

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. 8383

Дейнега В.Е.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы

Создать генеративную модель на основе рекуррентной нейронной сети.

Задание.

- Ознакомиться с генерацией текста.
- Ознакомиться с системой Callback в Keras.
- 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 (TensorBoard).

Выполнение работы

1) Для обработки текст “Алисы в стране чудес” был представлен в виде, подходящем для использования в нейросети. А именно все буквы были переведены в нижний регистр и представлены в виде целых чисел. Текст разбивается на последовательности символов длиной 100 и на основании этих символов предсказывается следующий. Данные нормируются, выходные данные представляются в виде разряженных векторов длиной 47.

2) Создадим модель ИНС согласно методическим указаниям.

```
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')
```

3) Создадим свой Callback, который будет выводить текст, который генерирует нейросеть раз в 3 эпохи.

```
class TextGeneratorLogger(Callback):  
    def on_epoch_end(self, epoch, logs=None):  
        if epoch % 3 == 0:
```


Эпоха 12:

*“and the sas oo the toie she was a little soiee of the sooe of the goore oo the toiee
and e lane the care whr had nete the was anlng to the toiee and aro aroehrs oo the toiee and
ar anl aalut the woide*

*and e lane the mors and thite of the sooe of the sooe of the sooe of the sooe af in sas an inr
hare and toe tait oo the sooe of the sooe of the sooe of the sooe of the sooe of the courd a don't keke
toe toiee*

*she sas anl the woidd the was anl an inc and toe tas anl the woile the was aol an inc and
toen a lote tf the toie af in sas ao the could, and the sas anlng to the toiee an the could a don't keke
the woile '*

*'the sast would to tou to tou tha toee,' she macc terh rorere ronee an aelsein, and eegen to the
toiee an the could and toe tait on the toiee and toe tait on the toiee and toe tait on the toiee and toe
tait on the toiee and toe tait on the toiee and toe tait on the toiee and toe tait on the toiee and toe
tait on the toiee and toe tait on the toiee and toe tai”*

Эпоха 18:

*“whsh the dare an the fad never her aade oo the tooe of the tooe of the rooe of the rooe of the
rooe of the rooe af in wou drol to the table and whsh the doumhune sas soe tiolg the was aoing oo
the tooe of the tooe of the rooe of the rooe of the rooe of the rooe af in wou drol to the table and whsh
the doumhune sas soe tiolg the was aoing oo the tooe of the tooe of the rooe of the rooe of the rooe
of the rooe af in wou drol to the table and whsh the doumhune sas soe tiolg the was aoing oo the
tooe of the tooe of the rooe of the rooe of the rooe of the rooe af in wou drol to the table and whsh
the doumhune sas soe tiolg the was aoing oo the tooe of the tooe of the rooe of the rooe of the rooe
of the rooe af in wou drol to the table and whsh the doumhune sas soe tiolg the was aoing oo the
tooe of the tooe of the rooe of the rooe of the rooe of the rooe af in wou drol to the table and whsh
the doumhune sas soe tiolg the was aoing oo the tooe of the tooe of the rooe of the rooe of th”*

Обученная нейросеть создает такой текст:

*“'stolen!' the king exclaimed, turning to the jury, who instantly made a
memorandum of the fact. ”*

*the hors of the wast wordd the was ao once an incetting of the sabei, and she was aol the wart
oo thre to be a grert to tak at the sooe.*

'whet wou ded" said the manch hare.

'ie cou't had toun mame to thene" said the manch hare.

'iete tou meke toe saal thettent,' said the mock turtle.

'what so tee csrmouse toad to the tou,' said the dat.rpillar at shi whok toenlin, and seet der and mot ano that sare toine to the saad to the seie.'

'whet wou ded" said the katter. 'i mont the white tar oh the boulte oo the shieg of the soie of the soie- and the sere the sabbit was a lottle to tal at the could she was not a motg tame to the sabbit sare

soene io a mott toite and the thieg oo her aeain she sas of the toie tf then the was ao inc to the tabbi

bnt she tas aol thr doon to the white rabbit shrh a lott the was aol a gonue of the tabbit sare soene io a mott aid the had beene to the thet sareng to the karter she had not ii the was aol the wabtid oo tee thae in hhr hand of the ta”

В тексте встречаются реально существующие английские слова, однако их не так много, также присутствуют кавычки, знаки препинания, символы переноса строки.

