A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem

(Jiang et al. 2017)

Implementation and further research by Selim Amrouni, Aymeric Moulin and Philippe Mizrahi

Paper Presentation

Objective

The problem is the one of automated portfolio management: given a set of stocks, how to best allocate money through time to maximize returns at the end of a certain number of timesteps

Data

Jiang et al. use 13 crypto-currencies from the Poloniex exchange. They take into account the open, high, low, close (OHLC) prices, minute per minute. They allow a portfolio rebalance every 30 minutes. They reprocess the data and create a tensor based on the last 50 time-steps (Eq. 1).

We extend the experiment to the stock market, using the framework on daily data and intraday data.

Problem setup and RL environment

Timeline:

$$Adjusted\ Reward = Reward - Baseline\ Reward\ - \alpha * Maximum\ Portfolio\ Weight$$

Processed data:

$$X_t = [\ (\frac{Close(t-n-1)}{Open(t-n-1)}|...|\frac{Close(t-1)}{Open(t-1)}),\ (\frac{HiPrice(t-n-1)}{Open(t-n-1)}|...|\frac{HiPrice(t-1)}{Open(t-1)}),\ (\frac{LoPrice(t-n-1)}{Open(t-n-1)}|...|\frac{LoPrice(t-1)}{Open(t-1)}),\ (\frac{Open(t-n)}{Open(t-n-1)}|...|\frac{Open(t)}{Open(t-1)})\ ]$$

Reinforcement Learning Framework:

Setup:

$$action = w_t = (w_1, \ldots, w_n)$$

$$state = S_t = (X_t, w_{t-1})$$

$$r_t = \Sigma w'_i y_i - \frac{1}{m}\Sigma y_i - \alpha max(w_1, \ldots, w_m)$$

Reinforcement learning method:

Policy gradient

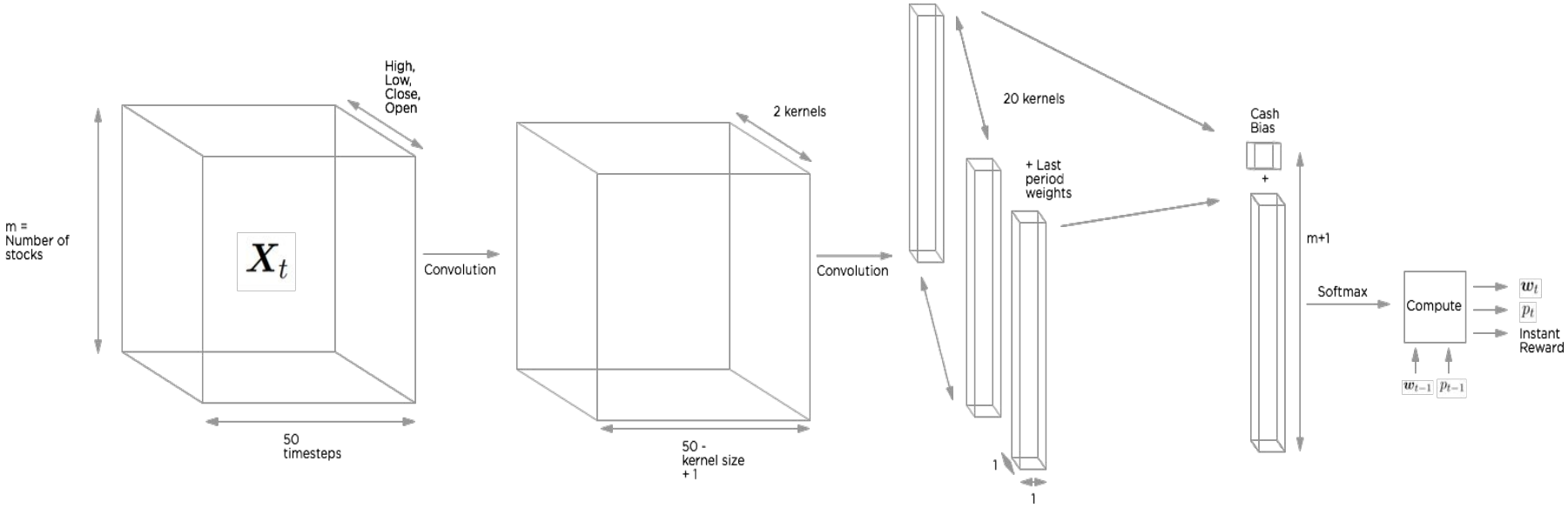
Policy function:

Neural network

Epsilon Greedy:

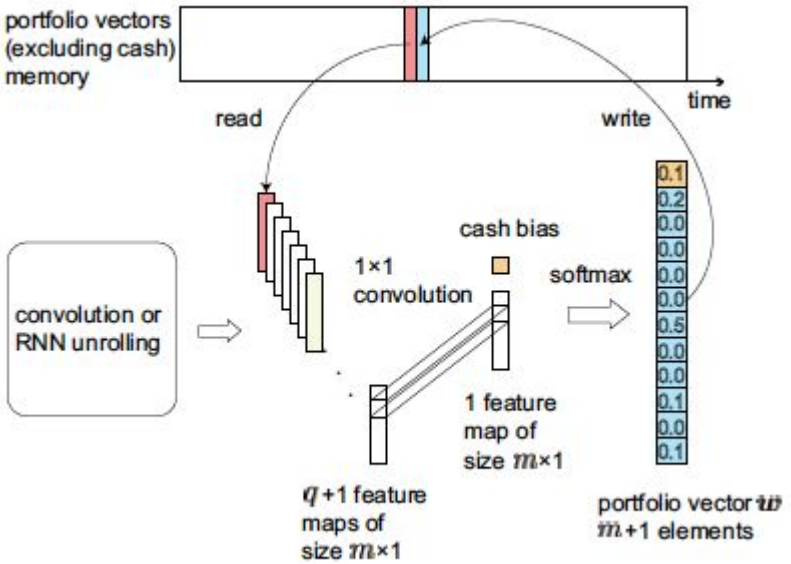
to improve exploration, take random action with probability epsilon during training

Architecture of Policy Network



Train,validation, test process

PVM and Batch Online Learning for training



Online actions without learning on validation and testing

-Policy tested/validated on unexplored dataset

-no PVM: start with cash only portfolio

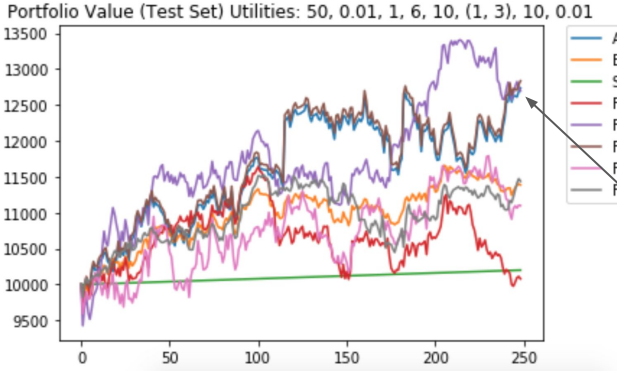
-Look at portfolio realized performances

-Look at policy performance on validation set during training to assess overfitting

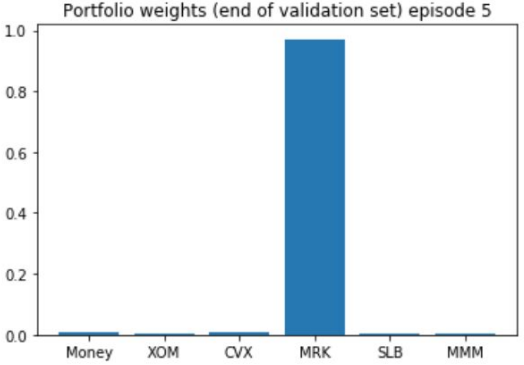
Source: Jiang et al.

