Skip-Gram Mini Project Report

Ilayda Soz Yilmaz

January 1, 2025

Introduction

The objective of this mini project is to implement a skip-gram model for word embedding generation. Skip-gram models are commonly used in natural language processing to predict context words from a target word within a predefined window size.

Methodology

Data Preparation

The input data consisted of the Text8 corpus, a dataset derived from the first 100 million characters of the English Wikipedia dump (March 3, 2006). The preprocessing steps included:

- Downloading the first 20 million characters of the Text8 corpus.
- Splitting the text into words using whitespace as the delimiter.
- Counting the frequency of each word in the dataset.
- Retaining the top 60,000 most frequent words to build the vocabulary.
- Replacing less frequent words with a special `<UNK>` token.
- Assigning a unique index to each word in the vocabulary, including the `<UNK>` token.
- Converting the text data into a sequence of word indices.
- Generating skip-gram pairs using a context window size of $C = 2$ to create word pairs for training.

Model Implementation

The skip-gram model was implemented using the following steps:

1. Creation of a vocabulary from the corpus.
2. Encoding of words into one-hot vectors.
3. Training a shallow neural network with one hidden layer to predict context words given a target word.

The implementation was done in Python using PyTorch library.

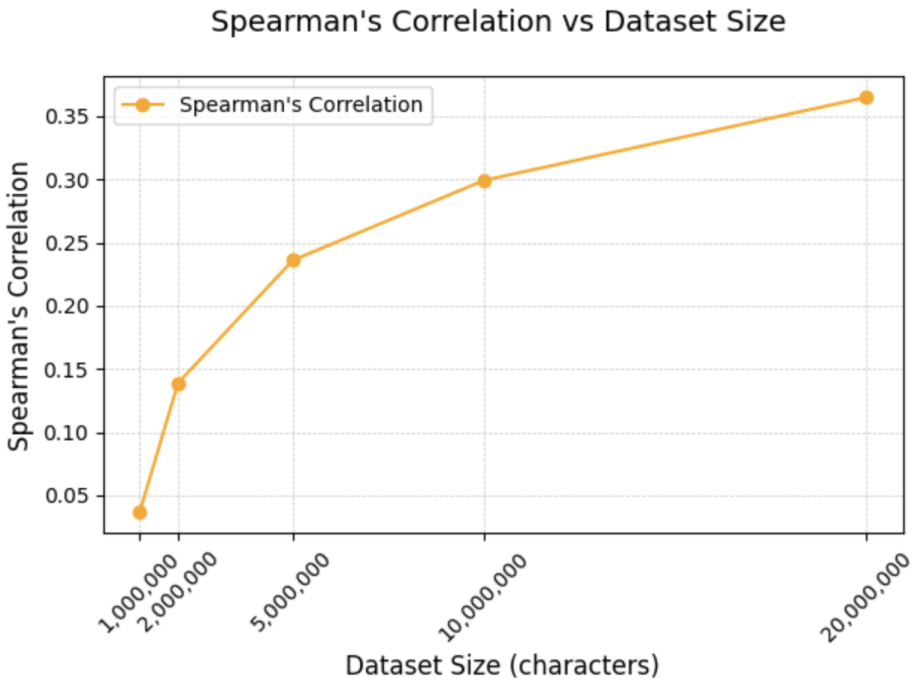

Figure 1: Spearman's Correlation vs Dataset Size

Training

The model was trained using the following parameters:

- Embedding dimension: 100
- Batch size: 1024
- Number of epochs: 10
- Learning rate: 0.01 (using Adam optimizer)
- Negative sampling: 5 negative samples per positive pair

Results

The trained model was evaluated using the WordSim-353 dataset. Figure 1 shows how Spearman's correlation improves as the dataset size increases. It starts at 0.0368 for 1 million characters and gradually rises to **0.3650** for 20 million characters. However, the improvement slows down with larger datasets.

Discussion

The skip-gram model captured semantic relationships between words in the corpus. Challenges included:

- Limited corpus size leading to sparse vocabulary.
- Balancing window size for meaningful word pairs.

Future improvements could involve experimenting with larger datasets and alternative models such as CBOW.

Conclusion

By implementing the model step-by-step, I learned about important concepts in natural language processing and how neural networks process data. The results showed that increasing the dataset size improves the embeddings, but the improvement slows down as the dataset gets larger.

References

- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint *arXiv:1301.3781*.
- PyTorch Documentation. `https://pytorch.org/docs/`