Optimizing Stock Price Prediction: Exploring LSTM Architectural Parameters in Financial Forecasting

Haoran Li *

Department of Material Science and Engineering, Tokyo Institute of Technology, Yokohama, 226-8503, Japan

* Corresponding Author Email: 1912231219@mail.sit.edu.cn

Abstract. Accurate stock price prediction is a crucial concern in the field of finance, impacting investors, traders, and the broader economic landscape. Traditional time series analysis and statistical models often fall short in capturing the dynamic and nonlinear nature of financial markets. Financial market forecasting has advanced significantly with the advent of deep learning technology, especially Long Short-Term Memory (LSTM) networks. This study conducts an in-depth exploration of LSTM networks' critical architectural parameters and their impact on stock price prediction. The performance of the LSTM-based model is assessed using historical stock price data from Amazon, with particular attention paid to hyperparameters like the number of neurons, dropout rates, and LSTM layers. The research reveals that a well-balanced model architecture with moderate units per layer (e.g., 64), a single layer, and a dropout rate of 0.7 is key to achieving accurate predictions in financial markets. The outcomes of this research hold practical significance for both investors and financial analysts. Furthermore, they emphasize the promising potential of LSTM networks in the realm of financial forecasting, opening avenues for future studies in finance. This investigation deepens the comprehension of LSTM network architecture and its applicability in predicting stock prices, providing valuable insights that extend beyond the finance sector.

Keywords: Stock Price Prediction, LSTM Networks, Hyperparameter Optimization.

1. Introduction

Financial markets have always been of extensive interest to investors, scholars, and financial analysts. Accurate prediction of stock prices has remained a fundamental issue in the field of finance [1]. It enables investors and traders to make informed decisions, mitigate risks, and optimize returns on their investments. Moreover, these predictions contribute to the overall stability and efficiency of financial markets, as they reflect economic trends, investor sentiment, and corporate performance. This information aids not only individual investors but also institutional investors, fund managers, and policymakers in making well-informed decisions, ultimately influencing the broader economic landscape.

While traditional time series analysis methods and statistical models have provided some predictive tools, they often prove to be insufficiently flexible when dealing with the highly dynamic and nonlinear nature of financial markets [2]. Traditional financial modeling approaches often struggle to adequately address the high uncertainty and complexity of the market. Financial market forecasting has significantly improved as a result of the deep learning technologies' quick growth, especially the financial sector's extensive use of LSTM networks [3, 4]. The aim of this study is to delve into the critical architectural parameters of LSTM networks, emphasizing their significance in predicting stock prices. For instance, researchers used LSTM networks for stock price prediction and compared them with traditional time series methods [5]. Additionally, Ding et al. combined historical stock price data with news textual information and employed deep learning methods for prediction [6]. In another paper, researchers introduced a deep learning framework that combines autoencoders and LSTM networks for predicting financial time series data [7]. Despite numerous successful use cases of LSTM networks in financial market prediction, there remains a relatively limited depth of research into the architectural hyperparameters of these networks, which deserves more attention.

This study carries significant theoretical and practical importance. Firstly, by conducting an in-depth exploration of the architectural parameters of LSTM networks, this study hopes to gain a better

understanding of how these parameters affect the performance of stock price predictions. Accurate stock price predictions are crucial for investors, traders, and financial decision-makers, by enhancing model performance, it is possible to mitigate investment risks and increase financial returns. Second, by revealing the possibilities of deep learning techniques in the financial markets, the study's findings will contribute to the body of knowledge on financial research. A better understanding of the applicability of LSTM networks can be obtained, thereby paving the way for new avenues in future financial research.

At the core of this study is an in-depth exploration of the critical architectural parameters of LSTM networks. The primary objectives of the research include evaluating the performance of models based on LSTM networks in stock price prediction, and investigating the impact of different hyperparameters on model performance, including LSTM layers, dropout rates, and quantity of neurons.

2. Method

2.1. Data Preparation

The study's dataset, which includes historical stock price data for Amazon (AMZN), was acquired via Kaggle [8]. The dataset contains only one feature, which is the adjusted closing price of the stock from May 1, 2013 to May 14, 2019. The model will use historical stock price data to make predictions about future prices, which is a common task in time-series forecasting. The training set, validation set, and test set comprise the entirety of the dataset. The training set has around 60% of the whole data, while the test set and validation set each comprise about 20% of the total data. Validation will be conducted during model training and the model's performance is ultimately evaluated on independent test data. The detailed information related to the dataset was provided in Fig. 1.

