**CSE538:Word2Vec Assignment**

* *Submitted by* **Utkarsh Garg**

*SBU#:* **112672834**

**System configuration:**

*OS:* Mac OS 10.14.2

*Python Version*: Python 3.6.3

**Implementation:**

1) **word2vec\_basic.py** - Main file where execution starts. I have implemented my batch method in this file. This method is used to generate labels and batches. Starting from pivot (initialised to skip-window), I store the data in a data buffer whose length is the same as window's size, picking num\_skips words from each side of current location of pivot and gradually move the window until the batch\_size is met. I keep moving the window i.e. keep on updating the data buffer and use it to update the batch and labels.

2) **loss\_func.py** - implementation for Noise contrastive estimation(NCE) and cross entropy functions.

Cross entropy – Here, we use the word vectors of the input word embeddings and the true labels to calculate the loss.

Noise contrastive estimation(NCE) - This method uses negative sampling, we get the regression value over the word vectors and unigram probability by applying sigmoid function.

3) **word\_analogy.py** - Data file containing the list of words is loaded and compare against the trained dev data values. Cosine similarity is used on given and trained data and output is logged accordingly.

This python file also contains code for evaluating similarity between words, given as inputs , against the trained model dictionary embeddings, to find 20 similar words for {first,American,would}.

*Output Files:*

- word2vec\_cross\_entropy.model (Include drive link)

- word2vec\_nce.model (Include drive link)

*Prediction files for word\_analogy\_test.txt:*

- word\_analogy\_dev\_predictions\_cross\_entropy.txt

- word\_analogy\_dev\_predictions\_nce.txt

**Ideal Configuration:**

batch\_size = 64, skip\_window = 2, num\_skips = 4, max\_num\_steps = 400001, num\_samples=64

***Cross Entropy*** - Average loss at step 400000 : 4.032

Best accuracy achieved for dev file on Cross\_Entropy model :

Generated by: score\_maxdiff.pl  
Mechanical Turk File: word\_analogy\_dev\_mturk\_answers.txt  
Test File: word\_analogy\_dev\_predictions\_cross\_entropy.txt  
Number of MaxDiff Questions: 914  
Number of Least Illustrative Guessed Correctly: 299  
Number of Least Illustrative Guessed Incorrectly: 615  
Accuracy of Least Illustrative Guesses: 32.7%  
Number of Most Illustrative Guessed Correctly: 276  
Number of Most Illustrative Guessed Incorrectly: 638  
Accuracy of Most Illustrative Guesses: 30.2%  
Overall Accuracy: 31.5%

***Noise Contrastive Estimation*** - Average loss at step 400000 : 1.36

Best Accuracy Achieved for Dev File on nce model :

Generated by: score\_maxdiff.pl  
Mechanical Turk File: word\_analogy\_dev\_mturk\_answers.txt  
Test File: word\_analogy\_dev\_predictions\_nce.txt  
Number of MaxDiff Questions: 914  
Number of Least Illustrative Guessed Correctly: 297  
Number of Least Illustrative Guessed Incorrectly: 617  
Accuracy of Least Illustrative Guesses: 32.5%  
Number of Most Illustrative Guessed Correctly: 280  
Number of Most Illustrative Guessed Incorrectly: 634  
Accuracy of Most Illustrative Guesses: 30.6%  
Overall Accuracy: 31.6%

Instructions to run the files:

-----------------------------------

1) to run word2vec\_basic.py ( for training the model )

- python word2vec\_basic.py nce

- python word2vec\_basic.py cross\_entropy

2) to run word\_analogy.py ( for generating the predictions using trained model )

- python word\_analogy.py

3) to run score\_maxdiff.pl ( to find the accuracy of the prediction )

- ./score\_maxdiff.pl word\_analogy\_dev\_mturk\_answers.txt word\_analogy\_dev\_predictions\_cross\_entropy.txt output\_cross\_entropy.txt

- ./score\_maxdiff.pl word\_analogy\_dev\_mturk\_answers.txt word\_analogy\_dev\_predictions\_nce.txt output\_nce.txt