IST 597: Deep learning
Assignment 10000
Report-tfv5097

GITHUB:https://github.com/thejasvi204/IST597_spring22

Bert Analysis:

I initially took 3 models small Bert, ALBERT and Electra with batch size 64,32 and 128 having learning rate as 3e-5. For all models and batch sizes and below are the train and testing losses:

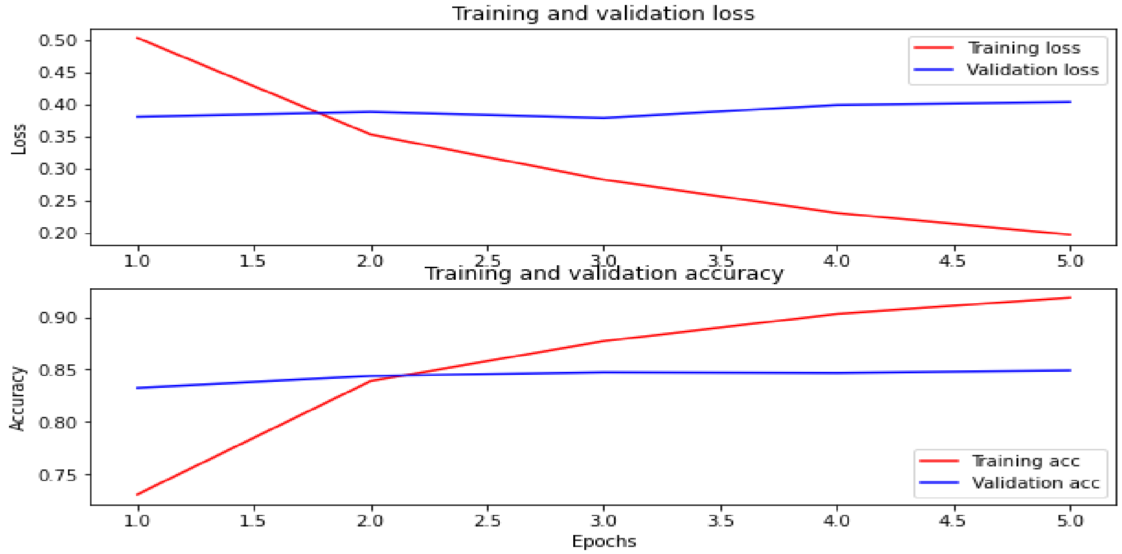
Electra with batch size 64 and 3e-5:

```
Training model with https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-512_A-8/1
Epoch 1/5
313/313 [==============================] - 336s 1s/step - loss: 0.5027 -
binary_accuracy: 0.7304 - val_loss: 0.3799 - val_binary_accuracy: 0.8322
Epoch 2/5
313/313 [==============================] - 324s 1s/step - loss: 0.3526 -
binary_accuracy: 0.8386 - val_loss: 0.3874 - val_binary_accuracy: 0.8436
Epoch 3/5
313/313 [==============================] - 322s 1s/step - loss: 0.2822 -
binary_accuracy: 0.8768 - val_loss: 0.3780 - val_binary_accuracy: 0.8472
Epoch 4/5
313/313 [==============================] - 323s 1s/step - loss: 0.2305 -
binary_accuracy: 0.9028 - val_loss: 0.3984 - val_binary_accuracy: 0.8466
Epoch 5/5
313/313 [==============================] - 323s 1s/step - loss: 0.1967 -
binary_accuracy: 0.9184 - val_loss: 0.4030 - val_binary_accuracy: 0.8490
```

Test loss and accuracy:

```
391/391 [==============================] - 194s 496ms/step - loss: 0.3804 -
binary_accuracy: 0.8553
Loss: 0.38042062520980835
Accuracy: 0.8552799820899963
```

Graphs:



Results from the saved model:

```
input: this is such an amazing movie! : score: 0.999026
input: The movie was great!           : score: 0.991501
input: The movie was meh.              : score: 0.760636
input: The movie was okish.           : score: 0.029231
input: The movie was terrible...      : score: 0.003795

Results from the model in memory:
input: this is such an amazing movie! : score: 0.999026
input: The movie was great!           : score: 0.991501
input: The movie was meh.              : score: 0.760636
input: The movie was okish.           : score: 0.029231
input: The movie was terrible...      : score: 0.003795
```

Electra with batch size 128 and 3e-5:

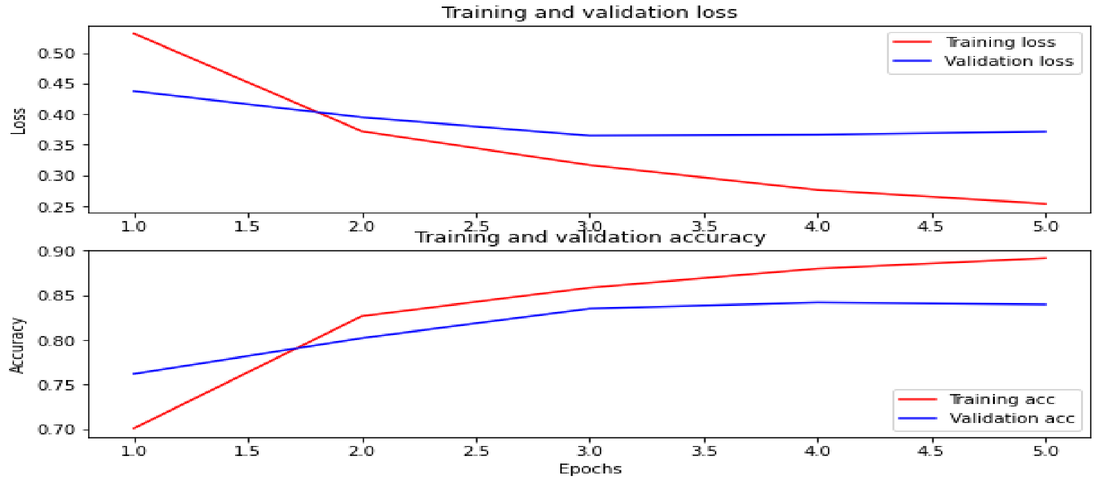
```
Training model with https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-512_A-8/1
Epoch 1/5
157/157 [==============================] - 401s 2s/step - loss: 0.5312 -
binary_accuracy: 0.7012 - val_loss: 0.4372 - val_binary_accuracy: 0.7620
Epoch 2/5
```

```
157/157 [==============================] - 383s 2s/step - loss: 0.3719 -
binary_accuracy: 0.8266 - val_loss: 0.3947 - val_binary_accuracy: 0.8020
Epoch 3/5
157/157 [==============================] - 382s 2s/step - loss: 0.3167 -
binary_accuracy: 0.8583 - val_loss: 0.3647 - val_binary_accuracy: 0.8350
Epoch 4/5
157/157 [==============================] - 384s 2s/step - loss: 0.2763 -
binary_accuracy: 0.8796 - val_loss: 0.3663 - val_binary_accuracy: 0.8418
Epoch 5/5
157/157 [==============================] - 383s 2s/step - loss: 0.2537 -
binary_accuracy: 0.8913 - val_loss: 0.3715 - val_binary_accuracy: 0.8396
```

Test loss and accuracy:

```
196/196 [==============================] - 257s 1s/step - loss: 0.3566 -
binary_accuracy: 0.8481
Loss: 0.35659059882164
Accuracy: 0.8481199741363525
```

Graphs:



Results from the saved model:

```
input: this is such an amazing movie! : score: 0.998267
input: The movie was great!           : score: 0.990168
input: The movie was meh.             : score: 0.626642
input: The movie was okish.           : score: 0.245236
input: The movie was terrible...      : score: 0.007207
```

```
Results from the model in memory:
input: this is such an amazing movie! : score: 0.998267
input: The movie was great!           : score: 0.990168
input: The movie was meh.             : score: 0.626642
```

```
input: The movie was okish.          : score: 0.245236
input: The movie was terrible...     : score: 0.007207
```

ALBERT with batch size 32 and 3e-5:

```
Training model with https://tfhub.dev/tensorflow/albert_en_base/2
Epoch 1/5
625/625 [==============================] - 795s 1s/step - loss: 0.3876 -
binary_accuracy: 0.8109 - val_loss: 0.2994 - val_binary_accuracy: 0.8710
Epoch 2/5
625/625 [==============================] - 795s 1s/step - loss: 0.2555 -
binary_accuracy: 0.8925 - val_loss: 0.3047 - val_binary_accuracy: 0.8788
Epoch 3/5
625/625 [==============================] - 800s 1s/step - loss: 0.1611 -
binary_accuracy: 0.9434 - val_loss: 0.4454 - val_binary_accuracy: 0.8738
Epoch 4/5
625/625 [==============================] - 800s 1s/step - loss: 0.0915 -
binary_accuracy: 0.9743 - val_loss: 0.6646 - val_binary_accuracy: 0.8764
Epoch 5/5
625/625 [==============================] - 800s 1s/step - loss: 0.0491 -
binary_accuracy: 0.9884 - val_loss: 0.6789 - val_binary_accuracy: 0.8836
```

Test loss and accuracy:

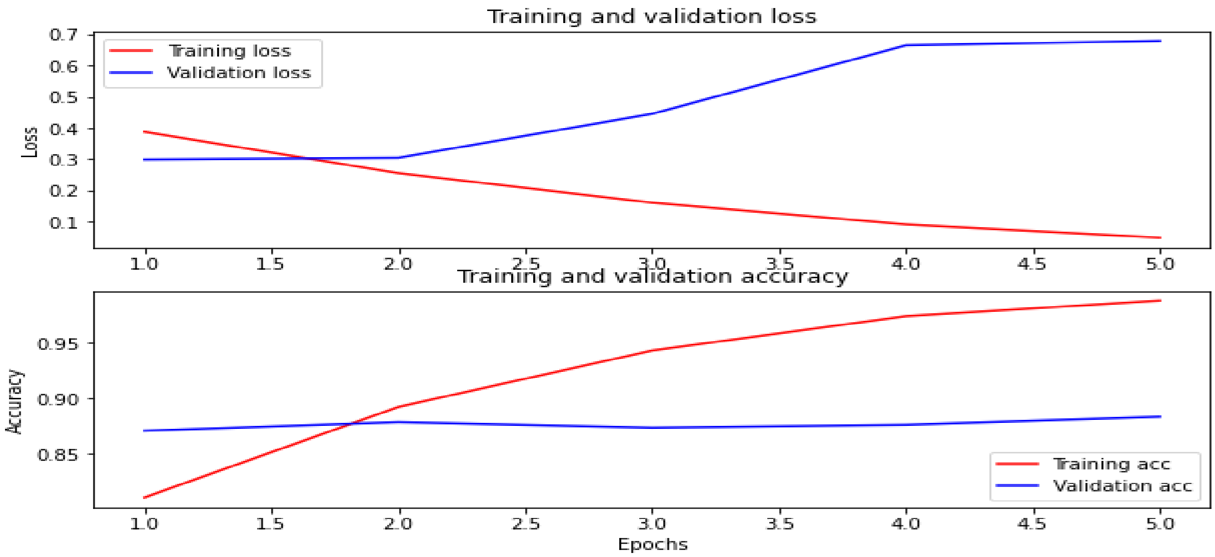
```
782/782 [==============================] - 278s 355ms/step - loss: 0.6548 -
binary_accuracy: 0.8868
Loss: 0.6547651886940002
Accuracy: 0.8867999911308289
```

Results from the saved model:

```
input: this is such an amazing movie! : score: 0.996450
input: The movie was great!          : score: 0.988944
input: The movie was meh.             : score: 0.468001
input: The movie was okish.          : score: 0.005260
input: The movie was terrible...     : score: 0.000772
```

```
Results from the model in memory:
input: this is such an amazing movie! : score: 0.996450
input: The movie was great!          : score: 0.988944
input: The movie was meh.             : score: 0.467993
input: The movie was okish.          : score: 0.005260
input: The movie was terrible...     : score: 0.000772
```

Graphs:



ALBERT with batch size 64 and 3e-5:

When increasing the batch size i am getting memory exhausting for batch size values 64,128 and 256. So, only batch sizes less than 32 ie 16,32 are working for albert.

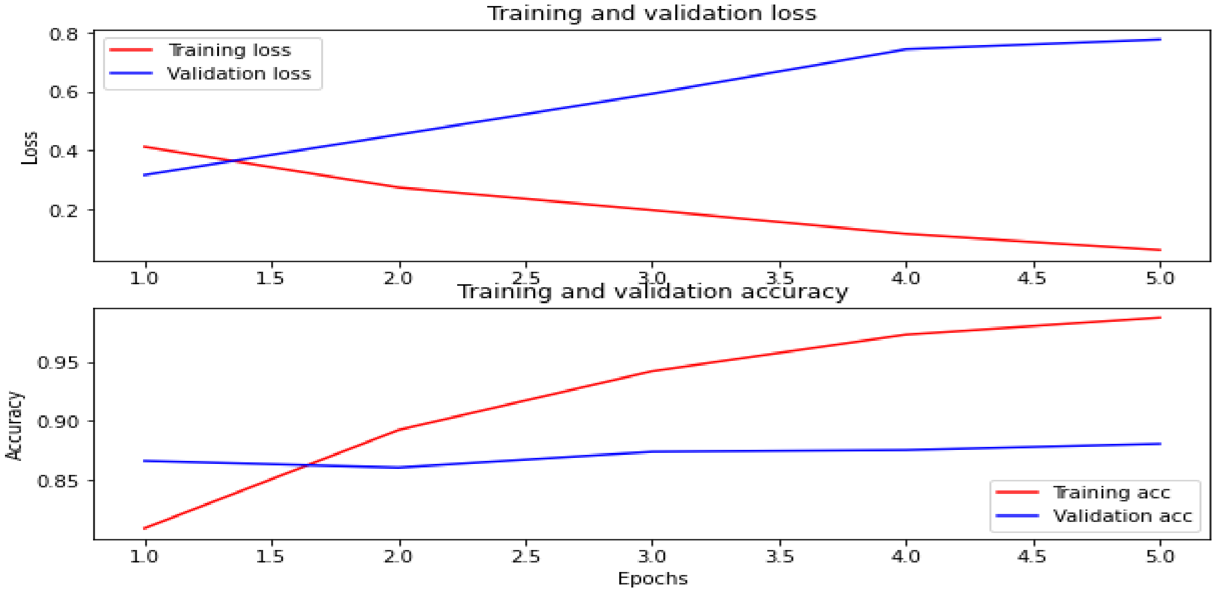
```
Training model with https://tfhub.dev/tensorflow/albert_en_base/2
Epoch 1/5
1250/1250 [==============================] - 470s 365ms/step - loss: 0.4124 -
binary_accuracy: 0.8090 - val_loss: 0.3167 - val_binary_accuracy: 0.8660
Epoch 2/5
1250/1250 [==============================] - 451s 360ms/step - loss: 0.2742 -
binary_accuracy: 0.8923 - val_loss: 0.4539 - val_binary_accuracy: 0.8604
Epoch 3/5
1250/1250 [==============================] - 450s 360ms/step - loss: 0.1968 -
binary_accuracy: 0.9422 - val_loss: 0.5929 - val_binary_accuracy: 0.8738
Epoch 4/5
1250/1250 [==============================] - 450s 360ms/step - loss: 0.1164 -
binary_accuracy: 0.9732 - val_loss: 0.7440 - val_binary_accuracy: 0.8752
Epoch 5/5
```

```
1250/1250 [==============================] - 450s 360ms/step - loss: 0.0611 -
binary_accuracy: 0.9876 - val_loss: 0.7776 - val_binary_accuracy: 0.8806
```

Test loss and Accuracy:

```
1563/1563 [==============================] - 185s 119ms/step - loss: 0.7651 -
binary_accuracy: 0.8827
Loss: 0.7650989890098572
Accuracy: 0.8826799988746643
```

Graphs:



Small Bert with batch size 64 and 3e-5:

```
Training model with https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-256_A-4/1
Epoch 1/5
313/313 [==============================] - 224s 690ms/step - loss: 0.6172 -
binary_accuracy: 0.6543 - val_loss: 0.4861 - val_binary_accuracy: 0.7846
Epoch 2/5
313/313 [==============================] - 208s 664ms/step - loss: 0.4291 -
binary_accuracy: 0.7931 - val_loss: 0.4066 - val_binary_accuracy: 0.8144
Epoch 3/5
313/313 [==============================] - 207s 663ms/step - loss: 0.3727 -
binary_accuracy: 0.8273 - val_loss: 0.3984 - val_binary_accuracy: 0.8246
Epoch 4/5
313/313 [==============================] - 207s 661ms/step - loss: 0.3415 -
binary_accuracy: 0.8450 - val_loss: 0.3938 - val_binary_accuracy: 0.8276
Epoch 5/5
```

```
313/313 [==============================] - 207s 660ms/step - loss: 0.3211 -
binary_accuracy: 0.8586 - val_loss: 0.3998 - val_binary_accuracy: 0.8238
```

Test Accuracy and loss:

```
391/391 [==============================] - 162s 415ms/step - loss: 0.3893 -
binary_accuracy: 0.8282
Loss: 0.3892654776573181
Accuracy: 0.8281599879264832
```

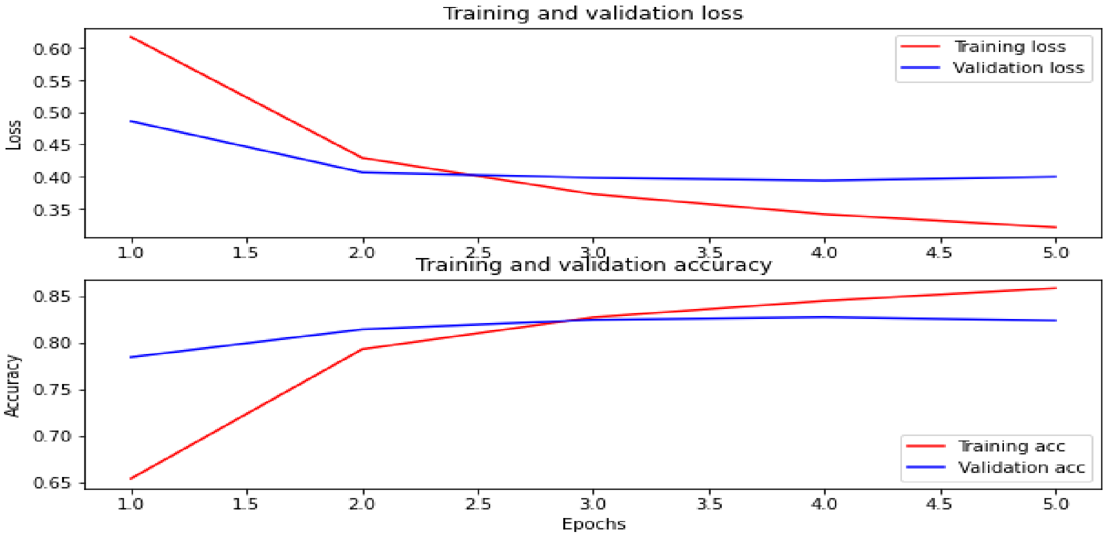
Results from the saved model:

```
input: this is such an amazing movie! : score: 0.967992
input: The movie was great!           : score: 0.890026
input: The movie was meh.             : score: 0.370098
input: The movie was okish.           : score: 0.268040
input: The movie was terrible...      : score: 0.022361
```

