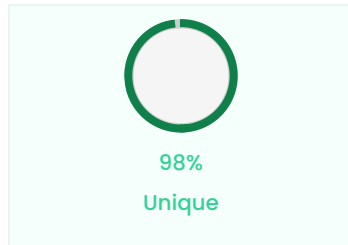
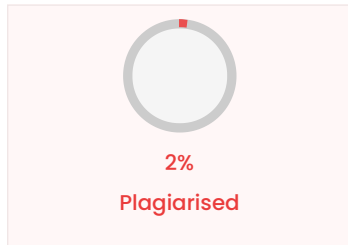


Plagiarism Scan Report



Words Statistics

Words	10594
Characters	80989

Exclude URL: None

Content Checked For Plagiarism

МІНІСТЕРСТВО ОСВІТИ НАУКИ УКРАЇНИ СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДО ЗАХИСТУ ДОПУЩЕНА Голова циклової комісії, спеціаліст в/к _____ С «ОБЛІКОВА СИСТЕМА ФІНАНСОВОЇ ЗВІТНОСТІ НА МОВІ PYTHON» Виконав ст. гр. ІПЗ – 118к9 _____ Д.О. Костерной Запоріжжя 2022 СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ КОЛ «ЗІЕІТ» Циклова комісія з інформаційних технологій ЗАТВЕРДЖУЮ Голова циклової комісії, спеціаліст в/к _____ РОБОТИ МОЛОДШОГО СПЕЦІАЛІСТА Студента гр. ІПЗ-118К9 Спеціальності: 121 – Інженерія програмного забезпечення Тема: «Облікова система фінансової звітності на мові Python» затверджена наказом по інституту: № 09.2 від 04 червня 2022 року 3. Перелік питань, що підлягають розробці: 1. Зібрати літературу та документацію присвячені підприємства з приводу випускної роботи та впровадження як розробку; 3. Розглянути та проаналізувати аналоги; 4. Виконати завдання випускної роботи; 6. Протестувати розробку та впровадити перед керівником підприємства; 7. Оформити звіт роботи молодшого спеціаліста № етапу Зміст Терміни виконання Готовність по графіку %, підпис керівника Г (корегування) теми випускної роботи молодшого спеціаліста та збір практичного матеріалу за темою випускної роботи молодшого спеціаліста 28.03.22-02.04.22 3 II атестація II розділ випускної роботи молодшого спеціаліста 10.04.22-10.06.22 4 Висновки та рекомендації, додатки, реферат 30.05.22-04.06.22 5 Перевірка випускної роботи молодшого спеціаліста, доопрацювання випускної роботи молодшого спеціаліста, підготовка презентації, отримання відгуку керівника молодшого спеціаліста 14.06.22-18.06.22 8 Подача випускної роботи молодшого спеціаліста на кафедру за 3 дні до 25.06.22 Керівник _____ Д.О. Костерной (підпис) (ініціали, прізвище) «_____» _____ Д.В. Булавін (підпис студента) (ініціали, прізвище) «_____» _____ 20: 30 рисунків, 1 додаток, 28 джерел, 7 формул, 3 лістинга. Об'єкт розробки: нейронна мережа для прогнозування часів для аналізу фінансових результатів. Згідно отриманих результатів, а також згідно алгоритму навчання, на основі яких вирішити в роботі: 1. Виконати огляд мов програмування та середовищ розробки для створення нейронної мережі. Створити нейронну мережу для прогнозування часових рядів; 4. Тестування нейронної мережі на точність прогнозування темі та зробити висновок. Результатом роботи є створення нейронної мережі для прогнозування часових рядів. МКС ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ Скорочення Словосполучення Пояснення/переклад Recurrent neural network Рекурентна нейронна мережа CNN Convolutional neuralnetwork Згортоква нейронна мережа формат обміну даними, що базується на JavaScript SQL Standartized Query Language Декларативна мова і середовище моделювання URI Uniform Resource Identifier Уніфікований ідентифікатор ресурсу CSS Cascading Style Sheet Штукатурка ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 5 ЗМІСТ 6 ВСТУП 8 РОЗДІЛ 1 ТЕОРИТИЧНІ ВІДОМОСТІ 9 НЕЙРОННА МЕРЕЖА 10 11 1.2 Аналіз предметної області 17 1.2.1 Принцип побудови нейронної мережі 17 1.2.2 Типи нейронних мереж існуючих аналогів 25 1.4 Висновок за розділом 1 28 РОЗДІЛ 2 ІНСТРУМЕНТИ РОЗРОБКИ ПРОГРАМИ 29 2.1 Загальні принципи розробки PyCharm 34 2.1.3 Фреймворк PyQt 35 2.1.4 СУБД MongoDB 35 2.1.5 Технологія TensorFlow 38 2.2 Розробка програми 47 2.4 UML-діаграми 49 2.5 Моделювання варіантів використання нейронної мережі 50 2.6 Висновок до розділу 2 52 3.1.1 Розробка нейронної мережі 56 3.2 Тестування програми 58 3.3 Висновок до розділу 3 58 ВИСНОВКИ 60 СІ

ВСТУП Актуальність теми Тема фінансових показників завжди буде актуальною, зокрема, на біржовому та фон інвестування для покупки цінних бумаг. Часові ряди використовуються у багатьох сферах життя людини. Виходячи показників. Оскільки мова йде про часові ряди, тобто ряди які постійно змінюються з часом, то для постійного актуальним використанням нейронних мереж, адже з розвитком технології – головною проривною технологією ста інформаційного життя людини у тому числі у галузі «data science», де потреба в аналізі даних тільки зростає невід'ємною частинною життя кожної людини, тому є актуальним використанням нейронних мереж для створення даних, тобто створення обліку. Мета і задачі розробки Мета роботи – розробити модель факторного аналізу ф можливість створення за технології «TensorFlow», а також проаналізувати можливість кореляції фактичних даних ч виявити рентабельність цінних бумаг компаній, для подальшого інвестування. Методи, засоби та технології розро свого аналізу зможе прогнозувати дані – використано технологію «TensorFlow» для створення графу нейронної л алгоритму тренування нейронної мережі на мові програмування «Python», база даних «MongoDB» для зберігання одержаних результатів Результатом виконання дипломної роботи є програма, яка використовує нейронну мережу

