НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА АНАЛІЗУ ДАНИХ

«До захисту допущено»
В.о. завідувача кафедри
_____ Ганна ЯЙЛИМОВА
«__» _____ 2024 р.

Дипломна робота

на здобуття ступеня бакалавра

зі спеціальності: 113 Прикладна математика
на тему: **«Нейронні диференціальні рівняння для прогнозування цін фінансових інструментів на прикладі американських опціонів»**

Виконав: студент 4 курсу, групи ФІ-02
Вітенко Ігор Олегович

Керівник: асистент кафедри ММАД Яворський О. А. _____

Рецензент: доцент кафедри ММЗІ, к.т.н. Яковлєв С. В. _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

Київ — 2024

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА АНАЛІЗУ ДАНИХ

Рівень вищої освіти — перший (бакалаврський)
Спеціальність (освітня програма) — 113 Прикладна математика,
ОПП «Математичні методи моделювання, розпізнавання образів та комп'ютерного зору»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Ганна ЯЙЛИМОВА
«__» _____ 2024 р.

ЗАВДАННЯ
на дипломну роботу

Студент: Вітенко Ігор Олегович

1. Тема роботи: «*Нейронні диференціальні рівняння для прогнозування цін фінансових інструментів на прикладі американських опціонів*»,

керівник: асистент кафедри ММАД Яворський О. А.,

затверджені наказом по університету №__ від «__» _____ 2024 р.

2. Термін подання студентом роботи: «__» _____ 2024 р.

3. Вихідні дані до роботи:

4. Зміст роботи: *Аналіз та застосування методу нейронних диференціальних рівнянь (англ. NDEs, Neural Differential Equations) для прогнозування цін на американські опціони. Огляд інших існуючих методів прогнозування та моделей глибинного навчання (англ. DL, Deep Learning) для данної задачі.*

5. Перелік ілюстративного матеріалу: *«Презентація доповіді»*

6. Дата видачі завдання: 10 грудня 2023 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	листопад-грудень 2023 р.	Виконано
2	Огляд та опрацювання опублікованих джерел за тематикою дослідження	грудень 2023 р - лютий 2024 р.	Виконано
3	Написання програмного забезпечення та проведення дослідження	березень-квітень 2024 р.	Виконано
4	Оформлення та опис результатів	травень 2024 р.	Виконано
5	Написання та оформлення дипломної роботи	травень-червень 2024 р.	Виконано
6	Отримання рекомендації до захисту	08.06.2024	Виконано

Студент _____ Вітенко І. О.

Керівник _____ Яворський О. А.

РЕФЕРАТ

Кваліфікаційна робота містить: 56 сторінок, 23 рисунки, 11 таблиць, 54 джерела.

У даній роботі розглядається метод прогнозування цін на американські опціони типу put та call за допомогою нейронних диференціальних рівнянь на основі данних з фондового ринку та синтетичних. Було розглянуто дві архітектури нейронної мережі, запропоновано підхід до генерації синтетичних данних, порівняння методів попередньої обробки та оптимізації для нашої задачі.

В ході дослідження, було показано що метод нейронних диференціальних рівнянь є гарним доповненням до класичних чисельних методів розв’язання диференціальних рівнянь в частинних похідних, зберігаючи за собою їх сильні сторони, та адекватним аналогом до методів, які використовують лише машинне навчання.

МАШИННЕ НАВЧАННЯ, ШТУЧНА НЕЙРОННА МЕРЕЖА, НЕЙРОННІ ДИФЕРЕНЦІАЛЬНІ РІВНЯННЯ, РІВНЯННЯ БЛЕКА-ШОУЛЗА, АМЕРИКАНСЬКІ ОПЦІОНИ, МЕТОДИ ОПТИМІЗАЦІЇ

ABSTRACT

This thesis explores a method for predicting prices of American-style put and call options using neural differential equations based on stock market data and synthetic. Two neural network architectures are examined, with a proposed approach for generating synthetic data, and a comparison of preprocessing and optimization methods for our problem.

The study demonstrates that the method of neural differential equations serves as a valuable complement to classical numerical methods for solving partial differential equations, retaining their strengths while also providing an adequate alternative to methods that rely solely on machine learning.

MACHINE LEARNING, ARTIFICIAL NEURAL NETWORK, NEURAL DIFFERENTIAL EQUATIONS, BLACK-SCHOLES EQUATION, AMERICAN OPTIONS, OPTIMIZATION METHODS, SYNTHETIC DATA

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ML — машинне навчання (англ. machine learning)

NDE — Нейронне диференціальне рівняння (англ. neural differential equation)

PINN — Фізично інформована нейронна мережа (англ. physics informed neural network)

PDE — Диференціальне рівняння з частинними похідними (англ. partial differential equation)

MSE — середньоквадратичне відхилення (англ. mean squared error)

ReLU — випрямлений лінійний вузол (англ. rectified linear unit)

LeakyReLU — витікаючий випрямлений лінійний вузол (англ. leaky rectified linear unit)

Tanh — гіперболічний тангенс (англ. hyperbolic tangent)

SGD — стохастичний градієнтний спуск (англ. stochastic gradient descent)

RMSProp — середньоквадратичне поширення (англ. root mean squared propagation)

Adam — адаптивна оцінка моменту (англ. adaptive moment estimation)

Adamax — доповнення до Adam, використовує норму максимуму замість другого моменту градієнту.

ВСТУП

Актуальність дослідження. Проблема визначення оптимальної ціни для опціона на фондовому ринку є добре досліджуваною. Її важливість випливає з завдань управління ризиками, у яких опціон функціонує як інструмент геджування ризиків для фінансових установ. Регуляторна відповідність також відіграє суттєву роль, оскільки неточне ціноутворення може призвести до штрафів для фінансових установ. Інфляція, зростання ВВП та інші загальноекономічні показники також вносять додатковий рівень впливу на невизначеність цін опціонів.

Існує низка методів, які дозволяють точно визначити ціну опціону, проте зазвичай в них доводиться вирішувати диференціальне рівняння, яке не має простого аналітичного розв'язку. Для цього часто використовуються традиційні чисельні методи. Хоча вони і надають дуже точні розв'язки і просто інтерпретуються, вони є дорогими з точки зору обчислень. З іншого боку є методи глибоких нейронних мереж, які дешевші в обчисленнях, але повністю не інтерпретуються. Наша робота зосереджена на методі, який забезпечує швидке виконання обчислень та водночас зберігає високу ступінь пояснюваності результатів.

Метою дослідження є побудова методу створення оцінок ціни американських опціонів за допомогою нейронних диференціальних рівнянь.

Об'єктом дослідження є методи машинного навчання для моделювання ціноутворення опціонів.

Предметом дослідження є методи нейронних диференціальних рівнянь для моделювання ціноутворення американських опціонів.

Наукова новизна полягає у використанні нейронних диференціальних рівнянь для задачі прогнозування цін на опціони фондових ринків.

Практичне значення полягає у використанні запропонованого підходу для оптимізації існуючих методів прогнозування цін на опціони.

1 МОДЕЛЬ ЦІНОУТВОРЕННЯ ОПЦІОНІВ

В даному розділі будуть основні теоретичні відомості про об’єкт дослідження та необхідні знання для роботи з ним.

1.1 Опціон: визначення, типи

Опціон — це фінансовий інструмент, який надає право придбати (опціон типу call) або продати (опціон типу put) базовий актив за певних умов протягом заздалегіть визначеного періоду часу. Одними з найбільш популярних видів опціонів є американський та європейський. «Європейський опціон» може бути проданим тільки у попередньо зазначену дату, коли «американський опціон» може бути проданим у будь-який час до завершення його строку дії. Ціна, яку платять за базовий актив називається ціною виконання (англ. strike price) [7].

Через цю специфіку ми маємо аналітичні розв’язки для європейських опціонів типу put та call, але не маємо жодного для американських [53]. Через це у цій роботі ми будемо розглядати лише американські опціони.

Для опціону типу call в цілому очевидно, що чим більша ціна базового активу, тим більша ціна опціону. Коли ціна активу набагато більша за ціну виконання, то ми можемо бути впевненними, що опціон буде продано І аналогічно для опціону типу put, але при умові, що ціна активу буде набагато менша за ціну виконання. З іншої сторони, для опціону call якщо ціна активу набагато менша за ціну виконання, то з великою долею вірогідності опціон не буде продано і тому його ціна близька до нуля і навпаки для опціону put якщо ціна активу набагато більша за ціну виконання, то скоріше за все опціон не буде продано і його ціна близька до нуля [7]. Якщо дата закінчення опціону далеко в майбутньому, то ціна активу, який виплачує ціну виконання буде низькою, а вартість опціону приблизно дорівнюватиме ціні активу.

1.2 Ціноутворення опціонів

Якщо опціони правильно оцінені на ринку, то не має бути можливості гарантовано отримувати прибуток, створюючи портфелі з довгих та коротких позицій в опціонах та їх базових активів [7]. Так як в реальності ринок не є ідеально оціненим, то існують різні види маніпуляцій, такі як арбітраж — одночасна купівля та продаж активу на різних ринках для отримання гарантованого прибутку з різниці цін. З цього і випливає важливість точного ціноутворення опціонів для унеможливлення таких видів торгівлі. Основна мета теорії ціноутворення опціонів полягає в розрахунку ймовірності того, що опціон буде продано, або він буде в грошах (англ. ITM, in the money), при закінченні терміну дії і призначенні йому доларової вартості. Для опціону call бути в грошах означає, що ціна виконання (англ. strike price) буде нижчою за ціну активу і власник опціону матиме можливість купити наперед зазначену кількість активів за ціну нижче ринку. Для опціону put це означає, що ціна виконання буде вищою за ціну активу, і власник опціону матиме можливість продати заздалегідь визначену кількість активів за ціну, що перевищує ринкову [43].

Ціна базового активу (наприклад, ціна акції), ціна виконання, волатильність, процентна ставка та час до закінчення терміну дії, який визначається кількістю днів між датою розрахунку та датою виконання опціону є типовими змінними, які вводяться в математичні моделі для визначення теоретичної справедливої вартості опціону.

Також варто зазначити про грецькі літери в опціонах (англ. greeks) — фінансові метрики, які можна використовувати для вимірювання факторів, які впливають на ціну опціону [29]. Основними є:

1) Дельта — визначає, на скільки зміниться ціна опціону при зміні ціні активу на 1 доллар.

2) Гамма — визначає, на скільки зміниться дельта при зміні ціни активу на 1 доллар.

3) Тета — вимірює щоденне зниження ціни опціону при наближені до строку завершення дії опціону.

4) Вега — вказує, наскільки ціна опціону чутлива до значних змін ціни активу.

Також вартим уваги є метрика «відкритий інтерес» (англ. Open Interest) — загальна кількість незакритих (невиконаних) контрактів для активу, таких як опціон чи фьючерси. Завдяки цій метриці можна зрозуміти, наскільки ліквідним та цікавим є даний контракт для інших трейдерів на ринку [6].

Однією з моделей прогнозування ціни на опціон є модель Блека-Шоулза — модель, що визначає теоретичну ціну на опціони. На даний момент існує досить багато інших моделей оцінки справедливої ціни опціону: модель Башельє, біноміальна модель оцінки опціонів, модель локальної волатильності (узагальненя моделі Блека-Шоулза), модель стохастичної волатильності, симуляція Монте-Карло тощо. Всі ці моделі пропонують свої підходи до вирішення але ми зупиномся саме на Блека-Шоулза через те, що вона є найбільш популярною, хоча і не оптимальною [30, 5]. Детальніше ми її розглянемо в наступному підрозділі.

1.3 Рівняння Блека-Шоулза

Для опису динаміки ціни опціону нашої моделі було введено рівняння Блека-Шоулза — параболічне нелінійне диференціальне рівняння в частинних похідних [17]. Перед його введеням, ми сформулюємо найважливіші умови для базового активу та опціону, як вони були сформульовані в [7]:

1) Процентна ставка заздалегіть відома і не змінюється з плином часу.

2) Ціна акції слідує випадковому руху в неперервному часі зі зміною дисперсії, пропорційної квадрату ціни акції. Таким чином, розподіл можливих цін акцій в кінці будь-якого обмеженого інтервалу є лог-нормальним. Швидкість зміни дисперсії прибутку від акцій є постійною.

3) Акція не виплачує дівідентів або інших виплат.

Також варто зазначити, що ми будемо розглядати американський опціон типу call як європейський через відсутність дівідентів та інших виплат, як це було запропоновано автором в роботі [5].

Саме рівняння Блека-Шоулза має вигляд [17, 14]:

$$f := \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0 \tag{1.1}$$

Де V це ціна опціону, t це час до завершення строку дії опціону, σ це волатильність — показує тенденцію зміни ціни на ринку впродовж певного часу [39], S це ціна базового активу на момент купівлі опціону та r це безризирока (процентна) ставка — теоретичне значення прибутку, яка отримується від активів з нульовим ризиком [11].

Для європейського опціону типу call, а тобто і для американського у нашому випадку, аналітичний розв'язок описується як [14]:

$$V_{\text{call}}(S,t) = S \cdot N(d_1) - K \cdot \exp(-r \cdot t) \cdot N(d_2) \tag{1.2}$$

Для деяких випадків американський опціон типу put дорівнює європейському [25]. Також, для американського опціону типу put ми вважатимемо, що аналітичним розв'язком нашого рівняння Блека-Шоулза є:

$$V_{\text{put}}(S,t) = -V_{\text{call}}(S,t) = K \cdot \exp(-r \cdot t) \cdot N(d_2) - S \cdot N(d_1) \tag{1.3}$$

де:

$$d_1 = \frac{\ln S/K + (r + \sigma^2/2)t}{\sigma\sqrt{t}}$$

$$d_2 = d_1 - \sigma * \sqrt{t}$$

і N — кумулятивна функція нормального розподілу, яка визначає ймовірність того, що випадкова величина прийме значення менше або еквівалентне заданому

значенню [18]:

$$N_X(x) = P(X \leqslant x), \forall x \in \mathbb{R} \tag{1.4}$$

Описавши математичну складову обраної моделі, ми можемо перейти до обговорення методів отримання її чисельних розв'язків, зважаючи на різні початкові та граничні умови. Далі ми детальніше розглянемо ці підходи, а також їхні переваги та обмеження для нашої задачі.

1.4 Методи скінченних різниць, елементів та Рунге — Кутти

Як було зазначено раніше, одним з найрозповсюджених методів розв'язання диференціальних рівнянь в частинних похідних є традиційні чисельні методи [38]. Їх загальний підхід — це розбиття пошуку розв'язку на кроки, які потім при обчисленні є апроксимацією розв'язків нашого рівняння. Методи скінченних різниць використовують апроксимацію розв'язків за допомогою ряду Тейлора — представлення функції у вигляді нескінченної суми доданків, які обчислюються зі значень функцій похідних в одній точці [36], коли методи скінченних елементів використовують інтегральний підхід [54].

Для простоти розглянемо функцію однієї змінної $f(x)$. Її похідна в точці x матиме вигляд [36]:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{f(h)} \tag{1.5}$$

де h це наш крок.

Формула (1.5) відома як апроксимація першого порядку точності. Взагалі існують інші більш точні, однак ця є загальноприйнятим варіантом при умові, що розмір кроку буде досить малим [31]. Варто зазначити, що методи скінченних різниць не можуть апроксимувати значення функції в усіх точках простору, а лише в точках сітки — вузлах, які визначаються шириною сітки [34]. Врешті, маючи апроксимоване значення похідних та підставляючи їх до нашого рівняння у частинних похідних забезпечить нас рекурсивною системою

рівнянь, розв'язання якої з використанням інформації про граничні умови надасть нам значення вузлів нашої сітки.

Хоча цей метод є досить точним в загальному, існують проблеми для більш складних диференціальних рівнянь в частинних похідних, таких як задачі з вільними межами [34]. Для нашої задачі при неправильній генерації сітки або визначенні домену — множині всіх можливих значень, для яких наше рівняння визначається ми можемо стикатися з проблемами точності та стабільності отриманих розв'язків [17]. Тому для таких типів задач існує метод скінченних елементів.

Загалом, метод скінченних елементів розділяє континуум системи, визначений нашим рівнянням у частинних похідних на скінченну кількість менших систем (елементів), поведінка яких визначається фіксованим набором параметрів [54]. Більше того, відомо, що рішення загальної системи — комбінації її окремих компонентів — працює за правилами, які є стандартними для дискретних проблем [54]. Спеціально для задачі прогнозування американських опціонів цей метод адаптували та створили алгоритм Бренана-Шварца [40], максимальна похибка апроксимації якого є меншою ніж для методів скінченних різниць.

Іншим підходом до апроксимації рішень диференціальних рівнянь в частинних похідних є метод Рунге—Кутти 4-го порядку, який є ефективним та широко використовуваним для звичайних диференціальних рівнянь [10]. Розглянемо задачу Коші — задача пошуку інтеграла диференціального рівняння, що задовольняє початковим умовам:

$$y' = f_\theta(x,y), \quad y(x_0) = y_0 \tag{1.6}$$

Тоді значення невідомої функції в точці x_{n+1} обчислюється відносно значення в попередній точці x_n за формулою:

$$\begin{aligned} y_{n+1} &= y_n + \frac{y}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ x_{n+1} &= x_n + h \end{aligned} \tag{1.7}$$

де h — крок, аналогічний (1.5) а коефіцієнти k_n розраховуються таким чином:

$$\begin{aligned} k_1 &= f_\theta(x_n, y_n), \\ k_2 &= f_\theta(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1), \\ k_3 &= f_\theta(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2), \\ k_4 &= f_\theta(x_n + h, y_n + hk_3) \end{aligned} \tag{1.8}$$

Похибка на кожному кроці становить $O(h^5)$, а сумарна похибка для кінцевого розв'язку є $O(h^4)$ [16].

1.5 Машинне навчання та нейронні мережі

Штучний інтелект — напрямок досліджень, який полягає у симуляції людської діяльності за допомогою комп'ютерних систем.

Машинне навчання — це підрозділ напрямку штучного інтелекту, який полягає у алгоритмічному досягнені симуляції людської поведінки. Для початку ми введемо такі поняття, як штучний нейрон (перцептрон), нейронна мережа та автоматичне диференціювання для кращого розуміння майбутніх розділів.

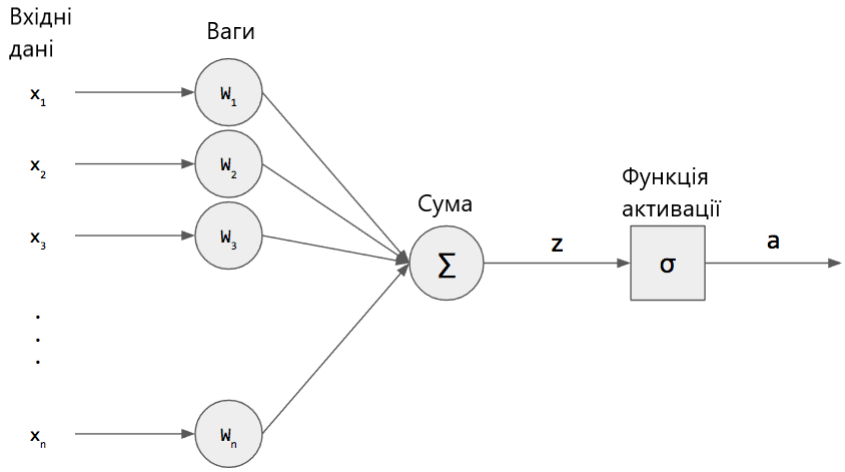


Рисунок 1.1 – Структура персептрона [13]

Персептрон — комп'ютерна модель сприйняття інформації мозком, яка була

запропонована Франком Розенблатом в 1958 році і вперше реалізована у вигляді електронної машини «Марк-1» в 1960 році [27]. Найбазовіший вид штучної нейронної мережі. Він складається з одного нейрону, вагів, вільний член (англ. bias) та функції активації. Згідно позначень рисунку 1.1 маємо $x_1, x_2, \dots x_n$ — вхідні значення, $w_1, w_2, \dots w_n$ — ваги. Також позначимо g — функція активації. Отже, вихідний результат нашого персептрону буде:

$$h(x) = g(w^T x + b) \tag{1.9}$$

Тепер, маючи уявлення про штучний нейрон, можемо перейти до нейронної мережі.

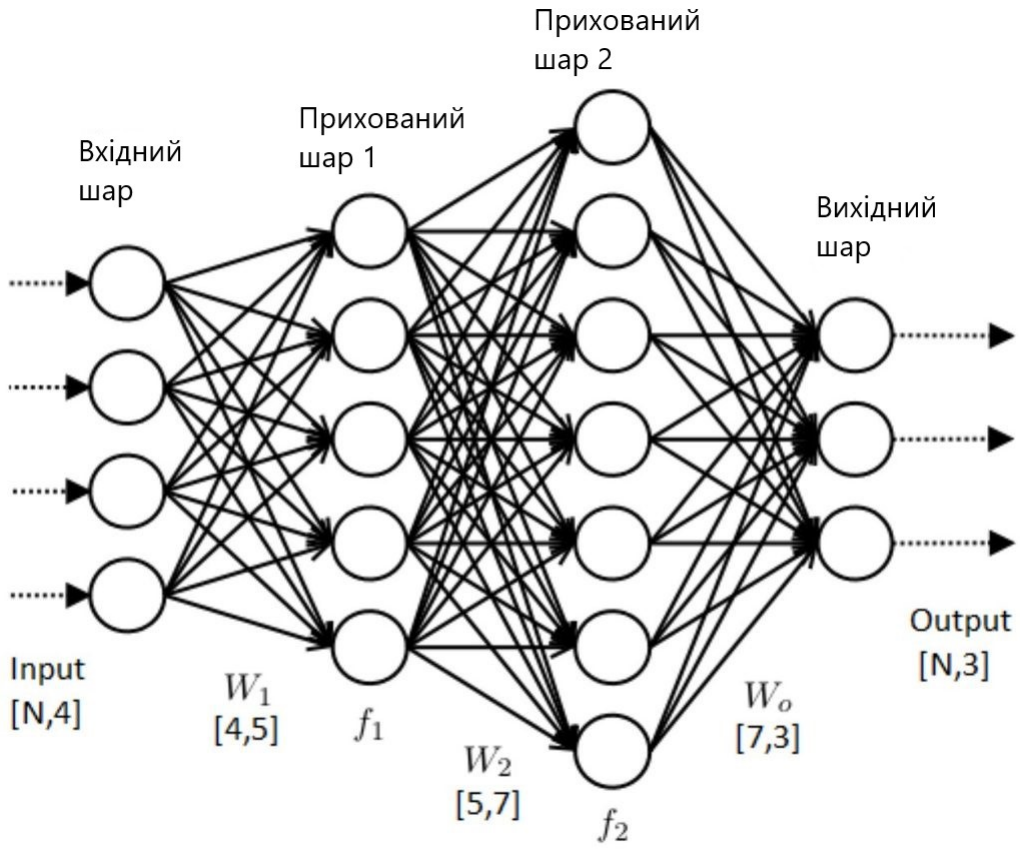


Рисунок 1.2 – Структура нейронної мережі прямого поширення [22]

Нейронна мережа — модель, яка грунтується на сукупності з’єднаних штучних нейронів, які приблизно моделюють поведінку нейронів головного мозгу. З розвитком нейромереж як апроксиматорів будь-якої функції [19], ми

маємо змогу розглянути велику кількість різних задач з нового боку. Є велика кількість різних архітектур нейронних мереж: нейронна мережа прямого поширення (англ. FNN, Feedforward neural network), згорткова нейронна мережа (англ. CNN, Convolutional neural network), рекурентна нейронна мережа (англ. RNN, Recurrent neural network) тощо. У цій роботі ми будемо фокусуватися саме на нейронних мережах прямого поширення.

Схема нейронної мережі прямого поширення показана на рисунку 1.2. Шар у нейронній мережі визначається як сукупність нейронів, кожен з яких окремо обробляє вхідні дані. Мережа визначається як сукупність нейронів, кожен з яких окремо обробляє вхідні дані і надає єдиний вихід. Отже, вихід шару — це вектор, розмірність якого дорівнює кількості нейронів у даному шарі. Хоча можливим є отримання прогнозу використовуючи лише один шар, більш оптимальним є підхід збільшення глибини нейромереж для покращення гнучності та моделюючих можливостей мережі [44]. Коли нейронну мережу називають «глибокою», це означає, що між вхідним і вихідним шарами є кілька прихованих шарів. Враховуючи, що кожен нейрон має вектор ваги, який використовується для обчислення його виходу, ми створюємо «матрицю вагів», щоб математично описати сукупність вагових векторів у кожному шарі. Інтуїція, що лежить в основі використання глибоких мереж замість одношарових з великою кількістю нейронів полягає у тому, що перші шари ідентифікують і навчаються розпізнавати низькорівневі закономірності у даних, коли останні шари ідентифікують високорівневі, що сильно допомагає у точності передбачення [44]. Також варто зазначити про важливість використання функцій активації. При їх відсутності передбачення нейронної мережі перетворюється на лінійне перетворення вхідних даних, що не дасть хороших результатів, якщо дані не є лінійно роздільними [15]. Нехай ми маємо два набори точок в n-вимірному евклідовому просторі X_0 і X_1. Якщо не існує

таких $n+1$ дійсних чисел $w_1, w_2, \ldots w_n, k$ при яких :

$$\begin{aligned} \sum_{i=1}^{n} w_i x_i > k, x \in X_0 \\ \sum_{i=1}^{n} w_i x_i < k, x \in X_1 \end{aligned} \tag{1.10}$$

Тоді X_0 і X_1 не є лінійно роздільними. Не менш важливою для структури нейронної мережі є процес її навчання та визначення оптимального набору вагових коефіцієнтів для нашої задачі. Щоб навчити нейронну мережу, нам потрібно надати їй навчальну вибірку та функцію втрат. Навчальна множина складається з вектору ознак і, можливо, відповідних цільових значень для кожного відповідного входу. Якщо навчальна множина містить цільові значення для входів, процес навчання називається контрольованим навчанням (англ. supervised learning), в іншому випадку - неконтрольованим навчанням (англ. supervised learning) [44]. У випадку нашої задачі ми будемо навчати модель використвуючи цільові значення. Оскільки кожна задача має свій набір відповідних функцій втрат, ми прагнемо надати огляд того, як оптимізується довільна функція втрат. Процес оптимізації ваг нейронної мережі складається з двох етапів: прямого та зворотного проходу. Під час прямого проходу вхідні дані надаються нейронній мережі і поширюються через мережу, що призводить до остаточного прогнозу. Після того, як прогноз зроблено, помилка вимірюється по відношенню до істинного (цільового) значення за допомогою визначеної заздалегіть функції втрат. Існує велика кількість різних функцій втрат, які використовуються для різних задач: середньоквадратичне відхилення, середнє відхилення по модулю, коефіцієнт детермінації, перехресна ентропія тощо [33]. Для нашої задачі ми будемо використовувати середньоквадратичне відхилення:

$$MSE = \frac{\sum_{i=1}^{n} \left(y_i - p(x_i)\right)^2}{n} \tag{1.11}$$

де $p(x_i)$ — передбачення моделі для запису x_i, а y_i — справжнє значення.

