Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Когнітивний сервіс для ідентифікації емоційного забарвлення голосових повідомлень, заснований на використанні глибинних нейронних мереж
(тема)

Виконав:
студент 2 курсу, групи СШІм-20-1
Запічний О.Ю.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник доц. Шевченко О.Ю.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Запічному Олексію Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Когнітивний сервіс для ідентифікації емоційного забарвлення голосових повідомлень, заснований на використанні глибинних нейронних мереж
затверджена наказом університету від 8 листопада 20 21 р. № 1695Ст
2. Термін подання студентом роботи до екзаменаційної комісії 7 грудня 20 21 р.
3. Вихідні дані до роботи специфікація мови програмування Python, документація фреймворків «Keras» та «Tensorflow», Google Colaboratory, теорія глибинних нейронних мереж, теорія інтелектуального аналізу даних
4. Перелік питань, що потрібно опрацювати в роботі аналіз предметної галузі,аналіз задач обробки природної мови, постановка задачі та вимог до неї, теоретичні дослідження архітектур нейронних мереж, експериментальні дослідження конфігурацій різних архітектур нейронних мереж

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1 – Уривок вибірки Large Movie Review Dataset, Рисунок 2 – Уривок вибірки RuTweetCorp, Рисунок 3 –Уривок повної вибірки Amazon Reviews for Sentiment Analysis, Рисунок 4 – Екранна формаінтерфейсу Google Colaboratory, Рисунок 5 – візуалізація методу векторизації one-hot-encoding, Рисунок 6 – Візуалізація методу щільного вектора для векторизації тексту, Рисунок 7 – Візуалізація явища упорядкування слів при використанні методу щільного вектора подання слів, Рисунок 8 – Візуалізація явища встановлення семантичних відношень при використанні методу щільного вектора подання слів , Рисунок 9 – Схематичне зображення архітектури повзнозв'язних нейронних мереж прямого поширення, Рисунок 10 – Графік сигмоїдної функції активації, Рисунок 11 – Графік функції активації гіперболічний тангенс, Рисунок 12 – Графік функції активації Rectified Linear Unit (ReLU), Рисунок 13 – Графік функції активації Leaky ReLU, Рисунок 14 – Схематична візуалізація методу dropout

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доц. Шевченко О.Ю.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	01.09.2021	виконано
2	Аналіз предметної галузі	02.09.2021 – 18.09.2021	виконано
3	Постановка задачі	19.09.2021 – 22.09.2021	виконано
4	Теоретичні дослідження	23.09.2021 – 14.10.2021	виконано
5	Експериментальні дослідження	15.10.2021 – 06.11.2021	виконано
6	Оформлення пояснювальної записки	07.11.2021 – 22.11.2021	виконано
7	Захист перед ЕК	09.12.2021	виконано

Дата видачі завдання 1 вересня 2021 р.

Студент _____
(підпис)

Керівник роботи _____ _____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Записка пояснювальна: 81 с., 32 рис., 3 дод., 8 джерел.

ВЕКТОРИЗАЦІЯ, ВИБІРКА, НЕЙРОННА МЕРЕЖА, ПОПЕРЕДНЯ ОБРОБКА, ПРИРОДНА МОВА, РЕКУРЕНТНА НЕЙРОННА МЕРЕЖА

Об'єкт дослідження – визначення сентиментальної тональності природної мови.

Предмет дослідження – використання нейронних мереж для визначення сентиментальної тональності природної мови .

Мета роботи – пошук найбільш оптимальних за результативністю та продуктивністю конфігурацій різних архітектур глибинних нейроних мереж.

Методи дослідження – аналіз технічної літератури з області інтеллектуального аналізу даних, експериментальний підбір конфігурацій нейронних мереж, порівняльний аналіз експериментів.

Проведено теоретичний аналіз різних вибірок тренувальних даних, архітектур нейронних мереж, методів оптимізації та інших параметрів. Проведення практичних дослідів передбачало підбір оптимальних значень параметрів та опис доцільності використання результатів кожного досліду. Практичні досліди проводилися на наборі реальних даних про відгуки до різних категорій товарів маркетплейсу Amazon, розміром у 400000 екземплярів. На основі отриманих результатів розроблено моделі, що за домогою узагальнюючої здаості нейромереж можуть достатньо точно визначати сентиментальну тональність тексту, що написаний англійською мовою.

РЕФЕРАТ

Пояснительная записка: 81 с., 32 рис., 3 прил., 8 источников.

ВЕКТОРИЗАЦИЯ, ВЫБОРКА, ЕСТЕСТВЕННАЯ РЕЧЬ, НЕЙРОННАЯ СЕТЬ, ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА, РЕКУРРЕНТНАЯ НЕЙРОННАЯ СЕТЬ

Объект исследования – определение сентиментальной тональности естественной речи.

Предмет исследования – использование нейронных сетей для определения сентиментальной тональности естественной речи.

Цель работы – поиск наиболее оптимальных по результативности и производительности конфигураций различных архитектур глубинных нейронных сетей.

Методы исследования – анализ технической литературы в области интеллектуального анализа данных, экспериментальный подбор конфигураций нейронных сетей, сравнительный анализ экспериментов.

Проведен теоретический анализ различных выборок тренировочных данных, архитектур нейронных сетей, методов оптимизации и других параметров. Проведение практических опытов подразумевало подбор оптимальных значений параметров и описание целесообразности использования результатов каждого опыта. Практические опыты проводились на наборе реальных данных об отзывах к разным категориям товаров маркетплейса Amazon, размером в 400000 экземпляров. На основе полученных результатов разработаны модели, которые по обобщающей способности нейросетей могут достаточно точно определять сентиментальную тональность текста, написанного на английском языке.

ABSTRACT

Explanatory note: 81 p., 32 fig., 3 ann., 8 sources.

DATASET, NATURAL SPEECH, NEURAL NETWORK, PRE-PROCESSING, RECURRENT NEURAL NETWORK, VECTORIZATION

The object of the research is the determination of the sentimental tonality of natural speech.

The subject of the research is the use of neural networks to determine the sentimental tonality of natural speech.

The aim of the work is to find the most optimal in terms of efficiency and performance configurations of various architectures of deep neural networks.

Research methods – analysis of technical literature in the field of data mining, experimental selection of neural network configurations, comparative analysis of experiments.

A theoretical analysis of various training data samples, neural network architectures, optimization methods and other parameters has been carried out. Conducting practical experiments meant the selection of the optimal values of the parameters and the description of the expediency of using the results of each experiment. Practical experiments were carried out on a set of real data on reviews for various categories of products on the Amazon marketplace, in the size of 400,000 examples. Based on the results obtained, models have been developed, according to the generalizing ability of neural networks, can quite accurately determine the sentimental sentiment of a text written in English.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Валідаційна вибірка – це набір даних, який використовується в процесі навчання моделі машинного навчання для оцінки якості епохи навчання;

Датасет – колекція однотипних даних, що застосовується в задачах машинної обробки даних;

API – Application Programming Interface (Прикладний програмний інтерфейс) – набір чітко визначених методів для взаємодії різних компонентів;

AI – Artificial Intelligence (Штучний Інтелект) – розділ комп'ютерної лінгвістики та інформатики, що опікується формалізацією проблем та завдань, які подібні до дій, що виконує людина;

GPU – graphics processing unit (графічний процесор)– окремий пристрій персонального комп'ютера,що виконує графічний рендеринг;

NLP – Natural language processing(Обробка природної мови) – міждисциплінарна галузь науки, що охоплює методики обчислювальної лінгвістики та теорії штучного інтелекту, основним проблемним полем якої є забезпечення взаємодії людських комунікативних актів та комп'ютерних систем.

ВСТУП

З часів виникнення інформаційних технологій людство навчилося використовувати їх засоби для вирішення широкого кола задач. Спектр автоматизації стрімко розширювався як з кількісного боку, поглинаючи нові предметні галузі, так і з якісного, підвищуючи складність методів вирішення та, власне, якість вирішення задач. На сьогодні вже досить важко переоцінити вплив засобів автоматизації процесів навіть на повсякденне життя, оскільки практично неможливо уявити повноцінне життя без використання тих чи інших здобутків інформаційних технологій. Окремо варто зазначити вплив світової пандемії коронавірусу на цей процесс: повсюдна необхідність вести діяльність віддалено значно прискорила попит та, як наслідок, темпи росту інформаційних технологій.

Незважаючи на значну розповсюдженість засобів автоматизації, у вирішенні задач діяльності людини існують задачі які визнано такими,що підлягають вирішенню тільки людиною. Нехарактерним та неповним прийнято вважати вирішення таких задач машиною. Досить часто мова йде про те, що для вирішення таких задач потрібні:

- творчість;
- художня уява;
- глибокий культурно-соціальний контекст.

Перелічені риси притаманні лише людській свідомості та жодним чином не здатні бути визначено задекларованими у площині комп’ютерних обчислень. Навіть у рамках людської свідомості вищезазначені елементи значно відрізняються від людини до людини.

Відповідно до загального визначення засобів штучного інтелекту, саме для виконання таких функцій, які традиційно є привілеями людського розуму, він і був створений. Його чисельні парадигми та підрозділи охопили фінансовий сектор, військову справу та державні спецслужби, важку

промисловість, медицину, менеджмент персоналу, музику, видавничу справу, телекомунікації, ігрову індустрію, транспорт. Така широка присутність у галузях діяльності пояснюється різноманіттям вирішуваних завдань:

− обробка природної мови: інформаційний пошук, машинний переклад, генерація текстів;

− комп'ютерний зір: розпізнавання образів, модифікація зображень, відстеження об’єктів;

− менеджмент знань: структурування та вивід нових знань з існуючої бази, експертні системи;

− машинне навчання: самостійне отримання знань з природи предметної галузі, розпізнавання образів з вчителем та без, прогнозування, нейронні мережі.

Особливе значення мають задачі пов’язані з результатами психологічних процесів та процесів мислення. Генерацію та аналіз тексту, зображень та мовний переклад на перший погляд зовсім неможливо довірити комп’ютеру, проте існує розділ штучного інтеллекту, який ставить на меті імітацію, власне, мислення у рамках сфери вирішуваної функції, і надає можливість вважати вищеописані задачі близькими до вирішення за допомогою машинних засобів. Цей розділ відомий як штучні нейронні мережі. У основі полягає концепція низькорівневої імітації роботи мозку живої істоти, а саме мережі нервових клітин, що виконують передачу сигналів між собою та результуючу обробку. Моделювання таких природних для організму процесів процесів значно розширює перспективи автоматизації.

Серед задач, що пов’язані з результатами процесів мислення, особливо виділяється задача сентиментального аналізу тексту. Вона є дуже актуальною та прогресивною. Це пояснюється швидкими темпами зростання інтернет спільноти і визначенням Інтернету як платформи думок

і суспільної думки взагалі. У просторі, де існує вільна можливість виказати свою думку і у той же час аналізувати чужі думки, засіб, що обробляючи великі об’єми інформації здатен давати узагальнення суспільної думки про певне явище, буде дуже доречним.

Тема сентиментального аналізу є дуже складною з морфологічної точки зору. Проблемними питаннями можуть стати в першу чергу структура мови, на якій ведеться аналіз тексту та її нормалізація, жаргонізми, сленгові вирази, різні види помилок.

У цій роботі буде розглянуто аналіз сентиментальної тональності природної мови засобами нейронних мереж. Сентиментальність є, мабуть, найбільш нехарактерним і не невловимим для комп’ютера явищем, що підкреслює світовий науковий інтерес до питання визначення емоції та тону у засобах передачі інформації.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Опис предметної галузі

Обробка природної мови – один із найважливіших напрямів досліджень у галузі штучного інтелекту. Зародження цього напряму пов'язують з появою перших обчислювальних машин і з ідеєю необхідності використовувати машини для вирішення корисних завдань, пов'язаних з природною мовою, якою розмовляють і пишуть люди. Обробка природної мови спирається на багато дисциплін: від класичних мовних наук, як-от лінгвістика, морфологія та синтаксис, до суттєво більш технічних інформатики та комп'ютерної лінгвістики.

Ще шість років тому NLP переважно вбирала у собі техніки та методи з інших галузей, але згодом вона почала експортувати їх. Методи, які розвинулися у сфері аналізу природних мов, почали успішно застосовуватися й інших областях. Зараз при аналізі природних мов теж починають широко застосовувати глибокі нейромережі.

Однією з центральних проблем для IT-рішень штучного інтелекту є завдання «розуміння» тексту, тобто, отримання сенсу з тексту природною мовою. Саме до неї, зрештою, зводяться практичні рішення розумних мовних технологій. Проте вона дуже далека від рішення у загальному вигляді, адже феномен чи специфікацію «розуміння» поки що не можуть пояснити чи змоделювати ні психологи, ні нейрофізіологи.

Особливо важливо підкреслити, що з точки зору аналізу інформації, природна мова є чи не найбільш неструктурованим та повним протиріччя видом вхідних даних. Людська мова досить складна і різноманітна для примітивної формалізації. Ми висловлюємо свої думки як усно, так і письмово. Окрім факту існування сотень мов та діалектів, варто зазначити факт унікальності граматичних та синтаксичних правил, термінів та сленгу

у просторі, навіть, однієї мови. З огляду на це, перед аналізом природної мови постають наступні типові проблеми:

– синтаксична невизначеність: «Час – не кінь, не підженеш і не зупиниш» для програми може бути абсолютно не зрозуміло, про що саме йдеться у реченні, про коня чи про час;

– смислова невизначеність: полягає у багатозначності елементів тексту. Розглянемо питальне речення «Де знайти ключ до того замку?» слово замок може мати два абсолютно різні значення, зважаючи на поставлений наголос або контекст;

– відмінкова невизначеність: у фразах «Усі були схвильовані перед концертом» та «Перед автівки був вщент розбитий» слово перед означає час або місце, що абсолютно змінює сенс фрази;

– референційна невизначеність: у фразі «Відкрий поличку та дістань мокру парасольку, я хочу її висушити» займенник її за смисловим значенням матиме відношення до мокрої парасольки, проте для машини, у якої повністю відсутнє розуміння реальності, даний займенник відноситиметься як до полички, так і до парасольки.

1.2 Огляд структури задач обробки природної мови

В цілому обробка природної мови є величезним спектром задач різного рівня.

Задачі рівня сигналу полягають у наявності певного типу вхідних даних, вигляд яких не дозволяє зробити його обробку. Такими даними можуть бути безпосередньо мова, рукопис або друкований відсканований текст. Суть задачі полягає у отриманні цифрового представлення вхідних даних, з яким може працювати програма. Отже, прикладами таких задач є машинне розпізнавання тексту, мови та їх синтез.

Рівень слова розглядає його багатомірність як лексичної одиниці, оскільки воно може бути описано багатьма характеристиками з боку лінгвістичних наук, що впливає на його вживаність у тій чи іншій ситуації. Типовими задачами тут є морфологічний аналіз, корекція помилок, нормалізація словоформ за допомогою стемінгу чи лематизації.

