USING NEURAL NETWORK TO PREDICT STOCK PRICE: USING SP500 AS AN EXAMPLE

YITIAN SUN

Canterbury School, New Milford, CT, USA
E-mail: ysun20@cbury.org

Abstract - The study uses multilayer neural network to predict the stock price and tests whether the stimulated prediction is aligned with the actual stock returns through mean squared error (MSE) value. Acquired from Yahoo Finance, S&P 500 stock data ranged from January 2002 to July 2019 are the sample for the study. Through comparing the MSE value obtained from multilayer neural network with those of other machine learning forecasting methods, random forest regression, gradient boosting, and support vector machine (SVM), the results demonstrate the former method to be the most effective.

Keywords - Machine Learning, Mean Squared Error, Multilayer Neural Networks, Stock Price Prediction

I. INTRODUCTION

Many papers have proven that stock market prediction is hard to achieve accurately. Early papers compared stock markets trend with Brownian movement which changes consistently over small period of time [1]. In 1827, Robert Brown discovered this molecular motion in which the little molecules move to all directions randomly [2]. When he put pollen particles floating on water surface under microscope, their constant “jittery motion” made Brown to be confused and wonder if the particles were alive. After analyzing more than 100 experiments, Brown failed to observe any trend of the particles’ movements. Using probabilistic model, Albert Einstein explained particles in Brownian motion to have random strength and move in random direction [3]. Some researchers had employed Geometric Brownian motion model into stock price predict, but the outcome was not pleasing. The stimulated result was slightly over 50% accurate with the real trend, though the researcher claimed that the odds would go up if the portfolios were formed [4].

In statistic, former data and linear combinations will occur repeatedly. However, the timing for buying and selling stocks are not linear, so statistical models cannot accurately predict the trend of stock price in the future [5]. Since the invention of the Neural Network in 1958 by psychologist Frank Rosenblatt, scholars have been trying to apply the network outside the biology field to predict the future financial nonlinear trend of different stocks. Biological Neural Networks are systems that consist neurons, cells, synapses and so on. The dendrites receive the input signals and output them through the axon terminal. Artificial Neural Networks (ANN) are inspired by animal rain models. The nodes and units which use machine learning to process data model the neurons from the Biological Neuron Network. The hidden layers which compose the units and nodes decide the complexity of the neural network. Since the complex relationship between financial and other input variables, regression tools like Artificial Neural Network (ANN) are proved to be competent to benefit the business field. ANN is applied to do Bankruptcy prediction [6]-[8], credit risk assessment [9], [10], and security market applications. The study will focus on its application to predict stock price.

Compare with other methods

Random forests method is a simple and powerful learning tool that can both analyze qualitative data and quantitative data. It can accomplish two kinds of prediction, classification and regression. Although the individual tree is weak, hundreds or thousands of them together can form a robust, accurate predictive method. The mode of the trees’ prediction results will be deemed as the tar-get output for classification task, while average number of all the trees’ outcomes is the final result for regression problem. Random forest does a great job at classification tasks such as face identification and the number of the recognition but performs poorly when doing extrapolation.

Although SVM is also a type of machine learning method that mainly deal with classification, it can also solve linear and nonlinear regression problems. The hyperplane which is used to separate the qualitative differences between the classes in classification issues is employed to predict the target value. The boundary lines separate the training data and the test data. The distance be-tween the hyperplane line and the two boundary lines are the same. In python, the model applies the SVR module to predict the target value and the least squared difference equation to calculate the MSE value. Gradient boost involves decision trees just like random forest does. Similar with AdaBoost, gradient boost’s trees are much bigger than simple stumps. Typically, gradient boost has 8 to 32 leaves. The model first takes the average value of the sample and subtract each value in training dataset by the mean value. After getting the pseudo residuals.

The following paper will divide in three parts. Section 2 presents the model of the multilayered neural network. Section 3 describes the result and the last section demonstrates the conclusion.

