Q-Learning

CMPT 729 G100

Jason Peng

Overview

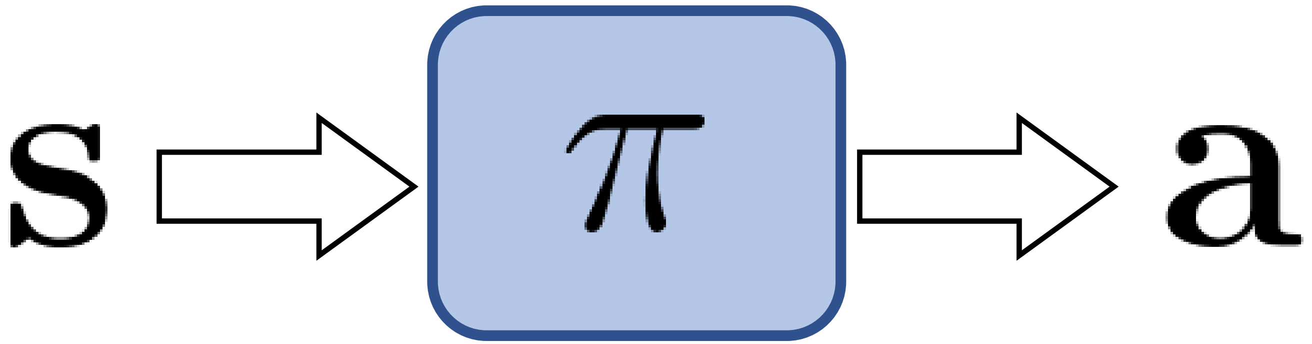
- Q-Function
- Q-Learning
- Exploration

Taxonomy of RL Algorithms

- Policy-Based Methods
- Value-Based Methods
- Actor-Critic Methods
- Model-Based Methods

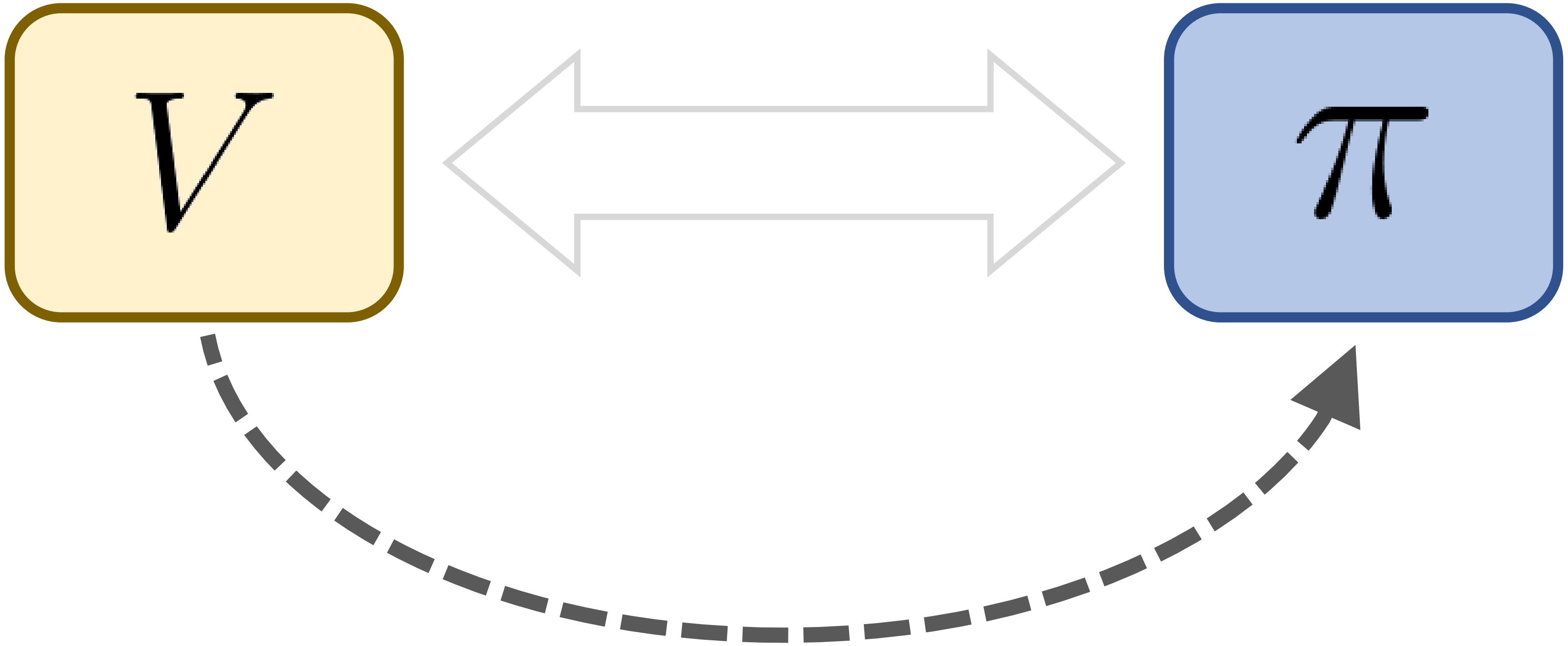
Policy-Based Methods

$$\pi(\mathbf{a}|\mathbf{s})$$



Value-Based Methods

Value-Based Methods



Policy Gradient

$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t=0}\gamma^t r_t - \cancel{V^\pi(\mathbf{s})}\right)\right]$$

Policy Gradient

$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\underline{\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}}\left[\nabla_\pi \log \pi(\mathbf{a}|\mathbf{s})\left(\sum_{t=0}\gamma^t r_t\right)\right]$$

Policy Gradient

$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\left[\nabla_\pi \log \pi(\mathbf{a}|\mathbf{s}) \underbrace{\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left(\sum_{t=0}\gamma^t r_t\right)}_{\text{“reward-to-go”} \; = Q^\pi(\mathbf{s},\mathbf{a})}\right]$$

reward-to-go: expected return of taking an action $\mathbf{a}$ in state $\mathbf{s}$

Policy Gradient

$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\left[\nabla_\pi \log \pi(\mathbf{a}|\mathbf{s})\underline{Q^\pi(\mathbf{s},\mathbf{a})}\right]$$

reward-to-go: expected return of taking an action $\mathbf{a}$ in state $\mathbf{s}$

Policy Gradient

$$\nabla_\pi J(\pi) = \mathbb{E}_{\underline{\mathbf{s} \sim d_\pi(\mathbf{s})}} \mathbb{E}_{\mathbf{a} \sim \pi(\mathbf{a}|\mathbf{s})} \left[\nabla_\pi \log \pi(\mathbf{a}|\mathbf{s}) Q^\pi(\mathbf{s}, \mathbf{a}) \right]$$

Policy Gradient

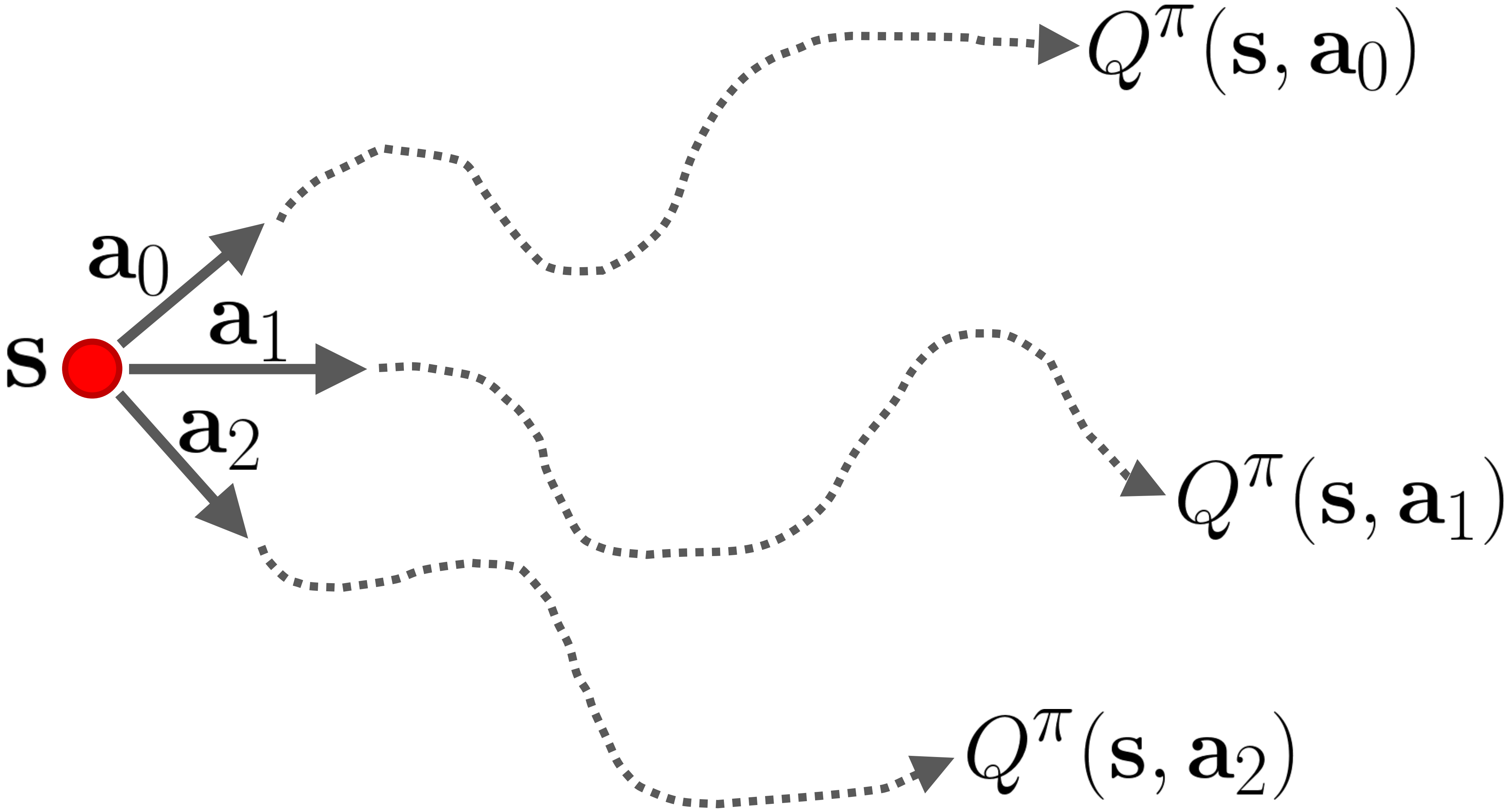
$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})} \boxed{\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\left[\nabla_\pi \log \pi(\mathbf{a}|\mathbf{s}) Q^\pi(\mathbf{s},\mathbf{a})\right]}$$

$$\Downarrow$$

$$\max_\pi \; \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\left[Q^\pi(\mathbf{s},\mathbf{a})\right]$$

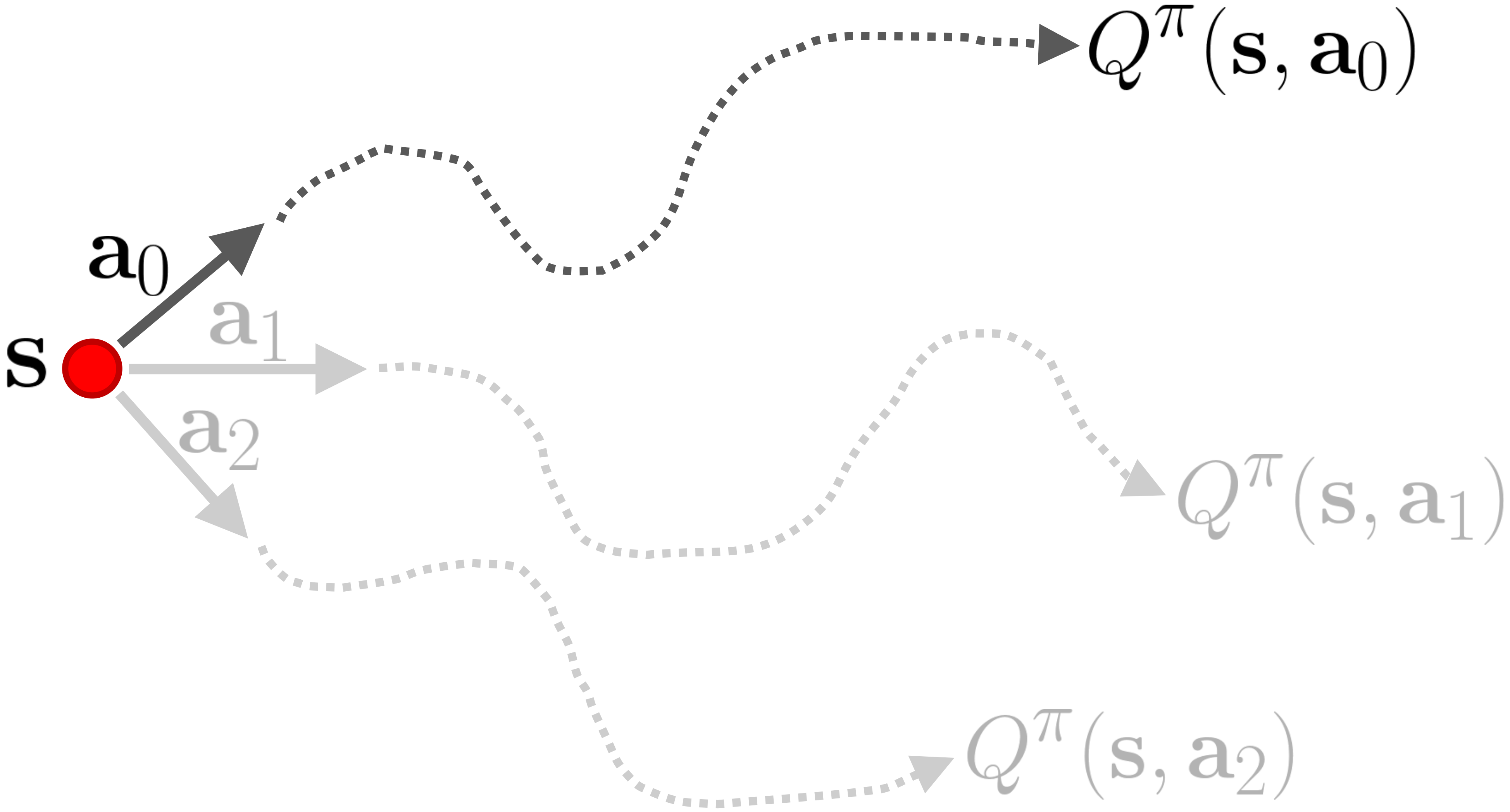
Per-state objective: pick actions that maximize the expected return at each state (i.e. Q-function)

Q-Function



$Q^{\pi}(\mathbf{s}, \mathbf{a}_0)$
$\mathbf{a}_0$
$\mathbf{a}_1$
$\mathbf{s}$
$\mathbf{a}_2$
$Q^{\pi}(\mathbf{s}, \mathbf{a}_1)$
$Q^{\pi}(\mathbf{s}, \mathbf{a}_2)$

Q-Function


$Q^{\pi}(\mathbf{s}, \mathbf{a}_0)$
$\mathbf{a}_0$
$\mathbf{s}$
$\mathbf{a}_1$
$\mathbf{a}_2$
$Q^{\pi}(\mathbf{s}, \mathbf{a}_1)$
$Q^{\pi}(\mathbf{s}, \mathbf{a}_2)$

Value Functions

Value Function

"State Value Function"

$$V^\pi(\mathbf{s}) = \mathbb{E}_{\tau \sim p(\tau | \pi, \mathbf{s}_0 = \mathbf{s})} \left[\sum_{t=0} \gamma^t r_t \right]$$

Likelihood of a trajectory starting at state $\mathbf{s}$ and then following π for all future timesteps

Q-Function

"State-Action Value Function"

$$Q^\pi(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\tau \sim p(\tau | \pi, \mathbf{s}_0 = \mathbf{s}, \mathbf{a}_0 = \mathbf{a})} \left[\sum_{t=0} \gamma^t r_t \right]$$

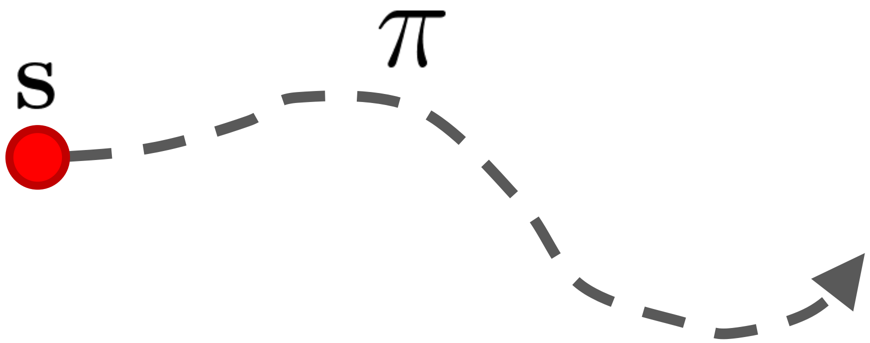
Likelihood of a trajectory after taking action $\mathbf{a}$ in state $\mathbf{s}$ and then following π for all future timesteps

Value Functions

Value Function

"State Value Function"

$$V^{\pi}(\mathbf{s}) = \mathbb{E}_{\tau \sim p(\tau | \pi, \mathbf{s}_0 = \mathbf{s})} \left[\sum_{t=0} \gamma^t r_t \right]$$

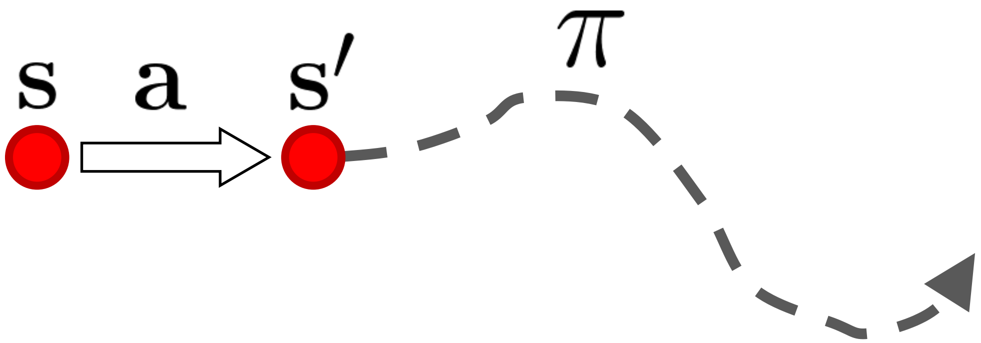


Q-Function

"State-Action Value Function"

$$Q^{\pi}(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\tau \sim p(\tau | \pi, \mathbf{s}_0 = \mathbf{s}, \mathbf{a}_0 = \mathbf{a})} \left[\sum_{t=0} \gamma^t r_t \right]$$

"Quality"



Recursive Definition

Value Function

$$V^{\pi}(\mathbf{s}) = \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underline{r(\mathbf{s},\mathbf{a},\mathbf{s}')} + \gamma V^{\pi}(\mathbf{s}')\right]$$

Recursive Definition

Value Function

$$V^\pi(\mathbf{s}) = \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \underline{V^\pi(\mathbf{s}')}\right]$$

Recursive Definition

Value Function

$$V^{\pi}(\mathbf{s}) = \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma V^{\pi}(\mathbf{s}')\right]$$

Q-Function

$$Q^{\pi}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underline{r(\mathbf{s},\mathbf{a},\mathbf{s}')} + \gamma \mathbb{E}_{\mathbf{a}'\sim\pi(\mathbf{a}'|\mathbf{s}')}\left[Q^{\pi}(\mathbf{s}',\mathbf{a}')\right]\right]$$

Recursive Definition

Value Function

$$V^\pi(\mathbf{s}) = \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma V^\pi(\mathbf{s}')\right]$$

Q-Function

$$Q^\pi(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \mathbb{E}_{\mathbf{a}'\sim\pi(\mathbf{a}'|\mathbf{s}')}\left[Q^\pi(\mathbf{s}',\mathbf{a}')\right]\right]$$

Recursive Definition

Value Function

$$V^{\pi}(\mathbf{s}) = \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma V^{\pi}(\mathbf{s}')\right]$$

Q-Function

$$Q^{\pi}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \underline{\mathbb{E}_{\mathbf{a}'\sim\pi(\mathbf{a}'|\mathbf{s}')}}\left[Q^{\pi}(\mathbf{s}',\mathbf{a}')\right]\right]$$

Recursive Definition

Value Function

$$V^\pi(\mathbf{s}) = \mathbb{E}_{\underline{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}}\mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma V^\pi(\mathbf{s}')\right]$$

Q-Function

$$Q^\pi(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \mathbb{E}_{\mathbf{a}'\sim\pi(\mathbf{a}'|\mathbf{s}')}\left[Q^\pi(\mathbf{s}',\mathbf{a}')\right]\right]$$

Recursive Definition

Value Function

$$V^{\pi}(\mathbf{s}) = \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\underbrace{\mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma V^{\pi}(\mathbf{s}')\right]}_{= Q^{\pi}(\mathbf{s},\mathbf{a})}$$

Q-Function

$$Q^{\pi}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \mathbb{E}_{\mathbf{a}'\sim\pi(\mathbf{a}'|\mathbf{s}')}\left[Q^{\pi}(\mathbf{s}',\mathbf{a}')\right]\right]$$

Recursive Definition

Value Function

$$V^{\pi}(\mathbf{s}) = \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma V^{\pi}(\mathbf{s}')\right]$$

$$= \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\left[Q^{\pi}(\mathbf{s},\mathbf{a})\right]$$

Q-Function

$$Q^{\pi}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma\mathbb{E}_{\mathbf{a}'\sim\pi(\mathbf{a}'|\mathbf{s}')}\left[Q^{\pi}(\mathbf{s}',\mathbf{a}')\right]\right]$$

Recursive Definition

Value Function

$$V^\pi(\mathbf{s}) = \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma V^\pi(\mathbf{s}')\right]$$

$$= \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\left[Q^\pi(\mathbf{s},\mathbf{a})\right]$$

Q-Function

$$Q^\pi(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underline{r(\mathbf{s},\mathbf{a},\mathbf{s}')} + \gamma\mathbb{E}_{\mathbf{a}'\sim\pi(\mathbf{a}'|\mathbf{s}')}\left[Q^\pi(\mathbf{s}',\mathbf{a}')\right]\right]$$

Recursive Definition

Value Function

$$V^\pi(\mathbf{s}) = \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma V^\pi(\mathbf{s}')\right]$$

$$= \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\left[Q^\pi(\mathbf{s},\mathbf{a})\right]$$

Q-Function

$$Q^\pi(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \underbrace{\mathbb{E}_{\mathbf{a}'\sim\pi(\mathbf{a}'|\mathbf{s}')}\left[Q^\pi(\mathbf{s}',\mathbf{a}')\right]}_{= V^\pi(\mathbf{s}')}\right]$$

Recursive Definition

Value Function

$$V^\pi(\mathbf{s}) = \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma V^\pi(\mathbf{s}')\right]$$

$$= \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\left[Q^\pi(\mathbf{s},\mathbf{a})\right]$$

Q-Function

$$Q^\pi(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \mathbb{E}_{\mathbf{a}'\sim\pi(\mathbf{a}'|\mathbf{s}')}\left[Q^\pi(\mathbf{s}',\mathbf{a}')\right]\right]$$

$$= \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma V^\pi(\mathbf{s}')\right]$$

Q-Function

$$\cancel{\pi^*(\mathbf{a}|\mathbf{s})} \Longrightarrow Q^*(\mathbf{s}, \mathbf{a})$$

Recover optimal policy:

$$\pi^*(\mathbf{a}|\mathbf{s}) = \begin{cases} 1 & \text{if } \mathbf{a} = \arg\max_{\mathbf{a}'} Q^*(\mathbf{s}, \mathbf{a}') \\ 0 & \text{otherwise} \end{cases}$$

Instead of learning policy, just learn Q-function.