Results from the model in memory:

```
input: this is such an amazing movie! : score: 0.967992
input: The movie was great!           : score: 0.890026
input: The movie was meh.             : score: 0.370098
input: The movie was okish.           : score: 0.268040
input: The movie was terrible...      : score: 0.022361
```

Graphs:



Small Bert with batch size 128 and 3e-5:

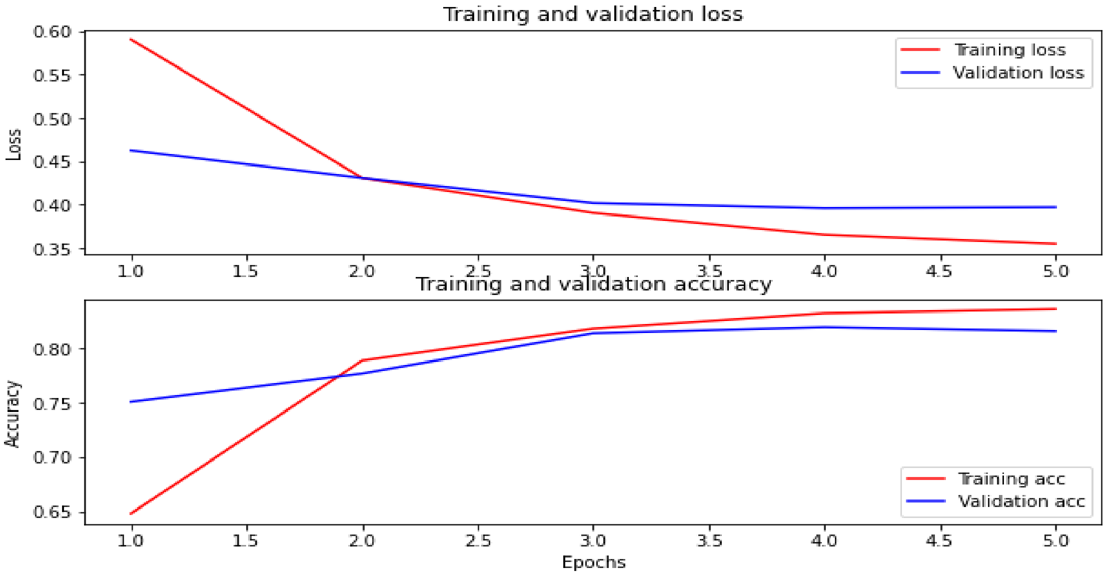
Training model with https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-256_A-4/1

```
Epoch 1/5
157/157 [==============================] - 1798s 11s/step - loss: 0.5901 -
binary_accuracy: 0.6482 - val_loss: 0.4623 - val_binary_accuracy: 0.7508
Epoch 2/5
157/157 [==============================] - 1760s 11s/step - loss: 0.4308 -
binary_accuracy: 0.7889 - val_loss: 0.4308 - val_binary_accuracy: 0.7770
Epoch 3/5
157/157 [==============================] - 1760s 11s/step - loss: 0.3905 -
binary_accuracy: 0.8183 - val_loss: 0.4018 - val_binary_accuracy: 0.8140
Epoch 4/5
157/157 [==============================] - 1794s 11s/step - loss: 0.3655 -
binary_accuracy: 0.8324 - val_loss: 0.3959 - val_binary_accuracy: 0.8196
Epoch 5/5
157/157 [==============================] - 1786s 11s/step - loss: 0.3550 -
binary_accuracy: 0.8361 - val_loss: 0.3969 - val_binary_accuracy: 0.8160
```

Test loss and accuracy:

```
196/196 [==============================] - 617s 3s/step - loss: 0.3884 -
binary_accuracy: 0.8227
Loss: 0.38835710287094116
Accuracy: 0.8226799964904785
```

Graph:



Results from the saved model:

```
input: this is such an amazing movie! : score: 0.984505
input: The movie was great!           : score: 0.769327
input: The movie was meh.              : score: 0.501706
input: The movie was okish.           : score: 0.342132
```

```
input: The movie was terrible...      : score: 0.107670

Results from the model in memory:
input: this is such an amazing movie! : score: 0.984505
input: The movie was great!           : score: 0.769327
input: The movie was meh.             : score: 0.501706
input: The movie was okish.           : score: 0.342132
input: The movie was terrible...      : score: 0.107670
```

Findings:

By changing the batch sizes 64,128 for small bert and electra and 32,64 for Albert when the learning rate is 3e-5, the small bert model proven to be the best model among so i tried changing the learning rate and increased the batch size and noted down the findings which resulted in 9 models of small bert with 64,128,256 batch sizes and learning rates as 3e-5,2e-4 and 1e-6. We observe that with small batch sizes we are getting good accuracy for every model.

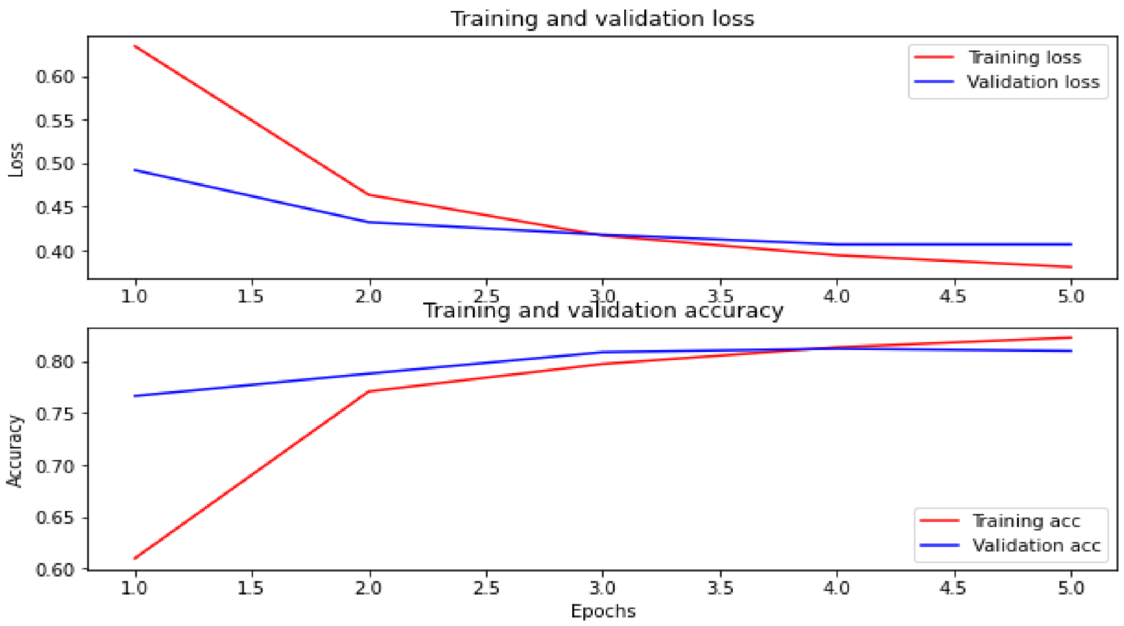
Small Bert with batch size 256 and learning rate 3e-5

```
Training model with https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-256_A-4/1
Epoch 1/5
79/79 [==============================] - 273s 3s/step - loss: 0.6342 - binary_accuracy:
0.6101 - val_loss: 0.4921 - val_binary_accuracy: 0.7664
Epoch 2/5
79/79 [==============================] - 257s 3s/step - loss: 0.4637 - binary_accuracy:
0.7709 - val_loss: 0.4324 - val_binary_accuracy: 0.7880
Epoch 3/5
79/79 [==============================] - 258s 3s/step - loss: 0.4170 - binary_accuracy:
0.7972 - val_loss: 0.4181 - val_binary_accuracy: 0.8086
Epoch 4/5
79/79 [==============================] - 259s 3s/step - loss: 0.3946 - binary_accuracy:
0.8134 - val_loss: 0.4069 - val_binary_accuracy: 0.8120
Epoch 5/5
79/79 [==============================] - 258s 3s/step - loss: 0.3811 - binary_accuracy:
0.8227 - val_loss: 0.4069 - val_binary_accuracy: 0.8098
```

Test Loss and Accuracy:

```
98/98 [==============================] - 240s 2s/step - loss: 0.3987 - binary_accuracy:
0.8150
Loss: 0.3986685276031494
Accuracy: 0.8149999976158142
```

