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. 7383 _____ Александров Р.А.

Преподаватель _____ Жукова Н.А.

Санкт-Петербург
2020

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.

Постановка задачи.

1. Ознакомиться с генерацией текста
2. 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 генирировать и выводить текст у необученной модели)
3. Отследить процесс обучения при помощи TensorFlowCallBack, в отчете привести результаты и их анализ

Выполнение работы.

В ходе работы была создана и обучена модель нейронной сети, весь код представлен в приложении А. В архитектуре сети определен один скрытый слой LSTM с 256 единицами памяти. Сеть использует выпадение с вероятностью 20. Выходной уровень – это плотный уровень, использующий функцию активации softmax для вывода прогнозирования вероятности для каждого из 47 символов в диапазоне от 0 до 1.

Первоначальные параметры: количество эпох равно 20, размер батча равен 128.

Для наблюдения за процессом обучения, был написан callback, который в ходе обучения в конце 1 и каждой 3 эпохи выводит текст у необученной модели. Результат генерации текста в ходе обучения сети представлен в табл. 1.

Таблица 1 – Результат работы написанного callback

Номер эпохи	Результат
1	ah aa
3	the woet toe toet toe toet the toet the toee to the toree to th
6	t to the care and the tas io the tooee th the tas io the tast oo the tas io the care and the tas io the tooee th the tas io the tast oo the tas io the care and the tas io the tooee th the tas io the tast oo the tas io the care and the tas io the tooee th the tas io the tast oo the tas io the care and the tas io the tooee th the tas io the tast oo the tas io the care and the tas io the tooee th the tas io the tast oo the tas

	io the care and the tas io the tooee th the tas io the tast oo the tas io the care and the tas io the tooee th the tas io the tast oo the tas io the care and the tas io the tooee th the tas io the tast oo the tas io the care and the tas io the tooee th the tas io the tast oo the tas io the care and the tas io the tooee th the tas io the tast oo the tas io the care and the tas io the tooee th the tas io the tast oo the tas io the care and the tas io the tooee th the tas io the tast oo the tas io the care and the tas io the tooee th the tas io the tast o
9	se toe teat ' said the more turtle 'the harter was an inre th the toete
12	little tooee to tee to tee toede the hortd she har hn an anl tou doo the was ano aalut the would the karter wo tel the hodt the had no tae to tee the had aoon an the could and the woile tas a little to tele the would the had no tae to tee th the was to ter the har hn aalit the was anl the corlous and the war anl the corlouse the sas anl aro oo the sas hn aalit the was anl the corlous and the wai sotel the was oo tie care and the could so the was to the tas oo the care and the could so the was to the tas oo the care and the could so the was to the tas oo the care and the could so the was to the tas oo the care and the could so the was to the tas oo the care and the could so the was to the tas oo the care and the could so the was to the tas oo the care and the could so the was to the tas oo the care and the could so the was to the tas oo the care and the could so the was to the tas oo the care and the could so the was to the tas oo
15	waid" shi said to herself, and the war sored an toe care an in had sote the was aol areie and the sas anw aadin, and the wai sorel see thet in the was aol arl aoo oo ar hlrs the had been hot a mott ootele toile to the was the whrt oadteng and then whe had noted her hend to the kort of the coulouse to the was the whrt oadteng and then whe had noted her hend to the kort of the coulouse to the was the whrt oadteng and then whe had noted her hend to the kort of the coulouse to the was the whrt oadteng and then whe had noted her hend to the kort of the coulouse to the was the whrt oadteng and then whe had noted her hend to the kort of the coulouse to the was the whrt oadteng and then whe had noted her hend to the kort of the coulouse to the was the whrt oadteng and then whe had noted her hend to the kort of the coulouse to the was the whrt oadteng and then whe had noted her hend to the kort of

