося аналогічно вищенаведеної методології, але без альтернативи, але кількісно, у вигляді співвідношення кількості обґрунтованих прогнозів (потрапляння в діапазон  $\pm 2^{\circ}\text{C}$ ) до загальної кількості прогнозів з заданим авансом.

Різні типи даних слід розуміти як фактичний існуючий ряд спостережень, так звані «сирі» дані, так звані «сирі» дані, і два його перетворення: по центру ряди - отримані шляхом віднімання від усіх значень ряду арифметичних засобів; і нормалізовані ряди – отримані шляхом ділення всіх значень ряду на максимальне модульне значення ряду. Всі серії спостережень були розділені на дві великі групи: 1-а група даних для навчання і 2-а група даних для прогнозування [21].

З першої групи даних, призначених для навчання, необхідні масиви вихідних даних формувалися наступним чином. Вся група була розділена на 3 частини. Перша частина використовувалася для навчання мережі (тренувальний набір). Друга використовувалася як тестовий (контрольний) набір для перевірки якості навчання.

Повторне повторення експериментів призводить до того, що контрольний набір починає відігравати ключову роль у створенні моделі, тобто стає частиною навчального процесу. Таким чином, його роль як самостійного критерію якості моделі ослаблена - при великій кількості експериментів існує ризик вибору мережі, яка дає хороший результат на контрольному наборі. Для того щоб надати остаточній моделі належну надійність, третя частина першої групи даних зробила резервний (тестовий) набір спостережень. Остаточна модель була протестована на даних з цього набору, щоб переконатися, що результати, досягнуті на навчальних і контрольних наборах, реальні. На основі отриманих даних показник якості навчання розраховувався за методикою, застосованою до розрахунку



обґрунтування прогнозу. екстрапольоване значення температури та наближення температурних значень тренувального масиву. Моделювання тут відноситься до процесу використання штучної нейронної мережі для прогнозування значень температури повітря.

З цієї групи даних всі масиви, необхідні для навчання (навчання, контролю і тестування), були сформовані методом, проілюстрованим на рис. 3.3.

Уздовж початкового масиву «сирих» (по центру або нормалізованих) даних, починаючи з заданого посилення (яке вибирається для аналізу певного сезону року), «вікно» зміщується по одному рахунку на кожному кроці з розміром, рівним заданій довжині вектора навчання (контролю, тесту) плюс один зворотний відлік. З температурних значень, що потрапили в «вікно», формувалися тренувальні, контрольні і тестові масиви, а з додаткових зразків формувалися цільові і обидва еталонні масиви (для набору управління і для тестового набору) [22].

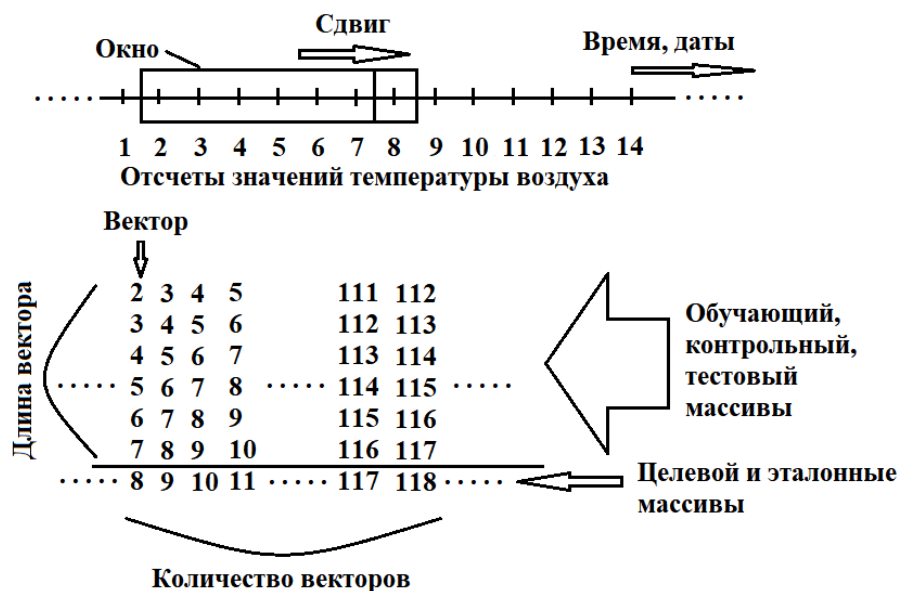


Рисунок 3.3 – Порядок формування масивів для навчання штучної нейронної мережі та прогнозування значень температур за її допомогою

Для навчання мережі був представлений тренувальний масив і

відповідний цільовий масив. Для контролю навчального, контрольного (або тестового) і довідкових масивів були представлені. В ході досліджень змінилася довжина і кількість навчальних векторів, а місце початку масиву було змінено для оцінки сезонного впливу даних на якість прогнозу.

Після навчання мережі з другої групи даних отримати прогноз в порядку, показаному на рис. 3.3 сформовано масив даних для прогнозування та довідковий масив для оцінки обґрунтування прогнозу. У мережі був представлений масив даних для прогнозування, на основі якого в мережі з'явилися екстрапольовані температурні значення (прогноз), яке порівнювали з еталонним значенням температури (істинне значення температури). Попередній прогноз в цій побудові масивів дає кількість векторів. При прогнозуванні на 3 години - це один вектор, при прогнозуванні на один день - це 8 векторів, а при прогнозуванні на три дні - це 24 вектори. У цих умовах єдине прогнозне моделювання на 3 години дало оцінку обґрунтування 0% або 100%, а при єдиному моделюванні прогнозу на добу і на три дні вже можна побудувати гістограму розподілу значення прогнозової похибки, оцінити стандартне відхилення і отримати значення оцінки обґрунтування прогнозу. При повторному моделюючому прогнозі кількість отриманих оцінок збільшується, з'являється можливість отримання статистичних оцінок обґрунтування, що підвищує надійність прогнозування.

З усіх парадигм і архітектур ШНМ для дослідження була обрана мережа прямого розподілу з пакету прикладного програмного забезпечення. Всі характеристики мережі (кількість входів - відповідає довжині тренувального вектору, кількості шарів, типу функції активації) були встановлені і змінені в рамках можливостей програми і використовуваного комп'ютера, а саме, кількість прихованих шарів варіювалися від одного до трьох, оскільки їх подальше збільшення недоцільно через складність мережі і збільшення часу навчання, що, втім, не призводить до поліпшення обґрунтування прогнозів. Функція активації штучних нейронів використовувалася двох типів: лінійна

без обмежень і з обмеженням сигмоїдального типу.

Оскільки загальне завдання дослідження включало в себе кілька конкретних завдань, ми проаналізуємо результати їх вирішення в порядку, зазначеному при постановці проблеми.

При визначенні оптимального, з точки зору обґрунтування прогнозів, була виконана довжина тренувального вектору і оптимальна кількість навчальних векторів нейронної мережі, багаторазове моделювання процедури навчання і прогнозування. При цьому розміри тренувального масиву (довжина векторів і їх кількість) підбиралися таким чином, щоб тренувальна процедура була завершена за час не більше однієї години. В іншому випадку значення короткострокового прогнозування з авансом в 3 години було втрачено, оскільки результат можна було отримати після настання прогнозного часу. При повторному моделюванні при повторному моделюванні оцінювалася якість навчання і обґрунтування всіх трьох прогнозів різних досягнень: а) тип вихідних даних ("сирий", центрований, нормалізований), б) коли кількість прихованих шарів змінювалася (1, 2, 3) і в) при зміні функції активації з лінійного на сигмоїдальний для прихованих і вихідних шарів. В результаті було отримано 72 тривимірних графіка. Уздовж осі абсцис кількість навчальних векторів переноситься, уздовж осі ординат - довжина векторів, а уздовж аплікаційної осі - значення параметра якості тренування або значення параметра вирівнювання прогнозу відповідного авансу. Кожна точка цих графіків вираховувалася на заданій довжині вектора, кількості векторів, кількості прихованих шарів ANN, типі функції активації і типі вихідних даних [24]. З математичної точки зору лівий графік на рис. 3.4 відображає якість мережевого наближення навчального масиву, а в правій - якість екстраполяції даних, на яких мережа не навчена. На правильному графіку нестабільність прогнозів добре видно при будь-якій довжині тренувальних векторів і з невеликою їх кількістю.

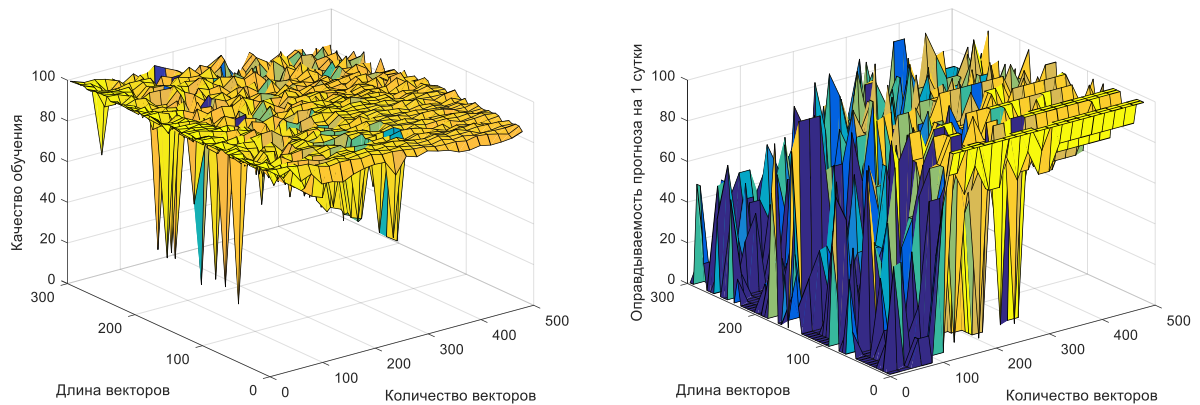


Рисунок 3.4 – Приклад відображення параметрів якості навчання та обґрунтування прогнозів

Аналіз всіх графіків показав, що наявність нелінійності (обмеження) в функції активації значно погіршує показник якості прогнозування - обґрунтування (рис. 3.5) і збільшує час навчання більш ніж в 2 рази. Тому при вирішенні такої проблеми нейрони мережі повинні мати повну функцію лінійної активації.

Крім того, збільшення кількості прихованих шарів мережі не покращується, але не погіршує якість прогнозування, обґрунтування істотно не змінюється (рис. 3.6), але архітектура мережі ускладнюється і з тим же алгоритмом навчання (алгоритм Левенберга-Марквардта в процедурі зворотного поширення помилки) час навчання множитья [25].

Тип вихідних даних не впливає на якість прогнозування, обґрунтування істотно не змінюється при заміні «сирих» вихідних даних центрованими або нормалізованими даними (рис. 3.7).

Описані вище дослідження дозволили визначити тип вихідних даних, обсяг вихідних даних, а також основні параметри прямого поширення ANN для прогнозування температурних значень.

