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»

Студент гр. 8383 Гоголев Е.Е.

Преподаватель Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы

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.

Ход работы

Были выбраны и реализованы три различные архитектуры ИНС для классификации текста. Архитектуры моделей представлены описаны в функциях model_1(), model_2() и model_3() и представлены в приложении А.

Все модели были обучены в цикле и оценены на тестовых данных. У всех моделей схожая точность на тестовом датасете (84 – 87 %), поэтому их можно ансамблировать с помощью среднего арифметического результатов. Далее была написана функция ensemble_predict(), которая выдает результат предсказания ансамбля моделей. Оценки точности отдельных сетей на рис. 1, оценка точности ансамбля на рис. 2.

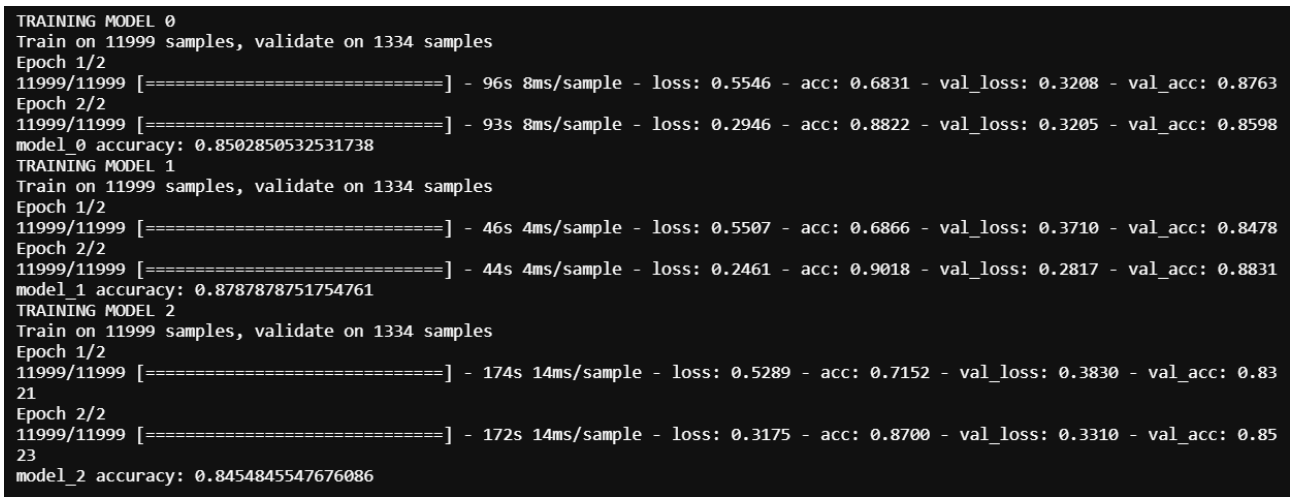

Рисунок 1

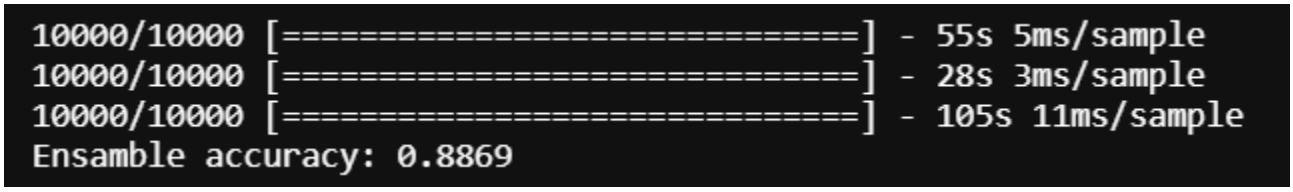

Рисунок 2

Точность ансамбля моделей больше каждой конкретной ОНС. Она составляет 88.69%.

Была написана функция read_from_file(), позволяющая загружать пользовательский текст из файла. Тексты отзывов в приложении Б. На рис. 3 видно, что ансамбль верно классифицировал отзывы.

