МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

Нейронные сети для обработки текста на естественном языке

РЕФЕРАТ

студентки 5 курса 531 группы
направления 10.05.01—Компьютерная безопасность
факультета КНиИТ

Кайдышевой Дарьи Сергеевны

Проверил
доцент _____ И. И. Слеповичев
подпись, дата

Саратов 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Эпоха цифровизации имеет последствие в виде огромного объема текстовых данных, генерируемых ежедневно – от сообщений в мессенджерах и отзывов на товары, до юридических документов и научно-исследовательских статей. Такой исход может представлять как проблему, так и возможность для улучшения и развития существующих технологий в области обработки информации. Понимание принципов работы по структуризации и извлечению смысла из поступаемых данных – это важнейшая задача для многочисленных приложений, сайтов и сервисов (переводчиков, чат-ботов, виртуальных ассистентов и т. д.), которыми пользуется среднестатистический человек на повседневной основе.

В последние годы нейронные сети активно используются для решения задач в области обработки текста на естественном языке. Теперь компьютеры лучше понимают, интерпретируют и генерируют «человеческий» язык с поразительной точностью.

Цель данной работы – исследовать роль нейронных сетей для обработки текста на естественном языке.

Задачи:

- Изучить основные понятия и определения, связанные с нейронными сетями и обработкой естественного языка;
- Изучить теоретические сведение об основных нейронных сетях, используемых для реализации сформулированных выше задач обработки текста: RNN, LSTM, GRU, и CNN;
- Продемонстрировать пример разработки модели анализа настроений на основе датасета отзывов на известные фильмы;
- Сделать выводы о проделанной работе.

1 Теоретические сведения обработки естественного языка нейронными сетями

Данный раздел посвящен взаимосвязи обработки текста на естественном языке и нейронных сетей. Эти понятия нельзя отождествлять, поскольку обработку естественного языка можно рассматривать как цель – понимание и обработка «человеческого» языка, а нейросетевые модели, как инструмент достижения этой цели. Далее будут рассмотрены основные понятия, определения и подходы, раскрывающие суть такого взаимодействия.

1.1 Основные определения и понятия

NLP (natural language processing, обработка естественного языка) – раздел компьютерной науки и искусственного интеллекта, использующий машинное обучение, чтобы позволить компьютерам понимать человеческий язык и общаться на нем [1].

На рисунке 1 схематически изображено понятие NLP, где:

- NLU (natural language understanding, понимание естественного языка) – переход от неструктурированной информации к структурированной;
- NLG (natural language generation, генерация естественного языка) – переход от структурированной информации к неструктурированной [2].

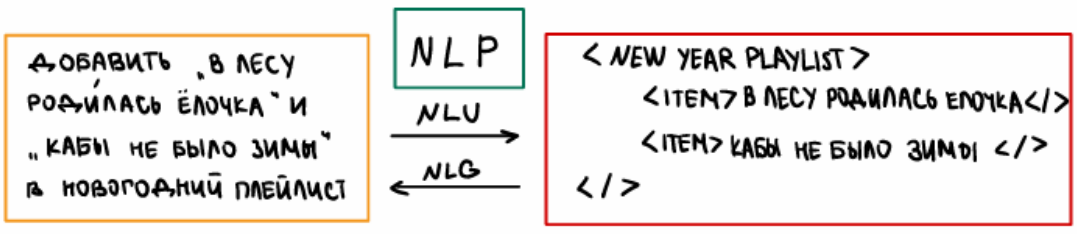

Рисунок 1 – Схема NLP [1]

Формулируются следующие преимущества NLP:

- Автоматизация повторяющихся задач;
- Более качественный анализ данных;
- Улучшенный поиск;
- Генерация контента [1].

Один из инструментов, реализующих NLP – это нейронные сети.

Нейронные сети – основополагающая концепция машинного обучения, вдохновленная структурой и функцией человеческого мозга. Нейронные сети состоят из взаимосвязанных узлов, организованных в слои. Входные слои получают данные, скрытые слои обрабатывают информацию, а выходные слои выдают результаты [3].

Сила нейронных сетей заключается в их способности обучаться на основе данных: внутренние параметры (веса) корректируются во время обучения для оптимизации производительности.

Нейрон – структурная единица, получающая информацию, производящая над ней простые вычисления и передающая ее дальше. Они также делятся на 3 типа: входной, скрытый, выходной [3].

Синапс – связь между нейронами. Параметр синапса – это вес [3].

На рисунке 2 показана нейронная сеть с прямой связью. В фазе прямого распространения происходит начальный этап обучения нейронной сети, при котором входные данные передаются через сеть для генерации прогноза (вычисления происходят на каждом уровне).

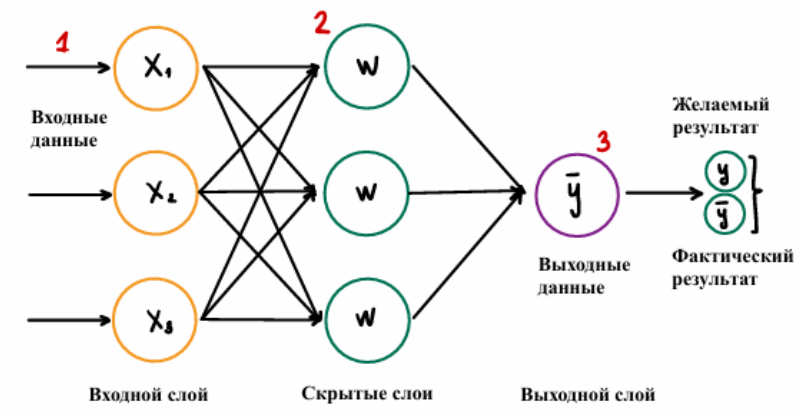

Рисунок 2 – Схема нейронной сети с прямой связью (Feedforward neural network (FNN)) [3]

В фазе обратного распространения ошибки (рисунок 3) осуществляется ключевой аспект обучения. С помощью таких методов, как градиентный спуск, сеть «уточняет» свои внутренние параметры, вычисляя градиент функции потерь относительно весов. Градиентный спуск – ключевая сила корректировки весов нейронной сети; алгоритм оптимизации, минимизирующий функцию потерь. Цепное правило на этом этапе играет значительную роль, позволяя сети соотносить потери с определенными весами, что в свою очередь обеспечивает большую точность. Так обновляются значения весов в нейронной сети.