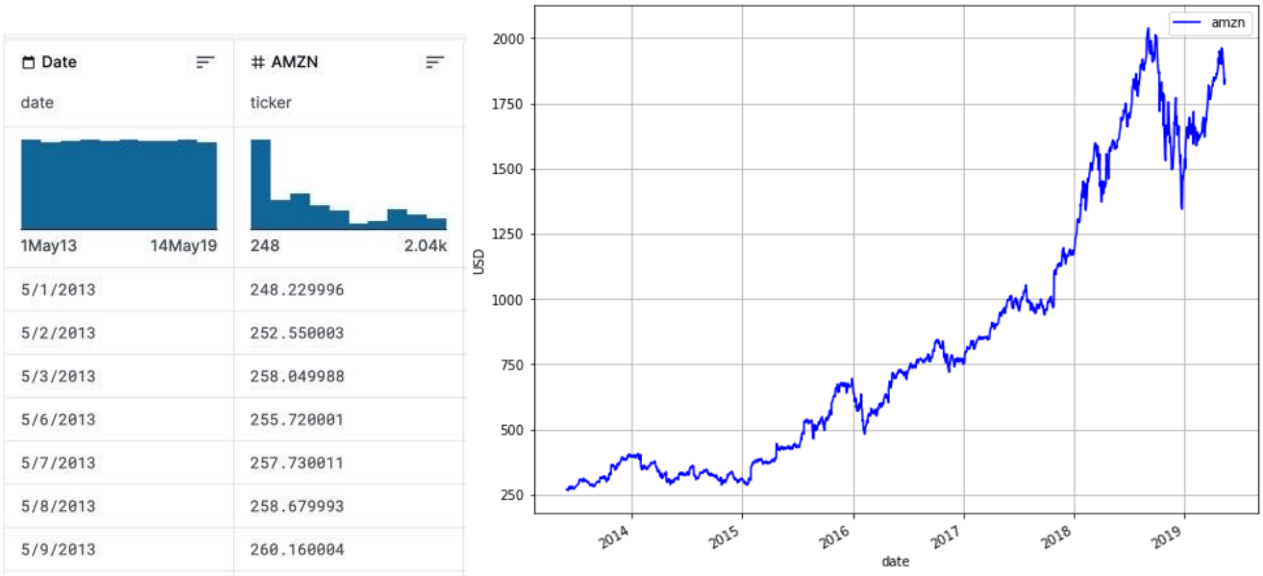


Figure 1. Details of the dataset [8].

The Min-Max normalization method was utilized to scale stock price data to fall within the range of 0 and 1, ensuring that features with different numerical ranges do not bias the model's training. Additionally, the time step was set to 100. When constructing the training, validation, and test datasets, 100 past data points were used as input to predict the next data point, enabling the model in learning dependencies and patterns in time-series data. By employing these dataset preparation methods, the LSTM model was furnished with normalized data and input sequences along with their corresponding target variables for the purpose of forecasting.

2.2. LSTM Model

Recurrent neural network (RNN) architecture of the LSTM type has shown to be very successful in a variety of sequential data processing applications [9]. It was created to get around the difficulties of vanishing gradients and long-range dependencies that conventional RNNs couldn't handle. In order to efficiently record and retain data across lengthy sequences, LSTM shown in Fig. 2 integrates a complex gating system that includes input, output, and forget gates. Forget gate is Responsible for deciding which information to retain or forget in the memory cell. It generates an output between 0 and 1 using a sigmoid activation function, where 0 signifies complete forgetfulness, and 1 means full retention. The input gate makes the decision about what fresh data goes into the memory cell. Similar to the forget gate, it employs a sigmoid function to generate weights, but it also utilizes a tanh activation function to decide what new memory values to update. An output gate uses a mix of sigmoid and tanh functions to determine whether data from the memory cell should be transmitted to the following layer or other areas of the network. Because of its design, LSTMs are well-suited for sequential data applications including voice recognition, time series analysis, natural language processing, and, as this study shows, stock price prediction. Their ability to capture intricate patterns and relationships within sequential data has made LSTMs a fundamental building block in modern deep learning for sequential data analysis.

