

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

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

Студент гр. 7382

Дрозд А.С.

Преподаватель

Жукова Н.В.

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 работы.**

- Ознакомиться с генерацией текста
- Ознакомиться с системой Callback в Keras

### **Требования работы.**

- 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

### **Основные теоретические положения.**

Искусственные нейронные сети - совокупность моделей, которые представляют собой сеть элементов - искусственных нейронов, связанных между собой синаптическими соединениями.

Нейронные сети используются как среда, в которой осуществляется адаптивная настройка параметров дискриминантных функций. Настройка происходит при последовательном предъявлении обучающих выборок образов из разных классов. Обучение - такой выбор параметров нейронной сети, при котором сеть лучше всего справляется с поставленной проблемой.

**Нейрон** — элемент, преобразующий входной сигнал по функции.

**Сумматор** — элемент, осуществляющий суммирование сигналов поступающих на его вход.

**Синапс** — элемент, осуществляющий линейную передачу сигнала.

### **Экспериментальные результаты.**

В ходе выполнения лабораторной работы, была построена модель ИНС, которая генерирует текст, код построенной модели представлен в Приложении А.

Также был реализован собственный Callback, который показывает то как генерируется текст во время обучения, код представлен ниже:

```
class gen_callback(callbacks.Callback):
    def __init__(self, epochs):
        super(gen_callback, self).__init__()
        self.epochs = epochs

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

def gen_sequence(model): # pick a random seed start =
    numpy.random.randint(0, len(dataX) - 1) pattern = dataX[start] 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 =
    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)]
```

Также был отслежен процесс обучения при помощи TensorFlowCallback, в отчете ниже приведены результаты и их анализ.

Номер эпохи	Выведенный результат
1	<p>" n them, after all. "--said  i could not swim--" you can't swim, can you?" he added,  turning to the  kn "  re toe and tee an shid the an</p>
5	<p>" y you have of  putting things!'  'it's a mineral, i think,' said alice.  'of course it is,' said the " caterplllll.  'io sou tou do a sare tou donn ' said the caterplllll.  'io sou tou do a sare tou donn ' said the caterplllll.  'io sou tou do a sare tou donn ' said the caterplllll.  'io sou tou do a sare tou donn ' said the caterplllll.  'io sou tou do a sare tou donn ' said the caterplllll.  'io sou tou do a sare tou donn ' said the caterplllll.  'io sou tou do a sare tou donn ' said the caterplllll.  'io sou tou do a sare tou donn ' said the caterplllll.  'io sou tou do a sare tou donn ' said the caterplllll.  'io sou tou do a sare tou donn ' said the caterplllll.</p>

	<p>'io sou tou do a sare tou donn ' said the caterplllll.</p> <p>'io sou tou do a sare tou donn ' said the caterplllll.</p> <p>'io sou tou do a sare tou donn ' said the caterplllll.</p> <p>'io sou tou do a sare tou donn ' said the caterplllll.</p> <p>'io sou tou do a sare tou donn ' said the caterplllll.</p> <p>'io sou tou do a sare tou donn ' said the caterplllll.</p> <p>'io sou tou do a sare tou donn ' said the caterplllll.</p> <p>'io sou tou do a sare tou donn ' s</p>
10	<p>" t moment she felt a violent blow underneath her chin: it had struck her foot!</p> <p>she was a good deal f "</p> <p>n a lort of the sired hard and the sabbit to the tab in a lottle oo the tab in a</p>

20	<p>" "there is another shore, you know, upon the other side.</p> <p>the further off from england the nearer is "</p> <p>too oo her head in the cooro, and she woodd of the soeer tate to beinn to the tam of the courd, and she jotked at the was oo the wiide and toene the was oo the wiide she was so the lact of the sabbit tal of the tooe and the moee turtle sase the wire the was so tiiek th the war of the courd, and she woodd of the soeer tate to be arreouion to the tam of the courd, and she jotked at the could, and sheould net th the sore of the coort, and she woode of the soeer tate to beinn to the tam of the tooe.</p> <p>'the dirst thing it would bo a latter wore,' she said to herself, and sen gegn to the toier of the coure, and she jotked at the was oo the wiide and toene the was oo the wiide she was so the lact of the sabbit tal of the tooe and the moee turtle sase the wire the was so tiiek th the war of the courd, and she woodd of the soeer tate to be arreouion to the tam of the courd, and she jotked at the could, and sheould net th the sore of the coort, and she woode of the soeer tate to beinn t</p>
30	<p>" Is, of course,' the gryphon replied rather impatiently: 'any shrimp could have told you that.'</p> <p>'if "</p> <p>ioo't hnow the doost tote, she muck turtle seplied vn the karter. 'no you'd aan tou doow that aoy oi the bortse oo the sorp.'</p> <p>'io sou dine to tee ' taid the magchn, wurning to alice, 'n ves a little piterp tooe as ie shotg,</p> <p>" thet aid thet sare the dormouse to she toids of thur sime,</p> <p>wou ortte</p> <p>woa boule.</p> <p>wot</p>

40	<p>" why is a raven like a writing-desk?"</p> <p>'come, we shall have some fun now!' thought alice. 'i'm glad t "</p> <p>fet iev head she said to herself, and she said the horsh th the was oo her ler an ohe was oo herd a aond of the gorse, and she sas oo her lere the care oo hend the cab</p> <p>anicl the mart rare then the was soeie the was soe of the courd to tee whe hatter,</p> <p>aot the codl sas all the cothouse she pabbit tase the was oo herden th the war oo tere the ras oo her hneng to her hn a crrdoue oanee and tound the care</p> <p>the dat saede the was soe oi the courd to the croress tf the katter whsh an any tfate and see white oas shat the houshon was soe oanei tee so her hn a crrdous oare an the was soe of the care</p> <p>and the was soling to thet thet sas she pagt of the rase and the cate tat shet she was soe oaken tfe saed to the waiki te toedk to the say of the way how in the sar oo herd an the sond oo the care ou tene the care</p>
----	---

Существует две основные дихотомии при использовании RNN для текста: символьная и словесная, мы рассматриваем последовательности символов. С каждой новой эпохой текст становится более осмысленным и связным, но всё же многие символы не объединились в существующие слова, поэтому можно прийти к выводу, что необходимо большее количество эпох, чтобы сгенерировать полностью человекопонятный текст, но это потребовало бы большее количество времени.

### Выводы.

В ходе выполнения лабораторной работы была реализована модель ИНС, которая генерирует текст на основе набора данных Приключения Алисы в Стране Чудес Льюиса Кэрролла.. Были изучены зависимости между символами и условные вероятности символов в последовательностях, чтобы мы могли, в свою очередь, генерировать совершенно новые и оригинальные последовательности символов. Был написан собственный Callback, который показывает то, как генерируется текст во время обучения. Также ознакомился с системой Callback в Keras.

## ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД

```
import sys import numpy import tensorflow 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)
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)

class gen_callback(callbacks.Callback):
def __init__(self, epochs):
```



```

super(gen_callback, self).__init__()
self.epochs = epochs

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

def gen_sequence(model): # pick a random seed start =
    numpy.random.randint(0, len(dataX) - 1) pattern = dataX[start] 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
    = 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)]

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') # define
the checkpoint filepath="weights-improvement-{epoch:02d}-{loss:.4f}.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1,
    save_best_only=True, mode='min') callbacks_list = [checkpoint,
    gen_callback([1, 3, 12, 20])] model.fit(X, y, epochs=20, batch_size=128,
    callbacks=callbacks_list)

```