IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

STOCK PRICE PREDICTION

Dakshita Jena[#1], Krutika Bhutada[#2], Pranjal Ghalme[#3], Reetu Sharma[#4], Kirti Wanjale[#5]

[#]Computer Department , Vishwakarma Institute of Information Technology

Survey No. 3/4, Kondhwa (Budruk) Pune – 411048, Maharashtra (India)

Abstract - Tinging stock market prices is a complex task that has traditionally involved extensive human-computer interaction. Due to the correlation of stock prices, traditional batch processing methods cannot be used effectively for stock market analysis. We propose an online learning algorithm that uses a type of repetitive neural network (RNN) called long-term memory (LSTM), where weights are adjusted for random gradient descent. It provides more accurate results than existing stock price estimation algorithms. Networks are trained and evaluated for accuracy with different sizes of data and the results are tabulated. Accuracy has been compared with the artificial neural network.

I. INTRODUCTION

Investment firms, hedge funds and individuals are also using financial models to better understand market behavior and make profitable investments and trades. Database is available in the form of historical stock prices and company performance data suitable for processing machine learning algorithms.

Can we really estimate stock prices with machine learning? Investors make educated predictions by analyzing the data. They read the news, assess the history of the company, industry trends and other data points. The current theory is that stock prices are completely random and unpredictable, but this raises the question of why top companies such as Morgan Stanley and Citigroup are using quantitative analysts to create forecasting models. On the trading floor filled with adrenaline, we have the idea that men with loose relationships are walking around screaming something on the phone, but these days they can be seen sitting quietly in front of a computer screen and rows of machine learning professionals. In fact, almost 70% of all orders on Wall Street are now delivered through software, and we now live in the age of algorithms.

Repetitive neural networks (RNNs) are useful for timing data, but recent research has shown that LSTMs are the most popular and useful versions of networked RNNs. Reduces regression error and helps LSTM [3] [4] to remember data and results longer. Finally, graphs are plotted for price fluctuations with dates (in the case of regression-based models) and between true and value-laden values (for LSTM-based models). Often, the forecast is random rather than confusing, which means it can be assessed by carefully analyzing the relevant stock market history. Machine learning is an effective way to refer to such processes. It estimates the market value close to the apparent value, thereby increasing accuracy. The introduction of machine learning in the field of stock prediction has attracted many researches due to its effective and accurate measurement [1].

The dataset used is an important part of machine learning. The dataset should be as concrete as possible because a small change in the data will result in a large-scale change [2]. This dataset consists of the following five variables: open, closed, low, high and volume. Open, Close, Low and High are almost bid prices for stocks at different times with almost direct names. Volume is the number of shares that went from one owner to another. The model is tested on test data

We use Keras to generate LSTM to estimate stock prices using historical closing prices and trading volumes, and to visualize the optimal value values over time and the optimal parameters for the model.

II. LITERATURE SURVEY

The initial focus of our literature survey was to explore common online learning algorithms and see if they fit our utility case, i.e. work on real-time stock price data. These include online AUC optimization [8], online transfer practice [9] and online feature selection [1]. However, since we could not find any of these optimizations for stock price estimation, we decided to look at existing systems [2], analyze its major flaws and see if we can improve on them. We have addressed the correlation between stock data (dynamic, long-term interdependence between stock prices) as the main problem we want to solve. A brief search for simple solutions to the above problem led us to RNN [4] and LSTM [3]. After deciding to use the LSTM neural network to estimate stocks, we considered several papers to stu dy the concept of gradient descent and its variants.

A. Algorithm and Technique

The goal of this project is to study time series data and explore more options to accurately estimate the stock price. Through our research we learned about recurrent neural nets (RNN) 8, which are used exclusively for learning scenes and patter ns. They carry information from the looped networks and therefore have the ability to remember the data accurately. Recurre nt neural nets have a disappearing gradient decent problem that d oes not allow them to learn in a way that is different from previ ous data. This problem is solved in long-term memory network 9, commonly known as LSTM. They have unique DNN, long-ter m dependency learning capabilities.

In addition to adjusting the structure of the neural network, a complete set of the following parameters can be tuned to optimize the prediction model.

B. Benchmark Model

We used the Linear Regression model as its primary benchmark for this project. As such, one of our goals is to u nderstand the performance and implementation differences bet ween machine learning versus deep learning models. It is based on the examples presented in the Linear Register Udacity's Machine Learning for Trading course.

The following are the estimated results We obtained from our benchmark model, as shown in Fig 1.

Fig 1: Green line: Adjusted Close price Blue Line: Predicted Close price

Train Score: 0.1852 MSE (0.4303 RMSE)

Test Score: 0.08133781 MSE (0.28519784 RMSE)

Input parameters:

- Preprocessing and Generalization (see Data Processing section)

Neural Network Architecture:

- Number of layer layers (how many layer nodes in the model; 3 used)
- Number of nodes (how many nodes per layer; 1,3,8, 16, 32, 64, 100,128 tested)

Para training parameters:

- Training / Test Separation (How many datasets for training versus test models; fixed at 82.95% and 17.05% for benchmark and LSTM models)

Certification set (fixed at 0.0 5% on training set)

- Batch size (how many steps should be included in the training phase; 1 will be kep t for the basic lstm model and 512 for the better lstm m odel)
- Optimizer function (which is used to optimize by minimizing errors)

C. LSTM Model

LSTM is a special subset of RNNs that have long-term context-specific temporal dependencies. Each LSTM neuron is a memory cell. In normal RNNs neurons take the current input only to their previous hidden state and the new hidd en state, an LSTM neuron takes its old cell state and produces its new cell state.