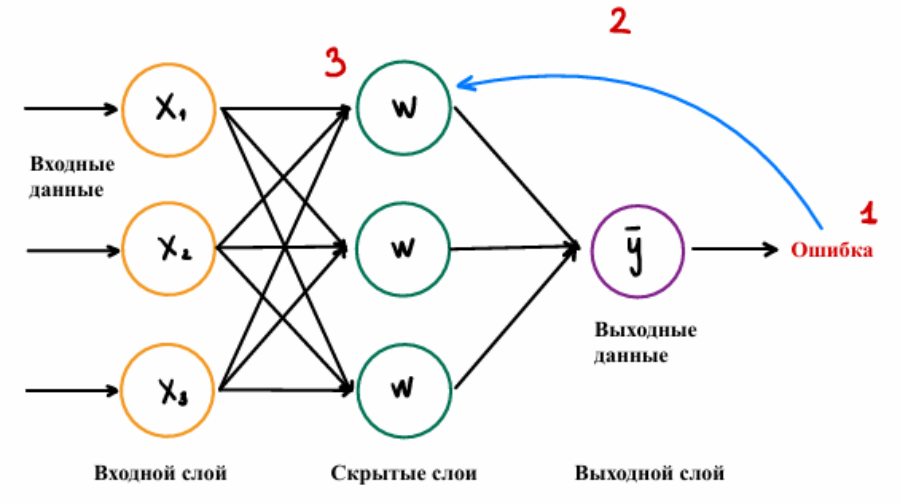

Рисунок 3 – Схема метода обратного распространения ошибки (Backpropagation) [3]

В обработке текста на естественном языке умение работы с последовательностями играет решающее значение. В отличие от традиционных задач машинного обучения, где точки данных независимы, язык включает в себя последовательную информацию. В обработке естественного языка порядок слов в предложении несет смысл, а контекст предыдущих слов влияет на интерпретацию последующих.

1.2 Задачи анализа текста

К задачам анализа текста можно отнести следующие:

- Классификация текста – определение темы или категории, к которой относится текст (например, категоризация электронной

почты или классификация «токсичности» – угроз, оскорблений и т. д.);

- Определение эмоциональной окраски текста (нейтральный, положительный или отрицательный текст) – именно такая задача будет рассмотрена в демонстрации раздела 2 на основе датаета с отзывами на фильмы;
- Извлечение сущностей из текста – примером подобного рода задач могут служить формулировки: «найти все названия компаний в тексте», «найти все имена в тексте» и т. д.;
- Генерация текста – то, что используется в чат-ботах (виртуальный собеседник, автозаполнение, ответы на вопросы);
- Автоматический (машинный) перевод;
- Реферирование (аннотирование) – извлечение основного смысла текста. Может быть извлекающее реферирование, которое фокусируется на извлечении наиболее важных предложений большого текста для их объединения и формирования в аннотацию. А может быть абстрактное реферирование, создающее аннотацию путем перефразирования;
- Исправление грамматических ошибок;
- Обнаружение спама [4, 5].

1.3 Основные архитектуры нейросетей для NLP

Работу моделей NLP можно описать так: они находят связи между составными частями языка – например, буквами, словами и предложениями, найденными в текстовом наборе данных. Архитектуры NLP используют различные методы для предварительной обработки данных, извлечения признаков и моделирования. Вот некоторые из этих процессов:

- Предварительная обработка данных («перевод» на язык, понятный для модели): стемминг (отрезание «лишнего» от корня)

и лемматизация (привод слова к канонической форме посредством морфологического анализа и словаря); сегментация предложений; удаление слов, не добавляющих много информации в текст (например, «the», «a», «an» и т. д.); токенизация (пример приведен на рисунке 4);

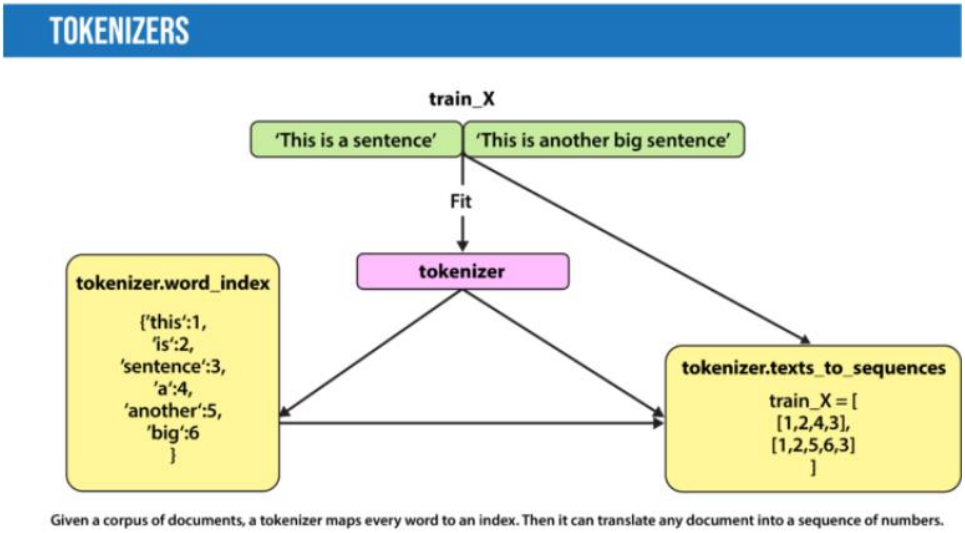

Рисунок 4 – Пример токенизации текста [5]

- Извлечение признаков:
  - Bag-of-words (мешок слов) – модель упрощенного представления текста, как набора его слов без учета грамматики и порядка слов, но с сохранением информации об их количестве (рисунок 5);