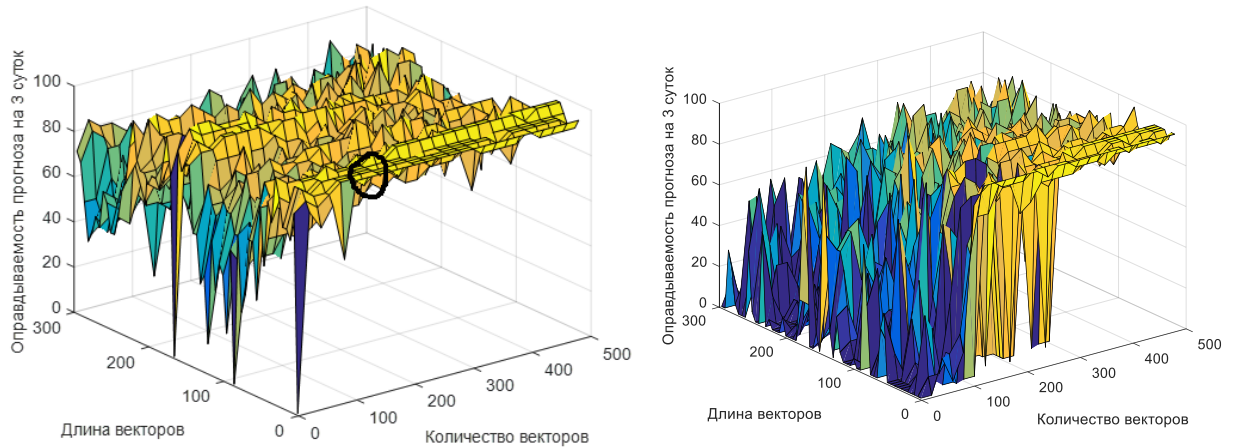


Рисунок 3.5 – Обґрунтування прогнозів в лінійній функції активації нейронів (ліворуч) і при наявності нелінійності в функції активації нейронів (праворуч) з однаковим типом вихідних даних і однаковою кількістю прихованих шарів

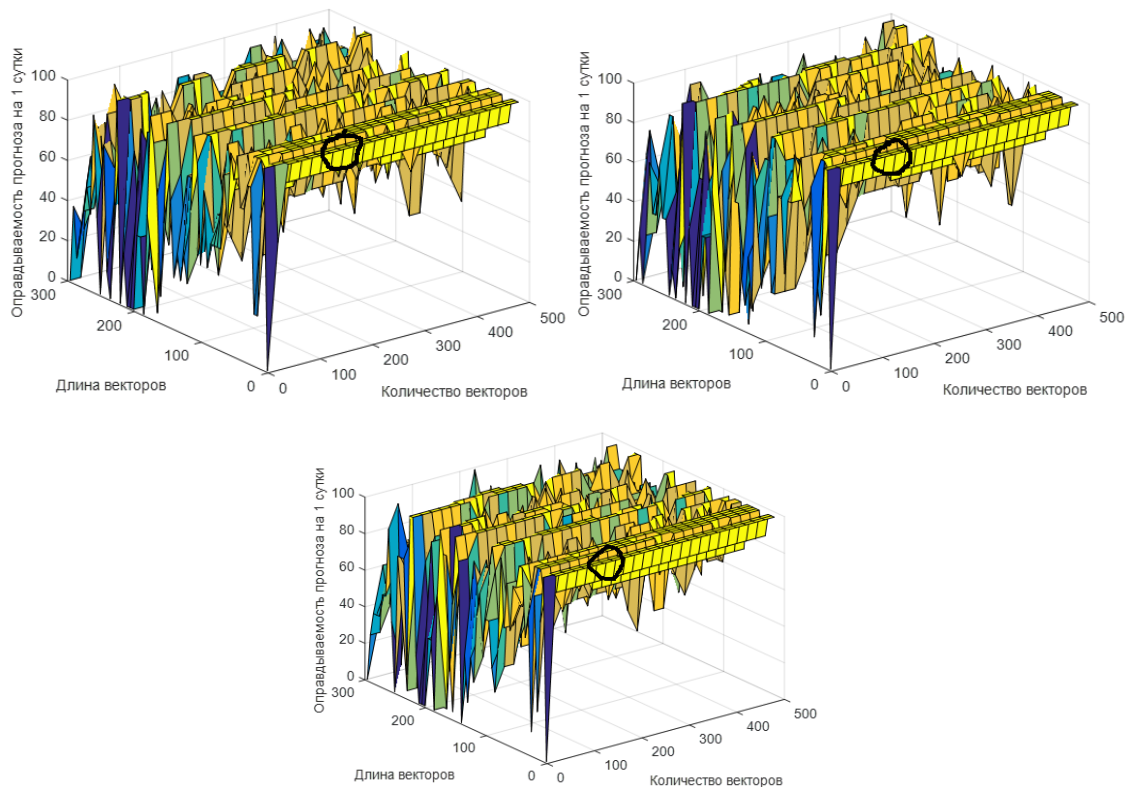


Рисунок 3.6 – Обґрунтування прогнозів при наявності одного прихованого шару (вгорі ліворуч), двох прихованих шарів (вгорі праворуч), трьох прихованих шарів (внизу)

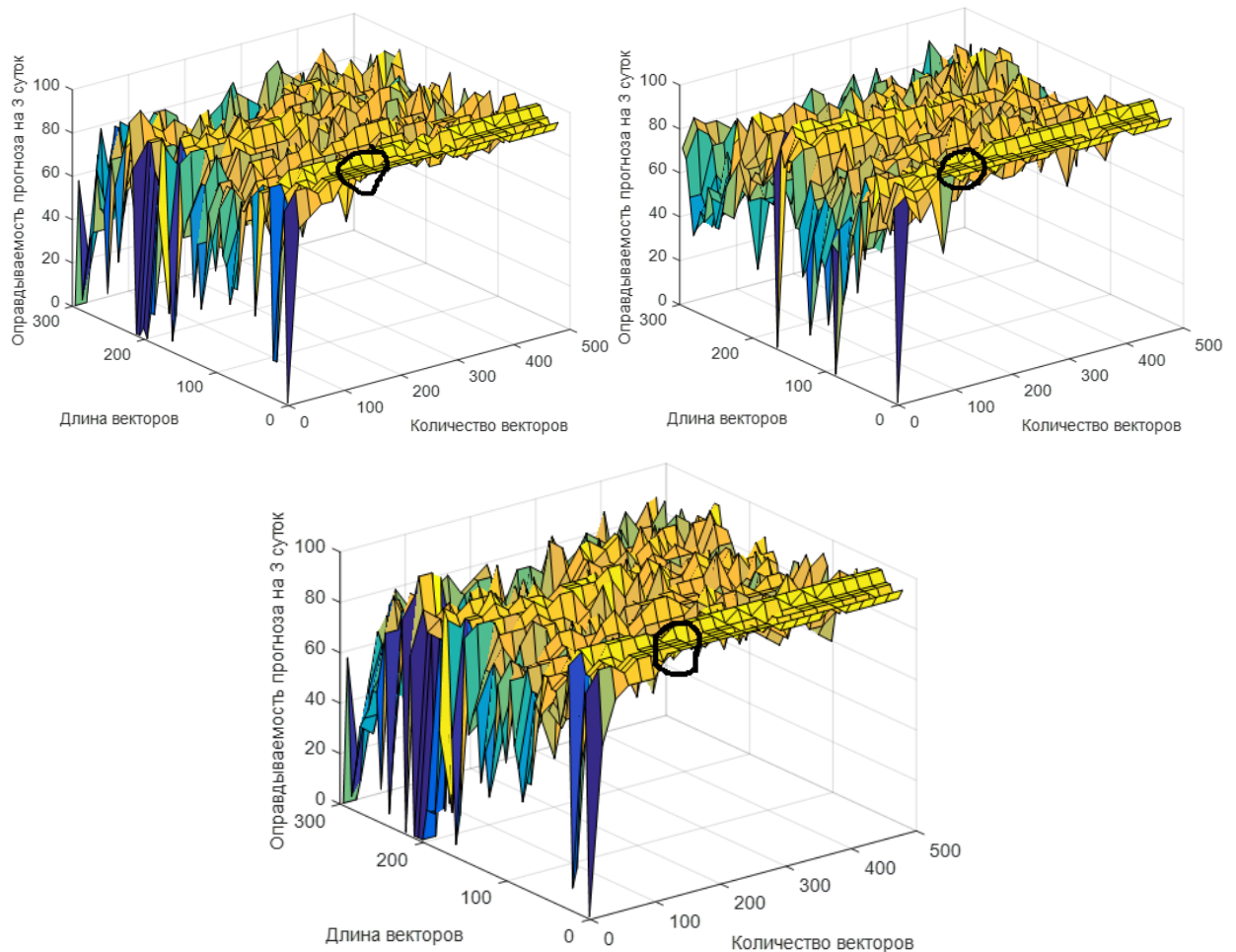


Рисунок 3.7 – Обґрунтування прогнозів для різних типів вихідних даних: "сирий" (вгорі ліворуч), центровий (вгорі праворуч), нормалізований (внизу)

#### **4 РОЗРОБКА АЛГОРИТМУ І ПРОГРАМИ ЕКСТРАПОЛЯЦІЇ ЗНАЧЕНЬ ТЕМПЕРАТУРИ ПОВІТРЯ НА ОСНОВІ ЛІНІЙНОЇ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ. АНАЛІЗ РЕЗУЛЬТАТІВ**

У системі комп'ютерної математики було розроблено та навчено штучну лінійну нейронну мережу на основі температурного ряду за період з 01.02.2005 по 31.12.2015.

Для коректної роботи програми спочатку було необхідно сформувати масив.

За допомогою функції `dlmread` було считано дані за період з 01.02.2005 по 31.12.2015:

```
D=(dlmread('T_31_12.txt'))';
s=length(D);
sr=mean(D);
mi=min(D);
ma=max(D);
m=max(abs(D));
```

За допомогою функції `mean` ми знаходимо середнє арифметичне значення елементів масиву, а `length` це кількість даних. Далі формуємо масив вхідних даних. Використовуючи функцію `plot` виводимо вхідні данні на екран у вигляді графіку.

```
ID=D
figure(1); plot(ID,'-b'); title('Исходные данные'); grid on;
```

Температура за період з 2005 по 2015 рік (рис. 4.1).

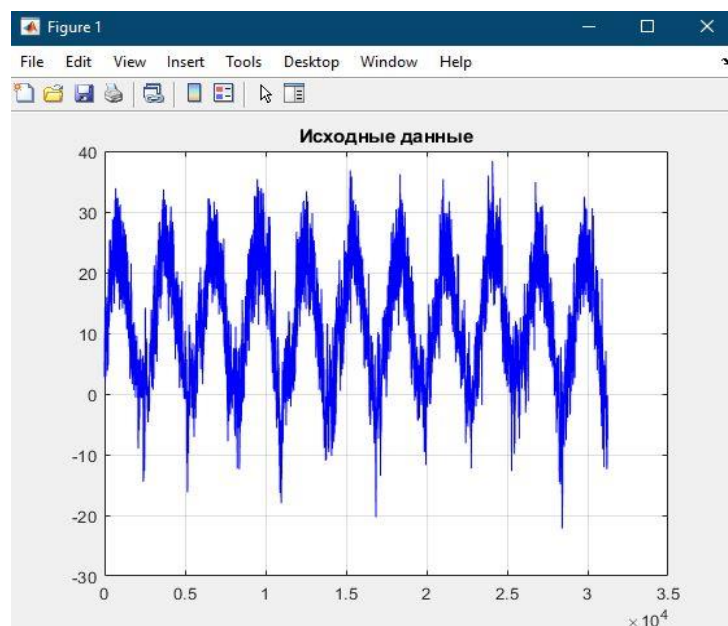


Рисунок 4.1 – Вхідні дані температур за період з 2005 по 2015 рік

Розділяємо вхідні данні на навчальну і перевірочну множину

```
OM=ID(1:29210);
PM=ID(29210+1:end);
```

Формуємо навчальні вектори і цільовий вектор

```
nachdv=5;
shagdv=5;
dldv=300;
nachnv=5;
shagnv=20;
kolnv=500;
DLI=0;
for dv=nachdv:shagdv:dldv
    KOL=0; DLI=DLI+1
    for nv=nachnv:shagnv:kolnv
        KOL=KOL+1;
        c=0;
        for jj1=1:nv
            for ii1=1:dv
                OCV(ii1,jj1)=OM(ii1+c);
            end;
            c=c+1;
        end;
        CV=OCV(dv,:);
        OV=OCV(1:dv-1,:);
```

Далі формуємо мережу прямого поширення

```
net=newff(OV,CV,1,{'purelin','purelin'},'trainlm');
```

Навчаємо мережу



```
[net1,TR]=train(net,OV,CV);
```

Результат роботи мережі на навченій множині(рис. 4.2)

```
figure(2);
subplot 211; plot(1:length(CV),CV(1:length(CV)),'b',...
    1:length(CV),YO(1:length(CV)),'g',...
    1:length(CV),ERR,'r'); grid on;
subplot 212; hist(ERR,50); grid on;
```

