Application of Reinforcement Learning on Asset Allocation

ENSAE 2020 Informatic Project

PRUGNIAUD Melchior TAN Jing

ENSAE Paristech

April 30, 2020

Outline

Reinforcement Learning Concepts

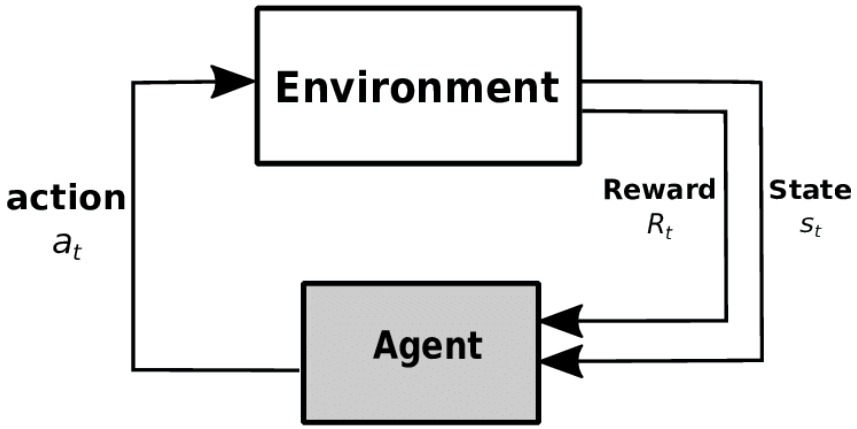

Figure 1: Agent and environment

Actions States Environment Reward Policy Q-value

More explications on reinforcement learning can be found on this website:
Reinforcement Learning Definitions

Deep Q Network

Since the return function $Q(s, a)$ depends on the state s and action a, applications of deep neural network can help approximate this function thus optimize the criteria for a sequence of portfolio. The goal of the agent is to choose actions that maximize the Q-value.
For more explications of deep Q network please check the following web page: *A Beginner's Guide to Deep Reinforcement Learning*

Deep Deterministic Policy Gradient

The deterministic policy gradient adapted Q-learning into continuous control space.
The target policy is not stochastic but a deterministic process, and there is a map function $\mu : \mathcal{S} \leftarrow \mathcal{A}$. By applying the Bellman equation, the Q-value can be written as

$$Q^{\mu}(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}[r(s_t, a_t) + \gamma Q^{\mu} s_{t+1}, \mu_{\theta}(s_{t+1})]$$

and

$$\mu_{\theta}(s) = \underset{a}{\operatorname{argmax}}\, Q(s, a)$$

where θ are trainable parameters of the deep network.

$$J(\mu_{\theta}) = Q^{\mu}(s, a)$$

Actor-critic

Besides, taking derivatives of the objective function is the same as taking derivatives of the policy. DDPG presents an actor-critic, model-free algorithm. The actor is a process to tune deep learning parameters θ for the policy function and return the best policy for a specific state.

$$\pi_\theta(s, a) = \mathbb{P}(a \mid s, \theta)$$

And the critic is used to evaluate the policy according to the temporal error.

$$r_{t+1} + \gamma Q^{\pi_\theta}(s_{t+1}) - Q^{\pi_\theta}(s_t)$$

Data Structure

Our portfolio contains Bitcoin and 6 other cryptocurrencies. We download the prices and volumns from the *Bitfinex* API. The input data is thus a three-dimensional tensor, which represents cryptocurrency, time and financial indicator respectively.

