КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет комп'ютерних наук та кібернетики
Кафедра теоретичної кібернетики

Кваліфікаційна робота

на здобуття ступеня бакалавра

за спеціальністю 122 Комп'ютерні науки

на тему:

ТЕЛЕГРАМ БОТ ДЛЯ ВИЗНАЧЕННЯ ЕМОЦІЙНОГО СТАНУ ЛЮДИНИ

Виконав студент 4-го курсу Кобзар Данило Миколайович (підпис)

Науковий керівник:

професор, доктор фіз.-мат. наук Пашко Анатолій Олексійович (підпис)

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

Студент _____
(підпис)

Роботу розглянуто й допущено до захисту на засіданні кафедри теоретичної кібернетики
«____»_____2021 р., протокол № _____
Завідувач кафедри

Ю. В. Крак _____
(підпис)

Київ – 2020

РЕФЕРАТ

Обсяг роботи 54 сторінки, 20 ілюстрацій, 35 джерел посилань.

НЕЙРОННА МЕРЕЖА, ЗГОРТКОВА НЕЙРОМЕРЕЖА, ЗГОРТКОВА НЕЙРОМЕРЕЖА, ВЕБ-СЕРВЕР, TELEGRAM API

Об’єктом роботи є телеграм бот, що за допомогою глибокого навчання визначає емоції людини за фотографією.

Предметом роботи є згорткова нейронна мережа та чат-бот в месенджері Телеграм.

Метою роботи є дослідити ефективність згорткових нейронних мереж у сфері розпізнавання емоцій та розробка телеграм боту.

Методи розроблення: глибоке навчання, згорткова нейронна мережа, метод оберненого поширення помилки, об'єктно орієнтоване програмування, веб-програмування.

Інструменти розроблення: TensorFlow, OpenCV, мова програмування Java та Python 3, фреймворк для веб-розробки Spring Framework.

Результати роботи: виконано загальний огляд глибокого навчання та згорткових нейронних мереж. Описаний алгоритм градієнтного спуску та зворотного поширення помилки. Розроблений веб-сервер телеграм боту.

Отриманий продукт не втрачає актуальність та може бути використаний для подальшого розвитку емоціонального інтелекту.

ЗМІСТ

ВСТУП

Оцінка сучасного стану об’єкта дослідження. Сучасні засоби комунікації стрімко розвиваються, набувають поширення такі засоби комунікації як месенджери та чат-боти. Чат-боти на сьогодні є складовою кожного бізнесу та зручним засобом розсилки реклами, новин, контролю та нагадуванню підписки на сервіс. В деяких компаніях, чат-бот є основним засобом взаємодії з користувачем. Також чат-боти можуть виконувати навчальну роль: проводити опитування та оцінювати знання, розвивати емоційний інтелект. Розвитком емоційного інтелекту, може слугувати чат-бот для розпізнавання емоцій людини.

Актуальність роботи та підстави для її виконання. Актуальність бакалаврської роботи зумовлена високою популярністю чат-ботів та їх звичним інтерфейсом. Чат-боти дозволяють спростити щоденні рутинні завдання, такі як отримання інформації про погоду, пробки, останні новини та інші. В цій роботі буде розроблятися чат-бот для розпізнавання емоцій людини. Вивчення теми розпізнавання емоцій людини може широко використовуватися в тих галузях, де потрібно запобігати людській втомі, поганому настрою: контроль втоми у пілотів та водіїв, адаптація процесу навчання за емоціями учня, збирання статистики емоцій клієнтів бізнесу та забезпечення правильного зворотного зв’язку. Телеграм бот для визначення емоцій людини може бути використаний для розвитку емоційного інтелекту.

Мета й завдання роботи. Метою роботи є взаємодія з користувачем телеграм через чат-бот, реалізація зручного інтерфейсу, обробка вхідних зображень від користувача та їх аналіз на емоції. Для досягнення мети поставлено та виконано такі завдання:

Дослідити поняття нейронних мереж.

Дослідити згорткові нейронні мережі для обробки зображень.

Дослідити предметну область чат-ботів.

Розробити архітектуру системи.

Розробити власного чат-бота для визначення емоцій людини за фотографією.

Об’єкт, методи й засоби розроблення. Об’єктом роботи є телеграм бот, що за допомогою глибокого навчання визначає емоції людини за фотографією. Методи розроблення є глибоке навчання, згорткова нейронна мережа, метод оберненого поширення помилки, об’єктно орієнтоване програмування, веб програмування. Засобами розроблення є TensorFlow, OpenCV, мова програмування Java та Python 3, фреймворк для веб розробки Spring Framework.

Можливі сфери застосування. Отриманий продукт не втрачає актуальність та може бути використаний для подальшого розвитку емоціонального інтелекту.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Месенджери як засіб взаємодії з сторонніми API сервісами

Сучасні технології стрімко розвиваються, кожного дня з’являються ідеї та стартапи в IT галузі, а вже існуючі намагаються адаптуватися, щоб йти в крок з часом та робити продукти свого бізнесу доступними для більшої аудиторії за допомогою мережі Інтернет, месенджер каналів, сторінок у соціальних мережах.

IT галузь також стрімко розвивається, кожен процес намагаються оптимізувати і якщо раніше існували “гарячі лінії”, де клієнти могли задавати свої питання і на підтримку лінії повинен був виділятися додатковий бюджет, то тепер стають популярними чат — боти, які вимагають бюджет тільки на їх розробку і підтримку, як IT продукт, здатні витримувати набагато більший клієнт-потік.

Чат-бот – це спеціалізований додаток, що дозволяє користувачам взаємодіяти зі сторонніми сервісами через зручний інтерфейс чату.

Найбільшу популярність чат-боти отримали, коли почалося їх використання в месенджерах і соціальних мережах (наприклад, в Telegram, Viber, Facebook).[1] Завдяки новому інструменту віртуальної комунікації з'явилася можливість дізнаватися про спеціальні пропозиції онлайн, отримувати посилання свіжих новин та спеціальних пропозицій у сфері товарів і послуг, здійснювати складні операції. Нині найбільше значення чат-боти мають у формуванні такої моделі поведінки, яка буде максимально наближена до людської. Через те, що контакт з чат-ботами є не тільки цікавим заняттям, але і корисним для людини з точки зору отримання нової інформації і рішення споживчих завдань, варто відзначити, що інтернет-аудиторія прийняла їх появу доброзичливо.

Чат-бот спілкується з користувачами через повідомлення і має множину певних функцій.

Тобто, можна отримати певну інформацію, написавши чат-боту спеціальну команду, яку останній інтерпретує певним чином.

Чат-боти можуть взаємодіяти з користувачем, відповідати на обмежену множину питань, надсилати регулярно рекламу або новини. Чат бот як будь-який сервіс можна інтегрувати в e-commerce платформи, свою архітектуру сервісів як сервіс по опитуванню та збору інформації о користувачах.[2] Чат-бот це помічник для будь-якого бізнесу, інтернет-магазину, банку, який технічно не є складною задачею, але приносить багато користі.

Такі боти дозволяють збільшити прибуток компаній, оскільки бот може обробляти заявки з більшою швидкістю, ніж людина, і зменшити навантаження на робочий персонал.

Telegram – це месенджер, що дозволяє обмінюватися повідомленнями і файлами (аудіо-, відео-, архівами тощо). Серверний код месенджеру є закритим та працює на серверах Німеччини і США.[3]

Telegram має ряд переваг:

- конфіденційність – чати зберігаються в зашифрованому вигляді
- швидкість – швидкість доставлення повідомлень вище, ніж у аналогів;
- відсутність підписок і реклами;

Розпізнавання емоцій людини

Обличчя добре вивчене як джерело інформації про емоції.[4] Закономірності рухів м’язів обличчя були визначені як мінімум для шести основних емоцій - щастя, смутку, гніву, страху, здивування та огиди. Ці моделі виробляють дуже послідовне розпізнавання емоцій у багатьох культурах світу. Це, у поєднанні з даними приматів та немовлят, передбачає біологічну основу для емоційного вираження та розпізнавання. Хоча міміка часто виявляє відчуття емоцій, вони також використовуються для соціальних сигналів. Вирази обличчя часто використовують як повідомлення про емоції заохочення, співпереживання або злості; вони також допомагають регулювати хід розмови в діалозі. Рухи очей сигналізують про увагу; у дружній взаємодії більш тривалий зоровий контакт означає більш позитивне ставлення. Більш зосереджений погляд, усмішка та більш природне положення тіла - все це передає психологічний комфорт та доброзичливість.

Основні напрямки використання розпізнавання емоцій

Розпізнавання емоцій є життєво важливим для спілкування людини з людиною, і для досягнення повної взаємодії між людьми та машинами необхідно враховувати розпізнавання емоцій. У наш час все більше і більше інтелектуальних систем використовують моделі розпізнавання емоцій, щоб поліпшити свою взаємодію з людьми. Це важливо, оскільки системи можуть адаптувати свої реакції та моделі поведінки відповідно до емоцій людей та зробити взаємодію більш природною. Одним із застосувань можна знайти в системі автоматичного навчання, де система регулює рівень навчального посібника залежно від афективного стану користувача, такого як хвилювання чи нудьга. В іншому прикладі застосування: стан втоми під час керування транспортним засобом та попередження водія. Також актуальним є розпізнавання мови, як, наприклад в кол-центрах.[5] Метою є успішне прогнозування афективного стану абонента або агента та надання зворотного зв'язку щодо якості послуги.

