COMP4211 PA2 Report

Name: TANG Wai Tin
Student ID: 20421717

Environment

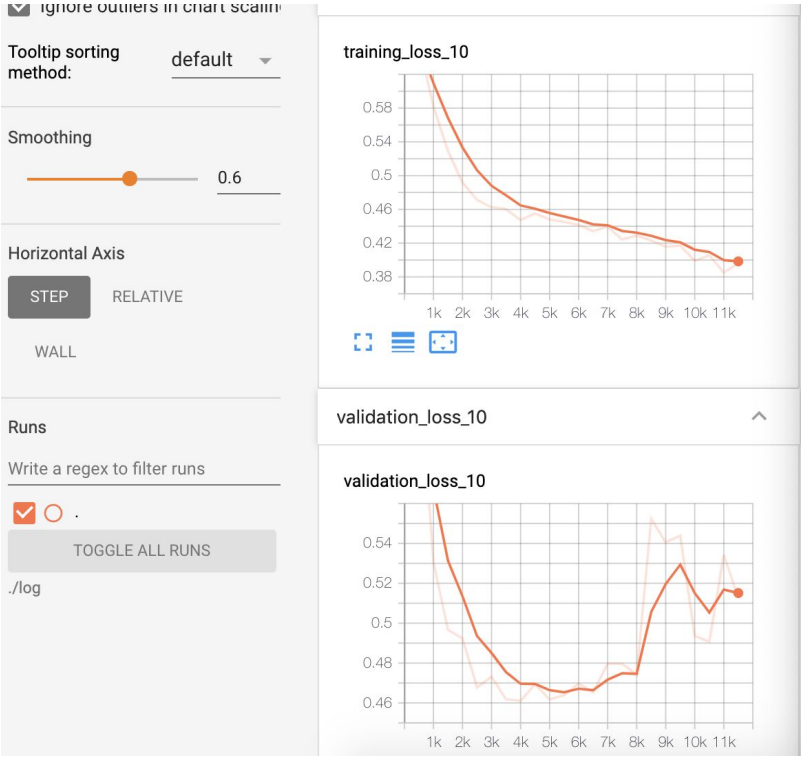
I used Google Colab to complete all tasks.
Create folder ‘Colab Notebooks’ on root folder of Google Drive, then put the datas and codes inside the folder. Read READMD.md for more details.

Part 1: Movie Genre Prediction

1. Print model architecture using torchsummary.summary and include it in the written report.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 16, 110, 110]           1,216
              ReLU-2         [-1, 16, 110, 110]               0
         MaxPool2d-3           [-1, 16, 55, 55]               0
            Conv2d-4           [-1, 32, 26, 26]           8,224
              ReLU-5           [-1, 32, 26, 26]               0
            Conv2d-6           [-1, 64, 12, 12]          32,832
         MaxPool2d-7             [-1, 64, 6, 6]               0
              ReLU-8             [-1, 64, 6, 6]               0
            Conv2d-9             [-1, 96, 4, 4]          55,392
             ReLU-10             [-1, 96, 4, 4]               0
           Conv2d-11            [-1, 128, 1, 1]         196,736
          Sigmoid-12            [-1, 128, 1, 1]               0
           Linear-13                  [-1, 32]           4,128
      BatchNorm1d-14                  [-1, 32]              64
           Linear-15                   [-1, 7]             231
================================================================
Total params: 298,823
Trainable params: 298,823
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 3.78
Params size (MB): 1.14
Estimated Total Size (MB): 5.50
----------------------------------------------------------------
```

2. Paste the screenshots of the plots and report the final training and validation losses obtained.



Final training loss: 0.39621920, validation loss: 0.51251315

3. Discuss the changes and improvement and report the final hyperparameter setting.

   To improve the model, the first thing that I noticed is that the training time is slow. Therefore I try to modify the last Conv2d layer, to output a larger shape, then I could add one more linear and batchnorm layer to make it faster. Accuracy is approximately the same after two changes.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 16, 110, 110]           1,216
              ReLU-2         [-1, 16, 110, 110]               0
         MaxPool2d-3           [-1, 16, 55, 55]               0
            Conv2d-4           [-1, 32, 26, 26]           8,224
              ReLU-5           [-1, 32, 26, 26]               0
            Conv2d-6           [-1, 64, 12, 12]          32,832
         MaxPool2d-7             [-1, 64, 6, 6]               0
              ReLU-8             [-1, 64, 6, 6]               0
            Conv2d-9             [-1, 96, 4, 4]          55,392
             ReLU-10             [-1, 96, 4, 4]               0
           Conv2d-11            [-1, 128, 2, 2]         110,720
          Sigmoid-12            [-1, 128, 2, 2]               0
          Linear-13                  [-1, 128]          65,664
      BatchNorm1d-14                  [-1, 128]             256
          Linear-15                   [-1, 32]           4,128
      BatchNorm1d-16                   [-1, 32]              64
          Linear-17                    [-1, 7]             231
================================================================
Total params: 278,727
Trainable params: 278,727
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 3.79
Params size (MB): 1.06
Estimated Total Size (MB): 5.43
----------------------------------------------------------------
```