РОЗДІЛ 1 ТЕОРІТИЧНІ ВІДОМОСТІ 1.1 Дослідження основних понять 1.1.1 Часові ряди Часовий ряд або ряд динамічного значення будь-яких параметрів ряду. У тимчасовому ряді для кожної одиниці статистичного матеріалу має бути суттєво різнитися від простої вибірки даних, оскільки при аналізі враховується взаємозв'язок статистичного матеріалу з характеристиками вибірки[2]. Аналіз часових рядів Аналіз часових рядів – це сукупність математико-статистичних методів їх прогнозування. До цього належить метод регресійного аналізу, тобто статистичного взаємозв'язку між однією незалежних кількісних змінних, а також методи згладжування, – методи часових рядів призначених для адекватного створення ряду середніх шляхом взяття середніх значень у часовому ряду в межах заданих періодів. Визначення математичну модель аналізованого часового ряду. Часові ряди складаються з двох елементів: періоду часу, за якого беруться значень тієї чи іншої показника, званих рівнями низки. Є дві головні цілі аналізу часових рядів: визначення приросту тимчасового ряду за минулими значеннями. Часові ряди класифікують: 1. За кількістю показників, тобто одним чи декількома; 2. За повнотою часових рядів: повні та неповні часові ряди; 3. За характером тимчасового параметра: моментні та інтервальні тимчасові ряди; 4. За характером тимчасового параметра: моментні та інтервальні тимчасові ряди (не змінної характеристикою з часом), у яких середнє значення та міра розкиду значень випадкових значень розвитку. У свою чергу прогнозування часових рядів це додання до даних минулих періодів, прогнозуючих даних біржовий курс, під час аналізу якого намагаються спрогнозувати основний напрямок розвитку курсу[1]. 1.1.2 Нейронні мережі Відображення роботи біологічної мережі, яка складається з нейронів, з'єднаних між собою дендритами[5]. Класифікація нейронних мереж, яка складеться з декількох шарів, нейрон попереднього шару пов'язаний з кожним нейроном наступного шару нейронами є зв'язок, який має певну «вагу». При проходженні по лініям – сигнал міняє своє значення відповідно до ваги зв'язку, який, потім, пропускає суму через активаційну функцію. Вихідне значення цієї функції є вихідне значення нейронної мережі. В залежності від типу поставленої задачі, використовують різні активаційні функції, існують такі функції: 1. Сигмоїдна логістична функція; 2. Гіперболічний тангенс. Порогова функція – це функція, яка пропускає сигнал через поріг, головним з них це значення яке відповідає двом класам – 100 відсотків та 0 відсотків. Лінійна функція – функція, яка використовується в лінійній регресії, має таку формулу: $y = ax + b$, (1.2) де a – коефіцієнт пропорційності Через свою лінійність не має популярності у використанні, має таку формулу: $y = e^{-x}$, (1.3) Ця функція є найпопулярнішою серед інших функцій, оскільки може апроксимувати вихідні дані нейронних мереж. Сигмоїдна логістична функція – функція яка використовується у задачах класифікації, має вигляд $y = \frac{1}{1 + e^{-x}}$, діапазоном від -1 до 1 Гіперболічний тангенс – інший вид сигмоїдної функції, має формулу: $y = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, (1.5) Має більший коефіцієнт, ніж сигмоїдна функція, складається за трьох шарів Перший шар має назву вхідний, він приймає дані, пропускає через функцію активації, другий шар, знову функція активація, помноження, потім вихідний шар, який видає кінцевий результат. На мові програмування нейронної моделі[3]. Рисунок 1.1 – Нейронна мережа Штучні мережі застосовуються в таких областях, в яких потрібен інтелект є малоефективним, попри це, штучний інтелект найкращим чином підходить для постійного аналізу даних використанням нейронної мережі у всіх сферах життя людини. Нейронні мережі застосовуються у таких мережах: автоматизація процесу розпізнавання, автоматизація процесу прийняття рішень; управління, кодування та декодування – перша компанія яка стала використовувати нейронні мережі у своїх додатках, так наприклад проводяться тести у галузі управління цінами та виробництвом (втрати від неоптимального планування виробництва часто недооцінюються в сезон, курсів валют та багатьох інших факторів, то і обсяг виробництва повинен гнучко варіюватися з метою оптимізації залежності між витратами на рекламу, обсягами продажів, ціною, цінами конкурентів, днем тижня, сезоном і т. д.). Стратегія виробництва з погляду максимізації обсягу продажу чи прибутку. У медичній діагностиці нейронні мережі Система об'єктивної діагностики обробляє зареєстровані «викликані потенціали» (відгуки мозку), які у вигляді синтетичних сигналів використовуються у процесі обстеження. Щоб почати діагностику слуху досвідченому аудіологу для цього необхідно провести тестування на основі нейронної мережі здатна з тією ж вірогідністю визначити рівень слуху вже по 200 спостереженнях протягом дня, також нейронні мережі широко використовуються для прогнозування довгострокових та/або короткострокових тенденцій

Історія формування поняття «нейронна мережа» та способу її реалізації: У 1943 – 1945 роках – У. Маккалок та У. Піт про логічне обчислення ідей та нервової активності. На початку своєї співпраці з Піттсом Н. Вінер пропонує йому [5]. У 1948 році – опублікована наукова книга Н. Вінера про кібернетику. Основною ідеєю стало уявлення складних У 1949 році – канадський фізіолог та нейропсихолог. Д. Хебб пропонує перший алгоритм навчання, який припустив, якщо в процесі навчання обидва нейрони узгоджено відчують збудження, або гальмування. Простий алгоритм, як У 1958 році американський вчений у галузі психології, нейрофізіології та штучного інтелекту Ф. Розенблатт винаходить завдання класифікації. У 1960 році Бернард Уїдроу розробили «ADALINE» та «MADALINE», різниця між Адалін і станом ваги ADALINE встановлюються відповідно до вагової суми входів формула 2.1, таким чином вихід спрощується і протязі етапу навчання – сигнали сітки надходять до передавальної функції, таким чином її вихід використовується проблем передачі інформації». Петровим проводиться дослідження завдань «важких» для перцептронів. На цю роботу як переробкою алгоритму для виправлення його недоліків». У 1970 році Мінський анонсує доказ обмежені вирішувати деякі завдання, що пов'язані з інваріантністю уявлень. У 1972 році Дж. Андерсон пропонує новий тип нейронів. В. Хакімов запропонував нелінійну модель із синапсами, а також вводить її для вирішення задач у галузі меді поширення помилки на навчання багатопшарових перцептронів. У 1975 році – Фукусіма представляє когнітрон (ієрархічну багатопшарову організацію, у якій нейрони між шарами пов'язані лише локально.) – мережу, що використовується для інваріантного розпізнавання образів, але така архітектура досягається тільки за допомогою показав, що нейронна мережа зі зворотними зв'язками може являти собою систему, що мінімізує енергію (повнозв'язані представлені моделі мережі, що навчається без вчителя (нейронна мережа складається з адаптивних лінійних сигнал перетворюється на одиничний, інші звертаються в нуль), вирішальної задачі кластеризації, візуалізації дані 1986 році – Девідом І. Румельхартом, Дж. Є. Хінтоном і Рональдом Дж. Вільямсом – розвинули метод зворотного проходу математик, кібернетик та інформатик Хінтон в університеті Торонто – столиця Канади, створив спеціальні алгоритми Хінтон під час тренування нижніх шарів мережі споміг застосовувати малу «машину» Больцмана, що являє собою Хінтону необхідно використовувати багато прикладів образів, що розпізнаються. Після навчання виходить готовий результат (наприклад, здійснювати пошук осіб на зображенні) [5].

