Research Project Documentation

vanGoethem, Joren Maerten, Andreas

January 31, 2022

Contents

1 Data Collection

We collect data from several sources in this project, crypto ticker info [**Open, Close, Low, High, Volume**] and along with that we collect data from Reddit, and RSS feeds.

1.1 Crypto Data

For our crypto data we've used the Binance Exchange as they have an API that allows us to fetch from the beginning till the end of all of their symbols. Their API ratelimits are pretty strict but as long as you keep this in mind you should able to fetch nearly every symbol in a day.

For context we're able to collect 15.3GB of raw minute candles from Binance. In the end we had over 360 different symbol pairs with USDT and our AI can properly train on this.

To use the collector you need to go to the Binance API management page, create an API-key and put it inside the crypto.py file.

Aside from that nothing will need to be changed and you can just run the file, it will start collecting all data from symbols trading with USDT.

1.2 RSS feeds

RSS Feeds don't require any API keys, they can be fetched by anyone at any point. We were unable to utilise this collector very well since there is no way to collect historical data, one needs to continously collect data to get any proper dataset. Also due to the data being collected from multiple different sources the formatting of the data can be vastly different which makes processing the data more difficult.

We have selected quite a few crypto related RSS feeds which we found on the following blog: feedspot

1.3 Reddit

For Reddit we've collected data from crypto subreddits, the ones we've chosen seem to contain the most general data about crypto as collecting data from specific crypto tokens/coins might skew the sentiment of the data collected.

So we've gone with the following subreddits:

- CryptoCurrency
- CryptoCurrencies
- bitcoin
- altcoin

With Reddit we were unable to collect old data as we can only fetch data from subreddits through tags "hot", "new" or "random". We were able however to get the 25 newest posts but then again for us to collect archival data we would need to collect this over a really long period and the data collected could be of very low quality since anyone is able to post at any time.

There exist archives of Reddit online but there is no information on the structure available and these are fairly large for just textual content so for now we will not be using this data to train our model but it might prove to be a useful expansion to our research.

1.4 Twitter

We initially tried to get data from Twitter their API but were unable to do so since they did not respond to our emails for a research oriented API key. This would allow us to collect data from Tweets in the Twitter archives, without this access we would not be able to gather enough significant data about crypto related tweets.

2 Ticker Timescale Swap

Optionally we have made a program with which you can swap a smaller time frame (1 minute) to another larger time frame (5min, 15min, 30min).

With this we can test how our artificial intelligence trains on a different time frame or additionally training on one time frame (1min) and attempting to make predictions on another time frame (5min).

By default the timescale ratio is 1:60, this is because our input data were 1min tickers from which we wanted to make 60min a.k.a. 1h tickers, but in theory any variant is possible 1:7, 1:15. In the end we stuck with 1 minute candles because the model that was trained on this time frame was able to make more predictions than any other model.

3 Data Pre-processing

3.1 Indicators

To augment our data for improved results we added technical indicators to our data. We have done extensive research into what indicators we should be using and what types of indicators exist and when which indicator is usefull. We have previous experience in making algoritmic trading bots using indicators only so we had some knowledge about these before starting this research project.

The types of indicators used are:

- Momentum Indicators
- Trend Indicators
- Volatility Indicators
- Overlap Studies
- Volume Indicators

More information about the indicators we used and their formulas can be found in our Indicators Documentation.

3.2 Normalization

The raw price data is not suitable for training a neural network with. We need to normalize all the data so the neural network can properly adjust to the data. For this we will be using the percentage change from one data point to the next for most of our data, for indicators that are already on a scale of 0 to 100 we will just divide this by 100 to get them on a scale of 0 to 1.

The following formula can be used to calculate the percentage change, where P is the percentage change, V1 is the previous value and V2 is the value of which we want to calculate the percentage change over V1. The result can be a positive or a negative value.

$P = \frac{V_2 - V_1}{V_1} \times 100$

3.3 Labeling

We used some basic logic to label our data with buy and sell targets for our model to train on, these targets should be the 'perfect' result. Since we can place these after seeing what comes next it is easy for us to determine what the best buying and selling points are. The AI will have to try and predict this which is a way more difficult task.

4 AI models

4.1 Training

We started using Pytorch to try and make a model that used Reinforcement Learning for training. However we quickly stumbled into problems with pytorch and LSTM layers, apparently the latest few versions of Pytorch have memory leaks in LSTM layers causing our model to crash after a short while. We then switched to Tensorflow but were not satisfied with the training speed of reinforcement learning and quickly moved on to labelled training which was a big improvement in training speed. This was probably the best choice because of the limited time we had for this project.

4.2 Long Short Term Memory

Our model is made up of mostly LSTM layers with 2 final Dense layers. We feed a series of data through our model, usually consisting of multiple hours or even days of data, depending on whether we were testing minute candles all the way up to hour candles. We tried multiple configurations of models ranging from 3 LSTM layers up to 10 LSTM layers. After these always a smaller dense layer and an output layer of 3 neurons with a softmax activation. One for each possible action: Hold, Buy and Sell.

4.3 Sentiment Analysis

For sentiment analysis we used a pretrained model from Huggingface, this model showed good results on our small dataset of reddit and rss feeds but sadly we did not have enough data to properly use for AI training. This would however be an interesting expansion to look into in the future with sufficient data.

4.4 Model Testing

To test and compare our models we let them make predictions on parts of our dataset and plot their predictions on charts to see how they perform. We also calculated their profit/loss with the predictions they made and compared these to each other. In general our largest model performed the best so it is possible we could achieve better results with a larger model but for that we would need a bigger GPU with more VRAM and more training time. Our bottleneck up to now was our VRAM capacity since this was very quickly filled up by our timeseries sequences. We could mitigate this by decreasing batch size even further but this would significantly increase training time which was already very long.