Нейронні мережі

Нейронні мережі складаються з простих блоків обробки, нейронів та спрямованих та збалансованих зв’язків між цими нейронами. Зв'язок між двома нейронами (або сполучною вагою) і і j називається w_{ij}. [6]

Нейронна мережа - це відсортована трійка (N, V, w) з двома множинами N, V і функцією w, де N - набір нейронів і V множина {(i, j) | i, j ∈ N}, елементами якої є зв’язки між нейроном i та нейроном j. Функція w: V → R визначає вагу, де w ((i, j)), вага зв'язку між нейроном i та нейроном j, скорочується до w_{ij}. Отже, масштабування може бути реалізоване як квадратна зважувальна матриця W або, як варіант, у ваговому векторі W, номер рядка матриці вказує, де починається з'єднання, а номер стовпця матриці вказує, який нейрон є кінцевим. У цьому випадку нульове значення означає відсутність зв'язку. Це матричне подання також називається графіком Хінтона. Нейрони та з'єднання складаються з наступних компонентів та змінних; [7]

Зв'язки несуть інформацію, що обробляється нейронами. Дані, що передаються між нейронами через з'єднання зі сполучною вагою, є збудливими або гальмівними.

Функція поширення перетворює векторні входи в входи скалярної мережі. Дивлячись на нейрон j, ми зазвичай знаходимо багато нейронів із зв'язком з j, тобто які передають свою продукцію в j. Функція поширення (англ. Propagation function) перетворює векторні входи в скалярні входи мережі. Дивлячись на нейрон j, ми зазвичай знаходимо багато нейронів із зв'язком з j, тобто які передають свій сигнал в j. Для нейрона j функція поширення отримує виходи $o_{i1}, \ldots, o_{in}$ нейронів $i_1, i_2, \ldots$ (які підключені до j), і перетворює їх з урахуванням ваг з’єднань w_{ij} у вхід мережі netj, який може бути додатково оброблений функцією активації. Таким чином, вхід в мережу є результатом функція поширення.

Функція розповсюдження та вхід мережі. I = $\{i_1, i_2, \ldots, i_n\}$ - набір нейронів, такі, що $\forall z \in \{1, \ldots, n\}: \exists w_{i_z,j}$. Тоді мережевий вхід j, званий netj,

обчислюється за допомогою функції поширення fprop наступним чином:

netj = fprop(oi1, . . . , oin, wi1,j , . . . , win,j).

Множення виходу кожного нейрона і за wi, j і підсумовування результатів: $net_j = \sum_{i \in I}(o_i * w_{ij})$. Активація - це "зміна стану" нейрона. Стан активації визначає реакцію нейронів на вхідні значення. Стан активації вказує на ступінь активації нейрона і часто коротко називається активацією. Активацію можна визначити наступним чином: нехай j - нейрон, стан активації aj, вказує ступінь активності нейрону та результати від функції активації.

Нейрони активуються, якщо вхід мережі перевищує порогове значення. Близько порогового значення активація функція нейрона реагує особливо чутливо. З біологічної точки зору порогове значення представляє поріг, при якому нейрон починає працювати. Порогове значення також здебільшого включається у визначення функції активації, але загалом визначення є наступне: (Порогове значення загалом). Нехай j - нейрон. Поріг значення Θj однозначно присвоюється j позначає положення максимального значення градієнта функції активації. Функція активації визначає активацію нейрона, що залежить від вхідного та порогового значення мережі. Активація aj нейрона j залежить від попереднього стану активації нейрона та зовнішнього входу. Функція активації : $a_j(t) = f_{act}(net_j(t), a_j(t-1), \Theta_j)$.

На відміну від інших змінних в нейронній мережі, функція активації часто визначається глобально для всіх нейронів або принаймні для набору нейронів і лише порогові значення різні для кожного нейрона. Ми також повинні пам’ятати про те, що порогові значення можна змінювати, наприклад

під навчання нейронної мережі. Функція активації також називається функцією передачі.

Загальні функції активації.

Найпростішою функцією активації є двійкова порогова функція (рис.), яке може приймати лише два значення (також звані функцією Heaviside). Якщо вхід вище певного порогу, функція змінює своє значення, в іншому випадку залишається константою. Це означає, що функція не диференціюється на пороговому значенні та для решти значень похідна дорівнює нулю. Завдяки

цьому навчання завдяки зворотному поширенню є неможливим (як ми побачимо пізніше).

Також дуже популярною є функція Фермі або логістична функція[8]

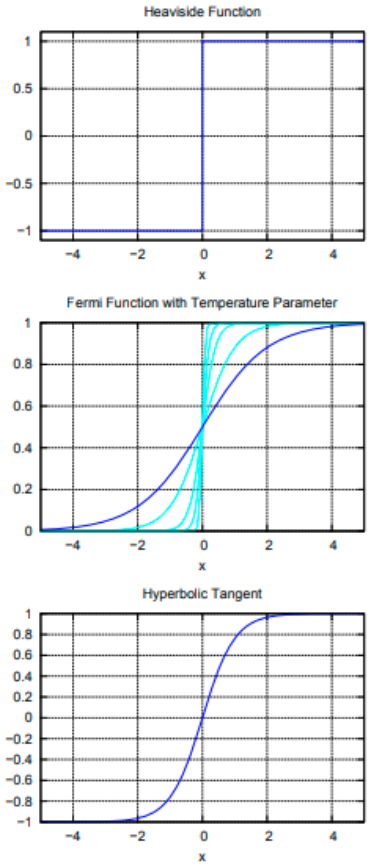

рисунок 1.1 — Функції активації

Нейрон зміщення - це технічна хитрість, щоб розглядати порогові значення як ваги з'єднання. На сьогодні ми знаємо, що в багатьох мережевих парадигмах нейрони мають порогове значення, яке вказує, коли нейрон стає активним. Таким чином, порогове значення є параметром функції активації нейрона. З біологічної точки зору це звучить найбільш правдоподібно, однак ускладнити доступ до функції активації під час виконання для підготовки порогових значень. [9]

Навчання нейронної мережі

Нейронна мережа може навчитися з багатьох речей, але, звичайно, завжди буде питання, як це реалізувати. Нейронна мережа змінюється, коли змінюються її компоненти, як ми дізналися вище. Теоретично, нейронна мережа може навчитися завдяки:

- розвитку нових зв'язків,
- видалення з'єднань, що вже існують
- зміни сполучної ваги,
- зміни порогових значень нейронів,
- варіювання однієї або декількох з трьох функцій нейрону (функції активації, функції поширення та вихідної функції),
- розробка нових нейронів, або видалення нейронів

Як зазначалося вище, ми вважаємо, що зміна ваги є найпоширенішою процедурою. Більш того, можна здійснити видалення з'єднань, заздалегідь визначити, що з'єднання більше не тренується, коли йому встановлено значення нуль. Також ми можемо розвивати подальші з'єднання, встановлюючи нове з'єднання (зі значенням нуль в матриці з'єднання) до значення, відмінного від нуля. Таким чином, ми виконуємо будь-яку з перших чотирьох парадигм навчання, тренуючи синаптичні ваги. Зміну функцій нейронів важко реалізувати, така зміна не є інтуїтивно зрозумілою і не зовсім біологічно мотивована. Можливості розробити або видалити нейрони не лише забезпечують добре відрегульовані ваги під час тренування нейронної мережі, але й оптимізують топологію мережі. З цієї причини, вони викликають інтерес і часто реалізуються за допомогою еволюційних процедур. Нейронна мережа навчається, змінюючи ваги відповідно до правил, шляхом зміни ваги. Процедура навчання - це алгоритм, який легко реалізувати за допомогою мови програмування.

Нехай навчальний набір визначається таким чином:

Набір тренувань (позначається як P) - це набір тренувальних патернів, які ми використовуємо для тренування своєї нейронної мережі.

Навчання без вчителя - це біологічно найбільш вірогідний метод, але він є неефективним для розв'язання деяких проблем. За цим методом мережа намагається виявити подібні закономірності та класифікувати їх за подібними категоріями. Набір тренувань складається лише із шаблонів введення, мережа сама намагається виявити подібність та сформувати класи шаблонів.

Під час навчання з підкріпленням мережа отримує логічне чи дійсне значення після завершення послідовності, яке визначає, правильний чи неправильний був результат. Інтуїтивно зрозуміло, що ця процедура повинна бути більш ефективною, ніж навчання без вчителя, оскільки мережа отримує конкретні критерії для розв'язання проблем. Навчальний набір складається із вхідних даних та даними що характеризують отриманий результат. Прикладом навчання з підкріпленням може бути гра, у якій за правильний хід, нейромережа в нагородження отримує додаткові бали і отримати якомога більше балів в кінці гри.

Методи навчання з учителем забезпечують схеми навчання разом із відповідними правильними результатами. Таким чином, для кожного вхідного навчального набору, є приклад правильного виходу нейромережі. Згідно з порівнянням виходу нейромережі та правильного результату, нейромережа змінює свої вагові коефіцієнти - що і є процесом навчання.

Процедури оптимізації градієнта

Градієнтний спуск та процедура зворотного розповсюдження помилок, що математично основана на градієнтному спуску та успадковує його переваги та недоліки, зазвичай використовуються там, де ми хочемо максимізувати або мінімізувати n-вимірні функції. Ілюстрація (рис.) показує лише два виміри, але принципово немає обмеження кількості вимірів. Градієнт - це вектор g, який визначений для будь-якої точки диференційованої функції, який вказує напрямок від цієї точки до найкрутішого підйому і вказує градієнт у цьому напрямку за допомогою своєї норми | g |. Таким чином, градієнт є узагальненням похідної для багатовимірних функцій.

