ГЕНЕРАТОР ЭКВИВАЛЕНТНОГО ТЕКСТА НА БАЗЕ LSTM СЕТИ

Гуменников Е. Д., Мурашко И. А.
Кафедра информационных технологий, Гомельский государсвтенный университет имени П. О. Сухого
Гомель, Республика Беларусь
E-mail: guma178@gstu.by

Данная работа посвящена методике применения нейронных сетей LSTM архитектуры для генерации текстов семантически идентичных предоставленным в качестве входных. Здесь рассматриваются особенности реализации подходящей нейронной сети, а также затрагивается методика сбора данных для обучения такой нейронной сети.

Введение

Цель данной работы состоит в адаптации нейросетевых методик автоматической генерации текстов, для решения задачи генерации текстов эквивалентных предоставленным в качестве входных данных. Система способная качественно решать подобную задачу может найти применение в таких отраслях как, разработка чатботов, голосовых интерфейсов, ну и конечно для автоматизации работы «копирайтеров».

Решение данной задач на базе простой рекуррентной нейронной сети имеет вероятно окажется неэффективным. Это оусловлено тем что такая архитектура не имеет механизма «памяти», что накладывает серьезные ограничения на способность такой системы оценивать контекст встречающихся в тексте слов. Однако LSTM сеть обладает такой памятью.

По этой причине, в качестве базовой архитектуры нейронной сети, способной решить поставленную задачу, рассматривается LSTM сеть. Ранее сети данной архитектура показывали наилучший результат в обработке последовательностей и текстов в частности.

I. Методика применения LSTM сети в контексте задачи генерации экивалентных текстов

LSTM является наилучшей архитекторой для решения поставленной задачи. Так как эти ИНС обладают долгосрочной памятью они способны учитывать контекст всего исходного текста и генерировать материалы наиболее близкие исходным. Данная архитектура нейронных сетей, другими словами, способна обрабатывать всю последовательность элементов, рассматривая каждый элемент как часть последовательности, то есть каждое слово рассматривается как часть текста, как исходного, так и генерируемого. Учитывать контекст генерируемого текста также важно, как и исходного, так как новые элементы в генерируемой последовательностью также являются частью текста.

LSTM типа многие ко многим отлично подходят для решения задач NLP, так как исходные и выходные данные являются последовательностями неопределенной длины.

LSTM сети не требуют каких-либо особенных модификаций для их успешного применения в контексте задачи генерации текста эквивалентного заданному.

Для решения поставленной задачи целесообразно применить подход "seq2seq". Исходный оригинальный текст подается на вход LSTM сети, носящей название кодировщик. Выход этой сети является состоянием ячейки, полученным при обработке последнего элемента исходного текста. Это состояние подается в качестве входных данных второй рекуррентной сети, которую называют декодировщиком, ее предназначении состоит в генерации последующего слова эквивалента. Ошибки декодировщика передаются кодировщику через состояние ячейки. Данный вектор в описанной модели называется вектором промежуточного представления. Промежуточное представление используется в популярных моделях предназначенных для решения задач автоматического перевода и, как правило, представляют граф представления интерпретирующий входной текста предназначенный для перевода. Система перевода генерирует выходной текст на основе этой промежуточной структуры. Подобная модель может быть применена и для решения задачи генерации эквивалентных текстов[3].

Генерация эквивалента из оригинального текста начинается с того, что первое слово оригинала подается в качестве входных данных в сеть, где оно вместе с информацией о предыдущих итерациях генерации следует к сигмоидальному слою, который называется «фильтром слоя забывания». Он определяет то, какую часть информации, содержавшейся в состоянии ячейки следует «забыть». Так можно окинуть семантическую информацию, которая вероятнее всего более не понадобится при генерации эквивалента. Математически этот шаг описан в как:

$$f_t = \sigma\left(W_t\left(h_{t-1}, x_t\right) + b_f\right)$$

где W_t – веса нейронных связей;

h_{t-1} – предыдущее слово сгенерированного эквивалента;

x_t – очередное слово оригинала;

b_f – величина смещения.

