UNIVERSITY OF TEHRAN

COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NEURAL NETWORK & DEEP LEARNING

MINI PROJECT#2

SIAVASH SHAMS

810197644

MOHAMMAD HEYDARI
810197494

UNDER SUPERVISION OF:

DR. AHMAD KALHOR

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

UNIVERSITY OF TEHRAN

May. 2022

1 CONTENTS

2 QUESTION #1: STOCK MARKET PREDICTION

2.1

Below we can see Google and Apple, day to day stock closing prices from 2010 to 2018.

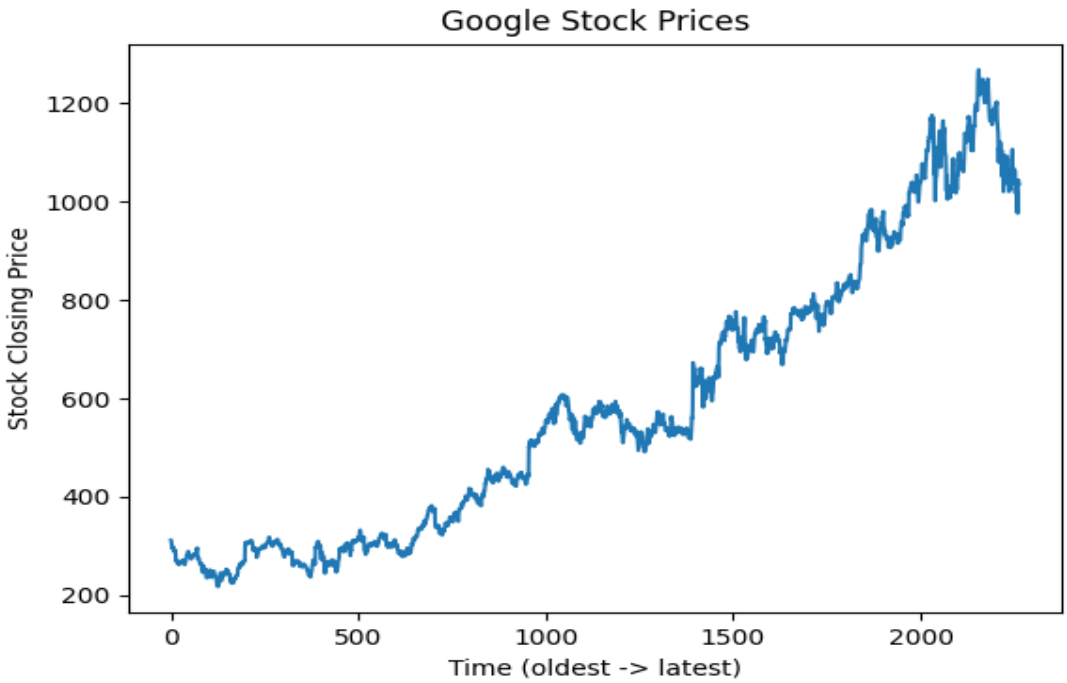

Figure1. Representation of Google Stoch Prices

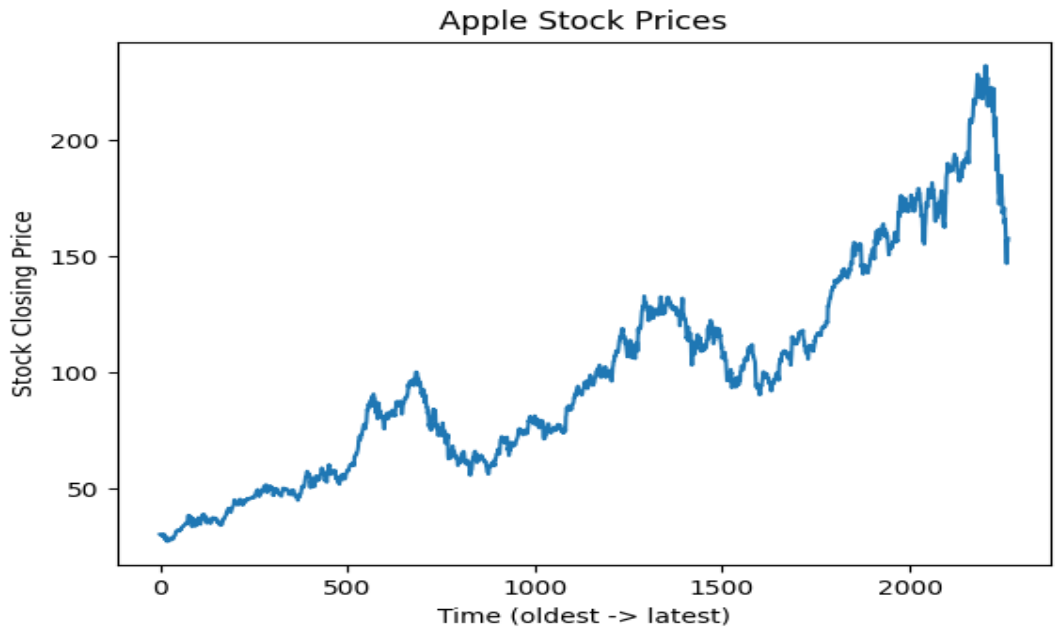

Figure2. Representation of Apple Stoch Prices

- **LSTM**

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 30, 64)            19712
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                33024
_________________________________________________________________
dense (Dense)                (None, 2)                 130
=================================================================
Total params: 52,866
Trainable params: 52,866
Non-trainable params: 0
```

Figure3. Model Architecture

We will train our model using Adam optimizer with 100 epochs.

```
Total training time: 106.46464157104492 seconds
```

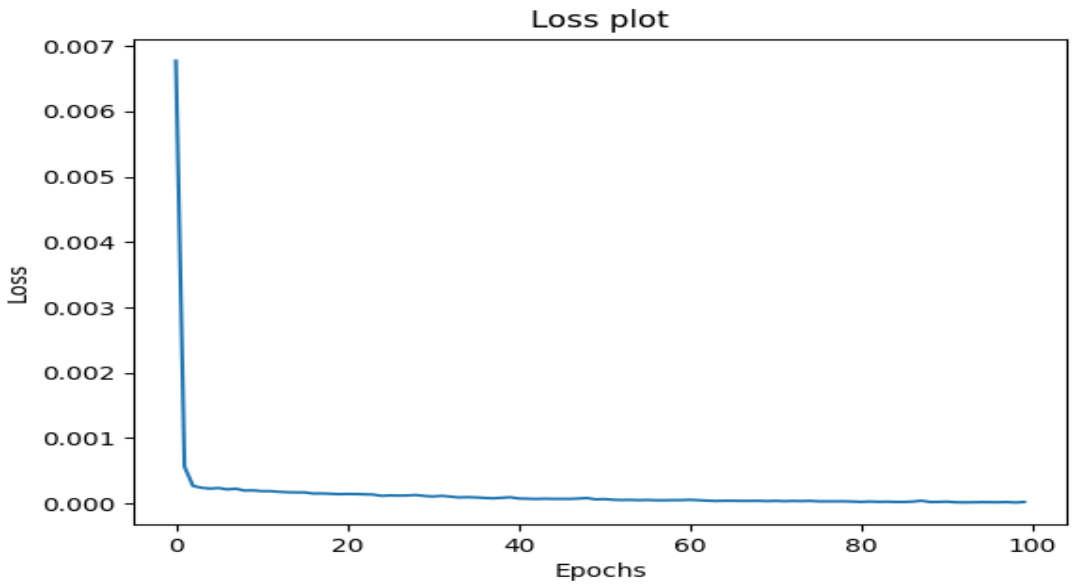

Figure4. Loss during the Epochs

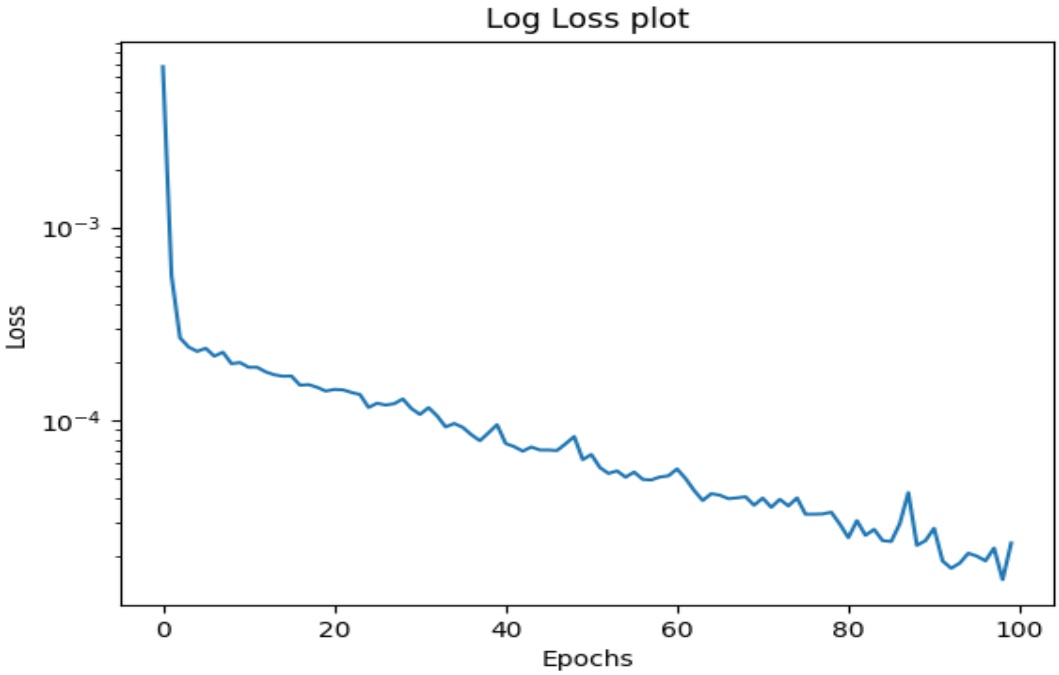

Figure5. Log-Loss representation during the Epochs

Figure6. Real-value VS Predicted-value (AAPL)

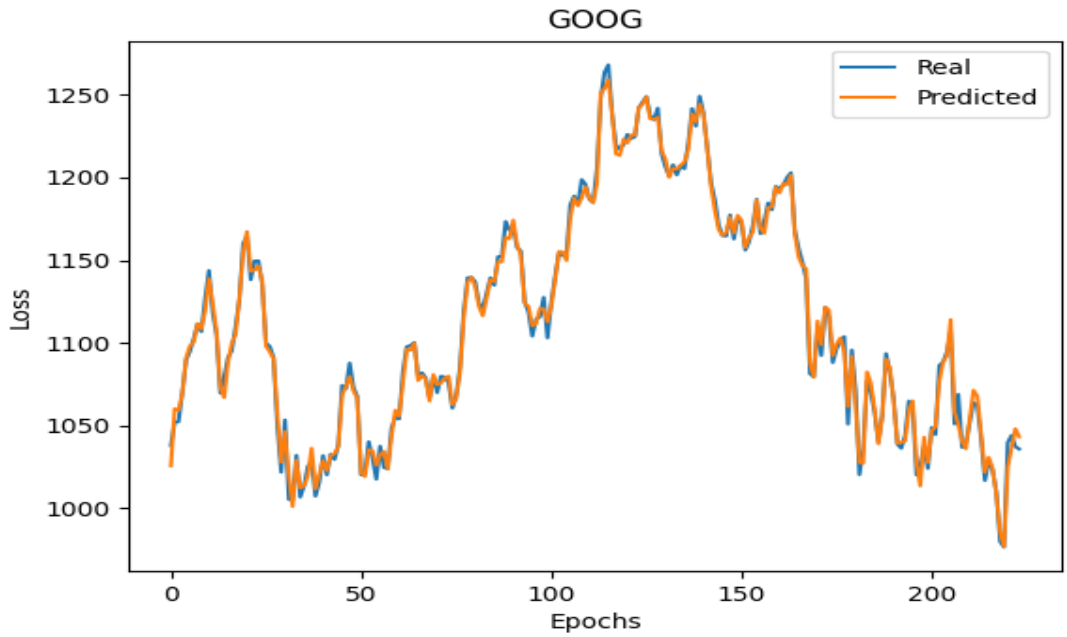

Figure7. Real-value VS Predicted-value (GOOG)

- **RNN**

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
simple_rnn (SimpleRNN)       (None, 30, 64)            4928
_________________________________________________________________
simple_rnn_1 (SimpleRNN)     (None, 64)                8256
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 130
=================================================================
Total params: 13,314
Trainable params: 13,314
Non-trainable params: 0
```

Figure8. Model Architecture

```
Total training time: 26.431206941604614 seconds
```

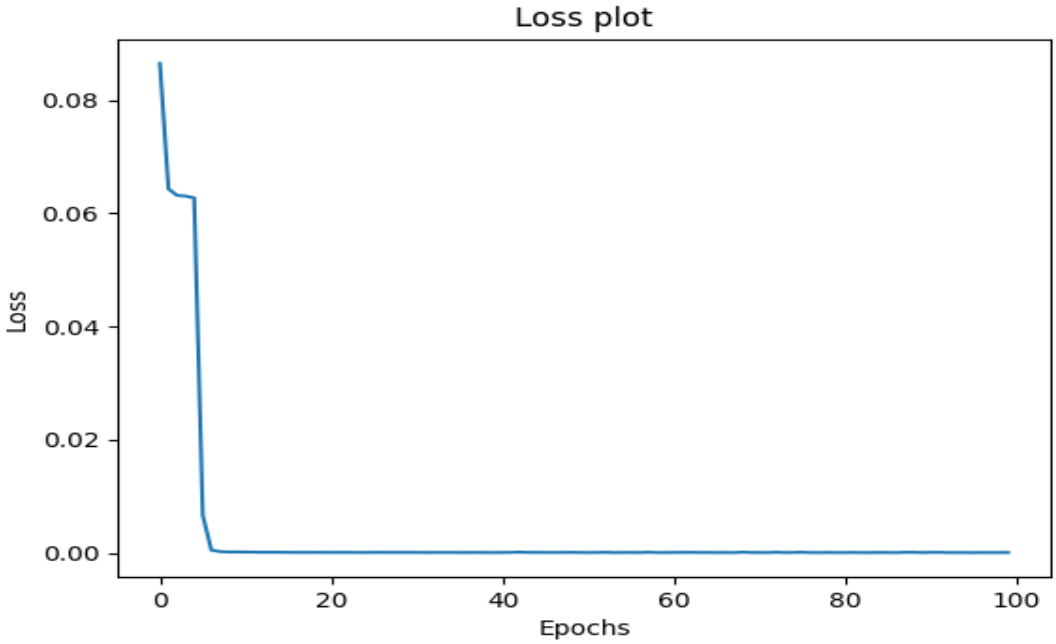

Figure9. Loss representation during the Epochs

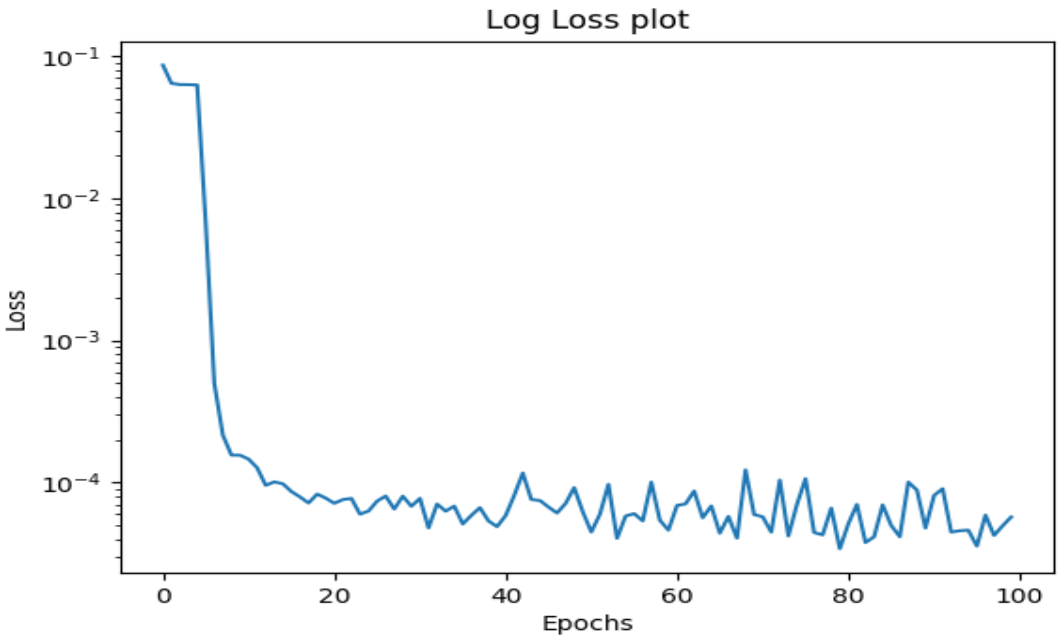

Figure10. Loss-Loss representation during the Epochs

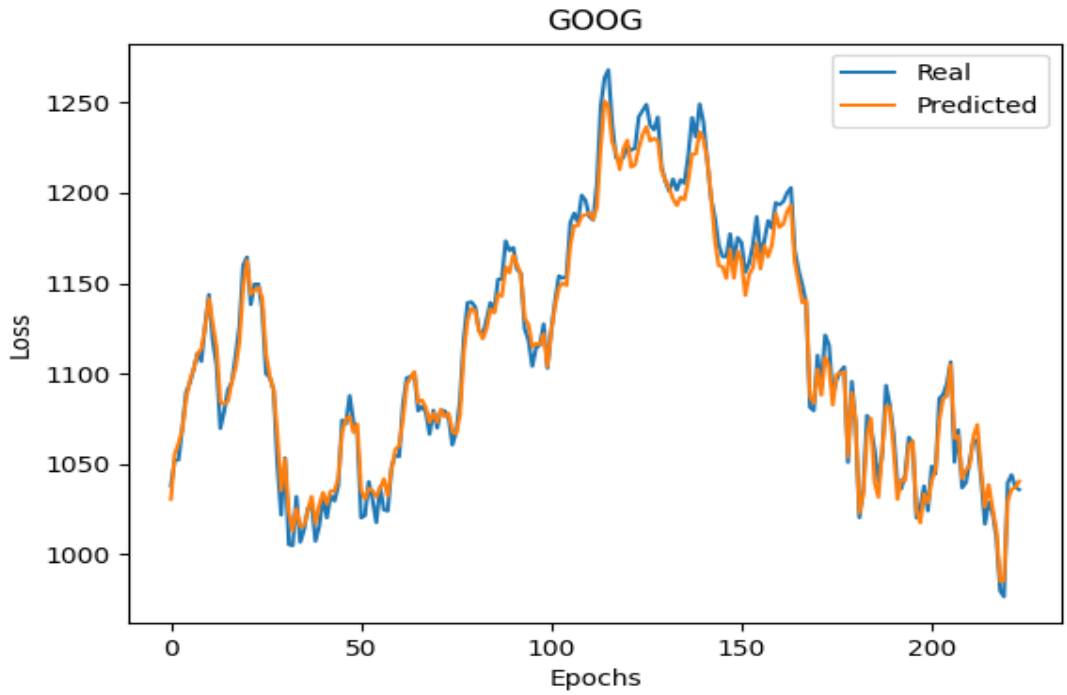

Figure11. Real-value VS Predicted-value (GOOG)

Figure12. Real-value VS Predicted-value (AAPL)

- **GRU**

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gru (GRU)                    (None, 30, 64)            14976
_________________________________________________________________
gru_1 (GRU)                  (None, 64)                24960
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 130
=================================================================
Total params: 40,066
Trainable params: 40,066
Non-trainable params: 0
```

Figure13. model architecture

```
Total training time: 81.92196726799011 seconds
```

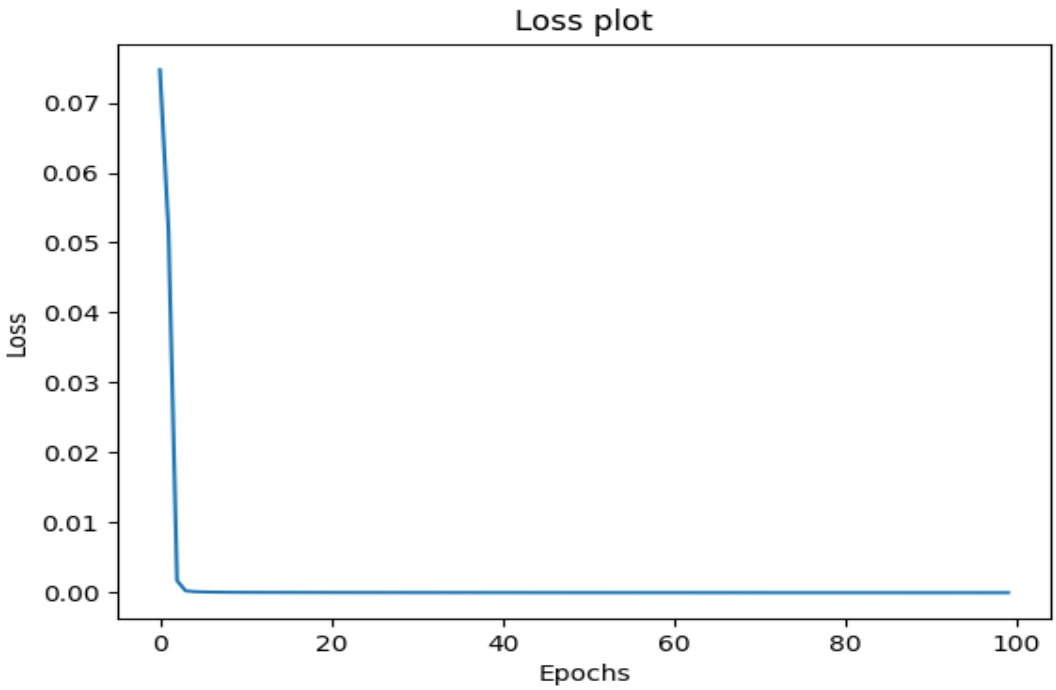

Figure14. Loss representation during the Epochs

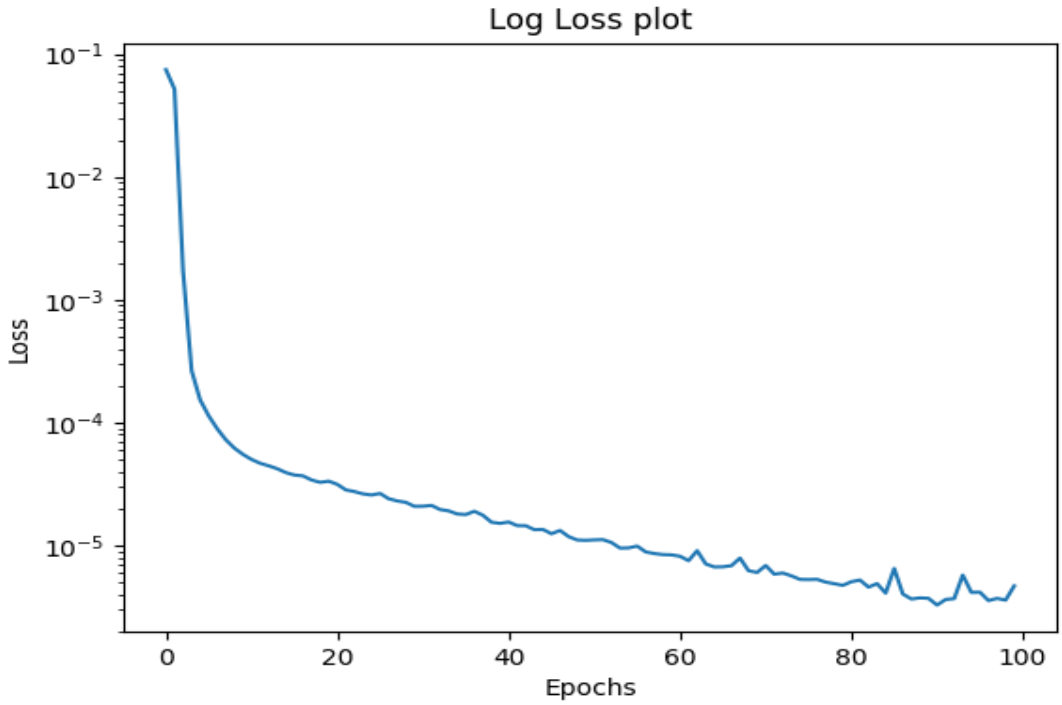

Figure15. Log-Loss representation during the Epochs

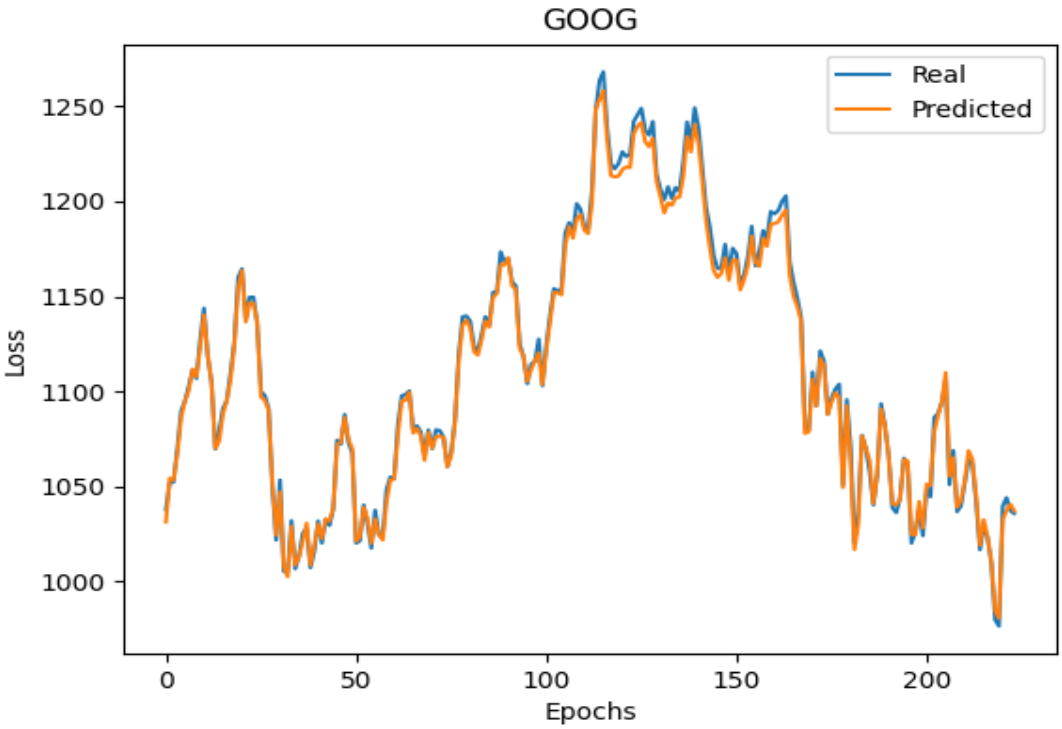

Figure16. Real-value VS Predicted-value (GOOG)

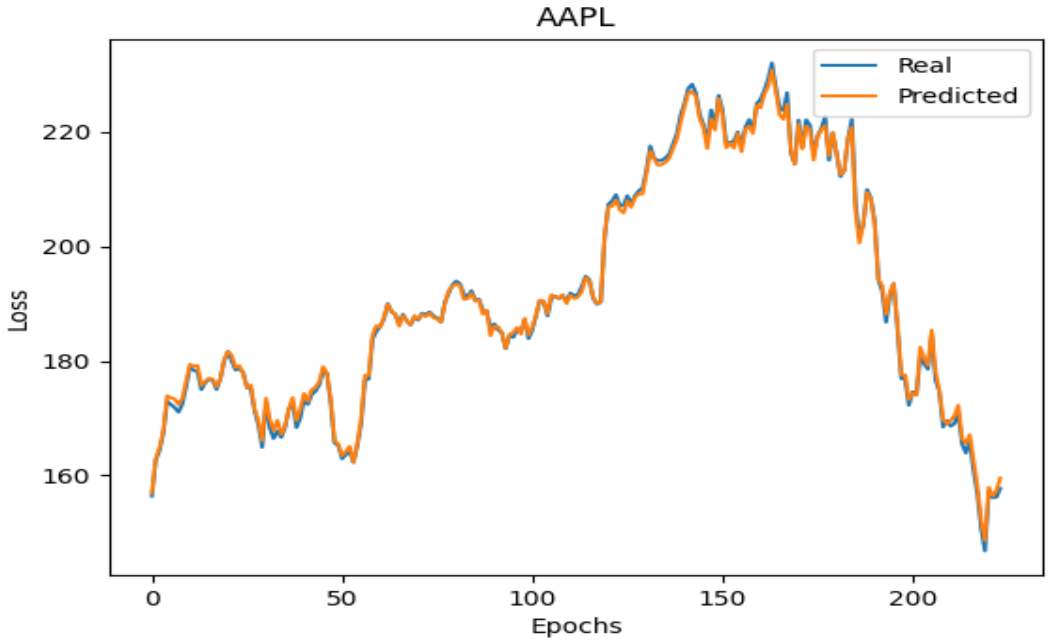

Figure17. Real-value VS Predicted-value (AAPL)

By looking at the results we conclude that GRU outperforms the other two memory cells

Performance: $GRU > LSTM > RNN$

also training time is least for RNN and GRU comes second, and LSTM takes the longest. We can see that as the memory unit complexity (number of parameters) gets bigger, it requires more training time.

Training time: $LSTM > GRU > RNN$

2.2

- GRU + MAPE