II. MODELS

In this study, we adopt data from the S&P 500 drew from Yahoo Finance. Through the data reader mode on python we directly read the data from the website and select the

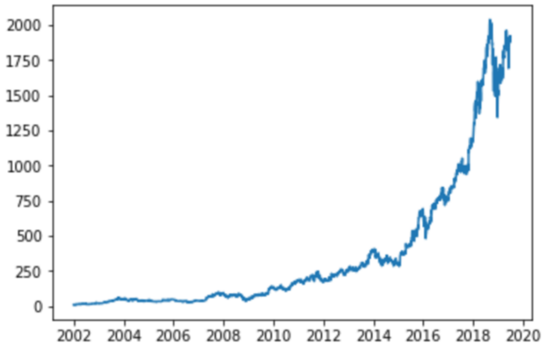

Figure 1: Time series plot of S&P 500

historical data from January 1st, 2002 to present day (July 1st, 2019), in total 4404 rows of data. Since we want the data to accurately reflect the stock value after any corporate actions, we put the adjusted closing price value for each day on the x-axis to represent the value of the stock. The graph shown above (Figure 1) is the time series plot of S&P 500 stock. In order to prevent the overfitting problem in which the model only fits the S&P 500 data, we separate the sample into two parts, training data and test data, to enable the model to do prediction outside this sample. For example, the model is like a T-shirt and the datasets it is going to fit are the people who are going to wear the T-shirt. To make the size of the T-shirts to be appropriate for everyone, the clothes should not fit one of the guys too much because none of the ten guys have completely similar figures. Similarly, the goal of the training process is to let the model to fit both training data and test data, just like the T-shirt is designed to fit all the ten guys. 80 percent of the data are denoted as training data while the other 20 percent are denoted as test data. Training data rages from January 2002 to January 2016; test data starts from January 2016 and ends in July 2019.

The magnitude of the stocks is different. For instance, the price of a S&P 500 index is about 2000, whereas some general electronics' stocks cost 7 dollars. Data scaling comes into play because we do not want the data with big magnitudes to influence our model. There are two ways to set the data to the same magnitude. We adopt the second way which is called the Min-Max scaler. Using each data value (X) to subtract the minimum value (X_{min}) of the sample, we then divide the data by the difference between the maximum value (X_{max}) and the minimum value. Through this method, the input data (X_{norm}) all have the magnitude between 0 and 1. In our code, we scale the data with Sklearn's Min-Max Scaler which works in the way above. The calculation for scaling data is applied for both training data and test data.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

With the help of TensorFlow, we establish placeholders to leave spaces for the input and target data from the graph and places for computations when being processed in the multilayer neural network. The shape of input placeholder is two dimensional and that of output is one dimensional.

Inspired by human brain, Artificial Neural Networks (ANN) consist of networks of "neurons" which are organized as layers. ANN is affiliated to deep learning which can establish relationship between pixels in an image to letters, numbers, and even human names. With enough amount of data, we can correlate between the current events with what is most likely to happen in the future. ANNs do not care about time. However, deep learning may read a string of inputs and predict the number most likely to occur next when given time series. A simple ANN (Figure 2) has only two layers, the input layer and the output layer. The two numerical variables, independent variable X and dependent variable Y form a linear relationship in this case.

A typical ANN (Figure 4) that are used to predict nonlinear trends includes hidden layers which the nodes in them receive predictors and further train them fulfill some directed task. Before this, each neuron in layers holds a number ranged from 0 to 1. For example (Figure 3), in this number recognition network, system is trained to recognize the number 7. The 28*28 picture can be separated into 784 little square pixels. Entirely colored pixel is represented by number 1, and completely white pixel is represented by number 0. Numbers from 0 to 1 which are denoted to the neurons are called "activations". All these 784 activations make up the first layer and are all connected to the next hidden layer which is designed to train the data. These activations decide which neurons in the next layer will be triggered. The signaled number at the output layer

