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

ОТЧЕТ

по лабораторной работе №7

по дисциплине «Искусственные нейронные сети**»**

Тема: "Классификация обзоров фильмов"

Студент гр. 8383 _____ Степанов В.Д.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург
2021

Цель.

Классификация последовательностей - это проблема прогнозирующего моделирования, когда у вас есть некоторая последовательность входных данных в пространстве или времени, и задача состоит в том, чтобы предсказать категорию для последовательности.

Проблема усложняется тем, что последовательности могут различаться по длине, состоять из очень большого словарного запаса входных символов и могут потребовать от модели изучения долгосрочного контекста или зависимостей между символами во входной последовательности.

В данной лабораторной работе также будет использоваться датасет IMDb, однако обучение будет проводиться с помощью рекуррентной нейронной сети

Выполнения работы.

При помощи методический материалов была построена нейронная сеть, классифицирующая обзоры фильмов по двум категориям.

Модель 1:

```
model = Sequential()
model.add(Embedding(top_words, embedding_vecor_length,
input_length=max_review_length))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))
```

Результат запуска модели 1:

```
Epoch 1/3
391/391 [==============================] - 129s 325ms/step - loss: 0.5883 - accuracy: 0.6795 - val_loss: 0.3230 - val_accuracy: 0.8644
Epoch 2/3
391/391 [==============================] - 136s 348ms/step - loss: 0.2737 - accuracy: 0.8917 - val_loss: 0.3094 - val_accuracy: 0.8692
```

```
Epoch 3/3
391/391 [==============================] - 139s 355ms/step -
loss: 0.2164 - accuracy: 0.9178 - val_loss: 0.2970 - val_accuracy:
0.8791
```

Точность данной модели 1 составила 91.78% на обучающих данных и 87.91% на тестовых. Попробуем расширить модель добавив в нее слои пула и свертки.

Модель 2:

```
model = Sequential()
model.add(Embedding(top_words, embedding_vecor_length,
input_length=max_review_length))
model.add(Conv1D(filters=32, kernel_size=3, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))
```

Результат запуска модели 2:

```
Epoch 1/3
391/391 [==============================] - 72s 180ms/step -
loss: 0.5983 - accuracy: 0.6383 - val_loss: 0.3397 - val_accuracy:
0.8618
Epoch 2/3
391/391 [==============================] - 80s 205ms/step -
loss: 0.2579 - accuracy: 0.9007 - val_loss: 0.2808 - val_accuracy:
0.8844
Epoch 3/3
391/391 [==============================] - 73s 186ms/step -
loss: 0.1953 - accuracy: 0.9282 - val_loss: 0.2948 - val_accuracy:
0.8778
```

Точность данной модели 2 составила 92.82% на обучающих данных и 87.78% на тестовых. Видно, что точность модели 2 практически не изменилась

по сравнению с моделью 1, но увеличилась скорость обучения модели. Добавим в модель Dropout, чтобы решить проблему переобучения.

Модель 3 :

```
model.add(Embedding(top_words, embedding_vecor_length,
input_length=max_review_length))
model.add(Conv1D(filters=32, kernel_size=3, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.2))
model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))
```

Результат запуска модели 3:

```
Epoch 1/3
391/391 [==============================] - 114s 287ms/step -
loss: 0.5709 - accuracy: 0.6735 - val_loss: 0.3208 - val_accuracy:
0.8654
Epoch 2/3
391/391 [==============================] - 105s 270ms/step -
loss: 0.2532 - accuracy: 0.9016 - val_loss: 0.2822 - val_accuracy:
0.8838
Epoch 3/3
391/391 [==============================] - 105s 268ms/step -
loss: 0.2020 - accuracy: 0.9241 - val_loss: 0.2836 - val_accuracy:
0.8838
```

Точность данной модели 3 составила 92.41% на обучающих данных и 88.38% на тестовых. Видно, что точность модели 3 практически не изменилась по сравнению с моделью 2.

Проведем ансамблирование моделей моделей. Всего будет 4 модели с конфигурацией модели 3 и обученные на разных датасетах, состоящих из 6250 данных. Бала написана функция, которая на вход принимает модели и данные для анализа. Функция возвращает среднее прогнозов моделей.

Результаты запуска ансамблирования моделей:

```
Model 1 accuracy: 90.00%
Model 2 accuracy: 88.00%
Model 3 accuracy: 89.00%
Model 4 accuracy: 90.00%
Models  accuracy: 91.00%
```

Видно, что ансамблирование моделей увеличило точность предсказаний. Для ввода пользовательского текста была взята функция из лабораторной №6.

Функция ввода:

```
def convertString(str):
    # удаляем все символы кроме букв, пробельных символов и апострофа
    # приводим к нижнему регистру и разделяем слова чере пробел
    str = re.sub('[^A-Za-z\s\']', '', str).lower().split(' ')
    index = imdb.get_word_index()  # послучаем словарь слов и их индексов
    list = []
    for i in str:
        if i in index.keys():
            list.append(index[i] + 3)  # преобразуем слова в индексы
    return list
```

Тестирование моделей на пользовательском вводе:

Enter your string or enter "exit" to out: *This is the best movie I've ever seen. Great actors, best director, great scenery. In this movie, everything is fine. I would give the highest rating.*

I predict it is **positive** feedback

Enter your string or enter "exit" to out: *This is the worst movie I've ever seen. Terrible acting, unbearably bad scenery, and disgusting cameraman work. I would not recommend the film to friends and acquaintances.*

I predict it is **negative** feedback

Предсказания являются верными. Оценим работу сети на отзыве, который содержит положительный слова, но имеет негативный контекстный и наоборот.

Enter your string or enter "exit" to out: *it is not good film. I'm disappointed*

I predict it is **negative** feedback

Enter your string or enter "exit" to out: *it is good film. I'm not disappointed*

I predict it is **positive** feedback

Видно, сеть правильно классифицирует предложения, состоящие из одинаковых слов, но имеющих разный контекстный смысл.

Вывод.

В ходе лабораторной работы была построена нейронная сеть, классифицирующая отзывы о фильмах. Были изучены рекуррентные нейронные сети и ансамблирование сетей. Сеть была протестирована на пользовательском тексте отзыва.