Deep Learning for Natural Language Processing

- NLP problems employed shallow machine learning models and time-consuming, hand-crafted features.
- Recent popularity and success of word embeddings neural-based models have achieved superior results.

Deep Learning for Natural Language Processing

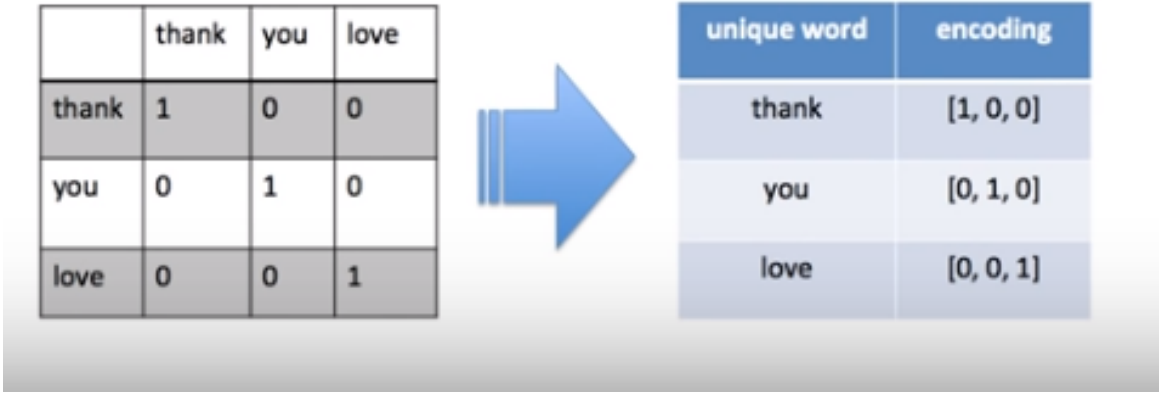
- Can text be input in Deep Learning?
  - NO
- Can number be input in Deep Learning?
  - Yes

What is word encoding?

Convert text to number

Vector Representation of Text

What is One Hot Encoding ? Convert text to Vector



	thank	you	love
thank	1	0	0
you	0	1	0
love	0	0	1

unique word	encoding
thank	[1, 0, 0]
you	[0, 1, 0]
love	[0, 0, 1]

One Hot Encoding doesn't have **similarity**

unique word	encoding
thank	[1, 0, 0]
you	[0, 1, 0]
love	[0, 0, 1]

One Hot Encoding doesn't have **similarity**

cosine similarity also 0 since angle is 90 degree

unique word	encoding
thank	[1, 0, 0]
you	[0, 1, 0]
love	[0, 0, 1]

Embedding

Embedding is dense vector with similarity

unique word	encoding	embedding
king	[1, 0, 0, 0]	[1, 2]
man	[0, 1, 0, 0]	[1, 3]
queen	[0, 0, 1, 0]	[5, 1]
woman	[0, 0, 0, 0]	[5, 3]

Word2vec approach to represent the meaning of word

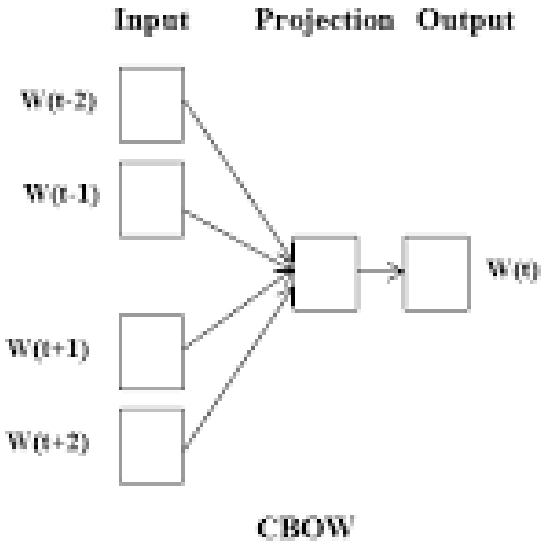
- Represent each word with a low-dimensional vector
- Word similarity = vector similarity
- Key idea: Predict surrounding words of every word

