

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
МО ЭВМ

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

Студент гр. 8383

Преподаватель

Костарев К.В.

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.

Постановка задачи.

1. Ознакомиться с генерацией текста
2. Ознакомиться с системой Callback в Keras

Выполнение работы.

Текст «Алисы в стране чудес» необходимо представить в виде, который подходит для использования в нейросети. Так как текст будет генерироваться посимвольно, для сокращения словаря сначала все символы текста преобразовываются в нижний регистр. Затем символы представляются в виде целых чисел. Исходный текст делится на подпоследовательности одинаковой длины. Было решено выбрать последовательности длиной 100 символов. На основании этих 100 символов нейронной сетью будет предсказываться следующий символ. Из последовательностей и следующих за ними символов формируются тензоры входных и выходных данных. Исходные данные преобразуются в форму, подходящую для слоя LSTM. Затем данные нормализуются. Выходные данные представляются в виде векторов длины, совпадающей с размером словаря, с нулями на всех позициях, кроме соответствующей индексу символа в словаре. На выходе модели будем ожидать вектор вероятностей того, что за последовательностью, переданной на вход, будет следовать каждый из символов словаря.

Далее была реализована функция, генерирующая символьную последовательность. Для генерации текста сначала генерируется случайное число от 0 до $\text{len}(\text{dataX})-1$. Здесь $\text{len}(\text{dataX})-1$ – индекс последней последовательности в массиве последовательностей, полученном из исходного текста. Сгенерированное число – это индекс последовательности, на основе которой будет сгенерирован первый символ. Входная последовательность преобразуется в форму, требуемую слоем LSTM, и нормализуется. Затем вычисляется предсказание модели для этой последовательности. Так как модель возвращает вектор вероятностей, запоминается индекс, которому соответствует наибольшее значение вероятности. С помощью словаря по этому индексу восстанавливается символ. Полученный символ добавляется к уже сгенерированной строке. Индекс добавляется к текущей последовательности, а первый элемент последовательности убирается – таким образом, мы сдвигаем «окно», на основе которого предсказывается следующий символ. После того, как таким образом были сгенерированы 1000 символов, функция выводит сгенерированную строку.

Также был реализован собственный CallBack, через заданное число эпох выводящий сгенерированный текст. При инициализации ему передается интервал – количество эпох, через которое необходимо генерировать текст, функция, генерирующая текст, входные данные, откуда функция, генерирующая текст, будет брать начальную последовательность, и словарь, в соответствии с которым предсказанные числа будут переводиться в символы.

Модель состоит из слоя LSTM из 256 нейронов и полносвязного слоя-классификатора с функцией активации softmax. Для модели задается категориальная энтропия в качестве функции потерь, оптимизатор adam и список CallBack'ов:

1. ModelCheckpoint – контрольные точки модели. В конце каждой эпохи сохраняет веса модели и регистрирует уменьшение потерь.

2. TensorBoard – позволяет визуализировать архитектуру, метрики модели, выводит гистограммы активаций и градиентов.
3. Собственный Callback – TextGeneratorCallback, генерирующий текст каждые 3 эпохи.

Модель обучается в течение 10 эпох пакетами по 256 образцов.

После первой эпохи был сгенерирован следующий текст:

ned it, and found in it a very small cake, on which the words
'eat me' were beautifully marked in cu ao ao ao ao ao ao ao ao ao ao
ao
ao
ao
ao
ao
ao
ao
ao
ao
ao ao

Можем видеть, что нейронная сеть заикнулась сразу же после начала предсказания, при этом значение потерь составило 3.0296.

После четвертой эпохи было сгенерировано следующее:

beautiful soup?

pennyworth only of beautiful soup?

beau--ootiful soo--oop!

beau--ootif toe ao c sont toe toet toe toet toe toet toe toet toe toet
toe toet toe toet the toet the toet the toet the toet the toet the
toet the toet the toet the toet the toet the toet the toet the toet
the toet the toet the toet the toet the toet the toet the toet the
toet the toet the toet the toet the toet the toet the toet the toet
the toet the toet the toet the toet the toet the toet the toet the
toet the toet the toet the toet the toet the toet the toet the toet
the toet the toet the toet the toet the toet the toet the toet the
toet the toet the toet the toet the toet the toet the toet the toet
the toet the toet the toet the toet the toet the toet the toet the
toet the toet the toet the toet the toet the toet the toet the toet

the toet the toet the toet the toet the toet the toet the toet the toet the
toet the toet the toet the toet the toet the toet the toet the toet the toet
the toet the toet the toet the toet the toet the toet the toet the toet the
toet the toet the toet the toet the toet the toet the toet the toet the toet
the toet the toet t

После начала предсказания модель опять практически сразу же заиклилась, но величина потерь уменьшилась до 2.7071.

После седьмой эпохи было сгенерировано следующее:

hinking over other
children she knew, who might do very well as pigs, and was just saying
to herself to the toeee
nhe soet the moree the hoote the hare an an an an an an a soee to the toee
,

Модель хоть и стала работать чуть лучше и даже генерировать осмысленные слова и фразы, но в все равно заиклилась. Значение потерь уменьшилось до 2.5365.

