Mini-Project 4: Stock price prediction using neural network, lstm and cnn

Version 1.0

CSC215, Fall 2018

Nov 5th, 2018

Prepared by: Chandini Nagendra

Siddharth Chittora

Contents

[1. Problem Statement 3](#_Toc529045958)

[2. Methodology 3](#_Toc529045959)

[2.1 Fully Connected Neural Network 3](#_Toc529045960)

[2.2 LSTM (Long Short-Term Memory) 4](#_Toc529045961)

[2.3 CNN (Convolutional Neural Network) 4](#_Toc529045962)

[3. Experimental Results and Analysis 5](#_Toc529045963)

[3.1 Regression Lift chart 6](#_Toc529045964)

[Activation ReLU 6](#_Toc529045965)

[Activation Sigmoid 6](#_Toc529045966)

[Activation Tanh 6](#_Toc529045967)

[4. Task Division 7](#_Toc529045968)

[4.1. Chandini Nagendra: 7](#_Toc529045969)

[4.2. Siddharth Chittora 7](#_Toc529045970)

[5. Project Reflection 7](#_Toc529045971)

[6. Additional Features 7](#_Toc529045972)

[6.1 Different Data Sets 7](#_Toc529045973)

[6.1.1 Regression Lift chart 7](#_Toc529045974)

[6.2 Predict Stock Price for Continues time period 10](#_Toc529045975)

[6.2.2 7 Day true and Predicted value for 3 sets of 7-day data 10](#_Toc529045976)

[6.2.3 Regression Lift chart 10](#_Toc529045977)

[6.3 Table showing RMSE and R2 for Additional Features 11](#_Toc529045978)

# Problem Statement

Task 1: Use the daily [Open, High, Low, Volume] to predict [Close] on that day using **a fully-connected neural network**. Use the first 70% of the records for training and the remaining 30% of the records for test. Report the RMSE of the model. Show the “regression lift chart” of your test data.

Task 2: Predict [Close] of a day based on the last 7 days’ data [Open, High, Low, Volume, Close] using a **LSTM model**. In other words, we want to predict the price in the green cell using all the numbers in the red cell. Use the first 70% of the available records for training and the remaining 30% of the available records for test. Report the RMSE of the model. Show the “regression lift chart” of your test data.

Task 3: Do the same as Task 2 but use a **CNN model**. Report the RMSE of the model. Show the “regression lift chart” of your test data.

# Methodology

* The data set considered for this project consist of following columns. Date, Open, High, Low, Close, Adj\_Close and Volume
* For data pre-processing, removed the columns Date and Adj\_Close
* Removed the null values if any from the data set.
* Normalized the columns Open, High, Low, Close and Volume
* Did not Normalize the column Close as this will be the label used for prediction
* Converted pandas dataframe to x y input that tensorflow requires
* Finally Split the data, 70% for training and 30% for testing.

## 2.1 Fully Connected Neural Network

* Trained the Tensorflow models with activation function ReLU, Sigmoid and Tanh.
* Using each of the activation function mentioned above experiment by using optimizers adam, sgd and rmsprop.
* Also experimented with 2, 3 and 4 layers and altered neuron counts
* Used early stopping and Model checkpointing to save the best weights
* Added a dropout layer to see how it affected the model
* Calculated RMSE and R2 score for each model
* Details of experimental results are as shown in the table 1.0

## 2.2 LSTM (Long Short-Term Memory)

* Using the data in the given dataset for seven days, predicting the close price for the 8th day
* Separated the Close field and saved it in a different dataframe to be used as a label
* Normalized the columns Open, High, Low, Volume and Close and used it as input data for the seven-day dataframe
* Did not normalize the separated close dataframe.
* Modified the function **to\_sequences(seq\_size, data, label)** (given in the tutorials for LSTM) to be able to use it for 7 day sliding window, extracted the x and y in the form of numpy arrays
* Using the x and y obtained, split the data into 70% training data and 30% testing data
* Trained the LSTM model. Experimented with 1 and 2 layers for LSTM
* Used the optimizer function adam
* Used early stopping and model checkpointing
* Performed prediction on test data
* Calculated the RMSE and R2 score. Details of the experiment is given below in the table 1.0

