CS7015: Deep Learning
Assignment4

Divya K Raman
EE15B085

4th December 2018

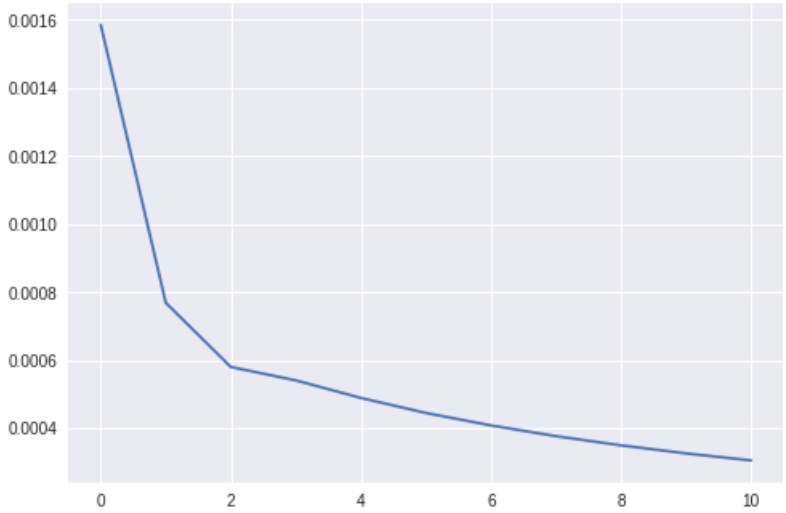
1 Introduction

This is an assignment on natural language processing. In the first part of the assignment, we train our own word2vec model for the text8 dataset using Bag-of-words, skip-gram and LSTM-based models. In the second part of the assignment, we classify movie reviews in one of the 5 classes based on the sentiment of the review.

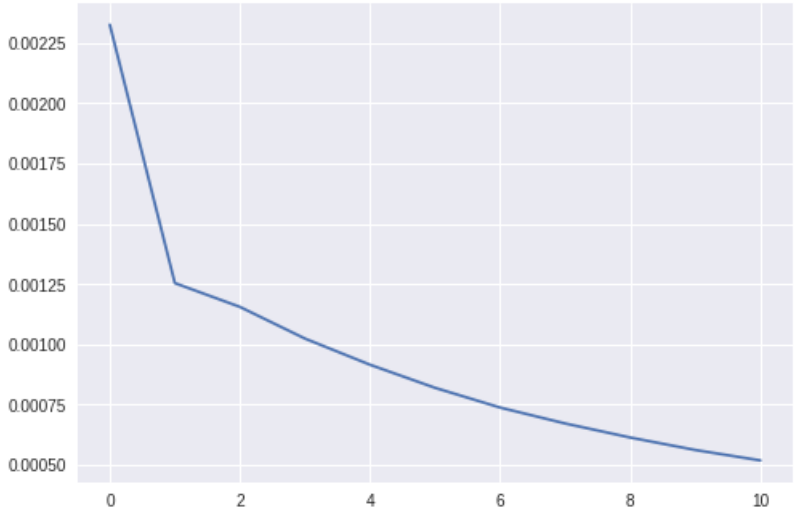
2 Part A

In PartA, we train our own word2vec models for the "text8" dataset. Text8 is downloaded from gensim. Stopwords and punctuations are removed and the text is cleaned. Number of unique words is counted. We then make a word to index and index to word mapping with the unique words. Pytorch is used for this assignment.