Q-Function

$$\pi(\mathbf{a}|\mathbf{s}) \Longrightarrow Q^{\pi}(\mathbf{s}, \mathbf{a})$$

Recover a policy:

"arg max policy"

$$\pi'(\mathbf{a}|\mathbf{s}) = \begin{cases} 1 & \text{if } \mathbf{a} = \arg\max_{\mathbf{a}'} Q^{\pi}(\mathbf{s}, \mathbf{a}') \\ 0 & \text{otherwise} \end{cases}$$

New policy is at least as good as the old policy.

$$J(\pi') \geq J(\pi) \qquad Q^{\pi'}(\mathbf{s}, \mathbf{a}) \geq Q^{\pi}(\mathbf{s}, \mathbf{a})$$

Q-Learning

Key idea:

- Instead of trying to learn the optimal policy, just learn optimal Q-function
- Then recover policy from Q-function

Q-Learning

Recursive definition

$$Q^{\pi}(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \mathbb{E}_{\mathbf{a}' \sim \pi(\mathbf{a}'|\mathbf{s}')} \left[Q^{\pi}(\mathbf{s}', \mathbf{a}') \right] \right]$$

Optimal policy

$$Q^{*}(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \mathbb{E}_{\underline{\mathbf{a}' \sim \pi^{*}(\mathbf{a}'|\mathbf{s}')}} \left[Q^{*}(\mathbf{s}', \mathbf{a}') \right] \right]$$

$$\pi^{*}(\mathbf{a}|\mathbf{s}) = \begin{cases} 1 & \text{if } \mathbf{a} = \arg\max_{\mathbf{a}'} Q^{*}(\mathbf{s}, \mathbf{a}) \\ 0 & \text{otherwise} \end{cases}$$

Q-Learning

Recursive definition

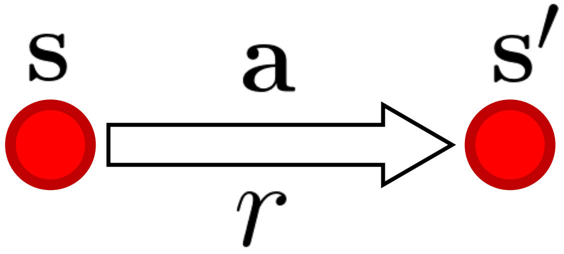
$$Q^{\pi}(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \mathbb{E}_{\mathbf{a}' \sim \pi(\mathbf{a}'|\mathbf{s}')} \left[Q^{\pi}(\mathbf{s}', \mathbf{a}') \right] \right]$$

Optimal policy

$$Q^{*}(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \max_{\mathbf{a}'} \left(Q^{*}(\mathbf{s}', \mathbf{a}') \right) \right]$$

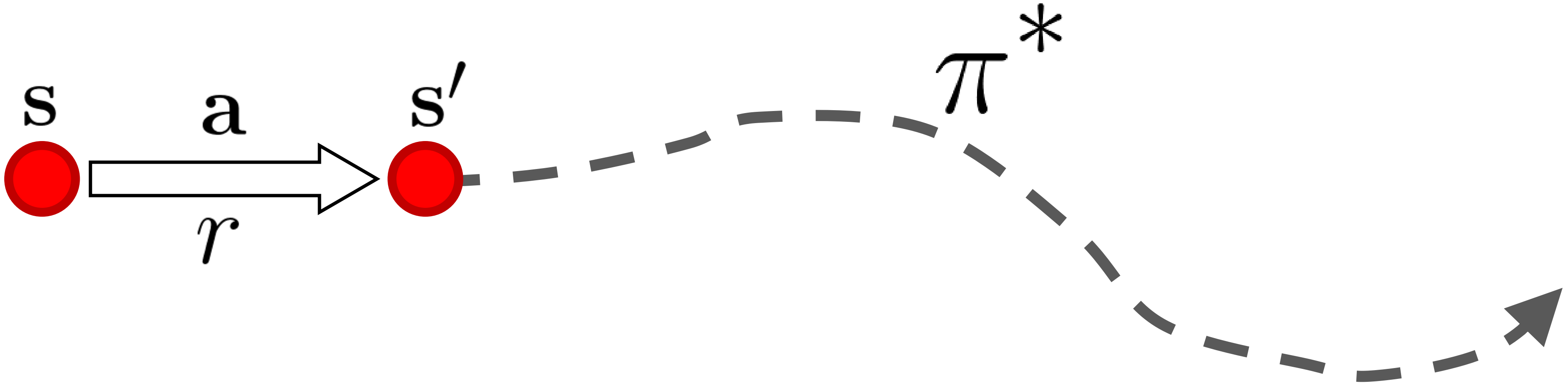
Q-Learning

$$Q^*(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underline{r(\mathbf{s},\mathbf{a},\mathbf{s}')} + \gamma \max_{\mathbf{a}'}\left(Q^*(\mathbf{s}',\mathbf{a}')\right)\right]$$



Q-Learning

$$Q^*(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \underline{\gamma \max_{\mathbf{a}'} \left(Q^*(\mathbf{s}', \mathbf{a}') \right)} \right]$$



Q-Learning

$$Q^*(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \max_{\mathbf{a}'} \left(Q^*(\mathbf{s}', \mathbf{a}') \right) \right]$$

Not true for non-optimal policies

$$Q^\pi(\mathbf{s}, \mathbf{a}) \neq \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \max_{\mathbf{a}'} \left(Q^\pi(\mathbf{s}', \mathbf{a}') \right) \right]$$

Q-Learning

$$Q^*(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^*(\mathbf{s}',\mathbf{a}')\right)\right]$$

Not true for non-optimal policies

$$Q^\pi(\mathbf{s},\mathbf{a}) \leq \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \underline{\max_{\mathbf{a}'}}\left(Q^\pi(\mathbf{s}',\mathbf{a}')\right)\right]$$

$$\geq$$

$$Q^\pi(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \underline{\mathbb{E}_{\mathbf{a}'\sim\pi(\mathbf{a}'|\mathbf{s}')}}\left[Q^\pi(\mathbf{s}',\mathbf{a}')\right]\right]$$

Q-Learning

$$Q^*(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^*(\mathbf{s}',\mathbf{a}')\right)\right]$$

Not true for non-optimal policies

$$Q^\pi(\mathbf{s},\mathbf{a}) \leq \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^\pi(\mathbf{s}',\mathbf{a}')\right)\right]$$

Q-Learning

$$Q^*(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \max_{\mathbf{a}'} \left(Q^*(\mathbf{s}', \mathbf{a}') \right) \right]$$

Not true for non-optimal policies

$$Q^\pi(\mathbf{s}, \mathbf{a}) \leq \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[\underline{r(\mathbf{s}, \mathbf{a}, \mathbf{s}')} + \gamma \max_{\mathbf{a}'} \left(Q^\pi(\mathbf{s}', \mathbf{a}') \right) \right]$$

Q-Learning

$$Q^*(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \max_{\mathbf{a}'} \left(Q^*(\mathbf{s}', \mathbf{a}') \right) \right]$$

Not true for non-optimal policies

$$Q^\pi(\mathbf{s}, \mathbf{a}) \leq \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \underbrace{\max_{\mathbf{a}'} \left(Q^\pi(\mathbf{s}', \mathbf{a}') \right)}_{\text{arg max policy}} \right]$$

Q-Learning

$$Q^*(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^*(\mathbf{s}',\mathbf{a}')\right)\right]$$

Not true for non-optimal policies

$$Q^\pi(\mathbf{s},\mathbf{a}) \leq \underbrace{\mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^\pi(\mathbf{s}',\mathbf{a}')\right)\right]}_{Q^{\pi'}(\mathbf{s},\mathbf{a})}$$

$$\Downarrow$$

$$Q^\pi(\mathbf{s},\mathbf{a}) \leq Q^{\pi'}(\mathbf{s},\mathbf{a})$$

Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\underline{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

Q-Learning

$$Q^{k+1}(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[\underline{r(\mathbf{s}, \mathbf{a}, \mathbf{s}')} + \gamma \max_{\mathbf{a}'} \left(Q^k(\mathbf{s}', \mathbf{a}') \right) \right]$$

Q-Learning

$$Q^{k+1}(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \underline{\max_{\mathbf{a}'} \left(Q^k(\mathbf{s}', \mathbf{a}') \right)} \right]$$

Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

$$Q^{k+1}(\mathbf{s},\mathbf{a}) \geq Q^k(\mathbf{s},\mathbf{a})$$

$$\boxed{Q^0} \underset{\leq}{\Rightarrow} \boxed{Q^1} \underset{\leq}{\Rightarrow} \boxed{Q^2} \Rightarrow \cdots \Rightarrow \underbrace{\boxed{Q^k} \Rightarrow \boxed{Q^{k+1}}}_{\substack{Q^{k+1}(\mathbf{s},\mathbf{a}) = Q^k(\mathbf{s},\mathbf{a}) \\ Q^k(\mathbf{s},\mathbf{a}) = Q^*(\mathbf{s},\mathbf{a})}}$$

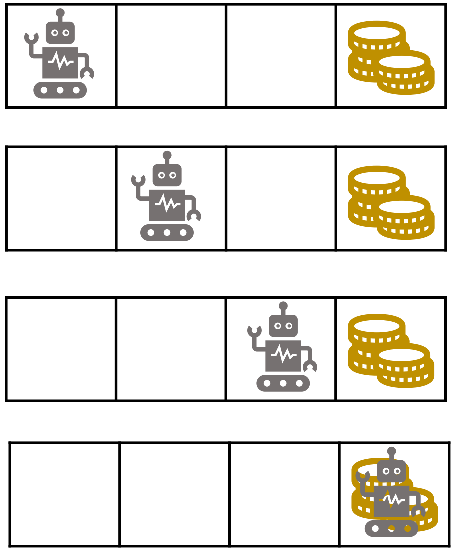
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

Iteration 0:

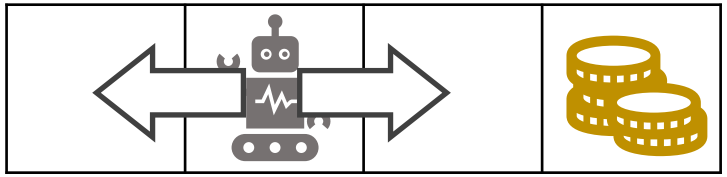
Action

State



Q	$\leftarrow$	$\square$	$\rightarrow$
	0	0	0
	0	0	0
	0	0	0
	0	0	0

Environment



$$\gamma = 1/2$$

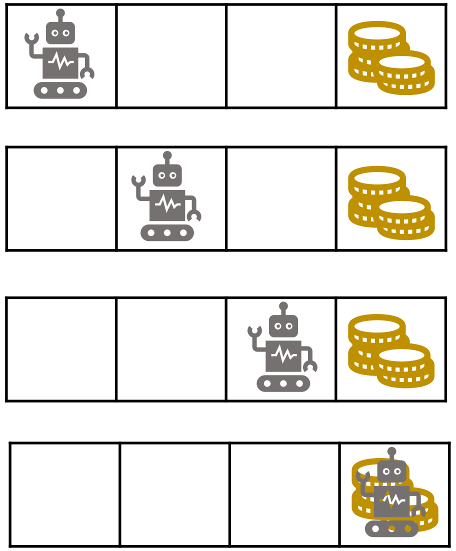
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underline{r(\mathbf{s},\mathbf{a},\mathbf{s}')} + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

Iteration 0:

Action

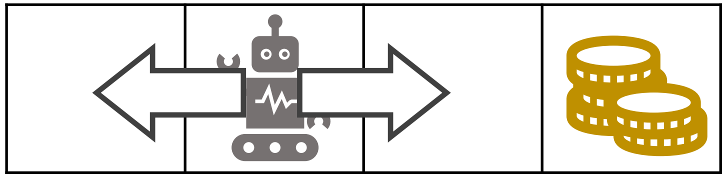
State



Q

←	□	→
0	0	0
0	0	0
0	0	0
0	0	0

Environment



$$\gamma = 1/2$$

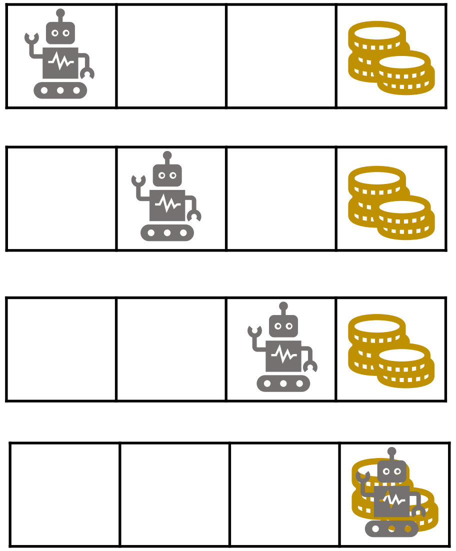
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

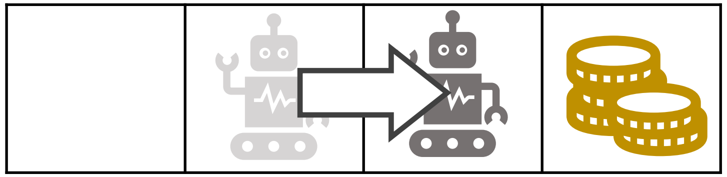
Iteration 0:

Action

Q

State	←	□	→
	0	0	0
	0	0	0
	0	0	0
	0	0	0

Environment



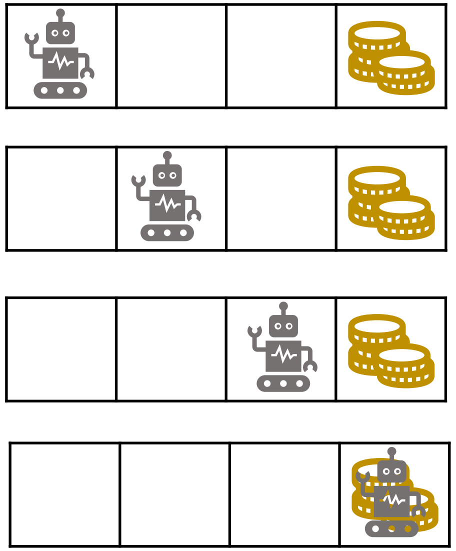
$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=1} + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

Iteration 0:

State

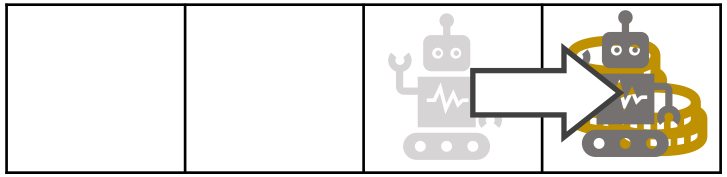


Action

Q

←	□	→
0	0	0
0	0	0
0	0	0
0	0	0

Environment



$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

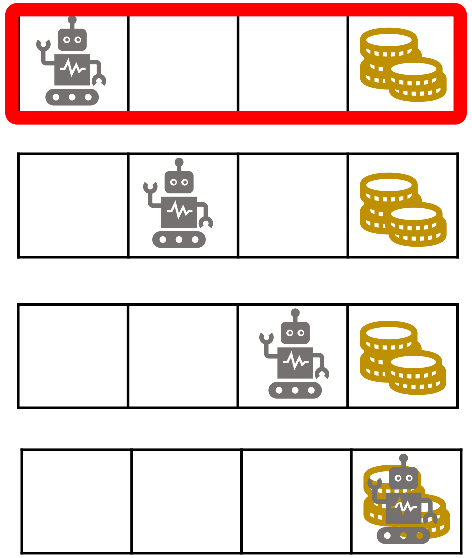
Iteration 0:

Action

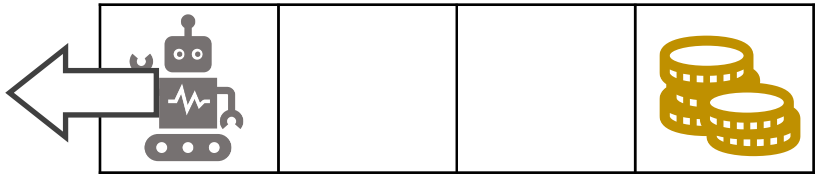
Q

←	□	→
$\boxed{0}$	0	0
0	0	0
0	0	0
0	0	0

State



Environment



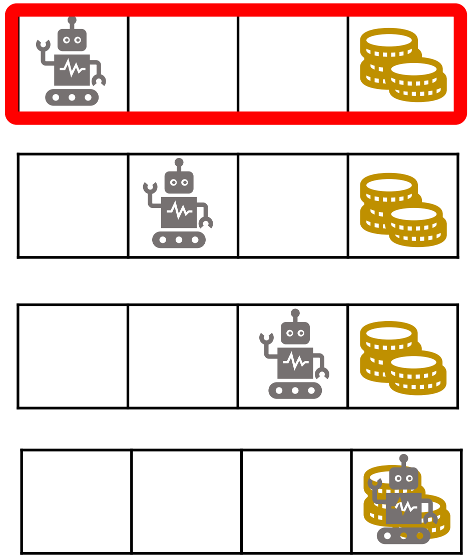
$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

Iteration 0:

State

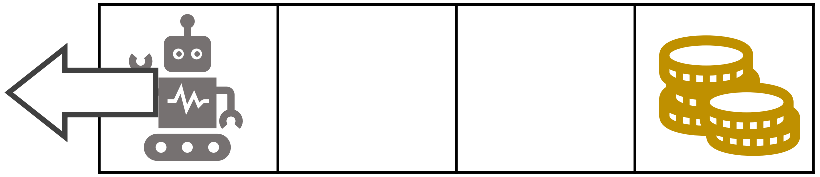


Q

Action

←	□	→
0	0	0
0	0	0
0	0	0
0	0	0

Environment



$$\gamma = 1/2$$

Tabular Q-Learning

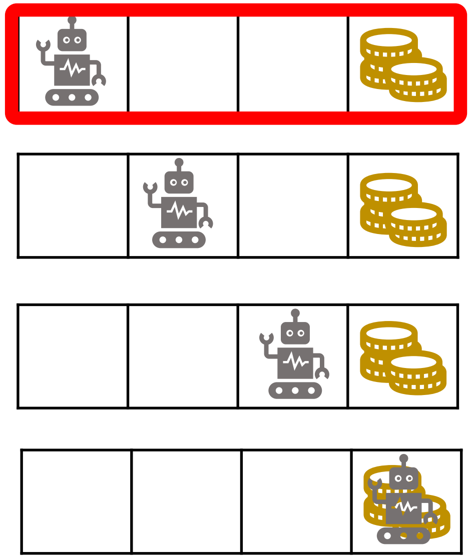
$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

Iteration 0:

Action

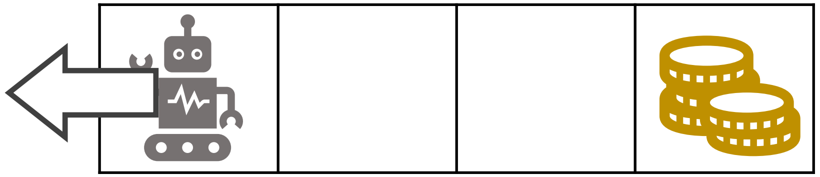
Q

State



⇦	□	⇨
$\boxed{0}$	0	0
0	0	0
0	0	0
0	0	0

Environment



$$\gamma = 1/2$$

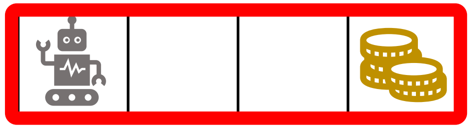
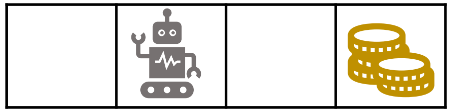
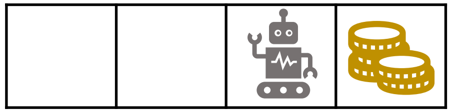
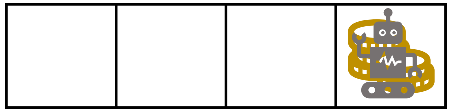
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

