**Word2Vec**

Существует несколько видов преобразования слов в числа. Один из них - Word Embeddings

(Векторное представление слова — вещественный вектор в пространстве с фиксированной невысокой размерностью.

Вход — коллекция текстов.

Выход — векторные представления слов из словаря коллекции).

Одна из ее наиболее популярной разновидностей - нейросеть Word2Vec. Она имеет две реализации: Skip-gram и CBOW (Сontinuous bag-of-words).

**Skip-gram**

Состоит из трех слоев:

- Входной слой, который принимает одно слово в формате one-hot. Суть one-hot encoding заключается в том, что слово кодируется бинарным вектором с одной единицей, которая представляет позицию слова в словаре. Например:

{ехала, по, дороге, машина}

[1, 0, 0, 0] - ехала

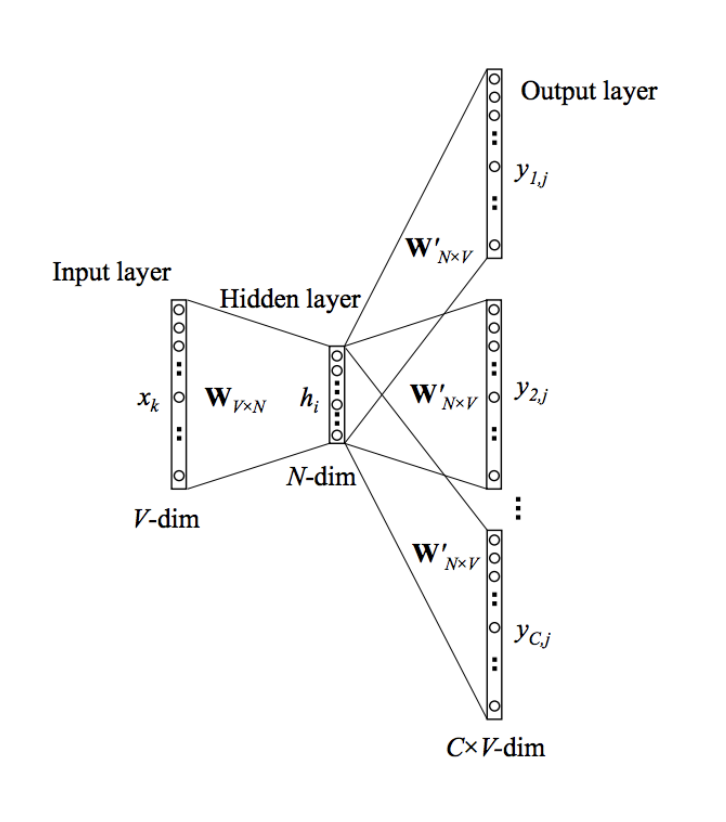
[0, 1, 0, 0] - по

[0, 0, 1, 0] - дороге

[0, 0, 0, 1] - машина

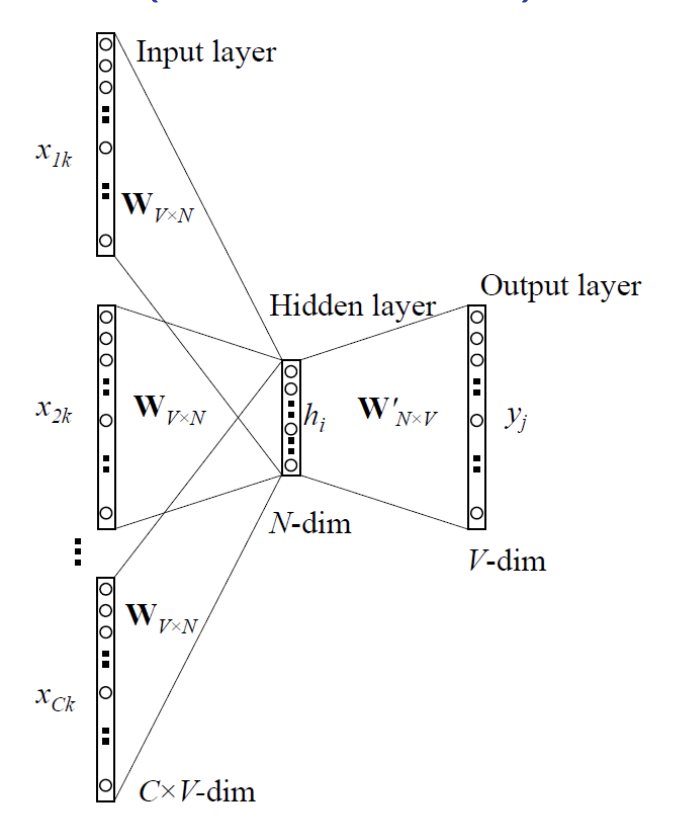
- Слой Embedding, который представляет собой матрицу размером NxP, где N– размер словаря, P – гиперпараметр, который подбирается эмпирически. Обычно P присваивают значение 300.

