Proposal – Bitcoin Price Prediction

Gabriel Gal
July – 2018

Domain Background

Cryptocurrencies have risen in both popularity and price since obscure beginning in the late 2000s. An alternative to hard currency, cryptocurrencies were designed to provide secure transactions independent from government monetary policy and traditional banking systems. Bitcoin, the first cryptocurrency, was created by the mysterious Satoshi Nakamoto who created a blockchain protocol that verifies the authenticity of a coin or transaction and records its existence in a public ledger. The blockchain protocol is managed by a peer to peer network where each member attempts to solve a computationally expensive problem to validate Bitcoin transactions. By successfully solving the problem, a member is rewarded with their very own Bitcoin that can be used to purchase goods and services – just like currency. Satoshi Nakamoto's identity is unknown and his involvement in Bitcoin ended in mid 2010 with maintenance and further coin progression left to the wider development community.

The price of Bitcoin is incredibly volatile with large price swings stemming from the demand to use Bitcoin in everyday transactions and the supply of Bitcoins produced from miners authenticating transactions. Government regulation, security vulnerabilities, media coverage, and pure market speculation all influence the price of cryptocurrencies. The price of Bitcoin began trading at just a few cents a coin with little interest from individuals outside the development community. As of January 2018, Bitcoin prices topped $20,000 per coin with large public exposure and dozens of investment products tracking the cryptocurrency's value. Investor purchases of Bitcoin for speculation have caused the price to skyrocket on multiple occasions. Speculation is followed by fear of price overreach and panic as investors sell Bitcoin sending the currency plummeting in value. Many investors believe that technical indicators signaling price movements exist in price charts for most financial instruments including Bitcoin. If this truly is the case, a pattern may be identified in a price chart that can allow an individual to predict the movement of Bitcoin.

Problem Statement

The purpose of this model is to correctly predict the direction, magnitude, and confidence level of price movements in a cryptocurrency called Bitcoin. The price of Bitcoin is publicly recorded relative to the value of the American dollar and updated almost instantaneously. The model will attempt to match this value as closely as possible with predictions for the 5, 15, and 30 minute price movements. A long short term memory (LSTM) algorithm will be created to predict values for each timeframe. The loss function of the model will minimize the mean square error of the predicted prices for each timeframe versus the actual prices during the timeframe. This process should be quantifiable, measureable, and replicable. The value of Bitcoin can be consistently observed and the loss function clearly demonstrating how successful the model is.

Datasets & Inputs

Financial markets record enormous amounts of data attracting researchers eager to utilize data for predictive modeling. Financial data is usually formatted in a time series where the order of the data matters and must flow in one consistent direction. Bitcoin is a financial instrument and uses data typical with other financial assets. At the simplest level, features for Bitcoin include open price, high price, low price, close price, and volume. Bitcoin also has unique features involving the computation required for authenticity calculations. Data for Bitcoin is recorded and summarized every second, minute, and day meaning a researcher could use varied timeframes to create predictions. Other data not directly linked to Bitcoin may be useful for price prediction. These include macroeconomic indicators, stock market indices, gross domestic product figures, and interest rates. Certain commodities such as gold and silver serve as a means to store value – along with their respected commercial applications. Bitcoin may be partially correlated to these and other financial assets that serve as a way to store value. Finally, market wide sentiment reflected in news articles, twitter posts, and online search counts may be predictive of Bitcoin price. Positive news reports about Bitcoin or an increase in online searches for the currency could suggest higher future demand which translates to a higher price.

Several different features will be used for this model. Simple financial features will include the ones previously mentioned – price and volume. This data is updated daily through a user on Kaggle.com, but can be substituted by various Bitcoin exchanges (1). The model will use different ratios, calculations, and metrics both unique to Bitcoin and common in financial markets. A full list features as well as their calculations is presented in Exhibit 1. These calculations make use of z scores to determine how far from the average a metric is (2). The hope is that these unusual metrics are correlated with either large gains or losses in the value of Bitcoin. Network value to transactions ratio is a metric only applicable to cryptocurrencies and is calculated to determine if current valuation is supported by the level of transactions. The data necessary to calculate the ratio is sourced from Blockchain, a software company specializing in digital assets (3). Additionally, several common and uncommon technical calculations are used. As implied, the data will be used as features into a LSTM neural network to help with Bitcoin price prediction.

Solution Statement

LSTM neural networks can be a viable solution to the price prediction problem. LSTMs are recurrent neural networks where the sequence of inputted features is remembered and used for prediction. Given that the value of Bitcoin flows in a time series, a recurrent neural network would capture the sequential information present in Bitcoin and its features as value changes over time. Unfortunately, standard recurrent networks are unable to learn long-term dependencies as loss functions used to optimize the model exponentially decay. Fortunately, LSTMs solve this issue by passing along new gates to the network that preserve the information found in long stretches of data. This is ideal for Bitcoin prediction since previous price trends may repeat in the future.

