

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «**РОЗРОБКА ПІДСИСТЕМИ
ФІНАНСОВОГО СЕНТИМЕНТ АНАЛІЗУ**»

Виконав: студент 2 курсу, групи 8.1211-з

спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)

освітньої програми інженерія програмного забезпечення
(назва освітньої програми)

С.Ю. Богуцький

(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії,
доцент, к.ф.-м.н. Кудін О.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент завідувач кафедри комп'ютерних наук,
професор, д.т.н. Чопоров С.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний
Кафедра програмної інженерії
Рівень вищої освіти магістр
Спеціальність 121 інженерія програмного забезпечення
Освітня програма інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри програмної
інженерії, к.ф.-м.н., доцент

Лісняк А.О.

(підпис)

« _____ » _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Богуцькому Станіславу Юрійовичу

(прізвище, ім'я та по-батькові)

1. Тема роботи (проєкту) Розробка підсистеми фінансового сентимент аналізу

керівник роботи (проєкту) Кудін Олексій Володимирович, к.ф.-м.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 10 » травня 2022 року № 514-с

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік питань до розробки.

3. Перелік літератури.

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

1. Огляд проблеми, постановка задачі.

2. Основні теоретичні відомості.

3. Розробка нейронної мережі.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Презентація за темою докладу

6. Консультанти розділів роботи

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

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану виконання кваліфікаційної роботи магістра.		
2.	Збір вихідних даних та аналіз предметної області.		
3.	Обробка методичних та теоретичних джерел.		
4.	Специфікація вимог до системи. Робота над першим розділом.		
5.	Проектування системи. Робота над другим розділом.		
6.	Реалізація та тестування системи. Робота над третім розділом.		
7.	Розробка керівництва користувача. Робота над додатками.		
8.	Оформлення та нормоконтроль кваліфікаційної роботи.		
9.	Захист кваліфікаційної роботи магістра.		

Студент _____
(підпис)

С.Ю. Богуцький _____
(ініціали та прізвище)

Керівник роботи _____
(підпис)

О.В. Кудін _____
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

А.В. Столярова _____
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка фінансового сентимент аналізу»: 53 с., 17 рис., 25 джерел.

БІБЛІОТЕКА, ДАТАСЕТ, ЕПОХА, МАШИННЕ НАВЧАННЯ, НЕЙРОНА МЕРЕЖА, ОБРОБКА ДАНИХ, ПРОГНОЗУВАННЯ, ШТУЧНИЙ ІНТЕЛЕКТ.

Об'єкт дослідження – нейронні мережі на основі рекурентної архітектури, їх можливості та перспективи у сфері фінансового прогнозування.

Мета роботи – підвищення якісних характеристик роботи рекурентних нейромереж для розв'язання задач прогнозування фінансових ринків.

Метод дослідження – порівняння, аналіз, описовий, структурний.

Актуальність кваліфікаційної роботи обумовлена високим розвитком нейромережевих технологій та штучного інтелекту, потреба в зростанні якості прогнозів фінансових ринків, незавершеність формування цілісного уявлення щодо прогнозування цін на фінансових ринках.

SUMMARY

Master's Qualification Paper «Development of the Financial Sentiment Analysis Subsystem»: 53 pages, 17 figures, 25 references.

LIBRARY, DATASET, ERA, MACHINE LEARNING, NEURAL NETWORK, DATA PROCESSING, FORECASTING, ARTIFICIAL INTELLIGENCE.

Object of research – neural networks based on recurrent architecture, their capabilities and prospects in the field of financial forecasting.

Subject of research – models and methods of using neural networks for forecasting tasks, ways of improving existing methods and forecasting systems.

The purpose of the master's dissertation is to increase the qualitative characteristics of the work of recurrent neural networks for solving tasks of forecasting of financial markets.

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат	4
Summary	5
Вступ.....	8
1 Огляд та характеристика методів	10
1.1 Прогнозування економічних процесів.....	10
1.2 Загальні методи прогнозування.....	14
1.3 Огляд методів	15
1.4 Методи на основі часових рядів.....	16
1.5 Каузальні методи	19
1.6 Структурні методи.....	20
1.7 Проблема зникаючого градієнту	21
1.8 Огляд програмних засобів, що реалізують нейромереві алгоритми для вирішення задач прогнозування.....	22
1.9 Висновки до розділу 1	26
2 Загальні поняття нейронних мереж для розробки.....	28
2.1 Основні поняття нейронних мереж.....	28
2.2 Методи навчання нейронних мереж	30
2.3 Згорткові нейронні мережі.....	32
2.4 Нейронні мережі прямого поширення.....	34
2.5 Мережа радіальних базисних функцій	35
2.6 Рекурсивні нейронні мережі	36
2.7 Рекурентні нейронні мережі	37
2.8 Довга короткострокова пам'ять (LSTM)	37
2.9 LSTM-архітектури для задачі прогнозування часових рядів.....	39
2.10 Висновки до розділу 2	39
3 Проєктування та розробка нейронної мережі	41

3.1 Налаштування середовища для розробки	41
3.2 Підключення необхідних бібліотек для роботи	42
3.3 Загальний опис розробки нейронної мережі.....	43
Висновки	50
Перелік посилань.....	51

ВСТУП

Актуальною малодослідженою проблемою економіки України сьогодні є ціноутворення на основні фінансові інструменти. Фінансові ринки вимагають від учасників високого рівня фахової підготовки. Фінансисти корпорацій, аналітики, торговці фінансових інститутів та біржові спекулянти щоденно приймають багато рішень про купівлю-продаж різних фінансових і матеріальних активів. Для зменшення ризику таких операцій та отримання очікуваних прибутків від своїх вкладень кожен із них повинен знати і аналізувати цілий ряд факторів, що впливають на ринкові ціни та курси і породжують тенденції зростання чи зниження.

В цілому в країнах розвинутого ринку поширені три методи аналізу ринку, а саме: фундаментальний, технічний, інтуїтивний.

Фундаментальний аналіз вивчає рух цін під впливом макроекономічних чинників. Технічний аналіз можна визначити як метод прогнозування динаміки цін, заснований на математичних, а не на економічних теоріях.

За умови незмінності попиту та пропозиції, коли ціни не коливаються, ринкова ситуація характеризується ціновою стабільністю. В останні 10-15 років на світових ринках цінова стабільність спостерігається надзвичайно рідко. Оскільки події на зарубіжних ринках впливають на становлення ринків в Україні, їх учасникам (продавцям, покупцям і посередникам) необхідно сприйняти цінову нестабільність як реальний факт та навчитися управляти відповідними ризиками.

Початок трансформації бухгалтерського обліку в рамках перебудови економіки на ринкові рейки (початок 1990-х років) знову повернув до життя такий найважливіший елемент аналітичної роботи, як фінансовий аналіз. У його основі – аналіз керування фінансовими ресурсами суб'єкта господарювання як основного і пріоритетного виду ресурсів. Щоби забезпечити життєспроможність підприємства в сучасних умовах,

управлінському персоналові необхідно насамперед уміти реально оцінювати фінансовий стан свого підприємства, його існуючих та потенційних контрагентів. Для цього потрібно:

- 1) володіти методикою оцінки фінансового стану підприємства;
- 2) користуватися відповідним інформаційним забезпеченням;
- 3) мати кваліфікований персонал, спроможний реалізувати дану методику на практиці.

Отже, фінансовий аналіз – це метод оцінки і прогнозування фінансового стану підприємства на основі його бухгалтерської звітності. Такий аналіз може виконувати як управлінський персонал даного підприємства, так і будь-який зовнішній аналітик, оскільки згаданий аналіз базований здебільшого на загальнодоступній інформації. Проте прийнято виділяти два види фінансового аналізу: внутрішній і зовнішній.

1 ОГЛЯД ТА ХАРАКТЕРИСТИКА МЕТОДІВ

1.1 Прогнозування економічних процесів

Інтерес наукового суспільства і бізнесу до нейронних мереж зростає з кожним роком. Все більше і більше з'являється прикладів успішного застосування штучного інтелекту у найрізноманітніших галузях діяльності людини, значно зростає кількість підприємств що впроваджують нейронні мережі у свою операційну діяльність. Розв'язання задач управління, прогнозування а також класифікації все частіше покладають на штучний інтелект у багатьох галузях діяльності людини. Окрім цього, нейронні мережі знайшли широке застосування в прогнозуванні фінансових ринків. Нова і перспективна технологія швидко привернула увагу дослідників. Спочатку нейронні мережі були застосовані у прогнозуванні попиту на певні товари, потім у прогнозуванні ризиків видачі кредитів. Були розроблені системи прийняття фінансових рішень, засновані на штучному інтелекті. Варто зауважити, що більшість економічних процесів мають нелінійний характер. Нейронні мережі мають можливість працювати не тільки з нелінійними процесами, а й з зашумленими даними. Стрімкий розвиток нейронних мереж водночас зі стрімким розвитком міжнародних біржових інституцій зумовлює потребу ринку у прогнозуванні фінансових активів.

Результативність застосування традиційних методів прогнозування фінансових активів, наприклад акцій, облігацій, та валюти, які вільно продаються і купуються на біржах, можна назвати недостатньою для потреб сучасного ринку. Це пов'язано із тим, що інвестиції на фондовому ринку тісно пов'язані із Інтернетом і залежні від інформаційного середовища. Для підвищення точності прогнозування доцільно застосувати таку модель, що не тільки базується на кореляціях факторів та особливостях часового ряду, а й тісно пов'язана з декількома джерелами даних. Сучасні системи

прогнозування не враховують комплексно кількісні та якісні фактори, що впливають на зміну курсів акцій або враховують у векторному вигляді, який обмежує можливості представлення вхідних даних до нейромережі. Такі системи прогнозування зазвичай мають точність 53-58 % вірного прогнозу. Цієї точності замало для того, щоб використовувати такі системи як повноцінний інструмент для економічного аналізу та прогнозування. Отже, метою дослідження є підвищення точності прогнозування тренду курсів акцій за допомогою рекурентних нейромереж.

