Stock Market Price Predicting with LSTM

By Kevin Sukher

Decades ago, before complex technology and computational algorithms, investors and brokerage companies would rely on daily news outlets and basic relayed information for insight on market stock values and trends. With these tidbits of data and info, general calculations and formulas were used to forecast how market price values would be affected and ultimately look like in the future. Back then, communication was slow paced and the collection and volume of data that needed to be used for computation was miniscule in comparison. This resulted in farfetched models and assumptions on the stock market, that grew in resourcefulness over the years. Today, with the use of complex computers, artificial intelligence, machine learning algorithm and more, financial analysis has been revolutionized with increased accuracy and anticipation of future changes.

With the volatility of the stock market being unpredictable, the use of financial modeling is important in marking past trends and historical outcomes to help determine how the market will behave in the future. For this project we will look at one company stock and apply machine learning algorithm to see if we can create a model and predict the future stock market price based on that. In terms of machine learning and deep learning, there are a bountiful different methods and algorithms that can be used for this problem, some better than others. In our case, we will look at a recurrent neural network (RNN) architecture called Long Short-Term Memory (LSTM). The expectation is to input historical market data of a specific stock into the network model for it to train with, and then test to see how well the model makes a prediction based on the given information.

Before diving into the machine learning model procedure, some background is needed on the technique at hand. Artificial neural networks are computational systems, generally inspired by the human brain, that are a complex collection of interconnected nodes. Similar to connected neurons and synapses in the human brain, these nodes send signal to one another to communicate in response of feedback. In a neural network these responses, or outputs, are a result of inputs that are passed into one end of the net and perform various computations as they travel through the nodes or layers. Starting from the input layer and travelling through the net to the output layer, many transformations can occur to the input. These transformations are the result of training the network. This process consists of giving the network an input and target result that it needs to reach. Based on the data given the network runs the input through the net multiple times, adjusting and reweighing nodes every time to get the output as similar to the target variable as possible. This is just an overview of a general neural network as there are many different

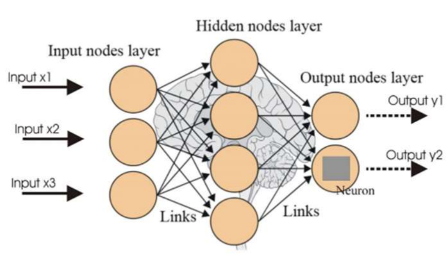

Figure 1: Depiction of an Artificial Neural Network (from "The Evolution and Core Concepts of Deep Learning & Neural Networks" by Syed Danish Ali, Aug 3, 2016, analyticsvidhya.com)

variations. In our case we will look at a type of Recurrent Neural Network (RNN), which has feedback memory nodes and layers. Compared to a typical feedforward neural network that consists of an input doing computations through to the output adjusting layers and creating different weights. An RNN has feedback that stores previous instances and shares parameters throughout the layers.

For the case of making predictions based on historical stock market prices we will use a type of RNN called Long Short-Term Memory (LSTM). LSTM is a network that learns order dependence based on sequential data and problems. Unlike other feedforward networks, LSTM has feedback connections and can process time series sequential data. Since stock prices consist of various points of data that are organized by date, this makes it a sequence of information and makes the LSTM model a great technique to process such data. LSTM are well suited for sorting, managing, and making prediction based on these inputs. The model commonly consists of three gates, input, output and forget. Unlike normal input-output networks, this one uses these three to regulate what data is needed and can be removed. This remembrance of arbitrary time intervals based on weighing requirements, makes LSTM applicable to many applications such as speech and picture recognition, segmentation, word comparison and in our case sequence prediction.

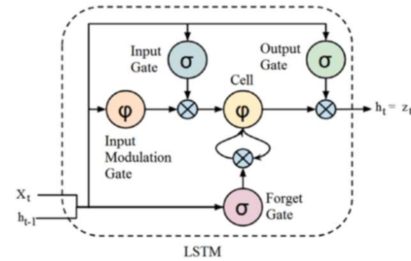

Figure 2: LSTM Cell Architecture (from "Long Short-Term Memory (LSTM): Concept" by Eugine Kang, Sep 1, 2017, medium.com)

Moving into the procedure of the model, Python code was written in Jupyter Notebook that consists of initializing, training, and testing given historical data of a stock market company. The ipynb file for this can be found in the GitHub Repository along with associated documents. In this case we will use Microsoft (MSFT), a well-known and successful corporation as our stock data set for the LSTM model. For the project, Yahoo Finance was used to obtain stock price history for Microsoft because it has a reliable interface that lets you filter the range of data to view, and download based on stock market dates. After obtaining the data, a set of only closing prices was filtered out to be used and processed. This plot from the code displays the closing prices of Microsoft from January 1, 2010 to January 1, 2020. This will be the underlying input data that will run through the LSTM as it trains and tests its own model to fit as close to the original as possible. For training, 75% of the closed market price data will be used while the other 25% will be for testing the model for an accurate prediction. Before creating the LSTM architecture and running the input data, its important and good practice to normalize any preprocessed data. After scaling and splitting the data into independent and dependent variables, it was just a matter of reshaping the set into a

