

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

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

Студент гр. 8383

Шишкин И.В.

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

Выполнение работы.

Был загружен текст ASCII для книги и преобразованы все символы в нижний регистр, чтобы уменьшить словарный запас, который сеть должна выучить.

Была создана модель, представленная в листинге 1.

Листинг 1 – Модель

```
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'))
```

Был написан собственный callback, который будет показывать то как генерируется текст во время обучения (то есть раз в какое-то количество эпох генерирует и выводит текст у необученной модели).

Помимо этого, написан checkpoint, который сохраняет веса

Листинг 2 – Callback

```
filepath = "weights-improvement-{epoch:02d}-{loss:.4f}.hdf5"
```

```

checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1,
save_best_only=True, mode='min')
num_epochs = 20
interval = 5

class MyCallback(keras.callbacks.Callback):
    def __init__(self):
        super(MyCallback, self).__init__()

    def on_epoch_end(self, epoch, logs=None):
        if epoch == 0 or epoch == num_epochs - 1 or epoch % interval ==
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(50):
                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]
                sys.stdout.write(result)
                pattern.append(index)
                pattern = pattern[1:len(pattern)]

            print("\nDone.")

tb_callback = TensorBoard(log_dir="./logs", histogram_freq=2,
write_graph=True, embeddings_freq=1)
callbacks_list = [checkpoint, MyCallback(), tb_callback]

```

Всего программа обучалась на 20 эпохах, каждая эпоха занимала примерно 250 секунд времени. По итогу были созданы файлы, представленные на рис. 1 (веса). В них же можно и отследить, как менялась ошибка в зависимости от эпохи.

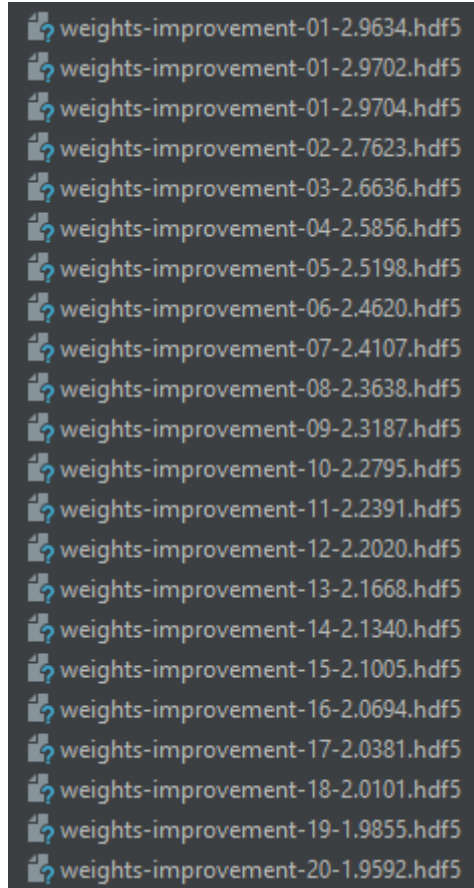


Рисунок 1 – Созданные программой файлы

Интервал для callback'а установлен раз в 5 эпох. На 1 эпохе программа вывела следующий текст:

Seed:

```
" there were three little sisters,' the dormouse began  
in a great hurry; 'and their names were elsie, "  
toe tee toe toe toe toe toe toe toe toe toe to  
Done.
```

На 6 эпохе:

Seed:

```
" see.'
```

```
'i don't see,' said the caterpillar.
```

```
'i'm afraid i can't put it more clearly,' alice replied "
an alice,
'the sas a lite the tait ' said the ma
Done.
```

На 11 эпохе:

```
Seed:
" o the earth. at last the gryphon said
to the mock turtle, 'drive on, old fellow! don't be all day ab "
lut it ' shi said to herself, 'no tou to te the to
Done.
```

На 16 эпохе:

```
Seed:
" dormouse, not choosing to
notice this last remark.

'of course they were', said the dormouse; '--we "
at a lirtle of then '
'i dane wai shi ganter aa
Done.
```

На 20 эпохе:

```
Seed:
" e different!' the mock turtle repeated thoughtfully. 'i
should like to hear her try and repeat somet "
hing toenele!'

'io soen ho hs ' said the kanter.
Done.
```

Как видно, если на 1 эпохе печаталось постоянно одно и то же слово, то на 20 эпохе у сети получилось выдавать существующие слова, и даже иногда связанные.

