Outline

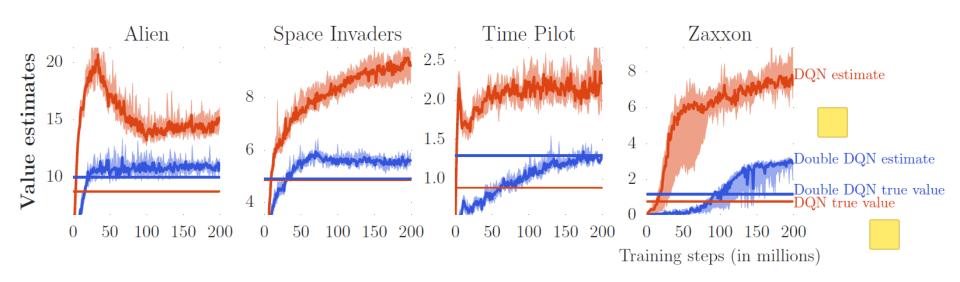
Introduction of Q-Learning

Tips of Q-Learning

Q-Learning for Continuous Actions

Double DQN

Q value is usually over-estimated

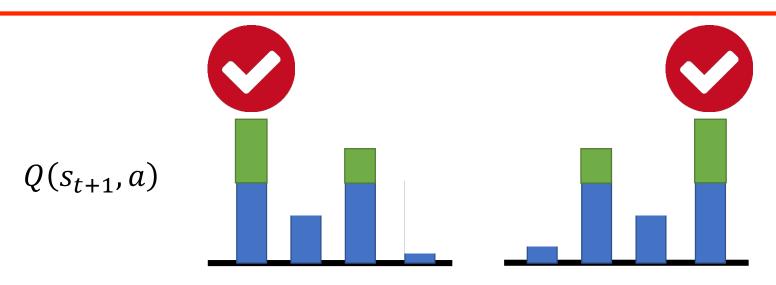


Double DQN

Q value is usually over estimate

$$Q(s_t, a_t) \longleftarrow r_t + \max_a Q(s_{t+1}, a)$$

Tend to select the action that is over-estimated

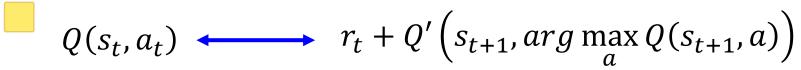


Double DQN

Q value is usually over estimate

$$Q(s_t, a_t) \longleftrightarrow r_t + \max_a Q(s_{t+1}, a)$$

• Double DQN: two functions Q and Q' Target Network

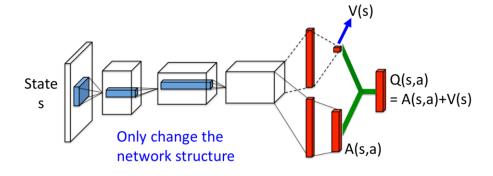


If Q over-estimate a, so it is selected. Q' would give it proper value. How about Q' overestimate? The action will not be selected by Q.

Hado V. Hasselt, "Double Q-learning", NIPS 2010 Hado van Hasselt, Arthur Guez, David Silver, "Deep Reinforcement Learning with Double Q-learning", AAAI 2016

