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

ОТЧЕТ

по лабораторной работе №6

по дисциплине «Искусственные нейронные сети»

Тема: Прогноз успеха фильмов по обзорам

Студентка гр. 8382 _____ Рочева А.К.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург
2021

Цель работы.

Прогноз успеха фильмов по обзорам.

Постановка задачи.

1. Ознакомиться с задачей регрессии
2. Изучить способы представления текста для передачи в ИНС
3. Достигнуть точность прогноза не менее 95%.

Требования к выполнению задания.

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).

Ход работы.

Для работы были использованы данные из датасета IMDb. Далее была произведена обработка данных:

```
   (training_data, training_targets), (testing_data, testing_targets) = imdb.load_data(num_words=500)
   data = np.concatenate((training_data, testing_data), axis=0)
   targets = np.concatenate((training_targets, testing_targets), axis=0)
   index = imdb.get_word_index()
   reverse_index = dict([(value, key) for (key, value) in index.items()])
   decoded = " ".join( [reverse_index.get(i - 3, "#") for i in data[0]] )
```

Архитектура построенной нейронной сети:

```
model = Sequential()
# Input - Layer
```

```
model.add(Dense(50, activation="relu", input_shape=(10000,)))
# Hidden - Layers
model.add(Dropout(0.2, noise_shape=None, seed=None))
model.add(Dense(50,                              activation="linear",
kernel_regularizer=regularizers.l2()))
model.add(Dropout(0.5, noise_shape=None, seed=None))
model.add(Dense(100,                               activation="relu",
kernel_regularizer=regularizers.l2()))
model.add(Dropout(0.5, noise_shape=None, seed=None))
model.add(Dense(50, activation="relu"))
model.add(Dense(1, activation="sigmoid"))
model.compile(Adam(),                    loss='binary_crossentropy',
metrics=['accuracy'])
history = model.fit(train_x, train_y, batch_size=500, epochs=10,
verbose=1, validation_data=(test_x, test_y))
```

Точность на тренировочной выборке – 0.848, на контрольной – 0.83. График точности показан на рис. 1, график ошибок – на рис. 2.

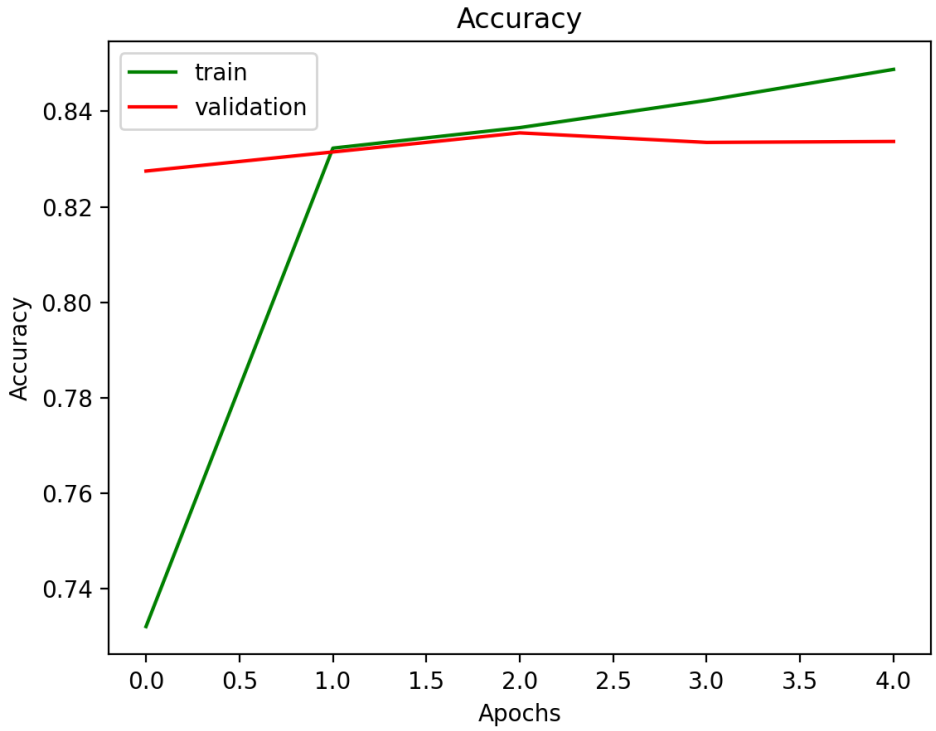

Рис. 1 – график точности при размере словаря в 10 000 обзоров

Потери на тренировочной выборке - 0.37, на контрольной - 0.39.

