Project 2: Time Series Forecasting using NN, LSTM and CNN

CSC 215-01 Artificial Intelligence (Spring 2023)

Raul Arambula (SID:302741322) and Bashar Allwza (SID:301780232)

Due March 6, 2023

Problem Statement

Time series forecasting and timely prediction are particularly important for predicting the outcome of data that is pointed out in a time order. Our goal is to create three models to predict today's closing value of a stocks based on the previous days. The three models implemented were fully-connected neural network, LSTM, and CNN. The stocks used were Apple.

Methodology

To have better results on our models we started preparing the data with the following steps.

1. Drop unnecessary columns from the dataset.
2. Duplicate the "Close" column in order to use it for the prediction.
3. Normalize the numeric values using "encode_numeric_zscore".
4. Split data 70/30 for training and testing.

A loop of five was used on all models to avoid local optimums. The patience for early stopping was set to five and the number of epochs was set to a hundred. The default batch size was used for the fully-connected neural network, LSTM models, and the CNN model using the last year and a half days of stock data. A batch size of 128 was used for the CNN model using the last seven days of stock data. A hyperparameter tuning table is presented at the end.

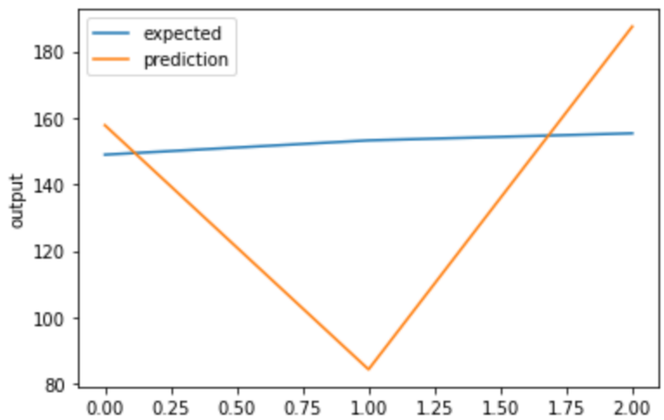
Experimental Results and Analysis

Fully-Connected Neural Network Model Using Last 7 Days of Stock Data

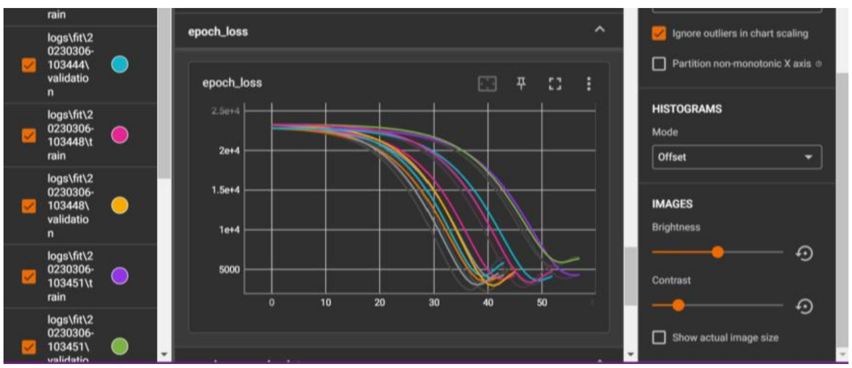
The fully-connected neural network performed best when using six hidden layers. The neuron counts that worked the best were 250 for the first hidden layer, 150 for the second hidden layer, 75 for the third hidden layer, 50 for the fourth hidden layer, 20 for the fifth hidden layer, and 10 for the sixth hidden layer. Relu was the activation function that performed the best. The adam optimizer with default arguments performed best. Check hyperparameter tuning table to see the impact on performance of other parameters.