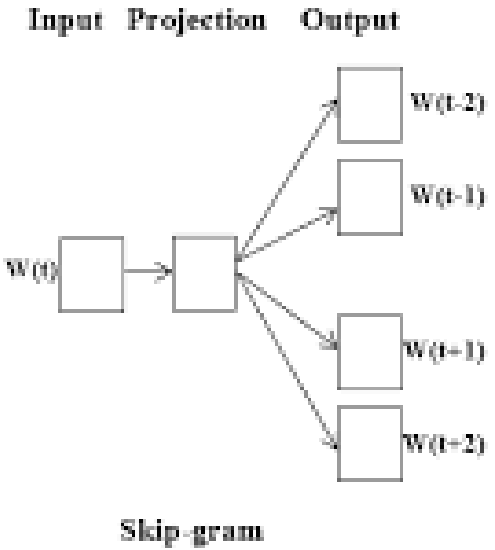
Represent the meaning of word – word2vec

There are 2 basic neural network models:

- Continuous Bag of Word (CBOW): use a window of word to predict the middle word
- Skip-gram (SG): use a word to predict the surrounding ones in window.

The Continuous Bag Of Words and the Skip Gram Model

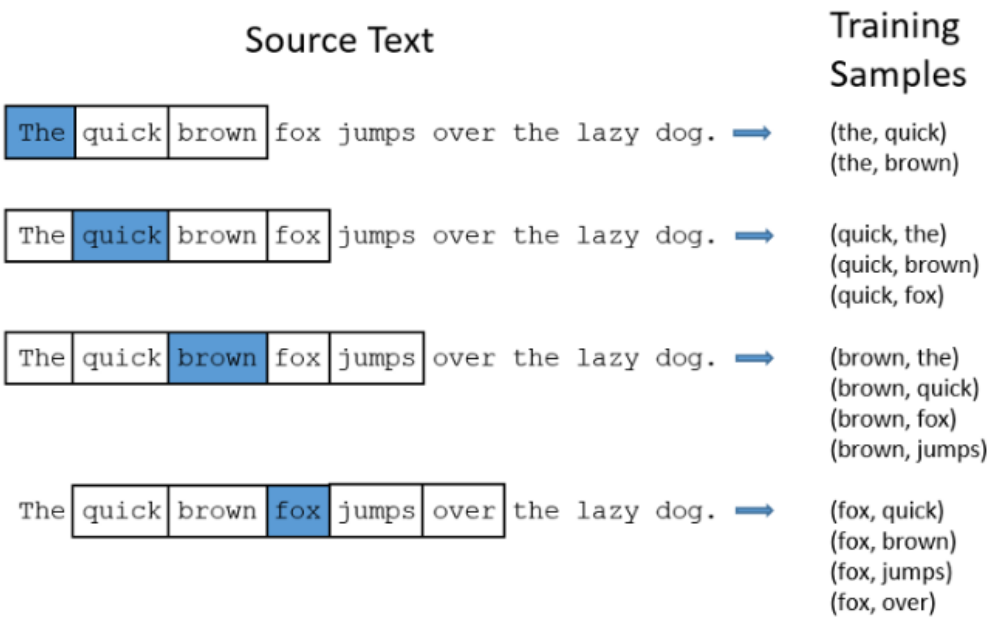




How word2vec works:

