CSC321 Assignment 1

Yueyang Zhang 1002091229

February 2018

Warm-up Questions

1.

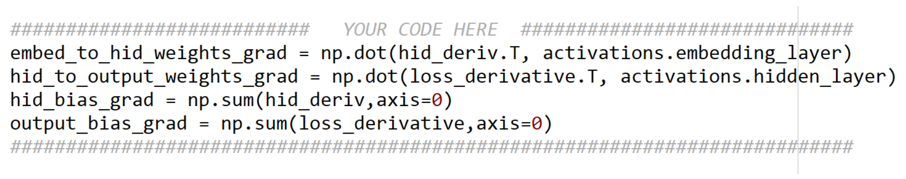
1. Since there are 250 words in the dictionary and the embedding dimension is 16, the dimension of the word embedding weights is 250 * 16. Thus, it contains 4000 trainable parameters.
2. The embedding layer contains 3 words, whose embedding dimension is 16, and thus one dimension of the embedding to hidden weights is 48. Also, due to 128 units in the hidden layer, the dimension of the embedding to hidden weights is 128*48, containing 6144 trainable parameters.
3. The dimension of the hidden bias is 128*1, containing 128 trainable parameters.
4. Because there are 128 units in the hidden layer and 250 words for the input of softmax function, the dimension of the output layer is 250*128, containing 32000 trainable parameters.
5. The dimension of the hidden bias is 250*1, containing 250 trainable parameters.

To sum up, the total number of trainable parameters is $4000 + 6144 + 128 + 32000 + 250 =$ **42522**. The part of hidden layer to output layer has the largest number of trainable parameters, which is **32250**.

2.

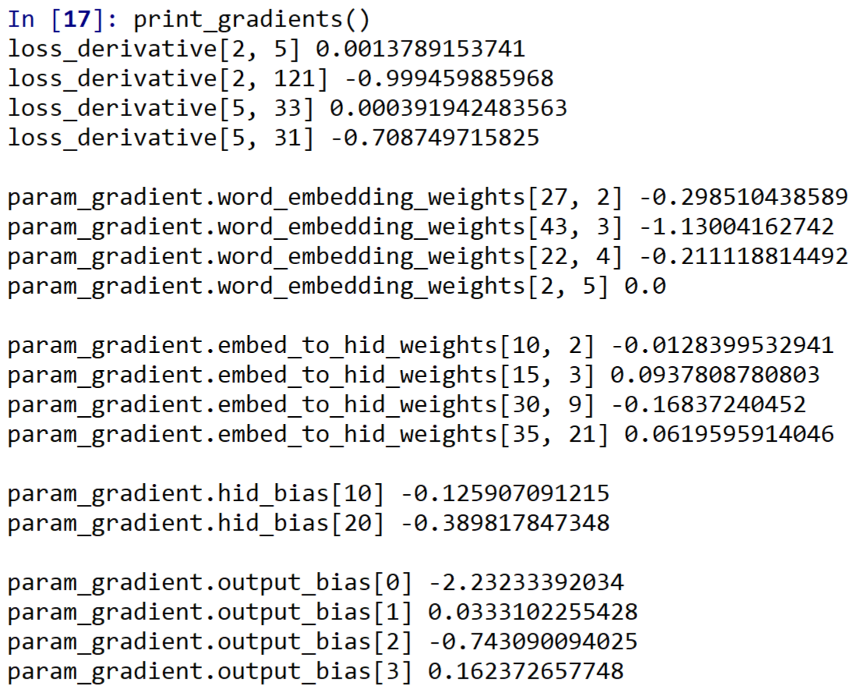
The number of entries in the table is $250^4 = 3906250000$.

Training the model Questions



```
##########################   YOUR CODE HERE   #############################
embed_to_hid_weights_grad = np.dot(hid_deriv.T, activations.embedding_layer)
hid_to_output_weights_grad = np.dot(loss_derivative.T, activations.hidden_layer)
hid_bias_grad = np.sum(hid_deriv,axis=0)
output_bias_grad = np.sum(loss_derivative,axis=0)
##########################################################################
```

Figure 1: The code of gradients



```
In [17]: print_gradients()
loss_derivative[2, 5] 0.0013789153741
loss_derivative[2, 121] -0.999459885968
loss_derivative[5, 33] 0.000391942483563
loss_derivative[5, 31] -0.708749715825

param_gradient.word_embedding_weights[27, 2] -0.298510438589
param_gradient.word_embedding_weights[43, 3] -1.13004162742
param_gradient.word_embedding_weights[22, 4] -0.211118814492
param_gradient.word_embedding_weights[2, 5] 0.0

param_gradient.embed_to_hid_weights[10, 2] -0.0128399532941
param_gradient.embed_to_hid_weights[15, 3] 0.0937808780803
param_gradient.embed_to_hid_weights[30, 9] -0.16837240452
param_gradient.embed_to_hid_weights[35, 21] 0.0619595914046

param_gradient.hid_bias[10] -0.125907091215
param_gradient.hid_bias[20] -0.389817847348

param_gradient.output_bias[0] -2.23233392034
param_gradient.output_bias[1] 0.0333102255428
param_gradient.output_bias[2] -0.743090094025
param_gradient.output_bias[3] 0.162372657748
```

Figure 2: The result of print_gradients()

Analysis Questions

1

1. ”government of united”:
”government of united own” Prob: 0.06786
”government of united states” Prob: 0.06067
”government of united life” Prob: 0.05791

