РАЗРАБОТКА РЕКУРРЕНТНОЙ НЕЙРОННОЙ СЕТИ ДЛЯ ГЕНЕРАЦИИ ТЕКСТОВ

Н.А. Кривошеев, И.А. Анфёров, К.В. Вик
Томский политехнический университет
E-mail: nikola0212@mail.ru

Введение

Люди не начинают думать с чистого листа каждую секунду. Читая этот текст, вы понимаете каждое слово, основываясь на понимании предыдущих слов текста. Мы не выбрасываем из головы ранее прочитанную информацию и не начинаем думать с чистого листа. Наши мысли обладают постоянством.

Традиционные нейронные сети не обладают этим свойством, и в этом их главный недостаток. Представим, например, что мы хотим классифицировать одежду, что это за элемент одежды из какой ткани, какого кроя и цвета. Непонятно, как традиционная нейронная сеть могла бы использовать рассуждения о предыдущих полученных данных, чтобы получить информацию о последующих.

Решить эту проблемы помогают рекуррентные нейронные сети (Recurrent Neural Networks, RNN). Recurrent Neural Networks, RNN [1] – рекуррентная нейронная сеть, это вид нейронных сетей, где связи между элементами образуют направленную последовательность. Благодаря этому появляется возможность обрабатывать серии событий во времени или последовательные пространственные цепочки.

LSTM [2] – Долгая краткосрочная память. Разновидность архитектуры рекуррентных нейронных сетей.

Задача

Целью данной работы является разработка рекуррентной нейронной сети RNN. Основная задача данной нейронной сети – научиться генерировать текст [3], напоминающий оригинал, предсказывая последующие символы, т.е. без получения каких-либо данных о грамматике языка, до этих правил, нейронная сеть догадается сама. На вход, для обучения, нейронная сеть будет получать небольшой фрагмент текста на английском языке. После обучения сети она будет выдавать предложения, в виде рассказа, на английском языке.

Ход работы

В ходе разработки рекуррентной нейронной сети RNN используются:

1. Архитектура LSTM (Долгая краткосрочная память) (смотреть рис. 1).
2. Dropout [4] (метод решения проблемы переобучения в нейронных сетях) (смотреть рис. 2)

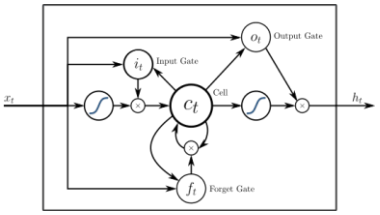

Рис. 1. LSTM блок с входным, выходным и гейтом забывания

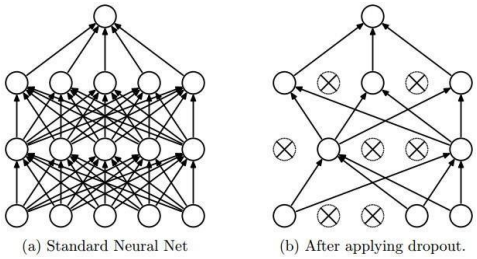

Рис. 2. Dropout – метод решения проблемы переобучения в нейронных сетях

Реализация

Для реализации нейронной сети была использована открытая нейросетевая библиотека Keras. В качестве среды разработки использовался сервис Google Colab, который позволяет работать с Jupiter Notebook'ами в облаке. В качестве метода реализации была выбрана посимвольная генерация, поскольку данная реализация на практике показала себя лучше, чем пословная.

Как было сказано выше, в данной работе модель строится на основе LSTM архитектуры. Основным является LSTM слой, состоящий из 100 нейронов, далее идут полносвязные слои из 75 и 50 нейронов, а также выходной слой.

В качестве функции активации используется гиперболический тангенс [5] и softmax [6]. В качестве оптимизатора используется Adam [7], а в качестве функции ошибки - categorial crossentropy.