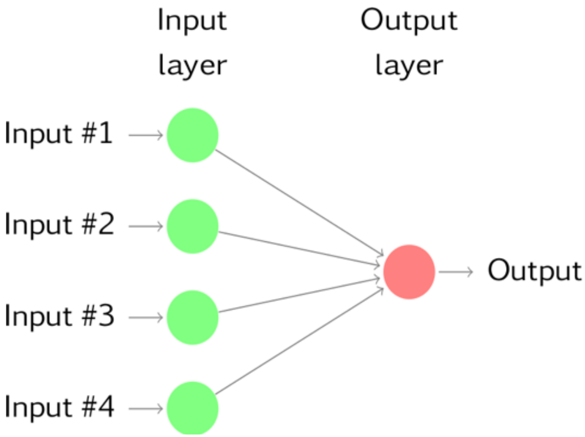

Figure 2: Simple ANN example

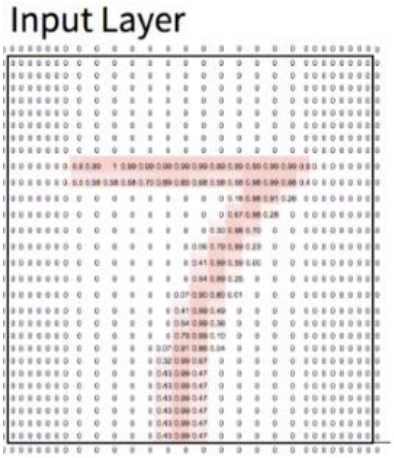

Figure 3: Number recognition network indicates the final result of the system.

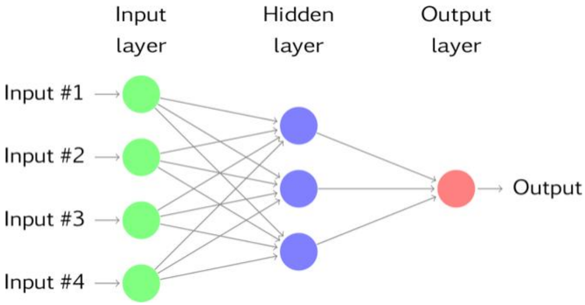

Figure 4: Structure of neural network

The structure of neural network requires us to set up layer by layer. The model contains four layers. The first layer consists of 1024 nodes and the subsequent hidden layers contain neurons halved the number of nodes in the previous layer which are respectively 512,256, and 126. Then, in order to optimize the variables, weights and bias which are parameters for the model are set up in the 4 hidden layers. Inputs are multiplied by weights and added to bias value before entering the next layer. After the weights are optimized in each layer, the overall multilayer neural network is optimized.

In regression problems, the cost function is used to calculate the difference between the prediction value and the actual training value. The mean squared error (MSE) is computed by taking the squared deviation between the output values from the model (y_{pred}) and the target values (y_{exp}). The prediction is more accurate when MSE is the smallest. To further minimize the MSE value, an optimizer is applied to the model. (Figure 3) It stimulates sets of calculation called gradients which can whether the direction of the weights and bias is positive or negative. In our code, Adaptive Moment Estimation (Adam) Optimizer is employed.

$$\mathrm{MSE} = \frac{\sum (y_{\mathrm{pred}\,.} - y_{\mathrm{exp}\,.})^2}{\mathrm{M}} \qquad (2)$$

The next step is to train and fit the neural network. After making the session and setting up the interactive batch, we randomly shuffle the indices through permutation during mini batch training in order to eliminate the rate of coincidence. The different permutation result yields different data. Then we use training data to fit the data batch by batch. Then, we run the initializer to activate Ad-am optimizer to decrease the MSE value. Finally, we ran the prediction and get the out of sample value through graphs. By comparing the prediction value and the test data and the MSE value, we can see how well our model works (Figure 5).

III. EXPERIMENTAL RESULTS

The four prediction methods all acquire excellent MSE value because all of them are under 0.3. However, the value for multilayered neural network is way lower than that of the other three.

