

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

**ОТЧЕТ**  
**по лабораторной работе №6**  
**по дисциплине «Искусственные нейронные сети»**  
**Тема: Прогноз успеха фильмов по обзорам**

Студентка гр. 8382

\_\_\_\_\_

Кулачкова М.К.

Преподаватель

\_\_\_\_\_

Жангиров Т.Р.

Санкт-Петербург

2021

## Цель работы

Прогноз успеха фильмов по обзорам.

## Задание

- Ознакомиться с задачей анализа настроений
- Изучить способы представления текста для передачи в ИНС

## Требования

1. Построить и обучить нейронную сеть для обработки текста
2. Исследовать результаты при различном размере вектора представления текста
3. Написать функцию, которая позволяет ввести пользовательский текст (в отчете привести пример работы сети на пользовательском тексте)

## 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  $len(sequences) \times dimension$ , где  $len(sequences)$  – число обзоров в датасете,  $dimension$  – число уникальных

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 функцией активации *sigmoid* для осуществления бинарной классификации. Ниже приведено краткое описание модели.

```
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 [=====] - 5s 24ms/step - loss: 0.5599 -
accuracy: 0.6974 - val_loss: 0.2700 - val_accuracy: 0.8894
Epoch 2/2
80/80 [=====] - 1s 14ms/step - loss: 0.2397 -
accuracy: 0.9086 - val_loss: 0.2588 - val_accuracy: 0.8953
Mean accuracy = 0.8923499882221222

```

Получили точность 89.23%.

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()
    f.close()
    print(txt)
    txt = re.sub(r"[^a-zA-Z0-9]", " ", txt)
    txt = txt.split()
    index = imdb.get_word_index()
    coded = [1]
    coded.extend([index.get(i, 0) for i in txt])
    for i in range(1, len(coded)):
        if coded[i]:
            coded[i] += 3
        if coded[i] >= max_words:
            coded[i] = 2
    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, закодированный обзор (чтобы узнать, какие слова были

распознаны, а какие нет, и просто потому, что мне интересно) и результат предсказания.

Имя файла	Текст обзора	Закодированный обзор	Предсказание сети
example1.txt	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	[1, 2, 357, 21, 776, 65, 8, 794, 6, 171, 163, 388, 11, 50, 17, 73, 156, 26, 608, 21, 5366, 535, 6, 933, 65, 33, 32]	0.29657927 - Обзор отрицательный
example2.txt	Mean Girls grabs the base formula of a teen movie and takes it to another level with clever 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.	[1, 384, 539, 5115, 4, 2810, 2063, 7, 6, 1629, 20, 5, 304, 12, 8, 160, 651, 19, 1096, 640, 35, 1728, 114, 5, 111, 906, 388, 4, 116, 9, 128, 74, 85, 212, 1629, 102, 5, 4, 1114, 6891, 5, 2425, 26, 73, 2142]	0.8510625 - Обзор положительный
example3.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.	[1, 384, 539, 9, 31, 7, 4, 91, 5590, 102, 7, 32, 58, 175, 347, 9, 5590, 5, 9, 2, 32, 4, 58, 17, 6, 415, 37, 8800, 102, 39, 4, 402, 2, 13, 238, 30, 6, 227, 7492, 54, 13, 135, 15, 384, 539, 407, 166, 61, 350, 158, 514, 102, 1029]	0.96799994 - Обзор положительный
example4.txt	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	[1, 13, 296, 88, 12, 16, 23, 248, 23, 6, 2324, 2655, 13, 197, 12, 16, 120, 5, 1194, 4809, 95, 1194, 145, 1784, 234, 303, 12, 16, 131, 395, 1088, 234, 99, 2, 5, 32, 6853, 2]	0.18843451 - Обзор отрицательный

	minutes too ling and all sickening mush.		
--	---	--	--

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 5000 наиболее часто встречающихся слов, векторизуем данные в соответствии с новым ограничением и построим модель, у которой входной слой будет состоять из 5000 нейронов. Обучим модель и оценим ее точность:

```
Epoch 1/2
80/80 [=====] - 1s 11ms/step - loss: 0.5722 -
accuracy: 0.6862 - val_loss: 0.2765 - val_accuracy: 0.8864
Epoch 2/2
80/80 [=====] - 1s 8ms/step - loss: 0.2675 -
accuracy: 0.8946 - val_loss: 0.2631 - val_accuracy: 0.8932
Mean accuracy = 0.8897999823093414
```

Точность модели по сравнению с 10000 слов уменьшилась незначительно.

Теперь сделаем то же самое с 1000 наиболее часто встречающихся слов.

Получим модель с точностью

```
Epoch 1/2
80/80 [=====] - 1s 5ms/step - loss: 0.6301 -
accuracy: 0.6210 - val_loss: 0.3574 - val_accuracy: 0.8456
Epoch 2/2
80/80 [=====] - 0s 4ms/step - loss: 0.3878 -
accuracy: 0.8303 - val_loss: 0.3239 - val_accuracy: 0.8608
Mean accuracy = 0.8532000184059143
```

Точность снизилась сильнее. 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.