## 2.3 CNN (Convolutional Neural Network)

* Using the data in the given dataset for seven days, predicting the close price for the 8th day
* Separated the Close field and saved it in a different dataframe to be used as a label
* Normalized the columns Open, High, Low, Volume and Close and used it as input data for the seven-day dataframe
* Did not normalize the separated close dataframe.
* Modified the function **to\_sequences(seq\_size, data, label)** (given in the tutorials for LSTM) to be able to use it for 7 day sliding window, extracted the x and y in the form of numpy arrays
* Using the x and y obtained, split the data into 70% training data and 30% testing data
* CNN model expects the data to be in 4d, therefore reshaped the train and test data using **reshape** function
* Visualized the sliding window as an image as CNN model is specifically made for an image. Visualized the window as an image with one row, seven columns and 5 channels
* Trained the cnn model with 1 and 2 Conv2d layer and Kernel size (1,5)
* Used max pooling and dropout in the model.
* Used early stopping and model checkpointing
* Used the activation function ReLU and optimizer adam
* Used a Dense layer
* Predicted the eight-day stock price using the last seven-day data
* Calculated the RMSE and R2 score
* Detailed experimental results are shown in the table 1.0

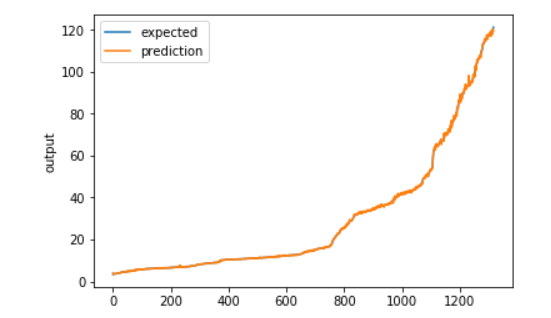
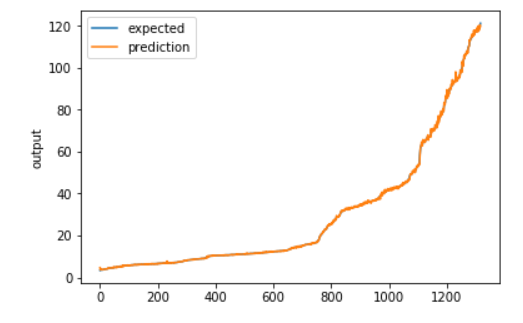
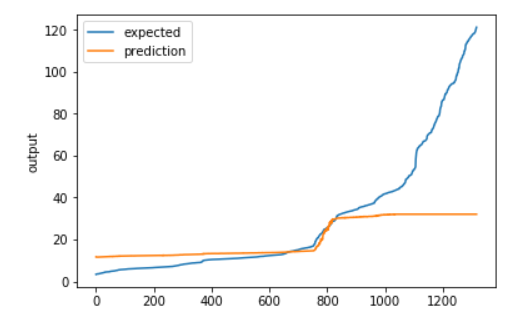
# Experimental Results and Analysis