4. Paste the screenshots of the classification reports. Describe how varying theta affects the precision and recall. Explain the reason behind.

	precision	recall	f1-score	support
0	0.37	0.28	0.31	413
1	0.16	0.69	0.26	254
2	0.14	0.55	0.22	148
3	0.43	0.94	0.59	774
4	0.60	0.00	0.01	1151
5	0.30	0.27	0.29	259
6	1.00	0.00	0.01	418
micro avg	0.30	0.34	0.32	3417
macro avg	0.43	0.39	0.24	3417
weighted avg	0.51	0.34	0.23	3417
samples avg	0.31	0.32	0.28	3417

Theta too high of too low would make both precision and recall become lower. As too high value of theta would leads to the situation that label are mostly not correctly identified, precision and recall value depends on correctly identified labels would sharply decrease. The situation also true for too low value of theta.

5. Given that cost of misclassification is higher than that of missing the true labels, how should you set the threshold and why?

   Set threshold lower than original. It is because recall is based on misclassification, and precision would also be lower when corrects is lower. It creates the case that misclassification pays higher cost.

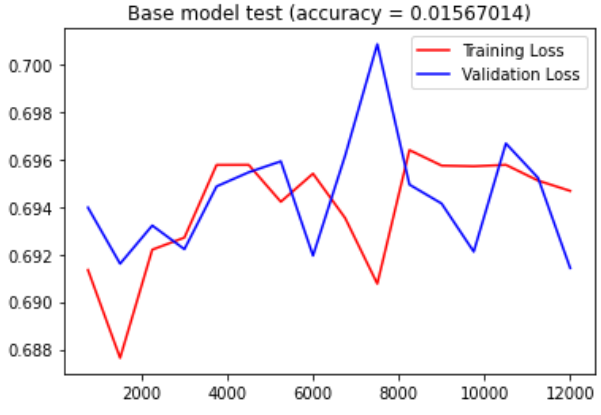
Part 2: Movie Review Sentiment Prediction

6. Print model architecture and the number of trainable parameters for the model and include it in written report

```
The model architecture:

 RNN(
  (emb_layer): Embedding(51167, 64)
  (fc): Linear(in_features=128, out_features=1, bias=True)
  (sig): Sigmoid()
  (rnn_layer): GRU(64, 128, batch_first=True, dropout=0.5)
)

The model has 3,349,313 trainable parameters

The model architecture:

 RNN(
  (emb_layer): Embedding(51167, 64)
  (fc): Linear(in_features=128, out_features=1, bias=True)
  (sig): Sigmoid()
  (rnn_layer): LSTM(64, 128, batch_first=True, dropout=0.5)
)

The model has 3,374,145 trainable parameters

The model architecture:

 RNN(
  (emb_layer): Embedding(51167, 64)
  (fc): Linear(in_features=256, out_features=1, bias=True)
  (sig): Sigmoid()
  (rnn_layer): LSTM(64, 128, batch_first=True, dropout=0.5, bidirectional=True)
)
```

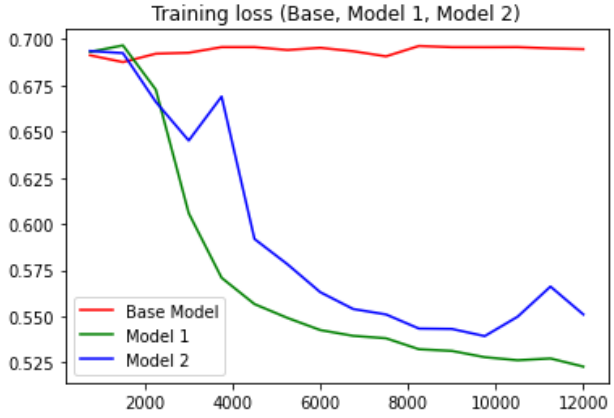
```
The model architecture:

 RNN(
  (emb_layer): Embedding(51167, 64)
  (fc): Linear(in_features=128, out_features=1, bias=True)
  (sig): Sigmoid()
  (rnn_layer): LSTM(1, 128, batch_first=True, dropout=0.5)
)

The model has 3,341,889 trainable parameters
```