Final score (RMSE)≈ 45

Regression Lift Chart:



TensorBoard:

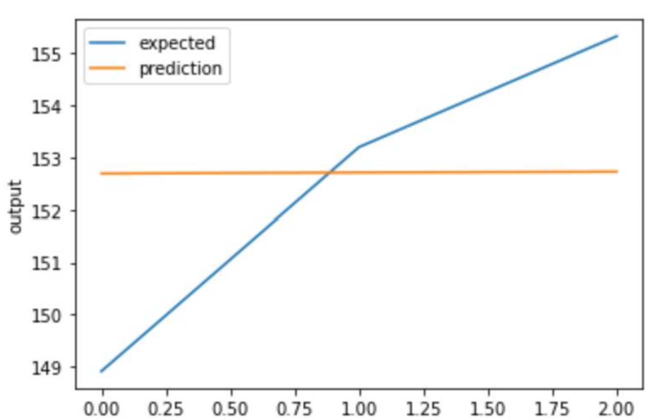


LSTM Model Using Last 7 Days of Stock Data

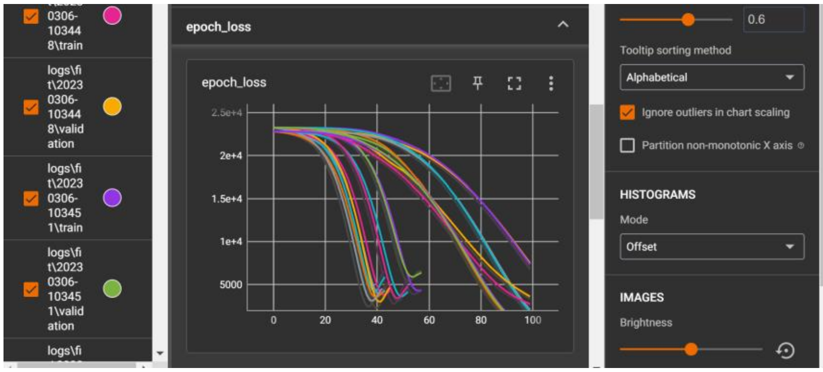
The LSTM model performed best when using a sequence size of five. Four LSTM layers and three dense layers was the optimal configuration. The nueron counts for the LSTM layers were 128, 64, 32, and 16 The nueron counts for the dense layers were 64, 32, and 16. The best activation function was relu and the best optimizer was adam. Check hyperparameter tuning table to see the impact on performance of other parameters.

`Final score (RMSE)≈ 3`

`Regression Lift Chart:`



TensorBoard:

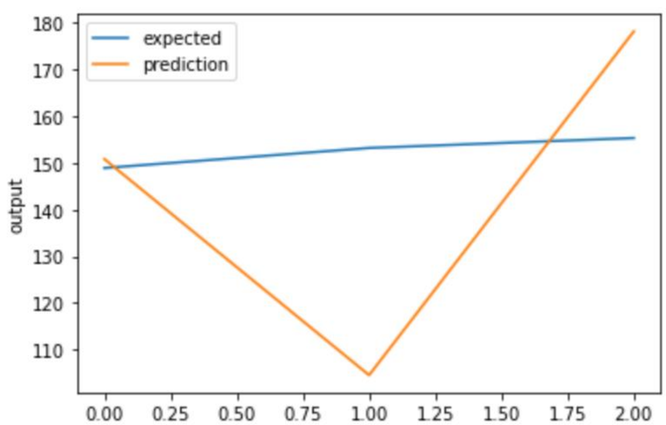


CNN Model Using Last 7 Days of Stock Data

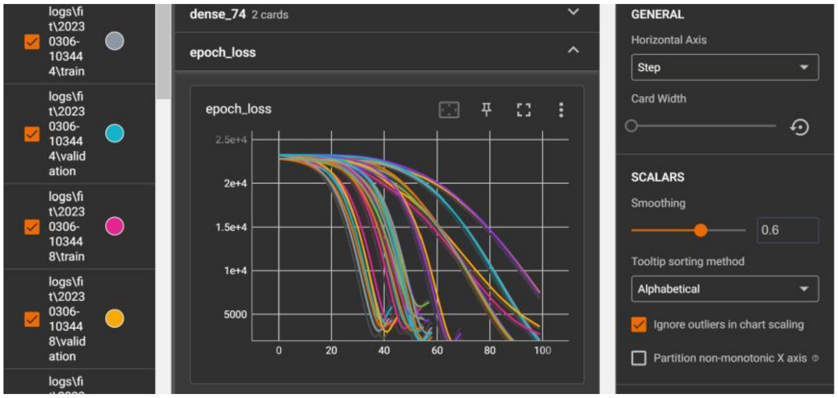
The CNN model performed best when the batch size was 128 and converting the data frame into image like data with dimensions (5,1). Then the image was made to be four dimensional (data.shape[0],5,1,1). The CNN model was optimal when using 4 kernels each with a size of (1,1). The filter count for the Conv2D layers were 64, 32, 16, and 8. Valid padding and relu activation were best for these layers. Three dense layers were used at the end with neuron counts of 128, 64, and 32. These layers also used relu. The best optimizer was adam. Check hyperparameter tuning table to see the impact on performance of other parameters.

```
Final score (RMSE)≈ 30
```

```
Regression Lift Chart:
```