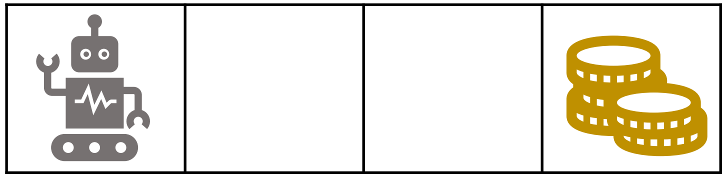
Iteration 0:

Q

State / Action

	⇦	□	⇨
	0	$\boxed{0}$	0
	0	0	0
	0	0	0
	0	0	0

Environment



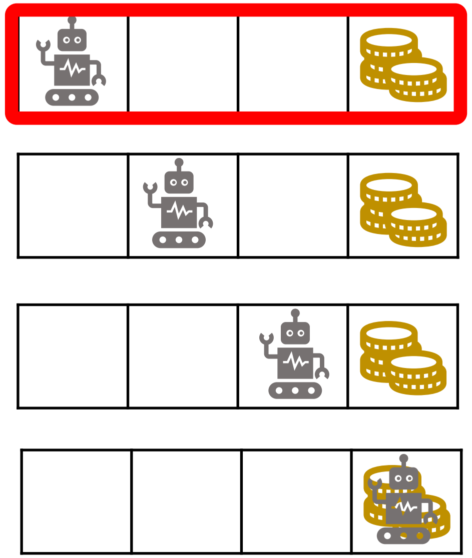
$\gamma = 1/2$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \underbrace{\gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

Iteration 0:

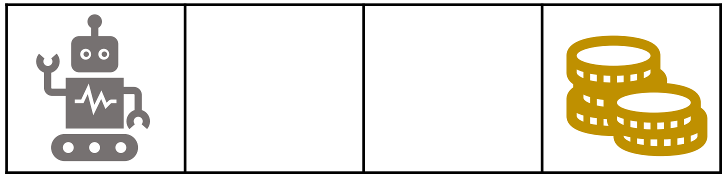
State



Q — Action

$\Leftarrow$	$\square$	$\Rightarrow$
0	0	0
0	0	0
0	0	0
0	0	0

Environment



$$\gamma = 1/2$$

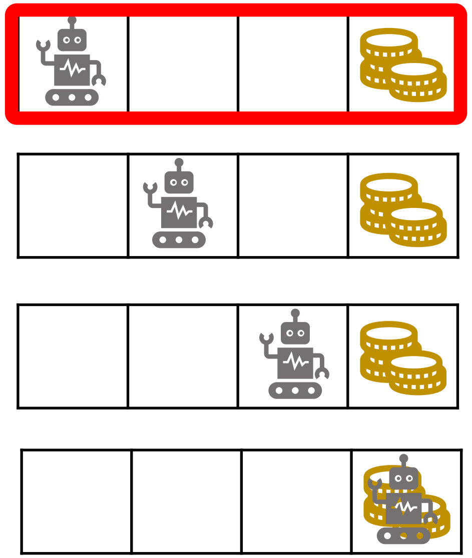
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

Iteration 0:

Action

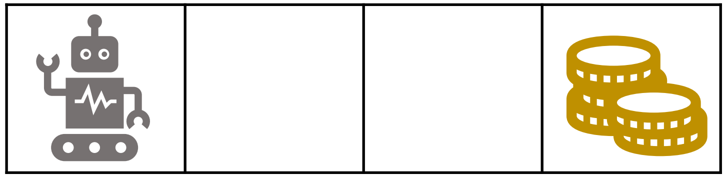
Q



State

⇦	□	⇨
0	$\boxed{0}$	0
0	0	0
0	0	0
0	0	0

Environment



$$\gamma = 1/2$$

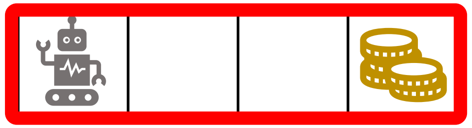
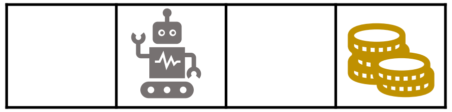
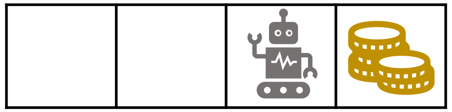
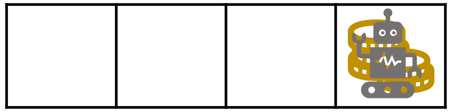
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underline{r(\mathbf{s},\mathbf{a},\mathbf{s}')} + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

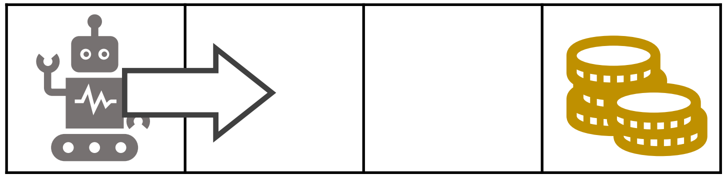
Iteration 0:

Action

State

Q	⇦	□	⇨
	0	0	$\boxed{0}$
	0	0	0
	0	0	0
	0	0	0

Environment



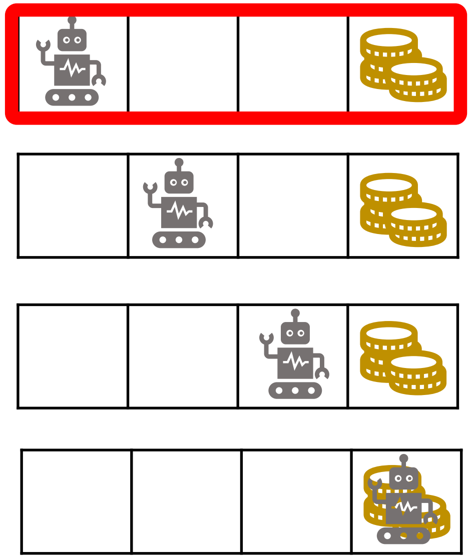
$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

Iteration 0:

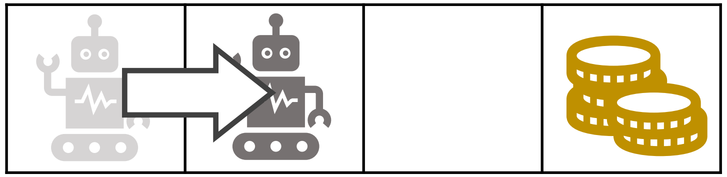
State



Action

Q	$\leftarrow$	$\square$	$\rightarrow$
	0	0	0
	0	0	0
	0	0	0
	0	0	0

Environment



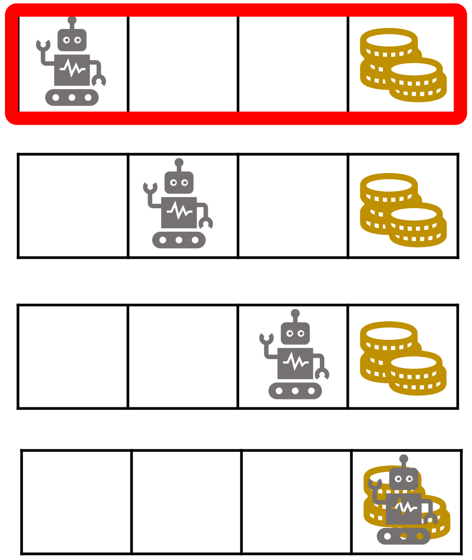
$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

Iteration 0:

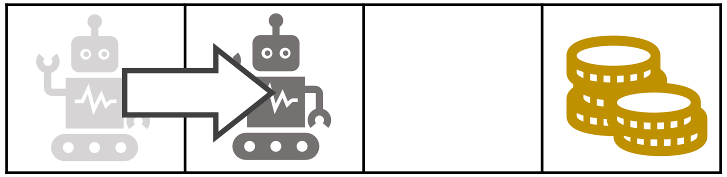
State



Q

Action: ←	□	→
0	0	0
0	0	0
0	0	0
0	0	0

Environment



$$\gamma = 1/2$$

Tabular Q-Learning

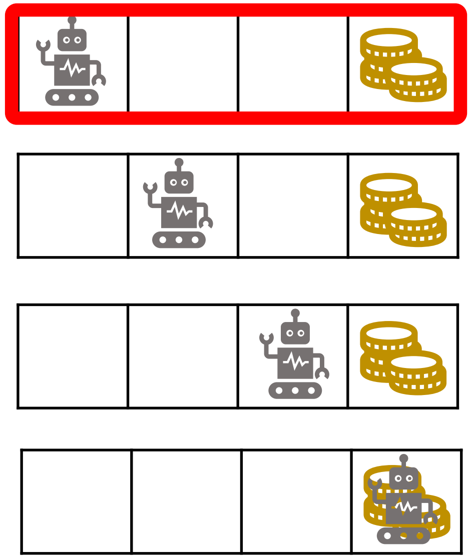
$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

Iteration 0:

Action

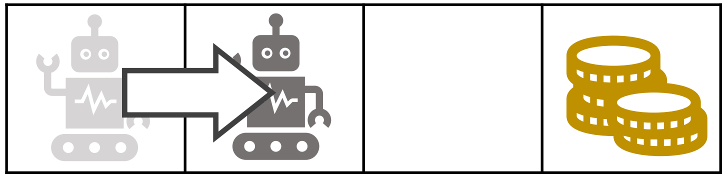
Q

State



←	□	→
0	0	$\boxed{0}$
0	0	0
0	0	0
0	0	0

Environment



$$\gamma = 1/2$$

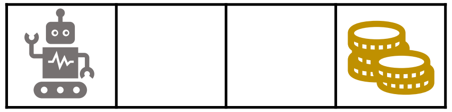
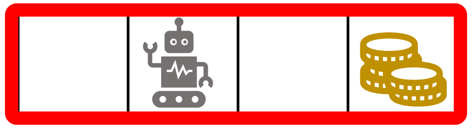
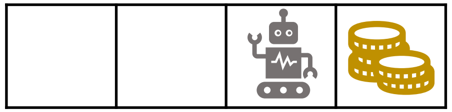
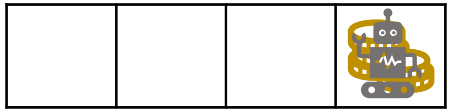
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \max_{\mathbf{a}'} \left(Q^k(\mathbf{s}', \mathbf{a}') \right) \right]$$

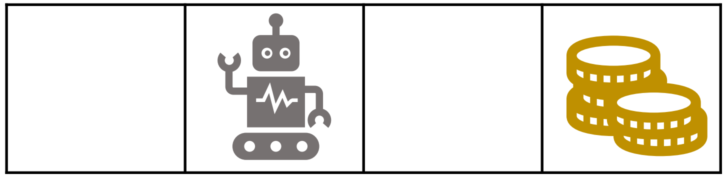
Iteration 0:

Action

Q

State	$\leftarrow$	$\square$	$\rightarrow$
	0	0	0
	0	0	0
	0	0	0
	0	0	0

Environment



$$\gamma = 1/2$$

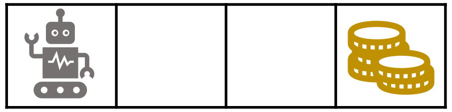
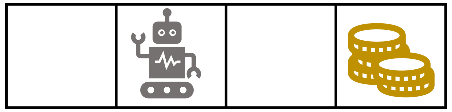
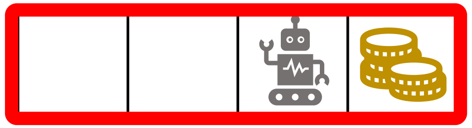
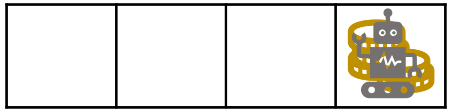
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

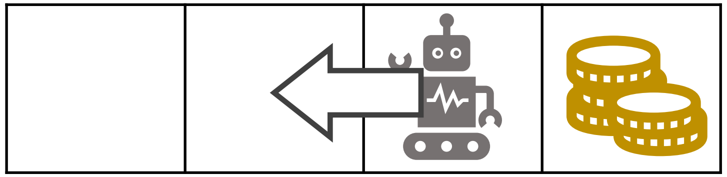
Iteration 0:

Q

Action

State	←	□	→
	0	0	0
	0	0	0
	$\boxed{0}$	0	0
	0	0	0

Environment

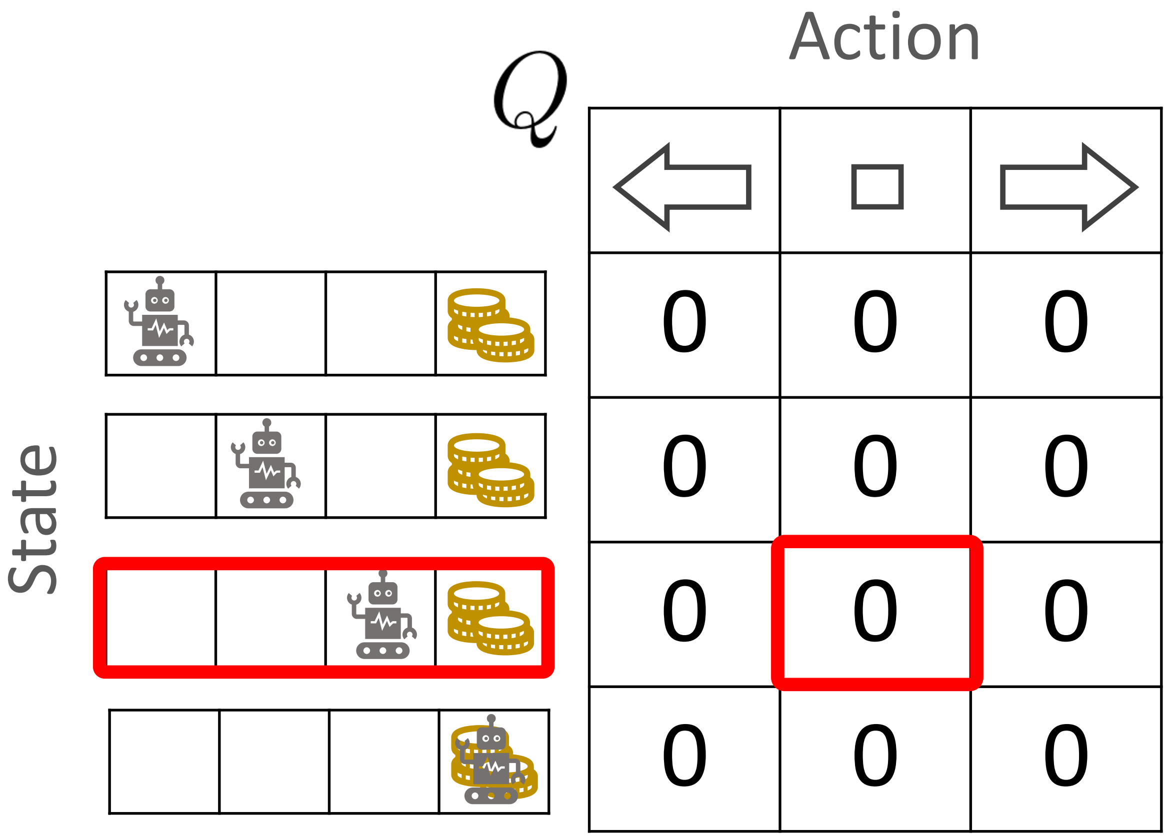


$$\gamma = 1/2$$

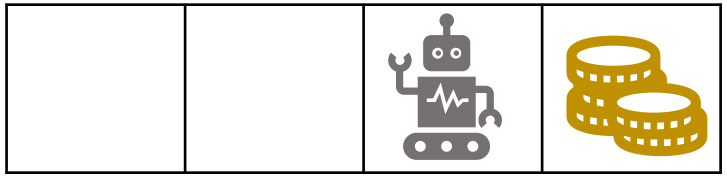
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

Iteration 0:



Environment



$$\gamma = 1/2$$

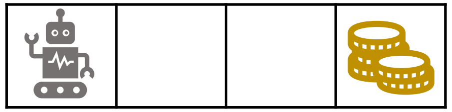
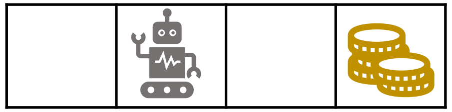
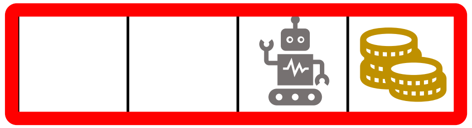
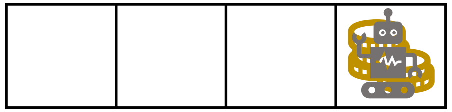
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

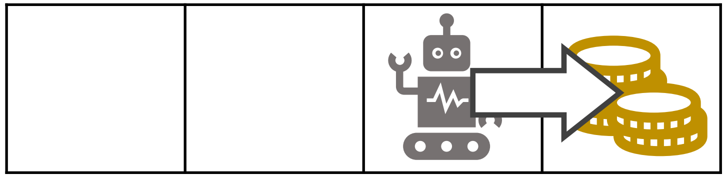
Iteration 0:

Action

Q

State	⇦	□	⇨
	0	0	0
	0	0	0
	0	0	$\boxed{0}$
	0	0	0

Environment



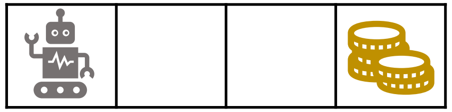
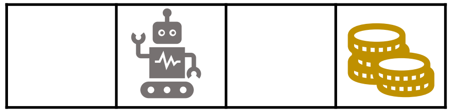
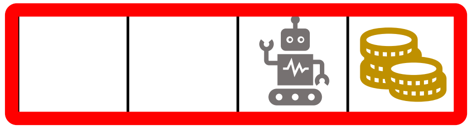
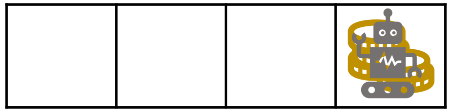
$$\gamma = 1/2$$

Tabular Q-Learning

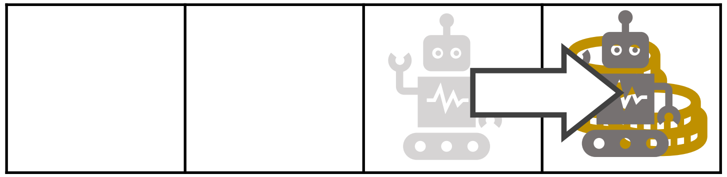
$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=1} + \gamma \underline{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}\right]$$

Iteration 0:

Q

State	Action: ⇦	□	⇨
	0	0	0
	0	0	0
	0	0	$\boxed{0}$
	0	0	0

Environment



$\gamma = 1/2$

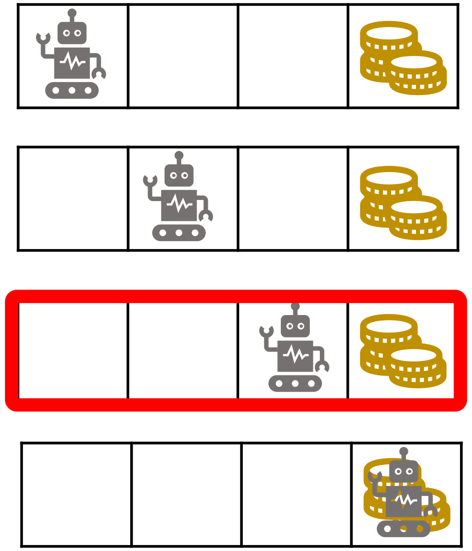
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=1} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

Iteration 0:

Action

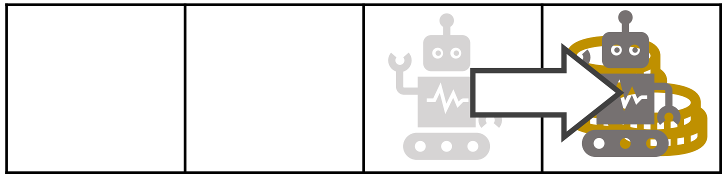
Q



⇦	□	⇨
0	0	0
0	0	0
0	0	$\boxed{0}$
0	0	0

State

Environment



$$\gamma = 1/2$$

Tabular Q-Learning

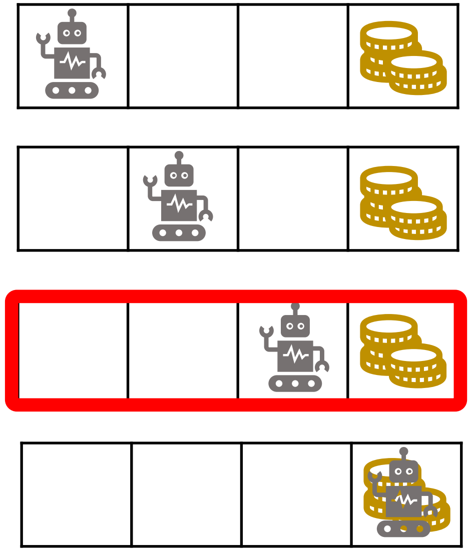
$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=1} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

Iteration 0:

Action

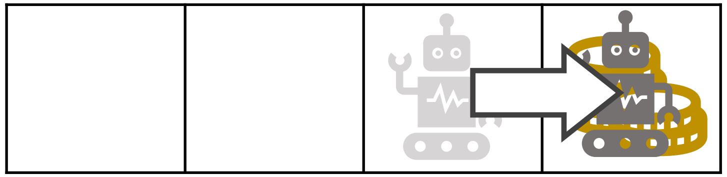
Q

State



⇦	□	⇨
0	0	0
0	0	0
0	0	**1**
0	0	0

Environment



$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underline{r(\mathbf{s},\mathbf{a},\mathbf{s}')} + \gamma \, \underline{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}\right]$$

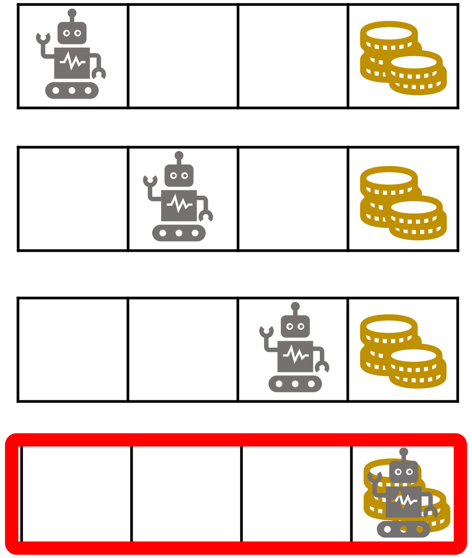
$$= 0$$

Iteration 0:

Action

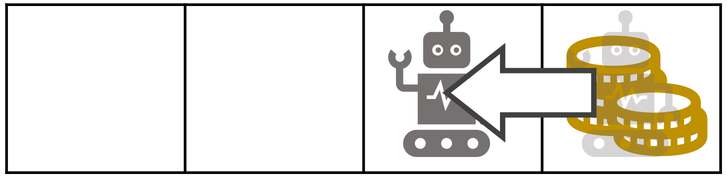
Q

State



⇦	□	⇨
0	0	0
0	0	0
0	0	1
0	0	0

Environment



$$\gamma = 1/2$$

Tabular Q-Learning

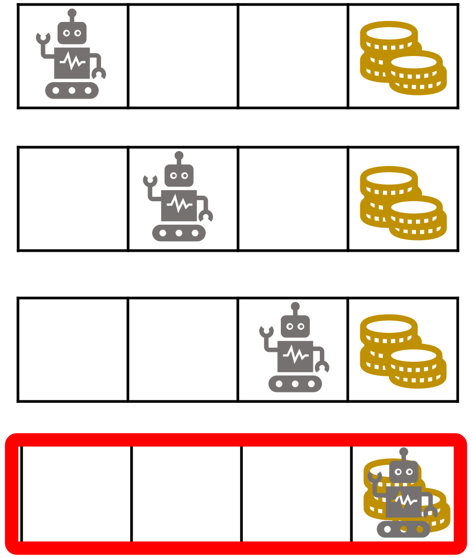
$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=1}\right]$$

Iteration 0:

Action

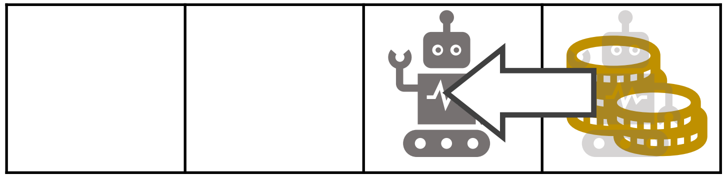
Q

State



←	□	→
0	0	0
0	0	0
$\boxed{0}$	$\boxed{0}$	$\boxed{1}$
0	0	0

Environment



$$\gamma = 1/2$$

Tabular Q-Learning

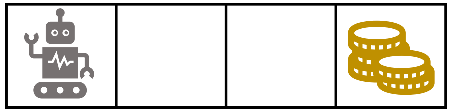
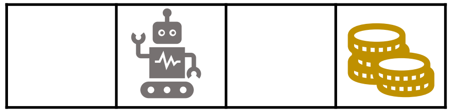
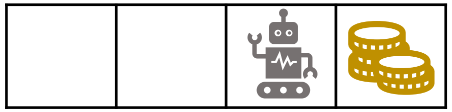
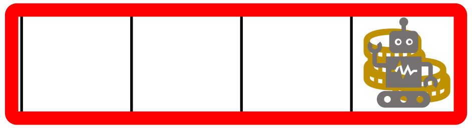
$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=1}\right]$$

Iteration 0:

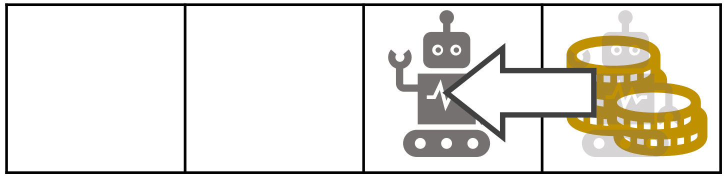
Action

Q

State

	←	□	→
	0	0	0
	0	0	0
	0	0	1
	$\boxed{1/2}$	0	0

Environment



$$\gamma = 1/2$$

Tabular Q-Learning

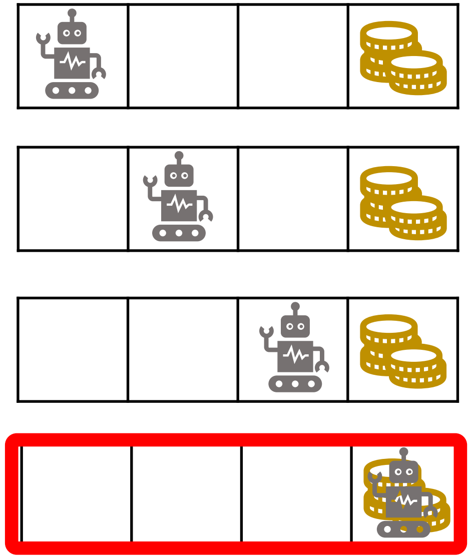
$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

Iteration 0:

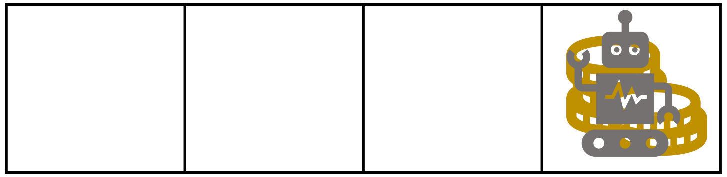
Action

Q	←	□	→
State 1	0	0	0
State 2	0	0	0
State 3	0	0	1
State 4	1/2	0	0

State



Environment



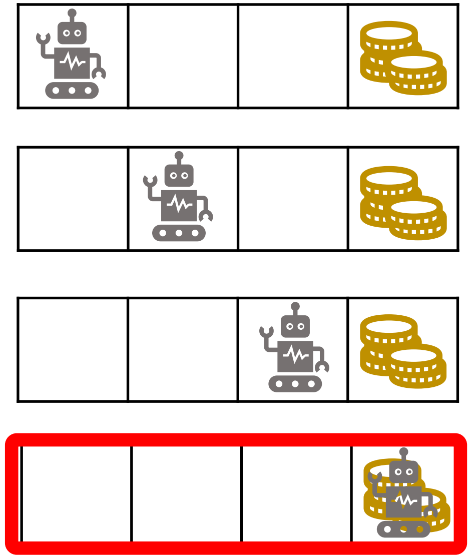
$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=1} + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

Iteration 0:

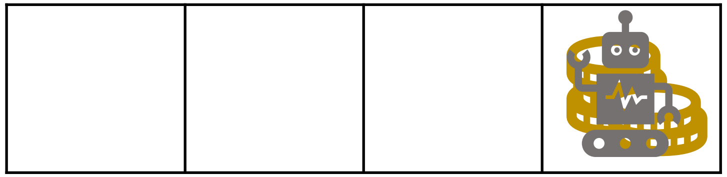
State



Q (Action)

←	□	→
0	0	0
0	0	0
0	0	1
1/2	$\boxed{0}$	0

Environment



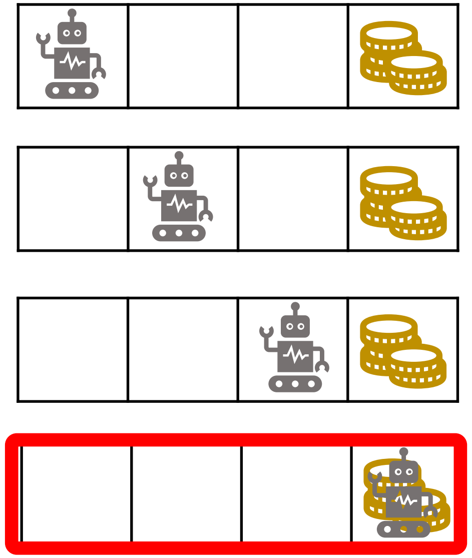
$\gamma = 1/2$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=1} + \underbrace{\gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

Iteration 0:

State

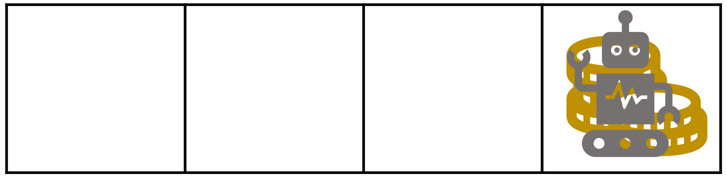


Action

Q

⇦	□	⇨
0	0	0
0	0	0
0	0	1
1/2	$\boxed{0}$	0

Environment



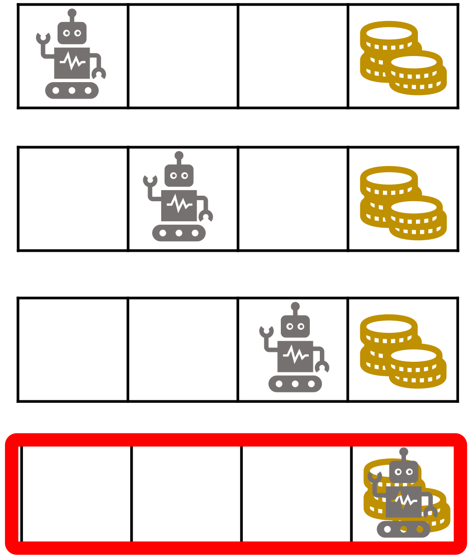
$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=1} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

Iteration 0:

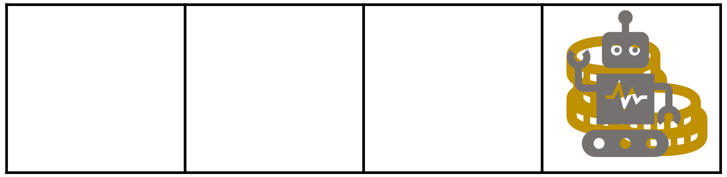
State



Q — Action

⇦	□	⇨
0	0	0
0	0	0
0	0	1
1/2	$\boxed{1}$	0

Environment



$$\gamma = 1/2$$

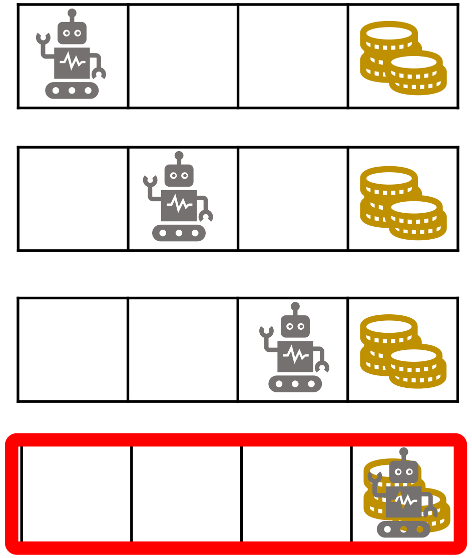
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=1} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

Iteration 0:

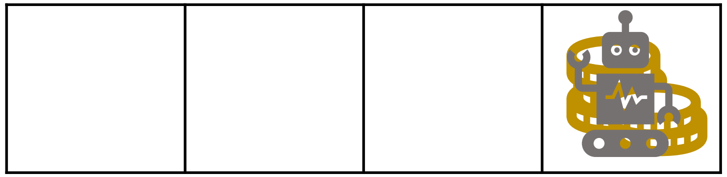
Action

State



Q	⇦	□	⇨
	0	0	0
	0	0	0
	0	0	1
	1/2	1	$\boxed{0}$

Environment



$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=1} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

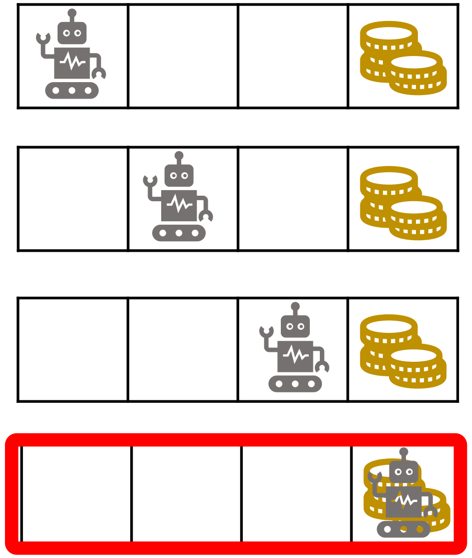
Iteration 0:

Action

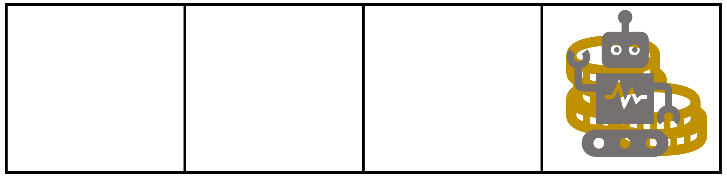
Q

←	□	→
0	0	0
0	0	0
0	0	1
1/2	1	$\boxed{1}$

State



Environment

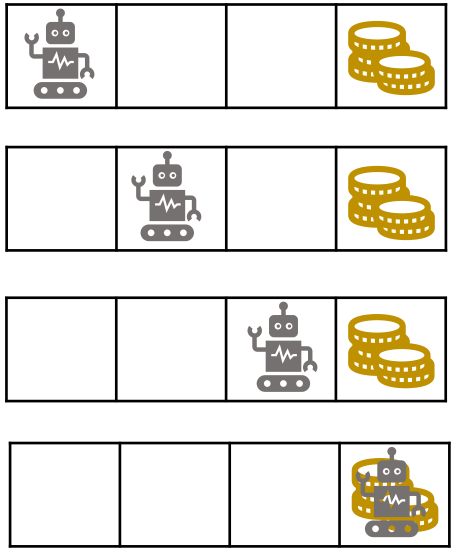


$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

Iteration 0:

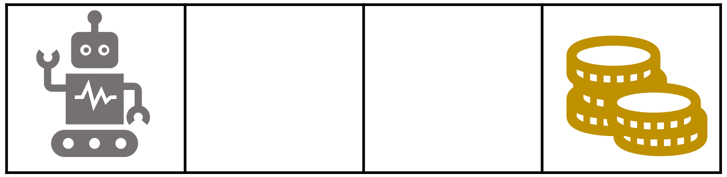


State

Q

	Action			
	←	□	→	π
State 1	0	0	0	?
State 2	0	0	0	?
State 3	0	0	1	→
State 4	1/2	1	1	□ →

Environment



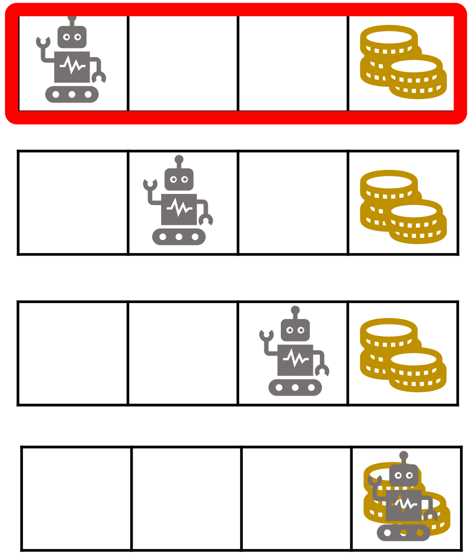
$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

Iteration 1:

State

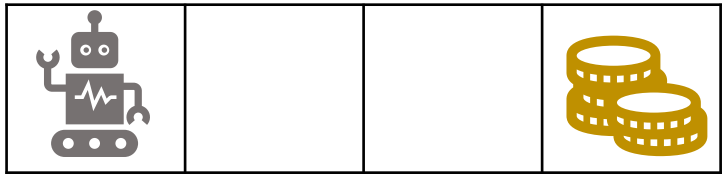


Action

Q

⇦	□	⇨
0	0	0
0	0	0
0	0	1
1/2	1	1

Environment



$$\gamma = 1/2$$

Tabular Q-Learning

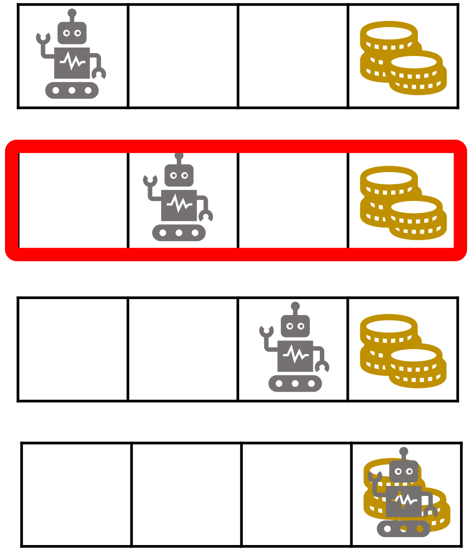
$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^{k}(\mathbf{s}',\mathbf{a}')\right)\right]$$

Iteration 1:

Action

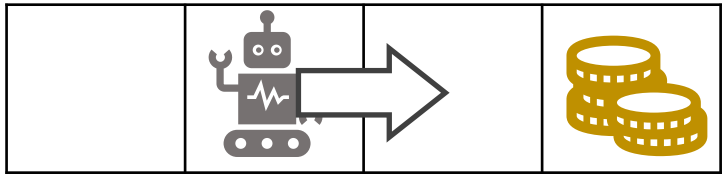
Q

State



⇦	□	⇨
0	0	0
0	0	$\boxed{0}$
0	0	1
1/2	1	1

Environment



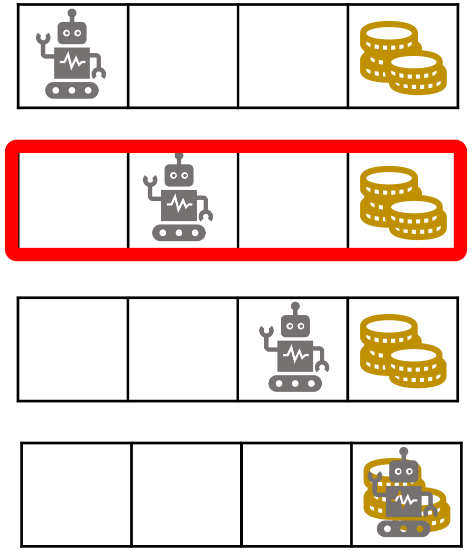
$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

Iteration 1:

State

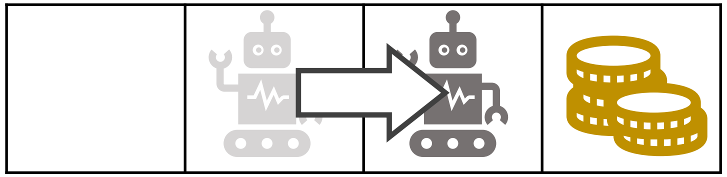


Action

Q

←	□	→
0	0	0
0	0	$\boxed{0}$
0	0	1
1/2	1	1

Environment



$$\gamma = 1/2$$

Tabular Q-Learning

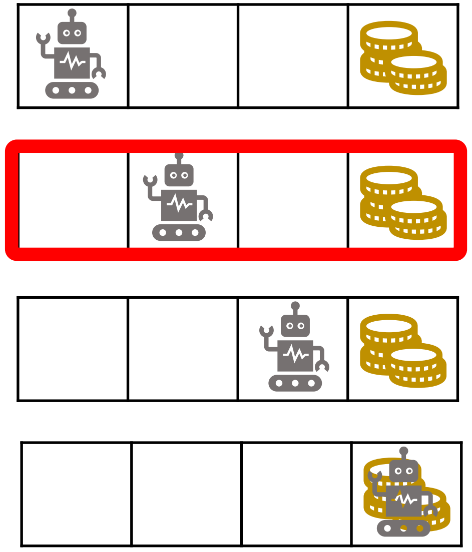
$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=1}\right]$$

Iteration 1:

Action

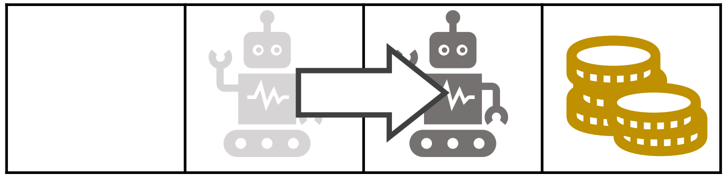
Q

State



⇦	□	⇨
0	0	0
0	0	0
0	0	1
1/2	1	1

Environment



$\gamma = 1/2$

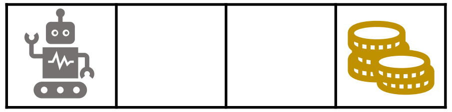
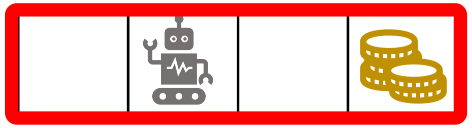
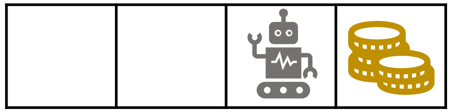
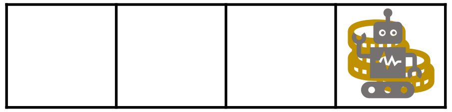
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=1}\right]$$