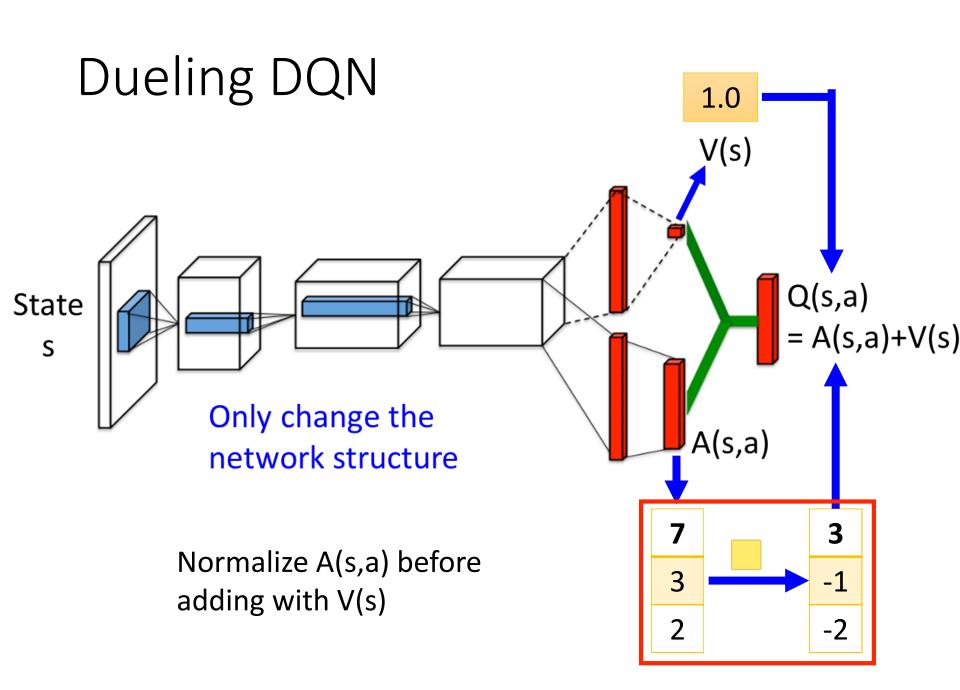
Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van 只改變原來 Deep Q Network 的架構 Hasselt, Marc Lanctot, Nando de Freitas, "Dueling Network Architectures for Deep Reinforcement **Dueling DQN** Learning", arXiv preprint, 2015 Q(s,a)State S V(s)Q(s,a) State = A(s,a)+V(s)S Only change the network structure

Dueling DQN



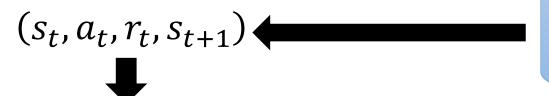


				_	
		3	3, 4	3	1
Q(s,a)	action	1	<u> </u>	6	1
II		2	-2 -1	3	1
				l	
V(s) Average of column		2	% 1	4	1
		+			
		1	3	-1	0
A(s,a)	sum of	-1	-1	2	0
	sum of column = 0	0	-2	-1	0

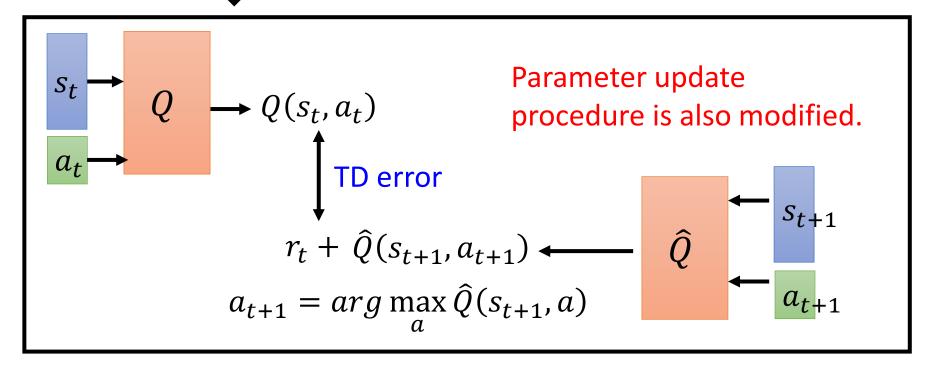


Prioritized Reply

The data with larger TD error in previous training has higher probability to be sampled.

