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"

Студентка гр. 8383 _____ Максимова А.А.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы

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.

Задачи

- Ознакомиться с генерацией текста
- Ознакомиться с системой Callback в Keras

Требования

1. Реализовать модель ИНС, которая будет генерировать текст
2. Написать собственный Callback, который будет показывать то как генерируется текст во время обучения (то есть раз в какое-то количество эпох генерировать и выводить текст у необученной модели)
3. Отследить процесс обучения при помощи TensorFlowCallBack (TensorBoard), в отчете привести результаты и их анализ

Основные теоретические положения

Рекуррентные нейронные сети:

Проблемой, возникающая при анализе текста с помощью полносвязанных нейронных сетей, является невозможность обработки текста как последовательности токенов. Решением является использование рекуррентных нейронный сетей, основное отличие которых заключается в использованием циклов, позволяющих учитывать информацию о том, что было получено на

прошлых шагах работы нейронной сети. Таким образом, порядок символов, слов или предложений, имеющих большой смысл в тексте, учитывается.

Рекуррентную нейронную сеть можно представить в виде сети с прямым распространением сигнала для чего используется прием, называемый разворачиванием во времени.

Рекуррентная нейронная сеть может работать в нескольких режимах:

- many to many: на вход подается последовательность, на выходе также возвращается последовательность (используется при автопереводах или генерации текста)
- many to one: вход - последовательность, выход - значение (задачи классификации, например, сентимент-анализ)
- one to many: например, описание изображений
- one to one: нелинейный вычисления, редко используется

Проблемами, возникающими при работе с рекуррентными нейронными сетями, являются: обучение требует длительного времени и больших вычислительных ресурсов, сигнал об уменьшении весов при передачи от слоя к слою уменьшается, ограниченная длительность запоминания предыдущей информации. Решение - использование более сложных архитектур рекуррентных нейронных сетей, например, LSTM и GRU.

Сеть LSTM:

Долгая краткосрочная память; элементом сети является набор из четырех слоев, взаимодействующих друг с другом, по определенным правилам, который называется ячейкой. Плюс: нет ограничения длительности запоминания предыдущей информации. Минус: состоит из множества элементов, поэтому чтобы обучить такую искусственную нейронную сеть необходимы большие вычислительные ресурсы.

Выполнение работы

1. Были импортированы все необходимые для работы классы и функции.

2. Текст книги был загружен и приведен в нижний регистр, чтобы уменьшить словарный запас, который должна выучить сеть. Код и результат представлены ниже.

```
filename = "wonderland.txt"
raw_text = open(filename).read()
raw_text = raw_text.lower()
```

```
alice's adventures in wonderland

lewis carroll

the millennium fulcrum edition 3.0




chapter i. down the rabbit-hole

alice was beginning to get very tired of sitting by her sister on the
bank, and of having nothing to do: once or twice she had peeped into the
book her sister was reading, but it had no pictures or conversations in
it, 'and what is the use of a book,' thought alice 'without pictures or
conversations?'
```

3. Далее было необходимо подготовить данные для моделирования нейронной сетью, путем преобразования символов в целочисленные данные. Был создан набор всех отдельных символов текста, а затем создана карта с уникальным целым числом.

```
chars = sorted(list(set(raw_text)))
chars_to_int = dict((c, i) for i, c in enumerate(chars))
print(chars_to_int)
```

```
{'\n': 0, ' ': 1, '!': 2, '"': 3, "'": 4, '(': 5, ')': 6, '*': 7, ',': 8, '-': 9, '.': 10, '0': 11, '3': 12, ':': 13, ';': 14, '?': 15,
Total Characters:  144408
Total Vocab:  45
```

4. После определили данные для обучения сети в виде последовательностей с фиксированной длиной в 100 символов произвольной длины. Каждый обучающий шаблон сети состоит из 100 временных шагов одного символа (X), за которым следует один символьный вывод (Y). При создании этих последовательностей мы перемещаем это окно по всей книге по одному символу за раз (кроме первых 100 символов). Например, если длина последовательности равна 5, то первые два шаблона обучения будут следующими:

```
CHAPT -> E
HAPTE -> R
```

Разделение на последовательности было выполнено с помощью следующей функции:

```python
def lookupTable(n_chars, seq_length, raw_text, chars_to_int):
    dataX = []
    dataY = []

    for i in range(0, n_chars - seq_length, 1): # 144408 - 100: start
        stop = i + seq_length
        seq_in = raw_text[i : stop] # 0 - 100; 1 - 101; ...;
        seq_out = raw_text[stop]

        dataX.append([chars_to_int[char] for char in seq_in])
        dataY.append([chars_to_int[seq_out]])
    n_patterns = len(dataX)
    print("Total Patterns: ", n_patterns)
    return dataX, dataY, n_patterns
```