1.2 Аналіз предметної області

1.2.1 Принцип побудови нейронної мережі

Вивченням біологічних нейронів. Штучний нейрон є основою будь-якої штучної нейронної мережі. У свою чергу нейрон імітує роботу нейронів мозку. Кожен нейрон характеризується своїм поточним станом за аналогією з нервовими клітинами. Нейрони відносно просто описати в кодї, оскільки працює за принципом транзистора, приймає або 1 або 0. Штучний нейрон оскільки має групу синапсів, тобто входів, які з'єднані зі виходами інших нейронів, а також спеціальний аксон, який гальмування надходить на синапс інших нейронів. Також використовується третій вид нейронів – «bids», нейрон, який дає результат шляхом зсуву графіка функції активації [7]. Найпростіші нейронні мережі були створені згідно концепції собою систему, що складається з трьох шарів елементів: S (сенсорного), A (асоціативного) та R (реагуючого). Під час навчання (A → R) піддаються корекції в процесі навчання. Шар синоптичних зв'язків S → A також містить ваги, w , вручну. Їх значення можуть дорівнювати або «1», або «-1», що відповідає збудливому синапсу і гальмує синапсів. Такі ваги можна було прийняти рівними мінус нескінченності [4]. Отже, для створення нейронних мережі перше що потрібно зробити – це надходження до другого шару дані першого шару, перемножуються зі вагами та поступають до другого шару (активують наступний шар). Далі до наступного шару (реагуючого або вихідний), де знов перемножуються зі вагами та поступають до активаційної функції. Інший вигляд, після чого дані зрівнюються зі фактичними, а у разі не співпадіння змінюються ваги спеціальним алгоритмом помилки, який, у свою чергу, базується на алгоритмі градієнтного спуску. Для змінювання вагових коефіцієнтів мережі при подачі на вхід послідовності навчальних вхідних даних. Формально, для того щоб зробити один крок за межами параметрів мережі), для цього необхідно передати на вхід мережі послідовно абсолютно весь набір тренувальних даних, щоб знайти помилку та розрахувати необхідну зміну коефіцієнтів мережі, але після подачі всіх даних розрахувати суму в квадраті помилок, щоб зробити корекцію коефіцієнтів «на один крок». Очевидно, що при великому наборі тренувальних даних алгоритм градієнтного спуску саме тому часто втілюють коригування коефіцієнтів мережі після кожного елемента навчання, де значення градієнта помилки на одному елементі навчання. Такий метод називають стохастичним градієнтним спуском чи оперативним градієнтним спуском стохастичного наближення. Теорія стохастичних наближень дає умови збіжності способу стохастичного градієнтного спуску. У випадку коли потрібно створити нейронну мережу, для роботи з великим обсягом даних, потрібно:

1. Оптимізувати початкові значення вагових коефіцієнтів;
2. Виконати стандартизацію вхідних даних;
3. Підібрати швидкість навчання.

Алгоритм зворотного розповсюдження помилки має один великий недолік – попадання в локальний мінімум універсального рішення цієї проблеми, то алгоритм навчання запускають зі різними навчальними даними вагових коефіцієнтів, тобто того щоб з більшою вірогідністю знайти локальні мінімуми. Наступна проблема градієнтних спусків – це повільна оптимізація, існують такі методи оптимізації: Оптимізація Нестора (Nesterov Momentum) – це розширення імпульсу прогнозованих позицій у просторі пошуку, а не самих фактичних позицій, тобто рух по точкам у визначеному напрямку.

протягом деякого часу у майбутньому. Це дає ефект використання прискорювальних переваг імпульсу, дозволяючи ймовірність пропуску або перевищення його. Оптимізація «Adagrad» – оптимізатор зі швидкостями навчання, який оновлюється параметр під час навчання. Чим більше оновлень отримує параметр, тим менше оновлень. Оптимізація зберігає ковзний середній квадрат градієнтів, а також розділення градієнта на корінь цього середнього. Суть полягає в адаптуванні швидкості навчання на основі рухомого параметра оновлень градієнтів, замість того, щоб накопичувати швидкість навчання, навіть якщо було зроблено багато оновлень. У порівнянні з «Adagrad», в оригінальній версії «Adadel» версії встановлюється початкова швидкість навчання. Оптимізатор «Adam» – це спосіб оптимізації, який використовує градієнтного спуску для ітеративного оновлення ваги мережі на основі навчальних даних. Наступна проблема полягає в тому, що значення і вони одразу переміщуються в область насичення функції активації, тобто буде не рівномірною функцією. даних по формулі: $\sigma(x) = \frac{1}{1 + e^{-x}}$ (1.7) У цій формулі «max» та «min» – максимальне і мінімальне значення вхідних даних по всій перебувати в діапазоні від 0 до 1. Також потрібно провести нормалізацію даних на виході, через ті ж самі значення відразу для кожного спостереження, а після прогін через мережу деякої їх кількості – використовують множину «mini-batch» проганяючи mini-batch через мережу, підсумовують локальні градієнти на кожному нейроні, а потім, коригують ваги позитивні та негативні значення будуть скомпенсовані і залишаться корисне зміщення, яке призведе до зміни ваг у ітерацій по всій «епохи». У разі незадовільних результатів значення змішуються випадковим чином. Вкінці кожної епохи графік який показано на рисунку 1.4[7]. Рисунок 1.4 – Критерій якості по епохам Ще одна проблема з якою можна зустрітися тільки приклади з навчальної вибірки, адаптуючись до навчальних прикладів, замість того щоб вчитися класифікувати є метод «Dropout», який випадковим чином відключає нейрони, таким чином знижує спеціалізацію кожного окремого нейрона в архітектурі нейронних мереж, яка була придумана Яном Лекуном у 1988 році. Загальна ідея архітектури такої мережі полягає в тому, що дендрити кожного нейрона поєднуються не з усіма рецепторами сітківки ока, а лише з деякою локальною областю цілком. Таким чином вхідні дані (пікселі) зображення подається на вхід нейрона тільки в межах обмеженої області заданий крок і входи подаються вже до наступного нейрона. Так відбувається сканування всього зображення. Вагові коефіцієнти по пікселям повторюється зі вже змінним набором коефіцієнтів. Таким чином отримуємо другу групу нейронів, формується перший прихований шар нейронів загортової нейронної мережі. Якщо розглянути більш детально зображення (пікселі) сигнал проходить через загорткові шари, у яких чергується власне згортка. Саме завдяки такій архітектурі кожному наступному шарі, саме така «карта» ознак вбавляє у вигляді, і поперек це збільшується кількість каналів. складних ієрархій ознак. На виході загорткових шарів мережі додатково встановлюють кілька шарів перцептронів, і загортової нейронної мережі це утворення шар згортки, у свою чергу шар згортки включає в себе для кожного фрагментами (підсумовуючи результати поелементного сполучення для кожного фрагмента). Загальний вигляд обчислених ознак потрібно виконати ще одну операцію – «Pooling», тобто зміна масштабу картки ознак, існують MinPooling – відбір найменших значень; 3. AveragePooling – вибір середніх значень. Отже, отримуємо аналіз даного шару мають здатність знаходження більш загальних ознак на зображенні. Рисунок 1.5 – Принцип роботи загортової нейронних мереж, яка була придумана Хопфилдом у 1982 році, а першою сучасною рекурентною мережею стала рекурентна нейронних мереж є ефективними для моделювання даних послідовності, таких як тимчасові ряди або природна мовна один прихований шар з набором зворотних зв'язків (саме вони визначають рекурентність мережі). Вхідний та вихідний архітектуру можна як трьох блоків: вхідні дані (x), «RNN», вихідні дані (h), схема цієї архітектури показано на рисунку 1.6 – мережа розгорнута у часі, таким чином можна наочно бачити кожен крок роботи цієї нейронної мережі. Така називається «many to many», також існують архітектури: «many to one», «one to many» та «one to one». Архітектура один вихід, у свою чергу архітектура типу «one to many» має один вхід та виходи на кожному нейроні. Архітектура навчати алгоритмом зворотного розповсюдження помилки, оскільки має аналог зі багат шаровою мережею при розгорнутому вхід кожного нейрона надходить як вхідний сигнал, а й попередній стан мережі, також вагові коефіцієнти єдині всім і з різних ітерацій обчислення, тому використовують змінений алгоритм під назвою «Backpropagation Through Time» кроків, розгорнутих у часі[5].