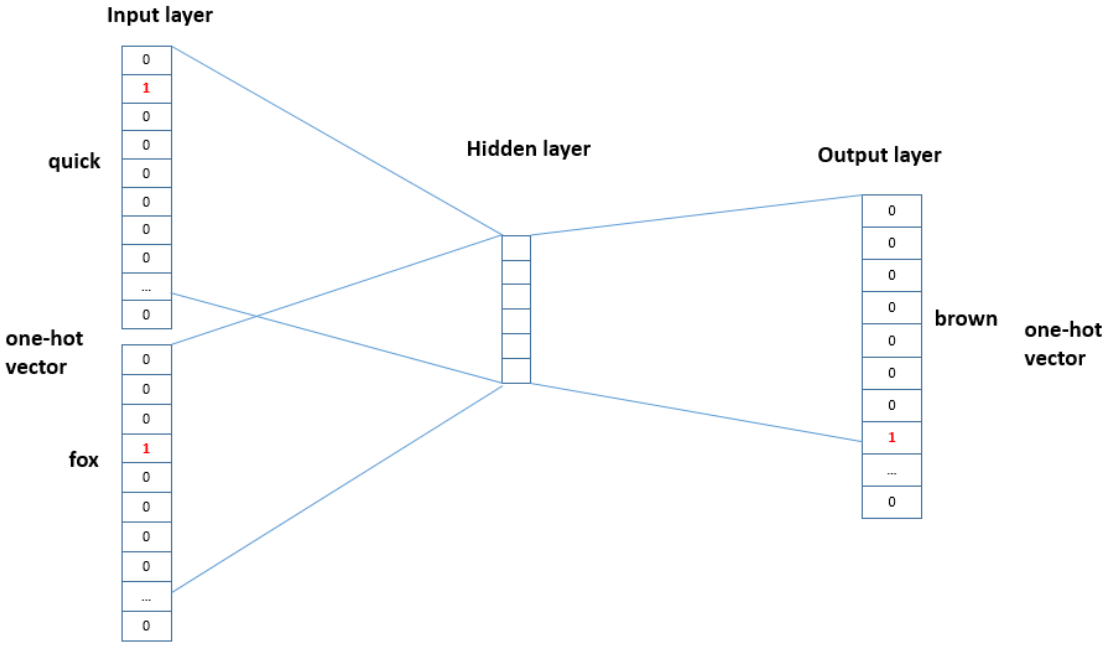
- Take a 3 layer neural network. (1 input layer + 1 hidden layer + 1 output layer)
- Feed it a word and train it to predict its neighbouring word.
- Remove the last (output layer) and keep the input and hidden layer.
- Now, input a word from within the vocabulary. The output given at the hidden layer is the 'word embedding' of the input word.

How word2vec works:

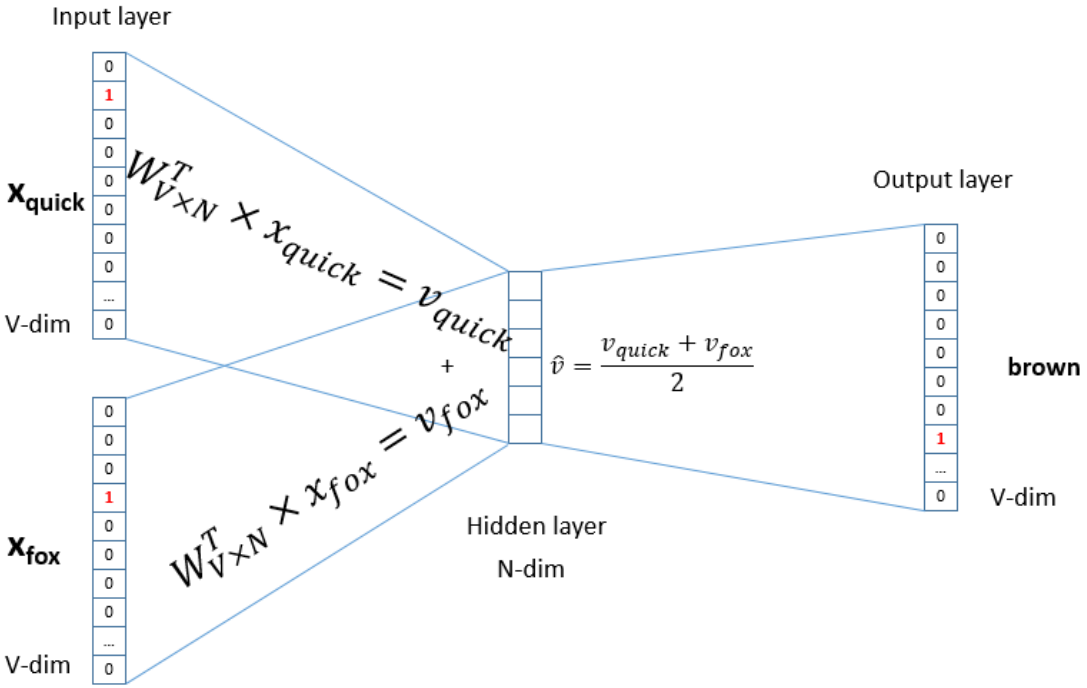


A training sample generation with a window size of 2.