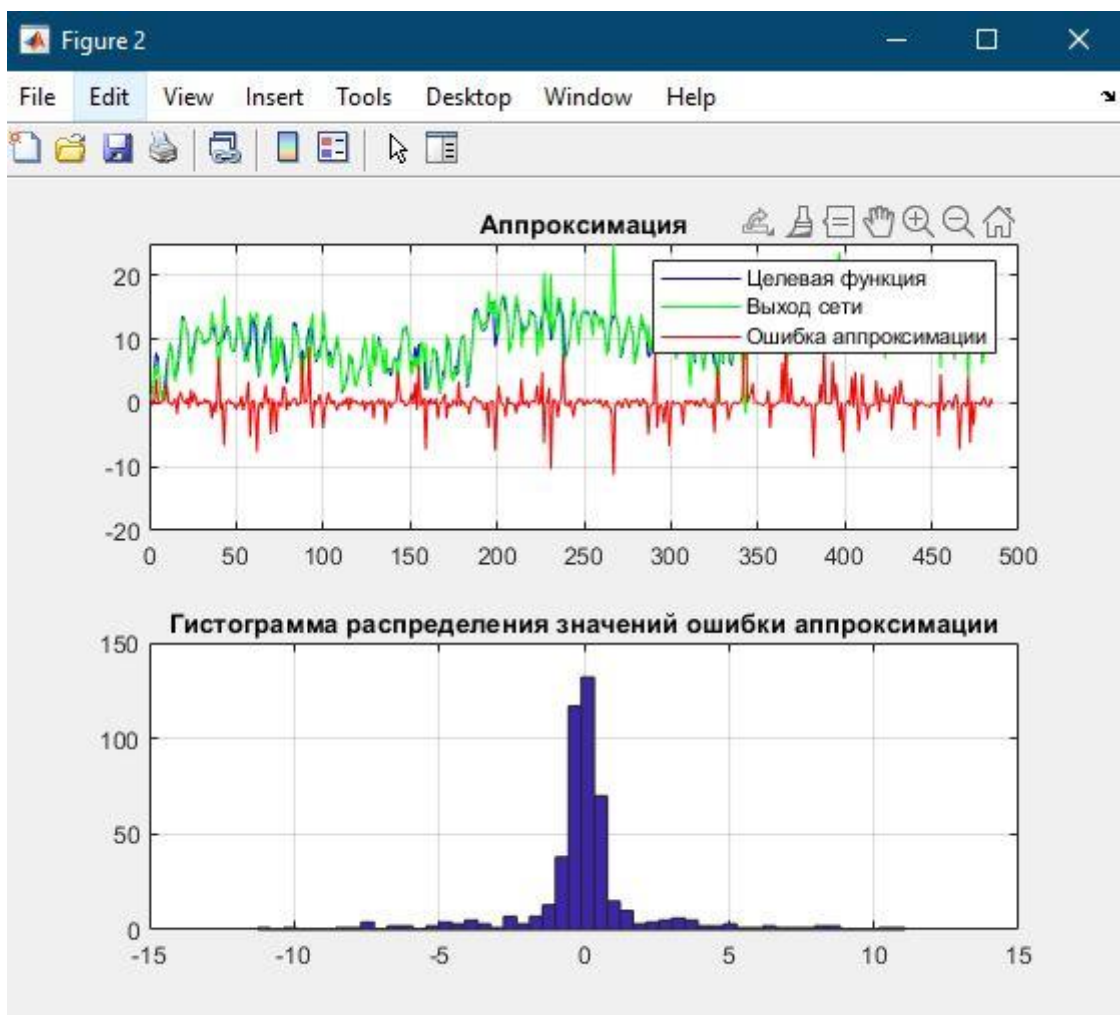


Рисунок 4.2 – Мережа на навченій множині

Оцінимо якість навчання мережі:

```

cc=0;
for i1=1:length(ERR)
    if ERR(i1)>=-2&ERR(i1)<=2
        cc=cc+1;
    end;
end;
KachObuch(DLI,KOL)=(cc/length(ERR))*100;
SKO=sqrt((sum(ERR.^2))/length(ERR));

```

Прогноз на 3 години (надкороткостроковий прогноз – 1 відлік).

Сформуємо перевірочні вектори і еталонний вектор:

```

npv1=1;
k1=0;
for jj1=1:npv1
    for ii1=1:dv
        PEV1(ii1,jj1)=PM(ii1+k1);
    end;
    k1=k1+1;
end;
EV1=PEV1(dv,:);
PV1=PEV1(1:dv-1,:);

```

Наступний крок це прогнозування мережі на перевірочній множині (рис. 4.3)

```

YP1=sim(net1,PV1);
ERR1=EV1-YP1;

figure(3);
subplot 211; plot(1:length(EV1),EV1(1:length(EV1)),'*b',...
    1:length(EV1),YP1(1:length(EV1)),'*g',...
    1:length(EV1),ERR1,'*r'); grid on;

SKO1=sqrt((sum(ERR1.^2))/length(ERR1));

```

Наступні кроки будуть визначення якості навчання (рис. 4.4) та виправдованості прогнозу на 3 години (рис. 4.5):

```
cc1=0;
for iii1=1:length(ERR1)
    if ERR1(iii1)>=-2&ERR1(iii1)<=2
        cc1=cc1+1;
    end;
end;
OPRAVD1(DLI,KOL)=(cc1/length(ERR1))*100;
end;
end;
```

```
[X,Y]=meshgrid(nachnv:shagnv:kolnv,nachdv:shagdv:dldv);
```

```
figure(7);
surf(X,Y,kachObuch); grid on;
figure(8);
surf(X,Y,OPRAVD1); grid on;
```

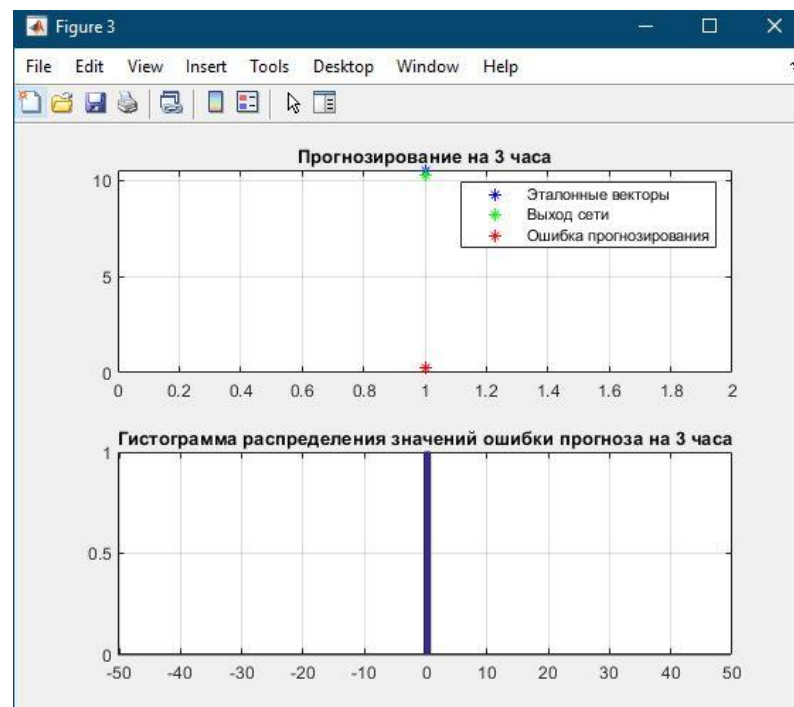


Рисунок 4.3 – Прогнозування мережі на перевірочній множині

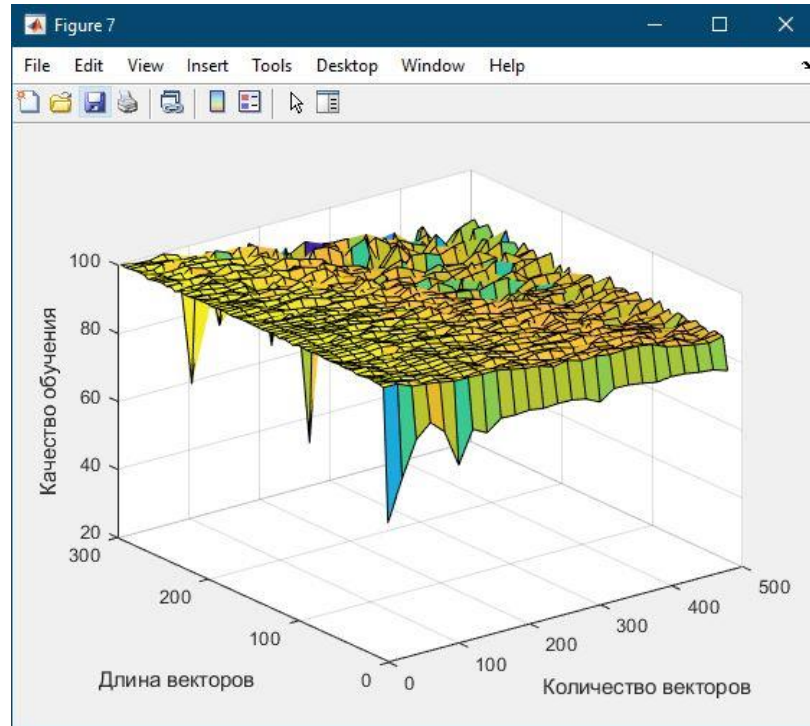


Рисунок 4.4 –Якість навчання

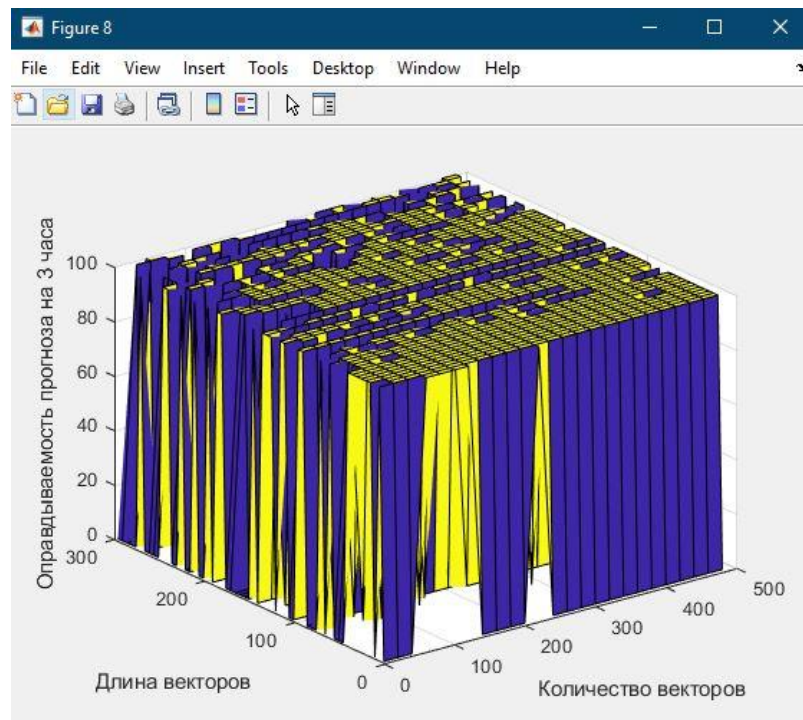


Рисунок 4.5 – Виправданість прогнозу на 3 години

Прогноз на 24 години (надкороткостроковий добовий прогноз - 8 відліків).

Сформуємо перевірочні вектори і еталонний вектор:

```
npv8=8;
k8=0;
for jj8=1:npv8
    for ii8=1:dv
        PEV8(ii8,jj8)=PM(ii8+k8);
    end;
    k8=k8+1;
end;
EV8=PEV8(dv, :);
PV8=PEV8(1:dv-1, :);
```

Наступний крок це прогнозування мережі на перевірочній множині (рис. 4.6)

```
YP8=sim(net1,PV8);
ERR8=EV8-YP8;
```

```
figure(4);
subplot 211; plot(1:length(EV8),EV8(1:length(EV8)), 'b',...
    1:length(EV8),YP8(1:length(EV8)), 'g',...
    1:length(EV8),ERR8, 'r'); grid on;
subplot 212; hist(ERR8,100); grid on;
SKO8=sqrt((sum(ERR8.^2))/length(ERR8));
```

Наступні кроки будуть визначення якості навчання (рис. 4.7) та виправданості прогнозу на 24 години (рис. 4.8):

```
cc8=0;
```

```

for iii8=1:length(ERR8)
    if ERR8(iii8)>=-2&ERR8(iii8)<=2
        cc8=cc8+1;
    end;
end;
OPRAVD8(DLI,KOL)=(cc8/length(ERR8))*100;
end;
end;
[X,Y]=meshgrid(nachnv:shagnv:kolnv,nachdv:shagdv:dldv);
figure(7);
surf(X,Y,kachObuch); grid on;
figure(9);
surf(X,Y,OPRAVD8); grid on;

