Forecasting Expected Move in Stock Prices Post Earnings Call Using Neural ODEs

Pratyush Menon
Department of Electrical and Computer Engineering
University of Toronto

Dhruv Patel
Department of Electrical and Computer Engineering
University of Toronto

Glen Liang
Department of Computer Science
University of Toronto

Abstract

Stock market prediction is an area of high interest in the research community, with current baselines often using recurrent neural network (RNN) models. However, recent research has focused on models using differential equations to model time-series data. Indeed, neural controlled differential equation (CDE) models are a continuous time extension of RNN models, outperforming them in some tasks. As such, neural CDE models and long short-term memory (LSTM) models were tested and compared on the stock market prediction problem. The specific scenario was to predict close prices of AAPL shares based on the past 20 days of daily data consisting of the company's past close, open, high, low and volume. Results show that LSTM models appear to outperform neural CDE models handily, but other differential equation based models may offer further improvements.

1 Introduction

A common understanding amongst investors to explain the volatility in a company's stock price is that, in the long term, the stock price strongly correlates to the company's earnings but in the short term, it is a cumulative result of investor sentiment. If investors expect the company to do better, then they start purchasing and send the prices higher. If they expect the company to underperform, then they start selling and send the prices lower. These price fluctuations form a price chart, and the short term price charts tend to have patterns since fluctuations in current prices sways investor sentiment. These price fluctuations and patterns are cyclic and tend to repeat. So, through the use of historical price charts, the patterns in the short term price charts can be identified and used to make one day stock price predictions. This problem can be viewed as a time-series, with the data being the company's historical price chart which includes all the important metrics used for technical analysis (the company's past close, open, high, low and volume). The performance of neural controlled differential equation (CDE) models on this problem will be compared as an alternative to current baseline models using long short-term memory (LSTM) cells.

2 Related Works

There has been much prior research done on predicting and forecasting company's stock prices and short term price trends. Some utilize the historic price data while some use corporate events to establish predictions. One such way is through the use of earnings figures from a company (corporate event) - especially in the case of "earnings surprise", where a company's reported figures drastically

Preprint. Under review.

differ from estimates. Some of the more traditional techniques to forecast closing prices and volume traded after results are reported include using autoregressive models with the additional information of Google search trends data [1].

More recently, statistical models for financial forecasting have moved towards more complex neural network-based models. Some approaches included using neural networks for regression, wavelet neural networks, and support vector machines [2]. However, the most successful method was to treat prior financial data as a time-series and rely on recurrent networks such as LSTM networks, while retaining the focus on feature engineering [2].

A new class of networks that appear to show better performance than LSTM networks on time-series data are neural differential equations, developed by researchers at the University of Toronto [3]. It was shown that neural ODE models were better able to learn the latent dynamics of a time-series, despite issues such as irregular sampling of input data [3]. This promising result suggests that neural ODEs may provide better predictive performance on other dynamical systems, such as financial forecasting. Specifically, neural controlled differential equations (CDEs) will be used, as they are likened to a continuous analogue of recurrent neural network models [4].

3 Method/Algorithm

3.1 Method

We developed a LSTM neural network, using the opensource Pytorch library [5], and a neural CDE network, using the open source torchcde library [4], and trained them at predicting the following day close price value of AAPL (Apple Inc.) using day data from the past 20 days. The historical day data is sourced from Yahoo Finance.

The model predicts the real closing price of the stock. The reason for predicting real price is based on related work using LSTM to do real price prediction, which has been shown to produce good results [6].

Additionally, we have chosen the AAPL stock as our benchmark due to its large repository of historical data (being a blue chip stock and old company), its close correlation with the major indices (due to its large market cap and being major part of them) and low short term volatility (due to strong consistency in its earnings and being a major part of indices).

12 years of historical day data of AAPL stock, from January 2010 up until April 2022 was collected. The data contains 5 features: the listing price upon the opening of an exchange on a trading day, the closing price of the trading day, the highest listing price of the trading day, the lowest listing price of the trading day, and the total number of shares that were traded during the day, which was found to have a positive relationship with the closing price [7].

The training set covered 80 percent of the original data set; and the validation and test sets covered the last 20 percent of the data set. The data is min-max normalized, set to be within range between 0 to 1 to the entire data set across all features. Normalization was done across all features to reduce complexity and increase training speed. Finally, results across both models were compared using the respective best performance achieved on each model.

3.2 Algorithm

Base LSTM Algorithm:

- The input x is the preprocessed data for the last 20 days.
- x is input to the LSTM layer(s), which have hidden and cell states initialized as 0.
- The hidden state at the last timestep is used as the final output of the LSTM.
- Finally, a simple feed-forward neural network is used to output the final prediction, based on the final hidden state of the LSTM as input. This prediction, y, is the stock closing price for the next day.

Neural CDE Algorithm as shown in equation (1) [8]:

- The input x is the preprocessed data for the last 20 days.

- Natural cubic spline interpolation is used to turn x into X, a continuous interpolation of x known as the control path.
- The initial hidden state $z(t_0)$ is computed as a function ζ_{θ_2} of the first observation of X, X_0.
- This function is parameterized by a simple feed-forward neural network, with parameters θ_2.
- Another simple feed-forward neural network, with parameters θ_1, is used to parameterize the CDE function, f_{θ_2}.
- Using the continuous path, initial hidden state, and CDE function, the CDE is solved with a black-box CDE solver. The solution to the CDE controlled by X is $z(t)$.
- The terminal value from the CDE, $z(t_n)$, is extracted. Then, another function g_{θ_3}, which is parameterized by a simple feed-forward neural network with parameters θ_3, is applied to $z(t_n)$ to obtain a final result, y. This prediction is the stock closing price for the next day.

$$z(t_0) = \zeta_{\theta_2}(X_0), \quad z(t) = z(t_0) + \int_{t_0}^{t} f_{\theta_1}(z(s)) \frac{dX}{ds} ds \quad \text{for} \quad t \in (t_0, t_n], \quad y = g_{\theta_3}(z(t_n)) \tag{1}$$

4 Experiments & Discussion

4.1 Experiments

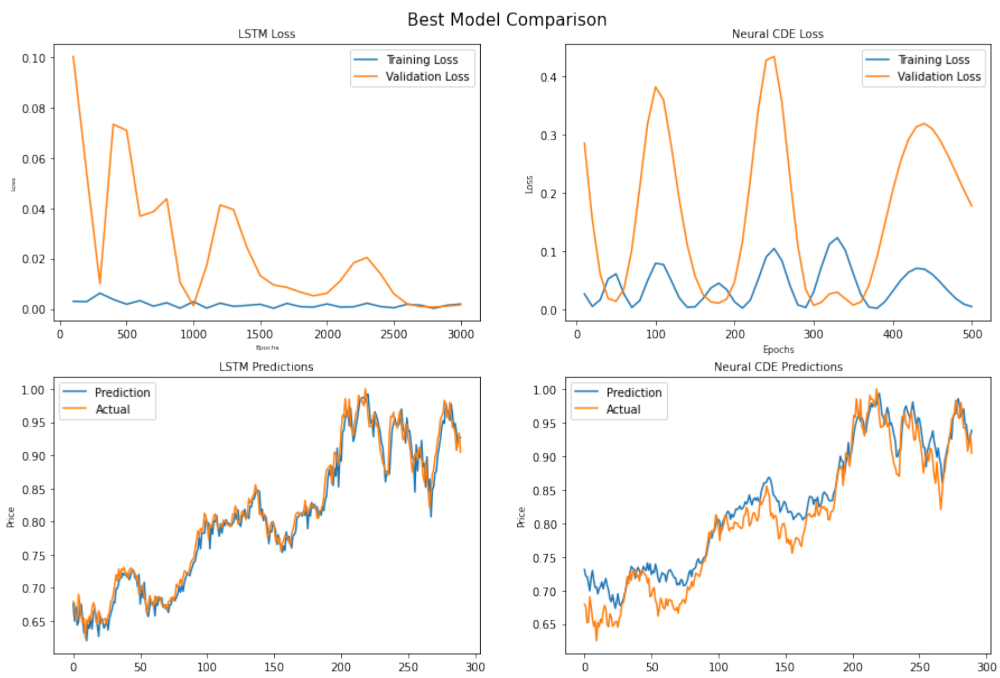

Figure 1: Training and final result plots for LSTM and neural CDE models.

To provide a baseline, we first experimented with LSTM to produce results similar to existing models [9]. Using existing baselines from previous work, we have used a fixed sequence length of 20 (number of days used for prediction). We experimented with our LSTM model by varying the learning rate, hidden size, number of layers, and number of training epochs. Similarly, we experimented with different hyperparameters for the neural CDE model with hidden size, learning rate, number of layers, and number of training epochs. Both the LSTM and neural CDE models are trained using the Adam optimizer and MSE loss function. To prevent overfitting, the model at the epoch with lowest validation loss through the entire training process was used for final comparisons.

The best LSTM model used a learning rate of 0.0001, hidden size 68, 1 LSTM layer, and trained with 3000 epochs. The optimal CDE model hyperparameter used are a hidden size of 128, learning rate of 0.0001, 1 layer, and trained with 500 epochs.

Figure 1 shows that, when comparing the best LSTM and neural CDE models we were able to train, the LSTM model clearly outperforms the neural CDE model at this task.

4.2 Discussion

Neural CDE models are intended to improve upon a shortcoming of recurrent neural network models when used to model a sequence of observations from an underlying process in continuous time. This results in RNN models being a discrete approximation of this process. While this often works well, situations where observations are irregularly sampled or missing result in worse performance. As shown in Figure 2, Neural CDE models allow the hidden state to evolve as a continuous process, to better model the continuous underlying process behind the observations, and this resulted in better performance in the aforementioned situations [4].

However, for the problem considered in this paper, the underlying process behind observations of the stock market (ie. the market sentiment) is heavily affected by discrete events such as press releases, product launches, competitor events, and any other news. These discrete events would heavily affect the overall continuity of the market sentiment behind a particular stock. As such, a continuous hidden state may not be the best model of market sentiment. This may explain the observed oscillatory nature of the training and validation loss, rather than the expected clearly decreasing losses.

Furthermore, the strengths of neural CDE models were shown to be in modeling functions of irregularly observed time series [8], where they outperformed RNN models. However, the stock market price prediction problem does not fit into that category, as data is fully observed at regular intervals. As such, it does not utilize the strengths of neural CDE models.

Overall, LSTM models appear to be a better choice for stock market prediction problems than neural CDE models. However, other differential equation based models could provide better performance than either. Future exploration in this area could be done by testing models such as neural ordinary differential equations [3] and neural jump stochastic differential equations [10] on a similar dataset.

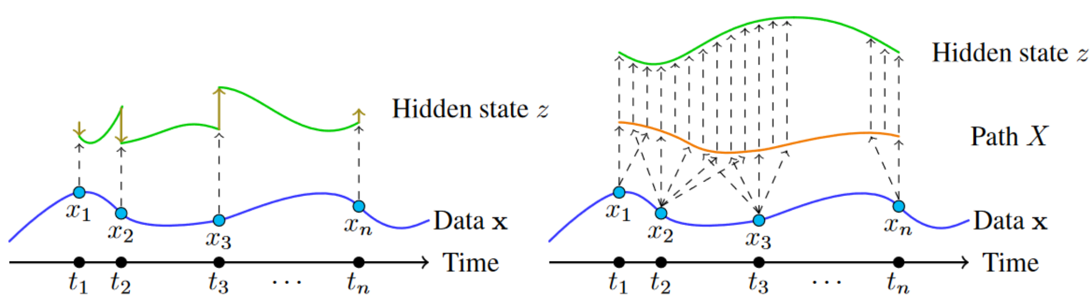

Figure 2: A comparison of hidden state between recurrent networks or neural ODEs (left) and neural CDEs (right). Recurrent networks and neural ODEs modify the hidden state at each observation, with neural ODEs continuously evolving the hidden state between each observation. Neural CDEs create a continuous path from the observations and as such the hidden state has a continuous dependence on the observations. [4]

5 Conclusion

LSTM and neural CDE models were used to explore the idea that, in the short term, a company's stock price is dependent on market sentiment and exhibits patterns that can be learned and used to make short term predictions. Models were trained on 12 years of historical data and a 20 day interval was used to make predictions for the next day's close price for AAPL. Multiple LSTM and neural CDE models were experimented on, with the final observation that LSTM models outperform neural CDE models on this particular task. This is likely due to the fact that stock market prediction does not utilize the strengths of neural CDE models, which outperform RNN models on irregularly observed time series data, which was not the case for this experiment.

References

[1] J. Baker, “Forecasting Volume and Price Impact of Earnings Surprises Using Google Insights,” The International Journal of Business and Finance Research, vol. 10, no. 4, pp. 53–62, 2016.

[2] J. Shen and M. O. Shafiq, “Short-term stock market price trend prediction using a comprehensive deep learning system,” Journal of Big Data, vol. 7, no. 1, 2020.

[3] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, “Neural Ordinary Differential Equations,” Advances in Neural Information Processing Systems, vol 31, 2018.

[4] P. Kidger, J. Morrill, J. Foster, T. Lyons, “Neural Controlled Differential Equations for Irregular Time Series,” Advances in Neural Information Processing Systems, 2020.

[5] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, … S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” Advances in Neural Information Processing Systems 32, 2019.

[6] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, and S. S., “Deep learning for stock market prediction,” Entropy, vol. 22, no. 8, p. 840, 2020.

[7] J. Karpoff, "The Relation Between Price Changes and Trading Volume: A Survey,” The Journal of Financial and Quantitative Analysis, vol. 22, no. 1, p. 109, 1987.

[8] J. Morrill, P. Kidger, L. Yang, T. Lyons, “Neural Controlled Differential Equations for Online Prediction Tasks,” ArXiv, abs/2106.11028, 2021.

[9] A. Moghar and M. Hamiche, “Stock market prediction using LSTM recurrent neural network,” Procedia Computer Science, vol. 170, pp. 1168–1173, 2020.

[10] J. Jia and A. R. Benson, “Neural Jump Stochastic Differential Equations,” Proceedings of the 33rd International Conference on Neural Information Processing Systems, Red Hook, NY, USA: Curran Associates Inc., 2019.

Appendices

Appendix A: Code Repository

https://github.com/GlenPSI/CSC413GroupProject

Appendix B: Contribution Table

Glen Liang: Developed and written learning method, data preparation, LSTM model, did hyperparameter tuning, model training, wrote experiment, method.

Pratyush Menon: Developed and trained neural CDE model, did hyperparameter tuning, wrote abstract, wrote algorithm, wrote discussion, wrote conclusion, edited document, compiled latex document.

Dhruv Patel: Research neural CDE model and latent ODEs, wrote abstract, wrote related works, wrote introduction, wrote algorithm, helped write conclusion, made the final latex document.