

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»
Тема: Классификация обзоров фильмов

Студент гр. 8383

Мирсков А. А.

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.

Выполнение работы.

Для решения поставленной задачи были реализованы следующие нейросети в соответствии с методическими указаниями:

```
model = Sequential()
model.add(Embedding(top_words, embedding_vector_length,
input_length=max_review_length))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
```

```
model = Sequential()
model.add(Embedding(top_words, embedding_vector_length,
input_length=max_review_length))
model.add(Conv1D(filters=32, kernel_size=3, padding='same',
activation='relu'))
model.add(MaxPooling1D(pool_size=2))
```

```
model.add(Dropout(0.5))
model.add(LSTM(100))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
```

Далее было проведено ансамблирование моделей. Для этого были написаны функции:

```
def ensemble_predict(models, X):
    return np.round(sum([one_model.predict(X) for one_model in
models])/len(models)).flatten()
```

```
def ensemble_evaluate(models, X, Y):
    count_good = np.count_nonzero(ensemble_predict(models, X) == Y)
    return count_good / len(Y)
```

функция `ensemble_evaluate` возвращает долю правильных ответов ансамбля.

Первая модель показала точность при валидации 0.858, вторая модель 0.9, а ансамбль показал точность 0.902. Точность ансамбля немного превышает точности обеих нейросетей.

Далее была написана функция для обработки пользовательского текста.

```
def user_text(text):
    for i in punctuation:
        text = text.replace(i, ' ')
    text = text.lower().split()
    index = imdb.get_word_index()
    coded_text = [index.get(i)+3 for i in text]
    return sequence.pad_sequences([coded_text], maxlen=max_review_length)
```

Пример работы пользовательского ввода.

Данный отзыв программа классифицировала как положительный: The film is very good. I will definitely watch the second part.

Данный отзыв программа классифицировала как отрицательный: The film is very bad. I do not recomend you to watch.

Выводы.

В ходе выполнения лабораторной работы была написана программа для обучения нейросети для классификации обзоров фильмов. Были изучены способы ансамблирования нейросетей.