Benchmark Model

Three different benchmarks will be used to determine the reliability of the LTSM neural network including a exponential smoothing strategy, moving average strategy, and an autoregressive integrate moving average (ARIMA) strategy. These benchmarks are typical rule of thumb methods analysts use to determine the future direction of any financial asset. The exponential smoothing strategy uses the weighted average of the previous data points with progressively less weight assigned to older values. Within the calculation, alpha is the smoothing parameter from zero to one that defines the weights. A value close to one places more emphasis on current values while a value close to zero places more emphasis on older values. A 96 lookback period will be used for this benchmark. A moving average strategy is similar to exponential smoothing, but all the values have an equal weight within the calculation. A lookback window of 96 will be used for this benchmark. The ARIMA benchmark is a sophisticated tool that splits price movements into autoregressive and moving average parameters. The autoregressive portion explains that the current value of an asset is partly a function of its previous value or values and some random error. The moving average part explains the portion of the current value that is just a function of the random error. The entire process takes into account the asset's trends, seasonality, cycles, errors, and non-stationary aspects of the data to create a forecast. A 96 period lookback will be used for the ARIMA benchmark. All these benchmarks values will be compared to the actual price of Bitcoin and the prediction given by the LSTM network.

Evaluation Metrics

Mean squared error (MSE) will be used to evaluate the performance and effectiveness of the model. The MSE metric finds the deviation of values from the predicted price produced from the model and the actual value of Bitcoin. Such a metric is relevant since it will attempt to find the peaks and troughs within a time series. Other metrics such as mean absolute error will not place as much emphasis on price outliers which are very relevant to a Bitcoin investor. This metric will also be calculated for the benchmark ARIMA strategy. The mean squared error for the entire time series in the LSTM network is updated at each iteration of training and compared to the benchmark error. As the model trains, the model should slowly converge to a model that minimizes MSE.

Project Design

1. Data Preprocessing & Feature Engineering

The simple Bitcoin data which includes price and volume is housed on a csv file that will be directly imported into python using a common importer module (1). The Bitcoin data will be scrubbed for missing values, transcription errors, and other abnormalities. Additional Bitcoin data will be imported from various different json links housed in the blockchain website (3). The dataset is too large to import completely into a recurrent neural network, so the data will be split into batches using a generator. This will improve training speed and model development. Values in each of the batches will be normalized to ensure the network can correctly backpropagate and the model can optimize to a value. The clean batch will then flow through to the LSTM network.

2. Model Implementation

The data has different time series frequencies with some sets quoted minutely and others daily. To address this asynchronous data, a phased LSTM network will be used (4). Phased LSTMs add a new time gate to the normal LSTM network. The new gate is controlled by a parametrized oscillation which updates the values of the cell only during a small portion of the entire LSTM cycle. This allows the minutely data to flow into the model uninterrupted while the daily data updates the cell only when the new gate is open – this will coincide with a daily frequency time series.

The network will be composed of two to four LSTM layers depending on the model performance with each successive layer. Dropout will be applied between each layer to ensure the model can generalize and not overfit the data. Batch normalization will be applied afterwards helping reduce the impact of previous layers by keeping the mean and variance fixed. This allows the optimizer to converge faster as each layer has more stable input distributions. A timedistributed layer will be added next to allow the model to use previous predicted values as inputs into subsequent predictions. The loss function of the model is mean square error where the distance between the predicted value versus the actual value is minimized. Adam will be used as the optimizer providing an all around effective algorithm to update the network weights during training.

3. Training & Prediction Comparison

The combined dataset will be split between training, validating, and testing 70%/15%/15%. The parameters for the phased LSTM will be tweaked to determine which combination give the best results. The model will start with 2,000 epochs to ensure the data is fed through enough times for the model not to underfit. Initially, the dataset will be divided into 1,000 input batches that will iterate through various times to equal a epoch. There will be 200 hidden nodes and three output timesteps for predictions of 5, 15, and 30 minutes. The learning rate will start at 0.001 and the dropout rate will be set at 0.20. Training will use these parameters and will be tested for accuracy using the validation set.

Three baseline models will be calculated to compare the new phased LSTM model to. They include an exponential smoothing strategy, moving average strategy, and an autoregressive integrate moving average (ARIMA) strategy. These model will be imported to python using various packages. These baseline models and the phased LSTM network will be tested on the testing set to see which model performs best. Comparison charts will be created to compare the results of the different models.