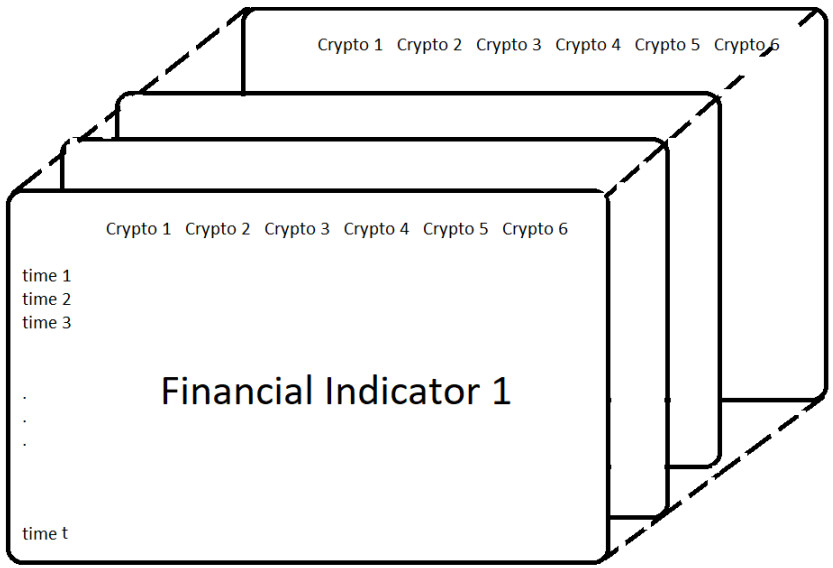

Figure 2: Data structure

Financial indicators

Feature	Explication
Close	Last price during specified interval
Open	First price during the period
High	Highest price level reached at
Low	Low price level reached at
Volume	Volume traded
ROC	Price rate of change
MA3	3 days' moving average
MA7	7 days' moving average
MACD	MA convergence divergence

Table 1: Financial indicators

Problem framing

The following figure proposed a reinforcement learning framework using deterministic policy gradient.

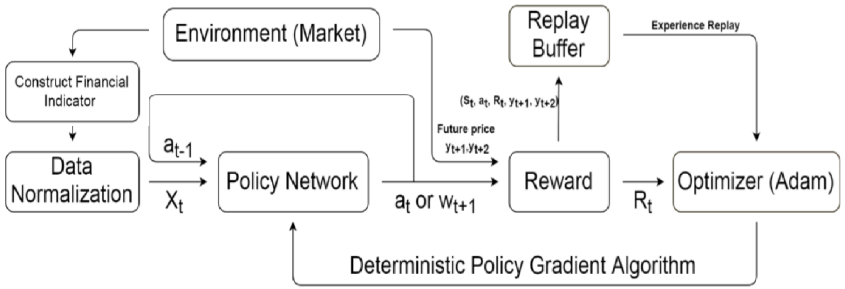

Figure 3: Problem framing

Policy network structure

The policy network can take forms of different structures of deep neural networks. For example, we used a structure of 3 convolutional layers.

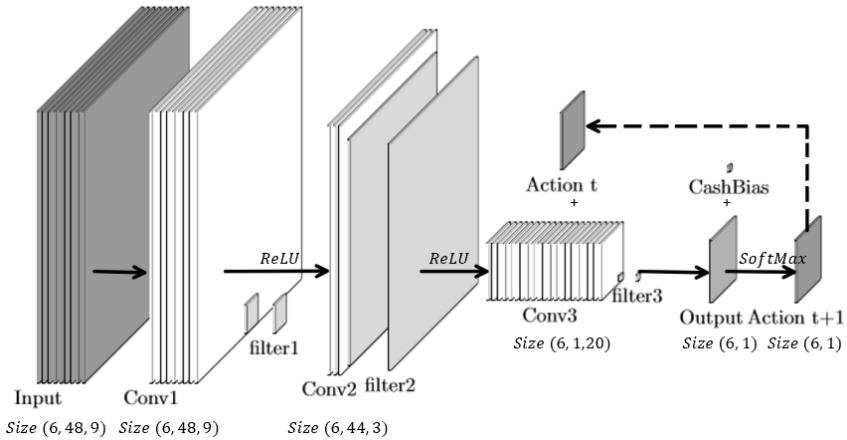

Figure 4: CNN Policy Network

Benchmark portfolio

In order to measure the performance of the deep reinforecement learning portfolio and to compare the results with other naive portfolio strategies, we created two benchmark portfolios.

- **UCRP** Uniform Constant Rebalanced Portfolio. The capital allocation to each asset is constant and we balance the portfolio weight while prices changing to make sure that the condition is respected.
- **UBHP** Uniform Buy and Hold Portfolio. It means that at the inception of the portfolio, capital is equally allocated to each crypto-asset and is never rebalanced.