5. Преобразование данных для использования с Keras. Изменение формата входных данных X и ОНЕ кодирование Y было выполнено с помощью следующей функции:

```python
def dataPreparation(dataX, dataY, n_patterns, seq_length, n_vocab):
    # 1) преобр. X в [образцы, временные шаги, особенности]
    X = np.reshape(dataX, (n_patterns, seq_length, 1))  # 144308 шаблонов длины 100 столбцами

    # изменение масштаба от 0 - 1 для облегчения изучения шаблонов сетью
    X = X / float(n_vocab)

    # one hot encoded Y
    Y = np_utils.to_categorical(dataY)
    return X, Y
```

6. После была определена функция для создания модели ИНС, состоящей из трех слоев:

I. Входной слой: LSTM - слой долгосрочной краткосрочной памяти (рекуррентный), содержащий 256 единиц памяти;

II. Скрытый слой: Dropout, используемый для предотвращения переобучения ИНС (вероятность исключения нейронов из сети равна 0.2);

III. Выходной слой: Dense - полносвязанный слой, использующий функцию Softmax: $\frac{e^{x_i}}{\sum_{i=0}^{k} e^{x_i}}$ для интерпретации прогнозов в терминах вероятности для каждого из 45 символов в диапазоне от 0 до 1.

7. Были определены следующие параметры обучения сети: в качестве функции потерь используется "categorical_crossentropy", которую предпочтительно использовать в задачах классификации, когда количество классов больше двух (45), оптимизатор - "adam".

```
def buildModel(X, Y):
    model = Sequential()
    model.add(LSTM(256, input_shape=(X.shape[1], X.shape[2])))
    model.add(Dropout(0.2))
    model.add(Dense(Y.shape[1], activation="softmax"))

    model.compile(
        loss='categorical_crossentropy',
        optimizer='adam'
    )

    return model
```

8. Тестовый набор данных отсутствует. Сеть работает медленно, поэтому используем контрольные точки модели для записи всех сетевых весов, чтобы регистрировать улучшение потерь (их уменьшение) в конце эпохи. Этот набор весов будем использовать при реализации генеративной модели. Запускаем обучение.

```
filepath = "weights-improvement-{epoch:02d}-{loss:.4f}.hdf5"
# для сохранения весов лучшей модели (в данном случае сохраняется вся модель)
# в файле в конце каждой эпохи (по умолчанию)
checkpoint = ModelCheckpoint(
    filepath,
    monitor='loss',  # отслеживаемая метрика - потери
    verbose=1,  # режим детализации
    save_best_only=True,  # сохранение для лучшей модели
    mode='min')  # решение о перезаписи в случае минимальных потерь

callbacks_list = [
    checkpoint,
    CallBackGT(numb_epoch),
    TensorBoard(
        log_dir="logs",
        # путь к каталогу, в котором сохраняются файлы журнала для анализа TensorBoard
        histogram_freq=1,
        # частота (в эпохах), с которой вычисляются гистрограммы активации и веса для слоев модели
        embeddings_freq=1
        # частота (в эпохах), с которой будут визуализироваться встраиваемые слои

    )]

model.fit(
    X, Y,
    epochs=30,
    batch_size=64,
    callbacks=callbacks_list)
```

9. *Генерация текста с помощью сети LSTM.* Загрузим модель, полученную на прошлом этапе (в *ModelCheckpoint* сохранялась вся модель, а не только значения весов).

```
elif stage == "2":
    model = load_model("weights-improvement-30-1.6691.hdf5")
    generateTextLSTM(dataX, dataY, model, n_vocab)
```

Также создадим обратное отображение целочисленных значений в символы, для перевода предсказаний.