Наступний рівень – рівень словосполучень. Тут ми вже оперуємо поняттями частин мови і їх співвідношеннями. Так як у деяких випадках визначення частини мови є неоднозначним без використання контексту вживання, то цю задачу неможливо помістити на рівень слова. Отже, на рівні словосполучень ми маємо задачі відокремлення слів, визначення частини мови та визначення іменованих сутностей.

Словосполучення формують речення, а тому і наступний рівень теж пов’язаний із простором речення. На ньому речення потребують виділення, синтаксичного аналізу членів речення, усунення неоднозначності слів, усунення референційної невизначеності.

Виходячи за рамки одного речення, ми потрапляємо на рівень абзаців. Сутність, описана у одному реченні, цілком закономірно може мати посилання у інших. Тому виникає питання розв’язання посилань і встановлення відносин між об'єктами, згаданими у різних реченнях. З абзацами, окрім розв’язання посилань і встановлення відношень, ми можемо вирішувати нові завдання: проаналізувати емоційну тональність тексту, визначити якою мовою він написаний.

Абзаци формують рівень документів. Документ є вже повністю самостійною одиницею інформації, з точки зору її розгляду під час аналізу. Тут варто згадати задачу семантичного аналізу тексту, тобто визначення змісту та автоматичного анотування. Окремим випадком задачі рівня документу є машинний переклад текстів.

Найбільшим та останнім рівнем задач обробки природної мови є рівень корпусу. Він містить задачі дедублікації та розгорнутого пошуку на великому корпусі документів [1].

Про зазначені вище рівні потрібно також мати на увазі наступні факти:

– кожний з рівнів задач, обов’язково потребує використання деяких задач нижчих рівнів. Так, наприклад, без відокремлення слів, що є задачею рівня словосполучення, неможливо уявити розв’язання задачі синтаксичного аналізу членів речення, що є задачею рівня речення;

– на всіх рівнях завдання фактично йдуть у два боки: пов'язані з розбором існуючої мови і з генерацією нового матеріалу.

2 ПОСТАНОВКА ЗАДАЧІ

2.1 Класифікація обраної задачі

Вирішувана задача відноситься до рівня документу, а отже поглинає у собі деякі задачі нижчих рівнів:

– на рівні сигналу потрібно визначити яким чином буде здійснюватися конвертація тексту у зручне та репрезентативне для програмної обробки представлення;

– на рівні словосполучення та речення нас цікавить задача відокремлення слів.

2.2 Визначення вимог до функціоналу

Дана робота ставить на меті отримання системи, здатної оцінити емоційне забарвлення тексту довільної довжини.

Емоційне забарвлення може бути визначено здебільшого як позитивне або негативне. Під текстом позитивного емоційного забарвлення будемо вважати текст, автор якого чітко дає зрозуміти що виражені емоції є позитивними, і зміст якого може включати схвалення чи захоплення чимось без домінуючого негативного підтексту. Під негативним текстом матимемо на увазі текст, автор якого чітко дає зрозуміти, що запропонована думка має негативний характер, і зміст якого може включати обурення, страх, розпач, погрози і невдоволення, токсичне ставлення.

Опціонально можна ввести поняття емоційно-нейтрального тексту, якщо його буде важко однозначно віднести до позитивного чи негативного класу. Додатково можна серед нейтрального виділити підкласи умовно-позитивного або умовно-негативного тексту

Архітектурне питання програмного застосунку вирішено розв’язати використанням нейронних мереж, оскільки дана структура здатна імітувати процеси наближені до реального мислення людини і теоретично забезпечити належну точність роботи. Використання нейронних мереж не зобов’язує дослідника приймати спрощувальні припущення, які несумісні з реаліями предметної області і створені для сумісності з більш простими засобами розв’язання задач. Так, наприклад, у галузі обробки природної мови існують архітектури нейронних мереж, у процесі роботи з якими зовсім необов’язково приймати припущення про те, що позиція одного слова у будь-якому тексті не залежить від позиції будь-якого іншого слова у тексті.

Використання нейронної мережі передбачає етап навчання. На ньому відбувається тренування мережі за допомогою вибірки розмічених даних, яка відповідає предметній галузі та вирішуванному питанню. Перед, власне, тренуванням необхідно зробити передобробку вибірки для виключення з неї невалідних для навчання елементів.

Для виконання завдання потрібно провести теоретичне дослідження, у рамках якого необхідно:

– на рівні словосполучення та речення нас цікавить задача відокремлення слів;

– обрати вибірку для тренування,що містить достатньо велику кількість екземплярів, розмічених на позитивну та негативну тональність тексту;

– обрати мову, середовище та програмні інструменти для навчання, виходячи з їх можливостей, потужності та перспективи використання;

– визначити які існують архітектури нейронних мереж, їх особливості та придатність до вирішення задач обробки природної мови;

– визначити оптимальний порядок та складові передобробки вхідних даних для тренування, тестування та використання програмного застосунку;

– обрати підходи для нормалізації словоформ вхідних даних;

– обрати спосіб цифрового представлення тексту;

– визначити інші, більш дрібні параметри навчання та передобробки даних.

Отримана система має:

– визначати клас поданого тексту щонайменше як позитивний або негативний. Опціональною є наявність нейтрального класу;

– виконувати етап навчання використовуючи оптимальну кількість апаратних ресурсів;

– виконувати видачу рішення використовуючи оптимальну кількість апаратних ресурсів;

– використовувати архітектуру штучних нейронних мереж.

Після отримання результатів теоретичних досліджень, потрібно переходити до реалізації системи та експериментальних досліджень визначених конфігурацій програмного застосунку.

3 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

3.1 Вибірка даних

Процес навчання нейронної мережі передбачає використання навчальної вибірки. Така вибірка є набором багатовимірних екземплярів, розмічених виходячи з характеристик предметної області та відповідно до потреб вирішуваної задачі.

Якщо розглядати загальні вимоги, не прив'язуюсь до конкретної задачі, то можна виділити три основних вимоги до неї:

– достатність – кількість екземплярів має бути достатньою для навчання. Для нейронної мережі необхідно, щоб кількість навчальних прикладів було в кілька разів більше, ніж кількість вагів міжнейронних зв'язків, у противному випадку модель може не набути здатності до узагальнення. Крім цього, розмір вибірки повинен бути достатнім для додаткового відокремлення тестової вибірки перед стартом навчання;

– різноманітність – велика кількість різноманітних комбінацій вхід-вихід у екземплярах. Здатність нейронних мереж до узагальнення не буде реалізована, якщо кількість екземплярів достатня, але вони однакові, тобто відповідають лише частині класів, характерних для вихідної множини;

– рівномірність присутності класів – екземпляри різних класів повинні бути представлені в навчальній вибірці приблизно в однакових пропорціях. Якщо один із класів буде переважати, це може призвести до спотворень у процесі навчання моделі, і цей клас буде визначений моделлю як найбільш ймовірний для будь-яких нових спостережень [2].

Відповідно до сформульованої задачі, оптимальна вибірка повинна виглядати як набір екземплярів з щонайменше двох атрибутів: текст природної мови та категорія його емоційного забарвлення.

Найбільш популярною вибіркою для вирішення задачі встановлення сентиментальної тональності є Large Movie Review Dataset. Тексти у ньому взяті з відгуків до фільму популярного сайту про кінематограф IMDB [3]. Відгуки присутні лише явно позитивні (оцінка більше 6\10) чи негативні (оцінка менше 5\10), нейтральні відгуки в набір даних не включалися. Загальний розмір набору даних 50 000 екземплярів. Кількість позитивних та негативних відгуків однакова.Зображення деяких екземплярів Large Movie Review Dataset наведено на рисунку 3.1.

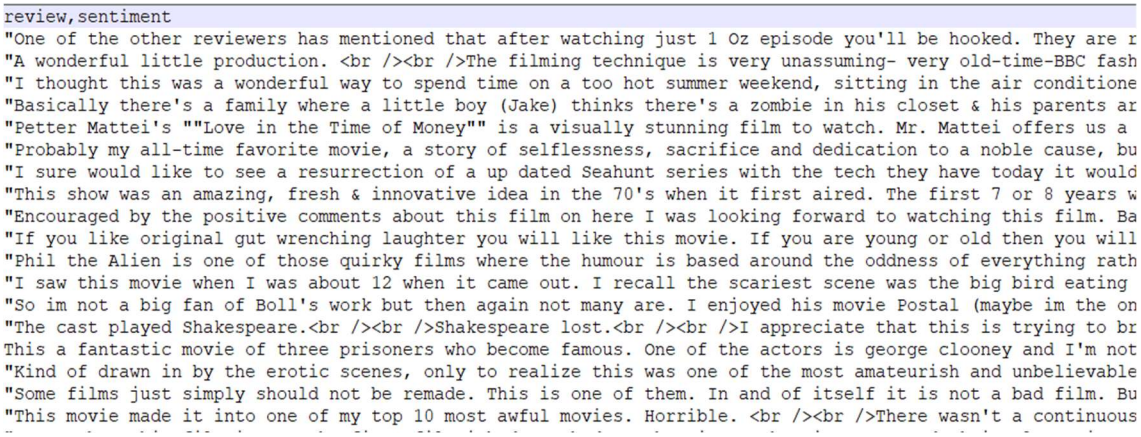

Рисунок 3.1 – Уривок вибірки Large Movie Review Dataset

До переваг датасету можна віднести:

– рівне співвідношення позитивних та негативних відгуків;

– через поширеність цього датасету, є багато його передоброблених різновидів;

Проте, є суттєві недоліки:

– замалий розмір повної вибірки – 50000 екземплярів. Враховуючи поділ на тестову та навчальну вибірку, навчання відбуватиметься на ще меншій кількості екземплярів;

– змістова однотипність екземплярів: оскільки відгуки відносяться лише для кінофільмів, то можливе зміщення якості аналізу в бік специфічної

для кіноіндустрії лексики та гірша робота з тематично-нейтральним текстом.

Наступним варіантом для розгляду є російськомовний корпус коротких текстів RuTweetCorp, створений аспірантом інституту систем інформатики ім.А.П. Єршова Юлією Рубцовою [4]. Джерелом текстів є російськомовний сегмент платформи мікроблогінгу Twitter. У корпусі, що складається з 114911 позитивних та 111923 негативних записів, містяться записи за час з кінця листопада 2013 року до кінця лютого 2014 року. Зображення деяких екземплярів датасету наведено на рисунку 3.2.

	Tweet	Class
1	коллеги сидят рубятся в urban terror, а я из-з...	0
2	@elina_4post как говорят обещаного три года жд...	0
3	желаю хорошего полёта и удачной посадки,я буду...	0
4	обновил за каким-то лешим surf, теперь не рабо...	0
5	котёнка вчера носик разбила, плакала и расстра...	0
...	...	...
226814	спала в родительском доме, на своей кровати.....	1
226815	rt @jebesilofyt: эх... мы немного решили сокра...	1
226816	что происходит со мной, когда в эфире #proacti...	1
226817	любимая,я подарю тебе эту звезду... имя како...	1
226818	@ma_che_rie посмотри #непытайтесьпокинутьомск ...	1

226818 rows × 2 columns

Рисунок 3.2 – Уривок вибірки RuTweetCorp

Переваги:

– корпус практично не містить нейтральних текстів та текстів менш ніж 40 символів;

– достатній розмір загальної вибірки – приблизно 225000 екземплярів;

– немає чіткої тематичної направленості;

– майже рівне співвідношення класів.

Недоліками є

– потреба значних зусиль для передобробки: велика кількість службових послідовностей платформи Twitter, посилань тощо;

– велика кількість лексичних помилок: етап передобробки додатково потребує їх виправлення, що не може бути гарантовано успішним;

– малоінформативність: після проведеної тестової передобробки середня довжина тексту – близько 5-8 слів, що є дуже малим значенням. Мережа, натренована на коротких повідомленнях зможе максимально успішно оцінювати також тільки короткі повідомлення. Окремо варто зазначити, що деякі види нейронних мереж враховують контекст мови і для них навчання на занадто коротких текстах не має сенсу.

Останьою розглянемо вибірку Amazon Reviews for Sentiment Analysis, знайдену на відкритому інтернет-ресурсі для машинного навчання Kaggle. Вона містить відгуки на товари з популярного маркетплейсу Amazon, що були оцінені у 4 або 5 зірок (трактуються як позитивні) та 1 або 2 зірки (трактуються як негативні) [5]. Відгуки з 3 зірками, тобто відгуки з нейтральними настроями, не були включені. Вибірка вже поділена на тестову (400000 екземплярів) та навчальну (близько 3600000 екземплярів) Зображення деяких екземплярів датасету наведено на рисунку 3.3. Перевагами можна назвати:

– достатню кількість екземплярів;

– тематичну нейтральність, оскільки об’єктами відгуків є різноманітні товари: музика, книги, техніка, медикаменти, тощо;

– відсутність необхідності поглибленої передобробки, у порівнянні з вибіркою RuTweetCorp;

	Class	Text
0	1	Great CD: My lovely Pat has one of the GREAT ...
1	1	One of the best game music soundtracks - for ...
2	0	Batteries died within a year ...: I bought th...
3	1	works fine, but Maha Energy is better: Check ...
4	1	Great for the non-audiophile: Reviewed quite ...
...	...	...
399995	0	Unbelievable- In a Bad Way: We bought this Th...
399996	0	Almost Great, Until it Broke...: My son recie...
399997	0	Disappointed !!!: I bought this toy for my so...
399998	1	Classic Jessica Mitford: This is a compilatio...
399999	0	Comedy Scene, and Not Heard: This DVD will be...

400000 rows × 2 columns

Рисунок 3.3 – Уривок повної вибірки Amazon Reviews for Sentiment Analysis

– достатню довжина текстів;

– рівне співвідношення класів.

Датасет також має певні недоліки,які з огляду на значну кількість переваг можна назвати умовними:

– виходячи з розміру тренувального набору даних, є необхідність значних апаратних ресурсів для її обробки;

– автор вибірки зазначає, що більшість оглядів написані англійською мовою, але незначна кількість відгуків написана іншими мовами, наприклад іспанською.

Найбільш оптимальним вибором є Amazon Reviews for Sentiment через тематичну нейтральність та достатню кількість екземплярів та відсутність серйозних недоліків. Розмір тестової вибірки дозволяє провести

процеси навчання та тестування лише на ній,тому для економії часу тренування у якості основної вибірки буде використана тестова. Саме цей датасет буде використовуватися у подальшій роботі.

3.2 Програмні ресурси

Для виконання роботи було вирішено обрати мову програмування Python.

Python є високорівневою мовою програмування загального призначення з автоматичним управлінням пам'яттю, орієнтованою на підвищення продуктивності розробника, читання коду та його якості, а також на забезпечення переносимості написаних на ній програм. Мова програмування Python останнім часом все частіше використовується для аналізу даних як у науці, так і комерційній сфері. Цьому сприяють наступні її переваги:

- простота опанування мови;
- спрощена робота з обчисленням надвеликих значень;
- велика різноманітність відкритих бібліотек, що значно спрощують роботу з однотипним кодом побудови та конфігурації моделей навчання;
- велика спільнота використання та підтримки Python інфраструктури;
- регулярні оновлення вищезгаданих бібліотек.