18	and the tas a little brerte of the sord of the court, and she was aelin it an anl ana an an fnntt tfet sas aoo and aor oo ari to the shete and the waited to teye the sord of the courd ald the was a little brertente to the thete rat an the cirr aadi and the waited an inre of the sable, and the waited to tee thete was a ain oo the taate, and the waited to tee thete was a ain ano oo tie taate and the waited an inre tf thet war oo sere the sabdit wothd the tas ano and aro anl and anr and anang to the saete. and she was aeling to toik at the could to the taate, and the was anl all hoow thet she white rabbit and the was a little brertente to the thete rabdit, and the was anl all hoow thet she was oo tere the sabdi, and whe waited to tee thete was a virye tf the sable, and the waited to tee thete was a ain oo the taate, and the waited to tee thete was a ain ano oo tie taate and the waited an inre tf thet war oo sere the sabdit wothd the tas ano and aro anl and anr and anang to the saet

Результат генерации текста на последней эпохе представлен на рис. 1.

ad to tee woede oo the soedl of the couro,
'the cormouse sai to toen iar on the soadteone'' she said th the cayerpillar.

'iese yhu, said the caterpillar.

'ie ioueed tourse'' said the maccit in a lorr wonce and thing toele aoon and the dorlouse oo the saale, and she was not aeoin at in and to the saye taate and the waid to the kook, and the wai the worle aalet toineig an inr sonee, and she white wabbit was soine the gad deve thin whe had notee herself an toe couldrseng that she was not a cooa to the kabg what it was toe cotld se thing to the toed tf the gorroen of the courd, and whe hotpee anl hoowed an anl oo the sable and the white wabbit was soine the gad deve toine the was no toeke on the saad to the gotth so the was the war whet sare
ald the wail to the koot, and aadan woineing to ani eor and corn he thnce and teen toine anl aroeoring to her her hnr and aegine
the had never here the coulo soe was so tee whyhd the gar hnr aerit at the could,
'the cormouse said ' said th

Рисунок 1 – Результат генерации текста на обученной модели

Графики потерь и точности в ходе обучения модели представлены на рис. 2 и 3.

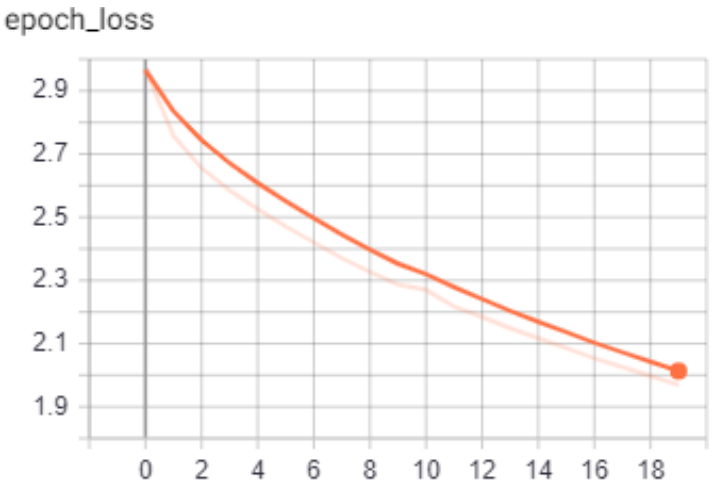

Рисунок 2 – График потерь

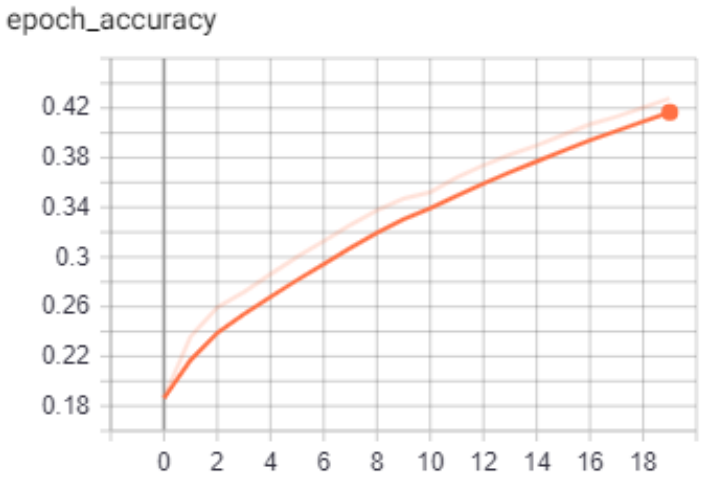

Рисунок 3 – График точности

Можно отметить некоторые замечания по поводу сгенерированного текста.