```
Total training time: 145.55148577690125 seconds
```

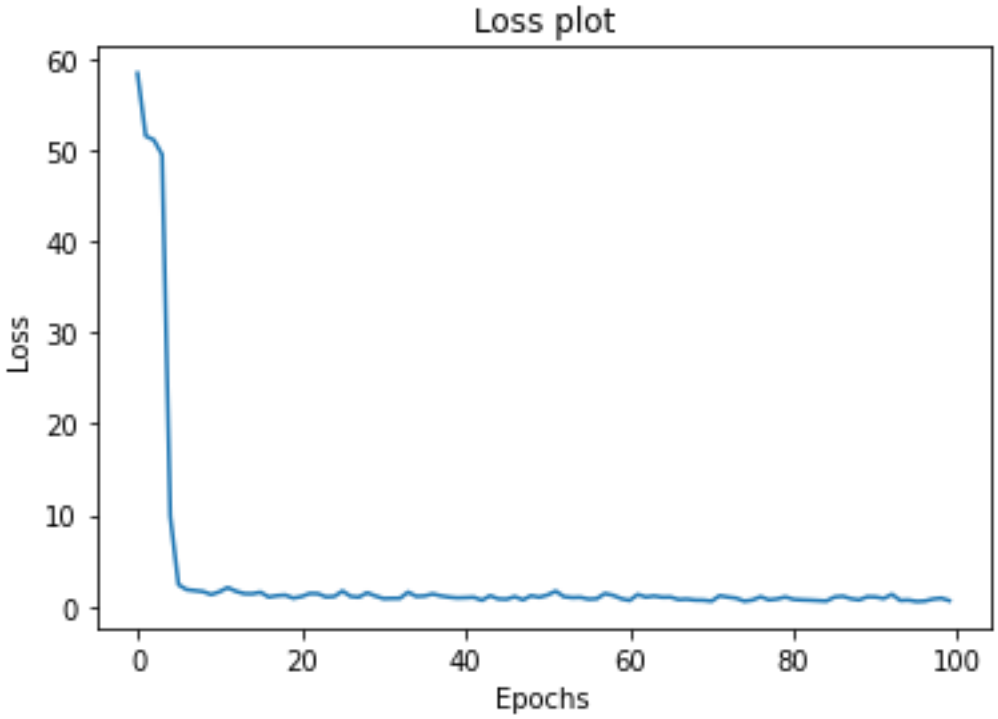

Figure18. Loss representation during the Epochs

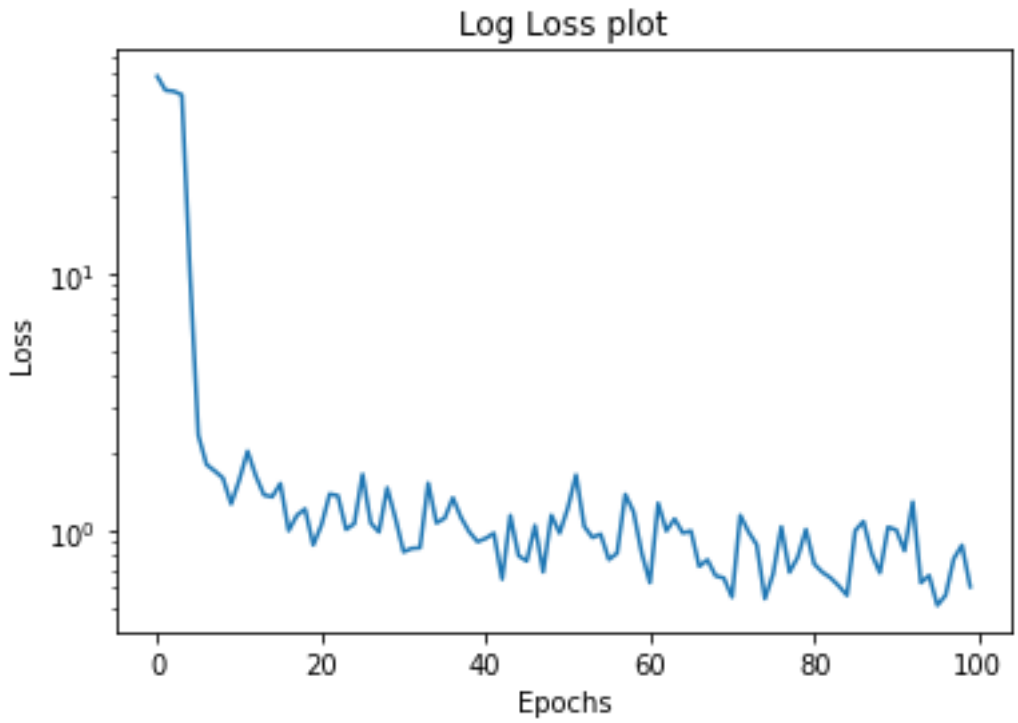

Figure19. Log-Loss representation during the Epochs

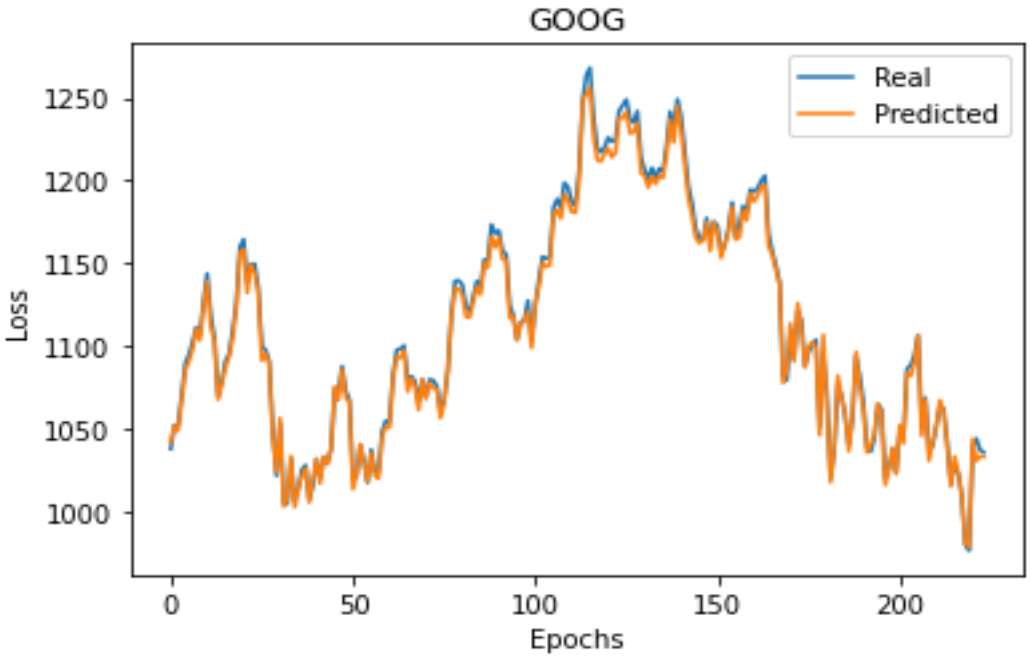

Figure20. Real-value VS Predicted-value (GOOG)

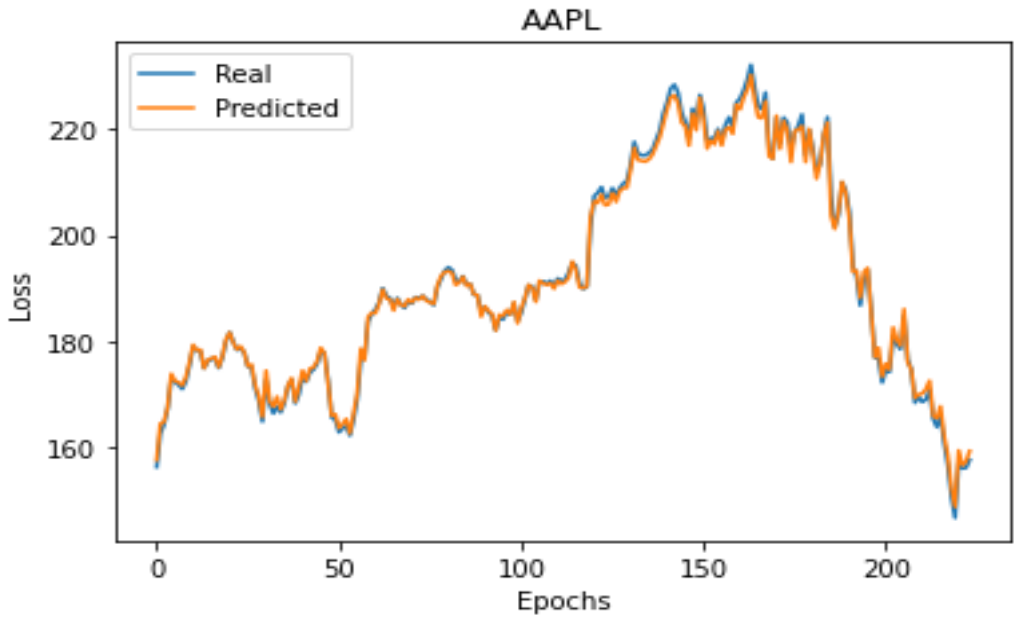

Figure21. Real-value VS Predicted-value (AAPL)

- MAPE + RNN

```
Total training time: 60.38407301902771 seconds
```

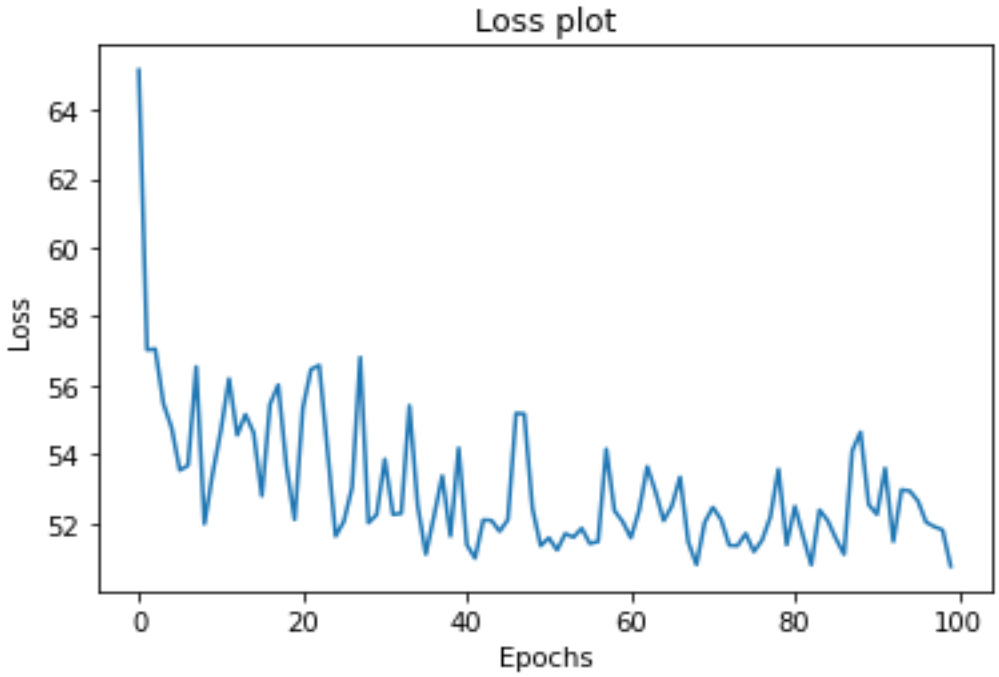

Figure22. Loss representation during the Epochs

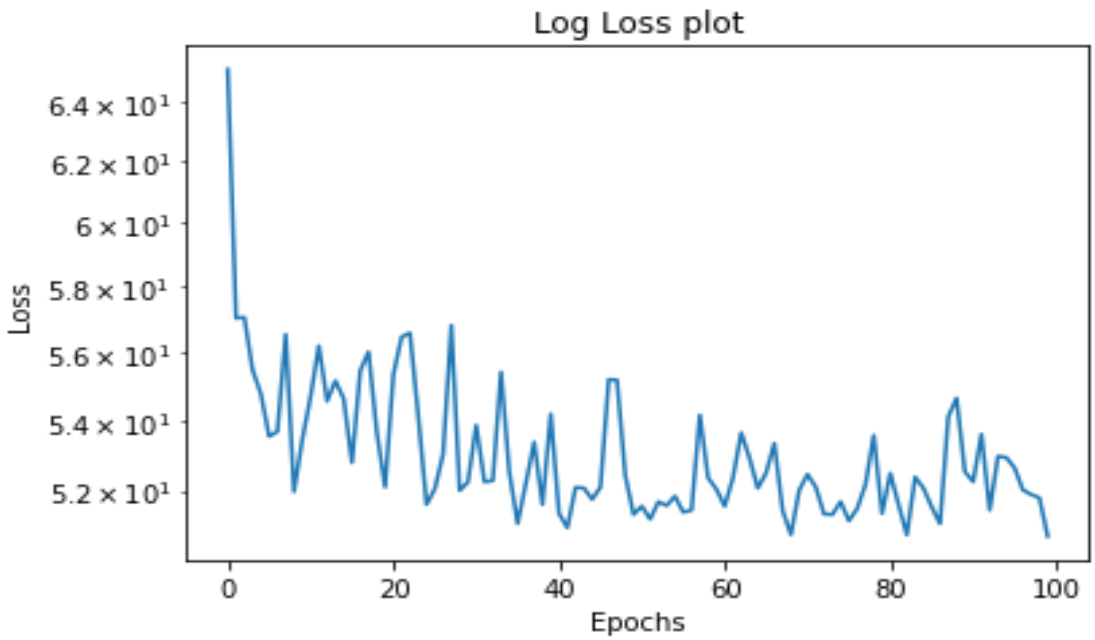

Figure23. simple LSTM network to learn sequences of characters

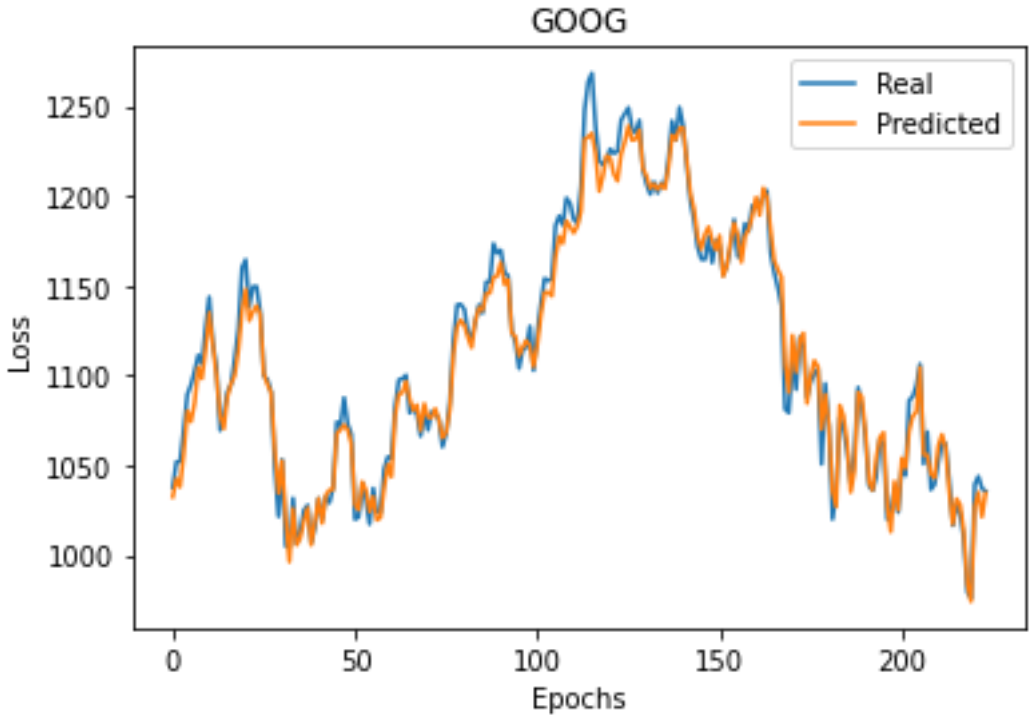

Figure24. Real-value VS Predicted-value (GOOG)

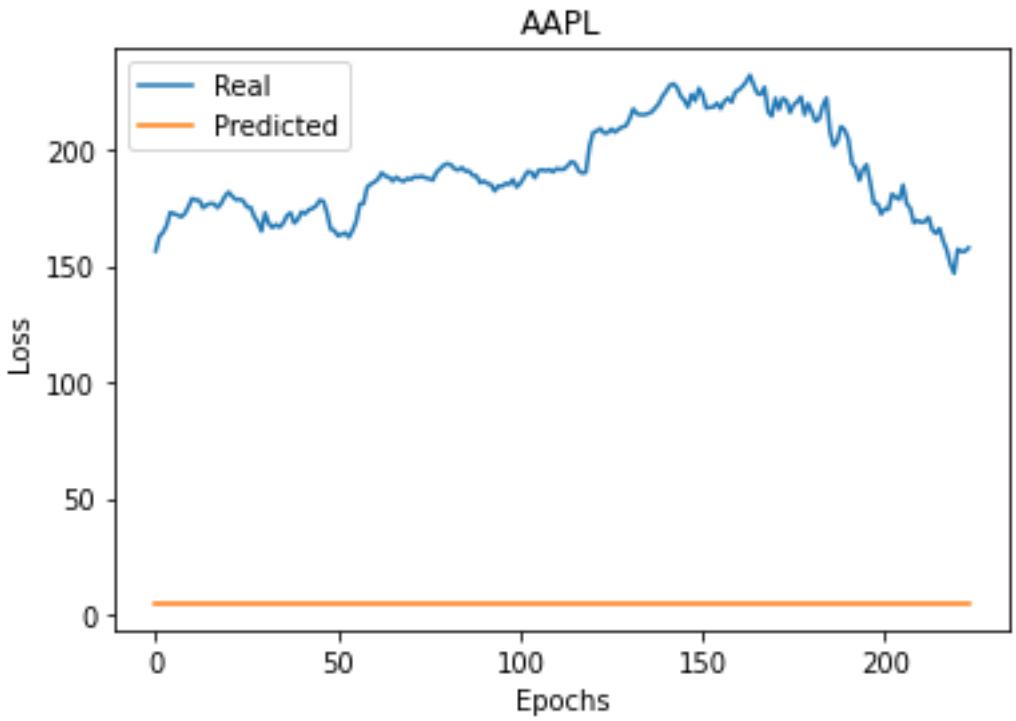

Figure25. Real-value VS Predicted-value (AAPL)

- MAPE + LSTM

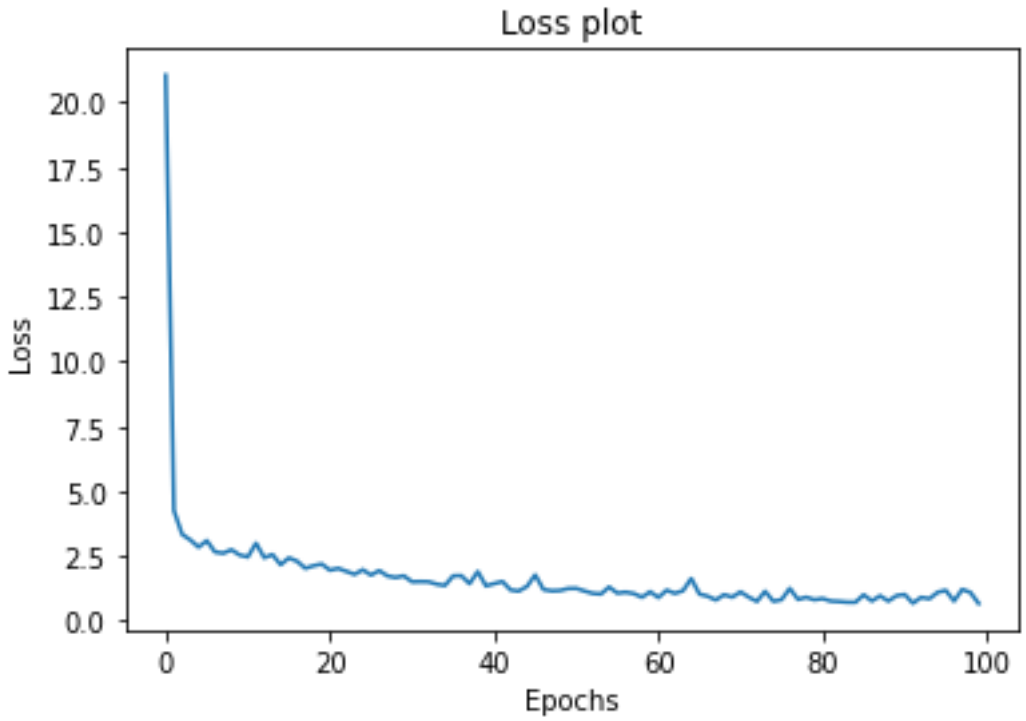

Figure26. Loss representation during the Epochs

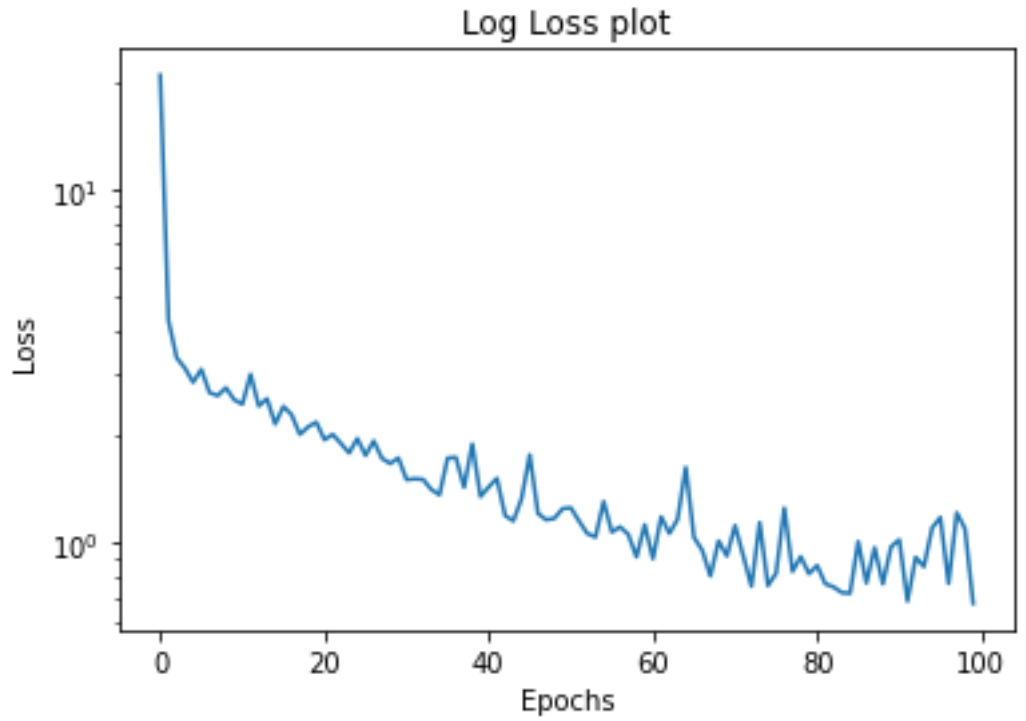

Figure27. Log-Loss representation during the Epochs

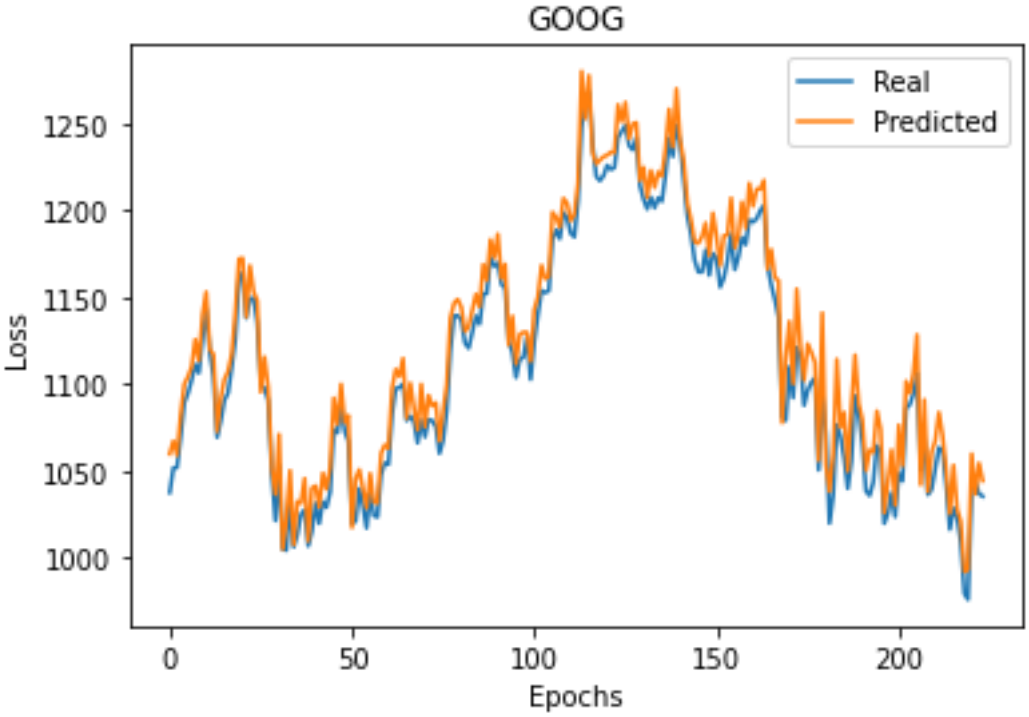

Figure28. Real-value VS Predicted-value (GOOG)

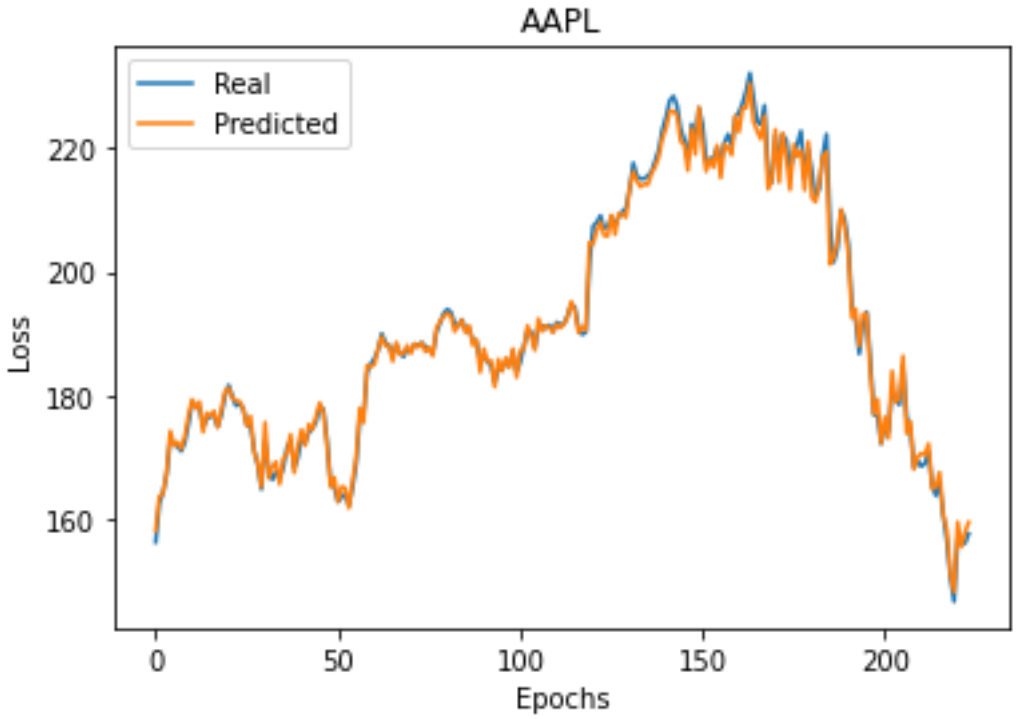

Figure29. Real-value VS Predicted-value (AAPL)

As we can see from the results, the MAPE loss function doesn't improve GRU and LSTM but reduces performance of RNN, as it has a constant gradient it suffers from vanishing gradient and thus the model is not trained well for RNN

2.3

