

Project 3: Final Cryptocurrency Trading

Sixian HUO

shuo@connect.ust.hk

1. Introduction

Cryptocurrency trading is a very interesting topic. Different from traditional risky assets, cryptocurrency doesn't have a reasonable pricing method. How much it's worth depends on how crazy people are.

In this project, I try to build a deep neural network based on technical analysis and use it to predict whether the return of next bar is positive. There are five layers of this DNN and each of them use sigmoid as activation function. The input vector has 18D and the output is 2D, which represents the probability of positive and negative.

2. Data Preprocessing

1-day bar

Unfortunately, I tried to use 1-minute data to fit the model but at last I found that it was hard to reduce the loss and the training process was time-consuming. So I have to use the lower complexity daily data, and 80% of them to train, 20% of them to test.

Features

I think the separate price data is meaningless. Here I used the log price, so that the difference between them can approximately represent return.

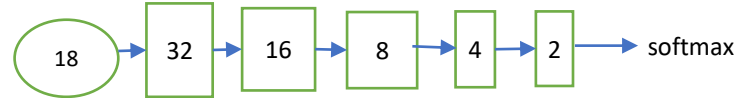
C_O means the close log price minus the open log price. I append H_O, C_O, C_L and H_L respectively. For volume, I used the quantity relative ratio in the past five bars instead. And C_O in the past five bars also should be included.

| | | | | | | | | |
|------|------|------|-------|------|------|------|------|------|
| Open | High | Low | Close | H_O | C_L | H_L | C_O | C_O1 |
| C_O2 | C_O3 | C_O4 | C_O5 | vol1 | vol2 | vol3 | vol4 | vol5 |

Features

3. Deep Neural Network

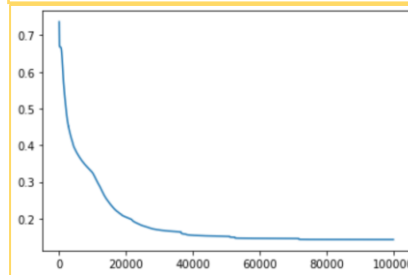
The first four hidden layers use sigmoid function to active and the last one should use softmax function because I want to get the probability. Softmax has a good characteristic which the value of it belongs to (0, 1).



4. Cross Entropy Loss Function

This loss function is very suitable in this classification problem. The more accurate the predicted value, the lower the loss.

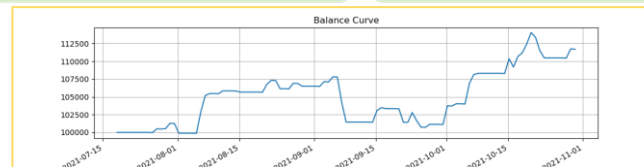
$$L = \frac{1}{N} \sum_i L_i = \frac{1}{N} \sum_i -[y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)]$$



LOSS

Total Return: 0.1170220101552355
Average Daily Return: 0.0010924103374430505
Sharpe Ratio: 2.384461094780931
Maximum Drawdown: -0.06589723444678097

Performance on daily BTC



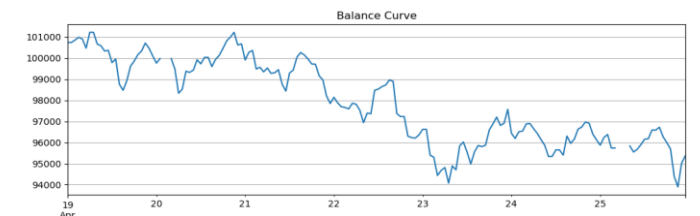
Strategy C&L

5. Trading Strategy

Here I want to establish a long only strategy based on the predicted probability. And the threshold I set is 0.95, which means if the DNN thinks the return of the next day will be positive at 95%, I will long the asset.

6. What about minute-level back-testing

It seems that there is a disaster, the model didn't work on minute-level data.



Performance on minute-level data

7. Conclusion

In practice, I found that the training process is very hard. The model is either over fitting or under fitting. And DNN doesn't have memory, which means something important happened in the past would be lost in this model, LSTM may have a better performance.

The minute-level trading will produce a huge amount of transaction fee and there are many noise in the data, your return may not be as much as the transaction fee. Many complex mathematical formulas may be required here.

8. Contribution

DATA CLEANING, CODING, REPORT

Sixian HUO, 20810798