|  |  |  |
| --- | --- | --- |
| **Model & Tuning** | **RMSE** | **R2 Score** |
| **Tensor flow regression neural network models** |  |  |
| ReLU + adam + 2 layers + early stopping + model checkpointing | 0.3784516751766205 | 1.00 |
| ReLU + adam + 3 layers + early stopping + model checkpointing | 0.3731230497360229 | 1.00 |
| ReLU + adam + 4 layers + early stopping + Model Checkpointing | 0.3739757239818573 | 1.00 |
| ReLU + adam + 4 layers + early stopping + Model Checkpointing + dropout | 0.3828038871288299 | 1.00 |
| ReLU + sgd + 3 layers + early stopping + Model Checkpoint | 29.916187286376953 | -0.00 |
| ReLU + sgd + 3 layers + early stopping + Model Checkpointing + dropout | 24.74211311340332 | 0.32 |
| ReLU + rmsprop + 3 layers + early stopping + Model Checkpointing | 0.3835900723934173 | 1.00 |
| ReLU + rmsprop + 3 layers + early stopping + Model Checkpointing + dropout | 0.3877429068088531 | 1.00 |
| Sigmoid + adam + 2 layers + early stopping + model checkpointing | 0.4187675714492798 | 1.00 |
| Sigmoid + adam + 3 layers + early stopping + model checkpointing | 29.907743453979492 | 0.00 |
| Sigmoid + adam + 4 layers + early stopping + Model Checkpointing | 29.90807342529297 | -0.00 |
| Sigmoid + adam + 4 layers+ early stopping + Model Checkpointing + dropout | 9.273457527160645 | 0.90 |
| Sigmoid + sgd + 3 layers + early stopping + Model Checkpoint | 0.6335227489471436 | 1.00 |
| Sigmoid + sgd + 3 layers + early stopping + Model Checkpointing + dropout | 0.8733727931976318 | 1.00 |
| Sigmoid + rmsprop + 3 layers + early stopping + Model Checkpointing | 4.313910007476807 | 0.98 |
| Sigmoid + rmsprop + 3 layers + early stopping + Model Checkpointing + dropout | 0.5480981469154358 | 1.00 |
| Tanh + adam + 2 layers + early stopping + model checkpointing | 0.3927091658115387 | 1.00 |
| Tanh + adam + 3 layers + early stopping + model checkpointing | 0.4639947414398193 | 1.00 |
| Tanh + adam + 4 layers + early stopping + Model Checkpointing | 7.534976959228516 | 0.94 |
| Tanh + adam + 4 layers + early stopping + Model Checkpointing + dropout | 7.3430399894714355 | 0.94 |
| Tanh + sgd + 3 layers + early stopping + Model Checkpoint | 28.205272674560547 | 0.11 |
| Tanh + sgd + 3 layers + early stopping + Model Checkpointing + dropout | 29.90804672241211 | 0.00 |
| Tanh + rmsprop + 3 layers + early stopping + Model Checkpointing | 0.5480981469154358 | 1.00 |
| Tanh + rmsprop + 3 layers + early stopping + Model Checkpointing + dropout | 3.440717935562134 | 0.99 |
| **LSTM** |  |  |
| 1 layer + early stopping + Model Checkpointing | 1.4062001345600446 | 0.9976032351042484 |
| 2 layers + early stopping + Model Checkpointing | 1.4062001345600446 | 0.997613912861848 |
| 2 layers + early stopping + Model Checkpointing + Dropout | 1.5346226538587044 | 0.9971441152140671 |
| **CNN** |  |  |
| 1 layer + early stopping + Model Checkpointing | 1.850536731132474 | 0.9956193848910099 |
| 2 layers + early stopping + Model Checkpointing | 1.3924339979203315 | 0.9976865261329603 |
| 2 layers + early stopping + Model Checkpointing + maxpooling + Dropout | 1.3930596890778992 | 0.9976893661267249 |

