System configuration:

OS: Mac OS 10.13.6
Python Version: Python 2.7.10

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. I store the data in a data buffer whose length is the same as window's size and gradually move the window and update the buffer i.e. data buffer will act as the window, initially i fill the data buffer with the data of the first window. 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 dunctions.
Cross entropy - I 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 data values. Cosine similarity is used on given and trained data and output is logged accordingly.

Output Files:
- word2vec_cross_entropy.model
- word2vec_nce.model

- Prediction files for word_analogy_test.txt
  - word_analogy_test_predictions_cross_entropy.txt
  - word_analogy_test_predictions_nce.txt

Ideal Configuration:
batch_size = 64, skip_window = 2, num_skips = 4, max_num_steps = 400001

->cross entropy - Average loss at step 400000 : 4.031693156671524

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_test_predictions_cross_entropy.txt
Number of MaxDiff Questions: 914
Number of Least Illustrative Guessed Correctly: 264

Number of Least Illustrative Guessed Incorrectly: 650
Accuracy of Least Illustrative Guesses: 28.9%
Number of Most Illustrative Guessed Correctly: 333
Number of Most Illustrative Guessed Incorrectly: 581
Accuracy of Most Illustrative Guesses: 36.4%
Overall Accuracy: 32.7%

->nce - Average loss at step 400000 : 1.1813166753590107

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_test_predictions_nce.txt
Number of MaxDiff Questions: 914
Number of Least Illustrative Guessed Correctly: 280
Number of Least Illustrative Guessed Incorrectly: 634
Accuracy of Least Illustrative Guesses: 30.6%
Number of Most Illustrative Guessed Correctly: 330
Number of Most Illustrative Guessed Incorrectly: 584
Accuracy of Most Illustrative Guesses: 36.1%
Overall Accuracy: 33.4%

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_test_predictions_cross_entropy.txt output_cross_entropy.txt
- ./score_maxdiff.pl word_analogy_dev_mturk_answers.txt word_analogy_test_predictions_nce.txt output_nce.txt