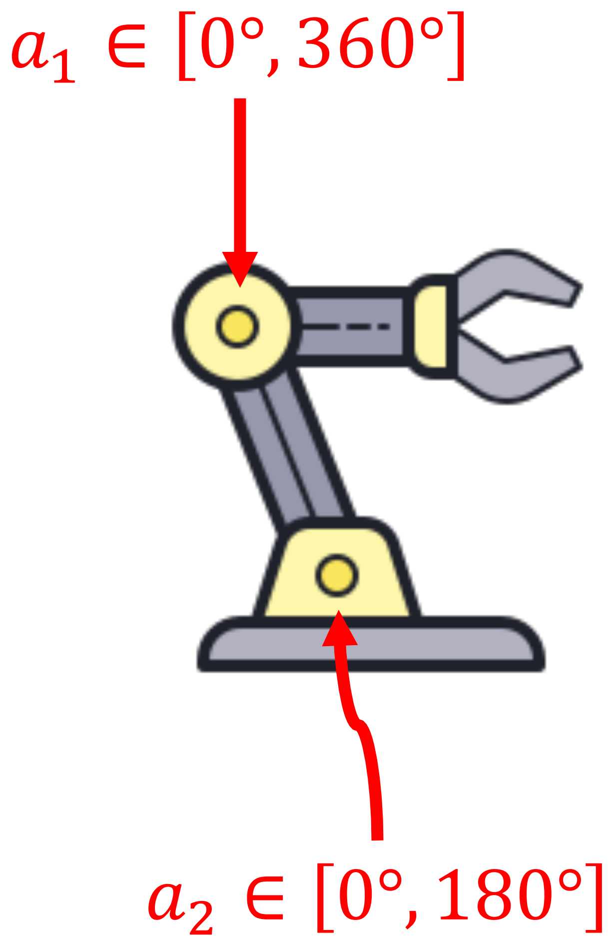
Deterministic Policy Gradient (DPG)

Shusen Wang

http://wangshusen.github.io/

Continuous Action Space



- The action space $\mathcal{A}$ is a subset of $\mathbb{R}^2$.
- The action space $\mathcal{A}$ is continuous:

$$\mathcal{A} = [0°, 360°] \times [0°, 180°].$$

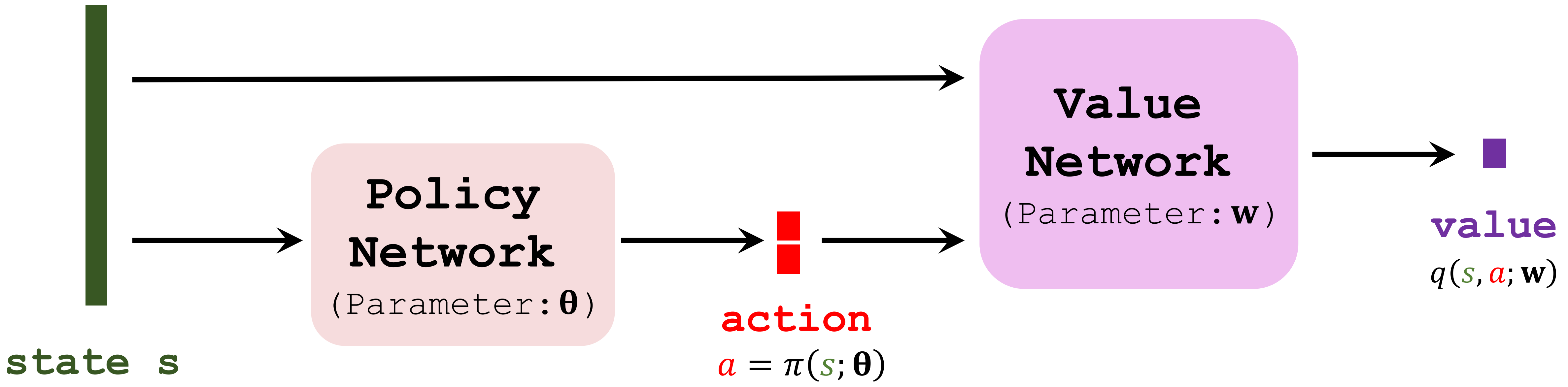
- Actions are 2-dim vectors.

Deterministic Policy Gradient (DPG)

Reference:

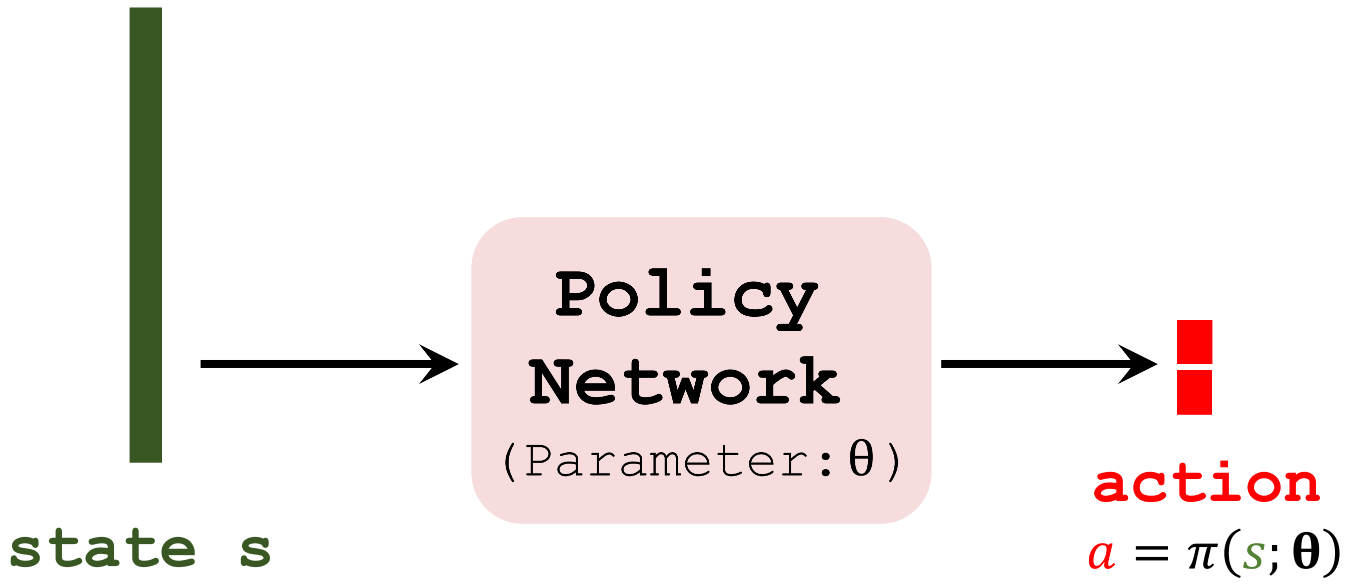
- Silver et al. Deterministic policy gradient algorithms. In *ICML*, 2014.
- Lillicrap et al. Continuous control with deep reinforcement learning. In *ICLR*, 2016.

Deterministic Actor-Critic



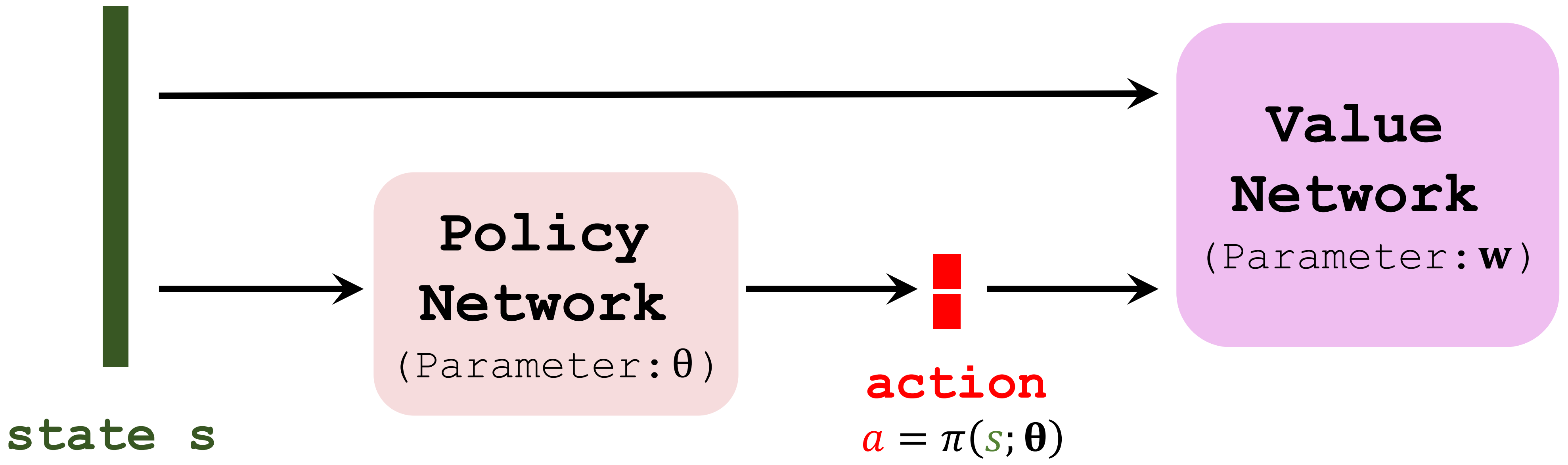
Deterministic Actor-Critic

- Use a deterministic policy network (actor): $a = \pi(s; \boldsymbol{\theta})$.



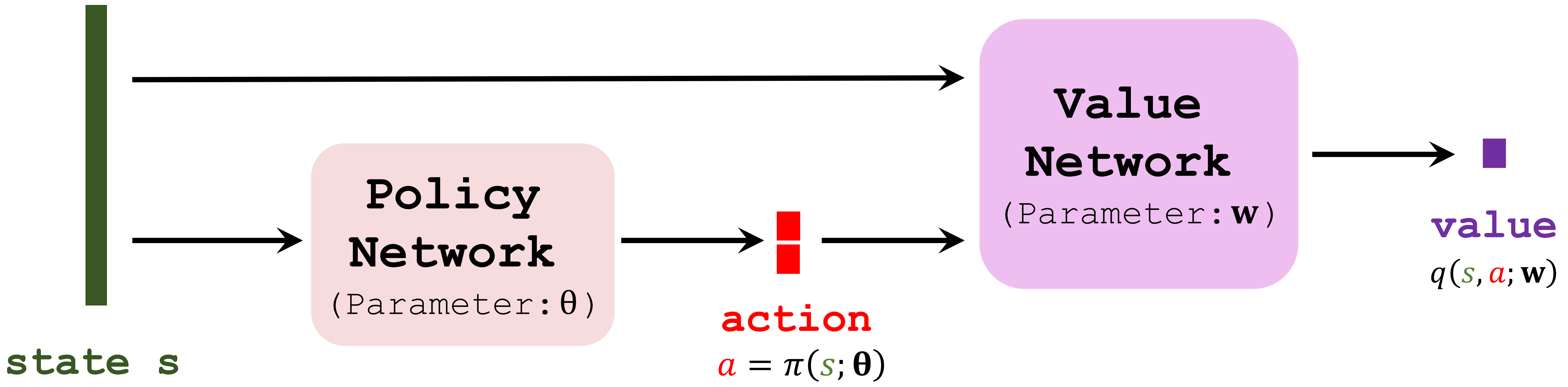
Deterministic Actor-Critic

- Use a deterministic policy network (actor): $a = \pi(s; \boldsymbol{\theta})$.
- Use a value network (critic): $q(s, a; \mathbf{w})$.



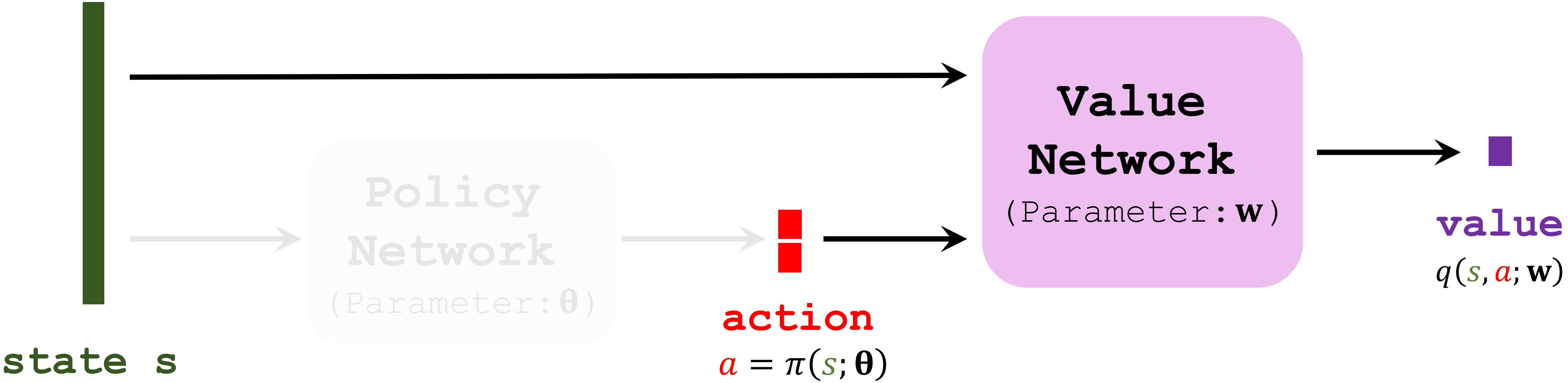
Deterministic Actor-Critic

- Use a deterministic policy network (actor): $a = \pi(s; \boldsymbol{\theta})$.
- Use a value network (critic): $q(s, a; \mathbf{w})$.
- The critic outputs a scalar that evaluates *how good the action* a *is*.



Updating Value Network by TD

- Transition: (s_t, a_t, r_t, s_{t+1}).



Updating Value Network by TD

- Transition: (s_t, a_t, r_t, s_{t+1}).
- Value network makes prediction for time t:

$$q_t = q(s_t, a_t; \mathbf{w}).$$

Updating Value Network by TD

- Transition: (s_t, a_t, r_t, s_{t+1}).
- Value network makes prediction for time t:

$$q_t = q(s_t, a_t; \mathbf{w}).$$

- Value network makes prediction for time $t+1$:

$$q_{t+1} = q(s_{t+1}, a'_{t+1}; \mathbf{w}), \text{ where } a'_{t+1} = \pi(s_{t+1}; \boldsymbol{\theta}).$$

Updating Value Network by TD

- Transition: (s_t, a_t, r_t, s_{t+1}).
- Value network makes prediction for time t:

$$q_t = q(s_t, a_t; \mathbf{w}).$$

- Value network makes prediction for time $t+1$:

$$q_{t+1} = q(s_{t+1}, a'_{t+1}; \mathbf{w}), \text{ where } a'_{t+1} = \pi(s_{t+1}; \boldsymbol{\theta}).$$

- TD error: $\delta_t = q_t - \underbrace{(r_t + \gamma \cdot q_{t+1})}_{\text{TD Target}}.$

Updating Value Network by TD

- Transition: (s_t, a_t, r_t, s_{t+1}).
- Value network makes prediction for time t:

$$q_t = q(s_t, a_t; \mathbf{w}).$$

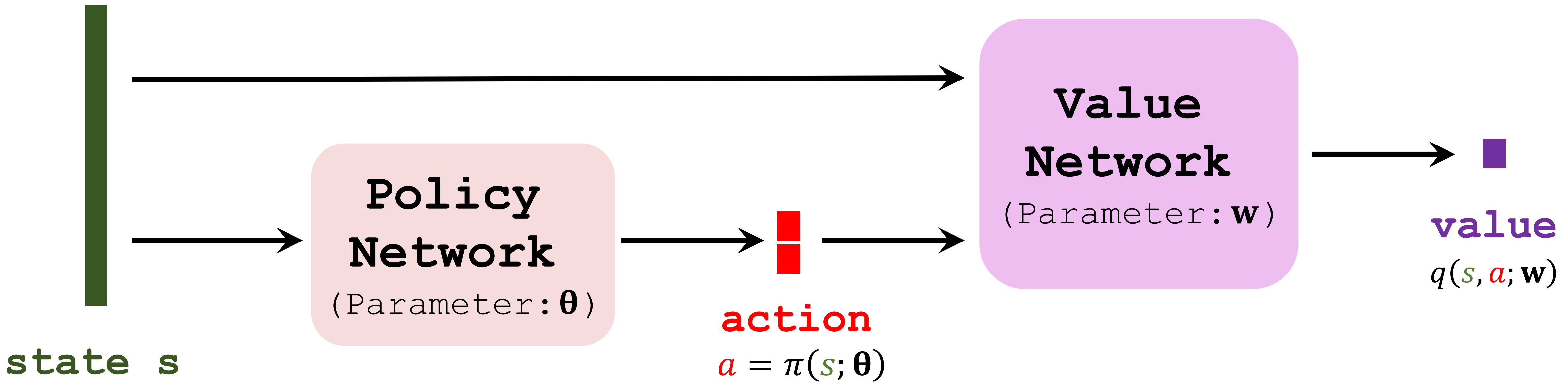
- Value network makes prediction for time $t+1$:

$$q_{t+1} = q(s_{t+1}, a'_{t+1}; \mathbf{w}), \text{ where } a'_{t+1} = \pi(s_{t+1}; \boldsymbol{\theta}).$$

- TD error: $\delta_t = q_t - (r_t + \gamma \cdot q_{t+1})$.
- Update: $\mathbf{w} \leftarrow \mathbf{w} - \alpha \cdot \delta_t \cdot \frac{\partial\, q(s_t, a_t; \mathbf{w})}{\partial \mathbf{w}}$.

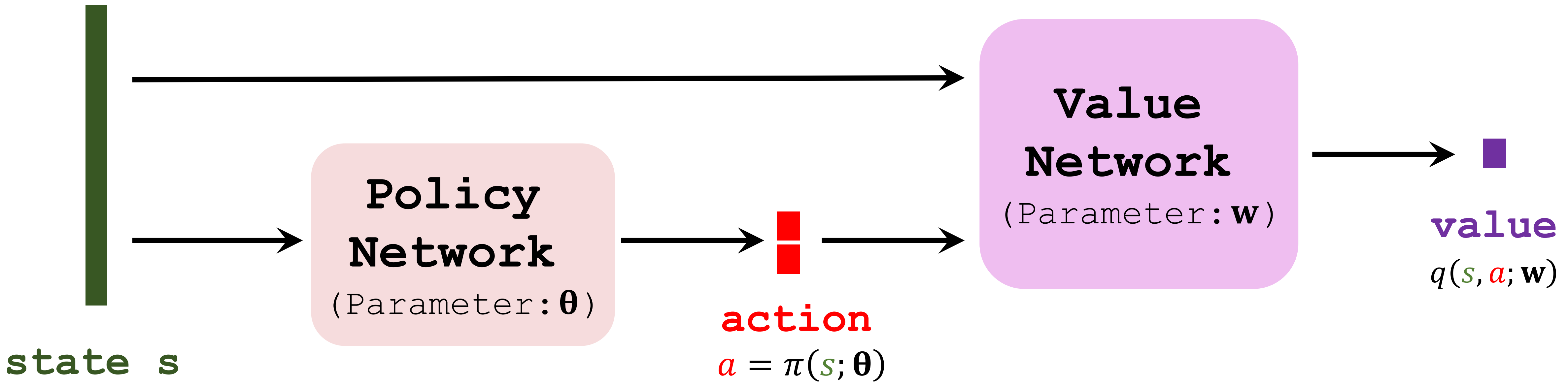
Updating Policy Network by DPG

- The critic $q(s, a; \mathbf{w})$ evaluates how good the action a is.