TensorBoard:

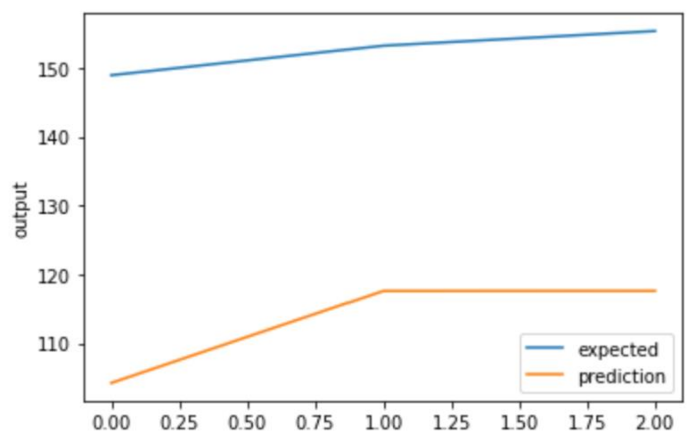


LSTM Model W/Bidirectional and Attention Layers Using Last 7 Days of Stock Data

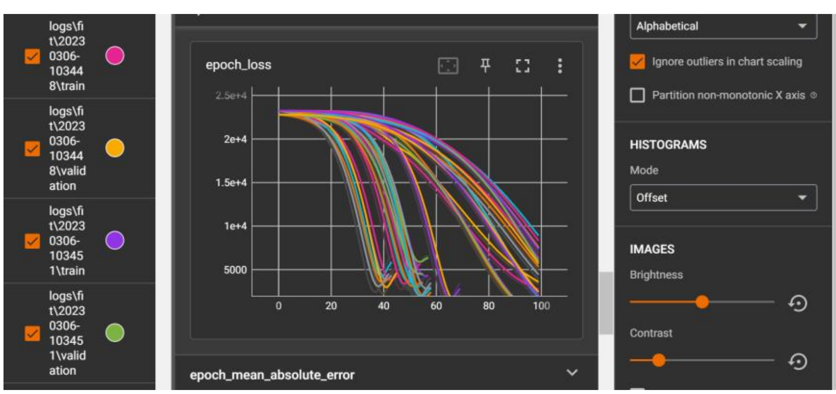
The LSTM model with bidirectional and attention layers performed best when using a sequence size of five. Four bidirectional LSTM layers, one attention layer, and three dense layers was the optimal configuration. The nueron counts for the bidirectional LSTM layers were 128, 64, 32, and 16. The number of units of the attention layer was 16. The nueron counts for the dense layers were 64, 32, and 16. The best activation function was relu and the best optimizer was adam. Check hyperparameter tuning table to see the impact on performance of other parameters.

`Final score (RMSE)≈ 40`

`Regression Lift Chart:`



TensorBoard:

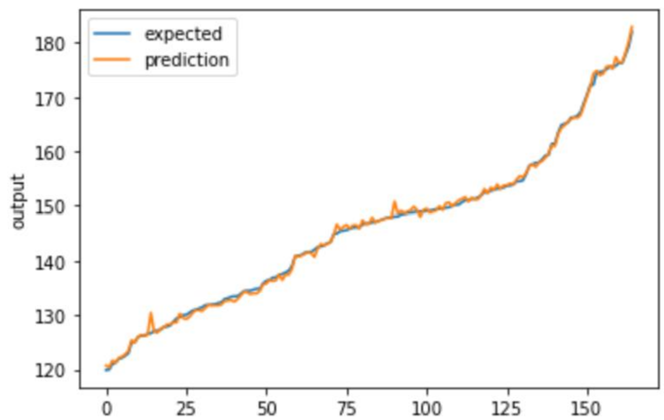


Fully-Connected Neural Network Model Using Last 548 Days of Stock Data

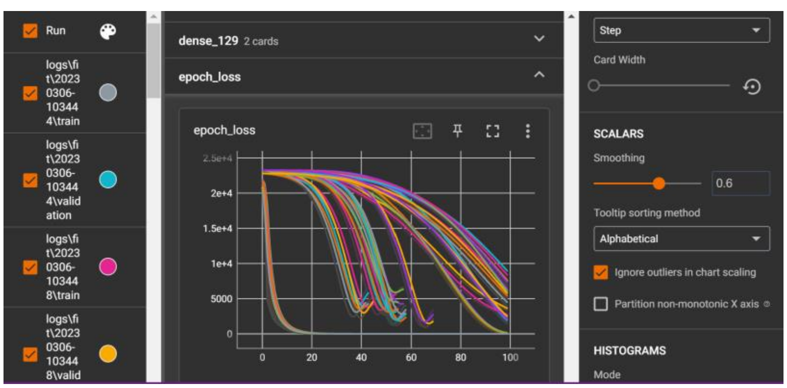
The fully-connected neural network performed best when using six hidden layers. The neuron counts that worked the best were 250 for the first hidden layer, 150 for the second hidden layer, 75 for the third hidden layer, 50 for the fourth hidden layer, 20 for the fifth hidden layer, and 10 for the sixth hidden layer. Relu was the activation function that performed the best. The adam optimizer with default arguments performed best. Check hyperparameter tuning table to see the impact on performance of other parameters.

`Final score (RMSE)≈  .6`

Regression Lift Chart:



TensorBoard:

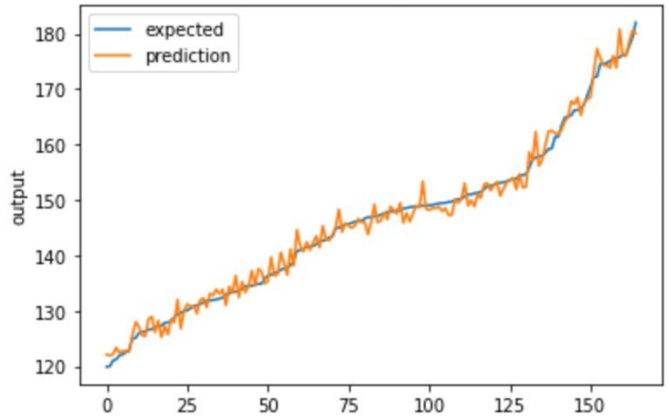


LSTM Model Using Last 548 Days of Stock Data

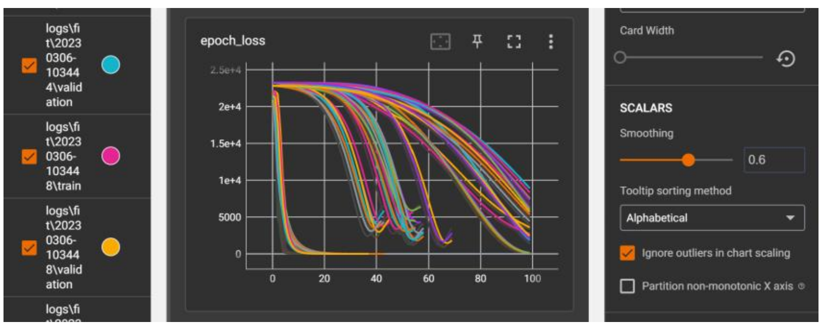
The LSTM model performed best when using a sequence size of five. Four LSTM layers and five dense layers was the optimal configuration. The nueron counts for the LSTM layers were 128, 64, 32, and 16. The nueron counts for the dense layers were 256, 128, 64, 32, and 16. The best activation function was relu and the best optimizer was adam. Check hyperparameter tuning table to see the impact on performance of other parameters.