An LSTM memory cell has the following three components or gates:

1. Forget the gate: The forgotten gate d etermines the replacement of specific parts of the cell state with recent information. It provides values close to 1 for part s of the cell state to be maintained and zeros for values to be ig nored.

2. Input Gate: Based on the input (i.e. past output O (t-1), input x (t) and previous cell position c (t-1)) this section of the network detects any conditions that the informa tion must be stored in the cell state (or Will be updated)

3. Output gate: Depending on the input and cell status, this component determines which information to forward (ie output O (t) and cell state C (t)) to the next node in the network.

Therefore, LSTM networks are ideal for exploring how variation in one stock price affects the prices of many other stocks in the long run. They also determine ho w long it takes to maintain information about specific past tre nds in the stock price movement (dynamically) to more accurately predict future trends in stock price changes.

The main advantage of LSTM is the ability to learn context-specific temporal dependencies. Each LSTM unit remembers information for a long or short time (hence the name) without explicitly using the activation function on repetitive parts.

The important thing to note is that any cell state is multiplied by the output of the forgotten gate, which varies between 0 and 1. That is, the forgotten gate in the LSTM cell is responsible for both the weight and activation function. Cell status. Therefore, the previous cell position information passes through an unchanging cell instead of rapidly increasing or decreasing at each time-phase or layer and the weights are converted to their correct values at reasonable intermission. This allows LSTM to solve the degraded gradient problem - since the value stored in the memory cell is not modified repeatedly, the gradient does not appear when trained with backpropagation.

We worked on LSTM's fine tuning parameters to get better estimates for this project. We improved by testing and analyzing each parameter and selecting the final value of each of them.

To improve LSTM, we did the following:

Increased the number of hidden nodes from 100 to 127. A 0.2 drop out was added to each layer of LSTM Batch size increased from 1 to 512 Ages increased from 1 to 20

Added verb = 2

Bought with batch size

This improves our average squared error for the test set from 0.01153170 MSE

0.00093063 MSE.

The estimated plot difference can be seen as follows in Fig 2 and Fig 3.

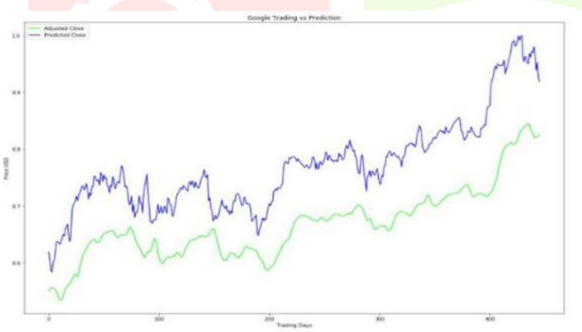

Fig 2. Plot For Adjusted Close and Predicted Close Prices of for basic LSTM model

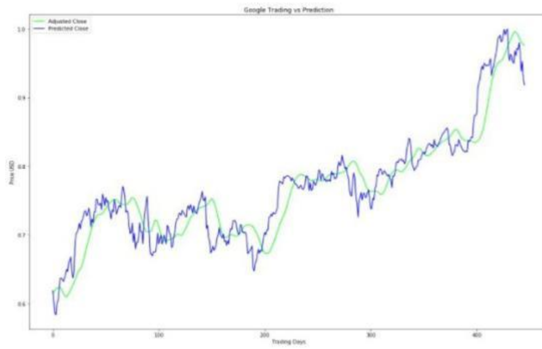

Fig 3. Plot For Adjusted Close and Predicted Close Prices for improved LSTM model

III. CONCLUSION

Comparison results between Long Short Term Memory (LSTM) and Artificial Neural Network (ANN) suggest that LSTM has better predictive accuracy than ANN.

The project seek to utilise Deep Learning Models, Long-Short Term Memory (LSTM) Neural Network Algorithm, to predict stock prices. LSTM shows more superior results over other models due to ability to assign different weights to input features hence automatically choose the most relevant features. Hence it's more able to capture the long term dependencies in the time series and more suitable in predicting financial time series. We have also used Keras to build a LSTM to predict stock prices using historical closing price and trading volume and visualise both the predicted price values over time & the optimal parameter for the model.

REFERENCES

[1] Nazar,NasrinBanu,andRadhaSenthilkumar.“Anonline approach for feature selection for classificationin big data.” Turkish Journal of Electrical Engineering&ComputerSciences 25.1 (2017):163-171.

[2] Soulas,Eleftherios,andDennisShasha.“Onlinemachinelearningalgorithmsforcurrencyexchangeprediction.”ComputerScienceDepartmentinNewYorkUniversity,Tech.Rep31(2013).

[3] Suresh,Harini, et al. “Clinical Intervention PredictionandUnderstandingusingDeepNetworks.”ArXivpreprint arXiv:1705.08498(2017).

[4] Pascanu, Razvan, Tomas Mikolov, and YoshuaBengio.“Onthedifficultyoftrainingrecurrentneuralnetworks.”InternationalConferenceonMachineLearning.2013.

[5] Zhu,Maohua,etal.“TrainingLongShort-TermMemoryWithSparsifiedStochasticGradientDescent.”(2016)

[6] Ruder,Sebastian.“Anoverviewofgradientdescentoptimizationalgorithms.”arXivpreprintarXiv:1609.04747.(2016).

[7] Recht, Benjamin, et al.“Hogwild: A lock-free approachtoparallelizingstochasticgradientdescent.”Advancesinneural informationprocessing systems.2011.

[8] Ding, Y., Zhao, P.,Hoi, S. C.,Ong, Y. S.“An AdaptiveGradientMethodforOnlineAUCMaximization”InAAAI(pp.2568-2574).(2015,January).

[9] Zhao,P.,Hoi,S.C.,Wang,J.,Li,B.“Onlinetransferlearning”.ArtificialIntelligence,216,76-102.(2014