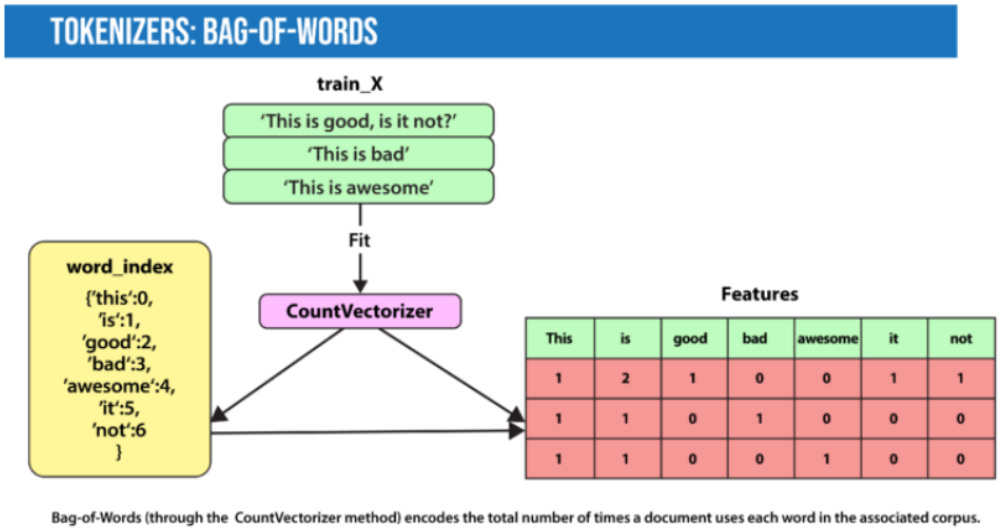

Рисунок 4 – Пример применения модели «Bag-of-words» для извлечения признаков текста [5]

- TF-IDF (term frequency (частота слова), inverse document frequency (обратная частота документа)) – статистическая мера, которая используется для оценки важности слова в конкретном документе (рисунок 6);

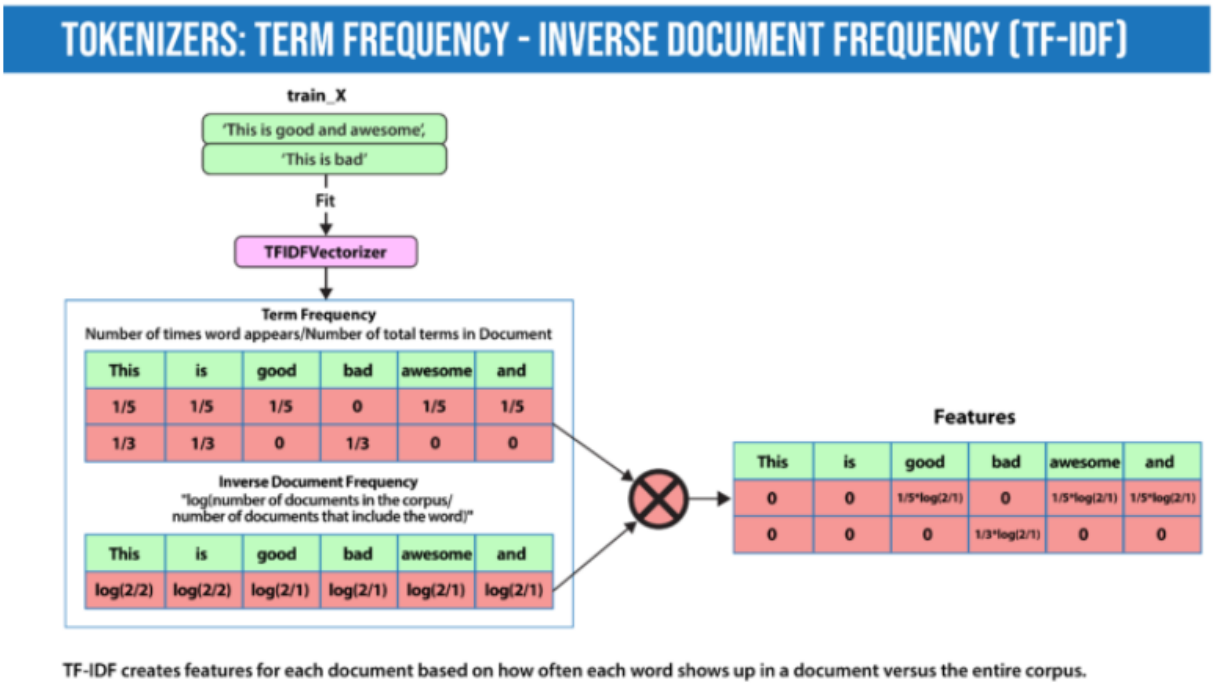

Рисунок 5 – Пример применения TF-IDF для извлечения признаков текста [5]

- Word2Vec – совокупность моделей на основе искусственных нейронных сетей, предназначенных для получения векторных представлений слов на естественном языке (рисунок 6);
- GLoVE – алгоритм обучения без учителя для получения векторных представлений слов. Этот алгоритм часто предпочтительнее Word2Vec, поскольку учитывает совместную встречаемость, а не полагается только на констекстную статистику;

- Моделирование: после предварительной обработки данные подаются в архитектуру обработки естественного языка, которая моделирует данные для выполнения различных задач [5].

В силу сформулированной темы реферата далее, в качестве архитектур обработки естественного языка, будут рассмотрены нейронные сети: RNN, LSTM, GRU и CNN.

1.3.1 RNN

RNN (recurrent neural network, рекуррентная нейронная сеть) – это специализированная форма нейронной сети, разработанная для обработки последовательных данных. Такая архитектура вводит концепцию памяти, что позволяет сети сохранить информацию о предыдущих вводах. Эта память имеет ключевое значение для задач, где важен контекст, таких как понимание и генерация языка (NLU и NLG).

Принципы работы RNN:

- Последовательная обработка;
- Рекуррентные соединения, демонстрирующие некоторую форму «памяти». На каждом шаге в последовательности RNN обрабатывает текущие входные данные вместе со «скрытым состоянием» с предыдущего шага. Это скрытое состояние содержит информацию, полученную из предыдущих входных данных;
- Скрытое состояние обновляется на каждом временном шаге на основе как новых входных данных, так и предыдущего скрытого состояния;
- Общие веса: В RNN веса (параметры) являются общими для всех временных шагов. Это означает, что для обработки каждого входного сигнала в последовательности используются одни и те же веса, что повышает эффективность модели и сокращает количество параметров [3].

RNN превосходно обрабатывают последовательные данные, что делает их подходящими для задач обработки языковых данных и анализа временных

рядов. Их способность запоминать предыдущие входные данные является неоспоримым преимуществом при работе с короткими и средними последовательностями.

Однако RNN сталкиваются с проблемой исчезающего градиента, что затрудняет их способность обрабатывать долгосрочные зависимости. Это ограничение существенно для задач, требующих обширного исторического контекста. Кроме того, их последовательный характер ограничивает использование современных технологий параллельной обработки, что приводит к увеличению времени обучения. Несмотря на эти трудности, рекуррентные нейронные сети остаются востребованными при обработке последовательностей. На рисунке 6 представлено схема RNN.

RNN находят применение в обработке естественного языка (языковое моделирование, машинный перевод), распознавании речи (распознавание фонем, синтез голоса) и прогнозировании временных рядов (прогнозирование цен на акции, прогноз погоды).