`Final score (RMSE)≈ 1.6`

`Regression Lift Chart:`



TensorBoard:

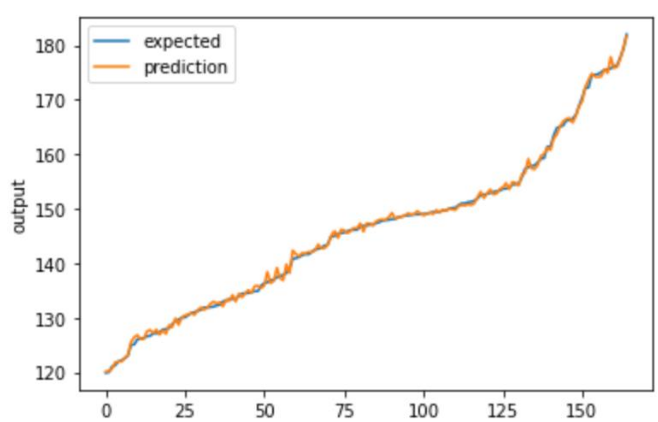


CNN Model Using Last 548 Days of Stock Data

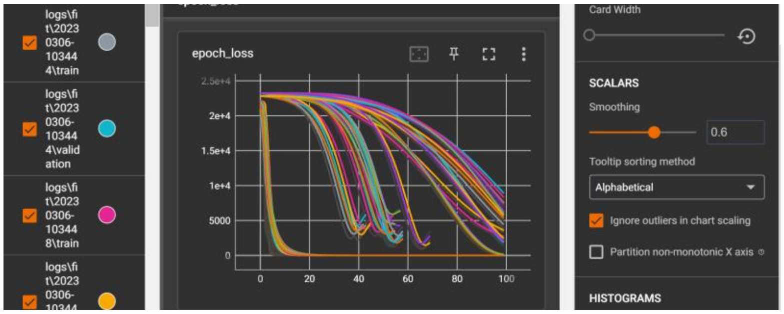
The CNN model performed best when the batch size was 32 and converting the data frame into image like data with dimensions (5,1). Then the image was made to be four dimensional (data.shape[0],5,1,1). The CNN model was optimal when using 2 kernels each with a size of (1,1). The filter count for the Conv2D layers were 32 and 16. Valid padding and relu activation were best for these layers. Five dense layers were used at the end with neuron counts of 256, 128, 64, 32, and 16. These layers also used relu. The best optimizer was adam. Check hyperparameter tuning table to see the impact on performance of other parameters.

`Final score (RMSE)≈ .5`

`Regression Lift Chart:`



TensorBoard:

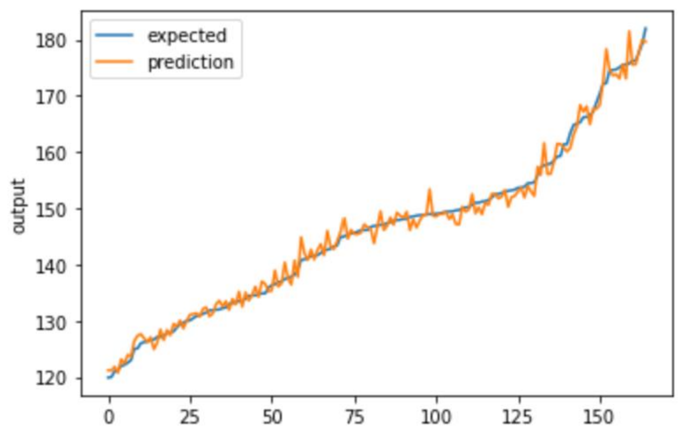


LSTM Model W/Bidirectional and Attention Layers Using Last 548 Days of Stock Data

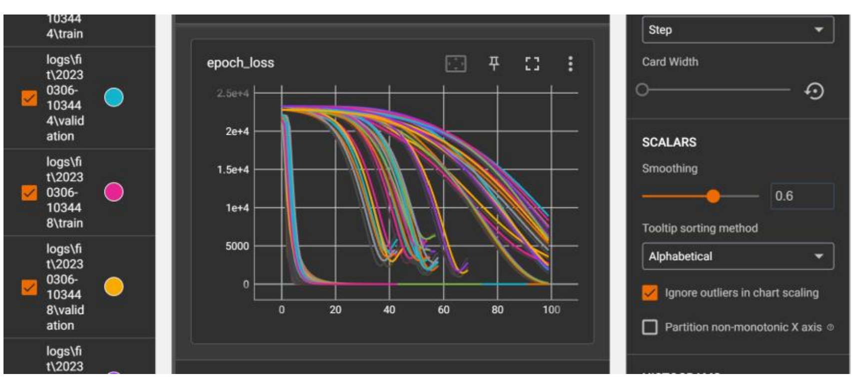
The LSTM model with bidirectional and attention layers performed best when using a sequence size of five. Three bidirectional LSTM layers, one attention layer, and five dense layers was the optimal configuration. The nueron counts for the bidirectional LSTM layers were 32, 16 and 8. The number of units of the attention layer was 128. The nueron counts for the dense layers were 256, 128, 64, 32, and 16. The best activation function was relu and the best optimizer was adam. Check hyperparameter tuning table to see the impact on performance of other parameters.