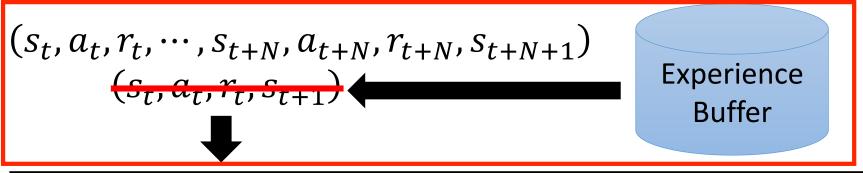


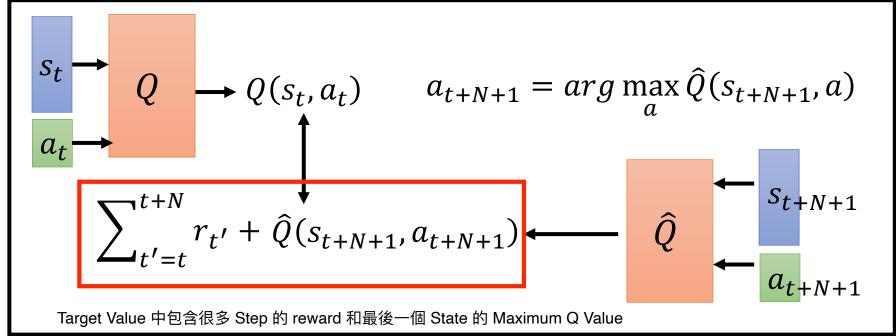
Experience Buffer



Multi-step

Balance between MC and TD





Noisy Net

https://arxiv.org/abs/1706.01905 https://arxiv.org/abs/1706.10295

Noise on Action (Epsilon Greedy)

$$a = \begin{cases} arg \max_{a} Q(s, a), & with probability 1 - \varepsilon \\ random, & otherwise \end{cases}$$

Noise on Parameters

 Inject noise into the parameters
 of Q-function at the beginning of each episode

$$a = arg \max_{a} \tilde{Q}(s, a)$$

$$Q(s, a) \longrightarrow \tilde{Q}(s, a)$$
Add noise

The noise would **NOT** change in an episode.

Noisy Net

- Noise on Action
 - Given the same state, the agent may takes different actions.
 - No real policy works in this way

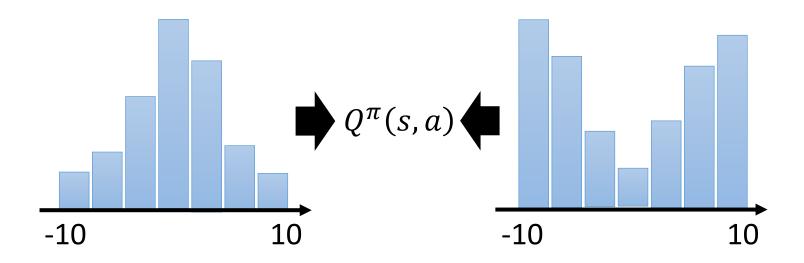
隨機亂試

- Noise on Parameters
 - Given the same (similar) state, the agent takes the same action.
 - → State-dependent Exploration
 - Explore in a consistent way

有系統地試

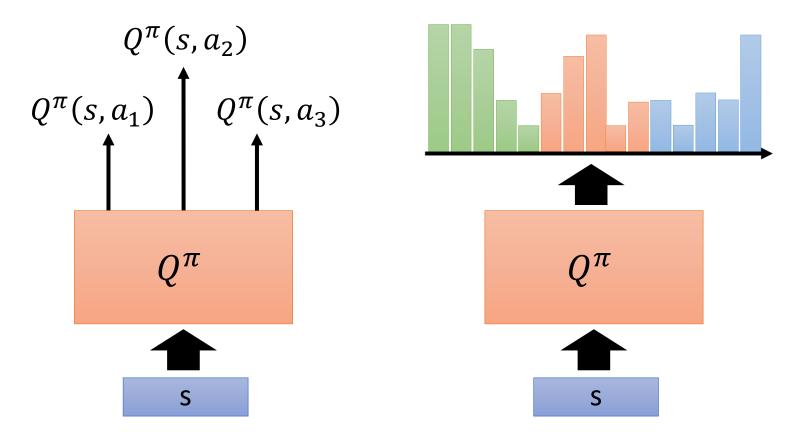
Distributional Q-function

- State-action value function $Q^{\pi}(s, a)$
 - When using actor π , the *cumulated* reward expects to be obtained after seeing observation s and taking a



Different distributions can have the same values.

Distributional Q-function



A network with 3 outputs

A network with 15 outputs (each action has 5 bins)

Rainbow