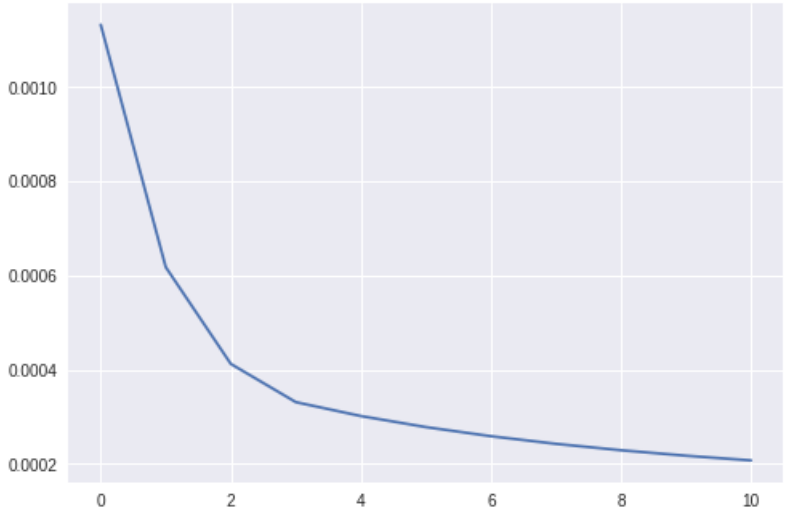
In the skip gram model, the surrounding words are predicted given the current word.We try using window sizes 1 and 2. Window size 2 works better than window size 1. We create index pairs in order to train. Embedding dimension 100, 200 and 50 is used. We use a two layer neural network and finally apply softmax. The code is run on a very small subset of text8 due to the time it takes to run. Also, it is run for 11 epochs.
Embedding Dimension 100 train plot:



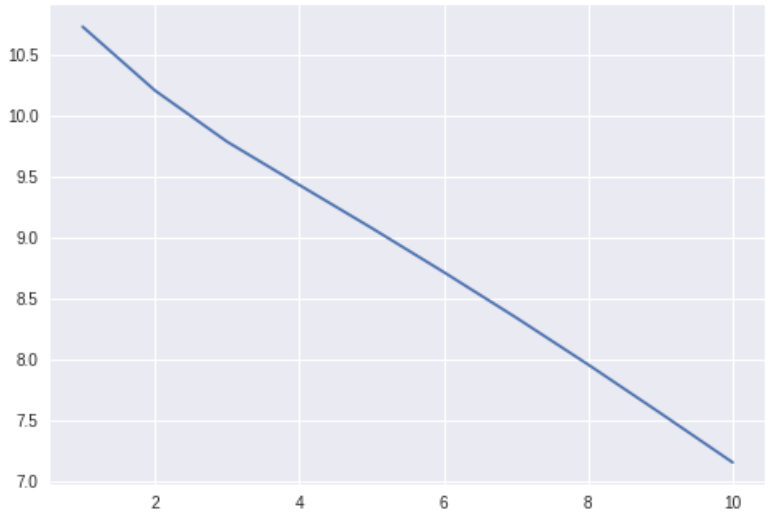
Embedding Dimension 100 train plot:



Embedding Dimension 100 train plot:



In the continuous bag of words model, the current word is predicted based on the context. Window size 2 and embedding dimension of 100 is used. Train data of context and target words is created. Two layer neural network with softmax at the end is used. Training is done for 10 epochs on a small subset of text8 dataset.
Training loss plot is given below:

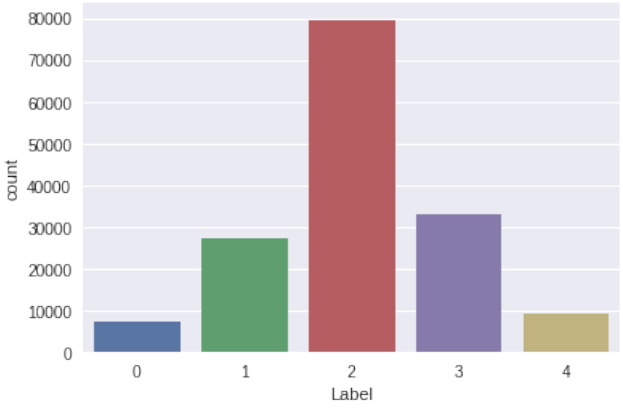


Observations and conclusions: CBOW takes much larger time to train than skip gram. It conerges slower than skip gram; it doesn't converge in 10 epochs whereas skip gram does. Loss in CBOW is very high whereas it is much lower in skip gram. The skip gram model is a much more efficient and accurate model

for wordtovec.