```
Total training time: 149.08284449577332 seconds
```

- LSTM + Adagrad

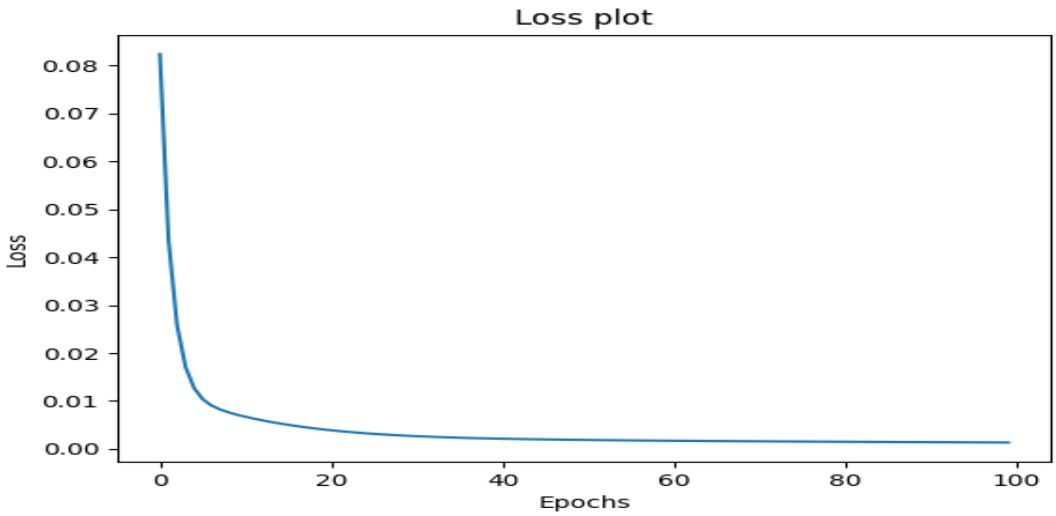

Figure30. Loss representation during the Epochs

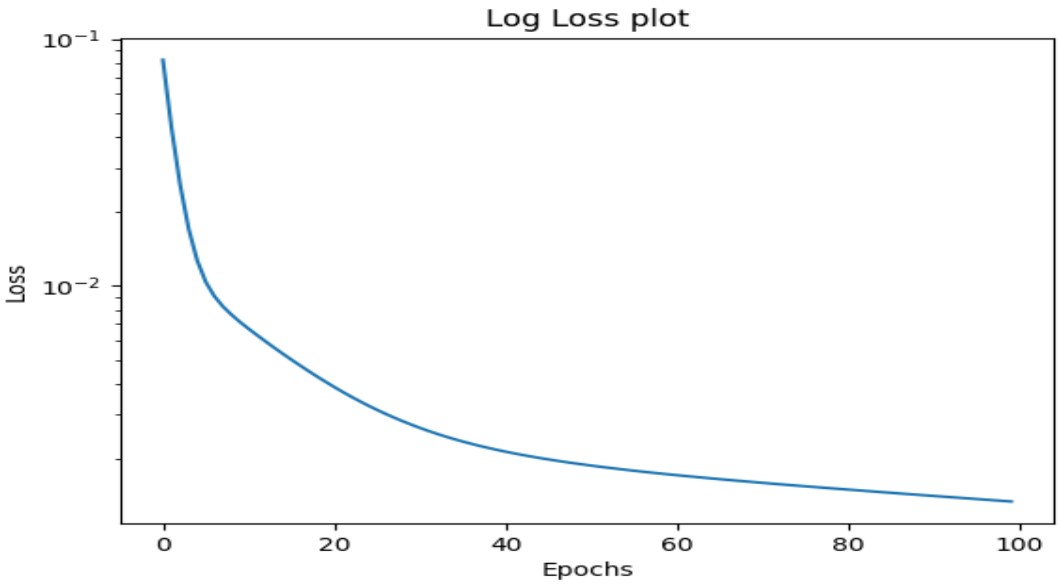

Figure31. Log-Loss representation during the Epochs

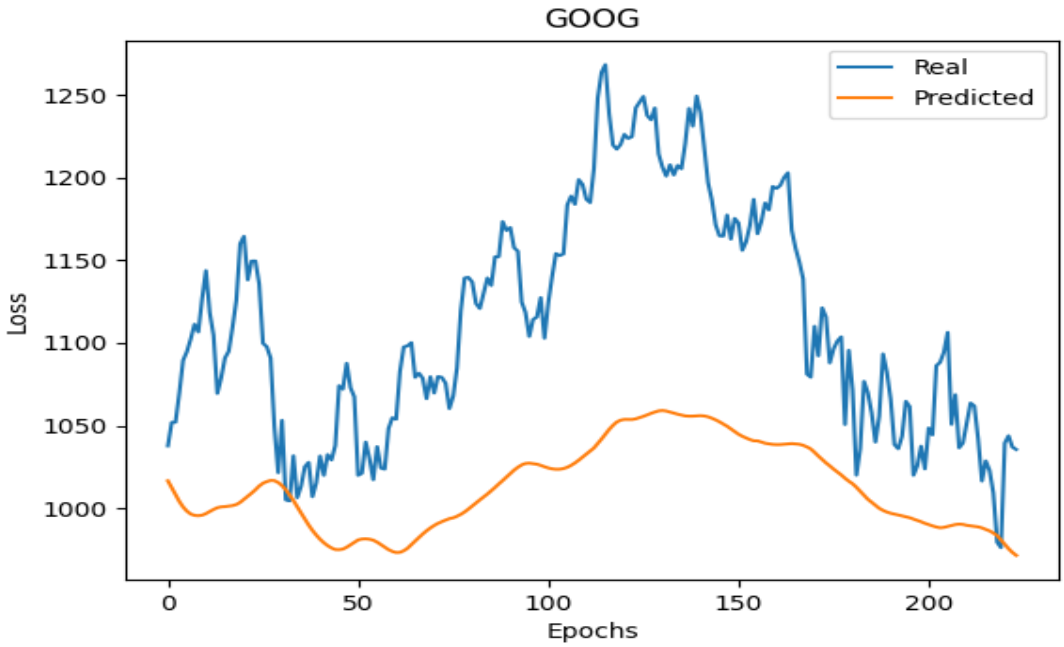

Figure32. Real-value VS Predicted-value (GOOG)

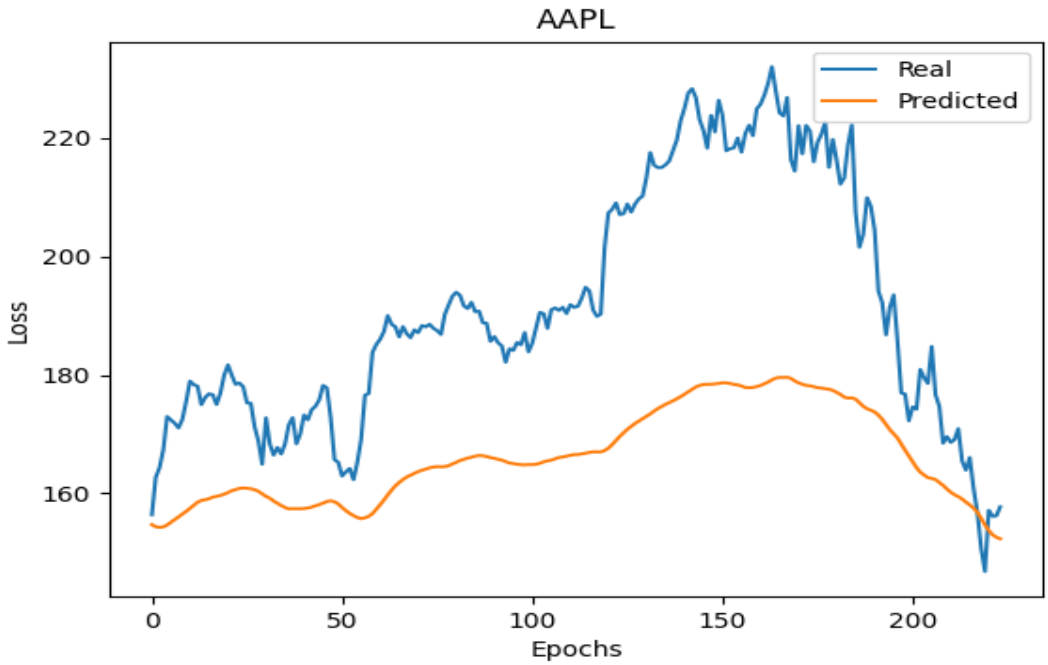

Figure33. Real-value VS Predicted-value (AAPL)

- LSTM + RMSprop

```
Total training time: 205.3320653438568 seconds
```

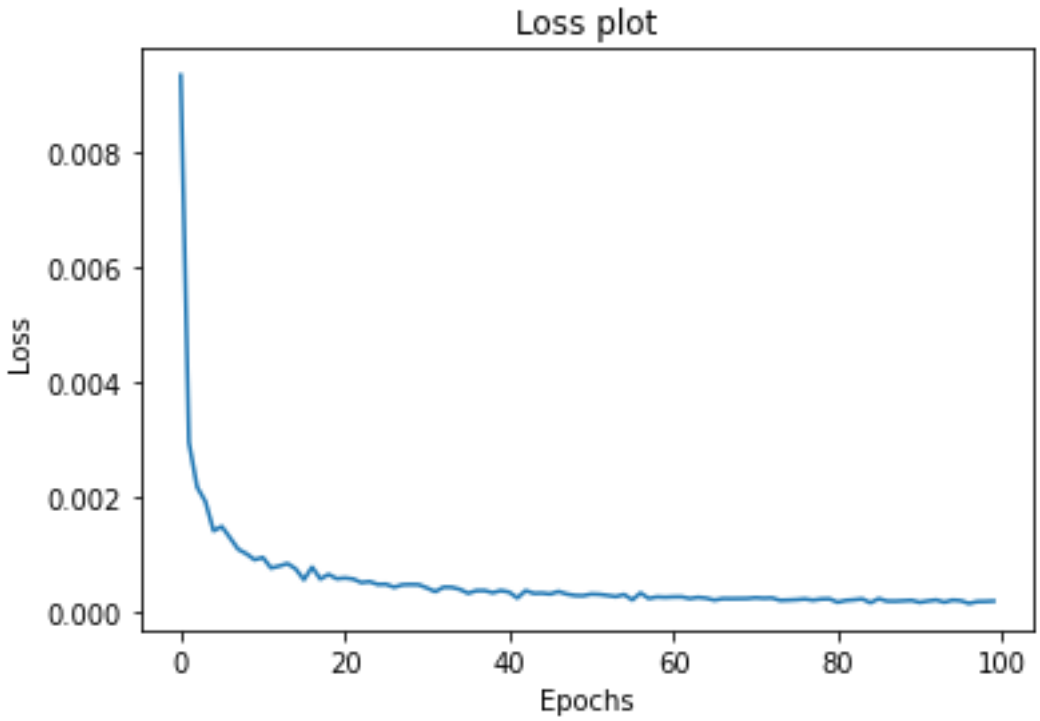

Figure34. Loss representation during the Epochs

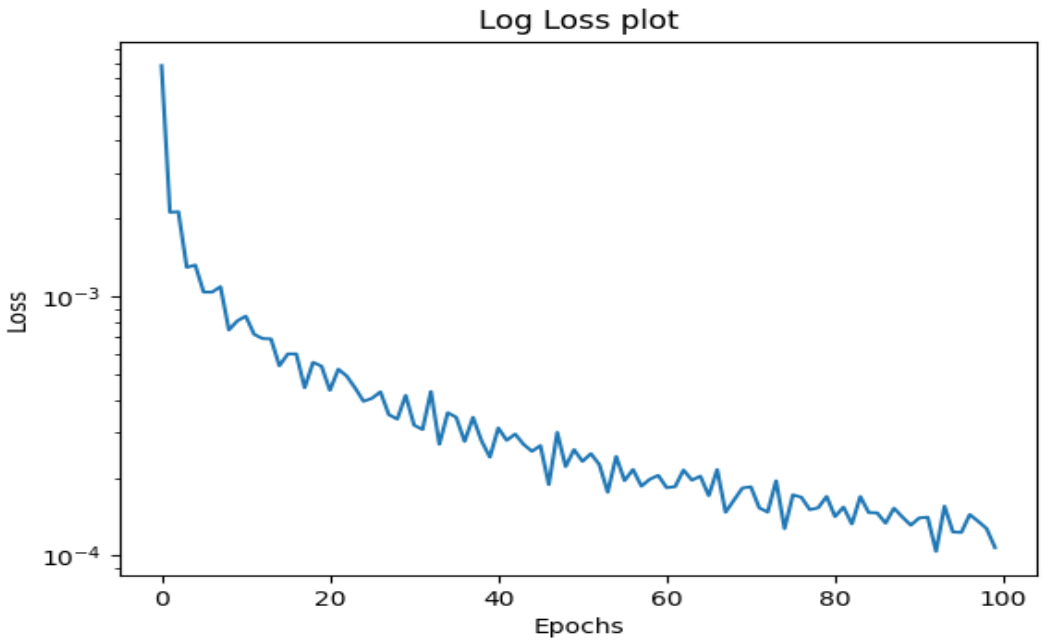

Figure35. Log- Loss representation during the Epochs

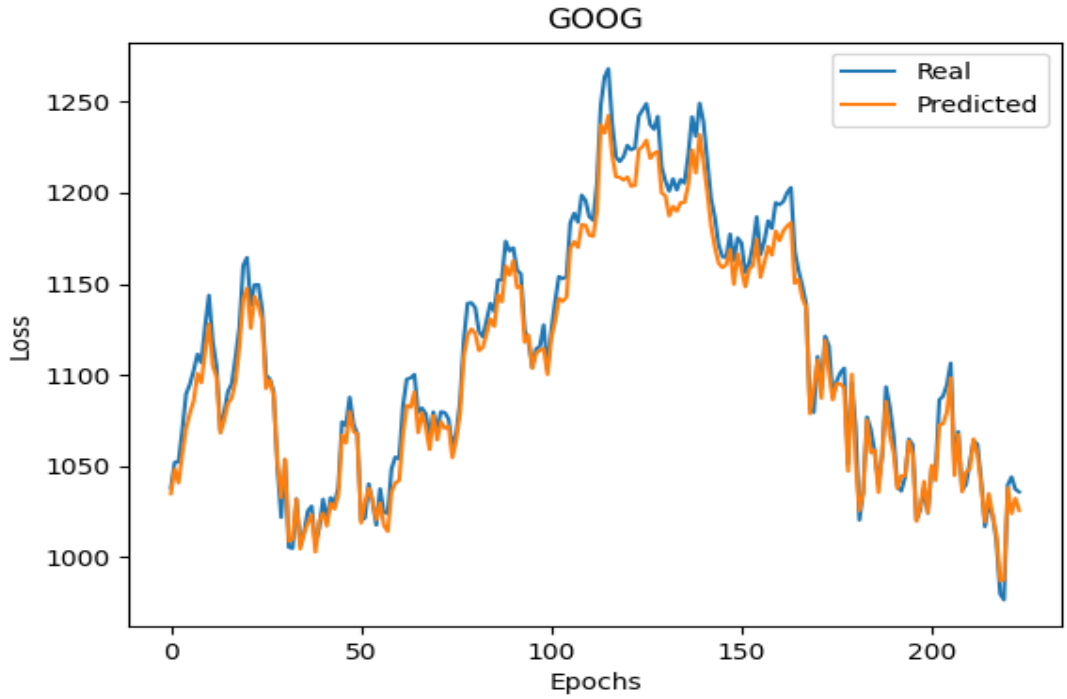

Figure36. Real-value VS Predicted-value (GOOG)

Figure37. Real-value VS Predicted-value (AAPL)

- ADAgrad + RNN

```
Total training time: 59.3671555519104 seconds
```

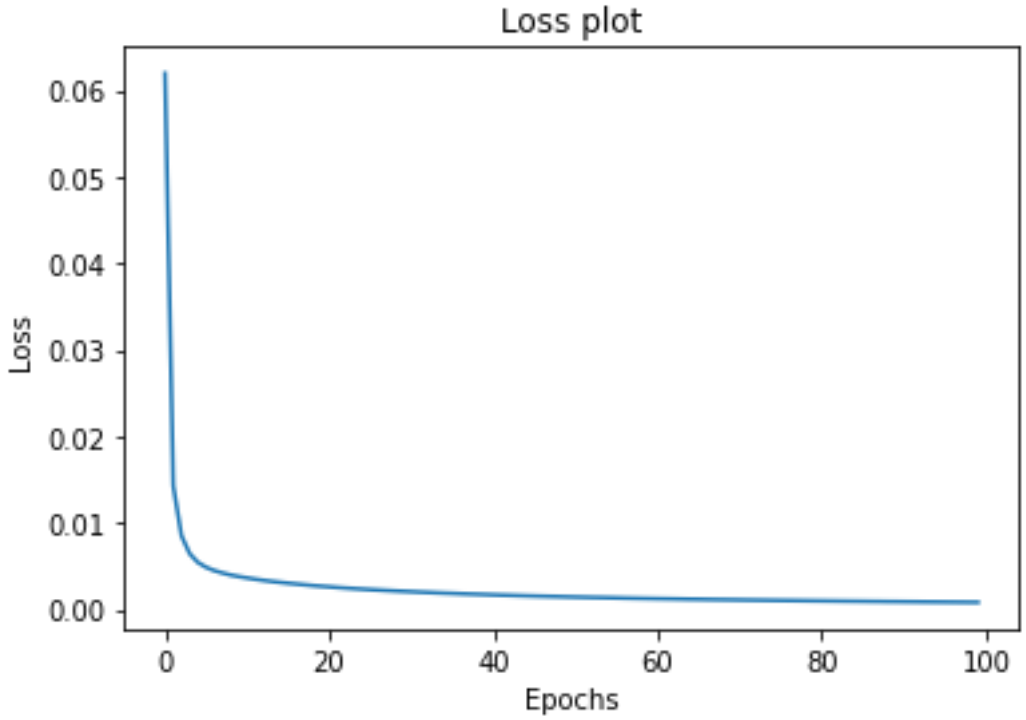

Figure38. Loss representation during the Epochs

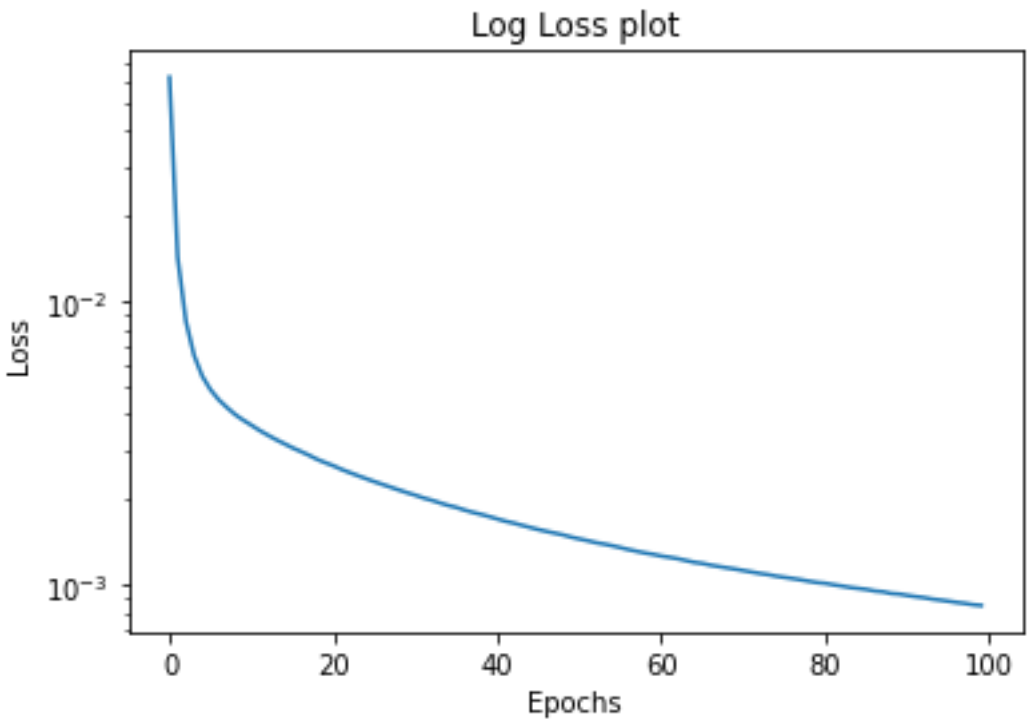

Figure39. Log-Loss representation during the Epochs

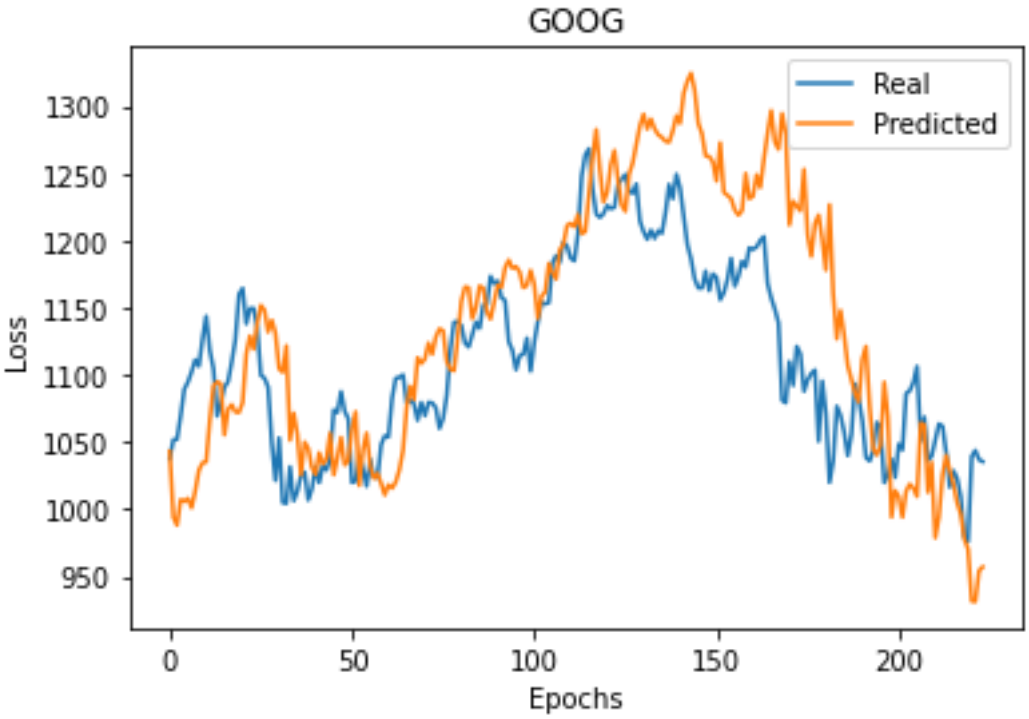

Figure40. Real-value VS Predicted-value (GOOG)

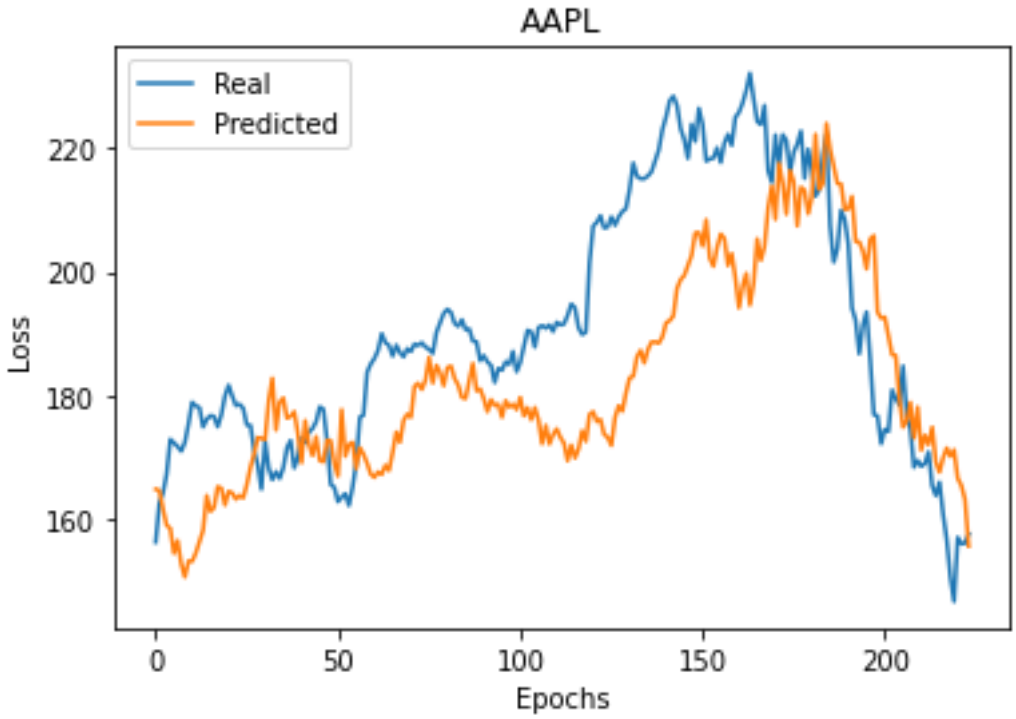

Figure41. Real-value VS Predicted-value (AAPL)

- RMSprop +RNN

