Machine Learning Engineer Nanodegree

Capstone Proposal

Benny Late
August 6, 2018

Proposal – Using Machine Learning to Predict Bitcoin Closing Spot Price

Domain Background

Bitcoin, and other cryptocurrencies, is the hottest thing around. Bitcoin is designed to be a decentralized "currency" with transactions verified and recorded in a public distributed ledger (blockchain) without the need for a trusted record keeping authority or central intermediary. Transaction blocks contain a SHA-256 cryptographic hash of previous transaction blocks, and are thus "chained" together, serving as an immutable record of all transactions that have ever occurred.

Market sentiment and the availability of exchanges have allowed individual investors to get into the game for BTC. In 2017, this pushed prices to astronomical returns. This has led to interest in crypto from established financial institutions, securities regulators, and even governments.

Given the increased interest, a number of attempts have been made to predict prices. Simeon Kostadinov (of Speechify) has a great rundown using RNN and ML to show the predictions, not so accurate though. https://towardsdatascience.com/bitcoin-price-prediction-using-lstm-9eb0938c22bd . Devika Mishra (of dataturks.com) has a great illustration not only of ML modeling for bitcoin, but of ARIMA and time series data.

Problem Statement

In 2018, we've seen billions of dollars in BTC investments disappear and prices have stabilized far below all-time highs. There are innumerable ways to try and predict the next spot price, but typical performance is not much better than a coin flip. That looks like a basic Naïve Bayes result, and I know we can do better! I want to create a model that can predict closing price for a single day. This model should exceed 60% prediction success. I enjoyed the reinforcement learning and deep learning sections, so I'm looking to implement RL and Deep Neural Networks.

Because our data and target features all are numbers, and can be any real number, this will be a regression problem.

Datasets and Inputs

Bitcoin price data is at 1-min intervals from the most trusted exchange, Coinbase, select exchanges, Jan 2012 to July 2018. My excel file contains the following data in more than 1.8 million rows:

- Open- Bitcoin price in Currency units at time period open
- High- Highest Bitcoin price in Currency units during time period
- Low- Lowest Bitcoin price in Currency units during time period
- Close- Bitcoin price in Currency units at time period close
- Volume (BTC) - Volume of BTC transacted in time period
- Volume (USD)- Volume of Currency transacted in time period
- Volume-weighted average price (VWAP)

Solution Statement

Fortunately, the minute by minute data allows us to see volume and price trends and volatility. We should be able to see how the model deals with this information and potential correlation with volume so we can achieve model predictability to 60% for closing spot price.

Given the nature of time series data, I will look to use a large same to train with. Because I can't random sample from data points which are dependent upon the timestamp, I will use the price data up through January 2018. I'll keep February 2018 data from the test set to avoid look-forward bias, and then use March-July 2018 data as my test group.

I will use an ARIMA regression model to start, and then I will create a Long Short Term Memory neural network and try replacement learning to see which one achieves 60% or better prediction for my target metric with optimal error/confidence levels.

Benchmark Model

I will use a standard ARIMA regression model.

Evaluation Metrics

I'll be targeting Closing price and volume weighted average price.

For model evaluation, I'll be using model evaluation metrics like RMSE, R2 score, explained variance score, and I'll use cross-validation to help confirm model viability.

Project Design

In the first phase of the project, I'll check the dataset to make sure the rows are usable.

Once the data is clean, I'll bring this together and do feature analysis and if any features are closely correlated and can be removed from the modeling.

I'll benchmark with the ARIMA model and then create a neural network model. Because of the sequence dependency and the vanishing gradient problem, I'll use a Recurrent Neural Network model, specifically a Long Short Term Memory model.

Finally, I'm going to explore reinforcement learning. For the action space, we will allow either 3 options –

- Buy
- Sell
- No_Action

The state space will be based on our budget of 10,000 USD. If the bot predicts correctly, it receives the difference between the buy and sell price. If it predicts incorrectly, it receives the buy and sell price difference (in the case a negative number).

We don't have information on the available bitcoins or current demand, so we will operate under the assumption that whatever order we place will be filled at the price the model expects.