## 3.1 Regression Lift chart

### 3.1.1 Fully Connected Neural Network

#### Activation - ReLU



###### Adam SGD Rmsprop

#### Activation - Sigmoid

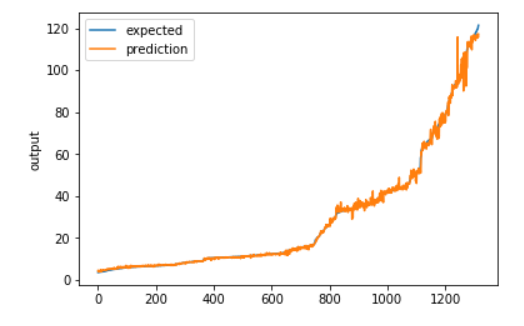
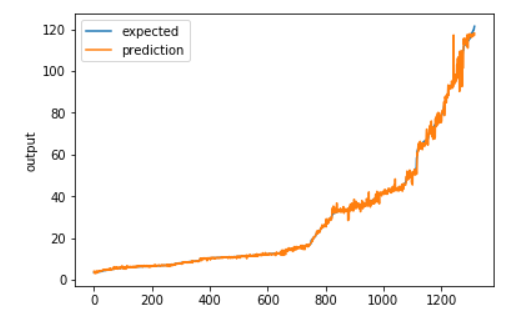
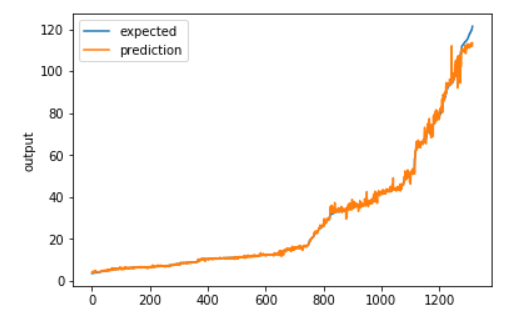
A close up of a map

Description automatically generatedA close up of a map

Description automatically generatedA close up of a map

Description automatically generated

###### Adam SGD Rmsprop

#### A close up of a map Description automatically generatedA close up of a map Description automatically generatedA close up of a map Description automatically generatedActivation - Tanh

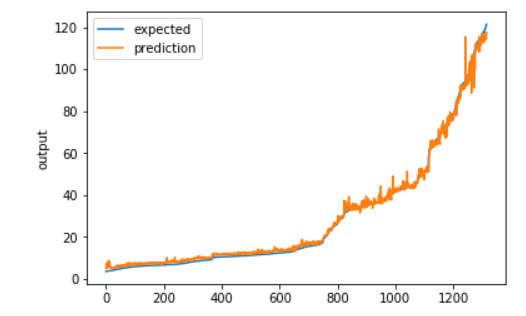
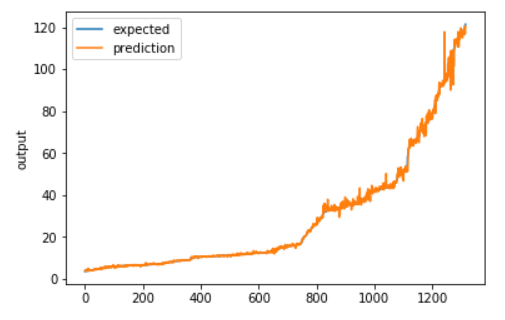
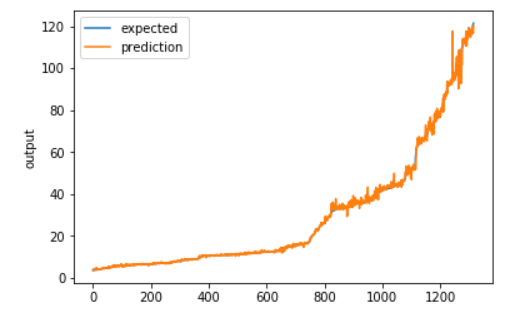
###### Adam SGD Rmsprop

### 3.1.2 LSTM (Long Short Time Memory)



###### 1 LSTM Layer 2 LSTM Layer 2 LSTM Layer + Dropout

### 3.1.3 CNN (Convolutional Neural Network)



###### 1 Conv2d Layer 2 Conv2d Layer 2 Conv2d Layer + dropout

# Task Division

## Chandini Nagendra:

* Fully Connected Neural Network
* CNN
* Report

## Siddharth Chittora

* Fully Connected Neural Network
* LSTM
* Report

Discussed together on how to improve the model and came up with the solution discussed in the additional features section.

