Assignment 5

1 a) In Assignment 4 we used 256-dimensional word embeddings (e_{word} = 256), while in this assignment, it turns out that a character embedding size of 50 suffices (e_{char} = 50). In 1-2 sentences, explain one reason why the embedding size used for character-level embeddings is typically lower than that used for word embeddings.

Ans) The embedding size for characters is smaller than the embedding size for words because the character vocabulary is much smaller than the word vocabulary, therefore a smaller embedding size is sufficient to represent each character.

b) Write down the total number of parameters in the character-based embedding model (Figure 2), and then do the same for the word-based lookup embedding model (Figure 1). Write each answer as a single expression (though you may show working) in terms of e_{char}, k, e_{word}, V_{word} (the size of the word-vocabulary in the lookup embedding model) and V_{char} (the size of the character-vocabulary in the character-based embedding model). Given that in our code, k = 5, V_{word} ≈ 50, 000 and V_{char} = 96, state which model has more parameters, and by what factor (e.g. twice as many? a thousand times as many?).

Ans) For character-based embedding model

No. of parameters $= e_{char}*V_{char} + f*e_{char}*k + f + e_{word}*e_{word} + e_{word} + e_{word}*e_{word} + e_{word}$

$= 50*96 + 256*50*5 + 256 + 256*256 + 256 + 256*256 +256$

$= 4800 + 64000 + 256 + 65536 +256 + 65536 + 256$

$= 200640$

For word-based embedding model

No. of parameters $= e_{word}*V_{word}$

$= 256*50000$

$= 12800000$

The word embedding model has more parameters by factor of approximately 64.

c) In step 3 of the character-based embedding model, instead of using a 1D convnet, we could have used a RNN instead (e.g. feed the sequence of characters into a bidirectional LSTM and combine the hidden states using max-pooling). Explain one advantage of using a convolutional architecture rather than a recurrent architecture for this purpose, making it clear how the two contrast. Below is an example answer; you should give a similar level of detail and choose a different advantage.

When a 1D convnet computes features for a given window of the input, those features depend on the window only – not any other inputs to the left or right. By contrast, a RNN needs to compute the hidden states sequentially, from left to right (and also right to left, if the RNN is bidirectional). Therefore, unlike a RNN, a convnet's features can be computed in parallel, which means that convnets are generally faster, especially for long sequences.

Ans) A convnet allows us to compute several local features of the characters in a particular window size (i.e. the size of the filter) while a RNN computes hidden states sequentially, hence it might not be able to capture only local features, because in theory even the first character can influence the last character. For some applications we might require these local features which can be computed efficiently and reliably by a convnet.

d) In lectures we learned about both max-pooling and average-pooling. For each pooling method, please explain one advantage in comparison to the other pooling method. For each advantage, make it clear how the two contrast, and write to a similar level of detail as in the example given in the previous question.

Ans) Max pooling takes the prominent i.e. the highest activation. The advantage is that the max value will sometimes be best value to describe its region. The disadvantage is that it discards the other values. In contrast average pooling takes the average of all the values, thereby preserving the information from the values of the region. The disadvantage is that because average pooling does not produce the most prominent signal it is sensitive to outliers and may not always produce the best word representations.

h) Highway Network: Once you've finished testing your module, write a short description of the tests you carried out, and why you believe they are sufficient.

Ans) The first test I performed was to make sure that all the tensors had the correct shape. The next test I performed was given an input tensor do the output an all the intermediate values match the ones calculated manually. A subtest of this test was to initialize the input as tensor of zeros. I think these tests are sufficient because they care of all the different cases that might be encountered at run time.

i) CNN: Once you've finished testing your module, write a short description of the tests you carried out, and why you believe they are sufficient.

Ans) The first test I performed was to make sure that all the tensors had the correct shape. The next test I performed was given an input tensor do the output an all the intermediate values match the ones calculated manually. A subtest of this test was to initialize the input as tensor of zeros. I think these tests are sufficient because they care of all the different cases that might be encountered at run time.