Final Assignment - trading cryptocurrencies with LSTM and refection on the LSTM journal

PAN Jiayi 20572023

Abstract

Machine Learning is a technique that trains machine to learn patterns, rules, and anomalies by feeding large amount of data to an pre-specified architecture. Applying machine learning technique, particularly the LSTM, in trading cryptocurrencies can generally gain a positive Sharpe ratio, but not very impressive. Why use LSTM? The reflection on the ground-breaking article of LSTM in 1997 provides answer. Though LSTM can solve many different AI problems and performs much better than its predecessors, it has certain limitations, which is also attested by the cryptocurrency trading project – the Sharpe ratio is not very high, even lower than the ordinary MACD strategy.

1 Cryptocurrencies Trading Project (Option A)

1.1 Introduction

I chose the project option A for this course MAFS6010U, and in particular, I chose the project cryptocurrency trading with AI. In this project, I need to use machine learning technique to build trading strategy to trade four different types of cryptocurrencies, namely, BitCoin, EOS, Ethereum and Litecoin. These cryptocurrency are digital assets created to serve as decentralized media of exchange as opposed to centralized digital currency and central banking systems. The decentralization of the cryptocurrency is supported by the distributed ledge technology, or sometimes called blockchains, which possess records of previous transactions so that all the transaction history would be kept by others. Since cryptocurrencies are created in 2009 (bitcoin), these asset have gain lots of attentions and become one of the popular trading assets that many investors and institution invested in. Cryptocurrencies have their own exchanges that are separated from ordinary exchanges for equity and futures, and they can be trade 24-7. The non-stopping trading hours implies that it becomes more necessary to use an automated trading method to trade cryptocurrencies compared to ordinary financial assets. That is one motivation for this project. The trading strategy in this project is essentially a model that is built to predict the price/return for next period, and acts accordingly. The characteristic of this trading strategy is that it uses machine learning technique, in particular, Long Short Term Memory (LSTM), to build the model, and thus it is different from most other strategies used in the industry, which do not use machine learning. One reason that in the industry machine learning has not yet fully adopted is that machine learning is, to some extent, a black box. Financial practitioners do not want to risk investing a large amount of money based on something that is not fully understood.

1.2 LSTM model

The strategy built in this project is a LSTM model. LSTM is a sub-category technique of Recurrent Neural Network (RNN), but it adds four gates in each cell of RNN to control the flow of previous information. These four gates are input gate, output gate, forget gate and gate gate. The cell remembers values over some time intervals and these four gates regulate the flow of information into and out of the cell. The outgoing information will be sent to the next cell and combined with new

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

Table 1: Comparison of Baisc LSTM and LSTM with Technical Indicators

	Annualized Sharpe Ratio	
Cryptocurrency	Basic LSTM	LSTM with Technical Indicators
Bitcoin	2.40	0.04
EOS	0.52	0.44
Ethereum	1.54	0.21
Litecoin	-0.84	1.50

information for processing, and thus it is called "recurrent". LSTM networks are particularly suitable for classifying, processing and making predictions based on time series data, because the past events might possess relevant information for predicting the next outcome. That is why, this project uses LSTM to build model for trading cryptocurrency based on the time-series trading data.

1.3 Considerations of other models

Before deciding to adopt LSTM, this project also considers other neural networks. Firstly, the ordinary neural network is a network in which the users need to specify what variables to input. For trading data, since it is hard to say what variables might be relevant and how long the lag is, it is not appropriate to use the plain neural network. Secondly, Convolutional Neural Network (CNN) is particularly suitable for processing information that have spatial structure, (e.g. images), as it uses filter to extract information and reduce dimension. However, trading data do not have spatial characteristic, using CNN might generate features that are non-sense. Thirdly, Reinforcement Learning (RL) is another consideration. Reinforcement Learning is suitable for cases where the reward can only be known after many steps, and the action of the player would also impact the environment. However, for trading, the ultimate objective is maximizing profit/Sharpe Ratio, and it is proven that maximizing the profit is equivalent to maximizing the profit in the next moment for all the moments. In addition, since this project does not do real trading, the impact of transactions to the market is not known. Then RL is also not suitable. Finally, the Generative Adversarial Networks (GAN) is a network involving one producer generating fake output to fool the distinguisher. There is no way to distinguish a return (output), as long as the return is in ordinary range, positive and negative both being normal. If GAN is used for feature extraction, the features are again hard to interpreted.