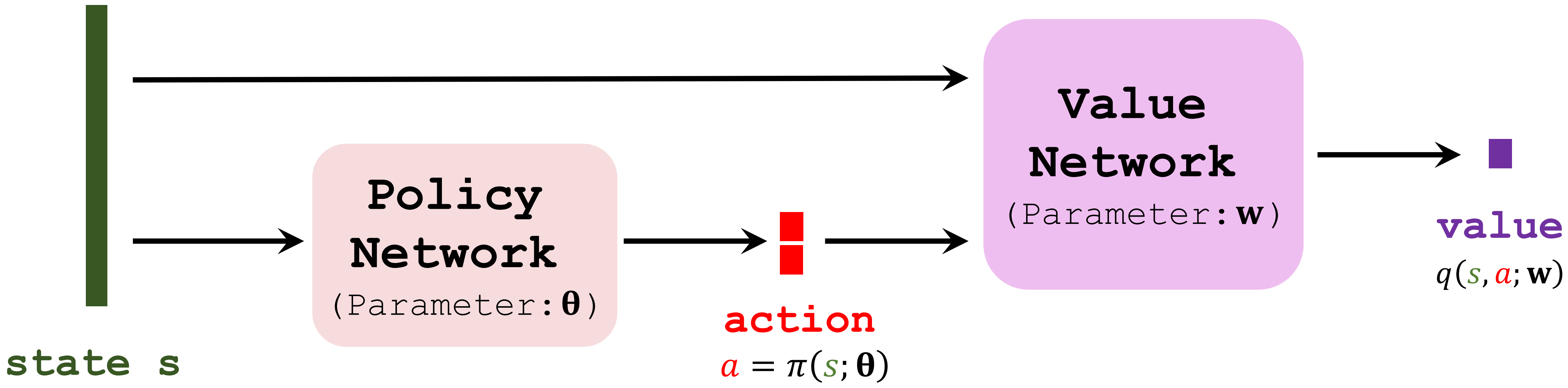
Updating Policy Network by DPG

- The critic $q(s, a; \mathbf{w})$ evaluates how good the action a is.
- Improve $\boldsymbol{\theta}$ so that the critic believes $a = \pi(s; \boldsymbol{\theta})$ is better.
- Update $\boldsymbol{\theta}$ so that $q(s, a; \mathbf{w}) = q(s, \pi(s; \boldsymbol{\theta}); \mathbf{w})$ increases.



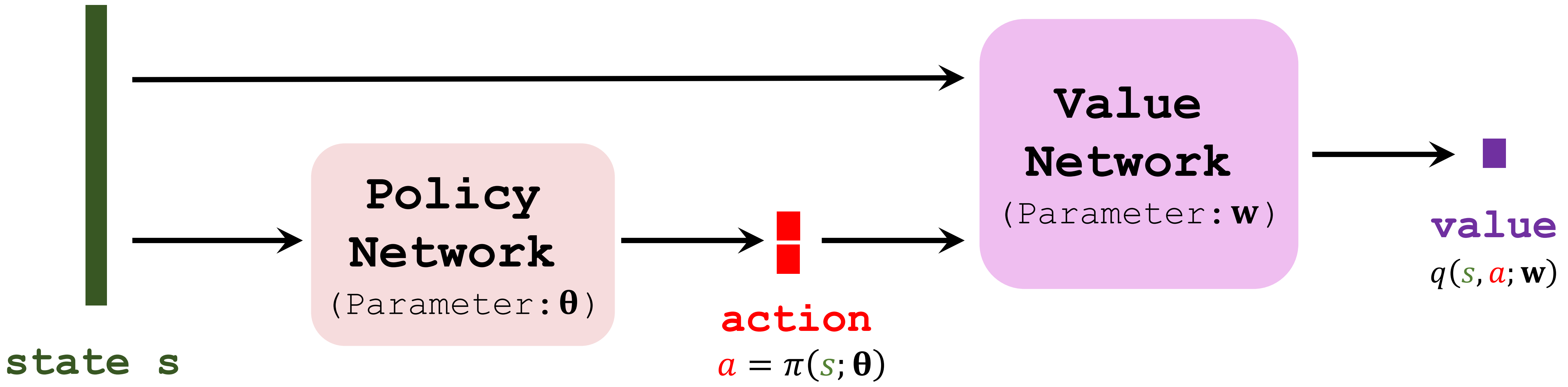
Updating Policy Network by DPG

- **Goal:** Increasing $q(s, a; \mathbf{w})$, where $a = \pi(s; \boldsymbol{\theta})$.