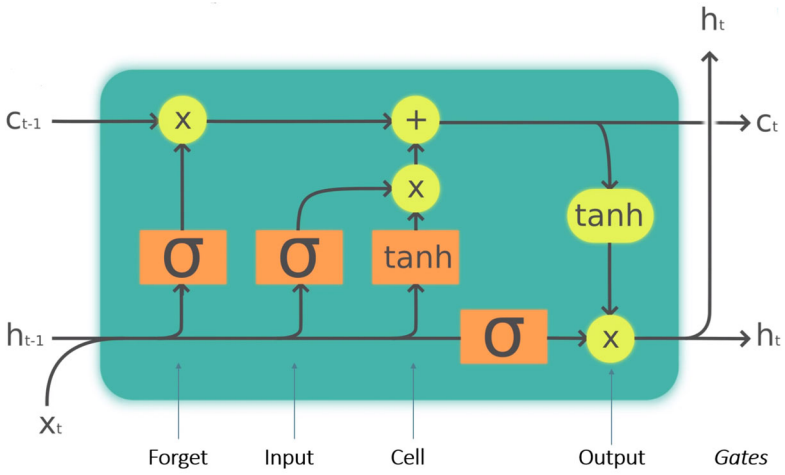


Figure 2. The structure of the LSTM neural network [10].

This study employed a multi-layer LSTM model shown in Fig. 3. Dropout layers are added after each LSTM layer to prevent from overfitting. Hyperparameters including layers, units within each layer, and dropout layer will be discussed in this study.

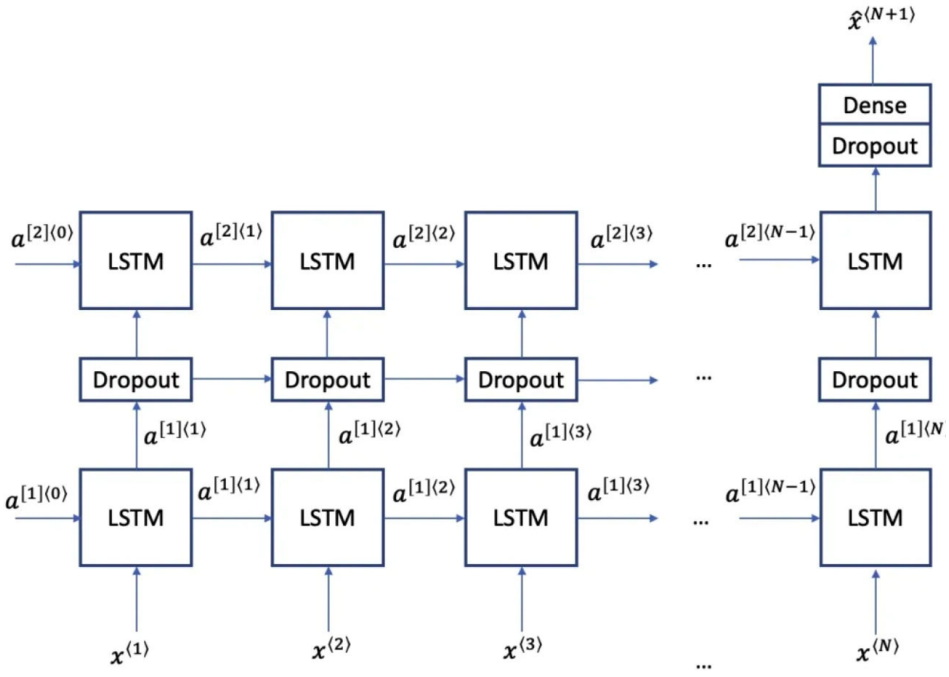


Figure 3. Multi-layer LSTM with dropout mechanism [11].

2.3. mplementation Details

Popular deep learning framework Pytorch was used to build the model. An adaptive learning rate optimization technique, Adam optimizer, was also used by the model. Typically, a learning rate of 0.001 was initialized in this model. The Adam optimizer could dynamically adjust the learning rate for each parameter, resulting in faster convergence. The Root Mean Squared Error (RMSE) is the main assessment statistic for this model. The model was able to handle all of the training data at once in each epoch since the batch size was chosen to match the size of the complete training dataset (812).

