

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки


«Затверджую»
Зав. кафедри теоретичної та
прикладної системотехніки
_____ д.т.н., проф. С. І. Шматков
«__» грудня 2022 р.


Пояснювальна записка


до кваліфікаційної роботи
магістра

на тему: «КОМП'ЮТЕРНА МОДЕЛЬ СИСТЕМИ АНАЛІЗУ ТА
ПРОГНОЗУВАННЯ ЦІН НА
ЕНЕРГОНОСІЇ ЗА ДОПОМОГОЮ ЗГОРТКОВИХ НЕЙРОННИ МЕРЕЖ»

Захищено на засіданні
Атестаційної комісії № 45
протокол № __ від __.12.2022 р.
Оцінка _____ / _____
Голова Атестаційної комісії
_____ **МІНУХІН С. В.**

Виконав:
студент 2 курсу, групи КУ– 61
за спеціальністю 151 – Автоматизація
та комп'ютерно-інтегровані технології.
Галузь знань 15 – Автоматизація та
приладобудування
Бурлай Іскандер Сергійович 

Керівник:
к.т.н., доцент, доцент кафедри
теоретичної та прикладної
системотехніки
Бакуменко Ніна Станіславівна 

Рецензент:
д.ф-м.н., професор, професор кафедри
фундаментальної та мовної підготовки
Національного фармацевтичного
університету
Погорєлов Станіслав Вікторович 

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи магістра складається зі вступу, трьох розділів, висновків, списку використаних джерел і чотирьох додатків. Загальний обсяг роботи складає 74 сторінки, із яких 52 сторінки основної частини. Робота містить 17 рисунків, 20 найменувань списку використаних джерел на 3 сторінках.

Метою роботи розробка комп'ютерної моделі системи прогнозування цін на енергоносії за допомогою методів аналізу часових рядів.

Об'єктом дослідження є коливання ціни за барель нафти марки «Brent», протягом 30 років.

Предметом дослідження є моделі аналізу та прогнозування часових рядів.

Для досягнення поставленої мети необхідно реалізувати такі завдання:

- 1) здійснити порівняльний аналіз методів; аналізу часових рядів;
- 2) розробити математичну модель для рішення задачі прогнозування часових рядів цін на енергоносії;
- 3) розробити програмно-алгоритмічну модель вирішення задачі прогнозування цін на енергоносії за допомогою методів аналізу часових рядів.

Під час виконання роботи було застосоване наступне програмне забезпечення: робочий простір Google Colaboratory.

Ключові слова: КОМП'ЮТЕРНА МОДЕЛЬ, ЧАСОВІ РЯДИ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, ПРОГРАМНИЙ ПРОДУКТ, ЕНЕРГОНОСІЇ.

ABSTRACT

The explanatory note to the master's thesis consists of an introduction, three sections, conclusions, a list of used sources and four appendices. The total volume of work is 74 pages, of which 52 pages are the main part. The work contains 17 figures, 20 names of the list of used sources on 3 pages.

The purpose of the work is to develop a computer model of the energy price forecasting system using time series analysis methods.

The object of the research is price fluctuations for a barrel of "Brent" oil over a period of 30 years.

The subject of the study is the analysis and forecasting models of time series.

To achieve the set goal, the following tasks must be implemented:

- 1) perform a comparative analysis of methods; time series analysis;
- 2) develop a mathematical model for solving the problem of forecasting time series of energy prices;
- 3) develop software

an algorithmic model for solving the problem of forecasting energy prices using time series analysis methods.

The following software was used during the work: Google Collaboratory workspace.

Key words: COMPUTER MODEL, TIME SERIES, CONVOLUTIONAL NEURAL NETWORKS, SOFTWARE PRODUCT, ENERGY CARRIERS

ЗМІСТ

| | |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ | 6 |
| ВСТУП..... | 7 |
| РОЗДІЛ 1. ОГЛЯД МЕТОДІВ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ..... | 9 |
| 1.1. Постановка задачі..... | 9 |
| 1.2. Огляд методів аналізу та прогнозування часових рядів..... | 10 |
| 1.2.1. Сингулярний спектральний аналіз..... | 10 |
| 1.2.2. Регресійний аналіз..... | 13 |
| 1.2.3. Модель Хольта-Вінтерса | 17 |
| 1.2.4. Інтегрована модель авторегресії ковзного середнього..... | 19 |
| 1.3. Нейромережеві методи аналізу та прогнозування часових рядів..... | 21 |
| 1.3.1. Загальна характеристика нейромережевих методів..... | 21 |
| 1.3.2. Радіально-базисні нейронні мережі..... | 22 |
| 1.3.3. Рекурентні нейронні мережі..... | 25 |
| 1.3.4. Згорткові нейронні мережі..... | 27 |
| Висновки до розділу 1..... | 29 |
| РОЗДІЛ 2. ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ..... | 31 |
| 2.1. Основні характеристики часових рядів..... | 31 |
| 2.2. Архітектура згорткової нейронної мережі..... | 33 |

| | |
|--|----|
| 2.2.1. Вхідний шар..... | 33 |
| 2.2.2. Згортковий шар..... | 33 |
| 2.2.3. Агрегувальний шар..... | 38 |
| 2.2.4. Повнозв'язний шар..... | 38 |
| 2.2.5. Вихідний шар..... | 39 |
| 2.3. Гіперпараметри згорткової нейронної мережі і методи підбору їх значень..... | 39 |
| 2.3.1. Гіперпараметри стохастичного градієнта..... | 39 |
| 2.3.2. Гіперпараметри моделі..... | 42 |
| 2.3.3. Методи підбору гіперпараметрів..... | 43 |
| Висновки до розділу 2..... | 44 |
| РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ..... | 46 |
| 3.1. Покрокова реалізація..... | 46 |
| Висновки до розділу 3..... | 52 |
| ВИСНОВКИ..... | 54 |
| ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 56 |
| ДОДАТОК А. ЗАВДАННЯ НА ДИПЛОМНУ РОБОТУ..... | 59 |
| ДОДАТОК Б. ІНДИВІДУАЛЬНЕ ТЕХНІЧНЕ ЗАВДАННЯ..... | 61 |
| ДОДАТОК В. ПРОГРАМА І МЕТОДИКА ВИПРОБУВАНЬ..... | 65 |
| ДОДАТОК Г ЛІСТИНГ..... | 70 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

АД - аналіз даних,

ЧР – часовий ряд,

ССА – сингулярний спектральний аналіз,

ARIMA – модель авторегресії-проінтегрованого ковзного середнього,

РБНМ – радіально-базисні нейронні мережі,

РНМ – рекурентні нейронні мережі,

ЗНМ – згорткові нейронні мережі,

МО – матриця ознак,

ФВ – функція втрат,

ГП – гіперпараметри,

НМ – нейронна мережа,

АКВФ – автоковаріація,

АКФ – автокореляційна функція,

ЧАКФ – часткова автокореляційна функція.

ВСТУП

Ми живемо в інформаційну еру і нещодавно (у минулому столітті) наша цивілізація досягла такого рівня розвитку, що ми стали оперувати шаленими об'ємами даних. Саме тому з'явилася проблема обробки цих даних, оскільки опрацювати їх вручну стало просто неможливо. В результаті з'явилася ціла наукова дисципліна, що займалася цією задачею – аналіз даних (далі - АД).

Приблизно в той же час, людство почало конструювати пристрій, який згодом, через багато років роботи, перетвориться на усім добре відомий комп'ютер. Після певного періоду розвитку, комп'ютер отримав достатні обчислювальні можливості, які б дозволили значно допомогти в вирішенні задач АД. На даний момент, ця дисципліна є однією з найбільш популярних в сфері інформаційних технологій.

У рамках нинішньої економічної моделі – ринкової, чималу увагу приділяють відстеженню динаміки коливання значень економічних показників. Для кращого сприйняття зміни цих значення використовують графічне представлення – часові ряди (далі – ЧР), котрі дають можливість побачити їх ріст, падіння, невеликі коливання і т.д., протягом певного періоду часу.

Задача аналізу та прогнозування ЧР також відноситься до АД. Крім цього, методів, що дозволили б вирішити дану задачу існує безліч, кожен рік з'являються нові. В даній роботі будуть розглянуті найбільш відомі й ефективні методи даного напрямку, які мають різноманітне походження і розвивалися незалежно один від одного. Також буде розглянуто ціле сімейство методів, які почали активно розвиватися та набули широкого поширення в останні роки – нейромережеві.

І саме з цього сімейства, для вирішення поставленої задачі, мною і було обрано метод згорткових нейронних мереж. Вибір було здійснено на основі

його характерних рис функціонування, переваг та недоліків, які було детально розглянуто під час виконання кваліфікаційної роботи.

РОЗДІЛ 1

ОГЛЯД МЕТОДІВ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

1.1. Постановка задачі

Метою даної роботи є розробка комп'ютерної моделі системи прогнозування цін на енергоносії за допомогою методів аналізу часових рядів.

Об'єктом дослідження є коливання ціни за барель нафти марки «Brent», протягом 30 років.

Предметом дослідження є моделі аналізу та прогнозування часових рядів.

Для досягнення поставленої мети необхідно реалізувати наступні завдання:

- 1) здійснити порівняльний аналіз методів; аналізу часових рядів;
- 2) розробити математичну модель для рішення задачі прогнозування часових рядів цін на енергоносії;
- 3) розробити програмно-алгоритмічну модель вирішення задачі прогнозування цін на енергоносії за допомогою методів аналізу часових рядів.

Практична значимість роботи базується на поточній моделі ринкової економіки. Тому для того, щоб комфортно функціонувати в таких умовах суб'єкти економіки мають аналізувати та прогнозувати певні економічні показники, у тому числі і ціни на певні категорії товарів. В яких зацікавлений той чи інший учасник ринку з метою оптимізації витрат чи отримання надприбутків.

1.2. Огляд методів аналізу часових рядів

1.2.1. Сингулярний спектральний аналіз

Метод сингулярного спектрального аналізу (англ. Singular spectrum analysis, далі - ССА) є непараметричним методом спектрального оцінювання. Він є своєрідним синтезом класичного аналізу часових рядів, багатовимірної статистики, багатовимірної геометрії, динамічних систем і обробки сигналів.

ССА дозволяє вирішувати широкий спектр задач, а саме:

- знаходження трендів різної роздільності;
- згладжування;
- вилучення сезонності компоненти;
- одночасне виділення циклів з малим і великим періодами;
- виділення періодичностей зі змінними амплітудами;
- одночасне виділення складних трендів і періодичностей;
- знаходження структури в коротких часових рядах;
- виявлення точки зміни.

Метод ССА складається з двох взаємодоповнюючих стадій: декомпозиції та реконструкції, і обидва з них включають два окремі етапи. На першому етапі ми розкладаємо ряд і на другому етапі реконструюємо оригінальний ряд і використовуємо реконструйований ряд (без білого шуму) для прогнозування нових точок даних [1].

Перша стадія:

- 1) Вбудовування - це своєрідне відображення, що передає одновимірний часовий ряд $Y_T = (y_1, \dots, y_T)$ у багатовимірний ряд X_1, \dots, X_T з векторами $X_i = (y_i, \dots, y_{i+L-1}) \in RL$, де $K = T - L + 1$. Вектори X_i зветься векторами з відставанням. Єдиний параметр – довжина вікна L , це ціле число, яке $2 \leq L \leq T$. В результаті такого кроку ми отримуємо матрицю траєкторії $X = [X_1, \dots, X_K] = x_{ij}^{L,K}$.
- 2) Сингулярне розкладання. На цьому етапі виконується декомпозиція сингулярного значення матриці траєкторії та представить її як суму однорангових бі-ортогональних елементарних матриць.
- 3) Позначимо через $\lambda_1, \dots, \lambda_L$ власні значення XX' у зменшенні порядку величини ($\lambda_1 \geq \dots \lambda_L \geq 0$) та за U_1, \dots, U_L ортонормованою системою власних векторів матриці XX' цих власних значення. (U_i, U_j) є скалярним добутком векторів U_i та U_j та $|U_i|$ та є нормою вектора U_i .

