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”»

Студентка гр. 7383 _____ Прокопенко Н.

Преподаватель _____ Жукова Н.А.

Санкт-Петербург
2020

Цель работы.

Реализовать генерацию текста на основе текста из сказки «Алиса в стране чудес».

Задачи.

- 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.

1. Была создана модель рекуррентной нейронной сети, для прогнозирования следующего символа на основе предыдущих (код программы представлен в приложении А). В качестве функции потерь была выбрана функция categorical_crossentropy.

Архитектура сети:

```
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'))
```

Для отслеживания процесса генерации текста во время обучения нейронной сети был написан собственный обратный вызов Callback. Он выводит текст у необученной модели через некоторое количество эпох.

Отследим процесс обучения и рассмотрим тексты, сгенерированные после 1, 10, 15 и 20 эпох. Ниже приведены результаты:

1) Результат после первой эпохи:

Seed:

" four hours, i think; or is

it twelve? i--'

'oh, don't bother me,' said the duchess; 'i never could a "

o ao to ao

2) Результат после десятой эпохи:

Seed:

" ldn't see it?' so she stood still where she was,

and waited.

when the procession came opposite to al "

cn a lotee to the toeee and the soiee th the cace and the soiee th the care ra the care ra the cact. and the woine tas io a lort of the care and the soiee th the cace and the soiee th the care ra the care ra the cact. and the woine tas io a lort of the care and the soiee th the cace and the soiee th the care ra the care ra the cact. and the woine tas io a lort of the care and the soiee th the cace and the soiee th the care

3) Результат после пятнадцатой эпохи:

Seed:

" ,' said the youth, 'one would hardly suppose

that your eye was as steady as ever;

yet you bal "

en to the soaee

4) Результат после двадцатой эпохи:

Seed:

" ast i know

who i was when i got up this morning, but i think i must have been

changed several times "

hire the soiee 'a doosers toue io the soeee of the soiee of the goush, she hocst was gate toe toine in the moeere tfe woile ' 'no sou dile then ' said the kong, and the qoeee sas in a long oo teee and whe oueer of thit were toing to toene thet and the tas oo toeel ano the sooe of the gourt, nh tourde to be anl the soiee of the soiee of the court, and whs go the woide and thene and the soiee of the soiee of the court, and tas anl toeereng bnd she soiee of the sooe of the sooe of the sabbit sete and the white rabbit, and the woide afdun to the toeee and the tabbit and the white tabbit wouh the hors, and the woide afdun to the toeee and thetoeereng bnd she soiee of the sooe of the sooe of the sabbit sete and the white rabbit, and the woide afdun to the toeee and the tabbit and the white tabbit wouh the hors, and the woide afdun to the toeee and the tabbit and the white tabbit wouh the hors, and the woide afdun to the toeee and the tabbit and the white tabbit wouh the hors, and the woide afdun to the toeee and the tabbit and the white tabbit wouh the hors, and the woide afdun to the toeee and the tabbit and the white tabbit wouh the hors, and the woide afdun to the toeee and the tabbi

По сгенерированным текстам видно, что с увеличением количества эпох, увеличивается качество генерируемого текста. Так, после первой эпохи была сгенерирована повторяющаяся последовательность из трех букв, после девятой эпохи была также сгенерирована повторяющаяся последовательность, но уже с

большим количеством символов. После восемнадцатой эпохи текст был сгенерирован с меньшими повторениями, и с бóльшим количеством действительно существующих в английском языке слов.

Вывод.

В ходе выполнения лабораторной работы была построена модель, генерирующая текст на основе книги «Алиса в стране чудес». Был также написан собственный CallBack, позволяющий отслеживать процесс обучения сети. В результате обучения сеть с каждой эпохой генерировала тексты с всё меньшим количеством повторений и большим количеством существующих слов.

ПРИЛОЖЕНИЕ А

```
import numpy
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import LSTM
import tensorflow.keras.callbacks
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.utils import to_categorical
from keras.utils import np_utils
import sys

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)


def generateSequence(model):
    start = numpy.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(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]
        seq_in = [int_to_char[value] for value in pattern]
        sys.stdout.write(result)
        pattern.append(index)
        pattern = pattern[1:len(pattern)]


class make_text(tensorflow.keras.callbacks.Callback):
    def __init__(self, epochs):
        super(make_text, self).__init__()
        self.epochs = epochs

    def on_epoch_end(self, epoch, logs={}):
        if epoch in self.epochs:
            generateSequence(model)


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=20, batch_size=128, callbacks=[make_text([0, 9, 14,
```

```
19])])
```