3. Results and Discussion

As is shown in Fig. 4., the ability of the model to capture intricate patterns in the data can be directly impacted by the number of units in a layer. Underfitting, in which the model is unable to identify the underlying patterns in the data, might result from using too few units (e.g., 16 units). This results in a high training and test error. On the other hand, too many units (e.g., 512 units) can lead to overfitting. With too much complexity, the model fits the noise in the training data and produces a low training error but a large test error. The model with 64 units per layer performs the best with an RMSE of 0.6689. However, the model with 32 units per layer has a relatively high RMSE of 0.6918, indicating potential underfitting. The results suggest that a moderate number of units can lead to better performance. The choice of the number of units can depend on the complexity of the problem and the available data [10].

When applying different numbers of layers, the result in Fig. 4. shows that having a single layer performs the best with an RMSE of 0.6556. As more layers were added, the RMSE slightly increases but remains relatively close. It appears that there is no discernible improvement in model performance when the number of layers is increased beyond 1. This observation aligns with the idea of avoiding overly deep networks if not necessary to prevent overfitting [11]. When considering number of units and layers together, it can be inferred that in AMZN dataset scenario, a model with a moderate number of units (e.g., 64) and a single layer performed best in terms of RMSE, indicating that model complexity can be managed effectively with a simpler architecture.

The experiments in Fig. 4. also indicates that a dropout rate of 0.7 results in the lowest RMSE of 0.6769, followed by 0.3 with an RMSE of 0.6743. Overall, a moderate dropout rate can improve

model generalization, but too high or too low dropout rates (e.g., 0 or 0.9) may adversely affect performance [12].

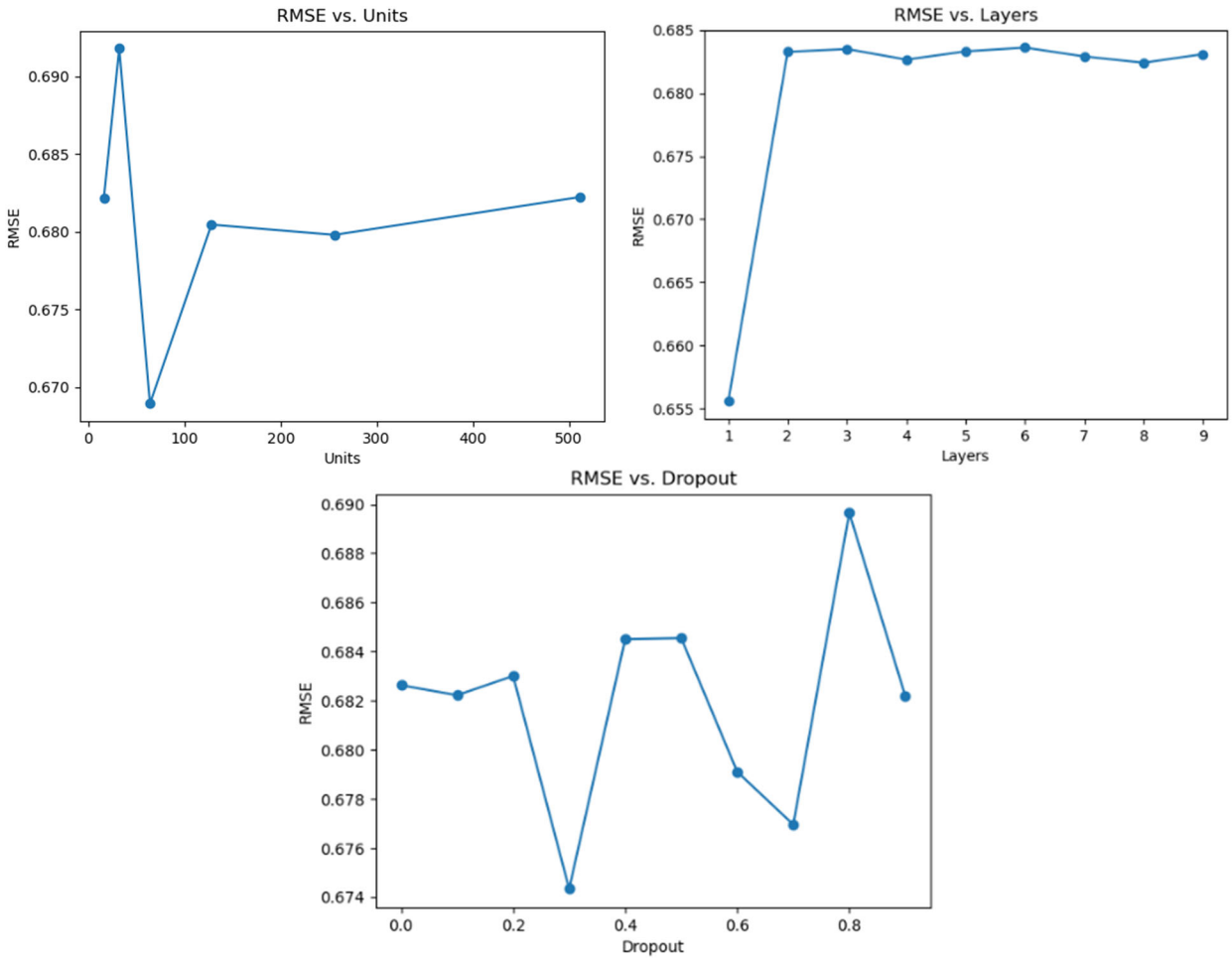


Figure 4. Relationships between RMSE and number of units, layers, and dropout rate (Photo credit : Original).

4. Conclusion

This study has provided valuable insights into the critical architectural parameters of LSTM networks and their significance in predicting stock prices. By evaluating the performance of LSTM-based models, this study has found that a moderate number of units (e.g., 64) per layer, a single layer, and a dropout rate of 0.7 are key factors in achieving accurate stock price predictions. These findings contribute to the understanding of how LSTM network architecture impacts model performance and can be applied to mitigate investment risks and enhance financial returns. The implications of this research are twofold. Firstly, it enhances the understanding of deep learning methods in financial markets, offering valuable lessons for investors and financial decision-makers. Secondly, the study's insights can guide future financial research by showcasing the potential of LSTM networks in time-series analysis. These findings facilitate the utilization of well-balanced model architectures for accurate stock price predictions. While this study sheds light on LSTM architectural parameters, there is room for further exploration. Future research may delve into additional hyperparameters or apply these insights to different financial datasets. Further research may also consider real-world market conditions, such as volatility and news sentiment, to improve model robustness.

References

[1] LeCun, Y., Bengio, Y., Hinton, G., et al. Deep Learning. Nature, 2015, 521 (7553), 436 - 444.

[2] Dong, H., Li, J., Peng, J., et al. Stock Price Prediction Using LSTM, RNN, and CNN-Sliding Models, 2016, IEEE Transactions on Neural Networks and Learning Systems, 28 (10), 2220 - 2230.

[3] Hochreiter, S., Schmidhuber, J. Long Short-Term Memory, 1997, Neural Computation, 9 (8), 1735 - 1780.

[4] Vinkler, M., Mekyska, J., Burget, et al. Financial Market Time Series Prediction using RNNs and CNNs, 2109, arXiv preprint arXiv: 1811. 05921.

[5] Heaton, J., Polson, N. G., et al. Stock Price Forecasting Using Deep Learning, 2016, ArXiv.

[6] Ding, X., Zhang, Y., Liu, T., et al. Deep Learning for Stock Prediction Using Numerical and Textual Information, 2017, ArXiv.

[7] Ahmed, M., Mahmood, A. N., Hu, J., Hu, J., & Luo, J. (2017). A Deep Learning Framework for Financial Time Series Using Stacked Autoencoders and Long Short-Term Memory, 2017, PLOS ONE.

[8] Amazon Stock Data. https://www.kaggle.com/datasets/varpit94/amazon-stock-data.

[9] Hochreiter, S., Schmidhuber, J. Long Short-Term Memory. Neural Computation, vol. 9, no. 8, pp. 1735 - 1780, 15 Nov. 1997.

[10] Understanding LSTMs. https://www.kushalj001.github.io/black-box-ml/lstm/pytorch/gates/vanishing%20gradient/2019/12/28/Understanding-LSTMs.html.

[11] Structure of the multi-layer LSTM network. https://www.researchgate.net/figure/Structure-of-the-multi-layer-LSTM-network_fig3_329532056.

[12] Srivastava, N., Hinton, G., et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting, 2014, Journal of Machine Learning Research, 15 (1), 1929 - 1958.