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

ОТЧЕТ

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

Студент гр. 7381 _____ Кортев Ю. В.

Преподаватель _____ Жукова Н.А.

Санкт-Петербург
2020

Цель работы.

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

Ход работы.

Графики точности и потерь предложенной архитектуры модели показаны на рисунках 1-2.

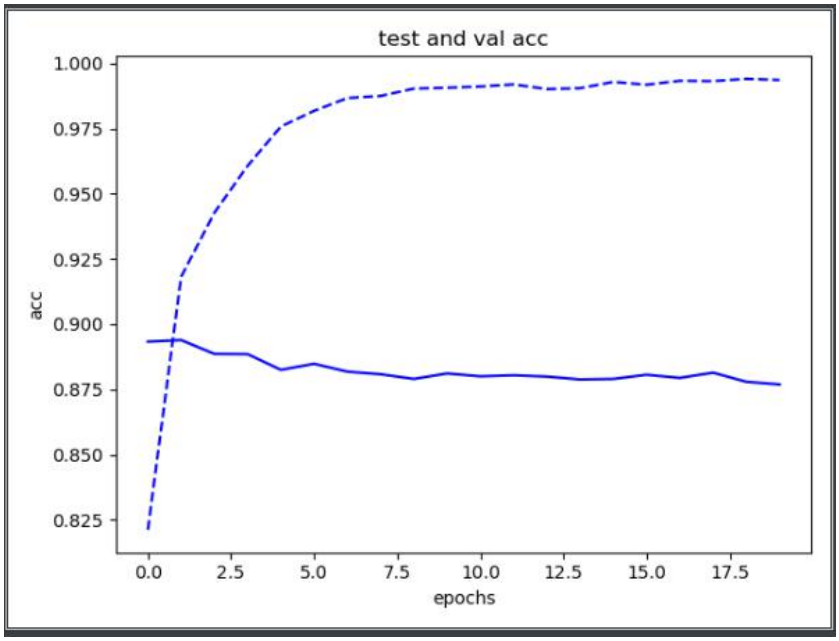

Рисунок 1 - Точность предоженной модели

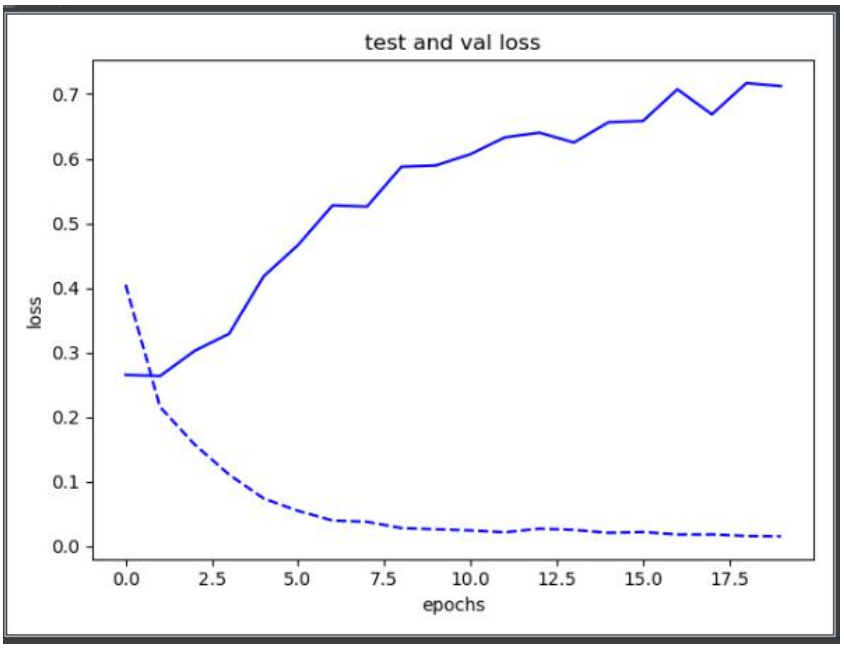

Рисунок 2 - Потери предложенной модели

Как видно переобучение наступает уже после 1й эпохи. Я попытался уменьшить размер модели, но независимо от этого точность достигает максимума в 90 процентов уже послей первой итерации. Инициализация итоговой модели показана в листинге 1. Графики точности и потерь показаны на рисунках 3-4.

```
model=Sequential()
model.add(Dense(16, activation = "relu", input_shape=(10000, )))
model.add(Dropout(0.3))
model.add(Dense(16, activation = "relu"))
model.add(Dense(1, activation = "sigmoid"))

model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['ac
curacy'])

h=model.fit(data[10000:],labels[10000:],
epochs=num_epochs,batch_size=512,validation_data=(data[:10000],labels[:
10000]))
```