Sources

(1) Zielak, (updated daily) Bitcoin Historical Data. Retrieved from http://www.kaggle.com
(2) Poon, Vincent (Sept. 6, 2017) Trading Bitcoin with Reinforcement Learning. Retrieved from https://launchpad.ai/blog/trading-bitcoin
(3) Smith, Peter (updated daily) Blockchain Charts & Statistics API. Retrieved from https://blockchain.info/api/charts_api
(4) Daniel Neil, Michael Pfeiffer, Shih-Chii Liu (Oct. 29, 2016) Phased LSTM: Accelerating Recurrent Network Training fo Long or Event-based Sequences from https://arxiv.org/pdf/1610.09513.pdf

Exhibit 1 - Features List

Simple Features R = return | P = price | V = volume | W = window

Name	Description	Type
r	R	Return
r1	r from 1 period prior	Lagged return
r2	r from 2 period prior	Lagged return
rZ12	Zscore (r,w) where w = 12	Price level
rZ96	Zscore (r,w) where w = 96	Price level
pma12	Zscore (p/ave(p,w1)-1,w2) where w1 = 12, w2 = 96	Change in price
pma96	Zscore (p/ave(p,w1)-1,w2) where w1 = 96, w2 = 96	Change in price
pma672	Zscore (p/ave(p,w1)-1,w2) where w1 = 672, w2 = 96	Change in price
ma4/36	Zscore (ave(p,w1)/ave(p,w2)-1,w3) where w1 = 4, w2 = 36, w3 =96	Change in price
ma12/96	Zscore (ave(p,w1)/ave(p,w2)-1,w3) where w1 = 12, w2 = 96, w3 =96	Change in price
ac12/12	Zscore ((p/ave(p,w1))/ave(p/ave(p,w2),w3),w4) where w1 = 12, w2 = 12, w3 = 12, w4 = 96	Acceleration in price
ac96/96	Zscore ((p/ave(p,w1))/ave(p/ave(p,w2),w3),w4) where w1 = 96, w2 = 96, w3 = 12, w4 = 96	Acceleration in price
vZ12	Zscore (v,w) where w = 12	Volume level
vZ96	Zscore (v,w) where w = 96	Volume level
vZ672	Zscore (v,w) where w = 672	Volume level
vma12	Zscore (v/ave(v,w1)-1,w2) where w1 = 12, w2 = 96	Change in volume
vma96	Zscore (v/ave(v,w1)-1,w2) where w1 = 96, w2 = 96	Change in volume
vma672	Zscore (v/ave(v,w1)-1,w2) where w1 = 672, w2 = 96	Change in volume
vol12	Zscore (std(r,w1),w2) where w1 = 12, w2 =96	Volatility level
vol96	Zscore (std(r,w1),w2) where w1 = 96, w2 =96	Volatility level
vol672	Zscore (std(r,w1),w2) where w1 = 672, w2 =96	Volatility level
dv12/96	Zscore (std(e,w1)/ave(std(r,w2),w3),w4) where w1 = 12, w2 = 12, w3 = 96, w4 = 96	Change in volatility
dv96/672	Zscore (std(e,w1)/ave(std(r,w2),w3),w4) where w1 = 96, w2 = 96, w3 = 672, w4 = 96	Change in volatility

Network Value to Transactions Ratio (NVT)

$$NVT = \frac{Market\ Capitalization}{USD\ Daily\ Transaction\ Value}$$

Network Value to Transactions Signal (NVTS)

$$NVTS = \frac{Market\ Capitalization}{90\ Period\ Moving\ Average\ of\ USD\ Daily\ Transaction\ Value}$$

Moving Average Convergence Divergence (MACD, Signal, Histogram)

$$MACD = 12\ Period\ Exponential\ Moving\ Average\ of\ Close\ Price - 26\ Period\ Exponential\ Moving\ Average\ of\ Close\ Price$$

$$Signal = 6\ Period\ Exponential\ Moving\ Average\ of\ MACD$$

$$Histogram = MACD - Signal$$

14 Period Relative Strength Index (RSI)

$$RSI = 100 - \frac{100}{1 + RS}$$

$$RS = \frac{\frac{(Previous\ Average\ Gain * 13) + Current\ Gain}{14}}{\frac{(Previous\ Average\ Loss * 13) + Current\ Loss}{14}}$$

Bollinger Bands (Middle Band, Upper Band, Lower Band)

$$Middle\ Band = 20\ Period\ Moving\ Average\ Price$$

$$Upper\ Band = Middle\ Band\ +\ 20\ Period\ Price\ Standard\ Deviation * 2$$

$$Lower\ Band = Middle\ Band\ -\ 20\ Period\ Price\ Standard\ Deviation * 2$$

Stochastics (K, D)

$$K = \frac{Price - 14\ Period\ Low\ Price}{14\ Period\ High\ Price - 14\ Period\ Low\ Price}$$

$$D = 3\ Period\ Average\ K$$