Machine Learning Nanodegree

Capstone Proposal

Michael Holberger
10/1/2018

Financial Market Combined Indicator using LSTM Model

Domain Background

Since the invention of computers, people have been engineering ways to use them to create an edge in investment markets. Being able to do calculations quickly, using current data enables traders to make educated trading decisions on when to enter or exit a market. Algorithmic trading has been a growing practice in recent decades as computational technologies became capable of employing high frequency trading strategies. Today, high frequency trading dominates public market trading.

Cryptocurrency markets democratize and accelerate this practice, in that many of the barriers for high-frequency trading in stock markets do not exist in the popular “alternative-coin” exchanges hosted on internet. Using the programmatic interfaces for these exchanges thousands of trades can be made daily, incurring minimal fees. Traders that employ systems to make split-second decisions (bots) have permeated these markets.

The incentive to create profitable trading system is primarily financial. Ideally an automated system can be created to gather current data, formulate predictions on market direction on which to trade, execute those trades, all with minimal intervention required by the user. However, a system which is perfectly profitable in all conditions is highly unlikely, as the stochastic movement of price line data his highly unpredictable, and is the equivalent of what is known as a “random walk”. No single technical analysis (TA) indicator that currently exists can be relied upon alone. However, it may be possible that using some combination of indicators will supplement the trader’s judgement, allowing them to make a career out of trading.
Many techniques for analyzing a market based on price data currently exist. New, machine learning algorithms offer the possibility of dramatic improvements. Some popularly used technical analysis indicators include Relative Strength Index and Moving Average Convergence/Divergence, which are calculated using price data from a set of previous intervals.

Problem Statement

This project uses machine learning techniques to attempt to predict the short term movement of certain cryptocurrency trading prices using multiple technical indicators, so that short term trades can realize profits in volatile market conditions. Success is measured by comparing the results of short term trades indicated by the model with buy-and-hold trading.

Datasets and Inputs

Price data was collected on the cryptocurrency assets traded against BTC on the Bittrex.com marketplace between Nov 2017 and June 2018. Data is organized into 5-minute period samples (referred to as *candles*) which include the opening and closing prices for that period, as well as the highest/lowest prices and trade volume.

This data is then pre-processed using several traditional technical analysis indicators such as Exponential Moving Averages (EMA), Commodity Channel Index (CCI), and Moving Average Convergence/Divergence (MACD), which are calculated using a selected number of previous data points. These indicators will be added as features alongside the price data as inputs into our predictive model.

Solution Statement

Using the price data obtained from several cryptocurrency markets, pre-processed for technical indicators commonly used by career stock market traders, we will implement machine/deep learning techniques to develop a model that will make price predictions in much the same way a successful human trader would. This will allow us to employ a more educated and profitable trading strategy.

Benchmark Model

The completed model will be used to make predictions several steps ahead in time. These can then be compared to a buy and hold strategy. During the time in which the data was collected, markets were generally falling in price, causing the buy-and-hold strategy to generally result in losses. Using the model to predict movement should allow us to make trading decisions that result in greater profit than a buy and hold strategy. By plotting the predicted price line against the true price data, it will then be apparent whether or not the predictions are useful.

Evaluation Metrics

Mean Squared Error (MSE) will be used to compute loss for each batch of time-sequences within each training session. For each training epoch, the model trains through 40 different sets

of market data, divided into training and cross-validation sets, from start to finish. After training a market data segment, the training MSE is recorded. Then point-by-point predictions from the cross-validation segment of that market's data are evaluated and recorded. The MSE results for training and cross-validation are then plotted for analysis. After the completion of several epochs, a final set of data reserved for testing will be used to make predictions to be evaluated against our previously stated benchmark.

Project Design

The data must first be pre-processed for the technical indicators that have been selected. This will be done using the python wrapper for TA-Lib: a programming library which includes a large collection of methods for calculating indicators, and is considered industry standard by financial technology developers. These new data features will then be added as columns to the dataset.

The nature of the price data is non-stationary, so the data must first be normalized before it is used to train the model. There are various methods for data normalization which must be tested for efficacy. Some techniques for data normalization that will be tested include standard scaling, which employs averaging, and MinMax scaling, which normalizes data based on its highest and lowest values. After either of these methods of normalization, a wavelet transform will be applied to each column and preliminarily tested to see if its application makes any significant improvement in prediction accuracy. This will be done by training a simple model with default parameters on a section of our total data. The prediction results will then be plotted and assessed based on factors such as preliminary MSE scores, prediction line accuracy, and how clearly the pre-normalized price data is represented after normalization.

In preparation for training the model, the data, now including TA indicators, is divided into batches of a specified number of consecutive data points. Data windows are generated frame-by-frame within each batch. The model is designed to predict one step ahead for each of these data windows. Each data window/sequence is then normalized before being input into a neural-network model, including several Long/Short Term Memory layers for training. LSTM layers were developed to be more effective in making predictions based on time-sequenced data than regular, fully-connected neural-network layers. The model's predictions will then be evaluated by using the segments of data pre-designated for cross-validation. Training will resume until it appears the model is making effective predictions, without overfitting the dataset. Different combinations of layer/node organization and hyper-parameter configuration will be trained and evaluated in this way for predictive power.