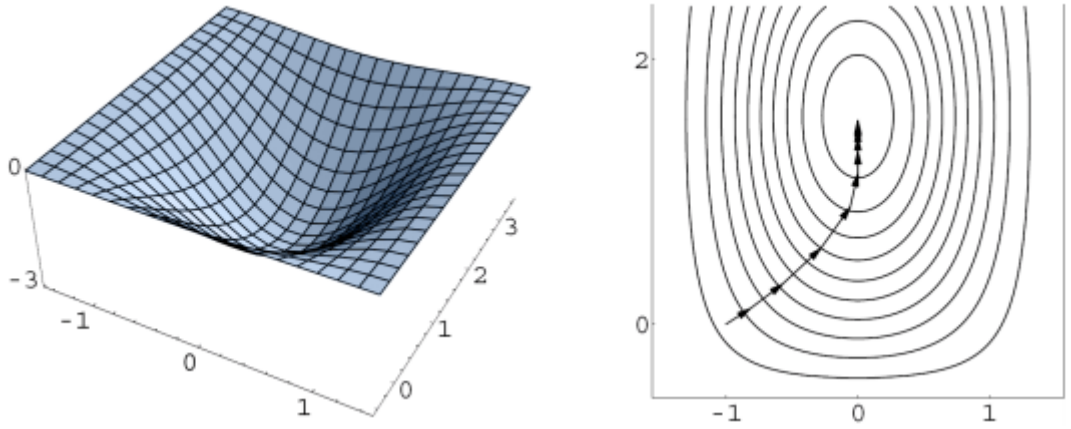

рисунок 1.2 — Візуалізація градієнтного спуску

Згорткові нейронні мережі

Згорткові нейронні мережі складаються з безлічі шарів штучних нейронів. Коли зображення входить в нейронну мережу, кожен з його шарів генерує кілька карт активації. Карти активації виділяють відповідні особливості зображення. Кожен з нейронів бере вхідний сигнал як ділянку пікселів, множить їх значення кольору на його вагу, підсумовує їх і запускає через функцію активації. Перший (або нижній) шар CNN зазвичай виявляє основні ознаки, такі як горизонтальні, вертикальні та діагональні ребра. Вихід першого шару подається як вхід наступного шару, який витягує складні функції, такі як кути та комбінації ребер. В міру проникнення в згорткову нейронну мережу шари починають виявляти функції вищого рівня, такі як об’єкти, обличчя тощо.[10]

рисунок 1.3 — Проходження зображення через згорткову нейронну мережу

Операція множення значень пікселів на ваги та їх підсумовування називається «згорткою» (звідси і назва згорткової нейронної мережі). CNN зазвичай складається з декількох згорткових шарів, але він також містить інші компоненти. Кінцевий рівень CNN - це класифікаційний рівень, який приймає вихідні дані остаточного рівня згортки як вхідні дані (пам’ятайте, вищі рівні згортки виявляють складні об’єкти). На основі карти активації остаточного рівня згортки рівень класифікації видає набір оцінок вірогідності (значення від 0 до 1), які вказують, наскільки ймовірно, що зображення належить до “класу”. Наприклад, якщо у вас є ConvNet, який виявляє котів, собак та коней, результат остаточного шару - це можливість того, що вхідне зображення містить будь-яку з цих тварин.

Навчання згорткової нейронної мережі є регулювання ваги окремих нейронів для вилучення правильних рис із зображень. Процес регулювання цих ваг називається «тренуванням» нейронної мережі.

На початку тренування CNN має набір випадкових вагів. Під час навчання розробники надають нейромережі великий набір зображень, анотованих відповідними класами (кішка, собака, кінь тощо). Нейронна мережа обробляє кожне зображення з його випадковими значеннями, а потім порівнює його вихід з правильним класом зображення. Якщо вихід мережі не відповідає класу - що, зазвичай відбувається на початку тренувального процесу - відбувається корекція ваги нейронів, щоб наступного разу, результат був трохи ближче до правильної відповіді.

Виправлення здійснюються за методом зворотного розповсюдження. По суті, зворотне розповсюдження оптимізує процес налаштування та полегшує мережі прийняття рішення про те, які групи нейронів налаштувати, замість випадкових виправлень.

Кожен цикл навчального набору даних називається "епохою". Під час тренувань згорткової мережі проходить кілька епох, регулюючи свою вагу в невеликих кількостях. Після кожної епохи нейронна мережа стає дещо

кращою при класифікації навчальних зображень. У міру вдосконалення CNN, коригування ваг стає все меншим.

Принцип роботи згорткові нейронні мережі з зображеннями

Для початку дамо визначення персептрону.

Персептрон - це лінійний класифікатор; тобто це алгоритм, який класифікує введення, розділяючи дві категорії прямою лінією. Вхідні дані, як правило, вектор функції x множиться на вагах wi додав до зміщення b: y = w * x + b. Персептрон має одиничний вихід на основі кількох дійсних входів, формуючи лінійну комбінацію, використовуючи свої вхідні ваги (а іноді передаючи вихід через нелінійну функцію активації). Математичний запис:

$$y = \varphi(\sum_{i=1}^{n} w_i x_i + b) = \varphi(\mathbf{w}^T \mathbf{x} + b)$$

де w позначає вектор ваг, x - вектор входів, b - зміщення, а phi - нелінійна функція активації.[11]

Багатошаровий персептрон (MLP) - це глибока, штучна нейронна мережа. Він складається з більш ніж одного персептрона. Вони складаються з вхідного рівня для приймання сигналу, вихідного рівня, який приймає рішення або передбачення щодо вхідного сигналу, а між цими двома - довільної кількості прихованих шарів, які є справжнім обчислювальним механізмом MLP. MLP з одним прихованим шаром здатні наблизити будь-яку безперервну функцію.[12]

Багатошарові перцептрони часто застосовуються до контрольованих навчальних проблем : вони тренуються на наборі пар вхід-вихід і вчаться моделювати кореляцію (або залежності) між цими входами та результатами. Навчання передбачає коригування параметрів, ваг та упереджень моделі, щоб мінімізувати помилки. Зворотне розповсюдження використовується для здійснення цих коригувань зважування та зміщення щодо похибки, а саму похибку можна виміряти різними способами, включаючи середньоквадратичну похибку (RMSE).[13]

Перше, що слід знати про згорткові мережі, це те, що вони не сприймають зображення, як люди. Отже, вам доведеться по-іншому подумати про те, що означає зображення і як воно обробляється згортковою мережею.

Коеволюційні мережі сприймають зображення як обсяги; тобто тривимірні об'єкти, а не плоскі полотна, які слід вимірювати лише шириною та висотою. Це пояснюється тим, що цифрові кольорові зображення мають червоно-синьо-зелене (RGB)[14] кодування, змішуючи ці три кольори, створюючи кольоровий спектр, який сприймають люди. Мережа поглинає такі зображення, як три окремі кольорові шари, складені одна на іншу.

Отже, згорткова мережа отримує звичайне кольорове зображення у вигляді прямокутного вікна, ширина та висота якого вимірюється кількістю пікселів уздовж цих розмірів, а глибина яких становить три шари глибиною, по одному на кожну букву в RGB. Ці глибинні шари називаються каналами.

Оскільки зображення рухаються згортковою мережею, ми опишемо їх з точки зору вхідних та вихідних обсягів, виражаючи їх математично як матриці з декількома вимірами у такому вигляді: 30x30x3. Від шару до шару їх розміри змінюються з причин, які будуть пояснені нижче.

Потрібно буде приділити увагу точним міркам кожного виміру об’єму зображення, оскільки вони є основою лінійних операцій алгебри, що використовуються для обробки зображень. Тепер для кожного пікселя зображення інтенсивність R, G і B буде виражена числом, і це число буде елементом в одній з трьох складених двовимірних матриць, які разом утворюють обсяг зображення. Ці цифри є початковими, необробленими, сенсорними властивостями, що надходять у згорткову мережу, і метою ConvNets є знайти, які з цих чисел є значущими сигналами, які насправді допомагають йому більш точно класифікувати зображення.

Замість того, щоб фокусуватись на одному пікселі за раз, згорткова мережа приймає квадратні ділянки пікселів і передає їх через фільтр. Цей фільтр також є квадратною матрицею, меншою за власне зображення та

рівною за розміром патчу. Завданням фільтра полягає у пошуку шаблонів у пікселях.

Уявімо дві матриці. Одна - 30x30, а інша - 3x3. Тобто фільтр покриває соту частину площі поверхні одного каналу зображення. Ми збираємося взяти точковий добуток фільтра з цим фрагментом каналу зображення. Якщо дві матриці мають високі значення в однакових позиціях, точковий результат буде високим. Якщо цього не сталося, воно буде низьким. Таким чином, одне значення - вихід точкового добутку - може сказати нам, чи відповідає шаблон пікселя на базовому зображенні шаблону пікселів, вираженому нашим фільтром.

Уявімо, що наш фільтр виражає горизонтальну лінію з високими значеннями вздовж другого рядка та низькими значеннями у першому та третьому рядках. Тепер малюнок, який ми починаємо у верхньому лівому куті базового зображення, і крок за кроком рухаємо фільтр по зображенню, поки він не дійде до верхнього правого кута. Розмір кроку відомий. Ми можемо переміщати фільтр за один правий стовпець за раз, або ви можете зробити більші кроки.

На кожному кроці беремо інший крапковий продукт і розміщуємо результати цього крапкового продукту в третій матриці, відомій як карта активації. Ширина або кількість стовпців карти активації дорівнює кількості кроків, які фільтр робить для обходу базового зображення. Оскільки більші кроки призводять до меншої кількості кроків, великий крок дасть меншу карту активації. Це важливо, оскільки розмір матриць, які згорткові мережі обробляють і виробляють на кожному шарі, прямо пропорційний тому, наскільки обчислення дорогі та скільки часу потрібно для навчання. Більший крок означає менше часу та обчислень.

Фільтр, накладений на перші три рядки, ковзатиме по них, а потім починатиметься знову з рядків 4-6 того самого зображення. Якщо він має крок три, то він створить матрицю точкових виробів, яка дорівнює 10x10. Цей самий фільтр, що представляє горизонтальну лінію, може бути застосований

до всіх трьох каналів основного зображення, R, G і B. І три карти активації 10x10 можуть бути складені, так що сукупна карта активації для горизонтальної лінії на всіх трьох каналах основного зображення також становить 10x10.

