

- Технологий. Лаборатория знаний, 2006. – 285 с
2. Канер, С. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений// 2005. – №4. – с47-52
 3. UIAutomationFundamentals [Электронный ресурс]. – Режим доступа: [https://msdn.microsoft.com/en-us/library/ms753107\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms753107(v=vs.110).aspx)
 4. Тестирование, управляемое данными [Электронный ресурс]. – Режим доступа: <http://abap-blog.ru/osnovy-abap/testirovanie-upravlyаемое-dannymi-data-driven-testing/>

МОДЕЛЬ SKIP-GRAM ТЕХНОЛОГИИ WORD2VEC

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Давыдовский С.В.

Сиротко С.И. – канд. физ.-мат. наук, доцент

Технология word2vec с момента своего появления в 2013 году [1] обрела большую популярность среди исследователей в области машинного обучения. Векторные представления слов, получаемые с помощью word2vec, нашли широкое применение в различных областях обработки естественного языка. Такое распространение обусловлено тем, что векторы word2vec довольно точно передают семантические значения слов. В данной работе рассматривается принцип работы одной из моделей word2vec – Skip-Gram.

Основная идея word2vec – слова, находящиеся в похожих контекстах, являются семантически близкими. Для вычисления векторного представления слов используется искусственная нейронная сеть. Во время обучения сети с помощью модели Skip-Gram формируются оптимальные векторы для каждого слова.

Модель Skip-Gram позволяет предсказывать близлежащие слова на основании центрального слова. Задача нейронной сети заключается в следующем: для заданного слова вычислить у всех остальных слов в словаре вероятности их появления рядом с этим словом. Под словом “рядом” понимается фиксированный размер окна, к примеру 5 слов слева и 5 слов справа.

Для обучения нейронной сети используются большие наборы текстов. В качестве тренировочных примеров используются пары “центральное слово – контекстное слово”. На рисунке 1 показаны примеры тренировочных данных полученные из предложения “The quick brown fox jumps over the lazy dog.” при размере окна, равном двум (центральное слово выделено синим) [2].

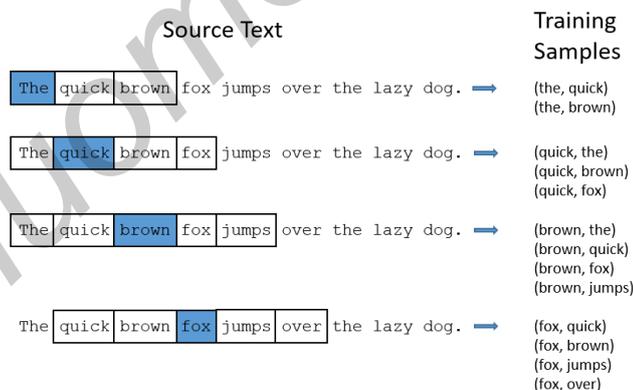


Рис. 1 – Примеры тренировочных данных

Предположим, что в наборе текстов находится 10000 уникальных слов. Каждое слово изначально представлено как one-hot вектор $\mathbf{l} \in \mathbb{R}^{10000}$. Выходной слой нейронной сети формирует вектор $\mathbf{O} \in \mathbb{R}^{10000}$, содержащий вероятности появления каждого из 10000 слов рядом с входным словом. Архитектура нейронной сети представлена на рисунке 2.

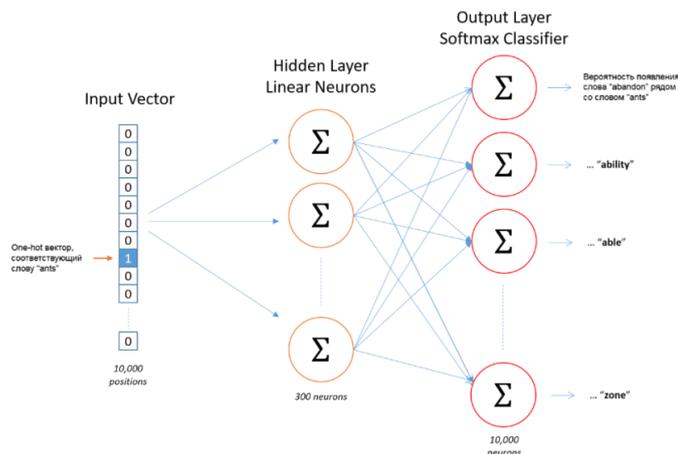


Рис. 2 – Архитектура нейронной сети, использующей модель Skip-Gram

Матрица весов скрытого слоя имеет размерность 10000x300. Каждая строка этой матрицы является векторным представлением соответствующего слова, причем приоритетной целью является не результат работы сети как таковой, а “обученные” (полученные в результате тренировок) веса скрытого слоя. В приведенном примере в скрытом слое находится 300 нейронов, соответственно векторные представления слов будут иметь 300 компонентов. Это число является гиперпараметром и определяется индивидуально для каждой задачи. Входной one-hot вектор используется для выборки соответствующего векторного представления слова из матрицы весов, на котором затем применяется softmax-функция нейронов выходного слоя.

В результате, если два слова имеют схожий контекст, то модель для этих двух слов должна вывести схожие результаты. Единственный способ этого добиться – иметь схожие векторные представления этих слов. При реализации на практике обычно применяют дополнительные механизмы, такие как иерархический софтмакс (Hierarchical Softmax) и/или негативное сэмплирование (Negative Sampling) [3].

Полученные векторные представления слов образуют векторное пространство, где векторы, обозначающие схожие по смыслу слова, расположены рядом друг с другом. Кроме того, имеют место различные интересные отношения между векторами. Например:

$$v_{queen} - v_{woman} + v_{man} \approx v_{king}$$

Из этого примера видно, что векторы word2vec позволяют определять нетривиальные взаимосвязи между словами. Поэтому они эффективны при решении многих задач обработки естественного языка.

Список использованных источников:

1. Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space
2. Word2Vec Tutorial – The Skip-Gram Model [Электронный ресурс], 2017. – Режим доступа: <http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model>. Дата доступа: 02.04.2017.
3. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality.

IOS-ПРИЛОЖЕНИЕ С ГОЛОСОВЫМИ ЗАМЕТКАМИ NOTEVOX

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Линник Г.И.

Жвакина А.В. – канд. техн. наук, доцент

Важнейшими из требований к современным программным продуктам являются удобство их использования, способность улучшать нашу повседневную жизнь. Также особое внимание в обществе уделяется лицам с ограниченными возможностями, например, имеющими дефекты зрения. Разработанное приложение позволяет экономить время за счет автоматизации записи речевых сообщений.

Цель работы: создание мобильного приложения для операционной системы iOS по записыванию голосовых заметок для автомобилистов, поэтов, людей с ограниченными возможностями, тех, кому неудобно записывать и тратить много времени на написание заметок вручную.

Описание работы мобильного приложения: