GLoVe Problem Set

November 2020

1 Model Analysis

1). Are all co-occurance matricies built by the GLoVe model symmetric? If so, explain your reasoning. If not, provide a counterexample.

2). What is is/are the major difference(s) between GLoVe and the closely related Word2Vec model? Hint: Think about how each of these methods *learn*. What are the advantages and disatvantages of each? More information on Word2Vec can be found here:
https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa

2 Implementation

For your implementation of the GLoVe model, we recommend using a smaller data set to test your code against. For your convenience, we went ahead and created a smaller corpus that can be found at our GitHub page.
GitHub: github.com/tsanford13/MLandOpt/blob/Add-Experiments/vocab.txt
For your implementation, we recommend defining three main functions:

1). make_cooccur(vocab, corpus, window) that takes in a map of words to word ID's (vocab), a list of all of the sentences being considered (corpus), and an integer representing the size of the sliding window used on each sentence in the corpus (window). This function should compute the co-occurance matrix as described in the presentation.

2). train(vocab, cooccur, iters) that takes in a map of words to word ID's (vocab), the co-occurance matrix (cooccur), and an integer (iters) representing the number of iterations to run. This function should initialize the model parameters given the co-occurance matrix and manage training at a high level.

3). glove_iter(vocab, data) that takes in a map of words to word ID's (vocab) and a list of co-occurance data tuples (data). Each tuple should contain any and all information needed to train the model. This function is called iteratively in train() and therefore should only perform one iteration of learning.

Compare your results against pre-trained word vectors available for download at nlp.stanford.edu/projects/glove/