Hyper parameters

Episode	200
Window size	48
Training step	820
Learning rate	2e-5 - 2e-4
Cash bias[1]	0
Sample bias[2]	1 - 1.1

Table 2: Hyper parameters

[1] Cash bias is a constant number associated with voting score layer. It determines the amount of cash holdings in next trading interval. The activeness of the trading strategy can be controlled by adjusting cash bias.

[2] Sample bias is usually defined in a range of 1 to 1.1, which a value above causes the model to overly bias the latest data and degrade the overall performance eventually.

Effect of learning rate

	Average			
Learning rate	Cumlative Return	Sharpe Ratio	Max Drawdown	Volatility
2.00E-04	0.961681979	-0.12734607	0.119236467	0.003492
2.00E-05	0.961763133	-0.12701175	0.11958631	0.003507
9.00E-05	0.961854508	-0.12767278	0.119149709	0.003488

Figure 5: Effect of learning rate on DDPG

This section demonstrates how the learning rate of the model affects the trading strategy and performance. The result demonstrated that the model generally achieved better performance for smaller learning rate.

Cumulative return

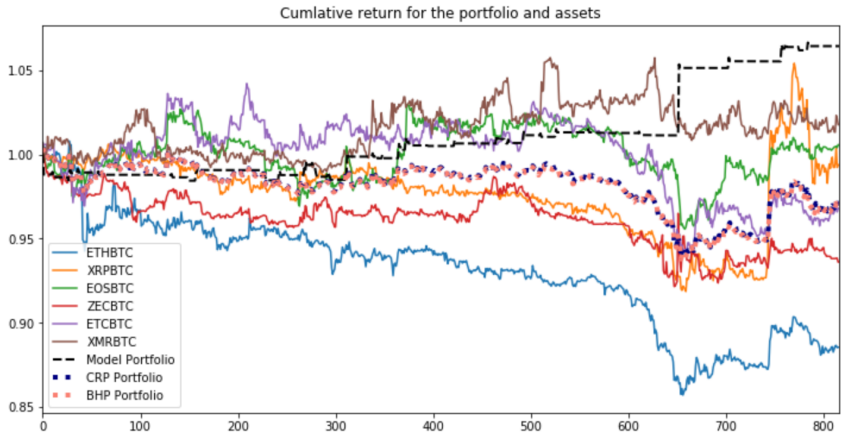

Figure 6: Cumulative return of different strategies

The graphic shows that our deep reinforcement portfolio has a better cumulative return than the benchmark portfolio. It also gained a positive return and outperformed all cryptocurrencies.

Over-fitting

Over-fitting: Our deep reinforcement learning model is empowered with large scale neural networks, carefully designed architectures, novel training algorithms and massively parallel computing devices. However, more training power comes sometimes with a potential risk of more over-fitting.