Python спільнота налічує багато зручних інструментів для роботи з нейронними мережами. Найбільш відомими з них є Keras та TensorFlow. TensorFlow – це бібліотека AI, яка допомагає розробникам створювати масштабні нейронні мережі з багатьма шарами, використовуючи графіки потоків даних. TensorFlow також полегшує побудову моделей глибокого навчання, просуває сучасну технологію ML/AI та дозволяє легко розгортати програми на базі ML.Keras – це високорівневий API TensorFlow для

створення та навчання коду глибоких нейронних мереж. З Keras, як обгорткою для дуже потужного TensorFlow, статистичне моделювання, робота із зображеннями та текстом є набагато легшою. Ця бібліотека дозволяє зосередитися на якісних характеристиках програмного рішення і буде основою виконуваної роботи.

Окрім побудови, власне, нейронних мереж, для навчання є необхідним мати зручні інструменти для різного роду обробки даних з датасету. Основна робота з завантаженням та обробкою датасету буде виконуватися пакетом Pandas. Він надає високоефективні, прості у використанні структури даних та інструменти аналізу та призначений для швидкої та простої обробки даних, читання, агрегування та візуалізації. Pandas бере дані у файлі CSV або TSV або базу даних SQL і створює об'єкт Python з рядками та стовпцями, який називається датафреймом даних. Фрейм даних дуже схожий на таблицю у статистичному програмному забезпеченні, наприклад, Excel або SPSS.

Для деяких робіт з багатовимірними даними буде використовуватися пакет NumPy. Він надає високопродуктивні об'єкти багатовимірних масивів та інструменти для роботи з масивами. NumPy використовується для обробки масивів, в яких зберігаються значення одного і того ж типу даних. NumPy полегшує математичні операції над масивами та їх векторизацію. Це значно підвищує продуктивність та, відповідно, прискорює час виконання.

Для візуалізації результатів навчання буде використано бібліотеку Matplotlib. Вона надає API для вбудовування графіків у програмні рішення,включаючи гістограми, стовпцеві діаграми, точкові діаграми та кругові діаграми.

Для роботи з NLP-специфічними аспектами буде використано NLTK (Natural Language Toolkit) бібліотеку. Вона містить засоби обробки тексту, за допомогою яких ви можна виконувати токенізацію, парсинг, класифікацію, виділення, тегування та семантичне обґрунтування даних.

3.3 Апаратні ресурси та середовище

Для виконання, власне, процесу обчислень було вирішено використовувати платформу Google Colaboratory – безкоштовне інтерактивне хмарне середовище для роботи з кодом від Google. Фактично це онлайн імплементацією відомої технології Jupyter Notebook – командної оболонки для інтерактивних обчислень на Python. Файл Google Colaboratory є схожим на традиційні python-скрипти, але код там розташовано у виконуваних комірках, які можуть бути довільних розмірів. Вони виконуються незалежно, з точки зору порядку, проте виконуються у одному середовищі, з точки зору процесу виконання. У безкоштовній версії середовище виконання діє 12 годин і видаляється після їх закінчення або після 30 хвилин бездіяльності.

У порівнянні з традиційним підходом, який передбачає виконання коду на локальній машині, існує ряд значних переваг:

− безкоштовно Google Colaboratory надає 12.69 GB оперативної пам’яті, які використовуються лише для обчислень, а не розділяються з потребами операційної системи та інших процесів, як у випадку з обчисленнями на локальній машині;

− немає обмежень на кількість активних середовищ;

− для середовища, окрім потужного процесора (CPU), є можливість безкоштовно отримати один графічний процесор (GPU) NVidia Tesla K80, що значно підвищує швидкість обчислень, через особливості GPU архітектури;

− Google Colaboratory працює як розподілена система, аналогічно до Google Документів;

− у середовище автоматично імпортовано пакети Keras та TensorFlow.

Екранна форма інтерфейсу Google Colaboratory наведена на рисунку 3.4.

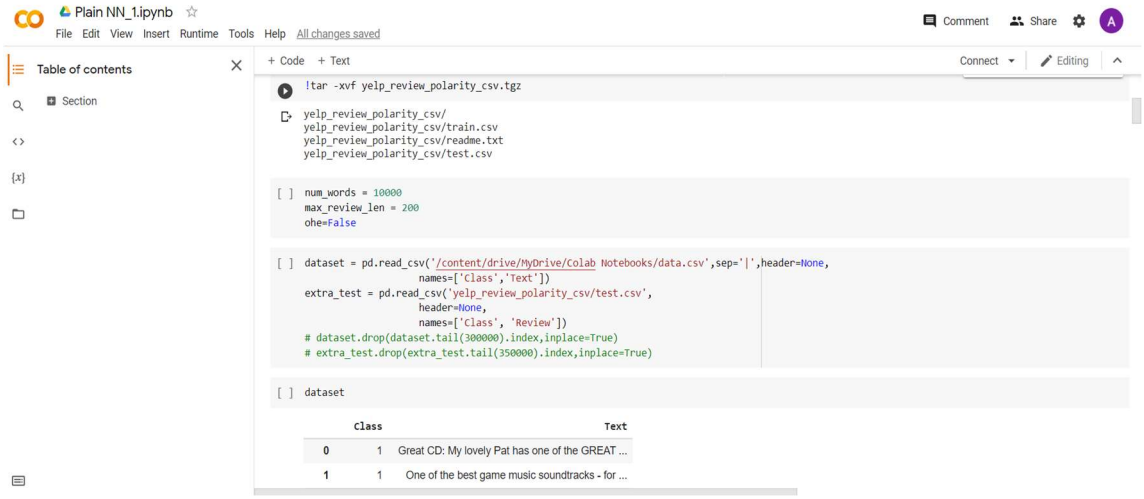

Рисунок 3.4 – Екранна форма інтерфейсу Google Colaboratory

3.4 Векторизація природної мови

Векторизацією називається процес отримання цифрового подання тексту. Оскільки нейронні мережі приймають на вхід цифрову інформацію, текст має бути певним чином змінено для обробки нейромережею. На сьогодні, у NLP галузі відомо 3 основних методи векторизації тексту:

- пряме кодування слів;
- one-hot-encoding;
- метод щільного вектору.

Метод прямого кодування слів передбачає наявність словника унікальних слів на базі навчальної вибірки. Власне, код слова може визначатися як порядковий номер у словнику, кодова послідовність символів слова у стандарті кодування або частота зустрічі слова у вибірці. Варто зазначити, що останній підхід є неоптимальним, через можливу колізію кодів, якщо будь-які два слова у вибірці будуть зустрічатися однакову кількість раз. Ситуації, коли є різні слова, яким відповідає один і той самий код, можуть спровокувати зниження якості оцінки тональності. Навіть з використанням порядкового номеру як коду, метод прямого

кодування є досить примітивним та малоефективним через той факт, що він є приватним випадком цілочисельного кодування ознак. Нейронні мережі погано працюють у випадку коли, категоріальні або частково-категоріальні ознаки кодуються цілими числами через неперервну природу останніх. Неперевність атрибуту свідчить про те, що його значення належать до неперевного числового простору та на множині цих значень змістовно задається відношення «більше», «менше». Різниця двох значень таких характеристик має зміст. Прикладами таких характеристик можуть бути зріст та вага людини. Категорійна природа свідчить про те що, домен атрибтуту задано множині дискретних значень та їх рівність чи не рівність може бути лише абсолютною. Встановення відношення «більше» або «менше», арифметичні дії, лінійні порівняння є нерелевантними, через їх якісний характер.

One-hot-encoding передбачає трансформацію тексту за допомогою його представлення як вектору, виміри якого позначають присутність певного слова у тексті і розташовані у порядку словника вибірки. Схематичне зображення цієї моделі наведено на рисунку 3.5.

Таким чином, трансформований текст є вектором довжини словника і у цьому векторі стоять одиниці на ознаках, що відповідають присутнім у тексті словам. Через агрегацію всіх присутніх слів у векторі такий метод також відомий як Bag-of-words. Цей засіб добре передає категоріальну природу слова у тексті, але має 2 недоліки. Він не вирішує проблему врахування взаємозв’язків та порядку слів у тексті та потребує значних витрат оперативної пам’яті для зберігання великих розріджених векторів, проте є доцільним до використання.

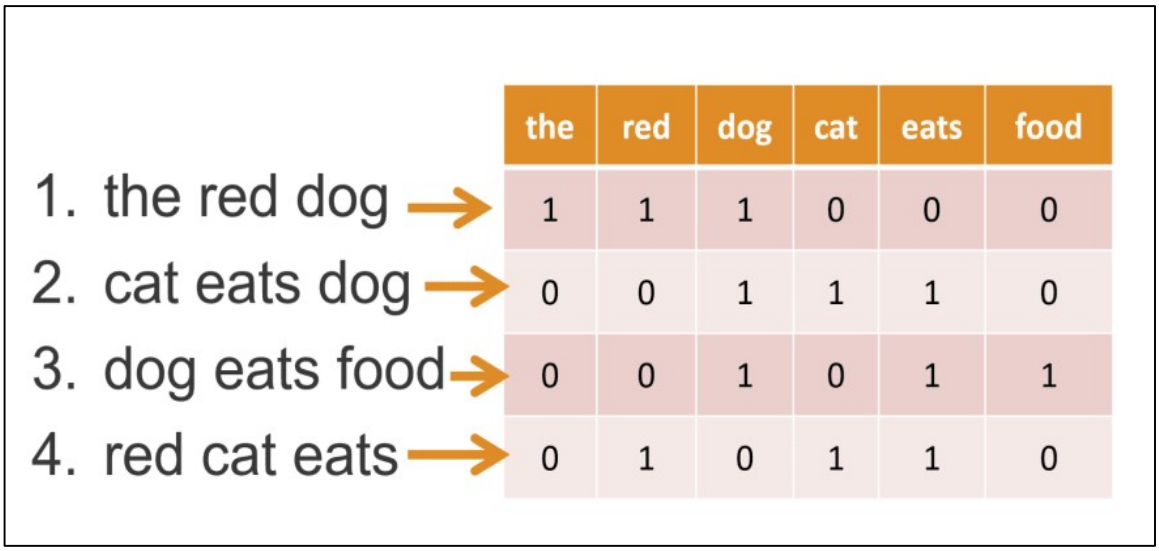

Рисунок 3.5 – Візуалізація методу векторизації тексту Bag-of-words з використанням one-hot-encoding

Останнім розглянемо метод щільного вектору. У ньому кожне слово у тексті трансформується у вектори довільних чисел однакової довжини. Власне, числа для кожного конкретного слова визначаються у процесі навчання. Архітектурно такий засіб можна уявити як вхідний шар нейронної мережі, який є не шаром нейронів у класичному розумінні, а матрицею вагів для генерації подання слова. Розмірність матриці визначається кількістю слів у словнику (для стовпчиків) та зазначеним розміром вектору, а отже кожний стовпчик матриці відповідає щільному векторному поданню певного слова. На першому етапі навчання матриця ініціюється випадковими числами, а потім, за допомогою методу зворотного поширення помилки, змінюються до оптимальних. Схематичне зображення роботи методу щільних векторів наведено на рисунку 3.6. Подання слів у вигляді щільних векторів або так званих embedding – спосіб ефективно з точки зору використання пам'яті уявити слово у вигляді, зрозумілому для нейронної мережі.

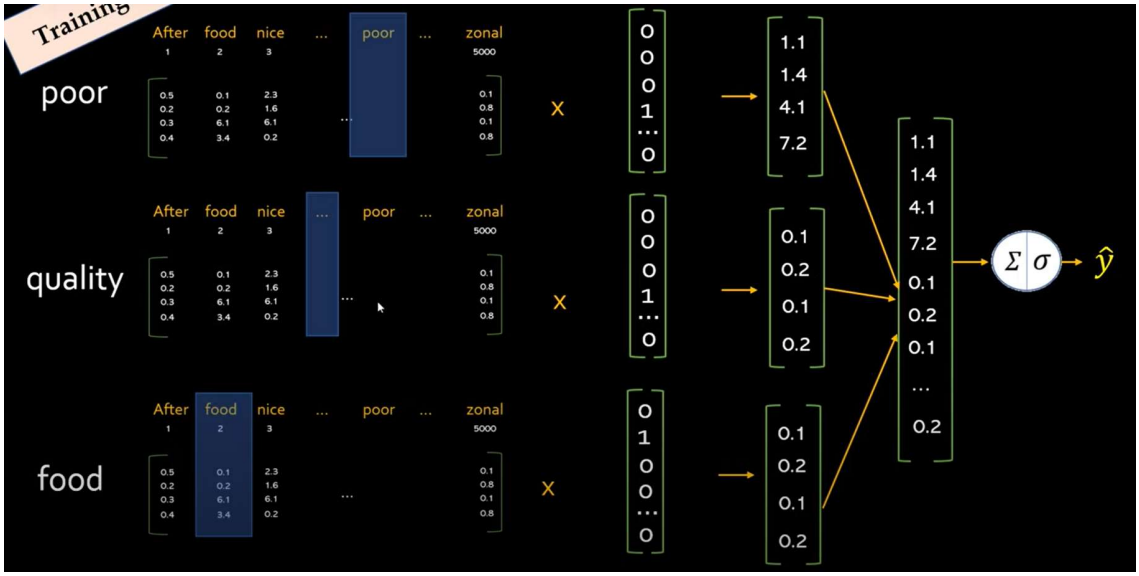

Рисунок 3.6 – Візуалізація методу щільного вектора для векторизації тексту

При цьому однакові слова мають однаковий вектор, що їх представляє. Розмір вектора задається дослідником і найчастіше варіюється від 8-мірного для невеликих датасетів, до 1024 у випадку великих текстових баз. Embedding подання з високою розмірністю дозволяють виокремлювати тонкі нюанси у взаємозв'язках між словами, але вимагають дуже великих баз для ефективного навчання для виправдання помітно більших витрат процесорного часу та пам’яті.

