

**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 успехов фильма по обзорам»**

Студент гр. 7382

\_\_\_\_\_

Находько А.Ю.

Преподаватель

\_\_\_\_\_

Жукова Н.В.

Санкт-Петербург

2020

### **Цель работы.**

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

### **Задачи работы.**

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

### **Требования работы.**

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)

### **Основные теоретические положения.**

Искусственные нейронные сети - совокупность моделей, которые представляют собой сеть элементов - искусственных нейронов, связанных между собой синаптическими соединениями.

Нейронные сети используются как среда, в которой осуществляется адаптивная настройка параметров дискриминантных функций. Настройка происходит при последовательном предъявлении обучающих выборок образов из разных классов. Обучение - такой выбор параметров нейронной сети, при котором сеть лучше всего справляется с поставленной проблемой.

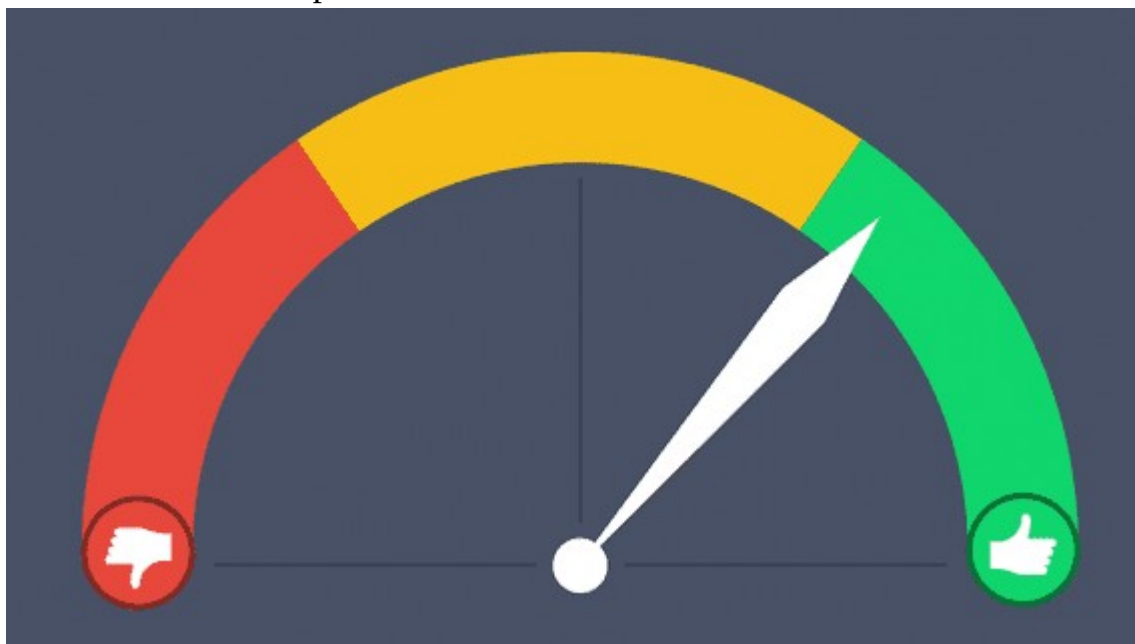
**Нейрон** — элемент, преобразующий входной сигнал по функции.

**Сумматор** — элемент, осуществляющий суммирование сигналов поступающих на его вход.

**Синапс** — элемент, осуществляющий линейную передачу сигнала.

Что такое анализ настроений (сентимент-анализ)?

С помощью анализа настроений можно определить отношение (например, настроение) человека к тексту, взаимодействию или событию. Поэтому sentiment-анализ относится к области обработки естественного языка, в которой смысл текста должен быть расшифрован для извлечения из него тональности и настроений.



Спектр настроений обычно подразделяется на положительные, отрицательные и нейтральные категории. С использованием анализа настроений можно, например, прогнозировать мнение клиентов и их отношение к продукту на основе написанных ими обзоров. Поэтому анализ настроений широко применяется к обзорам, опросам, текстам и многому другому.

### **Экспериментальные результаты.**

В ходе выполнения лабораторной работы была построена ИНС, код которой представлен в Приложении А.

