Word Embeddings

Recitation 3

CS273B : Deep Learning in Genomics and Biomedicine

October 14, 2016

Overview

1. Word Embeddings
2. Motivation
3. Word2Vec
4. Visualization

Word Embeddings

Refers to the technique of language modelling where words or phrases form vocabulary are mapped to vectors of real numbers

Motivation

- A computer can understand only numbers
- Just encoding/assigning id for each word, doesn't help in understanding the relationship with different words
- This is unlike the case of image/audio data
- The raw pixels/signals provide meaningful information

Motivation

- For ex: cat - 01 and dog - 02. Using this encoding, little can be understood about the relationship between cat and dogs (like they are both animals, 4 legged, etc).
- Word embedding is a way to resolve this issue of representation of text data as vectors such that the relationship among different words and their context are captured.

Word2Vec

- Computationally efficient predictive model for learning word embeddings from raw text
- 2 flavours - CBOW and Skip-gram
- CBOW - predict target word from context
- Skip-gram - do the inverse operation of CBOW - predict context words from target words

CBOW

- CBOW - Continuous Bag of Words
- Before going to this, let's consider a simpler model

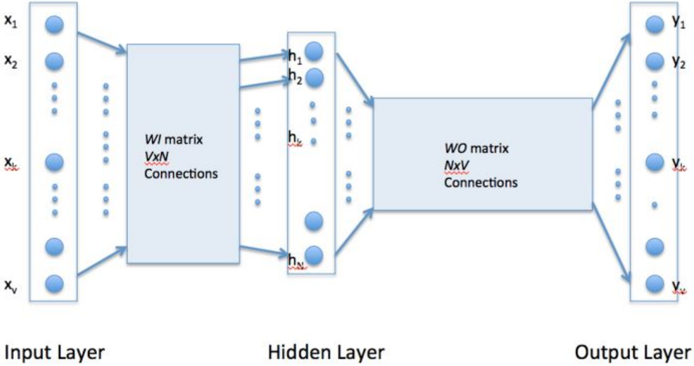


Figure: Simple word2vec [1]

CBOW

- Single hidden layer fully connected neural network
- Hidden layer size = dimensionality of word vector
- Output layer size = input layer size
- Ex: Cat climbed the wall
- The goal of the network is to predict the target word given a context word as input. For example, if "cat" is referred to as the context word, we want the network to show a high prob for "climbed" as "cat" and "climbed" are more likely to occur together.

CBOW

- Now onto CBOW
- Given multiple (say C) context words, need to determine the target word
- Idea : Replicate hidden layer C times

CBOW

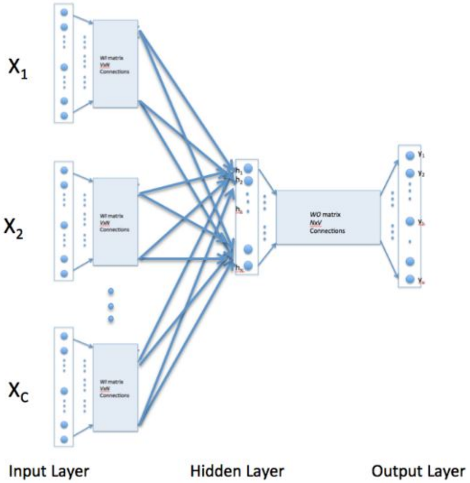


Figure: CBOW [1]

Skip-gram

- Let's do by example
- The quick brown fox jumped over the lazy dog
- Using a window of size 1 (this can be varied), we have ([the, brown], quick), ([quick, fox], brown), ([brown, jumped], fox), ... as (context, target) pairs
- Task : Given the target, need to predict the context words

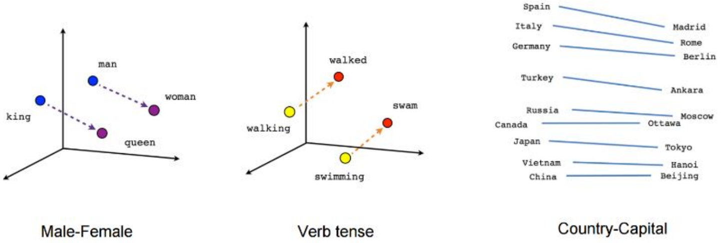
Skip-gram

- Create the training dataset as (quick, the), (quick, brown), (brown, quick), (brown, fox), ...
- You can use popular methods like GD, SGD to train
- Make the network learn the embedding and thereby relation between words
- Alternatively you can make the network produce two outputs which would form the pair of context words for the input and sum the errors for backprop training

Word2Vec

- These models can learn semantic relationships between words
- Skip-gram: works well with small amount of the training data, represents well even rare words or phrases [4]
- CBOW: several times faster to train than the skip-gram, slightly better accuracy for the frequent words

Visualization



Male-Female

Verb tense

Country-Capital

Figure: Word embeddings projected onto 2D [3]

Visualization

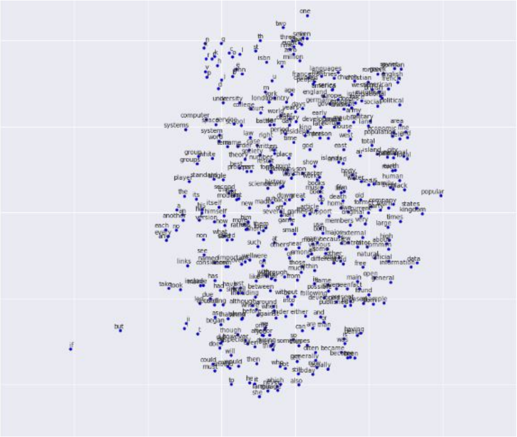


Figure: Word embeddings projected onto 2D [3]

References

Word2vec
https://iksinc.wordpress.com/tag/continuous-bag-of-words-c

Word Embedding
https://en.wikipedia.org/wiki/Word_embedding

Tensor Flow
https://www.tensorflow.org/versions/r0.11/tutorials/word2v

Word2Vec
https://www.quora.com/What-are-the-continuous-bag-of-words

Thank You