УДК 004.8

Нейросетевый метод снятия омонимии

Нгуен Нгок Зиеп, Ле Мань Ха - Московский Физико-Технический Институт

1 Введение

В лексической системе русского языка есть слова, которые звучат одинаково, но имеют совершенно разные значения. Такие слова называются лексическими омонимами, а звуковое и грамматическое совпадение разных языковых единиц, которые семантически не связаны друг с другом, называется омонимией [1].

Например:

1. Ключ(1) – «Родник» (Студеный ключ)

   Ключ(2) – Стальной ключ для отпирания и запирания замка

2. Лук(1) – «Растение» зеленый лук

   Лук(2) – «оружие» тугой лук

В отличие от многозначных слов лексические омонимы не обладают предметно-семантической связью, т.е. у них нет общих семантических признаков, по которым можно было бы судить о полисемантизме одного слова. Полная лексическая омонимия – это совпадение слов, принадлежащих к одной части речи, во всех формах. Пример: наряд(1) – «одежда», наряд(2) – «распоряжение»; Замок(1) – «123», Замок(2) – «здание»; Автоматическая система перевода делится на несколько этапов, одним из которых является морфологический. На этом этапе для каждого слова определяются морфологические характеристики: род, число, падеж, склонение, и т.п., а также начальная форма слова (Лемма). Процесс морфологической разметки осложняется омонимами [1].

2 Методы представления текста

В настоящее время теория и практика машинного обучения переживают настоящую «глубинную революцию», вызванную успешным применением методов deep learning, представляющих собой третье поколение нейронных сетей. Сети, обученные с помощью алгоритмов глубинного обучения, не просто превзошли по точности лучшие альтернативные подходы, но и в ряде задач проявили зачатки понимания смысла подаваемой информации (например, при распознавании

изображений, анализе текстовой информации и т.п.). Наиболее успешные современные промышленные методы компьютерного зрения, распознавания речи и машинного перевода построены на использовании глубинных сетей, а гиганты IT-индустрии, такие как Apple, Google, Facebook, скупают коллективы исследователей, занимающихся глубинным обучением целыми лабораториями.

Представление слова как вектор - в настоящее время одна из самых интересных областей исследований в глубинном обучении, хотя данный подход был изначально введен Bengio [2] и др. более десяти лет назад. Самые известные модели представления слова как вектор – "Continuous-Bag-of-Word" и "Skip-Gram", описание представленое Mikolov[3] и реализация моделей Word2vec [4], опубликованая на сайте Google project, которая привлекла большое внимание в последние два года. Эти модели использовали простую нейронную сеть для отображения слов, которые ближе к слову по значению. Поэтому процесс обучения для этих моделей быстро работает с большими данными.

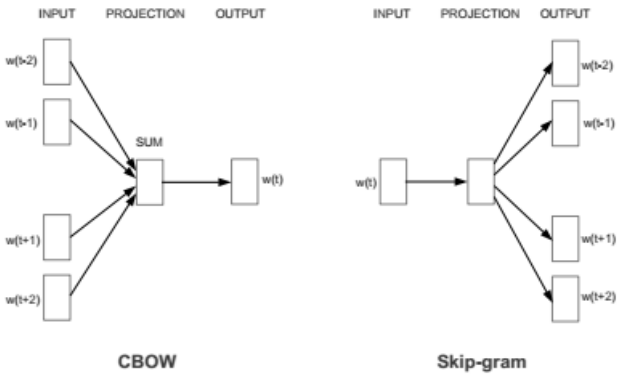

Рис. 1: Модели CBOW и Skip-gram.

CBOW модель – Использует окружающие слова (предложение без заданного слова), что бы предугадать заданное слово. Входные вектора в нейронную сеть W(t-2),W(t-1), W(t+1), W(t+2), а выходной вектор из сети это заданное слово W(t). В результате обучения получается 2 вектора представления слова. Входной вектор V - это вектор представления предложения описания слова, выходной вектор V' – Это вектор представления слова. SKIP-GRAM модель – Использует слово, что бы предугадать предложение описания заданного слова. Эти модели могут использоваться во многих задачах анализа естественного языка, таких как классификация документов, распознавание спама или машинный перевод.