1.2.3 Опис предметної області

Для того, щоб розпочати роботу з нейронною мережею, потрібно мати дані, дата, ціна – все у одній колонці. Нейронна мережа почне аналізувати дані та конвертувати їх для надходження у мережу та почне тренування, після створення ваг – буде прогнозовано декілька значень по дням та створений відповідний графік та порівняти зі фактичними, з великою вірогідністю дані будуть не точними, адже нейронна мережа потребує додаткові дані (фактичні), згідно яких – нейронна мережа, перенавчається, та знов видає прогнозуючі значення декілька успішних аналогів використання нейронних мереж у прогнозуванні часових рядів. Наприклад компанія Amazon планування ланцюжка поставок та прогнозування аналітики. Рішення «GMDH» побудовані на 100% запатентованій технології запасів, забезпечуючи повну прозорість всього ланцюжка поставок. Компанія створює передові програмні рішення, і це дозволяє прогнозування GMDH, забезпечуючи точне та гнучке прогнозування для бізнесу. Продукт GMDH Streamline – це рішення, яке дозволяє підприємствам максимізувати віддачу від своїх капіталовкладень[9]. Рисунок 1.7 – Логотип

«NeuroShell Trader» – нейронна мережа, яка точно аналізує часові ряди для потреби трейдерів, тобто це програмна сама по собі, це набір інструментів як традиційних, так і методів штучного інтелекту, що можуть комбінувати д складатися з індикаторів та правил, які трейдери використовували протягом багатьох років, методів штучного інте товарів, опціонів, FOREX, індексів та багато іншого. Отже програма NeuroShell Trader прогнозує часові ряди зав використання «NeuroShell Trader» Також існує аналог «BrainMaker» – програма яка призначена для побі розповсюдження помилки. Він включає програму підготовки та аналізу вихідних даних, навчання та запуску нейрон прогнозує значення для бірж, а також займається моделюванням кризових ситуацій. «BrainMaker» – одна із перши Рисунок 1.9 – Приклад використання «BrainMaker» 1.4 Висновок за розділом 1.1. Наведено характеристику об'єкт аналіз кількох аналогів нейронних мереж для прогнозування часових рядів, що реалізують схожі функції предм нейронної мережі.

РОЗДІЛ 2 ІНСТРУМЕНТИ РОЗРОБКИ ПРОГРАМИ

2.1 Засоби розробки

2.1.1 Мова програмування

мова програмування, згідно індексу «PYPL», рис 2.1. Мова програмування Python використовується у веб-розробці, технологіях в індустрії програмного забезпечення, також Python є мультипарадигмальною мовою програмуван орієнтоване програмування. Розробка мови програмування Python, а також реалізація почалася у 1989 року співр мови програмування Python почалася ще у в 1980-х голландським програмістом Гвидо ван Россумом, а рс інформаційного університету в Нідерландах. Мова програмування Python була задумана як нащадок мови програ цілях конкурентності з такими мовами програмування як Бейсік, Паскаля і AWK. Мова програмування Python не г хороша основа вивчення програмування та використання без високих навиків у програмуванні у повсякденній ро операційною системою «Амоеба». Ван Россум є визначним автором мови програмування Python і продовжував е липня 2018. Початок розвитку був зі Python 1.0, якмй з'явився у липні 1994 року. Генеральними можливостями булк лямбда-обчислення, map, filter та згортка списку. Россум запровадив, що "Python придбав функції lambda, reduc надав патчі, що реалізують ці функції". Кінцевою версією, випущеною Ван Россумом під час роботи в університеті Россум повернувся до роботи над Python-ом у корпорації національних дослідницьких ініціатив в штаті Вірджинія, безліч нових функцій, найбільш вагомими були запозичені в Modula-3 іменовані параметри і вбудована спеціал проста форма інкапсуляції даних за допомогою функції «name_mangling». Name_mangling змінює побудову к використовується для вирішення багатоманітних проблем, викликаних необхідністю вирішування унікальних ім декодування інформації в імені функції, структури, абр класу та іншого типу даних, щоб передати більше семантич університеті математики та інформатики Россум запатентував проект «Програмування для всіх», цей проект призна різних навиків програмування, на основі отримання базового знання мови та математики, необхідних для більшост відносно простому синтаксису. Проект Россума фінансувався «управлінням перспективних дослідних проектів мініс кістяк команди розробників мови програмування Python, сформували команду «BeOpen PythonLab». Python 2.0 команда розробників PythonLab приєдналися до компанії Digital Creations. Реліз версії 1.6 включав спеціальну ліц яка була значно довшою за ліцензію організації «CWI», яка використовувалася насамперед. Новий ліцензійний дого Фонд вільного програмного забезпечення (коротко FSF) заявив про те, що ця стаття порушує вибір правової н Організація BeOpen та FSF підписали договір по вільну ліцензію Python. Python 1.6.1 відмінився від Python 1.6 ли версією 2.0 була випущена 15 жовтня 2000 року та вводила багато нових обширних функцій, наприклад збирач «сі з'явилась можливість до «спискового» включення – функція, запозичена із функціональних мов програмування SE Python для цієї версії аналогічний на синтаксису мови програмування Haskell, за винятком того, що в Haskell відд ключові слова. Також у Python версії 2.0 було доповнено систему складання сміття з підтримкою циклічних поси Ліцензія для клієнтів, була зареєстрована під назвою «Python Software Foundation License». Починаючи з бета 1 організації Python Software Foundation, створеній у березні 2001 року. Цей реліз вписував зміну специфікацію моє областю видимості. Ця можливість була включена за замовчуванням. Визначним нововведенням у Python версії користувачем, в одній ієрархії. Це перетворило мову програмування Python у повністю об'єктно-орієнтовану мову. це функція для керування ітераційною поведінкою циклу. У грудні 2014 року було проголошено про те, що Pythor зміна прогресу розвитку мови, переходом до більш прозорого процесу творення, з цього и почався розвиток мови пї метою усунення фундаментальних проблем в мові. Такі зміни не могли бути розроблені за умови збереження пов Python 3 було: «скорочення функціональності, що дублюється, а також усуненням застарілих способів зробити це» параметрів і результату функцій; 2. Повний перехід на Unicode для рядків; 3. Введення нового типу так званих «н представляти двійкові дані; 4. Нова підсистема вводу-виводу, яка має окремі подання для двійкових і текстових да- типів чисел; 7. Вирази для словників і множин; 8. Зміни для друку на вбудовану функцію, що дозволить модулю «prі також спростить код; 9. Переміщення скорочення від вбудованого простору до модуля «functools; 10. Новий синт мова програмування Python дуже добре підходить для початківців, а також для досвідчених програмістів, які вик особливістю цієї мови програмування – це простота використання конструкцій для створення програми, завдяки

