Artificial Intelligence

SACRAMENTO STATE

Mini-Project 2
Time Series Forecasting

Sai Sri Meghana Dharmapuri (220250135)

Time Series Forecasting

Sai Sri Meghana Dharmapuri

Computer Science department, California State University Sacramento
6000 J street, Sacramento, CA - 95825

***Abstract*— Prediction of movement of the stock price index is seen as a challenging task of prediction of the financial time series. An accurate prediction of the change in stock prices will yield investor income. Development of efficient models for forecasting is very difficult due to the complexity of stock market data. This paper aims to use time series data to predict stock price. Models like Fully Connected Neural Networks, CNN and LSTM have been used.**

***Keywords*— Stock Prediction, Data Pre-processing, Data Analysis, Fully connected Neural Network, CNN (Convolution Neural Network), LSTM (Long short term memory networks)**

I. INTRODUCTION

The prediction of future event or events can be established by analyzing the historical data. Many of the issues with the forecasting include time analysis. For a selected variable (eg: stock price) a time series data may be described as a chronological sequence of observations.

Stock market prediction is regarded as a challenging task of the financial time series prediction process since the stock market is essentially dynamic, nonlinear, complicated, nonparametric, and chaotic in nature.

It is understood that information is abundantly provided in a short period of time. A quick action is a required attribute. In this, I consider time to be the independent variable and goal is to predict the future based on the past analysis.

II. METHODOLOGY

The datasets used follow an OHLC format (Open, high low, close). It is a way to represent price movement of a stock in a particular time frame. CSC215_P2_Stock_Price.csv and JPM.csv is the dataset that the models have been used.

A. Data Preprocessing

Data Preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format. Data processing services require skilled professionals to apply different techniques for analysing and processing data. For every business organization, data has become the most important tool to make critical decisions.

The first step I performed was to import our datasets, which are available in the form of csv files and I stored them into dataframes. A Dataframe is the most common Structured API and simply represents a table of data with rows and columns. I cross checked if the file was imported properly and then performed a search for null values and noticed that there Ire no null values. I remove the columns I think are not required like data and Adj_close. Normalisation is then performed on numeric attributes to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. While there are other normalisation techniques, I performed Zscore normalisation for our data.

B. Data Transformation

Data transformation is the process of converting data from one format to another, typically from the format of a source system into the required format of a destination system. Data transformation is a component of most data integration and data management tasks, such as data wrangling and data warehousing. I used a function named to_sequences to prepare 7 * 5 = 35 input features to predict the output value (‘y’)

III. EXPERIMENTS, RESULTS AND ANALYSIS

I used three machine learning models i.e Long short-term memory networks, convolution neural networks and fully connected neural networks in order to predict stock. I used sklearn library to split the dataset into two sets, one for training and the other for testing; and this is done before starting model training. After the split, I have 3069 rows for training dataset and 1316 rows for the test dataset.

A. Convolution Neural Network(CNN)

Is a deep neural network where each neuron from the previous layer is connected to all neurons int the next layer. The number of parameters in CNNs is smaller compared to a fully connected neural network. CNN operations include three stages, namely, convolutional stage, nonlinear transformation stage, and pooling stage.

Each CNN operation includes the stack of input, convolutional layer, activation, pooling layer, and fully connected layer.

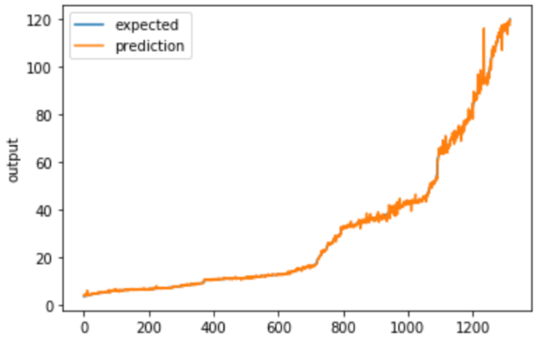
In this project I input data in 4D array shape of (N,C,X,Y), where N=number of samples, C= number of channels per sample, (X,Y)= sample size.

