Limit Order Execution Optimization with Reinforcement Learning

Mid-term thesis presentation

Marc Juchli

Table of contents

Motivation

Financial institutions buy or sell assets based on various reasons:

- Customer request
- Fundamental analysis
- Technical analysis
- …

→ Invariable outcome is the decision to buy or sell assets.

But how?

Background – Order book

Terminology

Bid: price in a buy order

Ask: price in a sell order

Spread: gap between the best bid and the ask price.

Order

side	:: buy | sell
type	:: limit | market
amount	:: float
price	:: float

Action	COUNT	AMOUNT	TOTAL	PRICE	PRICE	TOTAL	AMOUNT	COUNT	Action
-1	1	4.7	4.7	14,910	14,930	3.9	3.9	3	+1
-2	2	2.2	7.0	14,900	14,940	3.9	0.0	1	+2
-3	2	1.9	9.0	14,880	14,950	7.8	3.8	3	+3
-4	1	0.1	9.1	14,870	14,960	9.0	1.2	1	+4
…	2	0.1	9.2	14,860	14,970	13.2	4.1	5	…
	1	0.2	9.4	14,840	14,980	14.8	1.6	3	
	1	0.0	9.4	14,830	14,990	16.1	1.2	2	
	2	1.5	11.0	14,820	15,000	39.5	23.4	7	
	37	28.2	39.2	14,800	15,010	43.1	3.5	3	
	4	1.9	41.1	14,790	15,040	43.1	0.0	1	
	8	2.9	44.0	14,780	15,060	44.3	1.1	5	
	5	1.0	45.1	14,770	15,070	46.0	1.7	1	
	2	3.1	48.3	14,760	15,080	50.7	4.7	4	
	13	4.5	52.8	14,750	15,090	51.4	0.7	1	
	8	1.6	54.5	14,740	15,100	53.6	2.1	2	
	6	0.0	54.6	14,730	15,110	54.1	0.5	1	
	7	2.5	57.1	14,720	15,120	56.8	2.6	3	
	9	3.2	60.3	14,710	15,130	56.8	0.0	1	
	31	4.8	65.2	14,700	15,150	56.8	0.0	1	
	5	0.1	65.3	14,690	15,160	57.9	1.0	1	
	7	20.2	85.5	14,680	15,180	59.8	1.9	4	
	4	15.0	100.5	14,670	15,190	59.9	0.0	1	
	6	6.7	107.3	14,660	15,200	104.2	44.3	10	
	19	4.5	111.8	14,650	15,220	105.6	1.3	2	

https://www.bitfinex.com/t/BTC:USD

Background – Match Engine

Order Types:

1. Limit(price, amount)
2. Market(amount)

Action:
Order at limit level

Result:
Execution (e.g. Trade)

Action	COUNT	AMOUNT	TOTAL	PRICE	PRICE	TOTAL	AMOUNT	COUNT	Action
-1	1	4.7	4.7	14,910	14,930	3.9	3.9	3	+1
-2	2	2.2	7.0	14,900	14,940	3.9	0.0	1	+2
-3	2	1.9	9.0	14,880	14,950	7.8	3.8	3	+3
-4	1	0.1	9.1	14,870	14,960	9.0	1.2	1	+4
…	2	0.1	9.2	14,860	14,970	13.2	4.1	5	…
	1	0.2	9.4	14,840	14,980	14.8	1.6	3	
	1	0.0	9.4	14,830	14,990	16.1	1.2	2	
	2	1.5	11.0	14,820	15,000	39.5	23.4	7	
	37	28.2	39.2	14,800	15,010	43.1	3.5	3	
	4	1.9	41.1	14,790	15,040	43.1	0.0	1	
	8	2.9	44.0	14,780	15,060	44.3	1.1	5	
	5	1.0	45.1	14,770	15,070	46.0	1.7	1	
	2	3.1	48.3	14,760	15,080	50.7	4.7	4	
	13	4.5	52.8	14,750	15,090	51.4	0.7	1	
	8	1.6	54.5	14,740	15,100	53.6	2.1	2	
	6	0.0	54.6	14,730	15,110	54.1	0.5	1	
	7	2.5	57.1	14,720	15,120	56.8	2.6	3	
	9	3.2	60.3	14,710	15,130	56.8	0.0	1	
	31	4.8	65.2	14,700	15,150	56.8	0.0	1	
	5	0.1	65.3	14,690	15,160	57.9	1.0	1	
	7	20.2	85.5	14,680	15,180	59.8	1.9	4	
	4	15.0	100.5	14,670	15,190	59.9	0.0	1	
	6	6.7	107.3	14,660	15,200	104.2	44.3	10	
	19	4.5	111.8	14,650	15,220	105.6	1.3	2	