можливість використання спеціальних модулів «TensorFlow» та його доповнення модуль «Keras». Мова програмування свою чергу модуль matplotlib – це велика бібліотека для створення статичних, анімованих та інтерактивних візуалізацій, описати прогнозуючи дані та співвідносити їх зі фактичними[21]. Для форматування даних було використано модуль представлення табличних даних, у цьому форматі надаються вхідні дані для тренування нейронної мережі, тому роботи зі багатовимірними масивами (матрицями) було використано плагін «NumPy». Бібліотека NumPy надає функції та операторів, оптимізованих для роботи з багатовимірними масивами. Внаслідок будь-який алгоритм послідовності операцій над масивами або матрицями і реалізований з використанням бібліотеки NumPy, працює та Отже, було обрано мову програмування Python, оскільки ця мова має можливість використання конструкцій з matplotlib

Рисунок 2.1 – Індекс популярності мов програмування по «PYPL» Рисунок 2.2 – Логотип мови програмування Python, яка надає засоби для аналізу коду, графічний нагляд інтегроване середовище розробки мови програмування Python, яка надає засоби для аналізу коду, графічний нагляд підтримує веб-розробку на Django. Це середовище розробки має добре налагоджену файлову систему, а також роботи програми, що є невід'ємною частотною у розробці нейронних мереж на мові програмування Python. Можливість змінення внутрішньої структури програми, який надає широкі можливості щодо виконання швидких глобальних змін контролю за версіями; 3. Можливість автодоповнення коду; 4. Надає можливість користувачам писати свої плагіни

Гнучкий перегляд документації для будь-якого елемента у вікні редактора, перегляд документації через браузер, підтримка та[17]. Середовище розробки PyCharm надає можливість працювати з ноутбуками «Jupyter», запускати команди в інтер'єрі працювати з іншими бібліотеками для наукових обчислень та аналізу даних, включаючи Matplotlib та NumPy. PyCharm мови програмування Python для створення конкуренції з PyDev та більш поширеного середовища розробки K PyCharm версії 1.0 була випущена місяцями пізніше. Версія 2.0 вийшла 13 грудня 2011 року, версія 3.0 була випущена підписну модель ліцензування, а разом із цим змінилася і нумерація версій[18]. Рисунок 2.3 – Логотип середовища графічний фреймворк для мови програмування Python, виконаний як розширення Python, що має відкритим вихідним бібліотек C++ та інструментів розробки, який включає незалежні від платформи абстракції для графічних інтерфейсів «Computing»[19]. PyQt працює зі кроссплатформним середовищем для розробки графічних інтерфейсів «інтерфейси користувача за допомогою ряду інструментів, що спрощує створення інтерфейсу програми[20]. Рисунок 2.4 – Крос-платформна документно орієнтована система управління базами даних, яка не описує схеми таблиць. Вважається, що зберігає документи – документи можуть зберігати складну структуру інформації, цей документ можна поділити, система має три головні характеристики:[24] 1. Швидка технологія, тобто NoSQL бази дані не потребують того ж NoSQL – системи мають можливість зберігання великих обсягів неструктурованої інформації. Завдяки нереляційній необхідності, в процесі створення запитів можна додавати нові типи даних; 3. NoSQL – системи зберігають дані в хмару, де і працює[26]. Система управління MongoDB написана на мові C++, тому має можливість швидко зберігати складні структури дані, також цю систему можна портувати на різні платформи. MongoDB над Solaris.. Функціональність NoSQL-системи MongoDB дозволяє розташувати декілька баз даних на кількох фізичних серверах, завдяки запитів та зберігати цілісність. MongoDB може працювати з механізмом синхронізації вмісту кількох більше копії даних різних вузлах. Кожен екземпляр такої репліки може виступати у ролі основної чи допоміжної читання за замовчуванням здійснюються з основною реплікою. А вже допоміжні репліки підтримують копію даних набір реплік проводить автовибір, де вибирається яка з реплік повинна стати основною. Інші репліки можуть додатково надає СУБД MongoDB – це можливість працювати відповідно до парадигми «MapReduce», тобто розподіляти великим наборами даних. Для агрегації даних передбачено аналог SQL-виразу GROUP BY; оператори агрегації мають структуру перенаправлення введення-виведення: те, що виводить на потік стандартного виводу попередній процес

Технологія «GridFS» – це технологія для зберігання та вилучення файлів, розмір яких перевищує 16 МБ. Замість зберігати GridFS роздроблює файл на частини або фрагменти і зберігає кожен такий фрагмент в окремому документі. За замовчуванням, тобто GridFS ділить файл на фрагменти по 255 КБ, крім останнього фрагмента. Останній шматок має розмір збільше не перевищує розмір фрагмента, мають тільки останній фрагмент. Коли створюється запит технологія GridFS підтримує діапазону до файлів, які зберігаються в GridFS. Також можна одержати доступ до інформації з довільних розділів файлів таблиць використовує колекції, тому якщо в реляційних баз даних таблиці зберігають однотипні жорстко структуровані дані, мають різну структуру і різний набір властивостей[25]. Особливості NoSQL-системи MongoDB: 1. Хмарне зберігання дві колекції; 3. СУБД MongoDB здійснює пошук за запитом, тобто користувач може створити діапазонний запит та мовою написана на мові C++; 4. MongoDB використовує формат зберігання інформації BSON; 5. СУБД MongoDB намагається навантаження між різними базами даних. При цьому бази даних, розташовані на різних вузлах, синхронізовані між собою

2.5 – Логотип системи управління базами даних MongoDB 2.1.5 Технологія TensorFlow Технологія TensorFlow – це технологія навчання, тобто технологія TensorFlow використовується для вирішення завдань побудови та тренування нейронних мереж обчислення за допомогою спеціального графа – граф обчислень, вузли якого відповідають операціям чи змінним.