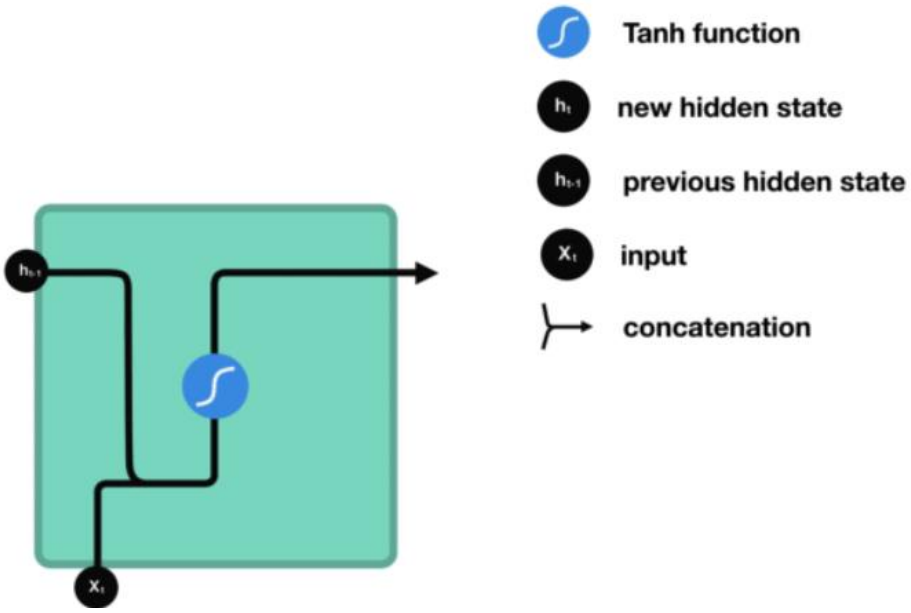

Рисунок 6 – Схема RNN [6]

1.3.2 LSTM

LSTM (long short-term memory, долгая краткосрочная память) – передовая разработка в мире рекуррентных нейронных сетей, специально

разработанная для устранения ограничений, присущих традиционным RNN, особенно при работе с долгосрочными зависимостями.

Принципы работы LSTM:

- Отличительной особенностью LSTM является его сложная ячейка памяти, известная как модуль LSTM. Этот модуль способен сохранять информацию в течение длительного времени;
- Механизм вентилей – LSTM включают в себя три типа вентилей, которые используются для контроля потоков информации на входах и на выходах памяти данных блоков. Эти вентили (входной вентиль, вентиль забывания, выходной вентиль) реализованы в виде логистической функции для вычисления значения в диапазоне [0; 1];
- В основе LSTM лежит состояние ячейки, своего рода конвейерная лента, проходящая по всей цепочке сети. Это позволяет передавать информацию относительно без изменений и гарантирует, что сеть эффективно сохраняет важную долгосрочную информацию и получает к ней доступ [3].

LSTM специально разработаны для того, чтобы избежать проблемы долгосрочной зависимости, что делает их более эффективными для задач, требующих понимания информации за длительные периоды времени.

Однако они более сложны и требуют больших вычислительных затрат по сравнению с базовыми RNN и GRU (см. раздел 1.3.3), что может быть проблемой с точки зрения времени обучения и распределения ресурсов.

LSTM доказали свою эффективность в различных областях, требующих обработки последовательностей с долгосрочной зависимостью, таких как сложные структуры предложений в тексте, распознавание речи и анализ временных рядов. Они являются мощным инструментом в арсенале архитектур нейронных сетей, особенно подходящим для задач глубокого обучения в NLP и за его пределами. На рисунке 7 представлена схема LSTM.

1.3.3 GRU

GRU (gated recurrent units, управляемые рекуррентные блоки) – это инновационная разновидность рекуррентных нейронных сетей, разработанная для улучшения и упрощения архитектуры LSTM. GRU предлагают более рациональный подход к обработке последовательных данных, особенно в сценариях, где важны долгосрочные зависимости.

Работа GRU:

- Упрощенная архитектура по сравнению с LSTM, что повышает эффективность с точки зрения вычислительных ресурсов (за счет уменьшенного количества вентилей).
- Механизм вентилей. По сравнению с LSTM отсутствует выходной вентиль;
- В отличие от LSTM, GRU не имеют отдельного состояния ячейки. Они объединяют состояние ячейки и скрытое состояние в единую структуру, упрощая поток информации и облегчая их моделирование и обучение [3].

GRU особенно известны своей эффективностью и скоростью обучения, что делает их подходящим выбором для моделей, где важны вычислительные ресурсы.

Хотя они, как правило, быстрее и проще, чем LSTM, они могут быть не столь эффективны при захвате очень долгосрочных зависимостей из-за своей упрощенной структуры.

GRU успешно применялись в различных областях, таких как языковое моделирование, машинный перевод и приложения преобразования речи в текст, где баланс между сложностью и производительностью имеет решающее значение. На рисунке 7 представлена схема GRU.

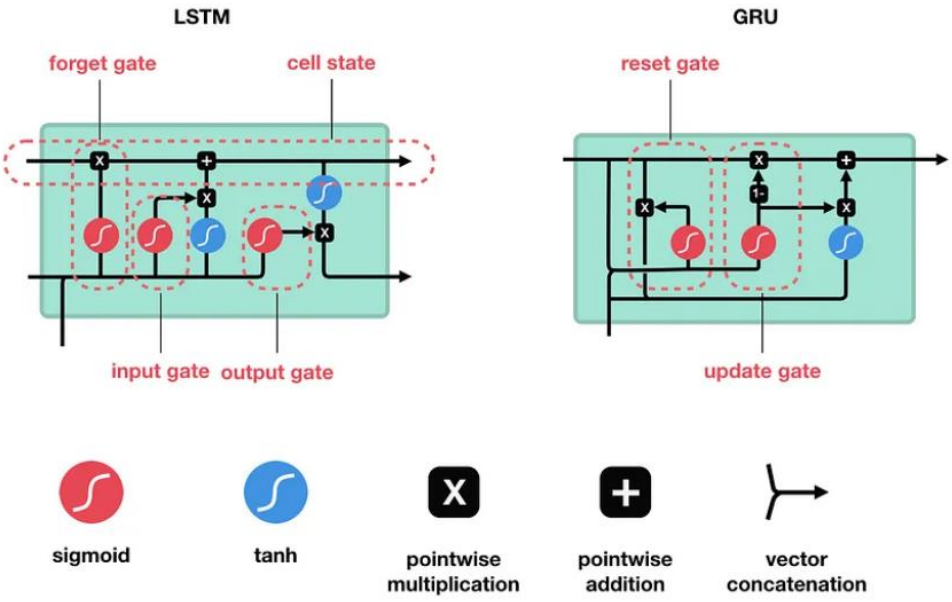

Рисунок 7 – Схемы LSTM и GRU [6]

1.3.4 CNN

CNN (convolutional neural networks, сверточные нейронные сети) ассоциируются в первую очередь с задачами компьютерного зрения (распознаванием изображений), но они также набирают популярность в области обработки естественного языка благодаря своей способности эффективно обрабатывать последовательные данные. В NLP CNN в основном используются для таких задач, как классификация текста, анализ настроений и категоризация документов.