Далі обчислюється градієнт функції втрат для кожної ваги і вони коригуються відповідно до напрямку, визначеного градієнтом. Градієнт диктує «напрямок», в якому мають бути скориговані ваги, щоб мінімізувати нашу функцію втрат [44]. Цей процес повторюється, ваги постійно оновлюються, а градієнт функції втрат відносно ваг перераховується при кожній ітерації. Таким чином, ми очікуємо, що в кінцевому підсумку отримаємо набір ваг, який достатньою мірою мінімізує нашу функцію втрат. Процес обчислення градієнтів і підбору ваг відповідно до градієнта функції втрат називається градієнтний спуск. У машинному навчанні прийнято масштабувати градієнт на коефіцієнт θ, також відомий як швидкість навчання (англ. learning rate), перед коригуванням вагів. Хоча реалізація градієнтного спуску не є складною, процес обчислення градієнта функції втрат відносно ваг є громіздким. Для цього ми використовуємо автоматичне диференціювання. Рисунок 1.3 точно відображає

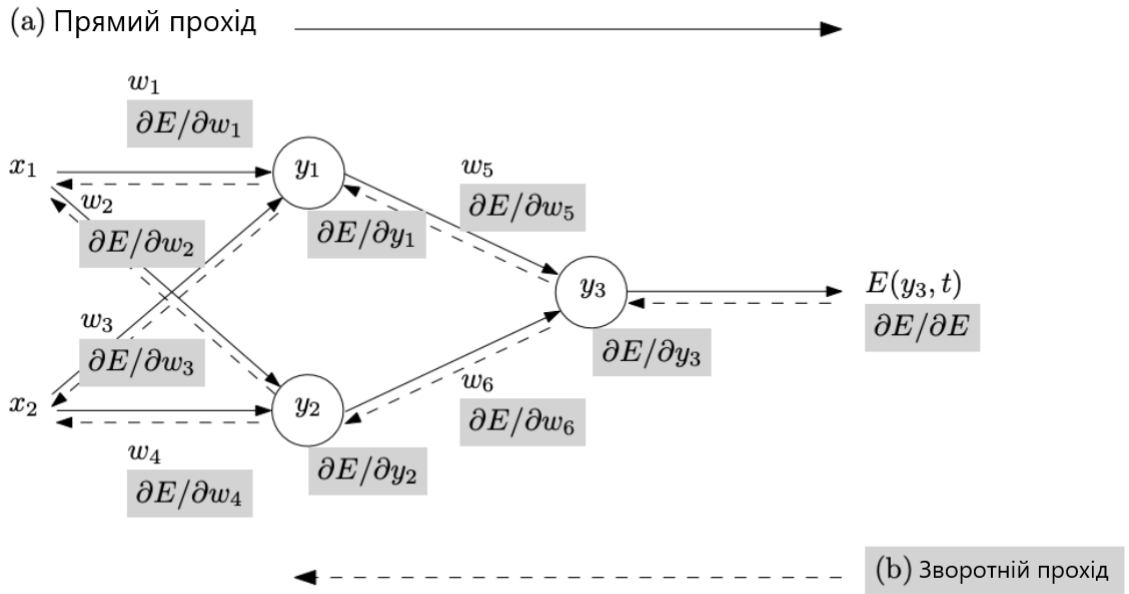


Рисунок 1.3 – Схема автоматичного диференціювання [34]

процес градієнтного спуску — зворотній прохід. Він залежить від правила ланцюга для диференціювання [50]. Нехай $f(y)$ — довільна функція однієї

змінної, а y — функція від деякого x. Правило ланцюга стверджує, що:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} \cdot \frac{\partial y}{\partial x} \tag{1.12}$$

Фактично весь процес диференціювання та у подальшому навчанні нейронної мережі заснований на рівності 1.12, незалежно від архітектури. Проте варто зазначити, що для знаходження чисельних розв'язків диференціального рівняння в частинних похідних звичайних нейромереж прямого поширення зазвичай недостатньо, так як ми не можемо гарантувати, що ті залежності в даних, які отримає нейронна мережа, задовольнятимуть граничні умови та закони природи, які впливають на наше рівняння. Саме для таких задач була запропонована нова архітектура — фізично інформована нейронна мережа [42].

Великою проблемою для вирішення диференціальних рівнянь в частинних похідних за допомогою нейронних мереж є потреба у великій кількості даних, які є дорогими для отримання. Автори статті [42] запропонували новий підхід до цього, який використовує окрім звичайних нейронних мереж прямого поширення також диференціальні рівняння, які дозволяють навіть при маленькій кількості даних отримувати закономірності з великорозмірних даних, отриманих з експерименту [42]. За своєю конструкцією ці моделі мають враховувати певні умови, задані за допомогою диференціальних рівнянь, що випливають з спостережуваних даних. Загальний вигляд фізично інформації нейронної мережі на рисунку 1.4. Головною відмінністю є функція втрат, яка складається не тільки з вимірювання помилки між передбаченними та цільовими значеннями, а і з відповідності передбачень граничним та початковим умовам [34]. Розглянемо приклад функції втрат для фізично інформованої нейронної мережі:

$$MSE = MSE_{x_b} + MSE_{t_b} + MSE_f \tag{1.13}$$

Де x_b та t_b відповідають за простір та час відповідно, у випадку нашої задачі прогнозування ціни опціону простором координат є ціна активу [34], а f

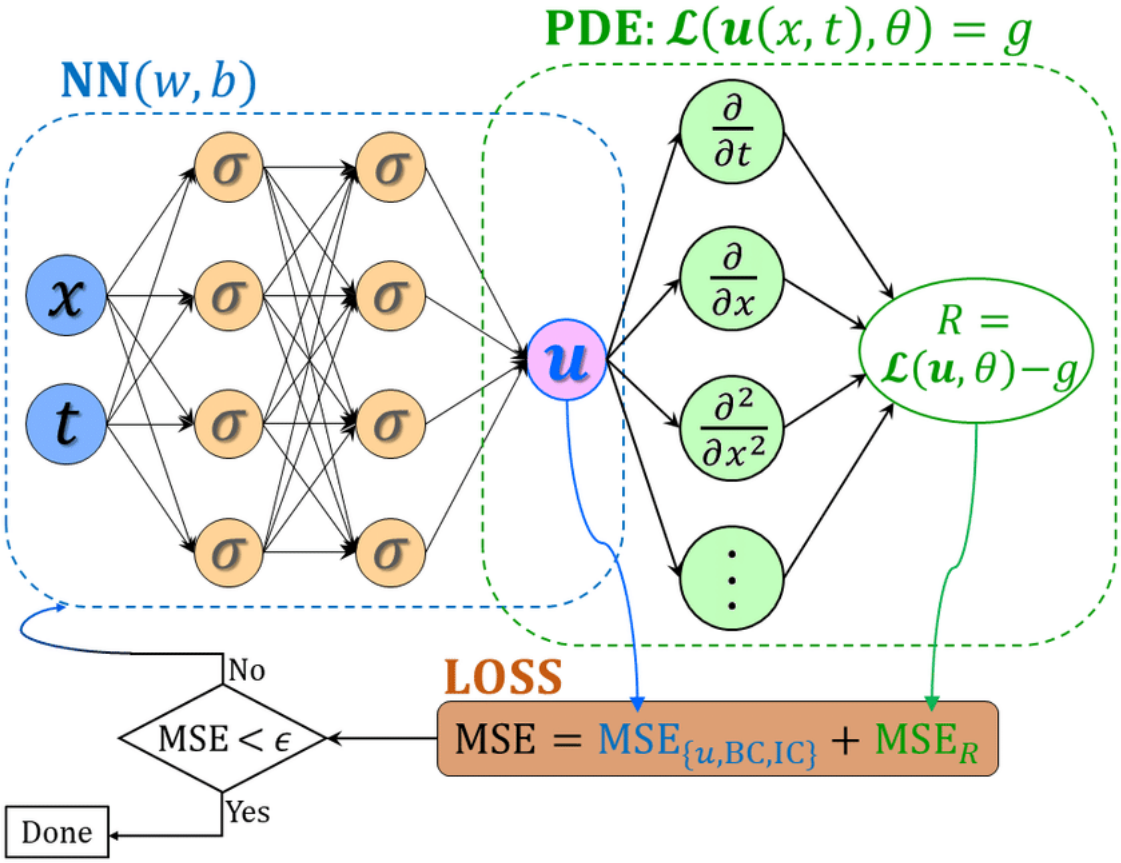


Рисунок 1.4 – Архітектура фізично інформованої нейронної мережі [32]

визначається як:

$$f := u_t + \mathcal{N}[u] \tag{1.14}$$

де $\mathcal{N}$ — нелінійний диференціальний оператор нашої моделі, а u — невідомий розв'язок системи [42]. Також загальне визначення диференціального рівняння з частинними похідними:

$$u_t + \mathcal{N}[u] = 0, \ x \in \Omega, \ t \in [0, T] \tag{1.15}$$

де Ω — простір параметрів. Тобто мінімізуючи 1.13, ми покращуємо передбачення нейронної мережі та одночасно з цим знаходимо розв'язки, які задовольнятимуть фізичні закони та саме диференціальне рівняння 1.15.

Використання цієї архітектури для нашої задачі буде описано детальніше у майбутніх розділах. Ми же зараз розглянемо ще один підхід до використання нейронних мереж для задач знаходження чисельних розв'язків диференціальних

рівнянь в частинних похідних — нейронні диференціальні рівняння, які схожі з фізично інформованими нейронними мережами модельно, але ідейно інші.

1.6 Нейронні диференціальні рівняння

Нейронне диференціальне рівняння — це диференціальне рівняння, яке використовує нейронну мережу для параметризації векторного поля [28]. Простим прикладом є звичайне диференціальне рівняння 1.6 [37]. Варто зазначити, що $f_\theta : \mathbb{R} \times \mathbb{R}^{d_1 \times \cdots \times d_k} \to \mathbb{R}^{d_1 \times \cdots \times d_k}$, де θ представляє собою деякий вектор вивчених параметрів, а f_θ — будь-яка звичайна архітектура нейронної мережі [28].

Для формулювання принципу роботи нейронних диференціальних рівнянь ми розглянемо залишкову нейронну мережу [12]:

$$y_{j+1} = y_j + f_\theta(j, y_j) \tag{1.16}$$

де $f_\theta(j, y_j)$ — це j-ий залишковий блок.

Тепер, застосувавши до нашого рівняння 1.6 метод Ейлера [1], ми отримуємо:

$$y(t_{j+1}) = y(t_j) + \Delta t f_\theta(t_j, y(t_j)) \tag{1.17}$$

Що фактично повторює формулу 1.16. Зробивши це спостереження можемо зробити висновок, що нейронні диференціальні рівняння можна розглядати як неперервний ліміт залишкових нейронних мереж [28].

Варто зазначити, що метод нейронних диференціальних рівнянь використовує нейронні мережі для уточнення диференціальних рівнянь, коли нами вище зазначений метод фізично інформованих нейронних мереж використовує їх для отримання числових розв'язків [28].

Описавши наше нейронне диференціальне рівняння теоретично, ми перепишемо формулу 1.17 у зручний для нас вигляд, перенісши $y(t_j)$ в ліву

частину, та отримаємо:

$$\Delta t f_\theta(t_j, y(t_j)) = y(t_{j+1}) - y(t_j) \tag{1.18}$$

Розбиваючи формулу 1.18 на компоненти, отримуємо що:

1) $y(t_{j+1})$ — це цільове значення ціни опціону.

2) $y(t_j)$ — це розв'язок нашого рівняння будь-яким методом.

3) $f_\theta(t_j, y(t_j))$ — це передбачення нашої нейронної мережі для уточнення розв'язку диференціального рівняння в частинних похідних.

4) Δt — це крок сітки для простору часу.

Так як центральною ідеєю є використання методів розв'язання диференціальних рівнянь як частини процесу навчання нейронної мережі [28], то ми посилаючись на [48] можемо сказати, що рівняння 1.18 — універсальне диференціальне рівняння, так як воно використовує універсальний апроксиматор у вигляді нейронної мережі [19] і надалі ми будемо оперувати саме цим терміном.

Маючи ці теоретичні знання ми тепер маємо можливість застосувати їх для подальшого моделювання процесу ціноутворення опціонів, деталі якого будуть окреслені у майбутніх розділах.

Висновки до розділу 1

В цьому розділі було оглянуто основні теоретичні відомості про об'єкт та предмет дослідження, а саме опціони як фінансовий інструмент, їх типи, а також існуючі методи ціноутворення. Були надані визначення основних концепцій, таких як рівняння Блека-Шоулза, метод Рунге-Кутти 4-го порядку, методи машинного навчання, зокрема нейронні мережі прямого поширення, та метод нейронних диференціальних рівнянь. Розглянуто основні метрики, що оцінюють якість передбачень нейронних мереж.

2 ПІДГОТОВКА ДО ПРОВЕДЕННЯ ДОСЛІДЖЕННЯ

В даному розділі знаходиться огляд основних інструментів і методів для аналізу даних та були зазначені використані ресурси для моделювання. Також були розглянуті роботи інших авторів за цією тематикою.

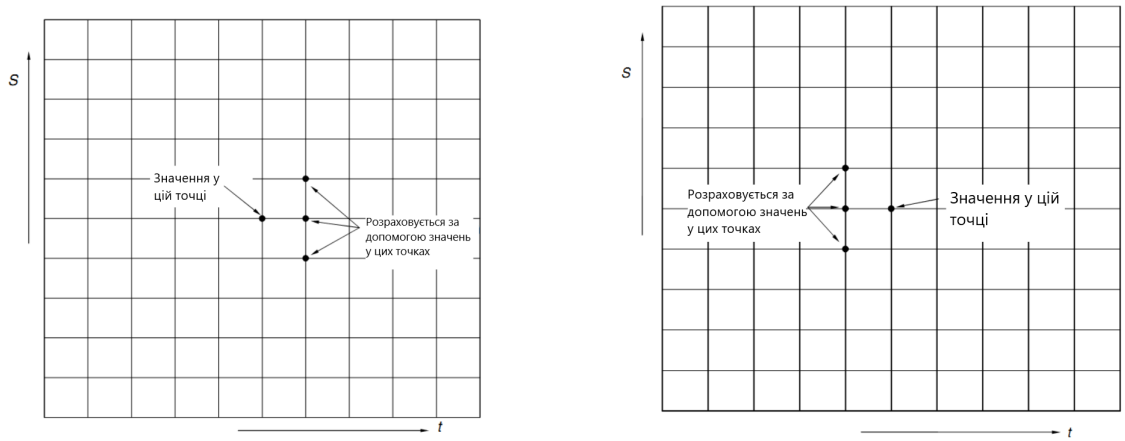
2.1 Огляд суміжних робіт

У минулих розділах ми розповіли про теорію різних підходів до задачі прогнозування ціни на опціони. В цілому, методи вирішення цієї задачі можна поділити на 3 класи за типом використання:

1) Використання виключно чисельних методів.
2) Використання виключно методів машинного навчання.
3) Використання і чисельних методів і методів машинного навчання.

Тепер ми детальніше розглянемо роботи та їх специфіку для кожного класу.

Для першого класу ми розглянемо роботи [23, 51, 4]. Однією з сильних сторін цього підходу є пояснюваність результатів та їх проста інтерпретація. Всі ці три роботи мають спільним використання методів скінченних різниць [31], а саме явний метод (англ. explicit method). Його аналогом є неявний метод (англ. implicit method), різницю між якими висвітлив [51] на рисунку 2.1. Також вони порівнювалися з методом Кренка-Нікольсона, та методом LU декомпозиції, який в загальному показав себе краще за явний метод скінченних різниць, але його обмеженням є неможливість використання для американських опціонів через можливість виконання його контракту у будь-який момент часу до закінчення часу дії. Також вартим уваги є дані, які використовувалися у цих роботах. Так як їх фокус був саме на порівнянні стабільності та точності отриманих розв’язків, автори використовували синтетичні дані, які генерували з статичними параметрами, окрім базової ціни активу, що є обмеженням даних робіт.