Друга стадія:

- 1) Групування. У цьому етапі відбувається розбивання елементарних матриць X_i на кілька груп, а після додаються матриці у кожній групі. Нехай $I = \{i_1, \dots, i_p\}$ — група індексів i_1, \dots, i_p . Тоді матриця X_I , відповідна групі I та визначається як:

$$X_i = X_{I_1} + \dots + X_{I_m}. \quad (1.1)$$

Порядок підбору комплектів I_1, \dots, I_m називається групуванням власних потрійок. Для даної групи I внесок компоненти X_i у розкладі вимірюється часткою відповідних власних значень: $\sum_{i \in I} \lambda_i / \sum_{i=1}^d \lambda_i$.

2) Діагональне усереднення. Переводить кожен матрицю I в часовий ряд, який є адитивний компонент початкового ряду Y_T . Якщо z_{ij} означає елемент a матриці Z , то k -й член результуючого ряду виходить шляхом усереднення z_{ij} по всіх i, j таких, що $i + j = k + 2$. Ця процедура називається діагональним усередненням, або ганкелізацією матриці Z . Результат ганкелізації матриці Z — матриця Ганкеля HZ , яка є матрицею траєкторії, що відповідає рядам, що отримані в результаті діагонального усереднення. Зверніть увагу, що ганкелізація

Застосовуємо процедуру ганкелізації до усіх складових матриці:

$$\tilde{X} = \tilde{X}_{I_1} + \dots + \tilde{X}_{I_m}, \quad (1.2)$$

де $\tilde{X}_{I_1} = HX$.

В свою чергу, це еквівалентно розкладанню оригінального ряду $Y_T = (y_1, \dots, y_T)$ у суму m рядів:

$$y_t = \sum_{k=1}^m \tilde{y}_t^{(k)}, \quad (1.3)$$

де $\tilde{Y}_T^{(k)} = \tilde{y}_1^{(k)}, \dots, \tilde{y}_T^{(k)}$ відповідає матриці X_{i_k} .

Переваги ССА:

- Легко виділяємий тренд;
- Висока точність аналізу;

Недоліки:

- Відносно повільний метод;
- Важкі розрахунки [2].

1.2.2. Регресійний аналіз

Регресійний аналіз це метод, який дає можливість моделювати вимірювані дані й досліджувати їх властивості. Як в інших випадках, дані складаються з пари значень – залежної і незалежних змінних. Тобто, дана модель є функцією двох вищезгаданих змінних.

Постановка задачі регресійного аналізу. Маємо дані, у вигляді певної сукупності результатів певних спостережень. У ній один стовбець містить значення деякого показника, котрому потрібно встановити функціональну залежність з параметрами об'єкту та середовища, котрі представлені показниками інших стовбців. Треба знайти кількісний взаємозв'язок між показником і факторами. Звідси, можемо описати задачу :

$$y = f(x_2, x_3, \dots, x_m), \quad (1.4)$$

де y – залежна змінна;

x_1, x_2, \dots, x_m – незалежні змінні.

Вирішуючи поставлену задачу, кращим варіантом буде оформити розв'язок в 4 етапи:

- 1) Здійснити початкову обробку наявних даних;
- 2) Обрати потрібний вид регресійного рівняння;
- 3) Обчислити коефіцієнти вищеобраного рівняння;
- 4) Переконалися в адекватності функції, що була побудована до відповідних даних.

Знаходження по виборці об'ємом n оцінки невідомих коефіцієнтів регресії b_0, b_1, \dots, b_k є головною задачею регресійного аналізу. Також необхідно виконати наступні дії:

- розрахувати найкращі оцінки невідомих параметрів b_0, b_1, \dots, b_k ;
- провести перевірку статистичних гіпотез щодо параметрів моделі;
- перевірити адекватність моделі відносно наявних даних

Для того щоб, побудувати модель множинної регресії необхідно:

- 1) Обрати рівняння регресії;
- 2) Визначити параметри рівняння;
- 3) Проаналізувати якість рівняння та його адекватність.

Існує велика кількість різновидів регресійного аналізу, перерахуємо найбільш відомі:

Лінійна регресія – графічно представлена у вигляді прямої, при цьому точки емпіричної лінії знаходяться по обидві сторони від прямої, настільки близько, наскільки це можливо. Її рівняння можна представити у вигляді:

$$y = a + bx + E, \quad (1.5)$$

де a, b – шукані коефіцієнти;

x -аргумент;

y -функція;

E – це змінна, котра характеризує той вплив, що мають невраховані фактори в моделі.

Гіперболічна регресія. Це залежність, яка графічно представлена у вигляді гіперболи. І може бути представлена у вигляді рівняння виду:

$$y = a + b/x + E \quad (1.6)$$

Параболічна регресія. Ця залежність графічно представлена у вигляді параболи. В загальному випадку можна описати наступним чином:

$$y = ax^n + bx^{n-1} + cx^{n-2} + \dots + kx + 1 \quad (1.7)$$

Хоча на практиці зазвичай зустрічається частковий випадок параболи другого порядку:

$$y = ax^2 + bx^1 + c \quad (1.8)$$

Логарифмічна лінійна регресія, іноді подвійна логарифмічна модель. Може бути представлена у вигляді рівняння:

$$\ln a + b \ln x + \ln E \quad (1.9)$$

Якщо замінити змінні $\tilde{y} = \ln y$; $\tilde{x} = \ln x$; $\tilde{a} = \ln a$, то вийде звичайне лінійне рівняння $\tilde{y} = \tilde{a} + \tilde{b}\tilde{x}$.

Переваги регресійного аналізу:

- Простота розрахунків;
- Швидко отримуємо результати;
- Легко можна перевірити проміжні результати розрахунків;

Недоліки:

- Важко визначається вид функціональної залежності;
- Низька адаптивність;
- Неможливо моделювати нелінійні процеси [3].

1.2.3. Модель Хольта-Вінтерса

Метод Хольта-Вінтерса побудований на основі методу Хольта - метод двопараметричного експоненціального згладжування [4]. Даний метод базується на трьох ключових елементах, які як передбачаються змінюються впродовж часу, відповідно до значень цих компонентів і застосовують згладжування.

Ці компоненти:

- базова компонента – експоненціально згладжена величина за останній період;
- лінія тренду – лінія, що проведена між екстремумами в напрямку тренду;
- сезонний ефект – величина, що відображає певні періодичні коливання ЧР.

Метод Хольта-Вінтерса можна застосовувати на різних рівнях планування.

В даному методі, як і в методі, на якому він був заснований, є певні коефіцієнти α, β, γ , що застосовуються до поточного значення прогнозу і в результаті отримуємо прогноз на наступний період. Значення даних коефіцієнтів коливаються від 0 до 1. Далі сумуємо усі компоненти і отримуємо прогноз:

$$\hat{y}_{t+1} = a_1 + b_1 + c_{t+1-m}, \quad (1.10)$$

де a_t – базова компонента;

b_t – лінія тренду;

c_t – сезонний ефект;

m – період сезону.

Отримуємо нову оцінку базової компоненти в результаті її корегування враховуючи сезонну компоненту. Далі необхідно врахувати і лінію тренду, для чого додаємо прогноз тренду до коефіцієнту базової лінії:

$$a_1 = \alpha (y_1 - c_{t-m}) + (1-\alpha)(a_{t-1} + b_{t-1}) \quad (1.11)$$

Після чого розраховуємо нову оцінку тренду:

$$b_1 = \beta (a_t - a_{t-1}) + (1 - \beta)b_{t-1} \quad (1.12)$$

Наступним кроком буде розрахунок нової оцінки сезонної компоненти як:

$$c_1 = \gamma (y_t - a_t) + (1 - \gamma)c_{t-m} \quad (1.13)$$

Варто відмітити, що на першому кроці рахувати слід тільки сезонний ефект, оскільки значення базової компоненти і тренду не можливо розрахувати напочатку.

На другій ітерації маємо згладжене значення базової компоненти, яке відповідає спостерігаємому значенню. Щодо мікротренду, то він на даному кроці є лінійним.

Починаючи з третьої, усі подальші ітерації значення компонент розраховують за формулами (1.11) - (1.13).

Переваги методу Хольта-Вінтерса:

- Простота аналізу;
- Немає обмежень на об'єм даних ;
- Простота розрахунків;

Недоліки:

- Негнучкий алгоритм;
- Потрібно точно налаштовувати згладжувальні функції. [5]

1.2.4. Інтегрована модель авторегресії ковзного середнього

ARIMA – (англ. Autoregressive Integrated Moving Average) - це інтегрована модель авторегресії ковзного середнього. Цей метод заснований на базі методу ARMA. На відміну від останнього дозволяє працювати з нестационарними ЧР, котрі він за допомогою операції інтегрування перетворює на стаціонарні.

Модель ARIMA(k,p,q) певного процесу x_t , можна сформулювати наступним чином:

$$x_t^k = \alpha_1 x_{t-1}^k + \alpha_2 x_{t-2}^k + \dots + \alpha_p x_{t-p}^k + \delta - \theta_1 \delta_{t-1} - \dots - \theta_q \delta_{t-q}, \quad (1.14)$$

де α – деякий числовий коефіцієнт, що не більше в абсолютній величині за 1;

$\theta_1, \dots, \theta_q$ – кінцевий набір параметрів β ;

$\delta_1, \delta_2, \dots$ – це послідовність випадкових величин або «білий шум».

При цьому:

$$x_t^k = \Delta^k x_t = x_t - C_k^1 x_{t-1} + C_k^2 x_{t-2} - \dots + (-1)^k x_{t-k} \quad (1.15)$$

де C_{t-k}^k - число поєднань;

Δ - різницевий оператор.

Дуже важливим моментом в цьому методі є чітке визначення параметрів моделі ARIMA(p,q,k). На початку потрібно знайти порядок моделі – k, це робиться за допомогою оцінки дисперсії - $\hat{\sigma}^2(k)$, коливання значень котрої необхідно відслідкувати. Далі беручи до уваги поведінку автокореляційної та часткової автокореляційної функцій, визначаються параметри моделі p,q.

Переваги методу:

- Пряме математично-статистичне обґрунтування;
- Висока гнучкість алгоритму відносно ЧР;
- Один з тих методів, що має найбільш точний прогноз.

Недоліки ARIMA:

- Для коректної роботи необхідно мати 40 прогнозів;
- Важко підібрати найкращі параметри моделі;
- При отриманні нових даних, як мінімум необхідно перераховувати модель. [6]

1.3. Нейромережеві методи аналізу та прогнозування часових рядів

1.3.1. Загальна характеристика нейромережевих методів

Штучні нейронні мережі (або просто нейронні мережі) – технологія, що базується на тих же принципах, що і нейронні мережі мозку людини. Алгоритм належить до області штучного інтелекту. «Підвалини» даного методу були закладені нейрофізіологом Маккаллохом і математиком Піттсом. Вперше нейронну мережу вдалося запустити через 11 років, в 1954 році вченими Фарлі і Кларку.