CNN в NLP работают аналогично тому, как они используются в компьютерном зрении, но с текстовыми данными. Вместо пикселей CNN обрабатывают слова или символы в качестве входных данных. Основная идея заключается в использовании сверточных слоев для обнаружения шаблонов или особенностей на разных уровнях детализации в тексте.

Принцип работы:

- Встраивание слов. Преобразование слов в плотные числовые векторы с использованием таких методов, как Word2Vec или GloVe. На этом этапе текст преобразуется в формат, подходящий для нейронных сетей;

- Сверточные слои. CNN используют сверточные слои для сканирования входного текста, фиксируя локальные закономерности. Каждый сверточный фильтр действует как детектор признаков, выявляя определенные языковые паттерны или сочетания слов;
- Слои пулинга. После свертки слои объединяют информацию из выходных данных свертки, уменьшая размерность при сохранении важных объектов;
- Полносвязная нейронная сеть. Эти слои объединяют данные, полученные с помощью сверточных слоев и слоев пулинга, сопоставляя их с желаемыми выходными классами [7].

CNN в NLP продемонстрировали эффективность в различных задачах благодаря своей способности улавливать локальные закономерности и иерархические отношения в текстовых данных. Они могут различать важные элементы независимо от их положения в предложении, что делает их надежными для таких задач, как анализ тональности или классификация документов, где важен контекст. Несмотря на свою простоту, CNN обеспечивают конкурентоспособную производительность и эффективность в решении задач NLP.

1.4 Сравнение архитектур

Сравнение будет удобно представить в виде таблицы 1.

Таблица 1 – Сравнение нейронных сетей для обработки естественного языка

RNN	**Сильные стороны:** • Обработка последовательностей и сохранение информации в течение коротких промежутков времени; • Простая архитектура делает рекуррентные нейронные сети вычислительно эффективными. **Ограничения:** • Проблемы с долгосрочными зависимостями из-за исчезающего градиента.

Продолжение таблицы 1

LSTM	**Сильные стороны:** • Высокоэффективны в изучении долгосрочных зависимостей; • Добавление вентилей ввода, забывания и вывода позволяет лучше контролировать ячейку памяти, что позволяет решить проблему исчезающего градиента. **Сложность:** • Сложнее, чем RNN, что приводит к более высоким вычислительным затратам.
GRU	**Сильные стороны:** • Управление долгосрочными зависимостям • GRU объединяют вентили ввода и забывания в один вентиль обновления, что снижает сложность архитектуры. **Эффективность:** • Быстрее обучается, чем LSTM (из-за меньшего количества параметров), при этом часто достигает аналогичной производительности.
CNN	**Сильные стороны:** • Обработка последовательностей и извлечение локальных закономерностей, простота архитектуры • Скорость обработки и обучения; **Ограничения:** • Менее эффективны в захвате долгосрочных зависимостей между словами в длинных текстах.

2 Демонстрация обработки текста на основе датасета IBDM

В данном разделе будет произведена демонстрация анализа тональности отзывов на фильмы IMDB [8] с помощью LSTM.

2.1 Подготовительный этап и требования

Демонстрация будет осуществляться на платформе Google Colab [9] – сервиса Jupyter Notebook, не требующего настройки для использования и предоставляющего бесплатный доступ к вычислительным ресурсам, включая графические и тензорные процессоры.

Язык демонстрации – Python. Используемые библиотеки и инструменты представлены листингом 1.

Листинг 1 – Используемые библиотеки

```
import numpy as np
from tensorflow.keras.preprocessing import sequence
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation, Embedding
from tensorflow.keras.layers import LSTM, SpatialDropout1D
from tensorflow.keras.datasets import imdb
import matplotlib.pyplot as plt
%matplotlib inline
```

В качестве датасета для обучения будет использоваться набор данных IMDB movie review (он уже включен в keras) [10]. Он был создан для задач определения тональности текста. Набор состоит из 50000 отзывов на фильмы с сайта IMDB:

- 25000 для обучения;
- 25000 для тестирования.

Нейтральные отзывы учитываться в демонстрации не будут. Отзывы могут быть либо положительными (если оценка не меньше 7), либо отрицательные (оценка не превосходит 4). Количество положительных и отрицательных отзывов одинаково.

Обозначения вводятся следующим образом:

- 0 – отрицательный отзыв;
- 1 – положительный отзыв.

Таким образом, это задача бинарной классификации.

2.2 Демонстрация

Первое, что нужно сделать после предварительной подготовки – загрузить данные. Загрузка представлена на листинге 2. Необходимо ввести переменную, отвечающую за максимальное количество слов, которое будет использоваться для анализа. В нашем случае – 5000. Это значит, что выбираются 5000 самых частовстречающихся слов в датасете imdb.

Листинг 2 – Загрузка данных

```
max_num_of_wrds = 5000
(x, y), (x_, y_) = imdb.load_data(num_words=max_num_of_wrds)
```

Чтобы продемонстрировать, как выглядят загруженные данные, воспользуемся командой, представленной листингом 3. То есть было произведено обращение к рецензии с номером 7. Фрагмент результата исполнения показан на рисунке 8 – каждое число соответствует одному слову из исходного отзыва (токенизация на уровне слов). Листинг 4 соответствует команде, демонстрирующей правильный ответ (положительный или отрицательный отзыв). Результат исполнения показан на рисунке 9 – отзыв отрицательный.

