ФЕДЕРАЛЬНОЕ АГЕНСТВО СВЯЗИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

Московский технический университет связи и информатики

(МТУСИ)

Кафедра «Математическая кибернетика и информационные технологии» Дисциплина «Интеллектуальные системы»

Лабораторная работа № 6

«Прогноз успеха фильмов по обзорам»

Выполнил: \_\_\_\_\_\_\_ Д.С. Калининский

Проверил: \_\_\_\_\_\_\_ А.В. Шереметьев

Москва 2022

**Оглавление**

[1 Цель работы 3](#_Toc119580633)

[2 Задачи 3](#_Toc119580634)

[3 Выполнение работы 3](#_Toc119580635)

[4 Результаты 5](#_Toc119580636)

[5 Вывод 7](#_Toc119580637)

# **1 Цель работы**

Прогноз успеха фильмов по обзорам (Predict Sentiment From Movie Reviews).

# **2 Задачи**

1) Ознакомиться с задачей классификации;

2) Изучить способы представления текста для передачи в ИНС;

3) Достигнуть точность прогноза не менее 95%.

# **3 Выполнение работы**

Начнем с импорта необходимых зависимостей для предварительной обработки данных и построения модели (рисунок 1).

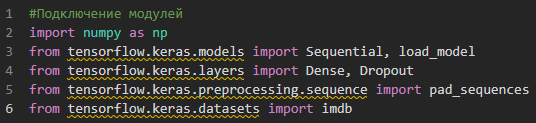


Рисунок 1 – Импорт необходимых модулей

Загрузим датесет IMDb, который уже встроен в Keras. Поскольку мы не хотим иметь данные обучения и тестирования в пропорции 50/50, мы сразу же объединим эти данные после загрузки для последующего разделения в пропорции 80/20. Для этого используем процедуру preprocess() (рисунок 2). Реализуется загрузка и чтение данных методом imdb.load\_data(), с параметром num\_words, который отвечает за размер вектора представления текста. Соединяются обучающая и тестовая выборки, обучающая векторизуется, и происходит преобразование выходных признаков в «float32» Функция preprocess() возвращает входные и выходные признаки обучающей и тестовой выборок, разбитые размером вектора представления текста в виде кортежа.

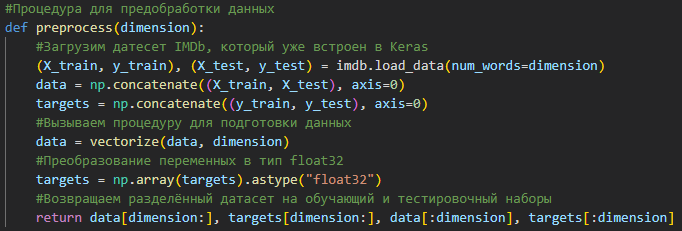


Рисунок 2 – Функция preprocess()

Пришло время подготовить данные. Нужно векторизовать каждый обзор и заполнить его нулями, чтобы вектор содержал ровно 10 000 чисел. Это означает, что каждый обзор, который короче 10 000 слов, мы заполняем нулями. Это делается потому, что самый большой обзор имеет почти такой же размер, а каждый элемент входных данных нашей нейронной сети должен иметь одинаковый размер (рисунок 3).

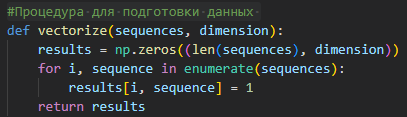


Рисунок 3 – Функция vectorize()

Для предобработки данных для тестирования модели были созданы две функции encode() (рисунок 4) и text\_pred() (рисунок 5). Для начала необходимо получить числовые индексы для слов, известных дата сету imdb при помощи метода get\_word\_index(). Заполняется словарь индексов my\_word\_index значениями больше на 3, так как для специальных символов pad, start\_char и oov\_char даны индексы 1, 2 и 3. Метод pad\_sequences в качестве максимального значения длины под-массивов берет 17, так как максимальная длина тестового отзыва равна 17 словам, затем заполняет каждый под-массив в arr нулями справа, ели длина меньше 17. Encode() возвращает полученный результат. Функция text\_pred разбивает предложение на слова, а затем выполняет преобразование созданной функцией encode. Полученный результат векторизуется. Так как модель загружается из каталога folder, всегда следует обращать внимание на соответствие значения dimension в vectorize() и размерности входной выборки в ИНС. Text\_pred() возвращает значения вероятности принадлежности отзыва к положительной категории.).

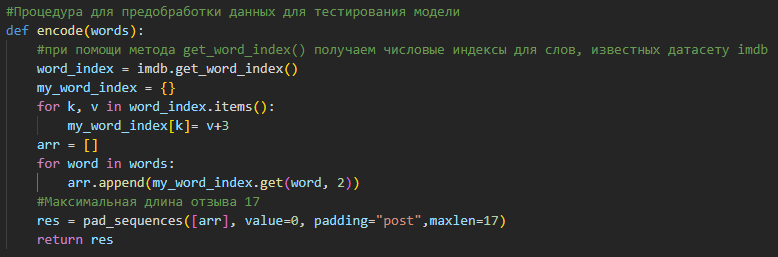


Рисунок 4 – Функция encode()

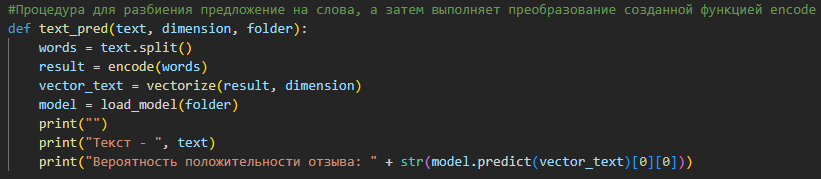


Рисунок 5 – Функция text\_pred()

Теперь можно создать простую нейронную сеть. Начнем с определения типа модели, которую мы хотим создать. В Keras доступны два типа моделей: последовательные и с функциональным API.