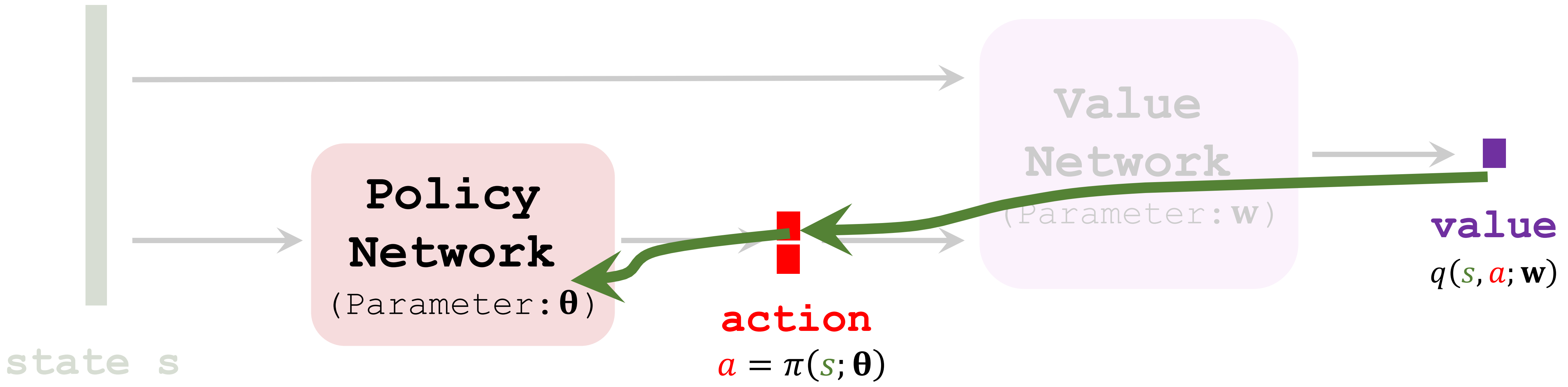
Updating Policy Network by DPG

- **Goal:** Increasing $q(s, a; \mathbf{w})$, where $a = \pi(s; \boldsymbol{\theta})$.
- DPG: $\mathbf{g} = \frac{\partial\, q(s,\pi(s;\boldsymbol{\theta});\mathbf{w})}{\partial\, \boldsymbol{\theta}} = \frac{\partial\, a}{\partial\, \boldsymbol{\theta}} \cdot \frac{\partial\, q(s,a;\mathbf{w})}{\partial\, a}$.