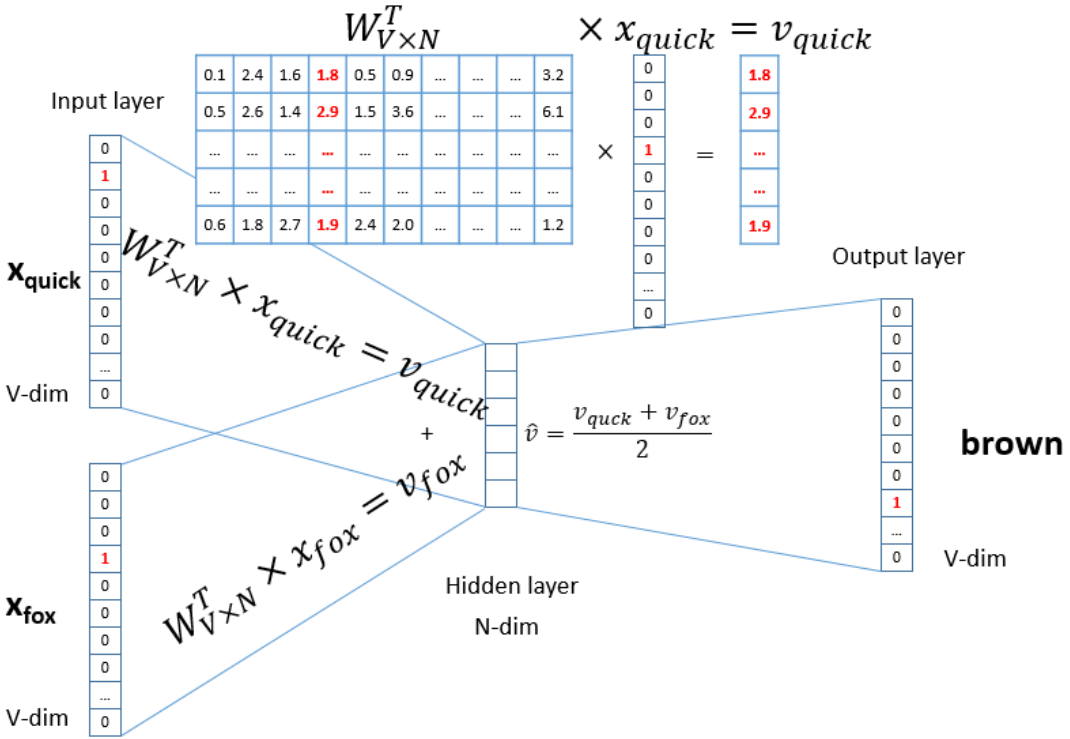
How word2vec works:



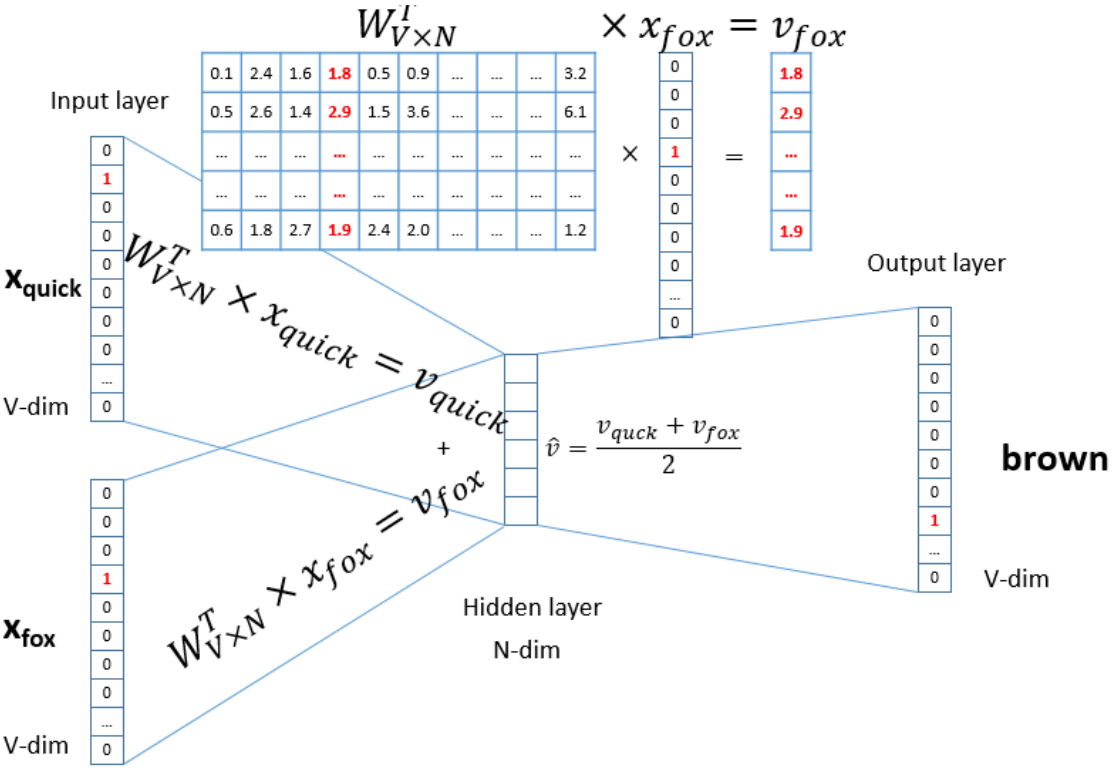
How word2vec works:



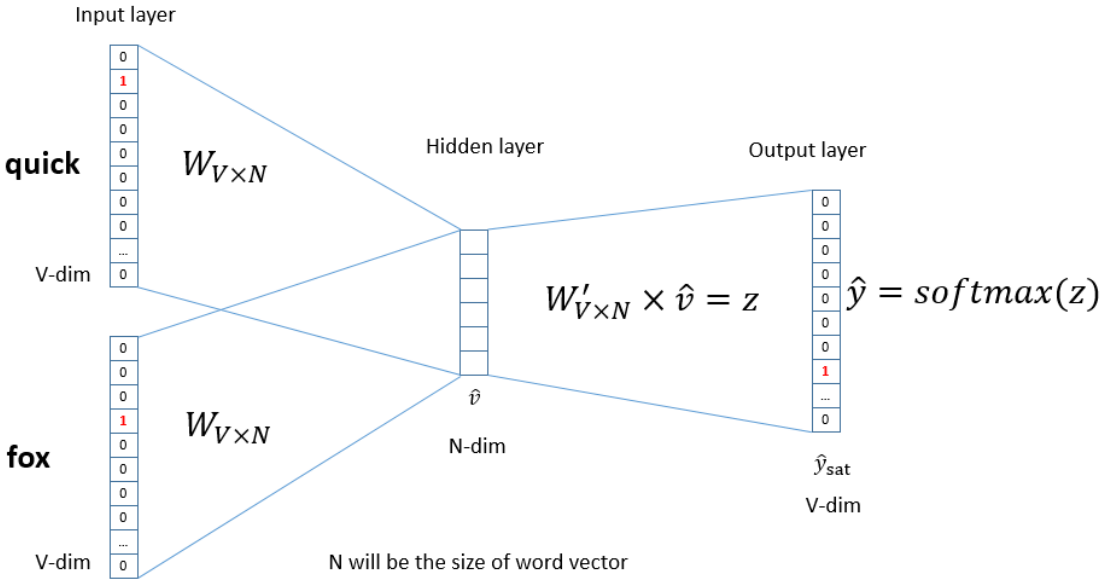
How word2vec works:



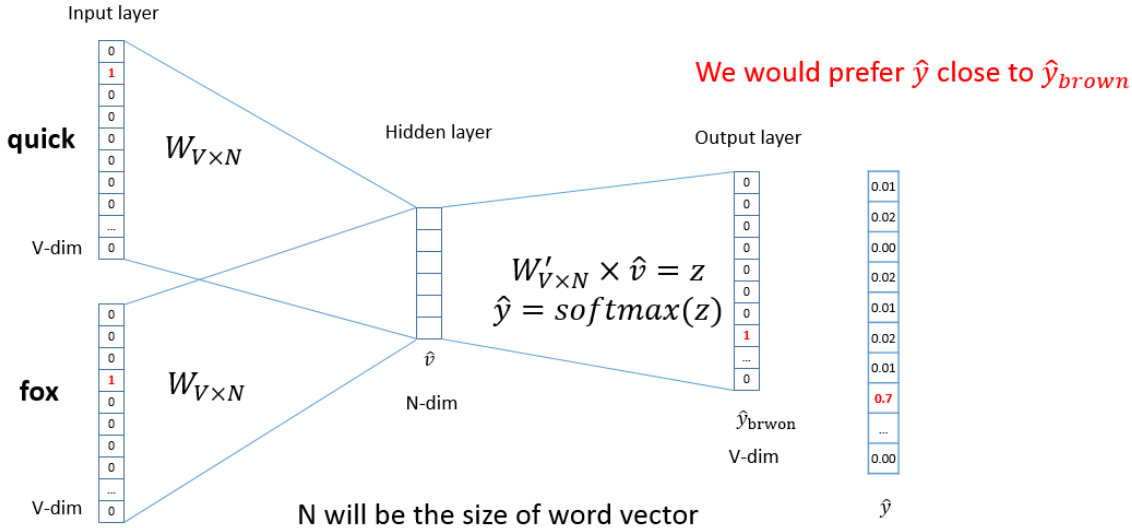
How word2vec works:



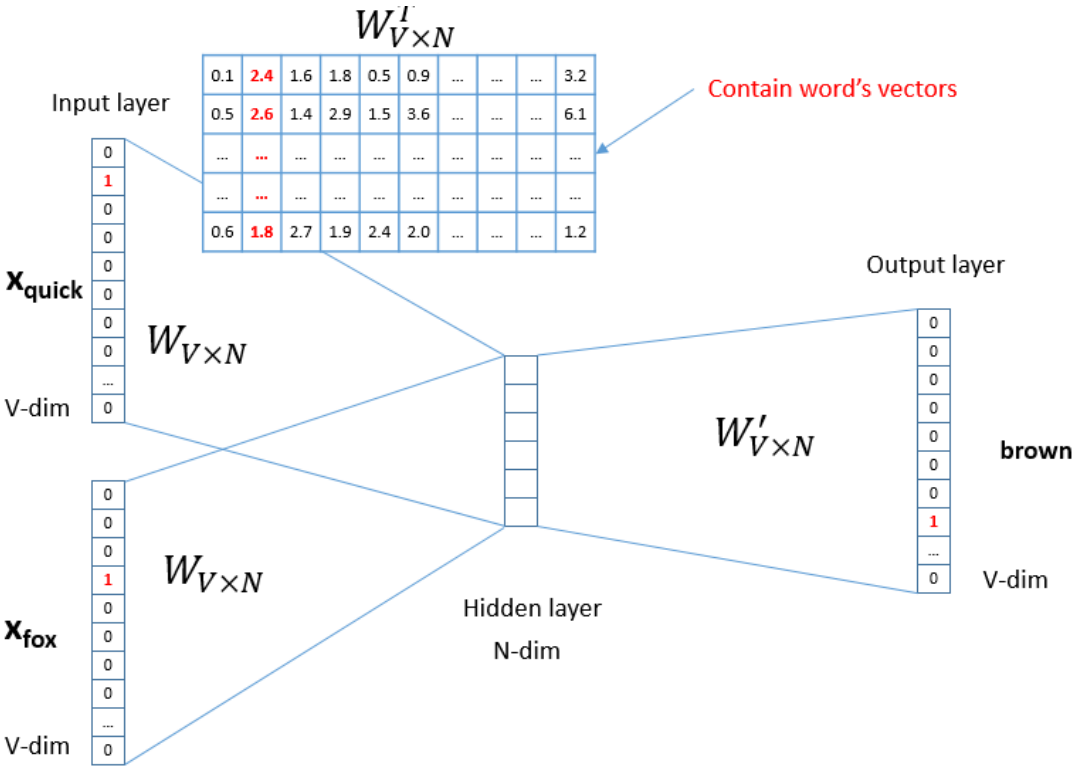
How word2vec works:



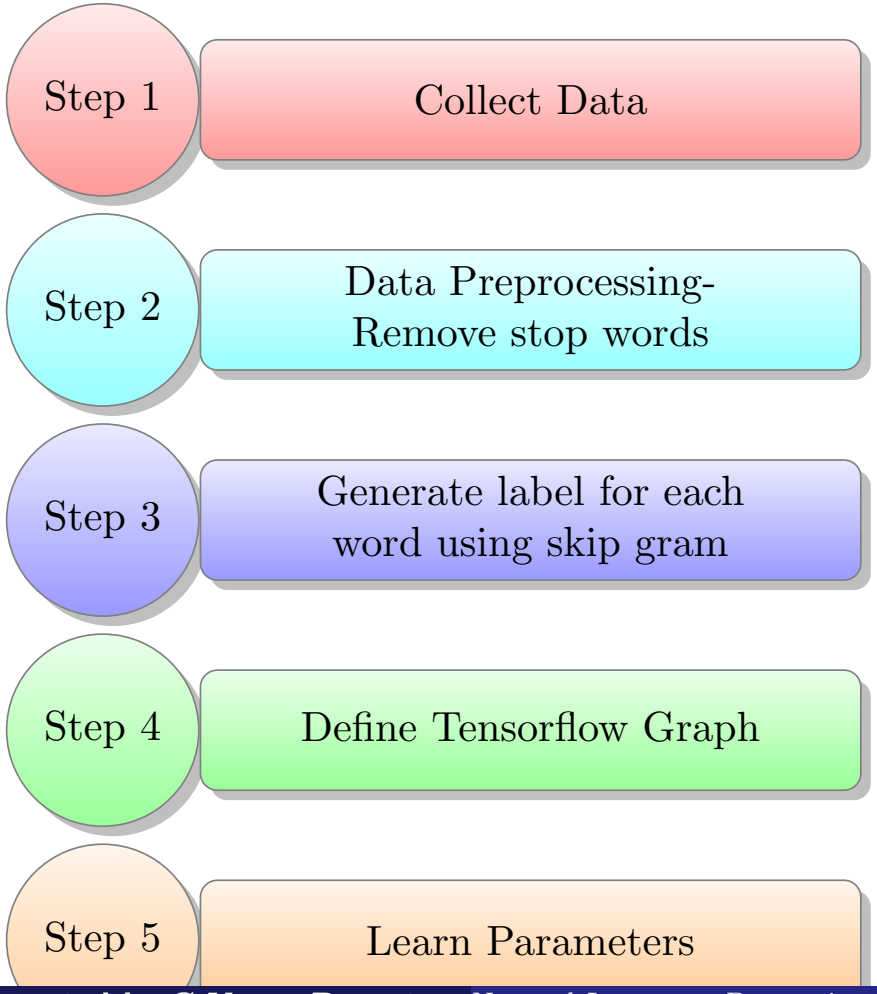
How word2vec works:



How word2vec works:



Implementation using Tensorflow



Collect Data

```
corpus = ['king is a strong man',
          'queen is a wise woman',
          'boy is a young man',
          'girl is a young woman',
          'prince is a young king',
          'princess is a young queen',
          'man is strong',
          'woman is pretty',
          'prince is a boy will be king',
          'princess is a girl will be queen']
```

Data Preprocessing

```
king strong man
queen wise woman
boy young man
girl young woman
prince young king
princess young queen
man strong
woman pretty
prince boy king
princess girl queen
```

Unique words in the corpus

```
{'boy',
 'girl',
 'king',
 'man',
 'pretty',
 'prince',
 'princess',
 'queen',
 'strong',
 'wise',
 'woman'
```

Generate Context word

```
['king', 'strong']
['king', 'man']
['strong', 'king']
['strong', 'man']
['man', 'king']
['man', 'strong']
['queen', 'wise']
['queen', 'woman']
['wise', 'queen']
['wise', 'woman']
```

Collect Data

	input	label
0	king	strong
1	king	man
2	strong	king
3	strong	man
4	man	king

Collect Data

```
[[0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
```

References

- Distributed Representations of Words and Phrases Tomas Mikolov and their Compositionality
- Stanford CS224d: Deep Learning for NLP http://cs224d.stanford.edu/index.html
- The best “word2vec Parameter Learning Explained”, Xin Rong https://ronxin.github.io/wevi/
- Word2Vec Tutorial - The Skip-Gram Model http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
- https://www.youtube.com/watch?v=64qSgA66P-8
- https://github.com/minsuk-heo/