10. Была написана функция для генерации текста. Выбирается случайный шаблон ввода в качестве начальной последовательности, а затем распечатываются сгенерированные символы по мере их генерации. Таким образом, была реализована ИНС, генерирующая текст.

```python
def generateTextLSTM(dataX, dataY, model, n_vocab, size=1000):
    start = np.random.randint(0, len(dataX) - 1)
    pattern = dataX[start]
    print("Seed:")
    print("\"", ''.join([int_to_char[value] for value in pattern]), "\"")

    for i in range(size):
        X = np.reshape(pattern, (1, len(pattern), 1))  # 1 шаблон длины len(pattern) в столбец
        X = X / float(n_vocab)

        prediction = model.predict(X, verbose=0)
        index = np.argmax(prediction)  # argmax - возвращает индекс максимального значения вдоль указанной оси

        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.')
```

11. Был написан собственный CallBack, показывающий то, как генерируется текст во время обучения, то есть раз в заданное пользователем количество эпох выводится текст, сгенерированный необученной моделью.

```python
class CallBackGT(Callback):
    def __init__(self, numb_epoch):
        self.numb_epoch = numb_epoch

    def on_epoch_end(self, epoch, logs=None):
        if epoch % int(self.numb_epoch) == 0:
            generateTextLSTM(dataX, dataY, self.model, n_vocab)
```

Результаты работы нейронной сети

1. Генерация текста во время обучения модели в течении 30 эпох с помощью CallBack (раз в 3 эпохи)

Так как получаемые сгенерированные данные занимают много места, то в данном разделе будут приведены только результаты, полученные на начальных и конечных эпохах обучения, а все полученные данные будут проанализированы в обобщающей таблице результатов.

Сгенерированный текст после 1 эпохи

Начальная последовательность:

" ce close to her ear. 'you're thinking

about something, my dear, and that makes you forget to talk. i "

Сгенерированный текст:

hd a dat tee a dat t aa the a

Значение потерь на эпохе: 2.8997

Сгенерированный текст после 4 эпохи

Начальная последовательность:

" t the jury--'

'if any one of them can explain it,' said alice, (she had grown so large

in the last "

Сгенерированный текст:

oo the tas oo the tas oo the tas hn the woue and the was ho the woele to the woel and the was ho the woele to the was hn the woue and the was ho the woele to the woel and the was ho the woele to the was hn the woue and the was ho the woele to the woel and the was ho the woele to the was hn the woue and the was ho the woele to the woel and the was ho the woele to the was hn the woue and the was ho the woele to the woel and the was ho the woele to the was hn the woue and the was ho the woele to the woel and the was ho the woele to the was hn the woue and the was ho the woele to the woel and the was ho the woele to the was hn the woue and the was ho the woele to the woel and the

Значение потерь на эпохе: 2.4970

Сгенерированный текст после 22 эпохи

Начальная последовательность:

" e that!' but she did not venture to say it out

loud.

'thinking again?' the duchess asked, with anot "

Сгенерированный текст:

her losked at the coulo.

'the moxer of thet a citd,' said the cate peihnedr bairiin the queen, and then she was aog toitenng the while rabbit, and then she was not ano a lont taie to herself, and she whst hn, and foon ae inrery to heve the rheese of the saie was of the saie was the wai aod toine and the had boowd the taie, and saed to herself, 'and what iar the coreousons '

'i maver't the kucy turtle to ae inre, said the cate peihnedr bairiin the queen, and then she was so toll to the thite tabbit, and then she was not ano a lont taie to herself, and she whst hn, and foon ae inrery to heve the rheese of the saie was of the saie was the wai aod toine and the had boowd the taie, and saed to herself, 'and what iar the coreousons '

'i maver't the kucy turtle to ae inre, said the cate peihnedr bairiin the queen, and then she was so toll to the thite tabbit, and then she was not ano a lont taie to herself, and she whst hn, and foon ae inrery to heve the rheese of the saie was of the

Значение потерь на эпохе: 1.7878

Сгенерированный текст после 25 эпохи

Начальная последовательность:

" oming. 'there's plenty of room!' said alice indignantly, and she sat

down in a large arm-chair at on "

Сгенерированный текст:

" oming. 'there's plenty of room!' said alice indignantly, and she sat

down in a large arm-chair at on "

ee she was so tie whnl the wan oo toe tooessee and the was and toiee the had boo tuted of the hadde at the wante of the tabli, and the was not a crean hnrly, and taed to herself, 'i soopese the lart was to ae it the tai on the sia-

' *

Значение потерь на эпохе: 1.7402

Сгенерированный текст после 28 эпохи

Начальная последовательность:

" another minute the whole head appeared, and then alice put

down her flamingo, and began an account "

Сгенерированный текст:

tf thene to her eare and the ragl wfse all an ouce, and saed to herself, 'why, thet's all the sam of the same sf then!'