Далее необходимо по текущему состоянию ячейки и новоприбывшему слову оригинала определить то, какую часть новой информации следует добавить к состоянию ячейки, то есть «запомнить». Этот этап состоит из двух шагов. Во-первых, сперва сигмоидальный слой, называемый «слоем входного фильтра» определяет то, какие значения в состоянии ячейки следует обновить, затем тангенс слой строит вектор значений кандидатов на добавление к состоянию ячейки. Так сеть может запомнить ключевые слова и семантическую информацию исходного и генерируемого текста, для ее последующего употребления при генерации. Математическая интерпретация описанных процессов:

$$i_t = \sigma\left(W_i\left(h_{t-1}, x_t\right) + b_i\right)$$

где W_i – веса нейронных связей;

h_{t-1} – предыдущее слово сгенерированного эквивалента;

x_t – очередное слово оригинала;

b_i – величина смещения.

$$C_t^{'} = than\left(W_C\left(h_{t-1}, x_t\right) + b_c\right)$$

где W_C – веса нейронных связей;

h_{t-1} – предыдущее слово сгенерированного эквивалента;

x_t – очередное слово оригинала;

b_C – величина смещения.

Обновление состояния ячейки таким образом происходит по следующему сценарию:

- старое состояние ячейки умножается на f_t, таким образом из состояния ячейки удаляется ненужная более информация;
- к состоянию ячейки прибавляется произведение $C_t^{'}$ и i_t.

Таким образом в состояние ячейки поступает новая информацию о сгенерированном и об оригинальном тексте. Математически это можно записать в виде:

$$C_t = f_t C_{t-1} + i_t C_t^{'}$$

где C_t – обновленное состояние ячейки;

f_t – коэффициент забывания;

C_{t-1} – предыдущее состояние ячейки;

i_t – вектор входного фильтра;

$C_t^{'}$ – вектор кандидатов на добавление к состоянию ячейки.

Далее необходимо сгенерировать новое слово эквивалентного текста. Для этого анализируется текущее состояние ячейки. Сперва применяется сигмоидальный слой, определяющий какую информацию из состояния ячейки необходимо применять. Затем состояние ячейки обрабатывается тангенс слоем, возвращающим вектор значений, величина которых находится в пределе от -1 до 1. Затем оба вектора перемножаются таким образом получается вектор значений кодирующий продолжение генерируемого текста. Затем результат отправляется в следующую итерацию, где будет обработано следующее слово оригинала. Также отправляется и новое состояние ячейки, таким образом формируется долгосрочная память о генерации и оригинальном тексте, что позволяет добиться наилучших результатов.

II. Сбор данных для обучения LSTM сети-генератора эквивалентов

Для обучения подобной LSTM сети необходим огромный объем обучающих данных. Каждый элемент обучающей выборки представляет из себя пару оригинал-эквивалент.

Собрать такие данные удобно с помощью нескольких переводов зарубежной литературы выполненных разными авторами.

Предполагается выбирать фрагменты из двух экземпляров текстов содержащие идентичную семантическую нагрузку. Один фрагмент станет оригиналом второй будет примером эквивалента.

Полностью автоматизировать процесс сбора данных не представляется возможным, так как авторы двольно своевольно форматируют текст.

1. Барто, Э. Обучение с подкреплением / Э. Барто // ДМК. – 2020. – С. 216–232.
2. Розанов, А. К. Быстрый алгоритм анализа словоформ естественного языка с трехуровневой моделью словаря начальных форм / А. К. Розанов. // РГРУ. – 2016. – С. 14.
3. Хайкин, С. Нейронные сети. Полный курс / С. Хайкин. // Вильямс. – 2006. – С. 912–987.
4. Тарик, Р. Создаем нейронную сеть / Р. Тарик. // Вильямс. – 2018. – С. 12–67.