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 _____ Ларин А.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2021

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.

Задание

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

Требования

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

Выполнение

Требуется обучить модель, которая в дальнейшем будет использоваться для генерации текста.
Исходный текст «Алисы в стране чудес» загружается в формате ASCII и подготавливается для обработки нейронной сетью. Для этого сначала текст преобразуется в целые числа. Для этого создается словарь всех уникальных символов, ставя им в соответствие их индексы в сортированном массиве массиве уникальных символов.

```
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))
```

Далее сводка по набору выводится на экран

```
n_chars = len(raw_text)
n_vocab = len(chars)
print "Total Characters: ", n_chars
print "Total Vocab: ", n_vocab
```

Так становится видно, что в тексте 144430, 45 уникальных

Далее текст разбивается на подпоследовательности фиксированной длины в 100 символов. Т.о обучающий надо с состоит из 100 символов на вход и 101-го на выход. Окно шириной в 100(101) символ движется по тексту, порождая 144330 паттернов.

```
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
```

Потом данные паттерны приводятся к виду приемлимому для Keras — форме [образцы, временные шаги, особенности] нормируются и векторизируется — приводится к векторы длинной 45(по количеству символов) с нулями во всех позициях, кроме позиции соответствующей искомому символу.

```
# 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)
```

Затем можно приступить к созданию модели. Она приняла следующий вид:

```
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')
```

Она содержит один скрытый слой LSTM, слой dropout и на конце плотный слой размера по количеству уникальных символов как выходной слой. Решается проблема классификации символов по n классам, потому используется прекрестная энтропия. В качестве оптимизатора используется Адам.
Модель работает медленно, потому используются контрольные точки для фиксирования весов, что бы регистрировать уменьшение потерь.

define the checkpoint

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

Написан Callback который показывает то как генерируется текст во время обучения. Он принимает на вход функцию-генератор, список паттернов, индекс для перевода чисел в символы, а так же количество шагов мжду срабатываниями. В конце эпохи он проверяет шаг и в случае срабатывания вызывает генератор, выводя результат на экран.

Полученные результаты

На самой первой эпохе сгенерировался полностью цикический текст

```
cb: epoch 0/30
Seed:
"t at present--at least i know
who i was when i got up this morning, but i think i must have been
cha"
 the toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe
toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe
toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe
toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe
toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe
toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe
toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe
toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe
toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe
toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe
toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe
toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe toe
toe toe toe toe toe toe toe toe toe toe
```

Потери = 2.9688

Из за присутствия символов разметки не укладывающиеся в общий текст они иногда появляются в выводе. Например текст на эпохе 14:

cb: epoch 14/30
Seed:
" knew that were of the same age as herself, to
see if she could have been changed for any of them.

"
'the war to all the harter whil ' said alice, "that i te thi thit to hese ' she taid to herself, 'i mene to tee thet io h madd to the that woul i seou to taek to the that '

'i mene to het th the sarter ' said the daterpillar.

'ie tou then to toee ' said the mock turtle. "
'   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *

Потери = 2.1570

Минимальные потери =1.6587 на эпохе 39
Получившийся текст:

```
cb: epoch 39/40
Seed:
"n end! 'i wonder how
many miles i've fallen by this time?' she said aloud. 'i must be getting
somewh"
in, and tound to be an a coud wf toredly, in whu aol thry moce to thy ao ince as
the sas no tari to her  she winl hir soacpiog
the care and the queen  and whnt husring ao the could,
'the kueye mu tish th the birthrce of thur mhae thet sale that so hete the sueen
shel she was no thre time to het hend, and whin alice wesught dererpully lo
tiehe ho at all th the other wite, the was anl the juryee and toone the tist sh
the ouher sade to the turen oh the care an thel whs oeg wo be anreot to letel
thet sae in would be aerertirlsy soocr in was the wister of the girten of the
court, 'an  the kuch turtle so hete than that ' she said to herself, 'alice had
boon hn a coeat caak th the sueer sad in ooeee to her eeao, and whsh tiry suoe
thri her head
so the sam and the bormouse of the court, and thin the pabbit was the cirtt te
the cane hnt lere an the was har hnce to her en the whste tore oi the cane,

'i mever say in aalie,  said the hatter. ''ie conrse the birto to the sarter '
the sueen sard
```

Текст почти не содержит посторяющихся паттернов, и в целом похож на английский текст. Ядерный вокабуляр (местоимения, артикли, предлоги, базовые слова) практически везде написаны правильно.

Исследование в TensorBoard

При помощи Callback-а `keras.callbacks.TensorBoard`
Были собраны логи, которые были визуализированы в tensor board.

