**Word2Vec (Skip-Gram + Negative Sampling)**

Word2Vec is a popular natural language processing (NLP) technique that is used for representing the word in multiple dimensions, which are vector representations of words in a continuous vector space. These vectors are called embeddings, and these are not explainable. Embeddings are widely used in various NLP tasks such as text classification, machine translation, and sentiment analysis.

There are two main approaches to train Word2Vec models:

1. CBOW (Continuous Bag of Words) – predicting a **target word** based on the **context (neighboring) words.**

A diagram of a diagram

Description automatically generated

For example,

There is a sentence, “Jay was shit by a red bus”

Target word

A close-up of a screen

Description automatically generated

Context wordd

A white rectangular box with green text

Description automatically generated

1. Skip-Gram – predicting **context words** based on **target word.**

A diagram of a diagram

Description automatically generated

From the same above example,

A pink and white rectangular box with black text

Description automatically generated with medium confidence

The above example is based on window size of 2 i.e. predicting 2 words left and 2 words right from the target word.

In this article, we will focus on the Skip-Gram method.

* **Skip-gram method** – The objective of the Skip-gram method is to get a word representation in multidimensional vector space by correctly predicting the surrounding words in the document. It means, in the above example the model is trying to represent word “red” in vector space so that, it can predict the words “by”, “a”, “bus” & “in”. Hence, if I have four words “w1”, “w2”, “w3”, “w4” then it tries to maximize the Log-likelihood of multiple conditional probabilities.

A white paper with writing on it

Description automatically generated

Each probability is defined by a softmax function,

A math equation with a black line

Description automatically generated with medium confidence

Where individual vectors are word representation of w’s and W is the vocabulary size.

**Problem!**

The above representation has been defined by Neural Network with all the vocabulary in both Input layer and output layer with a hidden layer. Due to this representation, the final network size (#of weights) is significantly large.

Say, vocabulary size = 10000 & #hidden units = 100. The total number weight becomes 10000 X 100 + 100 X 10000 + 10000 = 2010000.

Solution!

* **Negative Sampling** – It is a technique that addresses the above issue by first change the structure target (input) and context (output) into all target + context (paired one) called positive samples and a sample of target + non-context (non-paired one) called negative samples.

This eventually transforms the above problem into a binary logistic regression.

A screenshot of a computer

Description automatically generated

**Creation of negative samples**

**Creation of positive samples**

The above method if adopted from NCE (Non Contrastive Estimation) and it converts Skip-gram into Skip-gram with Negative sampling.

A diagram and input word

Description automatically generated

Hence, we need to maximize the below objective function,

A close-up of a logo

Description automatically generated

Where,  is sigmoid function, v’s are the vector representations and  is the noise distribution with negative samples.

The selecting of a wi becomes a negative samples is based on the below probability,

A math equations and formulas

Description automatically generated with medium confidence

Where, f(wi) is the frequency of wi in the document.

* Links for further reading –
* <https://jalammar.github.io/illustrated-word2vec/>
* <https://www.baeldung.com/cs/nlps-word2vec-negative-sampling>
* <https://zhuanlan.zhihu.com/p/42651829>
* <https://arxiv.org/pdf/1301.3781.pdf>
* <https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf>
* <http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/#:~:text=Word2Vec%20implements%20a%20%E2%80%9Csubsampling%E2%80%9D%20scheme,related%20to%20the%20word's%20frequency>