Financial Time Series Forecasting using Long Short-Term Recurrent Neural Network

Kyro Gibling (kyro.gibling@mycavehill.uwi.edu)
The University of the West Indies, Cave Hill Campus, Barbados

Abstract

Forecasting financial time series data is a formidable problem; but if successfully achieved can lead to financial success. The purpose of this research is to investigate the most effective Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) topology and generalization technique; with the prime objective of capturing the high variance inherent in a financial time series, when subject to a limited amount of training data. This paper presents the results of nine different LSTM RNN models; whereby, each model is compared using a consistent performance metric, the Mean Squared Error (MSE) function. Post-tests reveal that model performance is strongly effected by the network topology, as well as generalization techniques applied to the network learning procedure. The results of this research are informative, as it relates to implementing such models to forecast different financial time series.

Introduction

Financial time series forecasting involves the prediction of future values (prices of a financial instrument) based on previously observed values; something usually tackled with different Machine Learning (ML) algorithms (sometimes an ensemble of different models), or core Mathematical methods, such as the Fourier Series [1].

There are a variety of different approaches which can be taken; however, the underlying reasons which make this problem such a formidable opponent need to be explored. It is widely hypothesized that the challenges are directly linked with fact that there are many different variables (of varying degrees of correlation) which affect the time series [2]. Furthermore, the determination of future price points within the series must take into consideration the values of each of these features at many previous states (within the series) [3]; thus, demanding a learning procedure which has some internal memory, capable of learning long term dependencies [4].

There is extensive research which suggests that Recurrent Neural Networks (RNNs) are best suited to capture the recursive nonlinearity of time series data [5]. There are many different variants of the RNN model; however, research indicates that the Long Short-Term Memory Network (LSTM) is arguably the best for capturing long term dependencies within a time series [6].

The purpose of this paper is to determine the most effective methods/techniques, which maximize the ability of an LSTM RNN to generalize well to unseen data; especially, when there may not a sufficient amount of historical data from which to model the time series.

Literature Review

Several studies have indicated that Artificial Neural Networks (ANN) perform the best at forming nonlinear hypotheses in high dimensional space[7]. There are many different forms of ANNs, namely the traditional Feedforward Neural Network (FNN), and the RNN; however, there is strong evidence which suggests that FNNs perform poorly in such scenarios[8]. This is due to the fact that FNNs are designed to accept a set of data (different datum instances of input features) at a single state, and form a hypothesis based on those inputs.

When modeling a financial time series, the predicted time series rely on multiple datum points, over many previous states. A solution to modeling such sequential data is the RNN; however, research suggests that these models do not perform well either[9], and this can be attributed to their inability to capture long term dependencies; a direct consequence of the vanishing gradient problem[10]. An elegant solution to learning long-term dependencies is the Long Short-Term Memory (LSTM) model[6].

Studies indicate that LSTM RNNs have achieved high success in modeling time series data [8]; however, there are many challenges involved with creating effective LSTM RNNs; namely, determining the network topology (which directly influences the models predictive power), as well as the tuning of Hyper-parameters [11].

Methodology

In this case study ten years of Microsoft's historical stock data (specifically the price on market open) was used; whereby the entire dataset was split into training and testing subsets, of similar ratios (55% for training, 45% for testing), which would simulate having a limited amount of training data to make our predictions. Twelve different models were trained, tested, and evaluated (using the training and testing datasets respectively); whereby for each experiment the training and testing losses were recorded. These results were then compared in order to come to a conclusion of which models generalize the best when subject to limited training data.

The network was trained using the Adam Optimizer (for optimizing network weights and biases), in conjunction with the Mean Squared Error (MSE) function as a metric for measuring network performance. The batch size used for training was 32 training instances (input, output pairs). In addition, the two techniques used to teach the models to generalize well were Regularization and Dropout (used independently of each other).

Results

The experiments indicate that shallow networks with dropout perform the best; whereas deep networks with regularization perform the worst.