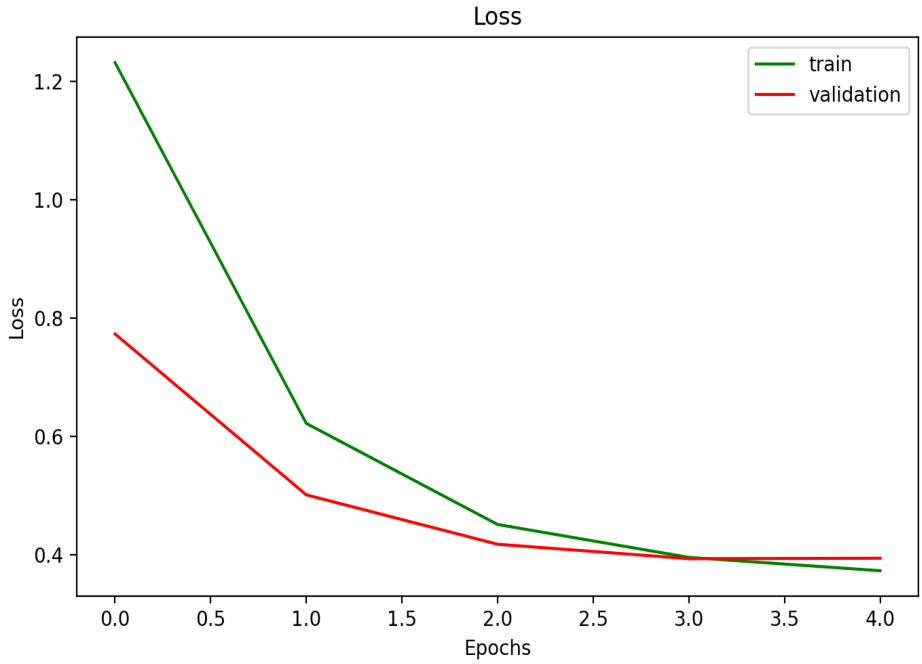

Рис. 2 – график потерь при размере словаря в 10 000 обзоров

Исследуем результаты при различном размере вектора представлении текста. Графики представлены на рис. 3 и 4.

Точность на тренировочной выборке – 0.85, на контрольной – 0.84.

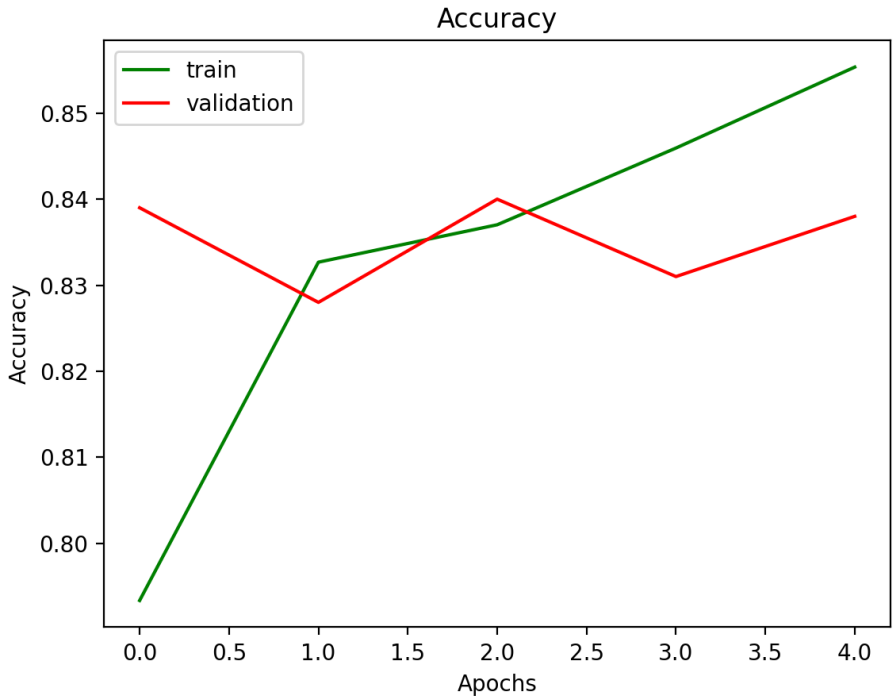

Рис. 3 - график точности при размере словаря в 1000 обзоров

Потери на тренировочной выборке - 0.36, на контрольной - 0.38.

