Proceedings of the 2nd International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/18/20230392

The study for word prediction based on Skip-gram and CBOW model

Peijun Li

Department of Applied Psychology, New York University, 50 West 4th Street, New York, NY, 1oo12, United States

pl1900@nyu.edu

Abstract. In the modern world, when people use their texting apps to communicate with each other, they find their phones are trying to predict what they want to type. The process of this word prediction is actually not a new topic, previous researchers have involved the Natural Language Processing (NLP) word prediction algorithms in many studies. Thus, in this article, a project implementing the Skip-gram model and the Continuous Bags of Words (CBOW) model is created in this study to find out whichever of the two newest and reversed word prediction models would be better to utilize. By comparing the efficiency items including total training time, total effective words, training time per epoch, and effective words per epoch in the experimental results, the project discovered that the CBOW model could be better utilized than the Skip-gram model, which means that the CBOW model can be considered more under this kind of task.

Keywords: Skip-gram models, CBOW, Word2vec.

1. Introduction

Nowadays, while people are immersed in the use of mobile phone communications, it can be noticed that when people try to type a sentence, there will be a bundle of words that might predict what people are trying to type about. Though sometimes the word list cannot precisely predict the word people want to use, people might still be confused about how the program can work to predict some synonyms that might help us to get the word people desire.

As sometimes people cannot experience the precision of predicting the words, studying to eliminate some of the distractions based on advanced algorithms are necessary. With the advent of deep learning, neural networks are considered as the effective method in this case due to their excellent performance in many tasks [1-5]. Distributional Semantic Models (DSMs) is an important branch of this method. However, these solely textual models, like conventional symbolic AI systems, are incredibly underdeveloped when compared to human semantic understanding since they lack a foundation in extra-linguistic modalities [6].

Thus, in order to enrich the linguistic vectors with perceptual information, previous scientists improvised using the Multimodal Distributional Semantic Models (MDSMs). Modern MDSMs beat text-based techniques on general semantic benchmarks as well as tasks that directly demand access to visual knowledge [7, 8]. However, even though it works clearly better than DSMs, MDSMs still have several drawbacks. Typically, they are created by first creating separate language and visual representations of the same concepts, and then combining them. By hearing words in a situated

DOI: 10.54254/2753-8818/18/20230392

perceptual situation, people learn about concepts in a way that is obviously very different from this [6]. In addition, MDSMs presuppose that all words have access to both linguistic and visual information, with no generalization of knowledge across modalities. Lastly, current MDSMs, however, cannot be applied to computer vision tasks like image identification or retrieval since they cannot generalize to images or words outside of their training set due to the latter assumption of full linguistic and visual coverage [6].

Hence, to improve this method based on the model of DSMs and MDSMs, Lazaridou et al. finally utilize the Skip-gram model, which solved all the issues that the DSMs and MDSMs have. This is a model that builds vector representations by gradually learning to anticipate the language settings in which target words will appear in a corpus. For a portion of the target words, pertinent visual cues from real-world images are shown alongside the corpus contexts (much as people hear words together with concurrent perceptual inputs when they hear speech) [6].

As in the progress of researching the Skip-gram model, this paper tries to implement the basic baseline to run the model of it, and this study found that it is convenient to work on any given raw text, while it takes for a long time to finish the computation. Thus, in order to elevate the computing time, the goal of this article is to find ways to improve the Skip-gram model.

2. Method

2.1. Data Description

In this project, the word-sentence dataset provided by Stamos was considered [9]. The original dataset consists of 91, 207 rows, each row of about 150 words. In other words, the dataset consists of a total of about 170, 224, 350 words.

When using the Skip-gram to work with this dataset, this paper uses the word2vec to run on it. In addition, it requires forward propagation and backpropagation. In a neural network, the path from the input layer (on the left) to the output layer (on the right) is called forward propagation. Backward propagation is the process of going from the output layer to the input layer backward, or from right to left.

2.2. Proposed approach

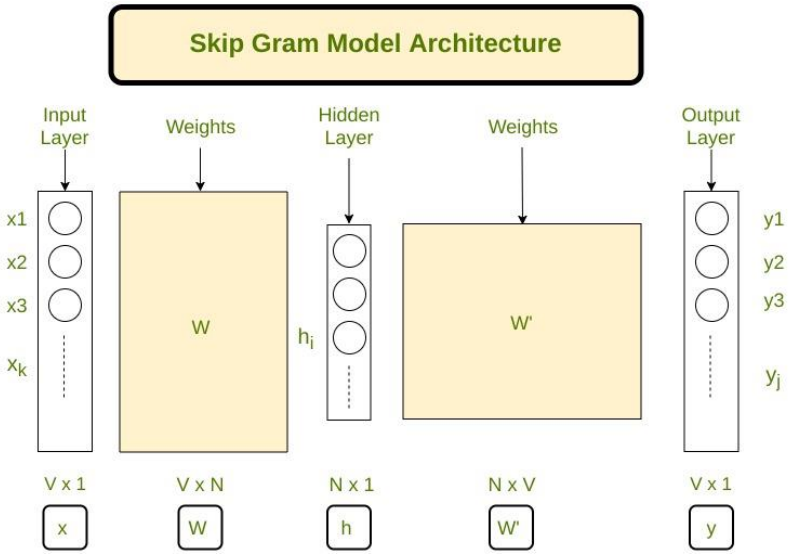

Figure 1. Model of Skip-gram [10].

To make a machine learn from the raw text people need to transform this data into a vector format which then can easily be processed by our computers. This transformation of raw text into a vector format is

DOI: 10.54254/2753-8818/18/20230392

known as word representation. The word is represented as a vector in a word representation, therefore if two words vectors are close to one another, then those words are connected.

Due to the size of any language's vocabulary and the difficulty of human labeling, unsupervised learning techniques are needed so that any word can understand its context on its own. One unsupervised learning method used to discover the words that are most connected to a given word is the skip-gram.

In this project, the Skip-gram model shown in Figure 1 is used to test the effective words per epoch. This is measured to find whether the word is similar to what people want to predict. In addition, this project also tests the training time per epoch and the total training time and total effective words.

Besides the research on the Skip-gram model, this project also tested on the Continuous Bags of Words (CBOW) as a comparison to the Skip-gram model to test which would be a better model on the word prediction.

2.3. Implementation Details

Firstly, to define some of the variables this project has to use in the Skip-gram model as shown in the figure above. V represents for the number of unique words in our corpus of text (Vocabulary), x represents the input layer (one hot encoding of our input word), N is the number of neurons in the hidden layer of neural network, W is the weights between input layer and hidden layer, W' is the weights between hidden layer and output layer, and y is a softmax output layer having probabilities of every word in the vocabulary.

The loss function could be expressed by the forward propagation to maximize the likelihood that w(c,j) will be predicted on the cth context position. This project used a Python-based word2vec as the framework. And the project classified the dataset similarity into 4 levels: 1) similarity between 2 words, 2) The most similar word, 3) Does not match groups, and 4) nth most similar words. For both of the models Skip-gram and CBOW, this project sets the epoch at 10, and input size at about 19.

3. Results and Discussion

Table 1. Total training Time.

CBOW	Skip-gram
956.5	3768.5

Table 2. Total effective Words.

CBOW	Skip-gram
1,327,456,338	1,327,454,735

Table 3. Training time per Epoch.

Epoch	CBOW	Skip-gram
Average	95.65	376.85
1	95.9	338.3
2	95.3	340.0
3	96.7	339.9
4	96.1	448.0
5	95.4	339.3
6	95.3	339.8
7	95.6	339.9
8	95.3	599.3
9	95.3	342.8
10	95.6	341.2

DOI: 10.54254/2753-8818/18/20230392

Table 4. Effective words per epoch.

