**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 Мирончик П.Д.

Преподаватель Жангиров Т.Р.

Санкт-Петербург
2021

## 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

Для выполнения работы используется датасет imdb из 10000 наиболее популярных слов. Длина отзывов обрезается либо дополняется нулями таким образом, чтобы составлять ровно 500 слов.

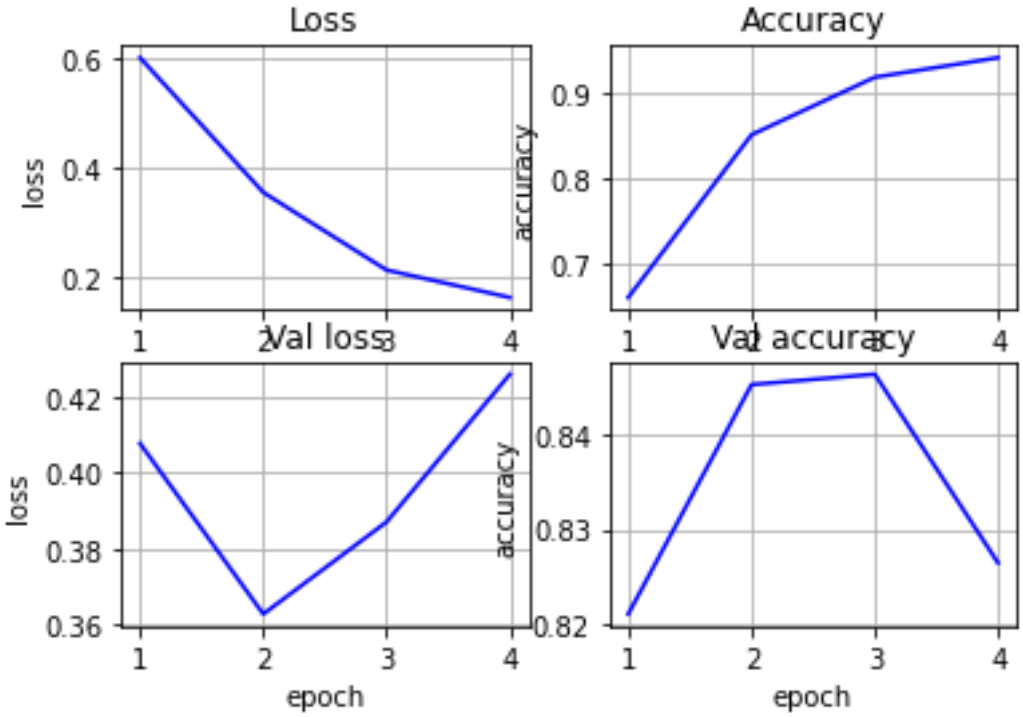
Для обучения используются также данные imdb сформированные таким образом, что соотношение данных на обучение и валидационных данных определяется как 80/20.

Сформируем модель:

```
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'])
```

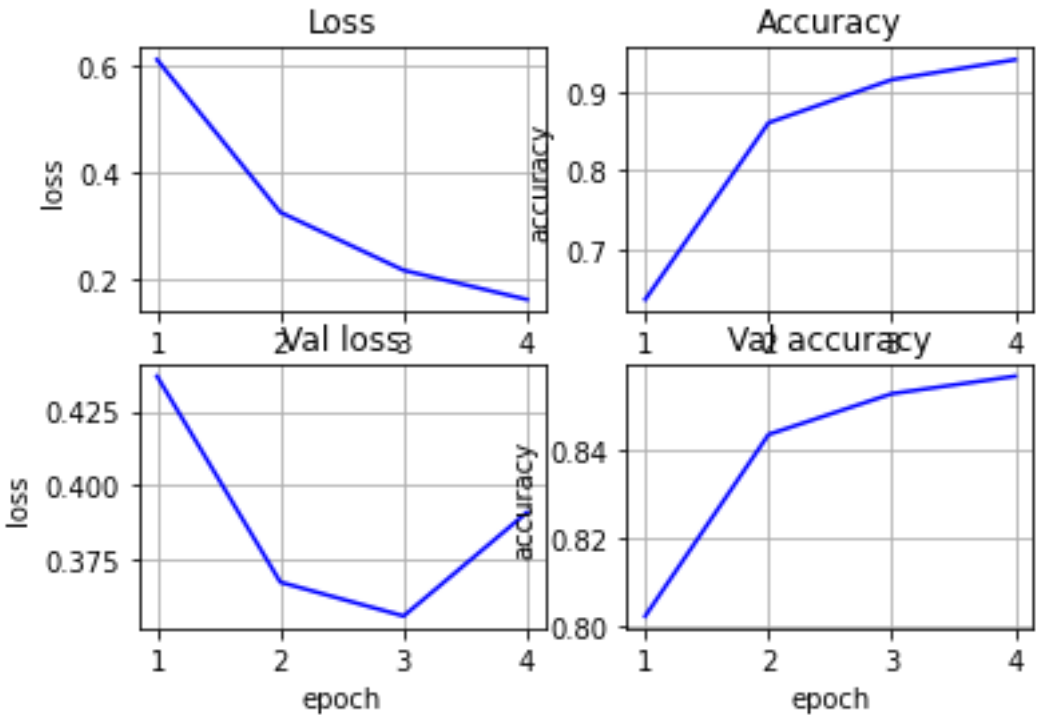
Результат обучения:

```
Accuracy: 82.64%
```