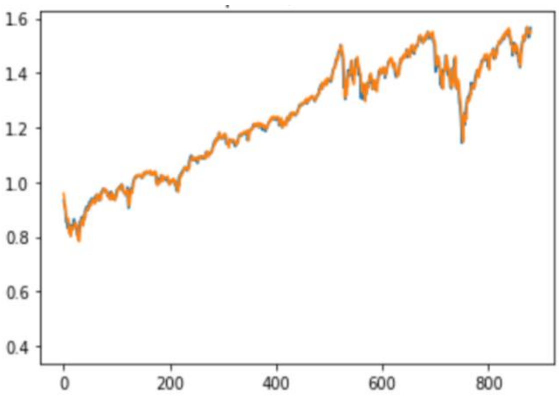

Figure 5: Predicted results

	Multilayer Neural Networks	Random Forest	SVM	Gradient Boosting
MSE	3.66×10^{-4}	1.42×10^{-1}	2.91×10^{-1}	9.95×10^{-1}

Table 1: MSE results

IV. DISCUSSION

The result of the study demonstrates the effectiveness and high accuracy of multilayered neural network when perform nonlinear stock price prediction. Other than the method, gradient boosting performs better than random forest and SVM.

Certainly, the study has some limitations. In order to perform the research more thoroughly and prove the priority of the multilayered neural network among the various machine learning methods, the study can include more techniques like linear regression, LSTM neural network, and recurrent neural network and compare them with multilayered neural network when per-forming prediction to the stock price. Since the limitation of time and resources, the study does not apply multilayered neural network to other stock than S&P 500. Future work for the study includes increasing more deep learning methods and applying multilayered neural network to more stocks.

V. CONCLUSION

The goal of the model is to fit the training data with the test data. As shown in Figure 5, the orange line which represents the training data is gradually overlapping the blue line which represents the test data. The gap between them is getting smaller and smaller which indicates the process of optimizing the prediction. Since the prediction is already close, as demonstrated in the graph, I did not include other features such as LSTM to improve the accuracy of the model. By using the model, we can predict the what will happen to the price of the stock in the next point-in-time. For instance, if the model prediction indicates the price is going up, people can buy in the stocks to benefit them-selves; and vice versa, if the model prediction indicates the price will go down, people can sell the stocks they hold to prevent losses. Thus, the model formed by neural network can benefit people's trading strategy.

REFERENCE

[1] A. Ermogenous, "Brownian Motion and Its Applications In The Stock Market," Undergraduate Mathematics Day, Electronic Proceedings, 15, 2005.

[2] R. Brown, "A brief account of microscopical observations on the particles contained in the pollen of plants and on the general existence of active molecules in organic and inorganic bodies," Philosophical Magazine, 4, pp. 161-173, 1828.

[3] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka, "Stock market prediction system with modular neural networks," 1990 IJCNN international joint conference on neural networks, pp. 1-6, IEEE, 1990.

[4] P. Hájek, "Municipal credit rating modelling by neural networks," Decision Support Systems, 51(1), pp. 108-118, 2011.

[5] G. E. Hinton, "Learning multiple layers of representation." Trends in cognitive sciences 11, no. 10, pp. 428-434, 2007

[6] E. Alfaro, N. García, M. Gámez and D. Elizondo, "Bankruptcy forecasting: An empirical comparison of AdaBoost and neural networks," Decision Support Systems, 45(1), pp. 110-122, 2008.

[7] K. Lee, D. Booth and P. Alam, "A comparison of supervised and unsupervised neural networks in predicting bankruptcy of Korean firms," Expert Systems with Applications, 29(1), pp. 1-16, 2005.

[8] J. Baek, S. Cho, "Bankruptcy prediction for credit risk using an auto associative neural network in Korean firms," Proceeding of the International Conference on Computational Intelligence for Financial Engineering, IEEE Press, pp. 25-29, 2003.

[9] H. Qu, G. Tang and Q. Lao, "Oil Price Forecasting Based on EMD and BP_AdaBoost Neural Network," Open Journal of Statistics, 8(04), p.660, 2018.

[10] E. Angelini, G. di Tollo and A. Roli, "A neural network approach for credit risk evaluation," The quarterly review of economics and finance, 48(4), pp.733-755.

★★★