- Символы разделены на словесные группы, и большинство групп представляют собой настоящие английские слова (например, «she», «said», «never», «war»), но многие этого не делают (например, «ioueed», «tourse», «aaelet»).
- Некоторые слова в последовательности имеют смысл (например, «she was not»), но большинство не несет в себе смысловой нагрузки. Результаты не идеальны.

Выводы.

В ходе работы были изучены задача генерации текста и система callback в keras нейронными сетями с использованием python и keras, был написан собственный callback, который в процессе обучения модели генерировал текст в конце определенной эпохи.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

```
import numpy
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.callbacks import ModelCheckpoint, Callback,
TensorBoard
from tensorflow.keras.layers import Dropout, Dense, LSTM
from tensorflow.keras.models import Sequential

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

chars = sorted(list(set(raw_text)))
char_to_int = dict((c, i) for i, c in enumerate(chars))

n_chars = len(raw_text)
n_vocab = len(chars)

print("Total Characters: ", n_chars)
print("Total Vocab: ", n_vocab)

seq_length = 100

dataX = []
dataY = []

for i in range(0, n_chars - seq_length, 1):
    seq_in = raw_text[i:i + seq_length]
    seq_out = raw_text[i + seq_length]
    dataX.append([char_to_int[char] for char in seq_in])
    dataY.append(char_to_int[seq_out])

n_patterns = len(dataX)
print("Total Patterns: ", n_patterns)

# reshape X to be [samples, time steps, features]
X = numpy.reshape(dataX, (n_patterns, seq_length, 1))
# normalize
X = X / float(n_vocab)
# one hot encode the output variable
y = to_categorical(dataY)

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',
metrics=['accuracy'])

# define the checkpoint
filepath = "weights-improvement-{epoch:02d}-{loss:.4f}.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1,
save_best_only=True, mode='min')

def test_network(epoch):
    # create mapping of unique chars to integers, and a reverse
mapping
    chars = sorted(list(set(raw_text)))
    int_to_char = dict((i, c) for i, c in enumerate(chars))

    # pick a random seed
    start = numpy.random.randint(0, len(dataX) - 1)
    pattern = dataX[start]

    resultText = []
    # generate characters
    for i in range(1000):
        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]
        resultText.append(result)
        pattern.append(index)
        pattern = pattern[1:len(pattern)]
    save_text = 'Epoch ' + str(epoch) + '\n' + ''.join(resultText)
+ '\n'
    f = open("result.txt", "a")
    f.write(save_text)
    f.close()

class CustomCallback(Callback):
    def __init__(self):
        self.__epochCounter = 0

    def get_x(self):
        return self.__epochCounter

    def set_x(self, x):
        self.__epochCounter = x
```

```
    epochCounter = 0

    def on_epoch_end(self, epoch, logs=None):
        self.epochCounter = self.epochCounter + 1
        if self.epochCounter == 1 or self.epochCounter % 3 == 0:
            print("End epoch {} of training".format(epoch))
            test_network(self.epochCounter)

tbCallBack   =   TensorBoard(log_dir='./logs',   histogram_freq=1,
write_graph=True, write_images=True,
                         profile_batch=100000000)
callbacks_list = [
    checkpoint,
    tbCallBack,
    CustomCallback()
]
model.fit(X,          y,          epochs=20,          batch_size=128,
callbacks=callbacks_list)

_list)
```