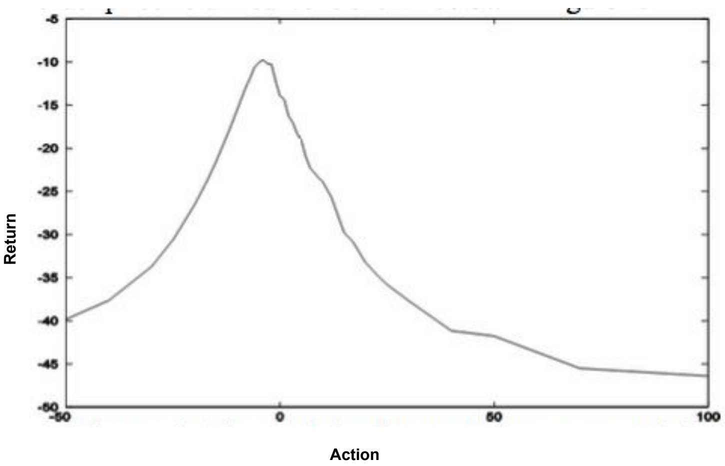
https://www.bitfinex.com/t/BTC:USD

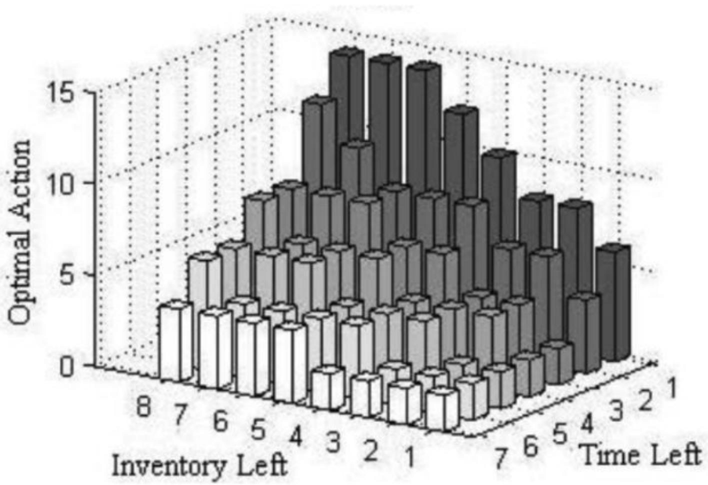
Background – Execution behaviour

"How should one buy (respectively sell) V shares of a given asset over a time horizon H while spending the least (respectively, receiving the most) of the counter asset (e.g. USD)."

$$P_{\mathrm{VWAP}} = \frac{\sum_j P_j \cdot Q_j}{\sum_j Q_j}$$

Return: Price_t0 - VWAP

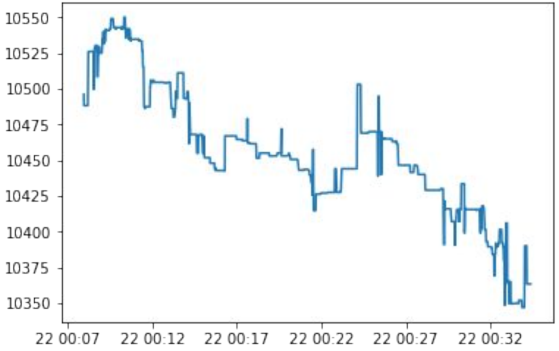




Nevmyvaka, Yuriy, Yi Feng, and Michael Kearns.
"Reinforcement learning for optimized trade execution."
Proceedings of the 23rd international conference on Machine learning. ACM, 2006.