```
Total training time: 63.164148807525635 seconds
```

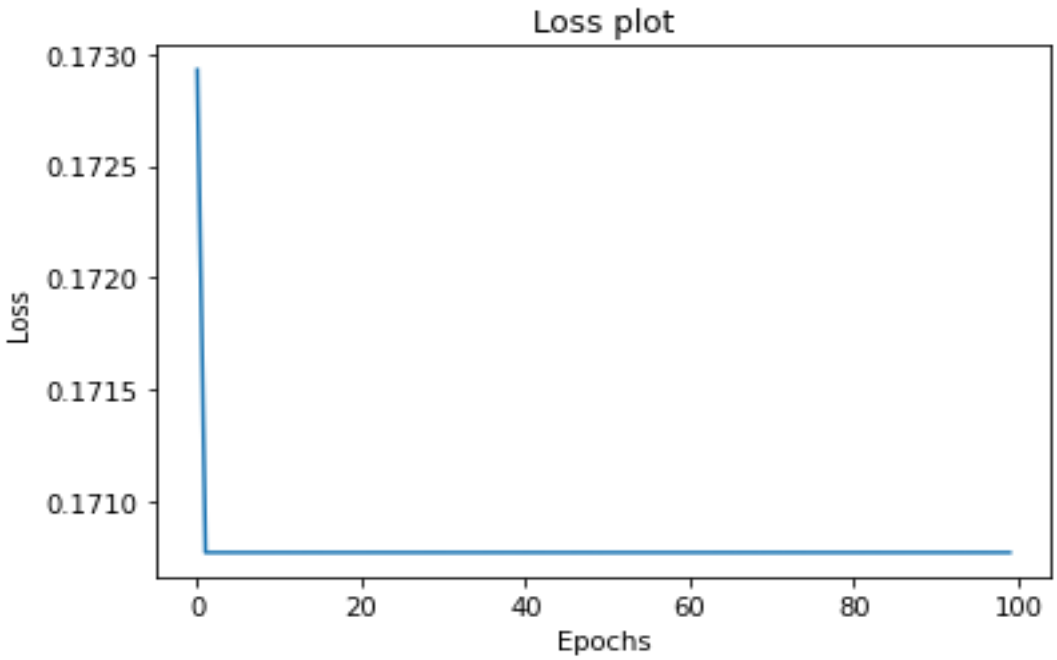

Figure42. Loss representation during the Epochs

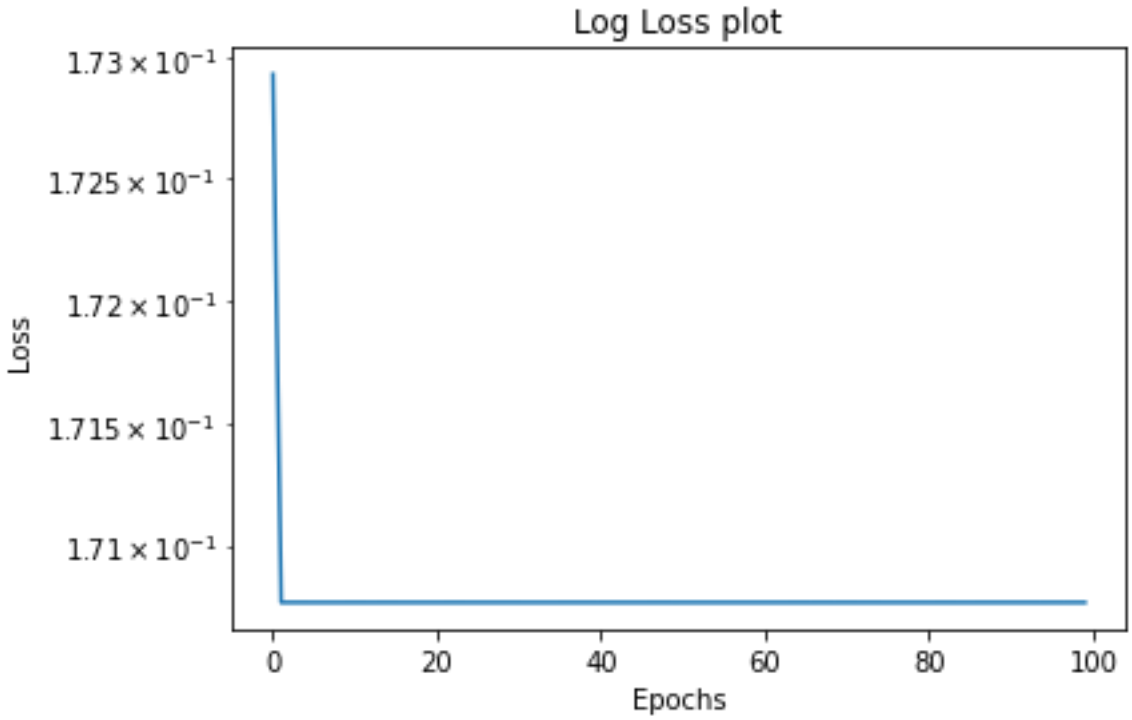

Figure43. Log-Loss representation during the Epochs

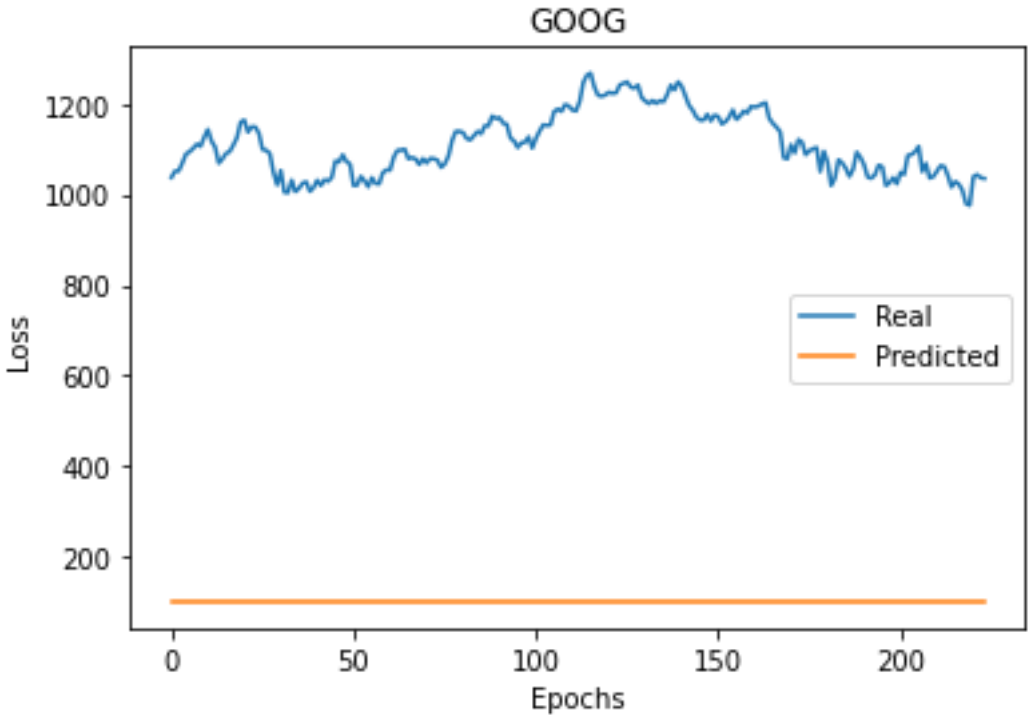

Figure44. Real-value VS Predicted-value (GOOG)

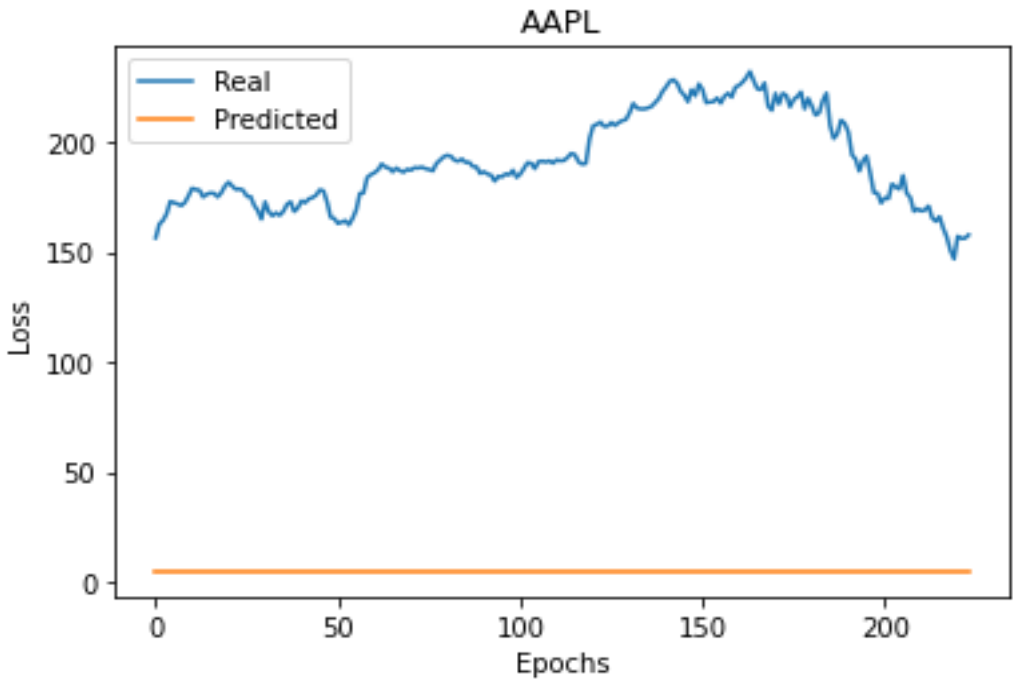

Figure45. Real-value VS Predicted-value (AAPL)

- ADAgrad + GRU

```
Total training time: 137.29109835624695 seconds
```

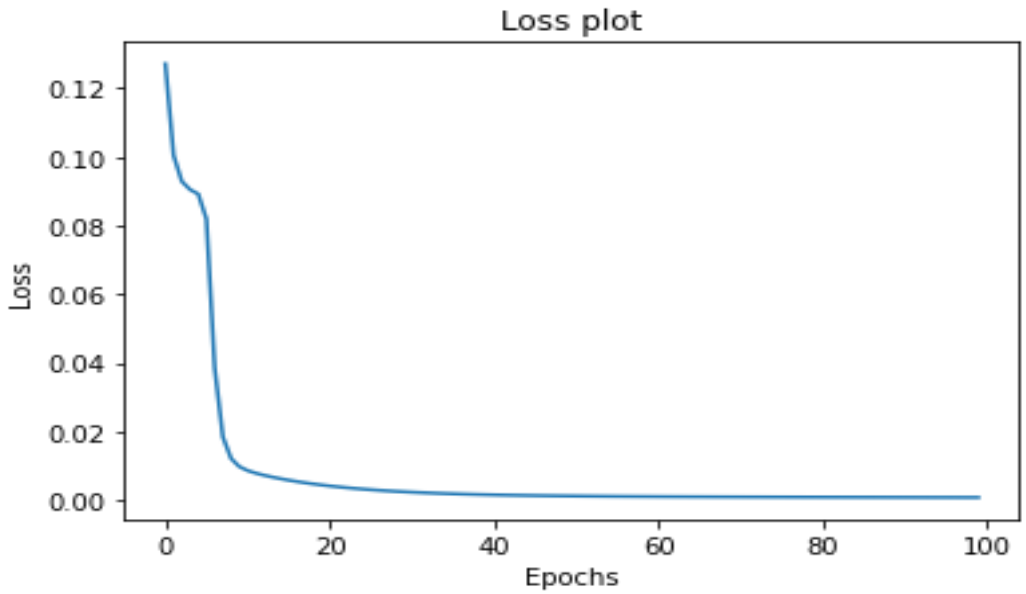

Figure46. Loss representation during the Epochs

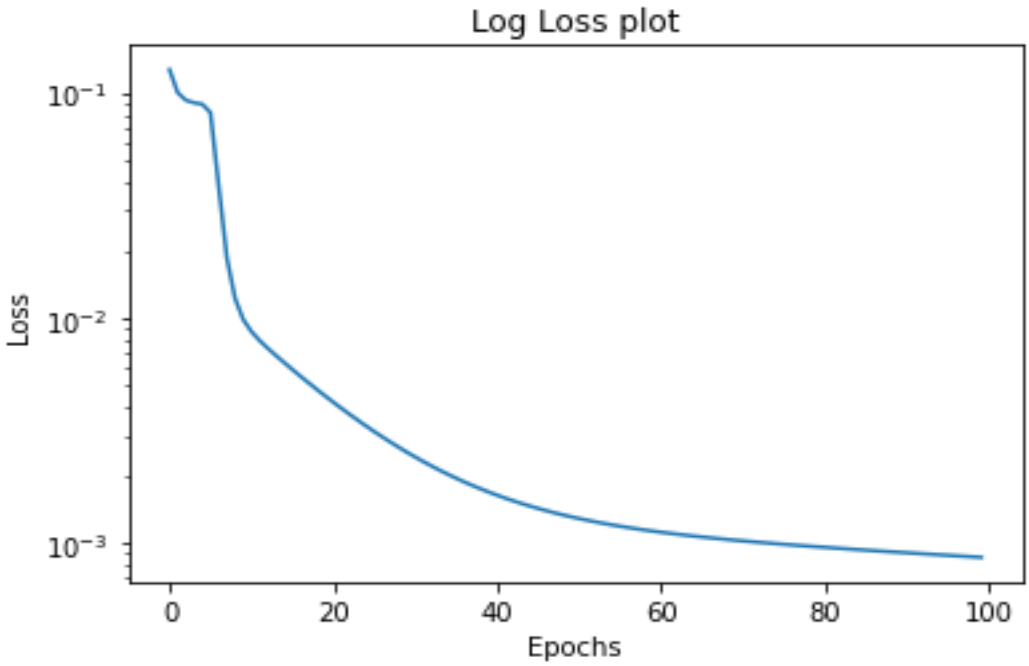

Figure47. Log-Loss representation during the Epochs

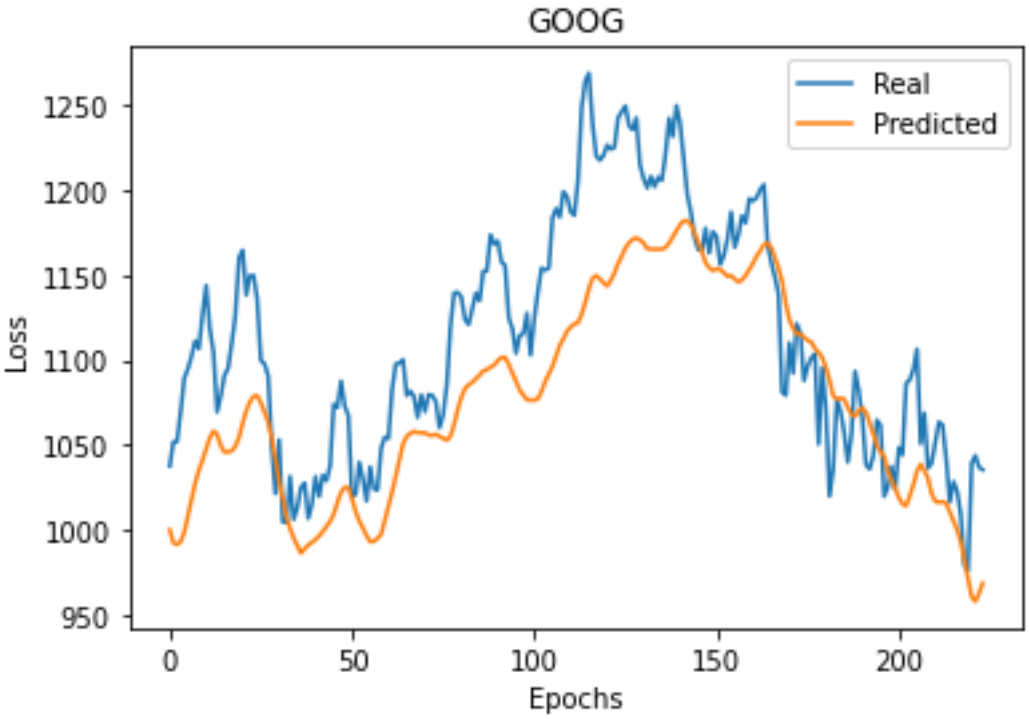

Figure48. Real-value VS Predicted-value (GOOG)

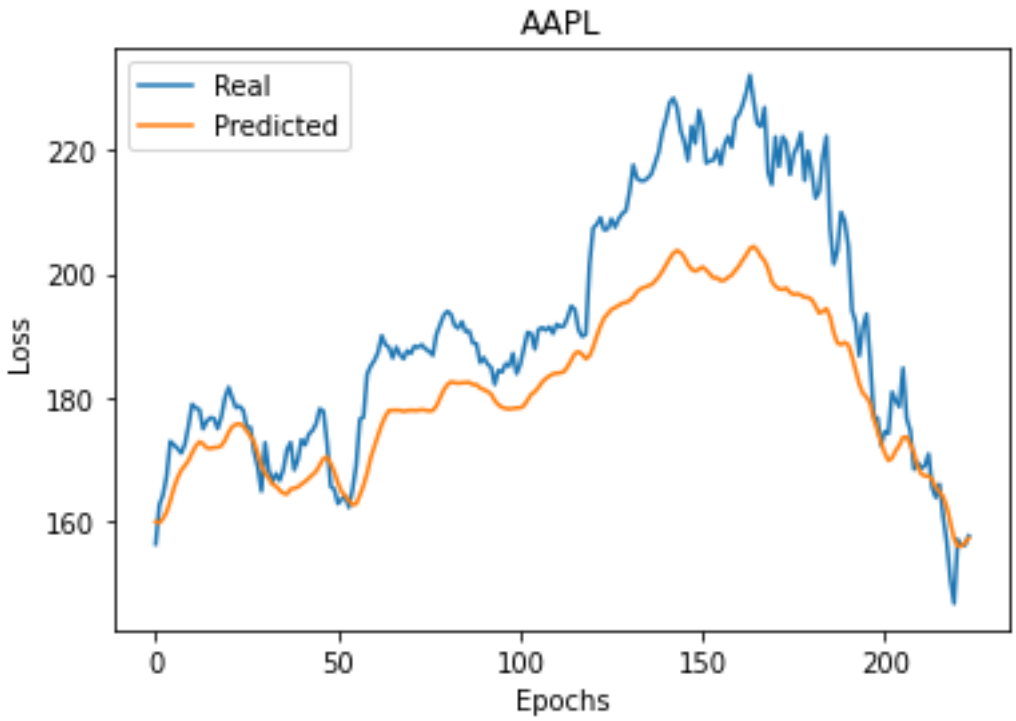

Figure49. Real-value VS Predicted-value (AAPL)

- RMSprop + GRU