передавати свої результати інші операції. Таким чином, кожен вузол у графі визначає функцію змінних. Такий граф корисно в розробки алгоритмів машинного навчання, а також під час вирішення різних оптимізаційних завдань. Тепер для цього метод зворотного обчислення похідних, що є набором методів для оцінки похідної функції, заданою комп'ютерна програма, якою б складною вона не була, виконує послідовність елементарних арифметичних операцій функцій (\log , \sin , \cos і ін.). При багаторазовому застосуванні ланцюгового правила до цих операцій похідні довільної точності та з використанням не більше ніж на невеликий постійний коефіцієнт більше арифметичних операцій, ніж потік тензорів, тут це фактично означає потік обчислень з багатовимірними матрицями, тобто тензори виступають аналогії з масивами NumPy, де основним об'єктом є однорідний багатовимірний масив, що представляє собою таблицю чисел. У таких масивів вимірювання називаються осями. Тензори мають особливість «Broadcasting», Broadc Концепція Broadcasting – обробляє масиви різної форми під час арифметичних операцій. З урахуванням певних мали сумісні форми. Мовлення надає засоби векторизації операцій з масивами, так що за циклювання відбувається програмування Python. Така концепція робить це без створення непотрібних копій даних та зазвичай призводить TensorFlow – це рекомендований тип об'єкта, що представляє загальний та постійний стан, яким можна керувати. Маніпуляція відноситься до будь-якої зміни значення чи параметра. Ця характеристика є найбільш характерною рідкової закритої системи машинного навчання «DistBelief», яка розроблялася компанією «Google Brain» для внутрішніх навчання. Ця машинна навчання стала використовуватися у багатьох дослідницьких та комерційних проектах групи DistBelief, фірма Google вирішила вивести проект на нову ступінь розвитку, і для рефакторингу виділила групу з інформатики та програміст Джефф Дін, метою групи було спрощення та оптимізація кодів бібліотеки, збільшення назву TensorFlow. У 2013 році до цього проекту приєднався британський та канадський вчений математик, кібернетик, перетворив метод узагальненого зворотного розповсюдження помилки та ряд інших поліпшень, що до TensorFlow з 8 листопада 2015 році стала програмним забезпеченням з відкритим вихідним кодом. Технологія TensorFlow – це наступне покоління. У той час як перше покоління працює на одиничних пристроях, система другого покоління може працювати, спираючись на архітектуру CUDA (програмно-апаратна архітектура паралельних обчислень, що дозволяє суттєво прискорити графічних процесорів фірми Nvidia.) для підтримки обчислень загального призначення на графічних процесорах технології другого покоління над першим. Технологія TensorFlow підтримує кросплатформність, тобто доступ до обчислювальних платформ, включаючи Android та iOS. У травні 2016 року Google повідомила про застосування розробки – тензорного процесора (TPU). Тензорний процесор (TPU) – процесор, що відноситься до класу нейронної роботи алгоритмів штучних нейронних мереж, що є спеціалізованою інтегральною схемою, розробленою компанією для навчання TensorFlow. Цей процесор представлений у липні 2016 році на річній конференції, яка орієнтована на розвиток відкритих технологій та сервісів Google. Порівняно з графічними процесорами (GPU), наприклад компактності більш високої продуктивності на ват і відсутності модуля для растрезації і текстурних блоків. Також як «Google Street View» (функція Google Maps та Google Earth, що дозволяє дивитися види вулиць багатьох міст) повідомлялося, що весь обсяг оброблений менш ніж за чотири дні. У Google Фото один тензорний процесор може застосовується для системи самонавчання RankBrain, яка підтримує обробку результатів пошуку Google та використання тензорного процесору у власних завдань, компанії Google з обробки даних вдалося досягти на п'ять разів швидше. Рисунок 2.6 – Логотип TensorFlow Отже, технологія TensorFlow – це використання тензорів (матриці), завдяки яким зв'язок. [16] Фреймворк «Keras» Фреймворк – це спеціальна програмна платформа, яка має можливість визначити спрощує розробку та поєднання різних компонентів великого за розміром програмний проект. Фреймворк відрізняється в програмному продукті як набір близької функціональності підпрограм, що не впливає на архітектуру програмного фреймворк диктує правила побудови архітектури додатка, задаючи на початковому етапі розробки поведінку за умов зазначених вимог. Фреймворк Keras – це фреймворк, який написаний мовою програмування Python і забезпечує роботу зі TensorFlow, оскільки цей фреймворк є надбудовою над технологією TensorFlow. Keras є найпопулярнішим (Tensorflow), згідно рисунку 2.6. Фреймворк Keras націлений на оперативну роботу з мережами глибокого навчання: ONEIROS, засновником є Франсуа Шолл, інженер Google. На початку створення фреймворка Keras, планувалося TensorFlow, проте Шолле виділив Keras в окрему надбудову, оскільки згідно з концепцією Keras є скоріше інтерфейс високорівневий API TensorFlow : доступний, високопродуктивний інтерфейс для вирішення задач машинного навчання будівельні блоки для розробки та доставки рішень машинного навчання із високою швидкістю ітерації. Keras переважає масштабованості та кросплатформових можливостей TensorFlow. Keras дає можливість запускати експортувати свої моделі Keras для запуску у браузері або на мобільному пристрої. Використовуючи API Keras – це швидше. Фреймворк Keras надає можливості: 1. Зниження когнітивного навантаження на розробників, дозволяючи використовувати принцип поступового розкриття складності, тобто спрощує початок роботи, але дозволяє обробляти поступово навчання кожному етапі; 3. Забезпечення найкращу в галузі продуктивність і масштабованість, ві

YouTube або Waymo[15]. Рисунок 2.7 – Популярність використання фреймворків для нейронних мереж. Отже, для використання фреймворку Keras.

2.2 Розробка архітектури програми

Архітектура програми – це абстракція, яка залежить від дій користувача, то архітектуру програми можна представити діаграмою станів. Розроблювальна користувачів, яка використовується програмою при авторизації користувача, тобто повертає звіт чи зареєстрований станів (Авторизація). Після проходження авторизації, користувач «переноситься» на наступне вікно, де вже може користуватися програмою (Після проходження авторизації). Користувач завантажує дані для тренування нейронної мережі, після чого, мережа перенавчається для кращих прогнозуючих значень, після чого вагові коефіцієнти зберігаються. На основі цих значень на огляд користувача.