```

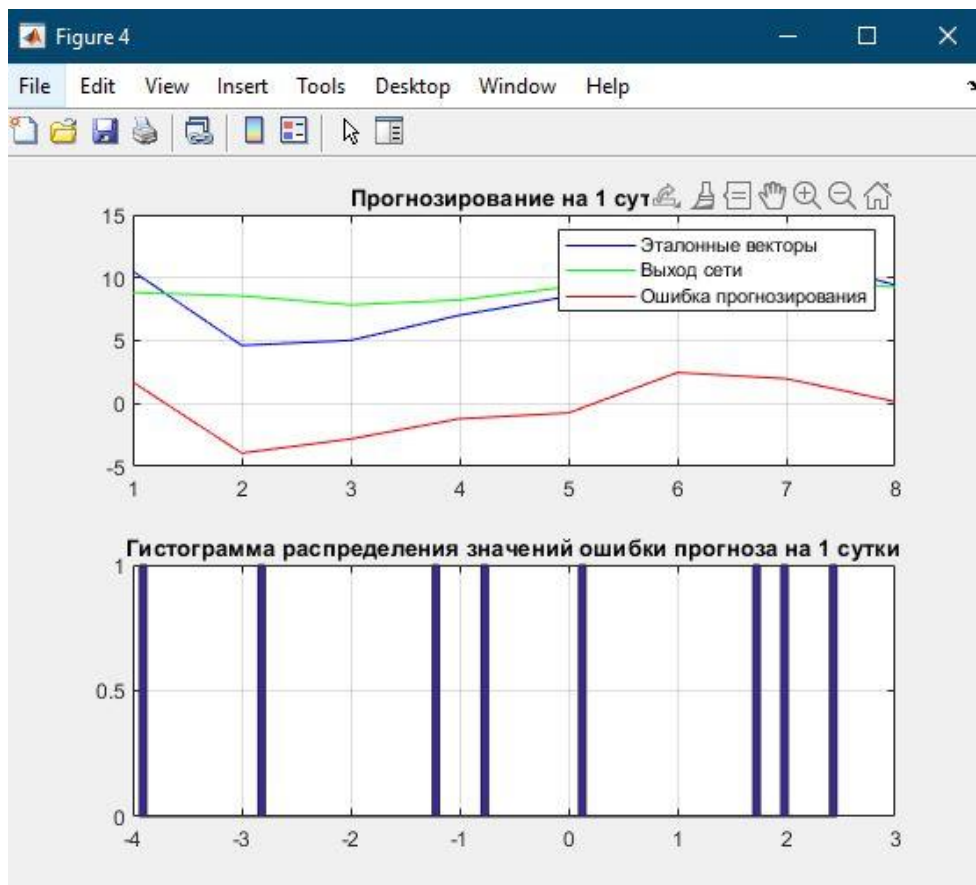


Рисунок 4.6 – Прогнозування мережі на перевіірчній множині

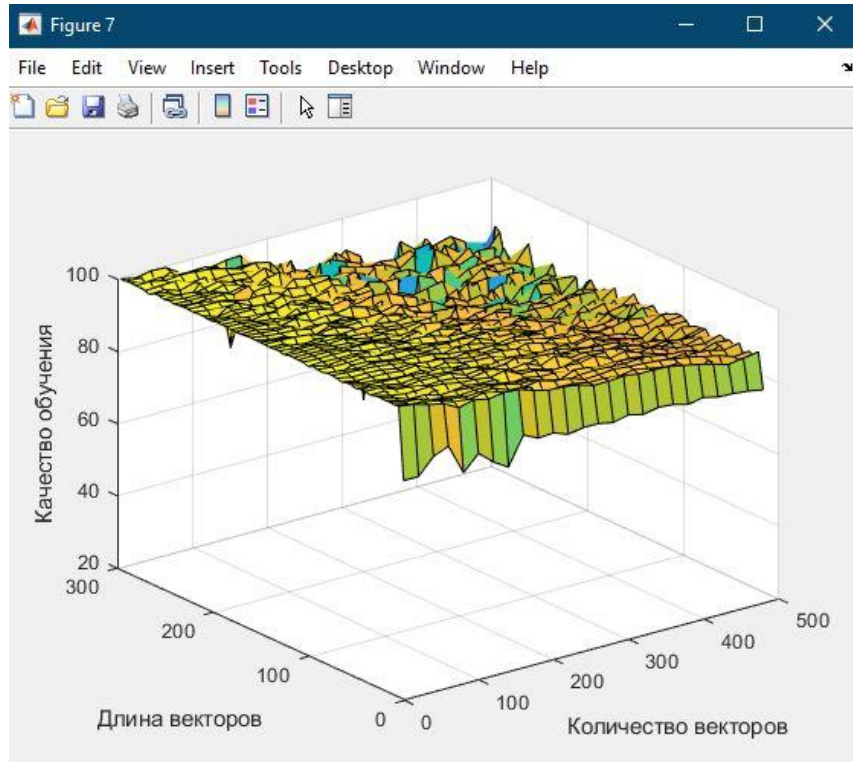


Рисунок 4.7 – Якість навчання

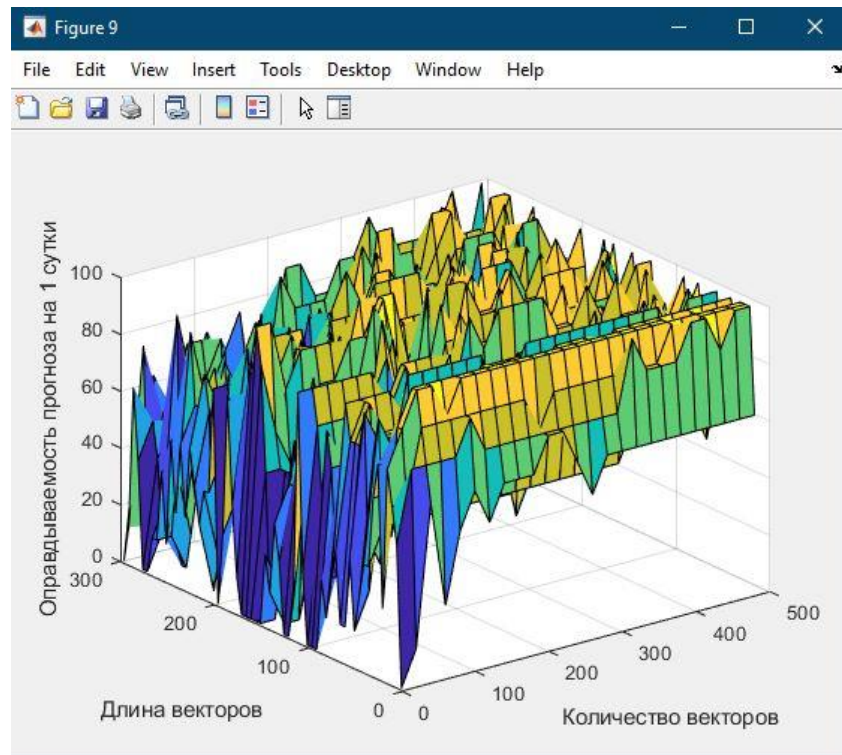


Рисунок 4.8 – Виправданість прогнозу на 24 години

Прогноз на 72 години (короткостроковий тридобовий прогноз - 24 відліки).

Сформуємо перевірочні вектори і еталонний вектор:

```
npv24=24;
k24=0;
for jj24=1:npv24
    for ii24=1:dv
        PEV24(ii24,jj24)=PM(ii24+k24);
    end;
    k24=k24+1;
end;
EV24=PEV24(dv,:);
PV24=PEV24(1:dv-1,:);
```

Наступний крок це прогнозування мережі на перевірочній множині (рис. 4.9)

```
YP24=sim(net1,PV24);
ERR24=EV24-YP24;

figure(5);
subplot 211; plot(1:length(EV24),EV24(1:length(EV24)),'b',...
    1:length(EV24),YP24(1:length(EV24)),'g',...
    1:length(EV24),ERR24,'r'); grid on;

subplot 212; hist(ERR24,100); grid on;

SKO24=sqrt((sum(ERR24.^2))/length(ERR24));
```



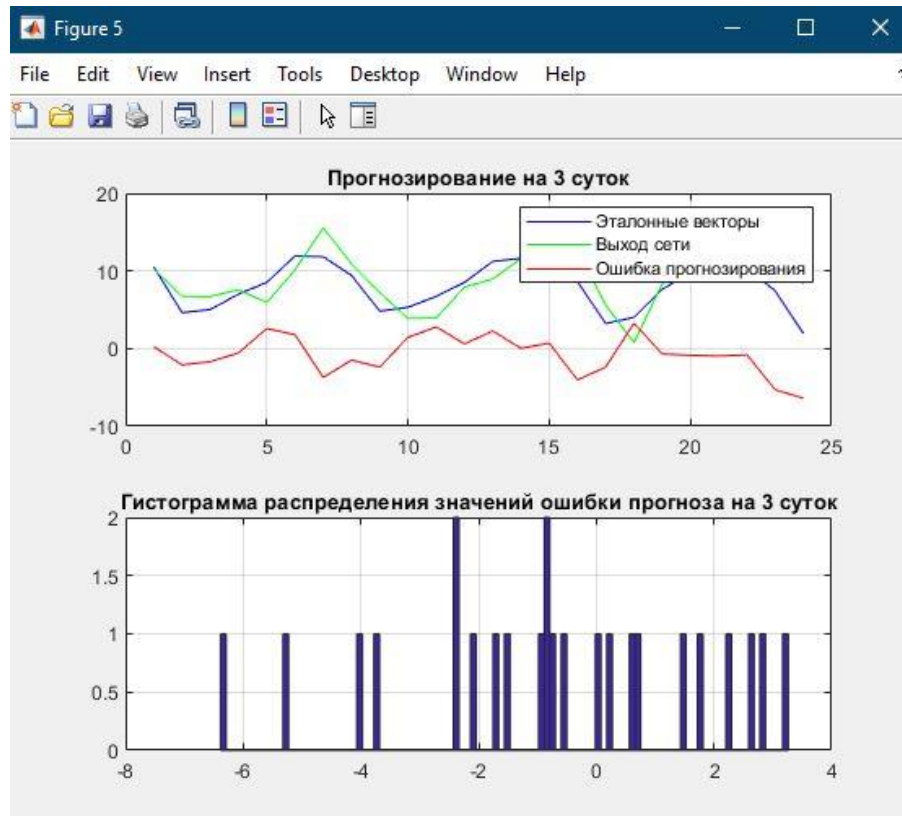


Рисунок 4.9 – Прогнозування мережі на перевірочній множині

Наступні кроки будуть визначення якості навчання (рис. 4.10) та виправданості прогнозу на 3 доби (рис. 4.11):

```
cc24=0;
for iii24=1:length(ERR24)
    if ERR24(iii24)>=-2&ERR24(iii24)<=2
        cc24=cc24+1;
    end;
end;
OPRAVD24(DLI,KOL)=(cc24/length(ERR24))*100;
end;
end;
```

```
[X,Y]=meshgrid(nachnv:shagnv:kolnv,nachdv:shagdv:dldv);
```

```
figure(7);
surf(X,Y,kachObuch); grid on;
```

```
figure(10);  
surf(X,Y,OPRAVD24); grid on;
```

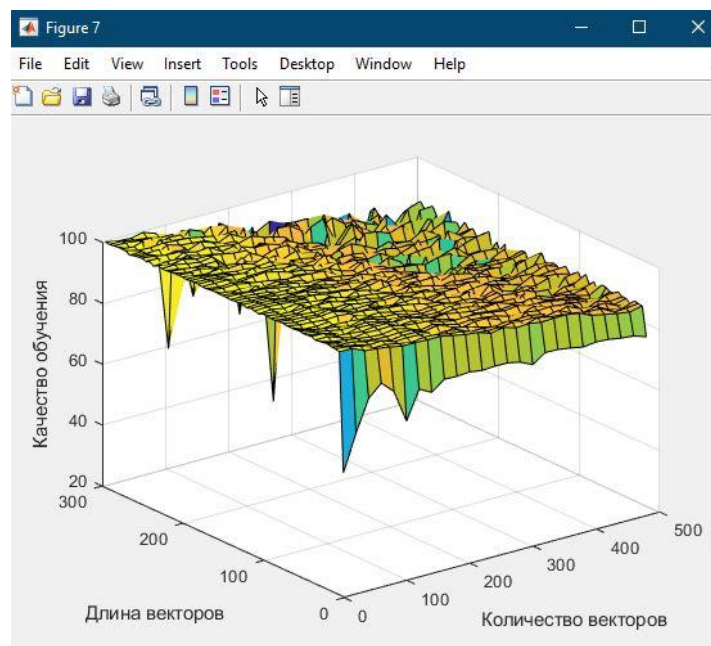


Рисунок 4.10 – Якість навчання

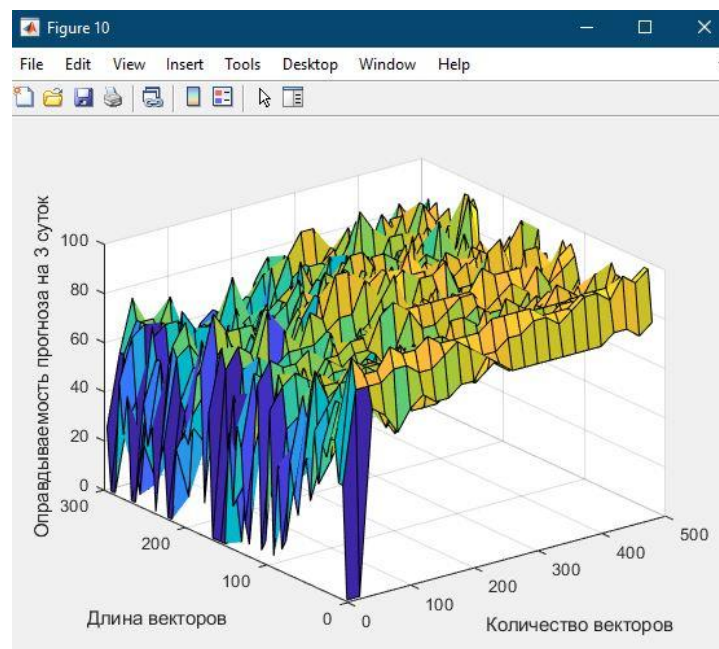


Рисунок 4.11 – Виправданість прогнозу на 3 доби

За результатами першого етапу для подальшої оцінки параметрів якості прогнозування, крім обґрунтування, була використана одношарова мережа прямого поширення з функціями лінійної активації нейронів, з навчанням на основі «сирих» даних і з використанням алгоритму Левенберга-Марквардта і з об'ємом вихідних даних у вигляді 150 векторів довжиною 16 кожен (окреслені області стабільного прогнозування на рис. 3.5...3.7). Причому при тренуванні за такою невеликою кількістю даних необхідно дотримуватися однієї умови: при прогнозуванні в певний сезон (пору року) потрібно брати більш ранні дані для тренувань, але обов'язково з точно такої ж пори року. В іншому випадку помилка прогнозування буде неприйнятно великою (рис. 4.12). Результати застосування даної мережі з вищевказаними параметрами з повторним короткочасним прогнозуванням з попереднім тригодинним, одним днем і трьома днями показані на рис. 4.13, 4.14, 4.15.