Листинг 1 - Инициализация итоговой модели

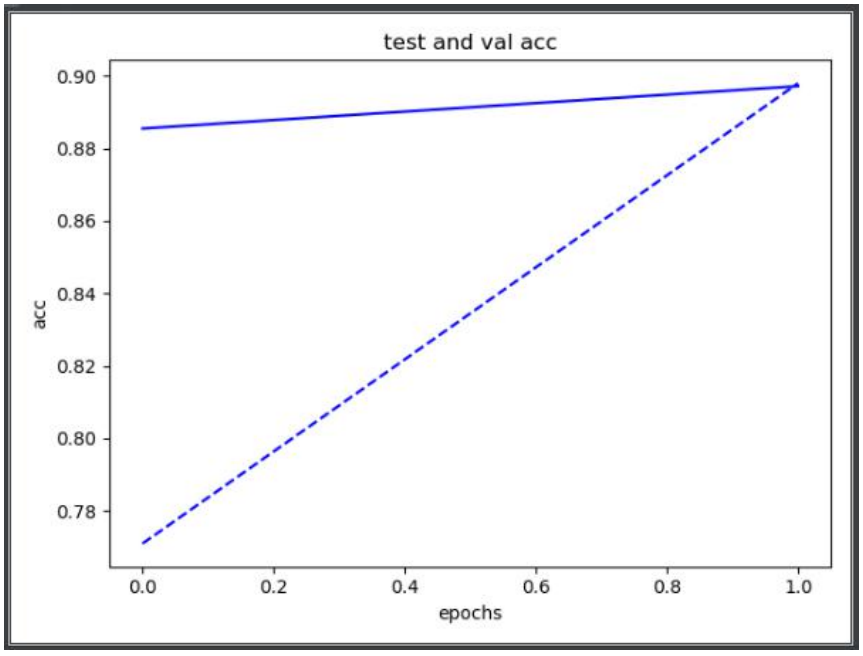

Рисунок 3 - Точность итоговой модели

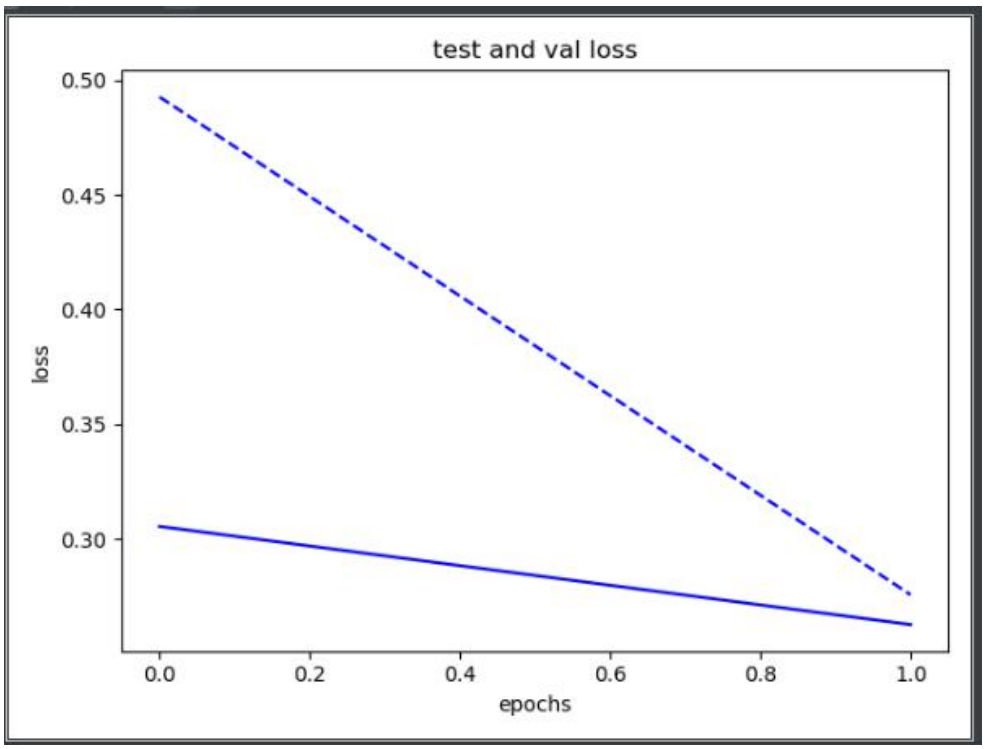

Рисунок 4 - Потери итоговой модели

В итоге модуля модель сохраняется в файл model.h5, модуль prediction_views, для предсказания по пользовательским обзорам загружает модель, получает пути к текстовым файлам с обзорами через аргументы командной строки. Модуль загружает обзоры, сохраняет в список и приводит к виду, который может принять модель. Пример работы модуля показан на рисунке 5.

```
(venv) C:\Users\green\.PyCharm2019.3\config\scratches>python predict_review.py rev0of10.txt rev10of10.txt
Using TensorFlow backend.
I'm very smart neuron network! And i think what...
Film №1 is a horrible movie!!!
Film №2 is a Great movie!
```