Заметно, что уже к 3 эпохе модель начинает переобучаться. Попробуем добавить слои Dropout для исправления этой проблемы:

```
Accuracy: 85.66%
```

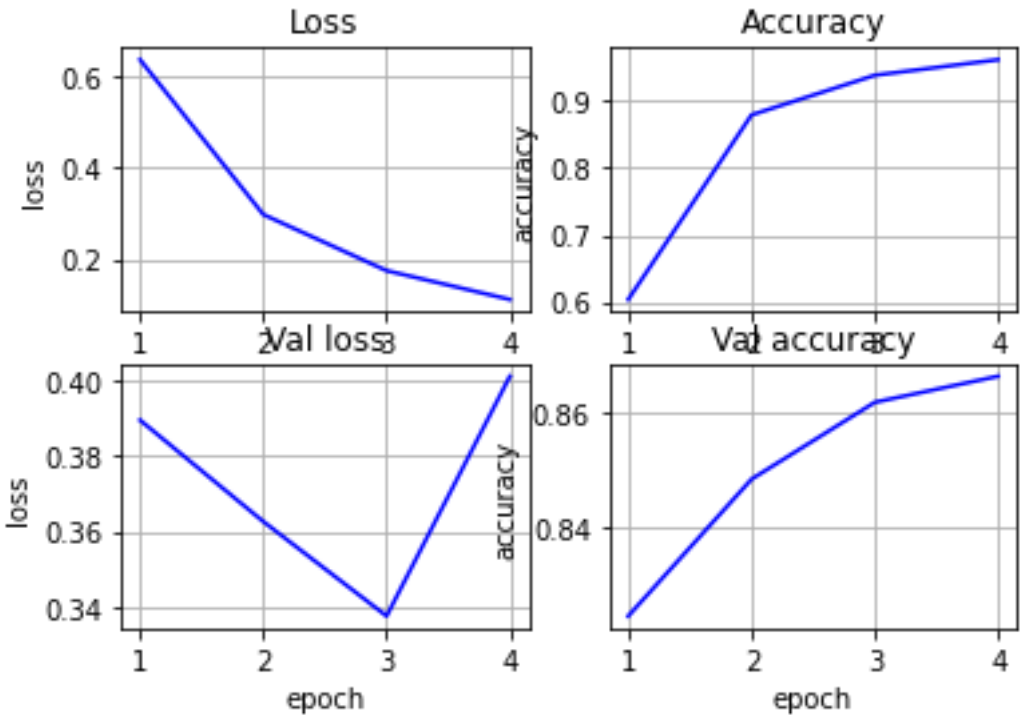



Результат стал заметно лучше, точность выросла на 3 процента и переобучение наблюдается достаточно слабо. Добавим в модель слои свертки и максимального пулинга:

```
model = Sequential()
model.add(Embedding(top_words, embedding_vector_length,
input_length=max_review_length))
model.add(Conv1D(filters=32, kernel_size=4, 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'])
```

Результат:

```
Accuracy: 86.64%
```




Точность выросла еще на процент. Теперь попробуем создать ансамбль из двух моделей второго типа, и двух - третьего:

```
Ensemble Accuracy: 86.57%
```

Точность ансамбля сетей фактически не изменилась.

Теперь прогоним два файла с реальными отзывами на обученном ансамбле сетей:

```
good_1.txt 0.996825
Positive
bad_1.txt 0.0031543388
Negative
```

Результаты выглядят подозрительно точными, но, учитывая содержимое файлов - много ключевых хороших и плохих слов, вполне допустимые.

## ВЫВОДЫ

В ходе лабораторной работы на датасете IMDB был построен и обучен ряд инс для классификации обзоров к фильмам. Среди построенных моделей были выбраны лучшие по ним проведено ансамблирование. Также написана

функция для обработки пользовательских отзывов, тестирование которой показало достаточно высокую точность предсказания.