

# CS 230

## A Deep Learning Approach for Stock Market Prediction

**Yan Miao**  
Computer Science Department  
Stanford University  
yanmiao@stanford.edu

### Abstract

The project explores a stock market prediction model using a LSTM network. A LSTM model with different parameters are tested to determine the effect of number of hidden layers, dropout regularization and batch size on the result accuracy. The model is tested on the stock price of Amazon, Google and Facebook.

### 1 Introduction

Financial time series are non-stationary, nonlinear and high-noise. While individuals and firms all have interests in gaining profits from the stock market, it is difficult to predict the trend of a stock's price by instinct. Despite the success of classical machine learning algorithms, the evolution of deep learning has provided new models for researchers to analyze big data with cheap computation devices. The input to my algorithm is the daily price value of a stock in a chosen time period. Then I use a long short-term memory (LSTM) recurrent neural network to output the predicted stock price.

### 2 Related Work

The deep learning neural network is a reliable predictor due to its ability to approximate nonlinear functions. There already exists many papers on this topic. For example, Artificial Neural Networks (ANN) are able to predict the stock price and direction of movement of the price [1][2]. Convolutional Neural Networks (CNN), though more frequently used for image processing, can also be used to predict the stock movement as a classification model with one day close, open, high, low, volume data [3]. In his paper, experiments on a one-dimensional convolutional network with three convolutional layers using MaxPooling and ReLu activation. In particular, LSTM neural networks have the characteristics of selectivity and memory cells that are suitable for random non-stationary sequences like stock-price time series [4]. LSTM can store memory and solve the gradient vanishing problem. The LSTM network is also the most popular model adopted right now. In a study that systematically reviews the Deep Learning models implemented for stock market forecasting, researchers find that the LSTM technique is widely applied (73.5%) [5]. In this project, an LSTM network is implemented to predict the Amazon stock price. Similar works can be found here: [6][7][8].

### 3 Dataset and Features

The experimental data comes from the data provided by Yahoo Finance. The stock price is selected from January 28, 2015 to January 29, 2020. Stocks from three companies are selected: Amazon(AMZN), Google(GOOG) and Facebook(FB). Such selection is to minimize unnecessary influences due to different industries.

	Date	Open	High	Low	Close	Adj Close	Volume
0	2015-01-28	309.809998	311.510010	303.799988	303.910004	303.910004	3063800
1	2015-01-29	304.730011	312.799988	299.329987	311.779999	311.779999	8656600
2	2015-01-30	346.320007	359.500000	340.739990	354.529999	354.529999	23856100
3	2015-02-02	350.049988	365.000000	350.010010	364.470001	364.470001	10231900
4	2015-02-03	360.290009	367.820007	360.209991	363.549988	363.549988	6212600

Fig. Sample representation of dataset

In the dataset, there are five features: the day of the transaction (date), the opening price value of the specified day (open), the closing price value of the specified day (close), the lowest price value of the specified day (low), the highest price value of the specified day (high). For the project, the goal is to predict the “close” stock price value. For each dataset, there are 1260 entries in total. The first 800 entries are selected for training, and the remaining 460 entries are used for testing. Neural networks are sensitive to unnormalized data. Before training the model, the data was rescaled into the range of [0, 1] using Min-Max normalization to make the model more reliable. Then, each row of the input matrix is a data structure with 30 time-steps. For training input, the shape is (770, 30, 1) and for testing input, the shape is (430, 30, 1).

## 4 Methods

The LSTM network model is implemented in this project. The structure of the memory unit of a LSTM is demonstrated below. The memory unit operates with input gate, forget gate and output gate. The process can be summarized as the following equations:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$\tilde{S}_t = \tanh(W_s[h_{t-1}, x_t] + b_s)$$