В конце концов, после десятой эпохи был сгенерирован следующий текст:

ke it put more
simply--"never imagine yourself not to be otherwise than what it might
appear to othe the toeee to the carter, and the woele tas anlin the toeee tf
the coureos and the more tar ior the toree to the coureon she woele tas anl
toe tar oo the toeee she soeee to the coureon she woele tas anl toe tar io
the cadt, and the more tar ior the toree to the coureon she woele tas anl toe
tar oo the toeee she soeee to the coureon she woele tas anl toe tar io the
cadt, and the more tar ior the toree to the coureon she woele tas anl toe tar
oo the toeee she soeee to the coureon she woele tas anl toe tar io the cadt,
and the more tar ior the toree to the coureon she woele tas anl toe tar oo
the toeee she soeee to the coureon she woele tas anl toe tar io the cadt,
and the more tar ior the toree to the coureon she woele tas anl toe tar oo
the toeee she soeee to the coureon she woele tas anl toe tar io the cadt,
and the more tar ior the toree to the coureon she woele tas anl toe tar oo
the toeee she soeee to the coureon she woele tas anl toe tar io the cadt,
and the more

Модель также заиклилась, но размер повторяющейся области уже больше и в нем появляются более осмысленные фразы и выражения.

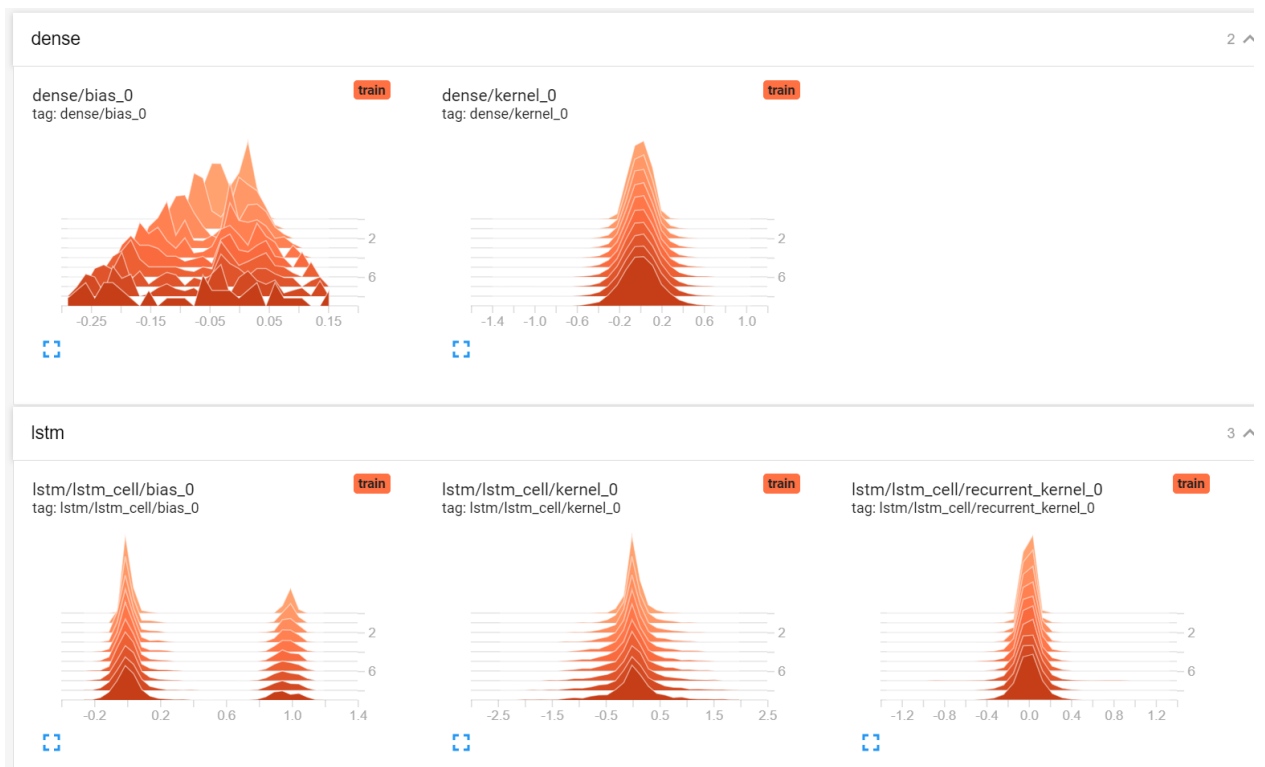
Можно было продолжить обучение с большим числом эпох, но к сожалению мощности устройства, на котором проводились вычисления, не позволяют провести их не занимая огромного количества времени.

Изучим характеристики модели с помощью TensorBoard.

Изменение величины потерь с течением эпох:



Гистограммы весов и смещений для каждого из слоев модели:



Выводы.

В ходе выполнения работы была построена рекуррентная нейронная сеть, с помощью которой генерировались символьные последовательности. Полученные последовательности были не совсем осмысленными, но содержали настоящие английские слова, а также символьные последовательности, похожие на английские слова. В процессе обучения использовались три Callback'а: ModelCheckpoint, сохраняющий веса после каждой эпохи, что позволяет затем использовать веса модели с наименьшими потерями, TensorBoard, позволяющий визуализировать процесс обучения, а также был реализован новый Callback, генерирующий текст недообученной моделью через заданное количество эпох. Благодаря генерации текста в процессе обучения можно было отследить, как улучшалось качество предсказаний: в тексте становилось меньше заикливаний и появлялось больше настоящих слов.