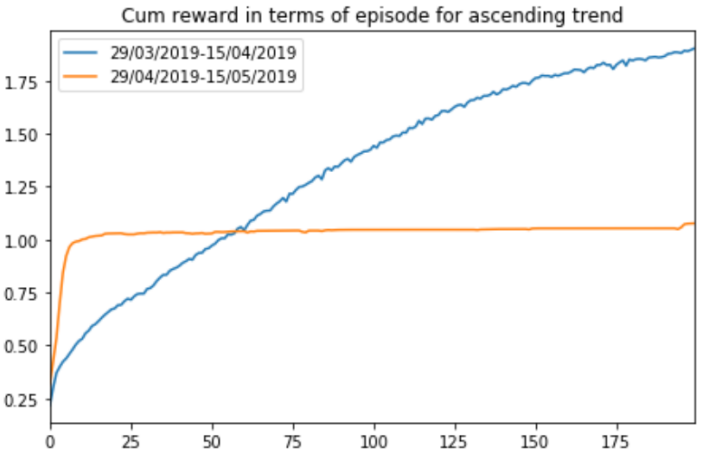

Figure 7: Cumulative rewards over-fitting and boring areas traps

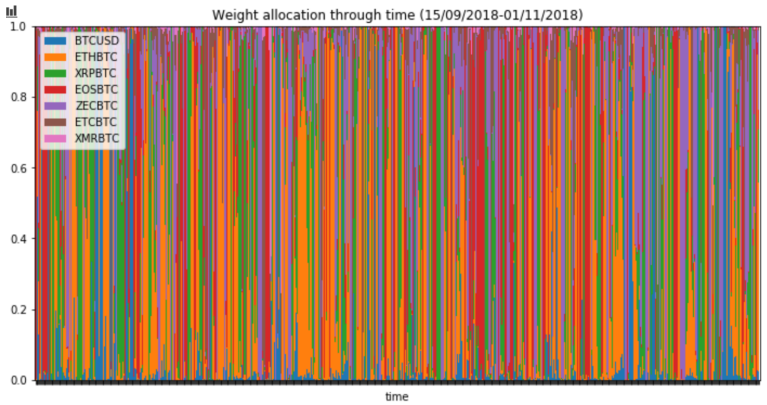

Figure 8: Over-fitted portfolio composition

Boring Areas Trap

Boring Areas Trap: It is possible that the agent falls into a boring areas trap. Due to lower variance differences the reward function hardly change. A lower learning rate will lower the possibility of falling in to this area however training will be very slow.

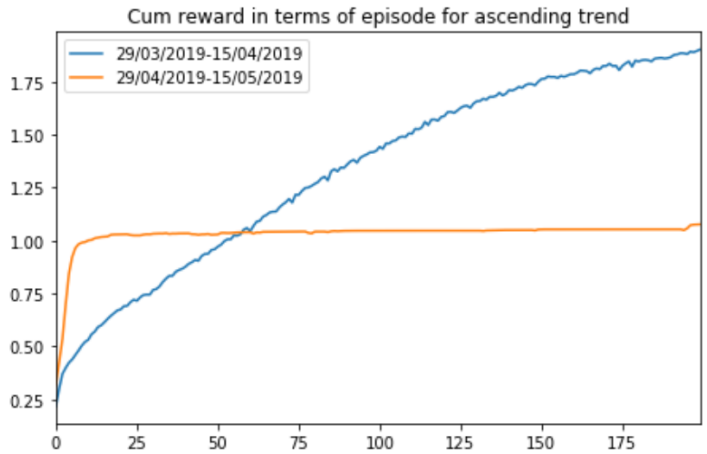

Figure 9: Cumulative rewards over-fitting and boring areas traps

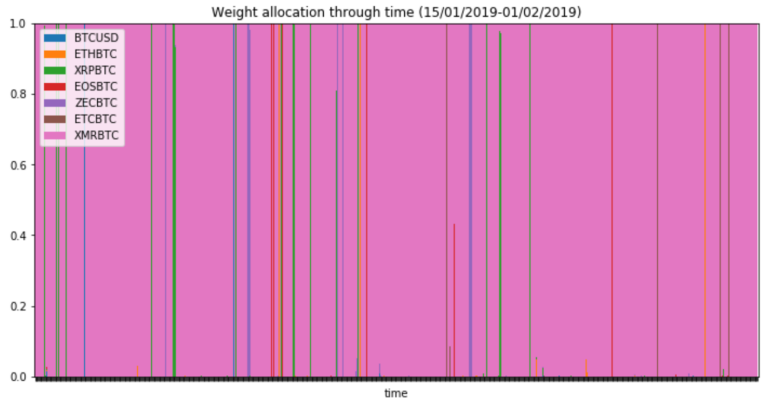

Figure 10: Trapped portfolio composition

Difficulties and future work

Practical problems: model training for with historical data, agent simulators, etc.
Theoretical difficulties: strong non-linearity and stochasticity of financial time-series.

Relevant readings

- Alexander Pritzel Nicolas Heess Tom Erez YuvalTassa David Silver Daan Wierstra Timothy P. Lillicrap,Jonathan J. Hunt. 2016. Continuous control with deepreinforcement learning.Conference paper at ICLR2016, arXiv preprint:1509.02971.
- Zhengyao Jiang, Dixing Xu, and Jinjun Liang. 2017. Adeep reinforcement learning framework for the finan-cial portfolio management problem.
- Roohollah Amiri, Hani Mehrpouyan, Lex Fridman,Ranjan Mallik, Arumugam Nallanathan, and DavidMatolak. 2018.A machine learning approach for power allocation in HetNets considering QoS.