

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №8**  
**по дисциплине «Искусственные нейронные сети»**  
**Тема: «Генерация текста на основе «Алисы в стране чудес»**

Студент гр. 8383

Степанов В.Д.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

## Цель.

Рекуррентные нейронные сети также могут быть использованы в качестве генеративных моделей.

Это означает, что в дополнение к тому, что они используются для прогнозных моделей (создания прогнозов), они могут изучать последовательности проблемы, а затем генерировать совершенно новые вероятные последовательности для проблемной области.

Подобные генеративные модели полезны не только для изучения того, насколько хорошо модель выявила проблему, но и для того, чтобы узнать больше о самой проблемной области.

## Выполнения работы.

Была реализована модель генерации текста из методических материалов. Для наблюдения генерации текста в зависимости от эпохи был добавлен Callback. После каждой эпохи генерируется текст и выводится в консоль.

Листинг 1 – Callback

```
class CustomCallback(keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs=None):

        if epoch % 4 == 0:
            # pick a random seed
            start = numpy.random.randint(0, len(dataX) - 1)
            pattern = dataX[start]
            print("Seed:")
            print("\n", ''.join([int_to_char[value] for value in pattern]),
                  "\n")

            # generate characters
            for i in range(10):
                x = numpy.reshape(pattern, (1, len(pattern), 1))
                x = x / float(n_vocab)
                prediction = model.predict(x, verbose=0)
                index = numpy.argmax(prediction)
                result = int_to_char[index]
                seq_in = [int_to_char[value] for value in pattern]
                sys.stdout.write(result)
                pattern.append(index)
                pattern = pattern[1:len(pattern)]
            print("\nDone.")
```

Для анализа процесса обучения был добавлен TensorFlowCallback.

## Листинг 2 – TensorFlowCallback

```
log_dir = "logs/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = keras.callbacks.TensorBoard(log_dir=log_dir,
histogram_freq=1, embeddings_freq=1)
```

## Листинг 3 – Генерация текста в разные эпохи

### Epoch 1

Seed:

" alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to "

Text:

**e toe toe toe toe toe toe toe toe toe toe toe**

Done.

### Epoch 5

Seed:

" alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing toe "

Text:

**tast to the sast oo the sast oo the sast oo the s**

Done.

### Epoch 10

Seed:

" alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing toe "

Text:

**thit hn the care an allce and the was aoiie an th**

Done.

### Epoch 15

Seed:

" alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing toe t "

Text:

**hat the was ho a lottee of the soeee of the soeee**

Done.

## Epoch 20

Seed:

" alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do "

Text:

**it wat oo the tas oo tie thst so the tas oo tie th**

Done.

Из листинга 3 можно увидеть, что нейронная сеть с каждой эпохой генерирует больше правильных слов и пытается образовать некоторую связь между ними.

Проведем анализ обучения нейронной сети.

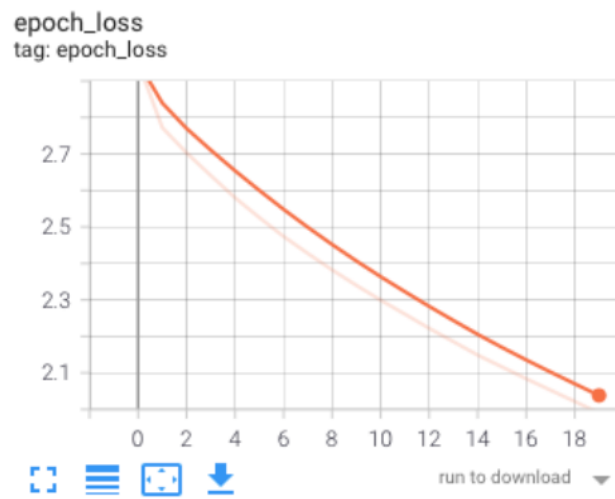


Рисунок 1 – График потерь

Из рис. 1 видно, что потери снижаются с каждой эпохой. На 20 эпохе потери равны примерно 2.