Analysis with different combinations of Kernel size, dropout, strides and dense layers along with optimizers and activation layers:

Kernel Size	strides	dropout	dense layers	optimizers	activation	MSE	RMSE
3,3	1,1	0.5	64,1	Adam	relu	2.871454802	1.694536751
3,3	1,1	0.5	64,1	Adam	sigmoid	4.92756585	2.21981212
3,3	1,1	0.5	64,1	Adam	tanh	3.451603053	1.857849039
3,3	1,1	0.5	64,1	sgd	relu	576.0375136	24.00078152
3,3	1,1	0.5	64,1	sgd	sigmoid	4.184324236	2.045562083
3,3	1,1	0.5	64,1	sgd	tanh	36.8953984	6.074158246
5,5	***1,1***	***none***	***64,1***	***Adam***	***relu***	***1.151133053***	***1.072908688***
5,5	1,1	none	64,1	Adam	sigmoid	1.284305378	1.133271979
5,5	1,1	none	64,1	Adam	tanh	1.671228611	1.292760075
5,5	1,1	none	64,1	sgd	relu	17690.91084	133.0071834
5,5	1,1	none	64,1	sgd	sigmoid	22.47658104	4.740947272
5,5	1,1	none	64,1	sgd	tanh	342.4299865	18.50486386

I noticed that the best results Ire when I used
Kernel size: 5, 5; Strides: 1,1; Dropout: none; Dense layer: 64,1; Optimizer: Adam; activation: relu

```
Final score (MSE): 1.1511330531687562
Final score (RMSE): 1.0729086881784284
```



B. *Long short term memory networks(LSTM)*

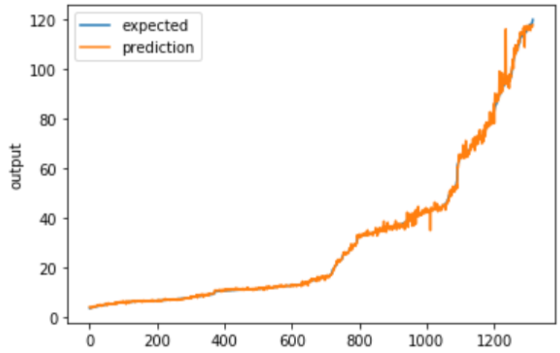
Is a recurrent neural network that is capable of learning long term dependencies. They are networks with loops in them, allowing information to persist. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior. It is composed of input, output, forget gate and the key feature i.e. the cell state.

I took data in a 3D format to predict the stock price with the below combinations of dense layers, optimizers and activation layers used.

LSTM Layers	Dense Layers	Optimizer	Activation Function	MSE	RMSE
64 (Dropout: 0.1)	32,32,1	adam	relu	4.068796991	2.017125924
64 (Dropout: 0.1)	32,32,1	adam	sigmoid	7.622987368	2.7609758
64 (Dropout: 0.1)	32,32,1	adam	tanh	6.631325781	2.57513607
64 (Dropout: 0.1)	32,32,1	sgd	relu	1591.255839	39.89054825
64 (Dropout: 0.1)	32,32,1	sgd	sigmoid	3.536089284	1.880449224
64 (Dropout: 0.1)	32,32,1	sgd	tanh	4.057501017	2.01432396
64 (Dropout: 0.1)	32,1	adam	relu	3.915065687	1.978652493
64 (Dropout: 0.1)	32,1	adam	sigmoid	4.361511622	2.088423238
64 (Dropout: 0.1)	32,1	adam	tanh	4.360648892	2.088216677
64 (Dropout: 0.1)	32,1	sgd	relu	4.360648892	2.088216677
64 (Dropout: 0.1)	32,1	sgd	sigmoid	4.360648892	2.088216677
64 (Dropout: 0.1)	32,1	sgd	tanh	4.360648892	2.088216677
64,32 (Dropout: 0.1)	32,1	adam	relu	7.948688997	2.819341944
64,32 (Dropout: 0.1)	32,1	adam	sigmoid	6.956585509	2.637533983
64,32 (Dropout: 0.1)	32,1	adam	tanh	9.090454144	3.015038
64,32 (Dropout: 0.1)	32,1	sgd	relu	2542.254634	50.42077582
64,32 (Dropout: 0.1)	32,1	sgd	sigmoid	4.915438785	2.217078886
64,32 (Dropout: 0.1)	32,1	sgd	tanh	5.629066768	2.37256544
64,32	32,1	adam	relu	3.949532163	1.987342991
64,32	32,1	adam	sigmoid	1.875706571	1.369564373
64,32	32,1	adam	tanh	1.613521617	1.270244708
64,32	32,1	sgd	relu	630.8682838	25.11709147
64,32	32,1	sgd	sigmoid	1.366957805	1.169169707
64,32	***32,1***	***sgd***	***tanh***	***1.29408119***	***1.137576894***