$$S_t = f_t * S_{t-1} + i_t * \tilde{S}_t$$

$$h_t = o_t * \tanh(S_t)$$

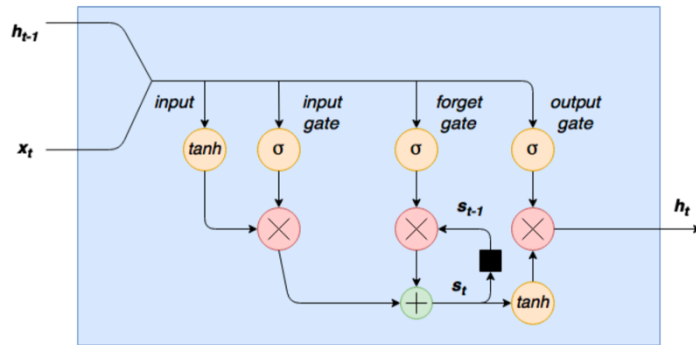


Fig. LSTM Memory Unit [9]

Here  $i_t$ ,  $f_t$  and  $o_t$  are the outputs of different gates,  $\tilde{S}_t$  is the new state of memory cell,  $S_t$  is the final state of memory cell and  $h_t$  is the final output of the memory unit.  $W_i, W_f, W_o, W_s, b_i, b_f, b_o$  and  $b_s$  are coefficients.

## 5 Experiments

The model is built with 50 neurons. For each hidden layer in the model, dropout regularization is employed. Then, the model is compiled with the MSE loss function and the Adam optimizer. Six models are tested with differences in layers, dropout rate and batch size. Models are tested with 3 and 4 hidden layers, dropout rate of 0.1 and 0.2, and batch size of 32 and 64. The first model is a LSTM network with 50 neurons, 3 layers, dropout rate of 0.2 and batch size of 32. To test if a smaller dropout rate could improve the model error, dropout is reduced to 0.1. To determine if a smaller number of layers is enough, one layer is removed from the model. Since batch size is critical for LSTM to learn the common pattern, models with different batch size are also tested.

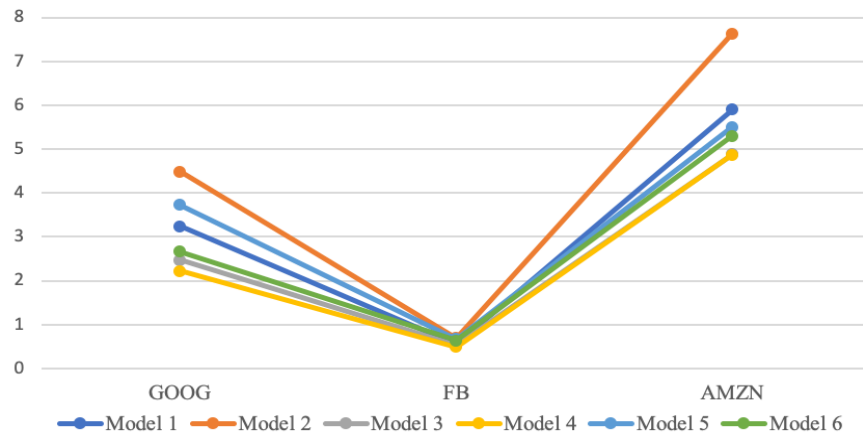
Model #	Neurons	Layers	Dropout	Batch Size	Epochs
1	50	4	0.2	32	100
2	50	4	0.1	32	100
3	50	3	0.2	32	100
4	50	3	0.1	32	100
5	50	3	0.2	64	100
6	50	3	0.1	64	100

*Table. Details on different models*

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
GOOG	32.41	44.86	24.66	22.23	37.31	26.69
FB	5.61	6.98	5.24	4.89	6.68	6.36
AMZN	58.97	76.34	48.85	48.66	55.00	53.00

