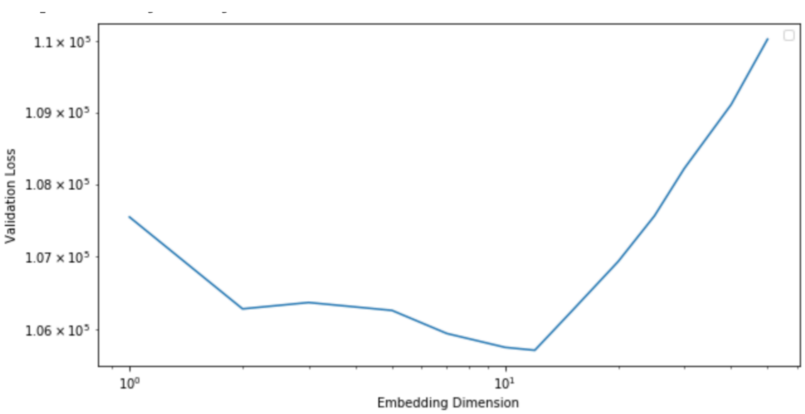
Question 1

1)

Output weight = d * V;
Output bias = V;
trainable parameters = V * d + V

2)
$$\frac{\partial L}{\partial w_i} = 2(\Sigma_{j=1}^{V}(w_i^T w_j + b_i + b_j - logX_{ij}) + \Sigma_{j=1}^{V}(w_j^T w_i + b_i + b_j - logX_{ij})$$
$$-(w_j^T w_j + b_i + b_j - logX_{ij}))w_j$$
$$= 2(2\Sigma_{j=1}^{V}(w_i^T w_j + b_i + b_j - logX_{ij}) - (w_j^T w_j + b_i + b_j - logX_{ij}))w_j$$
$$= 2(2\Sigma_{j\neq 1}^{V}(w_i^T w_j + b_i + b_j - logX_{ij}) + (w_j^T w_j + b_i + b_j - logX_{ij}))w_j$$
$$= 4\Sigma_{j=1}^{V}(w_i^T w_j + b_i + b_j - logX_{ij})w_j$$

4)



d = 12 leads to optimal validation performance, and larger d doesn’t always lead to better validation error since it will has higher chance to cause overfit.

Question 2

1)
Word embedding weights = 250 * 16 = 4000;
Embed to hid weights = 128 * 16 * 3 = 6144;
Hid bias = 128 * 16 * 1 = 128;
Hid to output weights = 250 * 128 = 32000
Output bias = 250 * 1 = 250;
total number of trainable parameters = 4000 + 6144 + 128 + 32000 + 250 = 42552
The total number of trainable parameters in the model is 42552, and the hid to output weights has the largest number of trainable parameters.

2)
We need 250^4 entries table to store all the possible of 4-grams for 250 words.

Question 3

```
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] 0.2537663873815642
param_gradient.hid_bias[20] -0.03326739163635357

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
```

Question 4

1)

```
1 trained_model.predict_next_word('life', 'in', 'the')
2 find_occurrences('life', 'in', 'the')
```

```
life in the world Prob: 0.14804
life in the united Prob: 0.05100
life in the right Prob: 0.04756
life in the game Prob: 0.04656
life in the first Prob: 0.03991
life in the market Prob: 0.03468
life in the country Prob: 0.03441
life in the place Prob: 0.03420
life in the city Prob: 0.03085
life in the end Prob: 0.02786
The tri-gram "life in the" was followed by the following words in the training set:
    big (7 times)
    united (2 times)
    world (1 time)
    department (1 time)
```

I use the example of "life in the" as the example. The training set only has occurrence for the "big", "united", "world" and "department", but the model gives the plausible prediction on some other words like "city"with probability of 3.85% and "game" with probability of 4.65% etc.

2)
the words in each cluster by the method tens_plot_representation are having similar meaning and can easily replace each other without changing the meaning of the sentence.
The method tens_plot_glove_representation have some short phrases are close to each other, such as "long" and "ago" or "new york city".
The t-SNE has the result words in some clusters but the method plot_2d_GLoVE_representation will have the result words more evenly separated in the 2D graph.

3)
'new' and 'york' are not close together in the learned representation, even through these two words are used together a lot. 'new' is an a djective and 'york' is a verb or noun. Thus they have different meanings and they are different type of words. Also, the word distance between these two words is 4.04920709249407.

4)
The word distance of 'government' and 'political' is 1.4759537372355684 and distance of 'government' and 'university' is 1.1895426605900286. I think the 'government' and 'university' are closer because they are both nouns and have the meaning of institution or agency. On the other aspect, in the training data there might exists a lot similar adjective in front on these two words. Thus the model will has the result that 'government' and 'university' are closer.