Проблемою залишається дослідження спільного впливу декількох джерел даних на курс акцій. У попередніх дослідженнях наведено два підходи до прогнозування курсів акцій за допомогою рекурентних нейромереж: прогнозування часового без урахування зовнішніх факторів та прогнозування з урахуванням вектору зовнішніх факторів.

Прогнозування без урахування зовнішніх факторів являє собою звичайне прогнозування часового ряду за допомогою нейромережі LSTM.

Переваги прогнозування без урахування зовнішніх факторів:

- швидка реалізація через відсутність модулів аналізу подій, настроїв інвесторів та кількісного аналізу;
- швидше навчання порівняно з іншими методами;
- відсутність потреби збирати величезну допоміжну кількість даних.

Все що треба для прогнозування – дані часового ряду (курс відкриття, закриття, максимальний, мінімальний, медіана, волатильність).

Недоліки прогнозування без урахування зовнішніх факторів:

- низька точність прогнозування (зазвичай не більше 54-56%);
- відсутність гнучкої реакції на фінансові події, що нівелює корись застосування системи у сучасному фінансовому ринку, який керується подіями.

Прогнозування із урахуванням вектору зовнішніх факторів поєднує набір факторів у вектор який подається на вхід нейромережі.

Переваги прогнозування із урахуванням вектору зовнішніх факторів:

- система враховує зовнішні фактори і гнучко реагує на події та фінансові новини;
- більша точність прогнозування порівняно з попереднім методом (57-59 %).

Недоліки прогнозування із урахуванням вектору зовнішніх факторів:

- вектор входів значно розростається;
- втрачаються контекстні зв'язки між джерелами інформації;
- необхідність збирати величезну кількість даних про зовнішні фактори, цінність яких частково втрачається на етапі формування єдиного вектору.

Отже, було проведено багато наукових досліджень на цю тему в останні роки. Загальна ідея минулих досліджень полягала в тому, щоб поєднати набір властивостей декількох джерел інформації у єдиний вектор ознак. Однак це призводило до того що зростали розміри вектора. Окрім того поступово слабнуть або взагалі втрачаються контекстні зв'язки між джерелами інформації. Інші ж дослідники, як наведено у публікації [2], ігнорували зв'язок між джерелами інформації і спиралися лише на аналіз часового ряду. Таке дослідження давало позитивні результати, однак дослідники, що урахували їх у роботі [3] отримували більш точне прогнозування, в середньому на 2-4 %. У дослідженні [4] приведений алгоритм, що поділяє зв'язки між джерелами даних на статичні та динамічні. Проте кожне прогнозування у ньому розглядається як окрема задача незалежна від інших прогнозів. Ціна акцій публічної компанії зумовлена прибутками, що має отримати компанія в майбутньому. Неодноразово у наукових дослідженнях [5] було зазначено що фінансові новини дуже сильно впливають на ціну акцій [6]. Деякі дослідники зауважують, що фондовий ринок керується подіями [7]. У дослідженні [8] був запропонований метод структуризації події як кортежу, що складається з агенту, предикату і об'єкту, а також метод, що базується на глибинному навчанні для захвату синтаксичної і семантичної інформації про подію.

Існує ціла серія досліджень з застосування результатів аналізу настроїв текстів джерел інформації до волатильності фондового ринку [9]. Методологія аналізу настрою у новинах та соціальних, мережах наведена у дослідженні [10]. У роботі [11] досліджується спільний вплив подій і настроїв інвесторів. Виявлено кореляції між фінансовими подіями та настроями інвесторів. Було доведено доцільність застосування багатовимірних даних для прогнозування фондових ринків. Однак, всі ці дані було поєднано у єдиний вектор. Проте існують скриті зв'язки між джерелами даних, що були втрачені у попередніх дослідженнях. Але тензорна структура дозволяє не втрачати ці данні, а навпаки, легко маніпулювати ними і виявляти все більш неочевидні зв'язки.

Прогнозування – це процес оцінювання майбутньої події (подій), який в тій чи іншій мірі використовує накопичені попередні дані і об'єднує їх визначеним шляхом, щоб отримати необхідну оцінку. Передбачення, крім цього, оперує суб'єктивними міркуваннями. Прогнозування неможливе без будь-яких історичних даних. Наприклад, виробник телевізорів на підставі минулих даних може спрогнозувати кількість телевізорів, які необхідно зібрати наступного тижня. Але припустимо, що виробник вирішив налагодити випуск нового товару, для якого у нього немає ніяких даних, для яких він міг би застосувати необхідні моделі і техніки. В даному випадку він не зможе здійснити прогноз, тільки передбачити майбутній результат.

Прогнози часто класифікують по періоду їх дії. У загальному випадку, короткострокові прогнози (до одного року) використовуються для вирішення поточних операцій; середньострокові (від року до трьох) і довгострокові (понад п'ять років) використовуються в стратегічних цілях.

Варто відзначити, що ідеальний прогноз неможливий через наявність великої кількості факторів, які важко оцінити з високим відсотком точності. Тому, замість пошуку ідеального прогнозу, набагато важливішим є добре знання існуючих моделей і методів та правильне їх застосування в залежності

від специфіки даних і предметної області, вміння пристосуватися до неідеального результату прогнозування.

Через те, що прогноз залежить від минулих даних, його надійність і точність будуть знижуватися відповідно до того, як далеко ми хочемо здійснювати прогнозування. Варто відзначити, що точність прогнозу і витрати на його здійснення взаємопов'язані. Кращі прогнози не обов'язково є найточнішими. Такі фактори, як його мета і доступність даних грають важливу роль у визначенні бажаного рівня точності.

1.2 Загальні методи прогнозування

Розглянемо класифікацію методів прогнозування, яка представлена на рисунку 1.1.

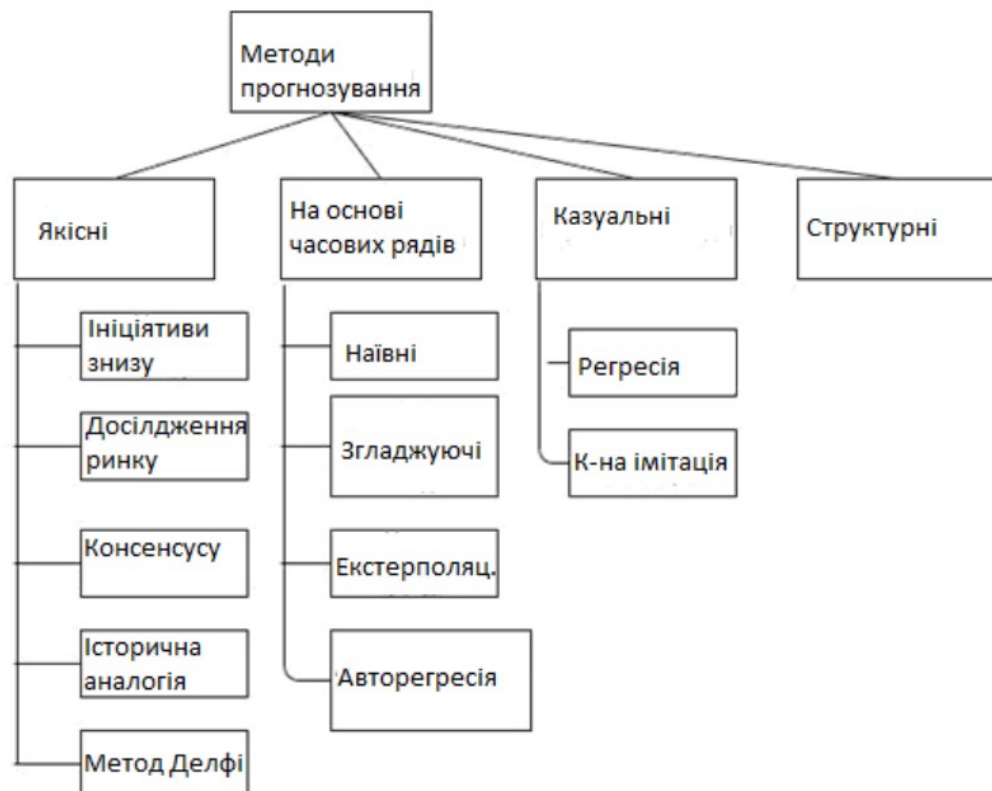


Рисунок 1.1 – Класифікація методів прогнозування

З рисунку можна зробити висновок, що існує 4 основних класи методів прогнозування, а саме: якісні методи, методи на основі часових рядів, казуальні та структурні. Проте, всі методи містять деякі загальні кроки.

Процес прогнозування в загальному випадку можна розбити на наступні кроки:

- 1) ідентифікація спільної мети;
- 2) вибір часового періоду прогнозу;
- 3) вибір моделі для прогнозу: для цього необхідно володіти знанням про різні моделі, застосування їх в різних ситуаціях, наскільки вони є надійними і в яких даних потребують. Виходячи з цих міркувань, може бути обрана одна або кілька моделей;
- 4) збір даних: дані повинні бути зібрані і представлені в тому вигляді, якому від них вимагає обрана модель;
- 5) прогнозування: застосування моделі до зібраних даних і обчислення прогнозу;
- 6) оцінка: прогноз, отриманий на попередньому етапі, розглядається з урахуванням довірчого інтервалу – більшість моделей дозволяють обчислити верхнє і нижнє значення, між якими розташовується прогнозована величина з певною часткою ймовірності.

Також, можуть бути застосовані інші методи для порівняння і поліпшення якості отриманого прогнозу.

1.3 Огляд методів

Якісні методи зазвичай застосовуються при участі групи експертів. Зрозуміло, що система прогнозування фінансових активів не може базуватися на постійній допомозі експертів. Проте, аналіз таких методів може бути корисним для розуміння специфіки задачі прогнозування у економіці [16].

Метод ініціативи знизу – даний метод будує прогноз послідовним шляхом, починаючи з самого «низу». Спочатку висувається гіпотеза, що людина, що знаходиться «ближче» всього до споживача або кінцевого використання продукту більш компетентна в прогнозуванні майбутнього. Це не завжди так, але в багатьох областях це припущення валідне і використовується в якості базису даного методу. Таким чином, прогнози з нижчого рівня підсумовуються і передаються на рівень вище (наприклад, районне управління). Процедура повторюється до тих пір, поки не досягне найвищого рівня [17].