# Project Reflection

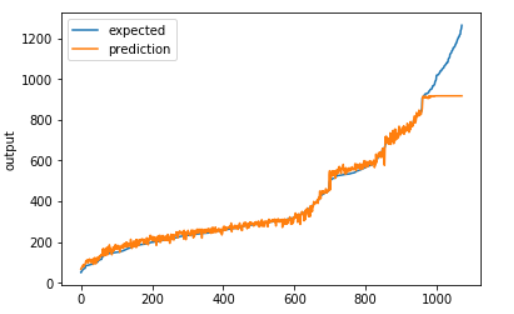
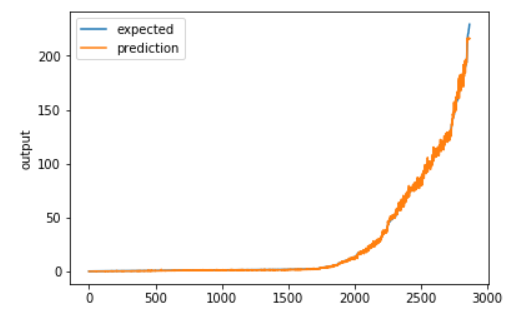
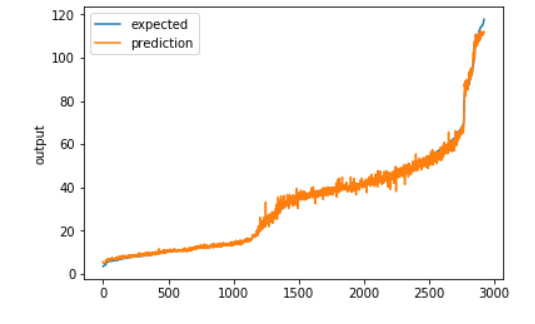
* For fully connected Neural Network, following combination of hyper parameters performed well,
  + activation function Relu and optimizer Adam and RMSprop
  + activation function Sigmoid and optimizer RMSprop
  + activation function Tanh and optimizer Adam
* Optimizer SGD performed poorly, irrespective of the activation function. The results are as shown in the table below.
* If the activation function softmax is used for a regression problem it brings the R2 score to 0. Need to be careful only to use ReLU, Sigmoid and tanh for regression problem and softmax for classification
* Both CNN and RNN with LSTM works well for the Stock prediction problem, be it a single day prediction or a series of multiple day prediction.

# Additional Features

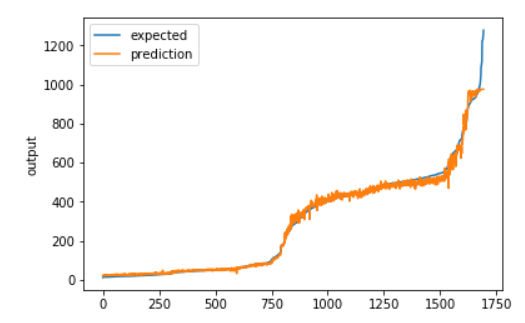
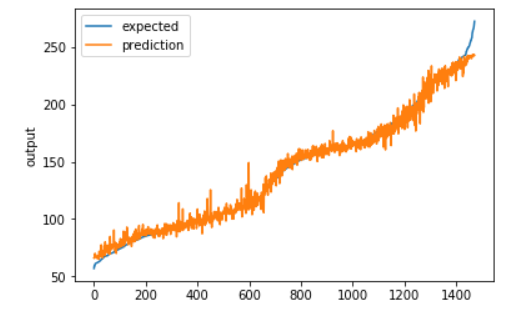
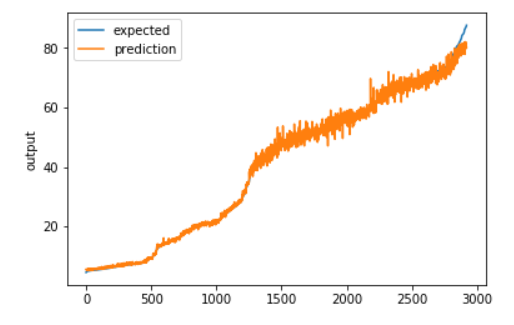
## 6.1 Different Data Sets

