Digital Currency

Group members

Bohan Li, Zhekai Dong, Hyeseong Choi

Agenda

- Introduction
- Algorithm
- Data
- Model
- Results and Analysis
  - Long-Term Profits
  - Short-Term Profits
- Conclusion
- Future works

Introduction

- Background:
  - Combination of deep learning with reinforcement learning model for the stock market was first proposed in 2016
  - Various models available for the stock market after that
- Challenge
  - The stock market models only rely on the price do not perform well on more unpredictable markets such as the digital currency market (ie. Bitcoin)
- Objective:

  Build DQN model to help trading digital currencies and make profit.
- Motivation
  - Develop deep reinforcement learning model for digital currency trading
  - The model includes more trading indexes

Algorithm

- Q-learning
- Deep neural network(Keras with tensorflow 2.0)

Data

- 7 data files
- 5 completed daily price history of bitcoin, ltc, eth, bch, eos for long term trading
- 2 hours price history for bitcoin and ltc for short term high frequency trading
- Training: 65% data
- Validation: 10%data
- Testing: 25% data

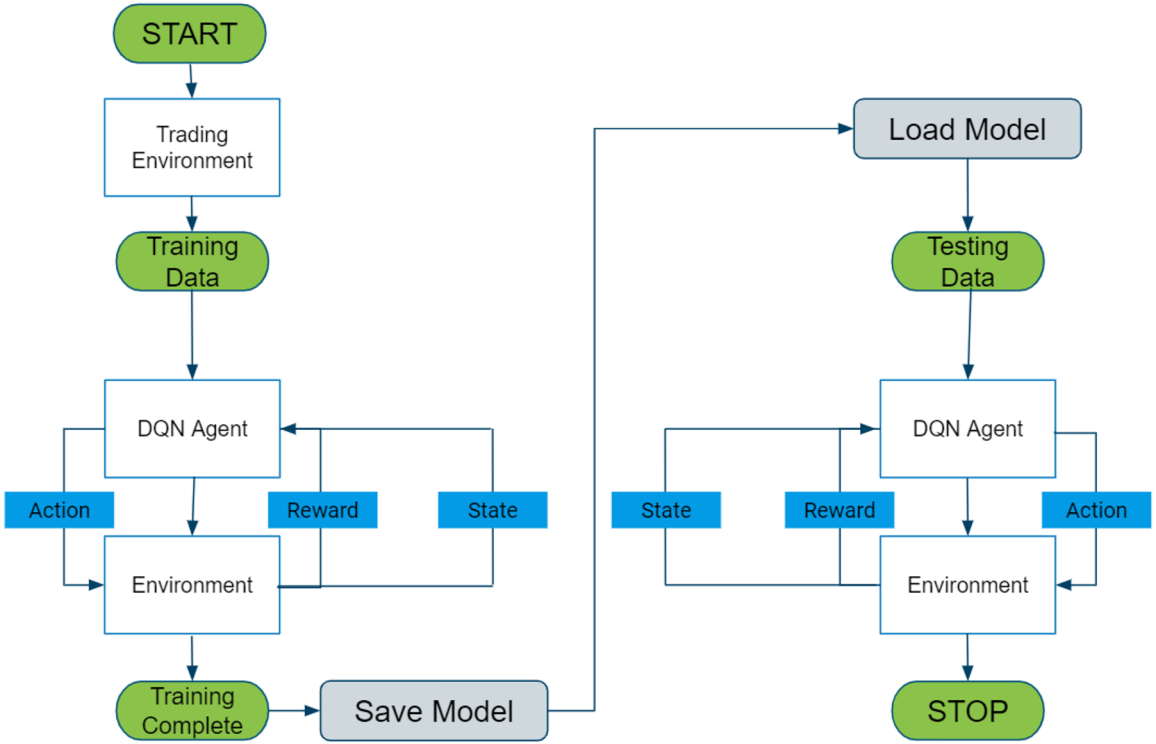
Data

Each sample will be process with indexes as a state in state space:

$$\{(ADX_t\ ,\ RSI_t\ ,\ \ CCI_t\ \ ,\ \ Vol_t\ ,\ Position_t\ ,\ Return_t\ ),(ADX_{t+1}\ ,\ RSI_{t+1}\ ,\ \ CCI_{t+1}\ \ ,\ \ Vol_{t+1}\ ,\ Position_{t+1}\ ,\ Return_{t+1}\ ).... \text{Terminate}\}$$

