Word2vec and CBOW

Apurba Das
40263612
contactapurbadas@gmail.com

November 3, 2024

Abstract. In this assignment, we undertook a practical investigation of the Continuous Bag of Words (CBOW) model[2], implemented using PyTorch, which is a key element of Word2Vec for generating word embeddings. Our exploration encompassed essential steps such as data preprocessing, vocabulary development, and the construction of a neural network aimed at predicting context-based embeddings. We employed the Wikitext dataset sourced from Wikipedia to train the CBOW model and evaluated its performance on analogy tasks to measure the quality of the produced word embeddings. Furthermore, we experimented with a different dataset and another subset of the analogy test dataset to analyze the influence of variations in training data on embedding quality and semantic task accuracy. This thorough examination enhanced our understanding of Word2Vec's underlying mechanisms and its applications in natural language processing, enabling us to utilize these techniques across a range of NLP contexts.

Contents

1 Introduction

The objective of this assignment was to explore the Continuous Bag of Words (CBOW) model, an essential element of the Word2Vec framework designed for generating word embeddings. CBOW (Continuous Bag of Words) and Skip-gram are two architectures used in word embedding models, particularly in the Word2Vec framework. They serve to learn word representations from large text corpora, but they approach the task differently, as illustrated by the diagram below.

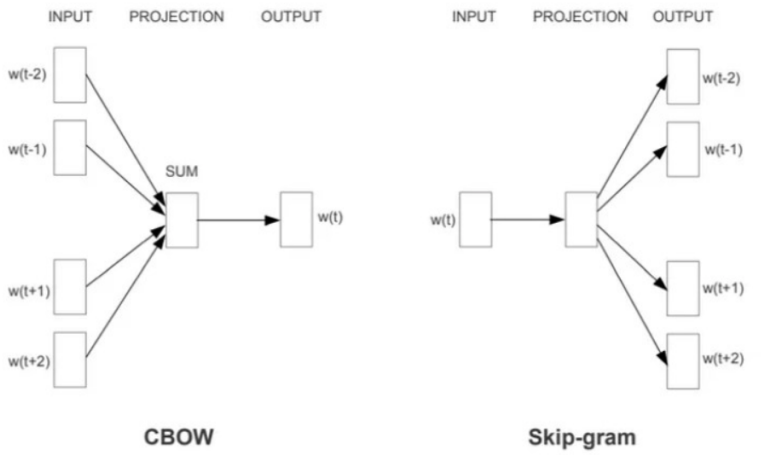

Figure 1: CBOW vs Skip Gram

We engaged in a hands-on implementation of CBOW using PyTorch, which encompassed several critical phases: data preprocessing, vocabulary development, and the construction of a neural network aimed at predicting context-based embeddings. Our focus was on training the CBOW model utilizing the Wikitext dataset and evaluating its effectiveness through analogy tasks, a common method for measuring the quality of word embeddings with a vocabulary size of 59352. Unfortunately, our experiments resulted in an accuracy of 0.0, underscoring significant obstacles encountered in generating effective embeddings. This was done with taking a random sample fo 5000 lines from the dataset using capital countries subset of the test analogy dataset. We then proceeded with a different dataset, this time taking the recommendation provided by our POD where we selected the first 5000 lines, with torch seed (due to resource constraints, the first model was not rerun again in the same manner), and got a vocabulary size of 21556. But,

we still ended up 0.0 accuracy. On changing the analogy subset to family from capital countries, we got an accuracy of 0.0047 with the new dataset.

2 Experiments

In this section, we detail our experimental investigations into the Continuous Bag of Words (CBOW) model. We outline the motivations behind each experiment, the methodologies employed, the results obtained, analyses of those results, and the limitations we encountered throughout the process.

2.1 Experiment 1

2.1.1 Motivation

The primary motivation for Experiment 1 was to evaluate the performance of the CBOW model on a different dataset, aiming to ascertain whether variations in the training data could lead to improvements in the quality of generated word embeddings.