Листинг 3 – Просмотр данных (рецензия)

```
x[7]
```

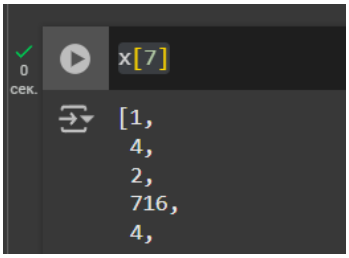

Рисунок 8 – Просмотр загруженных данных (рецензия)

Листинг 4 – Просмотр данных (правильный ответ)

```
y[7]
```

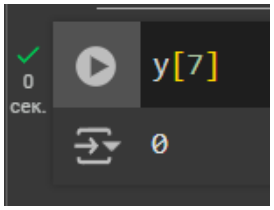

Рисунок 9 – Просмотр загруженных данных (правильный ответ)

Раскодируем текст рецензии – узнаем, что значат числа рецензии на рисунке 8. Для этого загрузим словарь, который использовался для кодирования – листинг 5. На рисунке 10 показан результат исполнения листинга 5.

Листинг 5 – Загрузка словаря

```
word_indx = imdb.get_word_index()
word_indx
```

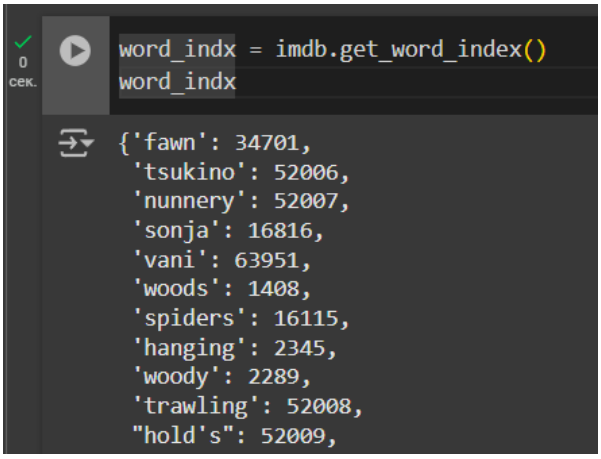

Рисунок 10 – Фрагмент словаря

Листинг 6 демонстрирует фрагмент, который преобразовывает словарь так, чтобы получать слово по номеру. Для получения раскодированной рецензии воспользуемся кодом, представленным на листинге 7. Результат исполнения этого кода показан на рисунке 11.

Листинг 6 – Преобразование словаря

```
rev_word_indx = dict()
for a, b in word_indx.items():
  rev_word_indx[b] = a
```

Листинг 7 – Раскодирование рецензии с номером 7

```
res = ''
for el in x[7]:
  word = rev_word_indx.get(el-3, '?')
  res += word + ' '
res
```

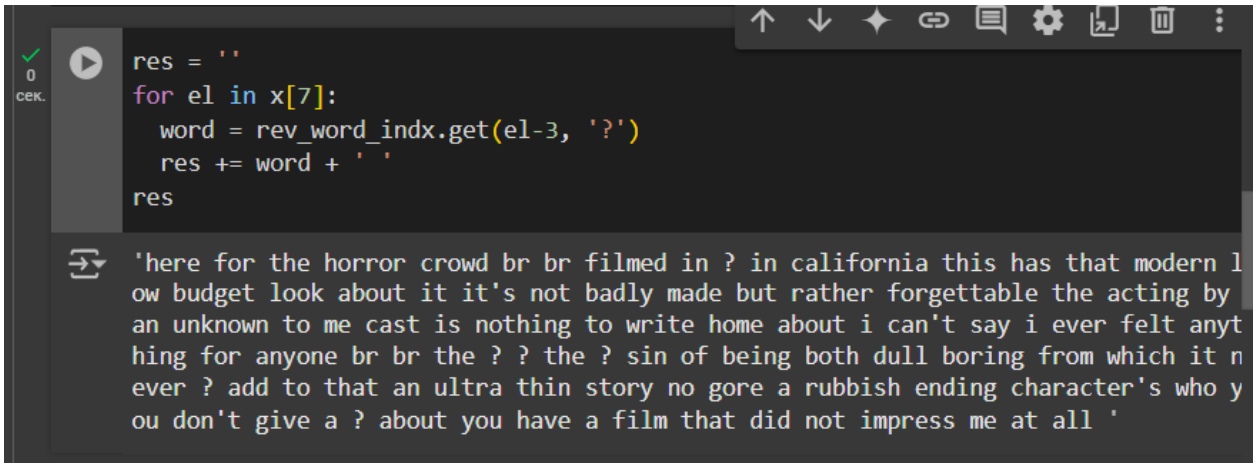

Рисунок 10 – Полученная рецензия (по содержанию видно, что тональность действительно негативная)

Знак вопроса в полученной рецензии – это слово, которое не вошло в 5000 используемых нами для анализа слов.

Кроме того, имеют место следующие служебные коды:

- 0 – символ заполнитель;
- 1 – начало последовательности;
- 2 – неизвестное слово.

Именно по этой причине, для корректной раскодировки, необходимо действие «$el - 3$» (листинг 7 строка 3).

Теперь необходимо произвести подготовку данных для обучения. Нужно сделать так, чтобы длина всех отзывов была одинакова – добавить символов-заполнителей или «отрезать» часть данных. Листинг 8 демонстрирует фрагмент, который выравнивает длину рецензий.

Листинг 8 – Подготовка данных для обучения (длина всех рецензий)

```
max_len = 100
x = sequence.pad_sequences(x,maxlen=max_len, padding='post')
x_ = sequence.pad_sequences(x_, maxlen=max_len, padding='post')
x
```

На рисунке 11 показано, как теперь выглядят данные.

