

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. 8382

Гордиенко А.М.

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.

Порядок выполнения работы.

- 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)

Основные теоретические положения.

Мы отобразим каждый обзор фильма в реальную векторную область, популярную технику при работе с текстом, которая называется встраивание слов. Это метод, в котором слова кодируются как действительные векторы в

многомерном пространстве, где сходство между словами в смысле смысла приводит к близости в векторном пространстве.

Keras предоставляет удобный способ преобразования положительных целочисленных представлений слов в вложение слов слоем Embedding.

Мы сопоставим каждое слово с вещественным вектором длиной 32. Мы также ограничим общее количество слов, которые нам интересны в моделировании, до 5000 наиболее часто встречающихся слов и обнуляем остальные. Наконец, длина последовательности (количество слов) в каждом обзоре варьируется, поэтому мы ограничим каждый обзор 500 словами, укорачивая длинные обзоры и дополняя короткие обзоры нулевыми значениями.

Теперь, когда мы определили нашу проблему и то, как данные будут подготовлены и смоделированы, мы готовы разработать модель LSTM.

Архитектура сети.

Первый слой - это Embedded, который использует 32 вектора длины для представления каждого слова. Следующий уровень - это слой LSTM с 100 единицами памяти (умными нейронами). Наконец, поскольку это проблема классификации, мы используем плотный выходной слой с одним нейроном и сигмоидной функцией активации, чтобы сделать 0 или 1 прогноз для двух классов (хорошего и плохого) в задаче.

Поскольку это проблема двоичной классификации, в качестве функции потерь используется журнал потерь (binary_crossentropy в Keras). Используется эффективный алгоритм оптимизации ADAM. Модель подходит только для 2 эпох, потому что она быстро решает проблему. Большой пакет из 64 обзоров используется для разметки обновлений веса.

LSTM и сверточная нейронная сеть для классификации последовательностей

Сверточные нейронные сети превосходно изучают пространственную структуру во входных данных.

Данные обзора IMDB действительно имеют одномерную пространственную структуру в последовательности слов в обзорах, и CNN может быть в состоянии выбрать инвариантные особенности для хорошего и плохого настроения. Эти изученные пространственные особенности могут затем быть изучены как последовательности уровнем LSTM.

Мы можем легко добавить одномерный слой CNN и максимальный пул после слоя Embedding, которые затем передают объединенные элементы в LSTM. Мы можем использовать небольшой набор из 32 объектов с небольшой длиной фильтра 3. Слой пула может использовать стандартную длину 2, чтобы вдвое уменьшить размер карты объектов.

Мы достигаем результатов, аналогичных первому примеру, но с меньшими весами и меньшим временем обучения.

Рекуррентные нейронные сети, такие как LSTM, обычно имеют проблему переобучения. Вам необходимо решить эту проблему путем добавления в архитектуру сети слоев Dropout.

Ход работы.

1. Были выбраны две модели нейронной сети.

Первая модель:

```
model = Sequential()
model.add(Embedding(top_words,
                    embedding_vector_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:

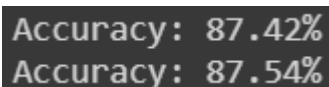
```

model = Sequential()
model.add(Embedding(top_words,
                    embedding_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(0.2))
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 имеют хорошую точность по отдельности, было решено в качестве объединения прогнозов использовать среднее по прогнозам.

Оценка работы сетей представлены на рис. 1.

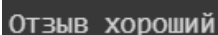


Accuracy: 87.42%
Accuracy: 87.54%

Рисунок 1 – Оценка точности первой и второй моделей сети.

3 . Была написана функция, которая принимает две модели сетей и вектор пользовательского текста.

4 . Проведено тестирование с пользовательским текстом, в котором содержится положительный отзыв. Классификация проведена верно.



Отзыв хороший

Рисунок 2 – Тестирование на пользовательском вводе.

Выводы.

В данной работе было продолжено изучение анализа настроений. В этот раз задача решалась с помощью рекуррентных сетей. Был создан ансамбль из

нескольких моделей, корректно отработавший с пользовательским текстом, для более объективной оценки настроения отзыва к фильму.