

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»

Студентка гр. 7381

Кревчик А.Б.

Преподаватель

Жукова Н.А.

Санкт-Петербург

2020

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.

Постановка задачи.

- Ознакомиться с рекуррентными нейронными сетями
- Изучить способы классификации текста
- Ознакомиться с ансамблированием сетей
- Построить ансамбль сетей, который позволит получать точность не менее 97%

Требования.

1. Найти набор оптимальных ИНС для классификации текста
2. Провести ансамблирование моделей
3. Написать функцию/функции, которые позволят загружать текст и получать результат ансамбля сетей
4. Провести тестирование сетей на своих текстах (привести в отчете)

Ход работы.

1. Были выбраны две модели нейронной сети.

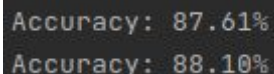
Первая модель:

```
model = Sequential()
model.add(Embedding(top_words, embedding_vecor_length,
input_length=max_review_length))
model.add(Dropout(0.2))
model.add(LSTM(100))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
```

Вторая модель:

```
model = Sequential()
model.add(Embedding(top_words, embedding_vecor_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(0.2))
model.add(LSTM(100))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
```

2. Так как обе сети одинаково хороши и имеют почти одинаковую точность, было решено в качестве объединения прогнозов использовать среднее по прогнозам.



Accuracy: 87.61%
Accuracy: 88.10%


Рисунок 1 - Точность первой и второй сети соответственно

3. Было написано несколько функций, которые позволяют загружать текст и получать результат ансамбля сетей.

Функция `test_text(filename)` - принимает имя файла с текстом и векторизует этот текст.

Функция `final_pred(model_1,model_2,text)` - принимает 2 модели и вектор с текстом. Далее получает оценку для этого по каждой модели и вычисляет среднее. Таким образом, мы имеем оценку по обеим моделям.

4. Проведено тестирование файла “text_good.txt”, в котором содержится положительный отзыв. Классификация произведена верно.



```
text_good.txt
Отзыв хороший
```

Рисунок 2 - Тестирование на своем тексте

Выводы.

В ходе данной лабораторной работы получен навык ансамблирования моделей. Также была написана функция, позволяющая проводить тестирование на собственном тексте.