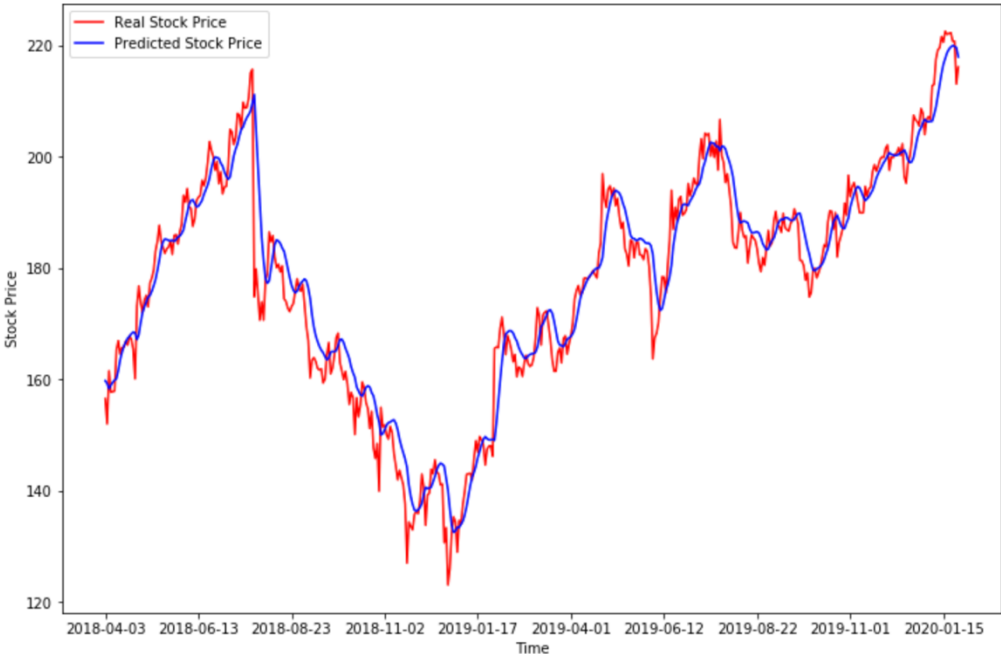
*Table. Model Performance on Three Datasets*

The metrics adopted for model performance is the Root Mean Squared Error. RMSE is commonly used to measure accuracy for forecasting. It penalizes big errors while small errors can be safely ignored. This choice of metrics is consistent with the goal of predicting stock market trend to generate revenues. Below is a plot that demonstrate the model performance. Different colors represent different models, and the horizontal axis represents three datasets respectively.