Action Space: Buy; Hold; Sell

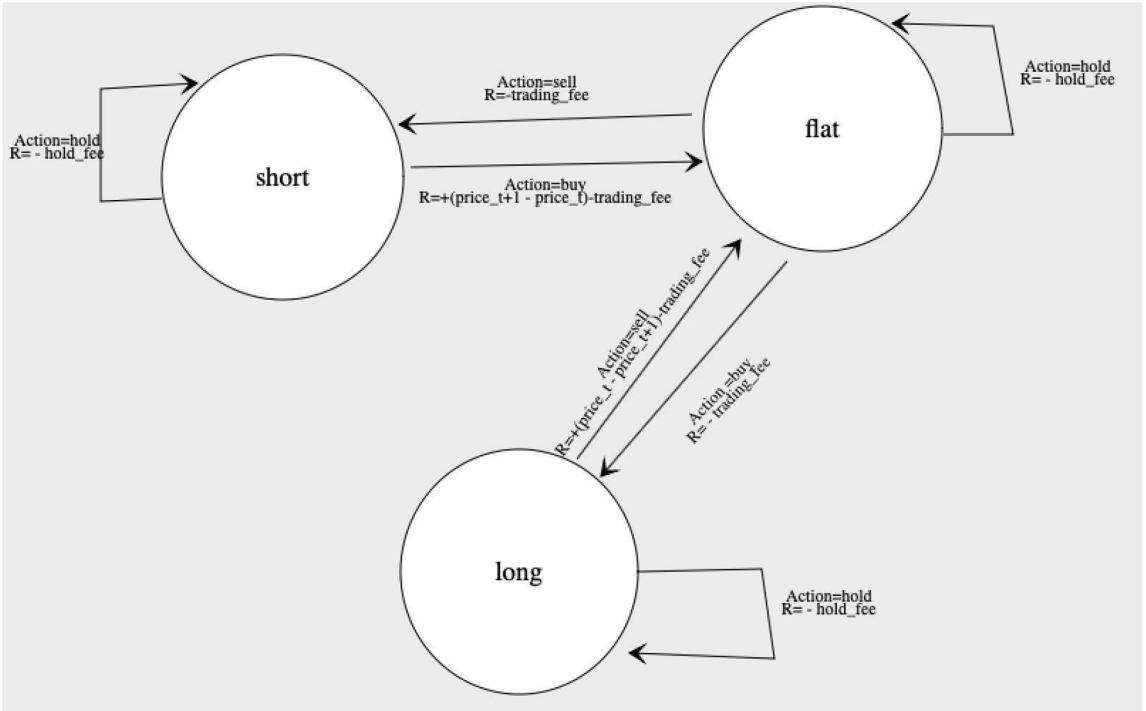
Model


START
Trading Environment
Training Data
DQN Agent
Action
Reward
State
Environment
Training Complete
Save Model
Load Model
Testing Data
DQN Agent
State
Reward
Action
Environment
STOP

Model

Reward Structure:

reward will depend on the pairs of position and action.



