Forecasting Cryptocurrency Price Changes with Artificial Neural Networks

Modulus Global, Inc. - Scottsdale, Arizona

Introduction

Predictive modeling of financial time series data in general is a complex task that has been the focus of intense research in academic and private institutions. These time series dynamics are so difficult to capture because they are known to be extremely volatile and chaotic (Deboeck 1994). Doing so with cryptocurrencies is an even more complicated matter because traders are not known to rely on a consistent source of technical indicators, as traditional financial traders would. Some naive resources might recommend the use of simple, linear forecasting methods, however, financial data rarely meet the assumptions of such statistical models, thus linear modeling techniques such as ARIMA should be avoided.

In order to capture these chaotic and nonlinear dynamics, we employ artificial neural networks to predict (binary) change in, and future prices of, cryptocurrencies. While there is no published procedure for effectively accomplishing this task in a commercial sense, we are able to glean some useful insights from several literature sources. For example, Oancea and Ciucu showed empirically how LSTM models perform better than feed forward neural networks for cryptocurrency price forecasting (2014). Additionally, we chose to use uniform distributions based on n, the number of neurons within a layer, rather than using default weight randomization processes for neural networks:

$$w_i \sim U(-\sqrt{3/n}, \sqrt{3/n})$$

which is suggested by LeCun et. al to decrease necessary training time (2012).

System Description

(i.) Data Collection & Preprocessing

Model architecture is a key component to predicting cryptocurrency fluctuations, but the selection of quality input data is as important if not more. With our extensive experience in financial trading and cryptocurrencies at Modulus Global, we have meticulously chosen data sources that are relevant to price fluctuations in the cryptocurrency market. Specifically, we collect blockchain details from blockchain.com, website traffic data from Amazon Web Services, and relevant keyword/topic information from Google trends. These data are not tested or transformed to force stationarity, for reasons mentioned above. First, we calculate the moving average and the coefficient of determination for daily market prices. Then principle components analysis is conducted on the model training sets to capture 95% of the variance and this transformation is applied to the validation sets. Once the data are reduced, we scale all independent variable values to be between -1 and 1, the relevant portions of the domain for sigmoid functions. Finally, the future price values are scaled between 0 and 1 for the forecasting model and price increases and decreases are set to 1 and 0, respectively

(ii.) Model Descriptions

We have constructed two classes of neural network models for predicting future prices and binary change. The price forecasting model is a shallow LSTM model, following the recommendations of Oancea and Ciucu. The binary change model is a shallow logistic regression model with the expansion of neurons in the hidden layers to improve differentiation between data corresponding to price increases/decreases. In order to avoid the common issue of vanishing gradients with logistic regression, we used a custom activation function, $h(x)$, for the input layer (LeCun et al. 2012):

$$h(x) = 1.7159 \tanh(2x/3) + 0.01x$$

$$\frac{\partial h}{\partial x} = 0.01 + 1.14393 \text{sech}((2x)/3)^2$$
$$\Rightarrow \frac{\partial h}{\partial x} > 0.01, \forall x \in \mathbb{R}$$

(iii.) Model Selection and Training

Let V be the set of validation datasets, L be a set of learning rate values, and D a set of decay rate values. To select the optimal hyperparameter values, we conduct a grid search over all possible combinations with the model validation predictions:

$$(\Lambda, H) = Max_{(\lambda,\eta)\in L\times D}\{\overline{score}(f(v, \lambda, \eta)); \forall v \in V\}$$

where

$$score = \begin{cases} \text{Binary Accuracy} & \text{Logistic Regression Model} \\ -(\text{Mean Squared Error}) & \text{LSTM Model} \end{cases}.$$

Once these hyperparameters are selected, we train the model on a series of sequential training and validation sets, clearing the weights with each iteration. The most recently trained model is then used to predict future changes in Bitcoin prices. We believe that this sliding window method is best to evaluate the performance of models due to the relatively low autocorrelation for Bitcoin prices, as opposed to the k-fold cross validation or bootstrapping methods often used in the literature.

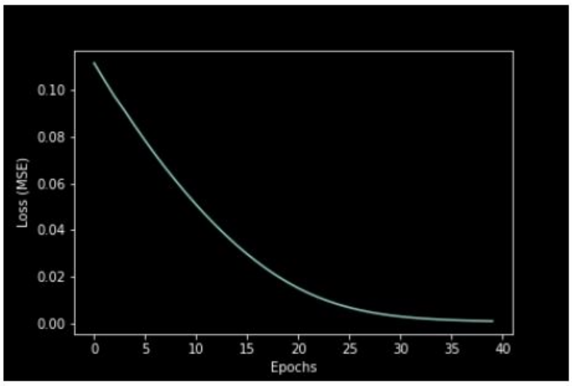
Example Cases

(i.) Setup

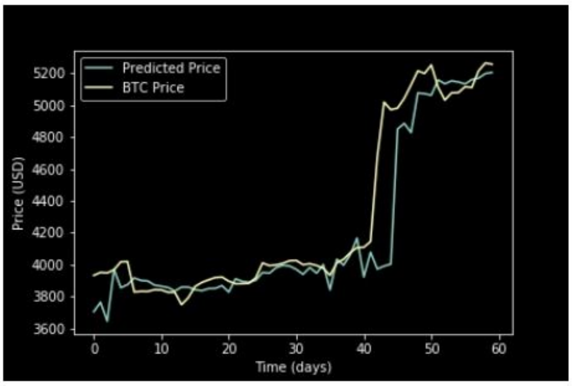
To illustrate our system's ability to predict changes in cryptocurrency prices, we trained out LSTM forecasting model to predict the price of Bitcoin 3 days in the future with 30 days of training data for each iteration. This was done for 20 iterations, producing 60 days worth of predictions.

(ii.) Results

As you can see from figure (a), the model is able to decrease the Mean Squared Error to nearly zero over 40 epochs, yielding predictions that are quite close to the actual prices, figure (b).



(a) Price Forecasting Predictions vs Actual Price



(b) Price Forecasting Predictions vs Actual Price

References

Deboeck, Guido. Trading on the edge: neural, genetic, and fuzzy systems for chaotic financial markets. Vol. 39. John Wiley & Sons, 1994.
LeCun, Yann A., et al. Efficient backprop. Neural networks: Tricks of the trade. Springer, Berlin, Heidelberg, 2012.
Oancea, Bogdan, and S¸tefan Cristian Ciucu. Time series forecasting using neural networks. arXiv:1401.1333 2014.