A Comparison of Long Short-Term Memory and ARMA Model on a Bitcoin Prices Dataset

Jose R. Navarro NAVARROJ@MIT.EDU
Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA. 02139

Manuel A. Mundo MMUNDO@MIT.EDU
Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA. 02139

Abstract

There are three review cycles for ICML 2013, with full paper submissions due on October 1, 2012; December 15, 2012; and February 15, 2013. Reviewing will be blind to the identities of the authors, and therefore identifying information must not appear in any way in papers submitted for review. Submissions must be in PDF, with an 8 page length limit (upto 9 pages including references).

1. Introduction

Our project presents a contrast of two different statistical methods to the problem of studying sequential data.

In particular we contrast a linear model as it is the ARMA model with a highly nonlinear estimation model as is a Long-Short Term Memory neural network.

2. Models

In this section, the models used to understand the data are described.

2.1. Autoregressive Moving Average

AutoRegressive (AR) Moving Average (MA) models are the most frequently used theoretical framework to model time series data. The following formal definition is taken from Brockwell and Davis, "Introduction to Time Series and Forecasting".

Definition 1. A stochastic process $\{X_t\}$ is an ARMA(p,q) process if it is stationary and, for every t,

$$X_t - \phi_1 X_{t-1} - \ldots - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + \ldots + \theta_q Z_{t-q}, \qquad (2.1)$$

where Z_t is a white noise process, i.e. $\{Z_t\} \sim N(0, \sigma^2)$, and the polynomials $1 - \phi_1 y - \ldots - \phi_p y^p$ and $1 + \theta_1 y + \ldots + \theta_q y^q$ have no common factors.

These models allow researchers to understand the influence of previous elements of the process, X_{t-i} on the current random variable X_t. This is clearly a linear model with a high level of interpretability. However, we suspect a LSTM neural networks might, generally, have higher accuracy in the prediction of X_t based on previous values of the process. This claim is tried empirically in this paper as will be clear later.

2.2. Autoregressive Integrated Moving Average

Another model useful for time series analysis is ARIMA also defined in Brockwell and Davis.

Definition 2. If d is a nonnegative integer, then X_t is an ARIMA(p, d, q) process if $Y_t := (1 - B)^d X_t$ is a causal ARMA(p, q) process.

2.3. Recurrent Neural Networks

Recurrent Neural Networks are used to model sequential data such as text and speech but variants can be created to perform time series analysis.

A RNN is typically represented as an unfolded computational graph as in figure 1 below:

RNN suffer from gradient vanishing problem which has been solved by Hochreiter et al among others with the introduction of LSTM recurrent networks which is the tool used in this paper.

Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

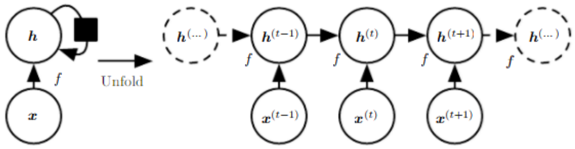


Figure 1. Unfolded Computational graph representing a RNN. From *Deep Learning* by Goodfellow et al.

2.3.1. Long Short-Term Memory

A Long Short-Term Memory architecture consists of a memory block as shown in figure 2 bellow. The memory block has three multiplicative units, known as input output and forget gates.

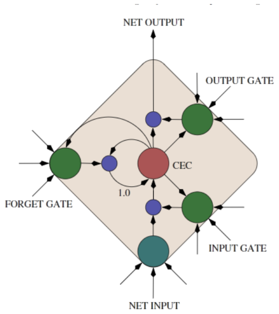


Figure 2. Memory cell for LSTM.

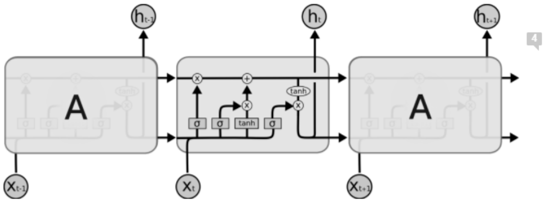
These gates perform the operations in the memory block are described as follow

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

where x_t is the input vector to the LSTM block; f_t, the forget gate's activation vector; i_t, the input gate's activation vector; o_t, the output gate's activation vector; h_t, the output vector of the LSTM block; c_t, the memory cell state vector. The activation functions are σ_g, a sigmoid function; σ_c and σ_h, hyperbolic tangent functions.