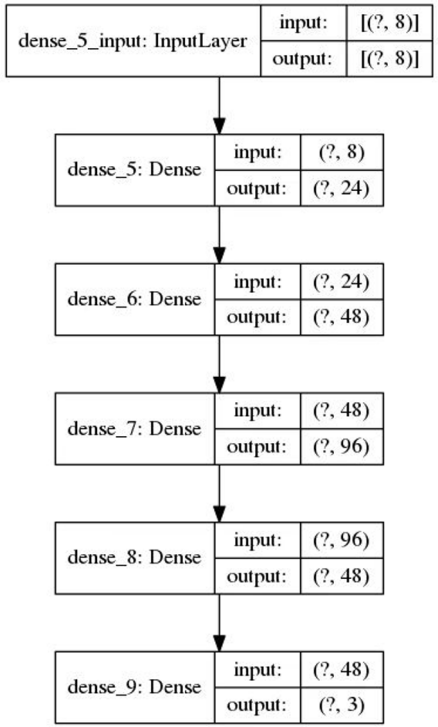
Model

- Q-learning:

$$Q^*(s_t,\ a) \rightarrow r(s_t,\ a_t) + \gamma max_a Q^*(s_{t+1},\ a)$$

- Deep Q learning:
  - Q function approximated by Deep neural network model (DQN)

$$L(w) = E[(r + \gamma max_{a'} Q(s',\ a',\ w) - Q(s,\ a,\ w))^2]$$



Results

- **Experiment setup:**

Data: 65% 10% 25%

iterations: 50

trading fee: 0.001

time fee: 0.001 (holding fee)

gamma: 0.96

batch_size: 64

learning rate: 0.001

activation function: (relu , relu, relu, relu, linear)

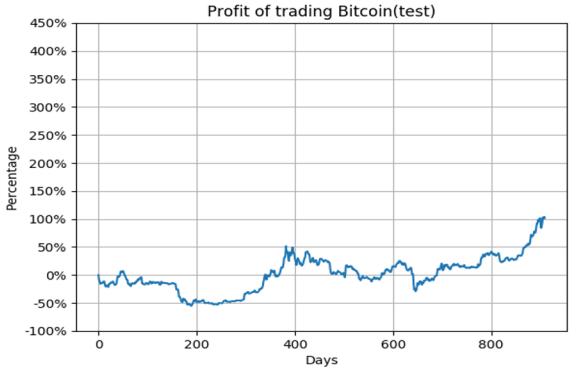
loss function: mse

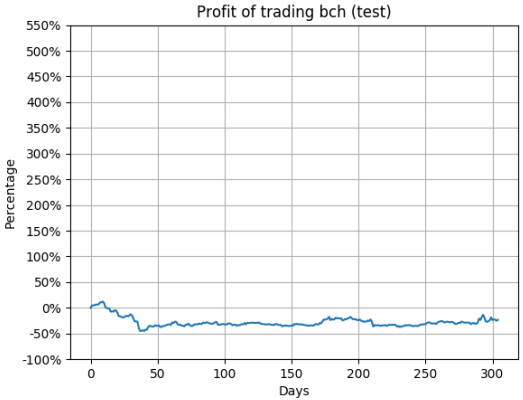
optimizer: Adam

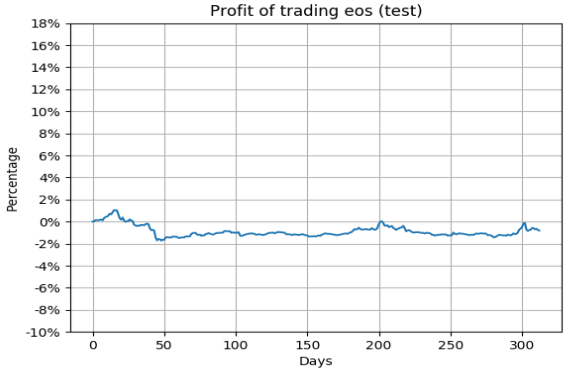
initial trading principal: 800 currencies and 100000$