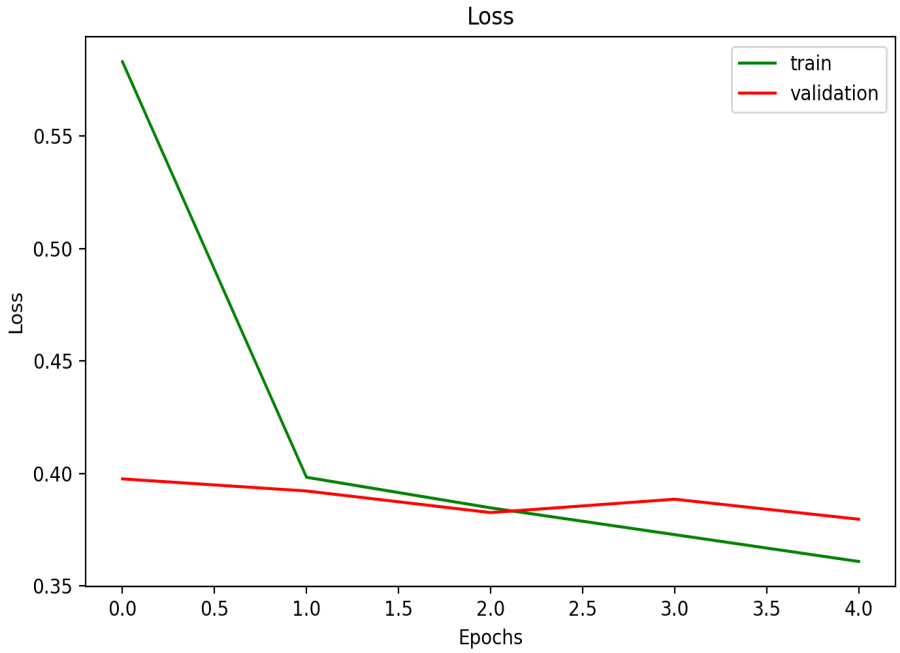

Рис. 4 – график потерь при размере словаря в 10 000 обзоров

Напишем функцию, которая позволяет ввести пользовательский текст.

```
def get_text(custom_x, word_index):
    def get_index(a, index):
        new_list = a.split()
        for i, v in enumerate(new_list):
            new_list[i] = index.get(v)
        return new_list
    for i in range(len(custom_x)):
        custom_x[i] = get_index(custom_x[i], word_index)
    return custom_x
```

При помощи данной функции можно получить из массива строк (обзоров) массив представлений в виде индексов слов в imdb датасете и подготовленные для прогона через модель. График точности оценки фильма, при прогоне через написанный датасет из 5 обзоров, предоставлена на рис 5.

Написанный датасет: "It is bad, i hate it", "It's too boring and awful", "It's amazing, fantastic and exiting", "Fine film, i love it too much", "Really good, fantastic actors, i like".

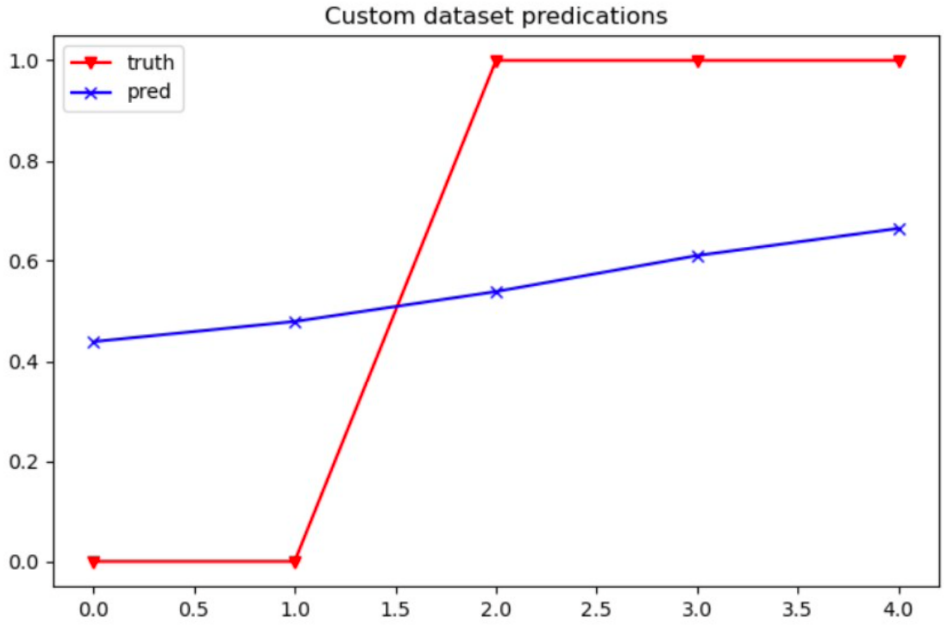

Рис. 8 – график точности оценки фильма

Точность оценки фильма равна 0.61.

Выводы.

В ходе работы была изучена задача классификация обзоров из датасета IMDB. Подобрана архитектура, дающая точность 83%. Функция для подготовки вручную введенных обзоров, продемонстрировала точность в ~65%