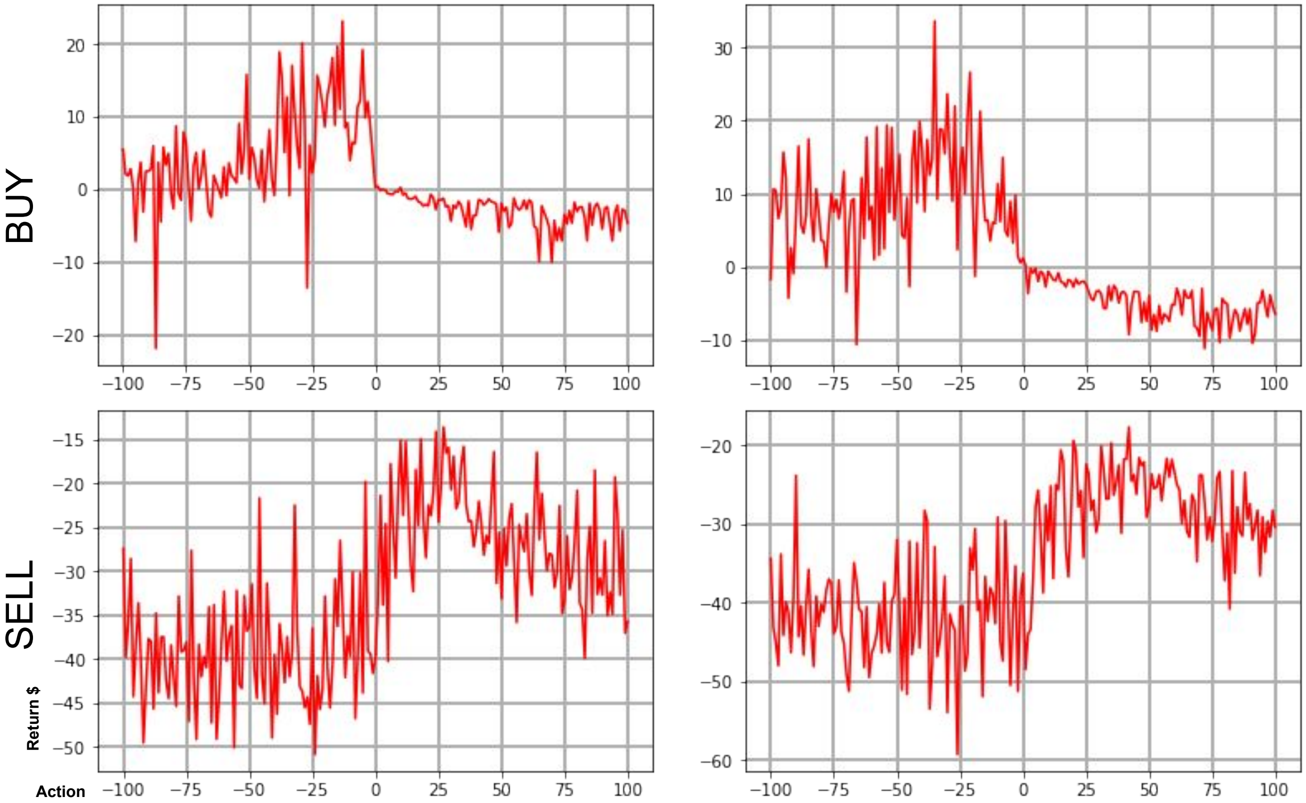
Background – Execution behaviour

USD/BTC (~30 min. hist. data)



T = 60s, I = 1.0 BTC

T = 120s, I = 1.0 BTC



Research objectives

- How should one build a reinforcement learning environment in order to buy (respectively sell) V shares of a given asset over a time horizon H while spending the least (respectively, receiving the most) of the counter asset?
- How can data, which is derived from a limit order book, be used as features in a reinforcement learning environment in order to contribute to the optimization of order execution?

