Assignment 1: E1-246
Word2Vec Skip-gram model

Aadesh Magare
aadeshmagare@iisc.ac.in

1 Introduction

This is final report for assignment 1. The problem statement was to train a skip-gram based word2vec model. The report contains experimental details for hyper parameter tuning. Learned embeddings are evaluated using simlex-999 benchmark. (?) Reuters dataset is used for training the word2vec model.

2 Task 1

The task was to implement word2vec skip gram based model. PyTorch is used for the implementation. Stochastic gradient descent is used as optimizer.

Pre-processing: Following pre-processing is carried out on the reuters data.

- All words are converted to lowercase
- punctuations are removed.
- Words with frequency less than 5 are ignored.
- Negative sampling is implemented as mentioned in paper. (?)

The dataset is converted into triplets of word, context word and negative samples which is further used for training.

2.1 Hyperparameter Tuning:

Batch Size: Following graph shows variation of loss function for different values of batch size. Batch sizes considered are 128, 256, 512.

Window Size: Following graph shows variation of loss function for different values of window size. Window sizes considered are 2, 4, 6.

Embedding Size: Following graph shows variation of loss function for different values of embedding dimension size. Embedding sizes considered are 100, 200, 300.

Negative Samples: Following graph shows variation of loss function for different values of negative samples. No of negative samples considered are 3, 5, 7.

Final Model: Based on above observations following model parameters were selected and used for further experiments.
Batch Size = 256
Windows Size = 2
Embedding Size = 300
Negative Samples = 5

2.2 Simlex-999:

The final model is evaluated on simlex-999 benchmark.
The performance metric is spearmans coefficient which is calculated by taking cosine similarity between embeddings of pair of words generated by model and similarity values given by the simlex-999 dataset.

3 Task 2:

3.1 Word Analogy task:

Final modle is evaluated against the qusetion words dataset to perform the word analogy task.
The final performance was evaluted on top-k values. i.e. we look at top k values of given analogy word to find resulting word.
Here the model performance is not as good as expected this might be due to the small vocabulary size.

3.2 Biases:

The training corpus is reuters dataset which is news articles thus the model also learns some biases commonly found in news.