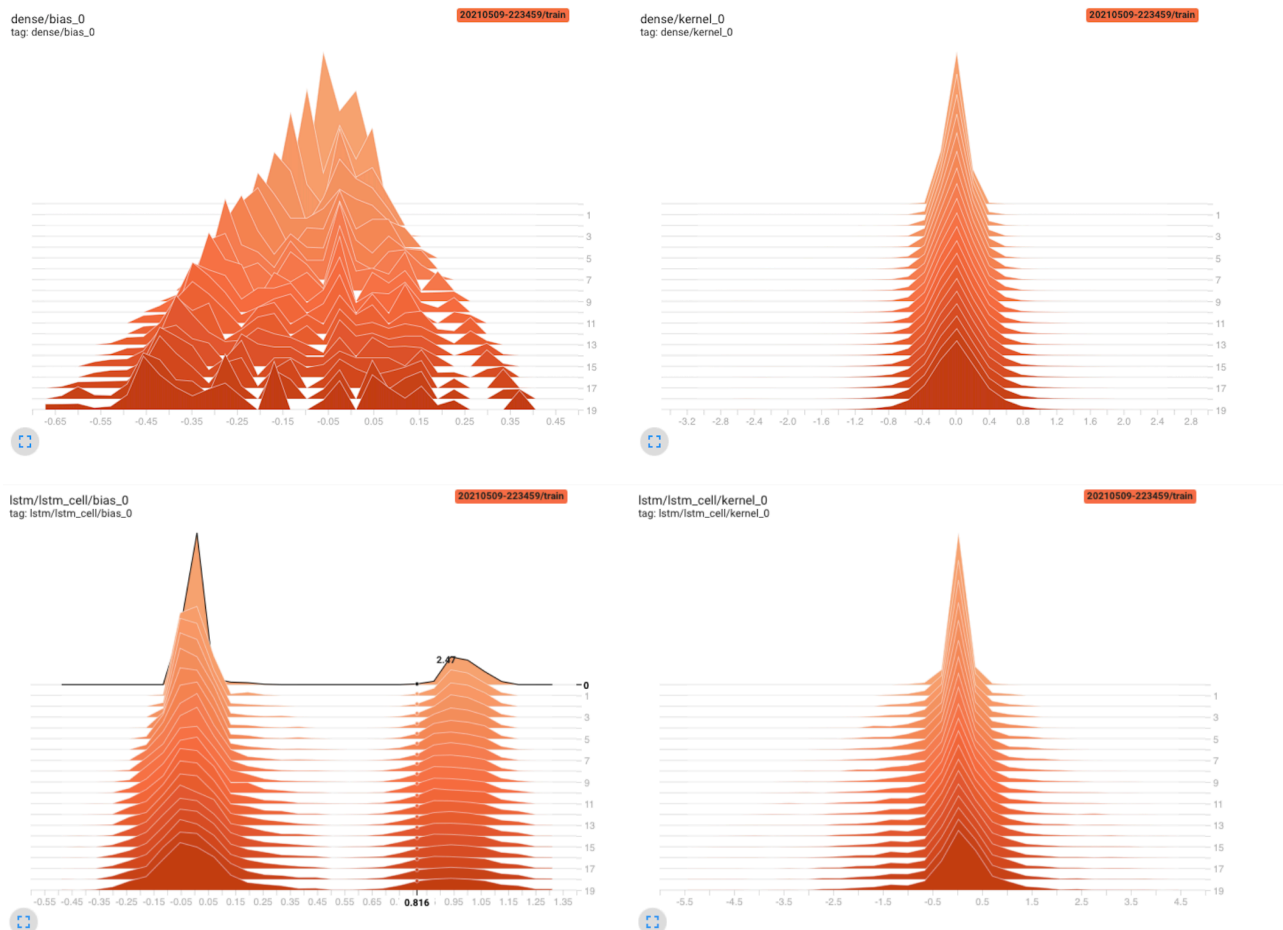


Рисунок 2 – Графики активации слоев

На рис. 2 можно увидеть значения весов, которые были установлены во время обучения.

### Вывод.

В ходе лабораторной работы была реализована нейронная сеть, которая генерирует текст на основе текста «Алиса в стране чудес». Сеть была изучена при помощи собственного CallBack и TensorFlowCallBack.