Окремою перевагою є фізичний зміст щільних векторів. Оскільки кожне слово у такій концепції має кілька вимірів, то цілком природно вважати, що кожен вимір буде відповідати за певну характеристику. Так, наприклад, якщо ми у рамках нейромережі тренуємо щільне векторне подання слів із вектором розмірністю 2 та відобразимо деякі слова у координатній площині, то побачимо, що координати слів підлаштувалися таким чином, що синоніми будуть розташовані відносно близько один до одного, а антоніми – на протилежних кінцях координатної площини. Візуалізацію описаного явища зображено на рисунку 3.7

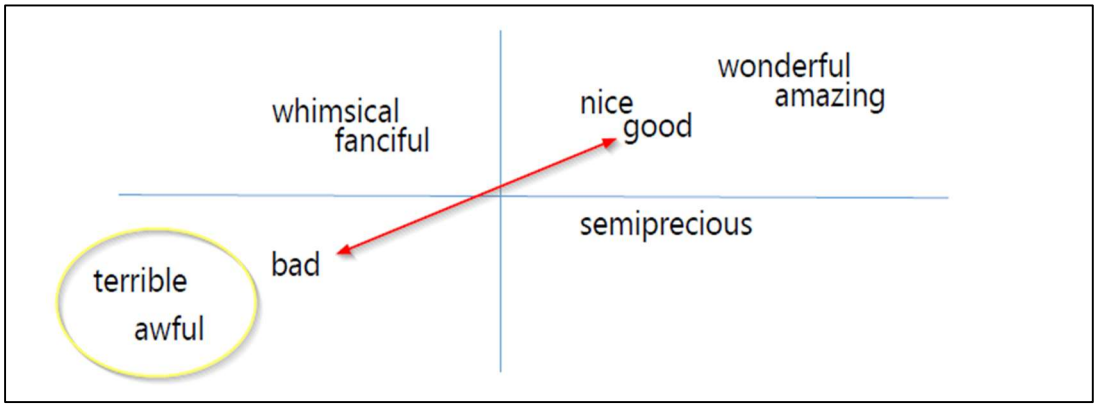

Рисунок 3.7 – Візуалізація явища упорядкування слів при використанні методу щільного вектора подання слів

Розглянутий випадок є прикладом лише одного відношення, яке здатне відчувати embedding подання слова. Маючи більшу кількість вимірів, embedding подання здатне відчувати інші досить складні за встановленням у машинній обробці відношення:

− відношення роду слова, тобто ставити поряд семантично однакові слова різного роду (король – королева), у той час як семантично різні слова не будуть присутні у зазначених паралелях, за виключенням випадків коли різна семантична природа все ж таки зберігає смислову близькість, яка є доречною (курка –півень);

− відношення числа слова, тобто множина та однина слова стоять поруч у просторі, тощо;

− часу, до якого належить словоформа. Валідне для дієслів відношення, коли форми різного часу одного й того ж дієслова розташовані впорядковано у координатному просторі;

− більш глибокі відношення. Прикладом такого відношення може бути близькість розташування у просторі таких слів як вівця та ягня. Дуже різні за написом слова позначаютьодну і ту ж тварину різного віку.

Візуалізацію описаного явища зображено на рисунку 3.8.

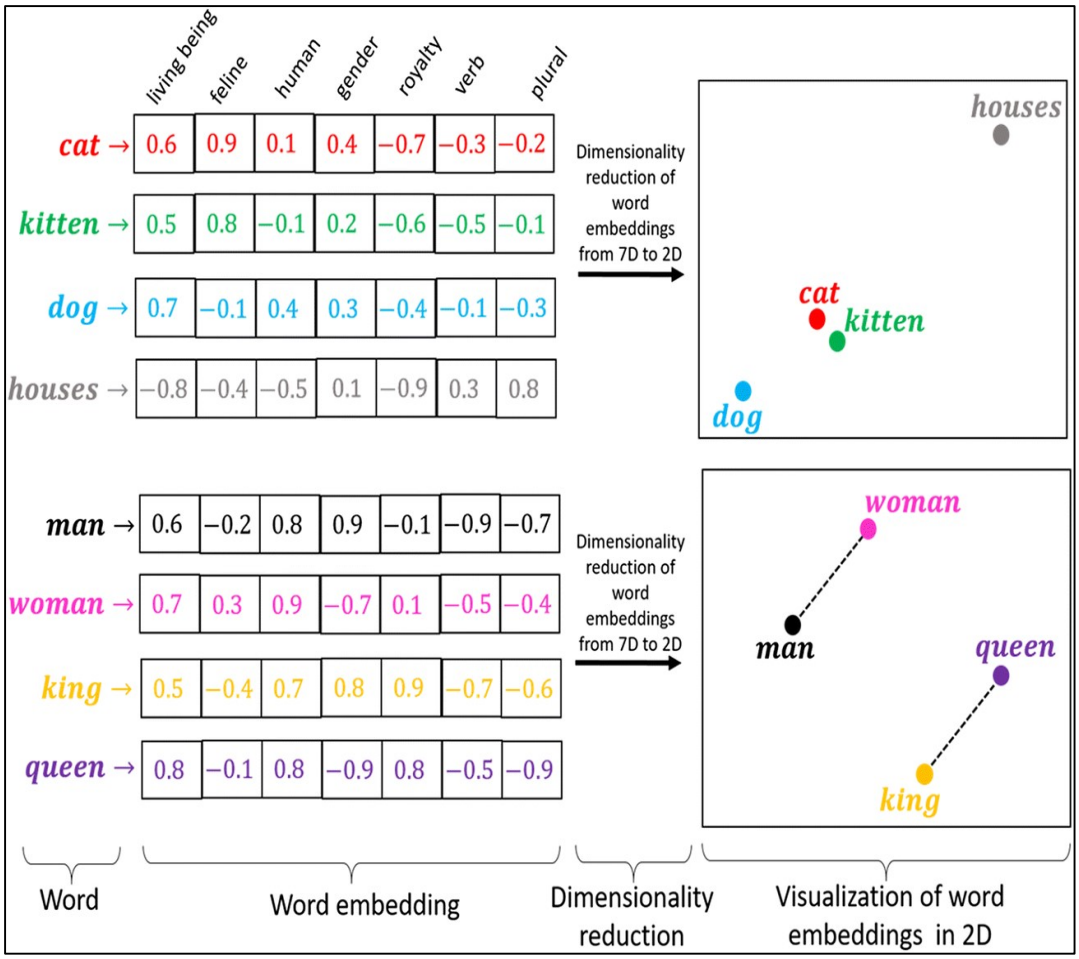

Рисунок 3.8 – Візуалізація явища встановлення семантичних відношень при використанні методу щільного вектора подання слів

Якщо embedding шар є частиною рекурентної нейронної мережі, то він послідовно віддає подання слів подальшим шарам, а у випадку звичайної мережі прямого поширення – вектори слів послідовно конкатенуються у загальний вектор тексту перед переходом до наступних шарів.

Враховуючи розглянуті переваги та недоліки методів векторизації тексту, було зроблено наступні висновки щодо їх подальшого використання у роботі:

– one-hot-encoding: незважаючи на недоліки є досить вичерпним методом, тому буде використовуватися у дослідах;

− метод щільного вектору: є досить зручним для використання і якісним з боку відображення відношень між словами;

− метод прямого кодування слів: через примітивність та наявність протиріч, вирішено не використовувати у подальших дослідах.

3.5 Передобробка вибірки

Передобробка тексту перед процесом навчання зазвичай включає в себе наступні етапи:

− видалення нетекстової інформації: цифри, спеціальні символи, зайві пробільні символи;

− приведення тексту до малих літер;

− вилучення стоп-слів: слова, які не несуть змістового навантаження, тому їх користь та роль для аналізу не суттєва. До них можна віднести прийменники, суфікси, дієприкметники, вигуки, числівники тощо;

− стемінг: процес скорочення слова до основи шляхом відкидання допоміжних частин, таких як закінчення чи суфікс;

− токенізація: розбиття тексту на слова;

− трансформація у цілочисельний вигляд: обчислення словнику вибірки та кодування слів відповідно до нього.

3.6 Архітектури нейронних мереж

3.6.1 Нейронні мережі прямого поширення

Нейронні мережі прямого поширення є найбільш простою архітектурою нейромереж. У них сигнали поширюються в одному напрямку, починаючи від вхідного шару нейронів, через приховані шари до вихідного шару і на вихідних нейронах отримується результат опрацювання сигналу.

Налаштування вагів для зв'язків між нейронами здійснюється методом зворотного поширення помилки (backpropagation). Початкові значення вагів моделі задаються випадковими числами. Приступаючи до їх корекції, окрім, власне, вагів потрібно розуміти структуру мережі та мати вибірку екземплярів з певними атрибутами та бажаним результатом на кожне спостереження. Отже, такий екземпляр, проходячи крізь мережу, буде створювати певні значення зважених входів у нейрони і виходів з них як результат роботи активаційних функцій. Ці значення знадобляться на наступних етапах тому їх варто запам’ятовувати. На вихідному шарі буде отримано результат проходу екземпляру через мережу і обчислена його різниця з бажаним спостереженням на кожному нейроні вихідного шару. Ця різниця буде називатися помилкою мережі. За допомогою значення помилки та похідної від функції активації нейрону, обчислюється значення локального градієнту на нейронах вихідного шару:

$$\delta = e \cdot f'(v_{out}). \tag{3.1}$$

Маючи значення градієнту ми можемо виконати корекцію вагів, пов’язаних з вихідним нейроном. Нове значення ваги буде дорівнювати різниці його поточного значення з добутком знайденого градієнту, сигналу що проходив по цьому зв’язку та кроку збіжності. Крок збіжності визначається дослідником вручну та зазвичай розташований у межах від 0,1 до 0,001. Переходимо до нейрона наступного з кінця шару і для його вхідних зв'язків повторимо ту саму процедуру. Локальний градієнт останнього нейрона зважується вагами поєднаних зв’язків. Отримані значення кожному нейроні множаться на похідну функції активації, взяту в точках вхідної суми. Отже, процес коректування вагових коефіцієнтів є оберненим до процесу проходу екземпляру крізь нейронну мережу, починаючись з самого останнього шару моделі і доходячи до самого першого.

Кількість шарів та нейронів на шар обирається дослідником експериментальним шляхом. В мережах такого виду немає зворотних зв’язків. Типову схему такої архітектури наведено на рисунку 3.9.

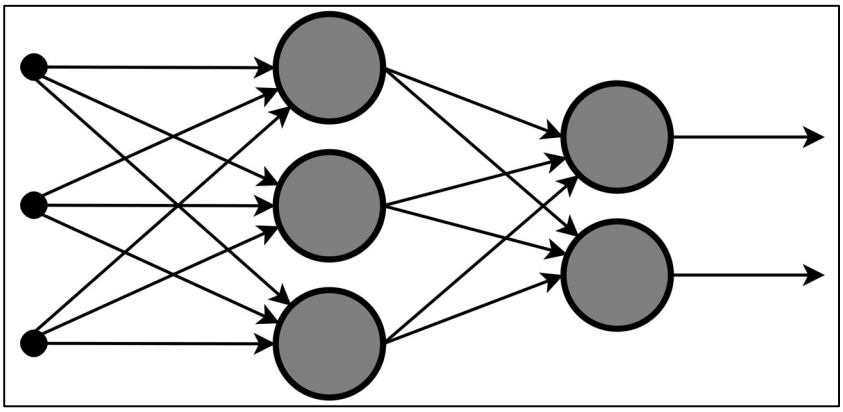

Рисунок 3.9 – Схематичне зображення архітектури повзнозв’язних нейронних мереж прямого поширення

Також екпериментальним шляхом обирається функція активації – засіб трансформації сигналу на етапі проходу через нейрон. Розглянемо основні різновиди.

Лінійна – є звичайною прямою лінійною залежністю. Функція пропорційна сигналу на вході входу (тобто зваженої сумі цьому нейроні). Формула виглядає наступним чином:

$$(x) = kx. \tag{3.2}$$

Вона дозволяє отримати діапазон значень на виході, а не тільки нуль або одиниця, що допомогає підвищити точність при застосуванні у задачах класифікації. Але лінійна функція має значний недолік. Він полягає у неможливості використання методу зворотного розповсюдження помилки. Так як в основі цього методу навчання лежить градієнтний спуск, який у

своїй роботі оперує значенням похідної, яка для цієї функції активації є константою. Тобто при оновленні ваг не можна сказати чи потребує існуюча вага на зв’язку змінення значення;

Сигмоїдна – гладка монотонно зростаюча функція, що виглядає наступним чином:

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{3.3}$$

І оскільки ця функція нелінійна, її можна використовувати в нейронних мережах з безліччю шарів, і навчати за методом зворотного поширення помилки. Сигмоїда обмежена двома горизонтальними асимптотами y=1 та y=0, що дає нормалізацію вихідного значення кожного нейрона, яка є перевагою при використанні сигмоїди у задачах класифікації. Графік функції наведено на рисунку 3.10.

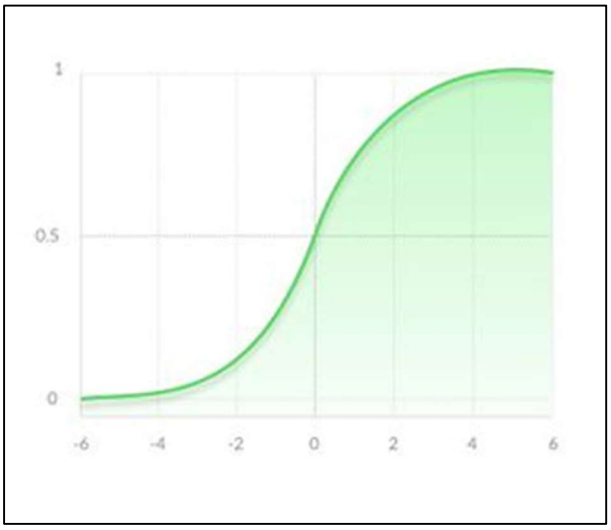

Рисунок 3.10 – Графік сигмоїдної функції активації

Значним недоліком є проблема зникаючого градієнта: похідна такої функції дуже мала у всіх точках, крім порівняно невеликого проміжку [6].

Це сильно перешкоджає формуванню градієнту, оскільки він стає стає близьким до нулю, та модель має схильність до перенавчання.

Функція гіперболічного тангенсу. Є видозміненою сигмоїдною функцією з нормалізацією у проміжку від -1 до 1. Має вигляд:

$$\text{th}(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}. \tag{3.4}$$

Фактично зберігає переваги та недоліки сигмоїди. Відсутність необхідності у нормалізації є найбільш вживаним сценарієм використання цієї функції активації. Область визначення цієї функції активації центрована щодо нуля, що знімає обмеження при підрахунку градієнта для переміщення у певному напрямку. Графік функції зображено на рисунку 3.11.

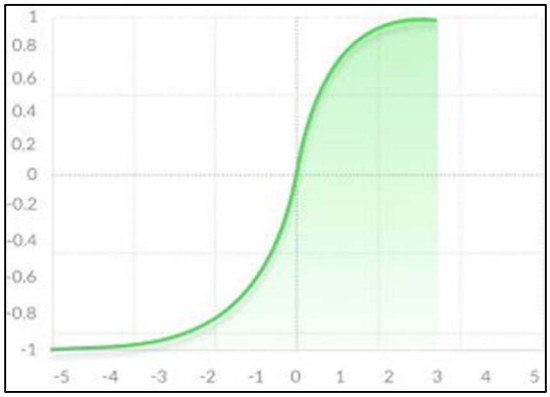

Рисунок 3.11 – Графік функції активації гіперболічний тангенс

Rectified Linear Unit (ReLU) – ця функція повертає нуль, якщо приймає негативний аргумент, у противному випадку, функція повертає аргумент як результат роботи. Тобто вона може бути записана наступною формулою:

$$f(x) = \max(0,z). \tag{3.5}$$

У такої функції дуже швидко і просто обчислюється похідна: для негативних значень – 0, для позитивних – 1. Графік функції наведено на рисунку 3.12.