Фірми часто наймають сторонні компанії, які спеціалізуються у вивченні і дослідженні ринку і займаються даним типом прогнозування. Вивчення ринку найчастіше застосовується для дослідження продукту з точки зору нових ідей, плюсів і мінусів існуючих рішень, аналогічних конкурентних продуктів в даному сегменті і т.д. Найчастіше, методами збору даних в цьому випадку є опитування та інтерв'ю [18].

Метод консенсусу – ключовою ідеєю даного методу є припущення, що група фахівців, що займають різні позиції, можуть дати більш точний прогноз, ніж маленька група осіб. В даному випадку основою методу є скликання зборів, на яких відбувається вільний обмін ідей і думок осіб з усіх рівнів керівництва («мозковий штурм»). Проблемою даного методу є той факт, що люди, що займають нижчі посади, зазвичай бояться висловлювати думку, відмінну від людей з більш високими посадами [18].

Метод Делфі – даний метод має схожі риси з методом консенсусу, при цьому виправляючи виникаючу при його використанні проблему – упередженість думки. Для її усунення пропонується дотримуватися анонімності осіб, що беруть участь у опитуванні; в цьому випадку кожен голос має однакову вагу. Кожному учаснику надсилається опитувальник, відповіді якого потім сумуються; учасникам потім надаються результати цього етапу, і висилається новий опитувальник. Зазвичай, даний метод займає близько трьох ітерацій [18].

1.4 Методи на основі часових рядів

Якщо ми розглядаємо прогнозування фінансових активів, то у більшості ситуацій, що виникають у нас є достатньо накопиченої історичної інформації для прогнозування. Існує безліч методів, які використовують статистичний аналіз використовуючи минулі дані, щоб зробити прогноз на майбутнє. Ключове припущення в даному випадку полягає в тому, що попередні характеристики і взаємозв'язок даних будуть зберігатися і в майбутньому. Різні методи по різному використовують попередні значення і по різному оцінюють їхній внесок в прогнозовані значення. Значення часового ряду складають фіксовані вимірювання в певні моменти часу конкретної змінної або характеристики. Часові інтервали, між якими відбувається вимір, можуть становити мілісекунди, секунди, хвилини, дні, тижні і т.д.

Тепер розглянемо докладніше методи засновані на аналізі часових рядів, та розглянемо їх доцільність до застосування у системі прогнозування фінансових ринків.

Наївні методи – методи, включені в цю групу, з точки зору математичного опису є найпростішими методами прогнозування. Найпростіший полягає в тому, що ми просто беремо останнє вимірне значення і використовуємо його в якості оцінки прогнозу. При цьому всі попередні значення часового ряду ніяк не враховуються. Для даних з високою сезонністю може бути використаний схожий метод, тільки в цьому випадку, прогноз наступного значення ряду приймається за останнє значення ряду в такому ж сезоні (наприклад, для всіх наступних прогнозів на серпень беруться значення за останній серпень) [19]. Також, варіацією першого методу є так званий метод дрейфу, який дозволяє оцінці прогнозу збільшуватися або зменшуватися з плином часу; зміни (так званий дрейф) в даному випадку рівні середньому зміні в попередніх даних, що схоже з проведенням прямої лінії між першим і останнім вимірами і екстрапольоване

далі. Також, одним з найпростіших методів прогнозування є так званий метод середнього, коли за прогнозоване значення береться середнє по всім попереднім даним. Приклади роботи даних методів наведені на рисунку 1.2.

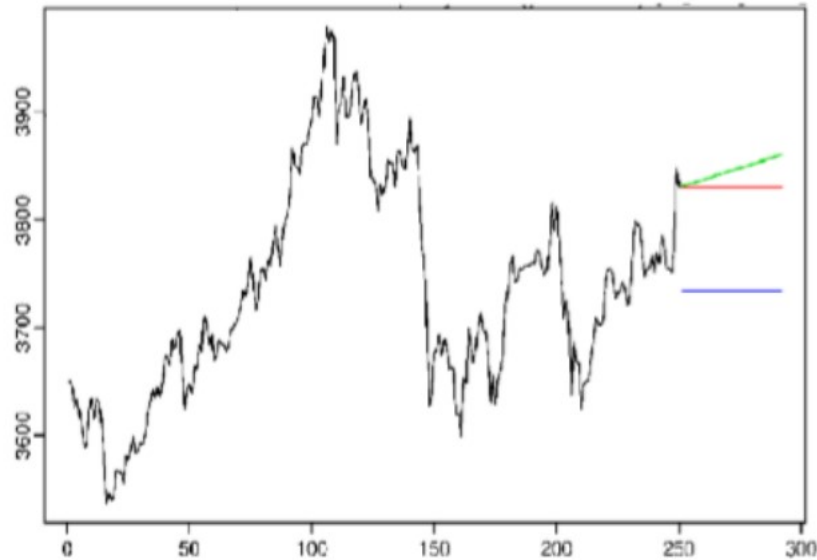


Рисунок 1.2 – Часовий ряд і його значення

Коли часовий ряд не змінюється занадто швидко і не має сезонних характеристик, то метод змінного середнього може бути використаний з метою видалення випадкових варіацій і здійснення оцінки прогнозу. У цьому випадку, ми використовуємо величину так званого змінного (локального) середнього. Її можна вважати відносним компромісом між простою моделлю середнього і методом дрейфу. В даному випадку дуже важливим є правильний вибір періоду для оцінки змінного середнього, так як, наприклад, при виборі великого часового періоду і наявності невеликого тренда дані будуть «відставати» від тренда, в той час як занадто короткий період буде надавати більшого значення різним варіаціям [20].

Попередні два методи мають один спільний недолік – ми повинні зберігати велику кількість історичних даних. Додаючи нові значення в ці методи, ми тим самим відкидаємо старі значення. У більшості додатків, як було згадано вище, прийнято вважати, що недавні значення є більш

інформативними і несуть більше корисної інформації, ніж старіші. Якщо це припущення є вірним, то метод експоненціального згладжування є найбільш логічним і простим методом в цій ситуації [21]. Даний метод автоматично привласнює ваги попереднім значенням таким чином, що ваги зменшуються експоненціально. Основний принцип даного методу полягає в тому, що ми вважаємо, що оцінка прогнозу наступного значення складається з двох компонент: старої оцінки прогнозу і певної величини помилки прогнозу.

На поточний момент даний метод є найбільш часто використовуваним методом прогнозування і включений в більшість програм здійснення прогнозу. Наступні фактори забезпечили цим методом таке успішне застосування:

- 1) даний метод є досить точним;
- 2) формула, що описує основну модель, відносно проста;
- 3) для його застосування потрібно відносно мало обчислювальної потужності.

Також варто відзначити, що у даного методу є різні розширення, які допомагають враховувати такі речі як сезонність і наявність тренда. Вони включають в себе подвійне або потрійне експоненціальне згладжування, а також, наприклад, метод Холта, який враховує наявність тренда.

1.5 Каузальні методи

Дані методи намагаються знайти та ідентифікувати фактори, які можуть вплинути на оцінку прогнозу даної змінної. Наприклад, якщо взяти до уваги інформацію про характеристику клімату, то теоретично можна краще спрогнозувати попит на парасольки. Подібні методи припускають, що подібні фактори можуть бути виміряні та враховані при оцінці прогнозу. Мета подібних методів – виробити найкращий статистичний взаємозв'язок між розглянутою залежною змінною і однією або декількома незалежними.

Найпоширеніший метод в даній категорії є метод регресійного аналізу, який ділиться на простий (залежність від однієї змінної) і багатовимірний (залежність від декількох змінних). Подібні методи найчастіше застосовуються для середньострокового і довгострокового прогнозування. Крім описаних вище, варто згадати метод комп'ютерної імітації, який, як правило, застосовується на великих обсягах даних [21].

1.6 Структурні методи

У даних методах визначальною ідеєю є припущення, що в основі спостережуваного процесу лежить якась структура, для якої задані деякі параметри, і, що найважливіше, базові правила і залежності. До подібних методів можна віднести наступні:

- 1) моделі на основі нейромереж;
- 2) моделі, що базуються на використанні Марковських ланцюгів;
- 3) моделі, що використовують класифікаційно-регресійні дерева;
- 4) моделі, засновані на використанні генетичних алгоритмів;
- 5) моделі, що використовують опорні вектора;
- 6) моделі, побудовані із застосуванням апарату нечіткої логіки.

Саме ці методи є найбільш придатними до застосування у системах прогнозування фінансових активів. Нейромережеві методи набирають все більшої популярності для вирішення задач прогнозування у останні роки. Дослідження показало, що застосування LSTM-нейромереж до часового ряду може давати прогнозування тренду вірне у 58 % випадках. Саме тому у цій магістерській дисертації обрано нейромережі як засіб прогнозування. Проте при прогнозуванні фінансових активів часто застосовують також технічний і фундаментальний аналіз.

Технічний і фундаментальний аналіз є невід'ємними складовими аналітики фінансових ринків. Дослідники фінансових ринків використовують

фундаментальний та технічний аналіз для прогнозування основних фінансових показників, курсів акцій а валют. Ці методи є класичними і традиційними вирішеннями задачі прогнозування фінансових ринків і успішно застосовуються вже більше тридцяти років.