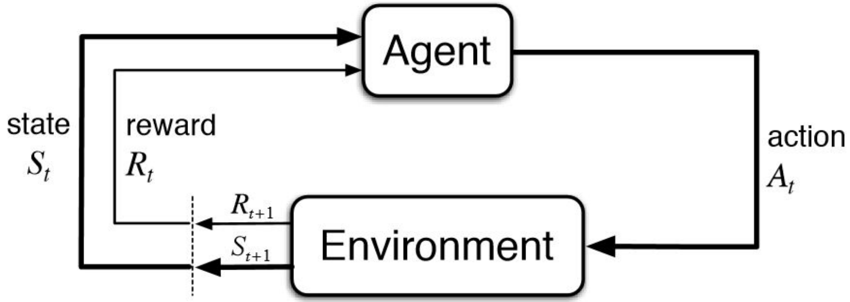
RL – Setup

<u>State:</u>
(runtime, inventory, features*)

<u>Action:</u>
limit level (basis points relative to spread)

<u>Reward:</u>

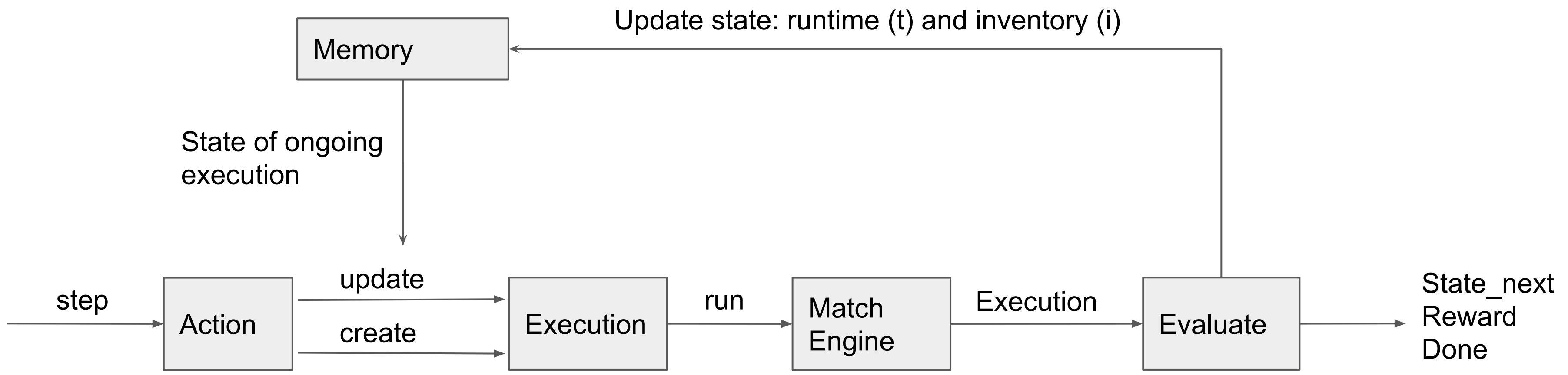
- Buy: Price_t0 - VWAP_executed
- Sell: VWAP_executed - Price_t0



"(Approximately) Markovian nature of trade execution: if our state space is properly defined, the optimal action at any given point in time is (approximately) independent of any previous actions." [Kearns et. al.]

E.g. executions do not affect the market for future executions.

RL – Environment



Q-Learning

Bellman equation: the maximum future reward is the reward the agent received for entering the current state *s* plus the maximum future reward for the next state *s'*.

$$Q(s,a) = r + \gamma \max_{a'} Q(s',a')$$

Q-Learning: we can iteratively approximate Q values using the Bellman equation described above.

$$Q_{t+1}(s_t,a_t) = Q_t(s_t,a_t) + \alpha(r_{t+1} + \gamma \max_{a} Q_t(s_{t+1},a) - Q_t(s_t,a_t))$$

Q-Table:

State: (runtime, inventory, features*)	Action	Value
...	...	...

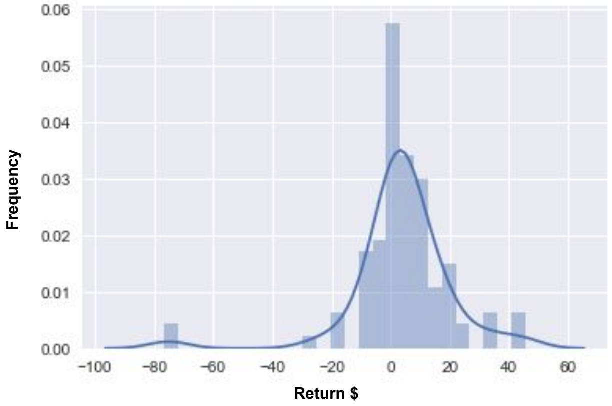
Q-Learning – Results

Buy without market variables.
30 minute training and test set (subsequent)