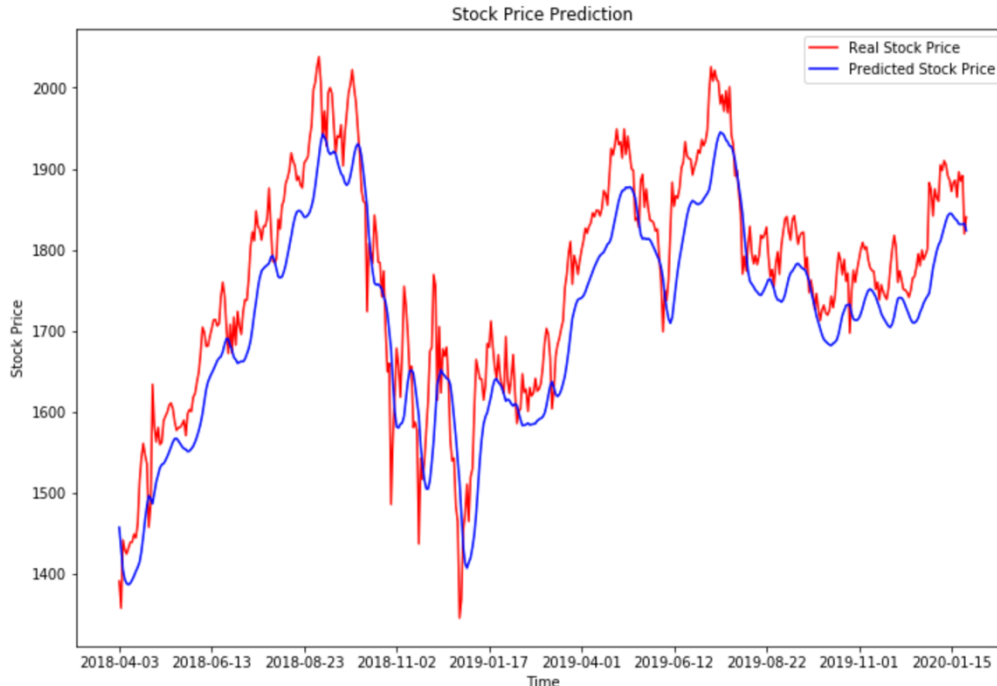


*Fig. Performance of different models on three stocks*

The observation is that while a three-layer model is created with a fixed batch-size (Model 3 vs. Model 4, Model 5 vs. Model 6), a smaller dropout rate (0.1) will produce a more accurate result. Also, in this case, when dropout rate is the same (Model 3 vs. Model 5 and Model 4 vs. Model 6), batch size of 32 produces more accurate results than batch size of 64. While batch size and dropout rate are the same (Model 1 vs. Model 3 and Model 2 vs. Model 4), a three-layer model produces more accurate result than a four-layer model.



*Fig. Model 4 Result on Facebook's Stock*



*Fig. Model 4 Result on Amazon's Stock Price*

According to the performance metrics, Model 4 produces the best result, while Model 2 gives the largest error value. In addition, Model 3 produces results almost similar to Model 4, and Model 6 is next. Therefore, the project finds the best model to be a LSTM network with 50 neurons, 3 layers, dropout rate of 0.1 and batch size of 32. Interestingly, the model used in this project seems to work particularly well on Facebook's stock price, while models to predict Amazon's stock price often produce the highest error. It might be due to factors related with the financial market.

## **6 Conclusion**

In this project, we find that the best performing algorithm is a LSTM network with 50 neurons, 3 layers, dropout rate 0.1 and batch size 32. If there is more time, it will be interesting to investigate the following two questions. The first question is what is the effect of number of neurons on the model? All models in the project have 50 neurons because they seem to work well. It is worth exploring how changing the number of neurons will influence the model's performance. The second question is how will a different timestep influence the model? All models use a timestep of 30 during data processing. While intuitively smaller timesteps should create more accurate results, it could take some experiments to determine the effect of timestep.

## References

- [1] Yakup Kara, Melek Acar Boyacioglu, and Ömer Kaan Baykan. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange. *Expert Systems with Applications*, 38(5):5311 – 5319, 2011.
- [2] K. Abhishek, A. Khairwa, T. Pratap, and S. Prakash. A stock market prediction model using artificial neural network. In *2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12)*, pages 1–5, July 2012.
- [3] Sheng Chen and Hongxiang He. Stock prediction using convolutional neural network. *IOP Conference Series: Materials Science and Engineering*, 435(1):012026, 2018.
- [4] Bin Weng, Ahmed M A, Megahed F M. Stock Market One-day ahead Movement Prediction Using Disparate Data Sources [J]. *Expert Systems with Applications*, 2017, 79(2): 153–163.
- [5] A. W. Li and G. S. Bastos, "Stock Market Forecasting Using Deep Learning and Technical Analysis: A Systematic Review," in *IEEE Access*, vol. 8, pp. 185232-185242, 2020, doi: 10.1109/ACCESS.2020.3030226.
- [6] G. Sismanoglu, M. A. Onde, F. Kocer and O. K. Sahingoz, "Deep Learning Based Forecasting in Stock Market with Big Data Analytics," 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), Istanbul, Turkey, 2019, pp. 1-4, doi: 10.1109/EBBT.2019.8741818.
- [7] K. Khare, O. Darekar, P. Gupta and V. Z. Attar, "Short term stock price prediction using deep learning," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2017, pp. 482-486, doi: 10.1109/RTEICT.2017.8256643.
- [8] J. WU, C. WANG, L. XIONG and H. SUN, "Quantitative Trading on Stock Market Based on Deep Reinforcement Learning," 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 2019, pp. 1-8, doi: 10.1109/IJCNN.2019.8851831.
- [9] Andy. 2017. Recurrent neural networks and LSTM tutorial in Python and TensorFlow. (Oct. 2020). <http://adventuresinmachinelearning.com/recurrent-neural-networks-lstm-tutorial-tensorflow/>
- [10] Framework used: Keras, Tensor Flow, Pandas, NumPy