```
Total training time: 137.48347330093384 seconds
```

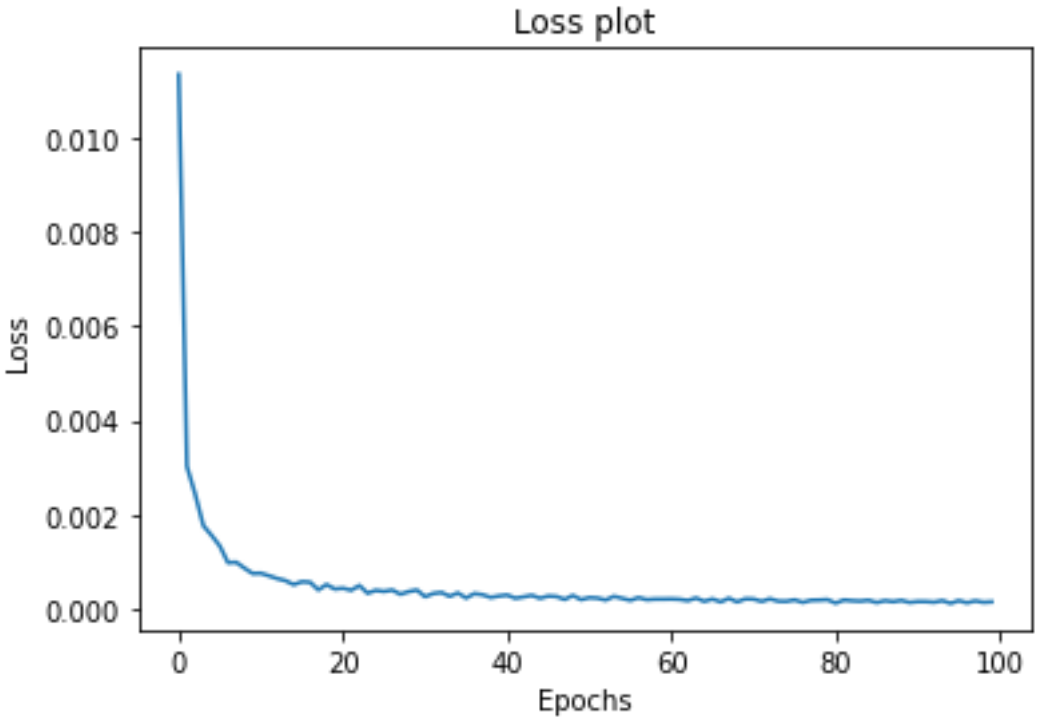

Figure50. Loss representation during the Epochs

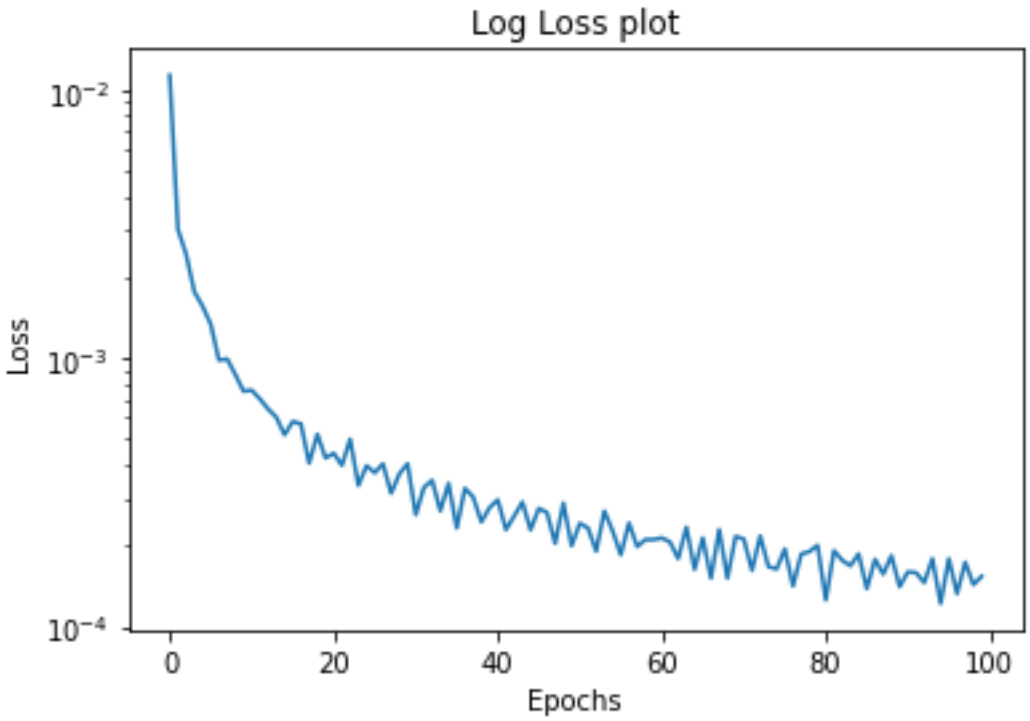

Figure51. Log-Loss representation during the Epochs

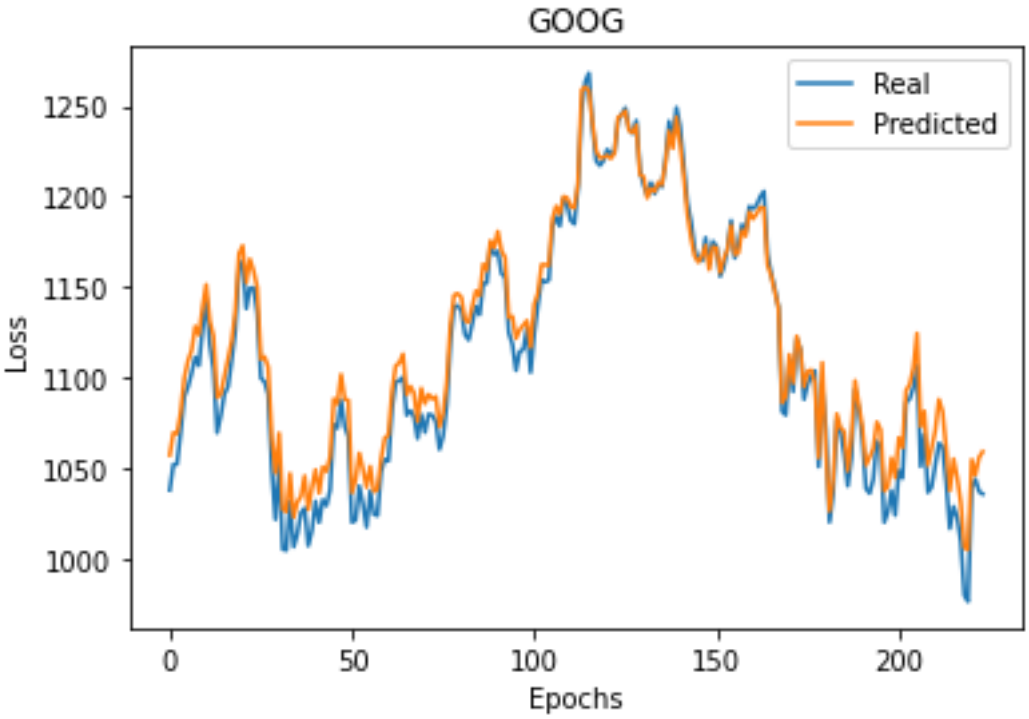

Figure52. Real-value VS Predicted-value (GOOG)

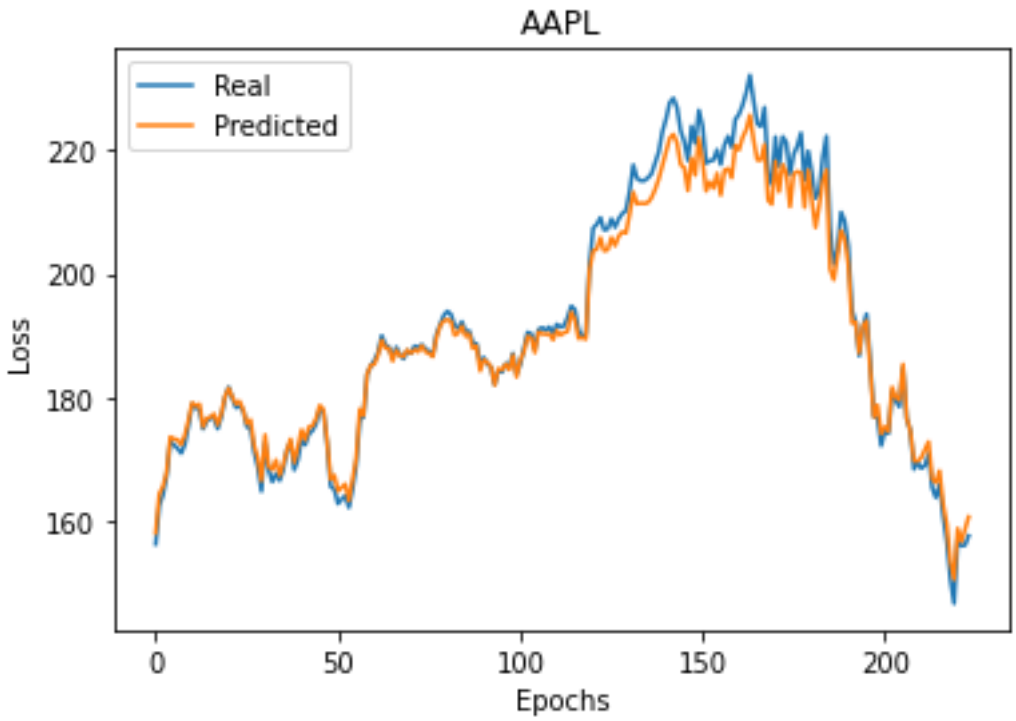

Figure53. Real-value VS Predicted-value (AAPL)

As we can see from the results both ADAgrad and RMSprop optimizers get outperformed by Adam, because Adam uses both first and second moments and benefits from advantages of two other optimizers, so it is generally the best choice.

Also, Adam takes noticeably less time to train.

Table 2.1- Comparison between optimizers

Optimiser	Year	Learning Rate	Gradient
Momentum	1964		✓
AdaGrad	2011	✓	
RMSprop	2012	✓	
Adadelta	2012	✓	
Nesterov	2013		✓
Adam	2014	✓	✓
AdaMax	2015	✓	✓
Nadam	2015	✓	✓
AMSGrad	2018	✓	✓

2.4

- LSTM + 0.1 Dropout

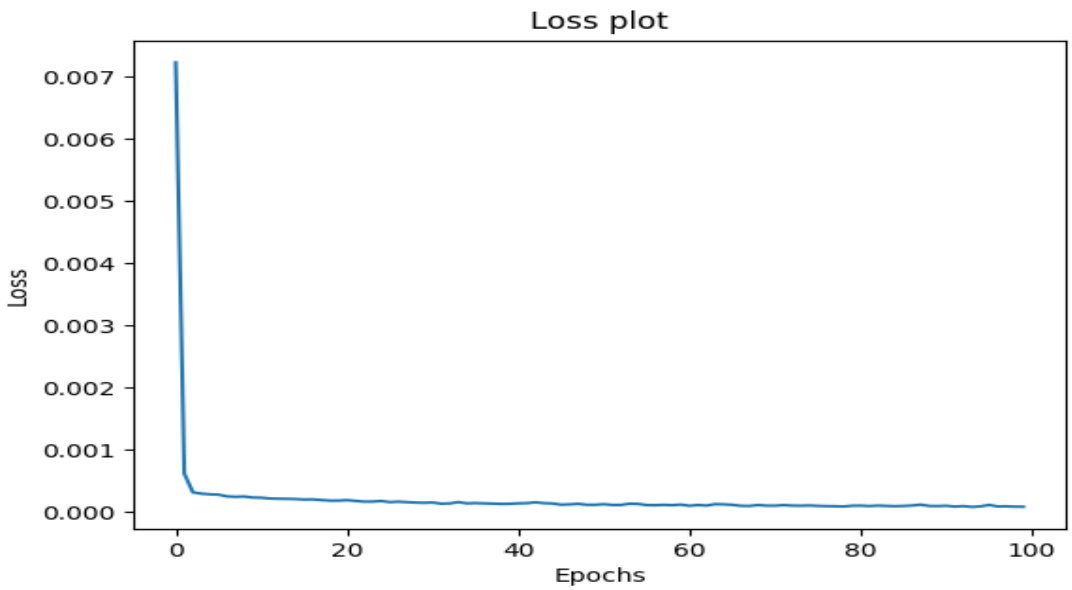

Figure54. Loss representation during the Epochs

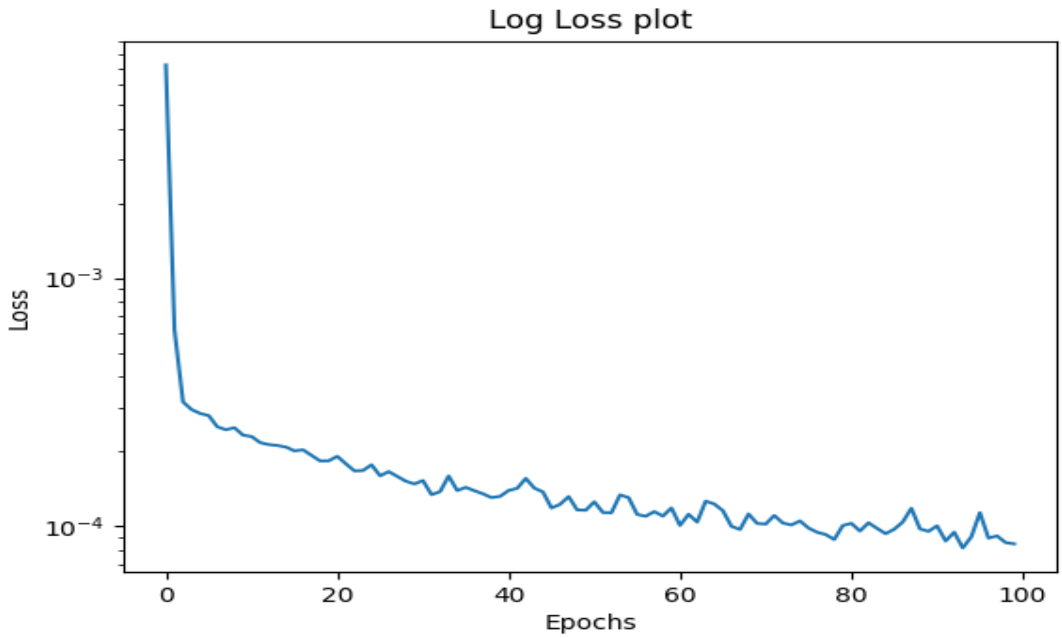

Figure55. Log-Loss representation during the Epochs

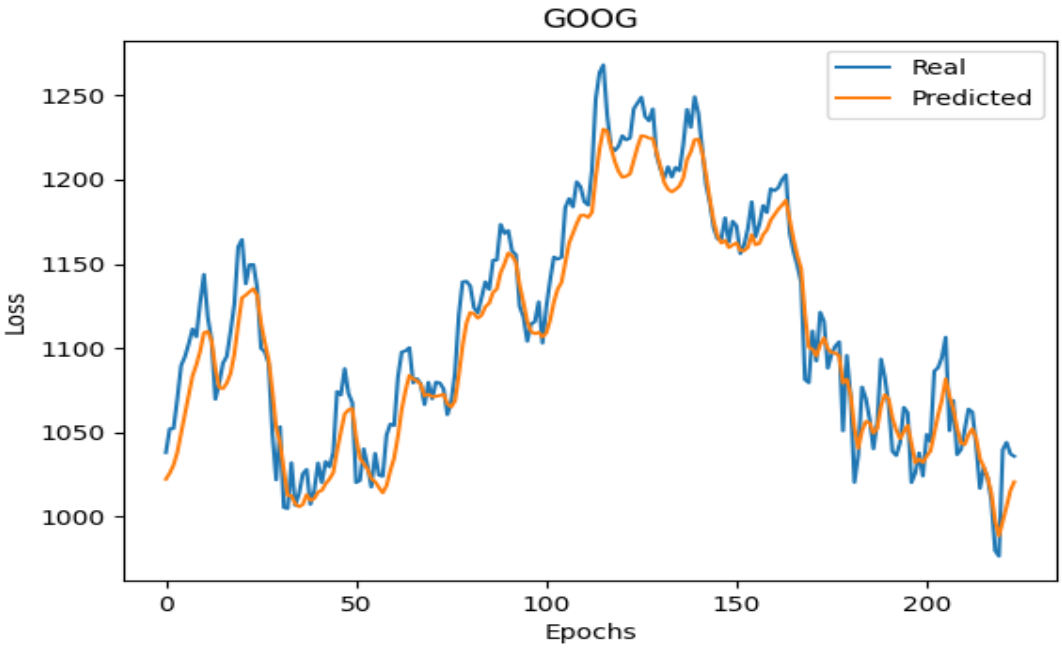

Figure56. Real-value VS Predicted-value (GOOG)

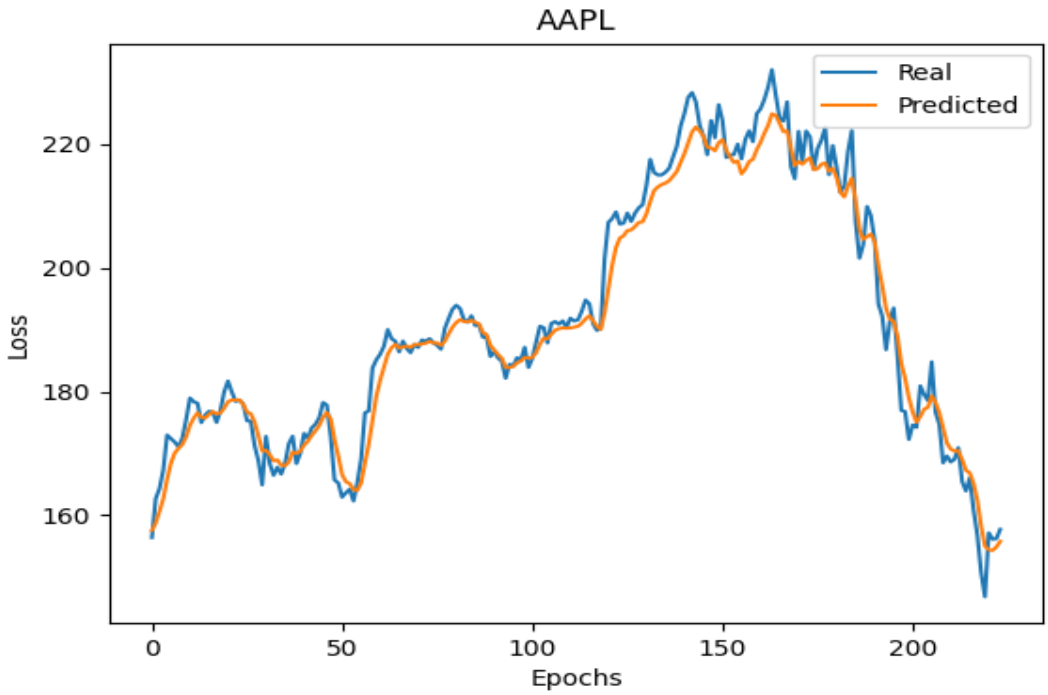

Figure57. Real-value VS Predicted-value (AAPL)

- RNN + dropout 0.1

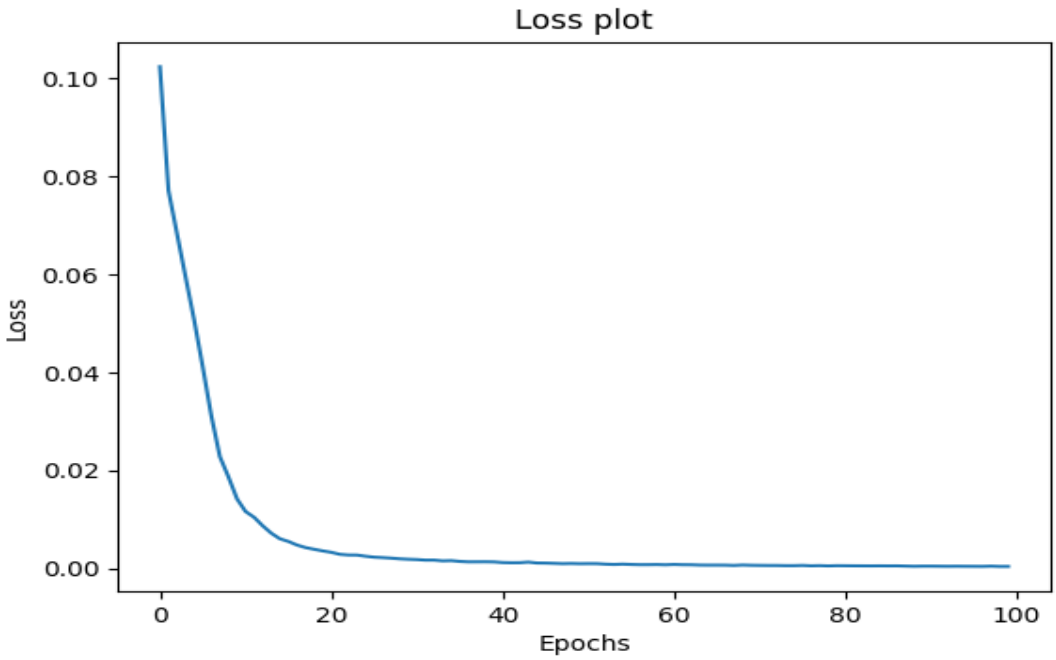

Figure58. Loss representation during the Epochs

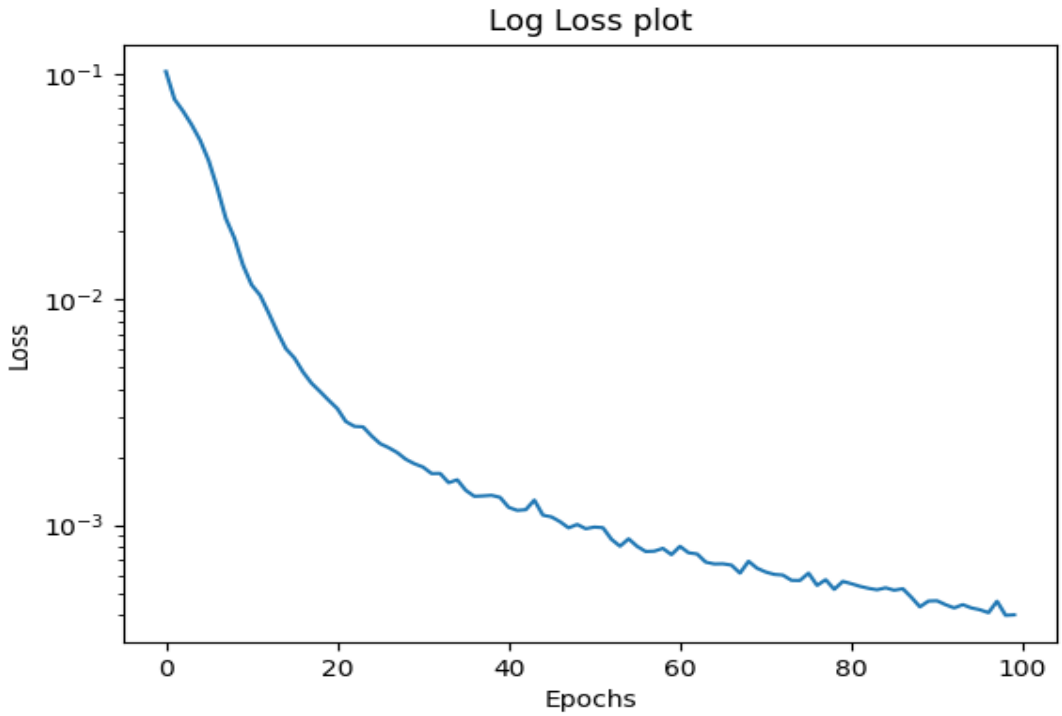

Figure59. Log- Loss representation during the Epochs

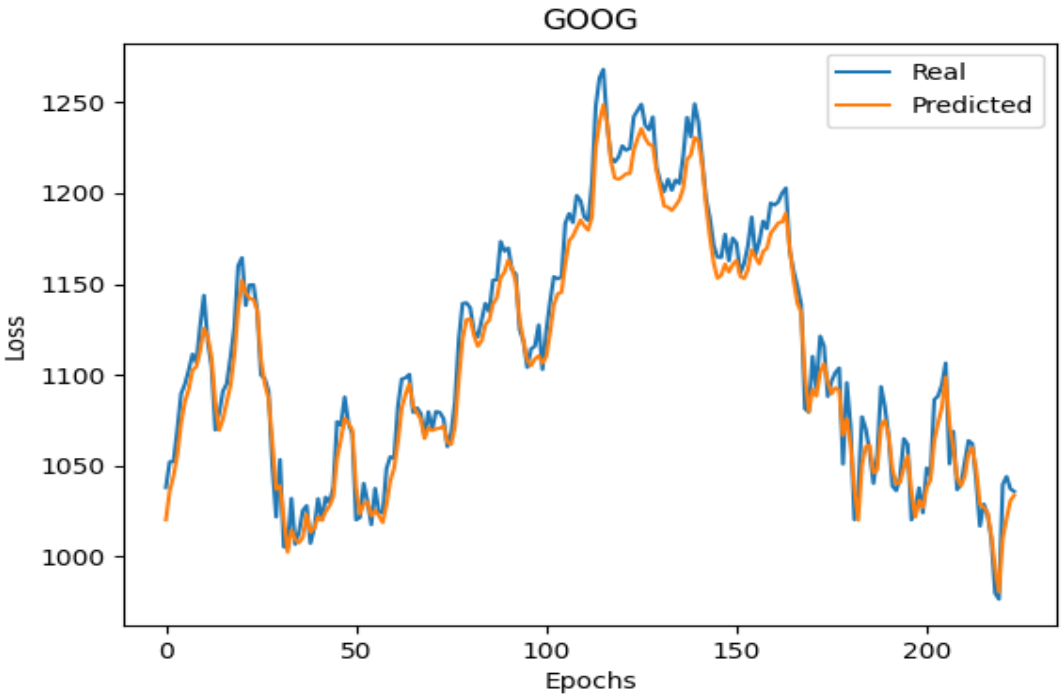

Figure60. Real-value VS Predicted-value (GOOG)

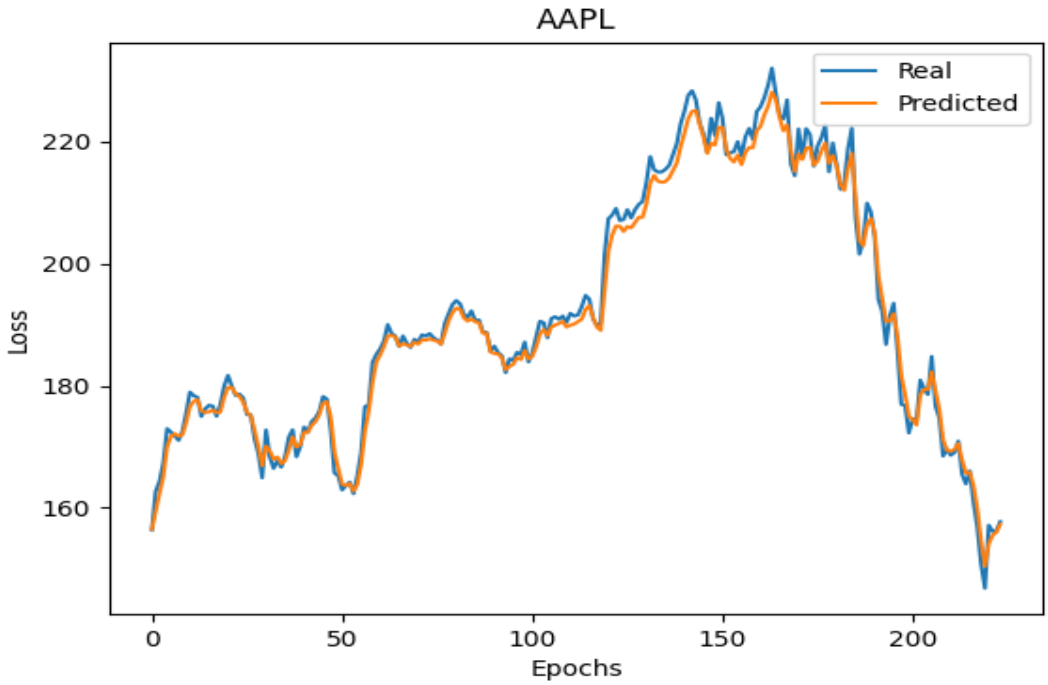

Figure61. Real-value VS Predicted-value (AAPL)

- GRU + 0.1 dropout

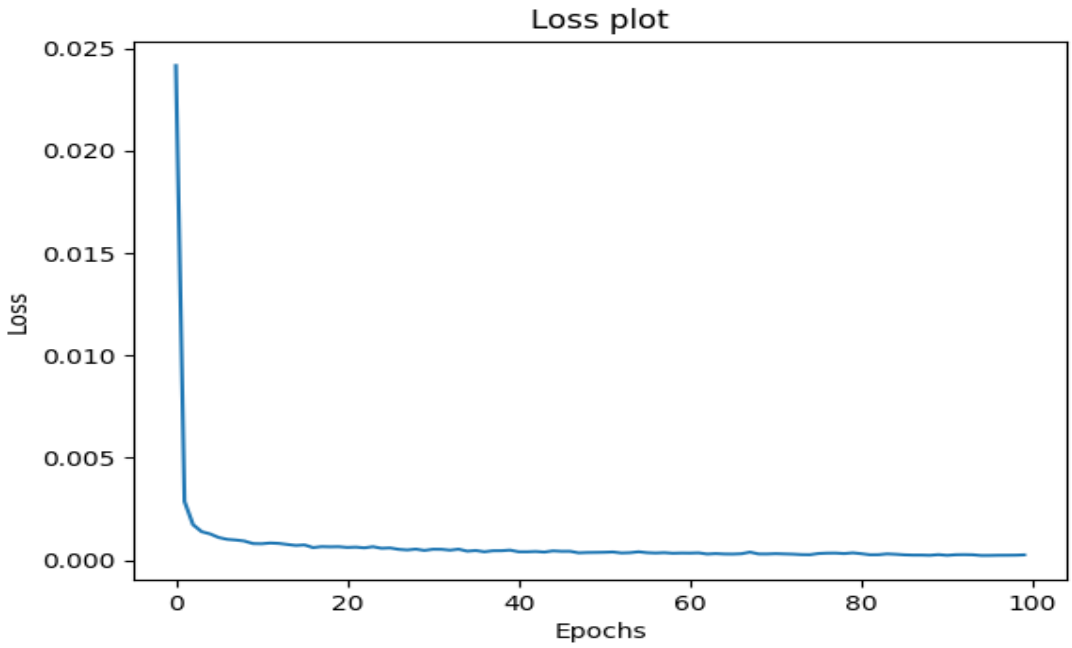

Figure62. Loss representation during the Epochs

Figure63. Log-Loss representation during the Epochs

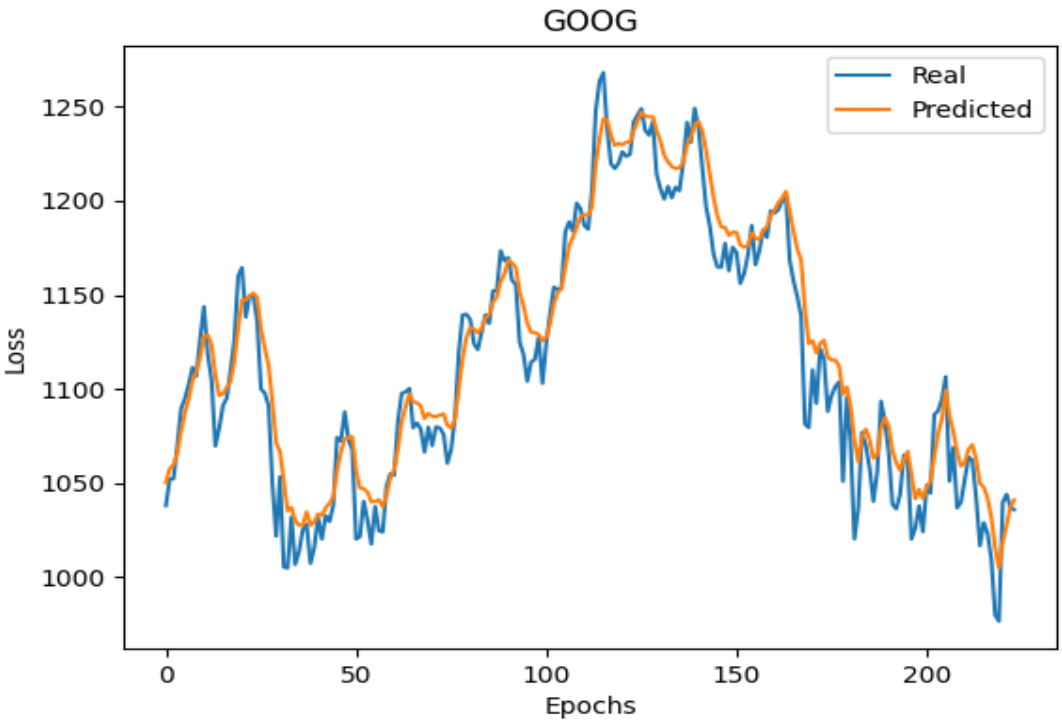

Figure64. Real-value VS Predicted-value (GOOG)

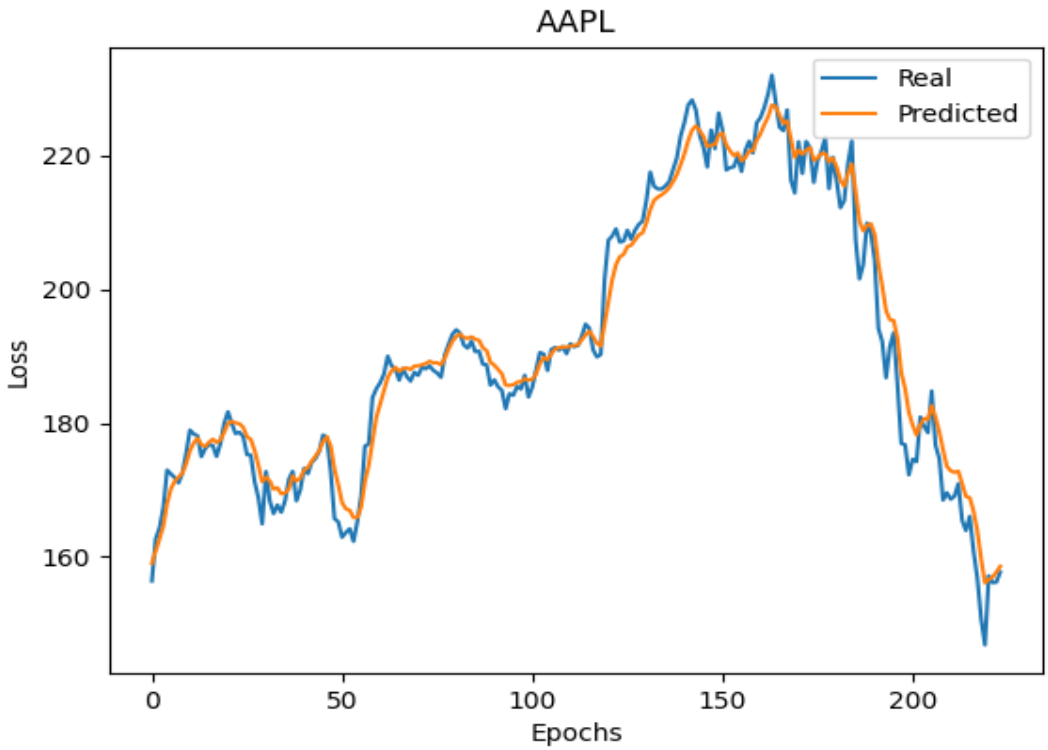

Figure65. Real-value VS Predicted-value (AAPL)

By comparing the results to part 2 we can see that adding dropout decreased performance of LSTM and GRU units but improved RNN unit.

Dropout is a regularization technique, and is most effective at preventing overfitting. However, there are several places when dropout can hurt performance.

- when the network is small relative to the dataset, regularization is usually unnecessary. If the model capacity is already low, lowering it further by adding regularization will hurt performance.

- when training time is limited. Usually, dropout hurts performance at the start of training, but results in the final "converged" error being lower. Therefore, if you don't plan to train until convergence, you may not want to use dropout.

3 QUESTION #2: TEXT GENERATION

In this part we intend to implement a Recurrent Neural Network which can be used to generate text.

First of all, it's so important to fit **an appropriate model** to our train-data which is the "**HARRY POTTER AND THE GOBLET OF FIRE** ".

3.1

In this part we are going to analyse the loss and accuracy of model and investigate whether the model is an appropriate one or not.

In this section we will develop a simple LSTM network to learn sequences of characters from **HARRY POTTER AND THE GOBLET OF FIRE**. In the next section we will use this model to generate new sequences of characters.

Further you can see the architecture of model:

```
model = Sequential()
model.add(LSTM(256, input_shape=(100, 1), stateful=False, return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(256))
model.add(Dropout(0.2))
model.add(Dense(len(chars)))
model.add(Activation('softmax'))
```

Figure66. simple LSTM network to learn sequences of characters

Some notes about our model:

we define a **single hidden LSTM layer with 256 memory units**. The network **uses dropout with a probability of 0.2**. The output layer is a **Dense layer using the softmax activation function** to output a probability prediction for each of **the 54 characters between 0 and 1**.

The problem is really a single character classification problem with 54 classes and as such is defined as optimizing the log loss (cross entropy), here **using the ADAM** optimization algorithm for speed.

we have trained the model **using the dataset** and further you can find **the results:**

training process:

```
Total Characters:  1107542
Total Vocab:  54
Total Patterns:  1107442
Epoch 1/40
8652/8652 [==============================] - ETA: 0s - loss: 2.7356 - accuracy: 0.2304
Epoch 1: loss improved from inf to 2.73558, saving model to weights-improvement-01-2.7356.hdf5
8652/8652 [==============================] - 133s 15ms/step - loss: 2.7356 - accuracy: 0.2304
Epoch 2/40
8651/8652 [============================>.] - ETA: 0s - loss: 2.4502 - accuracy: 0.3004
Epoch 2: loss improved from 2.73558 to 2.45016, saving model to weights-improvement-02-2.4502.hdf5
8652/8652 [==============================] - 130s 15ms/step - loss: 2.4502 - accuracy: 0.3004
Epoch 3/40
8650/8652 [============================>.] - ETA: 0s - loss: 2.2866 - accuracy: 0.3434
Epoch 3: loss improved from 2.45016 to 2.28662, saving model to weights-improvement-03-2.2866.hdf5
8652/8652 [==============================] - 130s 15ms/step - loss: 2.2866 - accuracy: 0.3434
Epoch 4/40
8651/8652 [============================>.] - ETA: 0s - loss: 2.1831 - accuracy: 0.3714
Epoch 5: loss improved from 2.18313 to 2.11050, saving model to weights-improvement-05-2.1105.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 2.1105 - accuracy: 0.3919
Epoch 6/40
8649/8652 [============================>.] - ETA: 0s - loss: 2.0561 - accuracy: 0.4070
Epoch 6: loss improved from 2.11050 to 2.05615, saving model to weights-improvement-06-2.0562.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 2.0562 - accuracy: 0.4070
Epoch 7/40
8650/8652 [============================>.] - ETA: 0s - loss: 2.0115 - accuracy: 0.4189
Epoch 7: loss improved from 2.05615 to 2.01147, saving model to weights-improvement-07-2.0115.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 2.0115 - accuracy: 0.4189
Epoch 8/40
8650/8652 [============================>.] - ETA: 0s - loss: 1.9762 - accuracy: 0.4285
Epoch 8: loss improved from 2.01147 to 1.97620, saving model to weights-improvement-08-1.9762.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.9762 - accuracy: 0.4285
Epoch 9/40
8649/8652 [============================>.] - ETA: 0s - loss: 1.9471 - accuracy: 0.4362
Epoch 9: loss improved from 1.97620 to 1.94710, saving model to weights-improvement-09-1.9471.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.9471 - accuracy: 0.4362
Epoch 10/40
8651/8652 [============================>.] - ETA: 0s - loss: 1.9230 - accuracy: 0.4427
Epoch 10: loss improved from 1.94710 to 1.92295, saving model to weights-improvement-10-1.9230.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.9230 - accuracy: 0.4427
Epoch 11/40
8649/8652 [============================>.] - ETA: 0s - loss: 1.9013 - accuracy: 0.4482
Epoch 11: loss improved from 1.92295 to 1.90136, saving model to weights-improvement-11-1.9014.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.9014 - accuracy: 0.4482
Epoch 12/40
8652/8652 [==============================] - ETA: 0s - loss: 1.8822 - accuracy: 0.4531
Epoch 12: loss improved from 1.90136 to 1.88222, saving model to weights-improvement-12-1.8822.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.8822 - accuracy: 0.4531
Epoch 13/40
8652/8652 [==============================] - ETA: 0s - loss: 1.8652 - accuracy: 0.4578
Epoch 13: loss improved from 1.88222 to 1.86518, saving model to weights-improvement-13-1.8652.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.8652 - accuracy: 0.4578
Epoch 14/40
8652/8652 [==============================] - ETA: 0s - loss: 1.8503 - accuracy: 0.4617
Epoch 14: loss improved from 1.86518 to 1.85031, saving model to weights-improvement-14-1.8503.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.8503 - accuracy: 0.4617
```

```
Epoch 15/40
8651/8652 [============================>.] - ETA: 0s - loss: 1.8373 - accuracy: 0.4655
Epoch 15: loss improved from 1.85031 to 1.83727, saving model to weights-improvement-15-1.8373.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.8373 - accuracy: 0.4655
Epoch 16/40
8650/8652 [============================>.] - ETA: 0s - loss: 1.8247 - accuracy: 0.4685
Epoch 16: loss improved from 1.83727 to 1.82468, saving model to weights-improvement-16-1.8247.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.8247 - accuracy: 0.4685
Epoch 17/40
8649/8652 [============================>.] - ETA: 0s - loss: 1.8136 - accuracy: 0.4712
Epoch 17: loss improved from 1.82468 to 1.81362, saving model to weights-improvement-17-1.8136.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.8136 - accuracy: 0.4712
Epoch 18/40
8650/8652 [============================>.] - ETA: 0s - loss: 1.8039 - accuracy: 0.4740
Epoch 18: loss improved from 1.81362 to 1.80392, saving model to weights-improvement-18-1.8039.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.8039 - accuracy: 0.4740
Epoch 19/40
8649/8652 [============================>.] - ETA: 0s - loss: 1.7947 - accuracy: 0.4768
Epoch 19: loss improved from 1.80392 to 1.79467, saving model to weights-improvement-19-1.7947.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.7947 - accuracy: 0.4768
Epoch 20/40
8649/8652 [============================>.] - ETA: 0s - loss: 1.7861 - accuracy: 0.4788
Epoch 20: loss improved from 1.79467 to 1.78615, saving model to weights-improvement-20-1.7861.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.7861 - accuracy: 0.4788

Epoch 21/40
8652/8652 [==============================] - ETA: 0s - loss: 1.7779 - accuracy: 0.4807
Epoch 21: loss improved from 1.78615 to 1.77785, saving model to weights-improvement-21-1.7779.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.7779 - accuracy: 0.4807
Epoch 22/40
8651/8652 [============================>.] - ETA: 0s - loss: 1.7694 - accuracy: 0.4835
Epoch 22: loss improved from 1.77785 to 1.76937, saving model to weights-improvement-22-1.7694.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.7694 - accuracy: 0.4835
Epoch 23/40
8651/8652 [============================>.] - ETA: 0s - loss: 1.7625 - accuracy: 0.4842
Epoch 23: loss improved from 1.76937 to 1.76254, saving model to weights-improvement-23-1.7625.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.7625 - accuracy: 0.4842
Epoch 24/40
8651/8652 [============================>.] - ETA: 0s - loss: 1.7556 - accuracy: 0.4867
Epoch 24: loss improved from 1.76254 to 1.75562, saving model to weights-improvement-24-1.7556.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.7556 - accuracy: 0.4867
Epoch 25/40
8649/8652 [============================>.] - ETA: 0s - loss: 1.7484 - accuracy: 0.4891
Epoch 25: loss improved from 1.75562 to 1.74839, saving model to weights-improvement-25-1.7484.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 1.7484 - accuracy: 0.4891
Epoch 26/40
8650/8652 [============================>.] - ETA: 0s - loss: 1.7433 - accuracy: 0.4896
Epoch 26: loss improved from 1.74839 to 1.74327, saving model to weights-improvement-26-1.7433.hdf5
```

```
8652/8652 [==============================] - 128s 15ms/step - loss: 1.7433 - accuracy: 0.4896
Epoch 27/40
8651/8652 [============================>.] - ETA: 0s - loss: 1.7380 - accuracy: 0.4913
Epoch 27: loss improved from 1.74327 to 1.73799, saving model to weights-improvement-27-1.7380.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.7380 - accuracy: 0.4913
Epoch 28/40
8652/8652 [==============================] - ETA: 0s - loss: 1.7318 - accuracy: 0.4928
Epoch 28: loss improved from 1.73799 to 1.73181, saving model to weights-improvement-28-1.7318.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.7318 - accuracy: 0.4928
Epoch 29/40
8652/8652 [==============================] - ETA: 0s - loss: 1.7276 - accuracy: 0.4940
Epoch 29: loss improved from 1.73181 to 1.72761, saving model to weights-improvement-29-1.7276.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.7276 - accuracy: 0.4940
Epoch 30/40
8652/8652 [==============================] - ETA: 0s - loss: 1.7220 - accuracy: 0.4951
Epoch 30: loss improved from 1.72761 to 1.72201, saving model to weights-improvement-30-1.7220.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.7220 - accuracy: 0.4951
Epoch 31/40
8649/8652 [============================>.] - ETA: 0s - loss: 1.7160 - accuracy: 0.4973
Epoch 31: loss improved from 1.72201 to 1.71603, saving model to weights-improvement-31-1.7160.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.7160 - accuracy: 0.4973
Epoch 32/40
8650/8652 [============================>.] - ETA: 0s - loss: 1.7131 - accuracy: 0.4981
```

```
Epoch 33/40
8649/8652 [============================>.] - ETA: 0s - loss: 1.7098 - accuracy: 0.4989
Epoch 33: loss improved from 1.71308 to 1.70980, saving model to weights-improvement-33-1.7098.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.7098 - accuracy: 0.4989
Epoch 34/40
8652/8652 [==============================] - ETA: 0s - loss: 1.7044 - accuracy: 0.5000
Epoch 34: loss improved from 1.70980 to 1.70438, saving model to weights-improvement-34-1.7044.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.7044 - accuracy: 0.5000
Epoch 35/40
8652/8652 [==============================] - ETA: 0s - loss: 1.7012 - accuracy: 0.5010
Epoch 35: loss improved from 1.70438 to 1.70121, saving model to weights-improvement-35-1.7012.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.7012 - accuracy: 0.5010
Epoch 36/40
8649/8652 [============================>.] - ETA: 0s - loss: 1.6973 - accuracy: 0.5017
Epoch 36: loss improved from 1.70121 to 1.69733, saving model to weights-improvement-36-1.6973.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.6973 - accuracy: 0.5017
Epoch 37/40
8651/8652 [============================>.] - ETA: 0s - loss: 1.6943 - accuracy: 0.5024
Epoch 37: loss improved from 1.69733 to 1.69428, saving model to weights-improvement-37-1.6943.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.6943 - accuracy: 0.5024
Epoch 38/40
8650/8652 [============================>.] - ETA: 0s - loss: 1.6898 - accuracy: 0.5040
Epoch 38: loss improved from 1.69428 to 1.68982, saving model to weights-improvement-38-1.6898.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.6898 - accuracy: 0.5040
Epoch 39/40
8649/8652 [============================>.] - ETA: 0s - loss: 1.6862 - accuracy: 0.5051
Epoch 39: loss improved from 1.68982 to 1.68615, saving model to weights-improvement-39-1.6861.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.6861 - accuracy: 0.5052
Epoch 40/40
8652/8652 [==============================] - ETA: 0s - loss: 1.6836 - accuracy: 0.5054
Epoch 40: loss improved from 1.68615 to 1.68356, saving model to weights-improvement-40-1.6836.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.6836 - accuracy: 0.5054
```

The Results :

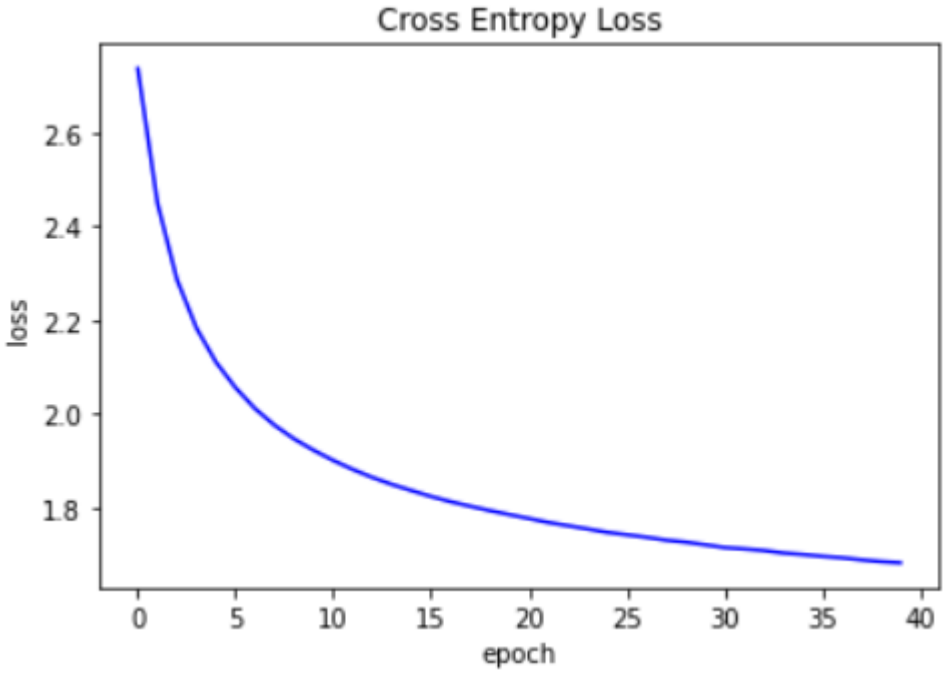

Figure67. cross-entropy loss during the epochs

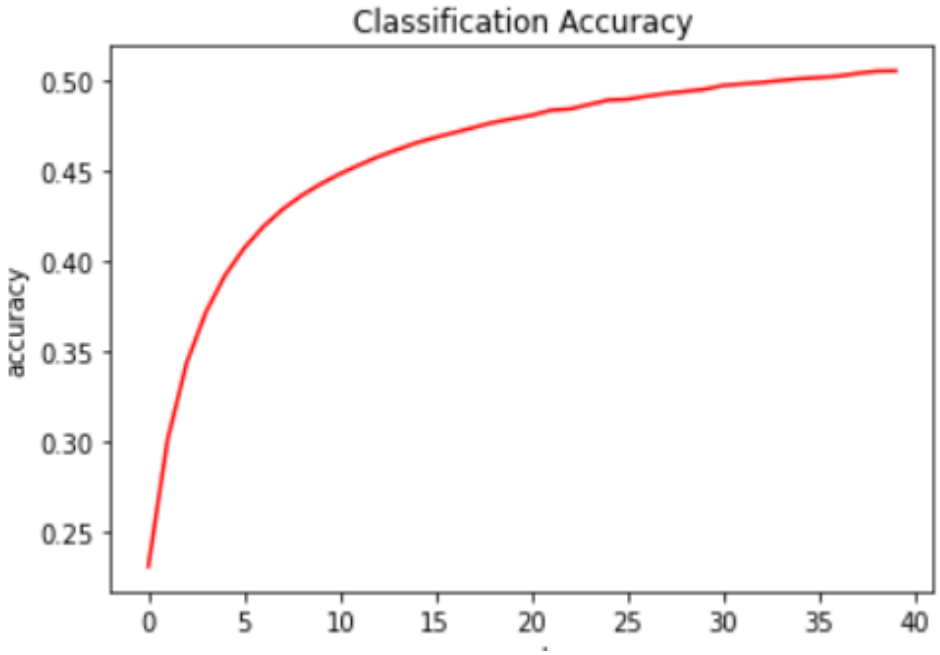

Figure68. Classification accuracy during the epochs

TEXT PREDICTION RESULTS:

Here is some output from the stateless recurrent neural network that takes predicts characters one at a time, using a seed text of 100 characters (with some autocorrect applied):

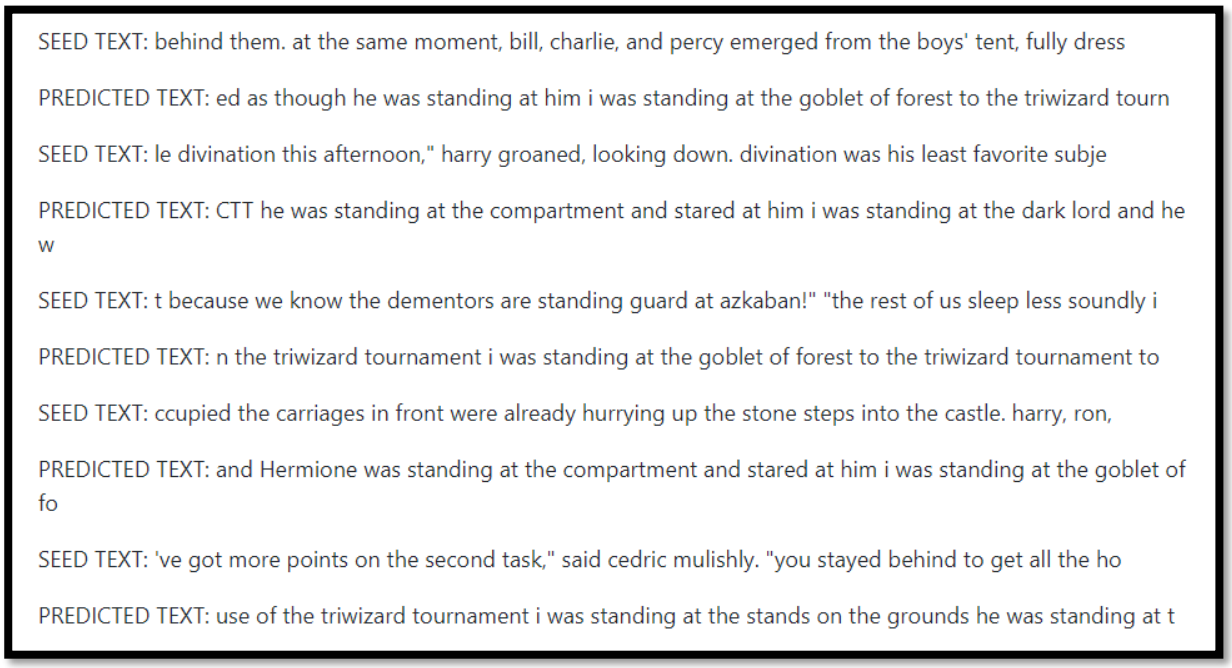

Figure69. Text Prediction Results

Why pre-processing can improve the model performance?

1) we need to load the ASCII text for the book into memory and convert all of the characters to lowercase **to reduce the vocabulary that the network must learn.**

2) when that the book is loaded, we must **prepare the data for** modeling **by the neural network**. We cannot model the characters directly, **instead we must convert the characters to integers.** We can do this easily by first creating a set of all of the distinct characters in the book, **then creating a map of each character to a unique integer.**

3) Next, we need to **rescale the integers to the range 0-to-1** to make the patterns **easier to learn by the LSTM network** that uses the sigmoid activation function by default.

4) Finally, we need to convert the output patterns (single characters converted to integers) into a **one hot encoding**. This is so that we can configure the network to predict the probability of each of the 54 different characters in the vocabulary (an easier representation) rather than trying **to force it to predict precisely the next character**. Each y value is converted into a sparse vector with a length of 54, full of zeros except with a 1 in the column for the letter (integer) that the pattern represents.

3.2

In this section, we are going to use another loss-function in our optimizer and investigate is it a good deal or not.

A loss function is going to serve as a measurement of how far our current set of predictions are from the corresponding true values.

First-one : Mean-Squared-Error

Model-evaluation:

accuracy curve and **loss curve** during epochs have been attached:

```
Total Characters:  1107542
Total Vocab:  54
Total Patterns:  1107442
Epoch 1/20
8652/8652 [==============================] - ETA: 0s - loss: 0.0157 - accuracy: 0.2780
Epoch 1: loss improved from inf to 0.01575, saving model to weights-improvement-01-0.0157.hdf5
8652/8652 [==============================] - 288s 32ms/step - loss: 0.0157 - accuracy: 0.2780
Epoch 2/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0137 - accuracy: 0.3953
Epoch 2: loss improved from 0.01575 to 0.01371, saving model to weights-improvement-02-0.0137.hdf5
8652/8652 [==============================] - 281s 33ms/step - loss: 0.0137 - accuracy: 0.3953
Epoch 3/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0129 - accuracy: 0.4392
Epoch 3: loss improved from 0.01371 to 0.01288, saving model to weights-improvement-03-0.0129.hdf5
8652/8652 [==============================] - 282s 33ms/step - loss: 0.0129 - accuracy: 0.4392
Epoch 4/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0123 - accuracy: 0.4658
Epoch 4: loss improved from 0.01288 to 0.01235, saving model to weights-improvement-04-0.0123.hdf5
8652/8652 [==============================] - 283s 33ms/step - loss: 0.0123 - accuracy: 0.4658
Epoch 5/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0120 - accuracy: 0.4828
Epoch 5: loss improved from 0.01235 to 0.01200, saving model to weights-improvement-05-0.0120.hdf5
8652/8652 [==============================] - 286s 33ms/step - loss: 0.0120 - accuracy: 0.4828
Epoch 6/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0117 - accuracy: 0.4953
Epoch 6: loss improved from 0.01200 to 0.01174, saving model to weights-improvement-06-0.0117.hdf5
8652/8652 [==============================] - 285s 33ms/step - loss: 0.0117 - accuracy: 0.4953
Epoch 7/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0115 - accuracy: 0.5057
Epoch 7: loss improved from 0.01174 to 0.01153, saving model to weights-improvement-07-0.0115.hdf5
8652/8652 [==============================] - 285s 33ms/step - loss: 0.0115 - accuracy: 0.5057
```