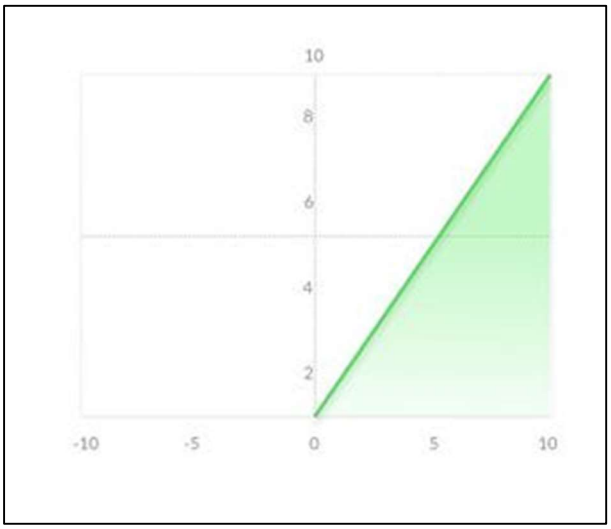

Рисунок 3.12 – Графік функції активації Rectified Linear Unit (ReLU)

Ця функція має один суттєвий недолік, що називається проблемою вмираючого ReLU. Так як частина похідної функції дорівнює нулю, то градієнт для неї буде нульовим, а це означає, що ваги не будуть змінюватися під час градієнтного спуску і нейронна мережа перестане навчатися.

Leaky ReLU – модифікація звичайного ReLU з вирішенням проблеми «вмирання» [7]. Графік функції активації на негативних значеннях утворює не горизонтальну пряму, а похилий, з невеликим кутовим коефіцієнтом (порядку 0,01). Тобто вона може бути записана як

$$f(x) = \begin{cases} 0{,}01x, & x < 0 \\ x, & x \geq 0 \end{cases}. \quad (3.6)$$

Графік функції наведено на рисунку 3.13.

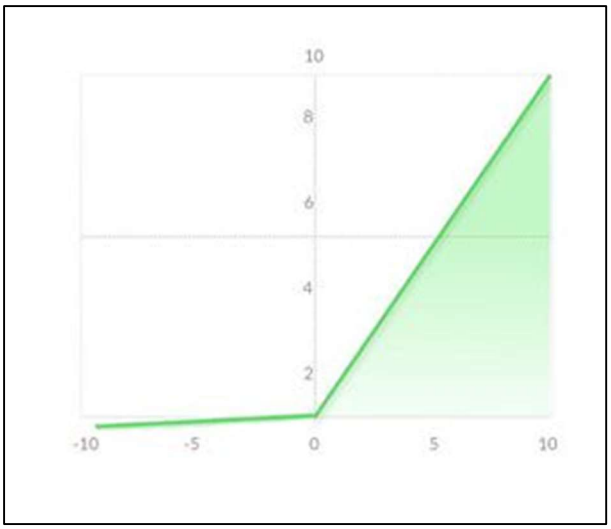

Рисунок 3.13 – Графік функції активації Leaky ReLU

Похідна такої функції на від’ємних значеннях буде ненульовою, що вирішує головну проблему класичного варіанту, але нівелює перевагу простої похідної, повертає проблему зникаючого градієнту та додає коефіцієнт як параметр до налаштування нейронної мережі.

Під час навчання існує ризик виникнення проблеми перенавчання. Вона полягає у тому, що модель добре пояснює лише приклади з навчальної вибірки, адаптуючись до навчальних прикладів, замість того, щоб вчитися класифікувати приклади, що не брали участь у навчанні (втрачаючи здатність до узагальнення). Проблему перенавчання у випадку його виникнення вирішено нейтралізувати методом Dropout. Головна ідея Dropout – на кожній ітерації зміни вагових коефіцієнтів частину нейронів потрібно виключати із заданою ймовірністю p, тобто при будь-яких вхідних даних або параметрах ці нейрони повертають 0. На кожній ітерації зміни вагових коефіцієнтів нейронної мережі частина нейронів штучно виключається з моделі. Кандидати на виключення обираються з зазначеною дослідником імовірністю. У результаті загальне число нейронів залишається незмінним, але їх спеціалізація та узагальнююча властивість зростає. Схематична візуалізація методу dropout зображена на рисунку 3.14.

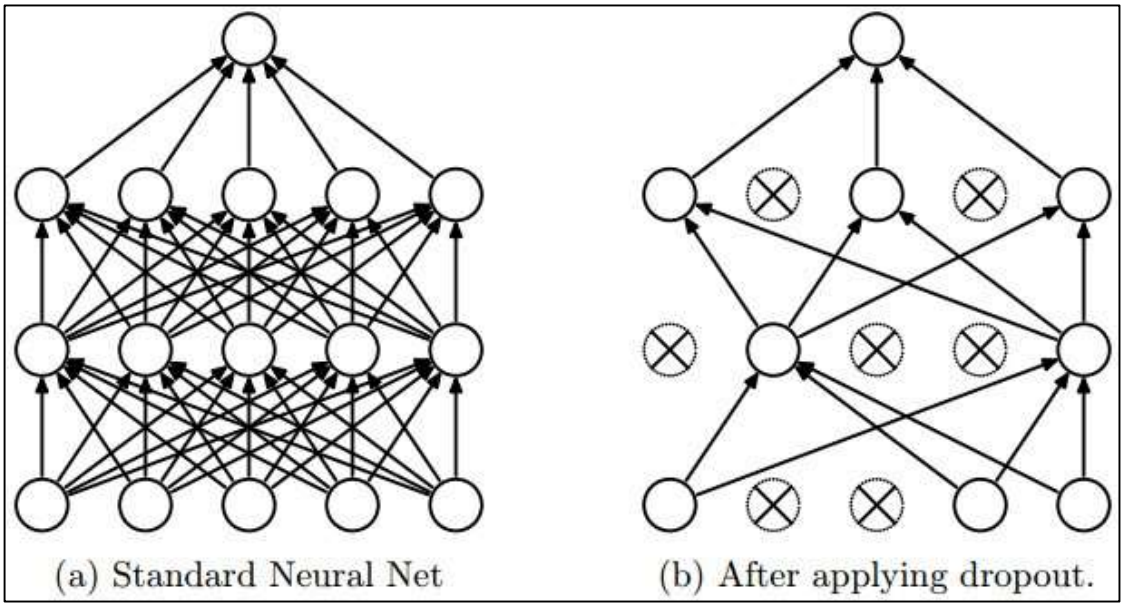

Рисунок 3.14 – Схематична візуалізація методу dropout

Цікавий момент такого методу полягає у тому, що в момент виключення певних нейронів загальна сума зважених сигналів на вході кожного нейрону буде відрізнятися від аналогічної суми у випадку, коли всі нейрони працювали б. Отже, вихідне значення нейрону після його активації теж буде іншим. У такому випадку модель може почати видавати некоректні результати. Запобігти цьому можна за допомогою модифікації зваженого вхідного сигналу нейрона. Її суть полягає у тому, що знаходиться величина, обернена до математичного очікування кількості невиключених нейронів і ця величина є нічим іншим як очікуваною кількістю працюючих нейронів. Вхідний сигнал на нейрон модифікованої нейронної мережі ділиться на цю величину, що створює ефект компенсації нормальної роботи попереднього шару.

Після того, як мережа завершила етап навчання, включаються всі нейрони і ефект перенавчання (зайвої спеціалізації) має помітно знизитись. Виключені нейрони не роблять свій внесок у процес навчання на жодному з етапів алгоритму зворотного поширення помилки (backpropagation). Тому

сумарний сигнал на входах нейронів масштабується, емулюючи поведінку повної мережі з усіма нейронами для того, щоб не допустити спотворення результатів роботи всієї мережі.

3.6.2 Рекурентні нейронні мережі

Рекурентними називають нейронні мережі, у яких вихід з нейронів одного шару необов’язково подається на вхід наступного шару, тобто сигнал може передаватися на більш ранні шари і утворювати так званий зворотній зв’язок. При цьому під зворотним зв'язком мається на увазі зв'язок від більш віддаленого елемента до менш віддаленого. Наявність зворотних зв'язків дозволяє враховувати досвід попереднього або декількох попередніх проходжень сигналів, а отже створити певний потоковий контекст обробки вхідної інформації. Це робить рекурентні нейронні мережі зручними для обробки послідовностей.

Приватним випадком обробки послідовностей є обробка природної мови, оскільки текст є, власне, послідовністю слів. Варто зазначити, що рекурентні нейронні мережі не так сильно відрізняються від звичайних. Їх можна уявити як безліч копій однієї і тієї ж мережі, причому, кожна копія передає повідомлення наступної копії. Схематичне зображення поетапної роботи рекурентної мережі наведено на рисунку 3.15.

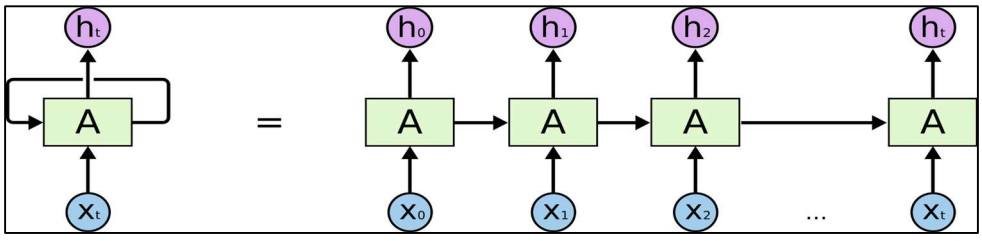

Рисунок 3.15 – Візуалізації поетапної роботи рекурентної мережі

Найбільш популярними архітектурами рекурентних нейронних мереж на сьогодні є LSTM та GRU модулі.

LSTM (Long Short Term Memory) – різновид рекурентних мереж, здатний регульовано накопичувати інформацію про попередні сигнали та використовувати її для обробки наступних. Структура, наведена на рисунку 3.16, має форму ланцюга повторюваних модулів (repeating module).У звичайній рекурентній нейронній мережі ці модулі, матимуть дуже просту структуру, наприклад, всього один шар з функцією активації гіперболічний тангенс.

Основним елементом LSTM модулю є клітинний стан. На зображенні вище він позначений горизонтальною лінією, що проходить крізь верхню частину модулю і існує фактично як конвеєр для проходу вхідної інформації через модуль з певними змінами або без них. LSTM модуль може видаляти або додавати інформацію до клітинного стану, але ця можливість ініціюється лише структурами, які називаються вентилями (gates) [8]. Вентилі є комбінацією сигмоїдного шару нейронів (на рисунку 3.16 позначені жовтими прямокутниками з літерою σ) та операції звичайного векторного множення.

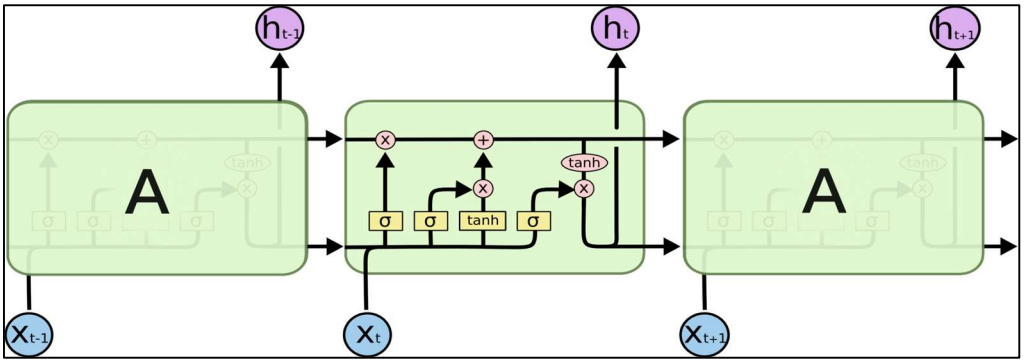

Рисунок 3.16 – Схематичне структури LSTM модулів у рекурентній нейронній мережі

Сигмоїдний шар подає на вихід числа між нулем та одиницею, описуючи таким чином, наскільки кожен компонент має бути пропущений крізь вентиль. Нуль фактично означає нічого не пропускати ,а одиниця -

беззмінний прохід інформації. LSTM має три такі вентилі, щоб захищати та контролювати клітинний стан.

Першим кроком роботи модуля є робота так званого забуваючого вентиля (forget gate layer). Він конкатенує вектори вихідної інформації з попереднього етапу та вхідної інформації нового екземпляру. Результуючий вектор проходить крізь сигмоїдний шар вентилю і трансформується у значення від 0 до 1. Одиниця під час подальшого векторного множення на конвеєр буде означати беззмінний прохід інформації по ньому, у той час як нуль активує «забування» того, що ми отримали з попереднього етапу. Отже, цей крок виконує функцію стирання контексту, який був накопичений для розгляду. Схематичний вигляд цього кроку зображено на рисунку 3.17.

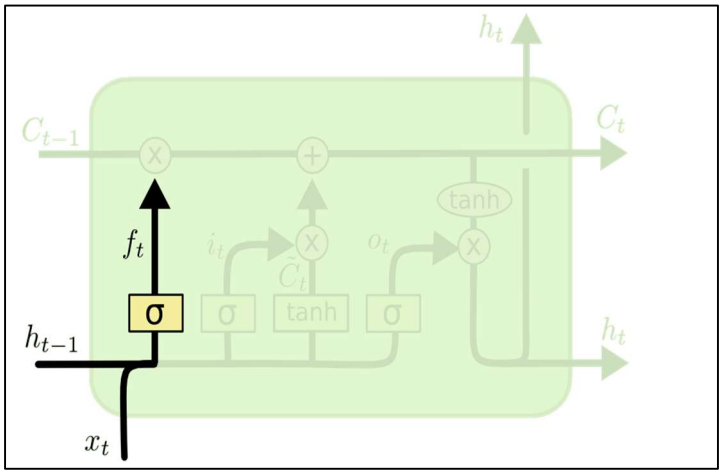

Рисунок 3.17 – Робота забуваючого вентилю у LSTM модулі

Наступний крок визначає інформацію, яку модуль буде зберігати у клітинному стані. Схематичну роботу цього кроку розглянуто на рисунку 3.18.

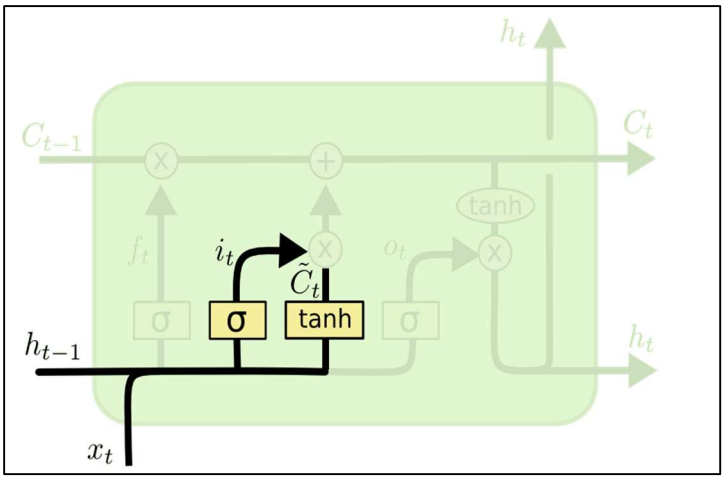

Рисунок 3.18 – Робота вхідного вентилю у LSTM модулі

Вже відомий з попереднього кроку сконкатенований вектор проходить через сигмоїдний шар вхідного вентилю (input gate layer). Аналогічно до попереднього кроку, одиниця є командою повного збереження, а нуль – тотальна відсутність впливу нової інформації на клітинний стан. Далі, шар гіперболічного тангенсу створює вектор кандидатів на нові значення, який може бути доданий до стану.