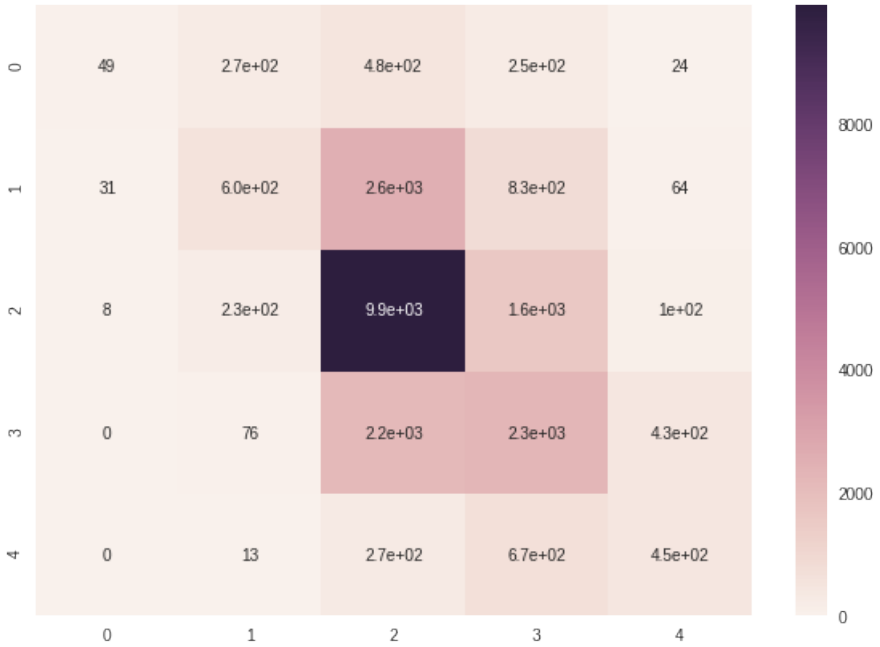
3 Part B

In this part of the assignment, we are required to classify movie reviews in one of the 5 classes based on the sentiment of the review. We first make a feature representation of the review using pre-trained word embedding GloVe(Global Vectors for Word Representation) and then use LSTM to classify into one of the 5 classes.We use a 100 dimensional word embedding. We use keras for this part of the assignment. First, the train and test files are loaded. We then do an analysis of the train set to find the number of phrases that belong to each label.This gives us an idea on whether the dataset is uniformly distributed or biased. A plot of the same is given below:



Clearly, the dataset is biased towards the label 'neutral'. Labels 'negative' and 'positive' are very rare. We then do a one-hot encoding of the labels. A train test split is performed on the provided train set to verify the network we apply for classification. We then use tokeniser to convert texts to sentences. Glove embedding of dimension 100 is loaded. It has 400000 word vectors. All sequences are zero padded to make their lengths uniform. An embedding matrix is then created. The neural network is then defined using LSTM; the embedding layer has non trainable parameters as pre trained GloVe weights are used. Softmax is applied in the final layer. Categorical cross entropy loss is applied. RMSprop optimiser is used and the metric used for checking convergence is accuracy.

The model is then trained. It converges in three epochs with a train score of 0.5831 and a validation score of 0.5719. We then test the model on the test set we have created from the train dataset. We get an accuracy of 0.568. We then plot the confusion matrix:



We then visualise a few predictions. They are as follows:
'will be needed' is classified as 2.
'is a lot like a well-made PB J sandwich : fa..' is classified as 3.
'plummets into a comedy graveyard' is classified as 3 (plummets is a negative word).
'aimless for much of its running time , until..' is classified as 1(until is a positive word).

This dataset is highly biased towards the label 2. A dataset which is more uniformly distributed and not biased towards a particular label can give us better results.

4 References

1.http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
2. https://towardsdatascience.com/word-embedding-with-word2vec-and-fasttext-a209c1d3e12c
3. https://towardsdatascience.com/machine-learning-word-embedding-sentiment-classification-using-keras-b83c28087456
4. https://www.opencodez.com/python/text-classification-using-keras.htm
5. https://github.com/FraLotito/continuous-bag-of-words