CSC413 Winter 2020 Assignment 1

Name: Ruijie Sun
SN: 1003326046

Part 1. SOLUTION

(1) Given the vocabulary size V and embedding dimensionality d, how many trainable parameters does the GLoVE model have?
$V(d+1)$

(2) Write the gradient of the loss function with respect to one parameter vector W_i.
$L(\{W_i, b_i\}_{i=1}^V) = \sum_{i\neq p}^V [\sum_{j\neq p}^V (W_i{}^T W_j + b_p + b_j - logX_{ij})^2 + (W_i{}^T W_p + b_i + b_p - logX_{ip})^2] + \sum_{j\neq p}^V (W_p{}^T W_j + b_p + b_j - logX_{pj})^2 + (W_p{}^T W_p + b_p + b_p - logX_{pp})^2$

So $\frac{\partial}{\partial W_i} L(\{W_i, b_i\}_{i=1}^V) = 2\sum_{j=1}^V (2W_j{}^T W_i + 2b_j + 2b_i - logX_{ji} - logX_{ij})W_j = 4\sum_{j=1}^V (W_j{}^T W_i + b_j + b_i - logX_{ij})W_j$

(4) The largest $d = 50$ performs best because the larger d enable to contain more information of the word after training.

Part 2. SOLUTION

(1) As above, assume we have 250 words in the dictionary and use the previous 3 words as inputs. Suppose we use a 16-dimensional word embedding and a hidden layer with 128 units. The trainable parameters of the model consist of 3 weight matrices and 2 sets of biases. What is the total number of trainable parameters in the model? Which part of the model has the largest number of trainable parameters?

word_embedding_weights: 250×16
embed_to_hid_weights: 128×48
hid_bias: 128×1
hid_to_output_weights:250×128
out_put_bias:250×1
$Total = 250 \times 16 + 128 \times 48 + 128 \times 1 + 250 \times 128 + 250 \times 1 = 42522$

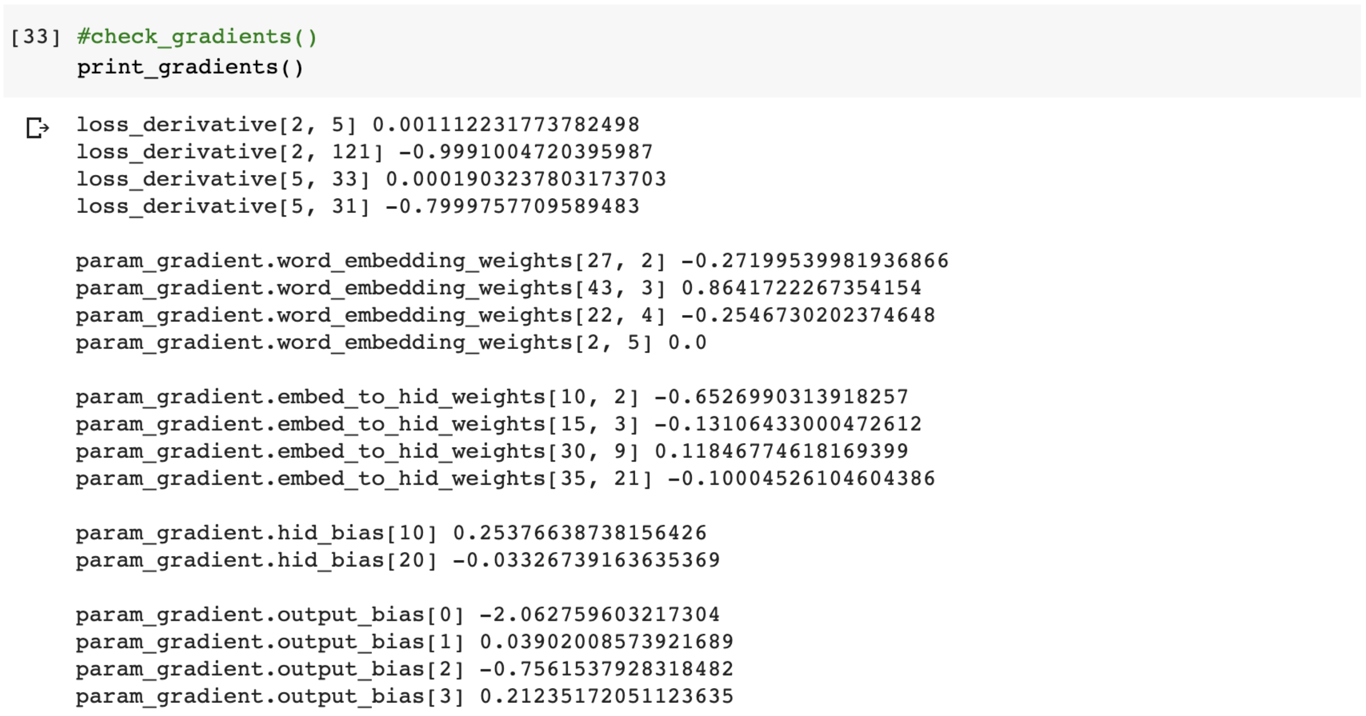
The part of the model has the largest number of trainable parameters: hid_to_output_weights