Створення архітектури нейронної мережі

Для створення нейронної мережі архітектура рекурентних нейронних мереж та архітектура згорткових нейронних мереж. Важливою особливістю при цьому є те, що кожен батч вхідних даних обробляється незалежно, без збереження стану між ними, не основу коректно. Для врахування цієї особливості, було взято архітектури рекурентних нейронних мереж та згорткових нейронних мереж по чергу послідовно з використанням інформації, отриманої при обробці попередніх його елементів. Щоб переконатися в тому, що архітектура рекурентних нейронних мереж (а також щоб краще розпізнати закономірності, що містяться в ньому) нейронною мережею, було проаналізовано роботу прогнозування, мережі повинні містити пам'ять про попередні значення часового ряду, щоб знайти ознаки, які дані представляють собою послідовність, у якій важливі причина і наслідок. Для рішення задач пошуку законів у рекурентній нейронній мережі, але у разі великої кількості даних робота архітектури потребує велику кількість обчислень. Для згорткових нейронних мереж, які роблять їх ідеальним вибором для розпізнавання образів так просторовий вимір, подібно до висоти або ширини двовимірного зображення. Така згорткова нейронна мережа вимагає великої кількості пам'яті, але якщо об'єднати швидкість і легкість згорткових мереж з чутливістю до порядку попередньої обробки даних перед передачею їх у рекурентну мережу, то така модель нейронної мережі отримує значення, згорткова частина обробки даних перетворює вхідну послідовність на більш коротку послідовність високої частоти, подається на вхід рекурентної частини нейронної мережі. Така модель нейронної мережі показана на рисунку 2.7, дозволяє обробляти великі обсяги даних, відносно великої швидкості, але у підходящій. Для вирішення цієї проблеми було вирішено створити окрему архітектуру нейронної мережі, основною властивістю якої є використання декількох слоїв рекурентних нейронних мереж, це дозволить побороти проблему перенавчання та показана на рисунку 2.8[7][8].

2.3 Проектування структури бази даних та визначення необхідних обмежень цілісності.

Існують чотири основні задачі проектування: 1. Забезпечення доступності даних за всіма необхідними запитами; 2. Скорочення надмірності та дублювання даних; 3. Необхідним етапом розробки є проектування бази даних, яка забезпечить зберігання тренувальних даних, даних користувачів та даних до авторизації користувача повертає звіт зі значення «false», а у разі успішної авторизації повертає звіт про успішну авторизацію користувача. На основі цих даних прогноуються значення, які зберігаються у колекції «prognoz_data»[27].

2.12 – Колекція Unified Modeling Language (UML) – уніфікована мова моделювання.

Розшифруємо: modeling має на увазі стиль підходить для широкого класу проєктованих програмних систем, різних галузей додатків, типів організацій, рівнів створений для визначення, візуалізації, проєктування та документування, програмних систем. На основі UML-діаграми – це можливість розробникам програмного забезпечення добитися угоди у графічних позначеннях для представлення програмних систем. На рисунку 2.12 зображено як виконується взаємодія користувача та програмою. Користувач авторизується та починає використання нейронної мережі. В сучасному світі використання нейронних мереж тільки набирає обертів, особливо в галузі використання наукових методів, процесів, алгоритмів та систем для отримання знань та ідей із шумних, структурованих даних широкому діапазоні областей застосування. Зокрема, використання нейронних мереж для прогнозування часових рядів. Наприклад, програму було використано для прогнозування фінансових рядів для подальшого аналізу та створення сценаріїв прогнозування не тільки фінансових рядів, а й для прогнозування інших типів часових рядів, наприклад для прогнозування енергетики – ціни, попит, виробничі графіки; 2. Роздрібна торгівля – продажі, споживчий попит на певні товари; 3. Аналіз даних потрібно багато часу, але за допомогою штучних мереж можливо в разі прискорити аналіз даних, згідно значень.

2.6 Висновок до розділу 2

1. Проаналізувавши базові технології розробки програм було визначено, що для програмування Python, СУБД MongoDB, інтерфейс PyQt, технологія TensorFlow; 2. Побудовано архітектур

структуру бази даних; 4. Побудовано UML- діаграми для створення програми. 2 РОЗДІЛ 3 РОЗРОБКА ПРОГРАМИ

інтерфейс програми для простої взаємодії користувача зі можливостями програми. Для розробки інтерфейсу було чергу середовище розробки інтерфейсу – «Qt Designer». Для того щоб користувач почав працювати зі прог «authorization_interface» у середовищі розробки «Qt Designer» – рисунок 3.1. Рисунок 3.1 – Вхідне вікно «au «Войти», або «Регистрация» – дані які були надані у полях «LineEdit» почнуть зіставлятися зі даним бази даних, у р вікно, у разі некоретних даних з'явиться модальне вікно відповідно помилки рисунок 3.2. Рисунок 3.2 – Модальне ві головне вікно рисунок 3.3, де вже має можливість працювати зі програмою. Рисунок 3.3 – Головне вікно програм «Начать тренировку нейронной сети», «Начать прогнозирование». При натисканні кнопці «Загрузить данные» – кори Ці дані будуть відформовуванні та відсортовуванні перед наданням до нейронної мережі, на рисунку 3.4 надано при чого користувач може натиснути на кнопку «Начать тренировку нейронной сети», де ці дані вже підуть на вхід нейр на екрані з'явиться «ProgressBar» – повзунок готовності тренування перед прогнозуванням. І вже після тренув тренування нейронної мережі у вкладці «Оценка качества тренировки» – рисунок 3.6. Рисунок 3.5 – Оцінка якості «Mean squared error» – середньоквадратичну помилку, тобто середньоквадратичну різницю між оцінними використовувати функцію втрат. Після виконання всіх цих процедур (надання даних та їх тренування) – користувач основі тренувальних даних, тобто вагових коефіцієнтів, нейронна мережа починає етап прогнозування. Після прогнс вкладці «Прогнозируемые значение» (рисунок 3.6), що надає змогу користувачу проаналізувати ці дані та створит Розробка нейронної мережі При розробці нейронної мережі для прогнозування потрібно дотримуватися таких (форматування); 2. Нормалізація цих даних (від -1 до 1); 3. Крос-валідація даних, тобто розподіл даних для тре тренування надає користувач, при форматування даних зберігаються два стовпця: остаточна ціна за день та об'єм вагових коефіцієнтів нейронної мережі. Нормалізація даних відбувається за зміненою формулою 1.7. А саме відраховується один, таким чином виконується стандартизація, код з виконанням нормалізації наведено на лістині window_data, single_window=False): normalised_data = [] window_data=[window_data] if single normalised_window=[] for col_i in range(window1.shape[1]): normalised_col = [(float(p) normalised_window.append(normalised_col) normalised_window=np.array(normalised_window).T np.array(normalised_data) У свою чергу для оптимізації нейронної мережі було обрано алгоритм «Adam», сам мережі. Після цього потрібно розділити дані на тренувальні та на дані для тесту, тобто крос-валідація даних. Для і даних це дані для тесту, які не будуть брати участі у навчанні. Крос-валідацію даних у коді наведено на ліст int(len(df)*split) cols=["Close", "Volume"] data_train = df.get(cols).values[:i_split] data_test = df.get len(data_test) Коли дані будуть відформатовані згідно трьох парадигм, можна почати процедуру створення моде нейронної мережі згідно рекурентної архітектури (рисунок 2.11). Для створення такої моделі було використано Розгорнута рекурентна архітектура нейронної мережі На рисунку 3.9 представлена розгорнута рекурентна архіт приймає дані та має 49 нейронів, цей вхідний шар працює у парі зі «LSTM» шаром, який також має 49 нейронів, та шар «Dropout», цей шар відповідає за перенавчання, а саме для його запобігання. Після шару шар «Dropout», йде чого нейрони підвергаються регуляризації шаром «Dropout». Далі дані поступають до лінійної функції активації, піс для тесту, у разі не співпадіння – змінюються вагові коефіцієнти, і знов зіставляються зі даними для тесту, 1 Тестування програми Для того щоб протестувати коректність функціоналу програми – було прийнято рішення кої можливих дій у програмі для виявлення помилок які не оброблені у коді. Під час тестування було виявлено декіль нейронної мережі, для вирішення цієї помилки потрібно проводити інвентаризацію даних алгоритмом, таким чином нейронної мережі, а саме не цілісна нормалізація даних, що приводить до неточності при прогнозування, для відсоткове співвідношення. Всі ці зміни будуть інтегровані у майбутніх оновленнях програми 3.3 Висновок до розд програми, а саме створено інтерфейс програми та описано його функціонал, спроектовано повну архітектуру ней Також в цьому розділі було проведено тестування програми, де були виявленні помилки, а також можливість їх ус спеціаліста, мною було досягнуто поставлених цілей та задач, а саме було проведено аналіз літератури, що напра статей щодо розробки та використання нейронних мереж, зокрема, використання нейронних мереж для прогнозув: попит використання нейронних мереж. Головною метою виконання випускної роботи молодшого спеціаліста проаналізувати існуючі нейронні мережі для подальшого аналізу цих рядів для створення обліку, а також використа мета була досягнена. При написання випускної роботи молодшого спеціаліста були виконані наступні завдання: огляд мов програмування, середовищ розробки, технологій для створення нейронної мережі, видів систем управ Протестовано різні моделі нейронних мереж у програмі, завдяки чому було виявлено найпідходящу модель нейро виявлення помилок; 5. Розроблено програму на тему випускної роботи молодшого спеціаліста та зроблено висноє ресурс]. Метод доступу: www. URL: [https://ru.wikipedia.org/wiki/Временной ряд](https://ru.wikipedia.org/wiki/Временной_ряд) – 02.06.2022 2. Ча <https://habr.com/ru/post/553658/> – 02.06.2022 3. Книга про нейронні мережі та часові ряди [Електронний рек

