**МИНОБРНАУКИ РОССИИ**
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**
**МО ЭВМ**

# ОТЧЕТ

**по лабораторной работе №6**

**по дисциплине «Искусственные нейронные сети»**

**«Прогноз успеха фильмов по обзорам»**

Студент гр. 8383 _____ Костарев К.В.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2021

**Цель работы.**

Спрогнозировать успех фильмов по обзорам.

**Постановка задачи.**

1. Ознакомиться с задачей анализа настроений
2. Изучить способы представления текста для передачи в ИНС\

**Выполнение работы.**

Анализ настроений – это определение по тексту отношения его автора к тому, о чем текст написан. Например, определение мнения клиентов о продукте по отзывам.

В программу загружается датасет IMDb, причем при загрузке сохраняются только 10000 самых часто встречающихся слов. Каждый обзор датасета представляет собой список индексов слов в словаре, причем сдвинутый: индекс 0 соответствует неизвестным словам, 1 – индикатор начала обзора, 2 заменяет слова, которые не вошли в заданное число самых часто встречающихся. Сами слова начинаются с индекса 3, что необходимо учитывать при декодировании обзоров и кодировании пользовательского текста.

Каждый обзор датасета необходимо представить в виде вектора. Так как мы знаем наибольшее число уникальных слов, которые могут встретиться в обзорах, представим датасет в виде матрицы размера len(sequences)×dimension, где len(sequences) – число обзоров в датасете, dimension – число уникальных 3 слов. Каждая строка матрицы соответствует одному обзору. В этой строке единицы проставляются в столбцах с индексами слов, которые присутствуют в обзоре, а в остальных столбцах ставятся нули.

Создадим модель нейронной сети. Входной слой модели соответствует длине вектора, задающего каждый обзор. Модель состоит из трех скрытых полносвязных слоев с 50 нейронами, использующими функцию активации relu, каждый. После первого слоя расположим слой прореживания с

вероятностью 0.5, после второго – слой прореживания с вероятностью 0.25. На выходном слое будем использовать один нейрон с функций активации sigmoid для осуществления бинарной классификации. Конфигурация сети:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 50)                500050
_________________________________________________________________
dropout (Dropout)            (None, 50)                0
_________________________________________________________________
dense_1 (Dense)              (None, 50)                2550
_________________________________________________________________
dropout_1 (Dropout)          (None, 50)                0
_________________________________________________________________
dense_2 (Dense)              (None, 50)                2550
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 51
=================================================================
Total params: 505,201
Trainable params: 505,201
Non-trainable params: 0
```

При обучении модели будем использовать оптимизатор adam, в качестве функции потерь – бинарную энтропию. Обучим модель в течение двух эпох пакетами по 500 образцов, а затем вычислим среднее значение точности в процессе обучения.

```
Epoch 1/2
80/80 [==============================] - 3s 28ms/step - loss: 0.4389 -
accuracy: 0.7969 - val_loss: 0.2665 - val_accuracy: 0.8940
Epoch 2/2
80/80 [==============================] - 2s 23ms/step - loss: 0.2381 -
accuracy: 0.9086 - val_loss: 0.2608 - val_accuracy: 0.8943
0.8941499888896942
```

Получена точность порядка 89,4%.

Исследуем, как модель работает с пользовательским текстом. Сначала напишем функцию, которая будет считывать текст из файла и кодировать его при помощи словаря.

```
def read_txt(filepath, max_words=10000):
    f = open(filepath, 'r')
    txt = f.read().lower()
    txt = re.sub(r"[^a-zA-Z0-9']", " ", txt)  # убираем лишние символы
    print(txt)
```

```
        txt = txt.split()  # разбиваем на слова
        index = imdb.get_word_index()  # загружаем словарь
        coded = [1]  # индекс начала последовательности
        coded.extend([index.get(i, 0) for i in txt])
        for i in range(len(coded)):
            if coded[i]:
                coded[i] += 3  # смещаем индексы
            if coded[i] >= max_words:
                coded[i] = 2  # отмечаем слова, не вошедшие в число max_words
самых популярных
        print(coded)
        return coded
```

Функция читает текст из файла и сразу же приводит все буквы к нижнему регистру. Затем при помощи регулярных выражений все символы в тексте, кроме английских букв, цифр и апострофа, заменяются пробелами. «Очищенный» текст делится на отдельные слова. Загружается словарь, где каждому слову сопоставляется его индекс. В список с индексами слов сразу добавляется число 1, обозначающее начало последовательности, затем в соответствии со словарем добавляются индексы слов обзора. Если слова нет в словаре, ему присваивается индекс 0. В полученном списке индексов все ненулевые индексы, кроме первого (индекса начала последовательности) увеличиваются на три, а индексы, которые оказываются больше, чем число учитываемых самых часто встречающихся слов, заменяются на 2. Функция возвращает закодированную последовательность, которую перед передачей в нейросеть необходимо будет представить в векторном виде.

Исследуем работу нейросети на четырех обзорах фильма «Дрянные девчонки», скопированных с сайта «Rotten tomatoes». В таблице ниже приведены исходный текст обзора и результат предсказания.

| Исходный текст | Результат предсказания |
|---|---|
| Cringy boring but easy story to follow ... a few funny moments in there as well. Actors are ok but dont expect a deep story at all (test1.txt) | 0.26525092 (отрицательный обзор) |
| Mean Girls grabs the base formula of a teen movie and takes it to another level with clever | 0.90140873 (положительный обзор) |

| | |
|---|---|
| jokes, an engaging plot, and many memorable moments. The acting is better than other comedy/teen movies, and the visual dynamics and movement are well executed. (test2.txt) | |
| Mean Girls is one of the most iconic movies of all time. Every line is iconic and is quoted all the time. As a person who prefers movies from the early 2000's I might be a bit bias when I say that Mean Girls definitely makes my top 10 favorite movies list. (test3.txt) | 0.9432686 (положительный обзор) |
| I watched because it was on TV on a Saturday afternoon. I thought it was over and changed channels, then changed back twenty minutes later. It was still playing--30 minutes too ling and all sickening mush. (test4.txt) | 0.30693814 (отрицательный обзор) |

Нейросеть правильно классифицировала все четыре примера, причем, сравнивая числа, которые нейросеть вернула для разных отзывов, можно заметить, как на результат влияет использование более эмоционально окрашенной лексики.

Исследуем работу нейросети при различных размерах вектора представления. Для этого при загрузке датасета необходимо сократить число загружаемых слов. Загрузим 1000 наиболее часто встречающихся слов, векторизуем данные в соответствии с новым ограничением. Обучим модель и оценим ее точность:

```
Epoch 1/2
98/98 [==============================] - 1s 8ms/step - loss: 0.5502 - accuracy: 0.7045 - val_loss: 0.3509 - val_accuracy: 0.8560
Epoch 2/2
98/98 [==============================] - 1s 6ms/step - loss: 0.3666 - accuracy: 0.8425 - val_loss: 0.3102 - val_accuracy: 0.8630
0.859499990940094
```

Точность снизилась. Учитывая, что при изменении числа загружаемых слов, остальные просто теряются, снижение точности было ожидаемо, и скорее удивительно то, что она снизилась не слишком сильно.

**Выводы.**

Была создана нейронная сеть, осуществляющая анализ настроений, то есть классификацию отзывов о фильмах на положительные и отрицательные. В ходе выполнения работы был изучен один из способов представления текста для передачи его в ИНС. Была реализована функция, осуществляющая чтение пользовательского текста из файла и его представление в виде индексов слов. Созданная нейросеть была протестирована на нескольких пользовательских примерах – все примеры были классифицированы правильно.