I noticed that the best results Ire when I used two LSTM layers (64, 32) two dense layers, neuron count to be (32, 1) with the optimizer to be sgd and activation layer to be tanh.

```
Final score (MSE): 1.2940811897438231
Final score (RMSE): 1.1375768939917086
```



C. *Fully connected Neural Network*

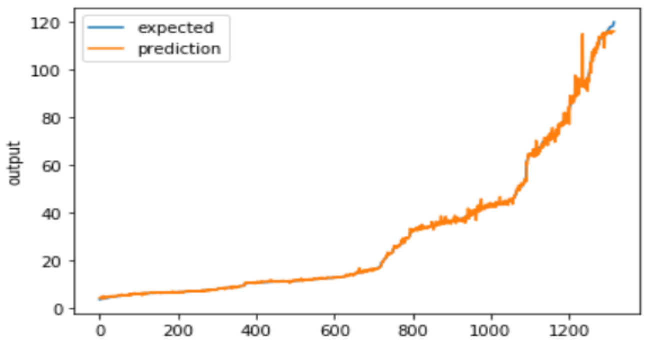
Is a series of algorithms that endeavours to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. Neural networks can adapt to changing input; so, the network generates the best possible result without needing to redesign the output criteria.

I took 2D data with 35 input features for both train and test sets to predict the stock price. I trained the model with different combinations of the activation layers and optimizers.

Dense Layers	Optimizer	Activition Function	MSE	RMSE
64, 32, 32, 1	adam	relu	1.4974044	1.223684764
64, 32, 32, 1	***adam***	***sigmoid***	***1.0311748***	***1.015467774***
64, 32, 32, 1	adam	tanh	1.443351373	1.201395594
64, 32, 32, 1	sgd	relu	7349.025543	85.72645766
64, 32, 32, 1	sgd	sigmoid	1.871782105	1.36813088
64, 32, 32, 1	sgd	tanh	86.18869072	9.283786443
64, 32, 1	adam	relu	1.359909252	1.16615147
64, 32, 1	adam	sigmoid	1.03962418	1.019619625
64, 32, 1	adam	tanh	1.326262834	1.151634853
64, 32, 1	sgd	relu	4263.087564	65.29232393
64, 32, 1	sgd	sigmoid	2.233770812	1.49458048
64, 32, 1	sgd	tanh	52.66792614	7.257267126
64, 32,32,16,1	adam	relu	1.700206542	1.303919684
64, 32,32,16,1	adam	sigmoid	1.06710527	1.033007875
64, 32,32,16,1	adam	tanh	1.416016291	1.189964827
64, 32,32,16,1	sgd	relu	6439.758106	80.24810344
64, 32,32,16,1	sgd	sigmoid	30.65600053	5.536786119
64, 32,32,16,1	sgd	tanh	80.89689206	8.994269957

From the analysis above, I understood that the best model was when I used 64, 32, 32, 1 dense layer combination, Adam as the optimizer and sigmoid as the activation layer.

```
Final score (MSE): 1.0311747997427967
Final score (RMSE): 1.0154677738573474
```



IV. Additional Features

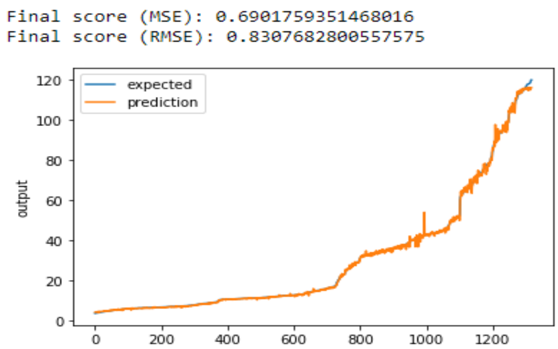
A. Best 'N' value

I used the fully connected neural networks to build a predictive model to find the best 'N' value i.e the number of days for which I could get the best result of the model.

I varied the number of days by varying the sequence size in the project. I used 64, 32, 32, 1 dense layer combination for adam optimizer and sigmoid activation.