(а) **(б)**

Рисунок 2.1 – Різні види методів скінченних різниць: (а) явний, (б) неявний [51].

На відміну від першого класу, клас робіт, які використовували виключно методи машинного навчання, для експериментів використовують здебільшого неперервні дані з фондових ринків у певний період часу [21, 8]. Перевагами данного класу робіт є швидкість обчислень та отримання розв’язків. Ці роботи фокусуються на порівнянні відомих методів машинного навчання для прогнозування ціни опціону, таких як метод опорних векторів, випадковий ліс (англ. Random Forest), XGBoost, нейронні мережі прямого поширення, рекурентні нейронні мережі а саме довга короткострокова пам’ять (англ. LSTM) та генетичні алгоритми. Хоча ці методи і показують хороші результати, вони можуть страждати від перенавчання, адже ніяк явно не опрацьовують граничні умови та фізичні закони, які мають диференціальні рівняння в частинних похідних та при зміні розподілів даних або поведінки ринку можуть призводити до неоптимальних розв’язків.

Для цієї задачі була створена модель фізично інформованої нейронної мережі, про яку ми детально говорили у розділі 1.5. Активно задачою прогнозування ціни на опціони використовуючи фізично інформовані нейронні мережі займалися Louskos A. в [34] та Ashish Dhiman і Yibei Hu в [14]. Цікавим підходом у цих роботах є процес навчання: явним методом з формули 1.13 обраховується лише значення функції втрат на її граничних точках, де значення ціни опціону відоме

з граничних умов моделі Блека-Шоулза, а передбачення для негранічних точок напряму не порівнюється з реальною ціною для цих записів. Замість цього відомі параметри заносяться у формулу 1.1, яка при коректних значеннях відповідних параметрів зануляється. Цей підхід вирішує описану проблему другого класу, але в свою чергу є довшим для навчання і показує гіршу точність відносно інших моделей машинного навчання. У роботі автора [34] для тестування якості моделі не використовуються дані з фондових ринків, а лише синтетичні, коли в свою чергу [14] використовує для тесту також дані з ресурсу Optionmetrics.

Також вартим уваги є те, що в описаних роботах першого класу автори використовують лише європейські опціони, в роботах другого класу — європейські call та американські put у [14], американські call у [34]. У нашій роботі, яка буде представляти собою третій клас змішанного підходу до прогнозування ціни опціону ми будемо розглядати одночасно американські опціони типу call та put, а також розглянемо підходи до генерації синтетичних даних не лише для ціни базового активу та часу до завершення дії опціону, а і для грецьких літер і метрики OpenInterest, які ми описали у розділі 1.2.

2.2 Використанні інструменти та ресурси

Мовою програмування було вибрано Python v3.11 [2], який є досить ефективним рішенням для задач машинного навчання, адже він має велику кількість готових бібліотек та ресурсів, які дозволяють ефективно розв’язувати різного типу задачі. Середовищем для програмування було обратно Jupyter Notebook [49] — інструмент з відкритим вихідним кодом, який виступає у ролі віртуального лабораторного щоденника, що підтримує робочі процеси, код, дані та їх візуалізації. Він є простим для розуміння та зручним для проведення експериментів і активно використовується науковою спільнотою для дослідницьких робіт. Основною бібліотекою для створення моделі був PyTorch v2.3 [41]. Ця бібліотека надає навчальні посібники, приклади та інші ресурси, які дозволяють швидко та ефективно будувати рішення з машинного навчання.

Основним ресурсом для експериментальних даних був вебсайт historicaloptiondata.com — цей вебресурс дає безкоштовний доступ до записів купівлі опціонів з фондових ринків. Записи містять всю необхідну нам інформацію, таку як ціна активу, ціна опціону, термін дії контракту опціону, волатильність а також всі необхідні Greeks та метрику Open interest. Для зберігання даних та їх обробки використовувалася бібліотека Pandas [3] через її швидкість, зручність та функціонал для попереднього аналізу даних. Також для обробки даних була використаною бібліотека Scikit-learn [45], яка поєднує в собі велику кількість вбудованих алгоритмів для будь-яких задач з машинного навчання. Для візуалізації була використана бібліотека matplotlib [35] — це портативний пакет для 2D графіки та обробки зображень, який переважно націлений на візуалізацію наукових, інженерних та фінансових даних. Особливості включають створення кількох осей та фігур на сторінці, інтерактивну навігацію, багато визначених стилів ліній та символів, зображення, антиаліазинг, змішування альфа-каналів, діаграми для дат та фінансів, управління шрифтами відповідно до стандартів W3C та підтримка FreeType2, легенди та таблиці, псевдокольорові графіки, математичний текст та багато іншого.

Висновки до розділу 2

В цьому розділі було проведено огляд основних інструментів для пошуку, зберігання, аналізу та обробки даних, які були взяті з вебресурсу historicaloptiondata, який є базою даних контрактів на опціони. Були зазначені основні інструменти для моделювання, такі як Python v3.11, PyTorch та Pandas. Було проведено аналіз робіт за нашою тематикою та було виділено 3 класи робіт за типом виконання методів прогнозування цін на опціони, були висвітлені сильні та слабкі їх сторони а також обмеження досліджень. Було зазначено, що обраний нами метод нейронних диференціальних рівнянь є представником третього класу, який має закріпити за собою сильні сторони першого і другого.

3 ПОБУДОВА МОДЕЛЕЙ ТА ПОРІВНЯННЯ ЇХ РЕЗУЛЬТАТІВ

В даному розділі описані основні та додаткові параметри побудованих моделей для дослідження, процес їх навчання та результати прогнозування. Описаний процес обробки даних та їх статистичний аналіз. Також було розглянуто та використано запропонований підхід до генерації синтетичних даних.

3.1 Попередня обробка даних

Для використання наших даних з фоднового ринку у тренуванні моделі їх потрібно було обробити. Всього на початку ми мали 958,448 записи купівлі опціонів за 20 серпня 2019 року. Для початку ми одразу залишили лише ті записи, де поточна ціна базового активу знаходилася між 40 та 200 долларами. Такий вибір обгрунтовано тим, що активи у цьому ціновому проміжку мають вищу ліквідність, що в свою чергу призводить до «вужчого» проміжку між Bid та Ask цінами та робить ціноутворення надійнішим. Внаслідок цього ми отримали 446,434 записи для подальшої роботи. Віднявши від кінця контракту опціону його початок ми знайшли кількість днів до завершення дії контракту.

Наступним нашим кроком було знаходження цільвого значення ціни опціону. З даних ми мали дві колонки: «Bid» — максимальна ціна, яку покупець хоче заплатити за контракт опціону та «Ask» — мінімальна ціна, яку продавець контракту прийме [9]. Як було запропоновано [14], ми будемо вважати, що справжня ціна опціону на ринку є середнім серед цих двох:

$$V_{market} = \frac{V_{bid} + V_{ask}}{2} \tag{3.1}$$

Після цього ми розділили наші дані на 2 датасети з опціонами типу call та put

відповідно, маючи по 223,167 записів в кожному. Для аналізу розподілів ціни на опціон та ціни базового окрім звичайних метрик, таких як мінімальне, максимальне, середнє, стандартне відхилення нам також потрібні коефіцієнт асиметрії — числова характеристика яка показує як розташовані дані біля середнього та коефіцієнт ексцесу — числова характеристика яка показує стрімкість підвищення кривої розподілу у порівнянні з нормальним розподілом [24]. В таблиці 3.1 зображено аналіз розподілу значень ціни опціону типу call, а на рисунку 3.1 сам розподіл.

Мінімальна ціна опціону	0.00
Максимальна ціна опціону	180.72
Середня ціна опціону	10.84
Стандартне відхилення	15.90
Коефіцієнт асиметрії	2.64
Коефіцієнт ексцесу	9.55

Таблиця 3.1 – Статистичний огляд цін опціонів типу call

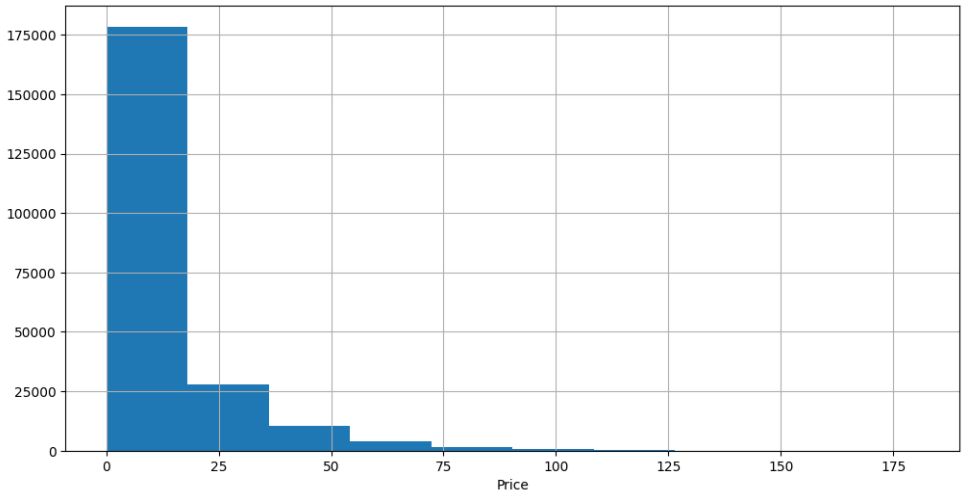


Рисунок 3.1 – Графік розподілу цін на опціони типу call

В таблиці 3.2 зображено аналіз розподілу значень ціни опціону типу put, а на рисунку 3.2 сам розподіл.

Як можна побачити по таблицям, для опціонів типу put коефіцієнти асиметрії та ексцесу значно більші, хоча їх середнє та стандартне відхилення схожі, що

свідчить про різницю в «хвостах» розподілів та концентрацію точок всередині відносно середнього.

Мінімальна ціна опціону	0.00
Максимальна ціна опціону	487.00
Середня ціна опціону	10.85
Стандартне відхилення	20.21
Коефіцієнт асиметрії	5.80
Коефіцієнт ексцесу	64.06

Таблиця 3.2 – Статистичний огляд цін опціонів типу put

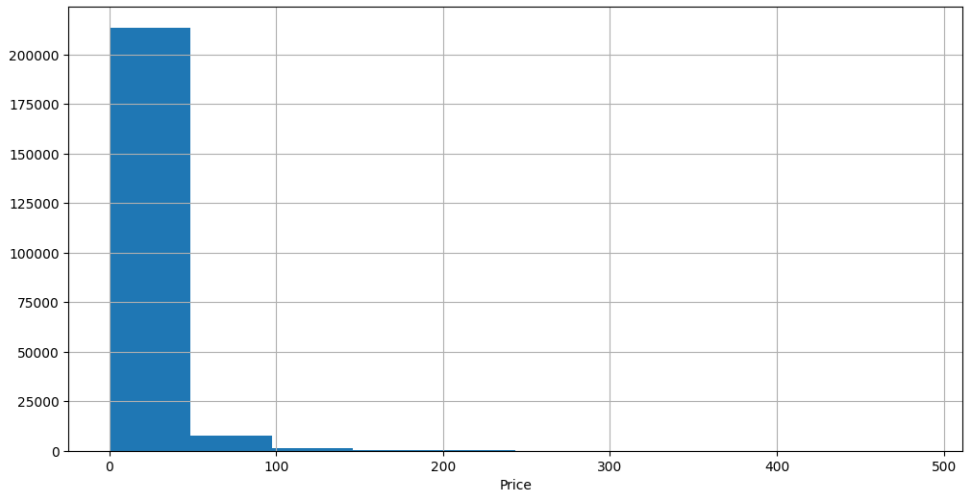


Рисунок 3.2 – Графік розподілу цін на опціони типу put

3.2 Генерація синтетичних даних

Перед побудовою моделей для прогнозування ціни на опціони та їх навчання на даних з фоднового ринку, описаних у розділі 3.1, ми розглянемо підхід до генерації синтетичних даних, адже вони є безкоштовними для отримання та їх розмір не обмежений ресурсом, який їх надає.

Для генерації часу до завершення дії опціону у днях, ціни базового активу в долларах ми будемо використовувати параметри приведені у таблиці 3.3 для

обох опціонів типу call та put.

Ціна виконання, K	120
Безризикова ставка, r	0.03
Волатильність, σ	0.4
Ціна базового активу, S_{range}	[40,200]
Час до завершення дії опціону, T_{range}	$[0, 1095]$

Таблиця 3.3 – Параметри згенерованих даних

Наш датасет для кожного з опціонів put та call буде складатися з 10,000 записів, серед яких 3000 записів мають граничні значення ціни базового активу, 1000 записів мають опціон контракти, термін дії яких минув, вимагаючи їх негайного виконання та інші 6000 записів між граничними умовами. Граничні умови дають нам точні значення ціни опціону без потреби у використанні чисельних методів. Так як граничні умови для ціни є двох видів, то з 3000 записів відповідно 1500 мають максимальну ціну базового активу і 1500 — мінімальну. Таке розбиття датасету обумовлене тим, що найбільш важливими для дослідження є записи, точної ціни яких ми не знаємо заздалегідь і вони мають складати більшість всього набору даних. Для обрахунку ціни опціону для таких записів ми будемо використовувати метод Бренана-Шварца [40], який ми описали в розділі 1.4. Для нього нам необхідно буде встановити штучну верхню межу для ціни базового активу так само, як і для часу до завершення дії опціону. У нашій роботі це буде 400 долларів. Відповідні кроки для сітки простору складають $\Delta V = 1$ та $\Delta t = 1$. Граничні умови ціни опціону put:

1) $S = S_{min} : \ V = K$

2) $S = S_{max} : \ V = 0$

3) $t = 0 : \ V = max(K - S, 0)$

Граничні умови ціни опціону call:

1) $S = S_{min} : \ V = 0$

2) $S = S_{max} : \ V = S_{max} - K \cdot e^{-r \cdot t}$

3) $t = 0 : \ V = max(S - K, 0)$

Для генерації записів ціни базового активу та часу до завершення дії ми

будемо випадковим чином генерувати значення з рівномірного розподілу на проміжку, який зазначений у таблиці 3.3 для відповідних параметрів.