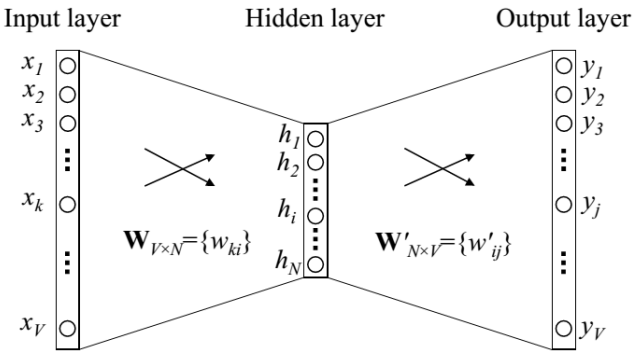

Рис. 2: Нейронная сеть, использованная для получения вектора представления слова. Входной слой и выходной имеют V нейронов, а скрытый имеет N нейронов.

На рисунке 3 отображены вектора для чисел и животных на английском и испанском языках [5], на рисунке можно легко заметить, что эти понятия имеют схожие геометрические композиции. Причина в том, что что понятия, которые основаны на реальном мире разделились одинаково, как и для множества распространенных языков.

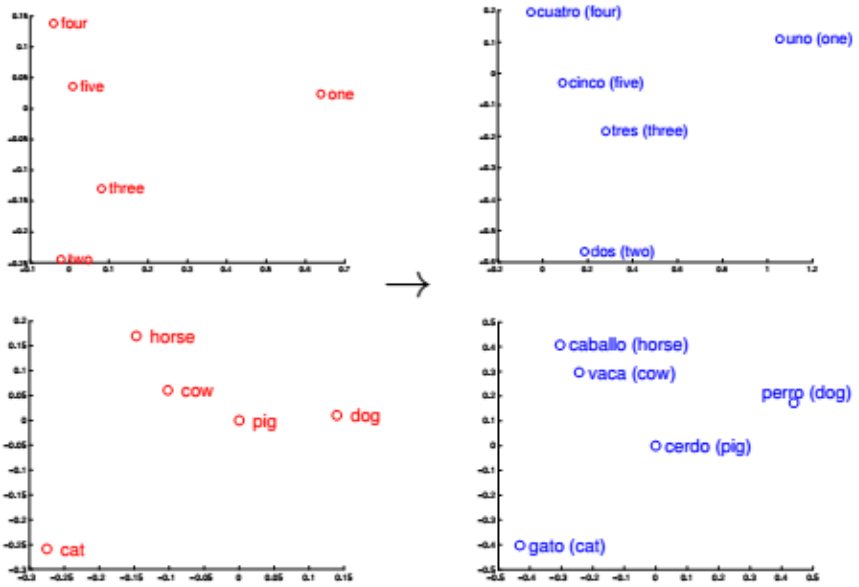

Рис. 3: Распределение векторных представлений слов чисел и животных на английском языке (слева) и испанском (справа). Пять векторов на каждом языке были спроектированы на два измерения с использованием метода PCA. Видно, что эти концепции имеют схожие геометрические представления в обоих пространствах, что позволяет предполагать возможное точное линейное отображение из одного пространства в другое.

Один хороший пример двуязычного вхождения слова, приводится в работе Socher и др. [6]. Было показано как можно научиться вставлять слова из двух разных языков в одно общее пространство. В рассмотренном в работе случае, учились вставлять английские и китайские слова в том же пространстве. Если значения слов окружающих омонимы разные, тогда расстояние между омонимами тоже большое.

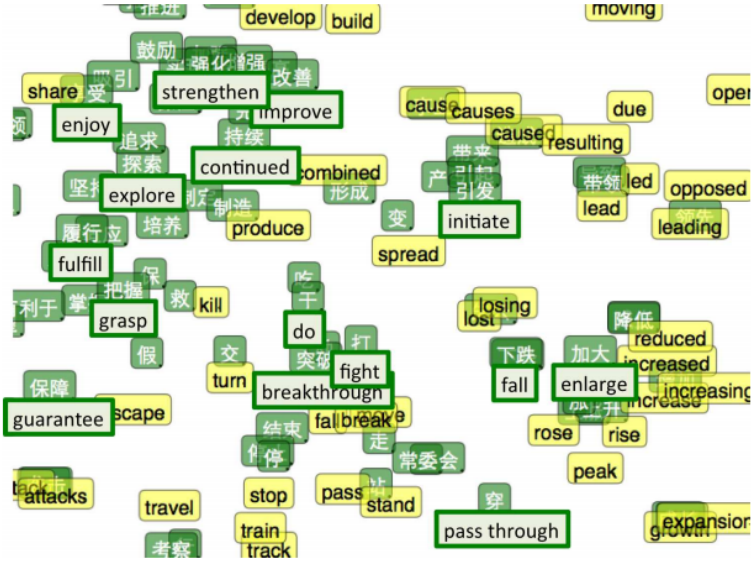

Рис. 4: Визуализация двуязычного пространства - Английский и китайский.