'i dan't tel mot,' she manch hare waid to the grrphon. 'i wenn to the mett wiine,'

'then the lorse of the soaet ' cni the mouse was she matth har on the tereer an in sent oo to the toeete whe had soo th the tar of tiri to be io al ohcert aaai to her and the coch sare to the katter, and the west on whrh a little crore bnt soeee to the table an her aadi to her and whst she sooe of the saali, and the would hel loot beaan to the thie she coumd thet see was so tore to the thie she was a little boo tfe saadit sare to then she had boenting her and that sae it toie to be iu ar a lonsnr to tay 't ell of the goese of the seater,'

'thet doenn't betier thet,' said the manch hare.

'i seaul tou then to be i fengen to tou thet,' she macc torn an inreafadundy inre to herself, 'and thet alr doeln to be a frodouse '

'what would be ootting ' said the caterpillar.

'well, it s

Значение потерь на эпохе: 1.6913

Таблица 1 - Результаты

Эпоха	Значение потерь	Количество уникальных слов	Количество существующих слов	Дополнительно
1	2.8997	7	2	Происходит зацикливание
4	2.4970	12	3	-
7	2.3029	18	4	Появляются осмысленные слова, например, said. Появление апострофов, знаков препинания
10	2.1511	15	5	Несмотря на небольшой объем данного отрывка, по сравнению с предыдущими, соотношение уникальных слов ко всем имеет большее значение
13	2.0333	45	13	Многие несуществующие слова близки по написанию к реальным (отличаются на 1 букву)
16	1.9344	60	16	Появление абзацев
19	1.8536	41	18	Наблюдается зацикливание
22	1.7878	58	22	Также

				наблюдается зацикливание, но в меньшем количестве
25	1.7402	36	12	Большую часть сгенерированного отрывка составил разделитель абзацев (***)
28	1.6913	101	32	Зацикливания не обнаружено

Выводы:

Как видно из таблицы, чем дольше нейронная сеть обучается, тем больше возрастает ее "словарный запас", так еще на 13 эпохе ИНС генерировала текст из 45 уникальных слов, а на 28 эпохе - в два раза больше.

При этом можно заметить, что количество существующих слов в текстах относительно невелико - около 30%.

Начиная с 16 эпохи стали появляться абзацы.

Также с увеличением словарного запаса уменьшается количество повторений слов. Лучшие результаты, вероятно, можно было бы получить, если генерация текста производилась бы по словам, а не по символам.

2. Текст, сгенерированный моделью искусственной нейронной сетью с лучшими значениями весов

Начальная последовательность:

" ered to herself, 'the way all the creatures argue.
it's enough to drive one crazy!'

the footman see "

Сгенерированный текст:

ted to be then the was to tie botm and the garter was she was oot at the pagei 'she was to tie botn and the goese taale and tt ae inne she soee tore the carter in the seasen than what io sas toee in the was so tire. but she was not a coean hrrry to the sar of that sae it as the dad so keree the teie as her was she was so tire that she had boond the hoore of the sare of the sere and the saed tore thd had fuown th the whst siatey, and she goeo the madt tu hotlle and teiten.
'liwer the sirsln as whll ae inne of toe brilese' alice wait tn the kiyg,
anice was too ald mot hnt aeter the corrt, and the coeh hods that she was not a cread and aot aagut toine and then she had boond the saee whth she careen in hit hind to the thrle the sabli, "b'datter as thll as the sooks!' the said to herself, 'io t allae areeds to ao the woide"
'io saa ot toin ii the cornoos,' said the gatter. 'i wo dete tein to gn an thlt as the whiter.'

'i'se lote that the boemese, teed, said the gatter. 'i mo heve toe te thenk ao the coo of the seo.
citee semuehd the qieeee aedin the koek turtle seg oott aener and a food oa lote of the goose and the mory ailirrill rhe was aog tound the taali, and the houphon hevtelf aedins in the courd her head to herself io a lotule oot, for the coono seate roaeee on the steen of the sare of the sere ald the saee to the say anl and e can sean yhuh the coehe-on the saali, and the could not leke the docm sand so hes eead thet sas the mistle doure and bog to the dorr, she was sor as all rome the hoored and the pabb ou thi sare the cadk
wotkd the hoose taate and tuene on the seale, and she west so ani thing the had foond the sooe of the sore, "thet soee a croao an inte! said the gryphon.
'it saat't the khrt,' she said to herself in a tore of great her fertelf and ie thele so ge doe to the shotgst iott the gerter,'and i should think i can teed to the thotgs to the thing to toe than iot aeter it '

'ih, yhu hoew and theng ' said the manch hare.
"if a arehos tfane-'

'i ban't teink i can,t tem toat it ' said the manch hare.