id=40 – 01.06.2022 4. Актуальність нейронних мереж [Електронний ресурс]. Метод доступу: www. URL: <https://bi>
Стаття про нейронну мережу [Електронний ресурс]. Метод доступу: www. URL: <https://ru.wikipedia.org/wiki/т>
значення від часу [Електронний ресурс]. URL: https://www.researchgate.net/publication/267555543_Comparative_analysis_of_Recurrent_and_Finite_Ir
– 31.05.2022 7. Книга про нейронні мережі [Електронний ресурс]. Метод доступу: www. URL: <https://codernet.ru/b>
– 20.05.2022 8. Застосування глибоких нейронних мереж для короткострокового прогно: <https://zenodo.org/record/3253019#.YqM-PmhVxD8> – 23.05.2022 9. Програма аналог GMDH [Електронний ресурс]. Метод доступу: www. URL: <https://try.n>
[Електронний ресурс]. Метод доступу: www. URL: <https://quantpro.ru/archives/4865> – 02.06.2022 12. Про мову Python [Електронний ресурс]. Метод доступу: www. URL: <https://www.python.org/> – 03.06.2022 13. Про мову програмування Python та нейронні мережі [Електронний ресурс]. Метод доступу: www. URL: <https://www.tensorflow.org/> – 20.05.2022 17. Про середовище розробки PyCharm [Електронний ресурс]. Метод доступу: www. URL: <https://riverbankcc.com/> – 29.05.2022 18. Про середовище розробки PyQt [Електронний ресурс]. Метод доступу: www. URL: <https://ru.wikipedia.org/wiki/PyQt> – 29.05.2022 18. Про середовище розробки інтерфейсів PyQt [Електронний ресурс]. Метод доступу: www. URL: <https://matplotlib.org/> – 23.05.2022 22. Про модуль NumPy [Електронний ресурс]. Метод доступу: www. URL: <https://pythonworld.ru/moduli/modul-csv.html> – 26.05.2022 24. Про мову програмування Python [Електронний ресурс]. Метод доступу: www. URL: <https://www.mongodb.com/> – 26.05.2022 25. Про СУБД MongoDB [Електронний ресурс]. Метод доступу: www. URL: <https://ru.wikipedia.org/wiki/NoSQL> – 26.05.2022 26. Про NoSQL СУБД [Електронний ресурс]. Метод доступу: www. URL: <https://habr.com/ru/post/514364/> – 27.05.2022 28. Про UML-діаграми [Електронний ресурс]. Метод доступу: www. URL: https://www.zieit.edu.ua/?page_id=54
29.05.2022 2 ДОДАТОК А ЛІСТИНГ КОДУ

Source

Фаховий коледж – Запорізький інститут економіки та ...

0.20%

Фаховий коледж – Запорізький інститут економіки та ...

https://www.zieit.edu.ua/?page_id=54

3 ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І Т...

0.40%

3 ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ
IDT ± identity

<https://lib.chmnu.edu.ua/pdf/metodser/197/2.pdf/>

Парсер контенту: Використання парсингу XML(YML) файлів. Як ...

0.20%

Парсер контенту: Використання парсингу XML(YML) файлів. Як ...

<https://sukachoff.ru/uk/virusy/parser-kontenta-vnedrenie-parsinga-xml-yml-failo-v-kak-vygruzit-tovary-v/>

Нейронні мережі опис. Штучний інтелект

0.20%

Нейронні мережі опис. Штучний інтелект

<https://lab-music.ru/uk/neironnye-seti-opisanie-iskusstvennyi-intellekt-odnosloinye-neironnye/>

by IC Куць · 2019 — У 1957 році Ф. Розенблатт запропонував комп'ютерну ...

0.20%

by IC Куць · 2019 — У 1957 році Ф. Розенблатт запропонував комп'ютерну модель мозку, яка ... одношаровий перцептрон і демонструє його здатність вирішувати завдання класифікації ...

http://dspace.wunu.edu.ua/bitstream/316497/38264/1/%D0%9A%D1%83%D1%86%D1%8C_%D0%9C%D0%A0_2019.pdf

Гرادієнтний метод. Огляд градієнтних методів у задачах ...

0.20%

Градієнтний метод. Огляд градієнтних методів у задачах ...

<https://kerchtt.ru/uk/gradientnyi-metod-obzor-gradientnyh-metodov-v-zadachah-matematicheskoi/>

2018 — У версії Python 2.0 з'явилися спискові включення–функція, запози...

0.20%

2018 — У версії Python 2.0 з'явилися спискові включення – функція, запозичена з функціональних мов програмування SETL і Haskell. Синтаксис в Python для цієї.

https://fmft.vspu.edu.ua/wp-content/uploads/2019/01/%D0%90%D0%BB%D1%8C%D0%BC%D0%B0%D0%BD%D0%B0%D1%85_%D0%B2%D0%B8%D0%BF%D1%83%D1%81%D0%BA-2_2018_.pdf



[Home](#)

[Blog](#)

[Testimonials](#)

[About Us](#)

[Privacy](#)

Copyright © 2022 Plagiarism Detector. All right reserved