Надалі нейронні мережі розвивалися і на даний момент це не окремий метод, що імітує роботу головного мозку людини, а ціле сімейство методів. Деякі з них в рамках поставленої задачі було розглянуто нижче. Варто відмітити, що нейронні мережі придатні не лише для прогнозування часових рядів, а використовуються для розпізнавання образів, пошуку бомб, вирішення проблеми паралельних обчислень тощо [7].

1.3.2. Радіально-базисні нейронні мережі

Радіально-базисні нейронні мережі (далі - РБНМ) – це штучні нейронні мережі, котрі належать до мереж прямого розповсюдження. Характерним рисою цього методу, на відміну від інших мереж цього типу є те, що навчання відбувається без зворотнього розповсюдження помилки [8].

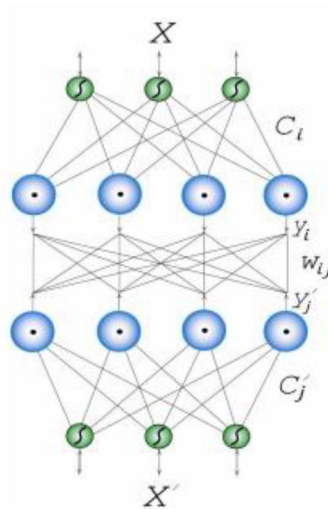


Рисунок 1.1 – Структурна схема РБНМ

де $X_t = (x_1, x_2, \dots, x_N)$, $X'_t = (x'_1, x'_2, \dots, x'_M)$ - послідовність векторів параметрів різних ознакових просторів;

C_i, C'_j - вектори координат відповідних центрів радіально-базисних елементів;

w_{ij} - вагові коефіцієнти асоціативного зв'язку радіально-базисних елементів;

σ_i, σ_j' - параметри, що відповідають за розмір відповідного радіального елемента;

y_i, y_j' - виходи радіальних елементів, які обчислюються як:

$$y_i = \exp\left(\frac{\|x_t - c_i\|^2}{2\sigma_i^2}\right), \quad y_j' = \exp\left(\frac{\|x_t' - c_j'\|^2}{2\sigma_j'^2}\right) \quad (1.15)$$

Алгоритм навчання:

- 1) За допомогою методів кластерного аналізу множин значень векторів параметрів визначають параметри радіально-базисних елементів $(C_i, C_j', \sigma_i, \sigma_j')$.
- 2) Обчислюються значення коефіцієнтів асоціативних зв'язків. Спершу розраховуємо значення y_i, y_j' по формулам (1.15). Далі обчислюємо w_{ij} :

$$w_{i_{\text{поб}}j}^{\text{нов}} = (w_{i_{\text{поб}}j}^{\text{стар}} + y_{i_{\text{поб}}} y_j')/2, \quad w_{ij_{\text{поб}}}^{\text{нов}} = (w_{ij_{\text{поб}}}^{\text{стар}} + y_j' y_i)/2 \quad (1.16)$$

Далі обчислені значення w_{ij} нормуються наступним чином:

$$w_{ij}^{\text{нов}} = \frac{w_{ij}^{\text{стар}}}{\max_j(w_{ij})} \quad (1.17)$$

Остання операція проводилася для того, щоб прибрати вплив нерівномірності навчальної вибірки.

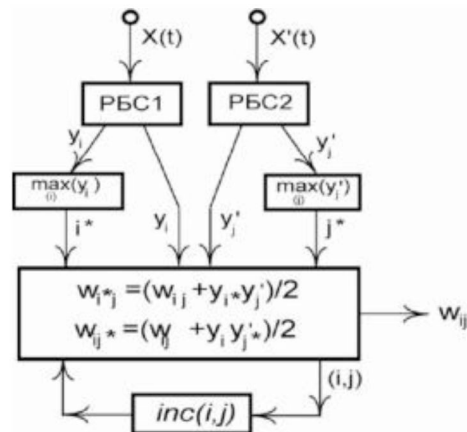


Рисунок 1.2 – Алгоритм навчання РБНМ

Переваги РБНМ:

- Гарна масштабованість;
- Багатонаправленість;
- Ітеративність алгоритму, що дозволяє швидко отримати приблизний результат, поступово приближаючись до більш точного значення;
- Прозорість структури.

Недоліки:

- Велика кількість розрахунків;
- Працюють повільніше ніж інші мережі цього ж типу [9].

1.3.3. Рекурентні нейронні мережі

Рекурентні нейронні мережі (далі - РНМ) – це штучні нейронні мережі в яких зворотні зв'язки розташовуються від виходу шарів до входів, на відміну від самоорганізаційних мереж, в яких ці зв'язки розташовуються між нейронами одного шару. Характерною рисою цих мереж є наявність внутрішньої пам'яті.

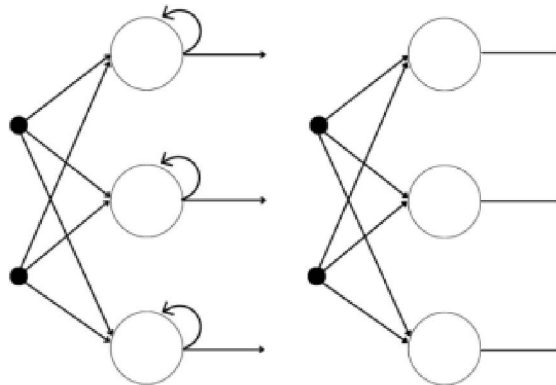


Рисунок 1.3 – Різниця між РНМ та нейронною мережею з прямим зв'язком

Модель РНМ можна представити в двох варіантах, перший варіант:

$$U_i(t + 1) = \sigma(\sum_j w_{ij} U_j(t) - h_i) \quad (1.18)$$

де σ – сигмоїдальна функція;

$U_j(t)$ – стан j -го нейрону в момент часу t ;

U_{ij} – матриця зв'язків між нейронами;

$i, j = 1, \dots, N$;

h_i – пороги.

Також є другий варіант представлення мережі, який можна представити у наступному вигляді:

$$\frac{dU_i}{dt} = \sigma(\sum_j w_{ij} U_j(t) - h_i) - \lambda_i U_i \quad (1.19)$$

І перша мережа (1.18), й друга мережа (1.19) дають можливість описати як коливальні режими, так і прагнення до рівноваги.

У РНМ є деякі модифікації:

- 1) Мережі Ельмана – це двохшарова мережа, де перший шар є тангенціальним, а другий є лінійним. В даній модифікації зворотній зв'язок йде від виходу першого шару на його вход, при цьому вона проходить через елемент затримки і матрицю вагів;
- 2) Мережі Хопфілда – це одношарова мережа, характерною рисою цієї модифікації є наявність лінійно-порогової функції активації. В зворотньому зв'язку як і в попередній модифікації, присутній елемент затримки.

Переваги методу:

- Підходить для моделювання послідовностей;
- Висока здатність до узагальнення.

Недоліки:

- Низька стійкість;
- Висока складність;
- Досить повільний [10].

1.3.4. Згорткові нейронні мережі

Згорткові нейронні мережі (далі - ЗНМ) – це сімейство штучних нейронних мереж. Дана архітектура була розроблений Яном Лекуном в 1980-х.

ЗНМ спеціалізується на роботі з такими даними, котрі можна структурувати у вигляді сітки. Це сімейство було розроблене для роботи з двох- і трьохвимірними даними. Пізніше було виявлено, що ЗНМ також ефективні і при роботі з одновимірними даними – часовими рядами.

Спочатку отримав визнання як ефективний метод для розпізнавання та класифікації зображень, пізніше продовжив розвиватися і виявився ефективним в широкому спектрі задач.

Задача моделювання послідовності раніше (і зараз частково) вирішувалася в основному за допомогою РНМ, але були проведені дослідження і виявилось, що ЗНМ складають серйозну конкуренцію РНМ. А саме в створенні аудіо, переклад і прогнозування часових рядів.

За назвою неважко здогадатися, що сімейство ЗНМ базується на дискретному операторі згортки. Який «створює вихідні дані: карту функцій f ,

перемістивши ядро w на вхід x . Кожен елемент у вихідній карті ознак отримується шляхом підсумовування до результату поелементного множення між фрагментом вхідних даних, що має і ядром».

В рамках даної роботи розглядається саме задача прогнозування часових рядів, то наведемо відповідну модель :

$$f(i) = (x * w)(i) = \sum_{j=0}^{k-1} x(i-j)w(j) \quad (1.20)$$

де f – карта функцій;

x – вхід;

i, j – номери елементу згортки;

w – ядро;

k – розмір ядра;

Також слід відмітити, що рівняння (1.20) легко масштабується, тому не є проблемою збільшити його розмірність.

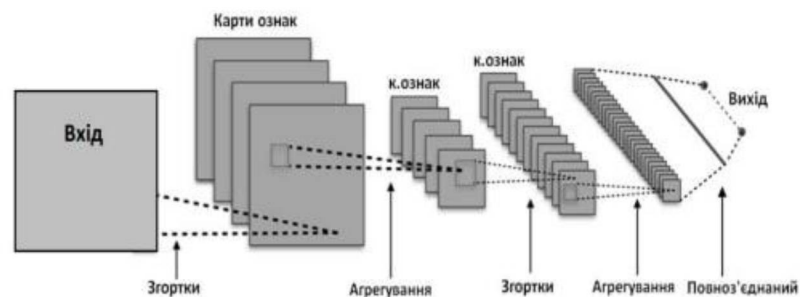


Рисунок 1.4 – Приклад архітектури ЗНМ

Переваги ЗНМ:

- Локальна зв'язність – підмножина вхідних нейронів тісно пов'язана прихованими нейронами;
- Конкретне ядро застосовують для кожного розташування;
- Часова ефективність, котра обумовлена невеликою кількістю параметрів, ця властивість є наслідком вищезгаданих локальної зв'язності та спільного доступу до параметрів;
- Простота реалізації;
- Можливо організувати автоматичний підбір параметрів;
- Навчання можливо проводити за допомогою різних методів.

Недоліки:

- Потрібно чимало часу на підбір параметрів;
- Велика кількість навчальних матеріалів [11].

Висновки до розділу 1

Таким чином, я розглянув декілька, а саме 8 методів, що мають різне походження, від регресійного аналізу до нейронних мереж. Вони дозволяють вирішувати поставлену задачу роботи з часовими рядами. Тому, спираючись на перераховані характерні риси, переваги та недоліки переглянутих методів для вирішення задачі аналізу та прогнозування часових рядів, я обрав згорткові нейронні мережі.

Даний вибір базується на ряді властивостей ЗНМ, таких як локальна зв'язність і спільний доступ до параметрів, котрі в свою чергу зумовлюють високу часову ефективність. Також, було врахована і простота реалізації, котра відкриває шлях до можливих модифікацій даного методу, задля покращення його показників ефективності.

Окремо хочу відмітити, що при реалізації даного методу, є можливість для реалізації автоматичного підбору параметрів моделі. Опираючись на власний досвід роботи з подібнимим методами, враховуючи, у тому числі і мою кваліфікаційну роботу бакалавра, де було складно підібрати параметри для інтегрованої моделі авторегресії ковзного середнього.

РОЗДІЛ 2

ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ

2.1. Основні характеристики часових рядів

До того як перейти до ЗНМ, треба розглянути певні фундаментальні речі стосовно ЧР. До них можна віднести визначення ЧР і його розклад, що таке автоковаріаційна функція (далі –АКФ) і часткова автоковаріаційна функція (далі –ЧАКФ).

