

# Machine Learning Nanodegree

## Capstone Proposal

Arun Dass

December 27, 2017

### **Domain Background**

Cryptocurrencies are quickly becoming a popular trend at this time. There are currently over 1000 different cryptocurrencies that are available across the web, with the most popular and polarizing being Bitcoin. Bitcoin has seen a massive surge in price over the last five years, and has created immense wealth for people who have invested early. The top 4 cryptocurrencies by market cap at the time of writing this proposal are Bitcoin, Ethereum, Bitcoin Cash, and Ripple.

### ***Personal Motivation***

With the vast amount of wealth that has been generated with early investments in cryptocurrency, it is a great opportunity for investors to increase wealth by understanding how and why prices fluctuate with certain cryptocurrencies. Therefore, it is my personal motivation to understand how to make similar investments.

### **Problem Statement**

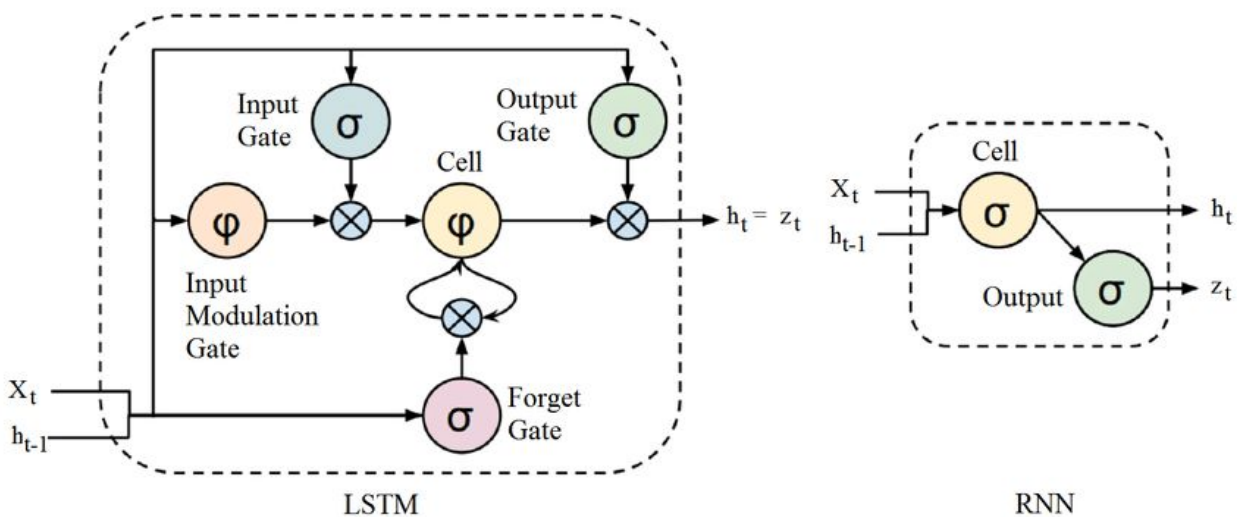
Due to the high volatility of the cryptocurrency markets, there is a very good opportunity to use the large swings in price to generate profits. The goal of this model is to be able to predict when a price is low, which is an optimal time to buy. As well as predict when the price is at a high, which is an optimal time to sell.

## Datasets and Inputs

I will be predicting the closing price of a chosen cryptocurrency, and will be using the standard trading features for trading currencies, stocks, along with information from Bitcoin. The features to be used include the daily volume, high, low, and close. This information will be accumulated from the Alpha Vantage API. In addition, I will be adding daily Google Trend data for Bitcoin and the chosen cryptocurrency. This information will be accumulated using the PyTrends API. Google trends is a public web facility that shows how often a term has been searched. This can be a useful tool when looking into how much “hype” a cryptocurrency is receiving, and if it will reflect the price.

## Solution Statement

Recurrent Neural Networks, specifically, Long Short Term Memory Networks, are a great tool for analyzing time series data. As there is a vast amount of web information regarding currency metrics, many find it fascinating and lucrative to be able to predict the future prices of investments. The LSTM is a version of the RNN that has a built in long and short term memory as well as a forget gate.



The potential solution is to create a LSTM model to take in a dataframe of historic data and train the model to predict the future price of the cryptocurrency in question. Knowing this information would put the creator in an advantageous position to increase the value of their investments.

## Benchmark Model

The selected benchmark will be the most recent close data used as ground truth for the cryptocurrency to be predicted by the model. This data will be the previous 30 days of actual close price for the cryptocurrency chosen.

## Evaluation Metrics

### *Root mean squared error(RMSE)*

RMSE is the standard deviation of prediction error and is a measure of how far the data points are from the regression line.

The formula is:

$$RMSE = \sqrt{(f - o)^2}$$

Where:

- $f$  = forecasts (expected values or unknown results),
- $o$  = observed values (known results).

Image sourced from <http://www.statisticshowto.com/rmse/>

Root mean square error is a commonly used metric when measuring the differences between values obtained from a predictor/model and values actually observed.

## Project Design

Programming Language: Python 3.5+

Libraries: SkLearn, Math, Keras, Tensorflow, Numpy, Pandas, Matplotlib

Process:

- Begin by wrangling the appropriate features from Alpha Vantage and Pytrends API
- Clean the data for pre processing
- Preprocess the inputs and label for correct input format into the LSTM
- Build the Long Short Term Memory model using Keras and Tensorflow
- Feed the the inputs into the model for training
- Predict the test data and score using Root Mean Squared Error metrics
- Plot predictions and true data using matplotlib