1.7 Проблема зникаючого градієнту

Проблема зникаючого градієнта це складність, яка виникає при навчанні штучних нейронних мереж з використанням методів навчання на основі градієнта і 32 зворотного поширення похибки. Стандартні функції активації, такі як гіперболічний тангенс, мають градієнти в діапазоні $(-1, 1)$, а метод зворотного поширення помилки обчислює їх за ланцюговим правилом. Після множення цих чисел для обчислення градієнтів «фронтальних» шарів в n-шаровій мережі, градієнт (сигнал помилки) експоненційно зменшується разом з n, а передні шари навчаються дуже повільно [19]. Коли використовуються функції активації, похідні яких можуть приймати великі значення, є ризик зіткнутися з проблемою вибухаючого градієнту. Можливими рішеннями є

- багаторівнева ієрархія: шар мережі попередньо навчається, використовуючи методи навчання без учителя, а потім його значення регулюється за допомогою методу зворотного поширення помилки;

- довга короткострокова пам'ять: різновид архітектури рекуррентних нейронних мереж, коли величини помилки поширюються в зворотному напрямку від вихідного шару, помилка не випускається з пам'яті LSTM-блоку, вона безперервно передається назад кожному з вузлів, поки вони не будуть навчені відкидати подібні значення;

- залишкові мережі (Residual networks): один з найбільш ефективних методів вирішення проблеми зникаючого градієнта є використання

залишкових нейронних мереж (ResNets). Більш глибока мережа матиме вищу помилку навчання, ніж мережа з меншою кількістю шарів.

Команда Microsoft Research виявила, що поділ глибокої мережі на частини (скажімо, кожна частина являє собою три шари мережі) і передача вхідних даних в кожен фрагмент до наступного фрагмента (поряд із залишковим виходом) допомогли усунути більшу частину проблеми зникнення градієнта. Ніяких додаткових параметрів або змін в алгоритмі навчання не потрібно. ResNets показали нижчу помилку навчання (і тестову помилку), ніж їх менші за кількістю шарів аналоги.

Дослідивши методи боротьби із проблемою зникаючого градієнту, можна зробити висновок, що одне з можливих рішень – а саме варіація рекуррентних нейромереж LSTM позбавлена цієї проблеми. Це підтверджує доцільність вибору мереж із довгою короткостроковою пам'яттю у якості інструменту прогнозування фінансових активів.

1.8 Огляд програмних засобів, що реалізують нейромережеві алгоритми для вирішення задач прогнозування

На сьогоднішній день вже розроблено велику кількість програмних продуктів, придатних для застосування там, де виникає необхідність використання нейромережевих технологій у прогнозуванні фінансових ринків. Існують універсальні нейромережеві пакети, призначені для вирішення будь-яких завдань прогнозування (Brain Maker Pro, NeuroSolution), але, як показує практика, такі програмні продукти не завжди зручні для вирішення задач прогнозування часових рядів і курсів фінансових активів.

Існує окремий клас нейромережевих програмних продуктів, призначених виключно для вирішення завдань прогнозування курсів акцій. Ці продукти орієнтовані на фінансових працівників та людей, зацікавлених у

дослідженні фінансової галузі – трейдерів, біржових аналітиків і інших. Часто такі програмні пакети мають дружній графічний інтерфейс, і проектуються таким чином, щоб людина, що має слабе поверхове уявлення про нейронні мережі, змогла швидко їх освоїти. До таких програмних продуктів належать: Neuro Builder 2015, NeuroShell Day Trader, BioComp Profit, NeuroScalp.

Найбільш популярні у світі програмні продукти для вирішення задач прогнозування часових рядів, що реалізують нейромережеві підходи: Brain Maker Professional, NeuroShell Day Trader, Neuro Builder 2015.

Пакет Brain Maker Professional – призначений для побудови нейронних мереж зворотного поширення. Пакет включає в себе програму підготовки та аналізу вихідних даних NetMaker, програму побудови, навчання і запуску нейромереж BrainMaker, а також набір утиліт широкого призначення. Програмний пакет орієнтований на широке коло завдань – від створення прогностичних застосувань до організації систем розпізнавання образів і нейромережевої пам'яті. Значна кількість функцій програми орієнтована на фахівців в області дослідження нейромереж. Слід зазначити, що організація внутрішніх нейромережевих моделей є «прозорою» і легко доступною для програмного розширення.

У програмі BrainMaker передбачена система команд для пакетного запуску. Існує інтерфейсна програма-функція для включення навчених мереж в програми користувача. В цілому пакет може бути інтегрований в програмний комплекс цільового використання.

Програма BrainMaker призначена для побудови нейромережі за деякими початковими умовами, її навчання в різних режимах, модифікації параметрів мережі. Програма має значну кількість функцій для оптимізації процесу навчання. Крім цього, програма надає ряд методів аналізу чутливості виходів мережі до різних варіацій вхідних даних, при цьому формується детальний звіт, згідно з яким можна додатково оцінити ступінь функціональної залежності вхідних і вихідних значень.

NeuroShell Day Trader – нейромережева система, яка враховує специфічні потреби трейдерів і досить легка у використанні. NeuroShell Trader має, як і в інші стандартні програми для трейдерів, дружній графічний інтерфейс користувача. Можливості графіки дозволяють відображати дані у вигляді японських свічок (candlestick), в формі open / high / low / close, high / low / close, лінійних графіків або гістограм різних типів. Існує можливість міняти кольори, стискати і розтягувати шкали, ховати і знову робити видимими потоки даних. У щоденній, тижневої або місячної часовій шкалі можна відображати курси акцій, ціни товарів (commodities), біржові індекси (indexes), взаємні фонди (mutual funds), обмінний курс валют (foreign exchange rates) і подібне.

Neuro Builder 2015 Advanced – продукт, що належить до категорії наукомістких, високотехнологічних, вузькопрофесійних інструментів. Це додаток, що працює під управлінням ОС Windows. У своїй галузі – спеціалізовані програми для фінансових аналітиків – Neuro Builder 2015 займають прикордонне положення між серійними програмами і системами на замовлення. Вона може бути використана як самостійний продукт, може виступати складовою частиною складного аналітичного комплексу.

Нижче перераховані сім головних характеристик програми Neuro Builder 2015, що відрізняють її від аналогів та конкурентів:

а) програма Neuro Builder 2015 – додаток, створений спеціально для вирішення завдань прогнозування на фінансових ринках;

б) програма Neuro Builder 2015 – додаток, що дозволяє використовувати нейромережі в повсякденній роботі так само просто, як і звичні для трейдерів інструменти – програми технічного аналізу та електронні таблиці;

в) програма Neuro Builder 2015 – додаток, що дозволяє користувачеві використовувати навички, набуті під час роботи з Microsoft Office – технологічність і регулярність, які забезпечуються автоматизацією роботи

програми за звичними для користувача сценаріями і оформлення звітів програмою за створеними ним шаблонами;

г) програма Neuro Builder 2015 – єдиний на сьогоднішній день серійний та оновлюваний програмний продукт, що містить модуль дослідження даних до визначення архітектури нейромережі – Best Builder, що дозволяє автоматизувати визначення вхідного вектора параметрів завдання з урахуванням впливу кожного параметра вхідного вектора на передбачуваний результат;

д) програма Neuro Builder 2015 – не є «чорним ящиком»; Детальна документація містить опис всіх структур і файлів, включаючи тимчасові файли. Всі файли з даними системи зберігаються тільки в двох форматах - текстовий і EXCEL;

е) програма Neuro Builder 2015 – продукт, що не залежить від джерела даних; до складу програми входить модуль Data Builder Light, що дозволяє перетворювати фінансові дані з безлічі популярних форматів, в формат даних програми Neuro Builder 2015 і виправляти помилки в даних паралельно з їх перетворенням;

ж) програма Neuro Builder 2015 – дозволяє використовувати знайдені рішення неодноразово; дані для конкретного завдання завжди формуються на етапі її рішення через запит до бази даних і відразу знищуються після отримання результату; між сеансами роботи зберігається тільки опис способу отримання даних з локальної бази;

з) програма Neuro Builder 2015 реалізована у вигляді безлічі незалежних модулів, взаємодіючих в рамках комплексу за документованими інтерфейсами:

– кожен з програмних модулів оформлений у вигляді виконуваної програми (EXE) і відповідає за вирішення однієї з конкретних підзадач в складі загальної задачі прогнозування на фінансових ринках;

– кожен з модулів в змозі працювати як в складі комплексу програми Neuro Builder 2015, так і спільно з будь-якими іншими програмами, що підтримують її інтерфейс.

Технологія застосування програми Neuro Builder 2015 орієнтована на регулярність отримання результатів і економію робочого часу аналітика. Так програма Neuro Builder 2015 забезпечує мінімальний період прогнозування, відповідний однієї доби. В кінці торгового дня в базу даних програми заносяться ціни поточного дня, і скрипт буде працювати на обробку нових даних за заздалегідь підготовленим сценарієм. Контроль оператора в процесі обрахунків не потрібен. На початку наступного торговельного дня за результатами обрахунку вже можна отримати прогноз цін закриття цього дня. Таким чином, основний час роботи програми припадає на ніч, і завдання – прогнозування на день вперед – вирішене.

Роль користувача полягає в підготовці коректних сценаріїв для роботи програми і забезпеченні безперебійної подачі живлення персонального електронної обчислювальної машини, на якій запущена програма.

Інші програмні продукти, що реалізують нейромережеві алгоритми, менш відомі і не набули широкого поширення. Таким чином, можна зробити висновок, що типовий програмний продукт ринку нейромережевих програм, призначений виключно для прогнозування фінансових ринків, оцінюється виробниками приблизно в \$ 800 – \$ 3000. Зазначена ціна за для українських споживачів досить висока, проте, порівняно з потенційними прибутками, які можна отримати із використанням розглянутих програмних продуктів при успішній торгівлі на фінансових ринках, це дуже незначна сума.

1.9 Висновки до розділу 1

У розділі були розглянуті основні поняття у сфері прогнозування фінансових ринків, розглянуто специфіку прогнозування фінансових ринків,

проведено огляд поточного стану проблемної області і огляд існуючих рішень на ринку.

Аналіз застосування традиційних методологій прогнозування показує, що вони не відповідають сучасним потребам ринку у прогнозуванні фінансових активів.

Більшість методів розглядають вирішення задачі прогнозування лише з одного боку, беручи до уваги лише частину факторів, що впливають на зміну курсу фінансового активу і не враховують надважливі фактори, такі як реакцію людей у соціальних мережах і фінансові новини. У той час як фундаментальний аналіз розглядає зовнішні фактори, його застосування у системах прогнозування досить обмежене через складність реалізації і підтримки, або ж навпаки його реалізація досить неповна і орієнтується на зміну основних економічних і політичних індексів і звітів.

