International Journal of Research in Advent Technology, Special Issue, March 2019
E-ISSN: 2321-9637
Available online at www.ijrat.org

Stock Market Prediction using Regression and Tensor Flow

Arpit Verma[1], Gaurav Raj[2]
[*]Department of Computer Science and Engineering
Jaypee University
Anoopshahr
arpitv066@gmail.com[1]

Abstract –Stock market prediction is a problem that has the potential to be worth billions of dollars and is researched by the largest financial corporations in the world. Everyday more than 5000 trade companies enlisted in Bombay stock Exchange (BSE) offer an average of 24,00,00,000+ stocks, making an approximate of 2000Cr+ Indian rupees in investments. Thus analyzing such a huge market will prove beneficial to all stakeholders of the system. This paper allows technique for stock market prediction including acquiring and analyzing a large data set and technique to train the program and predict potential outcomes.

Keywords: Machine learning, review paper, Tensor flow, Pandas ,Numpy, linear regression, types of programming languages for machine learning, types of libraries for machine learning, types of libraries for graphing, types of libraries for analysis.

1. INTRODUCTION

Predicting a stock market is very difficult because of its complexity and various factors which determines price changes. This is done in one of the ways that is using regression and tensor flow. It is apparent that the features that are used here are not truly independent, we know that volume and shares are not independent as well as the closing price and the return on investment not being independent. This is an assumption that we are making to simplify the model in order to use the chosen regression models. This research concerns closing prices of stocks, that's why day trading was not modeled. The model for the stock market was only concerned with the closing price for stocksat the end of a business day.

2. RELATED WORK

E. Guresan [11] evaluates the effectiveness of neural network models which are known to be dynamic and effective in stock-market predictions. The models analysed are multi-layer perceptron (MLP), dynamic artificial neural network (DAN2) and the hybrid neural networks which use generalized autoregressive conditional heteroscedasticity (GARCH) to extract new input variables. The comparison for each model is done in two view points: Mean Square Error (MSE) and Mean Absolute Deviate (MAD) using real exchange daily rate values of NASDAQ Stock Exchange index.

Y. Kara attempted to develop two efficient models and compared their performances in predicting the direction of movement in the daily Istanbul Stock Exchange (ISE) National 100 Index. The models are based on two classification techniques, artificial neural networks (ANN) and support vector machines (SVM)..

A wide range of researches has been performed on the MNIST database to explore the strength and weakness of the best recommended methodology. The best approach till date gives the training accuracy of 99.81% by Convolution Neural Network for feature extraction and used RBF network model for prediction of the handwritten digits. Research was extended at Concordia University for predicting handwritten digits.

99.17% accuracy was found on data set by preprocessing of data by Mexican hat wavelet transformation technique. This accuracy is lesser from the earlier one where the data was not preprocessed and the architecture was same.

3. PROPOSED METHODOLOGY

Tensor Flow

TensorFlow is an open source software library for numerical computation using data-flow graphs. It was originally developed by the Google Brain Team within Google's Machine Intelligence research organization for machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.TensorFlow is cross-platform. It runs on nearly everything: GPUs and CPUs—including mobile and embedded platforms—and even tensor processing units (TPUs).

TensorFlow is a great piece of software and currently the leading deep learning and neural network computation framework. It is based on a C++ low level backend but is usually controlled via Python. TensorFlow operates on a graph representation of the underlying computational task. This approach allows the user to specify mathematical operations as elements in a graph of data, variables and

operators. Since neural networks are actually graphs of data and mathematical operations,thats why it is quite useful.

3.1 Dataset

The dataset that was used was collected from the Kaggle.com as a collection of values where each row contains a stock on a specific day along with data on date, opening price, closing price, high price of stocks, low price of stocks, volume exchanged, change percentage and other features for that day.

Various scientific computing libraries of python was used along with data analysis library pandas. Pandas and numpy was used to import the data and manipulate the data. The data can be manipulated without these libraries too but these libraries make things easier.

3.2 Understanding Datasets:

The dataset that we used is data of gold stocks. The attributes that database contained was date, opening price of stocks ,end price of stocks ,highest price of stocks ,lowest price of stocks, volume exchanged ,change percentage of difference between start and end value. The end price, open price and amount of volume exchanged were the important attributes for the model. The other attributes like date, highest price and lowest price of stock on that particular day were ignored .The change percentage can also be ignored because a new change percentage will be calculated.

3.3 Importing and Formatting data:

In the dataset, at the end of volume there is K ,and it was desired to convert all these attributes to float, so to get rid of that K was removed from the volume. Similarly ,all of the prices was containing commas so to convert into float, these commas were removed. So, at the end there was bunch of float arrays(float array of end price, opening price, float array of volumes) and after that float array of price differences was created. Pandas and numpy libraries were used to import the data and manipulating the data, matplotlib library was used to plot data on some actual graphs, tensorflow is going to provide all the means to build the computational graph for training and testing the data.The outcome of the model will be based on price differences i.e.difference of next day's opening price and current day's final price. If the result will come as positive number that means stock prices will go up and it should be buy and sell at the beginning of next day.

3.4 Computational Graph

y=W*x + b

W and b will be some kind of loss variable that was tried to minimise using change in W and b.