* Downloaded and trained LSTM model with different data sets
* Data sets used were Google, Apple, JP Morgan, Goldman Sachs, Reliance, Shell

### 6.1.1 Regression Lift chart



#### Google Apple JP Morgan



#### Goldman Sachs Reliance Shell

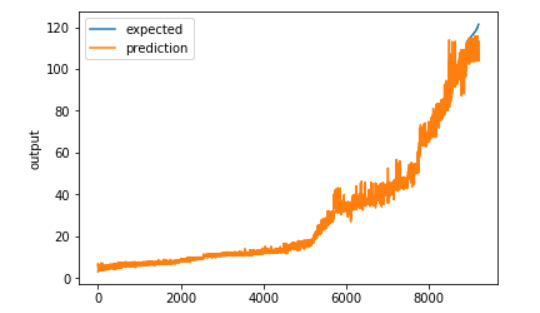
## 6.2 Predict Stock Price for Continues time period

* Using the data in the given dataset for seven days, predicting the close price for the next 7 days
* Separated the Close field and saved it in a different dataframe to be used as a label
* Normalized the columns Open, High, Low, Volume and Close and used it as input data for the seven-day dataframe
* Did not normalize the separated close dataframe.
* Modified the function **to\_sequences(seq\_size, data, label)** (given in the tutorials for LSTM) to be able to use it for 7 day sliding window, extracted the x and y in the form of numpy arrays, with y being the close stock values of next 7 days

### 6.2.2 7 Day true and Predicted value for 3 sets of 7-day data

|  |  |
| --- | --- |
| **True Value** | **Predicted Value** |
| **Set 1** |  |
| 3.953125 | 5.970350742340088 |
| 3.765625 | 5.4521918296813965 |
| 3.953125 | 4.616113185882568 |
| 3.9375 | 4.268948078155518 |
| 4.203125 | 4.303502082824707 |
| 4.125 | 4.355138301849365 |
| 4.140625 | 4.341532230377197 |
| **Set 2** |  |
| 52.630001068115234 | 54.08523178100586 |
| 52.595001220703125 | 53.506290435791016 |
| 58.380001068115234 | 52.01629638671875 |
| 58.5099983215332 | 50.512142181396484 |
| 58.5099983215332 | 50.221710205078125 |
| 60.0 | 50.21151351928711 |
| 60.13999938964844 | 51.08226776123047 |
| **Set 3** |  |
| 64.79000091552734 | 68.39259338378906 |
| 65.375 | 66.59864807128906 |
| 64.94999694824219 | 65.91062927246094 |
| 64.70999908447266 | 64.97370910644531 |
| 64.63500213623047 | 65.6392822265625 |
| 63.689998626708984 | 66.072998046875 |
| 63.86000061035156 | 66.35757446289062 |

### 6.2.3 Regression Lift chart



## 6.3 Table showing RMSE and R2 for Additional Features

|  |  |  |
| --- | --- | --- |
| **Model & Tuning** | **RMSE** | **R2 Score** |
| **LSTM on different datasets** |  |  |
| Google | 56.768039780720166 | 0.9488591165992385 |
| Apple | 1.6703499441326957 | 0.9987481603845102 |
| JP Morgan | 1.5011174123345588 | 0.9958564007849647 |
| Goldman Sachs | 5.726676372376688 | 0.9865372561295315 |
| Reliance | 25.614558826289425 | 0.9900416728137975 |
| Shell | 1.4574782728059565 | 0.9962500139483056 |
| **Close price prediction of 5 consecutive days** |  |  |
| 2 layers + early stopping + Model Checkpointing + dropout | 2.405463457107544 | 0.9932324522800101 |