2 ЗАГАЛЬНІ ПОНЯТТЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ РОЗРОБКИ

2.1 Основні поняття нейронних мереж

Штучні нейронні мережі – це певні обчислювальні структури, що відтворюють модель елементарних біологічних процесів, які асоціюють із процесами, що відбуваються у мозку людини. Це системи, що спроможні до адаптивного навчання, яке відбувається внаслідок аналізу впливів – негативних та позитивних [22]. Прототипом для назви елементарного повторювача, штучного нейрону, у цих мережах слугує біологічний нейрон. Для того, щоб зрозуміти дійсні евристичні можливості нейромережевого аналізу у соціологічному дослідженні та відтворити якісне тестування цих можливостей, потрібно зрозуміти походження штучних нейронних мереж. З цією метою надалі будуть описані структурні та функціональні елементи, що надають змогу визначити схожість чи відмінність штучних та біологічних нейронів між собою.

Людський мозок та її нервова система складаються з нейронів. Нейрони, у свою чергу, поєднуються між собою за допомогою нервових волокон, які передають імпульси між нейронами. Усі процеси, що відповідають за передачу подразнень від основних органів чуття: очей, вух, шкіри, а також ті, що відповідають за управління діями та процеси, що відповідають за мислення виконуються за допомогою передачі електричних імпульсів між нейронами. Тобто, нейрон – це біологічна клітина, особлива клітина, яка виконує функцію обробки інформації (рис. 2.1). На рисунку зображено будову біологічного нейрона. Він включає в себе сому або ж тіло та відростки нервових волокон.

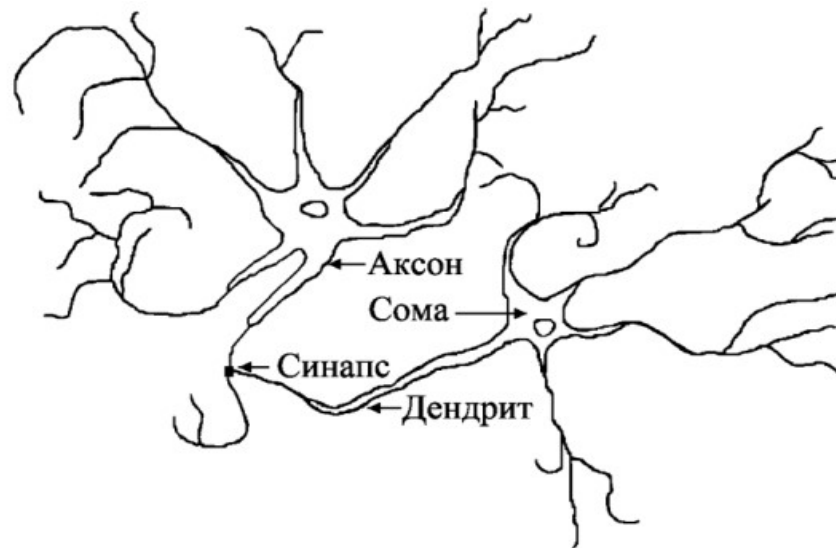


Рисунок 2.1 – Біологічний нейрон

Відростки існують двох типів: перший – аксон, за допомогою якого нейрон передає імпульс, другий – дендрити, за допомогою яких відбувається прийом імпульсів. У тілі нейрона знаходиться ядро, у якому міститься плазма та інформація про спадкові властивості. Плазма, у свою чергу, має певні молекулярні засоби для того, щоб виробляти матеріали, необхідні для нейрона.

Тобто, існує система отримання, передачі та обробки інформації, яка потрапляє до нейрона. Вона працює наступним чином: від аксонів або ж, можна сказати, приймачів нейрон отримує імпульси від інших нейронів, приймачами сигналів виступають дендрити, які потім ці сигнали передають вздовж власного аксона, вкінці він розгалужується на волокна. На кінцях таких волокон містяться синапси – утворення, що мають вплив на величину імпульсів. Ці утворення – функціональний вузол між двома нейронами.

Окрім цього важливо вказати поняття нейротрансмітерів. Це певні хімічні речовини, які вивільнюються під час досягнення імпульсом синаптичного закінчення. Їх головною функцією є гальмування або ж навпаки – збудження можливості нейрона, що відповідає за приймання інформації, генерувати електричні імпульси. Результативність такої передачі сигналу може бути різною, тобто сигнали навчаються, зважаючи на

активність процесів, учасниками яких вони стають. Такий процес залежності спрацьовує як пам'ять. Варто зауважити, що з плином часу поведінка нейронів може змінюватися, що залежить від синапсів.

2.2 Методи навчання нейронних мереж

Штучні нейронні мережі (ШНМ) – це програмна імплементація нейронних структур нашого мозку. Ми не будемо обговорювати складну біологію нашої голови, досить знати, що мозок містить нейрони, які є свого роду органічними перемикачами. Вони можуть змінювати тип переданих сигналів в залежності від електричних або хімічних сигналів, які в них передаються. Нейронна мережа у людському мозку – величезна взаємопов'язана система нейронів, де сигнал, який передається одним нейроном, може передаватися у тисячі інших нейронів. Навчання відбувається через повторну активацію деяких нейронних з'єднань. Через це збільшується імовірність виведення потрібного результату при відповідній вхідній інформації (сигналах). Такий вид навчання використовує зворотний зв'язок – при правильному результаті нейронні зв'язки, які виводять його, стають більш щільними [23].

Першим з методів навчання нейронних мереж є зворотне поширення помилки (Back-propagation) – це метод для обчислення часткових похідних (або градієнтів) функції, яка має форму функціональної композиції (як у нейронних мережах). При вирішенні задач оптимізації, використовуючи градієнтні методи (градієнтний спуск є лише одним з них), обчислюється градієнт функції на кожній ітерації.

Для нейронних мереж цільова функція має форму композиції. Є два загальні способи визначення функції: аналітична диференціація та приблизна диференціація. Аналітична диференціація – це обчислення похідних,

використовуючи правило ланцюга (основне обчислення). Приблизна диференціація – провадиться з використанням кінцевої різниці.

Другий метод навчання нейронних мереж – стохастичний градієнтний спуск (Stochastic Gradient Descent). Аби отримати базове розуміння стохастичного градієнтного спуску необхідно уявити річку, яка бере свій початок з вершини гори. Мета градієнтного спуску – досягнення найнижчої точки передгір'я. Якщо структура гори є такою, що річка безперешкодно протікає до пункту призначення (тобто до найнижчої точки передгір'я), то це є ідеальним варіантом, який необхідно отримати. В машинному навчанні це означає, що необхідно знайти глобальний мінімум (або оптимальне рішення), починаючи від початкової точки (вершини пагорба). Однак існують варіанти, коли природа місцевості має кілька можливих шляхів, що можуть змусити річку «потрапити в пастку» та утворити застій. В умовах машинного навчання такі ями називають локальними мінімумами, які не є бажаними. Існує безліч способів уникнути таких мінімумів.

Глибинні нейронні мережі з великою кількістю параметрів є дуже потужними системами машинного навчання. Однак серйозною проблемою в таких мережах є перенавчання. Це пов'язано, зокрема, із повільними темпами навчання самої системи, оскільки, аби визначити, що нейронна мережа правильно вирішує свою задачу, необхідно пройти етап тестування. Dropout є методом вирішення цієї проблеми (див. рис. 2.2).

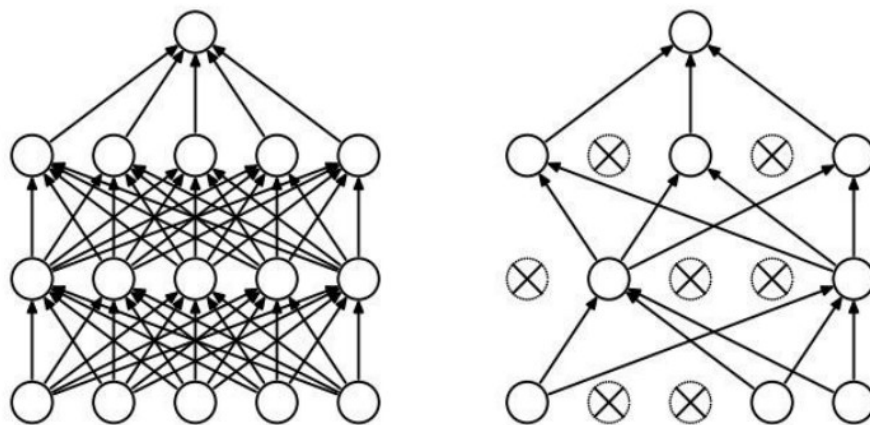


Рисунок 2.2 – Метод Dropout

Агрегування – це метод дискретизації на основі вибірки. При реалізації методу береться зразок вхідного представлення (матриця зображень, прихованого шару тощо), та зменшується її розмірність, що дозволяє робити припущення щодо функцій, які містяться в субрегіонах.

Це зроблено, для подолання проблеми перенавчання, шляхом надання абстрактної форми об'єкту, що розглядається. Крім того, це зменшує обчислювальні витрати, через зменшення кількості параметрів для навчання. Агрегування здійснюється шляхом застосування максимального фільтра до складових субрегіонів початкової матриці.

2.3 Згорткові нейронні мережі

Згорткові нейронні мережі були вперше описані Яном Лекуном та його командою у дослідженні [24], де вони вирішували задачу розпізнавання рукописних цифр із датасета MNIST. Через недостатню обчислювальну потужність та кількість даних для навчання згорткові нейронні мережі не набували широкого розповсюдження. Усе змінилося у 2012 році, коли Алекс Крижевський та його команда розробили модель AlexNet [11], здатну класифікувати 1.2 мільйони зображень на 1000 різних класів з похибками top-1 та top-5 у 37.5 % та 17 % відповідно. У статті [17] наведений детальний огляд принципу роботи згорткової нейронної мережі.

Згорткова нейронна мережа – це нейронна мережа прямого поширення спеціального виду, яка займається розпізнаванням ознак певних класів об'єктів на зображенні. У порівнянні з нейронною мережею прямого поширення згорткова нейронна мережа використовує значно меншу кількість параметрів.

