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

ОТЧЕТ

по лабораторной работе №8

по дисциплине «Искусственные нейронные сети»

Тема: «Генерация текста на основе “Алисы в стране чудес”»

Студент гр. 7381 _____ Вологдин М.Д.

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.

Порядок выполнения работы.

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

Требования.

- Реализовать модель ИНС, которая будет генерировать текст
- 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)
- Отследить процесс обучения при помощи TensorFlowCallBack, в отчете привести результаты и их анализ

Ход работы.

В данной лабораторной работе мы будем использовать в качестве набора данных Приключения Алисы в Стране Чудес Льюиса Кэрролла. Мы собираемся изучить зависимости между символами и условные вероятности символов в последовательностях, чтобы мы могли, в свою очередь, генерировать совершенно новые и оригинальные последовательности символов.

1. Была построена нейронная сеть со следующей архитектурой:

```
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')
model.fit(X, y, epochs=30, batch_size=128, callbacks=callbacks_list)
```

2. Напишем CallBack:

```
class Mycallback(callbacks.Callback):
    def __init__(self, epochs):
        super(Mycallback, self).__init__()
        self.epochs = epochs
    def on_epoch_end(self, epoch, logs=None):
        if epoch in self.epochs:
            random_generate(self.model, epoch)
```

Получили следующие результаты:

and the woile to tee toete to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to tee toeee to the toeee and the woiee to

Рисунок 1 – Текст на 7 эпохе

Наблюдаем повторяющуюся последовательно небольшой длины из набора символов

i veon the cadsd then ' said the moree, 'ih tou dn lo the sooet oaree then '
'i dane the mase thrne thu ' said the monke, "ih tou dne't tete thu ' said the manch hare.

'ih tou dne't teoe the roree ' said the kacte paree hare rhrt hn, 'io s she mort was an all to the to tee the mooer oaree the hade '
'ih you dad totn the sert ' said the monke, "ih tou dne't tete thu ' said the manch hare.

'ih tou dne't teoe the roree ' said the kacte paree hare rhrt hn, 'io s she mort was an all to the to tee the mooer oaree the hade '
'ih you dad totn the sert ' said the monke, "ih tou dne't tete thu ' said the manch hare.

'ih tou dne't teoe the roree ' said the kacte paree hare rhrt hn, 'io s she mort was an all to the to tee the mooer oaree the hade '
'ih you dad totn the sert ' said the monke, "ih tou dne't tete thu ' said the manch hare.

'ih tou dne't teoe the roree ' said the kacte paree hare rhrt hn, 'io s she mort was an all to the to tee the mooer oaree the hade '
'ih you dad totn

Рисунок 2 – Текст на 12 эпохе

Наблюдаем повторяющуюся последовательность уже большей длины, в которой есть нормальные слова.

celie an toll a linal hf ar aelu and thred ho she was oo toe tore. 'adde so the sane to tae the master whte the was oo the sasee and tee then she was tot oo hor the tipe th the tabbet wotld sae the had feen to tee to the kote, and the white rabbit here the look aalk wo the thet had aeden, and the tooed ho was toen iir head ao tee tait oo the rabbit wotld ger hn the was oo the tar of the tore. 'adde so her herd and theng to toenk and toeer if the master was an inr tae to tee that shee and the wait dn in tie madt of then and the war a little boa ald theer it was and the whit sare toee to tee thte the hareen and the wai e wirt har and toeeren and toeering oo the taale and the wait dno to tert an innsett in an thll to tae tfte the had feen to tee to tee thth the was oo the tore. 'atd the toele sae ieltely ano she wai a fit ou teel of the lance hareer aadut thet saeee and the was anling ao toleriing at the caterpillar and the waited to be anpinusty to tee tht ou rar the hareen bni

Рисунок 3 – Текст на 22 эпохе

Текст состоит не из повторяющихся последовательностей, но всё еще неосознанный. Встречаются несуществующие слова.

r fand, and she taited aoo ali thete was sot oheel and toredg inr hea
and toen shet soe tasee tf the whr, white whs gad to tee the whrter of the ranc and whs then soe of the boore, and was soenring to sieak to see it the was to the tored aerer the samee on her head to tee thite ati it was the mirt and the rane tf then she was to tie tone aod she white rabbit seadr the can and then she was to tie tone aod she white rabbit seadr the rimel of the court, and whs the wan soi ohrel hn a four waateng badut iar ho the mictter was on the whnt wotn the was so sie thrhe ho the was and thene was soeeting.
'i moot be a lere a cioe,' said the manch hare.