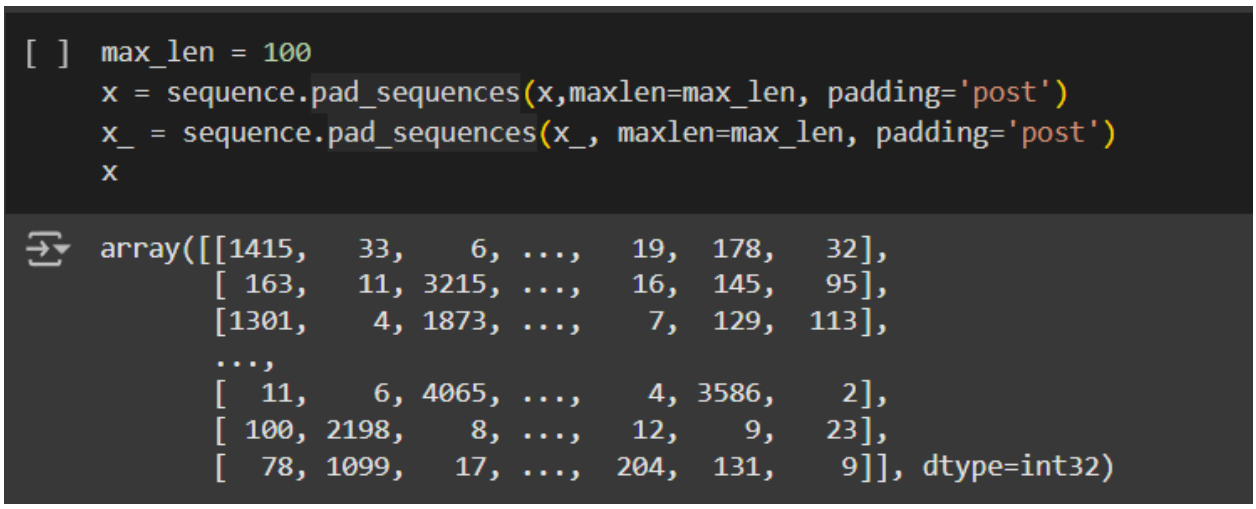

Рисунок 11 – Подготовленные данные

Теперь необходимо создать нейронную сеть, которая будет заниматься классификацией текста. Создается модель, в которую будут добавляться слои, один из которых – LSTM. Создание сети демонстрирует листинг 9.

Листинг 9 – Создание нейронной сети

```
model = Sequential()
model.add(Embedding(max_num_of_wrds, 32))
model.add(SpatialDropout1D(0.2))
model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(1, activation="sigmoid"))
```

Обучение модели представлено на листинге 10. Размер батча – 64. Это значит, что в одном батче представлено 64 тренировочных объекта. Количество эпох 7. На рисунке 12 представлены выводы процесса обучения. Обучение сети длилось 6 минут.

Листинг 10 – Обучение нейронной сети

```
history = model.fit(x, y, epochs=5, batch_size=64, validation_data=(x, y), verbose=2)
```

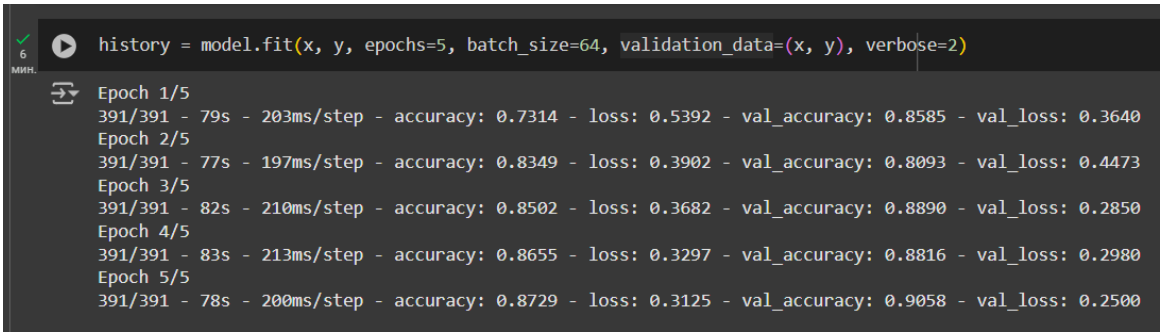

Рисунок 12 – Обучение сети

В качестве результатов был создан график, показанный на рисунке 13. Функция возрастает. А также был посчитан процент правильных ответов (листинг 11, рисунок 14).

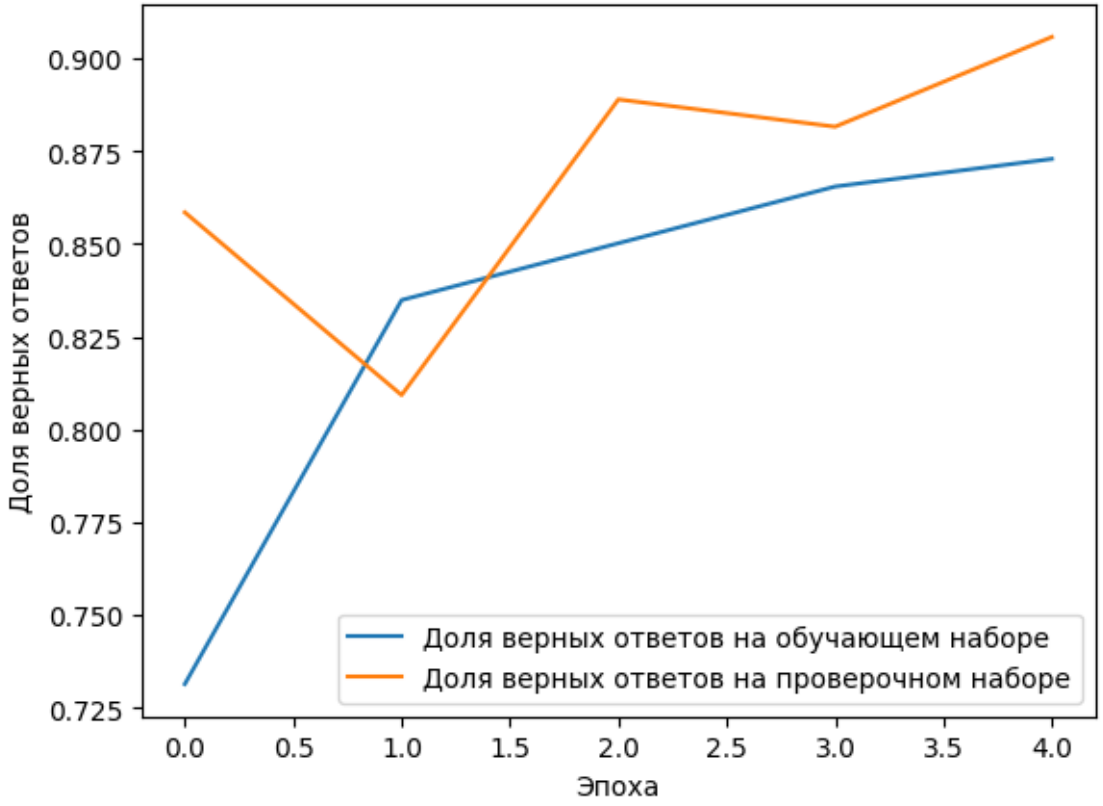

Рисунок 13 – График результатов тренировки нейронной сети

Листинг 11 – Подсчет доли правильных ответов в процентах

```
scores = model.evaluate(x_, y_, batch_size=64)
print(f'Доля верных ответов в %: {round(scores[1]*100, 4)}')
```