```
Epoch 7: loss improved from 0.01174 to 0.01153, saving model to weights-improvement-07-0.0115.hdf5
8652/8652 [==============================] - 285s 33ms/step - loss: 0.0115 - accuracy: 0.5057
Epoch 8/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0114 - accuracy: 0.5142
Epoch 8: loss improved from 0.01153 to 0.01135, saving model to weights-improvement-08-0.0114.hdf5
8652/8652 [==============================] - 284s 33ms/step - loss: 0.0114 - accuracy: 0.5142
Epoch 9/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0112 - accuracy: 0.5207
Epoch 9: loss improved from 0.01135 to 0.01122, saving model to weights-improvement-09-0.0112.hdf5
8652/8652 [==============================] - 281s 33ms/step - loss: 0.0112 - accuracy: 0.5207
Epoch 10/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0111 - accuracy: 0.5262
Epoch 10: loss improved from 0.01122 to 0.01111, saving model to weights-improvement-10-0.0111.hdf5
8652/8652 [==============================] - 282s 33ms/step - loss: 0.0111 - accuracy: 0.5262
Epoch 11/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0110 - accuracy: 0.5318
Epoch 11: loss improved from 0.01111 to 0.01100, saving model to weights-improvement-11-0.0110.hdf5
8652/8652 [==============================] - 282s 33ms/step - loss: 0.0110 - accuracy: 0.5318
Epoch 12/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0109 - accuracy: 0.5354
Epoch 12: loss improved from 0.01100 to 0.01092, saving model to weights-improvement-12-0.0109.hdf5
8652/8652 [==============================] - 282s 33ms/step - loss: 0.0109 - accuracy: 0.5354
Epoch 13/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0108 - accuracy: 0.5391
Epoch 13: loss improved from 0.01092 to 0.01083, saving model to weights-improvement-13-0.0108.hdf5
8652/8652 [==============================] - 282s 33ms/step - loss: 0.0108 - accuracy: 0.5391

Epoch 14/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0108 - accuracy: 0.5434
Epoch 14: loss improved from 0.01083 to 0.01076, saving model to weights-improvement-14-0.0108.hdf5
8652/8652 [==============================] - 282s 33ms/step - loss: 0.0108 - accuracy: 0.5434
Epoch 15/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0107 - accuracy: 0.5460
Epoch 15: loss improved from 0.01076 to 0.01070, saving model to weights-improvement-15-0.0107.hdf5
8652/8652 [==============================] - 281s 33ms/step - loss: 0.0107 - accuracy: 0.5460
Epoch 16/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0106 - accuracy: 0.5488
Epoch 16: loss improved from 0.01070 to 0.01064, saving model to weights-improvement-16-0.0106.hdf5
8652/8652 [==============================] - 282s 33ms/step - loss: 0.0106 - accuracy: 0.5488
Epoch 17/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0106 - accuracy: 0.5513
Epoch 17: loss improved from 0.01064 to 0.01059, saving model to weights-improvement-17-0.0106.hdf5
8652/8652 [==============================] - 283s 33ms/step - loss: 0.0106 - accuracy: 0.5513
Epoch 18/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0105 - accuracy: 0.5538
Epoch 18: loss improved from 0.01059 to 0.01054, saving model to weights-improvement-18-0.0105.hdf5
8652/8652 [==============================] - 282s 33ms/step - loss: 0.0105 - accuracy: 0.5538
Epoch 19/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0105 - accuracy: 0.5556
Epoch 19: loss improved from 0.01054 to 0.01050, saving model to weights-improvement-19-0.0105.hdf5
8652/8652 [==============================] - 282s 33ms/step - loss: 0.0105 - accuracy: 0.5556
Epoch 20/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0105 - accuracy: 0.5579
Epoch 20: loss improved from 0.01050 to 0.01045, saving model to weights-improvement-20-0.0105.hdf5
8652/8652 [==============================] - 284s 33ms/step - loss: 0.0105 - accuracy: 0.5579
```

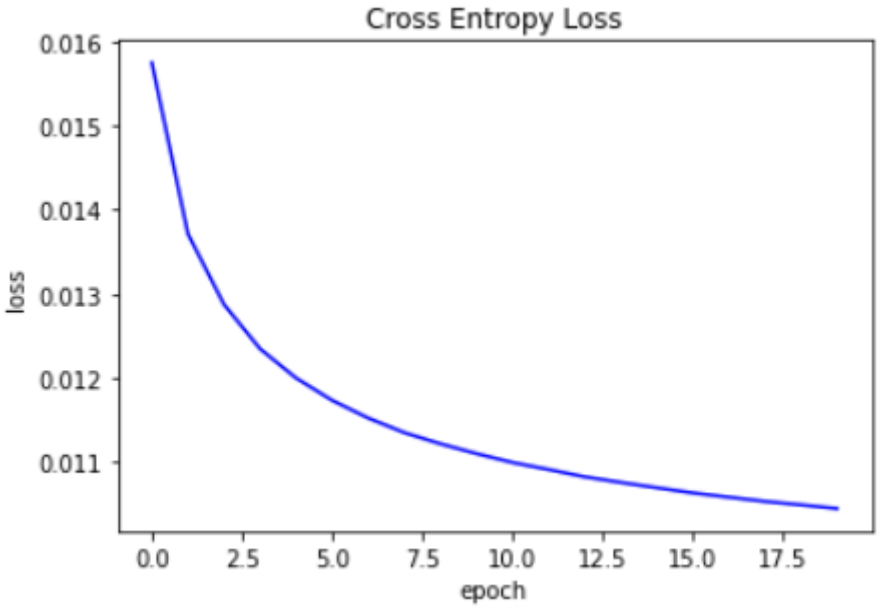

Figure70. cross-entropy loss during the epochs

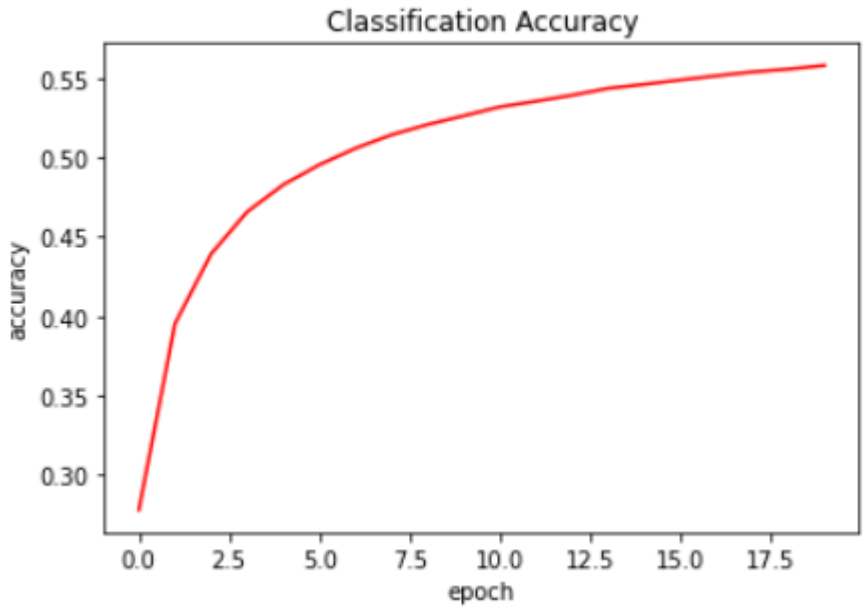

Figure71. Classification accuracy during the epochs

Second-one : Mean-Absolute-Error

Model-evaluation:

accuracy curve and **loss curve** during epochs have been attached:

```
Total Characters:  1107542
Total Vocab:  54
Total Patterns:  1107442
Epoch 1/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 1: loss improved from inf to 0.03047, saving model to weights-improvement-01-0.0305.hdf5
8652/8652 [==============================] - 283s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 2/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 2: loss improved from 0.03047 to 0.03046, saving model to weights-improvement-02-0.0305.hdf5
8652/8652 [==============================] - 275s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 3/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 3: loss did not improve from 0.03046
8652/8652 [==============================] - 276s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 4/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 4: loss did not improve from 0.03046
8652/8652 [==============================] - 275s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 5/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 5: loss improved from 0.03046 to 0.03046, saving model to weights-improvement-05-0.0305.hdf5
8652/8652 [==============================] - 276s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 6/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 6: loss did not improve from 0.03046
8652/8652 [==============================] - 275s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 7/20
8651/8652 [============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 7: loss did not improve from 0.03046
```

```
8652/8652 [==============================] - 276s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 8/20
8651/8652 [==============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 8: loss did not improve from 0.03046
8652/8652 [==============================] - 276s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 9/20
8651/8652 [==============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 9: loss did not improve from 0.03046
8652/8652 [==============================] - 277s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 10/20
8651/8652 [==============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 10: loss did not improve from 0.03046
8652/8652 [==============================] - 276s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 11/20
8651/8652 [==============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 11: loss did not improve from 0.03046
8652/8652 [==============================] - 276s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 12/20
8651/8652 [==============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 12: loss did not improve from 0.03046
8652/8652 [==============================] - 275s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 13/20
8651/8652 [==============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 13: loss did not improve from 0.03046
8652/8652 [==============================] - 277s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 14/20
8651/8652 [==============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 14: loss did not improve from 0.03046
8652/8652 [==============================] - 276s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 15/20
8651/8652 [==============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 15: loss did not improve from 0.03046
8652/8652 [==============================] - 276s 32ms/step - loss: 0.0305 - accuracy: 0.1774

Epoch 15: loss did not improve from 0.03046
8652/8652 [==============================] - 276s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 16/20
8651/8652 [==============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 16: loss did not improve from 0.03046
8652/8652 [==============================] - 276s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 17/20
8651/8652 [==============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 17: loss did not improve from 0.03046
8652/8652 [==============================] - 277s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 18/20
8651/8652 [==============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 18: loss did not improve from 0.03046
8652/8652 [==============================] - 276s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 19/20
8651/8652 [==============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 19: loss did not improve from 0.03046
8652/8652 [==============================] - 277s 32ms/step - loss: 0.0305 - accuracy: 0.1774
Epoch 20/20
8651/8652 [==============================>.] - ETA: 0s - loss: 0.0305 - accuracy: 0.1774
Epoch 20: loss improved from 0.03046 to 0.03046, saving model to weights-improvement-20-0.0305.hdf5
8652/8652 [==============================] - 277s 32ms/step - loss: 0.0305 - accuracy: 0.1774
```

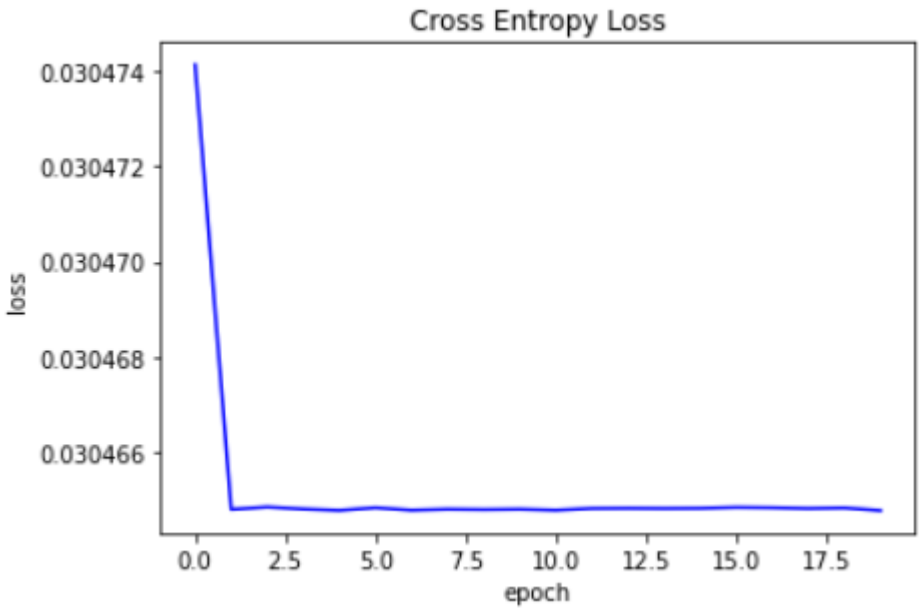

Figure72. cross-entropy loss during the epochs

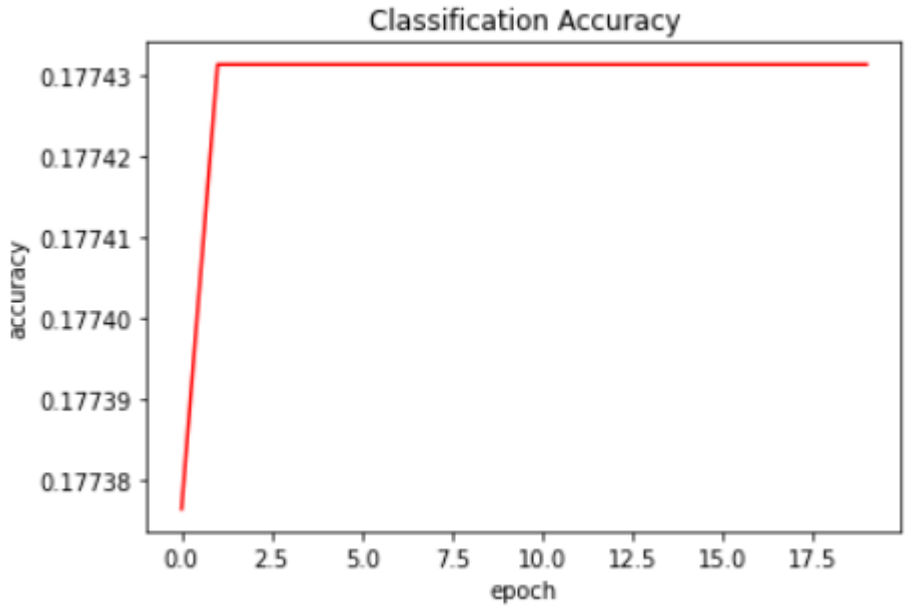

Figure73. Classification accuracy during the epochs

As you can see the results of MSE in comparison with MAE is pretty much better and also the curves which are belongs to the MSE are most smooth in other words has a better convergence rate.

But in both cases, we are dealing with a lower performance in comparison to cross-entropy loss which is highly better in all cases.

3.3

in this section we are going to change two of metrics and investigate whether the results tend to be better or not.

NUMBER OF EPOCHS # 20 :

TRAINING PROCESS:

```
Total Characters:  1107542
Total Vocab:  54
Total Patterns:  1107442
Epoch 1/20
8650/8652 [============================>.] - ETA: 0s - loss: 2.7279 - accuracy: 0.2328
Epoch 1: loss improved from inf to 2.72788, saving model to weights-improvement-01-2.7279.hdf5
8652/8652 [==============================] - 131s 14ms/step - loss: 2.7279 - accuracy: 0.2329
Epoch 2/20
8650/8652 [============================>.] - ETA: 0s - loss: 2.4492 - accuracy: 0.3017
Epoch 2: loss improved from 2.72788 to 2.44918, saving model to weights-improvement-02-2.4492.hdf5
8652/8652 [==============================] - 128s 15ms/step - loss: 2.4492 - accuracy: 0.3017
Epoch 3/20
8652/8652 [==============================] - ETA: 0s - loss: 2.2917 - accuracy: 0.3430
Epoch 3: loss improved from 2.44918 to 2.29173, saving model to weights-improvement-03-2.2917.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 2.2917 - accuracy: 0.3430
Epoch 4/20
8649/8652 [============================>.] - ETA: 0s - loss: 2.1936 - accuracy: 0.3695
Epoch 4: loss improved from 2.29173 to 2.19358, saving model to weights-improvement-04-2.1936.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 2.1936 - accuracy: 0.3695
Epoch 5/20
8652/8652 [==============================] - ETA: 0s - loss: 2.1242 - accuracy: 0.3879
Epoch 5: loss improved from 2.19358 to 2.12422, saving model to weights-improvement-05-2.1242.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 2.1242 - accuracy: 0.3879
Epoch 6/20
8649/8652 [============================>.] - ETA: 0s - loss: 2.0720 - accuracy: 0.4021
Epoch 6: loss improved from 2.12422 to 2.07192, saving model to weights-improvement-06-2.0719.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 2.0719 - accuracy: 0.4021
Epoch 7/20
8651/8652 [============================>.] - ETA: 0s - loss: 2.0300 - accuracy: 0.4133
Epoch 7: loss improved from 2.07192 to 2.03000, saving model to weights-improvement-07-2.0300.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 2.0300 - accuracy: 0.4133
Epoch 8/20
8651/8652 [============================>.] - ETA: 0s - loss: 1.9962 - accuracy: 0.4224
Epoch 8: loss improved from 2.03000 to 1.99621, saving model to weights-improvement-08-1.9962.hdf5
8652/8652 [==============================] - 130s 15ms/step - loss: 1.9962 - accuracy: 0.4224
Epoch 9/20
8652/8652 [==============================] - ETA: 0s - loss: 1.9671 - accuracy: 0.4308
Epoch 9: loss improved from 1.99621 to 1.96713, saving model to weights-improvement-09-1.9671.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.9671 - accuracy: 0.4308
Epoch 10/20
8650/8652 [============================>.] - ETA: 0s - loss: 1.9420 - accuracy: 0.4366
Epoch 10: loss improved from 1.96713 to 1.94198, saving model to weights-improvement-10-1.9420.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.9420 - accuracy: 0.4366
Epoch 11/20
8652/8652 [==============================] - ETA: 0s - loss: 1.9213 - accuracy: 0.4425
Epoch 11: loss improved from 1.94198 to 1.92135, saving model to weights-improvement-11-1.9213.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.9213 - accuracy: 0.4425
Epoch 12/20
8651/8652 [============================>.] - ETA: 0s - loss: 1.9017 - accuracy: 0.4478
Epoch 12: loss improved from 1.92135 to 1.90170, saving model to weights-improvement-12-1.9017.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.9017 - accuracy: 0.4478
```

```
Epoch 13/20
8650/8652 [============================>.] - ETA: 0s - loss: 1.8853 - accuracy: 0.4522
Epoch 13: loss improved from 1.90170 to 1.88533, saving model to weights-improvement-13-1.8853.hdf5
8652/8652 [==============================] - 130s 15ms/step - loss: 1.8853 - accuracy: 0.4522
Epoch 14/20
8651/8652 [============================>.] - ETA: 0s - loss: 1.8689 - accuracy: 0.4563
Epoch 14: loss improved from 1.88533 to 1.86894, saving model to weights-improvement-14-1.8689.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.8689 - accuracy: 0.4563
Epoch 15/20
8650/8652 [============================>.] - ETA: 0s - loss: 1.8555 - accuracy: 0.4602
Epoch 15: loss improved from 1.86894 to 1.85545, saving model to weights-improvement-15-1.8554.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.8554 - accuracy: 0.4602
Epoch 16/20
8650/8652 [============================>.] - ETA: 0s - loss: 1.8437 - accuracy: 0.4631
Epoch 16: loss improved from 1.85545 to 1.84365, saving model to weights-improvement-16-1.8437.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.8437 - accuracy: 0.4631
Epoch 17/20
8651/8652 [============================>.] - ETA: 0s - loss: 1.8326 - accuracy: 0.4658
Epoch 17: loss improved from 1.84365 to 1.83267, saving model to weights-improvement-17-1.8327.hdf5
8652/8652 [==============================] - 129s 15ms/step - loss: 1.8327 - accuracy: 0.4658
Epoch 18/20
8649/8652 [============================>.] - ETA: 0s - loss: 1.8212 - accuracy: 0.4689
Epoch 18: loss improved from 1.83267 to 1.82119, saving model to weights-improvement-18-1.8212.hdf5
8652/8652 [==============================] - 132s 15ms/step - loss: 1.8212 - accuracy: 0.4689
Epoch 19/20
8651/8652 [============================>.] - ETA: 0s - loss: 1.8119 - accuracy: 0.4709
Epoch 19: loss improved from 1.82119 to 1.81191, saving model to weights-improvement-19-1.8119.hdf5
8652/8652 [==============================] - 131s 15ms/step - loss: 1.8119 - accuracy: 0.4709
Epoch 20/20
8649/8652 [============================>.] - ETA: 0s - loss: 1.8027 - accuracy: 0.4735
Epoch 20: loss improved from 1.81191 to 1.80268, saving model to weights-improvement-20-1.8027.hdf5
8652/8652 [==============================] - 130s 15ms/step - loss: 1.8027 - accuracy: 0.4735
```

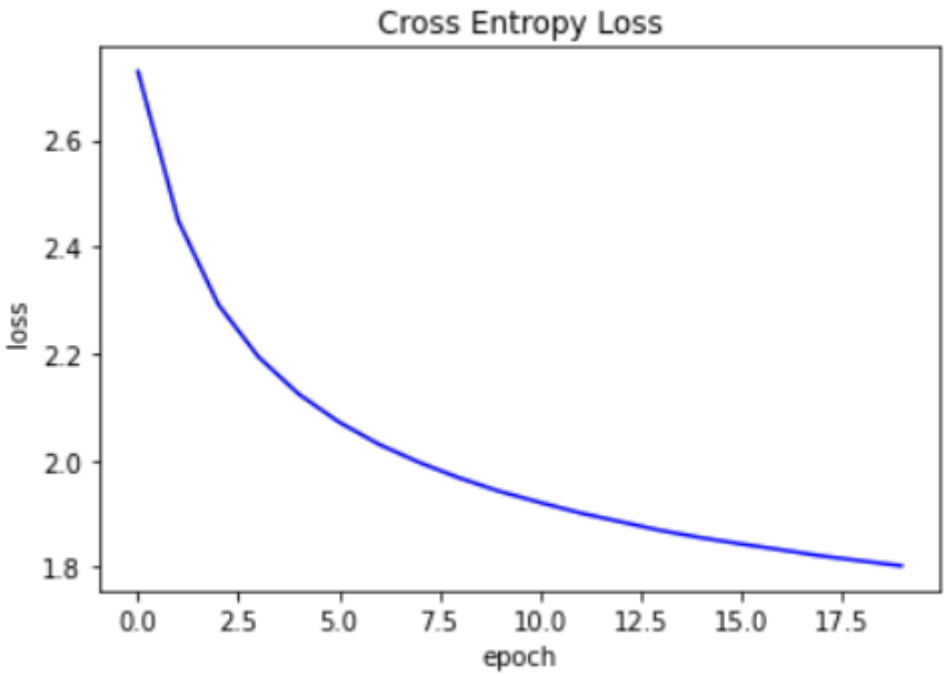

Figure74. cross-entropy loss during the epochs

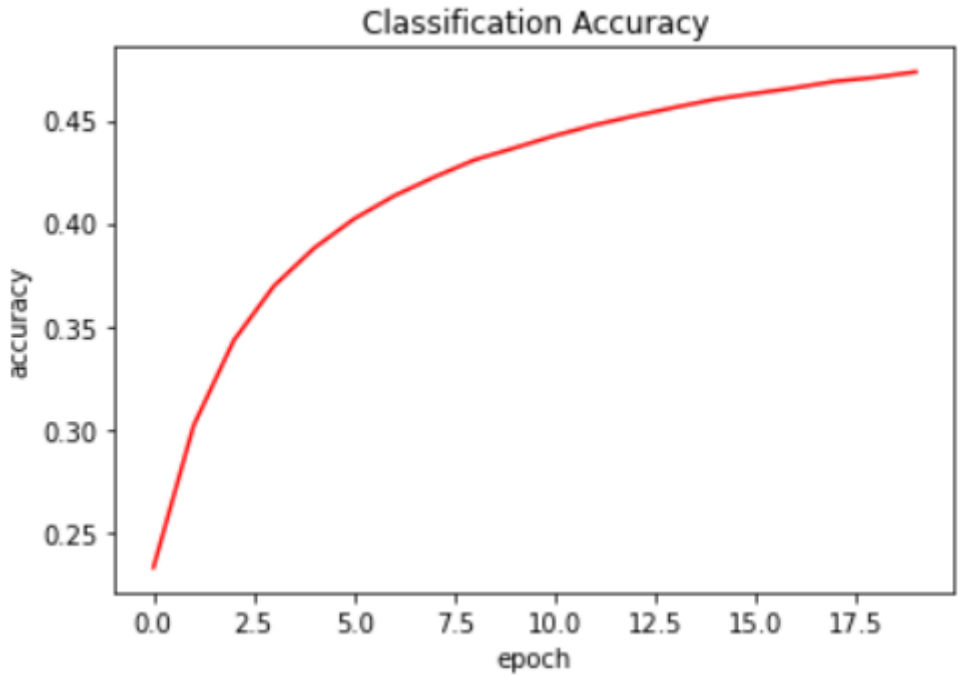

Figure75. Classification accuracy during the epochs

BATCH-SIZE # 64 :

TRAINING PROCESS:

```
Total Characters:  1107542
Total Vocab:  54
Total Patterns:  1107442
Epoch 1/20
17304/17304 [==============================] - ETA: 0s - loss: 2.3529 - accuracy: 0.3289
Epoch 1: loss improved from inf to 2.35287, saving model to weights-improvement-01-2.3529.hdf5
17304/17304 [==============================] - 338s 19ms/step - loss: 2.3529 - accuracy: 0.3289
Epoch 2/20
17304/17304 [==============================] - ETA: 0s - loss: 1.9421 - accuracy: 0.4373
Epoch 2: loss improved from 2.35287 to 1.94207, saving model to weights-improvement-02-1.9421.hdf5
17304/17304 [==============================] - 341s 20ms/step - loss: 1.9421 - accuracy: 0.4373
Epoch 3/20
17303/17304 [============================>.] - ETA: 0s - loss: 1.8051 - accuracy: 0.4737
Epoch 3: loss improved from 1.94207 to 1.80515, saving model to weights-improvement-03-1.8051.hdf5
17304/17304 [==============================] - 340s 20ms/step - loss: 1.8051 - accuracy: 0.4737
Epoch 4/20
17302/17304 [============================>.] - ETA: 0s - loss: 1.7224 - accuracy: 0.4955
Epoch 4: loss improved from 1.80515 to 1.72237, saving model to weights-improvement-04-1.7224.hdf5
17304/17304 [==============================] - 340s 20ms/step - loss: 1.7224 - accuracy: 0.4955
Epoch 5/20
17304/17304 [==============================] - ETA: 0s - loss: 1.6677 - accuracy: 0.5103
Epoch 5: loss improved from 1.72237 to 1.66774, saving model to weights-improvement-05-1.6677.hdf5
```

```
17304/17304 [==============================] - 340s 20ms/step - loss: 1.6677 - accuracy: 0.5103
Epoch 6/20
17302/17304 [==============================>.] - ETA: 0s - loss: 1.6268 - accuracy: 0.5209
Epoch 6: loss improved from 1.66774 to 1.62675, saving model to weights-improvement-06-1.6268.hdf5
17304/17304 [==============================] - 340s 20ms/step - loss: 1.6268 - accuracy: 0.5209
Epoch 7/20
17303/17304 [==============================>.] - ETA: 0s - loss: 1.5946 - accuracy: 0.5296
Epoch 7: loss improved from 1.62675 to 1.59461, saving model to weights-improvement-07-1.5946.hdf5
17304/17304 [==============================] - 341s 20ms/step - loss: 1.5946 - accuracy: 0.5296
Epoch 8/20
17303/17304 [==============================>.] - ETA: 0s - loss: 1.5700 - accuracy: 0.5356
Epoch 8: loss improved from 1.59461 to 1.56999, saving model to weights-improvement-08-1.5700.hdf5
17304/17304 [==============================] - 341s 20ms/step - loss: 1.5700 - accuracy: 0.5356
Epoch 9/20
17303/17304 [==============================>.] - ETA: 0s - loss: 1.5478 - accuracy: 0.5420
Epoch 9: loss improved from 1.56999 to 1.54780, saving model to weights-improvement-09-1.5478.hdf5
17304/17304 [==============================] - 340s 20ms/step - loss: 1.5478 - accuracy: 0.5420