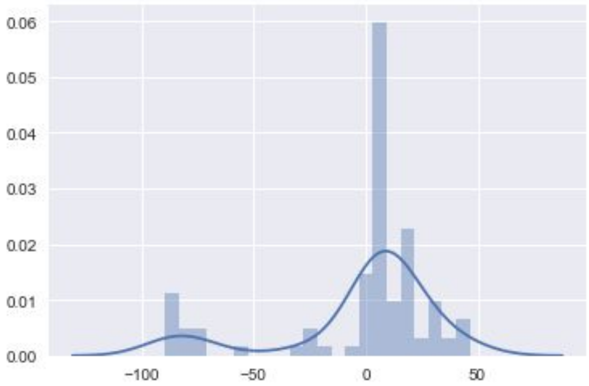
State: (runtime, inventory)
Inventory segmentation: [0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
epsilon=0.4, alpha=0.3, gamma=0.8

Time segmentation:

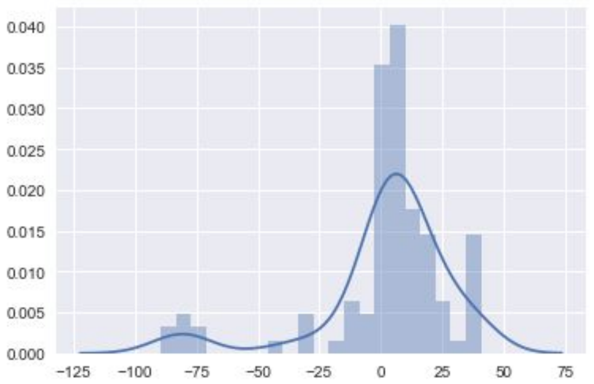
[0, 20, 40, 60]



[0, 20, 40, 60, 80, 100]



[0, 20, 40, 60, 80, 100, 120, 140]



Q-Learning – Conclusion

- + Fast
- + Capable of optimizing on a clear price trend
- - Not able to adapt to unexpected trend changes
- - Feature limitations

Feature Engineering – Difficulties

Obvious features:

- Volume
- Price fluctuation

Derived from the order book time series and will most likely not appear in the same constellation more than once.

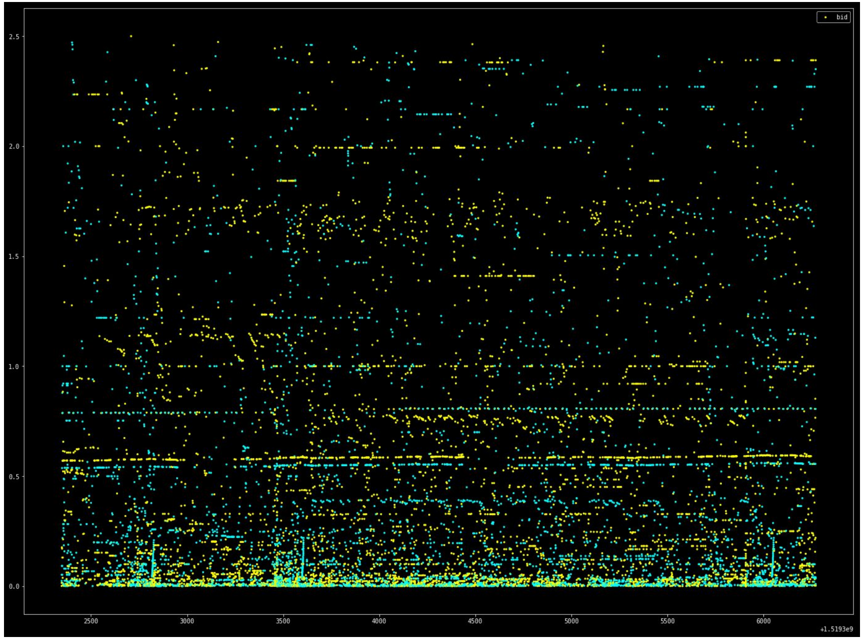
Approximation is required and therefore the capabilities of value iteration (Q-Learning) are quickly exceeded due to state space.

