

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

Кортев Ю. В.

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.

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 генирировать и выводить текст у необученной модели)
- Отследить процесс обучения при помощи TensorFlowCallBack, в отчете привести результаты и их анализ

Ход работы.

- **Реализовать модель ИНС, которая будет генерировать текст**

Инициализация и параметры обучения модели показаны в листинге 1.

```
model = Sequential()
model.add(layers.LSTM(128, input_shape=(maxlen, len(chars))))
model.add(layers.Dense(len(chars), activation='softmax'))
optimizer = keras.optimizers.RMSprop(lr=0.01)
model.compile(loss='categorical_crossentropy', optimizer=optimizer)
model.fit(x,y,batch_size=128,epochs=100,callbacks=[Gen_text(text_seed=
text_seed),checkpoint],verbose=2)
```

Листинг 1 - Архитектура и параметры обучения модели генерации текста

- **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)**

В листинге 2 показан callback, который генерирует текст из 400 символов в конце каждой эпохи, с использованием 2х разных подходов:

1. Наивный выбор символа, где выбирается индекс символа с максимальным значениям выхода softmax.
2. Стохастический выход, где символ выбирается случайно. Выход softmax лишь играет роль в распределении вероятности.

```

class Gen_text(callbacks.Callback):
    def __init__(self, text_seed):
        self.text_seed=text_seed

    def on_epoch_end(self, epoch, logs=None):

        if(epoch==20 or epoch==30 or epoch==60 or True):
            for temp in [0.5,0.01]:
                print("temp: ",temp)
                gen_text = self.text_seed[:]
                sys.stdout.write(gen_text)
                for i in range(400):
                    sampled = np.zeros((1, maxlen, len(chars)))
                    for t, char in enumerate(gen_text):
                        sampled[0, t, char_indices[char]] = 1

                    pred = self.model.predict(sampled, verbose=0)[0]
                    next_char = chars[sample(pred, temp)]

                    gen_text += next_char
                    gen_text = gen_text[1:]
                    sys.stdout.write(next_char)
                    if(i%100==0):
                        print()

                print('\n-----\n')

```

Листинг 2 - Callback генерации текста

- Отследить процесс обучения при помощи TensorFlowCallBack, в отчете привести результаты и их анализ

Текст, взятый из книги, который использовался, как начальная точка генерации:

'Never!' said the Queen furiously

Анализ:

1ая эпоха:

```
temp: 0.5
'never!' said the queen furiously
moungt to meats!' she batting the was and the see the which has the
mome the sook about the gooled t
he mooking to hit wasine the rayce, with the more the mound her of the
toied the goon the cores, 'in
ting the the way the queen the mutter the mousther the sooked of the
waingithing the mourge say the t
ooe the said the madle they gooked in the peame the was the hadper the
haster at on the toinge the
-----

temp: 0.01
'never!' said the queen furiously
the mound the mound the mound the mound the mound the mound the mound
the mound the mound the mound
the mound the mound the mound the mound the mound the mound the mound
the mound the mound the mound
the mound the mound the mound the mound the mound the mound the mound
the mound the mound the mound
the mound the mound the mound the mound the mound the mound the mound
the mound the mound the mound
-----
```

Сначала модель генерирует текст стохастическим методом, затем наивным. Во 2м случае структура текста постоянно повторяется, с другой стороны уже в конце 1й эпохи модель генерирует настоящие слова. В 1м случае слова разнообразнее, но при этом много несуществующих слов.

Эпоха 44:

```
Epoch 44/60
- 26s - loss: 0.7721
temp: 0.5
'never!' said the queen furiously.

'we went on at all.'

'i must be here,' said the queen.

'not i'll grow treal which it was lostlrie
n any canted a ray to her off and for some could go to think the thing
ear, and the right was
little
harte head!pooran any grow her face.

'yes, but it is the duches in must be not ence,rand the come
turning to a moras, that she can first about it, and rane it was a
large quite one, with a suddenly
-----

temp: 0.01
'never!' said the queen furiously.

'whalll well, with the dormouse she went on a little not.'

'i can't grinl will you have a ver ell
of such a little not.'

'i can't persony are she grow use or very little glars not.

'well, wish wha
t were that!'

'i can't grinl will you have a bit--
'i hear they'll curt
it on at all was
only thin
g, who had not got wordht, and was got to her near the sentence to her
had folling to great her han
-----
```

