

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. 7381

Процветкина А.В.

Преподаватель

Жукова Н.А.

Санкт-Петербург

2020

Цель работы.

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

Постановка задачи.

- Ознакомиться с задачей регрессии
- Изучить способы представления текста для передачи в ИНС
- Достигнуть точность прогноза не менее 95%

Теоретические положения.

ИНС не работают с текстом как таковым. Для обучения модели на основе текстовых данных сначала требуется закодировать их. Вместе с датасетом IMDB можно посмотреть на словарь, который и реализует данную идею

Ход работы.

Была построена и обучена НС. Точность модели составила 0.895. Процесс обучения модели показан на рис.1.

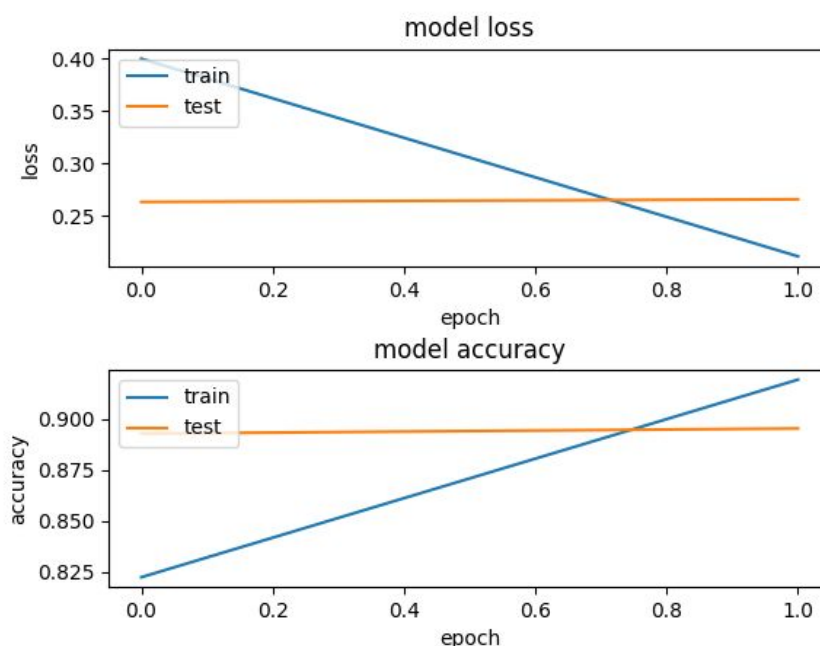


Рисунок 1 -- Процесс обучения модели

Получить большую точность не представляется возможным. Единственное изменение, которого можно добиться, -- в худшую сторону.

Исследуем поведение сети при уменьшении размера вектора представления текста.

При размере вектора равном 900, точность ухудшается и составляет 0.825. Процесс обучения показан на рис.2.

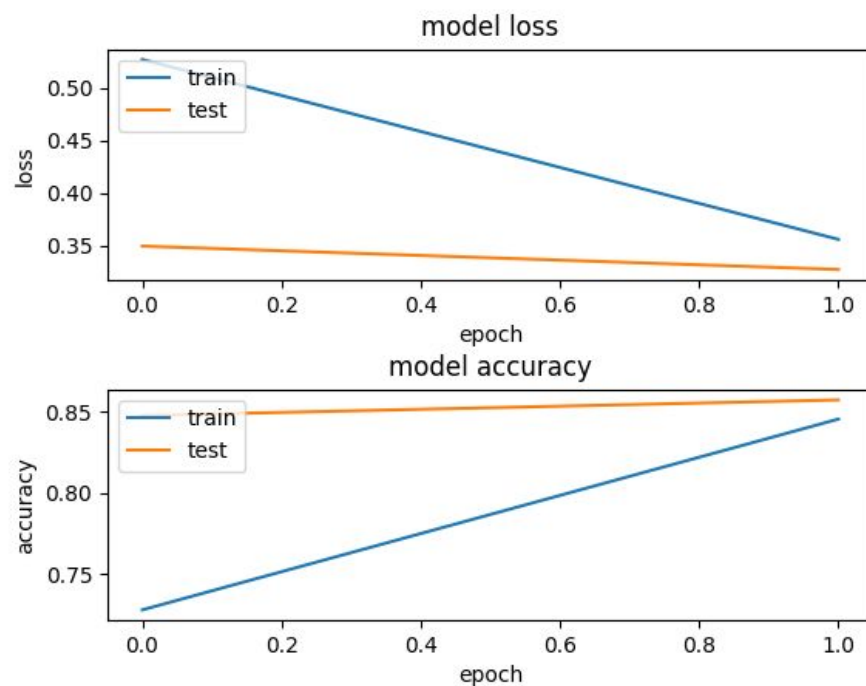


Рисунок 2 -- Процесс обучения модели при размере вектора 900

При размере вектора равном 500, точность ухудшается и составляет 0.821. Процесс обучения показан на рис.3.