- Выходной слой с размером Nx1, где N – размер словаря. Это единственный слой, который имеет функцию активацию (softmax). Каждый из нейронов этого слоя выдает вероятность того, что входное слово принадлежит соответствующему контексту (другим словам).



**CBOW**

Архитектура CBOW является зеркальным отражением Skip-gram, когда входной и выходной слой меняются местами: на вход подается контекст (множество слов), а модель предсказывает слово, подходящее этому контексту. Слой Embedding остается тем же.



**Представление входного и выходного векторов через контекстное окно**

После создания словаря необходимо выбрать входное слово и контекст к нему. Контекст - ближайшие слова, образованные в зависимости от размера контекстного окна. Например, есть предложение “Сегодня по дороге ехала большая красная машина”. Выбираем слово “ехала” в качестве входного слова и окно с размером 2. Тогда имеем по два контекста слева и справа:

(ехала, красная), (ехала, большая).

(по, ехала), (дороге, машина).

Число контекстных слов зависит от количества предложений и размера окна. Алгоритм word2vec ищет все предложения с входным словом и контекстом около него. Исходя из нашего предложения, можно составить входной и выходной векторы для одного слова:

{сегодня, по, дороге, ехала, большая, красная, машина}

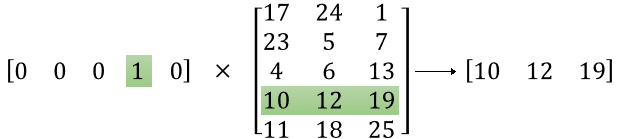
X\_1 = [0, 0, 0, 1, 0, 0, 0] - входное слово “ехала”

Y\_1 = [0, 1, 1, 1, 1, 1, 0] - контекст (слова “сегодня” и “машина” в него не входят)

Именно Y\_1 является тем вектором, с которым сравниваются результаты выходного слоя.

**Извлечение из слоя Embedding одной строки**

Входной вектор (one-hot) умножается на матрицу Embedding:



В итоге, из всей большой матрицы NxP выбирается только одна строка-вектор, которая и является векторным представлением слова (word embedding). Эта строка-вектор посылается на выходной слой, каждое соединение с нейроном которого имеет свои веса.

**Phrase Learning**

Это означает, что некоторые слова стоит рассматривать вместе, например, “New\_York”. Слово “New” в некоторых случаях может обозначать “новый”, но если “New” и “York” стоят вместе, то, скорее всего, имеется в виду “Нью-Йорк”. Также это поможет различать людей: “Петр\_Сидоров” от “Петр\_Козлов”.

**Subsampling**

Он подразумевает избавление от слишком повторяющихся слов. Предлоги, союзы могут быть в каждом контексте, но не раскрывать его смысл. Для каждого слова вычисляется вероятность того, что оно должно учитываться в обучении. Такая вероятность вычисляется следующим образом:

\[ P(w_i)=\frac{0.001}{f(w_i)}\sqrt{\frac{f(w_i)}{0.001}} \]

где – w\_i слово, а f(w\_i) – частота встречаемости этого слова в корпусе.

Таким образом, чем она больше, тем выше вероятность того, что слово не имеет информативной ценности.

**Negative Sampling**

Он необходим для уменьшения вычислительных затрат на обучение. Выходной слой имеет размерность N, равный размеру словаря. Если словарь содержит миллион слов, то и обновлять веса для каждого нейрона слишком затратно. Поэтому обновление весов можно осуществлять только для контекстных слов и 5-6 слов, которые не совпадают с контекстом (для большого датасета можно ограничиться 2-3). Кроме того, Negative sampling можно заменить на Hierarchical softmax, который разворачивает сеть в бинарное дерево, обновляя log(N) весов вместо N весов.