На наступному етапі ми з'єднаємо ці дві частини, щоб оновити клітинний стан. Схему цього етапу наведено на рисунку 3.19.

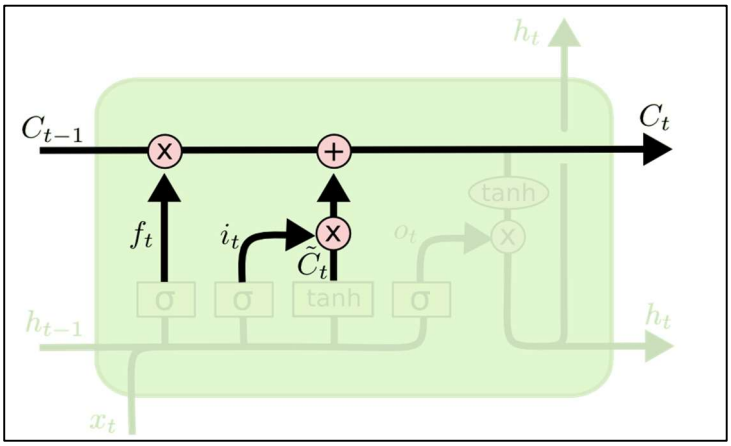

Рисунок 3.19 – Оновлення клітинного стану у LSTM модулі

Після логічного множення результатів сигмоїди та гіперболічного тангенсу ми додаємо отримане до конвейера. Доданками є нові кандидати, масштабовані відповідно до того, наскільки сильно мережа вирішила оновити кожне значення стану.

Зрештою, нам потрібно вирішити, який результат ми збираємось подати на вихід. Цей результат буде заснований на нашому клітинному стані, але фактично буде його певним чином відфільтрованою версією. Схему обчислення вихідного сигналу наведено на рисунку 3.20.

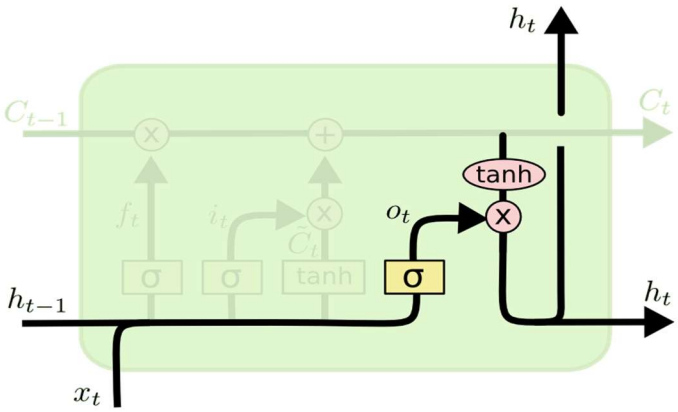

Рисунок 3.20 – Оновлення клітинного стану у LSTM модулі

Спочатку ми запускаємо сигмоїдний шар, який вирішує, які частини клітинного стану збираємося відправити на вихід. Потім ми пропускаємо клітинний стан крізь гіперболічний тангенс і модифіковане значення множимо на вихід сигмоїдного вентиля, отже ми відправляємо на вихід тільки ті частини, які пройшли фільтрацію.

Такі нейронні мережі можуть використовуватися у навчанні з учителем та методом зворотного поширення помилки. Особливість такої архітектури у тому, що нейромережа навчається не тільки видавати правильні сигнали на вихід, але й формувати сигнали вентилей (вхідного, вихідного та вентиля забування) для якісного впливу на результуючий

вихідний сигнал. Недоліком такої архітектури є структурна складність, що означає потребу у більших ресурсах для навчання такої мережі.

Існує спрощений різновид LSTM – вентильна рекурентна одиниця (Gated Recurrent Unit) або GRU. Подібно до LSTM архітектури, він виконує злиття клітинного стану із прихованим шаром та вносить деякі інші зміни. Забуваючий та вхідний вентилі тут поєднано в один так званий «поновлювальний вентиль» (update gate). Модель, що виходить у результаті, є простішою, ніж звичайна модель LSTM і практично не поступається у результативності, особливо на невеликих датасетах. Схему GRU модуля наведено на рисунку 3.21.

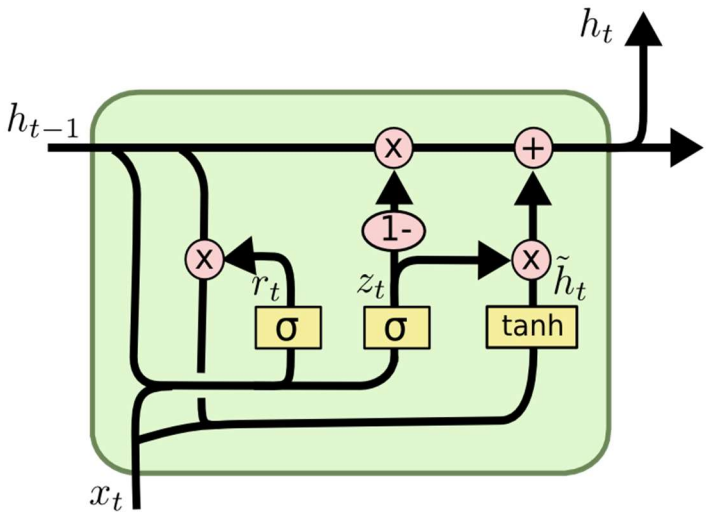

Рисунок 3.21 – Схематичне зображення GRU модулю рекурентної нейронної мережі

Отже, після вичерпного розгляду усіх аспектів дослідів, що будуть проводитися можна переходити до етапу експериментальних досліджень.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

4.1 Загальні умови дослідів

У результаті проведеного етапу теоретичних досліджень було визначено достатню для проведення експериментальних досліджень теоретичну базу етапу.

У якості засобів трансформації тексту вирішено обрати два методи: one-hot-encoding та метод щільного вектору слова. Вони будуть використовуватися по черзі, з метою порівняльного аналізу впливу на результати навчання.

Набір даних Amazon Reviews for Sentiment Analysis вирішено використовувати як основний, проте не його основну частину, а тестову вибірку. Дане рішення можна пояснити браком апаратних ресурсів для об’ємів оригінальної основної вибірки. Вибірка під час навчання має весь час знаходитися у оперативній пам’яті, тому доступних 12.69 GB оперативної пам’яті може бути недостатньо для роботи з повним набором даних, що налічує 3600000 екземплярів. У той час як 400000 екземплярів тестової вибірки, використані у якості основної, теоретично можуть забезпечити достатню якість та оптимальну швидкість процесу тренування. Фрагмент вибірки наведено у додатку Б.

Додатково вирішено проводити другий етап тестування – оцінку якості нейронної мережі на тестовій вибірці з набору даних YELP dataset. Вона містить 38000 екземплярів та є вибіркою відгуків до ресторанів, автосервісних та домашніх послуг та інших товарів. Відгуки розмічені відповідно до їх емоційної полярності: негативні або позитивні. Цей датасет є зручним для додаткової перевірки функціоналу нейронної мережі на даних, які мають інше від тренувальних або тестових джерело виникнення. Фактично таким набором даних можна емулювати тестування у реальних умовах. Фрагмент вибірки наведено на рисунку 4.1.

```
extra_test
```

	Class	Review
0	2	Contrary to other reviews, I have zero complai...
1	1	Last summer I had an appointment to get new ti...
2	2	Friendly staff, same starbucks fair you get an...
3	1	The food is good. Unfortunately the service is...
4	2	Even when we didn't have a car Filene's Baseme...
...	...	...
37995	1	If I could give 0...I would. Don't do it.
37996	2	Items Selected:\nChocolate Cinnamon Horn\nSmal...
37997	1	Expensive lunch meals. Fried pickles were goo...
37998	1	Highly overpriced and food was cold. Our waitr...
37999	1	I have been using this company for 11 months. ...

38000 rows × 2 columns

Рисунок 4.1 – Фрагмент вибірки YELP dataset

Етапи тренування та тестування будуть проводитися у середовищі Google Colaboratory. Інтерактивний редактор коду у ньому називається ноутбуком. Розглянемо структуру типового ноутбука, що буде використовуватися у рамках цієї роботи.

На прикладі секції імпортування необхідних бібліотек, яку наведено на рисунку 4.2, оглянемо їх використання у рамках даної роботи:

− Keras для побудови структури нейронної мережі, її конфігурації. Також будуть використовуватися окремі елементи пакету для токенізації тексту та формування послідовностей токенів;

− Pandas для завантаження та взаємодії з вибірками;

```
%tensorflow_version 2.x
from tensorflow.keras.models import Sequential,load_model
from tensorflow.keras.layers import Dense, Embedding, GRU, LSTM,Dropout,Flatten
from tensorflow.keras import utils
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.callbacks import ModelCheckpoint
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import re
import sklearn
from sklearn.model_selection import train_test_split
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
from nltk.stem import PorterStemmer
%matplotlib inline
```

Рисунок 4.2 – Секція імпортування необхідних для дослідів бібліотек

– NumPy для формування one-hot-encoding подання тексту;

– Matplotlib для візуалізації результатів навчання і оцінки точності роботи на валідаційній вибірці та тренувальній. Ця інформація допоможе виявити перенавчання у випадку його виникнення;

– NLTK для стемінгу та роботи з вилучення стоп-слів з датасету.

Для подальшої роботи потрібно заздалегідь визначити 3 питання конфігурації досліду:

– довжину вхідного тексту. Параметр визначає обробку якої кількості слів за одне повідомлення підтримуватиме система. Це є важливим у випадку використання методу щільного вектору. Повідомлення більші за розміром будуть скорочуватися до вказаної величини, а менші доповнюватися до неї пустими символами. Виходячи з середньої довжини тексту у датасеті 178 слів, було вирішено обрати довжину 200 слів;

– кількість унікальних слів з вибірки, які беруться до аналізу і формують словник для кодування тексту. Цей параметр визначає потужність множини слів, які буде здатна розпізнати мережа. Було обране значення 10000, яке є цілком достатнім, оскільки чисельні дослідження

експертів з лінгвістики показали, що середній словниковий запас однієї людини у просторі однієї мови оцінюється від 2500 до 4000 слів;

– флаг обрання методу one-hot-encoding. У випадку якщо він негативний – за замовчуванням використовуватиметься метод щільного вектору.

Наступним етапом є завантаження та передобробка даних, а саме двох вибірок: Amazon Reviews for Sentiment Analysis та YELP. Після завантаження проходить етап передобробки вибірок, код якого наведено на рисунку 4.3.

```
texts=preprocess(texts)
sws= dict(zip( stopwords.words('english'),range(len(stopwords.words()))))
ps = nltk.stem.PorterStemmer()
def token_preprocess(text):
  tokens=re.split("\s",text)
  tokens = [ps.stem(token.lower()) for token in tokens if len(token.strip())>0]
  # tokens = [token.lower() for token in tokens if len(token.strip())>1 and token.lower().strip() not in sws.keys()]
  return ' '.join(tokens)
def preprocess(data):
  print('here')
  return data             \
  .replace("/d+"," ",regex=True)      \
  .replace("[^A-Za-z\s]"," ",regex=True)\
  .transform(lambda x: token_preprocess(x))\
```

Рисунок 4.3 – Код попередньої обробки даних

У попередній обробці вирішено виконувати наступні зміни у тексті:

– вилучення з тексту цифр та їх заміни на пробіл;

– вилучення будь-яких символів, що не є символами латинки. Вилучаються знаки пунктуації, службові послідовності, спецсимволи, символи з абеток інших алфавітів, пробільні символи за виключенням одиночних пробілів, що позначають межі слів у тексті;

– розбиття тексту на слова за пробільними символами;

– видалення зайвих пробільних символів;

– перевірка на те, чи є це слово у списку стоп-слів і потенційне вилучення з розгляду. Виконується одночасно з вилученням дуже коротких малоінформативних слів. У рамках цієї роботи етап вилучення стоп-слів

вирішено зробити опцією етапу попередньої обробки, оскільки він немає вирішального значення і є лише допоміжним у рамках задачі і засобів її вирішення;

– стемінг – засіб нормалізації словоформ, який здійснюється переважно шляхом відкидання допоміжних частин, таких як закінчення, суфікс чи префікс, тобто фактично словоформа зводиться до кореня слова. Такий метод нівелює вплив на зміст вхідного тексту роду слова, відмінку або форми. Використання цього етапу у рамках даної роботи теж буде опціональним.

Різницю у вигляді вибірки до та після передобробки зображено на рисунках 4.4-4.5.

```
0          Great CD: My lovely Pat has one of the GREAT ...
1          One of the best game music soundtracks - for ...
2          Batteries died within a year ...: I bought th...
3          works fine, but Maha Energy is better: Check ...
4          Great for the non-audiophile: Reviewed quite ...
                                 ...
399995     Unbelievable- In a Bad Way: We bought this Th...
399996     Almost Great, Until it Broke...: My son recie...
399997     Disappointed !!!: I bought this toy for my so...
399998     Classic Jessica Mitford: This is a compilatio...
399999     Comedy Scene, and Not Heard: This DVD will be...
Name: Text, Length: 400000, dtype: object
```

Рисунок 4.4 – Вигляд тренувальної вибірки до попередньої обробки

Після етапу передобробки виконується розбиття вибірки на тестову і тренувальну у співвідношенні 75:25, що є рекомендованим для таких робіт значенням. Таким чином тренування нейронної мережі буде здійснено на 300000 екземплярів, а тестування на 100000. У кожному окремому випадку розбиття буде випадковим.

```
0         great cd lovely pat one great voices generatio...
1         one best game music soundtracks game really pl...
2         batteries died within year bought charger jul ...
3         works fine maha energy better check maha energ...
4         great non audiophile reviewed quite bit combo ...
                                ...
399995    unbelievable bad way bought thomas son huge th...
399996    almost great broke son recieved birthday gift ...
399997    disappointed bought toy son loves thomas toys ...
399998    classic jessica mitford compilation wide range...
399999    comedy scene heard dvd disappointment get hopi...
Name: Text, Length: 400000, dtype: object
```

Рисунок 4.5 – Вигляд тренувальної вибірки до попередньої обробки

Наступним етапом є трансформація текстів у цифровий вигляд. Для цього потрібно виконати розбиття текстів на окремі слова та обчислення словнику найбільш вживаних слів. Таку обробку можна зробити за допомогою класу Tokenizer, який приймає зазначений вище параметр максимальної кількості слів, що беруться у розгляд. Окрім присвоєння цілочисельних кодів словам, цей клас резервує кодами наступні спецсимволи:

- символ пробілу, заповнення або порожнього слова;
- символ доповнення текстової послідовності до зазначеного обсягу;
- символ для позначення невідомого словнику слова.