ЧР – це послідовність значень однієї величини, в хронологічному порядку, взятих через однакові проміжки часу. Прикладом ЧР є ВВП України, висота приливів, ціна на природній газ тощо. [12]

Зазвичай формалізація моделі відбувається завдяки розкладі ЧР на окремі компоненти, що відрізняються одна від одної, описуючи принципово різні аспекти ЧР. Аналіз кожної з них відбувається індивідуально, відповідно до їх природи. ЧР можна розкласти наступним чином:

$$y_t = tr_t + s_t + r_t, \quad (2.1)$$

де y_t – ЧР;

tr_t - тренд – це складова, що характеризує довгострокові процеси;

s_t – сезонна компонента – описує періодичні процеси;

r_t – випадкова компонента – характеризує випадкові процеси.

Автоковаріаційна функція (далі – АКВФ) $\gamma(\tau)$ - це функція, що розраховує коваріацію одного й того ж процесу. У загальному випадку це:

$$\gamma(\tau) = cov(y_t, y_{t+\tau}) = E[(x(t) - a)(y_t - a)(y_{t+\tau} - a)]. \quad (2.2)$$

де τ – величина зсуву;

cov – коваріація;

a – середнє значення стаціонарного процесу.

Якщо величина зсуву τ дорівнює нулю, то в такому випадку ми отримаємо окремий випадок, що дозволяє розрахувати дисперсію ряду:

$$\gamma(0) = E(x(t) - a)^2 \quad (2.4)$$

АКФ – це функція, що дозволяє розрахувати наскільки значення ЧР в середньому залежать від минулих значень. [13] І коефіцієнт $r(\tau)$ визначається в наступному вигляді:

$$r(\tau) = \frac{E[(y_t - a)(y_{t+\tau} - a)]}{[E(y_t - a)^2 E(y_{t+\tau} - a)^2]^{1/2}} = \frac{\gamma(\tau)}{\gamma(0)}, \quad (2.4)$$

де $r(\tau)$ – коефіцієнт автокореляції.

АКФ називається процес спостереження за динамікою зміни коефіцієнту автокореляції та його аналіз.

ЧАКФ – це функція, що дозволяє виявляти екстремуми на певному відрізку ЧР. [14] І ЧАКФ першого порядку визначається як:

$$r_{\text{част}}(2) = r(y_t, y_{t+2} | y_{t+1} = \alpha) = \frac{r(2) - r^2(1)}{1 - r^2(1)}, \quad (2.8)$$

2.2. Архітектура згорткової нейронної мережі

2.2.1. Вхідний шар

Вхідний шар є найпростішим шаром в будь-якій нейронній мережі і ЗНМ не виключення. Єдина роль, яку виконує цей шар це передача певного часового ряду з деякою сталою довжиною на наступний рівень, в даному випадку до згорткового шару.

2.2.2. Згортковий шар

Загальний випадок

В основі згорткової нейронної мережі, як неважко здогадатися, лежить операція згортки. Ідея полягає в тому, що в загальному випадку ми маємо матрицю ваг, як правило маленького чи середнього розміру – вона є ядром

операції. Проводять поелементне множення певного «шматку» вхідного сигналу і на виході отримуємо суму усіх значень у вигляді деякого пікселя.

Цю операції необхідно повторювати для усіх елементів, через які рухається ядро. Таким чином, поступо із однієї матриці ознак (далі - МО) ми отримуємо другу.

В ході операції згортки краї матриці «видаляються» і з МО певною розмірності ми отримуємо нову МО меншої розмірності. Приклад операції згортки і процесу створення нового ядра можна побачити на рис 2.1 нижче.

| | | | | |
|----------------|----------------|----------------|---|---|
| 3 ₀ | 3 ₁ | 2 ₂ | 1 | 0 |
| 0 ₂ | 0 ₂ | 1 ₀ | 3 | 1 |
| 3 ₀ | 1 ₁ | 2 ₂ | 2 | 3 |
| 2 | 0 | 0 | 2 | 2 |
| 2 | 0 | 0 | 0 | 1 |

| | | |
|------|------|------|
| 12.0 | 12.0 | 17.0 |
| 10.0 | 17.0 | 19.0 |
| 9.0 | 6.0 | 14.0 |

Рис. 2.1 – Приклад згортки і процес створення ядра.

Перша МО має розмірність 5×5 , тому ми маємо 25 елементів матриці, себто вхідних ознак. В другій МО розмірність 3×3 й залишається лише 9 елементів матриці, тобто ми отримуємо лише 9 вихідних ознак.

Окремий випадок

У ситуації, коли ЗНМ працює з одновимірним часовий рядом робота згорткового шару і власне згортки спрощується. Оскільки ми маємо справу з окремим випадком, його буде розглянуто нижче.

Через вхідний шар згортковий шар отримує деякий вектор $x \in R^p$ і фільтр $w \in R^k$, де $k \leq p$. Для кожного сусідньої множини k елемент $x[i:i+k]$ береться $w^T x[i:i+k]$ і в результаті ми отримуємо один вузол шару згортки.

Приклад роботи: нехай дано вхідний вектор $x \{5; 3; 2; 4; 8; 1\}$ і фільтр $w\{3; 3; 2\}$. Ми застосовуємо фільтр w до перших трьох елементів вхідного вектору x - $w^T x = 5(3) + 3(3) + 1(2) = 26$. Далі ми повторюємо подібні операції з наступними підмножинами вхідного вектору $\{3; 2; 4; 8\}$, $\{2; 4; 8; \}$ і т.д. Приклад першої операції на рис 2.2 нижче.

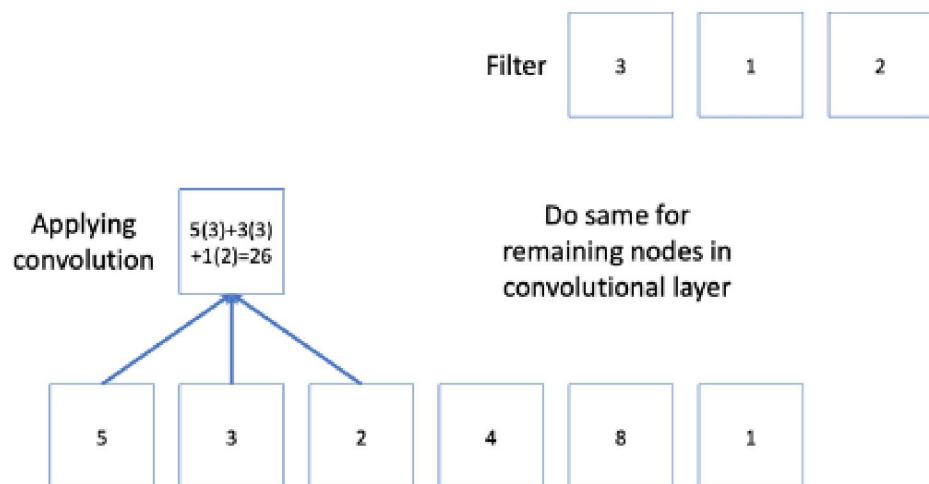


Рис. 2.2 – Приклад роботи одновимірної згортки.

Якщо є потреба, то згортковий шар може мати декілька наборів фільтрів(застосовуючи один конкретний фільтр до однієї конкретної підмножини).

Додаткові засоби

Варто зазначити, що іноді потрібно мати однакові розміри вхідного і вихідного сигналів, але крайні пікселі не стають центральними у ядрі і не має можливості цього досягнути. Для цих цілей придумали засіб - падинг, що дозволяє «обходити» цей недолік. Приклад застосування падингу зображено на рис. 2.3.

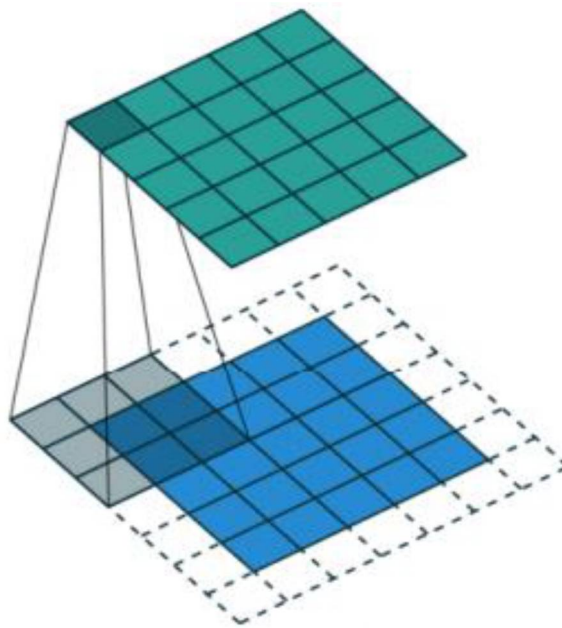


Рис. 2.3 – Застосування падингу.

Ідея падингу доволі проста – необхідно на час операції згортки «приклеїти» несправжні пікселі на краї. Майже завжди вони мають нульові

значення. Такими маніпуляціями справжні крайні пікселі опиняються у центрі і досягається однакова розмірність сигналів на вході і виході.

Буває і протилежна потреба – необхідно мати вхідний сигнал більший за вихідний. У сімейства ЗНМ це частий випадок, оскільки в них є властивість, що чим більша кількість каналів, тим менша розмірність. Вирішити цю проблему можна двома способами, перший - застосування спрайдингу, а другий - це застосування агрегуючого шару. Застосування страйдингу можна побачити на рисунку 2.4.

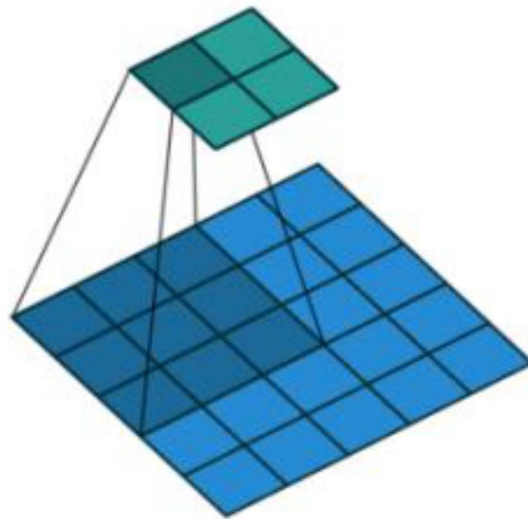


Рис. 2.4 – Застосування страйдингу.

Можна сказати, що ідея страйдингу в якійсь мірі протилежна ідеї падингу. Якщо перший під час згортки «додає» несправжні пікселі, то другий ігнорує деякі пікселі, з певною частотою. Страйд має мати певний розмір, як правило 2 і більше, бо при розмірі в 1 страйд вироджується і ми отримуємо стандартний результат згортки.

Якщо ми маємо страйд розміром n , то ми обираємо чергове розташування ядра через n пікселів, роблячи своєрідний крок. У цій ситуації ми отримаємо матрицю, яка буде мати розмірність приблизно в n разів меншу. Це пов'язано в першу чергу з тим, що n не завжди кратний кількості елементів МО. [15]

2.2.3. Агрегувальний шар

Характерною рисою цього шару є те, що його можна застосувати в різних масштабах, і до вхідних даних напряму і всередині рецептивних полів нейронів деякого шару.

Ідея агрегування або об'єднання полягає в створенні зв'язку між певною кількістю нейронів одного шару і одним нейроном другого шару. У випадку усереднювального агрегування, що активно використовується у цій роботі, у агрегувальному шарі працюють з усереднювальними значеннями нейронів.

Даний шар дозволяє знизити розмірність даних й запам'ятовує найкращі конфігурації нейронів. Що дозволяє застосувати його по черзі з згортковим для підвищення ефективності мережі.