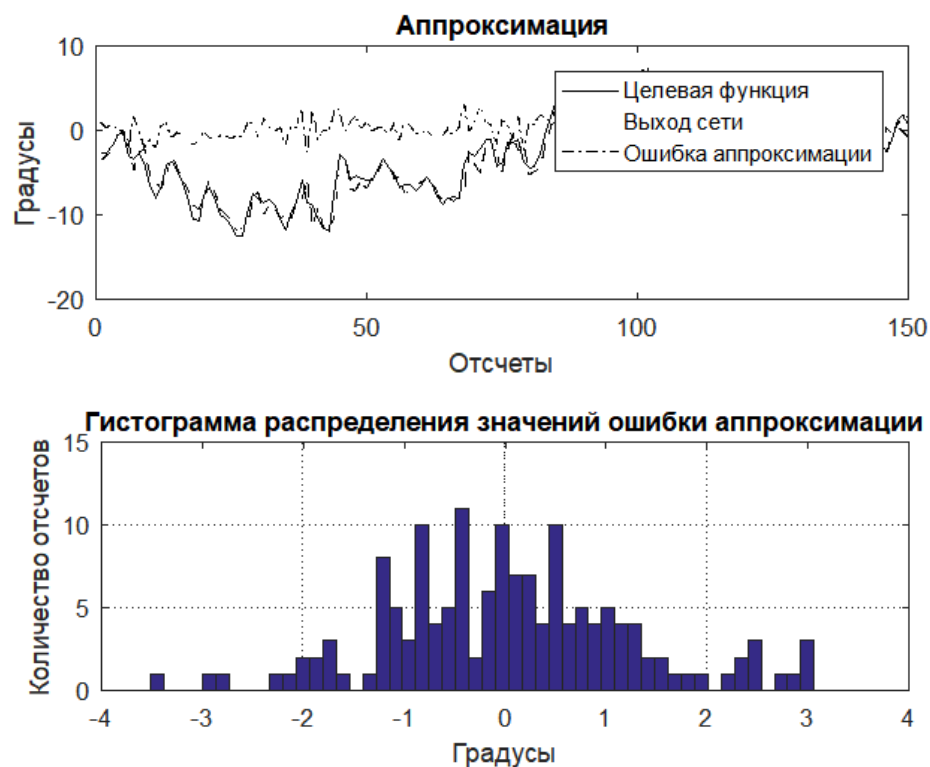


Рисунок 4.12 – Ілюстрація якості мережевого навчання

При цьому якість навчання склала 88%, обґрунтування прогнозу на три

години - 100%, за один день - 100%, за три дні - 91%. Ці показники були отримані з похибкою RMS (SKO) наближення 1,22 градуса, прогноз SKO на 3 години - 0,17 градуса, прогноз SKO на один день - 0,83 градуса і прогноз SKO на три дні - 1 градус.

У процесі проведення досліджень були вирішені всі завдання: визначено оптимальний вибірка даних для навчання мережі прямого розподілу, що забезпечує найкраще обґрунтування короткострокових температурних прогнозів і досліджено вплив на обґрунтування короткострокових температурних прогнозів параметрів нейронної мережі навчання даних ("сирих" даних, випадкова сезонність даних для навчання і прогнозування) і параметрів нейронів. Мережа (лінійні функції активації нейронів, один прихований шар, навчання на основі процедури зворотного зв'язків помилки з використанням алгоритму Левенберга-Марквардта як найшвидшого), якість отриманих короткострокових прогнозів оцінювалася як висока (обґрунтування прогнозу на три години і один день становить майже 100%, а обґрунтування прогнозу на три дні - не менше 91%).

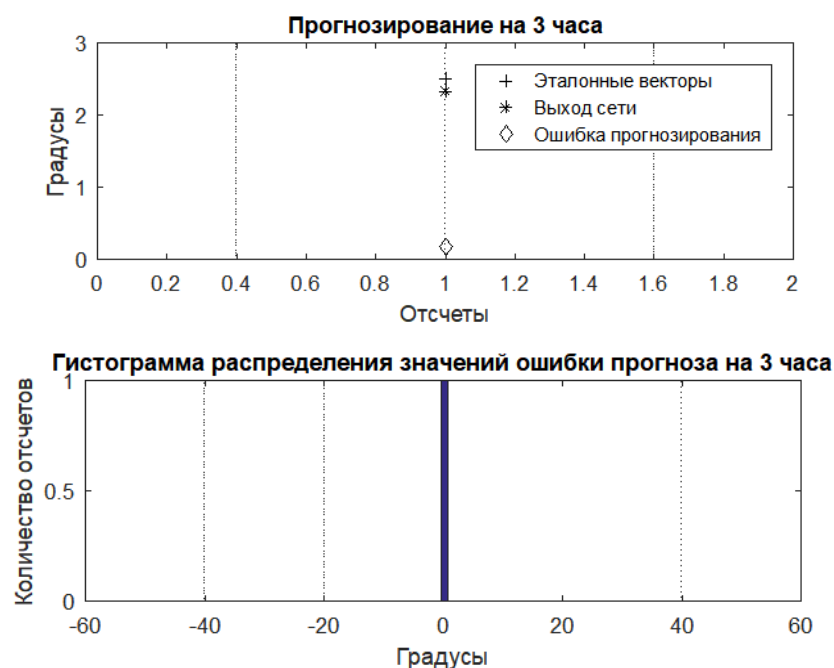


Рисунок 4.13 – Тригодинне прогнозування результатів

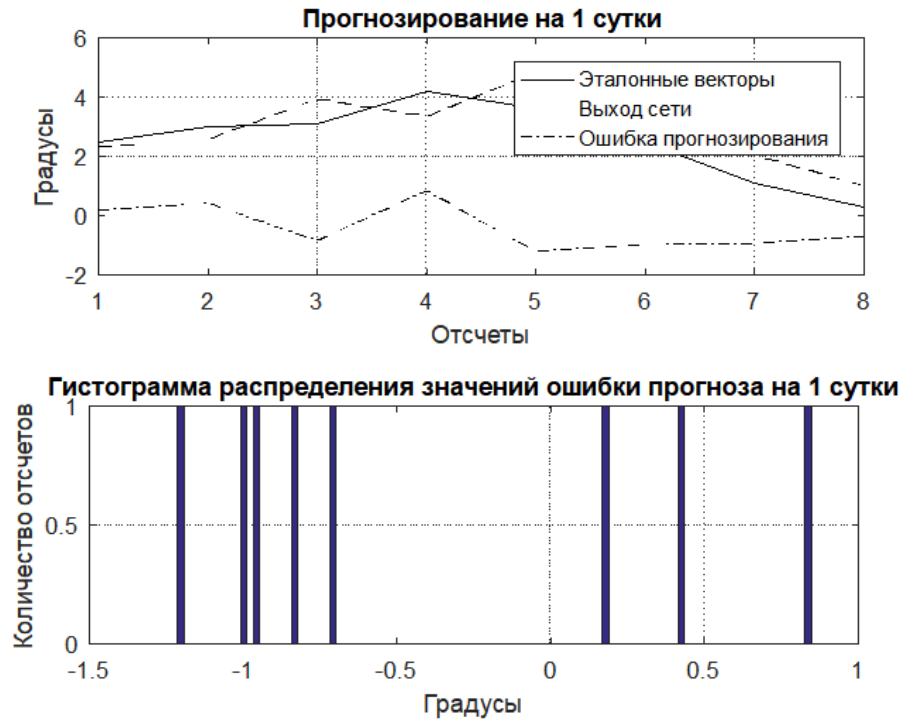


Рисунок 4.14 – Прогнозні результати на один день

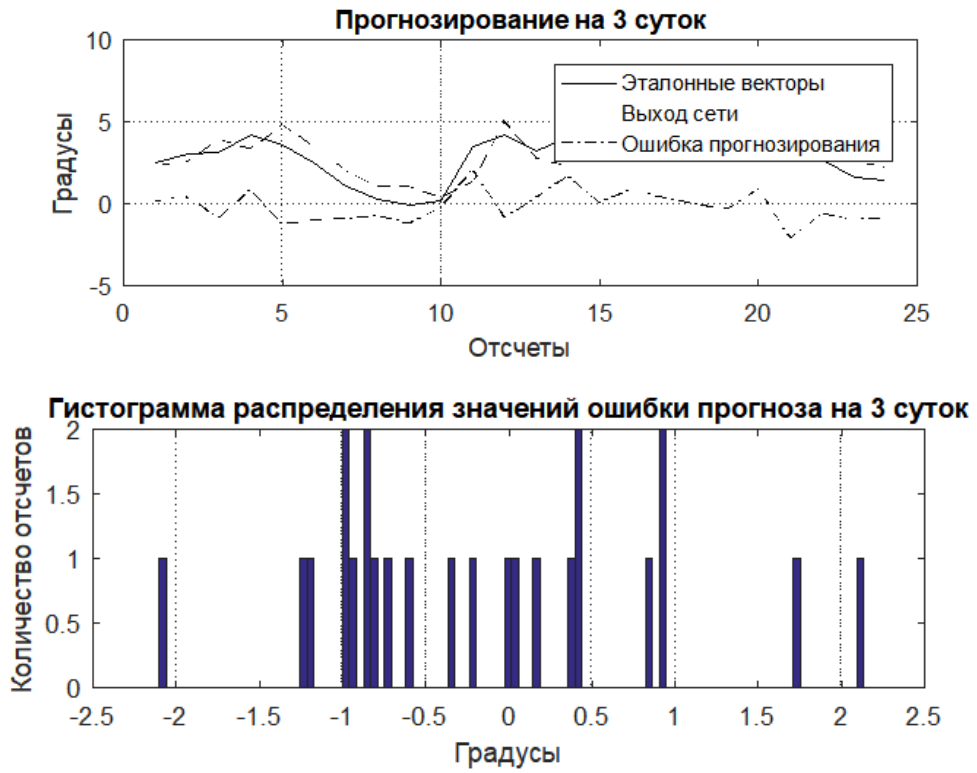


Рисунок 4.15 – Прогнозування результатів на 3 доби

Всі перераховані вище результати досліджень засновані на даних, що

відображають спокійну атмосферу в певний сезон. На таких даних проводилися як мережеві тренування, так і прогнозування. Безсумнівним інтересом є вивчення впливу порушеної атмосфери на якість прогнозування температури. Крім того, необхідно досліджувати процедури прогнозування з використанням штучних нейронних мереж та інших атмосферних факторів. довгострокове прогнозування окремих погодних елементів.

## **5 УПРАВЛІННЯ ПРОЕКТОМ**

Проект – унікальний набір процесів, що складаються з скоординованих і керованих завдань з датами початку та закінчення, зроблених для досягнення мети. Досягнення мети проекту вимагає отримання результатів, що відповідають певним заздалегідь вимогам, в тому числі обмеження на отримання результатів, таких як час, гроші і ресурси.

Управління проектом – область діяльності, в ході якої визначаються та досягаються чіткі цілі проекту при балансуванні між обсягом робіт, ресурсами (такими як гроші, працю, матеріали, енергія, простір і ін.), Часом, якістю та ризиками. Ключовим фактором успіху проектного управління є наявність чіткого заздалегідь визначеного плану, мінімізації ризиків і відхилень від плану, ефективного управління змінами.

### **5.1 Планування проекту**

Для поставленої задачі планування використовувалася програма Microsoft Project – програма управління проектами, розроблена і продається корпорацією Microsoft.