2.1.2 Methodology

In this experiment, we selected a new dataset - gutenberg book corpus, from that we selected austen-emma, shakespeare-hamlet, and melville-moby_dick. We conducted data preprocessing, including tokenization and vocabulary creation, followed by implementing the CBOW model in PyTorch. The focus was on training the model to predict target words based on their surrounding context.

2.1.3 Results

Despite our efforts of changing the dataset, using manual_seed and selecting the first 5000 lines, this experiment resulted in an accuracy of 0.0, indicating that the model was unable to effectively learn from the new dataset as well.

2.1.4 Analysis

The consistent 0.0 accuracy suggests potential issues with either the model's architecture or the dataset's characteristics. Possible explanations include inadequate training epochs, inappropriate hyperparameters, or a lack of sufficient context within the new training data.

2.1.5 Limitations

There were a lot of limitations particularly in terms of the resources, as we couldn't use powerful GPUs like A100 or increase the number of epochs/ sample size due to the GPU getting disconnected for running for long hours. Hence, we had to stick to the sample size of 5000 lines and 10 epochs only. This also limited our experimentation with hyperparameter tuning such as changing the learning rate (although the existing of 0.001 is an ideal suggested one usually)

2.2 Experiment 2

2.2.1 Motivation

In our second experiment, we aimed to assess the CBOW model's performance using a different subset of the analogy dataset, specifically focusing on familial relationships. This was motivated by the hope that varying the analogy context could yield better performance.

2.2.2 Methodology

We maintained the same architecture of the CBOW model but switched our focus to a different subset of analogy tasks. The training process remained consistent, but we anticipated that the familial context might lead to improved results in evaluating the model's effectiveness.

2.2.3 Results

This adjustment yielded a slight increase in performance, with an accuracy of 0.0047. While this result was still low, it indicated some potential for the model's effectiveness under this new analogy context.

2.2.4 Analysis

The marginal improvement in accuracy suggests that while the CBOW model struggles with generating high-quality embeddings, certain contexts or relationships may enhance its performance. This finding underscores the complexity of language representation and the challenges inherent in learning meaningful embeddings. We should try to ensure our training dataset and the test analogy dataset have similar genre.

2.2.5 Limitations

Despite the slight improvement, we recognized that the accuracy remained significantly below expectations. We lacked a thorough evaluation of the model's predictions to better understand its weaknesses, and couldn't experiment further due to resource constraints and time limits of google colab.

3 Conclusion and Future Work

3.1 Summary of Your Work and Findings

In this assignment, we explored the Continuous Bag of Words (CBOW) model, a crucial aspect of the Word2Vec framework, focusing on its implementation using PyTorch. Our investigations involved two key experiments: the first utilized a different dataset to assess the model's performance in generating word embeddings, while the second involved testing a different subset of the analogy dataset, specifically targeting familial relationships. Unfortunately, both experiments yielded unsatisfactory results, with the initial dataset producing an accuracy of 0.0, and even after switching to a new analogy subset, we only achieved an accuracy of 0.0047. These findings highlight the challenges encountered in generating effective embeddings and underscore the need for further investigation into model performance and training methodologies.

As part of our analysis to understand our dataset better, we had created a TSNE projection of 1000 words from our dataset [1]. Similar words are supposed to be grouped together. However, from the figures below and in the appendix section, we can see that vector semantics and embeddings of words might not have been as good as needed to achieve a good model with high accuracy. For the first wikitext dataset, words like might, new and fallen have been grouped nearby each other. For the second books related dataset, we have words like deal, lawn and regret present themselves nearby each other in our representation.

