Natural Language Understanding - E1246
Assignment 1
Implementation of prediction based models for learning Word Embeddings

Suman Gupta
Computer Science and Automation
Indian Institute of Science
sumangupta@iisc.ac.in

1 Introduction

In this assignment, we have implemented word2vec using the skipgram model for creating a word embedding on Reuters data set. We also experimented with different values of hyper parameters such as embedding size, context window size, negative sampling size, etc. For the first task, I have evaluated different models based on their loss function(performance on validation set) and their correlation with **Simlex-999** task scores. In the second task, we performed the analogy task e.g. V[king]-V[man]+V[woman]=V[queen].

2 Solution Sketch

I have implemented word2vec using a fixed batch size of 1 and fixed learning rate 0.001 (learning rate greater than this caused overflow). I have tried models with combinations of embedding dimensions of size 50, 100, 150 with negative sampling size of 10, 15 and window size 2, 3.
I have used numpy package for implementing the skipgram model. For preprocessing, I have converted the text to lower case and removed all the dots ('.'), in order to reduce unnecessary pairs. I aslo converted all the numbers in the corpus with a keyword "num" as there were a lot of numerical data in the corpus. Hyper-parameters for selected model
Negative sampling size (k)= 10
Context window size (w)= 3
Embedding dimension (d)= 50

2.1 Approach

1. Data Preporcessing

2. Vocabulary creation and tokens generation

3. Word-context pair generation based on window size (w=3)

4. For every word-context pair,
   (a) Calculate log of sigmoid loss of given context word pair with negative sampling (k=10)
   (b) Calculate the derivative of the loss function with respect to input word embedding vector, target word embedding vectors and negative samples embedding vectors
   (c) Update the weights of the input word, context word and negative samples based on the derivatives obtained.

5. Output: Word embedding in matrix format

3 Results

3.1 Task 1

Objective function value on validation set = $1.516095 \times e^{-06}$

Spearman Correlation coefficient value = 0.10597

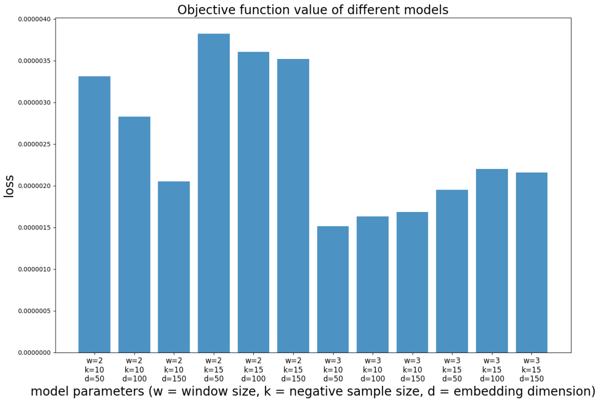

Figure 1: Objective function

3.2 Task 2

Accuracy in analogy task=0.09 percent.
The accuracy in the analogy task is very low which was expected, because our dataset is very small. Out of 4192 matching quadruples, it could predict only 4 of the outputs correctly.

1. competitive uncompetitive likely unlikely
2. enhancing enhanced falling fell
3. sit sits estimate estimates
4. shuffle shuffling say saying