"government of united money" Prob: 0.05206
"government of united ." Prob: 0.05050
"government of united end" Prob: 0.04033
"government of united time" Prob: 0.03520
"government of united house" Prob: 0.03320
"government of united say" Prob: 0.02317
"government of united team" Prob: 0.02231

2. "city of new":
"city of new york" Prob: 0.98987
"city of new ." Prob: 0.00139
"city of new ?" Prob: 0.00080
"city of new ," Prob: 0.00078
"city of new days" Prob: 0.00059
"city of new children" Prob: 0.00058
"city of new times" Prob: 0.00055
"city of new years" Prob: 0.00033
"city of new music" Prob: 0.00033
"city of new people" Prob: 0.00030

3. "life in the":
"life in the world" Prob: 0.15756
"life in the first" Prob: 0.13498
"life in the end" Prob: 0.05317
"life in the united" Prob: 0.04379
"life in the street" Prob: 0.04040
"life in the game" Prob: 0.03705
"life in the country" Prob: 0.03588
"life in the school" Prob: 0.02934
"life in the place" Prob: 0.02903
"life in the city" Prob: 0.02711

4. "he is the":
"he is the best" Prob: 0.22058
"he is the first" Prob: 0.20496
"he is the same" Prob: 0.09879
"he is the only" Prob: 0.02827
"he is the end" Prob: 0.02574
"he is the right" Prob: 0.01936
"he is the man" Prob: 0.01924
"he is the president" Prob: 0.01805
"he is the last" Prob: 0.01644
"he is the law" Prob: 0.01431

Generally, the algorithm can give the sensible predictions.
When making predictions for "city of new", "york" has the probability of nearly

99.0%, which quite makes sense.
When predicting the next word for "life in the" and "he is the", since there are so many reasonable phrase combinations, it does not give a solution with very high probability. But most of the predictions given are sensible.
When predicting the next word for "government of united", although we can see "states" in the solution list, it does not have a high probability. This is because the phrase "government of united states" does not occur in the dataset.

5. I make the prediction for "he is nt". From the solutions, there exist "he is nt going", "he is nt there", "he is nt a", "he is nt right", "he is nt like". All of these phrases are plausible but do not occur in the dataset.

2

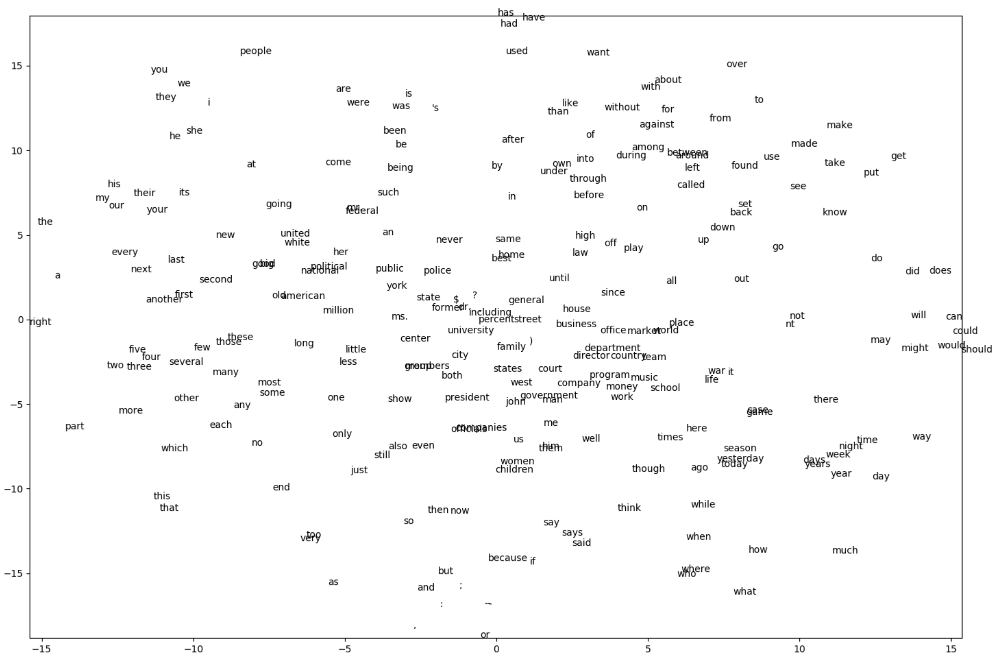


Figure 3: The result of tsne_plot()

1. "its", "their", "your", "our", "my", "his" are clustered together. All of them are pronominal possessive adjectives
2. "should", "will", "may", "can", "might", "would", "could" are clustered together. All of them are modal verbs.
3. "how", "what", "where", "when" are clustered together. All of them are

interrogative words.

3

By using the function of model.word_distance(), the distance between ”new” and ”york” in the map is 3.10966566781, which means they are not quite close. By using the function of model.display_nearest_words(), either of them is not the other’s 10 nearest words. This is because although ”new” and ”york” can form a phrase, they do not have much similarity(word characteristic or meaning) and are not mutable when constituting sentences or phrases.

4

The distance between ”government” and ”political” is 1.22760136884.
The distance between ”government” and ”university” is 1.04925552744.
In this case, the pair of ”government” and ”university” is closer. This is because they are all nouns and all represent institutions, which are relative relevant. On the other hand, ”government” and ”political” are not very far away from each other, since they all belong to the filed of politics.

Optional

I find the warm-up question the most valuable since it gives me an overall picture of the neural network in this assignment.
I find the code part the most difficult since it requires me to figure out what I have and what I can use from the code, which cost me some time.