Тепер, оскільки зображення мають лінії, що йдуть у багатьох напрямках, і містять безліч різних видів фігур та піксельних візерунків, ви захочете просунути інші фільтри через основне зображення у пошуках цих шаблонів. Наприклад, ви можете шукати 96 різних візерунків у пікселях. Ці 96 шаблонів створять стос із 96 карт активації, в результаті чого з’явиться новий том 10x10x96. На діаграмі нижче ми перемаркували вхідне зображення, ядра та вихідні карти активації.

Те, що ми щойно описали, - це згортка. Ви можете розглядати Convolution як вигадливий вид множення, який використовується при обробці сигналів. Інший спосіб думати про дві матриці, що створюють крапковий добуток, - це дві функції. Зображення - це основна функція, а фільтр - це функція, через яку проходить зображення.

Однією з головних проблем зображень є те, що вони мають великі розміри, а це означає, що вони обробляють багато часу та обчислювальних потужностей. Згорткові мережі створені для зменшення розмірності зображень різними способами. Ступінь фільтра - один із способів зменшити розмірність. Інший спосіб - зниження вибірки.

Наступний шар у згортковій мережі має три назви: максимальне об’єднання, зменшення вибірки та субдискретизація. Карти активації подаються в шар зменшення вибірки, і, як звивини, цей метод застосовується по одному патчу за раз. У цьому випадку максимальне об’єднання просто приймає найбільше значення з одного патча зображення, розміщує його в новій матриці поруч з максимальними значеннями інших патчів і відкидає решту інформації, що міститься на картах активації.

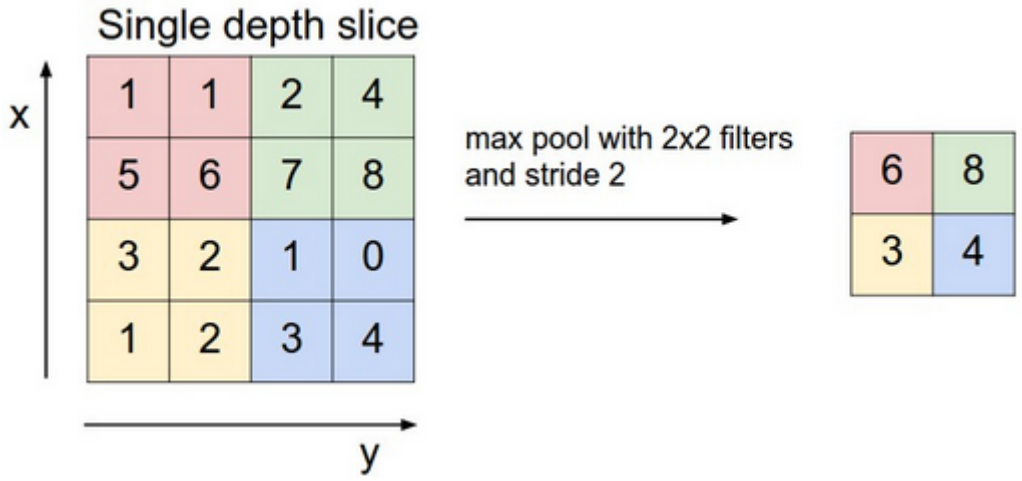

Рисунок 1.4 — Принцип роботи фільтру

Зберігаються лише місця на зображенні, які показали найсильнішу кореляцію з кожною ознакою (максимальне значення), і ці максимальні значення поєднуються, утворюючи простір нижчих розмірів. На цьому кроці втрачається багато інформації меншої цінності, що спонукало до дослідження альтернативних методів. Але зниження вибірки має ту перевагу, що саме через те, що інформація втрачається, зменшує обсяг необхідного зберігання та обробки.У міру того, як дедалі більше інформації втрачається, шаблони, оброблені згортковою мережею, стають все більш абстрактними та відрізняються від візуальних шаблонів, які ми розпізнаємо як люди.

Недоліки згорткових нейронних мереж

Попри свою потужність і складність, згорткові нейронні мережі, по суті, є засобом для розпізнавання образів. Вони можуть використовувати величезні обчислювальні ресурси, щоб розпізнати крихітні та непомітні візерунки, які можуть залишитися непоміченими для людського ока. Але коли справа доходить до розуміння значення змісту зображень, вони працюють погано.

Розглянемо наступне зображення:

Рисунок 1.5 — Приклад фото

Добре навчена нейронна мережа скаже, що це фото двох людей та річки. Але людина може дати детальний опис місця події, почуттів і настрою, яке передаю фото. Штучні нейронні мережі не розуміють цих понять. Різниця стає більш очевидною у практичному застосуванні згорткових нейронних мереж. Наприклад, CNN зараз широко використовуються для модерування вмісту в соціальних мережах. Але, всупереч величезному сховищу зображень та відео, на яких вони навчені, вони все ще блокують невідповідний вміст. Як приклад, модератор вмісту Facebook заблокував фотографію статуї як наготу.

Ще однією проблемою згорткових нейронних мереж є їх нездатність зрозуміти відносини між різними об'єктами. Розглянемо наступне зображення, яке відоме як "проблема Бонгарда"[15]. Завдання Bongard представляють вам два набори зображень (шість зліва та шість праворуч), і ви повинні пояснити ключову різницю між цими двома наборами. На прикладі зображення (рис. 1.6) в лівому наборі містять один об'єкт, а зображення в

правому наборі містять два об'єкти. Людям легко зробити такі висновки з такої невеликої кількості зразків. Якщо я покажу вам ці два набори, а потім запропоную вам нове зображення, ви зможете швидко вирішити, чи слід йому переходити в лівий чи правий набір.

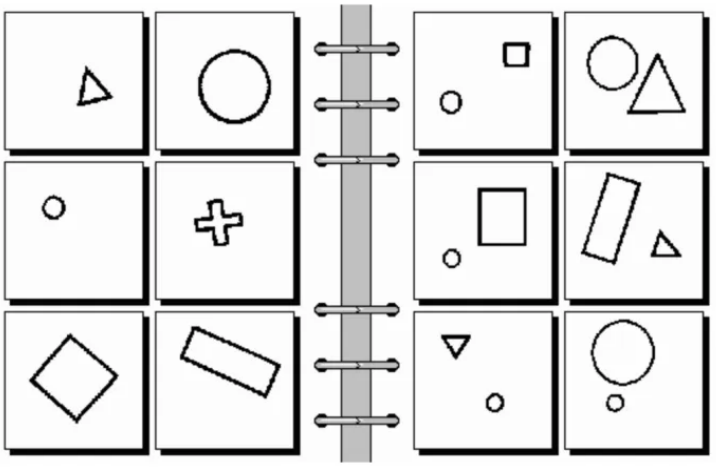

Рисунок 1.6— Проблема Бонгарда

Але все ще немає згорткової нейронної мережі, яка б могла розв'язувати проблеми Бонгарда за таку кількість прикладів навчання. В одному дослідженні, проведеному в 2016 році, дослідники ШІ навчили CNN на 20 000 зразках Bongard і протестували його ще на 10 000. Ефективність CNN була набагато нижчою, ніж у середніх людей.

Особливості ConvNets також роблять їх вразливими до змагальних атак, збурень у вхідних даних, які залишаються непоміченими для людського ока, але впливають на поведінку нейронних мереж. Суперечливі атаки стали головним джерелом занепокоєння, оскільки глибоке навчання, і особливо CNN, стали невіднятною складовою багатьох важливих програм, таких як самокеровані машини.

рисунок 1.7 — Приклад помилкової класифікації фото завдяки додаванню "шуму"

Класифікація чат-ботів

Класифікуємо чат-боти за своєю поведінкою:

а) боти на основі машинного навчання, які вміють підтримувати простий логічний діалог з користувачем;

б) боти які використовують скрипти або дерево рішень і мають обмежену кількість команд.

Чат-боти використовуються в освіті, засобах масової інформації та оптимізації бізнесу. Чат-бот може автоматично надсилати документи, нагадати про зустріч, консультувати клієнтів і таким чином оптимізувати навантаження на кол-центри і навіть збирати інформацію про користувача під час діалогу з ним.

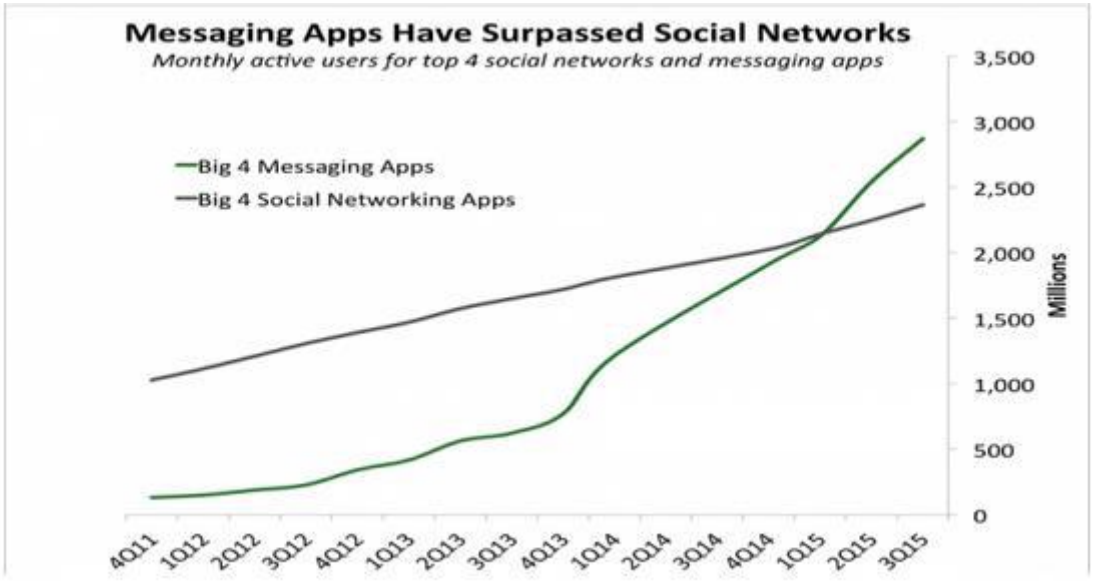

Рисунок 1.8 – Тенденції ринку месенджерів

За даним графіком спостерігається тенденція швидкого набуття популярності месенджерів, а отже і чат-ботів. Це зумовлено тим, що тепер у людей менше часу для того, щоб проводити його у соціальних мережах, тому потрібна швидка можливість написати повідомлення та швидко отримувати інформацію про новини, погоду тощо. Саме для такого завдання підходять інформаційні чат-боти в яких навіть є функція підписки на окремі теми новин.