В таблиці 3.4 зображено аналіз розподілу значень ціни опціону типу call для згенерованих даних, а на рисунку 3.3 сам розподіл.

Мінімальна ціна опціону	0.00
Максимальна ціна опціону	200.00
Середня ціна опціону	44.45
Стандартне відхилення	49.46
Коефіцієнт асиметрії	2.04
Коефіцієнт ексцесу	4.17

Таблиця 3.4 – Статистичний огляд цін опціонів типу call для згенерованих даних

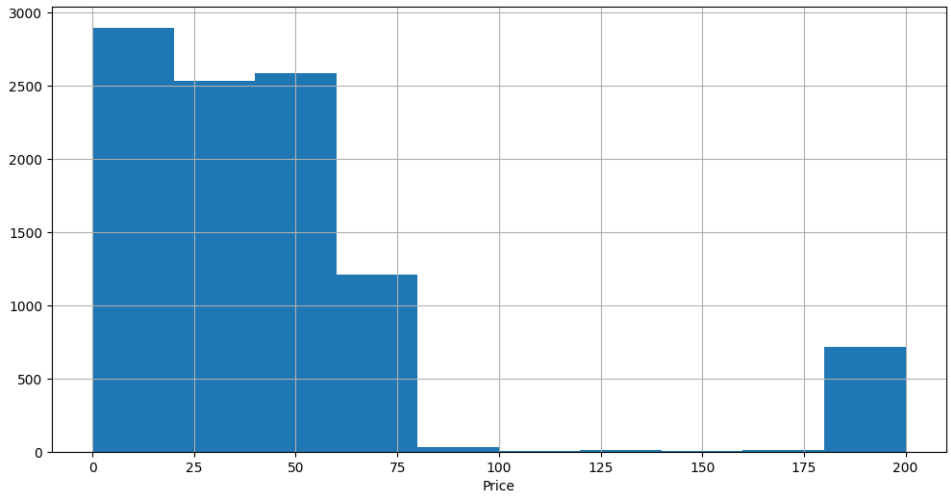


Рисунок 3.3 – Графік розподілу цін на опціони типу call для згенерованих даних

В таблиці 3.5 зображено аналіз розподілу значень ціни опціону типу put для згенерованих даних, а на рисунку 3.4 сам розподіл.

Проаналізувавши статистичні метрики з таблиць згенерованих даних і порівнявши їх з відповідними таблицями 3.1 та 3.2 для даних з фондових ринків можна побачити, що їх розподіли досить суттєво відрізняються у коефіцієнтах асиметрії та ексцесу, що у майбутньому відобразиться на результатах. Різницю у їх розподілах продемонстровано на прикладі ціни

Мінімальна ціна опціону	0.00
Максимальна ціна опціону	120.00
Середня ціна опціону	38.90
Стандартне відхилення	32.90
Коефіцієнт асиметрії	0.78
Коефіцієнт ексцесу	0.42

Таблиця 3.5 – Статистичний огляд цін опціонів типу put для згенерованих даних

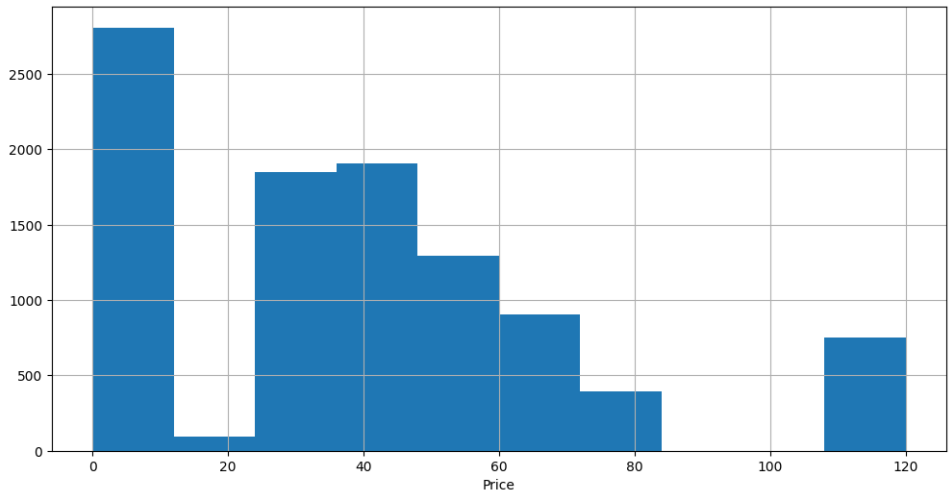


Рисунок 3.4 – Графік розподілу цін на опціони типу put для згенерованих даних

опціонів типу put на рисунку 3.5.

Як було вказано у розділі 1.6, а саме у формулі 1.18, для уточнення наших розв'язків нейронною мережею ми будемо використовувати грецькі літери та метрику Open interest, які фактично відображають динаміку ціни опціонів. Для тренування майбутніх моделей ми також маємо згенерувати їх. Для цього ми так само використали генератори випадкових чисел, але для кожної метрики розподіл визначався попереднім аналізом розподілу відповідної метрики даних з фондового ринку як у таблиці 3.2 та рисунку 3.2. Також для перевірки наскільки такий підхід до генерації є точним використовувався критерій узгодженості Колмогорова-Смірнова [52], де нульовою гіпотезою була статична відповідність двох розподілів і її альтернативою — їх суттєва відмінність.

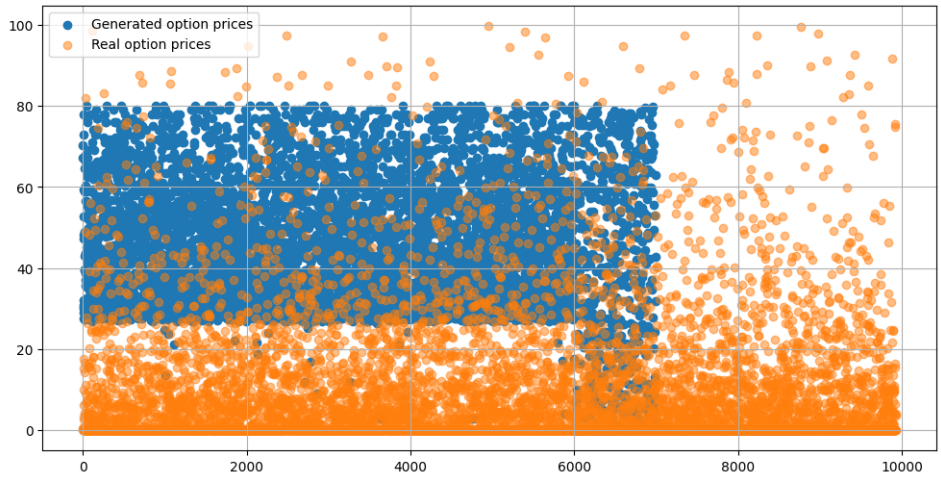


Рисунок 3.5 – Розподіл цін на опціони типу put для справжніх даних та згенерованих

Однією із особливостей нашої роботи є те, що ми випадковим чином призначали значення метрик до відповідного запису опціону. Відповідно до цього, фактичні взаємозв'язки між нашими метриками є також випадковими і не обов'язково відображають справжню динаміку опціону, що для професійного інвестора може бути проблемою через їх ірраціональність. Утім, з методологічної точки зору для роботи це не становить суттєвої проблеми, адже основною ціллю роботи є розробка алгоритму для знаходження чисельних розв'язків на широкому класі даних.

3.3 Підготовка архітектур моделей

Маючи повне розуміння розподілів наших даних, ми можемо перейти до побудови моделей, але для початку ми в деталях пояснимо призначені дані до компонент рівняння 1.18:

1) $y(t_{j+1})$ — це відома ціна опціону з наших даних.

2) $y(t_j)$ — це знайдена нами ціна опціону методом Рунге-Кутти 4-го порядку з розміром сітки 50×50.

3) $f_\theta(t_j, y(t_j))$ — це передбачення нашої нейронної мережі на основі метрик Open interest та грецьких літер, яке має уточнювати розв'язок нашого

чисельного методу.

4) Δt — 1 день.

У цій роботі ми розглянемо 2 архітектурних варіанти:

1) адаптовану версію моделі, запропонованої у роботі автора [34].
2) версію моделі, представленої у роботі автора [14].

Ми порівняємо різницю в їх точності, збіжності та зробимо відповідні висновки.

Модель у [34] фактично повторює звичайну структуру нейронної мережі прямого поширення, яка зазначена на рисунку 1.2. Вона має 10 прихованих шарів по 50 нейронів в кожному, з Tanh як функцією активації та Adam як оптимізатор. Наша ж перша модель буде мати трохи інший вигляд: Перший прихований шар складатиметься з 128 нейронів, другий з 256, між ними та вихідним шаром знаходиметься 20 шарів з 64 нейронами в кожному, ReLU як функція активації та бач нормалізація (англ. batch normalization) в кожному з шарів для покращення швидкості навчання, запобігання проблеми внутрішнього коваріантного зміщення та перенавчання [20].

Друга модель є архітектурно складнішою і показана на рисунку 3.6. Вона складається з:

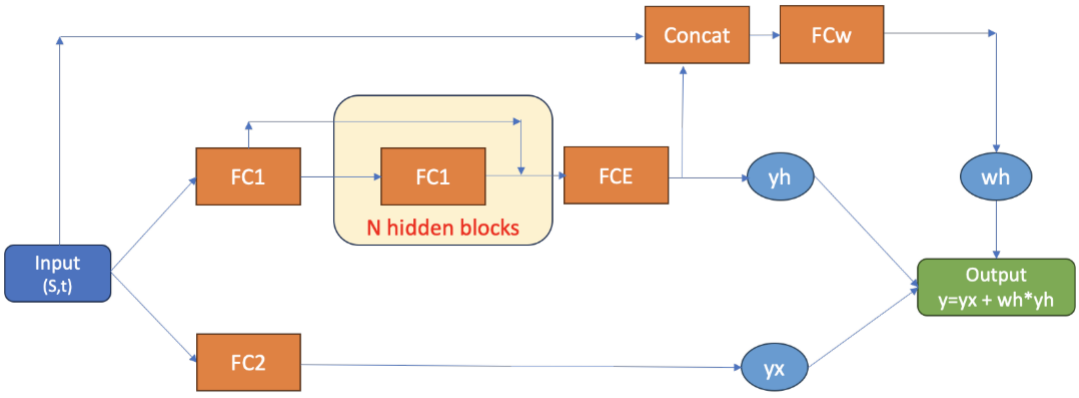


Рисунок 3.6 – Архітектура нейронної мережі для другої моделі [14]

1) FC1 — один прихований шар з 64 нейронами та функцією активації Tanh.
2) FCH — 20 прихованих шарів з 64 нейронами в кожному та функцією

активації Tanh, де після кожного шару використовується метод skip connection, описаний рівністю 1.16 з виходом шару FC1.

3) FCE — прихований шар, який переводить 64 вихідні ознаки у одну, не має функції активації.

4) FC2 — прихований шар, який переводить вхідну кількість ознак у одну, не має функції активації.

5) FCW — прихований шар, який отримує на вхід вихідне значення блоку FCH та вхідні ознаки та переводить їх у одну ознаку, використовує функцію активації Tanh.

6) OUTPUT — комбінація вихідного значення FC2 та FCE з вагою вихідного значення блоку FCW.

Тепер маючи дві різні архітектури моделей ми можемо перейти до розгляду оптимізаторів та гіперпараметрів до них.

3.4 Підготовка методів оптимізації, їх параметрів та алгоритмів масштабування

Перед тим як перейти до фінального навчання моделі, нам потрібно вибрати найкращий метод оптимізації, його параметри та гіперпараметри моделі, такі як швидкість навчання. Також ми порівняємо, який з алгоритмів масштабування (англ. scaler) кращий саме для нашої задачі: стандартний (англ. StandardScaler) або мінімум-максимум (англ. MinMaxScaler).

Для методів оптимізації ми вибрали наступні для обох опціонів типу put та call параметри:

Також варто зазначити, що всі експерименти по пошуку найращих параметрів проводилися на вибірці з даних у розмірі 20,000 випадкових записів, де 16,000 йшли на тренування та 4,000 на валідацію для перевірки та своєчасного запобігання перенавчання моделі на наших даних [47].

Тепер ми розглянемо результати порінянь двох типів архітектур, зазначених у розділі 3.2 для результатів пошуку найкращого методу оптимізації на прикладі

Метод оптимізації	Його параметри
SGD	$\theta = 5 \cdot 10^{-4}, momentum = 7 \cdot 10^{-1}$
RMSprop	$\theta = 5 \cdot 10^{-4}, momentum = 0$
Adam	$\theta = 5 \cdot 10^{-4}, betas = (9 \cdot 10^{-1}, 9.9 \cdot 10^{-1})$
Adamax	$\theta = 5 \cdot 10^{-4}, betas = (9 \cdot 10^{-1}, 9.9 \cdot 10^{-1})$

Таблиця 3.6 – Обрані методи оптимізації та їх гіперпараметри

опціону типу call.