3 Нейронные сети и Автоэнкодер

Нейронные сети прямого распространения часто используются для обучения с учителем и используются, например, для классификации [7].

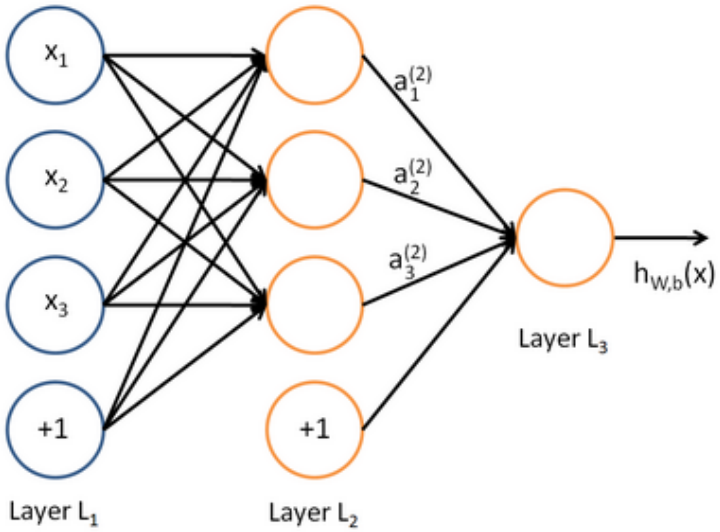

Рис. 5: Многослойная нейронная сеть

Часто используемая функция активации - сигмоида:

$$f(z_j) = \sigma(z_j) = (1 + e^{-z_j})^{-1} \tag{1}$$

Где:

$$z_j = \sum w_{ij} x_i \tag{2}$$

Полезное свойство сигмоиды - её производная функции:

$$\frac{\partial \sigma}{\partial z} = \sigma(z)(1 - \sigma(z)) \tag{3}$$

Обучение такой нейронной сети производится обычно методом обратного распространения ошибки таким образом, чтобы минимизировать среднеквадратическую ошибку сети на обучающей выборке. Таким образом, обучающая выборка содержит пары векторов признаков (входные данные) и эталонных векторов (маркированные данные) (x, y).

Метод обратного распространения ошибки:

Для выходного нейрона:

$$\delta = z - y \tag{4}$$

Для нейронов скрытых слоев:

$$\delta_i = \sum_{j=0}^{n} \delta_j w_{ij} \tag{5}$$

Коррекция весов:

Для выходного нейрона:

$$w'_{i0} = w_{i0} + \eta \delta y_i \tag{6}$$

Для нейронов скрытых слоев:

$$w'_{ij} = w_{ij} + \eta \delta_j y_j (1 - y_j) y_i \tag{7}$$

В реальной практике, маркированных данных очень мало, для них требуется много сил и времени. Автоэнкодер представляет собой алгоритм обучения без учителя, который использует нейронную сеть и метод обратного распространения ошибки для того, чтобы добиться того, что входной вектор признаков вызывал выход сети, равный входному вектору, т.е. y = x. Автоэнкодер является специальной архитектурой искусственных нейронных сетей, позволяющей применять обучение без учителя при использовании метода обратного распространения ошибки. Простейшая архитектура автоэнкодера — сеть прямого распространения, без обратных связей, наиболее схожая с перцептроном и содержащая входной слой, скрытый слой и выходной слой. В отличие от перцептрона, выходной слой автоэнкодера должен содержать столько же нейронов, сколько и входной слой [8].