Graphs:



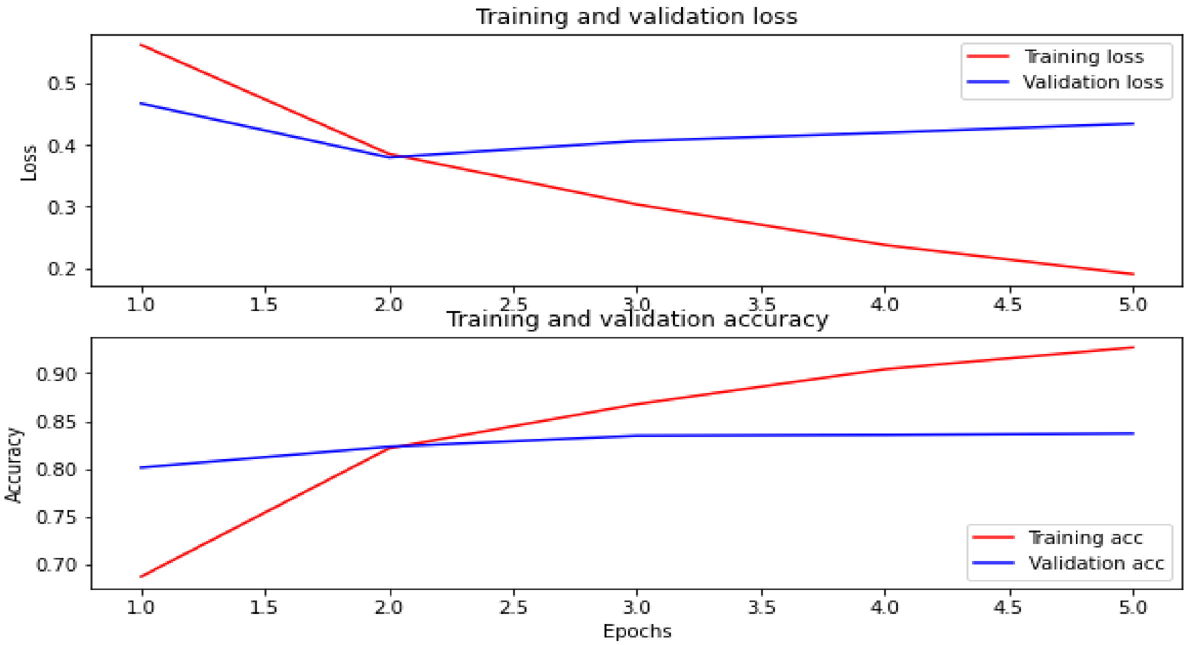
Small Bert with batch size 256 and learning rate 2e-4

```
Training model with https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-256_A-4/1
Epoch 1/5
79/79 [==============================] - 269s 3s/step - loss: 0.5614 - binary_accuracy:
0.6873 - val_loss: 0.4667 - val_binary_accuracy: 0.8014
Epoch 2/5
79/79 [==============================] - 258s 3s/step - loss: 0.3851 - binary_accuracy:
0.8214 - val_loss: 0.3794 - val_binary_accuracy: 0.8232
Epoch 3/5
79/79 [==============================] - 256s 3s/step - loss: 0.3034 - binary_accuracy:
0.8676 - val_loss: 0.4059 - val_binary_accuracy: 0.8346
Epoch 4/5
79/79 [==============================] - 257s 3s/step - loss: 0.2378 - binary_accuracy:
0.9043 - val_loss: 0.4194 - val_binary_accuracy: 0.8354
Epoch 5/5
79/79 [==============================] - 260s 3s/step - loss: 0.1908 - binary_accuracy:
0.9269 - val_loss: 0.4340 - val_binary_accuracy: 0.837
```

Test loss and Accuracy:

```
98/98 [==============================] - 241s 2s/step - loss: 0.4354 - binary_accuracy:
0.8309
Loss: 0.4353693723678589
Accuracy: 0.8308799862861633
```

Graphs:



Small Bert with batch size 256 and learning rate 4e-6

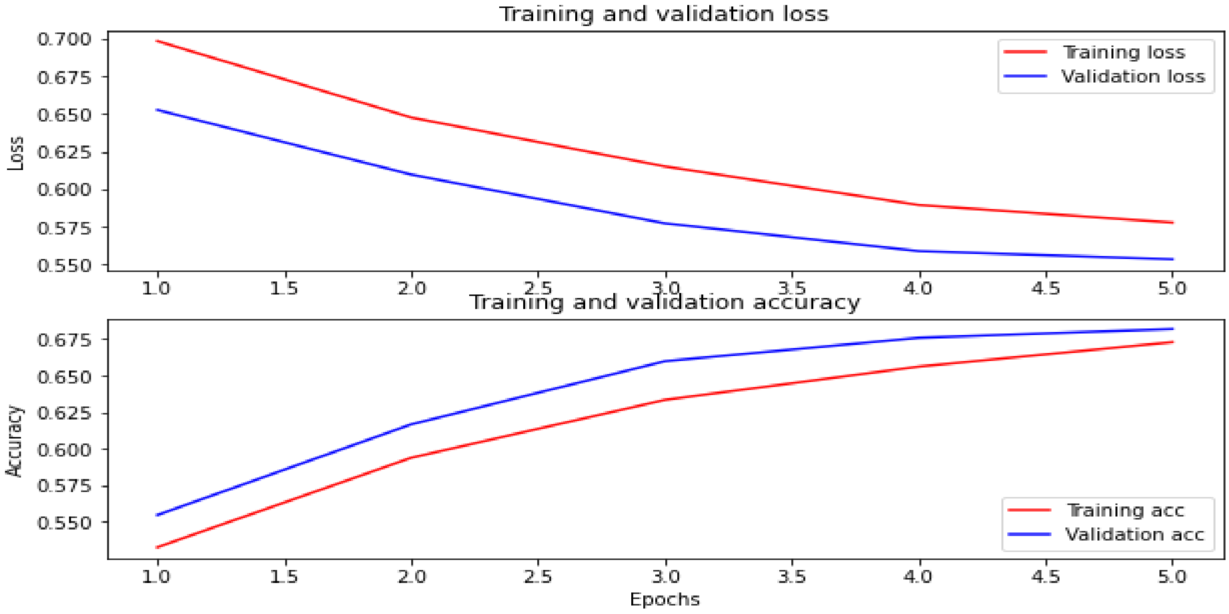
```
Training model with https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-256_A-4/1
Epoch 1/5
79/79 [==============================] - 281s 3s/step - loss: 0.6983 - binary_accuracy:
0.5325 - val_loss: 0.6526 - val_binary_accuracy: 0.5546
Epoch 2/5
79/79 [==============================] - 264s 3s/step - loss: 0.6475 - binary_accuracy:
0.5938 - val_loss: 0.6097 - val_binary_accuracy: 0.6166
Epoch 3/5
79/79 [==============================] - 264s 3s/step - loss: 0.6150 - binary_accuracy:
0.6334 - val_loss: 0.5773 - val_binary_accuracy: 0.6598
Epoch 4/5
79/79 [==============================] - 265s 3s/step - loss: 0.5895 - binary_accuracy:
0.6561 - val_loss: 0.5589 - val_binary_accuracy: 0.6758
Epoch 5/5
79/79 [==============================] - 261s 3s/step - loss: 0.5779 - binary_accuracy:
0.6729 - val_loss: 0.5535 - val_binary_accuracy: 0.6818
```

Test loss and Accuracy:

```
98/98 [==============================] - 242s 2s/step - loss: 0.5572 - binary_accuracy:
0.6820
Loss: 0.5571814179420471
```

Accuracy: 0.6819999814033508

Graphs:



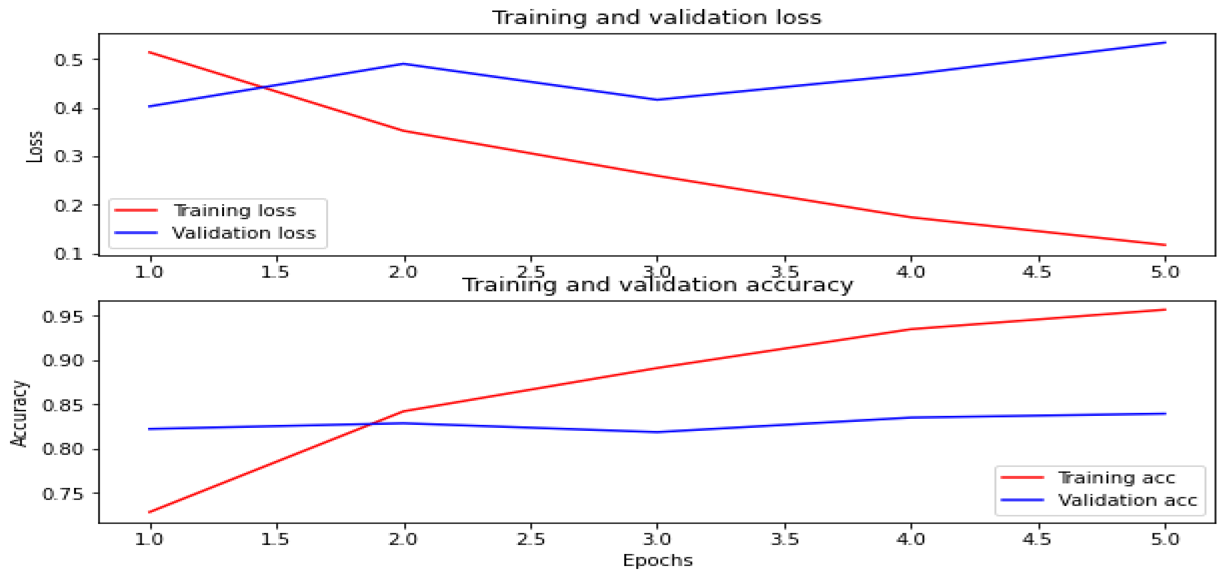
Small Bert with batch size 128 and learning rate 2e-4

```
Training model with https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-256_A-4/1
Epoch 1/5
157/157 [==============================] - 175s 1s/step - loss: 0.5132 -
binary_accuracy: 0.7283 - val_loss: 0.4022 - val_binary_accuracy: 0.8220
Epoch 2/5
157/157 [==============================] - 165s 1s/step - loss: 0.3518 -
binary_accuracy: 0.8418 - val_loss: 0.4897 - val_binary_accuracy: 0.8284
Epoch 3/5
157/157 [==============================] - 163s 1s/step - loss: 0.2593 -
binary_accuracy: 0.8907 - val_loss: 0.4156 - val_binary_accuracy: 0.8184
Epoch 4/5
157/157 [==============================] - 158s 1s/step - loss: 0.1737 -
binary_accuracy: 0.9345 - val_loss: 0.4678 - val_binary_accuracy: 0.8348
Epoch 5/5
157/157 [==============================] - 160s 1s/step - loss: 0.1168 -
binary_accuracy: 0.9565 - val_loss: 0.5333 - val_binary_accuracy: 0.8392
```

Test loss and Accuracy:

```
196/196 [==============================] - 138s 702ms/step - loss: 0.5697 -
binary_accuracy: 0.8257
Loss: 0.5696938037872314
Accuracy: 0.8256800174713135
```

Graphs:



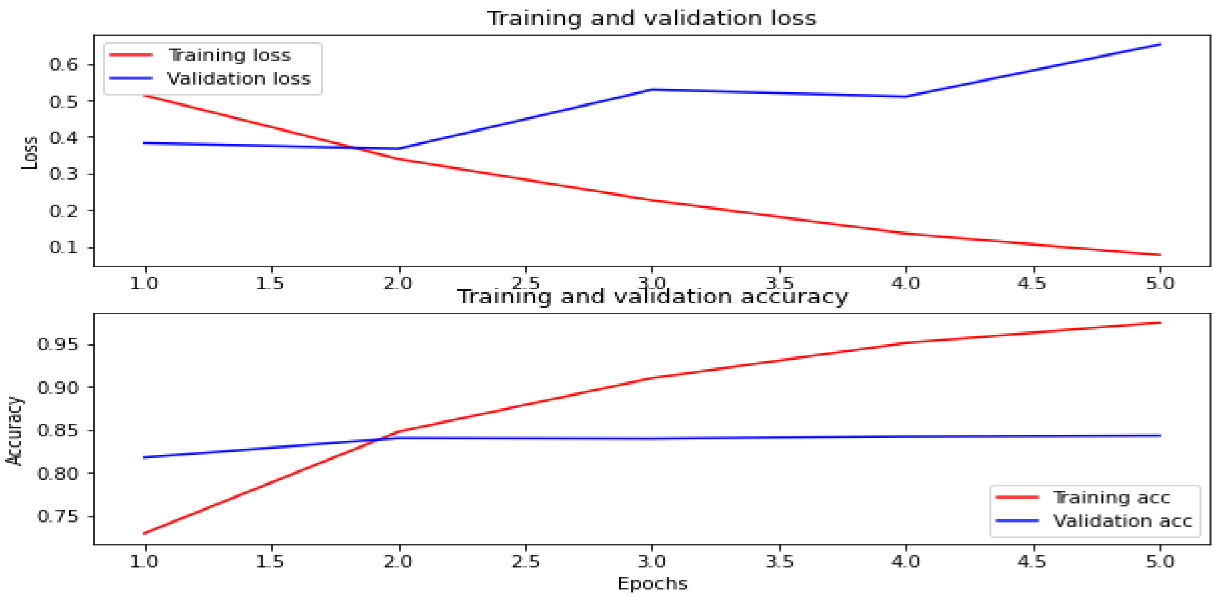
Small Bert with batch size 64 and learning rate 2e-4

```
Training model with https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-256_A-4/1
Epoch 1/5
313/313 [==============================] - 132s 401ms/step - loss: 0.5133 -
binary_accuracy: 0.7293 - val_loss: 0.3828 - val_binary_accuracy: 0.8176
Epoch 2/5
313/313 [==============================] - 117s 374ms/step - loss: 0.3395 -
binary_accuracy: 0.8475 - val_loss: 0.3673 - val_binary_accuracy: 0.8402
Epoch 3/5
313/313 [==============================] - 118s 376ms/step - loss: 0.2263 -
binary_accuracy: 0.9101 - val_loss: 0.5286 - val_binary_accuracy: 0.8396
Epoch 4/5
313/313 [==============================] - 120s 383ms/step - loss: 0.1354 -
binary_accuracy: 0.9510 - val_loss: 0.5099 - val_binary_accuracy: 0.8422
Epoch 5/5
313/313 [==============================] - 122s 389ms/step - loss: 0.0764 -
binary_accuracy: 0.9748 - val_loss: 0.6524 - val_binary_accuracy: 0.8432
```

Test loss and Accuracy:

```
391/391 [==============================] - 108s 275ms/step - loss: 0.7554 -
binary_accuracy: 0.8177
Loss: 0.7553879618644714
Accuracy: 0.8177199959754944
```

Graphs:



Small Bert with batch size 128 and learning rate 4e-6

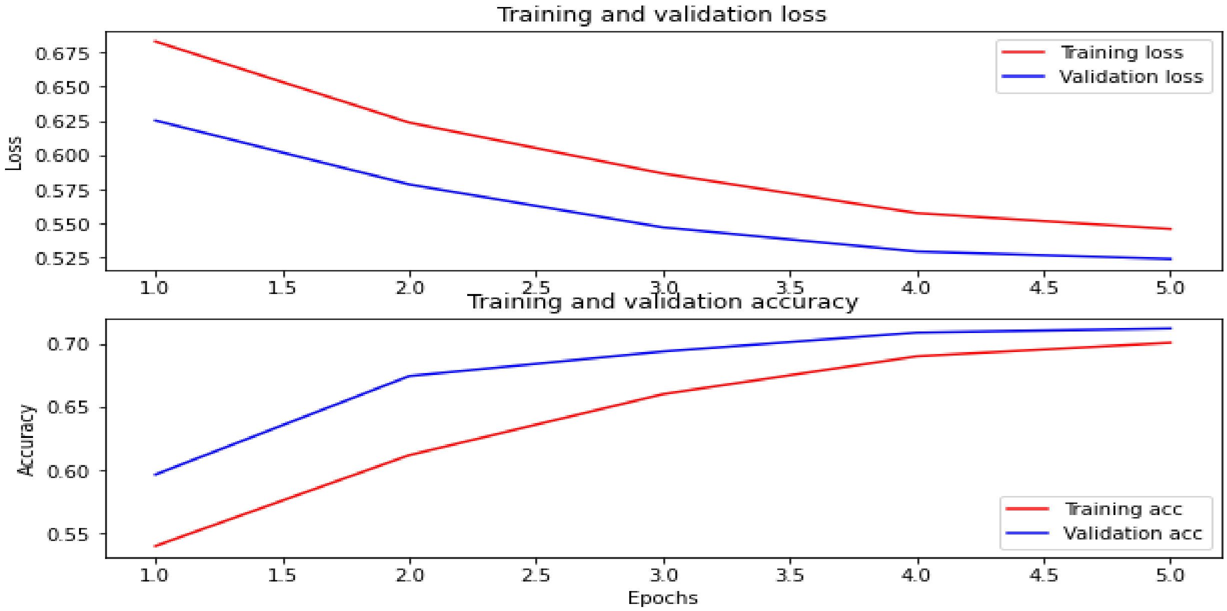
```
Training model with https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-256_A-4/1
Epoch 1/5
157/157 [==============================] - 177s 1s/step - loss: 0.6831 -
binary_accuracy: 0.5400 - val_loss: 0.6252 - val_binary_accuracy: 0.5964
Epoch 2/5
157/157 [==============================] - 165s 1s/step - loss: 0.6237 -
binary_accuracy: 0.6117 - val_loss: 0.5785 - val_binary_accuracy: 0.6742
Epoch 3/5
157/157 [==============================] - 164s 1s/step - loss: 0.5864 -
binary_accuracy: 0.6599 - val_loss: 0.5471 - val_binary_accuracy: 0.6936
Epoch 4/5
157/157 [==============================] - 163s 1s/step - loss: 0.5575 -
binary_accuracy: 0.6898 - val_loss: 0.5294 - val_binary_accuracy: 0.7086
Epoch 5/5
```

```
157/157 [==============================] - 164s 1s/step - loss: 0.5459 -
binary_accuracy: 0.7006 - val_loss: 0.5240 - val_binary_accuracy: 0.7118
```