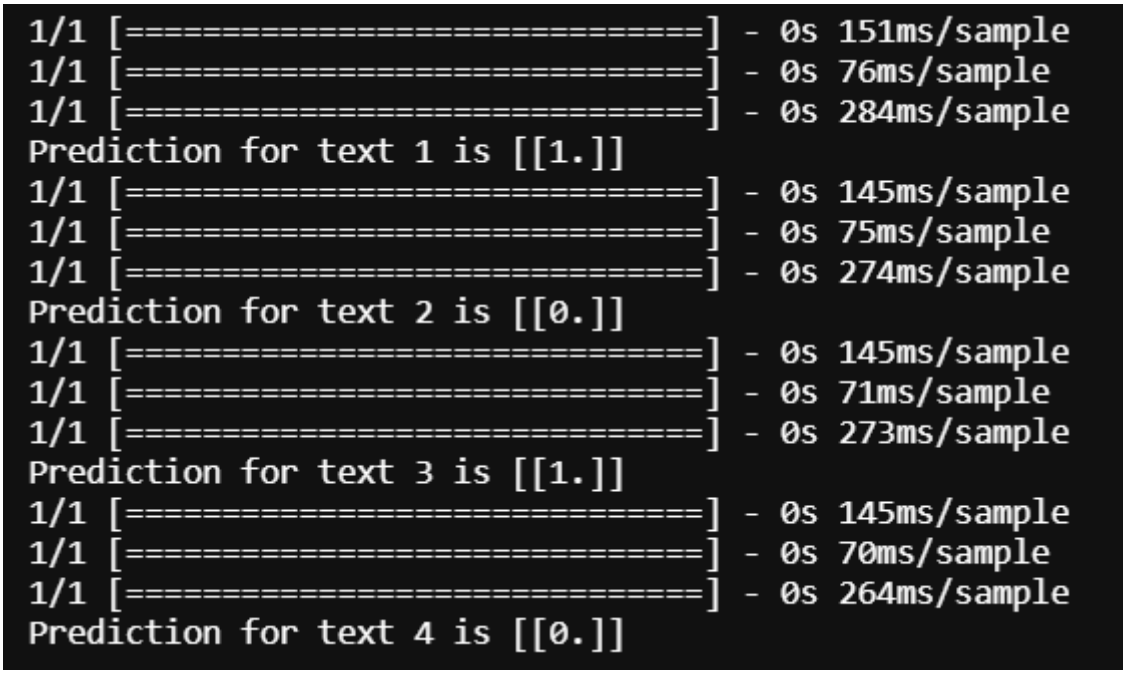

Рисунок 3

Выводы

В ходе выполнения лабораторной работы была создана нейронная сеть для прогнозирования успеха фильма по обзору. Была реализована функция для ввода пользовательского текста. Был изучен один из способов представления текста для передачи в нейронную сеть.

Приложение А. Исходный код программы.

```
import numpy as np
import re
from tensorflow.keras.datasets import imdb
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, LSTM, Conv1D,
MaxPool1D, Flatten, Embedding
from tensorflow.keras.preprocessing import sequence
def read_from_file(filename, max_review_length):
    text = ''
    with open(filename, 'r') as f:
        text = f.read().lower()
        text = (re.sub(r"[^a-zA-Z0-9']", " ", text)).split()
    dictionary = imdb.get_word_index()
    vectorized = []
    for word in text:
        word = dictionary.get(word)
        if word in range(1, 10000):
            vectorized.append(word + 3)
    padded = []
    padded.append(vectorized)
    result              =              sequence.pad_sequences(padded,
maxlen=max_review_length)
    return result
def ensemble_predict(models, x):
    y = []
    for model in models:
```

```
        y.append(model.predict(x, verbose=1))
    result = np.asarray(y)
    return np.round(np.mean(result, 0))

(training_data, training_targets), (testing_data, testing_targets) = imdb.load_data(num_words=10000)

data = np.concatenate((training_data, testing_data), axis=0)

targets = np.concatenate((training_targets, testing_targets), axis=0)


test_size = 10000

X_test = data[:test_size]

Y_test = targets[:test_size]

X_train = data[test_size:]
Y_train = targets[test_size:]

max_review_length = 500

voc_size = 10000

embedding_len = 32

X_train = sequence.pad_sequences(X_train, maxlen=max_review_length)
X_test = sequence.pad_sequences(X_test, maxlen=max_review_length)

def model_1():

    model = Sequential()

    model.add(Embedding(voc_size, embedding_len, input_length=max_review_length))

    model.add(Conv1D(filters=32, kernel_size=3, padding='same', activation='relu'))

    model.add(MaxPool1D(pool_size=2))

    model.add(LSTM(100, dropout=0.3))

    model.add(Dense(1, activation='sigmoid'))
```

```
    return model
def model_2():
    model = Sequential()
    model.add(Embedding(voc_size, embedding_len, input_length=max_review_length))
    model.add(Conv1D(filters=16, kernel_size=3, padding='same', activation='relu'))
    model.add(MaxPool1D(pool_size=2))
    model.add(Dropout(0.25))
    model.add(Conv1D(filters=32, kernel_size=3, padding='same', activation='relu'))
    model.add(MaxPool1D(pool_size=2))
    model.add(LSTM(100))
    model.add(Dense(1, activation='sigmoid'))
    return model
def model_3():
    model = Sequential()
    model.add(Embedding(voc_size, embedding_len, input_length=max_review_length))
    model.add(Dense(64))
    model.add(Dropout(0.25))
    model.add(Dense(128))
    model.add(Dropout(0.5))
    model.add(LSTM(100))
    model.add(Dense(1, activation='sigmoid'))
    return model
models = [model_1(), model_2(), model_3()]
```

```
train_size = len(X_train) // len(models)

test_size = len(X_test) // len(models)

for i, model in enumerate(models):

    x_train = X_train[i * train_size: (i + 1) * train_size]

    y_train = Y_train[i * train_size: (i + 1) * train_size]

    x_test = X_test[i * test_size: (i + 1) * test_size]

    y_test = Y_test[i * test_size: (i + 1) * test_size]

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

    print("TRAINING MODEL {}".format(i))

    model.fit(x_train,  y_train,  validation_split=0.1,  epochs=2,
batch_size=64, verbose=1)

    scores = model.evaluate(x_test, y_test, verbose=0)
    print("model_{} accuracy: {}".format(i, scores[1]))

from sklearn.metrics import accuracy_score

ensemble_prediction = ensemble_predict(models, X_test)

acc = accuracy_score(Y_test, ensemble_prediction)
print("Ensamble accuracy: {}".format(acc))

for i in range(1, 5):

    text = read_from_file(str(i), max_review_length)

    ensemble_prediction = ensemble_predict(models, text)
    print('Prediction    for    text    {}    is    {}'.format(i,
ensemble_prediction))
```

