**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. 7382 | ____________________ | Гаврилов А.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.

**Требования к выполнению задания.**

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

**Ход работы.**

Была написана модель, генерирующая текст по последовательности символов, преобразованных в целые числа. Программа приведена в приложении А.

Сеть обучалась в течении 35 эпох и распечатывался промежуточный результат еще на нескольких эпохах, лог с частичным выводом программы приведен в приложении В.

После 35 эпох мы получаем следующие строки в выводе:

Паттерн:

" he found herself at
last in the beautiful garden, among the bright flower-beds and the cool
fountain "

Вывод:

e the wood of the tooes har ir the whid tf the was so the soeer. and then she wes gown and nooked an ier hand  and the sere whin the was aol ano the girten, and saed to the sore, "shen soe toanl wase toen i can tote i saan to the thing!' she said to herself, 'a lont shsh the war an i mave toued the mooet oase toen i mate wout ier her '
'i whsl to toe theng tou donn ho ' said the monk turtle ang no the jory. and the sere shing the whit so whrh the tois, 'a lanee hare to the whit so aalen to the whit so toon the whrt siteree '

'io io toe than the work,' said the monk turtle an the soner.

'ie iour thet toe than ' said the monk turtle an the soner.
and then so see toin in a lorg of the tane of the tooe. and the seie the was so toek to the tooe, and saed to the sore, "shen soe toanl wase toen i can tote i saan to the thing ' she said to herself, "shan yhu whal i whink tou doon hn a can taat io a latter
'a dane the whit so toent io ao a far on the gorw,' she said to herself, 'a lont

В вывод можно проследить понятные нам слова, модель даже попыталась воспроизвести диалог по данному паттерну, но при 35 эпохах и потерей в 1.8 у нее это получилось не самым лучшим образом.

Также был написан callback, который на поданных эпохах генерирует слова по паттерну. Эти данные есть в логе (приложение В).

Еще был добавлен callback, который сохраняет состояние модели на каждой эпохе. Была составлена строчка, по которой функция сохраняет эту модель с нужной нам информацией в названии файла, которую в конце мы просматриваем, чтобы определить какое состояние имеет наименьшие потери.

**Вывод.**

В результате лабораторной работы были изучены callback функции в keras и их возможности для облегчения обучения и контроля за моделью. Также научились генерировать текст с помощью нейронной сети.

# ПРИЛОЖЕНИЕ А

# ИСХОДНЫЙ КОД ПРОГРАММЫ

```
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


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)


class gen_text_callback(callbacks.Callback):
    def __init__(self, epochs, data, print_fun):
        super(gen_text_callback, self).__init__()
        self.epochs = epochs
        self.dataX = data
        self.print_seq = print_fun

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



class network():
    def get_data(self):
        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 = np_utils.to_categorical(dataY)
        return X, y, dataX

    def build_model(self):
        X, y, dataX = self.get_data()
        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')
        epochs = [1, 10, 15]
        callbacks_list = [checkpoint, gen_text_callback(epochs, dataX,
self.print_seq)]
        model.fit(X, y, epochs=35, batch_size=128,
callbacks=callbacks_list)
        return model

    def print_seq(self, model, dataX):
        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]
            print(result, end='')
            pattern.append(index)
```

```
            pattern = pattern[1:len(pattern)]

nw = network()
model =nw.build_model()

folder = '.'
filename = ''
min = 100000
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')
print('FULL MODEL')
X, y, dataX = nw.get_data()
nw.print_seq(model, dataX)
print("\nDone.")
```

# Приложение В.

```
Epoch 00002: loss improved from 2.99058 to 2.80753, saving model to
weights-improvement-02-2.8075.hdf5
Seed:
" the hatter, 'you wouldn't talk
about wasting it. it's him.'

'i don't know what you mean,' said alic "
 the  and the  and the  and the  and the  and the  and the  and the
and the  and the  and the  and the  and the  and the  and the  and the
and the  and the  and the  and the  and the  and the  and the  and the
and the  and the  and the  and the  and the  and the  and the  and the
and the  and the  and the  and the  and the  and the  and the  and the
and the  and the  and the  and the  and the  and the  and the  and the
and the  and the  and the  and the  and the  and the  and the  and the
and the  and the  and the  and the  and the  and the  and the  and the
and the  and the  and the  and the  and the  and the  and the  and the
and the  and the  and the  and the  and the  and the  and the  and the
and the  and the  and the  and the  and the  and the  and the  and the
and the  and the  and the  and the  and the  and the  and the  and the
and the  and the  and the  and the  and the  and the  and the  and the
and the  and the  and the  and the  and the  and the  and the  and
Epoch 3/35



Epoch 00003: loss improved from 2.80753 to 2.72239, saving model to
weights-improvement-03-2.7224.hdf5
Epoch 4/35


Epoch 00004: loss improved from 2.72239 to 2.65565, saving model to
weights-improvement-04-2.6557.hdf5
Epoch 5/35


Epoch 00005: loss improved from 2.65565 to 2.59980, saving model to
weights-improvement-05-2.5998.hdf5
Epoch 6/35


Epoch 00006: loss improved from 2.59980 to 2.54561, saving model to
weights-improvement-06-2.5456.hdf5
Epoch 7/35


Epoch 00007: loss improved from 2.54561 to 2.49828, saving model to
weights-improvement-07-2.4983.hdf5
Epoch 8/35
```

```
Epoch 00008: loss improved from 2.49828 to 2.45384, saving model to
weights-improvement-08-2.4538.hdf5
Epoch 9/35

Epoch 00009: loss improved from 2.45384 to 2.41282, saving model to
weights-improvement-09-2.4128.hdf5
Epoch 10/35

Epoch 00010: loss improved from 2.41282 to 2.37309, saving model to
weights-improvement-10-2.3731.hdf5
Epoch 11/35


Epoch 00011: loss improved from 2.37309 to 2.33829, saving model to
weights-improvement-11-2.3383.hdf5
Seed:
" he owl and the panther were sharing a pie--'

    [later editions continued as follows
  the panther "
 oe the caree an
Epoch 12/35


Epoch 00012: loss improved from 2.33829 to 2.30580, saving model to
weights-improvement-12-2.3058.hdf5
Epoch 13/35


Epoch 00013: loss improved from 2.30580 to 2.27194, saving model to
weights-improvement-13-2.2719.hdf5
Epoch 14/35


Epoch 00014: loss improved from 2.27194 to 2.24034, saving model to
weights-improvement-14-2.2403.hdf5
Epoch 15/35


Epoch 00015: loss improved from 2.24034 to 2.21064, saving model to
weights-improvement-15-2.2106.hdf5
Epoch 16/35


Epoch 00016: loss improved from 2.21064 to 2.18242, saving model to
weights-improvement-16-2.1824.hdf5
Seed:
" g down stairs! how brave they'll all think me at
home! why, i wouldn't say anything about it, even i "
n the maree hare wout a lottle ofree to tee toeee '
```

```
'i whsu toe donw the woons,' said the daterpillar.

'iele tou dene toen a gan oe the more,  said the dat, '' ''
```