Test loss and Accuracy:

```
196/196 [==============================] - 141s 716ms/step - loss: 0.5266 -
binary_accuracy: 0.7180
Loss: 0.5265780091285706
Accuracy: 0.7180399894714355
```

Graphs:



Small Bert with batch size 64 and learning rate 4e-6

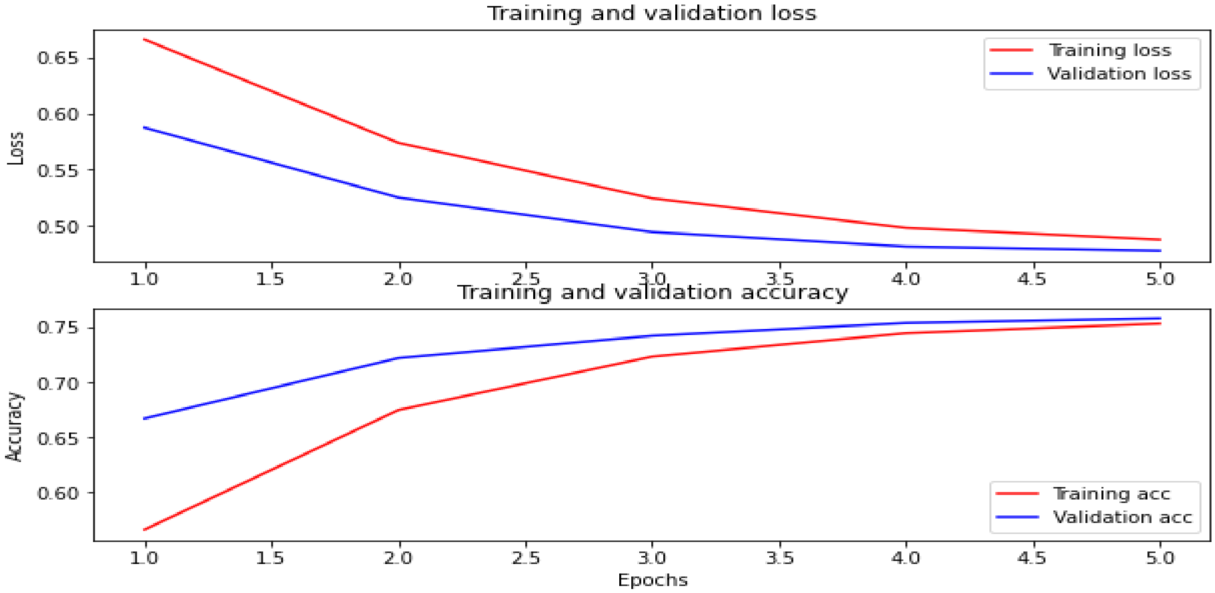
```
Training model with https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-256_A-4/1
Epoch 1/5
313/313 [==============================] - 134s 402ms/step - loss: 0.6661 -
binary_accuracy: 0.5662 - val_loss: 0.5875 - val_binary_accuracy: 0.6670
Epoch 2/5
313/313 [==============================] - 119s 380ms/step - loss: 0.5739 -
binary_accuracy: 0.6747 - val_loss: 0.5251 - val_binary_accuracy: 0.7218
Epoch 3/5
313/313 [==============================] - 119s 381ms/step - loss: 0.5245 -
binary_accuracy: 0.7231 - val_loss: 0.4944 - val_binary_accuracy: 0.7420
Epoch 4/5
313/313 [==============================] - 119s 381ms/step - loss: 0.4982 -
binary_accuracy: 0.7444 - val_loss: 0.4814 - val_binary_accuracy: 0.7536
```

```
Epoch 5/5
313/313 [==============================] - 123s 393ms/step - loss: 0.4876 -
binary_accuracy: 0.7530 - val_loss: 0.4777 - val_binary_accuracy: 0.7578
```

Test loss and Accuracy:

```
391/391 [==============================] - 108s 276ms/step - loss: 0.4743 -
binary_accuracy: 0.7599
Loss: 0.4743233621120453
Accuracy: 0.7599200010299683
```

Graphs:



RNN:

I have take RNN and LSTM models with batch size 256,512,1024 and learning rate as 1e-2,1e-3,1e-4 each which will create 18 models and the findings are as follows:

LSTM

LSTM with adam batch size 1024 and learning rate 1e-4:

```
Epoch: 1 -Train Accuracy:0.640350 - Eval Accuracy: 0.620400
Epoch: 2 -Train Accuracy:0.719900 - Eval Accuracy: 0.673000
Epoch: 4 -Train Accuracy:0.791850 - Eval Accuracy: 0.739800
Epoch: 6 -Train Accuracy:0.862950 - Eval Accuracy: 0.834800
Epoch: 8 -Train Accuracy:0.929650 - Eval Accuracy: 0.873400
```

```
Epoch: 10 -Train Accuracy:0.955600 - Eval Accuracy: 0.884200

Total time taken (in seconds): 433.16
```

LSTM with adam batch size 512 and learning rate 1e-3:

```
Epoch: 1 -Train Accuracy:0.910250 - Eval Accuracy: 0.877000
Epoch: 2 -Train Accuracy:0.958800 - Eval Accuracy: 0.873200
Epoch: 4 -Train Accuracy:0.984550 - Eval Accuracy: 0.864800
Epoch: 6 -Train Accuracy:0.998350 - Eval Accuracy: 0.869400
Epoch: 8 -Train Accuracy:0.996550 - Eval Accuracy: 0.861200
Epoch: 10 -Train Accuracy:0.998600 - Eval Accuracy: 0.875200

Total time taken (in seconds): 792.24
```

LSTM with adam batch size 256 and learning rate 1e-2:

```
Epoch: 1 -Train Accuracy:0.847400 - Eval Accuracy: 0.761200
Epoch: 2 -Train Accuracy:0.947250 - Eval Accuracy: 0.838800
Epoch: 4 -Train Accuracy:0.994400 - Eval Accuracy: 0.849400
Epoch: 6 -Train Accuracy:0.979450 - Eval Accuracy: 0.814200
Epoch: 8 -Train Accuracy:0.999400 - Eval Accuracy: 0.843000
Epoch: 10 -Train Accuracy:0.999550 - Eval Accuracy: 0.843200

Total time taken (in seconds): 1538.61
```

LSTM with RMS Prop batch size 1024 and learning rate 1e-4:

```
Epoch: 1 -Train Accuracy:0.520100 - Eval Accuracy: 0.527600
Epoch: 2 -Train Accuracy:0.561750 - Eval Accuracy: 0.568600
Epoch: 4 -Train Accuracy:0.585000 - Eval Accuracy: 0.584600
Epoch: 6 -Train Accuracy:0.709500 - Eval Accuracy: 0.690800
Epoch: 8 -Train Accuracy:0.790600 - Eval Accuracy: 0.791000
Epoch: 10 -Train Accuracy:0.843300 - Eval Accuracy: 0.818400
```

Total time taken (in seconds): 2087.20

LSTM with RMS Prop batch size 512 and learning rate 1e-3:

```
Epoch: 1 -Train Accuracy:0.550150 - Eval Accuracy: 0.554000
Epoch: 2 -Train Accuracy:0.638800 - Eval Accuracy: 0.631600
Epoch: 4 -Train Accuracy:0.904600 - Eval Accuracy: 0.850600
Epoch: 6 -Train Accuracy:0.945250 - Eval Accuracy: 0.862000
Epoch: 8 -Train Accuracy:0.951000 - Eval Accuracy: 0.861200
Epoch: 10 -Train Accuracy:0.965400 - Eval Accuracy: 0.854800
```

Total time taken (in seconds): 2384.80

LSTM with RMS Prop batch size 256 and learning rate 1e-2:

```
Epoch: 1 -Train Accuracy:0.675300 - Eval Accuracy: 0.658200
Epoch: 2 -Train Accuracy:0.781800 - Eval Accuracy: 0.748400
Epoch: 4 -Train Accuracy:0.909450 - Eval Accuracy: 0.858000
Epoch: 6 -Train Accuracy:0.987000 - Eval Accuracy: 0.881600
Epoch: 8 -Train Accuracy:0.996950 - Eval Accuracy: 0.877200
Epoch: 10 -Train Accuracy:0.998850 - Eval Accuracy: 0.880600
```

Total time taken (in seconds): 3084.40

RNN

RNN with adam batch size 1024 and learning rate 1e-4:

```
Epoch: 1 -Train Accuracy:0.571350 - Eval Accuracy: 0.557200
Epoch: 2 -Train Accuracy:0.679000 - Eval Accuracy: 0.591400
Epoch: 4 -Train Accuracy:0.770050 - Eval Accuracy: 0.596200
Epoch: 6 -Train Accuracy:0.869300 - Eval Accuracy: 0.561000
Epoch: 8 -Train Accuracy:0.928550 - Eval Accuracy: 0.546800
Epoch: 10 -Train Accuracy:0.964550 - Eval Accuracy: 0.532200
```

Total time taken (in seconds): 232.68

RNN with adam batch size 512 and learning rate 1e-3:

```
Epoch: 1 -Train Accuracy:0.705500 - Eval Accuracy: 0.690600
```

```
Epoch: 2 -Train Accuracy:0.911150 - Eval Accuracy: 0.839600
Epoch: 4 -Train Accuracy:0.962500 - Eval Accuracy: 0.841800
Epoch: 6 -Train Accuracy:0.992850 - Eval Accuracy: 0.856600
Epoch: 8 -Train Accuracy:0.976650 - Eval Accuracy: 0.787200
Epoch: 10 -Train Accuracy:0.996400 - Eval Accuracy: 0.837400

Total time taken (in seconds): 418.13
```