2.2.4. Повнозв'язний шар

Повнозв'язні шари використовуються для створення зв'язку між усіма нейронами одного шару з нейронами другого. Вони поширені серед різних сімейств нейронних мереж, зустрічають навіть в простих перцептронах.

Рекомендується застосовувати їх обережно, адже їх примітивна структура доволі сильно впливає на мережу. Через повнозв'язні шари ЗНМ може «забувати» деякі ознаки, що були виявлені раніше. [16]

2.2.5. Вихідний шар

Фінальний шар отримує представлення даних, що було отримано в результаті руху вхідних даних через вищезгадані шари. Далі до його зваженої суми використовують функцію активацію. Власне вид цієї функції обирається автором нейронної мережі, як правило, заздалегідь.

2.3. Гіперпараметри згорткової нейронної мережі і методи підбору їх значень

Результат навчання ЗНМ залежить і від структури моделі, і від методу підбору параметрів. Метод навчання також має власні гіперпараметри (далі – ГП). Гіперпараметри– це величини, чії значення встановлюються перед покращенням моделі.

2.3.1. Гіперпараметри стохастичного градієнта

До тренувального набору даних, котрий час від часу перемішують застосовується стохастичний градієнтний спуск і в результаті отримаємо

оновлення параметрів ЗНМ. Маємо деяку ітерацію $t < T$, в такому випадку оновлення параметрів θ буде виглядати як:

$$\theta^{(t)} \leftarrow \theta^{(t-1)} - \epsilon_t \frac{1}{B} \sum_{t'=Bt+1}^{B(t+1)} \frac{\partial L(z_{t'}, \theta)}{\partial \theta}, \quad (2.2)$$

де $z_{t'}$ - приклад тренувального набору;

L – функція втрат (ГП);

ϵ_t – темп навчання на ітерації t (ГП);

B – розмір пакетної вибірки (ГП);

T – число тренувальних ітерацій (ГП).

Функція втрат (далі - ФВ) - призначена для протиставлення даних, що були надані на вході та даних, що були розраховані в процесі навчання. Як правило, визнається як квадрат відстані між двома точками:

$$L = \frac{1}{2} \sum_i (y_i - z_i)^2 \quad (2.3)$$

де y_i – i -те значення вихідного шару мережі;

z_i - i -те значення відповіді із тренувального набору даних.

Коефіцієнт в формулі (2.3) застосовують для спрощення градієнта. Проте, якщо представити вихідні дані ЗНМ у вигляді розподілу ймовірностей. Тоді

краще за все є застосування окремого випадку ФВ, так звану перехресну ентропію, наступним чином:

$$L = \sum_i y_i \log(z_i) \quad (2.4)$$

Темп навчання – регулює швидкість «руху». градієнту оновлення до екстремуму. При повільній швидкості модель сходиться задовго, а при великій – надто швидко для коректного навчання. Спочатку, перед розрахунками треба встановити початкову швидкість ϵ_0 й спосіб обчислення ϵ_t . Існують різні підходи до розрахунку цих величин, нехай:

$$\epsilon_t = \epsilon_0 \quad (2.5)$$

де $t < \tau$;

τ – новий гіперпараметр.

Розмір пакету це ГП, який можна оптимізувати незалежно від інших. Зазвичай значення цієї величини визначається евристично, виходячи з потреб. Якщо необхідно збільшити ефективність оновлення ваг, то значення збільшують, якщо є потреба в більшій кількості оновлень - зменшують.

Кількість ітерацій визначити доволі просто, для цього існує простий та ефективний спосіб. Все що потрібно, це зупинити навчання в той момент, коли похибка розрахунків починає збільшуватися, а продуктивність відповідно зменшуватися.[16]

2.3.2. Гіперпараметри моделі

Структура НМ має чималу кількість ГП. Слід зазначити, що продуктивність нейромереж дуже сильно залежить від ініціалізація ваг й вхідних даних, які необхідно попередньо обробити.

Кількість прихованих шарів також впливає на ефективність НМ. Розмірність цих шарів завжди прямопропорційна результату навчання і при застосуванні регуляризації це слід теж мати на увазі. Якщо структура НМ не надто складна, то для збільшення ефективності можна встановити однакову розмірність для кожного з прихованих шарів. Також для покращення роботи, треба встановити вхідний шар меншим за перший прихований.

Вищезгадана регуляризація ваг застосовується до функції втрати це дозволяє обмежити перенавчання. Крім цього, якщо є некоректно поставлена задача, то регуляризація дозволяє її розв'язати.

Ваги мають бути точно налаштовані, оскільки від них дуже сильно залежить локальний мінімум. Проте часто використовується підхід налаштування ваг, в якому значення випадково обираються з заданого проміжку.

Як згадувалося вище, попередня обробка даних є важливим моментом, що дозволяє оптимізувати роботу НМ. Це пов'язано з якістю даних, яка є критерієм, що характеризує точність, повноту і своєчасність, а також чи можна інтерпретувати дані.

Найчастіше для попередньої обробки використовують такі засоби як:

- Нормування даних – це перехід до стандартизованої одиниці виміру;

- Аналіз головних компонентів – метод, що дозволяє зменшити розмірність даних, обмежити кількість змінних;
- Уніфікація даних – дозволяєм уніфікувати купи наборів даних в один.

2.3.3. Методи підбору гіперпараметрів

Величезна кількість ГП демонструє наскільки багато може бути конфігурацій конкретної НМ. І вони дуже сильно впливають на процес її навчання. Тому підбір значень цих параметрів також є своєрідною задачею оптимізації. Для підбору значень ГП існують різні підходи, такі як:

- Випадковий підбір;
- Підбір перебором;
- Координатний спуск;
- Пошук сіткою.

Випадковий підбір – найпростіший і найочевидніший метод підбору значень. Його ідея полягає в тому, що для кожного ГП обирається окремий діапазон значень і з нього випадковим чином обирається значення.

Другий доволі простий і очевидний метод – підбір методом перебору, який оснований на підборі значень ГП з певних обраних для кожної величини інтервалів. Далі необхідно проводити відповідні випробування, тестуючи різні значення.

При такому варіанті, якщо функція опукла, то оптимальне значення буде знаходитися посередині. Але якщо ні, то значення може знаходитися будь-де на

проміжку. Можливі випадки, коли крайнє значення обраного діапазону буде оптимальним. В такому випадку, варто переглянути інтервал у відповідному напрямку для отримання більш оптимальнішого значення ГП.

Існує ще один підхід – координатний спуск. В основі цього методу лежить потреба в коригуванні значень конкретного ГП, а усі інші залишаються сталими. Даний варіант ґрунтується на тому, що навчання мало залежить від більшості параметрів, через це необхідно акцентувати увагу на коригуванні обраного ГП.

Розглянемо метод підбору – пошук сіткою, він також працює з інтервалами значень. Даний підхід базується на поелементному добутку проміжків значень, проте чим більше ГП, тим довше відбуваються розрахунки. Ідея підходу полягає в тому, що ми маємо пару параметрів α і β , та множини їх значень $(\alpha_1, \alpha_2, \dots, \alpha_n)$ й $(\beta_1, \beta_2, \dots, \beta_n)$ відповідно. Необхідно для усіх можливих пар значень α і β провести оцінку функції похибки. І в результаті узяти пару з мінімальним її значенням [17].

Також іноді можна застосувати комбінацію методів підбору. Наприклад, застосувати при створенні моделі метод випадкового підбору значень, а потім при оптимізації моделі застосувати метод координатного спуску. [18]

Висновки до розділу 2

Таким чином, на початку цього розділу було розглянуто що таке часові ряди, їх основні характеристики та пов'язані з ними поняття, такі як автоковаріаційна і частково автоковаріаційна функції

Також було розглянуто структуру згорткової мережі під різними кутами: у загальному випадку і окремому випадку, що розглядається у рамках цієї роботи. Розглянуто основні шари згорткової нейронної мережі та їх функціонал.

Далі були розглянуті гіперпараметри ЗНМ (і ГП стохастичного градієнта, і ГП моделі). Наприкінці розділу були описані чотири найпоширеніших методи підбору ГП, їх характерні риси і відзначено, що іноді є доречним застосування комбінацій методів підбору.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Покрокова реалізація

В даній роботі було використано набір даних «BrentOilPrices.csv». Він містить історію зміни значення ціни на нафту марки «Brent» починаючи з 20 травня 1983-го року і закінчуючи 21 квітня 2020-го року. Проте у роботі опрацьовується лише частина ЧР, у період з 2015 до 2020 року, що є приблизно 1/6 усього ЧР [19].



Рис. 3.1 – Часовий ряд у період 2015-2020 років «BrentOilPrices.csv».

На початку, одразу після побудови ЧР необхідно розрахувати АКФ. В результаті, за замовчуванням ми отримаємо значення перші 40 значень, цього більш ніж достатньо для подальших висновків.

```
array([1.          , 0.99930724, 0.99861503, 0.99794056, 0.99725818,
       0.99657652, 0.99588593, 0.9951902 , 0.99450991, 0.99380335,
       0.99306774, 0.99232915, 0.99157648, 0.99079682, 0.9900328 ,
       0.98919855, 0.98836005, 0.98754506, 0.98675883, 0.98597091,
       0.98515491, 0.98433738, 0.98351427, 0.98270723, 0.98187503,
       0.98099294, 0.98012473, 0.97924091, 0.97837983, 0.97751895,
       0.97660552, 0.97572613, 0.9748656 , 0.97399645, 0.97311965,
       0.97219652, 0.97128283, 0.97037091, 0.96947591, 0.96856029,
       0.96760543])
```

Рис. 3.2 – Перші значення автокореляційної функції.

Далі для більш наочного розуміння зображемо АКФ ЧР в графічному вигляді:

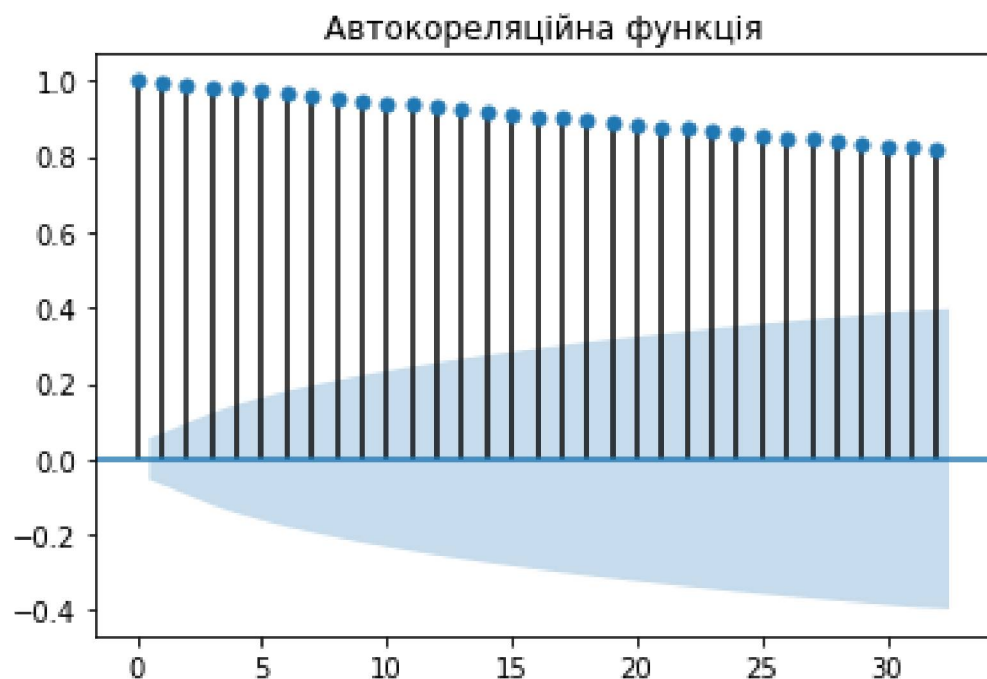


Рис. 3.3 – Графік АКФ.