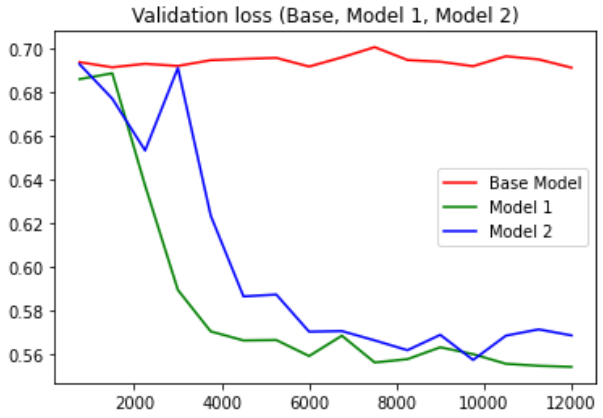
7. Paste screenshot of the plot and report the test accuracy.



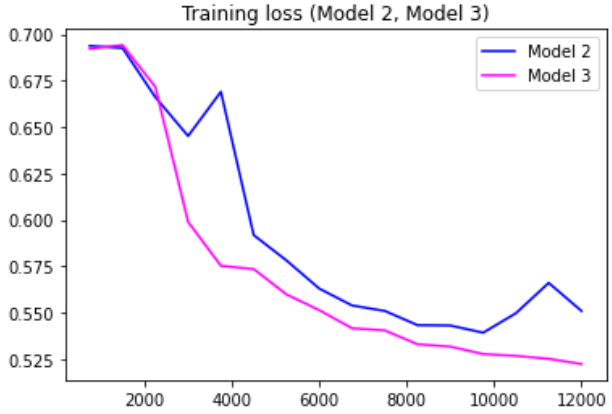
Test accuracy is 0.01567014.

8. Plot training and validation losses overtime of baseline model, model 1 and model 2 in one plot. Explain your observation and suggest some possible reasons.



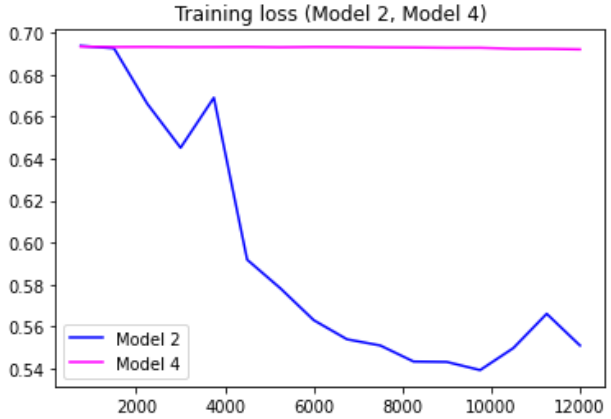


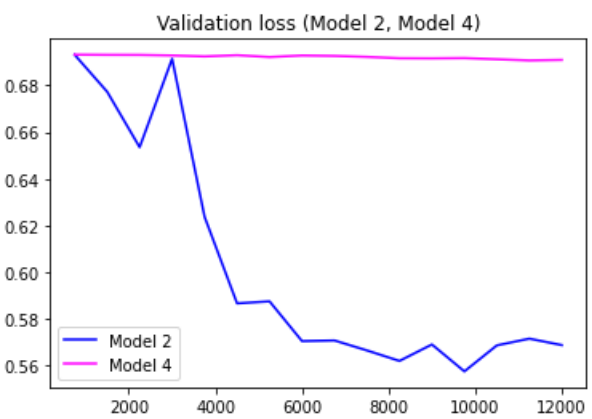
9. Plot training and validation losses over time of model 2 and model 3 in one plot. Explain effect of adding backward direction in LSTM compare to the standard one.



We can see that the curve of model 3, which is bidirectional is flatter than standard one. Also, for the accuracy, model 3 is more stable because it scan backward and modify better on its weight. Therefore, the accuracy of model 3 is increasing when epochs increase but not for model 2.

10. Plot the training and validation losses over model 2 and 4 in one plot. What is effect of removing embedding layer?





Without the embedding layer, we can see that the curve is flat line. That means removing embedding layer would leads to the model does not learn.

11. Rank performance of baseline model, model 1 to 4 according to accuracy.

    Models last evaluation training accuracy:

Baseline	Model 1	Model 2	Model 3	Model 4
0.01567014	0.02805556	0.02701042	0.02806597	0.01608681

    Rank: 3 > 1 > 2 > Baseline > 4

12. Report L2 Distances of three word pairs.