Основні напрямки використання

Основні напрямки використання чат-ботів:

1. У сфері бізнесу чат-боти мають найбільший набір функцій і вирішують такі завдання як консультація клієнтів, рекламні сповіщення, збір статистики по користувачах.
2. Інформаційні боти, використовуються в засобах масової інформації.
3. Чат-боти для розваг. Можуть вести діалог з користувачем, грати в ігри, проводити голосування в груповому чаті.

Огляд існуючих рішень

Перед розробкою власного Телеграм-боту, треба ознайомитися з існуючими чат-ботами, їх функціоналом та дизайном інтерфейсу.

@CryptoBot - бот для покупки та продажі криптовалюти. Бот також надає інформацію про курс криптовалюти та може інформувати користувача про його зміну. Це має сенс, коли користувач не має змоги постійно слідкувати за курсом криптовалюти, але хоче отримувати новини.[16]

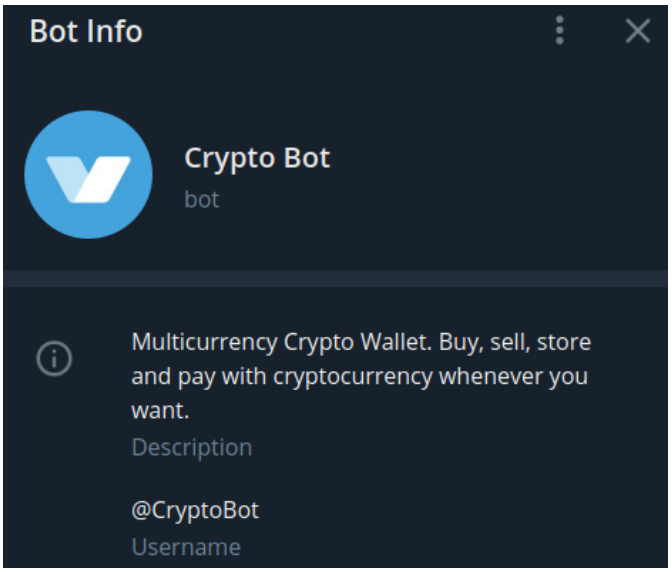

Рисунок 1.9 — Опис Телеграм-боту CryptoBot

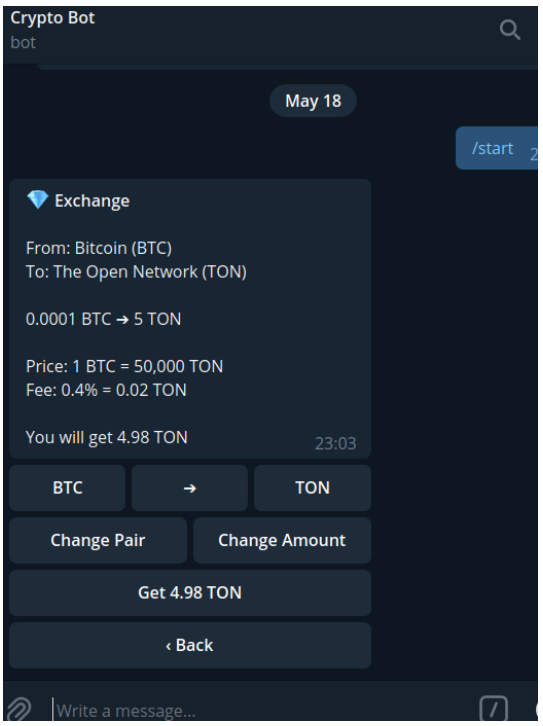

Рисунок 1.10 — Інтерфейс Телеграм-боту CryptoBot

@djinni_jobs_bot — телеграм бот сервісу пошуку вакансій Djinni.[17] В Телеграм-боті є можливість підписки на вакансії, які цікавлять користувача за ключовими словами.