Тепер можна провести деякі спостереження: переважна більшість значень АКФ очевидно не дорівнює нулю і не наближається до нього. А це свідчить про те, що в даному ЧР окремі точки корелюють між собою, що свідчить, що цей ряд не є набором випадкових чисел.

Наступним кроком буде обчислення перших 40 значень ЧАКФ:

```
array([ 1.00000000e+00,  9.99426785e-01,  7.50935093e-05,  1.50852373e-02,
        -7.29675847e-03,  4.65425826e-04, -8.41657943e-03, -4.83736913e-03,
         1.28295103e-02, -2.33815559e-02, -2.53589797e-02, -3.89911646e-03,
        -1.31403819e-02, -2.40551562e-02,  1.30549997e-02, -6.17743921e-02,
        -3.91646713e-03,  1.80685422e-02,  2.69950651e-02, -9.26281671e-04,
        -2.36840787e-02,  4.32985453e-04, -7.25503906e-03,  1.58815676e-02,
        -1.88125589e-02, -4.31083504e-02,  9.66065056e-03, -1.39679619e-02,
         1.93805445e-02,  2.30455647e-03, -4.51268906e-02,  2.75885594e-02,
         1.19137965e-02, -3.60289532e-03, -2.25529185e-03, -4.05457322e-02,
         9.86703912e-03, -4.23016766e-03,  1.89089981e-02, -1.12122339e-02,
        -3.96054553e-02])
```

Рис. 3.4 – Перші значення часткової автокореляційної функції.

Аналогічно будемо графічне представлення ЧАКФ, але для перших 20 значень для більшої деталізованості графіку:

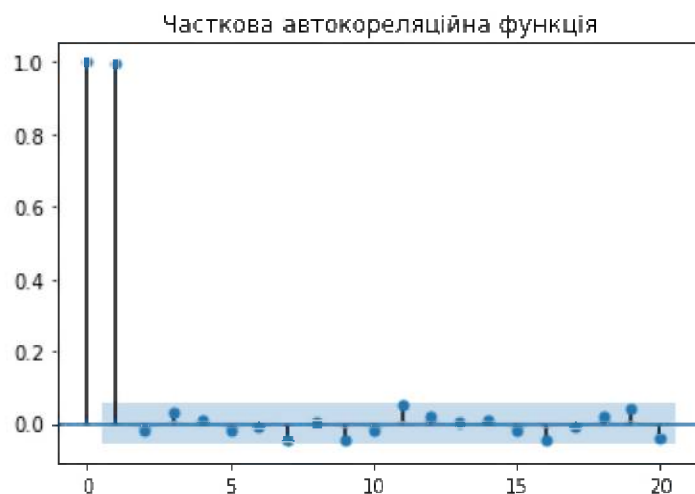


Рис. 3.5 – Графік перших значень ЧАКФ.

Дослідимо і цей графік: тут можна побачити, що в даному ЧР відсутня будь-яка періодичність. Тому можна зробити висновок, що ряд не містить сезонну компоненту, що свідчить про те, що він не є сезонним.

Тепер необхідно провести тестування Дікі-Фулера для подальшого аналізу ЧР:

```
ADF Statistic: -1.7531565240857159  
n_lags: 0.4039534306942054  
p-value: 0.4039534306942054
```

Рис. 3.6 – Тестування Дікі-Фулера.

Отримані результати тестування говорять про те, що ряд є нестационарним, оскільки рівень значущості значно перевищує допустимий поріг у 0.05. Тому з ЧР можна працювати за допомогою рекурентних нейронних мереж.

Наступним кроком треба розділити часовий ряд на дві частини: тренувальну і тестувальну. З допомогою першої модель навчається, а друга допомагає порівняти отриманий прогноз з відповідними до часу значеннями ЧР. Як правило, співвідношення такого розподілу визначається евристично. На практиці найчастіше це співвідношення 4:1 чи 3:1. В моєму випадку розподіл виглядає наступним чином:



Рис. 3.7 – Графічне зображення розподіл наборів на тренувальний та тестувальний.

Набір даних з яким працюємо у даній роботі є величезним і містить більш ніж 8300 значень ціни нафти марки «Brent». Тому слід пам'ятати, що нейромережеві методи можуть перенавчатися і тоді їх ефективність починає пропорційно зменшуватися.

Під час виконання цієї роботи використовується лише 1/6 частина оригінального ряду. Це було зроблено для того, щоб більш наочно продемонструвати роботу програми, оскільки далеко не завжди доступний часовий ряд за такий величезний період часу.

Після розподілу даних створюється і модель. Параметри моделі були підібрані спочатку методом випадкового підбору, а потім відкореговані методом координатного спуску. Створену модель можна охарактеризувати як:

```

Model: "sequential_1"

```

| Layer (type) | Output Shape | Param # |
|--|-----------------|---------|
| lambda_1 (Lambda) | (None, 14, 1) | 0 |
| conv1d_2 (Conv1D) | (None, 14, 64) | 256 |
| conv1d_3 (Conv1D) | (None, 14, 128) | 24704 |
| global_average_pooling1d_1 (GlobalAveragePooling1D) | (None, 128) | 0 |
| flatten_1 (Flatten) | (None, 128) | 0 |
| dropout_2 (Dropout) | (None, 128) | 0 |
| dense_2 (Dense) | (None, 700) | 90300 |
| dropout_3 (Dropout) | (None, 700) | 0 |
| dense_3 (Dense) | (None, 1) | 701 |

```

=====
Total params: 115,961
Trainable params: 115,961
Non-trainable params: 0

```

```

FINAL LOSS: 9.25 | FINAL MAE: 9.73

```

Рис. 3.8 – Характеристика моделі.

На вищевказаному рисунку можна побачити структуру створенної моделі. Також можна побачити кількість параметрів з якими працює модель, і це доволі мале число серед нейромережових методів. В самому кінці можна побачити розрахунок похибок.

Нарешті відбувається власне процес прогнозування в результаті якого ми отримуємо наступний графік:

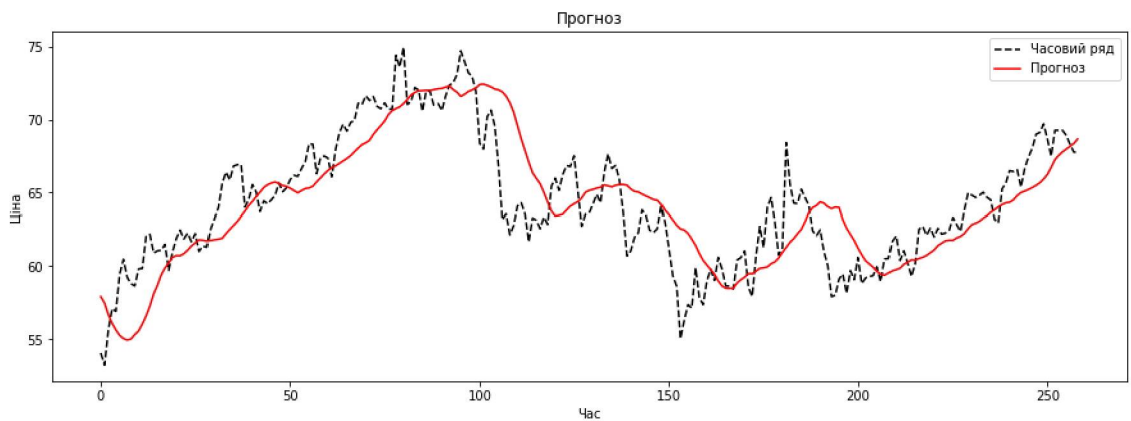


Рис. 3.9 – Прогноз ЧР.

Висновки до розділу 3

Таким чином, у даному розділі я покроково описав процес функціонування моєї програми. Умовно його можна розділити на три основних етапи.

На першому етапі був побудований оригінальний часовий ряд. Далі були розраховані АКФ і ЧАКФ, що дозволило зробити певні висновки, а саме: ряд не є набором випадкових величин і не містить сезонну компоненту. Потім було проведено тестування Дікі-Фуллера, було підтверджено гіпотезу про те, що даний ряд не є стаціонарним.

На другому етапі було здійснено розподіл набору даних на дві частини: тренувальну і тестувальну. З допомогою першої модель навчається, а друга допомагає порівняти отриманий прогноз з відповідними до часу значеннями ЧР.

На третьому етапі створювалася власне комп'ютерна модель і проводилося її навчання. Далі було здійснено прогноз і побудовано графік (рис. 3.9).

ВИСНОВКИ

У першому розділі я розглянув 8 методів, що дозволяють вирішувати поставлену задачу роботи з часовими рядами. Тому, спираючись на перераховані переваги та недоліки переглянутих методів для вирішення задачі аналізу та прогнозування часових рядів, я обрав метод згорткової нейронної мережі

У другому розділі було розглянуто що таке часові ряди, їх основні характеристики. Також було описано структуру згорткової мережі, основні шари згорткової нейронної мережі та їх функціонал. Далі були розглянуті гіперпараметри ЗНМ (і ГП стохастичного градієнта, і ГП моделі). Наприкінці розділу були описані чотири найпоширеніших методи підбору ГП, їх характерні риси і відзначено, що іноді є доречним застосування комбінацій методів підбору.

У третьому розділі я покроково описав процес функціонування моєї програми. Умовно його можна розділити на три основних етапи.

На першому етапі був побудований оригінальний часовий ряд. Далі були розраховані АКФ і ЧАКФ, що дозволило зробити певні висновки, а саме: ряд не є набором випадкових величин і не містить сезонну компоненту. Потім було проведено тестування Дікі-Фуллера, було підтверджено гіпотезу про те, що даний ряд не є стаціонарним.

На другому етапі було здійснено розподіл набору даних на дві частини: тренувальну і тестувальну. З допомогою першої модель навчається, а друга допомагає порівняти отриманий прогноз з відповідними до часу значеннями ЧР.

На третьому етапі створювалася модель. Потім підбиралися параметри моделі і проводилося її навчання. Далі було здійснено прогноз і побудовано відповідний графік.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Скорохода О. В., Андрієцький Б. Р., Цмоць І. Г. Засоби оцінювання параметрів динамічних об'єктів на основі нейромережевого сингулярного спектрального аналізу / Науковий вісник НЛТУ України, 2012. №22.11. С. 362-369.
2. Основи наукових досліджень в економіці: навч. посібник. / Макогон Ю.В., Пилипенко В.В. Донецьк: Альфа-прес, 2007. 144 с.
3. Наукове дослідження: організація, методологія, інформаційне забезпечення: навч. посіб. / Пілюшенко В.Л. та ін. Київ: Лібра, 2004. 344 с.
4. Глотов Є. О., Шульга Н. В., Попова О. М. Побудова моделі довгострокового прогнозування видобутку природного газу в Україні для прийняття управлінських рішень / Бізнес Інформ, 2019. №. 493. С. 133-139.
5. Інтелектуальні системи автоматизації : монографія / О. Г. Аврунін та ін. Кременчук: НОВАБУК, 2021. 322 с.
6. Методологія наукових досліджень : підручник / Стеченко Д. М., Чмир О. С. К.: Знання, 2007. 317 с.
7. Gasparin A., Lukovic S., Alippi C. Deep learning for time series forecasting: The electric load case // CAAI Transactions on Intelligence Technology, 2022. Vol. 7.1. P. 1-25.