Появляется пунктуация, цитаты героев обособляются кавычками, после цитат генерируются слово said+имя героя.

```
- 26s - loss: 0.5622
temp: 0.5
'never!' said the queen furiously:
  'but it was an unto the queen
and advence, and she said to herself knewis, whock and fetred it, and
  began get for the little now and
that mark, who had more cup in her facess, and had all them of the
  trees again.

'then the duchess so,' said alice, and ins: they was looking at the
shosing of this s
ired to say, 'i wonder?' and the moral
dy?' and the more turtle went on theild.

'i didn to your ta
-----

temp: 0.01
'never!' said the queen furiously:
  'but it's the was gone for the moment she was now.'

'i'm no bow use it such at all the time in her
e,' alice went down
anamouse, who it was a large know.'

'i cat get to go neen't be and more have co
uld go down that marks to he shook, but the door all the or say to the
mouse to lit made out that,'
the mock turtle said: 'now i must hall the ever had a new wo her sever
that she wasn't got to do, i
-----
```

Появляется более сложная структура: ‘цитата’ слова автора:’продолжение цитаты’

Больше правильной пунктуации.

```
temp: 0.5
'never!' said the queen furiously.

'we dist or she had to then she had flar being mempered to sense, and
the pance to say
out to be to be so, you could,' said alice, who fayling about in to
las the court what. but she turned all. don't rement was a book to the
first of
minder with one of it at once
timevly, and all that it was a larger, i teay,'w
she
was plopleten out to the jurymoon, and looking at the brin, and began
back on t
-----

temp: 0.01
'never!' said the queen furiously.

'we dentemay lest down it about
the goos next grow to her hand ambounay it very little glans compl
e, why,' said the dormouse, who had to listen the shark and funty fist
on which she had this could cat her hand, and all that,' said the mock
turtle.

'grop the ever!' said alice heard
the time the hatter went on taking the hatters.

alice was solline, and the
dads and great hurrying, and consideri
-----
```

Цитаты стали длиннее, в стохастическом случае генерируется более интересный текст, с более реалистичной в некоторых местах структурой, а несуществующие слова кажутся правдоподобными.

Вывод

В итоге данной лабораторной работы, была изучена работа рекуррентной нейронной сети. Создана модель генерирующая текст на основе Алисы в стране чудес.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ ТЕКСТ

```
import numpy as np
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
from keras import layers, callbacks
from keras.optimizers import RMSprop
from keras.models import Sequential, load_model

import sys

import random
random.seed(999)

def sample(preds, temperature=1.0):
    preds = np.asarray(preds).astype('float64')
    preds = np.log(preds) / temperature
    exp_preds = np.exp(preds)
    preds = exp_preds / np.sum(exp_preds)
    probas = np.random.multinomial(1, preds, 1)
    return np.argmax(probas)

class Gen_text(callbacks.Callback):
    def __init__(self, text_seed):
        self.text_seed=text_seed

    def on_epoch_end(self, epoch, logs=None):

        if(epoch==20 or epoch==30 or epoch==60 or True):
            for temp in [0.5,0.01]:
                print("temp: ",temp)
                gen_text = self.text_seed[:]
                sys.stdout.write(gen_text)
                for i in range(400):
                    sampled = np.zeros((1, maxlen, len(chars)))
                    for t, char in enumerate(gen_text):
                        sampled[0, t, char_indices[char]] = 1

                    pred = self.model.predict(sampled, verbose=0)[0]
                    next_char = chars[sample(pred, temp)]

                    gen_text += next_char
                    gen_text = gen_text[1:]
                    sys.stdout.write(next_char)
                    if(i%100==0):
                        print()
```

```

        print('\n-----\n')

text_seed="'Never!' said the Queen furiously".lower()

maxlen=len(text_seed)
step=3

text=open("wonderland.txt").read().lower()
chars=sorted(list(set(text)))
char_indices=dict((char,chars.index(char)) for char in chars)

text_seed_ind=random.randint(0,len(text)-maxlen-1)

senteces=[]
next_chars=[]

for i in range(0,len(text)-maxlen,step):
    senteces.append(text[i:i+maxlen])
    next_chars.append(text[i+maxlen])

x=np.zeros((len(senteces),maxlen,len(chars)),dtype=np.bool)
y=np.zeros((len(senteces),len(chars)), dtype=np.bool)

for i,sentece in enumerate(senteces):
    for t, char in enumerate(sentece):
        x[i,t,char_indices[char]]=1
        y[i, char_indices[next_chars[i]]]=1

filepath="weights-improvement.hdf5"
checkpoint = callbacks.ModelCheckpoint(filepath, monitor='loss',
verbose=1, save_best_only=True, mode='min')

model = Sequential()
model.add(layers.LSTM(128, input_shape=(maxlen, len(chars))))
model.add(layers.Dense(len(chars), activation='softmax'))
optimizer = keras.optimizers.RMSprop(lr=0.01)
model.compile(loss='categorical_crossentropy', optimizer=optimizer)
model.fit(x,y,batch_size=128,epochs=100,callbacks=[Gen_text(text_seed=
text_seed),checkpoint],verbose=2)

```