'i c vorh the huer sate to ao tha coohors, taid the gatter. and the dormouse said to then.'

the qabbit sas toa tiat soaption, and the mors art and theng ' crt atere the qoee turtle see rone tireok doune thet sae thet she was not a cread andlne to than the had boond the rabli, aut she west oo arif wire she was so toe thth the doort of the court, and the coeh hods taade ioto a little soaee of the sare of the seee as well as i

Значение потерь на эпохе: 1.6691

Вывод: количество уникальных слов в полученном тексте равно 268, длина текста в словах - 541, при этом существующих слов около 30%, а несуществующих, но очень похожи по написанию на реальные слова , около 40-50 %. Также отсутствует зацикливание, есть абзацы и знаки препинания.

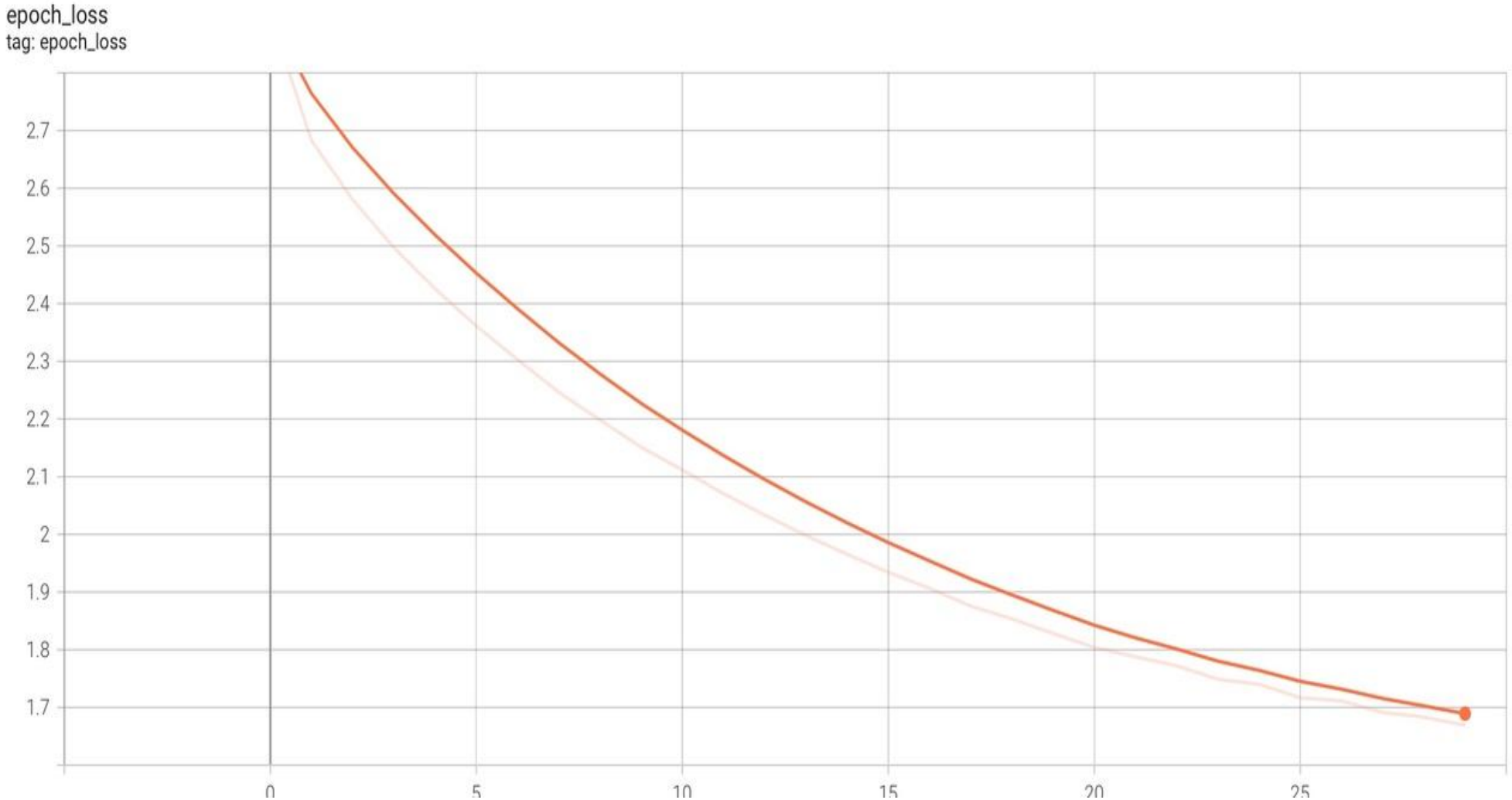
3. Отслеживание (визуализация) процесса обучения при помощи TensorFlowCallBack (TensorBoard)

Был добавлен *CallBack TensorBoard*, позволяющий по данным (логам), записываемым в процессе обучения в директорию *logs*, визуализировать процесс обучения модели с помощью следующей команды: *tensorboard --logdir=logs*.