Перший шар згорткової нейронної мережі – згортковий (convolution layer). Згорткова нейронна мережа працює на основі фільтрів, які займаються

розпізнаванням ознак певних класів об'єктів на зображенні: прямих ліній, кривих певного типу тощо.

Фільтр – це матриця чисел (ваг), які навчаються виявляти ознаки певних об'єктів на зображенні. Фільтр переміщається уздовж зображення та виявляє, чи присутня деяка ознака у його частині. Для отримання такої відповіді здійснюється операція згортки (convolution operation), яка математично визначається як сума добутків елементів фільтру та частини вхідного зображення. Назва згорткової нейронної мережі походить саме від назви цієї операції.

Якщо ознака присутня у певній частині зображення, в результаті згортки ми отримаємо велике значення. В іншому випадку, ми отримаємо або мале значення, або 0.

Зазвичай у згортковому шарі використовується декілька фільтрів. Ознаки, отримані в результаті операції згортки, об'єднуються, формуючи n -вимірну матрицю.

Наступний після згорткового шар – шар підвибірки (pooling layer). З метою прискорення процесу навчання та зменшення обсягу пам'яті, що споживає нейронна мережа, виконують операцію зниження розмірності (downsampling) вхідних даних (матриці ознак). Існує два способи зниження розмірності: максимальне об'єднання (max pooling) та середнє об'єднання (average pooling). Під час операції об'єднання уздовж вхідних даних переміщається так зване вікно просіювання. З даних, що потрапляють в його «поле зору», знаходиться максимальне чи середнє значення, яке переміщається в результуючу матрицю.

Перевагою операції об'єднання є те, що воно «змушує» нейронну мережу зосередитись на декількох нейронах замість усіх, що має регуляризує ефект, зменшуючи вірогідність перенавчання (overfitting).

Після згорткових шарів та шарів підвибірки йде один чи декілька повністю пов'язаних шарів (fully-connected layer). Отримані в результаті проходження через попередні шари ознаки (матриці) розгортаються у довгий

вектор, який проходить через декілька повністю пов'язаних шарів. У кожному такому шарі вектор ознак множиться на ваги шару, підсумовується зі значенням зміщення та проходить через нелінійну функцію активації.

Останній шар згорткової нейронної мережі – вихідний (output layer). Вихідний шар відповідає за обчислення вірогідності приналежності вхідних даних до певного класу об'єкта, наприклад, рукописної цифри «3». Кількість нейронів у вихідному шарі зазвичай дорівнює кількості класів об'єктів, які ми хочемо розпізнати. Виходи останнього повністю пов'язаного шару пропускаються через функцію активації, наприклад Softmax, яка формує вектор ймовірностей приналежності вхідного зображення до того чи іншого класу об'єктів.

Для того, щоб визначити, наскільки точно нейронна мережа розпізнає об'єкти, використовується функція втрат (loss function). Відповіддю цієї функції є дійсне число, що характеризує точність роботи нейронної мережі. Розповсюдженою функцією втрат, яка застосовується при прогнозуванні декількох класів об'єктів, є категоріальна функція кросентропійних втрат (Categorical Cross-Entropy Loss function)

2.4 Нейронні мережі прямого поширення

Нейронна мережа прямого поширення, нейромережа прямого розповсюдження (Feedforward neural network) – вид нейронної мережі, в якій сигнали поширюються в одному напрямку, починаючи від вхідного шару нейронів, через приховані шари до вихідного шару і на вихідних нейронах отримується результат опрацювання сигналу. В мережах такого виду немає зворотних зв'язків. Протилежним видом нейронних мереж із зворотними зв'язками є рекурентні нейронні мережі. Прикладом нейронної мережі прямого поширення є перцептрон Розенблатта, від якого і беруть свій

початок нейромережі прямого розповсюдження. Основна математична задача, з якою він здатний впоратися – це лінійне розділення довільних нелінійних множин, так зване забезпечення лінійної сепарабельності. В літературі часто термін перцептрон, багат шаровий перцептрон та нейромережа прямого поширення застосовуються синонімічно. Власне, між різними видами перцептронів спільне одне – вони усі є нейромережами з прямим поширенням сигналу, різняться в основному кількістю шарів, функцією активації та методом навчання.

2.5 Мережа радіальних базисних функцій

Мережа радіально базисних функцій використовується у математичному моделюванні – це штучна нейронна мережа, яка має радіальні базисні функції у якості функції активації. Виходом мережі є лінійна комбінація радіальних базисних функцій входу та параметрів нейрона. Мережі радіальних базисних функцій мають багато застосувань, зокрема, такі як апроксимацію функції, прогнозування часових рядів, задачі класифікації та керування системою.

Мережі радіальних базисних функцій мають ряд переваг перед розглянутими багат шаровими мережами прямого поширення. По-перше, вони моделюють довільну нелінійну функцію за допомогою всього одного проміжного шару, тим самим, позбавляючи розробника від необхідності вирішувати питання про кількість шарів. По-друге, параметри лінійної комбінації в вихідному шарі можна повністю оптимізувати за допомогою добре відомих методів лінійної оптимізації, які працюють швидко і не відчують труднощів з локальними мінімумами, так заважають при навчанні з використанням алгоритму зворотного поширення помилки. Тому мережа радіальних базисних функцій навчається дуже швидко – на порядок швидше, ніж з використанням алгоритму зворотного поширення.

Недоліки мереж радіальних базисних функцій: дані мережі володіють поганими екстраполюючих властивостями і виходять досить громіздкими при великій розмірності вектора входів.

Нейронна мережа радіальних базисних функцій містить у найбільш простій формі три шари: звичайний вхідний шар, що виконує розподіл даних зразка для першого шару ваг; шар прихованих нейронів з радіально симетричної активаційний функцією, кожен j -й з яких призначений для зберігання окремого еталонного вектора у вигляді вектора ваг w_j (h); вихідний шар.

2.6 Рекурсивні нейронні мережі

Рекурсивна нейронна мережа – тип глибокої нейронної мережі, сформована при застосуванні одних і тих же наборів ваг рекурсивно надструктурою, щоб зробити скалярне або структуроване пророкування надвхідною структурою змінного розміру через активацію структури в топологічному порядку (рис. 2.3).

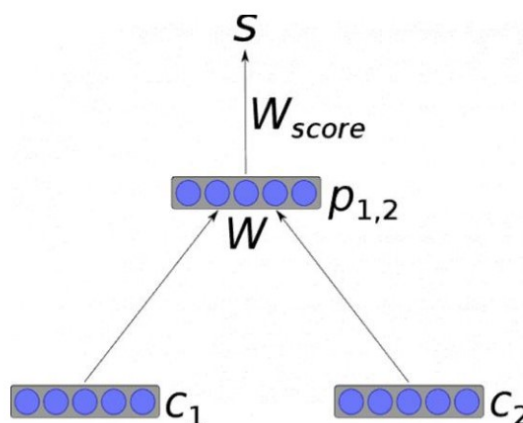


Рисунок 2.3 – Загальний вигляд рекурсивної нейронної мережі

У простій архітектурі нелінійність, така як тангенціальна функція активації і матриця ваг використовуються для об'єднання вузлів в батьківські об'єкти.

2.7 Рекурентні нейронні мережі

Рекурентні нейронні мережі – це клас штучних нейронних мереж, у якому з'єднання між вузлами утворюють граф орієнтований у часі. Це створює внутрішній стан мережі, що дозволяє їй проявляти динамічну поведінку в часі. На відміну від нейронних мереж прямого поширення, РНМ можуть використовувати свою внутрішню пам'ять для обробки довільних послідовностей входів. Це робить їх застосовними до таких задач, як розпізнавання несеgmentованого неперервного рукописного тексту та розпізнавання мовлення.

Рекурентна нейронна мережа, на відміну від прямої нейронної мережі, є варіантом рекурсивної ШНМ, в якій зв'язки між нейронами – спрямовані цикли. Останнє означає, що вихідна інформація залежить не тільки від поточного входу, але також від станів нейрона на попередньому кроці. Така пам'ять дозволяє користувачам вирішувати завдання нейролінгвістичного програмування: розпізнавання рукописного тексту або мови.

2.8 Довга короткострокова пам'ять (LSTM)

Нейронні мережі із використанням довгої короткострокової пам'яті здатний запам'ятовувати контекстні зв'язки як для елементів із довгою відстанню між ними, так і для елементів що розташовані поряд один з одним. Зауважимо, що сам LSTM-модуль не використовує функцію активації всередині своїх компонентів. Через це значення комірки не зникає із часом, а

градієнт не може зникати під час навчання нейронної мережі методом зворотнього поширення помилки. Метою розробки модулів довгої короткострокової пам'яті було формування більш стійкої пам'яті мережі, що значно би спростило фіксацію довгих контекстних залежностей. Загальним для всіх рекурентних мереж є те, що вони складені з ланцюга повторюваних блоків. Цей блок зазвичай має дуже просту структуру. Зазвичай, це єдина \tanh -функція у стандартних рекурентних мережах. Комірка довгої короткострокової пам'яті також має таку саму ланцюгову структуру, але комірка має вже не один, а чотири блоки що впливають один на одного.

LSTM-модуль має дуже корисну здатність до обробки послідовностей із довгостроковими залежностями: здатність додавати інформацію у стан комірки. При цьому цей процес регулюється "фільтрами" – регуляторами, які відповідають за процес забування чи зберігання інформації у комірці. Комірка пам'яті LSTM зберігає значення для довгих і коротких періодів часу. Це досягається за рахунок активаційної функції для кожної комірки пам'яті. Однією із переваг LSTM є те, що вона позбавлена ефекту зникнення градієнту.

Головний компонент LSTM – це стан ячейки горизонтальна лінія, що проходить по верхніх частинах схеми. Інформація у ній може зберігатися без змін. Проте, LSTM може видаляти інформацію зі стану ячейки; цей процес регулюється фільтрами (gates). Фільтри дозволяють пропускати інформацію на підставі деяких умов. Вони складаються з шару сигмоїдальної нейронної мережі і операції множення. По-перше треба визначити, яку інформацію можна викинути зі стану ячейки. Це рішення приймає сигмоїдальний шар, "шар фільтра забування" (forget gate layer). Наступний крок – вирішити, яка нова інформація буде зберігатися в стані ячейки. Цей етап складається з двох частин.

