FUNDAMENTALS AIDED STOCK FORECASTING WITH LONG SHORT TERM MEMORY NEURAL NETWORKS

Evan Kozliner, Debanjan Nandi, Isaac Goldthwaite, Aaditya Gavandalkar

Computer Science and Engineering, The Ohio State University

ABSTRACT

We build a Long Short Term Memory (LSTM) neural network model to evaluate how such a network performs when regressing stock price data and to examine the impact of fundamentals data on the LSTM. LSTM models are often used as predictive models for temporal data due to their ability to retain information about long term pattern in data. There have been some attempts to utilize LSTMs for time series predictions like stock pricing, currency pricing, and stock volatility (citation), but it seems that few have examined whether or not financial fundamentals such as balance sheet data can impact their accuracy. The results suggest that the LSTM does not effectively utilize the fundamentals information we provided.

***Index Terms*—** Long Short Term Memory Neural Networks, Stock Forecasting, Recurrent Neural Network

1. INTRODUCTION

To analyse the price of stocks generally one of two methods are employed: technical or fundamentals analysis. Fundamentals analysis attempts to measure the intrinsic value of a stock by studying quarterly metrics such as a companys assets and liabilities, or macro trends such as how the companys sector as a whole is doing. Technical analysis generally does not incorporate these figures and instead relies mostly on figures like a securitys price and volume traded, typically represented as a time series.

The accuracy of regression techniques to predict future stock prices is paramount in technical analysis because it is used to compute a stock's future value and volatility. The volatility and return potential of a stock relative to another dictates which stock is a better purchase.

LSTM models are type of recurrent neural networks (RNNs) that can handle temporal data and be used to make time series predictions. LSTM models are of a particular interest in the domain of time series prediction because they have the capability to pick up patterns in data spanning long periods of time, unlike traditional RNNs and other temporal models such as Hidden Markov Models (HMMs). Neural networks in general have the capacity to pick up complex relationships between input data, so there is some chance that that given the right data the LSTM could pick up difficult-to-spot but significant relationships between fundamentals data and stock prices.

Any overview of stock price prediction would be incomplete without mentioning the Efficient Market Hypothesis (EMH) and its implications on the feasibility of trading systems. EMH argues that it is impossible to beat the market because the stock price is always representative of its fair value. If EMH is correct, it would nullify the point of doing stock analysis at all and suggest that it would be better to purchase index funds.

Another potential problem for the incorporation of fundamentals data in stock price analysis is the risk that stock prices actually have little to do with the intrinsic value of a company. Behavioural economics suggests that investors are subject to cognitive biases when investing in a company. These biases can cause investors to pick growth stocks that are more frequently reported on over companies with good fundamentals. This behaviour inflates the price of hot stocks and leaves many companies unnoticed. The effect of human psychology could be good or bad for trading systems depending on if the trading systems are able to pick up on human patterns of undervalues and overvaluing.

2. DATA EXTRACTION

Historical stock price data is not frequently provided minute-by-minute, instead it is provided daily and only includes the opening price of the stock, the closing price, the highest price the stock reached, and the lowest price. This data is readily available online, we used the python backtesting framework bt to obtain it.

The data prior to adding fundamentals was indexed by day and looked like the following:

	Open	High	Low	Close
2011-01-03	181.36	186.00	181.21	184.22
2011-01-04	186.14	187.69	183.77	185.00
2011-01-05	184.10	187.44	184.07	187.41

Table 1. Stock data before adding fundamentals information

Fundamentals data is can be difficult to obtain because it needs scraped off 10-K or 10-Q (yearly or quarterly) reports.

All of these reports are available through an FTP server provided by the SEA, written in a standard format known XBRL. An open source tool called ScraXBRL can extract the data from these reports into a tree structure, however manual tree search algorithms needed to be written manually to extract specific values from the report. Scripts also needed to be written to properly join fundamentals data (which only comes quarterly) with daily stock prices.