Experiments Summary

Experiment 1: 1 LSTM Input Layer; 2 LSTM Hidden Layers; 1 Output Dense Layer; MSE Test: 0.0001261; MSE Train:
Experiment 2: 1 LSTM Input Layer; 1 LSTM Hidden Layer; 1 Output Dense Layer; MSE Test: 0.0001136; MSE Train: 0.00021086
Experiment 3: 1 LSTM Input Layer; 4 LSTM Hidden Layers; 1 Output Dense Layer; MSE Test: 0.000084; MSE Train: 0.00051268
Experiment 4: 1 LSTM Input Layer; 3 LSTM Hidden Layers (Dropout applied to each hidden layer); 1 Output Dense Layer; MSE Test: 0.000106; MSE Train: 0.00074963
Experiment 5: 1 LSTM Input Layer; 1 LSTM Hidden Layer (Dropout applied to hidden layer); 1 Output Dense Layer; MSE Test: 0.0000586; MSE Train: 0.000729
Experiment 6: 1 LSTM Input Layer; 1 LSTM Hidden Layer (L2 Regularization applied to hidden layer); 1 Output Dense Layer; MSE Test: 0.1096287; MSE Train: 0.0268
Experiment 7: 1 LSTM Input Layer; 4 LSTM Hidden Layers (L2 Regularization applied to each hidden layer); 1 Output Dense Layer; MSE Test: 0.67099; MSE Train: 0.1844
Experiment 8: 1 LSTM Input Layer; 1 LSTM Hidden Layer; 1 Output Dense Layer (L2 Regularization applied to output layer); MSE Test: 0.027417; MSE Train: 0.0875
Experiment 9: 1 LSTM Input Layer; 4 LSTM Hidden Layers (L2 Regularization applied to the last hidden layer); 1 Output Dense Layer; MSE Test: 0.21093; MSE Train: 0.0273

The following figures illustrate the true testing price (red) versus the predicted testing price (blue):

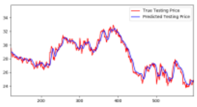

Figure 1. Resultant time series from experiment 5.

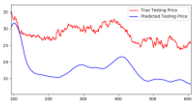

Figure 2. Resultant time series from experiment 9.

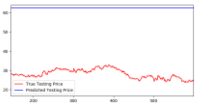

Figure 3. Resultant time series from experiment 7.

Discussion

These experiments indicate that shallow network architectures provide the best predictive power; which is due to the fact that the model is using a single feature (which equates to less complexity to capture) to base its predictions. It is also evident that the applying dropout to these shallow layers improves performance even more; a direct consequence of the fact that this reduces the chances of the network overfitting (keeping in mind that the testing proportion is a similar size to the training proportion; thus, the need to generalize extremely well).

The experiments also show that deep network architectures generalize poorly; a direct consequence of overfitting . All of the results mentioned above concur with the literature literature; thus, strongly supporting the claim that the complexity of network architecture should be directly proportional to the number of input features from which the network is basing its predictions.

The fundamental limitation of this experiment is the number of features which were used to base the predictions. Many different factors affect a financial time series; thus, more features should be taken into consideration in order to capture the variance of a financial time series.

Conclusions

This paper addressed the use of different techniques and network topologies with regard to their ability to help RNNs generalize better, when subject to a limited amount of training data. After conducting nine different experiments using different models, it became apparent that a three layer network, with dropout in the hidden layer, had the most predictive power. Future studies should focus on discovering more features (with high degrees of correlation) in order to effectively capture the variance of a non-linear time series.

References

[1] Erdal Kayacan, Baris Ulutas and Okyay Kaynak, *Grey System Theory-Based Models in Time Series Prediction*, (), 1786-1787

[2] Jan G. De Gooijer and Rob J. Hyndman, International Journal of Forecasting: *25 Years of Time Series Forecasting*, (Elsevier, 2006), 444

[3] Armando Bernal, Sam Fok, and Rohit Pidaparthi, *Financial Market Time Series Prediction with Recurrent Neural Networks*, (December 14, 2012), 1-2

[4] Mohammad Assaad, Romuald Bone, and Hubert Cardot, *A new boosting algorithm for improved time-series forecasting with recurrent neural networks*, (Elsevier, 2008), 45

[5] Nicole Maknickiene, Aleksandras Vytautas Rutkauskas, and Algirdas Maknicka, *Investigation of financial market prediction by recurrent neural network*, (2011), 3

[6] Ian Goodfellow, Joshua Bengio, and Aaron Courville, (The MIT Press, Cambridge, Massachusetts London, England, 2017), 397

[7] Ian Goodfellow, Joshua Bengio, and Aaron Courville, (The MIT Press, Cambridge, Massachusetts London, England, 2017), 363

[8] Zhang Jia-Shu and Xiao Xian-Ci, *Predicting Chaotic Time Series Using Recurrent Neural Network*, (2000), 88-89

[9] Ian Goodfellow, Joshua Bengio, and Aaron Courville, (The MIT Press, Cambridge, Massachusetts London, England, 2017), 367-370

[10] Ian Goodfellow, Joshua Bengio, and Aaron Courville, (The MIT Press, Cambridge, Massachusetts London, England, 2017), 390

[11] Ian Goodfellow, Joshua Bengio, and Aaron Courville, (The MIT Press, Cambridge, Massachusetts London, England, 2017), 415