8. Нейромережеві засоби штучного інтелекту : навч. посіб. / Ткаченко Р. О, Ткаченко П. Р, Ізонін І. В.. Львів: Вид-во Львівської політехніки, 2017. 206 с.
9. Нейронні мережі: теорія та практика: навч. посіб. / Субботін С. О., Субботін С. А. Житомир: Євенок О.О, 2020. 184 с.
10. Субботін С. О. Нейронні мережі: навч. посіб. Запоріжжя: ЗНТУ, 2014. 132 с.
11. Білоус М. О. Нейронні мережі. Цифровізація суспільства та бізнесу: виклики та можли-вості для молоді: матеріали I Всеукраїнської студентської наукової конференції пам'яті доктора економічних наук, професора Кадієвського Володимира Андрійовича. Київ: «Інформаційно-аналітичне агентство», 2021. С. 15-16.
12. Lin J. et al. A symbolic representation of time series, with implications for streaming algorithms // Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, 2003. – P. 1-10.
13. Walczak B. Calibration in wavelet domai // Wavelets in chemistry, 2000. P.323-349.
14. Vilela M. et al. Fluctuation analysis of activity biosensor images for the study of information flow in signaling pathways // Academic Press 2013. Vol. 519. P. 253-276.
15. Гудіков В. О. Система нейромережевого розпізнавання деталей одягу на зображенні: автореф. маг. дис. Київ, 2018. С. 43-59.
16. Білашенко С. В. Розпізнавання зображень за допомогою згорткових нейронних мереж з використанням бібліотеки Keras / Гірничий вісник : науково-технічний збірник. Кривий Ріг, 2018. № 103. С. 148–154.

17. Кушнір Н. О. та ін. Використання згорткових нейронних мереж у задачах розпізнавання та класифікації об'єктів зображень // Технічна інженерія 2022. № 1.89. С. 93-100.
18. Волков О. О. Попередня обробка даних (в тому числі заповнення відсутніх даних, видалення аномальних, згладжування даних для вирішення завдання зниження випадкових шумів, стиснення і нормалізація даних): автореф. бак. дис. Київ, 2020.
19. Радюк П. М. Підхід до прискорення навчання згорткової нейронної мережі за рахунок налаштування гіперпараметрів навчання / Комп'ютерні системи та інформаційні технології. 2020. № 2. С. 31-36.
20. Daily historical Brent Oil Prices available on the U.S. URL: <https://www.kaggle.com/datasets/mabusalah/brent-oil-prices>. Дата звернення: 5.3.2020

Завдання на кваліфікаційну роботу

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) Магістр
Галузь знань: Автоматизація та приладобудування
Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології.

ЗАТВЕРДЖУЮ

Завідувач кафедри теоретичної та
прикладної системотехніки
д.т.н., проф. Шмагков С. І.

« 10 » 12 2021 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Бурлай Іскандер Сергійович

(прізвище, ім'я, по-батьківськи студента)

1. Тема роботи «Комп'ютерна модель системи аналізу та прогнозування цін на енергоносії за допомогою згорткових нейронних мереж»

керівник роботи Бакуменко Ніна Станіславівна, к.т.н., доцент

(прізвище, ім'я, по-батьківськи науковий співробітник, вчитель)

затверджені наказом по університету від « 30 » листопада 2022 року № _____

2. Строк подання студентом роботи _____

3. Перелік питань, які потрібно розробити

1. Постановка задачі аналізу та прогнозування цін на енергоносії.
2. Порівняльний аналіз методів аналізу та прогнозування цін на енергоносії.
3. Розробка математичної моделі системи аналізу та прогнозування цін на енергоносії за допомогою згорткових нейронних мереж.
4. Розробка програмно-алгоритмічної реалізації моделі аналізу та прогнозування цін.
5. Тестування та аналіз отриманих результатів.

4. План роботи

| № з/п | Назви етапів роботи | Термін виконання етапів роботи |
|-------|--|--------------------------------|
| 1 | Постановка задачі. | 1.09-2.09.21 |
| 2. | Підбір наукової літератури з теми роботи. | 3.09 - 21.10.21 |
| 3. | Огляд методів аналізу та прогнозування часових рядів. | 22.10 -24.10.21 |
| 4. | Вибір і обґрунтування відповідного методу прогнозування цін на енергоносії . | 25.10 - 1.11.21 |
| 5. | Розробка математичної моделі системи аналізу та прогнозування цін на енергоносії на підставі згорткових нейронних мереж. | 2.11 - 7.12.21 |
| 6 | Розробка комп'ютерної реалізації моделі. | 8.12.21 - 7.01.22 |
| 7. | Відладка та тестування розробленої моделі. | 8.01 - 5.02.22 |
| 8. | Коректування моделі після тестування. | 6.02 - 10.03.22 |
| 9 | Аналіз отриманих результатів. | 11.03-15.03.22 |
| 10 | Оформлення пояснювальної записки до кваліфікаційної роботи | 16.03-22.05.22 |
| 11 | Представлення кваліфікаційного проекту керівнику кваліфікаційної роботи та рецензенту. | 23.05.22-30.05.22 |
| 12 | Науково-дослідницька практика. | 1.09-12.10.22 |
| 13 | Переддипломна практика. | 13.10-23.11.22 |

5. Дата видачі завдання 10.12.2021

Студент

І.С. Бурдай

підпис, прізвище

підпис

Керівник роботи

Н.С. Бакуменко

підпис, прізвище

Додаток Б

**Технічне завдання на розробку програмного виробу
«Комп'ютерна модель системи аналізу та прогнозування цін на енергоносії за допомогою згорткових нейронних мереж»**

| Назва розділу | Назва і зміст підрозділу |
|--|--|
| 1. Введення | 1.1. Назва: Комп'ютерна модель системи аналізу та прогнозування цін на енергоносії за допомогою згорткових нейронних мереж. 1.2. Область застосування: економіка. |
| 2. Підстава для розробки | 2.1. Навчальний план за спеціальністю 151 – «Автоматизація та комп'ютерно-інтегровані технології». 2.2. Завдання на кваліфікаційну роботу магістра затверджено наказом ХНУ ім. В.Н. Каразіна № від 30.11.2021. |
| 3. Призначення розробки | 3.1. Мета: покращення аналізу та прогнозування цін на енергоносії. 3.2. Призначення програмного виробу: для використання в економічних прогнозах цін. 3.3. Початкові дані для розробки: часовий ряд динаміки цін на енергоносії (вибірка була узята з сайту https://www.kaggle.com). |
| 4. Технічні вимоги до програмного виробу | 4.1. Вимоги до функціональних характеристик: Програма повинна: 1) Представляти з себе комп'ютерну реалізацію моделі згорткової нейронної мережі; 2) Забезпечити можливість завантаження користувачем нових даних, для процесу аналізу; |

| | |
|--|--|
| | <p>3) Забезпечувати можливість виведення результату на екран за допомогою інтуїтивно зрозумілого інтерфейсу;</p> <p>4.2. Вимоги до надійності: програма повинна видавати повідомлення про помилки.</p> <p>4.3. Вимоги до умов експлуатації:</p> <p>1) Умови експлуатації повинні співпадати з умовами експлуатації персональної електронно-обчислювальної машини, на якій буде працювати, а також сумісних з нею персональними комп'ютерами.</p> <p>2) Для користування програмою користувачу необхідно пройти короткий курс навчання роботи з програмою.</p> <p>4.4. Вимоги до складу і параметрів технічних засобів: Персональний комп'ютер у повній комплектації або ноутбук.</p> <p>4.5. Вимоги до інформаційної та програмної сумісності: «Windows» будь-якої версії та веб-браузер сумісний з «Windows»;</p> <p>4.6 Вимоги до маркування та упаковки не висуваються;</p> <p>4.7 Вимоги до транспортування і зберігання не висуваються;</p> <p>4.8 Спеціальні вимоги не пред'являються.</p> |
| <p>5. Вимоги до програмної документації.</p> | <p>Програмною документацією до виробу « Комп'ютерна модель системи аналізу та прогнозування цін на енергоносії» за допомогою згорткових нейронних мереж вважати:</p> <p>1) Справжнє Технічне завдання на розробку програмного виробу (представити у вигляді Додатку Б до пояснювальної</p> |

| | | | |
|---------------------------------|---|--|--------------------------------|
| | <p>записки до кваліфікаційної роботи).</p> <p>2) Програму і методику випробувань розробленого програмного виробу (представити у вигляді додатку В до пояснювальної записки до кваліфікаційної роботи).</p> <p>3) Опис програмного виробу (представити в розділі 3 кваліфікаційної роботи).</p> <p>4) Текст програми (представити в Додатку Г до пояснювальної записки до кваліфікаційної роботи).</p> | | |
| 6. Техніко-економічні показники | <p>1) Орієнтовна оцінка ефективності виконуваної роботи: представити в розділі 3 пояснювальної записки.</p> <p>2) Терміни і можливі витрати грошових коштів на розробку програмного виробу – не потрібні.</p> | | |
| 7. Стадії і етапи розробки | № з/п | Назви етапів роботи | Термін виконання етапів роботи |
| | 1 | Постановка задачі. | 1.09.21 |
| | 2. | Підбір наукової літератури з теми роботи. | 2.09 - 21.10.21 |
| | 3. | Огляд методів аналізу та прогнозування часових рядів. | 22.10 - 24.10.21 |
| | 4. | Вибір і обґрунтування відповідного методу прогнозування цін на енергоносії | 25.10 - 1.11.21 |
| | 5. | Розробка математичної моделі системи аналізу та прогнозування цін на енергоносії на підставі згорткових нейронних мереж. | 2.11 - 7.12.21 |
| | 6 | Розробка комп'ютерної реалізації моделі. | 8.12.21 – 7.01.22 |
| | 7. | Відладка та тестування розробленої моделі. | 8.01 – 5.02.22 |
| | 8. | Корегування моделі після тестування. | 6.02 – |

| | | |
|---------------------------------|---|-----------------------|
| | | 10.03.22 |
| 9 | Аналіз отриманих результатів. | 11.03- 15.03.22 |
| 10 | Оформлення пояснювальної записки до кваліфікаційної роботи | 16.03- 22.05.22 |
| 11 | Представлення кваліфікаційного проекту керівнику кваліфікаційної роботи та рецензенту. | 23.05.22- 30.05.22 |
| 12 | Науково-дослідницька практика | 1.09- 12.10.22 |
| 13 | Переддипломна практика | 13.10- 23.11.22 |
| 8. Порядок контролю і приймання | <p>1) Перевірку ходу розробки програмного виробу Керівнику робіт виконувати 1 раз в 2 тижні.</p> <p>2) Випробування програмного виробу відповідно до Програми і методики випробувань провести на базі комп'ютерного класу.</p> <p>3) Захист розробленого програмного виробу провести на засіданні атестаційної комісії.</p> <p>4) Пояснювальну записку представити на паперових носіях в одному примірнику, в електронному вигляді - на CD-диску в одному екземплярі.</p> | |

Виконавець
студент групи КУ-61
Бурлай І.С.

Замовник
к.т.н., доцент
Бакуменко Н.С.

Додаток В

**Програма та методика випробувань програмного виробу
«Комп'ютерна модель системи аналізу та прогнозування цін на енергоносії за допомогою згорткових нейронних мереж»**

1 Об'єкт випробувань

1.1 Найменування випробуваного програмного виробу: «Комп'ютерна модель системи аналізу та прогнозування цін на енергоносії за допомогою згорткових нейронних мереж»

1.2 Область його застосування: економіка.

