С. Д. Костюк

АЛГОРИТМ ВЕКТРОНОГО ПРЕДСТВАЛЕНИЯ СЛОВ ДЛЯ РЕШЕНИЯ ЗАДАЧ РАСПОЗНАВАНИЯ РЕЧИ

Рассматривается реализация алгоритма word2vec на выходе которого получаются векторные представления слов. Векторы слов лежат в основе многих систем обработки естественного языка (NLP), захлестнувших современный мир (Amazon Alexa, Google translate и т.д.) .

Введение

Для решения задачи обработки и разпознования речи предлагается использовать алгоритм word2vec который создает вектора слов, и подбирает наиболее подходящее слово на основе контекста.

I. Принципы действия алгоритмов word2vec

Создать кортежи данных в формате [входное слово, выходное слово], каждое слово представлено в виде двоичного вектора длины n, где i-ое значение кодируется единицей на i-ой позиции и нулями на всех остальных (one-hot кодировка); Создать модель, которая на вход и выход получает one-hot векторы; Определить функцию потерь, предсказывающую верное слово, чтобы оптимизировать модель; Определить качество модели, убедившись, что похожие слова имеют похожие векторные представления. Синее поле обозначает входной one-hot вектор (целевое слово), красное поле — выходной one-hot вектор (любое слово в контекстном окне за исключением целевого слова, так называемое контекстное слово). Из одного контекстного окна получаются два элемента данных (на одно целевое слово приходится два соседних). Размер окна обычно определяется пользователем. Чем больше размер контекстного окна, тем лучше наша модель, но это влияет на время выполнения алгоритма.

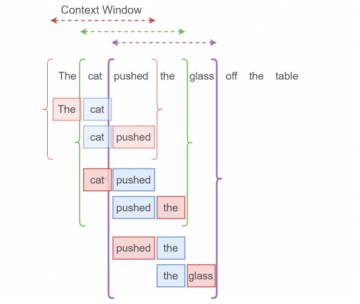

Рис. 1 – входные данные

II. Нейронная сеть,Embedding layer

Embedding layer хранит вектора всех слов в словаре.Это огромная матрица размера [число слов в словаре x размерность пространства сжатого векторного представления слов]). Эта гигантская матрица инициализируется случайным образом (как и нейросеть) и настраивается бит за битом в процессе оптимизации.

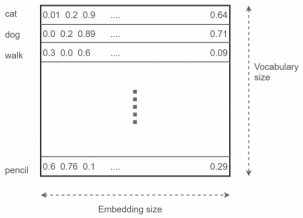

Рис. 2 – Embedding layer

В процессе обучения нейросеть получает входной вектор и пытается предсказать результат в виде распределения вероятностей слова быть в контексте входного слова на множестве всех слов (также его можно интерпретировать как линейную комбинацию one-hot кодировок этих слов). Затем с помощью функции потерь мы штрафуем модель за неправильную классификацию и награждаем за верную. Для данного введенного слова (целевого слова) найдем соответствующий вектор из embedding layer; Скормим этот вектор нашей нейросети, затем попытаемся предсказать правильное выходное (контекстное) слово; Сравнив предсказанное слово и то слово, которое на самом деле находится в контекстном окне, вычислим функцию потерь; Используя функцию потерь, оптимизируем нейросеть и embedding layer.

III. Выводы

Предлогаемый алгоритм поможет скорректировать распознанный текст, выбрать слово если при распознании или записи возникли помехи которые помешали корректно обработать слово. Так же этот алгоритм используется для предоставления слов на выбор пользователю при печати на клавиатуре телефона и т.д.

Костюк Сергей Дмитриевич, студент кафедры систем управления БГУИР, s.kostyuk11@gmail.com.

Научный руководитель: Захарьев Владимир Анатольевич, преподователь кафедры систем управления БГУИР доктор технических наук, доцент, zahariev@bsuir.by.