RNN with adam batch size 256 and learning rate 1e-2:

```
Epoch: 1 -Train Accuracy:0.717650 - Eval Accuracy: 0.645600
Epoch: 2 -Train Accuracy:0.869350 - Eval Accuracy: 0.712200
Epoch: 4 -Train Accuracy:0.909000 - Eval Accuracy: 0.703600
Epoch: 6 -Train Accuracy:0.723250 - Eval Accuracy: 0.583400
Epoch: 8 -Train Accuracy:0.708500 - Eval Accuracy: 0.593800
Epoch: 10 -Train Accuracy:0.872600 - Eval Accuracy: 0.675200

Total time taken (in seconds): 777.82
```

RNN with RMS Prop batch size 1024 and learning rate 1e-4:

```
Epoch: 1 -Train Accuracy:0.536300 - Eval Accuracy: 0.524000
Epoch: 2 -Train Accuracy:0.553800 - Eval Accuracy: 0.544200
Epoch: 4 -Train Accuracy:0.612850 - Eval Accuracy: 0.609400
Epoch: 6 -Train Accuracy:0.701900 - Eval Accuracy: 0.689200
Epoch: 8 -Train Accuracy:0.783000 - Eval Accuracy: 0.771600
Epoch: 10 -Train Accuracy:0.854750 - Eval Accuracy: 0.825800

Total time taken (in seconds): 2088.67
```

RNN with RMS Prop batch size 512 and learning rate 1e-3:

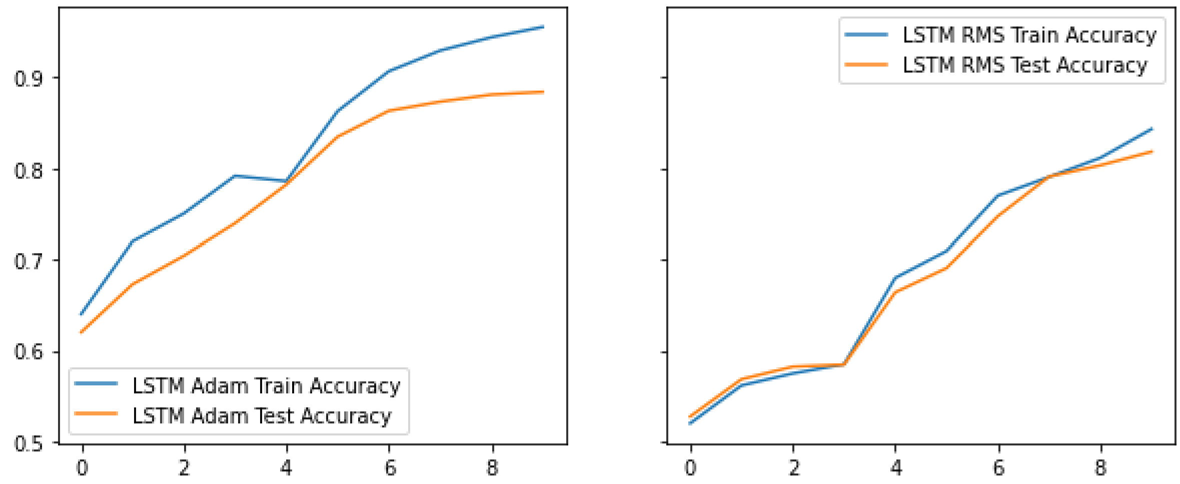
```
Epoch: 1 -Train Accuracy:0.560300 - Eval Accuracy: 0.562600
Epoch: 2 -Train Accuracy:0.665800 - Eval Accuracy: 0.650400
Epoch: 4 -Train Accuracy:0.899600 - Eval Accuracy: 0.839200
Epoch: 6 -Train Accuracy:0.902850 - Eval Accuracy: 0.841600
Epoch: 8 -Train Accuracy:0.944050 - Eval Accuracy: 0.851200
Epoch: 10 -Train Accuracy:0.988300 - Eval Accuracy: 0.867800

Total time taken (in seconds): 2458.82
```

RNN with RMS Prop batch size 256 and learning rate 1e-2:

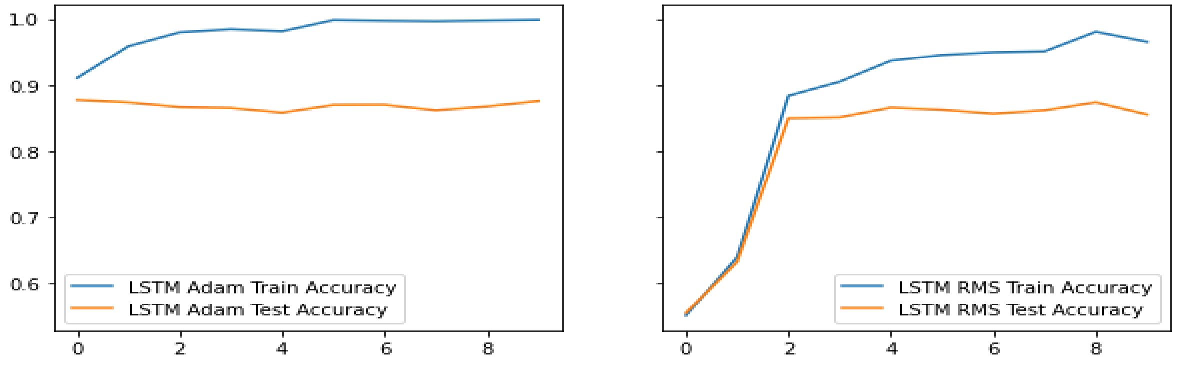
```
Epoch: 1 -Train Accuracy:0.631000 - Eval Accuracy: 0.622400
Epoch: 2 -Train Accuracy:0.771200 - Eval Accuracy: 0.723800
Epoch: 4 -Train Accuracy:0.924600 - Eval Accuracy: 0.875600
Epoch: 6 -Train Accuracy:0.985150 - Eval Accuracy: 0.885800
Epoch: 8 -Train Accuracy:0.997650 - Eval Accuracy: 0.882000
Epoch: 10 -Train Accuracy:0.999200 - Eval Accuracy: 0.884000

Total time taken (in seconds): 3354.97
```

Graphs:

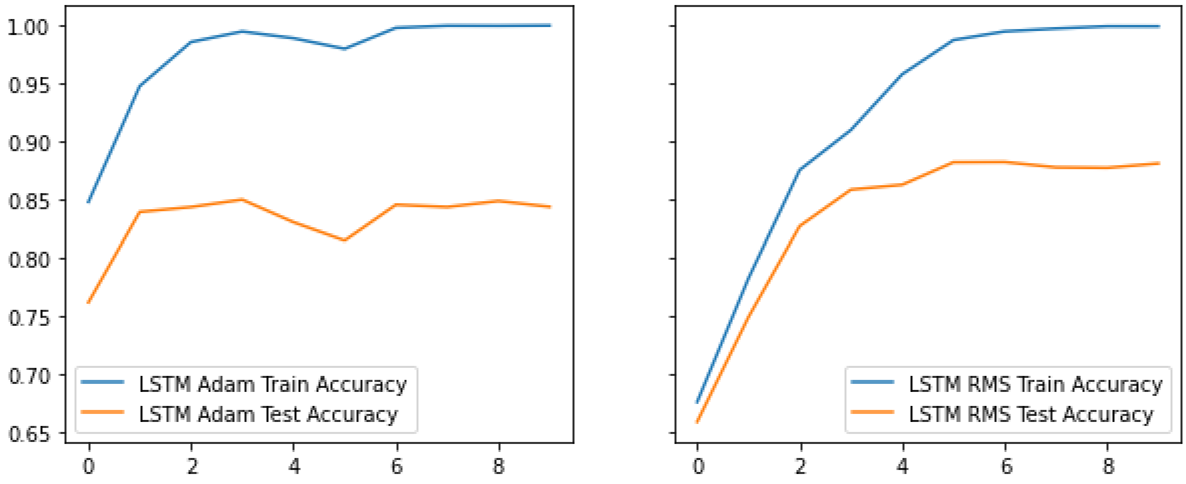
LSTM ADAM vs LSTM RMS Prop with batch size=1024 and learning rate =1e-4