Epoch	CBOW	Skip-gram
Average	132745634	132745474
1	132750757	132744876
2	132744712	132741580
3	132743879	132750658
4	132748376	132743435
5	132747942	132749631
6	132746112	132744974
7	132744511	132745877
8	132742194	132744706
9	132740767	132745693
10	132747088	132743305

From Table 1, it shows the total training time, in parameter of seconds, of CBOW and Skip-gram model. Table 2 shows the total effective words of CBOW and Skip-gram. Taking a direct look at these results yielded by the two reverse-algorithm models, it shows that CBOW can make a faster training and a relatively higher accuracy on the word prediction task.

From Table 3 and Table 4, it further proves that at each epoch of training time, CBOW is clearly faster, at about 4 times faster than the Skip-gram model. As for the effective words per epoch, CBOW has about 1200 more effective words compared with the Skip-gram model. In addition, Figure 2 and Figure 3 also present the comparison of effective words and training time per epoch.

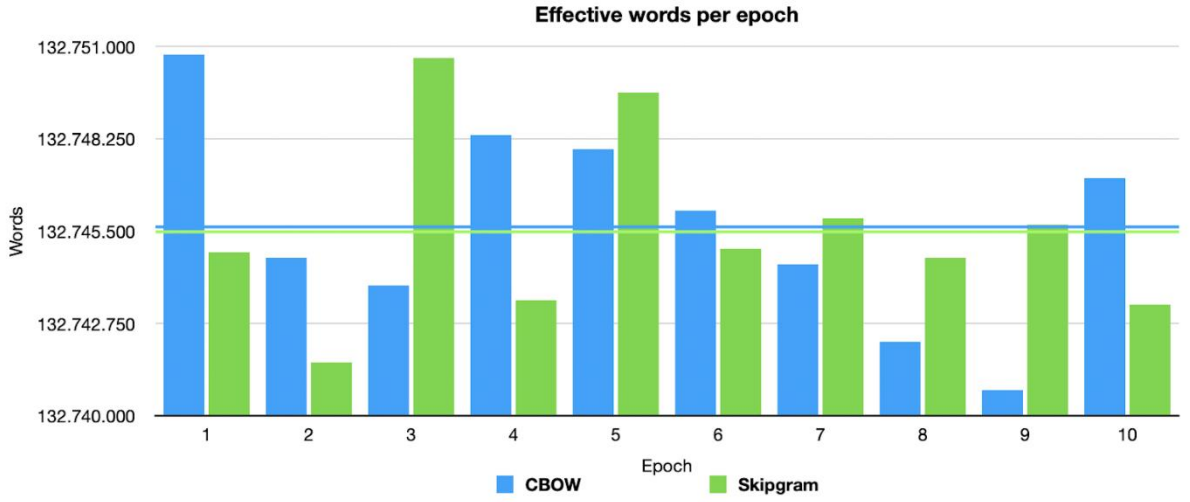

Figure 2. Comparison of Effective Words per Epoch.

