Wednesday, November 17, 2021

This report is not yet complete.

Trading Cryptocurrency with Neural Networks

Jashua Luna

Abstract

Neural networks are state of the art mathematical models. In this investigation, several models were evaluated based on their performance predicting the price of cryptocurrency. Ultimately, a transformer network was developed to predict changes in price with time. Evaluation criteria were prediction accuracy and performance in a simulated trading environment.

Table of Contents

Introduction

Trade Optimization

How Models are Evaluated

Models Tested

Results

Introduction

Cryptocurrencies are volatile assets whose prices are known to fluctuate wildly with no discernable pattern. Traders use mathematical models to regularize price data to make assessments and predictions. Could a neural network with training, develop its own mathematical understanding of price movements and be able to make accurate predictions and maximize profits from trading. Several network models are herein considered and evaluated for their performance, both in prediction accuracy, and trading results in a simulated market. Beyond different network architectures, two distinct methods for generating data from raw price data are considered. Traditional price interpretations - like exponential moving averages, moving average convergence divergence, or stochastic relative strength index – are purposely avoided while generating network data; the network model is expected to develop its own understanding of price changes.

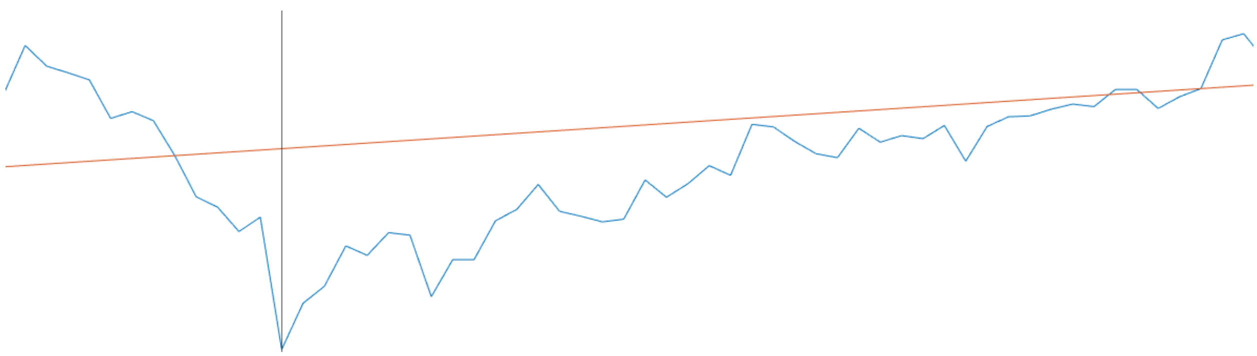
In practice, a trader can only buy or sell an asset. To profit from both increases, and decreases in price, a trader can long and short assets. A long is the act of purchasing an asset at a price with the expectation of selling it at a higher price. A short is the act of selling an asset, with the expectation of purchasing an equal amount to that sold at a lower price. If a trader could make trades with one hundred percent accuracy, they would at any given moment, either be in a long or a short. The first data model aims to have perfect accuracy in this sense, and the aim of this model is to transform price data to the corresponding long or short signal.

The other data model considered converts changes in price with respect to time, to a sequence of integers. Frequency bins corresponding to the histogram of the assets changes in price are used to determine the integer at a given time. This model's explicit task is to predict the next time step of the price change sequence.

Trade Optimization

Considering the price of the asset in question as unchangeable by the network's actions, a model can learn to maximize profits my maximizing the length of a line made between selected peaks and valleys on the assets price-time data; trades are made at peaks and valleys (buy at valley, sell at peak). In an ideal scenario, where trades are executed the instant the network makes a request, at the ideal price, and with no trading fees, a network could profit on a millisecond-to-millisecond basis, since maximum line length results in maximum profits. However, this is not the case. Request-response lag between the network and the trade platform, and trading fees make it such that a trade must be of a certain size for it to be profitable.

Since the network has two output states – long and short – the assets price data can be reduced to a binary signal: one for long, zero for short. The proposed networks are therefore transfer functions which convert the asset's price data to this binary signal from which trades are made.



For network training, historical price data is converted into its binary representation; that also satisfies the minimum trade size constraint. For this, a recursion is performed where a new point is inserted between the point at time t, and its right-hand neighbor, until a stopping condition is met, or the end of data reached. For this investigation, the stopping condition is that the trade (local maximum or minimum) to the right of the current trade, can be no closer than t + n minutes (data is sampled at one-minute intervals); several values of n were tested with different results. The new point is inserted at the maximum of the absolute difference between the secant – of the point at time t and its right-hand neighbor – and the price-time data. Once the recursion is complete, the binary signal is determined by the slopes of the lines between trade-points; long where slope is positive, short where negative.

How Models are Evaluated

Evaluation metrics are:

Training Accuracy & Training Loss

Trading performance on training sample of price data.

Trading performance on non-training sample of price data.

A simulation environment was created to test a trained network's profitability. An interface was created that would accept price data one time interval at a time. This interface uses a trained network to decide whether to enter a long or short position (and exit the previous position) and maintains a record of all entries and exits. When the simulation is complete, profits and win/loss ratios are calculated. For simplicity, it is assumed that each position is of the same size. Each model was run through a simulation over different samples of price data.

Tested Models

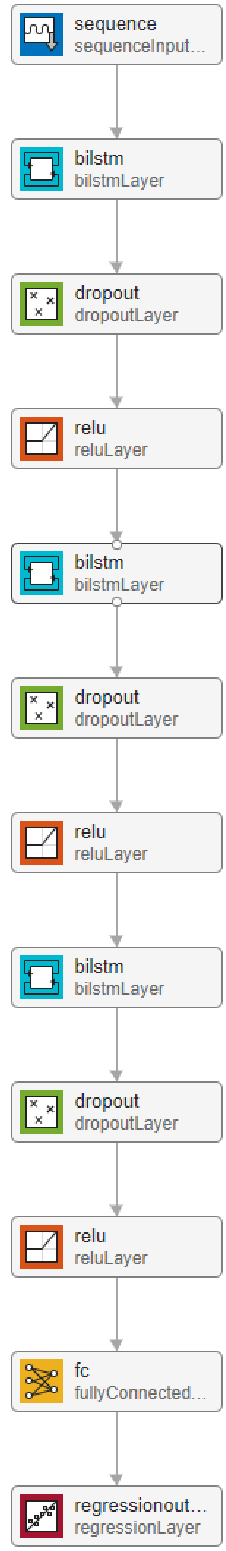
Each model is trained and evaluated ten times.

Different Window Sizes and Prediction Lengths

10,000 samples

Model One

Three Repeating blocks of bilayer lstm (128 hidden layers), dropout layer (0.2 probability and Relu layers. Followed by one fully connected layer which outputs 1 channel.



Results