Word2Vec

(sgn3@illinois.edu)

Contents

1. Introduction
2. Skip-gram Model Architecture
3. Model Variations
4. Conclusion

Introduction

Word embedding is the collective name for a set of language modeling and feature learning techniques in natural language processing (NLP) where words or phrases from the vocabulary are mapped to vectors of real numbers. Example: A word vector for word France from Google News corpus can be a vector of length 100 <0.023, -0.011, ……, 0.123>. Individual values in the above vector does not mean anything, we should consider the word vector as a point in 100-dimensional space (dimensions can change, in this example we have word vector of length 100). Word2Vec model builds these word embeddings.

We can calculate distance between two points (points in 100-dimension for two words) or use cosine angle between the vectors to find how similar two words are. Cosine Similarity is used in practice. Example: Words vectors of Spain, Belgium Netherlands will be similar to word vectors of France.

Word embedding encode meaning of words, it also captures linguistic regularities, for example vector operations vector('Paris') - vector('France') + vector('Italy') results in a vector that is very close to vector('Rome'), and vector('king') - vector('man') + vector('woman') is close to vector('queen'). The word vectors can be also used for deriving word classes from huge data sets. This is achieved by performing K-means clustering on top of the word vectors.

Skip-gram Model Architecture

Word2Vec uses Neural Network with hidden layer to perform a Task, we will not use that Neural Network for the task it was trained on. The weights in the hidden layer that was learnt are the word vectors. There is no activation function on the hidden layer. Input and Outputs are one-hot vectors. Task that the Neural Network will be trained on will predict the probability for every word in the vocabulary of being the nearby input word.

We cannot input words to Neural Network, so we need to build vocabulary of all the words in the document. We need to represent each word as one-hot vector. Let's say we have 100 words in the vocabulary and word France is second word in the vocabulary then one-hot vector for France will be of length 100 and it will have 1 in second position.

Let's say we want to learn 100 features about words then we can have hidden layer representing 100 features. Hidden layer will have number of rows equal to the length of vocabulary, each row for each word in the vocabulary, each column represents the features. So, when one-hot vector for France is passed to input (1xN vector), hidden layer (Nx300 Matrix) output will be 1x300.

Output layer taken output of hidden layer and outputs vector (1xN) which represent the probability for every word in the vocabulary of being the nearby input word. Output layer is an SoftMax regression classifier, each output neuron will produce an output between 0 and 1.

In detail, each output neuron has a weight vector which it multiplies against the word vector from the hidden layer and then applies function exp(x) and then divides by the sum of the results from all 10,000 output nodes.

Hidden layer and output layer have weight matrix with dimension NxF (F is number of features and N is vocabulary size). Training Neural network which has such huge weights in hidden layer and output layer will be time consuming and we need lot of data to avoid overfitting.

Subsampling frequent words is one of the methods used, when input is provided to Neural Network, if the word is frequent word then they are not passed in the input sometimes and sometimes they are passed.

Negative Sampling is another method, In Negative sampling each training sample only modify a small percentage of the weights, rather than all of them.

Model Variations

Continuous Bag of Words (CBOW) is an alternative to Skip-gram Model. CBOW changes the Neural Network final task. CBOW Neural Network final task is to predict the how likely a word will be at the center given all the words in the context.

The input of the network needs to change to take in multiple words. Instead of a “one hot” vector as the input, we use a “bag-of-words” vector.

With skip-gram, we saw that multiplying with a one-hot vector just selects a row from the hidden layer weight matrix. What happens when you multiply with a bag-of-words vector instead? The result is that it selects the corresponding rows and sums them together.

In Output Layer we also divide this sum by the number of context words to calculate their average word vector. So, the output of the hidden layer in the CBOW architecture is the average of all the context word vectors. From there, the output layer is identical to the one in skip-gram.

Hierarchical SoftMax is an alternative to Negative Sampling. Both are methods of reducing the compute cost of training. In Hierarchical SoftMax, the number of negative samples varies and words to use as negatives are predetermined for each word from the vocabulary.

Conclusion

Word2Vec model trains Neural Network for a Task with a hidden layer. The weights of the hidden layer are word embeddings. we’re not actually going to use that neural network for the task we trained it on.