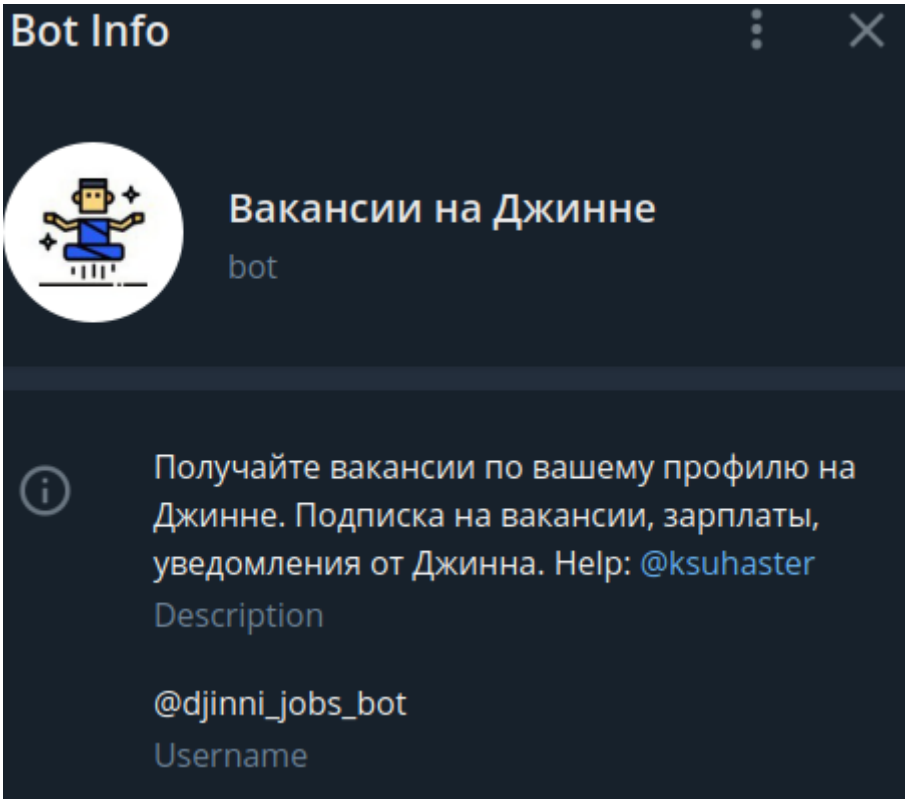

Рисунок 1.11 — Опис Телеграм-боту djinni_jobs_bot

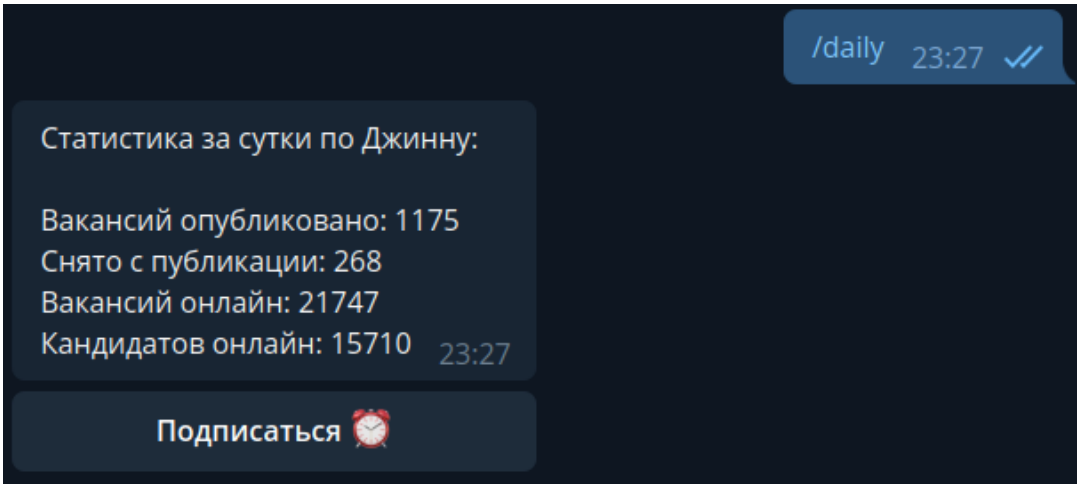

Рисунок 1.12 — Статистика вакансій в Телеграм-боті djinni_jobs_bot

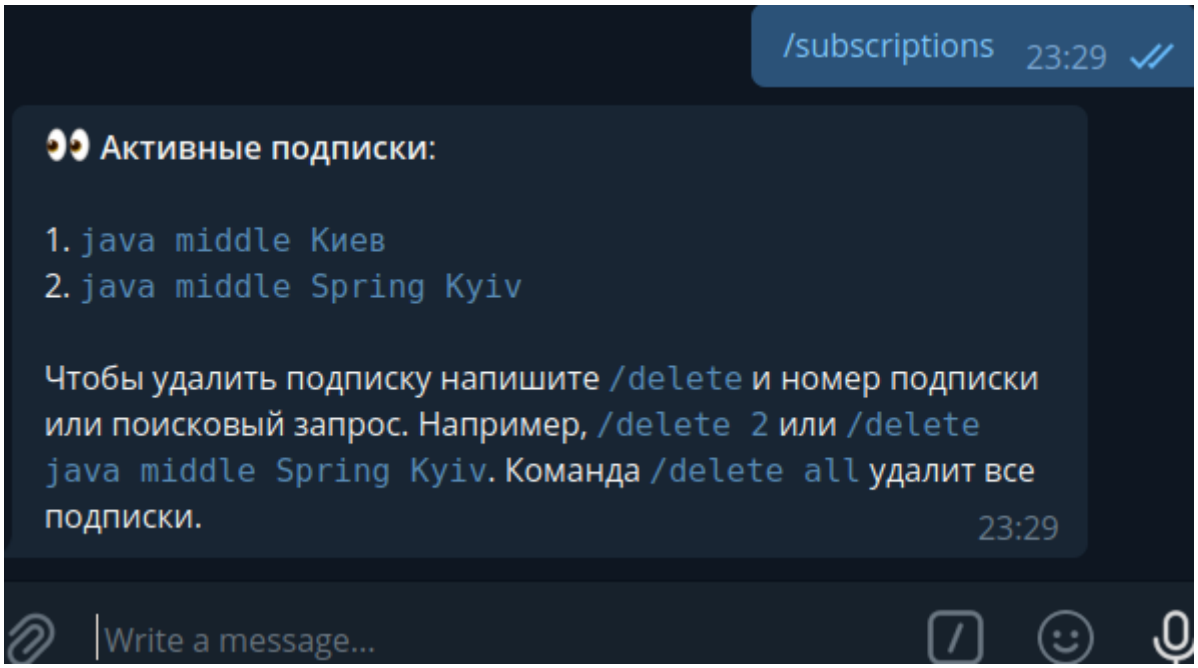

Рисунок 1.13 — Список підписок на вакансії в телеграм-боті djinni_jobs_bot

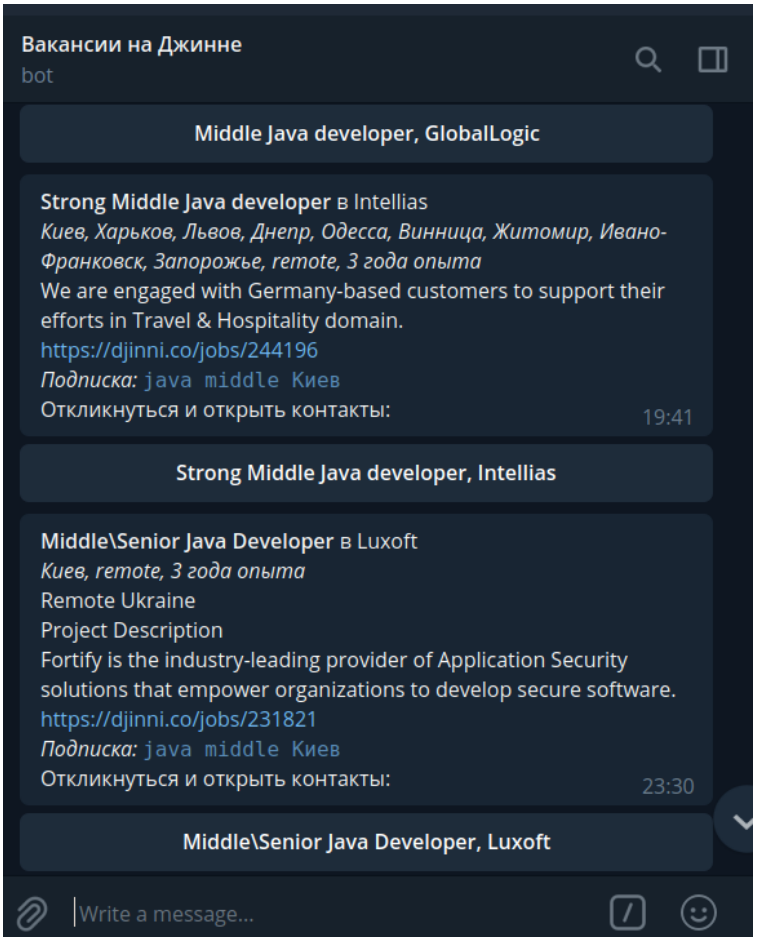

Рисунок 1.14— Приклад сповіщення про нові вакансії в Телеграм-боті djinni_jobs_bot

Бот дуже зручний у використанні та економить багато часу, тому що немає більше необхідності самому заходити на веб-сторінку, телеграм бот автоматично надсилає нові вакансії та статистику наймів.

Можливі види інтерфейсу чат-бота

Основний спосіб взаємодії користувача і бота це підтримка наперед визначених команд. Більш зручним способом є реалізація клавіатури в інтерфейсі телеграм, де команди подаються візуально у вигляді кнопок і для їх відправлення достатньо просто натиснути по ній на екрані.

РОЗДІЛ 2 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

Проєктування важлива частина розробки будь-якої програмної системи. Умовно, цей процес можна розділити на кілька послідовних, пов'язаних один з одним етапів: формулювання цілей (збір вимог), аналіз предметної області та створення прототипу системи. Аналіз предметної області має на увазі глибоке вивчення вимог, що пред'являються до функціонала проєктованої системи, опрацювання сценаріїв взаємодії з системою і алгоритмів в бізнес-логіці. Чітко сформульовані цілі і завдання, ретельно проведений аналіз вимог, опрацювання сценаріїв взаємодії з системою і проєктування бізнес-логіки є ключем успіху в досягненні необхідного результату. Таким чином, проєктування збільшує шанси успішного складання проєкту, а також економить час і гроші, адже внесення змін на початковому етапі розробки не так критично, як, наприклад, на етапі налагодження вже розробленої системи.[18]

Процес розробки можна поділити на наступні етапи:

- Архітектура програмного продукту
- Вибір мови програмування
- Вибір стеку технологій
- Реалізація програмного продукту
- Тестування
- Розгортка програмного продукту (англ. Software Deployment)

Вибір мови програмування JAVA для розробки Телеграм-боту

Згідно з документацією Telegram для розробки чат-бота необхідно знати хоча б одну з мов серверного програмування: Python, Ruby, Node.JS, PHP, Java.[20]

Спілкування нашого чат-бота з сервером Telegram відбувається за допомогою протоколу REST (Representational State Transfer) API (Application Programming Interface)[21], який надають месенджери, описаного в Telegram Bot API. Проаналізувавши вже існуючи рішення та навчальні матеріали, я дійшов висновку, що найчастіше Telegram-ботів пишуть на Python, оскільки це мова високого рівня і досить популярна.

В курсовій роботі реалізація буде описана на серверному язику програмування Java, оскільки для цієї мови існують потужні та перевірені часом фреймворки.

Огляд язику програмування та фреймворків :

Java - об’єктно орієнтована мова програмування, в офіційній реалізації Java-програми компілюється у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.[22]

Вибір стеку технологій для розробки Телеграм-боту

Оскільки наш чат-бот являє собою сервер, який повинен підтримувати протокол http, ми порівняємо можливі варіанти фреймворків та технологій, які можуть пришвидшити розробку нашого боту і використовувати якісні готові рішення.

Специфікація мови Java пропонує Servlet API.[23]

Java Servlet API - стандартизований API, призначений для реалізації на сервері та роботи з клієнтом за схемою запит-відповідь.

Сервлет - це клас, який вміє отримувати запити від клієнта і повертати йому відповіді. Сервлети в Java - саме ті елементи, за допомогою яких будується клієнт-серверна архітектура, є точкою входу для веб-сервера.

Для написання гнучкої архітектури на основі сервлетів, ми повинні реалізувати патерн проєктування Command.[24]

Патерн проєктування Command - перетворює запит на виконання дії в окремий об'єкт-команду. Така інкапсуляція дозволяє передавати ці дії інших функцій і об'єктів як параметр, наказуючи їм виконати запитану операцію. Команда - це об'єкт, тому над нею допустимі будь-які операції, що і над об'єктом.

Інтерфейс командного об'єкта визначається абстрактним базовим класом Command і в найпростішому випадку має єдиний метод execute (). Похідні класи визначають одержувача запиту (покажчик на об'єкт-одержувач) і необхідну для виконання операцію (метод цього об'єкта). Метод execute () підкласів Command просто викликає потрібну операцію одержувача.

Іншим варіантом реалізації вебсерверу є використання Spring framework, який “під капотом” використовує Servlet API у поєднанні з шаблоном проєктування Command і пропонує зручне написання вебсерверу з мінімальною кількістю коду.

Spring - Spring Framework пропонує комплексну модель програмування та конфігурації сучасних корпоративних Java-програм - на будь-якій платформі розгортання.[25]

Ключовим елементом Spring є інфраструктурна підтримка на рівні прикладних програм: Spring фокусується на оптимізації бойлер-коду корпоративних додатків, щоб команди могли зосередитись на бізнес-логіці на рівні додатків без зайвих зв'язків із конкретними середовищами розгортання.

Spring boot — Надбудова над фреймворком Spring, яка має наступні переваги:

• Створення stand-alone Spring програм.

• Вбудований сервер Tomcat, Jetty або Undertow безпосередньо (не потрібно розгортати файли WAR)[26]

• Вбудований набір популярних бібліотек без додаткових налаштувань

• Production-ready додатки, такі як метрики, перевірка стану сервера, легке підключення нових бібліотек.

