Word2Vec Skip-Gram Model using Reuters Corpus

Rajat Nagpal
M.Tech(SE)
rajatnagpal@iisc.ac.in

Github Link to this Report

Abstract

The recently introduced continuous Skip-gram model is an efficient method for learning high-quality distributed vector representations that capture a large number of precise syntactic and semantic word relationships. In this paper we present Word2Vec Skip-gram Model using Negative Sampling

1 Introduction

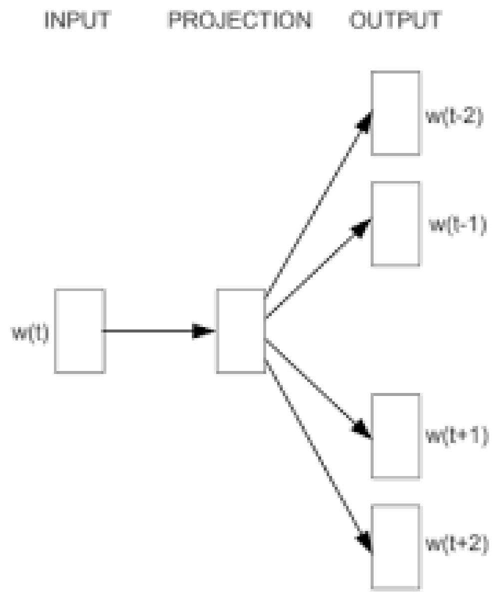
In this Model, I am using Word 2 Vector implementation from Reuters Corpus I am extracting Vocabulary and Pair-of-words with a window size of 2. I have used **noise-contrastive estimation training loss** for Loss calculation.I have used Tensorflow to implement this code.**Simlexx-999** Similarity Task is used for calculating the score.

2 Skip-Gram

The Basic Skip Gram formulation is Defined using Softmax function.

$$P(w_t|h) = \text{softmax}(\text{score}(w_t, h)) = \frac{\exp\{\text{score}(w_t, h)\}}{\sum_{\text{Word w' in Vocab}} \exp\{\text{score}(w', h)\}}$$

The Score generates the compatibility of input with respect to context. The Model is trained using Log-Likelihood because of the monotonic property of Log Function and Ease of computations of Log Function.



Skip-gram

In this figure, Skip Gram Architecture with window size 2 is shown. the center word is being paired with all the four context words to the right and left side of it , respectively.

The Skip-gram model architecture usually tries to achieve the reverse of what the CBOW model does. It tries to predict the source context words given a target word (the center word).

3 Spearmen‘s Correlation

The Spearman’s Correlation Coefficient, ρ , is a nonparametric measure of the strength and direction of the association that exists between two ranked variables. It determines the degree to which a relationship is monotonic, i.e., whether there is a monotonic component of the association between two continuous or ordered variables.

In the **Table1** above , Spearmen’s Correlation Coefficients for the different training mod-

Dataset	File Used	Num Pairs	Not Found	Rho
EN-SIMLEX-999.txt	`64-4-64.txt`	`999`	`375`	`0.1436`
EN-SIMLEX-999.txt	`64-4-128.txt`	`999`	`375`	`0.0874`
EN-SIMLEX-999.txt	`32_32_128.txt`	`999`	`375`	`0.0974`
EN-SIMLEX-999.txt	`32_32_256.txt`	`999`	`375`	`0.1773`
EN-SIMLEX-999.txt	`32_4_64.txt`	`999`	`375`	`0.1017`
EN-SIMLEX-999.txt	`32_6_32.txt`	`999`	`375`	`0.1235`

Table 1: **Spearmen Correlation** using different batch size, negative samples and embedding size respectively with max 0.1773

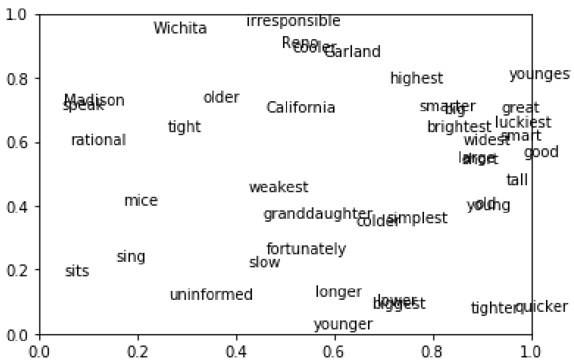
els which I have trained are given for the different combination of Hyper-parameters varied as per given in the problem statement in Task 2.

4 Negative Sampling

In negative sampling, we modify a small percentage of weights rather than modifying whole weights. This is because in the hidden layer of two layer neural network that we are using , only those weights corresponding to the input words are updated. More frequent words are likely to be selected as negative samples. In my model I have used different negative samples and trained my model and calculated Spearmen Correlation on **Simlex-999** Similarity Task.

Link to file with Max Correlation of 0.1773

5 Similarity



1=

Figure 1: Vector Representation of Words

In the Above Figure , I have taken some random words from my vocabulary and plotted their corresponding Embedding Vectors to show the similarity between two words , learnt from the context using Skip- Gram Model.

6 Analogical Reasoning Task

For Example , if we give London, England, Baghdad then following is the 10 nearest neighbours we get for Baghdad:

- Sweden
- Canada
- Australia
- Iraq
- Egypt
- Russia
- Vietnam
- Krona
- Greece
- cuba

All these are Country names and the most predictable amswer is Sweden followed by Canada and we get the right answer at fourth place . i.e , Iraq

There is some similarity captured between the predicted answers that all these are country names , which are nearby vectors in the corresponding Vector space of Word Embeddings.

I have used the logic that if we subtract corresponding Word Embeddings of **London** from **England** and add it to **Baghdad**, we get our desired Answer.

V(England)-V(London)+V(Baghdad)=Answer

7 10 Most Similar Words

I have given the link to K nearest neighbours here , This is my link: 10 Most Similar Neighbours found in Task 2 . I have noticed that since the data is less , if we increase the number of **Negative Samples** , the model gives more accuracy.

Github Link to this Report