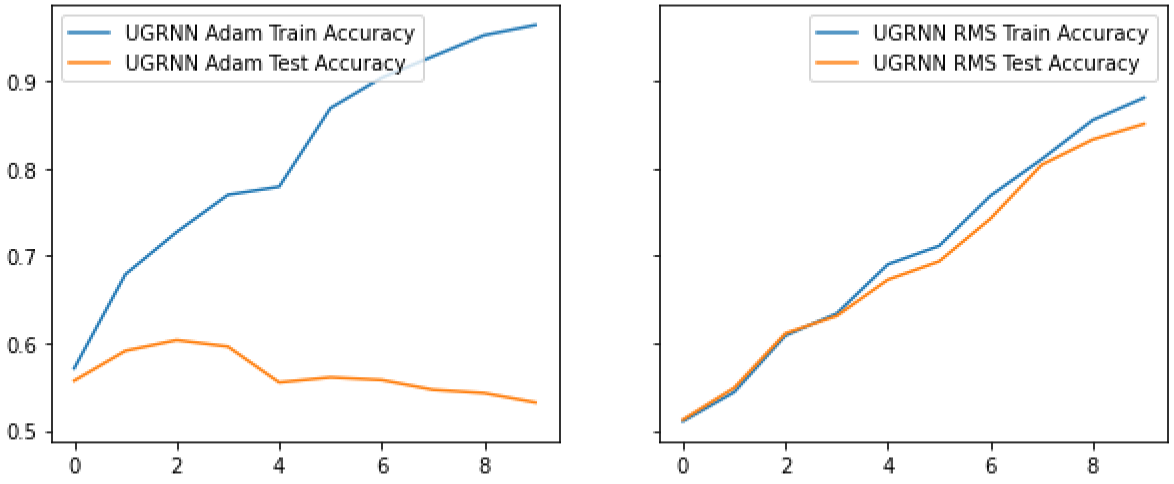
LSTM ADAM vs LSTM RMS Prop with batch size=512 and learning rate =1e-3



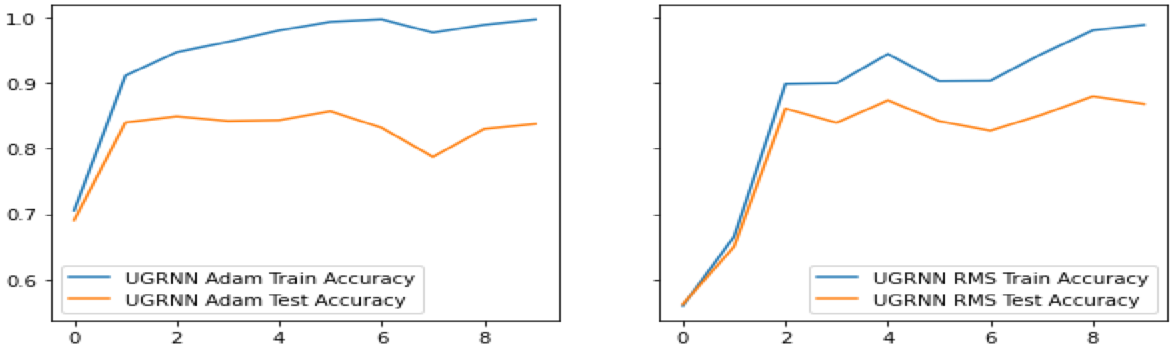
LSTM ADAM vs LSTM RMS Prop with batch size=256 and learning rate =1e-2



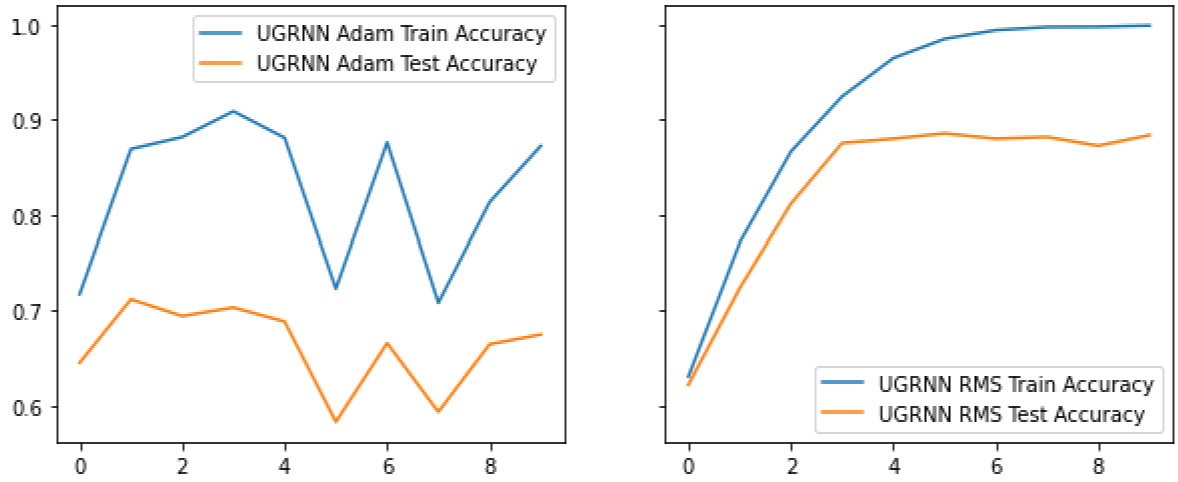
UGRNN ADAM vs UGRNN RMS Prop with batch size=1024 and learning rate =1e-4



UGRNN ADAM vs UGRNN RMS Prop with batch size=512 and learning rate =1e-3



UGRNN ADAM vs UGRNN RMS Prop with batch size=256 and learning rate =1e-2



Test Accuracy for all models:

For Adam with different learning rates and batch sizes(6 models):

Test accuracy for batch size 1024 is 0.864080011844635 in time 87.61

Test accuracy for batch size 1024 is 0.5242000222206116 in time 47.90

Test accuracy for batch size 512 is 0.8397200107574463 in time 87.36

Test accuracy for batch size 512 is 0.8191999793052673 in time 48.58

Test accuracy for batch size 256 is 0.8266800045967102 in time 117.12

Test accuracy for batch size 256 is 0.671239972114563 in time 64.79

For RMSProp with different learning rates and batch sizes(6 models):

```
Test accuracy for batch size 1024 is 0.8012400269508362 in time 73.80


Test accuracy for batch size 1024 is 0.8174800276756287 in time 92.39


Test accuracy for batch size 512 is 0.8416799902915955 in time 76.43


Test accuracy for batch size 512 is 0.8674799799919128 in time 74.25


Test accuracy for batch size 256 is 0.8674799799919128 in time 97.81


Test accuracy for batch size 256 is 0.8732399940490723 in time 96.84
```

Test network on new samples

```
Adam LSTM (1024,1e-4) The sentiment for the review with score 10 was found to be positive
Adam LSTM (1024,1e-4) The sentiment for the review with score 4 was found to be negative
Adam LSTM (1024,1e-4) The sentiment for the review with score 1 was found to be negative


Adam LSTM (512,1e-3) The sentiment for the review with score 10 was found to be positive
Adam LSTM (512,1e-3) The sentiment for the review with score 4 was found to be negative
Adam LSTM (512,1e-3) The sentiment for the review with score 1 was found to be negative


RMS LSTM (1024,1e-4) The sentiment for the review with score 10 was found to be positive
RMS LSTM (1024,1e-4) The sentiment for the review with score 4 was found to be negative
RMS LSTM (1024,1e-4) The sentiment for the review with score 1 was found to be negative


RMS LSTM (512,1e-3) The sentiment for the review with score 10 was found to be positive
RMS LSTM (512,1e-3) The sentiment for the review with score 4 was found to be negative
RMS LSTM (512,1e-3) The sentiment for the review with score 1 was found to be negative


RMS LSTM (256,1e-2) The sentiment for the review with score 10 was found to be positive
RMS LSTM (256,1e-2) The sentiment for the review with score 4 was found to be negative
```

RMS LSTM (256,1e-2) The sentiment for the review with score 1 was found to be negative

Adam UGRNN (1024,1e-4) The sentiment for the review with score 10 was found to be positive
Adam UGRNN (1024,1e-4) The sentiment for the review with score 4 was found to be positive
Adam UGRNN (1024,1e-4) The sentiment for the review with score 1 was found to be positive

ADAM UGRNN (512,1e-3) The sentiment for the review with score 10 was found to be positive
ADAM UGRNN (512,1e-3) The sentiment for the review with score 4 was found to be negative
ADAM UGRNN (512,1e-3) The sentiment for the review with score 1 was found to be negative

Adam UGRNN (256,1e-2) The sentiment for the review with score 10 was found to be negative
Adam UGRNN (256,1e-2) The sentiment for the review with score 4 was found to be positive
Adam UGRNN (256,1e-2) The sentiment for the review with score 1 was found to be positive

RMS UGRNN (1024,1e-4) The sentiment for the review with score 10 was found to be positive
RMS UGRNN (1024,1e-4) The sentiment for the review with score 4 was found to be negative
RMS UGRNN (1024,1e-4) The sentiment for the review with score 1 was found to be negative

RMS UGRNN (512,1e-3) The sentiment for the review with score 10 was found to be positive
RMS UGRNN (512,1e-3) The sentiment for the review with score 4 was found to be negative
RMS UGRNN (512,1e-3) The sentiment for the review with score 1 was found to be negative

RMS UGRNN (256,1e-2) The sentiment for the review with score 10 was found to be positive
RMS UGRNN (256,1e-2) The sentiment for the review with score 4 was found to be negative
RMS UGRNN (256,1e-2) The sentiment for the review with score 1 was found to be negative

Findings:

If we consider LSTM with different hyper parameters we can observe that batch size with 1024 and learning rate 1e-4 are the best parameters when compared to other values. We can observe that RMSProp with LSTM having batch size 1024 and learning rate 1e-4 is the best optimizer model among all LSTM models. Similarly RNN with RMS Prop having the batch size 1024 and learning rate 1e-4 is the best model among all other RNN models. We can also observe that with the decrease in batch size the training is taking more time for each model.