Работа ИНС была протестирована на собственном примере:

Very good movie, amazing play of actors, great scenery and soundtracks deserve separate praise

Полученный результат — 0.55, что соответствует хорошему отзыву.

Произвёл исследование результатов при различном размере вектора представления текста — исследованные размеры: 10000, 1000, 100.

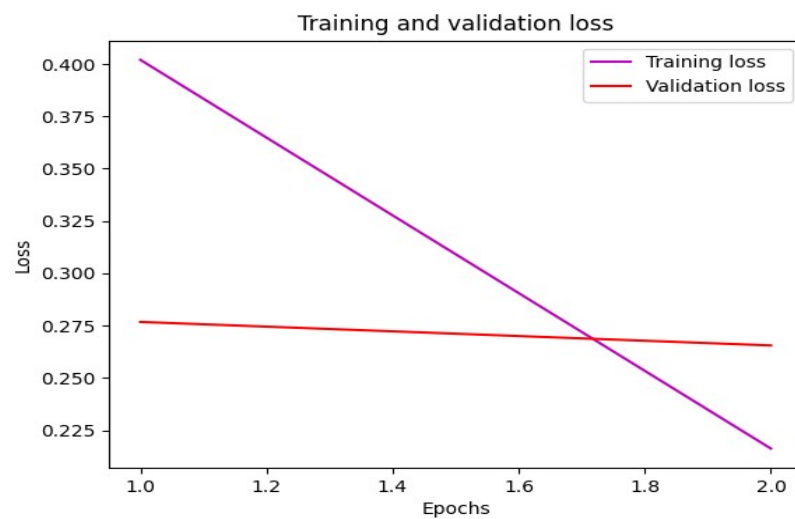


Рисунок 1 — график потерь при размере вектора 10000

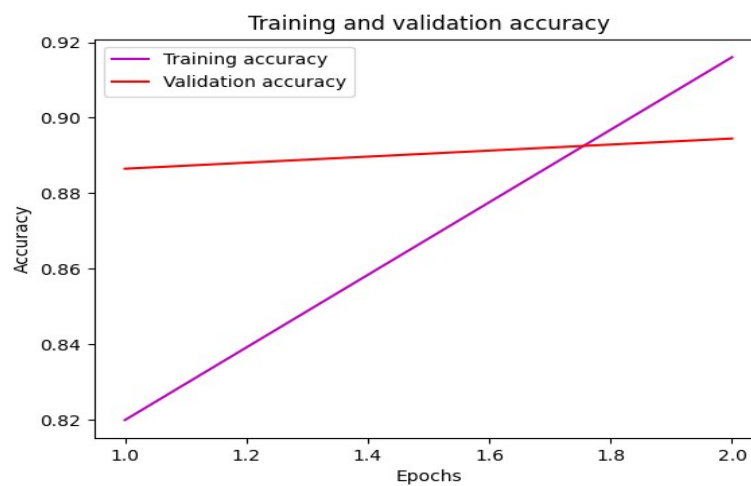


Рисунок 2 — график точности при размере вектора 10000

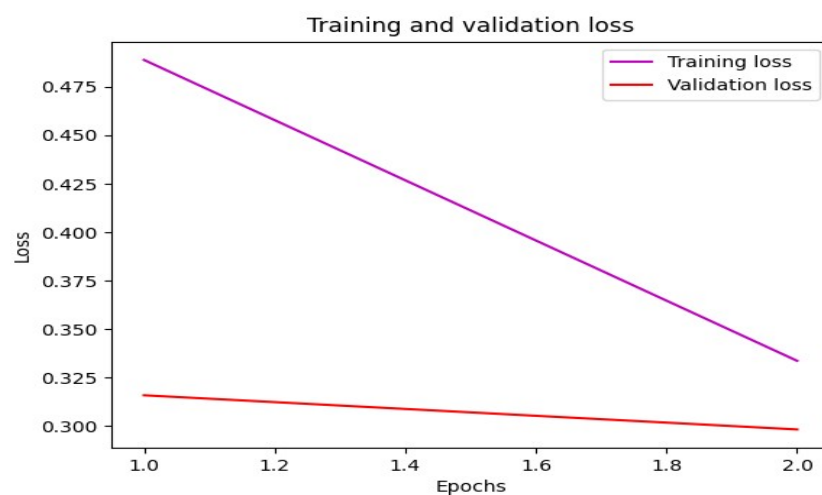


Рисунок 3 — график потерь при размере вектора 1000

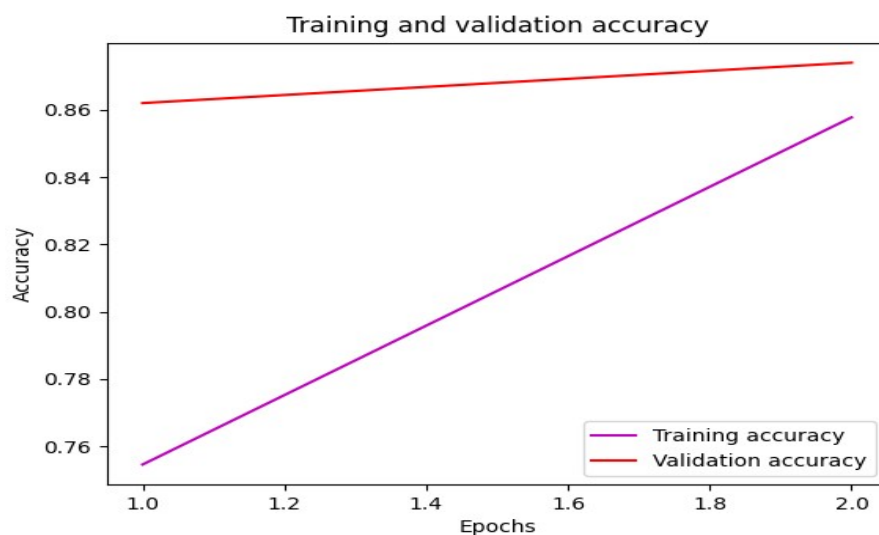


Рисунок 4 — график точности при размере вектора 1000

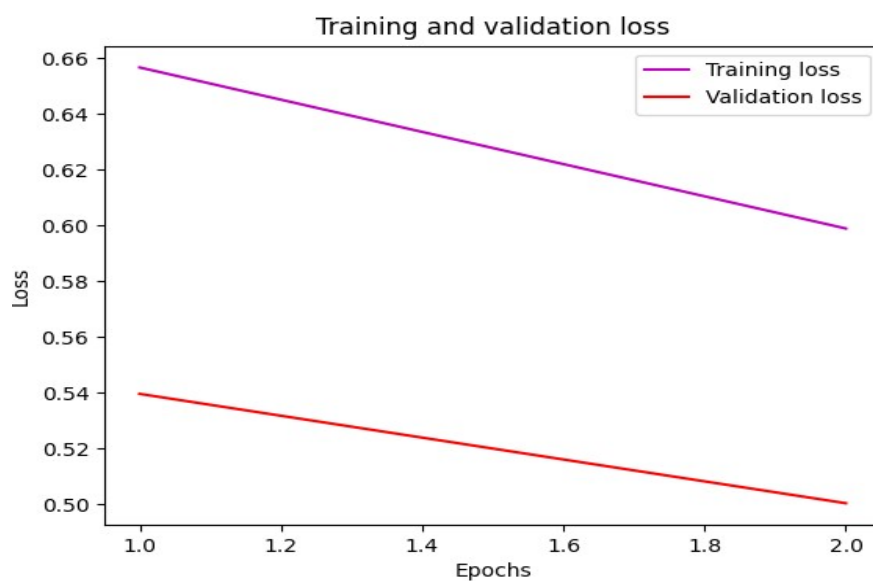


Рисунок 5 — график потерь при размере вектора 100

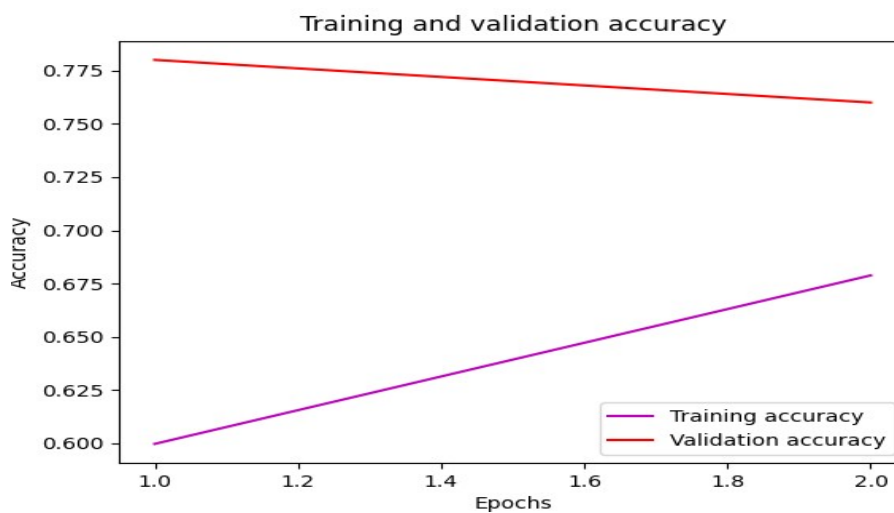


Рисунок 6 — график точности при размере вектора 100

Наилучшей точности ИНС добивается при размере 10000 образов, в этом случае точность составила 89%, в то время как при 1000 образах — 86%, а при 100 — 0.755. Исходя из полученных результатов, можем утверждать, что с уменьшением размера вектора — уменьшается точность и увеличивается ошибка, связано это с уменьшением в обзоре количества слов, характеризующие положительное или отрицательное отношение.