Был отслежен процесс обучения при помощи TensorFlowCallback (TensorBoard)

На рис. 2 представлен график потерь. На рис. 3 представлены гистограммы активация для слоев нейросети.



Рисунок 2 – График потерь

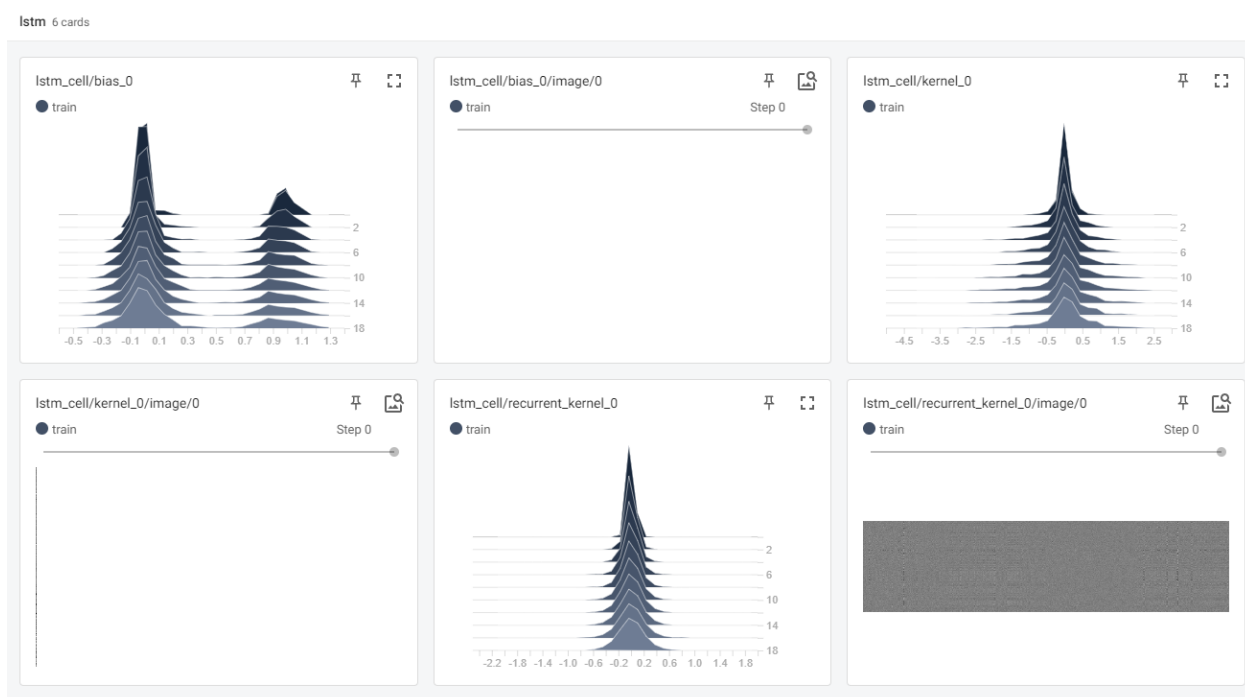


Рисунок 3 – Активации для слоев

Выводы.

Была реализована модель ИНС, которая будет генерировать текст. Написан собственный Callback, который будет показывать то, как генерируется текст во время обучения. Отслежен процесс обучения при помощи TensorFlowCallback (TensorBoard).