Lab 6: Text Based Data.

Data Set

We will now work with the IMDB review sentiment analysis dataset. This dataset is quite small with only 25,000 examples and it is binary data. We will use it to try and classify various text reviews for either a positive or negative sentiment (if someone liked the film or not). This is a classical NLP problem and would be good to understand if you plan to work in industry.

Problem

The goal of this lab will be to understand how RNNs work and how text-based data is processed and trained on. You will also compare 1D CNNs and LSTMs on sequential text data. You will need to submit your code and report. The report should include results. Also, you need to discuss your ideas and conclusions about the results. (e.g. You can say why a certain network architecture might be better than another.) Please use the starter python code.

1. We are going to import the IMDB dataset from keras.
2. We will need to pad the training and testing data. Keras has a built-in method to do this.
3. Now create a baseline to compare all our models to. We will use a 1D CNN as our baseline. We first need to use a trainable embedding layer as the input. Next we add a small amount of dropout to help prevent overfitting. Next we will add a 1D Convolutional layer with 64 filters that uses relu and has a stride of 1. Next we will apply BatchNorm to help minimize overfitting, we will apply a BatchNorm after every layer. Repeat the last two steps and add another Conv and BatchNorm layer. Now let's add a global pooling 1D layer; this will reduce overfitting and help our model retain a spatial understanding of the data. Finally, we will add a 'brain' to our model which will be a dense layer with 512 neurons and .5 dropout. Send this into a dense layer with 1 neuron and sigmoid activation since we are working with binary data. We will use binary crossentropy and Nadam to train. The batch size will be 128 and we will have 10 epochs.
   Is the model still overfitting even after everything we did? How could we fix this?
4. Apply those fixes to a network to try and create a better baseline. What worked best?
5. Now change the filters from 64 to 250. How does increasing the number of filters affect training and overfitting?
6. Now let us add an additional pair of Convolutional and BatchNorm layers. How did this affect training? Did these two changes have the same affect that they did in 2D CNNs from lab 5? Why do you think that is?
7. Now make a LSTM model to compare to our baseline. It will have the same input embedding layer. Next it will have two LSTM layers with the first one having 'return_sequences=True' (If you do not turn on return sequences than the model will not be able to train using two LSTM layers. If you are only using 1 LSTM layer you don't need to do this). Now add the same ending of a dense 512 with .5 dropout and a dense 1 with sigmoid. We will train using the same optimizer and loss the only change will be to reduce the number of epochs as RNNs take longer to train than CNNs. How does the LSTM compare to our baseline? Is it overfitting as much?

8. Now let us add another LSTM layer to the model. Remember to add in the return sequences parameter. How did increasing the number of LSTM layers affect training and overfitting? Based off of your knowledge of RNNs why do you think this is?
9. Next let us use larger LSTM layers. Increase the layer to 256 instead of 128. This will let us train on longer and longer sequences at a time. How did this affect training? Why do you think that is?
10. Finally let's work with a bi-directional LSTM and compare it to our baseline and our other LSTM model. All you need to do is add in the Bidirectional( ) wrapper around each LSTM layer. How did this affect training? Do you think we should only be using Bidirectional models for sequential data?
11. Now give an analysis of Attention mechanisms from this blog https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html. How do you think attention compares to the classical approach and should we abandon the RNN approach in favor of Attention ResNet CNNs?

Grading:

Each number with a question or questions is valued as equal points. Please post your code to the discussion board, but that will not affect your grade.