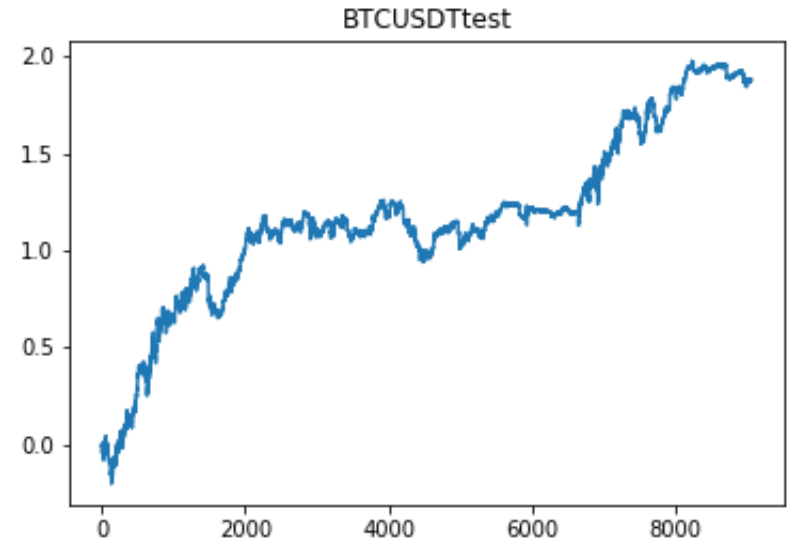
1.4 The strategy of this project

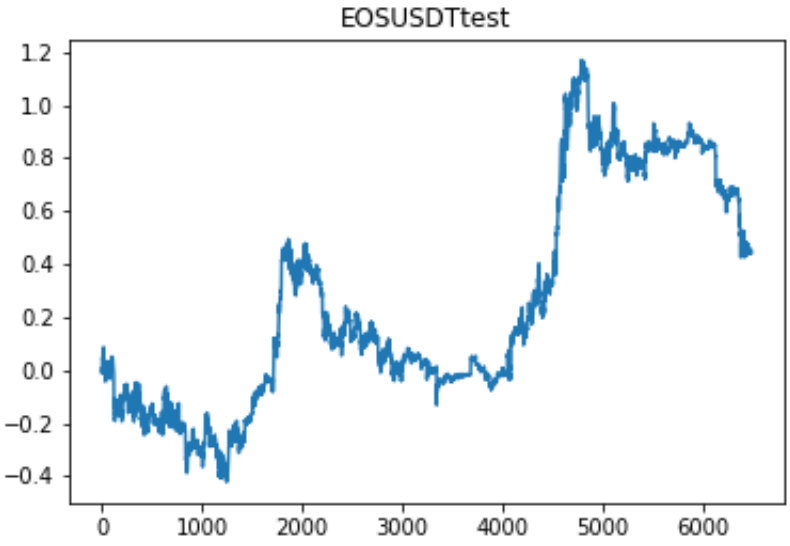
The charts for the back testing of the basic LSTM (without technical indicator) on each cryptocurrencies are presented below. And the Sharpe ratios of basic LSTM and the LSTM with Sharpe ratio are summarized in the following table.