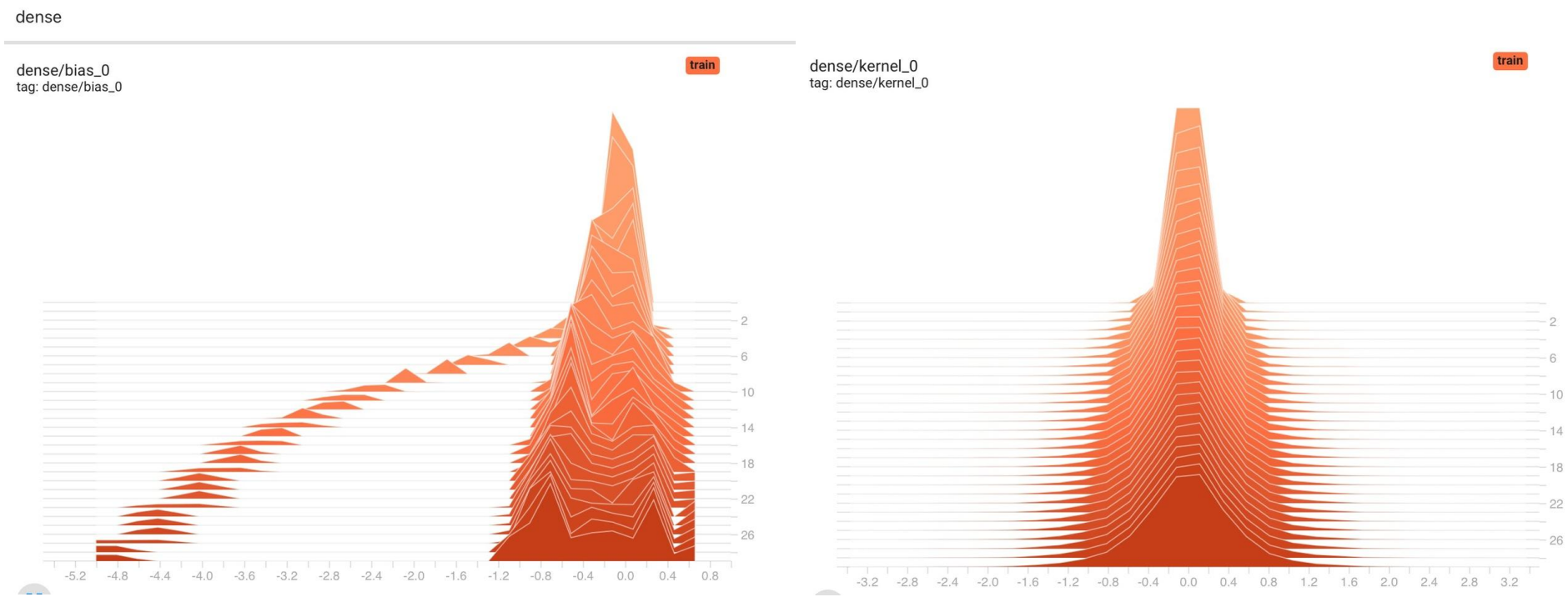
```
callbacks_list = [
    checkpoint,
    CallBackGT(numb_epoch),
    TensorBoard(
        log_dir="logs",
        # путь к каталогу, в котором сохраняются файлы журнала для анализа TensorBoard
        histogram_freq=1,
        # частота (в эпохах), с которой вычисляются гистрограммы активации и веса для слоев модели
        embeddings_freq=1
        # частота (в эпохах), с которой будут визуализироваться встраиваемые слои

    )]
```

Есть возможность просмотреть изменения значения потерь на обучающем наборе данных в процессе обучения, график представлен ниже. Как видно, данные графика соответствуют приведенным значениям потерь в таблице 1: после 15 эпохи значение потерь становится меньше 2.

