

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ
по лабораторной работе №8
Генерация текста на основе “Алисы в стране чудес”
по дисциплине «Искусственные нейронные сети»

Студентка гр. 8382

Ивлева О.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

Цель.

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.

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), в отчете привести результаты и их анализ

Ход работы.

1. Была создана модель рекуррентной нейронной сети. Код программы представлен в приложении А.

2. Был написан callback для генерации нейронной сетью текста во время обучения после заданных эпох.

3. Обучим модель и оценим ход обучения по сгенерированным текстам.

Эпоха 1.

Seed: "again: but he could think of nothing better to say than his first remark, 'it was the best butter,'"

Текст:

[illegible]

Сеть сгенерировала 2 уникальных слова и последовательность из повторяющегося слова “toe”.

Эпоха 5.

Seed: "t with either a waistcoat-pocket, or a watch to take out of it, and burning with curiosity, she ran"

Текст:

so the woet and the woet so the whe sore and the woete to the woete
and the woete to the woete and the woete to the woete and the woete to the woete and
the woete to the woete and the woete to the woete and the woete to the woete and
the woete to the woete and the woete to the woete and the woete to the woete
and the woete to the woete and the woete to the woete and the woete to the woete and
the woete to the woete and the woete to the woete and the woete to the woete and
the woete to the woete and the woete to the woete and the woete to the woete
and the woete to the woete and the woete to the woete and the woete to the woete and

the woete to the woete and the woete to the woete and the woete to the woete
and the woete to the woete and the woete to the woete and the woete to the woete
and the woete to the woete and the woete to the woete and the woete to the woete and
the woete to the woete and the woete to the woete and the woete to the woete and.

Сеть также генерирует повторяющийся набор слов, однако в наборе стало больше слов.

Эпоха 10.

Seed: " the gryphon. Alice did not quite like the look of the creature, but on the whole she thought it w"

Текст:

as toe tiite was anl the woide she was so the caree an the was horn the woide she
coulouse so tee thit would the was so the courouse the woide the was so the caree to
her and the woide toen a lange haree tas an the was ho a lote of the care 'whe mort oo
the canee she gateer reileed. "the woued toine to toe toine ' shiught alice. "the woite
was a little so tee tooee she coulouse so tee shit woule the was so the couro,se the care
and the was hot toee the woide she was so the caree an the was horn the woide she
coulouse so tee thit would the was so the courouse the woide the was so the caree to
her and the woide toen a lange haree tas an the was ho a lote of the care 'whe mort oo
the canee she gateer reileed. "the woued toine to toe toine ' shiught alice. "the woite
was a little so tee tooee she coulouse so tee shit woule the was so the couro,se the care
and the was hot toee the woide she was sothe caree an the was horn the woide she
coulouse so tee thit would the.

Сеть генерирует более разнообразные последовательности слов, также можно увидеть, что она научилась генерировать реплики.

Эпоха 15.

Seed: " wer-pot that stood near. the three soldiers wandered about for a minute or two, looking for them, an "

Текст:

d the woidd soe to the tabbit sored the had not oo the toene shecad nott at the coulo, and tas toe tint hn a lote of the sooe whe horke the was oo the table she cad not oo toe to tee the har hn the har hn the har hn the har hn the har hn the har hn the hard whe had not i sas tou dnen the was so the tooee sh the soreo of the horke of the horke of the horke of the sable she cad not oo toe to tee the har hn the har hn the har hn the har hn the har hn the har hn the hard whe had not i sas tou dnen the was so the tooee sh the soreo of the horke of the horke of the horke of the sable she cad not oo toe to tee the har hn the har hn the har hn the har hn the har hn the har hn the hard whe had not i sas tou dnen the was so the tooee sh the soreo of the horke of the horke of the horke of the sable she cad not oo toe to tee the har hn the har hn the har hn the har hn the har hn the har hn the hard whe had not i sas tou dnen the was so the tooee sh the soreo of the horke of the horke of.

Сгенерирован текст с повторяющимися элементами.

Эпоха20.

Seed: "continued the pigeon, raising its voice to a shriek, 'and just as I was thinking I should be free "

Текст:

of the saal tf the karter a dorso the har on the sooe oo taa the sas oo the saali, and she tert oote then the was so tie tooee sh the courouse, and sas so tie toted the was so the table, and she tert oo the sas sf the hoor, and she celln tas so tee thet whs whs toenk to toe theee an the sanle and arier. 'ie iour ba ir a leteln than ' said the monk, and the teite rabbit weid to the kory, and the wert oome the pooer of the sable and the coultuse sooe the was so the tabli, and shetert oo the hal su tee the sabbit hare sald tha sas of the hard and she cade oo a goorte of the sooer on tee taale, and she tert oote the woodd he har she was so tie tooeo of the sabbit sore, and she sert oo the

sas sf the hoor, and she celln tas so teethe whs whs toenk to toe theee an the sanle and arier. 'ie iour ba ir a leteln than ' said the monk, and the teite rabbit weid to the kory, and the wert oome the pooer of the sable and the coultuse sooe the was so the tabli, and she tert oo the hal su tee.

Сгенерирован текст с повторяющимися элементами, включая реплики.

Выводы.

В результате выполнения данной работы была создана нейронная сеть для генерации текстов на основе “Алисы в стране чудес”.

ПРИЛОЖЕНИЕ А

Исходный код программы. Файл lr8.py

```
import sys

import keras
import numpy as np
from keras.callbacks import ModelCheckpoint
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import LSTM
from keras.models import Sequential
from keras.utils import np_utils

# Loading book:
filename = "wonderland.txt"
raw_text = open(filename).read()
raw_text = raw_text.lower()

# Mapping unique chars to integers:
chars = sorted(list(set(raw_text)))
char_to_int = dict((c, i) for i, c in enumerate(chars))
# Reverse mapping:
int_to_char = dict((i, c) for i, c in enumerate(chars))

# Summarizing data:
n_chars = len(raw_text)
n_vocab = len(chars)
print("Total Characters: ", n_chars)
print("Total Vocab: ", n_vocab)

# Preparing the dataset:
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)

# Reshaping X to [samples, time steps, features]:
X = np.reshape(dataX, (n_patterns, seq_length, 1))
# Normalizing:
X = X / float(n_vocab)
```

```

# One hot encoding the output variable:
y = np_utils.to_categorical(dataY)

def generate_characters(model):
    start = np.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 = np.reshape(pattern, (1, len(pattern), 1))
        x = x / float(n_vocab)
        prediction = model.predict(x, verbose=0)
        index = np.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 GeneratingCallback(keras.callbacks.Callback):
    def __init__(self, epochs):
        super(GeneratingCallback, self).__init__()
        self.epochs = epochs

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

# Defining LSTM 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')

# Defining checkpoint callback:
filepath = "weights-improvement-{epoch:02d}-{loss:.4f}.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1,
save_best_only=True, mode='min')

# Fitting model:
model.fit(X, y, epochs=20, batch_size=128, callbacks=[checkpoint,
GeneratingCallback([0, 4, 9, 14, 19])])

```