Possible approach: function approximation.

Feature Engineering – Order Events

Volume (create orders) over time

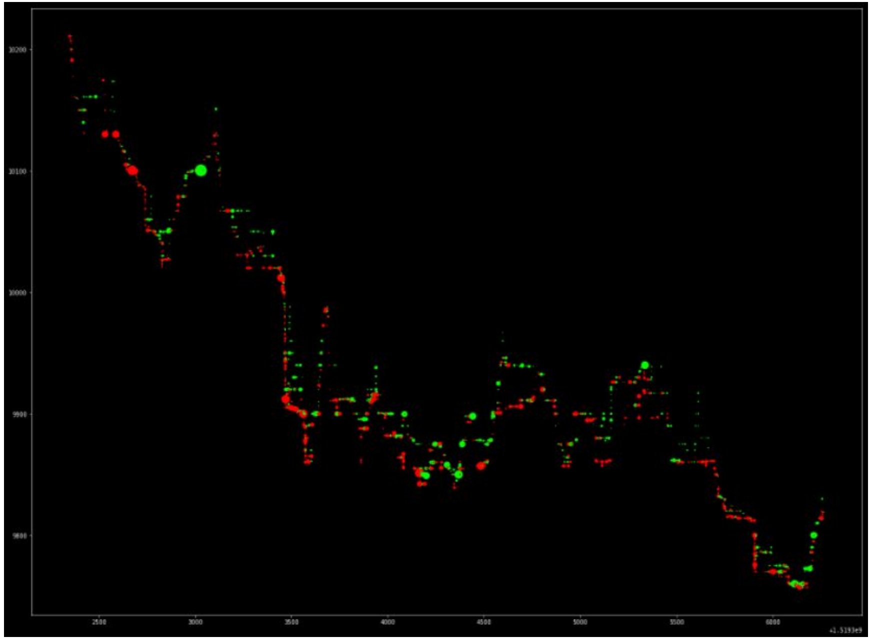
Volume of orders



t-m Time t

(Price, Volume) over time

Volume of orders at price



t-m Time t

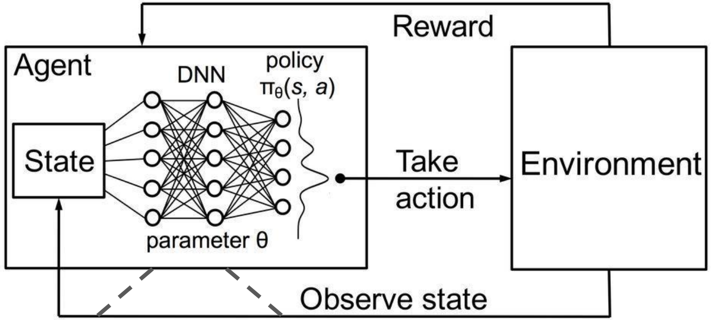
Feature Engineering – Order Events

COUNT	AMOUNT	TOTAL	PRICE	PRICE	TOTAL	AMOUNT	COUNT
1	4.7	4.7	14,910	14,930	3.9	3.9	3
2	2.2	7.0	14,900	14,940	3.9	0.0	1
2	1.9	9.0	14,880	14,950	7.8	3.8	3
1	0.1	9.1	14,870	14,960	9.0	1.2	1
2	0.1	9.2	14,860	14,970	13.2	4.1	5
1	0.2	9.4	14,840	14,980	14.8	1.6	3
1	0.0	9.4	14,830	14,990	16.1	1.2	2
2	1.5	11.0	14,820	15,000	39.5	23.4	7
37	28.2	39.2	14,800	15,010	43.1	3.5	3
4	1.9	41.1	14,790	15,040	43.1	0.0	1
8	2.9	44.0	14,780	15,060	44.3	1.1	5
5	1.0	45.1	14,770	15,070	46.0	1.7	1
2	3.1	48.3	14,760	15,080	50.7	4.7	4
13	4.5	52.8	14,750	15,090	51.4	0.7	1
8	1.6	54.5	14,740	15,100	53.6	2.1	2
6	0.0	54.6	14,730	15,110	54.1	0.5	1
7	2.5	57.1	14,720	15,120	56.8	2.6	3
9	3.2	60.3	14,710	15,130	56.8	0.0	1
31	4.8	65.2	14,700	15,150	56.8	0.0	1
5	0.1	65.3	14,690	15,160	57.9	1.0	1
7	20.2	85.5	14,680	15,180	59.8	1.9	4
4	15.0	100.5	14,670	15,190	59.9	0.0	1
6	6.7	107.3	14,660	15,200	104.2	44.3	10
19	4.5	111.8	14,650	15,220	105.6	1.3	2