DOI: 10.54254/2753-8818/18/20230392

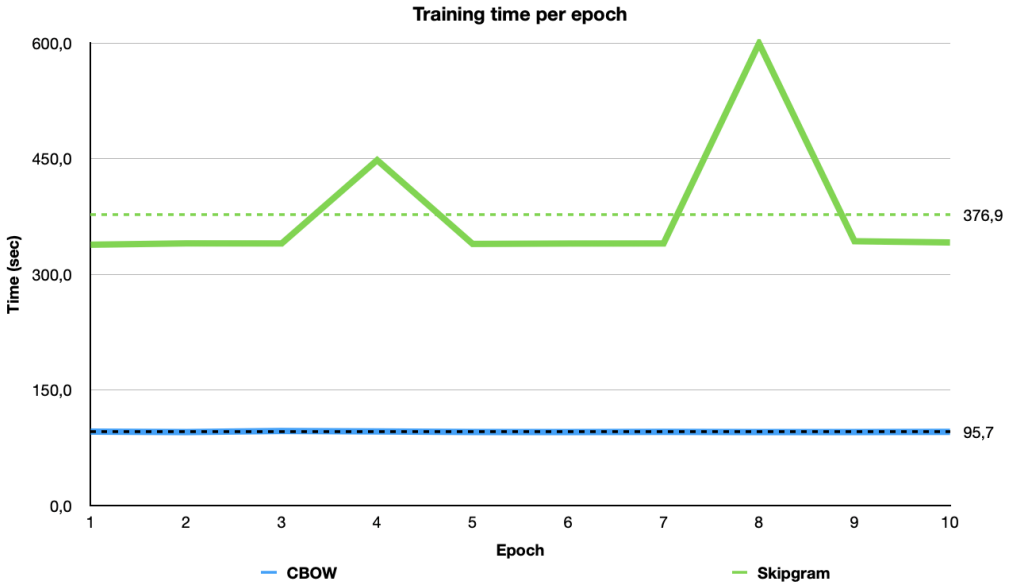

Figure 3. Comparison of Training Time per Epoch.

4. Discussion

This project has tested Skip-gram under the dataset of about 170, 224, 350 words, with an output of average 132,745,474 words at each epoch. This indicates that for the Skip-gram model, it can work on unsupervised learning, given any raw texts can be trained on it. In addition, it required comparatively less memory as running the model, compared with other word vector models.

However, it also has some limitations. Firstly, it would take a relatively longer time to finish the training. Moreover, the softmax function behind the Skip-gram model is computationally expensive.

To increase the time efficiency, the project used the CBOW, which has the reverse-algorithm of Skip-gram working on the same dataset. Clearly, it can train the dataset at a frequently higher accuracy with a much faster training time.

Moreover, the way Skip-gram acts seems to be hit or miss. It predicts a term that is nearby in certain case studies and a word that is semantically connected to or commutable in others. This observation suggests that CBOW would be a better fit for situations involving query expansion and synonyms.

5. Conclusion

In this project, the Skip-gram model and the reverse version of it (CBOW) were utilized. The project compared the efficiency of both of the models, including total training time, total effective words, training time per epoch, and effective words per epoch. In the result the project shows that although the Skip-gram model can be used as an NLP tool on any given raw texts, the model would take longer time to finish the goal, compared to CBOW model. In the future, the behavioural research of word2vec in phrases and to a relatively new approach in which word2vec technique is trained on multi-dimensional data attributes, depending on application, would be a highly interesting expansion of this thesis.

References

[1] Moradkhani, A et al. 2022. A portable medical device for detecting diseases using Probabilistic Neural Network. Biomedical Signal Processing and Control, 71, 103142.

[2] Do C et al. 2022 Application of deep-learning for medical text analysis: a comparison of different types of recurrent neural network. Journal of Data Science and Artificial Intelligence, 1(1).

DOI: 10.54254/2753-8818/18/20230392

[3] Zhang P et al. 2022. A dual-domain neural network based on sinogram synthesis for sparse-view CT reconstruction. Computer Methods and Programs in Biomedicine, 107168.
[4] Yu Q et al. 2022. Pose-guided matching based on deep learning for assessing quality of action on rehabilitation training. Biomedical Signal Processing and Control, 72, 103323.
[5] Vétil R et al. 2022 Improving The Automatic Segmentation Of Elongated Organs Using Geometrical Priors. In 2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI) (pp. 1-4). IEEE.
[6] Lazaridou A et al. 2015 Combining language and vision with a multimodal skip-gram model. arXiv preprint arXiv:1501.02598.
[7] Bruni E et al. 2012 Distributional semantics in technicolor In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp 136-145)
[8] Silberer C et al. 2014 Learning grounded meaning representations with autoencoders In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 721-732)
[9] Github 2022. Estamos (n d) Releases Estamos/word2vec-thesis. https://github.com/estamos/word2vec-thesis/releases
[10] GeeksforGeeks 2022. Implement your own word2vec(skip-gram) model in Python https://www.geeksforgeeks.org/implement-your-own-word2vecskip-gram-model-in-python/