Figure 2: Wikitext TSNE zoomed in

Figure 3: Books dataset TSNE zoomed in

3.2 Limitations

Several limitations affected our experiments. In the first experiment, we did not perform extensive hyperparameter tuning, which may have impacted the model's ability to learn effectively from the new dataset. Additionally, the choice of datasets could have constrained our results, as the quality and structure of the training data are crucial for embedding generation. In the second experiment, while we altered the analogy context, we still operated within a narrow framework wherein due to resource constraints we only worked with 5000 lines, instead of the entire dataset, ran the model with only 10 epochs, and stuck to the same test analogy dataset provided.

3.3 Future Work

Future work should aim to address the limitations identified in this study. We propose conducting a more systematic exploration of hyperparameter tuning like changing the learning rates and experimenting with various neural network architectures to improve the CBOW model's performance. We could also try increasing the number of epochs or change the optimizer from Adam to something else and see the effect on the result. Additionally, expanding the range of datasets used in training, including domain-specific corpora, might provide richer linguistic contexts for the model to learn from. For example, had we taken a geography textbook as the training dataset, and then used the capital-countries subset from the test analogy dataset, there would be much higher chances of getting better accuracy. We could also try and use Skip-gram model and check if it improves our model for the existing dataset and analogy test set.

References

[1] Jurafsky, D., and Martin, J. H. Ch 6. vector semantics and em-

beddings. https://web.stanford.edu/~jurafsky/slp3/6.pdf, 2024.

[2] Kosseim, L. Comp 6781 vector semantics and embeddings. https://moodle.concordia.ca/moodle/mod/resource/view.php?id=4001819, 2024.

3.4 Appendix

This section contains the extra figures for understanding our analysis and experiments.

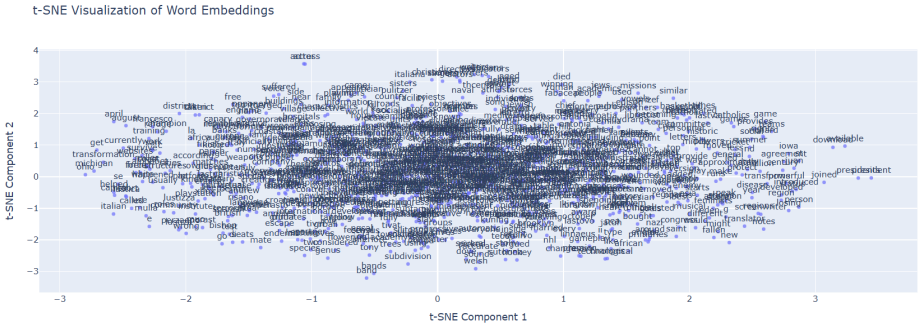

Figure 4: Wikitext TSNE for 1000 words

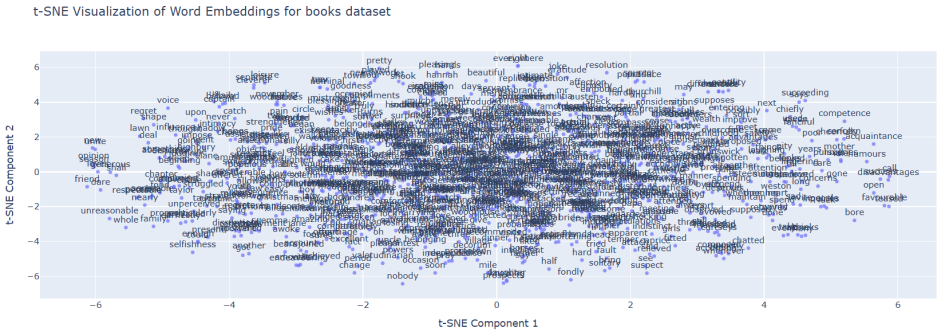

Figure 5: Books dataset TSNE for 1000 words

```
vocabulary
```

```
{'emma': 865,
 'jane': 301,
 'austen': 1,
 'volume': 17,
 'chapter': 229,
 'woodhouse': 314,
 'handsome': 41,
 'clever': 28,
 'rich': 25,
 'comfortable': 43,
```

Figure 6: Glimpse of a part of the vocabulary of second dataset