Результаты

В результате мы можем получить варианты сгенерированных текстов:

----- Generating text after Epoch: 0

----- Diversity: 0.2

----- Generating with seed: «they seem to be handmade so the flowers are not i»

they seem to be handmade so the flowers are not i was a small and i was a size and the straped and the dress is a size and the short and a little size 4 and the size and it was a little short and the fabric is a size and the small and the size is a

----- Diversity: 0.5

----- Generating with seed: "they seem to be handmade so the flowers are not i"

they seem to be handmade so the flowers are not is so colors out the dress and the size. i thought this top. i was a size and the pants and i don't love the spring is the model is very and beautiful and i tried the small short with so i am 5'4 and a

----- diversity: 1.0

Как видно из результатов генерации, на выходе можно получить сгенерированные слова, которые нейросеть пытается построить в предложения. Данные предложения, как и некоторые слова имеют свои проблемы, и если слова нейросеть строит вполне логичные, то предложения получаются не особо осмысленные.

Обучение нейронной сети

Обучение нейронной сети состоит из следующих этапов:

1. Генерация обучающих и тестовых примеров на основе выборки данных. Данный этап состоит из следующих шагов:

1.1. Взять пример текста из выборки данных;

1.2. Записать 50 символов текста в обучающую выборку как входной пример;

1.3. Записать 51 символ текста в обучающую выборку как желаемый результат;

1.4. Сдвинуть окно считываемых символов текста на шаг N;

1.5. Повторять шаги 1.2 и 1.3 пока не закончится текст примера;

1.6. Повторять шаги 1.1-1.5, пока не закончатся примеры текстов из выборки данных;

2. Обучение нейронной сети на основе сгенерированной обучающей выборки данных;
3. Тестирование нейронной сети на основе сгенерированной тестовой выборки данных.

Генерация текста заключается в последовательности следующих операций:

1. На нейронную сеть подаются первые 50 символов текста, задачей нейронной сети является предсказание 51 символа;
2. Окно подаваемых на нейронную сеть символов текста сдвигается на 1 символ вправо, в результате данное окно включает в себя новый сгенерированный символ;
3. Нейронная сеть предсказывает следующий символ текста;
4. Операции 2 и 3 повторяются, пока не будет сгенерировано заданное количество символов текста.

Сравнение с аналогами

Если сравнивать реализованную модель со схожими рекуррентными сетями [7] можно увидеть, что реализованная нейросеть позволяет получить схожий результат, используя при этом меньшее количество нейронов, и затрачивая меньше времени и ресурсов на обучение.

Если же сравнить с более продвинутыми аналогами [8], реализованная нейросеть проиграет в качестве, однако затраты ресурсов будут в разы больше. Так же стоит учесть, что нейросети, показывающие себя лучше [8], зачастую используют дополнительные доработки, например, это могут быть определенным образом подготовленные входные данные или расширение нейросети алгоритмами с дополнительными правилами.

Заключение

В ходе работы была реализована рекуррентная нейронная сеть для генерации текста, несмотря на недочеты в результатах, сеть способна генерировать осмысленные слова и составлять из них предложения. Возможно, улучшить результат, если доработать модель сети и использовать большее количество нейронов, и обучающую выборку с большим количеством примеров. В данном же случае пришлось идти на компромисс из-за времени, затрачиваемого на обучение, а также вследствие недостатка памяти при обучении нейронной сети.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-08-00977 А и в рамках Программы повышения конкурентоспособности ТПУ.

Список источников

1. “Understanding RNN and LSTM” - URL: https://towardsdatascience.com/understandingrnn-and-lstm-f7cdf6dfc14e Дата обращения: 27.12.19
2. “LSTM - сети долгой краткосрочной памяти” URL: ttps://habr.com/ru/company/wunderfund/blog/331310/ Дата обращения: 10.01.2020
3. “Генерация текста” URL: https://ru.wikipedia.org/wiki/Генератор_текста Дата обращения: 18.12.19
4. “Dropout - методы решения проблем переобучения в нейронных сетях” URL: https://habr.com/ru/company/wunderfund/blog/330814/ Дата обращения: 26.12.19
5. “Функции активации нейросети: сигмоида, линейная, ступенчатая, ReLu, tanh” URL: https://neurohive.io/ru/osnovy-datascience/activation-functions/ Дата обращения: 20.12.19
6. “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION” URL: https://arxiv.org/pdf/1412.6980v8.pdf Дата обращения: 22.12.19
7. “Text Generation With LSTM Recurrent Neural Networks in Python with Keras” URL: https://machinelearningmastery.com/textgeneration-lstm-recurrent-neural-networkspython-keras/ Дата обращения: 26.12.19
8. “Как научить свою нейросеть генерировать стихи” URL: https://habr.com/ru/post/334046/ Дата обращения: 20.10.19