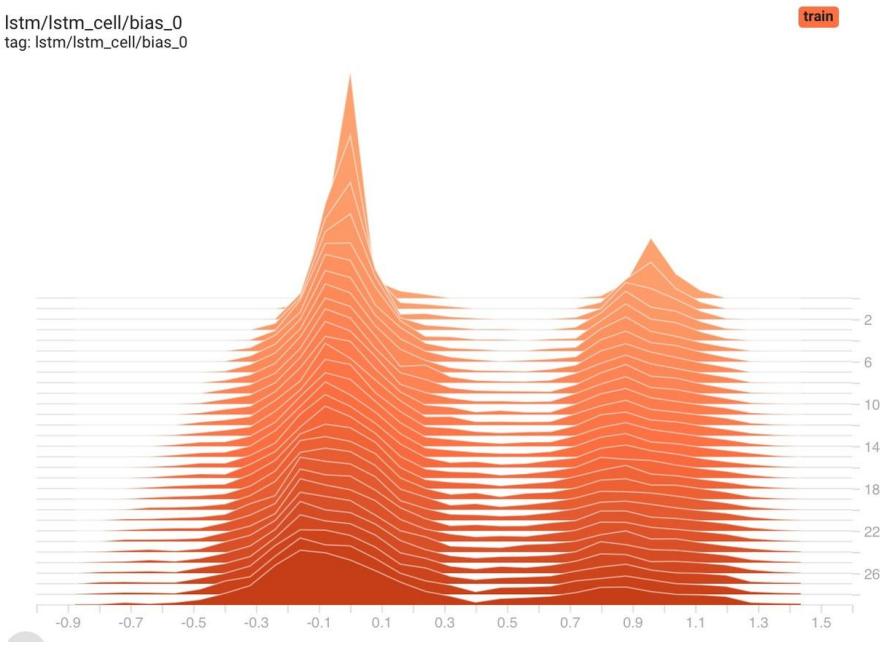


Также есть возможность просмотреть гистограммы активации слоев ИНС, представленные ниже.