Epoch 10/20
17302/17304 [==============================>.] - ETA: 0s - loss: 1.5305 - accuracy: 0.5456
Epoch 10: loss improved from 1.54780 to 1.53057, saving model to weights-improvement-10-1.5306.hdf5
17304/17304 [==============================] - 341s 20ms/step - loss: 1.5306 - accuracy: 0.5456
Epoch 11/20
17302/17304 [==============================>.] - ETA: 0s - loss: 1.5161 - accuracy: 0.5497
Epoch 11: loss improved from 1.53057 to 1.51610, saving model to weights-improvement-11-1.5161.hdf5
17304/17304 [==============================] - 341s 20ms/step - loss: 1.5161 - accuracy: 0.5497
Epoch 12/20
17303/17304 [==============================>.] - ETA: 0s - loss: 1.5035 - accuracy: 0.5526
Epoch 12: loss improved from 1.51610 to 1.50351, saving model to weights-improvement-12-1.5035.hdf5
17304/17304 [==============================] - 340s 20ms/step - loss: 1.5035 - accuracy: 0.5526
Epoch 13/20
17303/17304 [==============================>.] - ETA: 0s - loss: 1.4909 - accuracy: 0.5556
Epoch 13: loss improved from 1.50351 to 1.49084, saving model to weights-improvement-13-1.4908.hdf5
17304/17304 [==============================] - 340s 20ms/step - loss: 1.4908 - accuracy: 0.5556
Epoch 14/20
17303/17304 [==============================>.] - ETA: 0s - loss: 1.4809 - accuracy: 0.5589
Epoch 14: loss improved from 1.49084 to 1.48088, saving model to weights-improvement-14-1.4809.hdf5
17304/17304 [==============================] - 341s 20ms/step - loss: 1.4809 - accuracy: 0.5589
Epoch 15/20
17304/17304 [==============================] - ETA: 0s - loss: 1.4706 - accuracy: 0.5614
Epoch 15: loss improved from 1.48088 to 1.47056, saving model to weights-improvement-15-1.4706.hdf5
17304/17304 [==============================] - 339s 20ms/step - loss: 1.4706 - accuracy: 0.5614
Epoch 16/20
17302/17304 [==============================>.] - ETA: 0s - loss: 1.4623 - accuracy: 0.5634
Epoch 16: loss improved from 1.47056 to 1.46225, saving model to weights-improvement-16-1.4623.hdf5
17304/17304 [==============================] - 340s 20ms/step - loss: 1.4623 - accuracy: 0.5634
Epoch 17/20
17304/17304 [==============================] - ETA: 0s - loss: 1.4553 - accuracy: 0.5649
Epoch 17: loss improved from 1.46225 to 1.45534, saving model to weights-improvement-17-1.4553.hdf5
17304/17304 [==============================] - 340s 20ms/step - loss: 1.4553 - accuracy: 0.5649
Epoch 18/20
17303/17304 [==============================>.] - ETA: 0s - loss: 1.4476 - accuracy: 0.5672

Epoch 19/20
17302/17304 [==============================>.] - ETA: 0s - loss: 1.4410 - accuracy: 0.5689
Epoch 19: loss improved from 1.44757 to 1.44098, saving model to weights-improvement-19-1.4410.hdf5
17304/17304 [==============================] - 341s 20ms/step - loss: 1.4410 - accuracy: 0.5689
Epoch 20/20
17303/17304 [==============================>.] - ETA: 0s - loss: 1.4351 - accuracy: 0.5698
Epoch 20: loss improved from 1.44098 to 1.43511, saving model to weights-improvement-20-1.4351.hdf5
17304/17304 [==============================] - 341s 20ms/step - loss: 1.4351 - accuracy: 0.5698
```

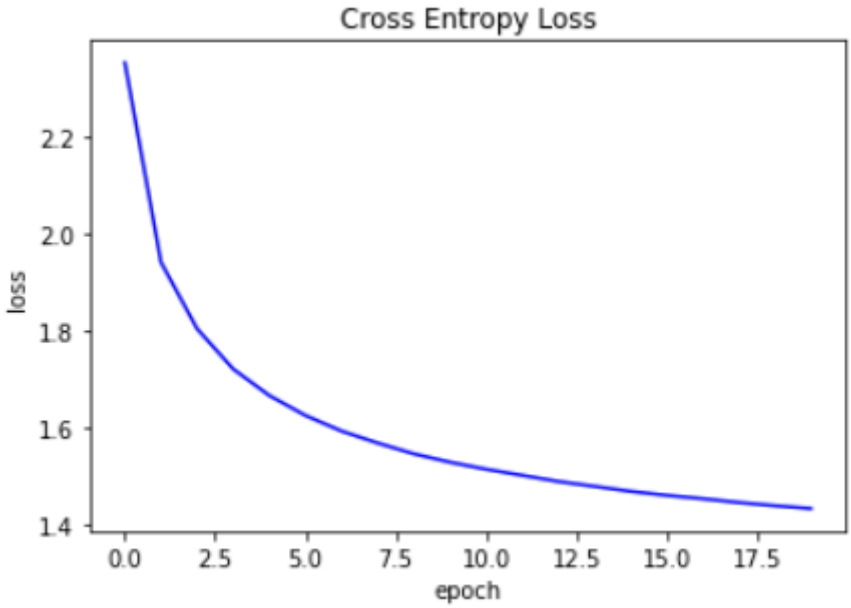

Figure76. cross-entropy loss during the epochs

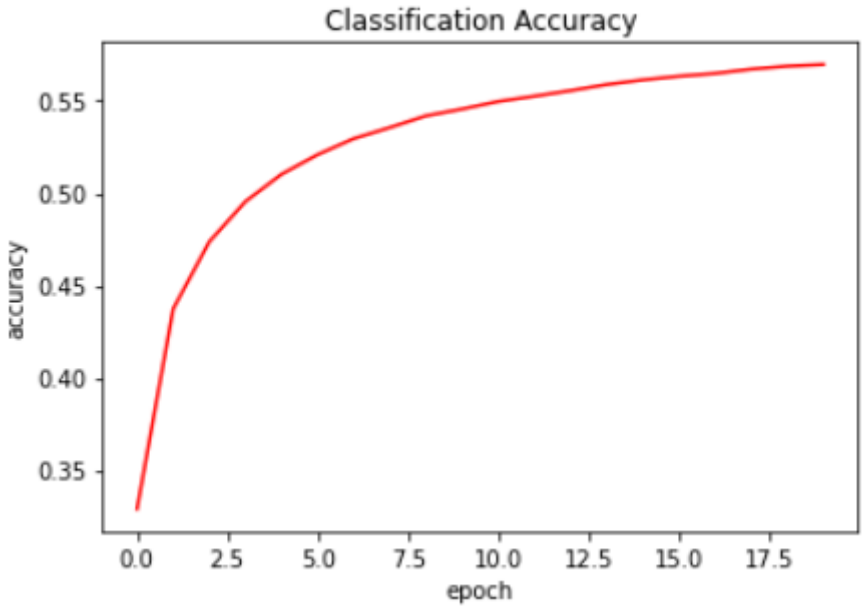

Figure77. Classification accuracy during the epochs

As you can see the best parameters for fitting the model based on our dataset are :

Batch-size=128

Number-of-epochs: 100 (to have a really good text-generator)

PLEASE NOTE THAT THE WEIGHTS OF BEST TRY HAS BEEN ATTACHED IN RELATED DIRECTORY.

(WEIGHTS-IMPROVEMENT-97-1.2709.HDF5)

3.4

HOW MEMORY OF NEURAL NETS IN RNN CAN BE HELPFUL IN MODEL PERFORMANCE?

The main difference between the functioning of neural networks and the biological neural network is memory. While both the human brain and neural networks have the ability to read and write from the memory available, the brain can create/store the memory as well

Memory in neural networks is required to store input data, weight parameters and activations as an input propagates through the network. In training, activations from a forward pass must be retained until they can be used to calculate the error gradients in the backwards pass.

NNs effectively have an internal memory that allows the previous inputs to affect the subsequent predictions. It's much easier to predict the next word in a sentence with more accuracy, if you know what the previous words were.

All of above statements lead to have better performance when we are dealing with RNNs.

4 QUESTION3: CONTEXTUAL EMBEDDING + RNNS

4.1

- Remove entities like < > & which gets embedded in the original data.
- In the twitter datasets, there is also other information as retweet, Hashtag, Username and modified tweets. All of this is ignored and removed from the dataset.
- Replace digits with <number>

- Remove Punctuations
- Character normalization (tooooooday -> today)
- Tweets shorter than a length are removed

4.2

We use pretrained Bert model and add LSTM and fully connected layers to it and fine tune the parameters as we do in transfer learning.

From figure? We can see that the length of tweets are shorter than 40 so we pad all the data to 40 maximum length.

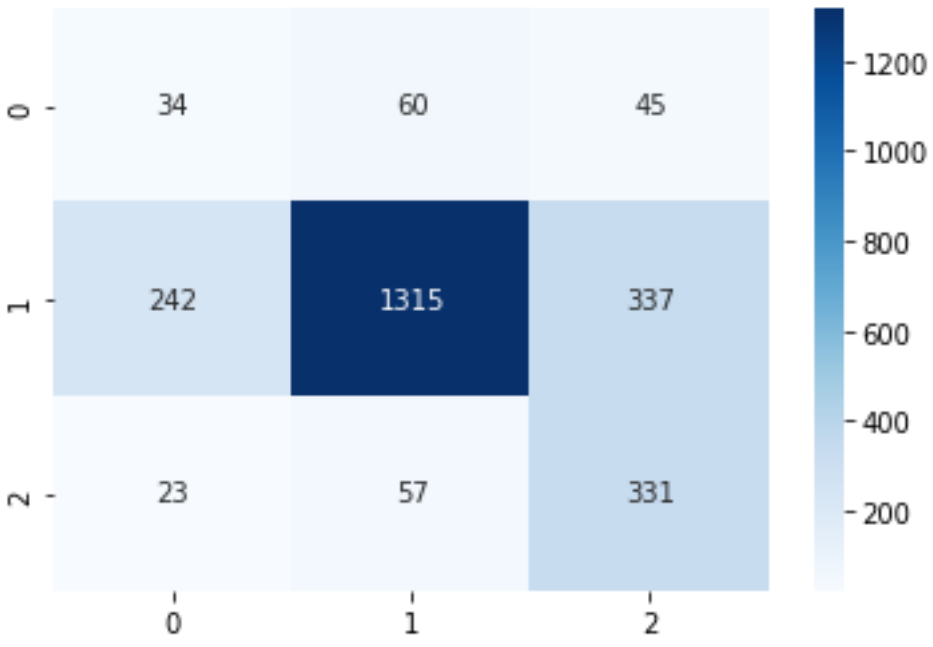

Figure78. Confusion matrix

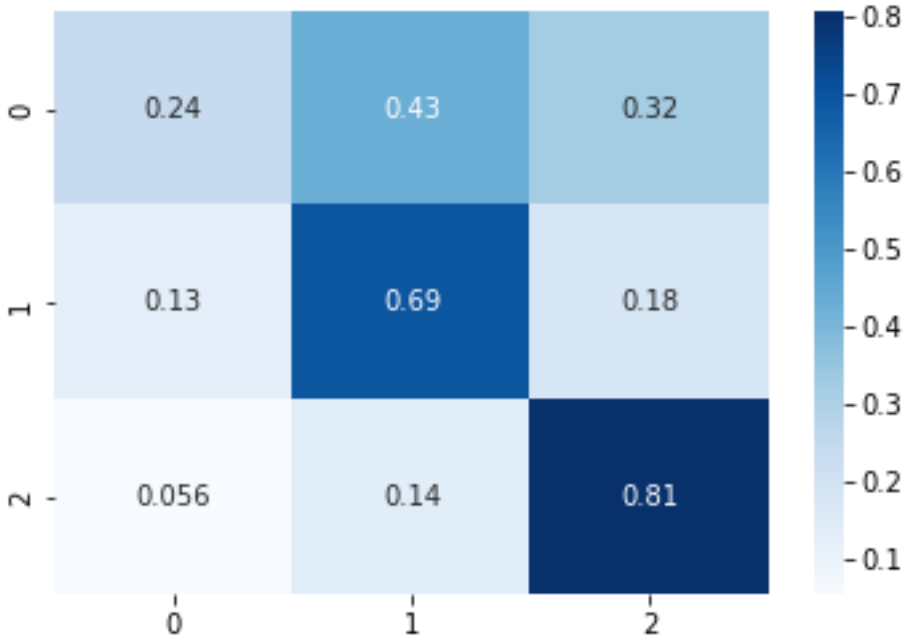

Figure79. Normalized confusion matrix

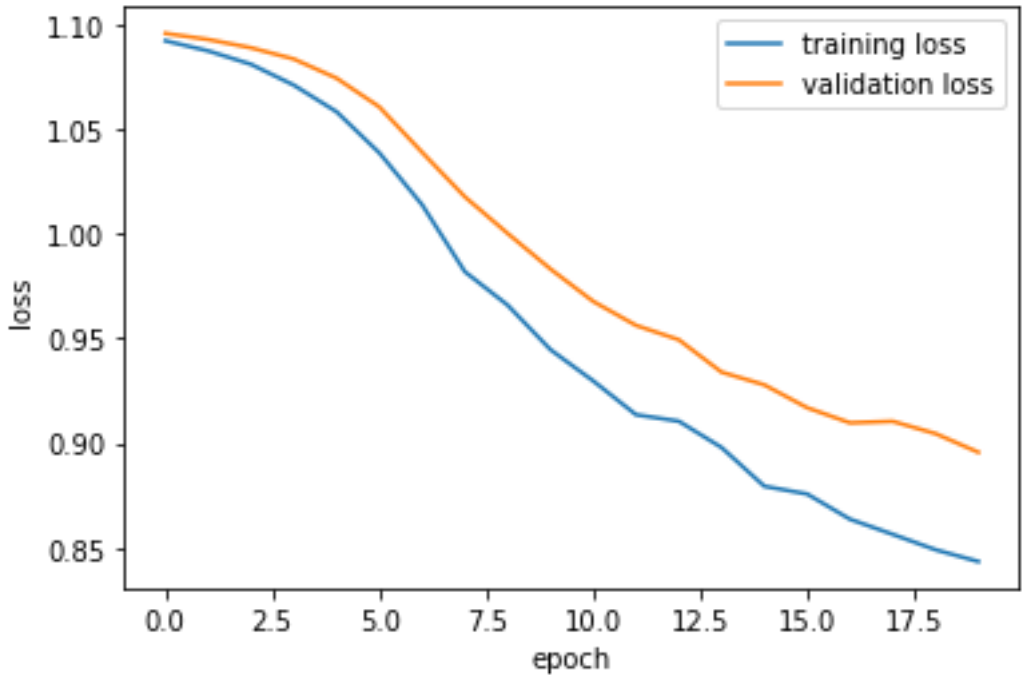

Figure80. validation and train loss in each epoch

	precision	recall	f1-score	support
0	0.15	0.41	0.22	139
1	0.93	0.70	0.80	1894
2	0.51	0.79	0.62	411
accuracy			0.70	2444
macro avg	0.53	0.63	0.55	2444
weighted avg	0.81	0.70	0.73	2444

Figure81. evaluation metrics

4.3

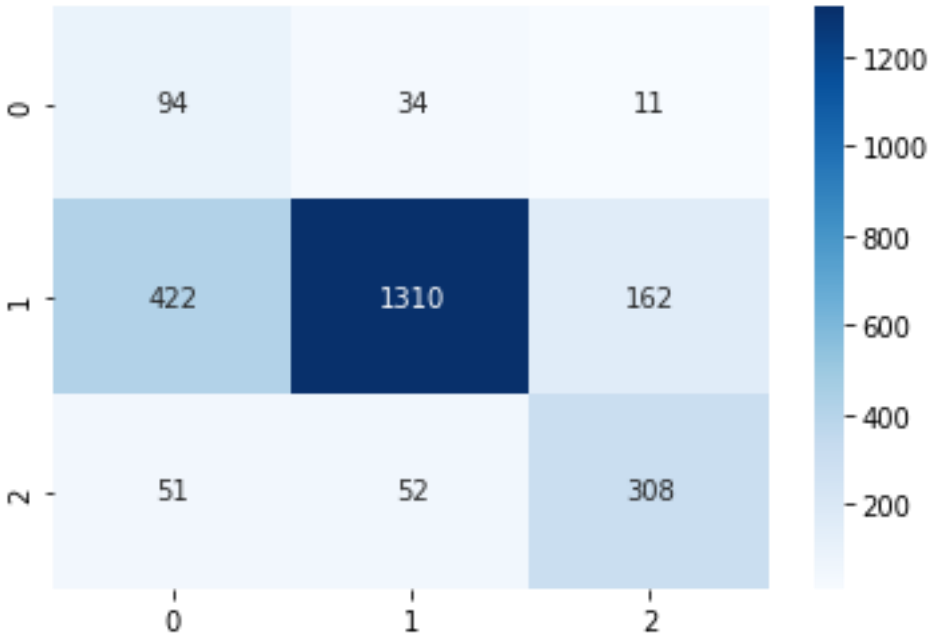

Figure82. confusion matrix

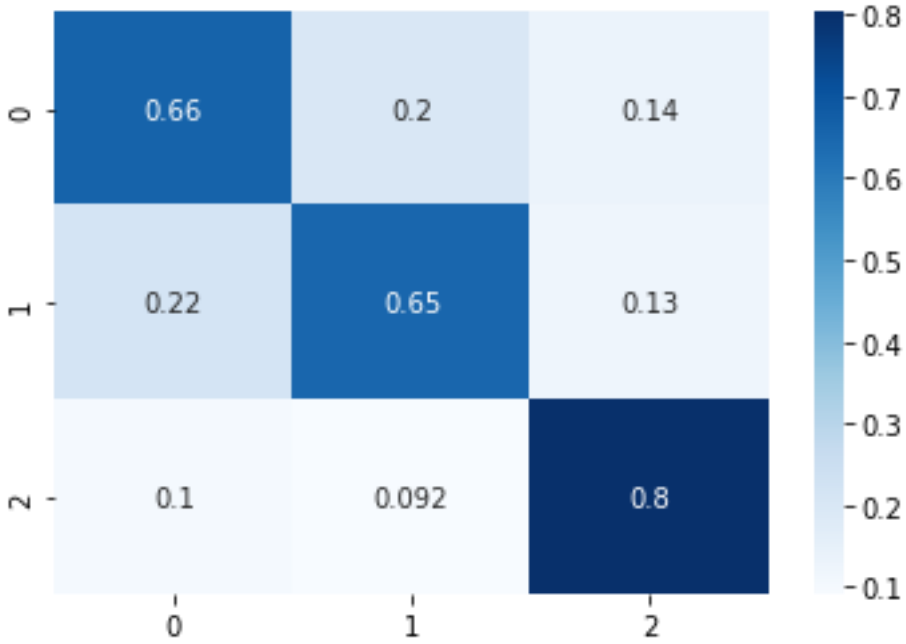

Figure83. Normalized confusion matrix

	precision	recall	f1-score	support
0	0.17	0.68	0.27	139
1	0.94	0.69	0.80	1894
2	0.64	0.75	0.69	411
accuracy			0.70	2444
macro avg	0.58	0.71	0.58	2444
weighted avg	0.84	0.70	0.75	2444

Figure84. evaluation metrics

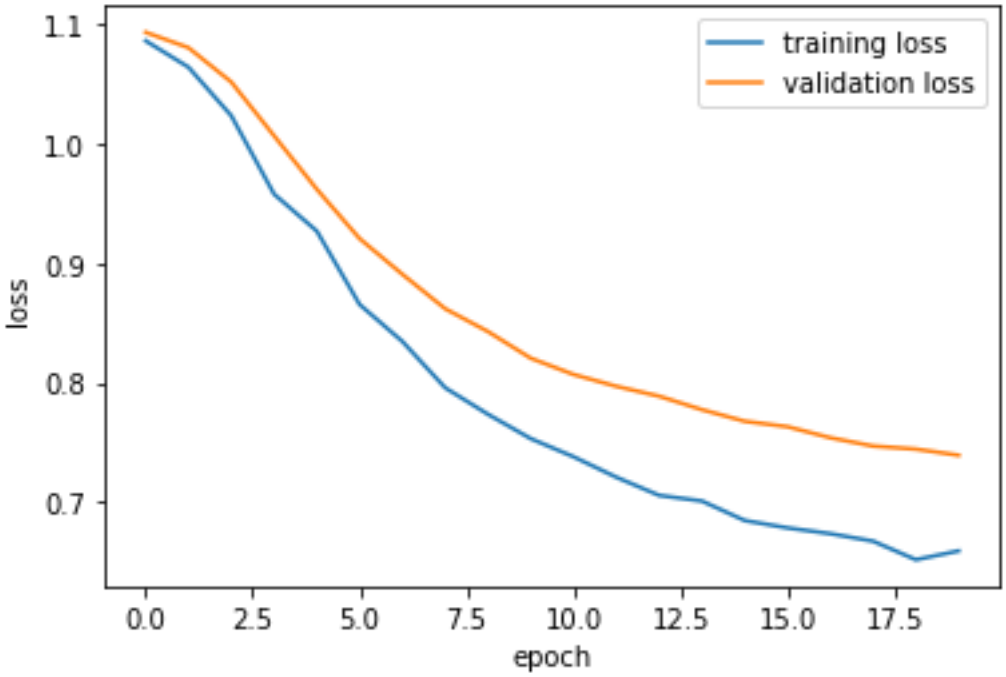

Figure85. validation and train loss in each epoch

4.4

As we can see from the results the pretrained hatebert model has better performance on predicting hate tweets. Normal (class 2) and hate tweets (class 0) precision, recall and f1-score are higher in hatebert model because this model is re-trained by specific dataset that mostly contained hate comments.

So, models like Hatebert are generally trained to perform better at classifying of a particular subject for example hate or political comments.

4.5 (Bonus)

T5 is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks and for which each task is converted into a text-to-text format. T5 works well on a variety of tasks out-of-the-box by prepending a different prefix to the input corresponding to each task, e.g., for translation: *translate English to German: …*, for summarization: *summarize: …*.

T5 uses relative scalar embeddings. Encoder input padding can be done on the left and on the right.

the differentiator that truly sets T5 apart from BERT-style models is that it does not output a label or a span of the input to the input sentence, but the output is a text string as well.

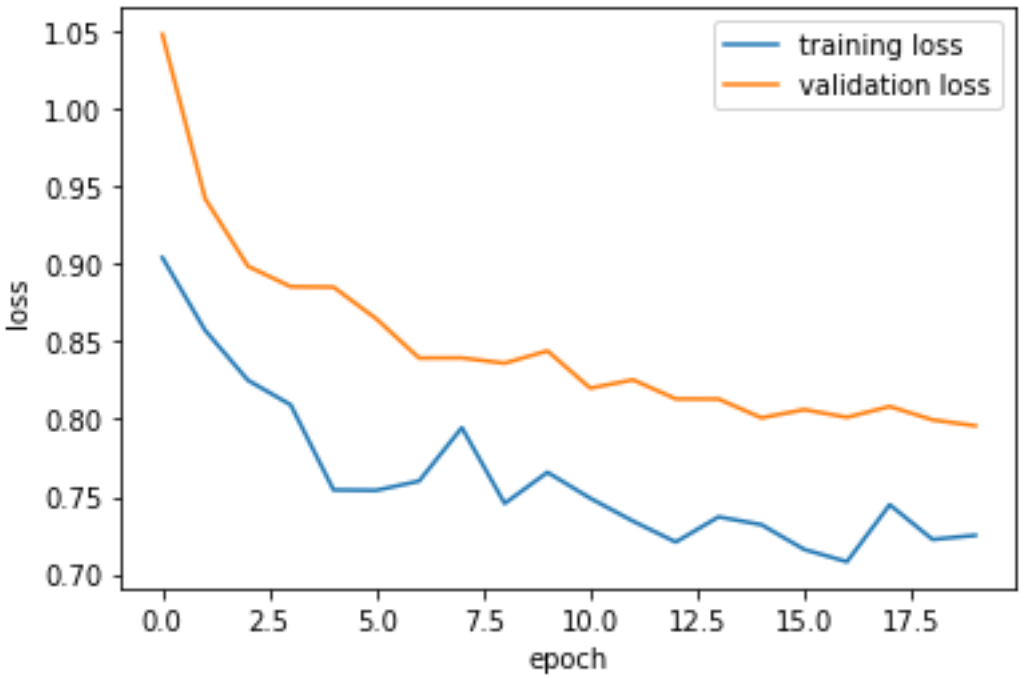

Figure86. validation and train loss in each epoch

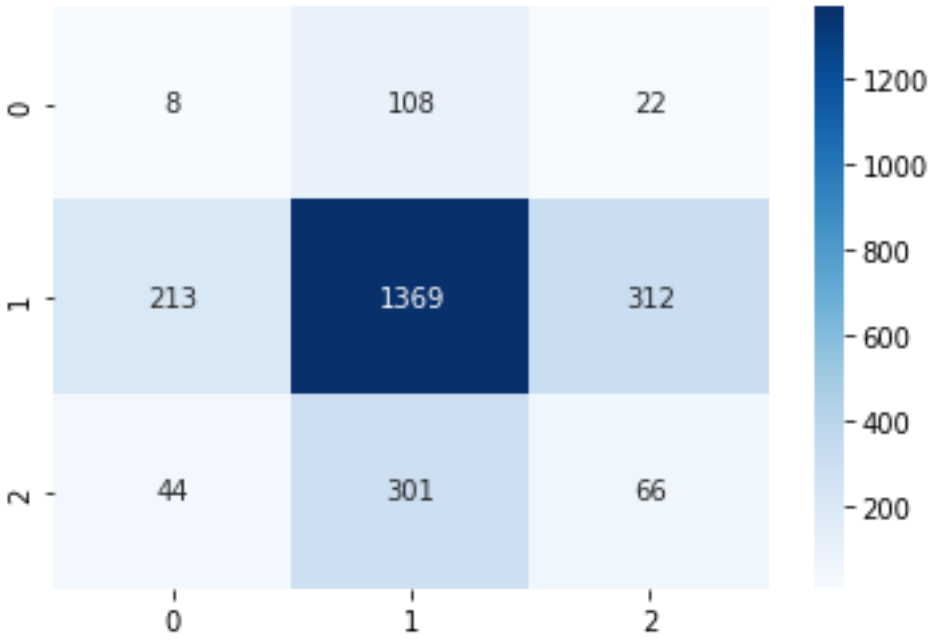

Figure87. confusion matrix

	precision	recall	f1-score	support
0	0.03	0.06	0.04	138
1	0.77	0.72	0.75	1894
2	0.17	0.16	0.16	411
accuracy			0.59	2443
macro avg	0.32	0.31	0.32	2443
weighted avg	0.63	0.59	0.61	2443

Figure88. evaluation metrics