Microsoft Project створений, щоб допомогти менеджеру проекту в розробці планів, розподілі ресурсів за завданнями, відстеження прогресу і аналізі обсягів робіт.

Microsoft Project створює розкладу критичного шляху. Розклади можуть бути складені з урахуванням використовуваних ресурсів. Ланцюжок візуалізується в діаграмі Ганта [26].

Проект дозволяє досягти певного результату в певні терміни і за певні гроші. План проекту складається для того, щоб визначити, за допомогою яких робіт буде досягатися результат проекту, які люди й устаткування потрібні для виконання цих робіт, в який час ці люди і обладнання будуть зайняті роботою по проекту. Тому проектний план містить три основні елементи: завдання (завдання), ресурси (ресурс) і призначення (призначення).

## **5.2 Оцінка трудомісткості і термінів розробки програмного забезпечення**

На даному етапі необхідно визначити, скільки часу займе реалізація проекту і в які терміни необхідно вирішити ту чи іншу задачу. Це, в свою чергу, визначить, коли саме потрібно залучити ті чи інші ресурси і як буде контролюватися розподіл коштів проекту.

Варто відзначити, що в управлінні термінами пріоритетним є визначення взаємозв'язків завдань проекту та способів їх реалізації; саме це і впливає в першу чергу на те, як вибудовуються завдання в календарному плані робіт. Даний підхід має ряд переваг:

По-перше, робота повинна визначати графік, а не навпаки, тому що досить легко розробити календарний план, який на ділі буде розходитися з намірами.

По-друге, даний підхід повністю збігається з інструментами управління проектами, такими як Microsoft Project.

По-третє, використовуючи даний підхід, можна отримати цілком реалістичний графік результатів, менш схильних до змін, на відміну від графіків робіт, які є більш гнучкими і визначаються іншими способами.

Таким чином, процеси управління термінами проекту включають в себе наступні фази:

1) Визначення складу операцій – виявлення всіх планових операцій, які необхідно здійснити. Це досягається за допомогою таких методів як: декомпозиція, шаблони, планування методом набігає хвилі, експертна оцінка і т.д.

2) Визначення взаємозв'язків операцій включає ідентифікацію та документування логічних взаємозв'язків між плановими операціями. Завдання послідовності може бути виконання за допомогою програмного забезпечення для управління проектами або вручну.

3) Оцінка ресурсів операцій покликана визначити, які ресурси (людські, матеріальні і т.д.) будуть використовуватися і в якій кількості, і коли кожен з цих ресурсів буде доступний для виконання проектних рішень.

4) Оцінка тривалості операцій задіє інформацію про зміст робіт планової операції, типах необхідних ресурсів, розрахунковій кількості ресурсів і календарях ресурсів із зазначенням їх доступності.

5) Розробка розкладу здійснюється за допомогою визначення планових дат початку та закінчення кожної з операцій на проекті. Розробка розкладу проводиться безперервно по всьому проекту в міру виконання робіт, зміни плану управління проектом і виникнення або припинення очікуваних ризиків або виявлення нових.

6) Управління розкладом інтегрує в собі всі вищеописані фази.

Коли всі плановані результати відображені на діаграмі, то не важко визначити початок і закінчення кожного результату.

Сама рання дата початку виконання завдання відповідно до логіки і обмеженнями розкладу називається «ранній старт». Її початок залежить від того, чи повинна попередня завдання бути завершена або достатньо її почати. Додаючи до дати «раннього старту» відрізок часу, призначений на виконання завдання, можна знайти дату «раннього фінішу», або найбільш раннього



терміну завершення завдання відповідно до логіки і обмеженнями розкладу. Таким чином, можна визначити ранній старт і фініш для кожної з задач на діаграмі. Такий метод називається «прямий прохід». Він має на увазі прочитання діаграми зліва направо [27].

Отже, використовуючи метод прямого проходу, можна також дізнатися якою є загальна тривалість всього проекту, і в який час знадобляться ресурси для виконання проектних завдань. Отримані для даного проекту результати статистики проекту показані на малюнку.

При розробці даного наукового проекту не здійснювалося серйозне управління ризиками, проте враховувалася можливість виникнення певних форс-мажорів, таких як хвороби виконавців, забуті роботи і т.п.

Всі ці процедури взаємодіють один з одним, а також з іншими процедурами. Кожна процедура виконується, принаймні, один раз в кожному проекті.

Для оцінки трудомісткості були складені зведений звіт про трудовитратах і ресурсів та звіт про базові трудозатратах ресурсів. Результати цих двох звітів наведені на рис. 5.1 і рис. 5.2.

|                                          | Начало       | Окончание    |         |
|------------------------------------------|--------------|--------------|---------|
| Текущее                                  | Пн 03.10     | Пт 28.12     |         |
| Базовое                                  | НД           | НД           |         |
| Фактическое                              | Пн 03.10     | Пт 28.12     |         |
| Отклонение                               | Од           | Од           |         |
|                                          | Длительность | Трудозатраты | Затраты |
| Текущие                                  | 55д          | 1 349,2ч     | 0,00р.  |
| Базовые                                  | 0д           | 0ч           | 0,00р.  |
| Фактические                              | 55д          | 1 349,2ч     | 0,00р.  |
| Оставшиеся                               | 0д           | 0ч           | 0,00р.  |
| Процент завершения                       |              |              |         |
| Длительность: 100%    Трудозатраты: 100% |              |              |         |
| Закреть                                  |              |              |         |

Рис 5.1 – Статистика розробленого проекту

Як видно з обох звітів, представлених на малюнках вище, загальні трудовитрати ресурсів були розподілені приблизно рівномірно між всіма виконавцями даного проекту протягом його розробки (рис. 5.3).

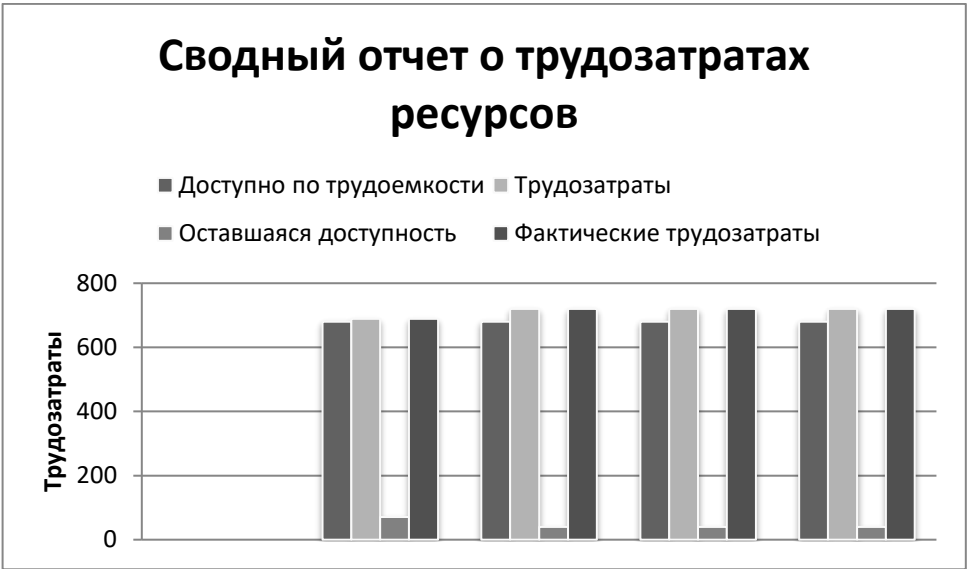


Рис 5.2 – Зведений звіт про трудовитратах ресурсів



Рис 5.3 – Звіт про базові трудозатратах

### 5.3 Організаційне забезпечення проекту

Причиною виникнення ризиків є невизначеності, що існують в кожному проекті. Ризики можуть бути відомі – ті, які визначені, оцінені, для яких можливо планування. Ризики невідомі – ті, які не ідентифіковано і не можуть бути спрогнозовані. Хоча специфічні ризики і умови їх виникнення не визначені, менеджери проекту знають, виходячи з минулого досвіду, що більшу частину ризиків можна передбачити.

Реалізуючи проекти, що мають високий ступінь невизначеності в таких елементах, як цілі і технології їх досягнення багато компаній приділяють увагу розробці та застосуванню корпоративних методів управління ризиками. Дані методи враховують як специфіку проектів, так і корпоративних методів управління [28].

Управління ризиками – це процеси, пов'язані з ідентифікацією, аналізом ризиків та прийняттям рішень, які включають максимізацію позитивних і мінімізацію негативних наслідків настання ризикових подій. Процес управління ризиками проекту зазвичай включає виконання наступних процедур:

- 1) Планування управління ризиками – вибір підходів і планування діяльності з управління ризиками проекту.
- 2) Ідентифікація ризиків – визначення ризиків, здатних вплинути на проект, і документування їх характеристик.
- 3) Якісна оцінка ризиків – якісний аналіз ризиків і умов їх виникнення з метою визначення їх впливу на успіх проекту.
- 4) Кількісна оцінка – кількісний аналіз ймовірності виникнення та впливу наслідків ризиків на проект.
- 5) Планування реагування на ризики – визначення процедур і методів по ослабленню негативних наслідків ризикових подій і використання можливих переваг.

б) Моніторинг та контроль ризиків – моніторинг ризиків, визначення залишаються ризиків, виконання плану управління ризиками проекту і оцінка ефективності дій з мінімізації ризиків.

Всі ці процедури взаємодіють один з одним, а також з іншими процедурами. Кожна процедура виконується, принаймні, один раз в кожному проекті. Незважаючи на те, що процедури розглядаються як дискретні елементи з чітко визначеними характеристиками, на практиці вони можуть частково збігатися і взаємодіяти [29].

При розробці даного наукового проекту не здійснювалося серйозне управління ризиками, проте враховувалася можливість виникнення певних форс-мажорів, таких як хвороби виконавців, забуті роботи і т.п. Для цього в кожен етап була додана фіктивна завдання з мінімальним пріоритетом, під назвою «інші роботи» для кожного ресурсу. Після вирівнювання ресурсів ці завдання виявляються в кінці етапу. Тривалість цих завдань залежить від імовірності виникнення і ступеня впливу ризиків, вона залежить від способу визначення оцінок тривалості завдань, здоров'я членів команди та інших факторів. Тривалість «інших робіт» для кожного етапу розробки виставлялася приблизно до чверті довжини етапу. Такий метод обліку ризиків дозволяє, по-перше назвати терміни виконання проекту і його етапів, причому аргументовано і з високим ступенем достовірності, і по-друге оцінити приблизні трудовитрати по проекту.

## **ВИСНОВОК**

Моїм завданням було дослідити застосування лінійної штучної нейронної мережі у програмному середовищі системи комп'ютерної математики для екстраполяції ряду спостережень температури повітря, розробити та навчити штучну лінійну нейронну мережу на основі температурного ряду періода з 01.02.2015 по 31.12.2015, визначення

виправдовуваність прогнозу на 3 години, добу, 3 доби.

У процесі проведення досліджень були вирішені всі завдання: визначено оптимальний вибірка даних для навчання мережі прямого розподілу, що забезпечує найкраще обґрунтування короткострокових температурних прогнозів і досліджено вплив на обґрунтування короткострокових температурних прогнозів параметрів нейронної мережі навчання даних ("сирих" даних, випадкова сезонність даних для навчання і прогнозування) і параметрів нейронів. Мережа (лінійні функції активації нейронів, один прихований шар, навчання на основі процедури зворотного зв'язків помилки з використанням алгоритму Левенберга-Марквардта як найшвидшого), якість отриманих короткострокових прогнозів оцінювалася як висока (обґрунтування прогнозу на три години і один день становить майже 100%, а обґрунтування прогнозу на три дні - не менше 91%).

## **ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Великанова Л.И. Краткосрочное прогнозирование влажности горных грунтов // Проблемы автоматизации и управления. – 2015, №4.– С. 158–166.
2. Верзунов С.Н. Применение глубоких нейронных сетей для краткосрочного прогноза дальности видимости. Проблемы автоматизации и управления. 2019, №1 (36) С. 118-130.
3. Верзунов С.Н., Лыченко Н.М. Мультивейвлетная полиморфная сеть для прогнозирования геофизических временных рядов // Проблемы автоматизации и управления. –2017. – № 1 (32). – С. 78–87.
4. Грибин А.С. Применение алгоритмов искусственных нейронных сетей для краткосрочного метеопрогноза: Дис. ... канд. физ.-матем. наук. СПб. 2005. 154 с.
5. Настанова з гідрометеорологічного прогнозування. Український гідрометеорологічний центр. Київ, 2019. – 35 С.