Після формування словнику виконується, власне, процес трансформації: кожне слово замінюється числовим еквівалентом з словника. Наступна обробка виконується в залежності від обраного засобу подання слова:

- one-hot-encoding: створюється матриця вибірки, що відповідає за розмірами кількості екземплярів вибірки з одного виміру та розміру словника з іншої. Відповідно до таких умов, кожний рядок є відображенням певного екземпляру. Матриця спочатку складається з нулів. За допомогою циклу по трансформованим послідовностям слів на пересіченнях

словникових позицій слів з вибірки та екземплярів, що ці слова містять, ставляться одиниці;

− метод щільного вектору: у залежності від довжини трансформованої послідовності слів, функцією pad_sequences вона зводиться до обраної дослідником довжини за допомогою доповнення або вирізання кінцевих символів. Варто зазначити, що доповнення виконується спецсимволами, які у необхідній кількості розміщуються перед основним текстом. Такий підхід є більш оптимальним для використання з рекурентними нейронними мережами, що розглядаються у цій роботі, оскільки беззмістовне наповнення йде перед змістом тексту і тому дозволяє мінімально вплинути на контекст та стан, який здатен змінити модуль мережі під час обробки цього повідомлення.

Порівняння вихідних текстів та їх подань у форматах one-hot-encoding та щільного вектору наведено на рисунках 4.6-4.7.

```
index = 25
print(texts[index])
print(x_train[index])
```

```
waste of money like many of the barbie cd roms the playtime is limited averaging about mins for each of my two daughters ages like the barbie movies
[   0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0   34  139    2   23  240  388 2433 9954
 3206    8  220    9 3773    3  272  139    6  342   48   35    4 5795
   70    4  220   91    9    1  196    2 1397   28   13   40  873    6
   18 3123   39 2272]
```

Рисунок 4.6 – Приклад тексту та його трансформації методом щільного вектору

```
index = 20
print(texts[index])
print(x_train[index][:200])
```

```
long boring read book much expectation boring book
[0. 0. 1. 0. 0. 1. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 1.
 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 1. 0. 0. 1. 0. 0. 0.]
```

Рисунок 4.7 – Приклад тексту та частини його трансформації методом one-hot-encoding

Далі потрібно сформувати архітектуру нейронної мережі для досліду. На етапі експериментальних досліджень будуть використовуватися наступні архітектури нейронних мереж:

- класична архітектура нейронних мереж прямого поширення;
- рекурентні мережі з використанням LSTM модулів;
- рекурентні мережі з використанням GRU модулів;
- змішані за типом модуля рекурентні мережі.

Конфігурація також включає наступні параметри моделі:

- оптимізатор: алгоритм оновлення моделі, який враховує тренувальні дані та значення функції втрат. За замовчуванням обрано оптимізатор ADAM (adaptive moment estimation), оскільки він є найбільш загальноприйнятим і перевіреним рішенням у розв’язанні подібних задач;
- функція втрат: за замовчуванням обрано бінарну перехресну ентропію, оскільки вона є найбільш оптимальним варіантом для задач бінарної класифікації, приватним випадком якої є аналіз сентиментальної тональності;

− метрика оцінки якості: за замовчуванням обрано точність роботи на вибірці;

− кількість епох тренування: для порівняння обрано 10 епох;

− розмір пакетної нормалізації: обрано 128 як рекомендоване початкове значення;

− відсоток екземплярів, що обираються для валідаційної вибірки: за замовчуванням обрано 20%.

Для отримання найкращого результату навчання вирішено використовувати програмну сутність ModelCheckpoint з бібліотеки Keras, яка вбудовується у конфігурацію нейронної мережі та дозволяє перехопити та зберегти останню найбільш якісну епоху за вказаною ознакою. У якості критерія відбору буде використовуватися значення точності роботи нейронної мережі на валідаційній вибірці.

Таким чином, маючи як фінальну модель так і модель збережену за ознакою максимальної точності на валідаційній вибірці ми маємо можливість оцінювати результат досліду за 4 основними показниками:

− точність роботи фінальної моделі нейронної мережі на оригінальній тестовій вибірці;

− точність роботи оптимальної моделі нейронної мережі на оригінальній тестовій вибірці;

− точність роботи фінальної моделі нейронної мережі на додатковій тестовій вибірці;

− точність роботи оптимальної моделі нейронної мережі на оригінальній додатковій вибірці.

Код реалізації типового ноутбуку, що використовувався для проведення дослідів у даній роботі наведено у Додатку А. З деякими змінами, за його допомогою можна виконати будь-який з запланованих у роботі дослідів.

4.2 Нейронні мережі прямого поширення

Для початку визначимо відносно просту конфігурацію нейронної мережі і отримаємо результати роботи на ній, а потім за допомогою зміни певних параметрів вивчимо їх вплив та найбільш оптимальні конфігурації.

Конфігурація за замовчуванням включатиме:

- кількість слів у словнику вибірки – 10000;
- довжина тексту – 200;
- використання методу щільного вектора як засобу подання слова;
- вилучення стоп слів та слів коротше за 2 символи;
- невикористання стемінгу;
- для методу щільного вектора розмірність вектора одного слова становитиме 2;
- мережа буде складатися з одного прихованого повнозв’язного шару з 8 нейронами;
- функція активації прихованого шару – сигмоїдна;
- вихідний шар буде складатися з одного нейрону з сигмоїдною функцією активації, вихід якої буде означати оцінку тональності тексту, де 1 – абсолютно позитивна оцінка, а 0 позначає негативний текст.

Навчання однієї епохи такої моделі тривало близько 11 секунд та завершилося з результатом 94% точності на тренувальній вибірці. Найбільш високе значення точності на валідаційній вибірці було зафіксовано на 4 епосі – 87.9% і відтоді не мало тенденцій до покращення чим спровокувало несильне перенавчання. Тестування фінальної епохи показало 86% точності на оригінальній тестовій вибірці. Тестування кращої епохи показало точність у 87.5%.

Відмова від вилучення стоп слів дещо пришвидшила передобробку даних, але негативно відмітилася на якості роботи нейронної мережі, що зменшилася на 3 відсотки.

Збільшення розмірності щільного вектора до 32 підвищило якість результатів навчання, але помітно додало перенавчання та незначно підвищило час тренування однієї епохи. Графік точності роботи на тренувальному та валідаційному наборах зображено на рисунку 4.8.

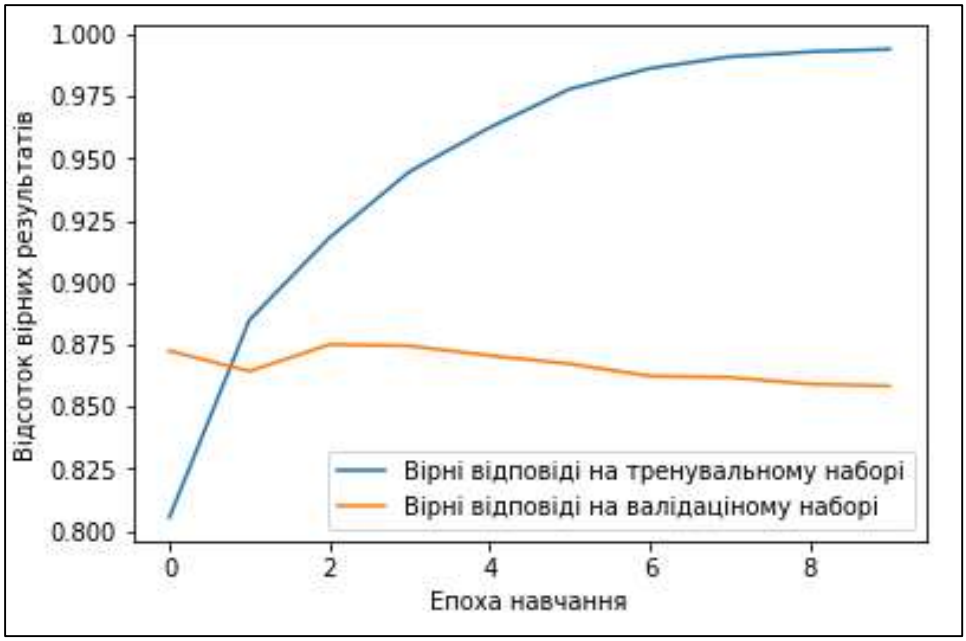

Рисунок 4.8 – Точність роботи на тренувальному та валідаційному наборах при використанні 32-мірного вектору

Отримані результати свідчать, що така розмірність вектору слова для даних умов є надмірною і потребує зменшення. Шляхом проведення додаткових дослідів було встановлено оптимальну розмірність від 4 до 12 (за умови використання dropout).

Невелике зниження якості роботи спостерігається при використанні методу подання слова one-hot-encoding. Це можна пояснити його неспроможністю передавати якісні характеристики слова, а лише відносну присутність та примусовим зниженням об’ємів вибірки тренування, спричиненим нестачею оперативної пам’яті для повних об’ємів вихідного датасету.

Використання стемінгу виявило здебільшого негативний його вплив при використанні основної вибірки. З одного боку спостерігається незначне покращення якості роботи нейронної мережі, але з іншого – серйозне погіршення часу передобробки вибірки. Серед усіх розглянутих засобів стемінгу з популярних для аналізу даних бібліотек найшвидший результат було отримано близько 10 хвилин. З огляду на незначне збільшення точності, та значну затримку продуктивності передобробки було вирішено відмовитися від стемінгу у подальших дослідах.

На початковій конфігурації цілком очікувано зміна функції активації єдиного прихованого шару не вплинула на точність роботи моделі і не змінила характеристику продуктивності навчання і роботи моделі. Це можна пояснити замалою структурою нейронної мережі для того щоб різна поведінка активаційних функцій стосовно змін градієнту встигла проявитися.

Поступове збільшення числа нейронів у прихованому шарі до 32 не відмітилось помітним збільшенням точності роботи. Варто зазначити, що конфігурація із 32 нейронами показала ознаки перенавчання, які зникли після додавання dropout, проте точність залишилася на попередньому рівні.

Наступна серія дослідів ставила на меті розширити структуру нейронної мережі до кількох прихованих шарів. Найбільш оптимальним виявилося рішення обмежитися двома прихованими шарами. Перший прихований шар складається з 64 нейронів, а другий з 32. Для зниження можливого перенавчання перший шар було вирішено доповнити використанням методу dropout з імовірнісним коефіцієнтом 60%. На практиці, у дослідах з більшою кількістю шарів або більшою кількістю нейронів було дуже важко контролювати проблему перенавчання. Також використання більш складної архітектури потребувало збільшити розмірність щільного вектору до 8. Перші досліди на такій конфігурації передбачали сигмоїдну функцію на всіх прихованих шарах. Точність роботи на найкращій моделі становить 85-86%. Дещо гірші результати

спостерігаються у випадку використання на обох шарах функції гіперболічного тангенсу, але точність повертається на попередній рівень, якщо шару з 64 нейронами повернути сигмоїдну функцію активації. Найкраща ж точність, а саме 90-91% була отримана при використанні на другому прихованому шарі функції активації ReLU. Код моделі наведено на рисунку 4.9.

```
model = Sequential()
model.add(Embedding(num_words, 8, input_length=max_review_len))
model.add(Flatten())
model.add(Dense(64,activation='sigmoid'))
model.add(Dropout(0.6))
model.add(Dense(32,activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

Рисунок 4.9 – Код двошарової моделі нейронної мережі прямого поширення

Використання шару Flatten дозволяє сконкатенувати усі векторні представлення слів у один вектор. Оскільки Embedding шар віддає масив щільних векторів, звичайні лінійні шари не можуть з ними працювати. Тому використання конкатенації є необхідним у такій конфігурації, хоча не є прихованим шаром у класичному розумінні.

Навчання такої однієї епохи такої моделі тривало трохи довше ніж одношаровий варіант і не є помітним недоліком, враховуючи більшу точність на тестових вибірках.

Отже, було знайдено дві найбільш оптимальні архітектури повнозв’язних нейронних мереж прямого поширення. Перша модель може бути описана наступними параметрами:

- один прихований шар з 8 нейронами;
- подання слова щільним вектором з довжиною вектора 4;

- вилучення стоп-слів;
- функція активації прихованого шару – сигмоїдна.

Друга модель може бути описана наступним чином:

- прихований шар з використанням 64 нейронів та функцією активації сигмоїда;
- dropout з імовірнісним коефіцієнтом 0.6;
- прихований шар з використанням 32 нейронів та функцією активації ReLU;
- подання слова щільним вектором з довжиною вектора 4.

4.3 Рекурентні нейронні мережі

Для початку визначимо конфігурацію нейронної мережі і за замовчуванням, а потім за допомогою зміни певних параметрів вивчимо їх вплив та найбільш оптимальні конфігурації.

Конфігурація за замовчуванням включатиме:

- кількість слів у словнику вибірки – 10000;
- довжина тексту – 200;
- використання методу щільного вектора як засобу подання слова;
- вилучення стоп слів та слів коротше за 2 символи;
- невикористання стемінгу;
- для методу щільного вектора розмірність вектора одного слова становитиме 4;
- мережа буде складатися з одного прихованого повнозв’язного шару з 4 LSTM модулями;
- вихідний шар буде складатися з одного нейрону з сигмоїдною функцією активації, вихід якої буде означати оцінку тональності тексту, де 1 – абсолютно позитивна оцінка, а 0 позначає негативний текст.

Навчання однієї епохи такої моделі тривало близько 90 секунд та завершилося з результатом 92% точності на тренувальній вибірці. Найбільш

високе значення точності на валідаційній вибірці було зафіксовано на 7 епосі – 88.9% і відтоді не мало тенденцій до покращення. Оцінка точності на тестовій вибірці показало 88.3%.

Виходячи з факту про більш складну структуру LSTM модулю, було сформовано припущення, що така модель потребує більшої розмірності щільного вектору на вхідному шарі. Зниження цієї розмірності до 2 трохи знизило ефективність роботи до 87.2% на тестовій вибірці. Послідовні досліди з цим значенням виявили правильність припущення: найбільш оптимальне значення знаходиться у проміжку від 8 до 16. Починаючи з 16 помітне зростання якості роботи не має місце, у той час як цілком очікувано зростає вплив перенавчання. Для розмірностей вище за 16 вже помітно важче контролювати вплив перенавчання. Далі буде використовуватися розмірність 8, оскільки отримана точність на тестовій вибірці 89.8% є найбільшою серед проведених дослідів. Час навчання однієї епохи зріс приблизно на 10 секунд.

Через особливості архітектури LSTM та GRU модулів використання подання one-hot-encoding є конструктивно неможливим, тому такі досліди не проводилися.

Помітний якісний стрибок було зафіксовано, при відмові від вилучення стоп слів – точність роботи на тестовій вибірці склала близько 91.4%. Такий ефект можна пояснити тим, що стоп-слова все ж беруть участь у формуванні певного контексту, незважаючи на власну окрему беззмістовність. Можливість аналізувати глибший контекст дозволила LSTM модулям отримати кращу узагальнюючу здатність. Виходячи з таких результатів опція використання стемінгу є недоцільною.

Експерименти зі збільшенням кількості нейронів у прихованому шарі показали, що збільшення числа нейронів до 16 дає невеликий приріст у якості роботи – близько 0.7%.

Заміна LSTM модулів на GRU дала приріст у швидкості навчання – одна епоха обробляється у середньому на 20 секунд швидше. Також

спостерігається незначний приріст у якості роботи. Це можна пояснити спрощеністю будови GRU, у порівнянні з LSTM, що дає більшу стійкість до перенавчання. Приріст швидкості пояснюється кращою обчислювальною ефективністю.

Пошук багатошарової архітектури встановив, що для заданих умов навчання не існує необхідності мати більше ніж один прихований шар рекурентної нейронної мережі. Точність на тестовій вибірці архітектури, що наведена на рисунку 4.10 склала 92%.Варто зазначити, що час навчання однієї епохи за такої конфігурації втричі більший за мережу з одним прихованим шаром – близько 3 хвилин. Такі витрати ресурсів, з огляду на незмінну точність, є нераціональними. Можливо, на вибірці більших розмірі або іншої її структури така архітектура показала б помітне покращення якості, у порівнянні з більш простою архітектурою. У такому випадку використання двох прихованих шарів було б доцільним.