`Final score (RMSE)≈ 1.5`

`Regression Lift Chart:`



`TensorBoard:`



Task Division and Project Reflection

Work was done together as a team with online meetings. All code was written together in a pair programming style. The report was written together simultaneously. The first challenge was correctly splitting and shaping the data for the LSTM and CNN models. It was also difficult figuring out the correct way to stack the layers for the LSTM and CNN models. These problems were solved through trial and error. The final challenge was correctly implementing the LSTM model with bidirectional and attention layers. This issue was solved by looking up guides and tutorials. The first lesson we learned was that LSTM and CNN models are very sensitive to the way their layers are constructed. We have also learned cooperation and effective communication regarding pair programming. Finally, we learned that there are multitudes of arguments and parameters that can have vastly varying effects on the performance of models.

Additional Features

The additional features we chose were changing the number of days of data to use and building a LSTM model with bidirectional and attention layers. The number of days we found to be the most optimal for our data was a year and a half. Performance gains were negligible beyond a year and a half. Our LSTM model with bidirectional and attention layers did perform better than the baseline fully-connected neural network but could not beat the baseline LSTM or CNN models when using the last 7 days of data. When using the last year and a half of data the LSTM model with bidirectional and attention layers performs marginally better than baseline the LSTM model but not better than the fully-connected neural network or CNN models. Its inability to perform better than the baseline LSTM when using only 7 days of data is most likely a tuning issue, however we do not think it can perform much better when using the last year and a half of data.

Hyperparameter Tuning Table

Model	Number of Days of Data	Hyperparameters	RMSE Scores
Neural Network	7	6 Hidden Layers	45
Neural Network	7	3 Hidden Layers	47
Neural Network	7	Nueron Count: 250,150,75,50,20,10	45
Neural Network	7	Nueron Count: 100,75,50,25,15,8	46
Neural Network	7	Relu	45
Neural Network	7	Sigmoid	149
Neural Network	7	Tanh	147
Neural Network	7	Adam	45
Neural Network	7	SGD	Nan
LSTM	7	2 LSTM Layers	38
LSTM	7	4 LSTM Layers	3
LSTM	7	LSTM Nueron Count:128,16	38
LSTM	7	LSTM Nueron Count:32,8	48
CNN	7	Kernel Size: (2,1)	38
CNN	7	Kernel Size: (1,1)	33
CNN	7	Kernel Number: 2	38
CNN	7	Kernel Number: 4	30
LSTM W/Attention and Bidirectional	7	2 LSTM Layers	40
LSTM W/Attention and Bidirectional	7	4 LSTM Layers	45
LSTM W/Attention and Bidirectional	7	LSTM Nueron Count:128,16	40

LSTM W/Attention and Bidirectional	7	LSTM Nueron Count:32,8	40
Neural Network	548	Nueron Count: 250,150,75,50,20,10	0.6
Neural Network	548	Nueron Count: 100,75,50,25,15,8	0.6
Neural Network	548	Relu	0.6
Neural Network	548	Sigmoid	135
Neural Network	548	Tanh	128
Neural Network	548	Adam	0.6
Neural Network	548	SGD	Nan
LSTM	548	2 LSTM Layers	1.8
LSTM	548	4 LSTM Layers	1.6
LSTM	548	LSTM Nueron Count:32,16	1.8
LSTM	548	LSTM Nueron Count:128,64	1.9
CNN	548	Kernel Size: (2,1)	0.6
CNN	548	Kernel Size: (1,1)	0.5
CNN	548	Kernel Number: 2	0.6
CNN	548	Kernel Number: 4	0.6
LSTM W/Attention and Bidirectional	548	1 LSTM Layer	1.5
LSTM W/Attention and Bidirectional	548	3 LSTM Layers	1.5
LSTM W/Attention and Bidirectional	548	LSTM Nueron Count:256,128,64	1.8
LSTM W/Attention and Bidirectional	548	LSTM Nueron Count:32,16,8	1.5