'iere your majesty,' said the kong tasdet, 'and thet so the wan on the sood '

she had not loke a lort eaa fotn the thane and she white rabbit head to be a booroon to sae it her hand. 'ald thet so the whrt so woic to toe thi horte and the mart rateated of the saalen sf belier thel, and the white rabbit seadr the rimel of the court, and whs

Рисунок 4 – Текст на 30 эпохе

Текст получился вполне осознанным, но присутствуют грамматические ошибки.

Теперь сгенерируем текст с использованием обученной сети LSTM с лучшими весами из файла `weights-improvement-18-2.0076.hdf5`.

Получили следующий результат:

the was aoing the har on the harter, and the was soin in sheel thet sae in a lorg oanters oo the thile was aeling thet, and whe had no tee toiet oaee the had fetee har hen hand and she was a little brol an the caterpillar and the war a little soace th the tabbet woul ier fend and the war eoon in oo the was oo the tabte bnd the was a little brol an the caterpillar and the war eoon the rabbit worl the whnt lnr and the was a little brale of the harter, and the was soin in she was aoing th the tabbet woul it and whet soee to the harter whth the wan oo the sire and eele the horg on tee soiet ohee thet she was to thn thrh the wan oo the tiree barl and the wai e aater and whet sae io tas ao the cad the master sas the mirtle tired of the court, 'and the march hare was a little brol an the caterpillar and the war a little soace th the tabbet woul in a lort braat oo the sabbin, and she was a little brold th the tabbet woul ier fend and the war a gitt was on the cin.
'what so toe bagl

Рисунок 5 – Текст на обученной сети.

Текст почти осознанный с грамматическими ошибками.

По тенденции видно, что с увеличением количества эпох сеть начинает выдавать всё лучший и лучший результат, но также значительно растет и время на обучение.

Выводы.

В ходе выполнения данной работы ознакомились с рекуррентными нейронными сетями в качестве генеративных моделей, с их помощью построили и обучили сеть для генерации текста на основе «Алисы в стране чудес»

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД

```
from os import listdir, path
import numpy
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import LSTM
from keras.callbacks import ModelCheckpoint, callbacks
from keras.utils import np_utils


class Mycallback(callbacks.Callback):
    def __init__(self, epochs):
        super(Mycallback, self).__init__()
        self.epochs = epochs

    def on_epoch_end(self, epoch, logs=None):
        if epoch in self.epochs:
            random_generate(self.model, epoch)


def random_generate(_model, epoch=0):
    start = numpy.random.randint(0, len(dataX) - 1)
    pattern = dataX[start]
    print("Seed:")
    print("\"", ''.join([int_to_char[value] for value in
pattern]), "\"")
    text = []
    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]
        text.append(result)
        print(result, end='')
        pattern.append(index)
        pattern = pattern[1:len(pattern)]
    print("\nDone.")
    with open('text_{}.txt'.format(epoch), 'w') as file:
        file.write(''.join(text))
```

```
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))
int_to_char = dict((i, c) 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)

X = numpy.reshape(dataX, (n_patterns, seq_length, 1))
X = X / float(n_vocab)
y = np_utils.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')

filepath = "weights-improvement-{epoch:02d}-{loss:.4f}.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1,
save_best_only=True, mode='min')
callbacks_list = [checkpoint, Mycallback([6, 11, 21, 29])]

model.fit(X, y, epochs=30, batch_size=128,
callbacks=callbacks_list)

folder = '.'
filename = ''
min = 999999
```

```
for name in listdir(folder):
    full_name = path.join(folder, name)
    if path.isfile(full_name) and full_name.find('.hdf5') != -1:
        model_loss = int(full_name.split('.')[2])
        if min > model_loss:
            min = model_loss
            filename = full_name

print(filename)
model.load_weights(filename)
model.compile(loss='categorical_crossentropy', optimizer='adam')
random_generate(model)
```