Additionally, $W \in R^{h \times d}$, $U \in R^{h \times h}$ and $b \in R^h$ are the weight matrices and bias vector parameters which need to be learned during training.

The memory blocks are recurrently connected as figure 3 shows.



The repeating module in an LSTM contains four interacting layers.

Figure 3. Memory blocks for LSTM.

3. Experimental Setup

3.1. Dataset construction

3.2. Setup for ARMA/ARIMA

Data with different time spans was used, from year long data with resampling of one week to month long data with resampling of an hour or 30 minutes. Resampling was used to make the data less variable and to be able to read the autocorrelation and partial autocorrelation plots. Two different time series analysis models were tried with mixed results. To fit ARMA and ARIMA model data was detrended using logarithmic transformation and a differencing. Once transformations were applied, stationarity condition was also checked using the Dickey-Fuller Test.

To pick the correct lags p,q the autocorrelation and partial autocorrelation functions were plotted.

3.3. Topology of Neural Network

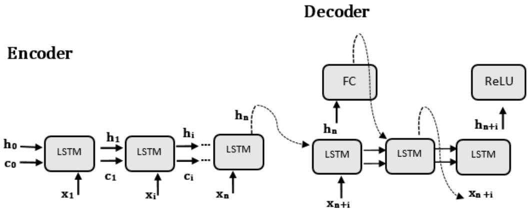


Figure 4. Architecture of our neural network. Encoder and Decoder both use LSTM layers.

3.4. Hyperparemeters for LSTMs

Different hyperparameters were tried for the LSTM newtorks. The batch sizes of 100 and 200 datapoints were tried, learning rates of 0.05 and 0.01 were tried, size of hidden layers of 100, 200, 500 were used.

3.5. Training

Loss is RMSE and training is done through adagrad.

4. Results

This section presents the project results.

4.1. ARMA

Detrending was also performed as can be seen in figure 6 below. Logarithmic transformation of the data was necessary and also differencing helped make the data stationary as ARIMA and ARMA models assume data is stationary.

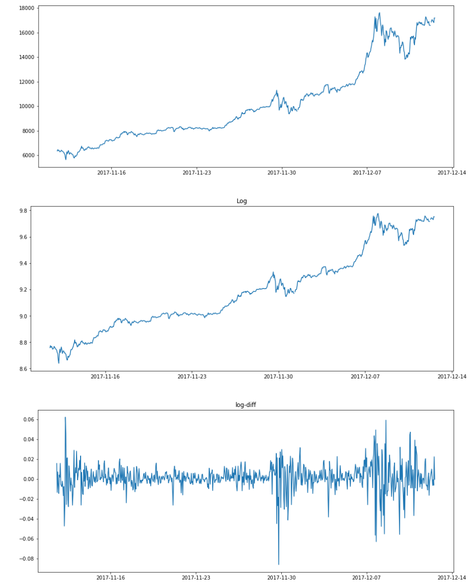


Figure 5. Detrending. Above plot is initial dataset, second plot is a log transformation, in this case trend is almost linear, third plot shows logarithmic and differencing transformations being applied.

StatsModels (python library) was used in order to implement the ARMA model. Since the ARMA model is denoted as Eq. (2.1) above, hyper parameters are p and q which denote the number of previous prices to look at and the number of previous errors the model should consider, respectively. Preliminary results are considered which where obtained by taking p parameter to be in [1,5,10]. ARMA models were fitted for these values on previous 100 time-stamps and then were tasked with predicting the next immediate 4 time-stamps . Actual prices in the next 4 time-stamps with the predicted ones were compared and MSE (mean squared error) was calculated. We sampled the 100 time-stamps uniquely 60 times and found MSE for each sample. MSE for each p considered:

lag, (p)	MSE
1	9.7514974036202844
5	9.4308973740076762
10	9.3925080749340033

Table 1. Change in mean squared error, MSE, due to change in Lag parameter.