$$
\begin{array}{cccc}
\text{Price Size} & \text{Price Size} & & \text{Price Size} \\
\left[\begin{pmatrix} bp_1 & bs_1 \\ bp_2 & bs_2 \\ \vdots & \vdots \\ bp_n & bs_n \\ ap_1 & as_1 \\ ap_2 & as_2 \\ \vdots & \vdots \\ ap_n & as_n \end{pmatrix}\right. &
\begin{pmatrix} bp_1 & bs_1 \\ bp_2 & bs_2 \\ \vdots & \vdots \\ bp_n & bs_n \\ ap_1 & as_1 \\ ap_2 & as_2 \\ \vdots & \vdots \\ ap_n & as_n \end{pmatrix} & \cdots &
\left.\begin{pmatrix} bp_1 & bs_1 \\ bp_2 & bs_2 \\ \vdots & \vdots \\ bp_n & bs_n \\ ap_1 & as_1 \\ ap_2 & as_2 \\ \vdots & \vdots \\ ap_n & as_n \end{pmatrix}\right] \\
\textbf{t} & \textbf{t-1} & & \textbf{t-m}
\end{array}
$$

State: (lookbackWindow m, 2*bookSide n, 2)

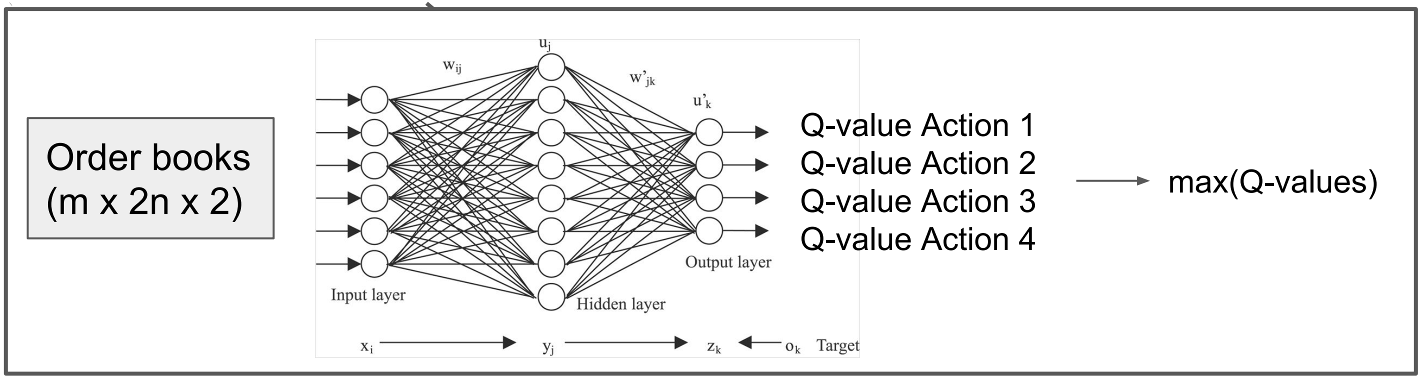
Deep Q-Learning



<u>State</u>: (lookbackWindow m, 2*bookSide n, 2)

<u>Loss function</u>: $\frac{1}{2}\cdot[\underbrace{r + max_{a_{t+1}}(Q(s_{t+a}, a_{t+1}; \theta_{t-1}))}_{\text{target}} - \underbrace{Q(s, a; \theta)}_{\text{prediction}}]^2$

<u>Experience Replay</u>: train on random mini-batches



Conclusion & Outlook

- Building the environment for order execution is hard
- Vast amount of order book data is challenging
- Q-Learning is limited (state space)
- Current results are moderate
- Deep Q-Learning is flexible and promising

❏ Focus on feature engineering
❏ Evaluation of Deep Q-Learning
❏ Define research question more precisely