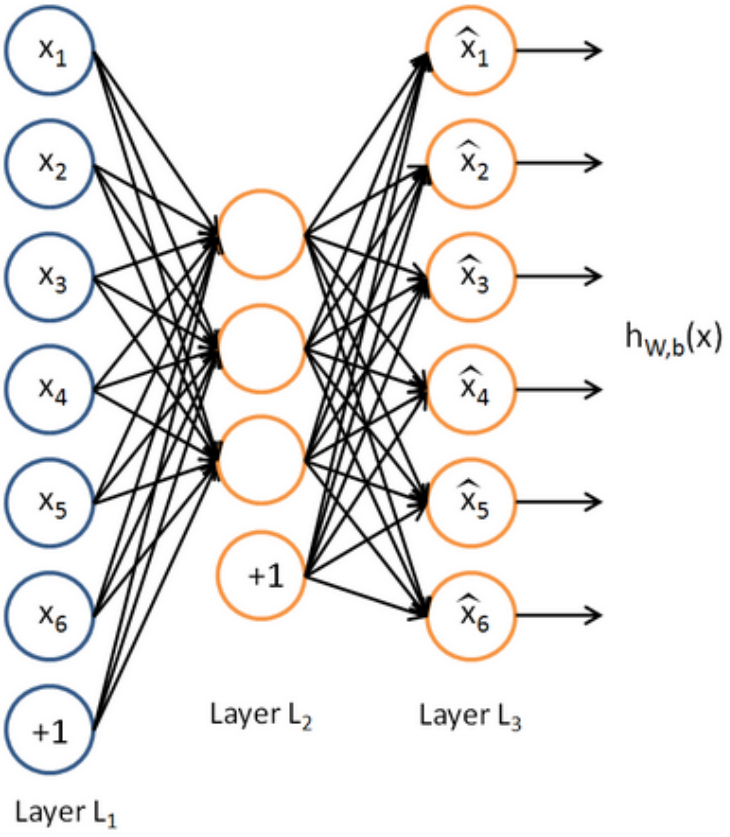

Рис. 6: Автоэнкодер

Цель автоэнкодер - чтобы выход нейронной сети был наиболее близким к входному вектору. Для того, чтобы решение этой задачи было нетривиальным, на топологию сети накладываются особые условия:

1. Количество нейронов скрытого слоя должно быть меньше, чем размерность входных данных.

2. Активация нейронов скрытого слоя должна быть разреженной.

Первое ограничение позволяет получить сжатие данных при передаче входного сигнала на выход сети. Такая сжатие возможно, если в данных есть скрытые взаимосвязи, корреляция признаков или структура. Второе ограничение – требование разреженной активации нейронов скрытого слоя, — позволяет получить нетривиальные результаты даже когда количество нейронов скрытого слоя превышает размерность входных данных. Будем считать нейрон активным, когда значение его функции передачи близко к 1 и наоборот, неактивным если значение его функции передачи близко к 0. Разреженная активация – это когда количество неактивных нейронов в скрытом слое значительно превышает количество активных.

Эти ограничения заставляют нейросеть искать обобщения и корреляцию в поступающих на вход данных, выполнять их сжатие. Таким образом, нейросеть автоматически обучается выделять из входных данных общие признаки, кото-

рые кодируются в значениях весов сети. Необходимо чтобы средняя активация каждого скрытого нейрона приняла значение, наиболее близкое к заданному разреженному параметру (порядка 0.05). Для этого был добавлен в каждый нейрон скрытого слоя параметр разреженности ρ:

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^{m} \left[a_j^{(2)}(x^{(i)}) \right] \tag{8}$$

Необходимо чтобы средняя активация каждого скрытого нейронна приняла значение, наиболее близкое к ρ:

$$\hat{\rho}_j = \rho \tag{9}$$

Штрафная функция:

$$S = \sum_{j=1}^{s_2} KL(\rho|\hat{\rho}_j) \tag{10}$$

$$KL(\rho|\hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j} \tag{11}$$

Производная штрафной функции:

$$\frac{\partial KL(\rho|\hat{\rho}_j)}{\partial \rho_j} = -\frac{\rho}{\hat{\rho}_j} + \frac{1-\rho}{1-\hat{\rho}_j} \tag{12}$$

4 Рекурсивный автоэнкодер и векторное представление текста

В данном разделе рассмотрим как представит текст в виде числового вектора. Для решения задачи классификации и других задач с помощью нейронных сетей, вход должен иметь фиксированную длину, а длинна текста является произвольной. Для фиксирования размера входов нейронных сетей используется метод векторного представления текста с помощью рекурсивного автоэнкода - входной и выходной слои которого имеют 2K нейронов, а скрытый слой - K нейронов [9]:

Автоэнкодер объединяет два слова x1, x2 (вектор длины 2K) в один вектор y (длина K):

$$y = f(W^{(1)}[x1, x2] + b^{(1)}) \tag{13}$$

Этот процесс повторяется N-1 раз для текста длинной N. В результате получается конечный вектор - семантическое векторное представление текста, этот вектор используется как вход для системы обучения.