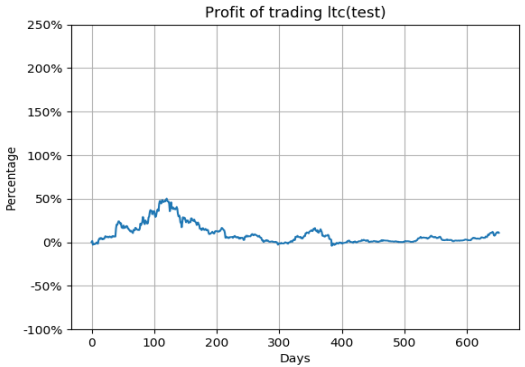
profit = (final trading principal - initial trading principalinitial)/ trading principal*100%

Results and Analysis - Long-Term Profits

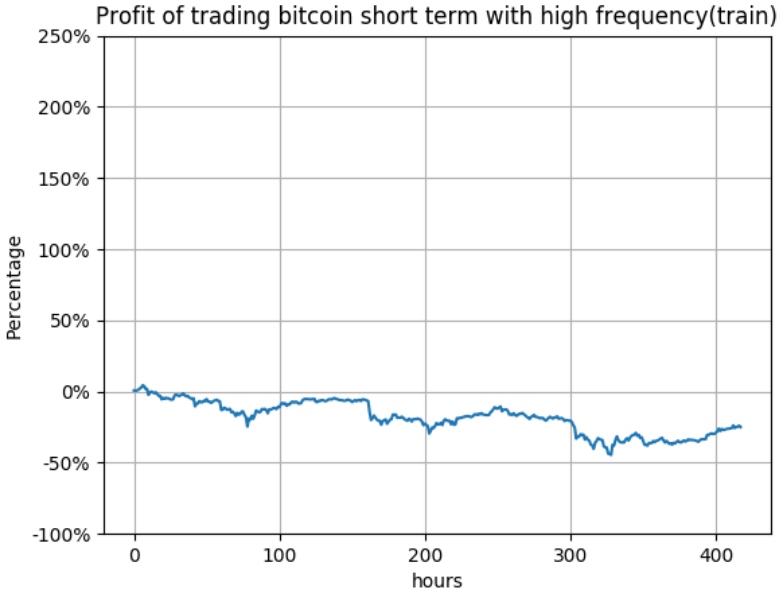


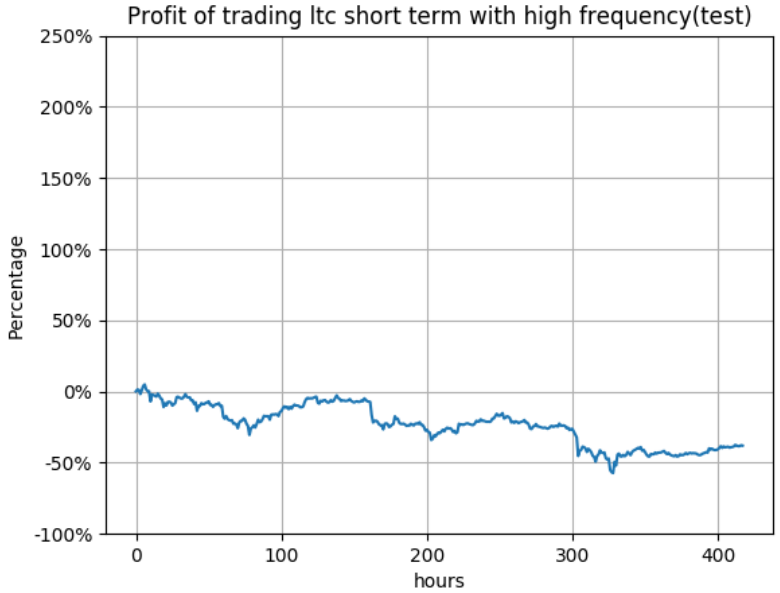






Results and Analysis - Short-Term Profits





Conclusion

The model is good at long term low frequency trading but not good at short term high frequency trading.(Indexes don’t help a lot in short term high frequency trading)

Future works

- Add some more complex layers into deep neural network like CNN and LSTM to strong our model.
- Redefine our state space by using more efficient and independent trading indicators or indexes
- Redefine our reward structure and make it closer to the reality.

Thank You!