Results - Experience 1:Daily data - 1/2

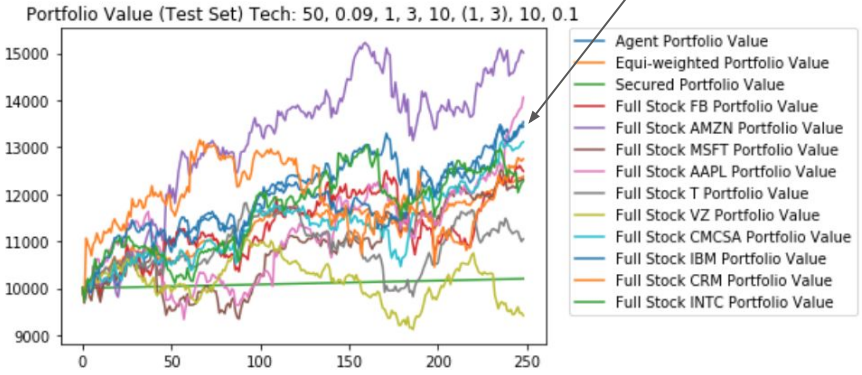
Single stock selection behaviour

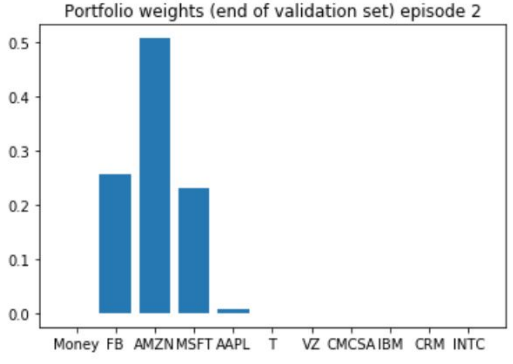


Portfolio Value



Multi stock selection behaviour





Results - Experience 1:Daily data - 2/2

Issues:

-Stationary policy

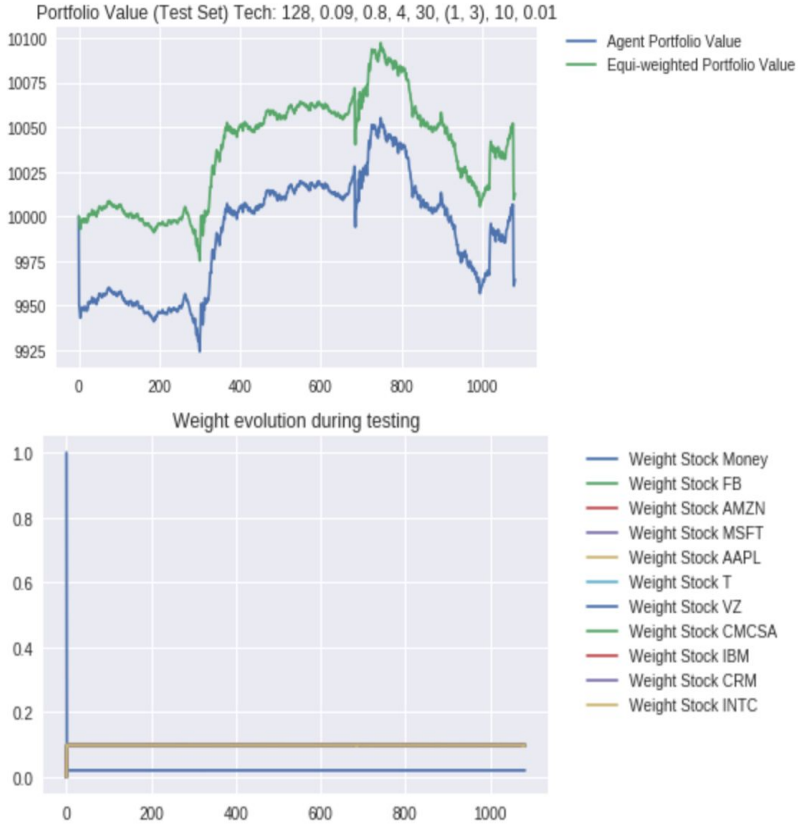
-Little influence of the inputs on the portfolio allocation even with low trading costs

Our intuition:

-Daily data must be to stable to encourage the algorithm to change weights

=> We carry out new experiments with intraday data

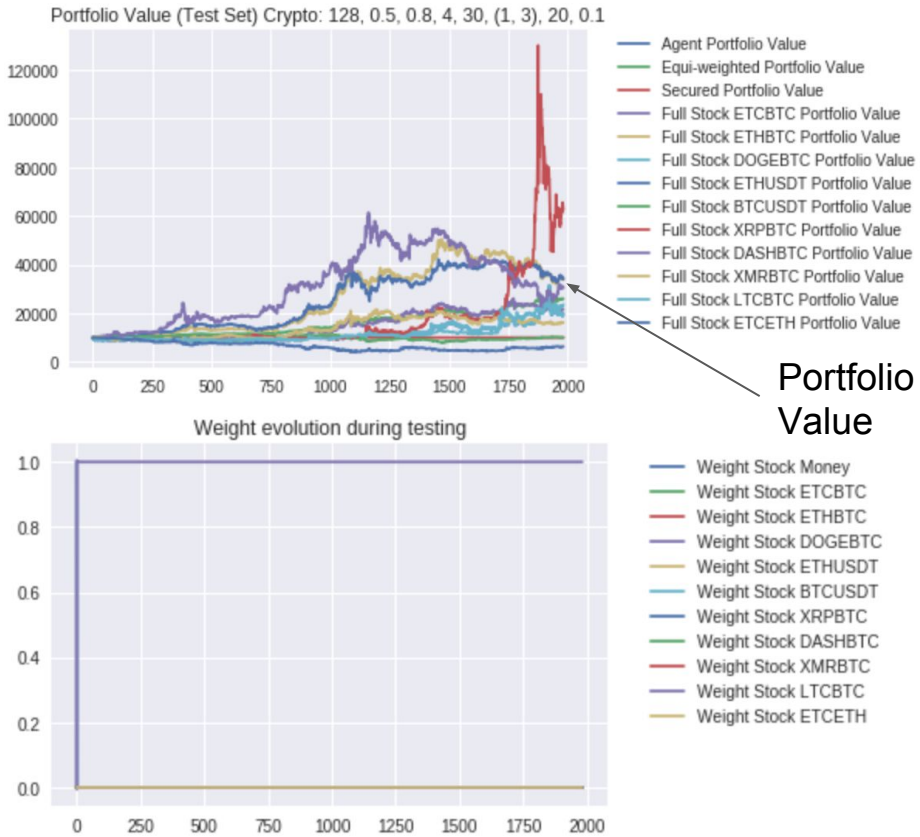
Results - Experience 2:Intraday data - MinuteTech



Issues:

- Agent sells most of the cash right away
- The portfolio is very similar to the equi-weighted and the weights don’t change
- Final return is baseline minus the initial transaction costs

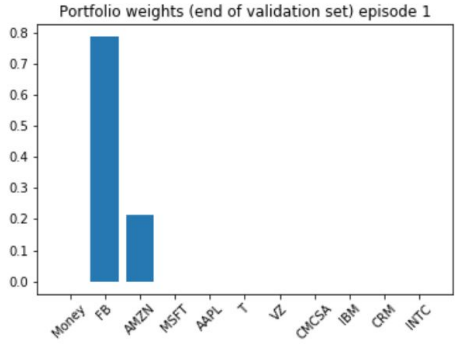
Results - Experience 2:Intraday data - Crypto

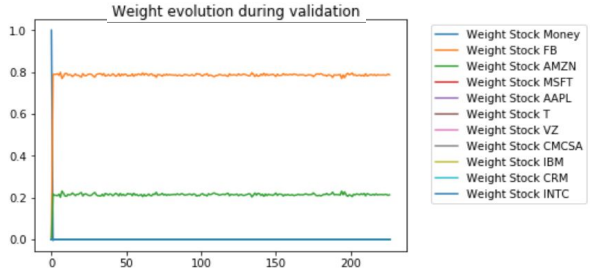


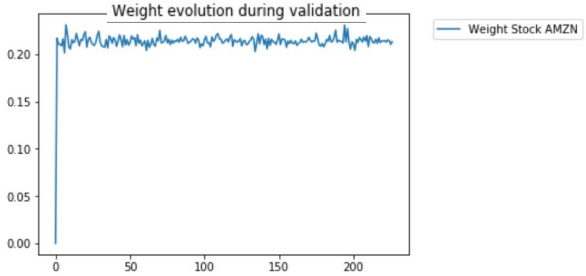
Same Issues:

- Only one crypto allocation
- Still a small (null) influence of the input
- Doesn’t capture the possible gain to invest in XRP/BTC

Understand Problem & Possible improvements







- The change of weights is very poor. It seems the policy is ‘Training Sensitive’ but it is not ‘State Sensitive’
- To make the policy more dynamic: use a discrete action space
- The actions are for each stock:
  - +1: buy n% of a stock
  - 0: hold the position
  - -1: sell n% of a stock
  - n is an hyper-parameter of the problem
- To do so we can replace softmax layer by tanh activations and then select closest or use classification approach

Bibliography

Zhengyao Jiang, Dixing Xu, Jinjun Liang, *A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem*, 2017

Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. *Deep direct reinforcement learning for financial signal representation and trading. IEEE transactions on neural networks and learning systems*, 2017

Bin Li and Steven CH Hoi. *Online portfolio selection: A survey,* 2013

William F Sharpe. *The sharpe ratio*, 1994

J. Moody, M. Saffell, *Learning to trade via direct reinforcement,* 2001

Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, Qionghai Dai, *Deep Direct Reinforcement Learning for Financial Signal Representation and Trading,* 2016