

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. 7381

_____ Габов Е.С.

Преподаватель

_____ Жукова Н.А.

Санкт-Петербург

2020

Цель

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.

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) Построены и обучены две модели ИНС для решения задачи определения настроения обзора.

Первая модель — рекуррентная с добавлением полносвязных слоев и слоя разреживания, код представлен на рис. 1.

```
def build_model_1():
    model = models.Sequential()
    model.add(layers.Embedding(top_words, embedding_vector_length, input_length=max_review_length))
    model.add(layers.Dropout(0.3))
    model.add(layers.LSTM(100))
    model.add(layers.Dropout(0.3))
    model.add(layers.Dense(1, activation="sigmoid"))
    model.compile(loss="binary_crossentropy", optimizer="adam", metrics=["accuracy"])
    return model
```

Рисунок 1 — Модель рекуррентная с добавлением полносвязных слоев и слоя разреживания.

Вторая модель — рекуррентная с добавлением слерточного слоя и слоя пуллинга, код представлен на рис. 2.

```
def build_model_2():
    model = models.Sequential()
    model.add(layers.Embedding(top_words, embedding_vector_length, input_length=max_review_length))
    model.add(layers.Conv1D(filters=32, kernel_size=3, padding='same', activation='relu'))
    model.add(layers.MaxPooling1D(pool_length=2))
    model.add(layers.LSTM(100))
    model.add(layers.Dense(1, activation="sigmoid"))
    model.compile(loss="binary_crossentropy", optimizer="adam", metrics=["accuracy"])
    return model
```

Рисунок 2 — Модель рекуррентная с добавлением полносвязных слоев и слоя разреживания.

2) Проведено ансамблирование моделей — средне арифмитическое предсказаний моделей. Результат составил 0.88324329

3) Функция для получения результата ансамбля сетей представлена на рис. 3

```
def predict_for_text(text):
    results = []
    for model in models:
        model.fit(train_x, train_y, epochs=2, batch_size=64, validation_data=(test_x, test_y),)
        results.append(model.predict(text))
    print(sum(results) / 2)
```

Рисунок 3 - Получения результата ансамбля сетей

Функция для загрузки текста представлена на рис. 4

```

def textPreparing(func):
    def FWrapper(*args, **kwargs):
        with open(args[0]) as f:
            text = f.read().lower()
            text = text_to_word_sequence(text)
        indexes = imdb.get_word_index()
        text_indexes = []
        for el in text:
            if el in indexes and indexes[el] < 10000:
                text_indexes.append(indexes[el])
        text_indexes = sequence.pad_sequences([text_indexes], maxlen=max_review_length)
        return func(text_indexes, **kwargs)
    return FWrapper

```

Рисунок 4 — Функция для загрузки текста

Вывод:

В результате выполнения лабораторной работы. Проведено ансамблирование моделей, в его результате результаты оказались выше чем в лабораторной работе номер 6. Также реализована функции, которые позволяют загружать текст и получать результат ансамбля сетей.