CSC321: Assignment #1

Xiangyu Kong
kongxi16

February 5, 2018

Problem 1

1. For the word embedding weights, since there are 250 words in total and each word has 16 features, the weight matrix's dimension is $250 \times 16 = 4000$

   For the embedding to hidden weights, there are 3 embeddings each with 16 features and the hidden layer consists of 128 units, so the dimension of the weight matrix is $3 \times 16 \times 128 = 6144$ and the dimension for the bias vector is 1×128

   For the hidden to output weight, there are 128 hidden units in the hidden layer and the output is the softmax over 250 words, so the dimension is $128 \times 250 = 32000$ and the bias has dimension of 1×250.

   Thus the total number of trainable features is $4000 + 6144 + 128 + 32000 + 250 = 42522$ and the layer with the largest number of trainable features is from hidden to output.

2. Since there are 250 total vocabulary, there are $250^3 = 15625000$ permutations of 3-word prefixes. Combining with 250 possible predictions, the total number of entries is $15625000 * 250 = 3906250000$.

Problem 2

Listing 1: Print Gradient Result

```
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
```

Problem 3

1. The words "he had some" appeared in the training set for twice, and the predictions are:

```
The tri-gram ``he had some" was followed by the following words in the training
        life (1 time)
        years (1 time)

he had some time Prob: 0.20988
he had some of Prob: 0.16279
he had some more Prob: 0.07394
he had some money Prob: 0.06740
he had some . Prob: 0.06341
he had some to Prob: 0.02488
he had some children Prob: 0.02266
he had some life Prob: 0.02182
he had some good Prob: 0.01899
he had some people Prob: 0.01821
```

   The top predictions make perfect sense, and the result is very interesting because although the words appeared in the training set, the most likely predictions do not include the training set words (life and years).

   The words I chose were "it was president", and the predictions are:

```
The tri-gram ``it was president" did not occur in the training set.
it was president . Prob: 0.16863
it was president of Prob: 0.16586
it was president , Prob: 0.14467
it was president for Prob: 0.07258
it was president now Prob: 0.04877
it was president ? Prob: 0.02850
it was president in Prob: 0.02043
it was president to Prob: 0.01792
it was president and Prob: 0.01520
it was president at Prob: 0.01456
```

   As shown above, "it was president." and "it was president of" make sense in a normal context.

2. A cluster would be { may, might, will, would , could, should, can }. These are all auxiliary verbs.

   Another cluster is { my, your, his, our, their }. These are all possessive determiners.

   A cluster means a set of words have the same structural positions in a sentence.

3. The words "new" and "york" are close together. This is because they are a fixed combination and are used together very often.

4. "government" is closer to "political" and this is because they are more closely related and they are used together more often.