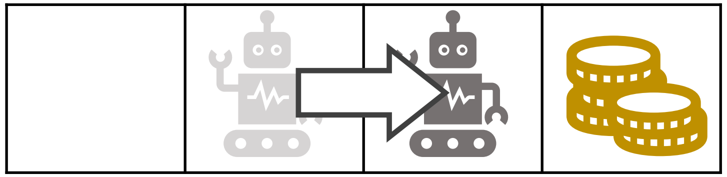
Iteration 1:

Action

Q

State	⇦	□	⇨
	0	0	0
	0	0	$\boxed{1/2}$
	0	0	1
	1/2	1	1

Environment



$$\gamma = 1/2$$

Tabular Q-Learning

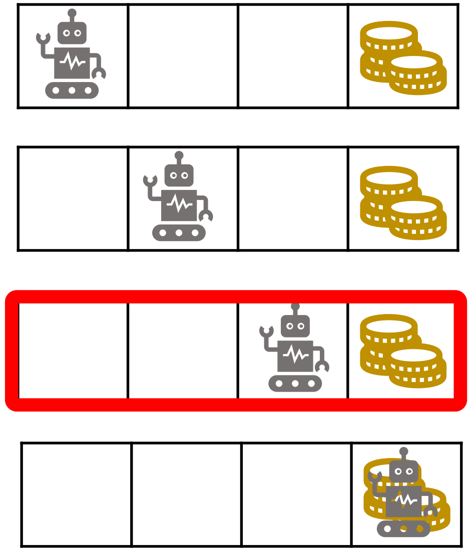
$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

Iteration 1:

Action

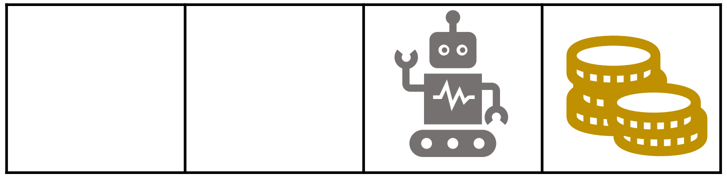
Q

State



⇦	□	⇨
0	0	0
0	0	1/2
0	0	1
1/2	1	1

Environment



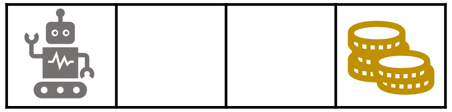
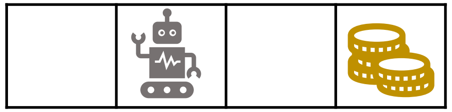
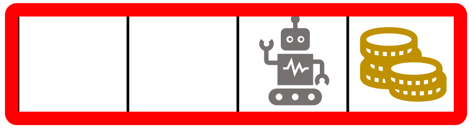
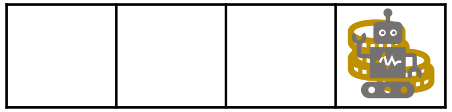
$$\gamma = 1/2$$

Tabular Q-Learning

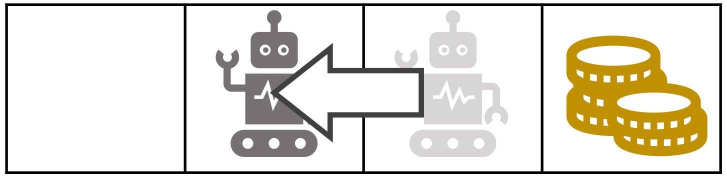
$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

Iteration 1:

Action

Q (State)	⇦	□	⇨
	0	0	0
	0	0	1/2
	$\boxed{0}$	0	1
	1/2	1	1

Environment



$$\gamma = 1/2$$

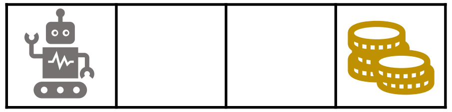
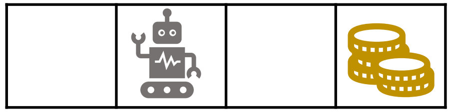
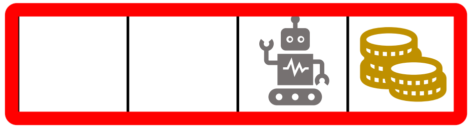
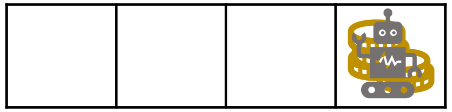
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \gamma\,\underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=1/2}\right]$$

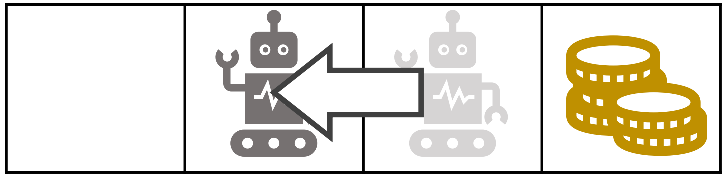
Iteration 1:

Action

Q

State	⇦	□	⇨
	0	0	0
	0	0	1/2
	0	0	1
	1/2	1	1

Environment



$$\gamma = 1/2$$

Tabular Q-Learning

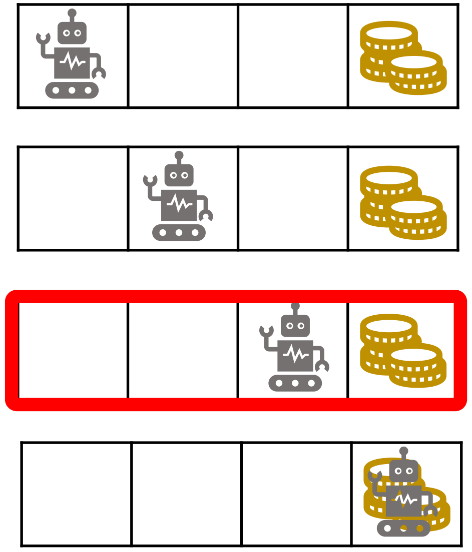
$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=1/2}\right]$$

Iteration 1:

Action

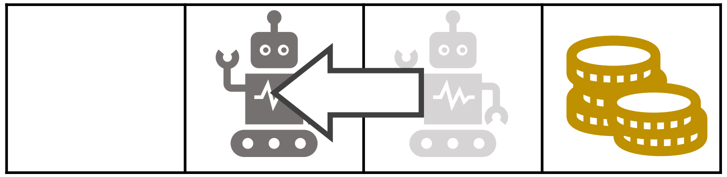
Q

State



⇦	□	⇨
0	0	0
0	0	1/2
$\boxed{1/4}$	0	1
1/2	1	1

Environment



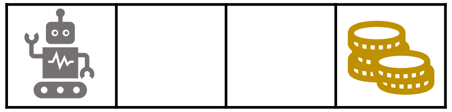
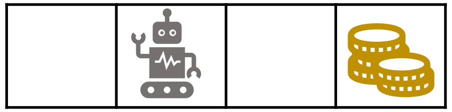
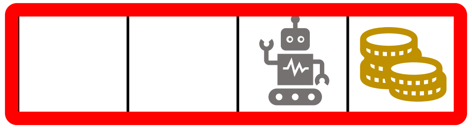
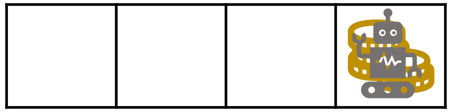
$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=0}\right]$$

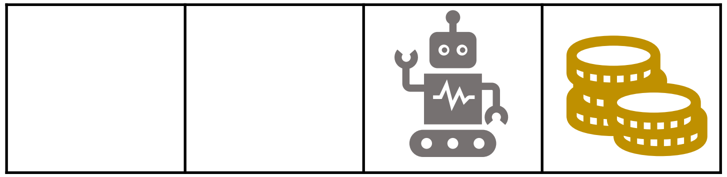
Iteration 1:

Action

Q	←	□	→
	0	0	0
	0	0	1/2
	1/4	0	1
	1/2	1	1

State

Environment



$$\gamma = 1/2$$

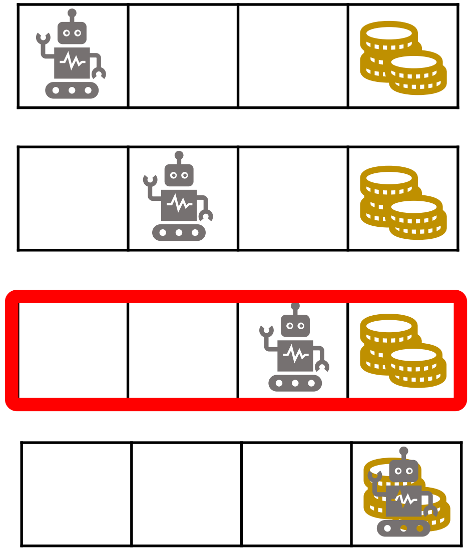
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=1}\right]$$

Iteration 1:

Action

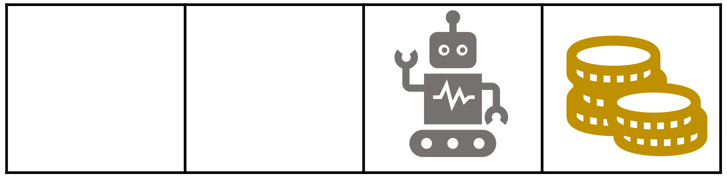
Q



State

⇦	□	⇨
0	0	0
0	0	1/2
1/4	0	1
1/2	1	1

Environment



$$\gamma = 1/2$$

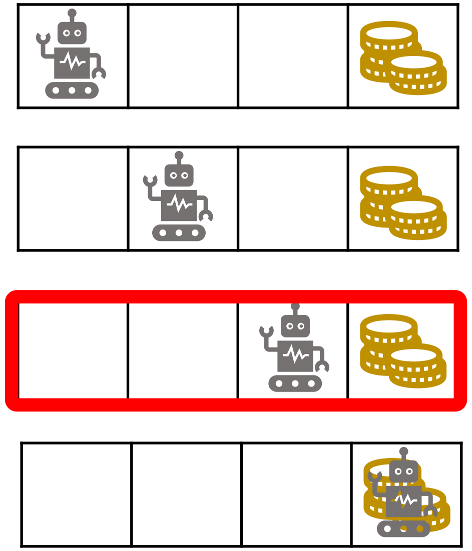
Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=1}\right]$$

Iteration 1:

Action

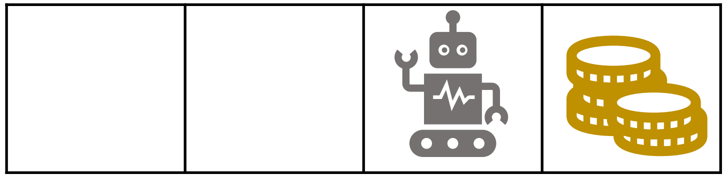
State



Q

←	□	→
0	0	0
0	0	1/2
1/4	$\boxed{1/2}$	1
1/2	1	1

Environment



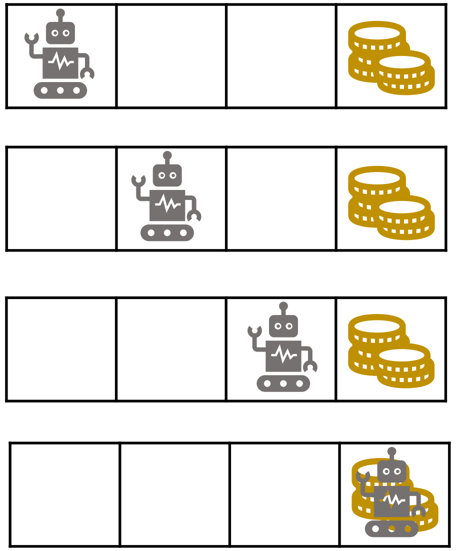
$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[\underbrace{r(\mathbf{s},\mathbf{a},\mathbf{s}')}_{=0} + \gamma \underbrace{\max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)}_{=1}\right]$$

Iteration 1:

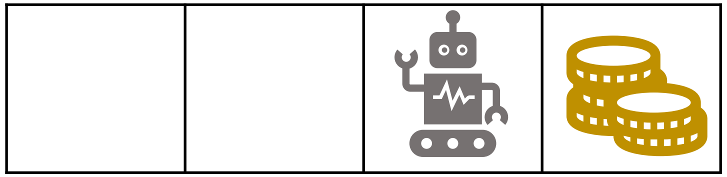
State



Action

Q	⇦	□	⇨	π
	0	0	0	?
	0	0	1/2	⇨
	1/4	1/2	1	⇨
	1/2	1	1	□ ⇨

Environment

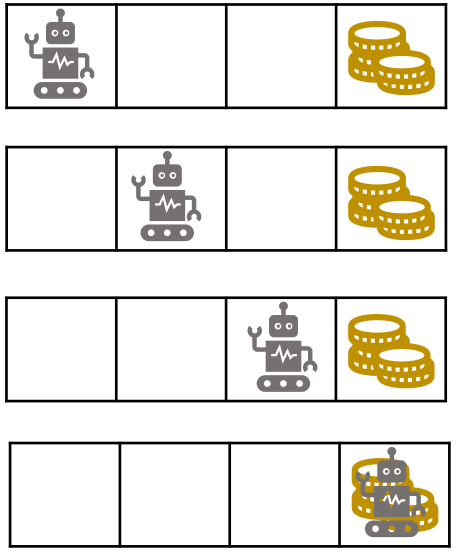


$\gamma = 1/2$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

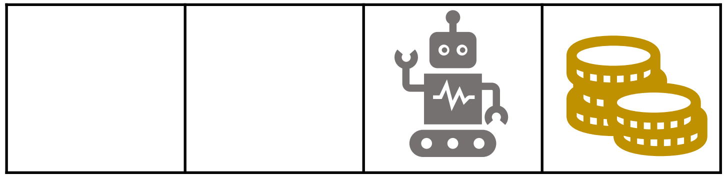
Iteration k:



State

Q	⇦	□	⇨	π
	1/8	1/8	1/4	?
	1/8	1/4	1/2	⇨
	1/4	1/2	1	⇨
	1/2	1	1	□ ⇨

Environment



$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

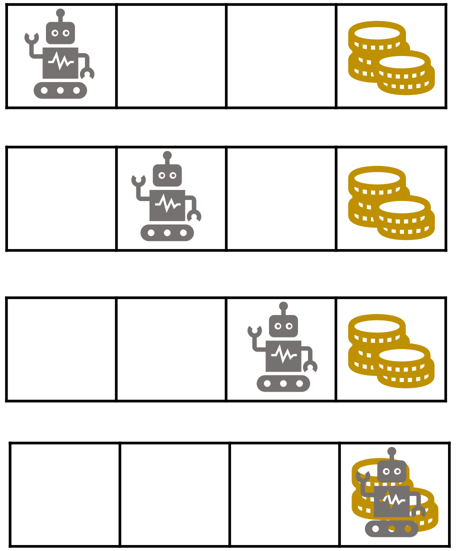
Iteration k:

Action

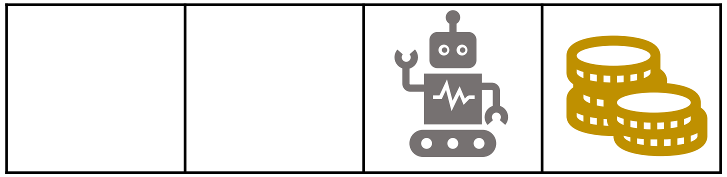
Q

←	□	→	π
1/8	1/8	1/4	→
1/8	1/4	1/2	→
1/4	1/2	1	→
1/2	1	1	□ →

State



Environment



$$\gamma = 1/2$$

Tabular Q-Learning

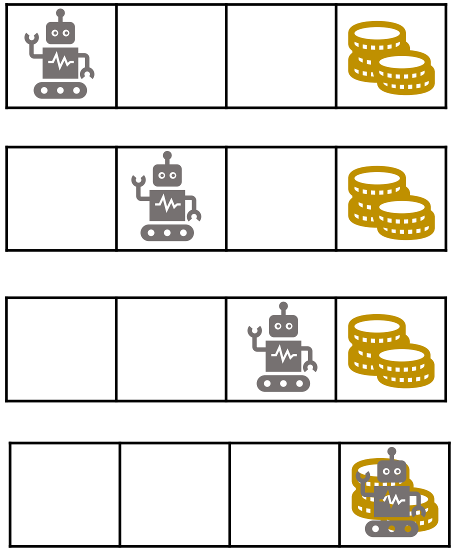
$$Q^{k+1}(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

Iteration k:

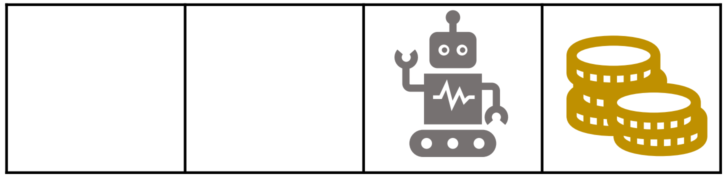
Action

Environment

State



Q^*	⇦	□	⇨	π^*
	1/8	1/8	1/4	⇨
	1/8	1/4	1/2	⇨
	1/4	1/2	1	⇨
	1/2	1	1	□ ⇨



$$\gamma = 1/2$$

Tabular Q-Learning

$$Q^{k+1}(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \max_{\mathbf{a}'} \Big(Q^k(\mathbf{s}', \mathbf{a}') \Big) \right]$$

In tabular setting:

- Every iteration leads to a better Q-function + policy
- Converges to optimal Q-function + policy

Limitations:

- Can only be applied to discrete states and actions
- Need to enumerate over all states and actions every iteration

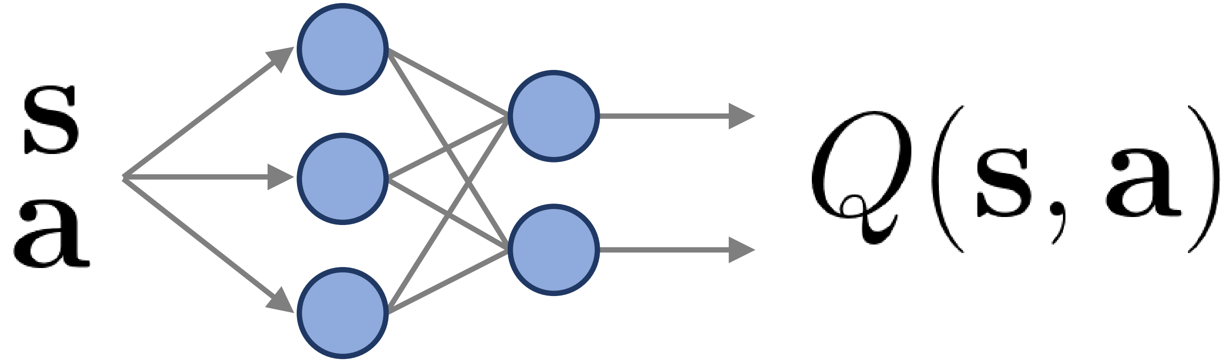
Large State Spaces

Observation:

- 64 x 64 image
- 8 bits per pixel
- $2^{8\times64\times64}$ different states!