Lombok — фреймворк для генерації бойлер коду, наприклад getter, setter методи, конструктори, реалізація патерну Builder.[27]

```java
public class User {
    private String name;
    private String secondName;
    private int age;

    public User() {
    }

    public User(String name, String secondName, int age) {
        this.name = name;
        this.secondName = secondName;
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getSecondName() {
        return secondName;
    }

    public void setSecondName(String secondName) {
        this.secondName = secondName;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
```

Рисунок 2.1 — клас User без використання фреймворку Lombok

```java
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class User {
    private String name;
    private String secondName;
    private int age;
}
```

Рисунок 2.2 — клас User з використанням фреймворку Lombok

На малюнках (рис)представлена різниця коду без фреймворку Lombok та з ним.

Фреймворк Lombok використовує метадані @Data, @NoArgsConstructor, @AllArgsConstructor, @Builder для генерації коду getter і setter методів, конструктора без параметрів, конструктора з параметрами і внутрішнього класу який реалізовує шаблон проєктування Builder.

Основною технологією для роботи з СУБД в Java є JDBC.

JDBC - промисловий стандарт взаємодії Java-додатків з різними СУБД, реалізований у вигляді пакета java.sql, що входить до складу Java SE.[28]

JDBC працює на основі так званих драйверів, що дозволяють отримувати з'єднання з базою даних по спеціально описаного URL. Драйвери можуть завантажуватися динамічно (під час роботи програми). Завантажившись, драйвер сам реєструє себе і викликається автоматично, коли програма вимагає URL, що містить протокол, за який драйвер відповідає.

Більш сучасною технологією для роботи з СУБД є Hibernate.

Hibernate — реалізація Java ORM framework для розробки бази даних за принципом code-first, працює на основі JDBC.[29]

Основні переваги використання ORM:

- для реалізації однакового функціонала потрібно написати менше коду;
- Не потрібно писати SQL запити та власноруч заповнювати об’єкти даними, цю роботу виконую фреймворк автоматично, ми повинні тільки писати об’єкти, в нашому випадку Java класи з додатковою метаінформацією;
- Легко підтримувати, адже зміна entity в коді зразу робить зміни в базі даних

Spring data - Мета Spring Data полягає у наданні звичної та послідовної моделі програмування базованої на фреймворку Spring для доступу до даних, зберігаючи при цьому особливі риси базового сховища даних.

Це полегшує використання технологій доступу до даних, реляційних та не реляційних баз даних. Це великий проєкт, який містить багато підпроєктів, характерних для певної бази даних. Проєкти розробляються спільно з багатьма компаніями та розробниками.

Особливості Spring data framework :

- Великий рівень абстракції об’єктів
- Динамічне генерування запитів з імен методів репозиторію
- Можливість інтеграції спеціального коду репозиторію.

Отже, проаналізувавши існуючі варіанти технологій для розробки чат-боту, ми виберемо :

- Spring boot — для реалізації вебсерверу с вбудованим http сервером Tomcat
- Hibernate, Spring data - ORM framework для зручної роботи з базою даних.
- Lombok — для генерації boilerplate code.

Архітектура програмного продукту

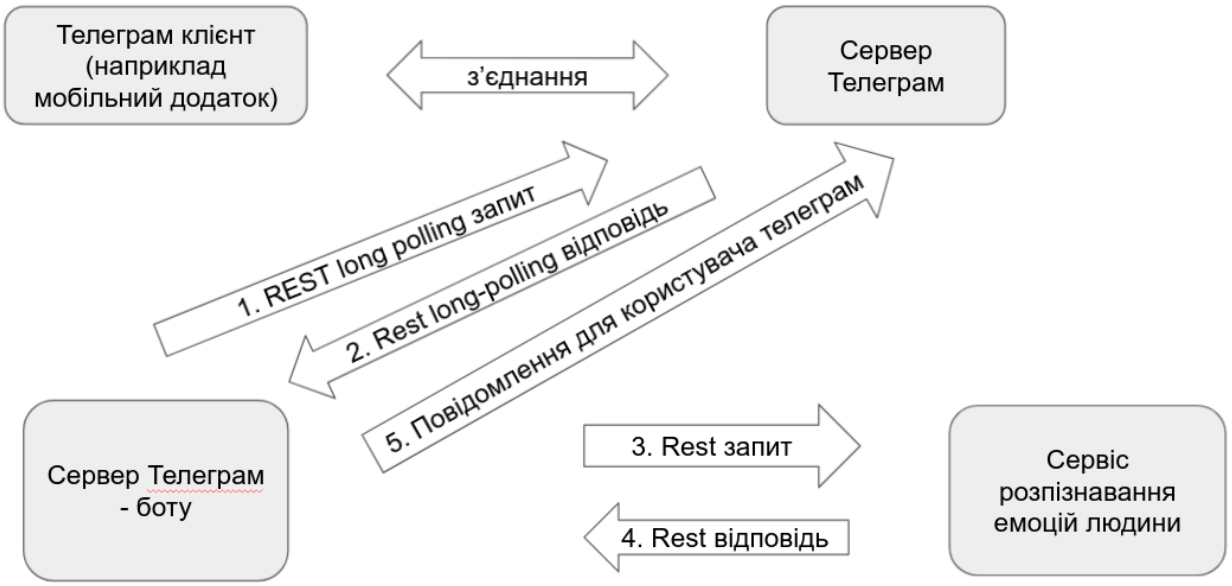

Рисунок 2.3 — Архітектура програмного продукту

Архітектура складається з наступних частин:

- Телеграм клієнт - мобільний додаток абе веб версія інтерфейсу
- Сервер телеграм - бекенд Телеграму
- Сервер Телеграм-боту - бекенд, який буде розроблятися в цій роботі.
- Сервіс розпізнавання емоцій, також буде реалізований в цій роботі

Телеграм-бот взаємодіє з сервісом розпізнавання емоції через протокол REST[19], надсилає запит з фотографією і отримую відповідь в якому описані емоції, які визначені за надісланою фотографією.

Телеграм бот відповідає за підказки команд користувачеві та валідацію вхідних даних.

Потрібно розробити два сервіси:

- Телеграм бот
- Сервіс розпізнавання емоцій людини

Для кожного сервісу буде визначена мова програмування та стек технологій.

Тип отримання оновлення на сервер бота

Існує два варіанти роботи чат-бота як бекенд сервісу[30] :

- Long polling bot
- Webhook bot

Long polling bot. Цей метод використовує принцип long poll. Long poll — принцип за яким сервер відповідає не зразу, а збирає всі запити за проміжок часу T і потім обробляє всі запити пачкою.[31]

В нашому випадку long poll реалізований з боку Telegram сервера, а ми згідно API з нашого боку, тобто на сервері Telegram bot повинні реалізувати метод getUpdates, який раз в період T опитує сервер Telegram на наявність оновлень та є точкою входу в нашому коді.

Webhook bot. Використовуйте цей метод, щоб вказати URL-адресу для отримання вхідних даних оновлення. Щоразу, коли є оновлення для бота, Telegram надсилає HTTPS POST-запит на вказану URL-адресу, що містить JSON-серіалізоване оновлення.

Для забезпечення безпеки і гарантії, що запит Webhook надходить від Telegram, рекомендується використовувати секретний шлях в URL-адресі, наприклад https://www.example.com/ <token>. Оскільки ніхто інший не знає секретний токен бота, ми можемо бути впевнені, що це телеграм.

Webhook bot в реалізації складніший, бо вимагає :

- Статичний IP адрес нашого чат-бот сервера, оскільки сервер телеграм повинен мати змогу напряму надіслати нам запит(hook). При наявності маршрутизатора, додатково потрібно настроїти кидок портів.
- SSL сертифікат, Сертифікат може бути підтверджений або самостійно підписаний. Налаштування Webhook із сертифікатом, який підписуємо самостійно, дещо відрізняється від встановлення Webhook із підтвердженим сертифікатом.
- Більш складне налаштування та менше документації. В курсовій роботі буде розглядатися Long polling метод написання бота.

В цій роботі буде розглядатися Long polling метод написання бота.

Реєстрація боту в Telegram

Для взаємодії телеграм серверу з сервером боту потрібен API token. Він забезпечує безпечне з’єднання і підтвердженням, що ми зареєстрували бот в системі Telegram.

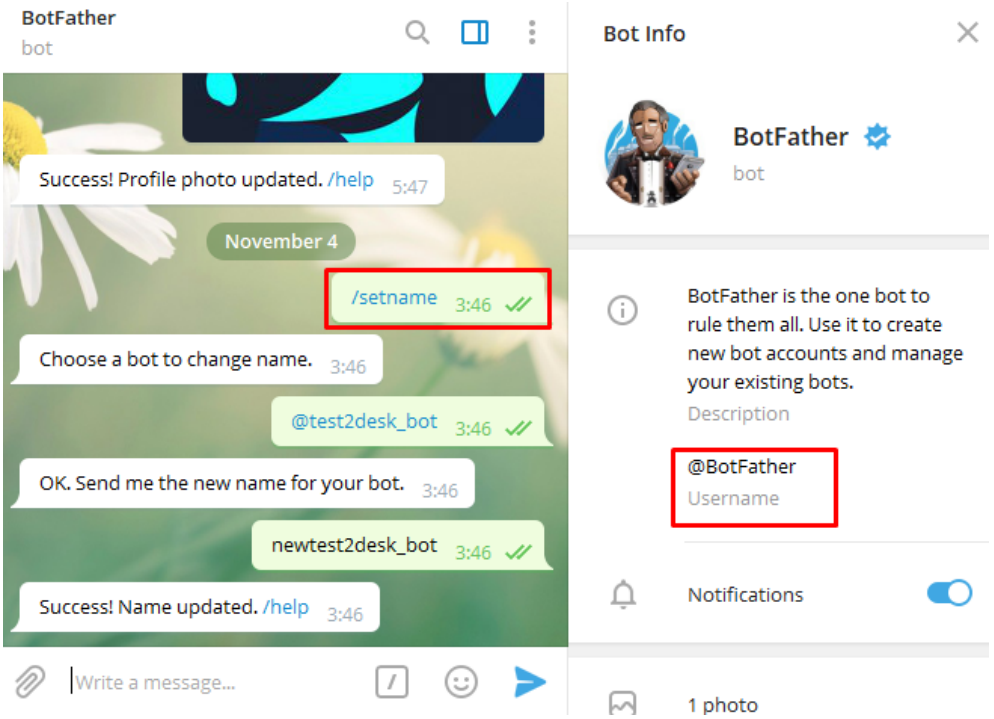

Рисунок 2.4 — Телеграм бот для реєстрації телеграм боту

Для реєстрації боту в Telegram необхідно знайти Telegram бот BotFather, мета якого є реєстрація і налаштування створення користувацького бота (рисунок). BotFather дозволяє присвоїти ім'я нашому боту, задати його опис, зображення і список команд, доступних користувачам. Необхідно заповнити всі пункти, бо надалі від цього буде залежати те, наскільки ботом зручно користуватися. Після реєстрації видається необхідний нам API token.