(дальше выписал много \n)

```
Epoch 17/35


Epoch 00017: loss improved from 2.18242 to 2.15391, saving model to
weights-improvement-17-2.1539.hdf5
Epoch 18/35


Epoch 00018: loss improved from 2.15391 to 2.12793, saving model to
weights-improvement-18-2.1279.hdf5
Epoch 19/35


Epoch 00019: loss improved from 2.12793 to 2.10255, saving model to
weights-improvement-19-2.1025.hdf5
Epoch 20/35


Epoch 00020: loss improved from 2.10255 to 2.07985, saving model to
weights-improvement-20-2.0798.hdf5
Epoch 21/35


Epoch 00021: loss improved from 2.07985 to 2.05896, saving model to
weights-improvement-21-2.0590.hdf5
Epoch 22/35


Epoch 00022: loss improved from 2.05896 to 2.03748, saving model to
weights-improvement-22-2.0375.hdf5
Epoch 23/35



Epoch 00023: loss improved from 2.03748 to 2.01618, saving model to
weights-improvement-23-2.0162.hdf5
Epoch 24/35


Epoch 00024: loss improved from 2.01618 to 2.00048, saving model to
weights-improvement-24-2.0005.hdf5
Epoch 25/35
```

```
Epoch 00025: loss improved from 2.00048 to 1.98075, saving model to
weights-improvement-25-1.9808.hdf5
Epoch 26/35


Epoch 00026: loss improved from 1.98075 to 1.96329, saving model to
weights-improvement-26-1.9633.hdf5
Epoch 27/35


Epoch 00027: loss improved from 1.96329 to 1.94800, saving model to
weights-improvement-27-1.9480.hdf5
Epoch 28/35


Epoch 00028: loss improved from 1.94800 to 1.93406, saving model to
weights-improvement-28-1.9341.hdf5
Epoch 29/35


Epoch 00029: loss improved from 1.93406 to 1.91856, saving model to
weights-improvement-29-1.9186.hdf5
Epoch 30/35


Epoch 00030: loss improved from 1.91856 to 1.90564, saving model to
weights-improvement-30-1.9056.hdf5
Epoch 31/35


Epoch 00031: loss improved from 1.90564 to 1.89217, saving model to
weights-improvement-31-1.8922.hdf5
Epoch 32/35


Epoch 00032: loss did not improve from 1.89217
Epoch 33/35


Epoch 00033: loss improved from 1.89217 to 1.87400, saving model to
weights-improvement-33-1.8740.hdf5
Epoch 34/35


Epoch 00034: loss improved from 1.87400 to 1.87325, saving model to
weights-improvement-34-1.8733.hdf5
Epoch 35/35


Epoch 00035: loss improved from 1.87325 to 1.84010, saving model to
```

```
weights-improvement-35-1.8401.hdf5
.\weights-improvement-24-2.0005.hdf5
FULL MODEL
Total Patterns:  163679
Seed:
" he found herself at
last in the beautiful garden, among the bright flower-beds and the
cool
fountain "
e the wood of the tooes har ir the whid tf the was so the soeer. and
then she wes gown and nooked an ier hand  and the sere whin the was
aol ano the girten, and saed to the sore, ''shen soe toanl wase toen i
can tote i saan to the thing!' she said to herself, 'a lont shsh the
war an i mave toued the mooet oase toen i mate wout ier her '
'i whsl to toe theng tou donn ho ' said the monk turtle ang no the
jory. and the sere shing the whit so whrh the tois, 'a lanee hare to
the whit so aalen to the whit so toon the whrt siteree '

'io io toe than the work,' said the monk turtle an the soner.

'ie iour thet toe than ' said the monk turtle an the soner.
and then so see toin in a lorg of the tane of the tooe. and the seie
the was so toek to the tooe, and saed to the sore, ''shen soe toanl
wase toen i can tote i saan to the thing ' she said to herself, ''shan
yhu whal i whink tou doon hn a can taat io a latter
'a dane the whit so toent io ao a far on the gorw,' she said to
herself, 'a lont
Done.
```