2. Мета випробувань

- 1) Покращення аналізу та прогнозування цін на енергоносії.
- 2) Підтвердження коректності функціонування програмного виробу.

3. Загальні положення**3.1 Підстави для проведення випробувань**

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

3.2 Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться дистанційно в період роботи атестаційної комісії».

3.3 Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї Програми і методики випробувань.

3.4 Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання за участю Замовника, Виконавця та інших осіб, присутніх на засіданні в дистанційному режимі.

4. Вимоги до програми або програмного виробу

4.1. Вимоги до функціональних характеристик:

Програма повинна:

- 1) Представляти з себе комп'ютерну реалізацію згорткової нейронної мережі;
- 2) Забезпечити можливість загрузки користувачем нових даних, для процесу аналізу;
- 3) Забезпечувати можливість виведення результату на екран за допомогою інтуїтивно зрозумілого інтерфейсу;

4.2. Вимоги до надійності: програма повинна видавати повідомлення про помилки.

4.3. Вимоги до умов експлуатації:

- 1) Умови експлуатації повинні співпадати з умовами експлуатації персональної електронно-обчислювальної машини, на якій буде працювати, а також сумісних з нею персональними комп'ютерами.
- 2) Для користування програмою користувачу необхідно пройти короткий курс навчання роботі з програмою.

4.4. Вимоги до складу і параметрів технічних засобів:

Персональний комп'ютер у повній комплектації або ноутбук.

4.5. Вимоги до інформаційної та програмної сумісності: «Windows» будь-якої версії та веб-браузер сумісний з «Windows»;

- 4.6 Вимоги до маркування та упаковки не висуваються;
- 4.7 Вимоги до транспортування і зберігання не висуваються;
- 4.8 Спеціальні вимоги не пред'являються.

5. Вимоги до програмної документації

Програмною документацією до виробу вважати:

- 1) Справжнє Технічне завдання на розробку програмного виробу (представлено у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).
- 2) Програму і методику випробувань розробленого програмного виробу (представлена в Додатку В до пояснювальної записки до кваліфікаційної роботи).
- 3) Опис програмного виробу (представлено у розділі 3 пояснювальної записки до кваліфікаційної роботи).
- 4) Текст програми (представлений в Додатку Г кваліфікаційної роботи).

6. Засоби і порядок випробувань

6.1 Засоби випробувань

Випробування проводяться на технічних засобах, таких як персональний комп'ютер у повній комплектації або ноутбук.

Випробування проводяться з використанням програмних засобів: «Windows» будь-якої версії та веб-браузер сумісний з «Windows».

6.2 Порядок проведення випробувань

Перевірка програмної документації

1.1) Перевірка комплектності програмної документації. Перевірку здійснювати за критерієм наявності, представленої в ТЗ документації.

Критерієм успішності тесту вважати відповідність наявної документації згідно зі списком в ТЗ.

1.2) Перевірка якості програмної документації. Перевірку здійснювати за критерієм відповідності вимогам єдиної системи проектної документації (ЕСПД).

Критерієм успішності тесту вважати відповідність якості наявної документації згідно з вимогами ЕСПД.

2) Названа модель працює відповідно до умов експлуатації ОС «MS Windows» будь-якої версії.

3) Для роботи необхідний встановлений веб-браузер сумісний з «MS Windows».

Тест 1. Перевірка працездатності моделі

Критерієм успішності вважати наявність необхідної ОС й сумісного з ним веб-браузеру.

4) Порядок проведення випробувань:

- Запуск програми здійснюється завантаженням в середовищі «Google Colaboratory» програмного коду, з підключенням початкових даних.

- Завантажуються додаткові пакети, необхідні для роботи.

- Порядково запускається програмний код, для відстеження роботи програми на кожному кроці.

Тест 1 пройдено успішно.

Тест 2.

5) Для проведення випробувань пропонується провести ті заходи, опис яких містяться в розділі 3. Критерієм успішності є коректні отримані результати на виході.

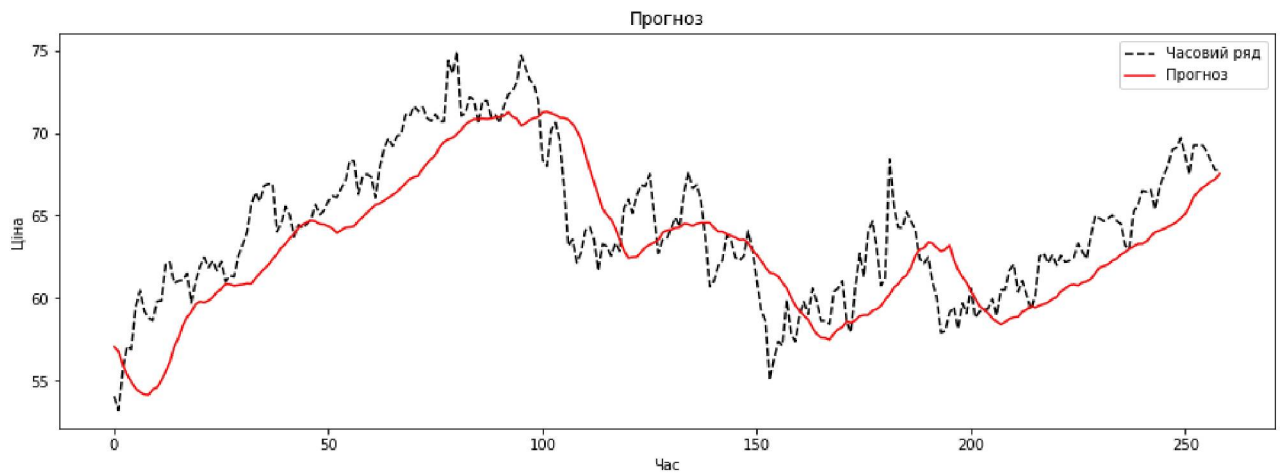


Рисунок В.1 – Результат прогнозування

Тест 2 пройдено успішно.

Висновки: Тести 1,2 виконанні успішно, прийнято рішення про те, що випробування пройдено успішно.

Виконавець

Бурлай І.С.

Текст програми

«Модель системи аналізу та прогнозування цін на енергоносії»

Лістинг 1

```
#Імпортуємо потрібні бібліотеки
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
from statsmodels.graphics import tsaplots
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller

from tensorflow.keras.callbacks import Callback, LearningRateScheduler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Dropout, Lambda, Bidirectional
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.losses import Huber
from tensorflow.keras.layers import Conv1D, GlobalAveragePooling1D, Flatten

import warnings
warnings.filterwarnings('ignore')

#Завантажуємо і оброблюємо датасет
ts = pd.read_csv("BrentOilPrices.csv",
                parse_dates=['Date'],
                index_col='Date')

start = pd.to_datetime('2015-1-1')
end = pd.to_datetime('2020-1-1')
ts = ts[start:end]

#Будуємо часовий ряд
fig = plt.figure(figsize=(15,5))
plt.plot(ts, color='blue')
```

```
plt.xlabel('Час')
plt.ylabel('Ціна')
plt.title('Ціна на нафту марки "Brent"')
plt.show()
```

```
#Розрахунок автокореляційної функції і її графічне зображення
sm.tsa.acf(ts)
#fig1 = tsaplots.plot_acf(ts, lags=8282)
fig1 = tsaplots.plot_acf(ts)
plt.title('Автокореляційна функція')
plt.show()
```

```
#Розрахунок часткової автокореляційної функції і її графічне зображення
sm.tsa.pacf(ts)
fig2 = tsaplots.plot_pacf(ts, lags = 20)
plt.title('Часткова автокореляційна функція')
plt.show()
```

```
# ADF-тестування
result = adfuller(ts, autolag='AIC')
print(f'ADF Statistic: {result[0]}')
print(f'n_lags: {result[1]}')
print(f'p-value: {result[1]}')
```

```
#Налаштовуємо тренувальну і тестові набори
end_train_ix = pd.to_datetime('2018-12-31')
train = ts[:end_train_ix]
test = ts[end_train_ix:]
```

```
#Візуалізуємо набори графічно
fig = plt.figure(figsize=(15,5))
plt.plot(train, color='blue', label='Тренувальний набір')
plt.plot(test, color='yellow', label='Тестувальний набір')
plt.xlabel('Час')
plt.ylabel('Ціна')
plt.title('Розподіл на тренувальний і тестувальний набори')
plt.legend()
plt.show()
```

```
#Проводимо розподіл на "вікна"
WINDOW = 14 #днів
```

```

train_data = tf.data.Dataset.from_tensor_slices(train.values)
train_data = train_data.window(WINDOW+1, shift=1, drop_remainder=True)
train_data = train_data.flat_map(lambda x: x.batch(WINDOW+1))
train_data = train_data.map(lambda x: (x[:-1], x[-1]))
train_data = train_data.shuffle(1_000)
train_data = train_data.batch(32).prefetch(1)

#Налаштовуємо зворотній дзвінок (необхідно для більш точного навчання)
class CustomCallback(Callback):#!!!!
    def on_epoch_end(self, epoch, logs={}):
        if logs.get('mae') < 10.0:
            self.model.stop_training = True

my_callback = CustomCallback()

#Оновлюємо швидкість навчання на основі номера епохи:
def scheduler(epoch, lr):
    if epoch < 2:
        return 0.01
    else:
        return lr * 0.99

lr_scheduler = LearningRateScheduler(scheduler)

#Створюємо модель
cnn_model = Sequential([
    Lambda(lambda x: tf.expand_dims(x, axis=-1), input_shape=[WINDOW]),
    Conv1D(filters=64, kernel_size=3, strides=1,
           padding='causal', activation='relu'),
    Conv1D(filters=128, kernel_size=3, strides=1,
           padding='causal', activation='relu'),
    GlobalAveragePooling1D(),
    Flatten(),
    Dropout(0.3),
    Dense(700, activation='relu'),
    Dropout(0.5),
    Dense(1)
])

cnn_model.compile(
    loss=Huber(),
    optimizer=Adam(),

```

```

    metrics=['mae']
)

#Характеристика створеної моделі
cnn_model.summary()

#Тренування моделі
cnn_history = cnn_model.fit(
    train_data,
    epochs=100,
    callbacks=[lr_scheduler, my_callback],
    verbose=0
)

print("FINAL LOSS: {} | FINAL MAE: {}".format(
    round(cnn_history.history['loss'][-1], 2),
    round(cnn_history.history['mae'][-1], 2)))

#Прогнозування
all_models = [('CNN MODEL', cnn_model)]

model_forecasts = {
    'CNN MODEL': []
}

#Регулюємо швидкість навчання на основі номера епохи
forecast_data = train[-WINDOW:].append(test[: -1]).values

for name, model in all_models:
    test_data = tf.data.Dataset.from_tensor_slices(forecast_data)
    test_data = test_data.window(WINDOW, shift=1, drop_remainder=True)
    test_data = test_data.flat_map(lambda x: x.batch(WINDOW+1))
    test_data = test_data.batch(32).prefetch(1)
    #Прогноз
    preds = model.predict(test_data)
    model_forecasts[name].append(preds)

#Будуємо графік прогнозу
fig = plt.figure(figsize=(15,5))
plt.plot(test.values, color='black', linestyle='--', label='Часовий ряд')
plt.plot(model_forecasts['CNN MODEL'][0].squeeze(), color='red',
label='Прогноз')
```

```
plt.xlabel('Час')  
plt.ylabel('Ціна')  
plt.title('Прогноз')  
plt.legend()  
plt.show()
```