Спочатку сигмоїдальний шар що зветься "шар вхідного фільтра" (input layer gate) визначає, які значення слід оновити. Потім \tanh -шар будує вектор нових значень кандидатів. Нарешті, потрібно вирішити, яку інформацію ми хочемо отримувати на виході.

Вихідні дані будуть засновані на нашому стані ячейки, до них будуть застосовані деякі фільтри. Спочатку ми застосовуємо сигмоїдальний шар, який вирішує, яку інформацію зі стану ячейки ми будемо виводити. Потім значення стану осередку проходять через \tanh -шар, щоб отримати на виході значення з діапазону $[-1; 1]$, і перемножуються з вихідними значеннями сигмоїдального шару, що дозволяє виводити тільки необхідну інформацію.

2.9 LSTM-архітектури для задачі прогнозування часових рядів

У дослідженні компанії Google [25] розглядалася можливість покращення архітектури довгої короткострокової пам'яті. Було використано тисячі надпотужних ЕОМ і перевірено сотні варіацій архітектури. Результатом дослідження є те, що класична структура нейронної мережі довгої короткострокової пам'яті загалом краще справляється із задачею прогнозування ніж її варіації за критерієм точності прогнозу. Однак, було виявлено варіацію архітектури яка прогнозує деякі нестационарні часові ряди краще за класично. Це архітектура із рекурентним вузлом. Особливістю архітектури із вентиляним рекурентним вузлом є те, що фільтри мають доступ до стану комірки.

Рекурентний вузол поєднує вхідний фільтр та фільтр забування у новий тип фільтру – фільтр оновлення, а також поєднує стан комірки з прихованим станом.

2.10 Висновки до розділу 2

У даному розділі були розглянуті основні математичні підходи до складових системи прогнозування фінансових активів. Було розглянуто специфіку застосування нейромереж у задачі прогнозування фінансових

активів. Через ланцюговий характер вхідних даних, вибір було зроблено на користь рекурентних нейромереж.

Було розглянуто проблему зникнення контекстних зв'язків у рекурентних нейромережах і запропоновано механізм її вирішення – додавання LSTM-модулю до комірок рекурентних нейромереж. Результати аналізу показали, що нейронні мережі із довгою короткостроковою пам'яттю більш придатні до задачі прогнозування фінансових ринків через здатність до роботи із довгими контекстними залежностями та відсутності проблеми зникнення градієнту.

3 ПРОЄКТУВАННЯ ТА РОЗРОБКА НЕЙРОННОЇ МЕРЕЖІ

3.1 Налаштування середовища для розробки

В даний час існує величезна кількість мов програмування, а також програмних бібліотек для роботи з штучними нейронними мережами та генетичними алгоритмами, реалізованих на різних мовах програмування, таких як C ++, C #, Python і ін. В якості інструменту для вирішення задач, і реалізації алгоритмів вибрана мова програмування Python.

Python – високорівнева мова програмування загального призначення, орієнтований на підвищення продуктивності розробника і читання коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій. Python підтримує структурний, об'єктно-орієнтоване, функціональне, імперативне і аспектно-орієнтоване програмування.

Основні архітектурні риси – динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень, високорівневі структури даних. Підтримується розбиття програм на модулі, які, в свою чергу, можуть об'єднуватися в пакети.

Виходячи з цього, в якості мови програмування була обрана мова програмування Python, як середовище програмування було обрано програмне забезпечення Microsoft Visual Studio Code.

Для вирішення завдань машинного навчання існує безліч бібліотек і систем, що підтримують інтерфейс мови Python. Деякі з них: Apache Singa, Caffe, Keras, Microsoft Cognitive Toolkit, MXNet, TensorFlow, Theano, PyTorch.

Для реалізації розпізнавання обрана бібліотека Keras – відкрита нейромережева бібліотека, написана на мові Python – вона націлена на оперативну роботу з мережами глибинного навчання, при цьому спроектована

так, щоб бути компактною, модульною та розширюється та OpenCV – бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом.

Для розробки нейромережі була використана платформа Kaggle. Середовище організоване як публічна веб-платформа, на якій користувачі та організації можуть публікувати набори даних, досліджувати та створювати моделі, взаємодіяти з іншими фахівцями з даних та інженерами з машинного навчання, організовувати конкурси з дослідження даних та брати участь у них. У системі розміщено набори відкритих даних, надаються хмарні інструменти для обробки даних та машинного навчання. Також реалізовані навчальні ресурси, є розділ для розміщення вакансій роботодавцями, де також можлива організація конкурсів для відбору найкращих кандидатів.

Також середовище Kaggle – платформа для змагань з аналітики та передбачувального моделювання, в рамках якого статистики та добувачі даних конкурують у створенні найкращих моделі для прогнозування та опису даних, запропонованих компаніями або користувачами. Цей краудсорсинговий підхід ґрунтується на тому, що є безліч стратегій, які можуть бути застосовані до будь-якого завдання з передбачувального моделювання, і наперед не відомо, яка методика або аналітичний підхід буде найбільш ефективним.

Для роботи з даною платформою необхідно зареєструватися та налаштувати свій профіль. Після цього можна знаходити набори даних, що є у відкритому доступі та використовувати за своїми потребами.

3.2 Підключення необхідних бібліотек для роботи

Сентимент аналіз: процес обчислювальної ідентифікації та класифікації думок, висловлених у фрагменті тексту, особливо для того, щоб визначити,

чи є ставлення автора до певної теми, продукту тощо позитивним, негативним чи нейтральним.

Для розробки нейронної мережі була використана рекурентна нейронна мережа.

Для створення нейронної мережі в першу чергу були імпортовані всі необхідні бібліотеки (див. рис 3.1).

```
import numpy as np # Linear algebra
import pandas as pd # data processing, CSV file I/O (e.g.
pd.read_csv)

from sklearn.feature_extraction.text import CountVectorizer
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM,
SpatialDropout1D
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
import re
```

Рисунок 3.1 – Приклад коду для імпорту бібліотек

Налаштовуємо таблицю тільки з необхідними даними (див. рис 3.2).

```
data = pd.read_csv('../input/Sentiment.csv')
# Keeping only the necessary columns
data = data[['text', 'sentiment']]
```

Рисунок 3.2 – Код для підгруження даних

3.3 Загальний опис розробки нейронної мережі

Далі я відкидаю «нейтральні» почуття, оскільки моєю метою було розрізняти лише позитивні та негативні твіти. Після цього я фільтрую твіти, щоб залишилися лише дійсні тексти та слова. Потім я визначаю максимальну

кількість функцій як 2000 і використовую Tokenizer для векторизації та перетворення тексту на послідовності, щоб мережа могла розглядати його як вхідні дані (див. рис 3.3).

```
data = data[data.sentiment != "Neutral"]
data['text'] = data['text'].apply(lambda x: x.lower())
data['text'] = data['text'].apply((lambda x: re.sub('[^a-zA-z0-9\s]', '', x)))

print(data[ data['sentiment'] == 'Positive'].size)
print(data[ data['sentiment'] == 'Negative'].size)

for idx,row in data.iterrows():
    row[0] = row[0].replace('rt', ' ')

max_fatures = 2000
tokenizer = Tokenizer(num_words=max_fatures, split=' ')
tokenizer.fit_on_texts(data['text'].values)
X = tokenizer.texts_to_sequences(data['text'].values)
X = pad_sequences(X)
```

Рисунок 3.3 – Перетворення тексту у дані зрозумілі для нейромережі

Далі я складаю мережу LSTM (див. рис 3.4).

```
embed_dim = 128
lstm_out = 196

model = Sequential()
model.add(Embedding(max_fatures, embed_dim, input_length =
X.shape[1]))
model.add(SpatialDropout1D(0.4))
model.add(LSTM(lstm_out, dropout=0.2,
recurrent_dropout=0.2))
model.add(Dense(2, activation='softmax'))
model.compile(loss = 'categorical_crossentropy',
optimizer='adam', metrics = ['accuracy'])
print(model.summary())
```

Рисунок 3.4 – Код складання мережі LSTM

Необхідно відмітити, що змінні `embed_dim`, `lstm_out`, `batch_size`, `dropout_x` є гіперпараметрами, їхні значення певною мірою інтуїтивно зрозумілі, їх будемо використовувати далі, щоб досягти хороших результатів.