### **Выводы.**

В ходе выполнения лабораторной работы была реализована программа прогнозирующая успех фильмов. Нейронная сеть для обработки текста была корректно построена и обучена. Также ознакомился с задачей регрессии, изучил способы представления текста для передачи в ИНС, была достигнута точность прогноза 95%. Была написана функция, которая позволяет ввести пользовательский текст.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД

```
import matplotlib.pyplot as plt
import numpy as np
from keras import Sequential
from keras import layers
from keras.utils import to_categorical
from keras import models
from keras.datasets import imdb

dmns = 10000

review = ["Very good movie, amazing play of actors, great scenery and
soundtracks deserve separate praise"]

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

def user_func(review):
    index = imdb.get_word_index()
    test_x = []
    words = []
    for line in review:
        lines = line.translate(str.maketrans('', '', ',.? !:;()')).lower()
        for chars in lines:
            chars = index.get(chars)
            if chars is not None and chars < 10000:
                words.append(chars)
        test_x.append(words)
    test_x = vectorize(test_x)
    model = no_user_func()
    prediction = model.predict(test_x)
    print(prediction)
```

```

def no_user_func():
    (training_data, training_targets), (testing_data, testing_targets) =
    imdb.load_data(num_words=dmns)
    data = np.concatenate((training_data, testing_data), axis=0)
    targets = np.concatenate((training_targets, testing_targets), axis=0)

    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:]

    model = Sequential()
    # Input - Layer
    model.add(layers.Dense(50, activation = "relu", input_shape=(dmns, )))
    # 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.compile( optimizer = "adam", loss = "binary_crossentropy", metrics
    = ["accuracy"])

    history = model.fit(train_x, train_y, epochs= 2, batch_size = 500,
    validation_data = (test_x, test_y))

    loss = history.history['loss']
    val_loss = history.history['val_loss']
    epochs = range(1, len(loss) + 1)
    plt.plot(epochs, loss, 'm', label='Training loss')
    plt.plot(epochs, val_loss, 'r', label='Validation loss')
    plt.title('Training and validation loss')

```



```
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
plt.clf()
```

```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
plt.plot(epochs, acc, 'm', label='Training accuracy')
plt.plot(epochs, val_acc, 'r', label='Validation accuracy')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```

```
results = model.evaluate(test_x, test_y)
print(results)
return model
```

```
user_func(review)
```