Изменение функции ошибки представлено на рис. 1

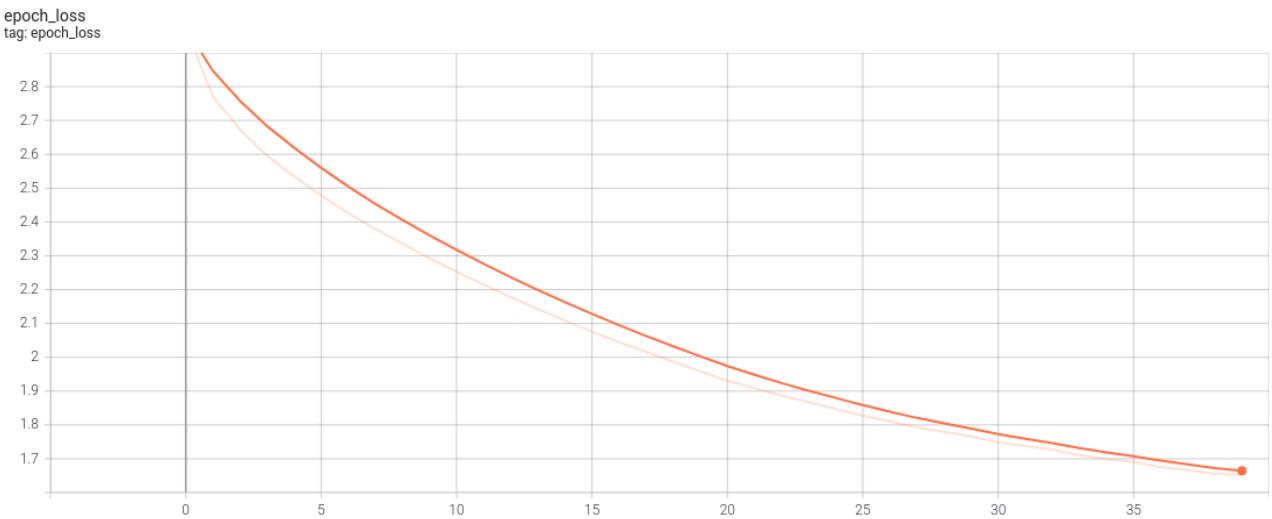

Рисунок 1 — Функция ошибки визуализированная через TensorBoard

На рис. 2 представлены веса и смещения для слоя dense

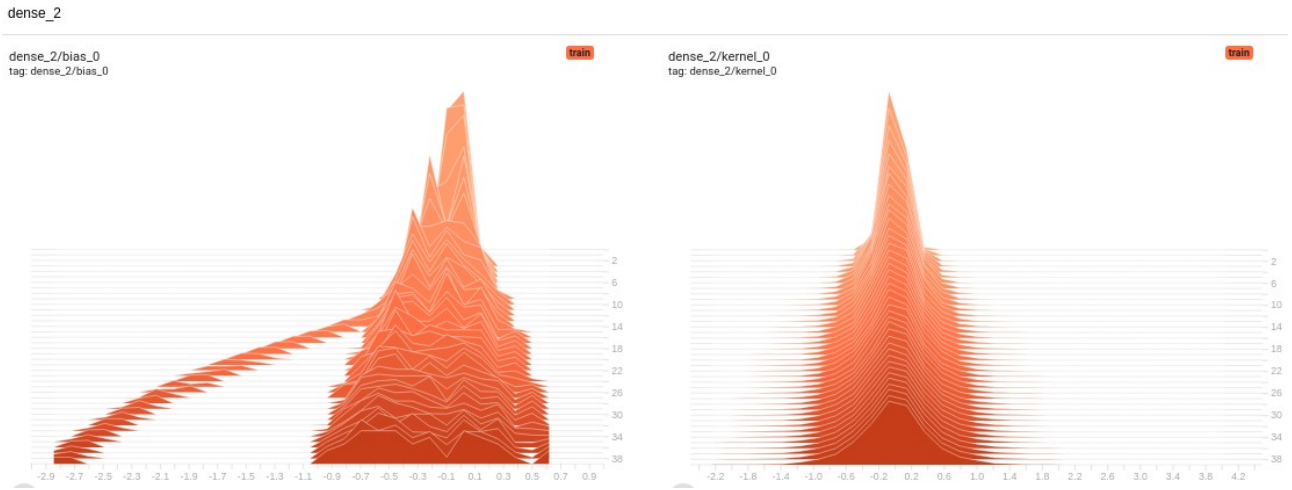

Рисунок 2 — kernel и bias слоя dense

На рис. 3 представлены веса и смещения для слоя LSTM

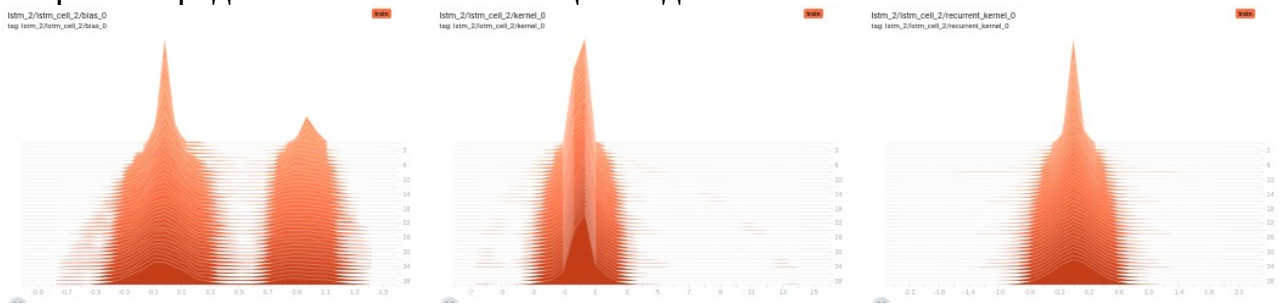

Рисунок 2 — kernel и bias слоя LSTM

Выводы.

Был изучен принцып работы генератиывных нейросетей. Написана нейросеть, обученая на тексте предсказывать следующий символ, с помощью которого был сгенерирован текст. Текст вышел похожим на английский, но не осмысленным и не читаемым. Для оценки работы был написан Callback, вызывающийся в процессе обучения и выводящий на екран пример генерации текста недообученной сетью. Наглядно видны улучшения в процессе.
Были собраны логи обучения при помощи callback-а tensorboard, которые затем были визуализированы. Был использован коллбэк чекпоинтов, сохраняющий веса и позволяющий вернуться к лучшему набору после окончания обучения.