```
model = Sequential()
model.add(Embedding(num_words, 4, input_length=max_review_len))
model.add(LSTM(16,return_sequences=True))
model.add(LSTM(8))
model.add(Dense(1, activation='sigmoid'))
```

Рисунок 4.10 – Код двошарової моделі рекурентної нейронної мережі з використанням LSTM модулів

Отже, після серії експериментів було знайдено найбільш оптимальні архітектури рекурентних нейронних мереж. Опис першої наведено нижче:

- один прихований шар з 16 нейронами;
- подання слова щільним вектором з довжиною вектора 8;
- відмова від вилучення стоп-слів;
- архітектура модулів GRU.

Друга модель може бути описана наступним чином:

- один прихований шар з 8 нейронами;
- подання слова щільним вектором з довжиною вектора 4;
- відмова від вилучення стоп-слів;
- архітектура модулів LSTM.

4.4 Проблеми етапу експериментальних досліджень

На етапі дослідів та підготовки до них виник ряд проблем, пов’язаних з продуктивністю системи.

Перша проблема, що виникла на етапі розробки пов’язана з вилученням стоп-слів з датасету. Було помічено, що передобробка з використанням цього етапу займає близько двох хвилин. Під час аналізу продуктивності було встановлено, що левова частка витраченого часу витрачається саме на пошукові операції у списку стоп-слів. Використовуваний список стоп-слів англійської мови налічував близько 200 слів, що пояснює такі витрати процесорного часу у випадку пошуку простим перебором списку. Проблему було усунено обранням для зберігання стоп-слів іншої структури даних – вбудованого у Python словника.

Така структура даних є реалізацією геш-таблиці. Така таблиця зберігає значення у комірках масиву. Позиція значення у масиві визначається так званим ключем, який за допомогою геш-функції перетворюється на індекс комірки. На відміну від перебору значень, операції обчислення індексу та взяття значення є константними за алгоритмічною складністю, що пояснює більш швидку роботу. З використанням такої структури даних для пошуку слова серед стоп-слів час передобробки основної вибірки зменшився до 30 секунд. Схематичне зображення внутрішньої роботи геш-таблиці зображено на рисунку 4.11.

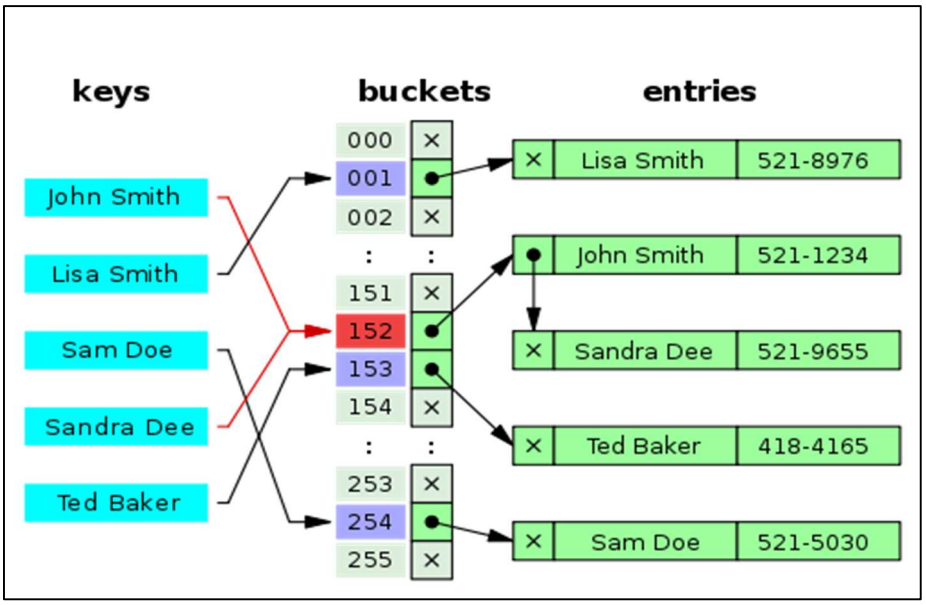

Рисунок 4.11 – Схематичне зображення внутрішньої роботи геш-таблиці

Інша проблема була пов’язана з дослідами, де використовувався метод one-hot-encoding. Під час їх проведення, а саме на етапі трансформації тексту у подання, вони передчасно закінчувалися системною помилкою середовища outOfMemoryError. Ця помилка зазвичай свідчить про відсутність вільної оперативної пам’яті, у той час коли було запрошено її виділення. Факт виникнення такої проблеми було вирішено підтвердити розрахунками. Оскільки на розмірність матриці впливають кількість екземплярів вибірки та розмір словнику, то методом підбору були визначені такі значення цих параметрі, на яких виділення пам’яті під one-hot-encoding матрицю виконувалось б успішно. Виявилось, що вибірка, обмежена першими 37500 екземплярами на словнику з 5000 слів, потребувала для виділення матриці близько 1.4 GB оперативної пам'яті. За методом пропорції було визначено, що повний об’єм вибірки 300000 екземплярів на словнику з 10000 слів мав би зайняти близько 22 GB пам’яті, у той час коли середовище має лише 12.69 GB. Очевидно, що питання зменшення цих параметрів є безальтернативним рішенням при використанні цього методу

подання тексту. Було обчислено і на практиці підтверджено, що максимальні доступні значення цих параметрів можуть бути 75000 екземплярів вибірки на словнику з 5000 екземплярів. Ці значення і були використані на експериментальному етапі.

4.5 Підсумки експериментального етапу

Було розглянуто 2 види архітектур нейронних мереж для вирішення задачі аналізу сентиментальної тональності природної мови. Серед серії дослідів з різними параметрами, та оцінюванням їх впливу на результат було обрано 4 найбільш оптимальні конфігурації:

- повнозв'язна мережа прямого поширення, що включає один прихований шар з 8 нейронів сигмоїдної функції активації. Точність роботи 90.6%;
- повнозв'язна мережа прямого поширення, що включає два прихованих шари: перший має 64 нейрони, сигмоїдну функції активації та використання dropout з імовірнісним коефіцієнтом для запобігання перенавчанню. Другий шар має 32 нейрони та функцію активації ReLU. Точність роботи – 91.2%;
- рекурентна мережа з одним прихованим шаром з 16 GRU модулями. Точність роботи – 92.8%;
- рекурентна мережа з одним прихованим шаром з 8 LSTM модулями – 92.4%;

У всіх випадках використовується метод щільного вектору для подання слова.

Перейдемо до оцінювання результатів роботи. Імітувати роботу на реальних даних будемо на тестовій вибірці YELP, яку було описано у попередніх розділах. Результати перевірки виявилися наступними:

- одношарова конфігурація з 8 нейронами: 88.5%;
- двошарова конфігурація з 64 та 12 нейронами: 89.4%;

- рекурентні GRU модулі: 91.5%;
- рекурентні LSTM модулі: 91.6%.

Дані показники свідчать, що етап експериментальних досліджень можна вважати успішним. Було натреновано моделі, які отримали узагальнюючу здатність та придатні до використання у реальних умовах. Результати є логічними і цілком обґрунтованими. Відносно близькі цифри результатів свідчать про максимальний рівень розкриття тренувального потенціалу вибірки. Маючи помітно потужніші апаратні ресурси, що здатні працювати з більшими об’ємами вибірок, можна отримати більш істотні розриви результатів конфігурацій та в цілому більш високі показники якості.

З проведених дослідів та їх результатів можна вивести певну порівняльну характеристику традиційних нейронних мереж та рекурентних. Останні у більшості дослідів переважали за якісними показниками. Як вже було зазначено, це можна пояснити архітектурними особливостями рекурентного підходу у поєднанні з особливостями аналізу природної мови, а саме: можливістю аналізувати слова, спираючись на їм передуючі, тобто враховуючи певний контекст. Але існує ряд інших відмінностей GRU та LSTM модулів від архітектур з прямим поширенням, які вдалося встановити суто з практичної сторони:

- GRU та LSTM модулі піддавались впливу перенавчання, але цей вплив був суттєво менший ніж той, з яким доводилося боротися при тренуванні звичайних мереж прямого поширення;
- модулі виявили більшу здатність до навчання: якщо на більшості конфігурацій мереж прямого поширення точність робот на валідаційний вибірці переставала зростати на 3-4 епосі, то у рекурентних конфігураціях нерідко зростання спостерігалося на 7-8 епохах з 10;
- час навчання однієї епохи модулів у 7-9 разів перевищував навчання звичайних мереж, проте враховуючи попередні дві особливості та

тенденцію до отримання помітно кращих результатів навчання, збільшені витрати часу слід вважати доцільними.

Програмні рішення засновані на отриманих моделях мають багато варіантів застосування:

− автомодерація коментарів або постів: фільтрація токсичного контенту у соцмережах або на онлайн-заходах;

− допомога у оцінюванні стану людини у когнітивних системах: виявлення агресії за кермом, аналіз роботи call-центрів;

− автоматична оцінка відгуків на великих маркетплейсах: перевірка на відповідність оцінки відгука його вмісту, автоматичне обчислення оцінки на відгуках, опублікованих без неї;

− вивчення суспільної думки у соцмережах стосовно певної теми: керування репутацією брендів, дослідження та прогнозування ринку, політики, попередження наслідків негативної реакції спільноти.

Для покращення користувальницького досвіду системи, отриманої на базі натренованих моделей потрібно ввести зручну логіку інтерпретацій її результатів. Отримана система, завдяки нейрону з функцією активації сигмоїда на вихідному шарі видає результат роботи у вигляді нормалізованого значення у проміжку від 0 до 1, де нуль означає найбільш негативну тональність, а одиниця – безумовно позитивну. Отже, при значеннях від 0 до 0.5 включно можна повідомляти користувачу про негативну сентиментальну тональність поданого тексту, а від 0.6 і більше – про позитивну. Враховуючи потенційно схожий характер тональності для текстів значення яких буде близьким до класової межі, можна штучно ввести наступні псевдокласи тональності:

− нейтрально-негативний текст : оцінка від 0.4 до 0.5 включно;

− нейтрально-позитивний текст: оцінка від 0.5 до 0.6 включно.

Така логіка інтерпретації допоможе правильніше оцінювати граничні значення оцінки.

Отримані моделі розраховані лише на текстові послідовності довжиною до 200 слів. У випадку необхідності аналізу тексту, що є довшим за цю межу, можна розбити його на відповідні проміжки, які можуть бути проаналізовані моделлю і взяти середнє значення отриманих оцінок або зважену суму відповідно до пропорцій об’ємів проміжків тексту.

ВИСНОВКИ

У результаті виконання роботи було розроблено систему, яка здатна втсановлювати емоційну тональність природної мови. Розробка такого продукту є дуже цінною з боку здобутого досвіду,адже цей процес супроводжувався вирішенням різних особливостей розробки і використання програмного забезпечення. Розробка передбачала повний цикл з аналізу предметної області проектування, розробки та тестування програмного забезпечення.

Першим етапом було важливо зробити поглиблений аналіз предметної області кваліфікаційної роботи та розробити постановку задачі з формулюванням основних вимог. Виконаний аналіз предментної області та сформульвана постановка задачі дозволили перейти до етапу теоретичних досліджень та реалізації програмного продукту.

Етап теоретичних досліджень охлпив лосттаньо велику кількість питань, без визначення яких було б неможливим ставити досліди. Проведено аналіз різних датасетів, доступних досліднику програмних, апаратних ресурсів та середовища. Розгорнуто розглянуто основні методи векторизації природної мови та архітектури нейронних мереж. Архітектури розглядалися як традиційні прямого поширення,так і рекурентні.

На етапі експериментальних досліджень було проведено тренування та тестування близько 3 десятків моделей. Визначено вплив окремих параметрів навчання на його результати та найбільш оптимальні за точністю конфігурації. Етапу, власне, дослдів передував етап визначення загальних умов дослідів

Результатом є оптимальні моделі нейронних мережі, на базі яких можливо побудуати когнітивну систему аналізу сентиментальної тональності природної мови.

Отримана система має велику кількість варіантів розвитку,зокрема:

− використання більших об’ємів тренувальної вибірки для покращення точності та доцільного викростання більш складних архітектур нейронних мереж;

− викорестання засобів корекції лексичних помилок, характерних природній мові на етапі попередньої обробки даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Буданов М. Обработка текстов на естественных языках. URL: https://habr.com/ru/company/vk/blog/358736/ (дата звернення 20.10.2021)

2. Галушка В.В., Фатхі В.А. Формирование обучающей выборки при использовании искусственных нейронных сетей в задачах поиска ошибок баз данных. URL: http://ivdon.ru/magazine/archive/n2y2013/1597 (дата звернення 20.10.2021)

3. Large Movie Review Dataset. URL: http://ai.stanford.edu/~amaas/data/sentiment/ (дата звернення 29.10.2021)

4. Рубцова Ю.В. RuTweetCorp корпус. URL: https://study.mokoron.com/ (дата звернення 30.10.2021)

5. Bittlingmayer A. Amazon Reviews for Sentiment Analysis. URL: https://www.kaggle.com/bittlingmayer/amazonreviews (дата звернення 30.10.2021)

6. Xiu-Shen Wei Must Know Tips for Deep Learning Neural Networks. URL: https://www.kdnuggets.com/2016/03/must-know-tips-deep-learning-part-1.html (дата звернення 30.10.2021)

7. Практики реализации нейронных сетей. URL: https://neerc.ifmo.ru/wiki/index.php?title=Практики_реализации_нейронных_сетей (дата звернення 30.10.2021)

8. Olah C. Understanging LSTM Networks. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (дата звернення 30.10.2021)