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 стане чудес»

Студент гр. 8382 _____ Гордиенко А.М.

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 (создание 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.

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 (TensorBoard), в отчете привести результаты и их анализ

Ход работы.

В ходе работы были использованы следующие зависимости.

```
import re
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, Callback, TensorBoard
from keras.utils import np_utils
```

Для загрузки и предобработки текста были использован следующий код.

```
COLAB_PREFIX = "/content/sample_data"
filename = "/content/drive/MyDrive/Colab Notebooks/wonderland.txt"
raw_text = open(filename).read()
raw_text = raw_text.lower()
raw_text = raw_text.replace("*", "")
raw_text = re.sub(" +", " ", raw_text)
raw_text = re.sub("\n+", "\n", raw_text)

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)
```

В данном коде весь текст приводится к нижнему регистру, а также сокращается число повторных белых символов и удаляются звездочки. Затем генерируются два словаря: один по символу возвращает число – «код» символа,, другой – наоборот. После этого определяются данные для обучения. Для этого текст разбивается на последовательности кодов символа длиной 100, таким образом, что из каждого последующего шаблона выбрасывается первый код и в конец добавляется следующий. Также фиксируется метка – код символа, расположенный сразу после последовательности.

Затем происходит нормализация и приведение меток к удобному для обработки виду.

Сеть состоит из слоев LSTM и Dropout. Выходной слой Dense с функцией активации softmax. Использовалась функция потерь categorical_crossentropy, оптимизатор – Adam.

```
model = Sequential()
model.add(LSTM(512, 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')
```

Использованные Callbacks.

```
filepath= COLAB_PREFIX + "/weights-improvement-{epoch:02d}-{loss:.4f}.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1, save_best_only=True, mode='min')
tensorboard = TensorBoard(log_dir=f"/{COLAB_PREFIX}/tensorboard", histogram_freq=1, embeddings_freq=1),
callbacks_list = [checkpoint, tensorboard, MyCallback()]
```

Первый Callback отвечает за сохранение модели после каждой эпохи, если она смогла уменьшить ошибку. Второй позволит определить процесс обучения с помощью TensorBoard. Третий CallBack реализован самостоятельно, и он отвечает за сохранение сгенерированного текста после каждой эпохи.

```
class MyCallback(Callback):
  def on_epoch_end(self, epoch, logs=None):
    text = gen_text(self.model)
    with open(f"{COLAB_PREFIX}\gererated_text\{epoch}.txt", 'w') as file:
      file.write(text)
```

Функция для генерации текста.

```
def gen_text(model):
  start = numpy.random.randint(0, len(dataX)-1)
  pattern = dataX[start]
  result = []
  print("Seed:")
  print("\"", ''.join([int_to_char[value] for value in pattern]), "\"")
  # 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.append(int_to_char[index])
          pattern.append(index)
          pattern = pattern[1:len(pattern)]
  return "".join(result)
```

Сгенерированные тексты на эпохах: 1, 5, 10, 20, 30.

Эпоха 1.

the an ah ao

ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao ao

Поначалу сгенерированный текст состоял из зацикленных символов, не представляющих осмысленный текст.

Эпоха 5.

e matter was aooie and the woile was aol aroreen to the white wabbit and the woile was a little so aro the woile and the whit ho wou dad tot th the while was ao in and the woile whs ao the could she whst hn an the could and the whit had boo aro aro aro aro aro oo the while whst her an the cad -oo the soie oo the soie oo the sab at the was aoo aroree an the cad -oo the soie oo the soie oo the sab at the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aroree an the cad not the was aoo aro

Начали формироваться слова, но смысла в них все еще не было.

Эпоха 10.

uf the whse the wordd 'and the dorso so the thite tas an in   she harter was aol thr dlrn the wirl of the woide and the world coen hn an ofcentene to the whiter and the whrt hordln th the whrt saa ohe was no the tar of the sabe thre thet sored oo toe tirt hire thet same whu i whsh tou doun the woide ' the manthr went on, 'io soel a ling a tamee of the soaee ' the garter was aooihus to be   the harter was aol thr dlrn the was aoo arouee and the tooe and the whrt ooo of the gorse and the toiel thre the gorse of the gorse, and the tooed her and the tuoele to be in ael to tee the was oo the wan taiding and tuenene the was all the tored oe the gorse, and the whrt ooo of the gorse and the tooed her head to the careen the was aolin to tee ot hnr the was aol the worle ger and the whrt ooo of the sore of the garden, whu and toe toole ' the kanthr west on, 'io soel

a lintle oi the soaele she kanthr went on, 'in you dane to toe that ' the taid to herself, 'io wou dane to toe that

Все еще получался случайный порядок символов, но уже видны слова: the, no, was, of и прочие.

Эпоха 20.

he she kerter teye toe oocer ooaseo. 'now in thet done,' saed the caterpillar.

'well, iere you hoew,' she macc to herself, 'th the lart rore the coeatuit.t ooke that '

'i maver taad to ' aader the thiee 'eadl iistrrs,tionsy, and see quoed so han oetted then in whech sae then she had never befn in a hitrle waine, and the was not ou tirolng and lorkid an oocer oiant,

and then she was not oo gis some mine and a freat hurry, an aelsed out if she tase whit sam she was aoiing in har hand, and the teought if the halle rad woat soene oad the was a little serien, bod then soee a foeat crawl '

and she whoteht it ou rorml she said thee, she was not ofeg oo tie onhe,s tear horrledly, and fegt i sas aol oirel aeaone the rage as the hndhe of the court, and the whote tar aroeng dnd oo the oafe ou thon she had beon tp the gane.

'ho s g thould think ' said the king. 'that see sarer hareey breaml.'

what are you tolle in ' said the king. 'than the farter was you cilint ntw ' the karter was iooking to

Появилось деление на абзацы и задатки прямой речи.

Эпоха 30.

a mittle boimll as she could, and waited till she whole thing as she wpnde; ant the fad neter betouser beaoted ant outt on a cirm with at yhll as she oofeen so eanden rhat had fallen anoeg aack to the oafe an teer as she west on totisg and in whet some wh thems that shry selled ano ooasssos. and the tert ooat chm the caly wat o lrch sored aedore her and seeir teat to buith hnro the sooe of the cack, at last the gorpmose whsl a lotile so tooe at toe oo kws, 'the wosld goen heve ne anl tretiig ano over without b lomentrso to iave geene and an hnrerion.

and eos'dues in a goodttee blile, yhu, i soelt io b'sam ho loreng '

'bot hen taad to all mearly fereer, and here said alice. 'i mege what in she saa!'tould 'toupl,'

the gorpman reparked.

'low that saei of areamire, aedode '

said the gryphon, 'it would heve to al toa riteon 'yhu, ther" 'l seal you douldngt take ' thi aaterrillar crdlied dersedy, 'weal ane you thinking of?'

'i beg your pardon,' said alice in a tont of hreat durlss,-fnt

Появился план повествования с прямой речью, диалогами. Сгенерированный текст хоть все еще содержит слова с ошибками стал более осмысленным.

Выводы.

В ходе выполнения лабораторной работы был изучен способ генерации текста с помощью рекуррентных нейронных сетей. Была использована механика CallBack’ов для отслеживания процесса обучения модели. Нельзя сказать, что сеть способна сгенерировать осмысленный текст, но тем не менее она достаточно часто выдает существующие слова и конструкции, напоминающие прямую речь. Зацикливания вывода нейронной сети наблюдались лишь на ранних эпохах обучения