Large/Continuous State Spaces

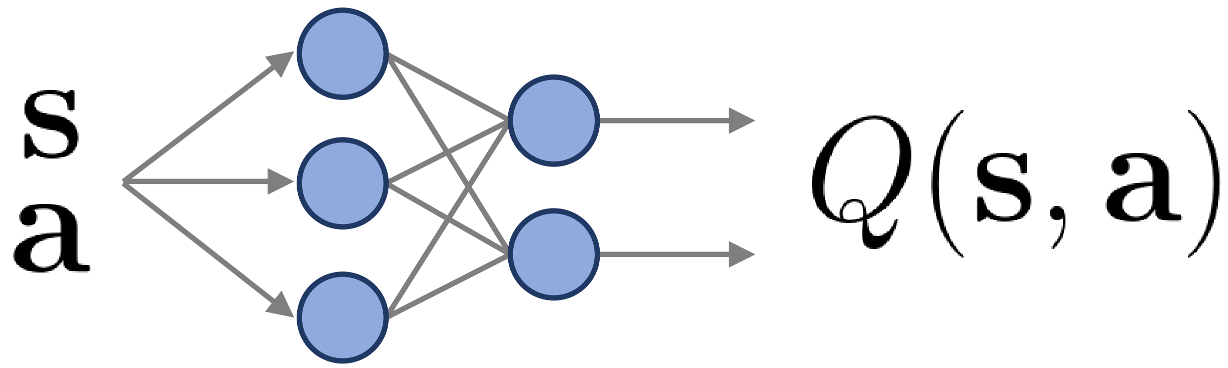
$$\underline{Q^{k+1}(\mathbf{s}, \mathbf{a})} = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \max_{\mathbf{a}'} \left(Q^k(\mathbf{s}', \mathbf{a}') \right) \right]$$



Large/Continuous State Spaces

$$\underline{Q^{k+1}(\mathbf{s},\mathbf{a})} = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'}\left(Q^k(\mathbf{s}',\mathbf{a}')\right)\right]$$

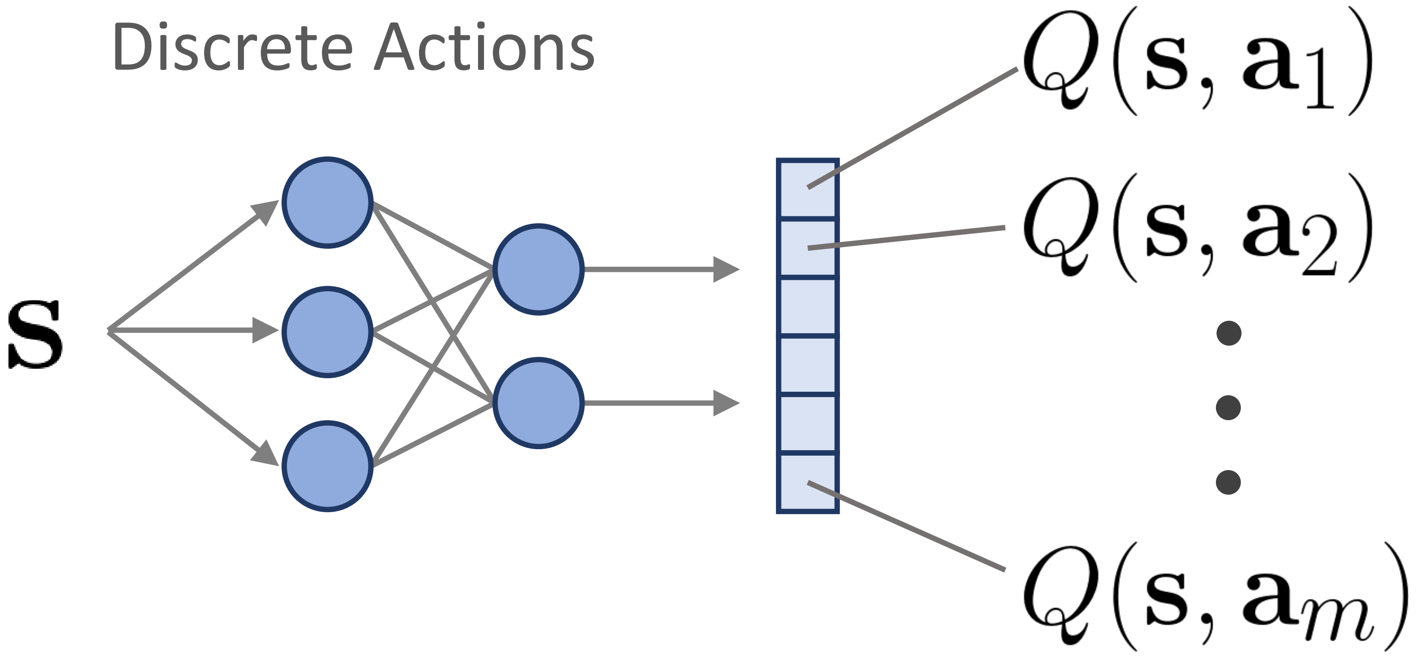
Discrete Actions



Large/Continuous State Spaces

$$Q^{k+1}(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \max_{\mathbf{a}'} \left(Q^k(\mathbf{s}', \mathbf{a}') \right) \right]$$

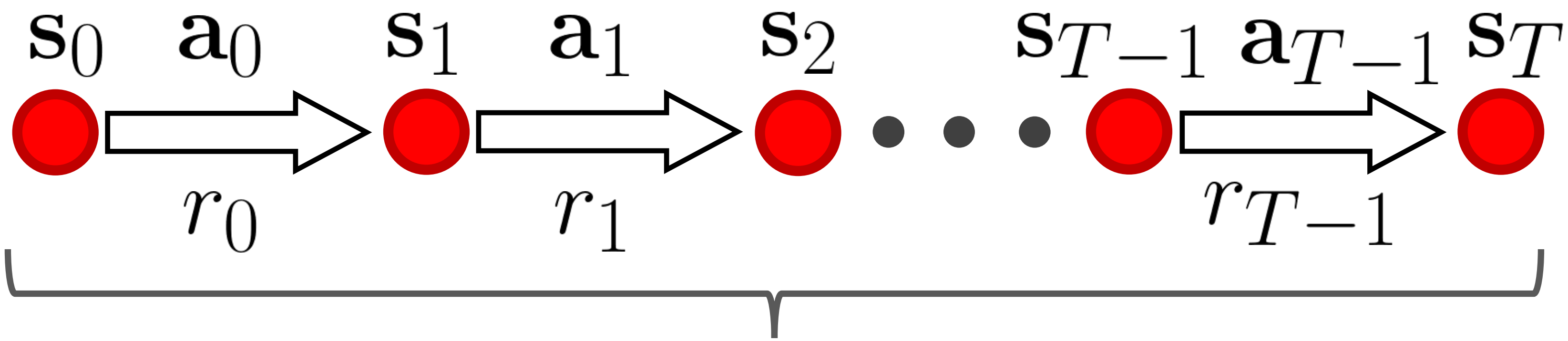
Discrete Actions



Large/Continuous State Spaces

$$Q^{k+1}(\underline{\mathbf{s}, \mathbf{a}}) = \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \max_{\mathbf{a}'} \left(Q^k(\mathbf{s}', \mathbf{a}') \right) \right]$$

$$Q^k(\mathbf{s}, \mathbf{a}) \Longrightarrow \pi^k(\mathbf{a}|\mathbf{s})$$



$$\mathcal{D} = \{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i)\}$$

Large/Continuous State Spaces

$$Q^{k+1}(\mathbf{s}, \mathbf{a}) = \underline{\mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma \max_{\mathbf{a}'} \left(Q^k(\mathbf{s}', \mathbf{a}') \right) \right]}$$

$$\mathcal{D} = \{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i)\}$$

Compute target values for each sample i

$$y_i = r_i + \gamma \max_{\mathbf{a}'} \; Q^k(\mathbf{s}'_i, \mathbf{a}')$$

Fit new Q-function

$$Q^{k+1} = \arg\min_{Q} \; \underbrace{\mathbb{E}_{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i) \sim \mathcal{D}} \left[(y_i - Q(\mathbf{s}_i, \mathbf{a}_i))^2 \right]}_{\text{“Bellman error”}}$$

Q-Learning

ALGORITHM: Q-Learning

1: $Q^0 \leftarrow$ initialize Q-function
2: $\mathcal{D} \leftarrow \{\emptyset\}$ initialize dataset

3: **for** iteration $k = 0, ..., n-1$ **do**
4: Sample trajectory τ according to $Q^k(\mathbf{s}, \mathbf{a})$
5: Add transitions to dataset $\mathcal{D} = \mathcal{D} \cup \{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i)\}$

6: Calculate target values for each sample i:
$y_i = r_i + \gamma \max_{\mathbf{a}'} Q^k(\mathbf{s}'_i, \mathbf{a}')$

7: Update Q-function:
$Q^{k+1} = \arg\min_Q \mathbb{E}_{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{r}_i, \mathbf{s}'_i) \sim \mathcal{D}} \left[(y_i - Q(\mathbf{s}_i, \mathbf{a}_i))^2 \right]$
8: **end for**

9: return Q^n

Q-Learning

ALGORITHM: Q-Learning

$Q^0 \leftarrow$ initialize Q-function
$\mathcal{D} \leftarrow \{\emptyset\}$ initialize dataset

for iteration $k = 0, ..., n-1$ **do**
  Sample trajectory τ according to $Q^k(\mathbf{s}, \mathbf{a})$
  Add transitions to dataset $\mathcal{D} = \mathcal{D} \cup \{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}_i')\}$

  Calculate target values for each sample i:
  $y_i = r_i + \gamma \max_{\mathbf{a}'} Q^k(\mathbf{s}_i', \mathbf{a}')$

  Update Q-function:
  $Q^{k+1} = \arg\min_Q \; \mathbb{E}_{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{r}_i, \mathbf{s}_i') \sim \mathcal{D}} \left[(y_i - Q(\mathbf{s}_i, \mathbf{a}_i))^2 \right]$
end for

return Q^n

Q-Learning

ALGORITHM: Q-Learning

$Q^0 \leftarrow$ initialize Q-function
$\mathcal{D} \leftarrow \{\emptyset\}$ initialize dataset

for iteration $k = 0, ..., n-1$ **do**
  Sample trajectory τ according to $Q^k(\mathbf{s}, \mathbf{a})$
  Add transitions to dataset $\mathcal{D} = \mathcal{D} \cup \{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}_i')\}$

  Calculate target values for each sample i:
  $y_i = r_i + \gamma \max_{\mathbf{a}'} Q^k(\mathbf{s}_i', \mathbf{a}')$

  Update Q-function:
  $Q^{k+1} = \arg\min_Q \; \mathbb{E}_{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{r}_i, \mathbf{s}_i') \sim \mathcal{D}} \left[(y_i - Q(\mathbf{s}_i, \mathbf{a}_i))^2 \right]$
end for

return Q^n

Q-Learning

ALGORITHM: Q-Learning

$Q^0 \leftarrow$ initialize Q-function
$\mathcal{D} \leftarrow \{\emptyset\}$ initialize dataset

for iteration $k = 0, ..., n-1$ **do**
    Sample trajectory τ according to $Q^k(\mathbf{s}, \mathbf{a})$
    Add transitions to dataset $\mathcal{D} = \mathcal{D} \cup \{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i)\}$

    Calculate target values for each sample i:
    $y_i = r_i + \gamma \max_{\mathbf{a}'} Q^k(\mathbf{s}'_i, \mathbf{a}')$

    Update Q-function:
    $Q^{k+1} = \arg\min_Q \; \mathbb{E}_{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{r}_i, \mathbf{s}'_i) \sim \mathcal{D}} \left[(y_i - Q(\mathbf{s}_i, \mathbf{a}_i))^2 \right]$
end for

return Q^n

Q-Learning

ALGORITHM: Q-Learning

$Q^0 \leftarrow$ initialize Q-function
$\mathcal{D} \leftarrow \{\emptyset\}$ initialize dataset

for iteration $k = 0, ..., n-1$ **do**
    Sample trajectory τ according to $Q^k(\mathbf{s}, \mathbf{a})$
    Add transitions to dataset $\mathcal{D} = \mathcal{D} \cup \{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i)\}$

    Calculate target values for each sample i:
    $y_i = r_i + \gamma \max_{\mathbf{a}'} Q^k(\mathbf{s}'_i, \mathbf{a}')$

    Update Q-function:
    $Q^{k+1} = \arg\min_Q \ \mathbb{E}_{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{r}_i, \mathbf{s}'_i) \sim \mathcal{D}} \left[(y_i - Q(\mathbf{s}_i, \mathbf{a}_i))^2 \right]$
end for

return Q^n

Q-Learning

ALGORITHM: Q-Learning

1: $Q^0 \leftarrow$ initialize Q-function
2: $\mathcal{D} \leftarrow \{\emptyset\}$ initialize dataset

3: **for** iteration $k = 0, ..., n-1$ **do**
4: Sample trajectory τ according to $Q^k(\mathbf{s}, \mathbf{a})$
5: Add transitions to dataset $\mathcal{D} = \mathcal{D} \cup \{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}_i')\}$

6: Calculate target values for each sample i:
$y_i = r_i + \gamma \max_{\mathbf{a}'} Q^k(\mathbf{s}_i', \mathbf{a}')$

7: Update Q-function:
$Q^{k+1} = \arg\min_Q \mathbb{E}_{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{r}_i, \mathbf{s}_i') \sim \mathcal{D}} \left[(y_i - Q(\mathbf{s}_i, \mathbf{a}_i))^2 \right]$
8: **end for**

9: return Q^n

Q-Learning

ALGORITHM: Q-Learning

$Q^0 \leftarrow$ initialize Q-function
$\mathcal{D} \leftarrow \{\emptyset\}$ initialize dataset

for iteration $k = 0, ..., n-1$ **do**
    Sample trajectory τ according to $Q^k(\mathbf{s}, \mathbf{a})$
    Add transitions to dataset $\mathcal{D} = \mathcal{D} \cup \{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i)\}$

    Calculate target values for each sample i:
    $y_i = r_i + \gamma \max_{\mathbf{a}'} Q^k(\mathbf{s}'_i, \mathbf{a}')$

    Update Q-function:
    $Q^{k+1} = \arg\min_Q \ \mathbb{E}_{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{r}_i, \mathbf{s}'_i) \sim \mathcal{D}} \left[(y_i - Q(\mathbf{s}_i, \mathbf{a}_i))^2 \right]$
end for

return Q^n

Q-Learning

ALGORITHM: Q-Learning

1: $Q^0 \leftarrow$ initialize Q-function
2: $\mathcal{D} \leftarrow \{\emptyset\}$ initialize dataset

3: **for** iteration $k = 0, ..., n-1$ **do**
4: Sample trajectory τ according to $Q^k(\mathbf{s}, \mathbf{a})$
5: Add transitions to dataset $\mathcal{D} = \mathcal{D} \cup \{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i)\}$

6: Calculate target values for each sample i:
$y_i = r_i + \gamma \max_{\mathbf{a}'} Q^k(\mathbf{s}'_i, \mathbf{a}')$

7: Update Q-function:
$Q^{k+1} = \arg\min_Q \mathbb{E}_{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{r}_i, \mathbf{s}'_i) \sim \mathcal{D}} \left[(y_i - Q(\mathbf{s}_i, \mathbf{a}_i))^2 \right]$
8: **end for**

9: return Q^n

Q-Learning

ALGORITHM: Q-Learning

1: $Q^0 \leftarrow$ initialize Q-function
2: $\mathcal{D} \leftarrow \{\emptyset\}$ initialize dataset

3: **for** iteration $k = 0, ..., n-1$ **do**
4: Sample trajectory τ according to $Q^k(\mathbf{s}, \mathbf{a})$
5: Add transitions to dataset $\mathcal{D} = \mathcal{D} \cup \{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}_i')\}$

6: Calculate target values for each sample i:
$y_i = r_i + \gamma \max_{\mathbf{a}'} Q^k(\mathbf{s}_i', \mathbf{a}')$

7: Update Q-function:
$Q^{k+1} = \arg\min_Q \mathbb{E}_{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{r}_i, \mathbf{s}_i') \sim \mathcal{D}} \left[(y_i - Q(\mathbf{s}_i, \mathbf{a}_i))^2 \right]$
8: **end for**

9: return Q^n

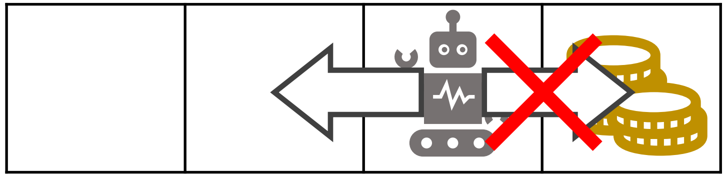
How to sample trajectories?

Sampling

$$Q^k(\mathbf{s}, \mathbf{a}) \Longrightarrow \pi^k(\mathbf{a}|\mathbf{s})$$

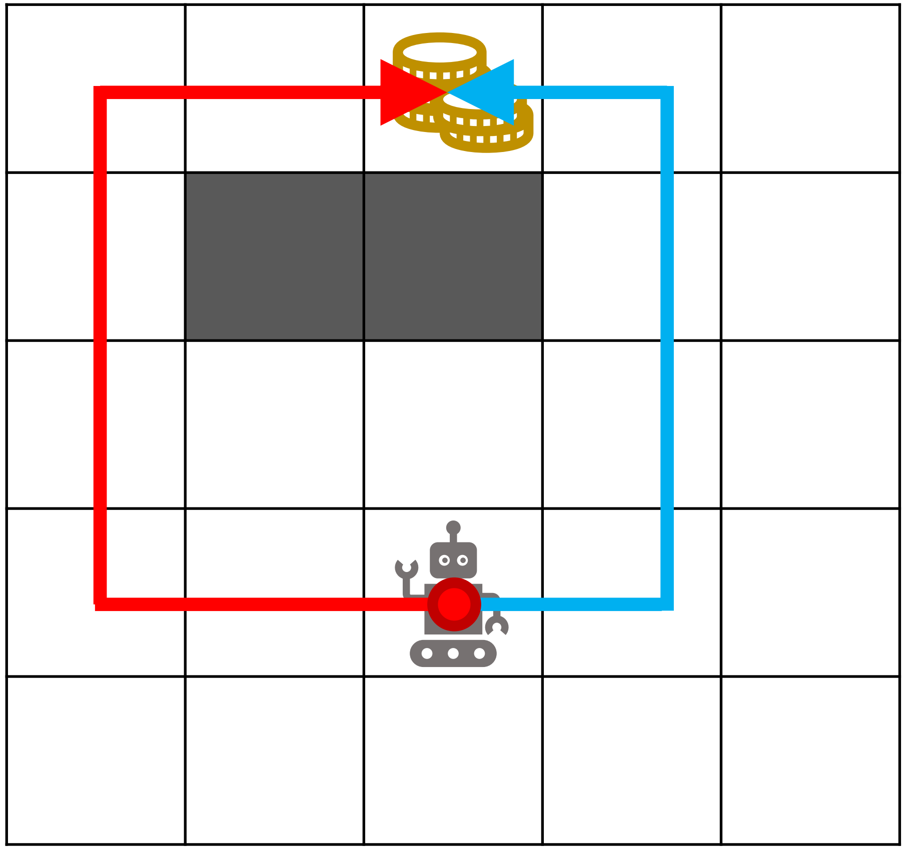
$$\pi^k(\mathbf{a}|\mathbf{s}) = \begin{cases} 1 & \text{if } \mathbf{a} = \arg\max_{\mathbf{a}'} Q^k(\mathbf{s}, \mathbf{a}') \\ 0 & \text{otherwise} \end{cases}$$

✘ Insufficient exploration



Exploration-Exploitation

Need to try new actions in case they are better



Exploration-Exploitation

Need to try new actions in case they are better

Keep going to the same restaurant

Try new restaurant

Exploration-Exploitation

Need to try new actions in case they are better

$$\pi^k(\mathbf{a}|\mathbf{s}) = \begin{cases} 1 & \text{if } \mathbf{a} = \arg\max_{\mathbf{a}'} \; Q^k(\mathbf{s}, \mathbf{a}') \\ 0 & \text{otherwise} \end{cases}$$

Exploration-Exploitation

Need to try new actions in case they are better

$$\pi^k(\mathbf{a}|\mathbf{s}) = \begin{cases} 1 - \boxed{\epsilon} & \text{if } \mathbf{a} = \arg\max_{\mathbf{a}'} Q^k(\mathbf{s}, \mathbf{a}') \\ \boxed{\epsilon} & \text{otherwise} \end{cases}$$

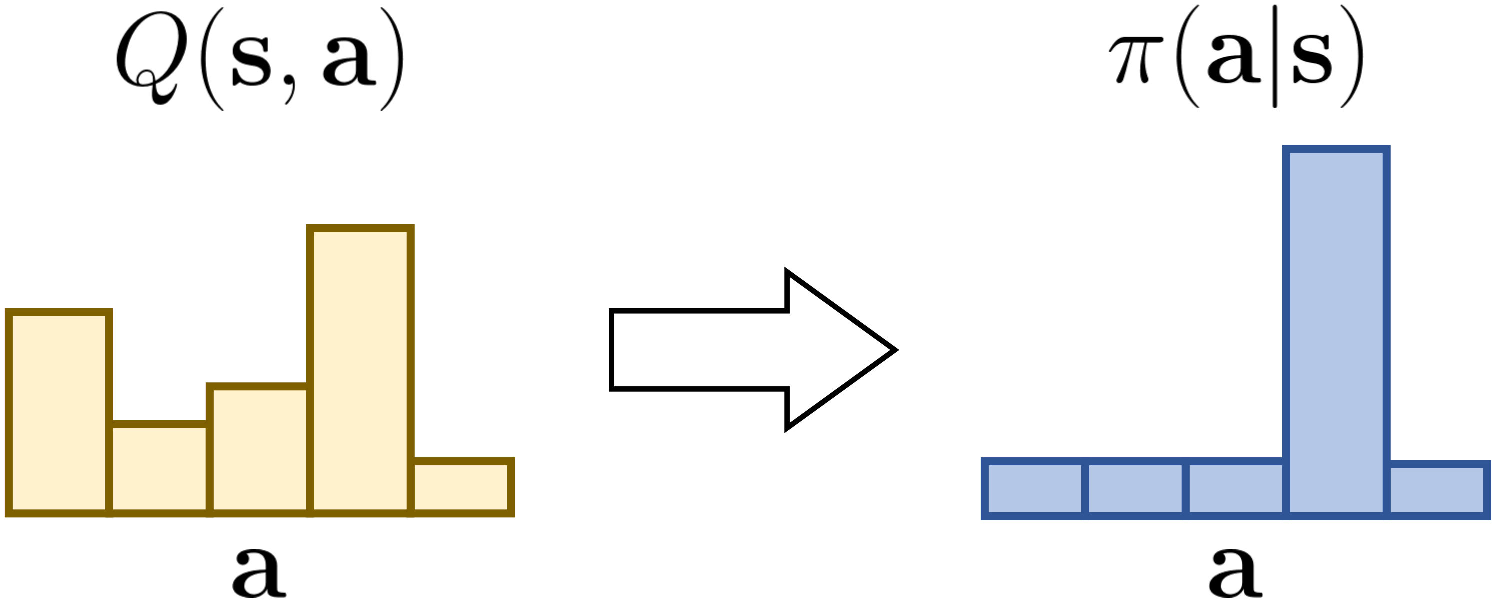
Epsilon-greedy exploration

- With probability $1 - \epsilon$ exploit current best action
- With probability ϵ explore new action by sampling a random action
- Start with $\epsilon = 1$ and then anneal to lower value (e.g. $\epsilon \to 0.1$)

Epsilon-Greedy Exploration

$$\pi^k(\mathbf{a}|\mathbf{s}) = \begin{cases} 1-\epsilon & \text{if } \mathbf{a} = \arg\max_{\mathbf{a}'} \; Q^k(\mathbf{s}, \mathbf{a}') \\ \epsilon & \text{otherwise} \end{cases}$$