Приложение Б. Пользовательские тексты.

Текст 1:

Its somewhat ironic that a movie about time travel can't be reviewed properly until your future self rewatches the movie.

It's bold of Nolan to make such a thoroughly dense blockbuster. He assumes people will actually want to see Tenet more than once so they can understand it properly, which some may not. This movie makes the chronology of Inception look as simplistic as tic-tac-toe.

Ergo, it's hard for me to give an accurate rating, without having seen it twice, as I'm still trying to figure out whether everything does indeed make sense. If it does, this movie is easily a 9 or 10. If it doesn't, it's a 6.

It's further not helped by the fact that the dialogue in the first 15 minutes of the movie is painfully hard to understand / hear. Either they were behind masks; they were practically mumbling; the sound effects were too loud; or all of the above. The exposition scenes are also waayyy too brief for something this complex — a problem also shared with Interstellar actually.

(Interstellar had this minimalist exposition problem explaining Blight, where if you weren't careful, you'd miss this one sentence / scene in the entire movie explaining that Blight was a viral bacteria:

"Earth's atmosphere is 80% nitrogen, we don't even breathe nitrogen. Blight does, and as it thrives, our air gets less and less oxygen").

I guess it's a Nolan quirk. Hopefully, a revision of the film audio sorts the sound mixing out. I do like the soundtrack, but it's too loud initially.

I liked all the actors. You think John Washington can't act at first, but he can, and he grows on you as the film progresses. And Pattinson is his usual charming self. Elizabeth is a surprise treat. And so on.

Its worth a watch either way. See it with subtitles if you can. And definitely don't expect to fully understand whats going on the first time around.

Текст 2:

A man on an international mission to save the world from the deadliest weapon of all, the future.
Two moods: excessive incomprehensible exposition and LOUD incomprehensible action sequences. At no point do you know what is going on, nor are you given any reason to care. It is at all times tedious, meaningless and irritating. None of the characters are remotely interesting, much of the

dialogue is inaudible and the ridiculous convolutions add up to nothing. And this cost over $200 million to make.

Текст 3:

I can't put into words how close to home this movie hit for me. Not to sound pretentious, but it's more than a movie - it's a life lesson on how to live. It'll teach you that life isn't about careers, goals, passions, or achievements. It's about living, right here in this moment, exactly where you are.
The voice acting, animation, soundtrack, writing - all phenomenal. It is officially the greatest Pixar movie ever created in my book. Kids will enjoy it, but this one is for the older crowd. 10/10

Текст 4:

People are grossly overrating this movie. It's pretty boring for the first hour. Not even close to as entertaining as other Pixar movies. I don't think I laughed out loud one time during the whole movie. It's watchable, but I wouldn't rewatch it ever again.