Рисунок 5 - Пример работы модуля предсказания пользовательских обзоров

Как видно модель совершенно верно определила, что 1й фильм ужасный, а 2й отличный.

Вывод

В итоге данной лабораторной работы, была получена нейронная сеть предсказывающая успех фильма по обзорам, разработан модуль для обработки пользовательских обзоров.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ ТЕКСТ

```
import numpy as np
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
from keras.layers import Dense, Dropout
from keras.models import Sequential
from matplotlib import pyplot as plt
from keras.utils import to_categorical
from sklearn.preprocessing import LabelEncoder

from keras.datasets import imdb

num_wprds=10000
num_epochs=2

(train_data,train_labels),(test_data,test_labels)=imdb.load_data(num_w
ords=num_wprds)
data=np.concatenate((train_data,test_data),axis=0)
labels=np.concatenate((train_labels,test_labels),axis=0)

def vectorize_sequences(secuences, dimension=num_wprds):
    results=np.zeros((len(secuences),dimension))
    for i, secuence in enumerate(secuences):
        results[i, secuence]=1.
    return results


data=vectorize_sequences(data)
labels=np.asarray(labels).astype('float32')

model=Sequential()
model.add(Dense(16, activation = "relu", input_shape=(10000, )))
model.add(Dropout(0.3))
model.add(Dense(16, activation = "relu"))
model.add(Dense(1, activation = "sigmoid"))

model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['ac
curacy'])

h=model.fit(data[10000:],labels[10000:],
epochs=num_epochs,batch_size=512,validation_data=(data[:10000],labels[:
10000]))


model.save('model.h5')

plt.plot(range(num_epochs),h.history['val_accuracy'],'b-',label='val')
plt.plot(range(num_epochs),h.history['accuracy'],'b--',label='test')
```

```
plt.title('test and val acc')
plt.xlabel('epochs')
plt.ylabel('acc')
plt.show()

plt.plot(range(num_epochs),h.history['val_loss'],'b-',label='val')
plt.plot(range(num_epochs),h.history['loss'],'b--',label='test')
plt.title('test and val loss')
plt.xlabel('epochs')
plt.ylabel('loss')
plt.show()
```

ПРИЛОЖЕНИЕ Б

ИСХОДНЫЙ ТЕКСТ МОДУЛЯ ОБРАБОТКИ ПОЛЬЗОВАТЕЛЬСКИХ ОБЗОРОВ

```
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
from keras.models import load_model
from keras.datasets.imdb import get_word_index
from sys import argv
import numpy as np

from keras.preprocessing.text import text_to_word_sequence

def vectorize_sequences(secuences, dimension=10000):
    results=np.zeros((len(secuences),dimension))
    for i, secuence in enumerate(secuences):
        results[i, secuence]=1.
    return results

def filter(list_of_revs):
    for i in range(len(list_of_revs)):
        j=0
        while j<len(list_of_revs[i]):
            if list_of_revs[i][j] > 10000:
                del list_of_revs[i][j]
            else:
                j+=1

def decode(rev):
    reverse_word_index = dict(
        [(value, key) for (key, value) in word_index.items()])
    decoded_review = ' '.join(
        [reverse_word_index.get(i - 3, '?') for i in rev])
    return decoded_review


reviews=argv[1:]
word_index=get_word_index()

for i, nameOfFile in enumerate(reviews):
    with open(nameOfFile, 'r') as file:
        strings = text_to_word_sequence(file.read())
    for j, word in enumerate(strings):
        if word in word_index:
            #print(word, word_index[word])
            strings[j]=word_index[word]+3
        else:
            strings[j]=2
    reviews[i]=strings
```

```
filter(reviews)

reviews=vectorize_sequences(reviews)

model=load_model('model.h5')

prediction=model.predict_classes(reviews)

print("I'm very smart neuron network! And i think what...")
for i, pr in enumerate(prediction):
    if pr[0]==1:
        print('Film №{} is a Great movie!'.format(i+1))
    else:
        print('Film №{} is a horrible movie!!!'.format(i+1))
```