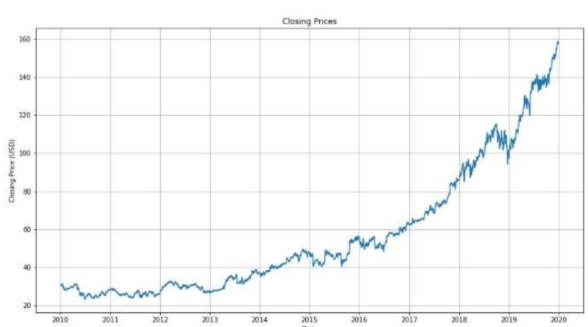

Figure 3: Closing Price Plot (from project)

three-dimensional array, so it will be accepted as an input for the LSTM model. Creating the model, for this project, it consists of 4 layers; two LSTM layers with 50 neurons each, the input having a return sequence, and two standard dense neural network layers with one having 25 neurons and the other having one as the output. Once built and completed its training run with the data, it's time for testing the model. Similar to the training data set, a testing set of the remaining 25% of the closing prices is used. It is normalized, reshaped and used for a prediction of the model. Displayed is a plot of the original and predicted closing price data. In blue it shows the initial training data, in orange the testing data and in green the prediction that the model made. By simply visualizing, it is clear to see that the model did surprising well and was very close to matching the validation. Calculating the Root Mean Square Error (RMSE) value, we approximately get 3.24. How this is interpreted is the marginal difference or error of the predicted to the original fit. The lower the number the more precise it is. When looking at the validation table of the model that compares the actual prices to the predicted prices, the results can be concluded as reliably close.

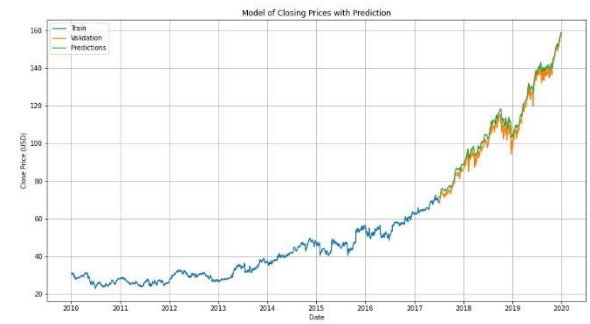

Figure 4: Predicted Closing Price Plot (from project)

Date	Close	Predictions
2017-07-03	68.169998	71.732788
2017-07-05	69.080002	71.362709
2017-07-06	68.570000	71.271782
2017-07-07	69.459999	71.170433
2017-07-10	69.980003	71.299210
...	...	...
2019-12-24	157.380005	157.351608
2019-12-26	158.669998	157.753860
2019-12-27	158.960007	158.311234
2019-12-30	157.589996	158.817291
2019-12-31	157.699997	158.833054

Figure 5: Validation Comparison Table (from project)

Based on the outcome of the model and data, it can be concluded that expectations were met for a basic introductory usage of LSTM networking. For future predictions we set up a new data set with the Microsoft closed market prices and ran it through the prediction model for the following day outside the range. The result of the prediction came out to be $158.75 for January 2, 2020 while the actual was $160.61 for the same day. This equates to a 2% margin of error between the sets for this run of the model. While improvements can be made to the code and model to increase accuracy and better predictions, it was an interesting challenge and learning experience in and of itself. Learning LSTM was fascinating as well and how powerful of an RNN technique it is for computing financial patterns and trends. But even so, not all algorithms and techniques are perfect. Many systems and methods are being improved and developed every day. Some methods that can be used to improve this model could be to increase the number of layers in the LSTM architecture, pushing the model through more training and testing runs, optimizing the code or method of LSTM modeling for better efficiency and run time, and many more. It is hard to predict when they will become perfect or to decrease the margin of error when it comes to something as volatile as the stock market. For as many years and decades have gone by with investors and big firms trying every kind of computation to predict what the market will be, we will see what technology and innovation awaits in due time for this everlasting challenge.

Sources

- Nelson, D. M., Pereira, A. C., & De Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. *2017 International Joint Conference on Neural Networks (IJCNN)*. doi:10.1109/ijcnn.2017.7966019

- Ryo Akita, Akira Yoshihara, Takashi Matsubara, and Kuniaki Uehara. Deep learning for stock prediction using numerical and textual information. In 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pages 1–6. IEEE, 2016.

- Zhou, H., Zhang, Y., Yang, L., Liu, Q., Yan, K., & Du, Y. (2019). Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism. *IEEE Access, 7*, 78063-78074. doi:10.1109/access.2019.2923006.

- Zou, Z., & Qu, Z. (2020, December). Using LSTM in Stock prediction and Quantitative Trading. Retrieved from http://cs230.stanford.edu/projects_winter_2020/reports/32066186.pdf