$\epsilon \to 1$

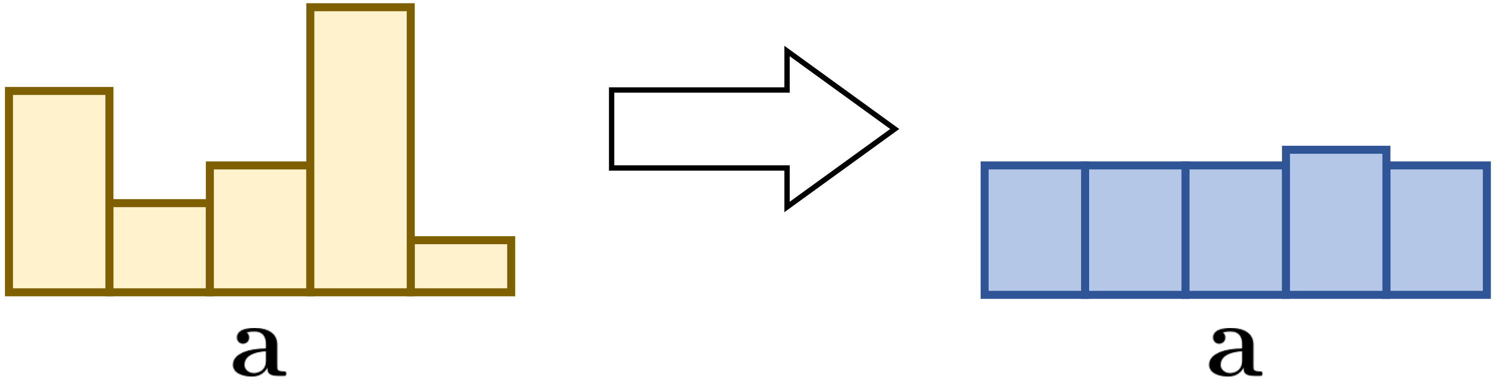


Epsilon-Greedy Exploration

$$\pi^k(\mathbf{a}|\mathbf{s}) = \begin{cases} 1-\epsilon & \text{if } \mathbf{a} = \arg\max_{\mathbf{a}'} Q^k(\mathbf{s},\mathbf{a}') \\ \epsilon & \text{otherwise} \end{cases}$$

$\epsilon \to 1$

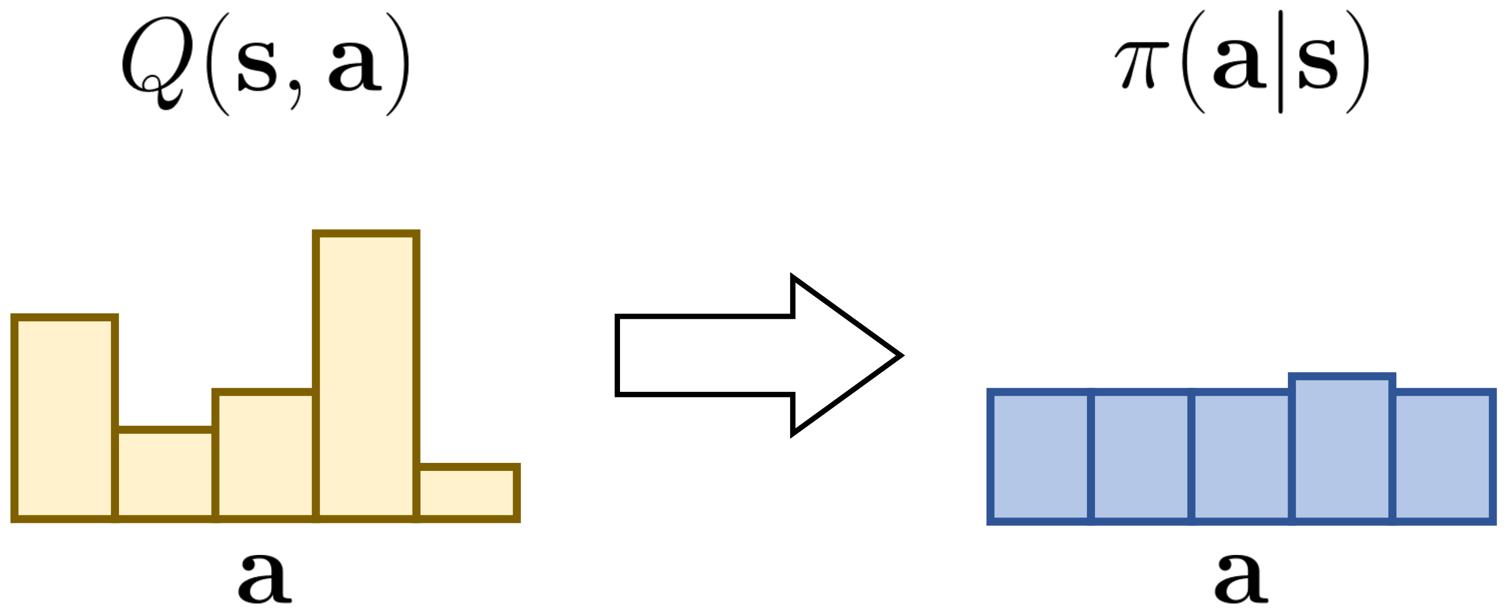
$Q(\mathbf{s},\mathbf{a})$ $\pi(\mathbf{a}|\mathbf{s})$



Epsilon-Greedy Exploration

$$\pi^k(\mathbf{a}|\mathbf{s}) = \begin{cases} 1-\epsilon & \text{if } \mathbf{a} = \arg\max_{\mathbf{a}'} \; Q^k(\mathbf{s}, \mathbf{a}') \\ \epsilon & \text{otherwise} \end{cases}$$

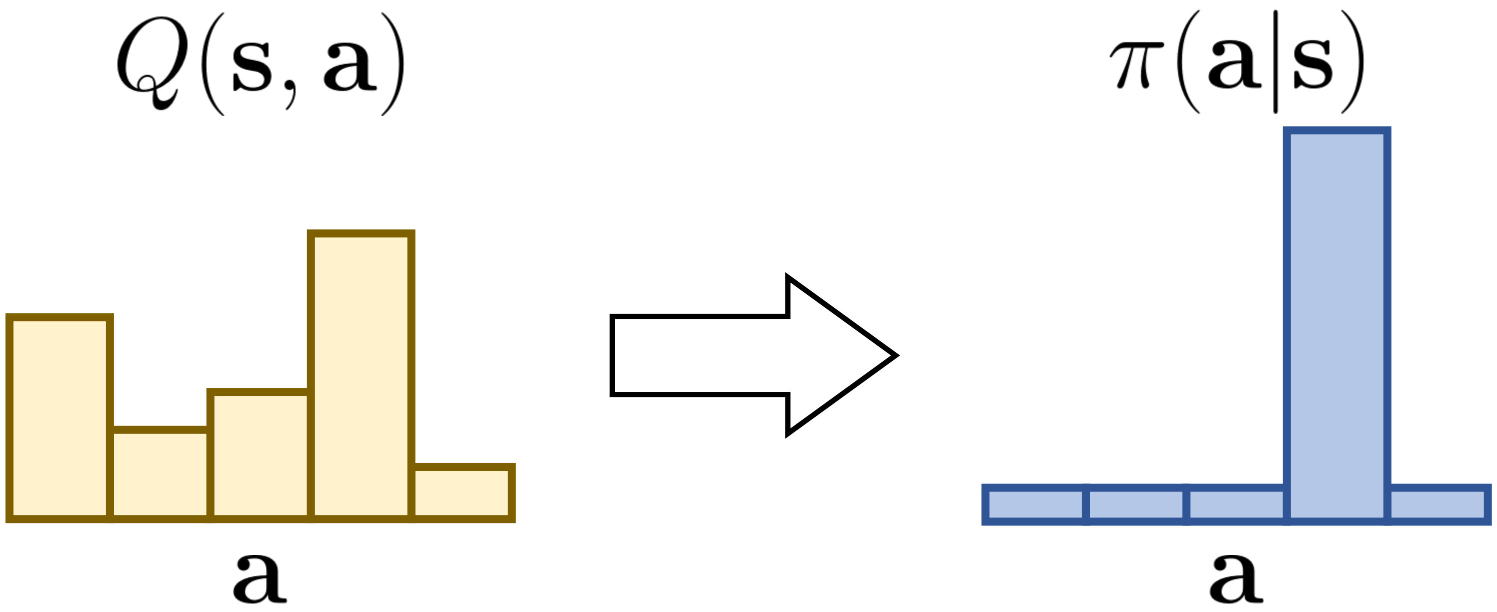
$\epsilon \to 0$



Epsilon-Greedy Exploration

$$\pi^k(\mathbf{a}|\mathbf{s}) = \begin{cases} 1-\epsilon & \text{if } \mathbf{a} = \arg\max_{\mathbf{a}'} Q^k(\mathbf{s}, \mathbf{a}') \\ \epsilon & \text{otherwise} \end{cases}$$

$$\epsilon \to 0$$



Boltzmann Exploration

Probability of an action is proportion to its “goodness”

$$\pi^k(\mathbf{a}|\mathbf{s}) = \frac{1}{Z}\exp\left(\frac{1}{\beta}Q^k(\mathbf{s},\mathbf{a})\right)$$

where,

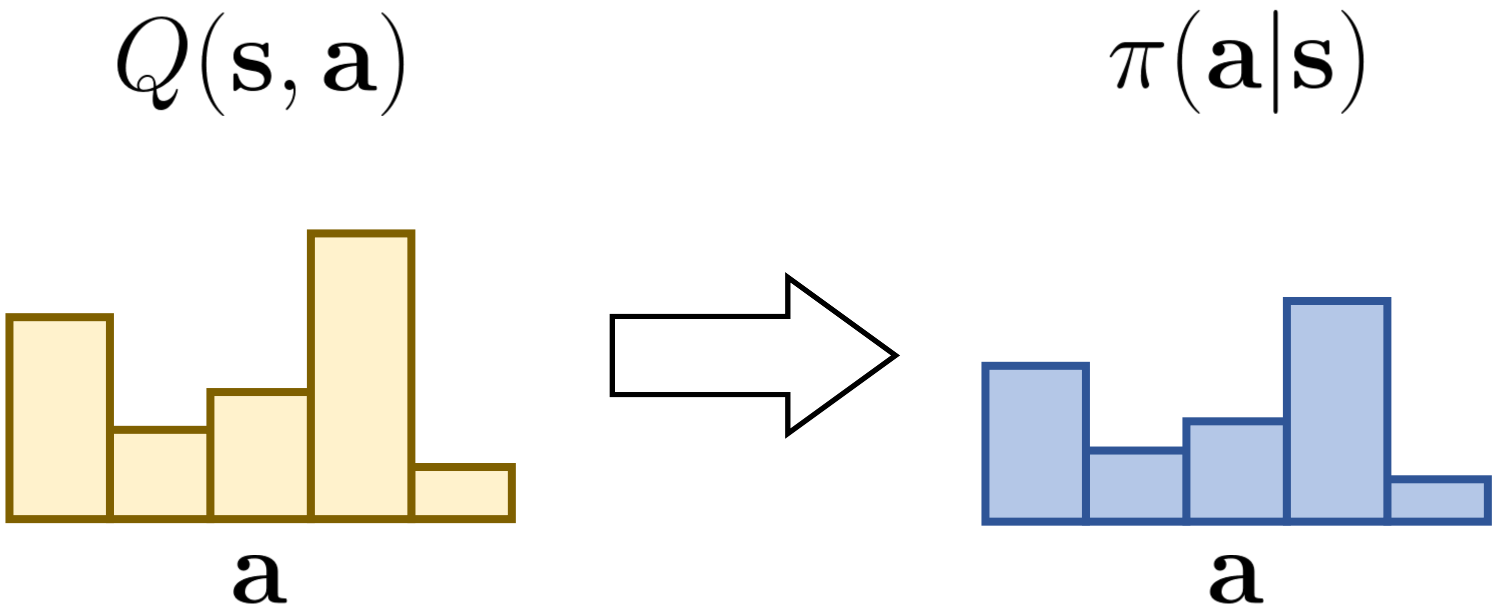
temperature parameters: $\beta \in \mathbb{R}$

normalization factor: $Z = \sum_{\mathbf{a}'} \exp\left(\frac{1}{\beta}Q^k(\mathbf{s},\mathbf{a}')\right)$

Boltzmann Exploration

$$\pi^k(\mathbf{a}|\mathbf{s}) = \frac{1}{Z}\exp\left(\frac{1}{\beta}Q^k(\mathbf{s},\mathbf{a})\right)$$

$$\beta \rightarrow \infty$$

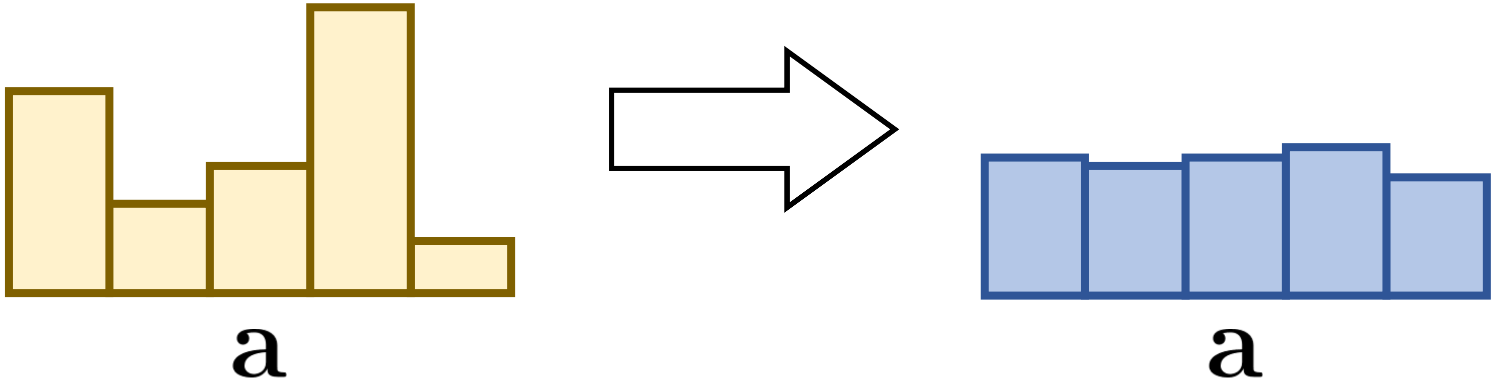


Boltzmann Exploration

$$\pi^k(\mathbf{a}|\mathbf{s}) = \frac{1}{Z}\exp\left(\frac{1}{\beta}Q^k(\mathbf{s},\mathbf{a})\right)$$

$\beta \to \infty$

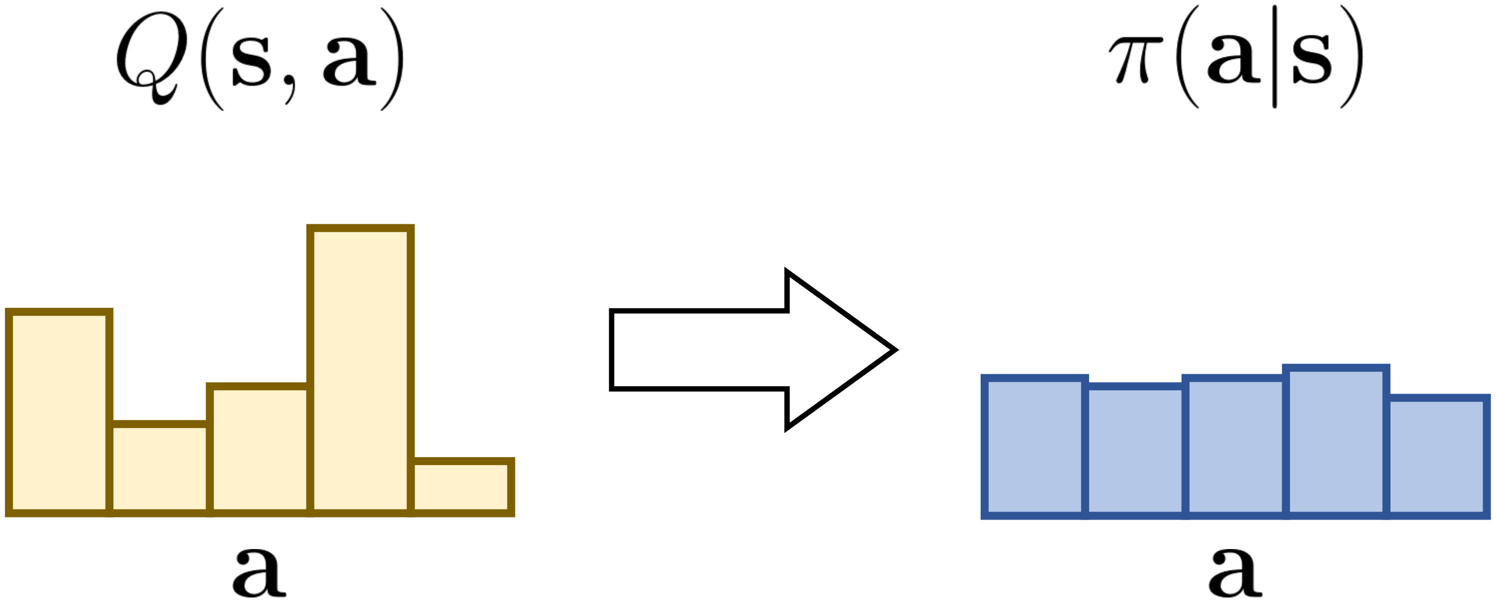
$Q(\mathbf{s},\mathbf{a})$ $\qquad\qquad$ $\pi(\mathbf{a}|\mathbf{s})$



Boltzmann Exploration

$$\pi^k(\mathbf{a}|\mathbf{s}) = \frac{1}{Z}\exp\left(\frac{1}{\beta}Q^k(\mathbf{s},\mathbf{a})\right)$$

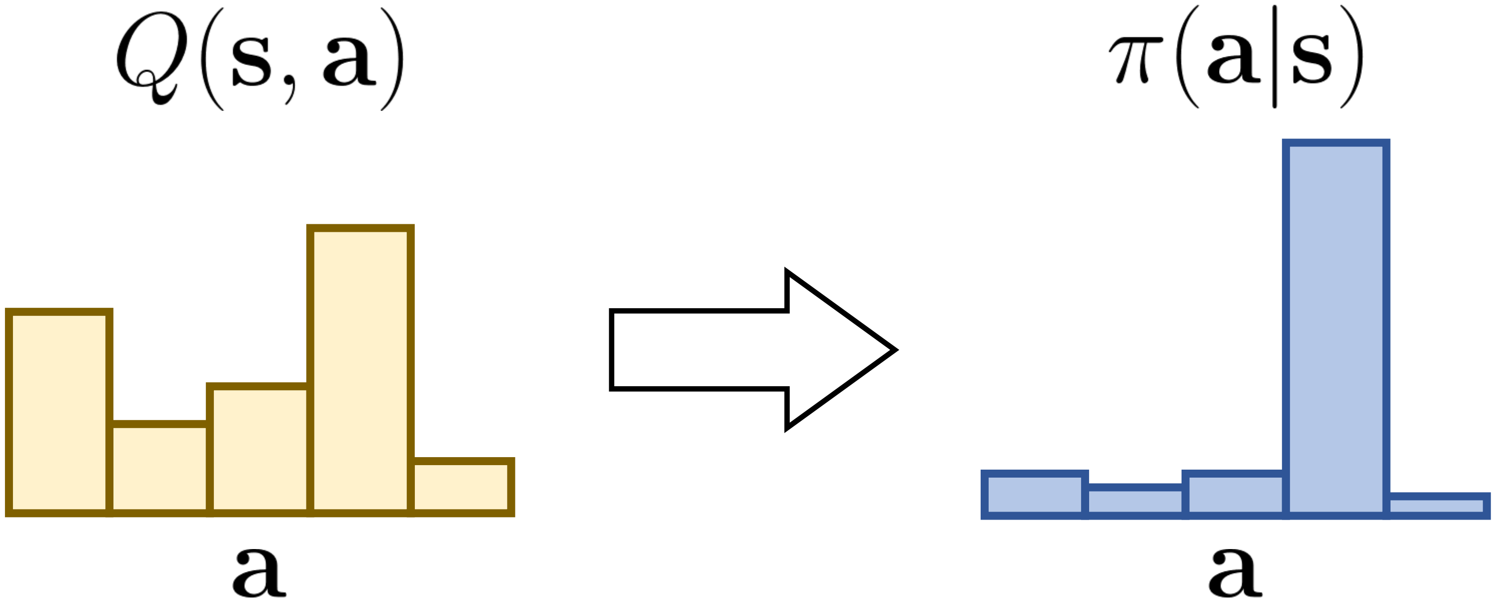
$$\beta \to 0$$



Boltzmann Exploration

$$\pi^k(\mathbf{a}|\mathbf{s}) = \frac{1}{Z}\exp\left(\frac{1}{\beta}Q^k(\mathbf{s},\mathbf{a})\right)$$

$\beta \to 0$



Testing

After training, test with greedy policy

$$\pi^k(\mathbf{a}|\mathbf{s}) = \begin{cases} 1 & \text{if } \mathbf{a} = \arg\max_{\mathbf{a}'} \; Q^k(\mathbf{s}, \mathbf{a}') \\ 0 & \text{otherwise} \end{cases}$$

Q-Learning

ALGORITHM: Q-Learning

1: $Q^0 \leftarrow$ initialize Q-function
2: $\mathcal{D} \leftarrow \{\emptyset\}$ initialize dataset

3: **for** iteration $k = 0, ..., n-1$ **do**
4: Sample trajectory τ according to $Q^k(\mathbf{s}, \mathbf{a})$
5: Add transitions to dataset $\mathcal{D} = \mathcal{D} \cup \{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i)\}$