6. Koprinska, Irena et al. Convolutional Neural Networks for Energy Time Series Forecasting // 2018 International Joint Conference on Neural Networks (IJCNN). 2018: p 1–8.
7. J.W. Taylor, R. Buizza Neural Network Load Forecasting with Weather Ensemble Predictions. // IEEE Trans. on Power Systems, 2002, Vol. 17, PP. 626-632. DOI: 10.1109/TPWRS.2002.800906
8. Васильев А.А., Вильфанд Р.М. Прогноз погоды СПб.: Типография «Моби Дик», М.: 2008. 60 с.
9. Оке Т.Р. Климаты пограничного слоя/Т.Р. Оке: – Ленинград: Гидрометеиздат, 1982. 360с.
10. Матвеев Л.Т. Особенности метеорологического режима большого города Л.Т.Матвеев// Метеорологияи гидрология. – 1979. – №5. 432 с.
11. Розенблатт Ф. Принципы нейродинамикн. Персептрон и теория механизмов мозга: Пер. с англ. — М.: Мир, 1965, 451с.
12. Капустин А.В., Сторожук Н.Л. Технические средства гидрометеорологической службы. СПб, 2005. 283 с.
13. Гордин В. А. Математика, компьютер, прогноз погоды. Л.: Гидрометеиздат, 1991. 224 с.
14. Будыко М.И. Климат и жизнь. Л.: Гидрометеиздат., 1971. 472 с.
15. Коновалова Н.В., Коробов В.Б., Васильев Л.Ю. Интерполирование климатических данных при помощи ГИС-технологий. Метеорология и гидрология, 2006, №5. 628 с.
16. Минский М. Л., Пейперт. С. Персептроны: Пер. с англ. — М.: Мир, 1971. 280 с.
17. Carpenter G. A., Grossberg S. The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network // Computer. — 1988. — March. P. 77-88.
18. Бодянский Е.В. Искусственные нейронные сети: архитектуры, обучение, применения / Е.В. Бодянский, О.Г. Руденко — Харьков: ТЕЛЕТЕХ, 2004. 369с.

19. Waibel A. Consonant Recognition by Modular Construction of Large Phonetic Time — Delay Neural Network. / In Advances in Neural Information Processing Systems 1. D.S. Touretzky (ed.), 1989. P. 215-223.
20. Whitley D., Dominic S., Das R. Genetic Reinforcement Learning with Multilayer Neural Networks / In Belew and Booker. — 1991. P. 562-570.
21. Кнут Д. Э. Искусство программирования. — 2-е изд. — М.: Вильямс, 2000. — Т. 3. Сортировка и поиск. 632 с.
22. Poggio T., Girosi F. Networks for Approximation and Learning // Proc. of IEEE. — 1990. — 78(9). P. 1481-1497.
23. Вазан М. Стохастическая аппроксимация. — М. : Мир, 1972. 295 с.
24. Медведев В.С, Потемкин В.Г. Нейронные сети. MATLAB 6 / Под общ. ред. В.Г.Потемкина. — М.: ДИАЛОГ-МИФИ, 2002. 496 с.
25. Hecht-Nilsen R. Neurocomputing. — Addison-Wesley, 1990. P. 185-204.
26. Афонин, А.М. Управление проектами: Учебное пособие / А.М. Афонин, Ю.Н. Царегородцев, С.А. Петрова. - М.: Форум, 2010. 184 с.
27. Гонтарева, И.В. Управление проектами: Учебное пособие / И.В. Гонтарева, Р.М. Нижегородцев, Д.А. Новиков. - М.: КД Либроком, 2013. 384 с.
28. Джалота, П. Управление проектами в области информационных технологий / П. Джалота. - М.: Лори, 2014. 224 с.
29. Ларсон, Э.У. Управление проектами: Учебник / Э.У. Ларсон, К.Ф. Грей; Пер. с англ. В.В. Дедюхин. - М.: ДиС, 2013. 784 с.

## **ДОДАТОК А Графічна частина магістерської роботи**



# Дослідження застосування лінійної штучної нейронної мережі для екстраполяції ряду спостережень температури повітря

Тимченко Владислав  
МКМ-20

Рисунок А.1 – Вступ

## Вхідні дані температур

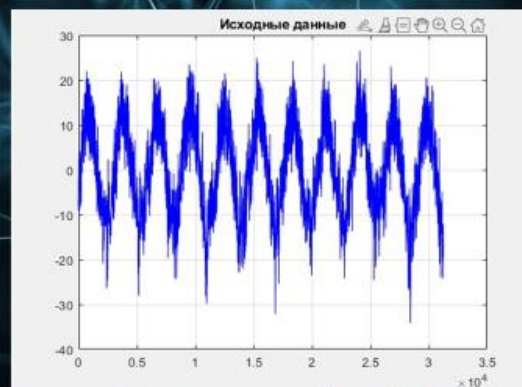


Рисунок А.2 – Вхідні дані температур

## Мережа на навченій множині

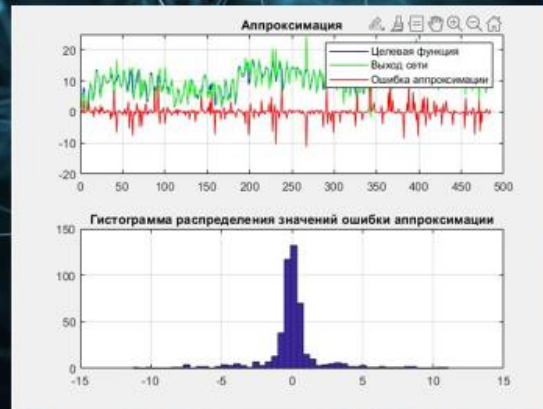


Рисунок А.3 – Мережа на навченій множині

## Прогнозування мережі на перевірочній множині

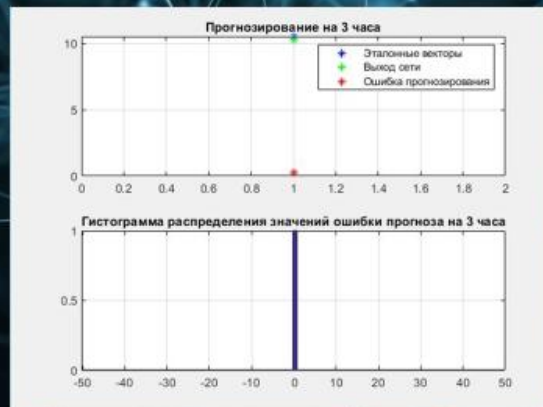


Рисунок А.4 – Прогнозування мережі на перевірочній множині

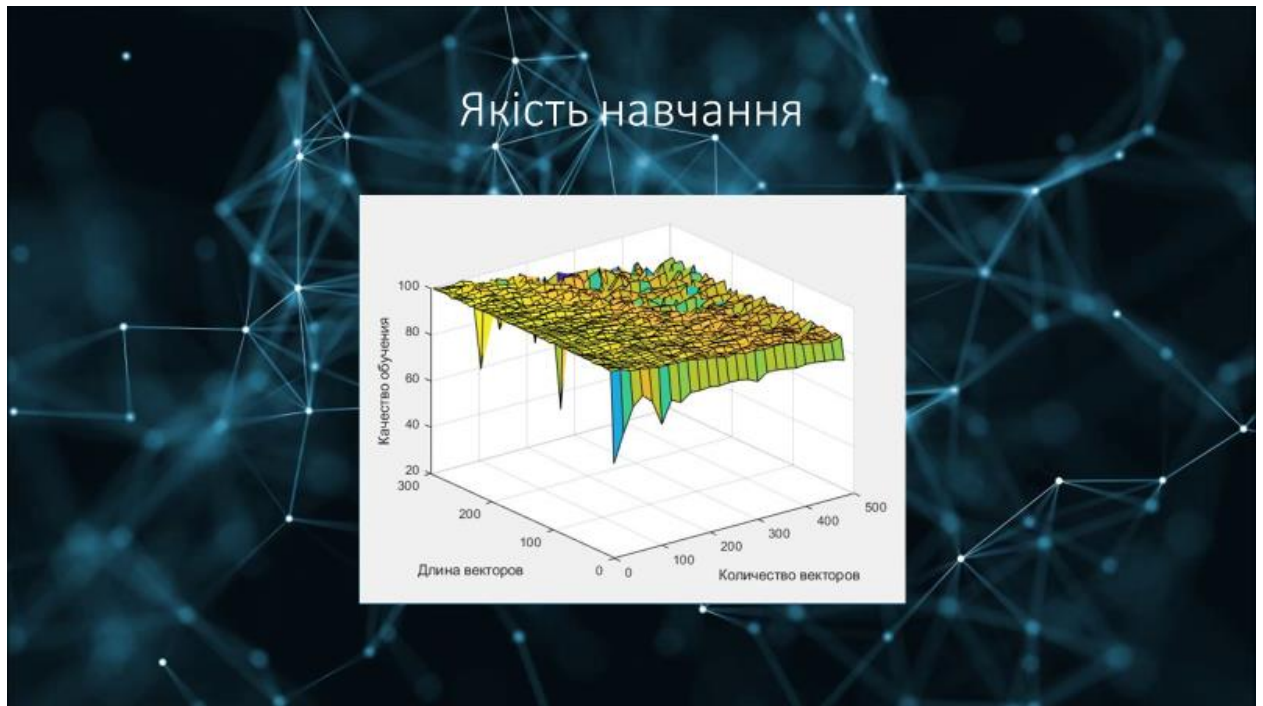


Рисунок А.5 – Якість навчання

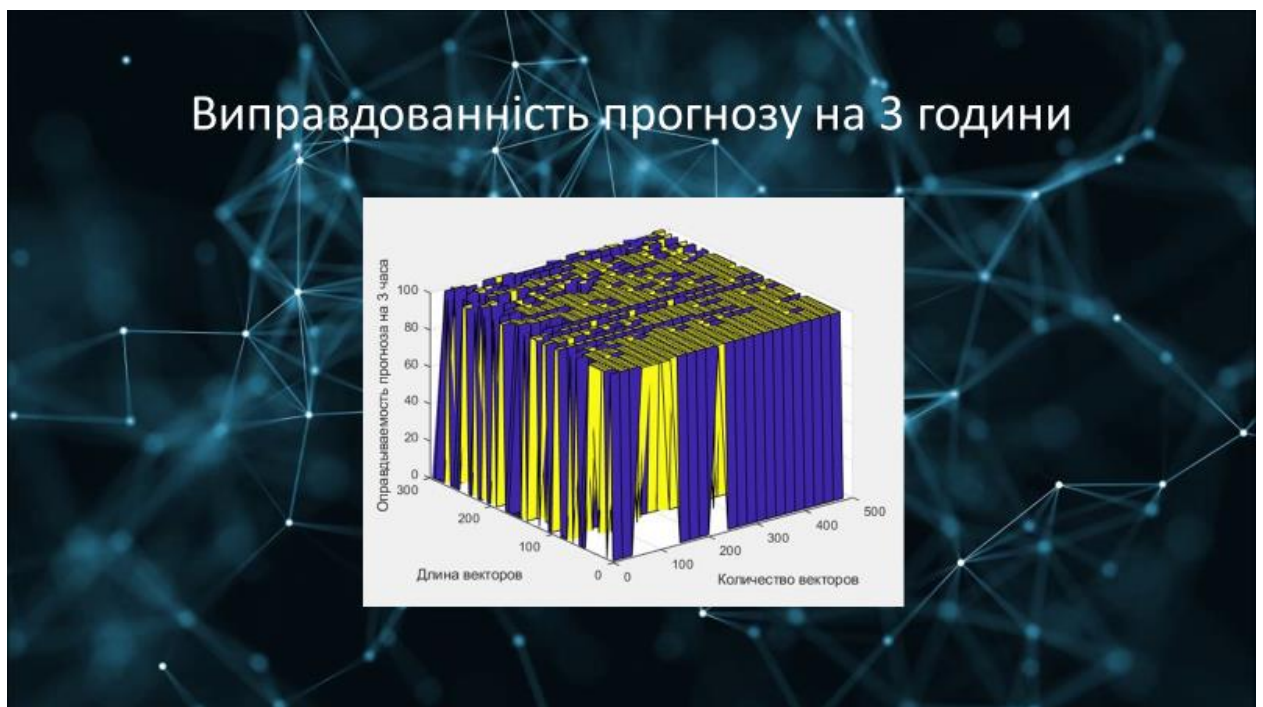


Рисунок А.6 – Виправданість прогнозу на 3 години



## Прогнозування мережі на перевірочній множині

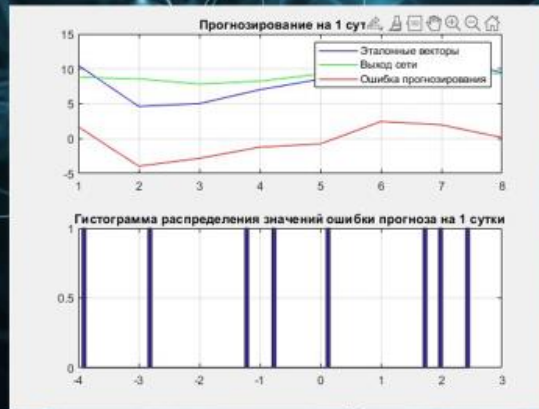


Рисунок А.7 – Прогнозування мережі на перевірочній множині

## Якість навчання

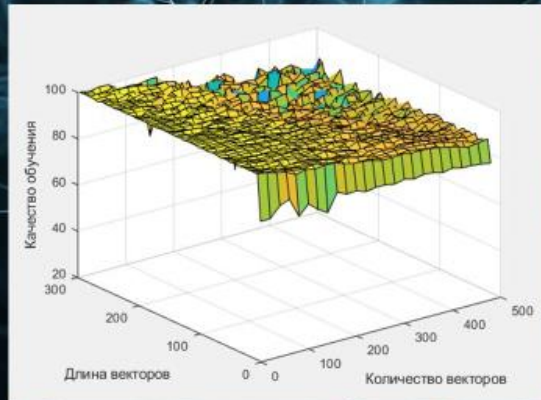


Рисунок А.8 – Якість навчання

## Виправдованність прогнозу на 24 години

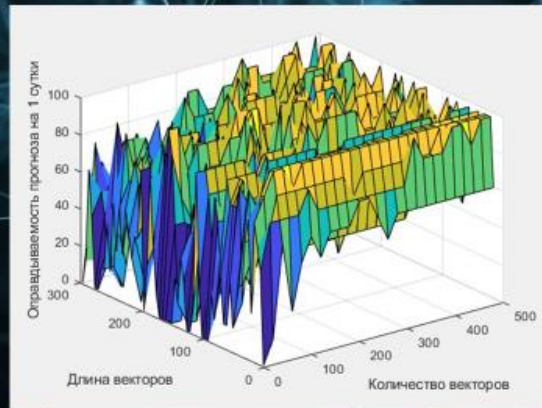


Рисунок А.9 – Виправданість прогнозу на 24 години

## Прогнозування мережі на перевіірчній множині

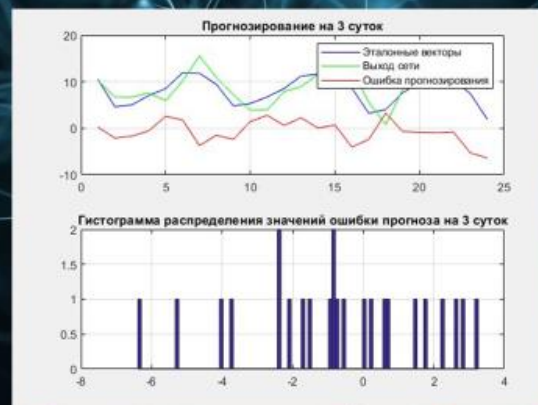


Рисунок А.10 – Прогнозування мережі на перевіірчній множині

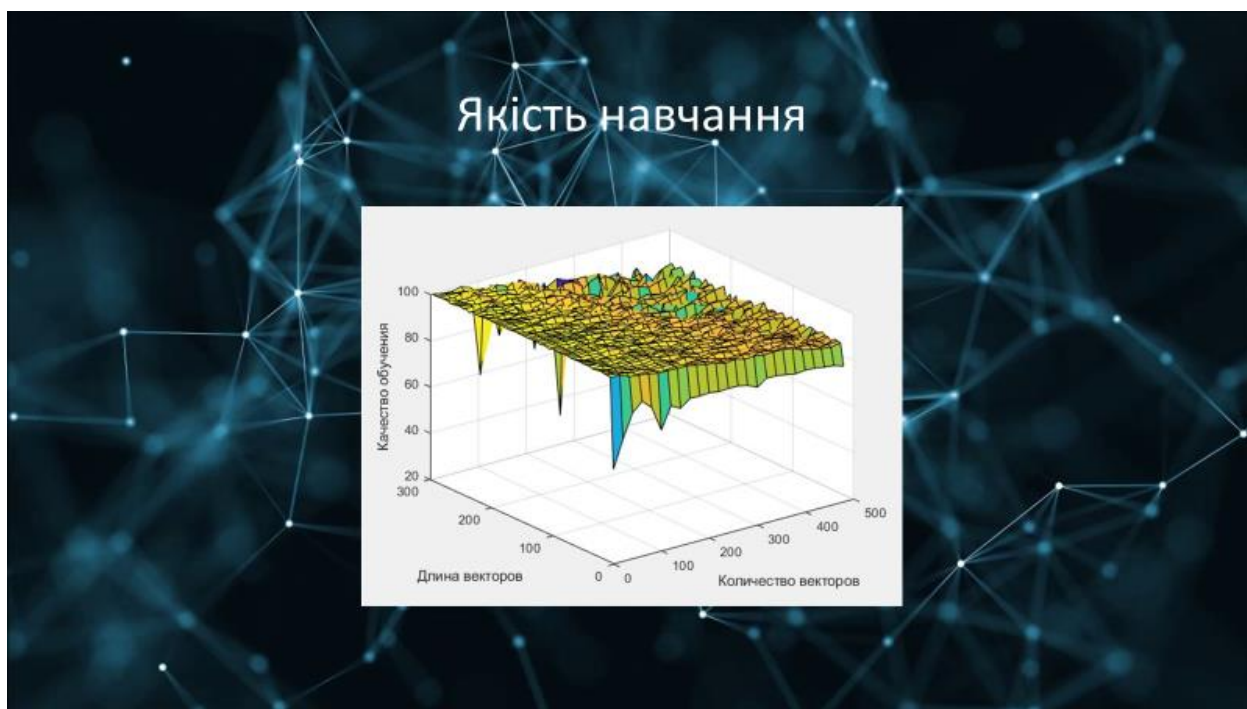


Рисунок А.11 – Якість навчання

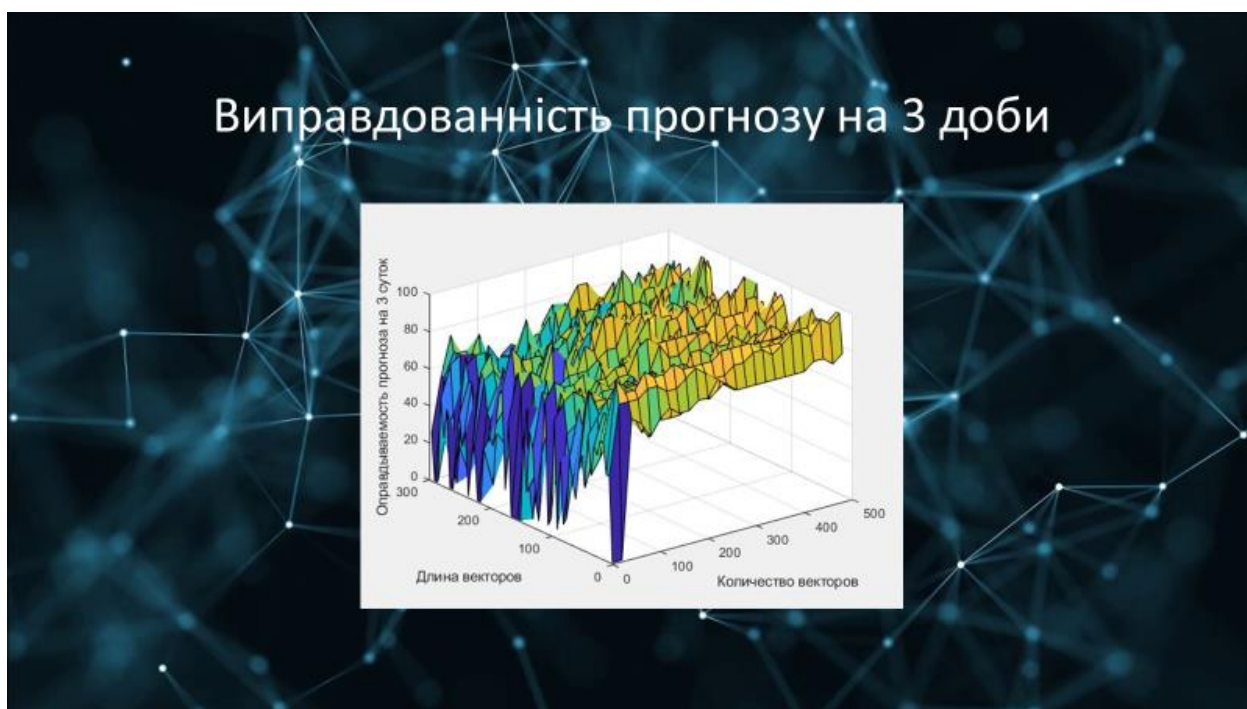


Рисунок А.12 – Виправданість прогнозу на 3 доби



## Ілюстрація якості мережевого навчання

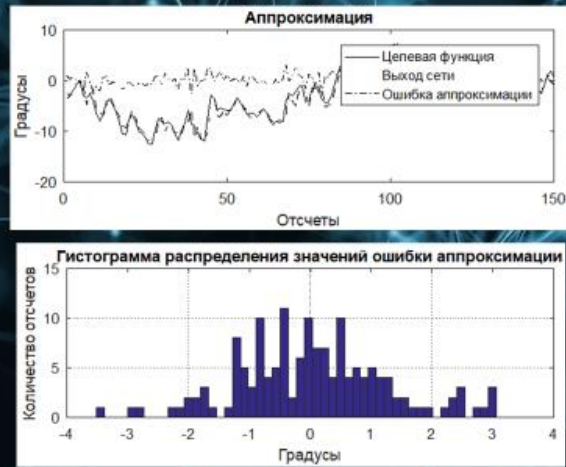


Рисунок А.13 – Ілюстрація якості мережевого навчання

## Тригодинне прогнозування результатів

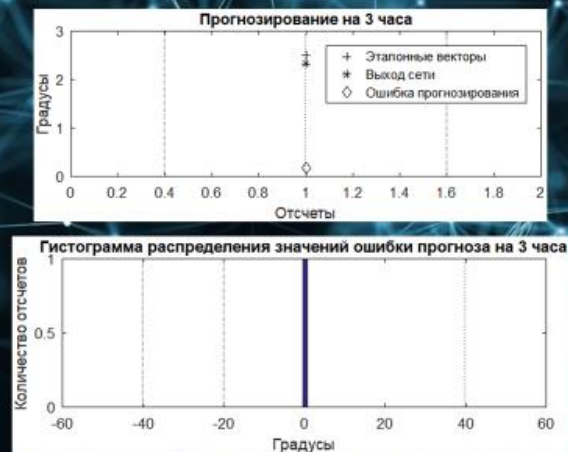


Рисунок А.14 – Тригодинне прогнозування результатів

## Прогнозні результати на один день

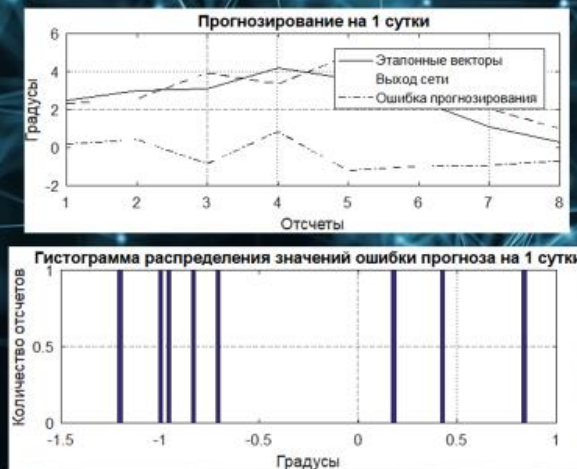


Рисунок А.15 – Прогнозні результати на один день

## Прогнозування результатів на 3 доби

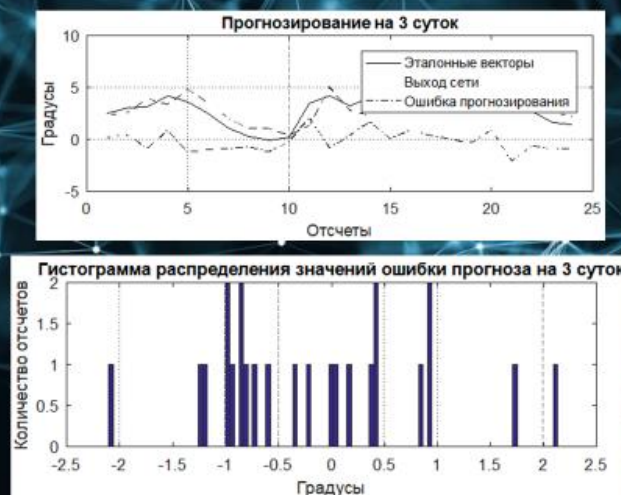


Рисунок А.16 – Прогнозування результатів на 3 доби