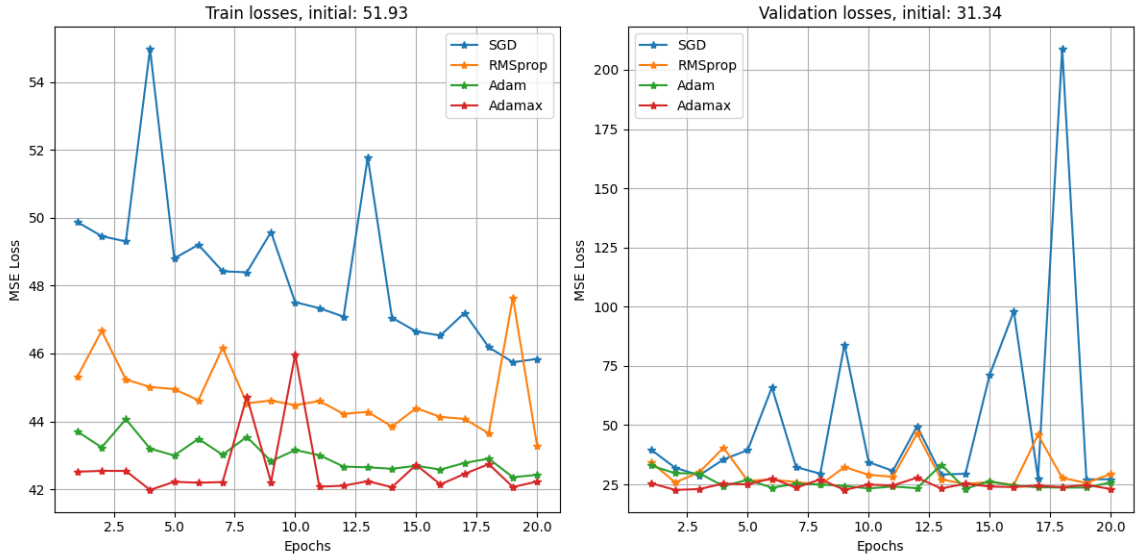


Рисунок 3.7 – Значення функції втрат для архітектури №1 для різних оптимізаторів для опціону типу call

Як можна побачити з порівняння рисунків 3.7 та 3.8, друга архітектура показує себе краще і має більш «гладку» поверхню, внаслідок чого сходження до мінімуму відбувається плавніше і не має таких сильних скачків у значеннях функції втрат. Абсолютно аналогічні результати були для опціонів типу put. Також треба додати, що під час цих експериментів також використовувалися 2 алгоритми масштабування, які ми зазначали вище, у і всіх з них результати були кращими з стандартним, тому для усіх наступних експериментів ми будемо використовувати архітектуру №2 лише з ним.

Одразу розглянемо опціон типу call, а саме рисунок 3.8. З аналізу його

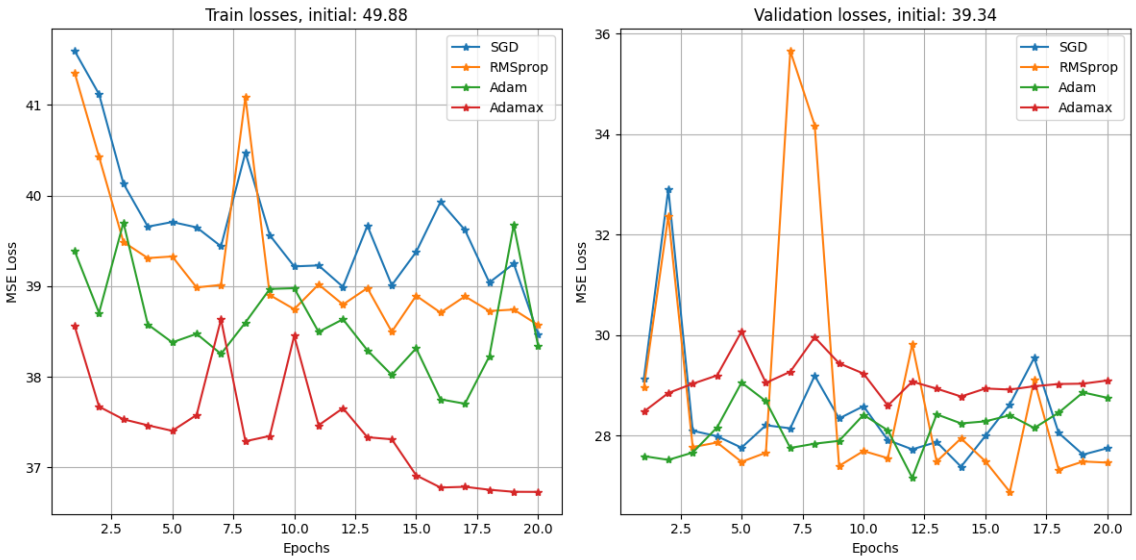


Рисунок 3.8 – Значення функції втрат для архітектури №2 для різних оптимізаторів для опціону типу call

графіків може здатися, що Adamax має найкращий результат, але якщо детальніше розглянути графік справа починаючи з десятої епохи можна побачити, що наш оптимізатор потрапив на певне плато, при якому на графіку зліва функція втрат продовжила спуск до глобального мінімуму, що свідчить про певний зсув у розподілах даних тренувальної та валідаційної вибірки. Також вартим уваги є метод SGD, адже для нього є досить властивим «негладкий» графік функції втрат і як можна побачити з графіку, хоча він і не дійшов до найкращих результатів на графіку зліва, він має майже найкращий результат на графіку справа. Також знаючи, що цей метод зазвичай дає надійні розв'язки у сенсі узагальнення закономірностей даних [46], далі ми будемо використовувати його для фінальної моделі. На рисунку 3.9 зображено порівняльний експеримент для методів оптимізації, але вже для опціону типу put. Одразу розглянувши графік справа можна побачити, що починаючи з 15 епохи всі оптимізатори зайшли на певне плато. Одночасно проаналізувавши обидва графіки можна зробити висновок, що у цьому випадку найоптимальніших оптимізатором буде Adam, адже із усіх він має найкращі показники функції втрат і відсутність великих стрибків у значеннях функції втрат, тому далі ми будемо використовувати його для фінальної моделі.

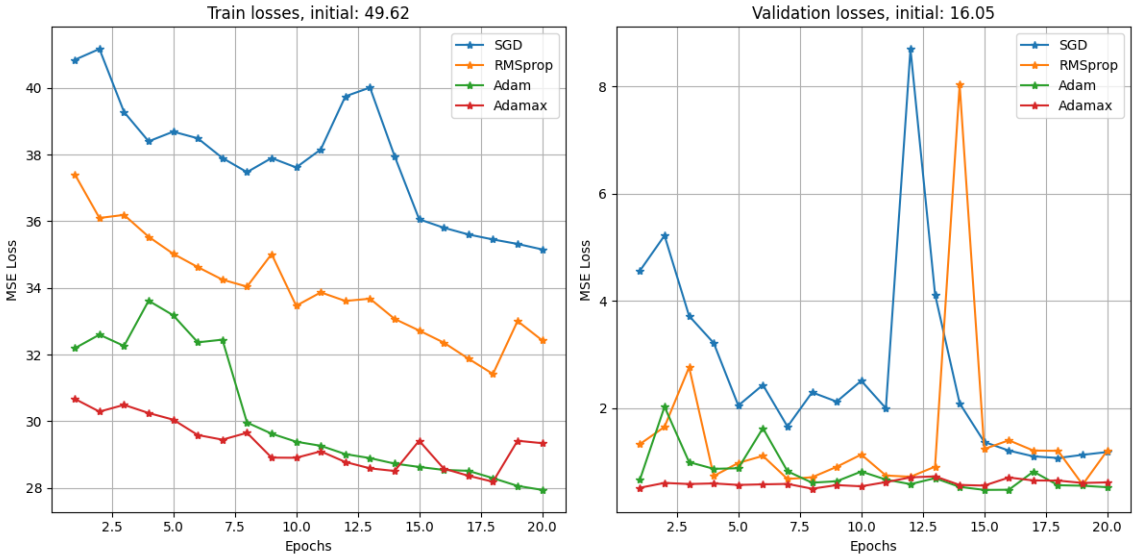


Рисунок 3.9 – Значення функції втрат для архітектури №2 для різних оптимізаторів для опціону типу put

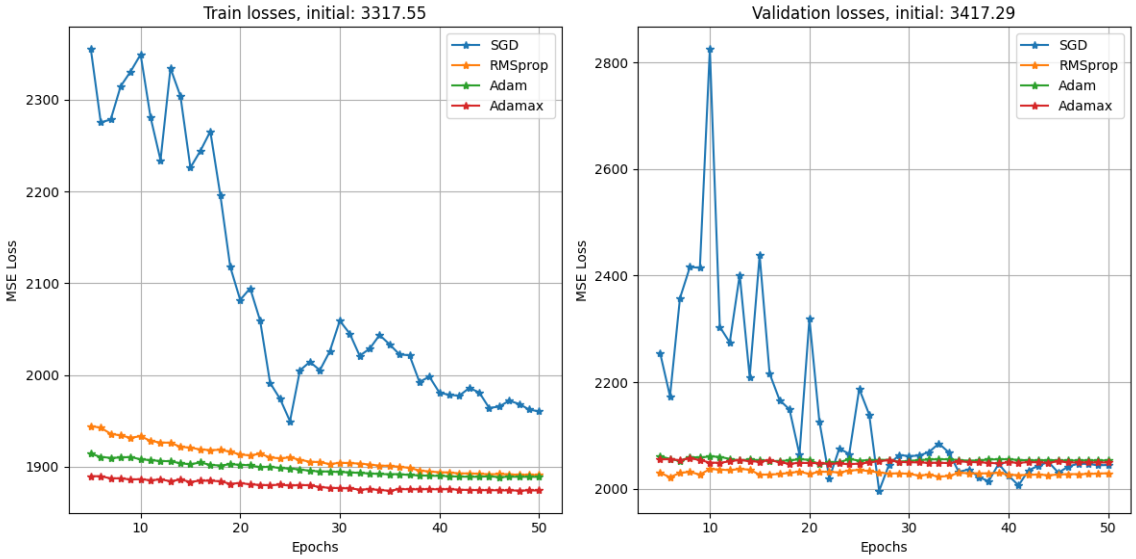


Рисунок 3.10 – Значення функції втрат для архітектури №2 для різних оптимізаторів для опціону типу call для згенерованих даних

Тепер проаналізуємо значення функції втрат для згенерованих даних та визначимо найкращі гіперпараметри моделей для них.

На рисунку 3.10 зображено порівняльний експеримент для методів оптимізації для опціонів call для згенерованих даних.

Як можна побачити, хоча всі оптимізатори, окрім SGD, мають початкове значення функції втрат нижче за нього, але на протязі 50 епох вони майже не змінили свого значення. З цього ми зробимо висновок, що для цього випадку кращим варіантом буде використання SGD.

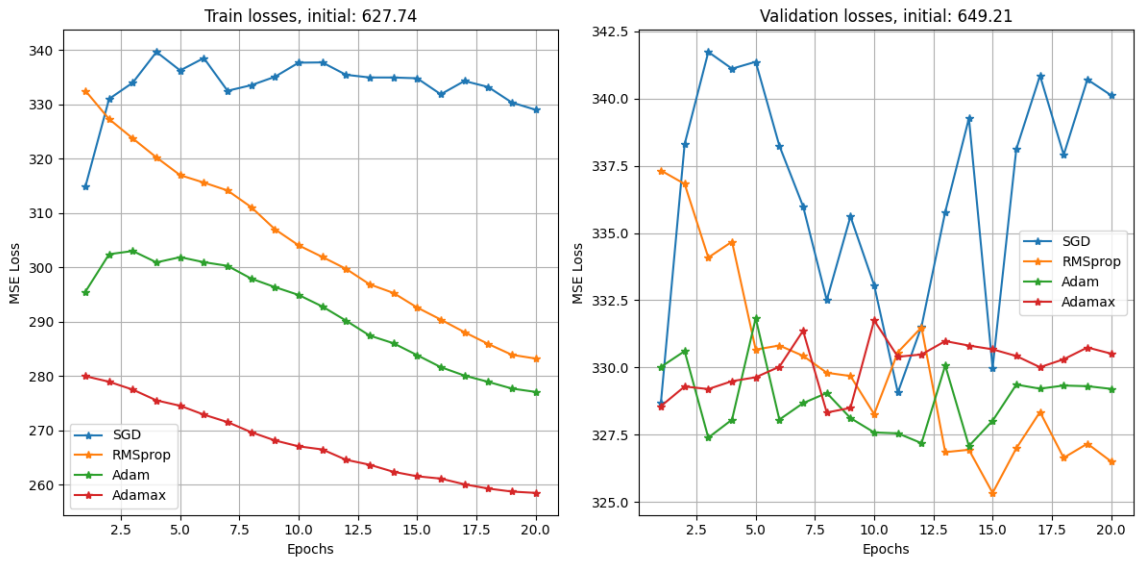


Рисунок 3.11 – Значення функції втрат для архітектури №2 для різних оптимізаторів для опціону типу put для згенерованих даних

На рисунку 3.11 зображено порівняльний експеримент для методів оптимізації для опціонів put для згенерованих даних. З обох графіків можна побачити, що RMSprop має найкращі показники і тому ми будемо використовувати його для відповідного опціону.

Тепер маючи найкращі гіперпараметри для всіх моделей можна перейти до повноцінного навчання та аналізу результатів.

3.5 Навчання та оцінка моделі

Оцінювальним параметром якості моделі є відношення середньоквадратичного відхилення до дисперсії цільового значення у відсотках. Це широковикористовувана метрика для оцінки якості моделей, так як вона дає зрозуміти, який відсоток дисперсії наша модель не може пояснити. Тобто перед початком тренування моделі ми порахуємо середньоквадратичне відхилення як задано формулою 1.11 для $y(t_j)$ та $y(t_{j+1})$ у формулі 1.18 та розділимо це значення на дисперсію їх різниці, яку позначимо VAR. Отримаємо формулу:

$$V_1 = \frac{MSE\left(y(t_j), y(t_{j+1})\right)}{VAR\left(y(t_j) - y(t_{j+1})\right)} \cdot 100\% \tag{3.2}$$

і після навчання моделі обрахуємо теж саме, але вже для отриманних передбачень для наших даних:

$$V_2 = \frac{MSE\left(p(x_j), y(t_{j+1}) - y(t_j)\right)}{VAR\left(y(t_j) - y(t_{j+1})\right)} \cdot 100\% \tag{3.3}$$

І знайшовши різницю 3.3 та 3.2 ми знайдемо, на скільки відсотків передбачення нашої моделі покращило початковий результат методу Рунге-Кутти 4-го порядку відносно дисперсії.

Навчати моделі для call та put опціонів ми будемо 50 епох, проте для call опціону згенерованих даних ми будемо використовувати 100 епох, початкова швидкість навчання $1 \cdot 10^{-5}$, де тренувальна вибірка буде містити 178,573 записи та 44,644 на тест відповідно. Для згенерованих даних тренувальна вибірка містить 8000 записів та 2000 на тест. Також ми будемо використовувати метод ReduceLROnPlateau [26], який буде зменшувати швидкість навчання в 10 разів при «плато» в навчанні — коли метрика якості не покращується на протязі 10 епох. Це дуже зручна та корисна техніка у випадках, коли ми не можемо потрапити у глобальний мінімум через завелику швидкість навчання, і кожного разу «перестрибуємо» його. Загальна кількість параметрів, які навчаються — 83,725.