6: Calculate target values for each sample i:
$y_i = r_i + \gamma \max_{\mathbf{a}'} Q^k(\mathbf{s}'_i, \mathbf{a}')$

7: Update Q-function:
$Q^{k+1} = \arg\min_Q \; \mathbb{E}_{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{r}_i, \mathbf{s}'_i) \sim \mathcal{D}} \left[(y_i - Q(\mathbf{s}_i, \mathbf{a}_i))^2 \right]$
8: **end for**

9: return Q^n

Q-Learning with Function Approximators

- No improvement guarantees

$$Q^{k+1}(\mathbf{s}, \mathbf{a}) \cancel{\geq} Q^k(\mathbf{s}, \mathbf{a}) \qquad J(\pi^{k+1}) \cancel{\geq} J(\pi^k)$$

- No convergence guarantees

$$Q^k \cancel{\Longrightarrow} Q^*$$

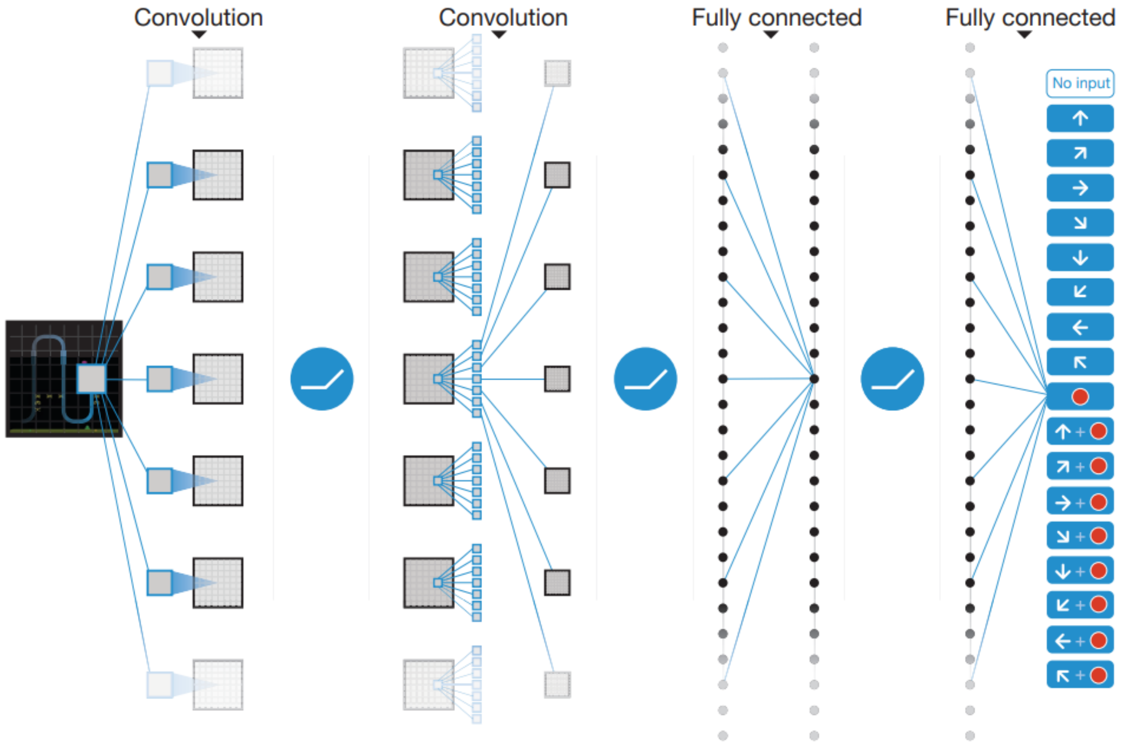
- But in practice, it works!

Deep Q-Networks (DQN)

Human-Level Control Through Deep Reinforcement Learning
[Mnih et al. 2015]

Deep Q-Networks (DQN)

Input: 84 x 84 images



Output: Q-values for joystick controls

Human-Level Control Through Deep Reinforcement Learning
[Mnih et al. 2015]

Deep Q-Networks (DQN)

Human-Level Control Through Deep Reinforcement Learning
[Mnih et al. 2015]

Q-Learning

✓ Often much more sample efficient than policy gradient

✓ Off-policy learning

✗ Limited to relatively small discrete action spaces

✗ Does not directly optimize performance

- Lower Bellman error $\neq$ better performance

✗ No convergence guarantees with function approximators

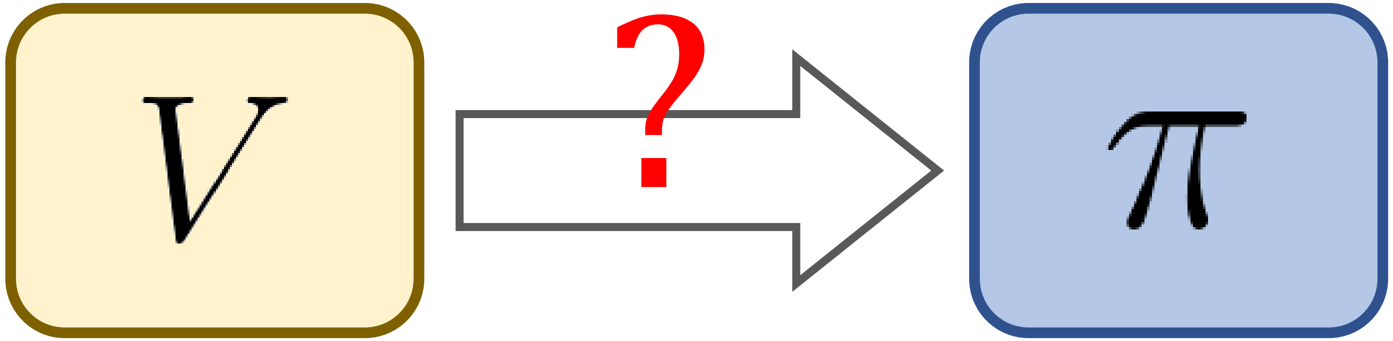
$$Q^{k+1} = \arg\min_{Q} \mathbb{E}_{(\mathbf{s},\mathbf{a},r,\mathbf{s}')\sim\mathcal{D}}\left[\left(\left(r + \gamma \underline{\max_{\mathbf{a}'} Q^k(\mathbf{s}',\mathbf{a}')}\right) - Q(\mathbf{s},\mathbf{a})\right)^2\right]$$

Intractable in large/continuous action spaces

Value Functions

$$\pi(\mathbf{a}|\mathbf{s}) = \begin{cases} 1 & \text{if } \mathbf{a} = \arg\max_{\mathbf{a}'} Q(\mathbf{s}, \mathbf{a}') \\ 0 & \text{otherwise} \end{cases}$$

What about $V(\mathbf{s})$?

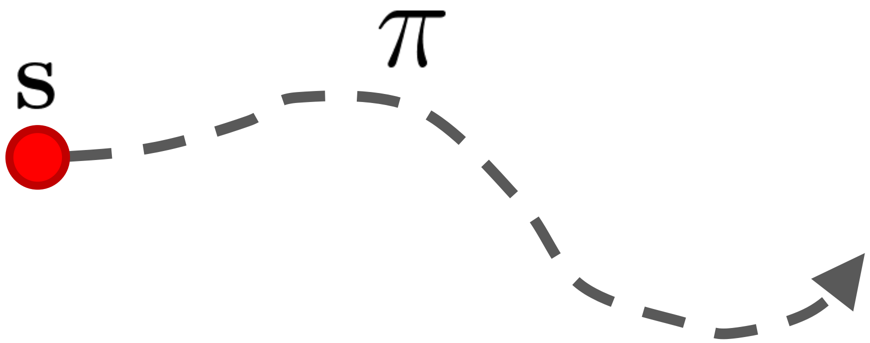


Value Functions

Value Function

"State Value Function"

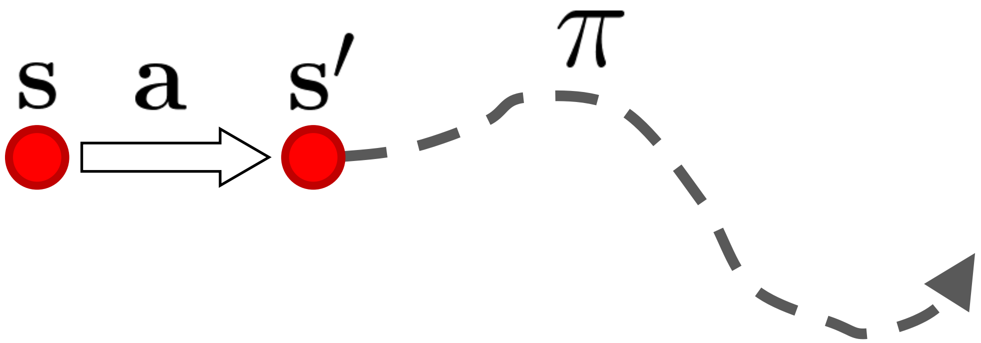
$$V^{\pi}(\mathbf{s}) = \mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s})}\left[\sum_{t=0}\gamma^t r_t\right]$$



Q-Function

"State-Action Value Function"

$$Q^{\pi}(\mathbf{s},\underline{\mathbf{a}}) = \mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\sum_{t=0}\gamma^t r_t\right]$$

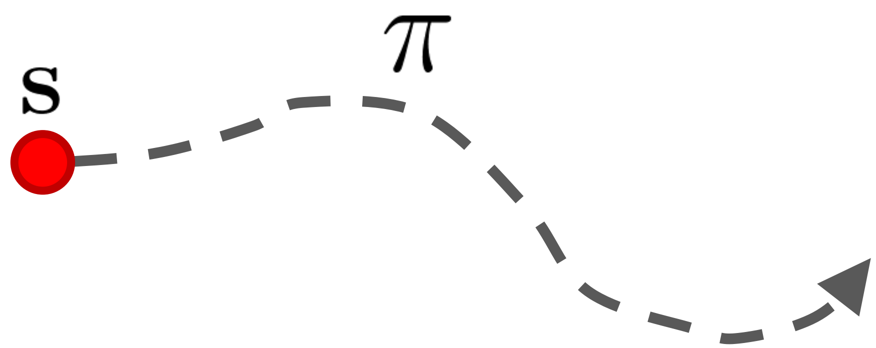


Value Functions

Value Function

"State Value Function"

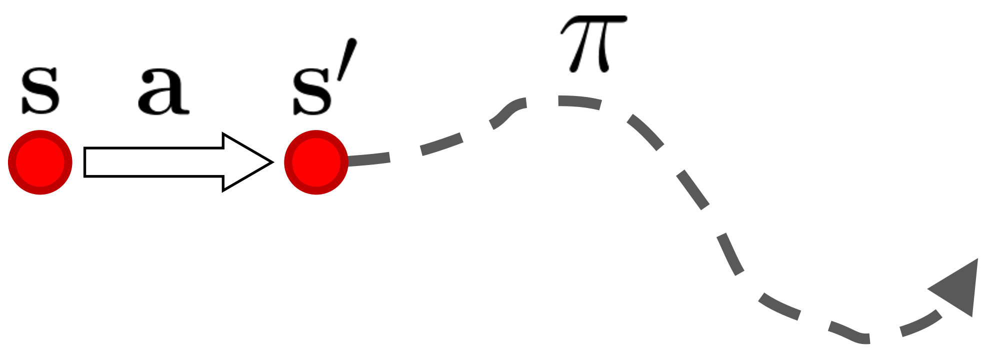
$$V^\pi(\underline{\mathbf{s}}) = \mathbb{E}_{\tau \sim p(\tau|\pi, \mathbf{s}_0 = \mathbf{s})}\left[\sum_{t=0} \gamma^t r_t\right]$$



Q-Function

"State-Action Value Function"

$$Q^\pi(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\tau \sim p(\tau|\pi, \mathbf{s}_0 = \mathbf{s}, \mathbf{a}_0 = \mathbf{a})}\left[\sum_{t=0} \gamma^t r_t\right]$$



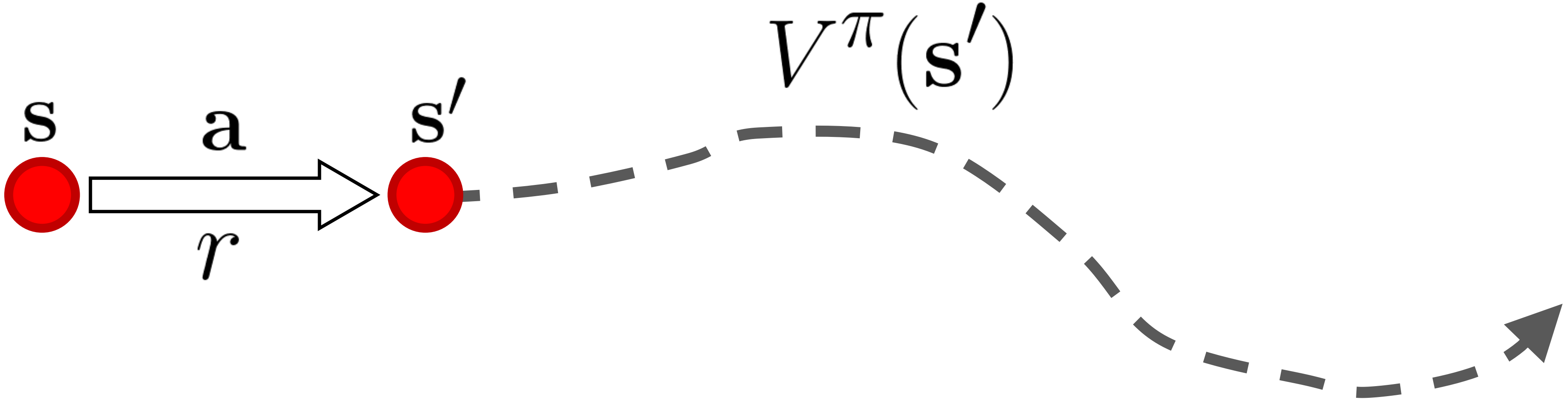
Recursive definition

$$Q^\pi(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \underline{\mathbb{E}_{\mathbf{a}'\sim\pi(\mathbf{a}'|\mathbf{s}')}\left[Q^\pi(\mathbf{s}',\mathbf{a}')\right]}\right]$$

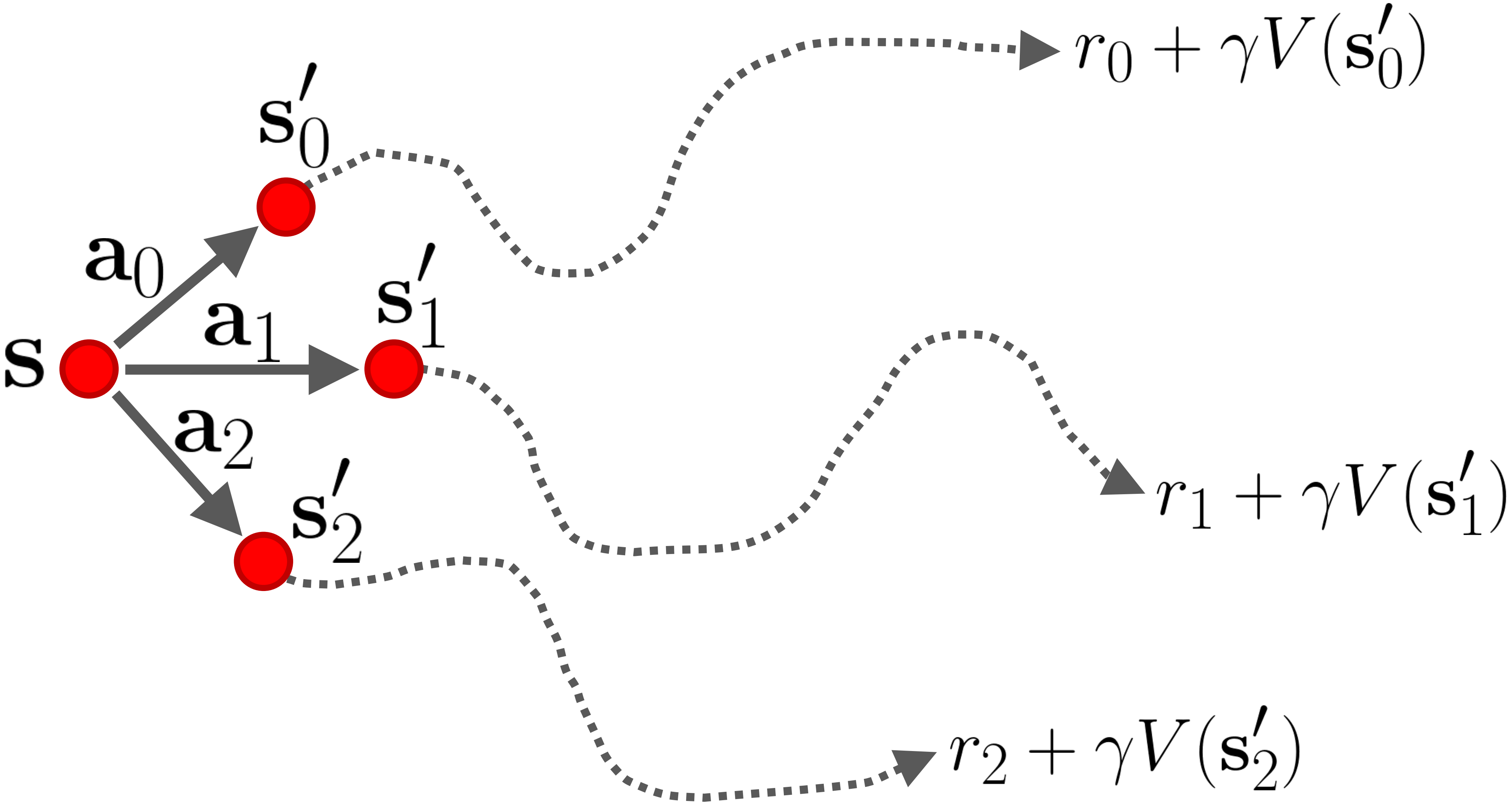
Recursive definition

$$Q^\pi(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \mathbb{E}_{\mathbf{a}'\sim\pi(\mathbf{a}'|\mathbf{s}')}\left[Q^\pi(\mathbf{s}',\mathbf{a}')\right]\right]$$

$$= \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \underline{V^\pi(\mathbf{s}')}\right]$$



Value Function



Value Function

Value-function:

$$\arg\max_{\mathbf{a}} \mathbb{E}_{\underline{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma V(\mathbf{s}') \right]$$

Need access to dynamics

Value Function

Value-function:

$$\arg\max_{\mathbf{a}} \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[r(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma V(\mathbf{s}') \right]$$

Q-function:

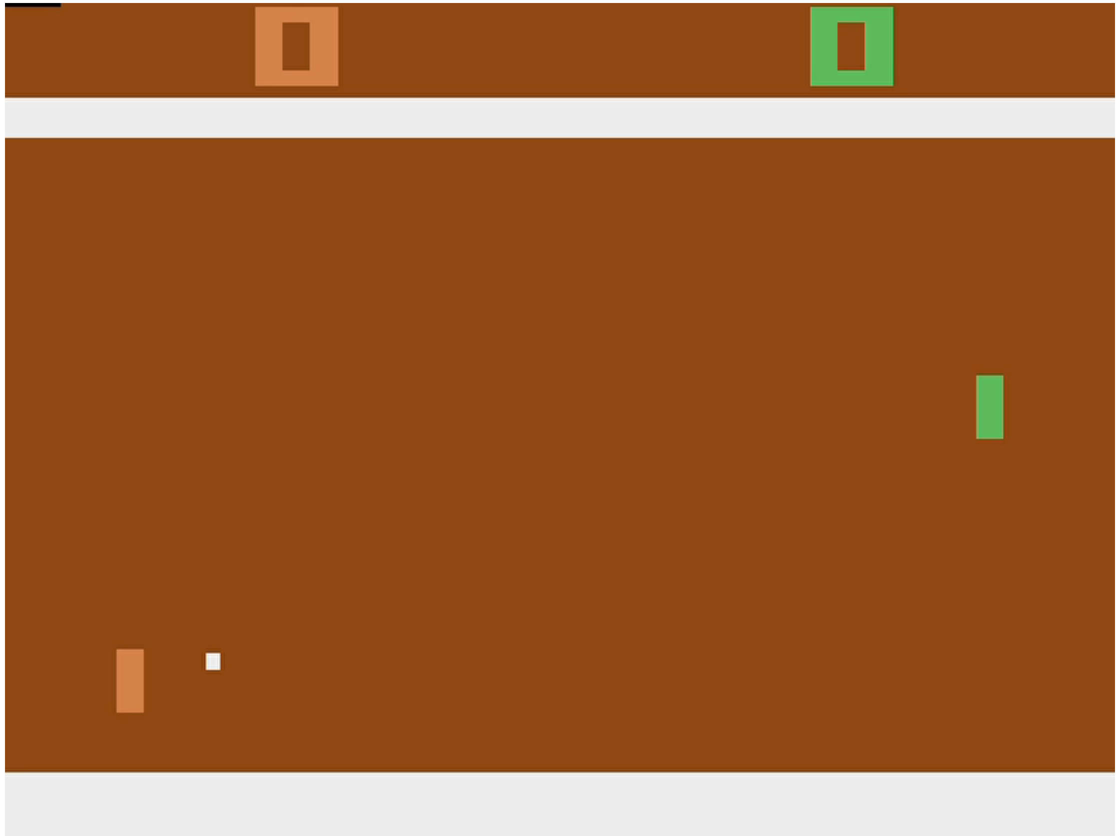
$$\arg\max_{\mathbf{a}} \underline{Q(\mathbf{s}, \mathbf{a})}$$

Do not need dynamics

Summary

- Q-Function
- Q-Learning
- Exploration

Assignment 3: Q-Learning



Pong

Breakout