Note that, we expected that larger p values would lead to less error since the model has more data to work with (pprevious values) in order to make a prediction.

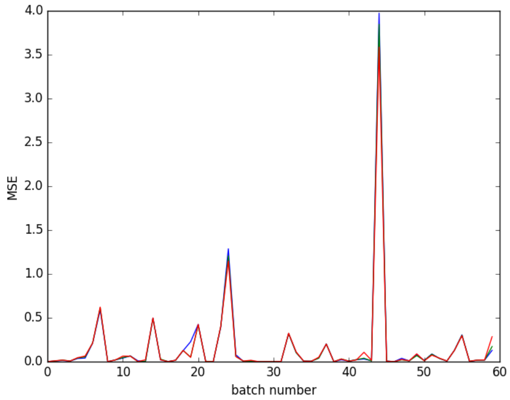


Figure 6. Change in MSE due to change in batch number.

The slight distinction of color in the top most peak (blue, green, red, in that order) represent the MSE for lags p=1, p=5, p=10.

After these preliminary results the autocorrelation and partial autocorrelation functions where calculated and plotted. An example of one such plot is presented in Figure 5 below. The graph indicates that p=1,2 and

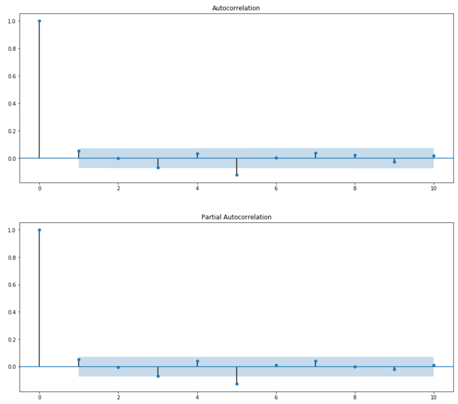


Figure 7. Autocorrelation Function with lags=10. Partial Autocorrelation Function with lags=10

q=1,2 should be the parameter values tried for ARMA and ARIMA modeling of the data.

	MSE
1	9.7514974036202844
5	9.4308973740076762
10	9.3925080749340033

Table 2. Change in mean squared error, MSE, due to change in Lag parameter.

4.2. LSTM

We present some additional results. We start by trying to use dropout and given that we are using only one hidden layer with four inputs the accuracy gets reduced by a lot. Figure 8 shows the results in training and verification set. The predicted results on the verification set is presented in green and the training set predictions is shown in orange. The original data is shown in shown always in blue.
Additionally, dropout was added, however since only one hidden layer is used dropout reduces the overall accuracy of the predictions both in the training and testing set, as expected.

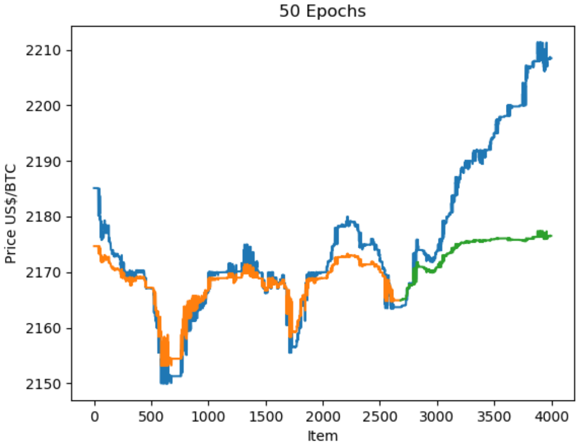


Figure 8. 50 epochs of learning.Loss is reduced from 0.3 to 2.4e-03

In figure **??** we present results for an LSTM layer with 4 units. The neural net was trained for 10 epochs and the results improved significantly to those of figure 1.

When increasing the number of epochs to 20 the training errors decrease to Train Score: 0.95 RMSE and Test Score: 3.01 RMSE. Figure 10 When increasing the number of epochs to 20 the training errors decrease to Train Score: 0.94 RMSE Test Score: 2.55 RMSE. Figure **??** includes recurrent dropout, however the improvements thanks to recurrent dropout are not very noticeable, as evidenced below.

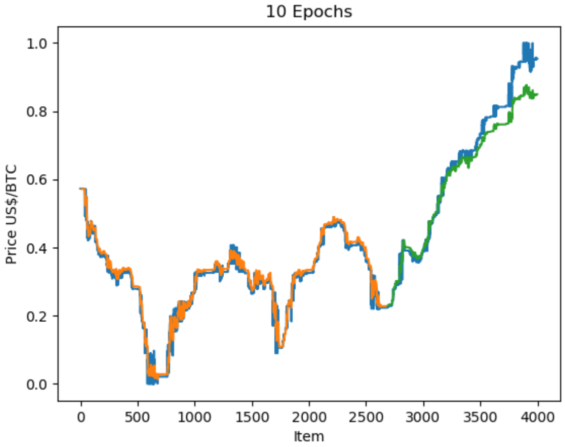
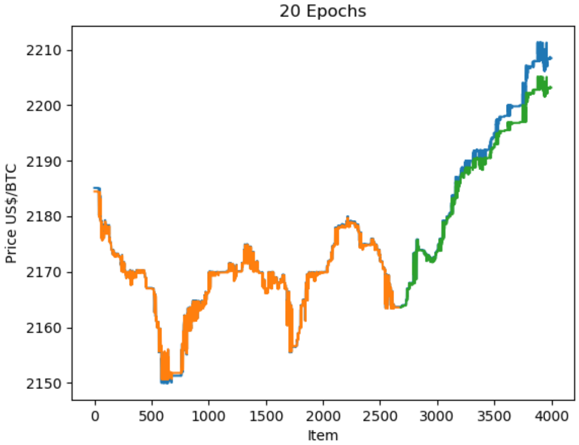


Figure 9. This frog was uploaded via the project menu.



(a)
h

Figure 10. This frog was uploaded via the project menu.

4.3. AutoEncoder

5. Conclusion

[Concluding Remark]

6. Acknowledgments

[ARDAVAN FOR REVIEWING OUR MILESTONES AND PROVING FEEDBACK]

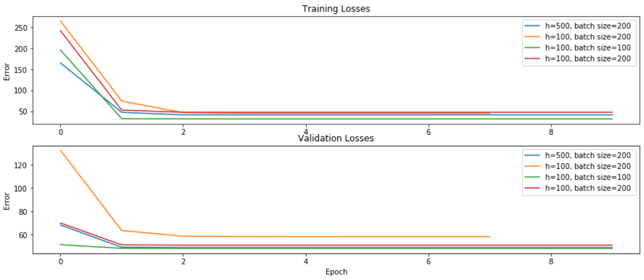


Figure 11. This frog was uploaded via the project menu.

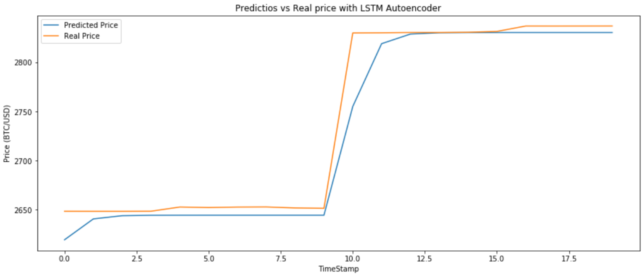


Figure 12. This frog was uploaded via the project menu.

7. References

[Brocwell, Davis] [6.036 Notes] [Hochreiter 1997 LSTM Paper]

Acknowledgments

Do not include acknowledgements in the initial version of the paper submitted for blind review.

If a paper is accepted, the final camera-ready version can (and probably should) include acknowledgements. In this case, please place such acknowledgements in an unnumbered section at the end of the paper. Typically, this will include thanks to reviewers who gave useful comments, to colleagues who contributed to the ideas, and to funding agencies and corporate sponsors that provided financial support.

(**?**)

References