CSC421 Assignment 1

Yubo Wang 1002138377

Part 1

1. For embedding layer, the weight matrix has dimension $250 * 16$ since there are 250 words and each of them has 16 features.
   From embedding layer to hidden layer, the weight matrix has dimension $3 * 16 * 128$ since there are 3 embedding with 16 features and 128units hidden layer. The dimension of the bias vector is simply $1 * 128$
   From hidden layer to output layer, dimension of the weight matrix is $128 * 250$ and dimension of bias vector is $1 * 250$.
   Therefore, the total number of trainable parameters is $250 * 16 + 3 * 16 * 128 + 1 * 128 + 128 * 250 + 1 * 250 = 42522$

2. There are 4 slots in total and each slot can be fitted using 250 words. Therefore, the number of total entries is $250^4 = 3{,}906{,}250{,}000$

Part 2

loss_derivative[2, 5] 0.001112231773782498
loss_derivative[2, 121] -0.9991004720395987
loss_derivative[5, 33] 0.0001903237803173703
loss_derivative[5, 31] -0.7999757709589483

param_gradient.word_embedding_weights[27, 2] -0.27199539981936854
param_gradient.word_embedding_weights[43, 3] 0.8641722267354156
param_gradient.word_embedding_weights[22, 4] -0.25467302023746496
param_gradient.word_embedding_weights[2, 5] 0.0

param_gradient.embed_to_hid_weights[10, 2] -0.6526990313918258
param_gradient.embed_to_hid_weights[15, 3] -0.13106433000472614
param_gradient.embed_to_hid_weights[30, 9] 0.11846774618169388
param_gradient.embed_to_hid_weights[35, 21] -0.1000452610460437

param_gradient.hid_bias[10] 0.2537663873815639
param_gradient.hid_bias[20] -0.033267391636353484

param_gradient.output_bias[0] -2.0627596032173052
param_gradient.output_bias[1] 0.0390200857392169
param_gradient.output_bias[2] -0.7561537928318482
param_gradient.output_bias[3] 0.21235172051123635

Part 3

1. For 3-gram that is not in the dataset, the prediction result is not that good:

   The tri-gram "government of united" did not occur in the training set.
   government of united own Prob: 0.19242
   government of united team Prob: 0.06020
   government of united . Prob: 0.05763
   government of united ? Prob: 0.05491
   government of united life Prob: 0.05265
   government of united states Prob: 0.03151

   For 3-gram presented in the dataset, the prediction result makes more sense.

   The tri-gram "city of new" was followed by the following words in the training set:
   york (8 times)
   city of new york Prob: 0.98472
   city of new . Prob: 0.00562
   city of new life Prob: 0.00101

   For the 3-gram 'life in the', we actually get some plausible predictions like
   life in the world Prob: 0.09968
   life in the united Prob: 0.08340
   life in the city Prob: 0.07089

2. Words in each cluster usually have similar meanings/usage, or simply just word in different tenses.
   For example, (and, or), (what, where, who), (go, come), (make, made)

3. They are not close together since they do not have similar meaning/usage. These two words are related only based on the word 'New York". Words in one cluster in the graph usually can replace each other in a sentence, but word "new" can't replace "York" under any circumstance.

4. Government and university should be closer to each other since they are both noun. Although government and political are more similar based on their meanings, they cannot replace each other in a sentence.

   By using model.word_distance function, we actually get

   1.1456468079733895 for "government" and "university"

   1.639315759575435 for "government" and "political"