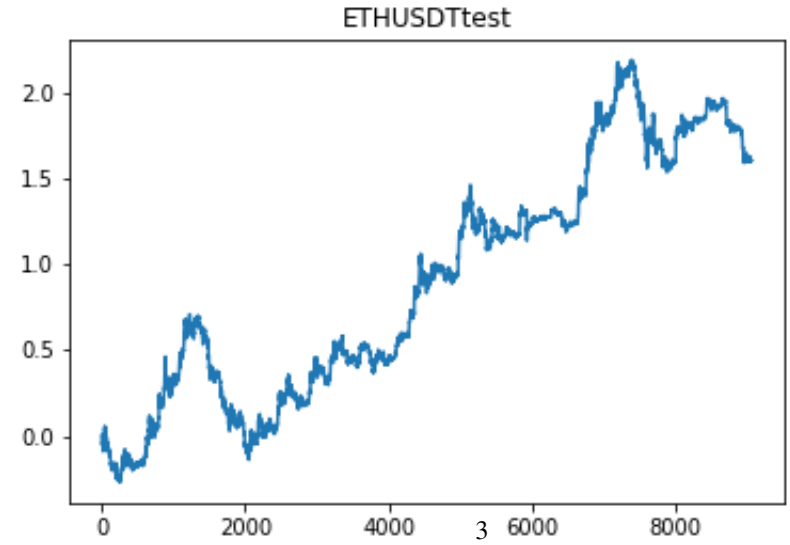
1.5 Result and comparison

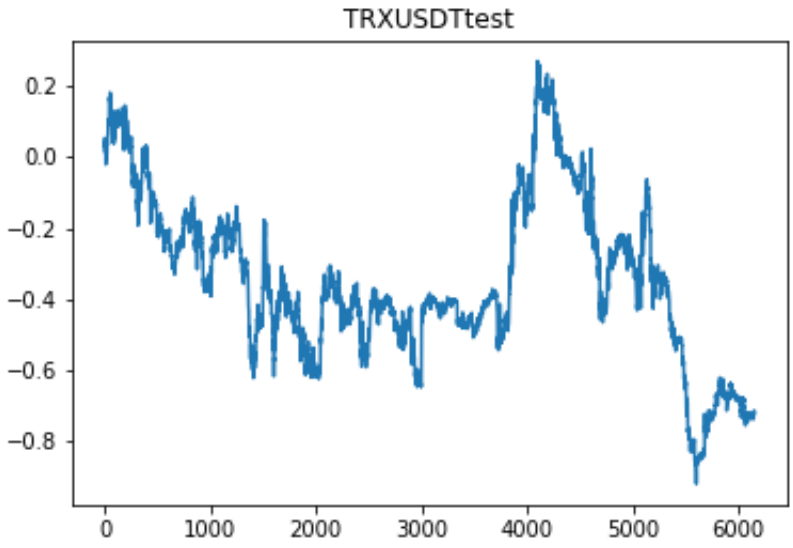
The charts for the back testing of the basic LSTM (without technical indicator) on each cryptocurrencies are presented below. And the Sharpe ratios of basic LSTM and the LSTM with Sharpe ratio are summarized in Table 1.

It can be seen that the performance of the basic LSTM model is better than the LSTM with technical indicators in three cryptocurrencies, and only in the Litecoin it underperforms. This underperformance could possibly due to different market characteristics of Litecoin. Why a simpler LSTM with fewer input performs generally better? One explanation could be that the technical indicators are not very useful in predicting the future return. However, the MACD strategy (but not presented in this report) shows an average Sharpe ratio of 7, implying that MACD provides a good signal for buying and selling. So this prompts me propose another explanation that the LSTM cannot learn from technical indicators, which most investors and trader find useful. This is a more reasonable explanation as the relation between indicators and the price returns are not that simple. For example, the indicators are usually combined with other signals for predicting, and the prediction of price movement might be long-term return instead of next period's return, etc. These complex relation between technical indicators and the return do not help the model learn, even worse, import redundant information that might cause overfitting.


BTCUSDTtest
2.0
1.5
1.0
0.5
0.0
0
2000
4000
6000
8000


EOSUSDTtest
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
1000
2000
3000
4000
5000
6000


ETHUSDTtest
2.0
1.5
1.0
0.5
0.0
0
2000
4000
6000
8000



2 The first journal proposing LSTM

Since the cryptocurrency trading project uses LSTM, I read the first journal proposing LSTM in 1997 [1] and it was written by Hochreiter and Schmidhuber. This part is about the main content of the journal and my own reflection on it.

2.1 Introduction

The model prior to the invention of the LSTM is the vanilla recurrent neural network (RNN). RNN is characterized by that it can learn to store information over extended time interval through processing the data recurrently and backpropagating. This characteristic of RNN makes it an appealing application in many fields, including language processing and time-series trading strategy building, because the past information, or context, is important for these tasks.

2.2 Problems of other RNN models

However, despite its potential in many fields and simplicity, RNN have some problems. The first problem is gradient exploding and vanishing. As the error backpropagates through longer time, the gradient either goes to infinity or to zero, which makes learning the early time features impractical. The second is that recurrent backpropagation takes a very long time to compute, and the learning is very inefficient. The problems of vanilla RRN has been known by scholars for long, and articles acknowledge that some scholar have proposed either remedy or improvement methods. These methods include Time Delays (update unit activations based on a weighted sum of old activations), Time Constant (uses time constant influencing change of unit activation), Ring's Approach (add higher-order unit influencing appropriate connections), Bengio et al.'s Approach, Kalman Filters, Second Order Nets (uses multiplicative units), etc. None of the above mentioned methods can solve the problems of RNN very well, or some of them incur other problems. These are the motivations prompt authors to construct the new architecture LSTM on the ground of RNN.

2.3 The architecture of LSTM

Firstly, a multiplicative input gate unit is introduced to protect the memory content, and then a multiplicative output gate unit is incorporated to protect other units from perturbation. The resulted cell is a memory cell. Each memory cell is built around a central linear unit with a fixed self-connection (the CEC). Inside the memory cell, there are gates to control the flow of information, deciding whether to keep or override information in the previous cells. Each signals trapped within

the memory cell's CEC cannot change, but different error signals flowing into the cell via its output gate may get superimposed. The LSTM network contains one input layer, one hidden layer and one output layer. The hidden layer contains memory cells and gates units. The LSTM is trained by a variant of real-time recurrent learning (RTRL) that takes into account the altered, multiplicative dynamics caused by input and output gates, and the computational complexity is O(W), where W is the number of weights. It is worth noting that, in the beginning of learning, since there is few inputs, the error reduction may be possible without storing information over time. Consequently, the network tends to abuse memory cells. But there are solutions to this problems. One of them is using sequential network construction where a memory cell and the corresponding gates are created later whenever the error stop decreasing in training. And the other is using output gate bias where each gate is initialized with a negative bias, and push initial memory activation to zero.