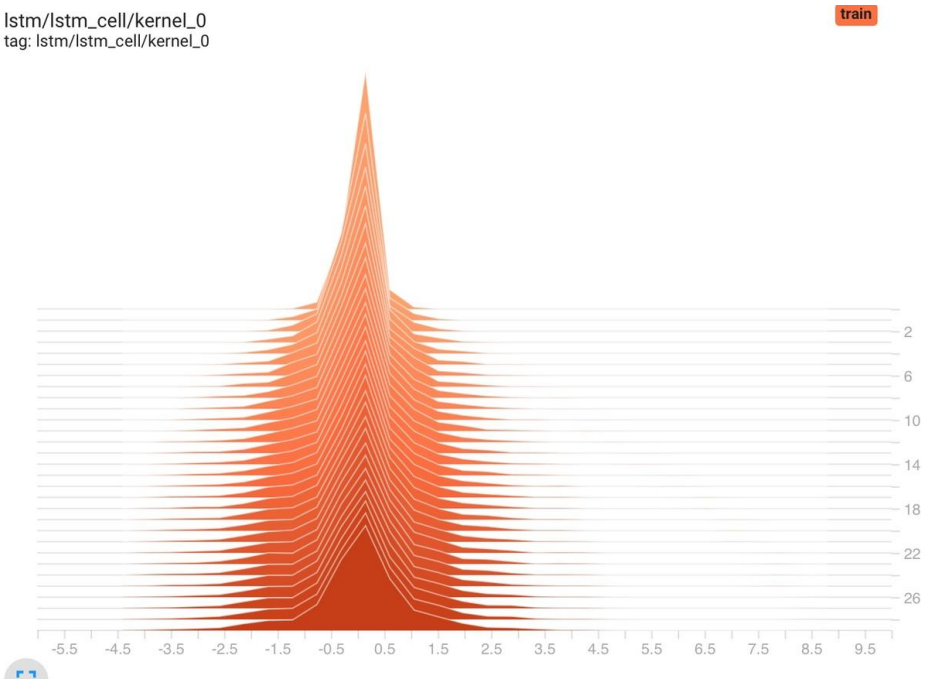


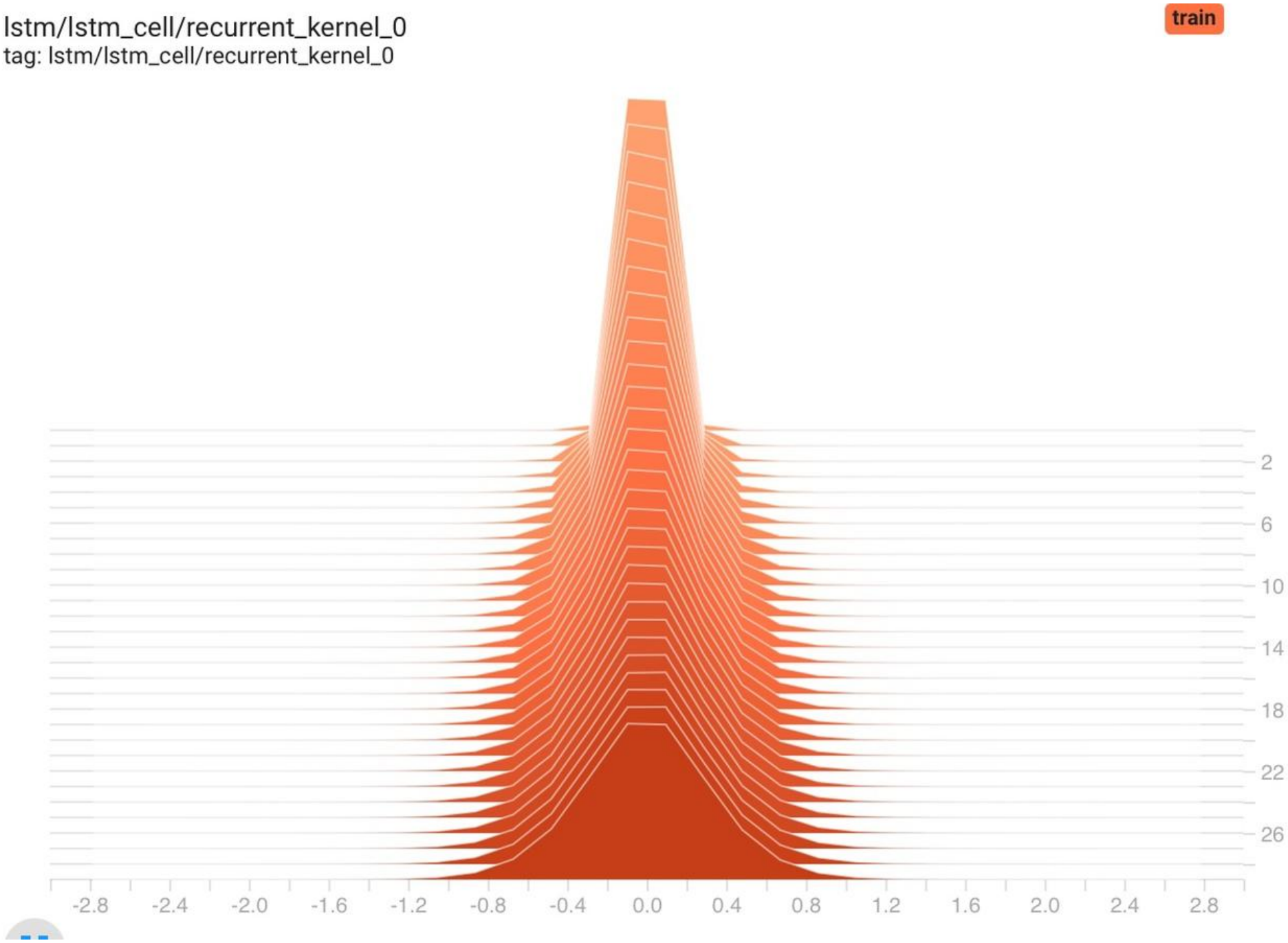
lstm

lstm/lstm_cell/bias_0
tag: lstm/lstm_cell/bias_0



lstm/lstm_cell/kernel_0
tag: lstm/lstm_cell/kernel_0




lstm/lstm_cell/recurrent_kernel_0
tag: lstm/lstm_cell/recurrent_kernel_0
train
2
6
10
14
18
22
26
-2.8
-2.4
-2.0
-1.6
-1.2
-0.8
-0.4
0.0
0.4
0.8
1.2
1.6
2.0
2.4
2.8

Выводы

В результате выполнения лабораторной работы была реализована модель искусственной нейронной сети, решающая задачу генерации текста. Наименьшее достигнутое значение потерь было равно 1.6691. Был написан и протестирован собственный Callback, позволяющий просматривать то, как генерируется текст у еще необученной модели раз в некоторое, заданное пользователем, количество эпох. Также было произведено отслеживание процесса обучения модели с помощью *TensorBoard*.