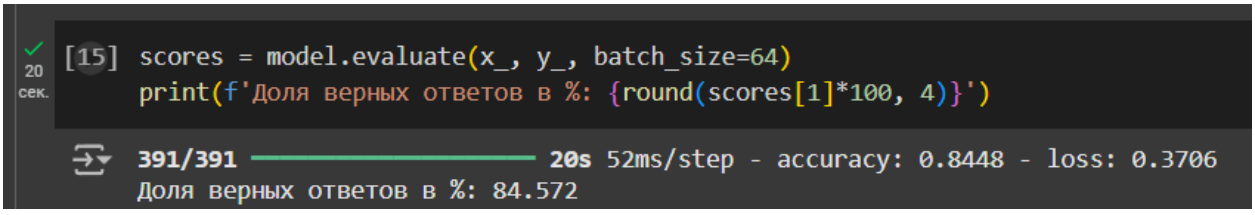

Рисунок 14 – Доля верных ответов в процентах

Листинг кода целиком приведен в приложении А.

ЗАКЛЮЧЕНИЕ

В данной работе исследовалась роль нейронных сетей в развитии обработки текста на естественном языке. Были приобретены новые знания как в области нейронных сетей, так и в области NLP, включая базовые концепции и определения. Эти знания позволили глубже изучить архитектуру рекуррентных нейронных сетей (RNN), в частности LSTM и GRU, а также сверточных нейронных сетей (CNN) и подчеркнуть их сильные стороны и возможности применения в обработке текста.

Практическое применение этих теоретических знаний было продемонстрировано при разработке модели анализа настроений с использованием набора данных с отзывами на фильмы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 What is NLP? [Электронный ресурс] // IBM [Электронный ресурс]. - URL: https://www.ibm.com/think/topics/natural-language-processing (дата обращения: 17.12.2024). - Загл. с экрана. - Яз. англ.

2 What is NLP (Natural Language Processing)? [Электронный ресурс] // YouTube [Электронный ресурс]. - URL: https://www.youtube.com/watch?v=fLvJ8VdHLA0 (дата обращения: 17.12.24). - Загл. с экрана. - Яз. англ.

3 Neural Networks in NLP: RNN, LSTM, and GRU [Электронный ресурс] // Medium [Электронный ресурс]. - URL: https://medium.com/@mervebdurna/nlp-with-deep-learning-neural-networks-rnns-lstms-and-gru-3de7289bb4f8 (дата обращения: 17.12.2024). - Загл. с экрана. - Яз. англ.

4 Нейронные сети для обработки естественного языка | Нейросети для анализа текстов [Электронный ресурс] // YouTube [Электронный ресурс]. - URL: https://www.youtube.com/watch?v=O5-qYR09zQw (дата обращения: 17.12.2024). - Загл. с экрана. - Яз. рус.

5 A Complete Guide to Natural Language Processing [Электронный ресурс] // DeepLearning.AI [Электронный ресурс]. - URL: https://www.deeplearning.ai/resources/natural-language-processing/ (дата обращения: 17.12.2024). - Загл. с экрана. - Яз. англ.

6 Illustrated Guide to LSTM’s and GRU’s: A step by step explanation [Электронный ресурс] // Medium [Электронный ресурс]. - URL: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21 (дата обращения: 17.12.2024). - Загл. с экрана. - Яз. англ.

7 How Is Convolutional Neural Network (CNN) Used In NLP ? [Электронный ресурс] // LinkedIn [Электронный ресурс]. - URL:

https://www.linkedin.com/pulse/how-convolutional-neural-network-cnn-used-nlp-korvage-wppcc (дата обращения: 17.12.2024). - Загл. с экрана. - Яз. англ.

8 IMDb [Электронный ресурс] // IMDb [Электронный ресурс]. - URL: https://www.imdb.com/ (дата обращения: 18.12.2024). - Загл. с экрана. - Яз. англ.

9 Demonstration.ipynb [Электронный ресурс] // Google Colab [Электронный ресурс]. - URL: https://colab.research.google.com/drive/15p56PYCtyt2vIBsbV-0RYscXYr4iOGlF#scrollTo=ZxsAGHKBq8F0 (дата обращения: 18.12.2024). - Загл. с экрана. - Яз. англ.

10 Large Movie Review Dataset [Электронный ресурс] // ai.stanford.edu [Электронный ресурс]. - URL: https://ai.stanford.edu/~amaas/data/sentiment/ (дата обращения: 18.12.2024). - Загл. с экрана. - Яз. англ.

ПРИЛОЖЕНИЕ А

Листинг программы imdb.py

```
import numpy as np
from tensorflow.keras.preprocessing import sequence
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation, Embedding
from tensorflow.keras.layers import LSTM, SpatialDropout1D
from tensorflow.keras.datasets import imdb
import matplotlib.pyplot as plt
%matplotlib inline

""" Демонстрация создания нейронной сети с использованием keras и
tensorflow """

# загрузка данных
max_num_of_wrds = 5000
(x, y), (x_, y_) = imdb.load_data(num_words=max_num_of_wrds)

# обработка словаря
word_indx = imdb.get_word_index()
rev_word_indx = dict()
for a, b in word_indx.items():
  rev_word_indx[b] = a

# получение текста рецензии
res = ''
for el in x[7]:
  word = rev_word_indx.get(el-3, '?')
  res += word + ' '

# нормализация данных по длине
max_len = 100
x = sequence.pad_sequences(x,maxlen=max_len, padding='post')
x_ = sequence.pad_sequences(x_, maxlen=max_len, padding='post')

# создание нейронной сети
model = Sequential()
model.add(Embedding(max_num_of_wrds, 32))
model.add(SpatialDropout1D(0.2))
model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(1, activation="sigmoid"))

# компиляция нейронной сети
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])

# обучение нейронной сети
history = model.fit(x, y, epochs=5, batch_size=64, validation_data=(x,
y), verbose=2)

# построение графика
plt.plot(history.history['accuracy'],
```

```
         label='Доля верных ответов на обучающем наборе')
plt.plot(history.history['val_accuracy'],
         label='Доля верных ответов на проверочном наборе')
plt.xlabel('Эпоха')
plt.ylabel('Доля верных ответов')
plt.legend()
plt.show()

# подсчет доли верных ответов
scores = model.evaluate(x_, y_, batch_size=64)
print(f'Доля верных ответов в %: {round(scores[1]*100, 4)}')
```