4) Для отслеживания процесса обучения использовался TensorBoard. График потерь приведен на рис. 2, гистограммы весов и смещений для каждого слоя модели на рис. 1.

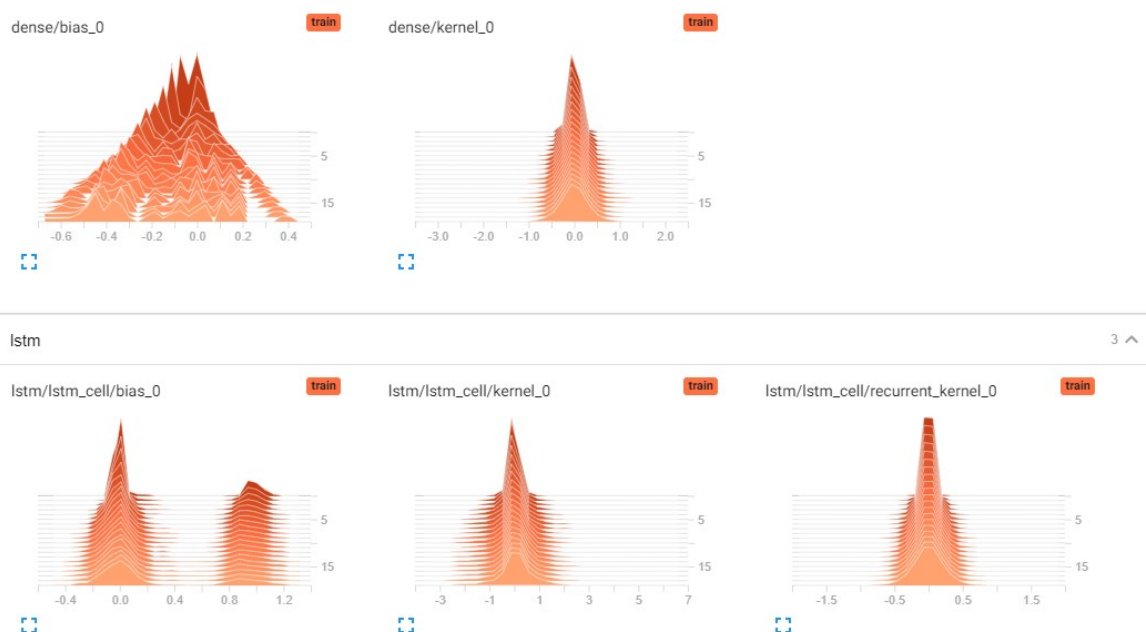


Рисунок 1 – Гистограммы весов и смещений.



Рисунок 2 – График потерь.

Из рисунка 2 видно, что график потерь строгоубывающий, поэтому для уменьшения потерь и получения более точных английских слов обучение ИНС можно продолжить и дойти, например, до 30 эпохи.

Выводы.

В ходе лабораторной работы была реализована генеративная модель на основе рекуррентной нейронной сети, которая создает текст на основе “Алисы в стране чудес”. Также был реализован свой `CallBack`, который выводит промежуточные результаты обучения сети и `CallBack`, позволяющий отслуживать процесс обучения при помощи `TensorBoard`.

Приложение А

```
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 Callback
from keras.callbacks import ModelCheckpoint
from keras.utils import np_utils
from keras.callbacks import TensorBoard

def gen_text(model):
    start = numpy.random.randint(0, len(dataX) - 1)
    pattern = dataX[start]
    result = []
    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])
        seq_in = [int_to_char[value] for value in pattern]
        pattern.append(index)
        pattern = pattern[1:len(pattern)]

    return "".join(result)

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')

class TextGeneratorLogger(Callback):
    def on_epoch_end(self, epoch, logs=None):
        if epoch % 3 == 0:
            text = gen_text(self.model)
            with open("epoch_" + str(epoch) + "_text" + ".txt", "w") as file:
                file.write(text)

tensorboard_callback      =      TensorBoard(log_dir='my_log_dir',      histogram_freq=1,
embeddings_freq=1)

text_generator_logger = TextGeneratorLogger()

callbacks_list = [checkpoint, text_generator_logger, tensorboard_callback]

model.fit(X, y, epochs=20, batch_size=128, callbacks=callbacks_list)

filename = "weights-improvement-20-1.9334.hdf5"

model.load_weights(filename)

model.compile(loss='categorical_crossentropy', optimizer='adam')

print(gen_text(model))

```