Також потрібно звернути увагу, що використовується `softmax` як функцію активації. Причина полягає в тому, що наша мережа використовує категоріальну крос-ентропію, і `softmax` є саме правильним методом активації для цього. Результат роботи цього блоку на рисунку 3.5.

```
-----
Layer (type)                Output Shape                Param #
=====
embedding_1 (Embedding)     (None, 28, 128)            256000
-----
spatial_dropout1d_1 (Spatial (None, 28, 128)            0
-----
lstm_1 (LSTM)                (None, 196)                 254800
-----
dense_1 (Dense)              (None, 2)                   394
=====
Total params: 511,194
Trainable params: 511,194
Non-trainable params: 0
-----
None
```

Рисунок 3.5 – Отриманий результат програми

Цим я визначаю набір даних тренування та тестування (див. рис 3.6). Результати показані на рисунку 3.7.

```
Y = pd.get_dummies(data['sentiment']).values
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size = 0.33, random_state = 42)
print(X_train.shape, Y_train.shape)
print(X_test.shape, Y_test.shape)
```

Рисунок 3.6 – Визначення набору даних тренування та тестування

```
(7188, 28) (7188, 2)  
(3541, 28) (3541, 2)
```

Рисунок 3.7 – Виведені на екран результати

Наступним кроком є тренування нейронної мережі. Спочатку було вибрано всього 7 епох (див. рис 3.8) .

```
batch_size = 32  
model.fit(X_train, Y_train, epochs = 7,  
batch_size=batch_size, verbose = 2)
```

Рисунок 3.8 – Код параметрів тренування нейронної мережі

Але навіть сім епох показали досить хороший результат, майже 93 % (рис. 3.9).

```
Epoch 1/7  
- 18s - loss: 0.4359 - acc: 0.8136  
Epoch 2/7  
- 16s - loss: 0.3222 - acc: 0.8649  
Epoch 3/7  
- 16s - loss: 0.2836 - acc: 0.8798  
Epoch 4/7  
- 16s - loss: 0.2565 - acc: 0.8962  
Epoch 5/7  
- 16s - loss: 0.2313 - acc: 0.9072  
Epoch 6/7  
- 17s - loss: 0.2048 - acc: 0.9206  
Epoch 7/7  
- 16s - loss: 0.1908 - acc: 0.9245
```

Рисунок 3.9 – Тренування нейронної мережі

Далі відбувається отримання набору перевірки та вимірювання точності (див. рис 3.10).

```
validation_size = 1500

X_validate = X_test[-validation_size:]
Y_validate = Y_test[-validation_size:]
X_test = X_test[:-validation_size]
Y_test = Y_test[:-validation_size]
score, acc = model.evaluate(X_test, Y_test, verbose = 2,
batch_size = batch_size)
print("score: %.2f" % (score))
print("acc: %.2f" % (acc))
```

Рисунок 3.10 – Отримання набору перевірки та вимірювання точності

Нарешті вимірюється кількість правильних відповідей (див. рис 3.11).

```
pos_cnt, neg_cnt, pos_correct, neg_correct = 0, 0, 0, 0
for x in range(len(X_validate)):

    result =
model.predict(X_validate[x].reshape(1,X_test.shape[1]),batch
_size=1,verbose = 2)[0]

    if np.argmax(result) == np.argmax(Y_validate[x]):
        if np.argmax(Y_validate[x]) == 0:
            neg_correct += 1
        else:
            pos_correct += 1

    if np.argmax(Y_validate[x]) == 0:
        neg_cnt += 1
    else:
        pos_cnt += 1
print("pos_acc", pos_correct/pos_cnt*100, "%")
print("neg_acc", neg_correct/neg_cnt*100, "%")
```

Рисунок 3.11 – Код для вимірювання кількості правильних відповідей

Зрозуміло, що знайти негативні записи дуже добре для нейронної

мережі, але вирішити, чи є вони позитивними, насправді не так. Моє обґрунтоване припущення полягає в тому, що позитивний навчальний набір значно менший, ніж негативний, отже, «погані» результати для позитивних записів.

Я розширив ядро за допомогою прикладу передбачення (див. рис 3.12), а також оновив виклики API до Keras 2.0. Мережа працює погано. Оскільки тренувальні дані є дуже незбалансованими (pos: 4472, neg: 16986), можливо слід отримати більше даних, використавши інший набір даних, використати попередньо підготовлену модель або вагові категорії, щоб отримати надійні прогнози. Приклад отриманих результатів продемонстровано на рисунку 3.13.

```
twt = tokenizer.texts_to_sequences(twt)
twt = pad_sequences(twt, maxlen=28, dtype='int32', value=0)
print(twt)
sentiment = model.predict(twt, batch_size=1, verbose = 2)[0]
if(np.argmax(sentiment) == 0):
    print("negative")
elif (np.argmax(sentiment) == 1):
    print("positive")
```

Рисунок 3.12 – Код для розширення ядра

```
pos_acc 63.43042071197411 %
neg_acc 90.09235936188077 %
```

Рисунок 3.13 – Отримані результати

Результатом розробки стала система прогнозування фінансових ринків із використанням рекурентних неймереж, що включає у себе модулі аналізу повідомлень у соціальних мережах, модуль обробки фінансових новин, модуль обробки технічних параметрів, модуль скорочення тензору, та, власне, неймережевий модуль.

Досліджено переваги і недоліки розробленої системи.

До переваг системи можна віднести легкість розширення, великий спектр підтримуваних форматів вхідних даних, можливість до інтеграції у більші системи за допомогою зручного API, порівняно невелику потребу у оперативній і постійній пам'яті електроннообчислювальної машини а також кросплатформеність, оскільки система викристоує кросплатформені технології, мови програмування та бібліотеки.

До недоліків системи можна віднести порівняно довгий час навчання, обмеженість у роботі із даними що не відповідають вхідному формату.

Розроблену систему для прогнозування курсів фінансових активів було навчено та протестовано на реальних даних.

Експеримент показав, що запропонований метод може підвищити точність прогнозування. Результат був досягнутий за допомогою комбінованого підходу до факторів що впливають на курси акцій що базується на тензорному підході. Також дослідження показало що LSTM-нейромережі загалом краще справляються із прогнозуванням часових рядів ніж звичайні RNN нейромережі.

Подальші дослідження доцільно спрямувати на оптимізацію алгоритмів обробки текстів новин, виявлення настроїв інвесторів та покращення алгоритмів навчання нейронних мереж.

ВИСНОВКИ

В роботі було проаналізовано сьогоdnішній стан проблемної області прогнозування фінансових ринків. Було розглянуто і проаналізовано основні сучасні методи прогнозування як часових рядів, так і безпосередньо курсів фінансових активів. Виявлені переваги і недоліки кожного з методів і запропоновано метод, що поєднує переваг технічного і фундаментального аналізу з прогнозуванням за допомогою нейромереж. Також було виконано порівняння систем, вже існуючих на ринку прогнозування фінансових активів.

Виходячи з результатів порівняльного аналізу було затверджено основні пріоритети і напрямки при побудові власної системи прогнозування фінансових активів. Було описано і вибрано її архітектуру і наведений набір користувацьких вимог до системи.

Після цього було обрано набір інструментів для вирішення поставлених задач. Було обрано архітектуру нейронної мережі яка найбільш доцільна для вирішення поставленої задачі – мережа з довгою короткостроковою пам'яттю. Було розглянуто можливі модифікації такої мережі. На основі обраних методологій і інструментів, враховуючи користувацькі вимоги було розроблено систему прогнозування фінансових ринків із використанням рекурентних нейромереж.

Розроблена система виконує поставлені задачі і задовольняє поставленим вимогам. Розроблену систему було протестовно на реальних даних і експериментально підтверджено оптимальність обраної архітектури довгої короткостроковій пам'яті за критерієм точності прогнозу у відсотках.

ПЕРЕЛІК ПОСИЛАНЬ

1. Haykin Simon O. *Neural Networks and Learning Machines*, 2009. 906 с.
2. Dasgupta D., McGregor D. Designing Application-Specific Neural Networks using the Structured Genetic Algorithm. *Proceedings of International Workshop on Combinations of Genetic Algorithms and Neural Networks*. 1992.
3. Савчук О. В., Кривенко К. С. Інтелектуальний аналіз діагностичної інформації електро- радіокомпонентів в умовах невизначеності. Збірник праць «Інтелектуальний аналіз інформації ІАІ-2013». Київ : Просвіта. 2013. С.211–217.
4. Huang H. *Statistical Mechanics of Neural Networks*. 2022. 302 с.
5. Zhang M. (ed.) *Artificial Higher Order Neural Networks for Economics and Business*, 2009. 542 с.
6. Шеремет О., Садовой В. Метод опорних векторів. *Математичне моделювання*. № 1. 2013. №28. С. 13.
7. Domingos P., Pazzani P. On the optimality of the simple Bayesian classifier under zeroone loss. *Machine Learning*. 1997. №29. С. 103–137.
8. Зорін Ю. М. Конспект лекцій з предмету «Комп'ютерні системи штучного інтелекту». URL : <http://m.scs.kpi.ua/course/view.php?id=107> (дата звернення 08.09.2022).
9. Біла Н. І. Інформаційні системи та технології в управлінні. Теоретичні відомості і завдання до лабораторних робіт. С. 6–19.
10. Goodfellow I., Bengio Y., Courville A. *Deep Learning*, 2016.
11. Alanyali M., Moat H. S., Preis T. Quantifying the relationship between financial news and the stock market, *Sci. Rep.* 3.
12. Mao H., Counts S., Bollen J. Quantifying the effects of online bullishness on international financial markets, *European Central Bank Workshop on Using Big Data for Forecasting and Statistics*, Frankfurt, Germany.

13. LeCun Y. LeNet-5, convolutional neural networks. URL : <http://yann.lecun.com/exdb/lenet/> (дата звернення 08.09.2022)
14. Zhang W. Shift-invariant pattern recognition neural network and its optical architecture. Proceedings of annual conference of the Japan Society of Applied Physics. 1988. 88 с.
15. Zhang W. Parallel distributed processing model with local space-invariant interconnections and its optical architecture. Applied Optics. 1990. №29. С. 4790–4797.
16. Subject independent facial expression recognition with robust face detection using a convolutional neural network. Neural Networks. 2003. №16. С. 555–559.
17. Николенко С. И., Кадурич А. А., Архангельская Е. О. Глубокое обучение. Погружение в мир нейронных сетей. 480 с.
18. Glorot X., Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. 2010.
19. Матеріали курсу CS231n: Convolutional Neural Networks for Visual Recognition. URL : <https://cs231n.github.io/convolutional-networks/> (дата звернення 08.09.2022).
20. Scherer D., Müller A., Andreas C., Behnke S. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. Springer. 2010. С. 92– 101.
21. Graham B. Fractional Max Pooling. URL : <https://arxiv.org/abs/1412.6071> (дата звернення 08.09.2022)/
22. Paul C. Tetlock. 2007. Giving content to investor sentiment: The role of media in the stock market. Journal of Finance 62, 3 (2007), 1139–1168.
23. Richard Ernest Bellman and Stuart E. Dreyfus. 1962. Applied Dynamic Programming. Rand Corporation.

24. Q. Li, Y. Chen, L. L. Jiang, P. Li, H. Chen, A tensor-based information framework for predicting the stock market, *ACM Trans. Inform. Syst. (TOIS)* 34 (2) 116 (2016) 11.
25. D. M. Cutler, J. M. Poterba, L. H. Summers, What moves stock prices, *J.Portf. Manag.* 15 (1989) 4 –12.