Як було зазначено у розділі 3.3, для опціонів типу call ми будемо використовувати оптимізатор SGD, для опціонів типу put — Adam та RMSprop для справжніх та згенерованих даних відповідно.

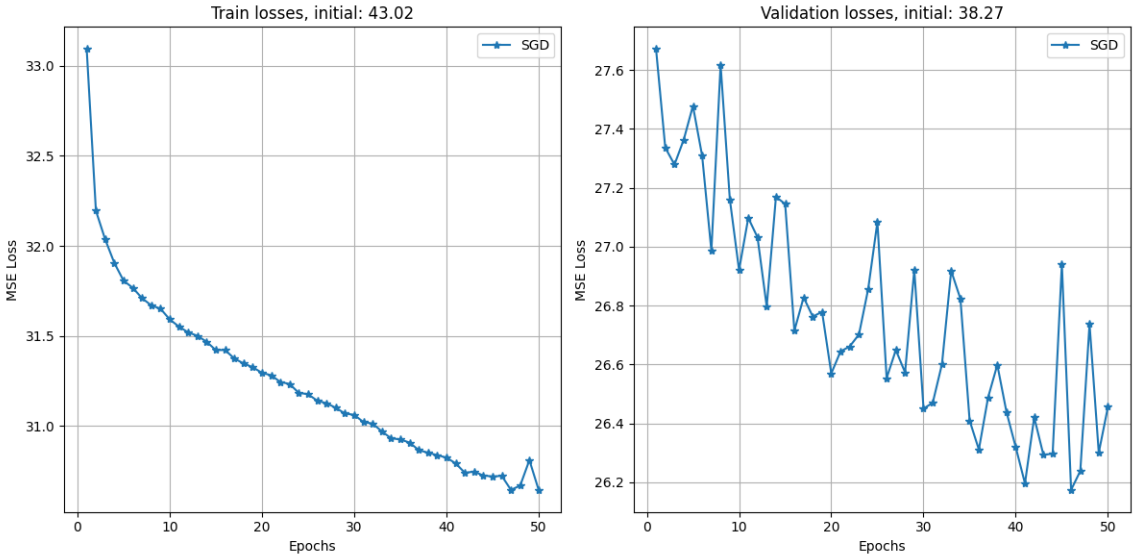


Рисунок 3.12 – Значення функції втрат під час навчання моделі для опціону типу call справжніх даних

Номер епохи	Train loss	Test loss
0	43.02	38.26
10	31.59	26.92
20	31.29	26.57
30	31.06	26.45
40	30.82	26.31
46	**30.72**	**26.17**
50	30.64	26.46

Таблиця 3.7 – Результати навчання моделі для опціонів типу call справжніх даних

На рисунку 3.12 зображено графіки функції втрат під час навчання для опціону call справжніх даних, а в таблиці 3.7 відповідні значення функції втрат, а саме середньоквадратичне відхилення під час навчання.

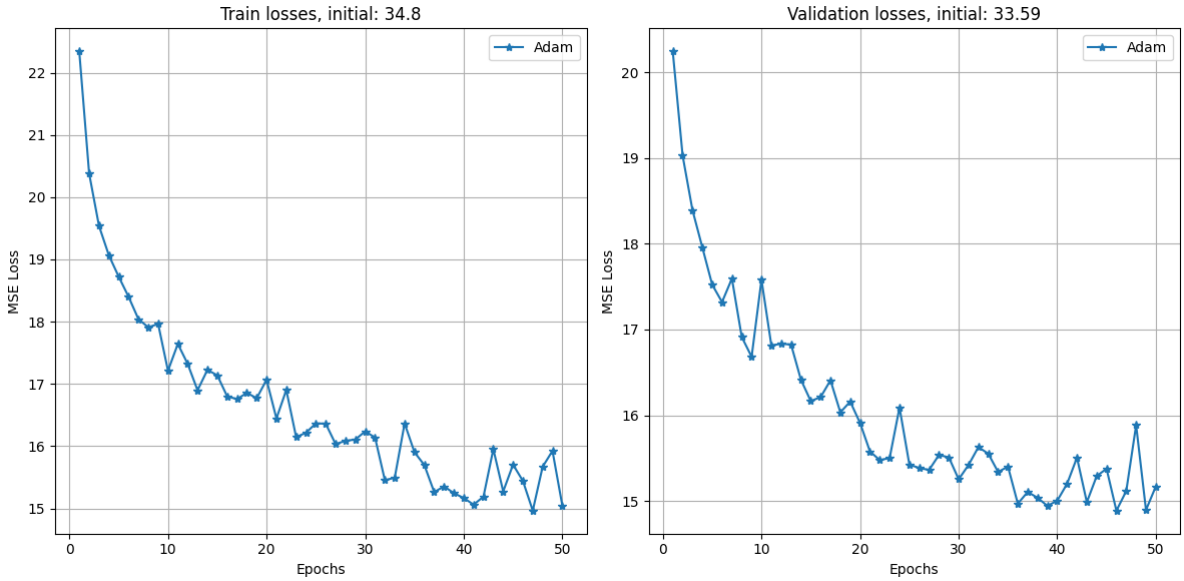


Рисунок 3.13 – Значення функції втрат під час навчання моделі для опціону типу put справжніх даних

Номер епохи	Train loss	Test loss
0	34.80	33.59
10	17.21	17.58
20	17.06	15.91
30	16.23	15.25
40	15.17	15.00
47	**14.96**	**15.11**
50	15.05	15.17

Таблиця 3.8 – Результати навчання моделі для опціонів типу put справжніх даних

На рисунку 3.13 зображено графіки функції втрат під час навчання для опціону put справжніх даних, а в таблиці 3.8 відповідні значення функції втрат.

Як видно з аналізу, результати навчання обох моделей після 50 епох не є оптимальними. Тому в якості остаточних результатів навчання для моделей ми обрали ваги після 46-ї епохи для опціонів типу call та після 47-ї епохи для опціонів типу put для реальних даних.

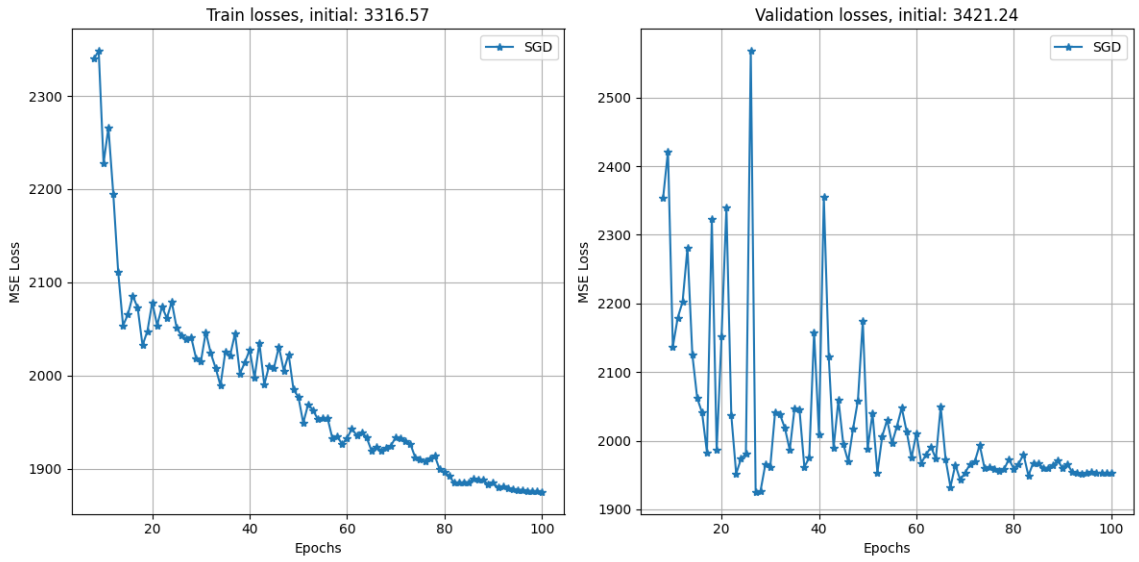


Рисунок 3.14 – Значення функції втрат під час навчання моделі для опціону типу call згенерованих даних

На рисунку 3.14 зображено графіки функції втрат під час навчання для опціону call згенерованих даних, а в таблиці 3.9 відповідні значення функції втрат.

На рисунку 3.15 зображено графіки функції втрат під час навчання для опціону put згенерованих даних, а в таблиці 3.10 відповідні значення функції втрат.

Як видно з аналізу, результати навчання обох моделей на останніх етапах не є оптимальними. Тому, за аналогією з моделями для реальних даних, ми вважаємо остаточними результатами ваги моделей після 67-ї епохи для опціонів типу call та після 41-ї епохи для опціонів типу put для згенерованих даних.

Маючи чотири моделі та оптимальні значення функції втрат після навчання, перейдемо до розрахунку оцінки якості моделей.

Номер епохи	Train loss	Test loss
0	3277.34	3578.15
10	2227.57	2136.91
20	2078.34	2151.39
30	2015.38	1961.68
40	2027.31	2009.19
50	1976.92	1988.56
60	1932.93	2010.51
67	**1821.37**	**2001.89**
70	1933.55	1953.35
80	1895.99	1959.28
90	1884.73	1960.19
100	1874.92	1952.69

Таблиця 3.9 – Результати навчання моделі для опціонів типу call згенерованих даних

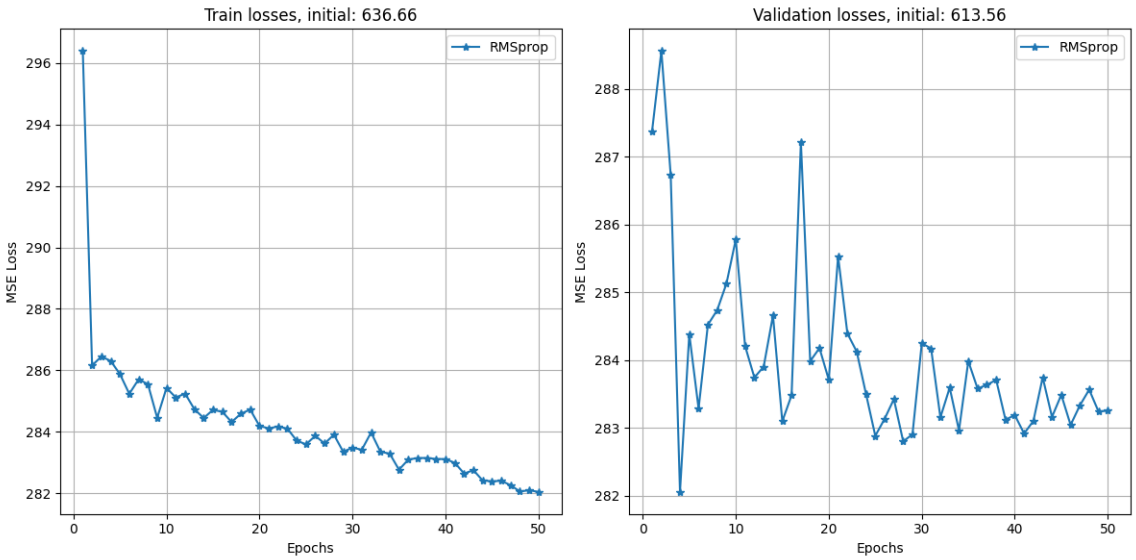


Рисунок 3.15 – Значення функції втрат під час навчання моделі для опціону типу put згенерованих даних

Номер епохи	Train loss	Test loss
0	636.65	613.56
10	285.42	285.78
20	284.19	283.71
30	283.48	284.24
40	283.11	283.18
41	**283.00**	**282.91**
50	282.04	283.25

Таблиця 3.10 – Результати навчання моделі для опціонів типу put згенерованих даних

Тепер, щоб краще зрозуміти, наскільки наш результат є вагомим відносно початкового, розглянемо таблицю 3.11, де представленні відношення середньоквадратичного відхилення та дисперсії до і після навчання. Як можна побачити:

1) для справжніх даних для опціону типу call ми уточнили початкове значення ціни опціону, знайдене чисельним методом, на 29.76% та 32.75% для тренувальної та тестувальної вибірки відповідно, а для опціону типу put — на 61.86% та 59.71%.

2) для згенерованих даних для опціону типу call ми уточнили початкове значення ціни опціону на 128.89% та 118.44% для тренувальної та тестувальної вибірки відповідно, а для опціону типу put — на 124.37% та 116.97%.

Тип моделі	До навчання		Після навчання	
	Train	Test	Train	Test
Call real	103.45%	104.18%	73.69%	71.43%
Call generated	225.41%	218.24%	96.52%	99.8%
Put real	108.54%	108.57%	46.68%	48.86%
Put generated	223.67%	217.19%	99.30%	100.22%

Таблиця 3.11 – Відношення середньоквадратичного відхилення до дисперсії до і після навчання для обох опціонів

Варто зазначити, що хоча з першого погляду результат моделей для згенерованих даних є кращим за результат моделей для справжніх даних, варто розглянути детальніше самі значення середньоквадратичного відхилення у таблицях 3.9 та 3.10 та порівняти їх із значеннями у таблицях 3.7 та 3.8. Така різниця пов'язана з апроксимацією чисельного розв'язку методом Бренана-Шварца [40], який ми використовували для знаходження ціни опціону згенерованих даних і вважали за справжнє значення. Як можна було побачити на рисунку 3.5, розподіл цих цін має суттєві відмінності від розподілу цін опціонів справжніх даних та значень, які ми отримали з методу Рунге-Кутти 4-го порядку, внаслідок чого наше початкове середньоквадратичне відхилення для згенерованих даних суттєво відрізняється від початкового для справжніх.

Висновки до розділу 3

У цьому розділі було проведено статистичний аналіз і передобробку справжніх даних і генерацію синтетичних даних та їх порівняння зі справжніми за допомогою статистичного тесту. Ми проаналізували чотири популярні методи оптимізації та два методи масштабування для двох архітектур нейронних мереж. Отримано, що архітектура №2 є найкращою у всіх випадках для нашої задачі, а метод масштабування StandardScaler продемонстрував найкращі результати для всіх експериментів, але не вдалося визначити єдиний метод оптимізації, який би забезпечував найкращі результати для всіх моделей. Також було встановлено, що найоптимальніші рішення для наших моделей досягаються не на останніх епохах тренування. Як наслідок, тренування наших моделей дало результат, що покращує початкові значення чисельного методу на 88% для опціонів типу put та на 76% для опціонів типу call. Це свідчить про те, що метод нейронних диференціальних рівнянь є хорошим доповненням до класичних чисельних методів розв'язання диференціальних рівнянь в частинних похідних, зберігаючи за собою їх сильні сторони, та гарним аналогом до методів, які використовують лише машинне навчання.