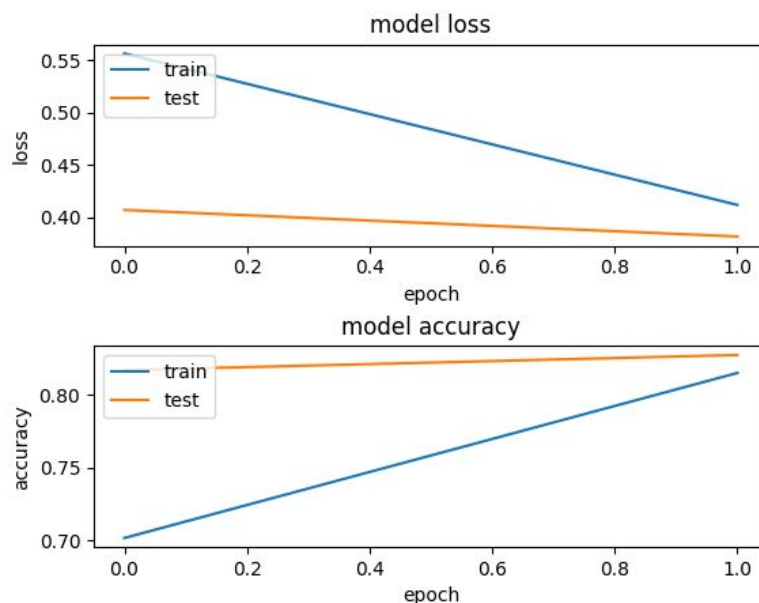


Рисунок 3 -- Процесс обучения модели при размере вектора 500

Увеличивать размер вектора не имеет смысла, поскольку самый длинный отзыв содержит около 1000 слов.

Работа сети была протестирована на двух отзывах, текст которых приведен в приложениях Б и В соответственно. Предсказанный результат показан на рис.

```
Testing positive:
Positive review
Testing negative:
Negative review
```

Рисунок 4 -- Результат предсказаний модели

Выводы.

В ходе лабораторной работы была изучена задача анализа настроений (сентимент-анализ), построена и обучена ИНС на основе датасета IMDB. Модель была проверена на пользовательском отзыве. Точность модели в результате обучения составила 0.895.

Приложение А

Исходный код программы

```
import matplotlib as matplotlib
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
from keras.utils import to_categorical
from keras import models

from keras import layers
from keras.datasets import imdb

N = 1000
(training_data, training_targets), (testing_data, testing_targets) =
imdb.load_data(num_words=N)
data = np.concatenate((training_data, testing_data), axis=0)
targets = np.concatenate((training_targets, testing_targets), axis=0)

def vectorize(sequences, dimension=N):
    results = np.zeros((len(sequences), dimension))
    for i, sequence in enumerate(sequences):
        results[i, sequence] = 1
    return results

data = vectorize(data)

targets = np.array(targets).astype("float32")
test_x = data[:10000]

test_y = targets[:10000]

train_x = data[10000:]

train_y = targets[10000:]
# Input - Layer
model = models.Sequential()
model.add(layers.Dense(50, activation="relu", input_shape=(N, )))

# Hidden - Layers
model.add(layers.Dropout(0.3, noise_shape=None, seed=None))
model.add(layers.Dense(50, activation="relu"))
model.add(layers.Dropout(0.2, noise_shape=None, seed=None))
model.add(layers.Dense(50, activation="relu"))
# Output- Layer
model.add(layers.Dense(1, activation="sigmoid"))
```

```

model.summary()
model.compile(optimizer="adam", loss="binary_crossentropy",
metrics=["accuracy"])
results = model.fit(train_x, train_y, epochs=2, batch_size=500,
validation_data=(test_x, test_y))
print(np.mean(results.history["val_accuracy"]))

plt.subplot(211)
plt.plot(results.history['loss'])
plt.plot(results.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')

plt.subplot(212)
plt.plot(results.history['accuracy'])
plt.plot(results.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

def custom_review(name):
    f = open(name, 'r')
    f_text = f.read()
    index = imdb.get_word_index()
    txt = []
    for i in f_text:
        if i in index and index[i] < 10000:
            txt.append(index[i])

    txt = vectorize([txt])
    return txt

text = custom_review("pos_review.txt")
print("Testing positive:")
result = model.predict_classes(text)
if result == 1:
    print("Positive review")
else:
    print("Negative review")

print("Testing negative:")
text = custom_review("neg_review.txt")
result = model.predict_classes(text)

```

```
if result == 1:  
    print("Positive review")  
else:  
    print("Negative review")
```

Приложение Б

Положительный отзыв

How often do you see a movie like this? It absolutely rocks. Even though it tones down the violence of the dark horse comics, it seamlessly blends Tex Avery style wackiness into the real world. If that doesn't reflect the inner child in us and what everyone wants to be, then what does? Firstly, the main character is brilliant. Hes an explosive combination of all the wacky toons we all love. But the difference is that its all amplified 10 times over and its all real! You actually combine all this wackiness into a very serious, realistic, dark and gritty world. The guy who came up with the 'Tornado' routine needs to be given a major award. This has to be Jim Carrey's best performance in his career and for once, hes not overacting or being annoying. Cameron Diaz looks her best in this movie.

Приложение В

Негативный отзыв

This is hilarious! I cannot believe I have been waiting for this movie so long to just get disappointed. It looks like no one makes good horrors anymore. Why making main characters such idiots? Moreover actors look like they understand they're playing idiots. Wanted to start yawning after an hour of watch. Boring. I regret spending money on a such rubbish.