3.5 Cost function

The cost function of the network is used to generate a measure of deviation between the network's predictions and the actual observed training targets. For regression problems, the mean squared error (MSE) function is commonly used. MSE computes the average squared deviation between predictions and targets. Basically, any differentiable function can be implemented in order to compute a deviation measure between predictions and targets.

```
loss=tf.reduce_sum(model output - real value)
```

3.6 Optimizer

The optimizer takes care of the necessary computations that are used to adapt the network's weight and bias variables during training.It indicate the direction in which the weights and biases have to be changed during training in order to minimize the network's cost function.

```
opt = tf.train.AdamOptimizer().minimize(loss)
```

Here the Adam Optimizer is used, which is one of the current default optimizers in deep learning. Adam stands for "Adaptive Moment Estimation". Adam can be looked at as a combination of RMS prop and Stochastic Gradient Descent with momentum. It uses the squared gradients to scale the learning rate like RMS prop and it takes advantage of momentum by using moving average of the gradient instead of gradient itself like SGD with momentum. Adam is an adaptive learning rate method. It computes individual learning rates for different parameters. Its name is derived from adaptive moment estimation, and the reason it's called that is because Adam uses estimations of first and second moments of gradient to adapt the learning rate for each weight of the neural network.

3.7 Initializers

Initializers are used to initialize the network's variables before training. Since neural networks are trained using numerical optimization techniques, the starting point of the optimization problem is one the key factors to find good solutions to the underlying problem.Here,the global variable initializer was used.

3.8 Fitting the neural network

After definition of placeholders, variables, initializers, cost functions and optimizers of the network, the model was trained.A sampled data of input flows (X) through the network until it reaches the output layer. There, TensorFlow compares the models predictions against the actual observed targets (Y) . Afterwards, TensorFlow conducted an optimization step and updated the networks parameters. After updated the weights and biases, the next batch is sampled and the process repeats itself. The procedure continues until all batches have been presented to the network. One full sweep over all batches is called an epoch. Total 100 epochs were used here.

After training,(y=W*x+b) W and b got trained values. During the training, predictions were evaluated on the test set—the data which is not learned and visualized it.

Fig 1:Price difference vs Volumes

4. RESULT

y=W*x+b
After training, the trained values of W=0.08 and for b=1.27 came.On passing different values of x that is volume exchanged on that particular day ,the results that came are:

x	y
200	-0.3
300	-1.07
400	-1.85

Fig 2:Price difference Vs Volumes in boo form

The accuracy is the criteria for assessment of the performance. The accuracy rate is given by

$$\frac{Number\ of\ \ correct\ predictions}{Total\ Number\ to\ actual\ values} * 100$$

The Accuracy is 61.9% on the test data in 100 epochs as shown in Fig 3.

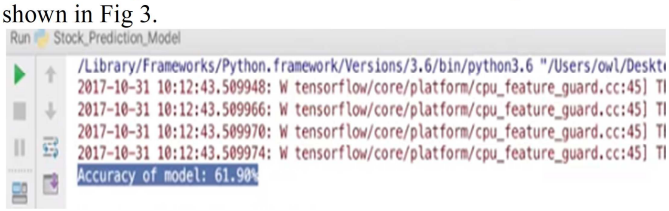

Fig 3: The Output Screen

5. CONCLUSION

The release of TensorFlow was a landmark event in deep learning research. Its flexibility and performance is beneficial for researchers to develop all kinds of complex neural network architectures as well as other ML algorithms..The proposed system was proved efficient with an overall accuracy of 62%. The accuracy is not the highest but it is satisfactory as it is very complex to predict the stock market.

REFERENCES

[1] P.PANDI SELVI, Dr.T.MEYYAPPAN [2012] P Pandi Selvi et al ,*Int.J.Computer Technology & Applications*,Vol 3 (6), 1939-1944

[2] Saeed AL-Mansoori [2015] *Int. Journal of Engineering Research and Applications ISSN* : 2248-9622, Vol. 5, Issue 5, (Part -3), pp.46-51

[3] Handwritten digit recognition using neural network approaches and feature extraction techniques: A survey Ankita Mishra, D.S.Singh,Dept. Of CSE, MMMUT Gorakhpur

[4] Handwritten Digit Recognition:
A Neural Network Demo
Berend-Jan van der Zwaag
Euregio Computational Intelligence Center
Dept. of Electrical Engineering
University of Twente
Enschede, the Netherlands

[5] Saleh Ali K. Al-Omari, Putra Sumari, Sadik A. Al-Taweel and Anas J.A. Husain *Journal of Computer Science* 5 (6): 427-434, 2009 ISSN 1549-3636
© 2009 Science Publications

[6] Y.Le Cuu, B.Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard and L.D. Jackel,"[1990] Handwritten Digit Recognition with a Back-Propagation Network," *AT&T Bell Laboratories, Holmdel*, N.J.07733.

[7] Daniel Cruces Alvarez, Fernando Martin Rodriguez, Xulio Fernandez[1998] Hermida,"Printed and Handwritten Digits Recognition Using Neural Networks", *E.T.S.I.T. Ciudad Universitaria* S/N.36200 Vigo.SPAIN.

[8] Dewi Nasien, Siti S. Yuhaniz, Habibollah Haron,"[2010] Recognition of Isolated Handwritten

Latin Characters Using One Continuous Route of Freeman Chain Code Representation and Feed forward Neural Network Classifier", World Academy of Science, Engineering and Technology 67 2

[9] Calin Enachescu, Cristian-Dumitru Miron,"Handwritten Digits Recognition Using Neural Computing", Scientific Bulletin of the Petru Maior University of Tirgu Mures, Vol.6(XXIII), 2009, ISSN 1841-9267.