Шаблон боту на GitHub

Для спрощення реалізації, ми візьмем шаблон проєкту з GitHub, в якому вже написані усі потрібні нам DTO які відповідають усім JSON об’єктам описаним в документації Telegram Bot API. https://github.com/rubenlagus/TelegramBots

Оскільки бот працює на основі команд, було зручне реалізувати аналог MVC архітектури, але замість REST контролеру, буде BotController.

MVC архітектура — Модель–вигляд–контролер (або Модель–представлення–контролер, англ. Model-view-controller, MVC) — архітектурний шаблон, який використовується під час проєктування та розробки програмного забезпечення.[32]

Цей шаблон передбачає поділ системи на три взаємопов'язані частини: модель даних, вигляд (інтерфейс користувача) та модуль керування. Застосовується для відокремлення даних (моделі) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

Мета шаблону — гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах сприяє впорядкованості їхньої структури і робить їх більш зрозумілими шляхом зменшення складності.

На GiHub є готове рішення контролеру для телеграм команд і підтримкою основних об’єктів TelegramBot API : https://github.com/OlegNyr/java-telegram-bot-mvc

Згідно документації нам потрібно :

- Клонувати проєкт з GitHub з MVC реалізацією, як шаблон
- Включити підтримку MVC для Telegram за допомогою @EnableTelegram

- Реалізувати інтерфейс TelegramMvcConfiguration для конфігурації
- Написати клас контролер з анотацією @BotController і метод з @BotRequest("/start"), де “/start” - це команда до якої прив’язується даний метод

Далі розробка проєкту повністю аналогічна звичайному вебпроєкту.

ВИКОРИСТАННЯ СЕРВІСУ ВИЗНАЧЕННЯ ЕМОЦІЙНОГО СТАНУ ЛЮДИНИ

Вибір мови програмування Python

Для розробки сервісу визначення емоційного стану людини будемо використовувати мову програмування Python з наступних причин:

- Python має сотні різних бібліотек та фреймворків, що є чудовим доповненням до процесу розробки. Вони економлять багато часу. Багато з цих бібліотек будуть зосереджені на аналізі даних та машинному навчанні. Крім того, існує величезна підтримка Big Data.
- Python підтримує фреймворки для розробки веб додатку.

Python є найкращою мовою програмування для розробки проєктів з аналізом даних. Оскільки потрібна проста у використанні мова, яка має велику кількість бібліотек. Python все ще перебуває в розробці, тобто регулярно отримує оновлення.[33]

Вибір стеку технологій

OpenCV - це open source бібліотека комп'ютерного зору, яка призначена для аналізу, класифікації та обробки зображень. У неї входять понад 2500 алгоритмів, в яких є як класичні, так і сучасні алгоритми для комп'ютерного зору і машинного навчання. Ця бібліотека має інтерфейси на різних мовах, серед яких є Python.[34]

Django - це високоякісний вебфреймворк Python, який стимулює швидкий розвиток та чистий, прагматичний дизайн. Побудований досвідченими розробниками, він оптимізує веброзробку, тому розробник може зосередитись на написанні свого додатку, не витрачаючи час на написання непотрібного коду.[35]

Тестування Телеграм-боту

Рисунок 2.5 — Тестування Телеграм-боту на позитивні емоції.

Рисунок 2.6 — Тестування Телеграм-боту на негативні емоції

ВИСНОВКИ

З розвитком месенджерів Телеграм, тема розробки чат-ботів стає все більш актуальною та використовуються в багатьох галузях. Телеграм бот може бути використаний як зручний засіб інтерфейсу зі сторонніми API сервісами. Таким чином в даній роботі був реалізований чат-бот для визначення емоцій людини за фотографією.

Було проведено аналіз предметної області та огляду існуючих рішень чат-ботів. Досліджувались різні види інтерфейсу та архітектури чат-ботів. Були описані принци отримання оновлень з сервера телеграм Long polling та Webhook. Також була розроблена архітектура взаємодії чат-боту з сторонніми сервісами через їх API.

Були розглянуті існуючі рішення та сервіси для розпізнавання емоцій людини за фотографією та їх API. Проаналізовано документацію сервісу.

У роботі використовується об’єктно орієнтована серверна мова програмування Java та фреймворк Spring. Розглянуті переваги даного стеку технологій та базові принципи використання його використання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Чат і месенджер-боти: тенденції в 2020 році [Електронний ресурс]. – 2020. – Режим доступу до ресурсу https://marketer.ua/ua/chat-and-messenger-bots-trends-2020.
2. Що таке електронна комерція? E-commerce для початківців [Електронний ресурс]. – 2021. – Режим доступу до ресурсу https://www.interkassa.com/ua/blog/chto-takoe-elektronnaya-kommerciya-e-commerce-dlya-nachinayushchih.
3. Телеграм [Електронний ресурс]. – 2021. – Режим доступу до ресурсу https://telegram.org.
4. Пол Екман. (1982). Теорія Брехні (PDF).
5. Емоційний AI: як технологія набуває людських рис [Електронний ресурс]. – 2020. – https://evergreens.com.ua/ua/articles/emotion-ai.html.
6. David Kriesel (2007). A Brief Introduction to Neural Networks.
7. Капсульные сети от Хинтона [Електронний ресурс]. – 2017. – https://habr.com/ru/company/recognitor/blog/343726.
8. Логістична рівність [Електронний ресурс]. – 2020. – https://uk.wikipedia.org/wiki/Логістична_рівність.
9. Штучний нейрон [Електронний ресурс]. – 2020. – https://uk.wikipedia.org/wiki/Штучний_нейрон.
10. What are convolutional neural networks (CNN)? [Електронний ресурс]. – 2020. – https://bdtechtalks.com/2020/01/06/convolutional-neural-networks-cnn-convnets.
11. Перцептрон [Електронний ресурс]. – 2020. – https://uk.wikipedia.org/wiki/Перцептрон.
12. Многослойный персептрон [Електронний ресурс]. – 2019. – https://wiki.loginom.ru/articles/multilayered-perceptron.html.
13. Среднеквадратичная ошибка (RMSE) [Електронний ресурс]. – 2019. – https://gis-lab.info/qa/rmse.html.

14. RGB [Електронний ресурс]. – 2019. – https://ru.wikipedia.org/wiki/RGB.
15. Проблема Бонгарда - Bongard problem [Електронний ресурс]. – 2019. – https://livepcwiki.ru/wiki/Bongard_problem.
16. Crypto Trade [Електронний ресурс]. – 2018. – https://itmaster-soft.com/en/telegram-cryptobot.
17. Анонімний пошук роботи Djinni [Електронний ресурс]. – 2020. – https://djinni.co.
18. 7 Steps of Effective Software Product Development Life Cycle [Електронний ресурс]. – 2020. – https://relevant.software/blog/7-steps-for-effective-software-product-development.
19. Representational State Transfer [Електронний ресурс]. – 2021. – https://ru.wikipedia.org/wiki/REST.
20. Телеграм API [Електронний ресурс]. – 2021. – https://core.telegram.org/api.
21. API [Електронний ресурс]. – 2021. – https://ru.wikipedia.org/wiki/API.
22. Java [Електронний ресурс]. – 2021. – https://ru.wikipedia.org/wiki/Java.
23. Сервлет [Електронний ресурс]. – 2021. – https://ru.wikipedia.org/wiki/Сервлет_(Java).
24. Шаблон проєктування Команда [Електронний ресурс]. – 2019. – https://refactoring.guru/ru/design-patterns/command.
25. Spring framework [Електронний ресурс]. – 2021. – https://spring.io/projects/spring-boot.
26. Java Web Servers [Електронний ресурс]. – 2019. – https://www.baeldung.com/java-servers.
27. Lombok возвращает величие Java [Електронний ресурс]. – 2019. – https://habr.com/ru/post/438870.
28. Java Database Connectivity [Електронний ресурс]. – 2015. – https://ru.wikipedia.org/wiki/Java_Database_Connectivity.

29. Hibernate [Електронний ресурс]. – 2019. – https://uk.wikipedia.org/wiki/Hibernate.
30. Telegram Bot API Getting Updates [Електронний ресурс]. – 2021. – https://core.telegram.org/bots/api#getting-updates.
31. Polling и long polling [Електронний ресурс]. – 2018. – https://devman.org/encyclopedia/about-chatbots/long-polling.
32. MVC [Електронний ресурс]. – 2020. – https://ru.wikipedia.org/wiki/Model-View-Controller.
33. Python [Електронний ресурс]. – 2020. – https://ru.wikipedia.org/wiki/Python.
34. OpenCV [Електронний ресурс]. – 2020. – https://ru.wikipedia.org/wiki/OpenCV.
35. Django [Електронний ресурс]. – 2020. – https://ru.wikipedia.org/wiki/Django.