2.4 Experiments for measuring LSTM's performance

In the second part of the journal, the authors conducts six experiments to apply LSTM to different tasks, measure performance in these six experiments and compare them with other neural networks architectures. The first task is learning the embedded Reber grammar. The authors use three different pairs of training and testing sets, and run 10 trials with different initial weights. The result for the first task is that LSTM learns faster and more accurately. The second task is learning both noise-free and noisy sequence with different length of lag interval. Three subtasks with different length of lag interval are conducted and the LSTM performs much better than other models. The third task is processing Noise and signal on the same channel. Similarly, three different tasks are conducted and their performances are measured, the LSTM still solve the problem faster and more accurately. The forth experiment is called Adding Problem, which is a task that has never been solved by other recurrent net algorithms before. LSTM can solve one scenario, which is the long-time-lag problems involving distributed, continuous-valued representation. The fifth task is Multiplication Problem, and in this experiment, LSTM can solve tasks involving both continuous-valued representations and nonintegrative information processing. The final task is called Temporal Order. Two sub-experiments are done in this experiment, and LSTM can solve other difficult tasks that have never been solved by previous recurrent neural network architectures.

2.5 Conclusion and discussion

The experiments conducted in this journal demonstrates that the LSTM has superior performance than all prior recurrent neural network models, and it can even solve some problems that have never been solved before. Besides, the LSTM is more computationally efficient in that it trains faster and produces more accurate result. However, the LSTM also has its limitations and the authors in this journal acknowledge them. Firstly, the efficient truncated backpropagation version of the LSTM cannot easily solve XOR-like problems. Secondly, LSTM generally run into problems where feedforward nets see the entire string at once, due to LSTM's constant error flow through CECs within memory cells. Compared to its limitations, the LSTM brings lots more advantages. The first advantage is that, the constant error backpropagation allows LSTM to bridge very long time lags. Secondly, the LSTM can handle noise, distributed representations and continuous values. Thirdly, LSTM generalizes problems well, even if the input's lag relation is not so obvious. In addition, LSTM works well in a broad range of parameters such as learning rate, input gate bias, etc. Last but not the least, LSTM algorithm has lower complexity when updating weights and thus consume less computational power.

3 Synthesis and suggestion for further study

This part of the report looks at the cryptocurrency trading project and the LSTM journal together. This part considers the limitation of the project and machine learning, and provides suggestion for future improvement.

- Recurrent Neural Network is group of neural network architectures that are very useful and practical in solving problems involving sequential inputs and context. We have seen its application in language processing, like machine translation, which requires the context of the sentence to translate differently.

- LSTM is firstly proposed by Hochreiter and Schmidhuber in 1997. In their journal that marks the invention of LSTM, it has been shown that LSTM performs much better than other preceding RNN architectures. These outstanding performances attract many practitioners to apply the LSTM architectures in the industry, and it now becomes very popular. Comparing LSTM and other RNN architecture, it can be observed that LSTM is more general and robust, thus it can be adopted more broadly for different problems. This gives us an inspiration that, when we want machine to learn patterns, discover rules or find anomalies, sufficient but not exceeding flexibility should be provided to the algorithms. The vanilla RNN model is too rigid to learn, but adding gates to control the flow of information, LSTM certainly performs better.
- Considering the features of LSTM, it could potentially be the most suitable architecture for trading strategy development. However, the basic LSTM, though generates mostly positive Sharpe Ratio, does not outperform the basic MACD strategy.
- However, in the project, it is noticed that adding more parameters (technical indicators) does not help improving the model. One possible explanation is that the LSTM architecture does not learn too complex relations between the input and the output. Thus, when applying LSTM, it is important to know by analyzing its architecture, what can be learned and what are its limitations.
- Similar to the limitations found in the this cryptocurrencies project, other neural networks may also be limited to learn some relations but are unable to learn other more complex problems. Whenever we use a machine learning technique, the first step is learning its architecture, knowing its limitations and keep them in mind.
- This cryptocurrencies trading project is, after all, a simple simulation. It does not consider the price impacts of real transaction, and some constraints of trading (e.g., cost of shorting, whether there is limited number of transactions in a day, etc.). More sophisticated back testing model/code should be developed to incorporated these real-world consideration.

4 Individual contribution

I did this project completely by myself. I myself forms the group. The link of video: `https://youtu.be/ufY7f8fc5hU`

Acknowledgments

The LSTM model is improved by me based on Cryil's orginal example sent in week 7.

References

[1] Hochreiter, S., & Schmidhuber, J. (1997). *Long short-term memory*. *Neural computation*, **9**(8): 1735-1780.