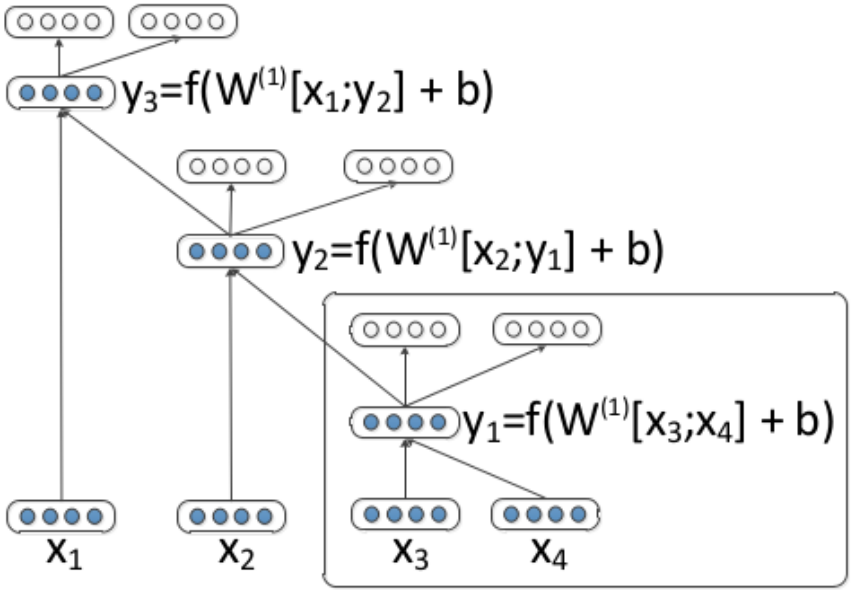

Рис. 7: Автоэнкодер

5 Кластеризация значений слова

5.1 K-means

K-means - наиболее популярный метод кластеризации. Был изобретён в 1950-х годах математиком Гуго Штейнгаузом и почти одновременно Стюартом Ллойдом. Особую популярность приобрёл после работы Маккуина. Действие алгоритма таково, что он стремится минимизировать суммарное квадратичное отклонение точек кластеров от центров этих кластеров [10]:

$$J(V) = \sum_{j}^{c} \sum_{j}^{c_i} (x_i - v_i)^2 \tag{14}$$

Следующим шагом является принятие каждую точку, принадлежащую к данному набору данных и связать его в ближайший центр. В этот момент нужно пересчитать K новые центра, как барицентра кластеров в результате предыдущего шага. После того как вычислили эти новые K центроиды, новую привязку должно быть сделано между теми же набора данных точек и ближайшей нового центра. Цикл был сформирован. В результате этого цикла можно заметить, что K-центры меняют шаг за шагом местоположения пока не прекращаются изменения или другими словами центры больше не двигаются.

5.2 Маркировка значений слова

В первых получить все возможные вектора слов, затем вектора используются, чтобы построить вектора контекстов с применением автоэнкодер[Session 4]. Например у нас есть миллион векторов описания о слове Замок, необходимо выполнять вычисления по алгоритму K-Means для этих векторов. В результате, можно маркировать каждое слово с значением. Таким образом, все омонимы будут маркированы по значению, тогда можно использовать эти результаты для задачи снятия омонимий.

Список литературы

[1] Омонимы в русском языке http://gramma.ru/

[2] Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. In Journal of Machine Learning Research, pages 1137–1155.

[3] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality."Advances in neural information processing systems. 2013.

[4] Word2Vec Project https://code.google.com/p/word2vec/

[5] Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever. "Exploiting similarities among languages for machine translation."arXiv preprint arXiv:1309.4168 (2013).

[6] Zou, W. Y., Socher, R., Cer, D. M., Manning, C. D. (2013). Bilingual Word Embeddings for Phrase-Based Machine Translation. In EMNLP (pp. 1393-1398)

[7] Stanford UFLDL tutorial, Multilayer neural networks
http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/

[8] Stanford UFLDL tutorial, Autoencoders
http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/

[9] Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions, Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, Christopher D. Manning, 2011

[10] Алгоритм K-means https://ru.wikipedia.org/wiki/K-means