(2) Another method for predicting the next word is an n-gram model, which was mentioned in Lecture 7. If we wanted to use an n-gram model with the same context length as our network, we'd need to store the counts of all possible 4-grams. If we stored all the counts explicitly, how many entries would this table have?

For the 4-grams model, the given context length is 3. Considering repeating cases, the number of entries in the $table = 250 \times 250 \times 250 \times 250 = 3906250000$

Part 3. SOLUTION

The output of checking.print_gradients():



```
[33] #check_gradients()
     print_gradients()
```

```
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] -0.27199539981936866
param_gradient.word_embedding_weights[43, 3] 0.8641722267354154
param_gradient.word_embedding_weights[22, 4] -0.2546730202374648
param_gradient.word_embedding_weights[2, 5] 0.0

param_gradient.embed_to_hid_weights[10, 2] -0.6526990313918257
param_gradient.embed_to_hid_weights[15, 3] -0.13106433000472612
param_gradient.embed_to_hid_weights[30, 9] 0.11846774618169399
param_gradient.embed_to_hid_weights[35, 21] -0.10004526104604386

param_gradient.hid_bias[10] 0.25376638738156426
param_gradient.hid_bias[20] -0.03326739163635369

param_gradient.output_bias[0] -2.062759603217304
param_gradient.output_bias[1] 0.03902008573921689
param_gradient.output_bias[2] -0.7561537928318482
param_gradient.output_bias[3] 0.21235172051123635
```

Part 4. SOLUTION

(1) Pickthreewordsfromthevocabularythatgowelltogether(forexample,'government of united', 'city of new', 'life in the', 'he is the' etc.). Use the model to predict the next word. Does the model give sensible predictions? Try to find an example where it makes a plausible prediction even though the 4-gram wasn't present in the dataset (raw_sentences.txt). To help you out, the function language_model.find_occurrences lists the words that appear after a given 3-gram in the training set.

Yes, the model gives sensible prediction.

And "life in the country" is an example where it makes a plausible prediction even thought it wasn't present in the dataset.

(2) Plot the 2-dimensional visualization using the method Model.tsne_plot. Look at the plot and find a few clusters of related words. What do the words in each cluster have in common? (You don't need to include the plot with your submission.)

The words in same cluster share same property in terms of grammar. For example, "may","might", "will","can","should","might", and "would" are all Modal verbs. And "I","he","she","you",and "they" are all PersonalPronouns.

(3) Are the words 'new' and 'york' close together in the learned representation? Why or why not?

No. From the results of TSNE Plot, function display_nearest_words('new',10) and word_distance('new', 'york'), we can know that 'new' and 'york' are not in the same cluster in TSNE Plot, 'york' does not appear in 'new' top 10 neartest words and their distance is 3.8492334683809077. So 'new' and 'york' are not close to each other in learned representation. The reason behind it is that 'new' is adjective and 'york' is noun. In other words, they have different properties in term of grammar.

(4) Which pair of words is closer together in the learned representation: ('government', 'political'), or ('government', 'university')? Why do you think this is?

('government', 'university'). Because they are both noun and they share same property in term of grammar. Furtherly, we can justify the conclusion by comparing word distance of ('government', 'political') and word distance of ('government', 'university'). The first one is 1.456761070875144 and the second one is 1.142257412653768.