Затем нужно добавить входные, скрытые и выходные слои. Для предотвращения переобучения будем использовать между ними исключение («dropout»). Обратите внимание, что вы всегда должны использовать коэффициент исключения в диапазоне от 20% до 50%. На каждом слое используется функция «dense» для полного соединения слоев друг с другом. В скрытых слоях будем используем функцию активации «relu», потому это практически всегда приводит к удовлетворительным результатам. Не бойтесь экспериментировать с другими функциями активации. На выходном слое используем сигмоидную функцию, которая выполняет перенормировку значений в диапазоне от 0 до 1 (рисунок 6).

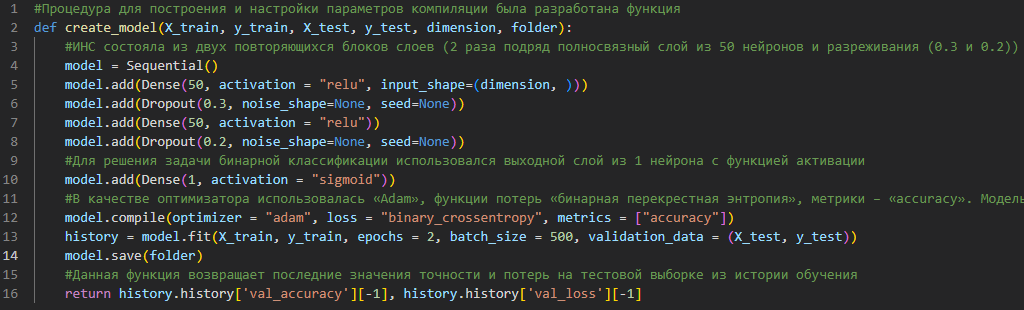


Рисунок 6 – Функция создания модели

# **4 Результаты**

В данной лабораторной работе помимо построения и обучения нейронной сети, нужно исследовать результаты при различном размере вектора представления текста (рисунок 7). Результаты тестирования размеров вектора представлены в таблице 1.

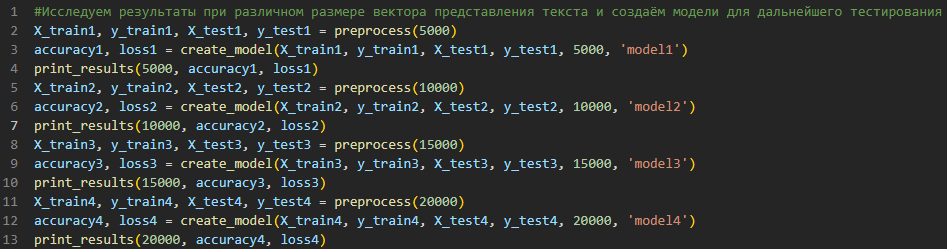


Рисунок 7 – Тестирование модели при различных размерах вектора

Таблица 1 – Результаты тестирования модели при различных значения вектора

|  |  |  |  |
| --- | --- | --- | --- |
| Модель | Размер вектора | Точность | Потери |
| 1 | 5000 | 0.894599 | 0.265854 |
| 2 | 10000 | 0.896001 | 0.257521 |
| 3 | 15000 | 0.892933 | 0.266741 |
| 4 | 20000 | 0.890649 | 0.276002 |

Исходя из полученных результатов делаем вывод, что вторая модель является самой оптимальной так, как имеет наивысшую точность и наименьшие потери, поэтому дальнейшие тестирования фильмов по образам проведём при помощи неё (рисунок 8).

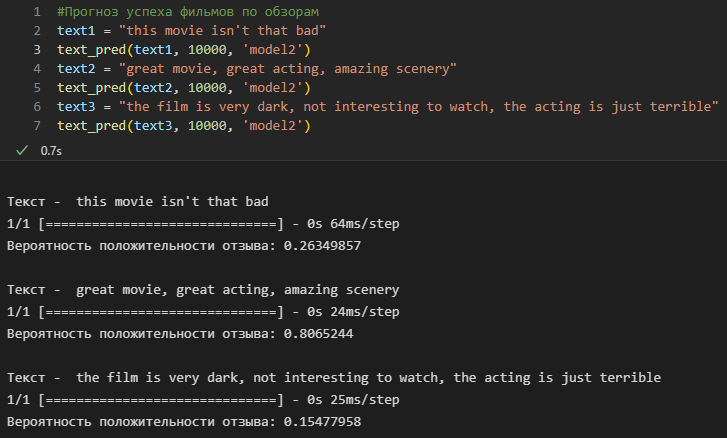


Рисунок 8 – Результат оценки

# **5 Вывод**

В данной лабораторной работе, был реализован прогноз успеха фильмов по обзорам. Для реализации задачи была построена модель и протестирована при различных размерах вектора, была выбрана наилучшая модель и при помощи неё произведена оценка, в результате которой были выявлены уязвимости данной оценки. Данная модель ошибается при подачи отрицательных слов, хотя образ может быть хорошим в контексте, это происходит из-за того, что модель даёт результат анализирую каждое слово по отдельности.