DAYS	MSE	RMSE
1	***0.690175935***	***0.83076828***
2	0.714363733	0.84520041
3	0.715637465	0.845953583
5	0.748785738	0.865324066
6	0.859081499	0.926866495
7	1.0311748	1.015467774
4	1.327388427	1.152123443

From the analysis above, I found that the best result was yield when I calculated the stock prices on daily basis.



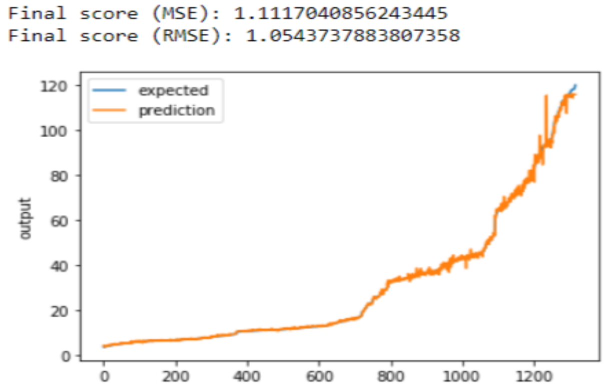
B. Bidirectional LSTM

Bidirectional RNNs connect two hidden layers of opposite directions to the same output. By this form of deep learning model, the output layer can get information from past and future states simultaneously.

I applied the bidirectional to the two LSTM layers (64, 32), two dense layers, neuron count to be (32, 1) with the optimizer to be adam and activation layer to be relu, sigmoid, and tanh. I compared the results of the bidirectional LSTM and regular LSTM and got a reduce in error for the model.

Optimizer	Activation Function	Regular LSTM RMSE	BiDirectional LSTM RMSE
adam	relu	1.987342991	1.221775948
adam	sigmoid	1.369564373	1.054373788
adam	tanh	1.270244708	1.061802808

I found out that the adam optimizer and sigmoid activation function gave the most accurate result for the BiDirectional LSTM.

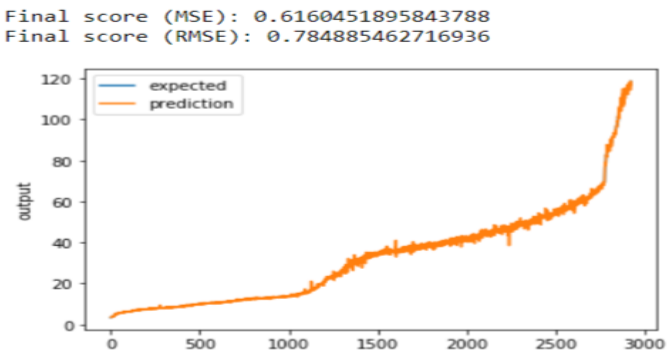


C. Fully Connected Neural Network on JPMorgan data

I used another data of JPMorgan to detect the RMSE using the fully connected neural networks. I used 64, 32, 32, 1 dense layer combination and Adam,sgd as the optimizers and relu,sigmoid,tanh as the activation layers.

Dense Layers	Optimizer	Activition Function	MSE	RMSE
64, 32, 32, 1	***adam***	***relu***	***0.61604519***	***0.784885463***
64, 32, 32, 1	adam	sigmoid	0.661066874	0.813060191
64, 32, 32, 1	adam	tanh	0.661023466	0.813033496
64, 32, 32, 1	sgd	relu	36967.44549	192.2692006
64, 32, 32, 1	sgd	sigmoid	0.837718337	0.915269543
64, 32, 32, 1	sgd	tanh	4.635555863	2.153034106

Upon analysis, I found that I got the best results with adam as optimizer and relu as activation function.



V. Project Reflection

Project Reflections

- I Initially took lot of time to figure out how to convert or reshape into required format like 2D,3D
- I learnt how to handle time-series data and use it for CNN model

VI. Challenges

- Initially I took lot of time to figure out how to convert or reshape into required format like 2D,3D.
- During our analysis a few models display loss value as nan, on performing some research I understood that it happened when loss value exceeds the bound value it

displays as nan. The solution to this is, a few layers are to be removed and the neuron count is to be decreased. Both in neural networks and LSTM.

VII. CONCLUSIONS

For the time series Forecasting, I found that **fully connected neural networks** with 64, 32, 32, 1 dense layer combination was the most accurate among the Neural Networks, LSTM, and CNN.