Out of the large amount of fundamentals data only Cash and Cash Equivalents, Net Assets, and Net Liabilities were selected as features for the LSTM. These features were selected because they are some of the simplest indicators of a company's financial success. Common financial ratios such as the P/E ratio would be a logical next step, however we only wanted to examine some of the most bare-bones fundamentals for our analysis.

After the addition of the fundamentals data our input vectors included below features in addition to previously mentioned stock features like this (note the fundamentals data is repeated because the same fundamentals apply for the whole quarter):

	Cash	Assets	Liabilities
2011-01-03	3.77e+09	1.879e+10	1.037e+10
2011-01-04	3.77e+09	1.879e+10	1.037e+10
2011-01-05	3.77e+09	1.879e+10	1.037e+10

Table 2. Additional fundamentals data added to input vector

3. LINEAR REGRESSION BASELINE

A common approach used when predicting stock market data is to model past data with a series of linear regressions to predict future results. As a baseline to compare our prediction model, we used a linear regression model which trained on past open values versus same-day close values. This model was then used to predict future close values given the same-day open values in our datas test set. **[insert data results here]**. We also ran tests on a linear regression model that took into account the fundamentals data, **[results + specific methodology here]**. Overall, the fundamentals data did not have a significant effect on the accuracy of the linear regression baseline model.

4. LONG SHORT-TERM MEMORY (LSTM)

Long Short Term Memory (LSTM) architecture (citation 1) uses special purpose-built memory cells to store information and is better at finding and exploiting long range dependencies in the data. Fig (insert fig no) illustrates a single LSTM memory cell. The memory cells, with self-connections storing the temporal state of the network, are key to the LSTM. Each LSTM block also contains an input gate, an output gate, and a forget gate. The input gate controls the input flow of data and determines by how much we should update each memory cell value. The forget gate determines which values of the cell state should be retained and which values to be forgotten. The output gate controls the output flow of data and determines what parts of the cell state we are going to output.

A LSTM calculates the network unit activations using the following equations iteratively from t = 1 to T:

$$ft = \sigma(Wf.[ht-1, xt] + bf) \tag{1}$$

$$It = \sigma(Wi.[ht-1, xt] + bi) \tag{2}$$

$$ot = \sigma(Wo.[ht-1, xt] + bo) \tag{3}$$

$$ct = ft \otimes ct - 1 + it \otimes \sigma(Wc.[ht-1, xt] + bc) \tag{4}$$

$$ht = ot \otimes tanh(ct) \tag{5}$$

Where denotes the Hadamard product, is the sigmoid function, i, f, o and c are respectively the input gate, forget gate, output gate and cell activation vectors, all of which are of the same same size as the hidden vector h. W and b represent the corresponding weight matrices and bias vectors for the respective gates.

5. IMPLEMENTATION

We used an embedding dimension of 64 for all the attributes of the feature vectors before using them as input to the LSTM. We used a fixed hidden state dimension of 1024 for all the LSTM modes. Additionally we use a dense hidden layer of a single neuron on top of the 2 hidden LSTM layers to determine the closing price. We used a learning rate of 0.001 and an Adam Optimizer minimizing the RMS Error for training the model for 100 epochs. The model was trained on a single GPU with Tensorflow implementation. We used a sliding window of 20 sequences. The batchsize was maintained at 128 sequences per batch

5.1. Subheadings

Subheadings should appear in lower case (initial word capitalized) in boldface. They should start at the left margin on a separate line.

5.1.1. Sub-subheadings

Sub-subheadings, as in this paragraph, are discouraged. However, if you must use them, they should appear in lower case (initial word capitalized) and start at the left margin on a separate line, with paragraph text beginning on the following line. They should be in italics.

6. REFERENCES

[1] C.D. Jones, A.B. Smith, and E.F. Roberts, “Article title,” in *Proceedings Title*. IEEE, 2003, vol. II, pp. 803–806.