ВИСНОВКИ

У ході даної роботи було розглянуто метод нейронних диференціальних рівнянь для прогнозування ціни на американські опціони типу put та call. Проведено огляд трьох класів можливих підходів до вирішення цієї задачі та аналіз робіт інших авторів для кожного з них. Розглянуто різні архітектури нейронних мереж та методи оптимізації для цієї задачі та їх порівняння.

Нашою метою в роботі було проаналізувати, чи зможе метод нейронних диференціальних рівнянь уточнювати розв'язки чисельних методів для рівняння Блека-Шоулза. Це було мотивовано тим, що для отримання теоретичної ціни американського опціону ми використовували аналітичні розв'язки рівняння для європейських опціонів, які лише за певних обставин дорівнюють американським [25]. Для цього ми зібрали дані з фондового ринку за контрактами на американські опціони, провели попередню обробку та статистичний аналіз, згенерували синтетичні дані та порівняли їх розподіли за допомогою статистичного тесту Колмогорова-Смірнова, обрали архітектури нейронних мереж прямого поширення, їх гіперпараметри, методи масштабування та оптимізації.

У результаті ми виявили, що для нашої задачі метод масштабування StandardScaler є кращим за метод MinMaxScaler, показавши кращі значення функції втрат для обраних нами методів оптимізації та архітектур нейронних мереж.

Модифікована версія моделі, запропонованої у роботі автора [34], показує гіршу стабільність при навчанні порівняно з адаптованою версією моделі, представленою у роботі автора [14], в усіх проведених нами експериментах.

Єдиного методу оптимізації, який би забезпечував найкращі результати для наших моделей, ми не виявили, але SGD у двох із чотирьох моделей показав себе краще за інші використані нами методи, які «застрягали» в локальних мінімумах та не могли вийти з них.

За результатами дослідження було виявлено, що метод нейронних

диференціальних рівнянь є гарним доповненням до класичного методу Рунге-Кутта 4-го порядку. Зокрема, моделі для опціону типу put мають покращення в уточненні чисельного розв’язку на 88%, а моделі для опціону типу call на 76% за метрикою відношення середньоквадратичного відхилення до дисперсії. Також варто зазначити, що під час дослідження для тренування моделей використовувалася невелика кількість епох. Це залишає можливість для дослідників розглянути цю тематику, вдосконалити запропонований підхід до генерації синтетичних даних та використати більш потужні обчислювальні ресурси для тренування моделей.

ЛІТЕРАТУРА

1. A DISCUSSION ON EULER METHOD: A REVIEW / B. N. Biswas [та ін.] // Electronic Journal of Mathematical Analysis and Applications. — 2013. — Черв. — Т. 1. — С. 294—317.
2. *Akshit J. Dhruv R. P.*, *Doshi N.* Python: The Most Advanced Programming Language for Computer Science Applications. — 2022.
3. An Empirical Study on How the Developers Discussed about Pandas Topics / S. K. S. Joy [та ін.]. — 2023. — arXiv: 2210.03519 [cs.SE].
4. *Anwar M. urul*, *Sazzad Andallah L.* A Study on Numerical Solution of Black-Scholes Model // Journal of Mathematical Finance. — 2018. — Т. 372—381, № 8.
5. *Bakshi G.*, *Cao C.*, *Chen Z.* Empirical Performance of Alternative Option Pricing Models // The Journal of Finance. — 1997. — Т. 52, № 5. — С. 2003—2049. — DOI: https://doi.org/10.1111/j.1540-6261.1997.tb02749.x. — eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1997.tb02749.x. — URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1997.tb02749.x.
6. *Bhuyan R. M.*, *Chaudhury M.* Trading on the Information Content of Open Interest: Evidence from the Us Equity Options Market // McGill Finance Research Centre Working Paper. — 2001.
7. *Black F.*, *Scholes M.* The Pricing of Options and Corporate Liabilities // Journal of Political Economy. — 1973. — Т. 81, № 3. — С. 637—654. — ISSN 00223808, 1537534X. — URL: http : / / www . jstor . org / stable / 1831029 (дата зверн. 20.04.2024).

8. Black–Scholes Option Pricing Using Machine Learning / S. Sood [та ін.] // Proceedings of International Conference on Data Science and Applications / за ред. M. Saraswat [та ін.]. — Singapore : Springer Nature Singapore, 2023. — С. 481—493. — ISBN 978-981-19-6631-6.

9. *Chaudhury M.* Option Bid-Ask Spread and Liquidity // The Journal of Trading. — 2011. — Серп. — Т. 10. — DOI: 10.2139/ssrn.1769126.

10. Comparing Numerical Methods for Ordinary Differential Equations / T. E. Hull [та ін.] // SIAM Journal on Numerical Analysis. — 1972. — Т. 9, № 4. — С. 603—637. — ISSN 00361429. — URL: http://www.jstor.org/stable/2156215 (дата зверн. 23.04.2024).

11. *Damodaran A.* What is the riskfree rate? A Search for the Basic Building Block // Journal of New Finance. — 2020. — Т. 1, № 3.

12. Deep Residual Learning for Image Recognition / K. He [та ін.]. — 2015. — arXiv: 1512.03385 [cs.CV].

13. *Deshpande M.* Perceptrons: The First Neural Networks for Machine Learning. — 2023.

14. *Dhiman A.*, *Hu Y.* Physics Informed Neural Network for Option Pricing. — 2023. — arXiv: 2312.06711 [q-fin.PR].

15. *Dubey S. R.*, *Singh S. K.*, *Chaudhuri B. B.* Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark. — 2022. — arXiv: 2109.14545 [cs.LG].

16. *H. M.*, *Saidu I.*, *Waziri M.* A Simplified Derivation and Analysis of Fourth Order Runge Kutta Method // International Journal of Computer Applications. — 2010. — Листоп. — Т. 9. — DOI: 10.5120/1402-1891.

17. *Han H.*, *Wu X.* A Fast Numerical Method for the Black-Scholes Equation of American Options // SIAM Journal on Numerical Analysis. — 2004. — Т. 41, № 6. — С. 2081—2095. — ISSN 00361429. — URL: http://www.jstor.org/stable/4101191 (дата зверн. 20.04.2024).

18. *Hogg R. V.*, *McKean J. W.*, *Craig A. T.* Introduction to mathematical statistics. — Pearson, 2019.

19. *Hornik K.*, *Stinchcombe M.*, *White H.* Multilayer feedforward networks are universal approximators // Neural Networks. — 1989. — Т. 2, № 5. — С. 359—366. — ISSN 0893-6080. — DOI: https : / / doi . org / 10 . 1016 / 0893 - 6080(89) 90020 - 8. — URL: https://www.sciencedirect.com/science/article/pii/0893608089900208.

20. *Ioffe S.*, *Szegedy C.* Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. — 2015. — arXiv: 1502.03167 [cs.LG].

21. *Ivaşcu C.-F.* Option pricing using Machine Learning // Expert Systems with Applications. — 2021. — Т. 163. — С. 113799. — ISSN 0957-4174. — DOI: https://doi.org/10.1016/j.eswa.2020.113799. — URL: https://www.sciencedirect.com/science/article/pii/S0957417420306187.

22. *JayeshBapuAhire*. The Artificial Neural Networks handbook: Part 1. — 2018.

23. *Jeong D.*, *Kim J.*, *Wee I.-S.* An accurate and efficient numerical method for Black-Scholes equations // Commun. Korean Math. Soc. — 2009. — Жовт. — Т. 24. — С. 617—628. — DOI: 10.4134/CKMS.2009.24.4.617.

24. *Joanes D. N.*, *Gill C. A.* Comparing Measures of Sample Skewness and Kurtosis // Journal of the Royal Statistical Society. Series D (The Statistician). — 1998. — Т. 47, № 1. — С. 183—189. — ISSN 00390526, 14679884. — URL: http://www.jstor.org/stable/2988433 (дата зверн. 28.04.2024).

25. *Johnson H. E.* An Analytic Approximation for the American Put Price // The Journal of Financial and Quantitative Analysis. — 1983. — Т. 18, № 1. — С. 141—148. — ISSN 00221090, 17566916. — URL: http://www.jstor.org/stable/2330809 (дата зверн. 21.04.2024).

26. *Al-Kababji A.*, *Bensaali F.*, *Dakua S. P.* Scheduling Techniques for Liver Segmentation: ReduceLROnPlateau Vs OneCycleLR. — 2022. — arXiv: 2202.06373 [cs.CV].

27. *Karimboyevich S. E.*, *Nematullayevich A. O.* Single Layer Artificial Neural Network: Perceptron // European Multidisciplinary Journal of Modern Science. — 2022. — Квіт. — Т. 5. — С. 230—238. — URL: https://emjms.academicjournal.io/index.php/emjms/article/view/253.

28. *Kidger P.* On Neural Differential Equations. — 2022. — arXiv: 2202.02435 [cs.LG].

29. *Kodwani D.* An Analytical Study of Option Greeks on Derivative Markets in India //. — 01.2005. — С. 17—33. — ISBN 978-1-349-54286-4. — DOI: 10.1057/9780230596368_3.

30. *Larsson J.* Optimization of option pricing : - Variance reduction and low-discrepancy techniques. — 2020.

31. *LeVeque R. J.* Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems. // Society for Industrial and Applied Mathematics. — 2007.

32. *Li Z.* PPINN: Parareal Physics-Informed Neural Network for time-dependent PDEs - Scientific Figure on ResearchGate.

33. Loss Functions and Metrics in Deep Learning / J. Terven [та ін.]. — 2023. — arXiv: 2307.02694 [cs.LG].

34. *Louskos A.* Physics-Informed Neural Networks and Option Pricing. — 2021. — URL: https://math.dartmouth.edu/theses/undergrad/2021/Louskos-thesis.pdf.

35. matplotlib – A Portable Python Plotting Package / P. Barrett [та ін.] //. — 12.2005.

36. *Mazumder S.* Chapter 5 - Treatment of the Time Derivative (Parabolic and Hyperbolic PDEs) // Numerical Methods for Partial Differential Equations / за ред. S. Mazumder. — Academic Press, 2016. — С. 219—275. — ISBN 978-0-12-849894-1. — DOI: https://doi.org/10.1016/B978-0-12-849894-1.00005-6. — URL: https://www.sciencedirect.com/science/article/pii/B9780128498941000056.

37. Neural Ordinary Differential Equations / R. T. Q. Chen [та ін.]. — 2019. — arXiv: 1806.07366 [cs.LG].

38. NeuralPDE: Automating Physics-Informed Neural Networks (PINNs) with Error Approximations / K. Zubov [та ін.]. — 2021. — arXiv: 2107.09443 [cs.MS].

39. *Petrov V.*, *Golub A.*, *Olsen R.* Instantaneous Volatility Seasonality of High-Frequency Markets in Directional-Change Intrinsic Time // Journal of Risk and Financial Management. — 2019. — Т. 12, № 2. — ISSN 1911-8074. — DOI: 10.3390/jrfm12020054. — URL: https://www.mdpi.com/1911-8074/12/2/54.

40. Pricing of American options, using the Brennan–Schwartz algorithm based on finite elements / S. Madi [та ін.] // Applied Mathematics and Computation. — 2018. — Т. 339. — С. 846—852. — ISSN 0096-3003. — DOI: https://doi.org/10.1016/j.amc.2018.06.028. — URL: https://www.sciencedirect.com/science/article/pii/S0096300318305174.

41. PyTorch: An Imperative Style, High-Performance Deep Learning Library / A. Paszke [та ін.]. — 2019. — arXiv: 1912.01703 [cs.LG].

42. *Raissi M.*, *Karniadakis G. E.* Hidden physics models: Machine learning of nonlinear partial differential equations // Journal of Computational Physics. — 2018. — Бер. — Т. 357. — С. 125—141. — ISSN 0021-9991. — DOI: 10.1016/j.jcp.2017.11.039. — URL: http://dx.doi.org/10.1016/j.jcp.2017.11.039.

43. *Reilly F.*, *Brown K.* Investment Analysis and Portfolio Management. — South-Western/Thomson Learning, 2003. — (Dryden Press series in finance). — ISBN 9780324171730. — URL: https://books.google.com.ua/books?id=LYUgAQAAIAAJ.

44. *Schmidhuber J.* Deep learning in neural networks: An overview // Neural Networks. — 2015. — Січ. — Т. 61. — С. 85—117. — ISSN 0893-6080. — DOI: 10.1016/j.neunet.2014.09.003. — URL: http://dx.doi.org/10.1016/j.neunet.2014.09.003.

45. Scikit-learn: Machine Learning in Python / F. Pedregosa [та ін.]. — 2018. — arXiv: 1201.0490 [cs.LG].

46. *Shah V.*, *Wu X.*, *Sanghavi S.* Choosing the Sample with Lowest Loss makes SGD Robust. — 2020. — arXiv: 2001.03316 [stat.ML].

47. *Tennenholtz G.*, *Zahavy T.*, *Mannor S.* Train on Validation: Squeezing the Data Lemon. — 2018. — arXiv: 1802.05846 [stat.ML].

48. Universal Differential Equations for Scientific Machine Learning / C. Rackauckas [та ін.]. — 2021. — arXiv: 2001.04385 [cs.LG].

49. Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study / B. M. Boscoe [та ін.]. — 2018. — arXiv: 1804.05492 [cs.DL].

50. *Verma R.* A Novel Approach to the Chain Rule. — 2015. — Жовт.

51. *Vidic I.* Numerical methods for option pricing. — 2012.

52. *Watada J.* Kolmogorov-Smirnov Two Sample Test with Continuous Fuzzy Data //. — 01.2010. — С. 175—186.

53. *Wilmott P.*, *Dewynne J.*, *Howison S.* Option Pricing: Mathematical Models and Computation. — Oxford Financial, 1998. — URL: https://books.google.com.ua/books?id=cGuGcgAACAAJ.

54. *Zienkiewicz O. C.*, *Taylor R. L.*, *Zhu. J. Z.* The Finite Element Method: Its Basis and Fundamentals. Seventh edition. — Oxford, UK: Butterworth-Heinemann, 2013.