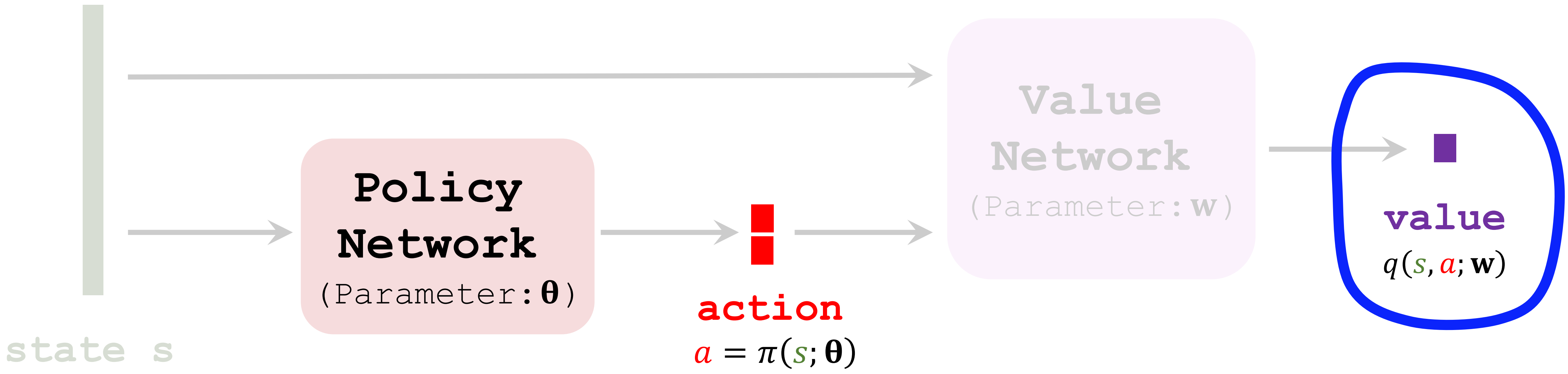
Updating Policy Network by DPG

- **Goal:** Increasing $q(s, a; \mathbf{w})$, where $a = \pi(s; \boldsymbol{\theta})$.
- DPG: $\mathbf{g} = \frac{\partial\, q(s,\pi(s;\boldsymbol{\theta});\mathbf{w})}{\partial\, \boldsymbol{\theta}} = \frac{\partial\, a}{\partial\, \boldsymbol{\theta}} \cdot \frac{\partial\, q(s,a;\mathbf{w})}{\partial\, a}$.



Updating Policy Network by DPG

- **Goal:** Increasing $q(s, a; \mathbf{w})$, where $a = \pi(s; \boldsymbol{\theta})$.
- DPG: $\mathbf{g} = \frac{\partial\, q(s,\pi(s;\boldsymbol{\theta});\mathbf{w})}{\partial\, \boldsymbol{\theta}} = \frac{\partial\, a}{\partial\, \boldsymbol{\theta}} \cdot \frac{\partial\, q(s,a;\mathbf{w})}{\partial\, a}.$
- Gradient ascent: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \beta \cdot \mathbf{g}.$



Improvement: Using Target Networks

Bootstrapping

- Value network makes a prediction for time t:

$$q_t = q(s_t, a_t; \mathbf{w}).$$

- Value network makes a prediction for time $t + 1$:

$$q_{t+1} = q(s_{t+1}, a'_{t+1}; \mathbf{w}), \text{ where } a'_{t+1} = \pi(s_{t+1}; \boldsymbol{\theta}).$$

Bootstrapping

- Value network makes a prediction for time t:

$$q_t = q(s_t, a_t; \mathbf{w}).$$

- Value network makes a prediction for time $t+1$:

$$q_{t+1} = q(s_{t+1}, a'_{t+1}; \mathbf{w}), \text{ where } a'_{t+1} = \pi(s_{t+1}; \boldsymbol{\theta}).$$

- TD error: $\delta_t = q_t - \underbrace{\boxed{(r_t + \gamma \cdot q_{t+1})}}_{\text{TD target}}.$

Bootstrapping

- Value network makes a prediction for time t:

$$q_t = q(s_t, a_t; \mathbf{w}).$$

- Value network makes a prediction for time $t+1$:

$$q_{t+1} = q(s_{t+1}, a'_{t+1}; \mathbf{w}), \text{ where } a'_{t+1} = \pi(s_{t+1}; \boldsymbol{\theta}).$$

- TD error: $\delta_t = \underline{q_t} - (r_t + \gamma \cdot \underline{q_{t+1}}).$

Bootstrapping: Using one's estimate to update oneself.

Bootstrapping

- Value network makes a prediction for time t:

$$q_t = q(s_t, a_t; \mathbf{w}).$$

- Value network makes a prediction for time $t+1$:

$$q_{t+1} = q(s_{t+1}, a'_{t+1}; \mathbf{w}), \text{ where } a'_{t+1} = \pi(s_{t+1}; \boldsymbol{\theta}).$$

- TD error: $\delta_t = q_t - \boxed{(r_t + \gamma \cdot q_{t+1})}.$

TD target

Basic idea: Computing the TD target using different networks.

Using target networks

- Value network makes a prediction for time t:

$$q_t = q(s_t, a_t; \mathbf{w}).$$

Using target networks

- Value network makes a prediction for time t:

$$q_t = q(s_t, a_t; \mathbf{w}).$$

- Target networks make a prediction for time $t+1$:

$$q_{t+1} = q(s_{t+1}, a'_{t+1}; \mathbf{w}^-), \text{ where } a'_{t+1} = \pi(s_{t+1}; \boldsymbol{\theta}^-).$$

Target value network

Target policy network

The same network structure, but different parameters.

Updating policy and value networks

- Policy network makes a decision: $a = \pi(s; \boldsymbol{\theta})$.
- Update policy network by DPG: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \beta \cdot \frac{\partial\, a}{\partial\, \boldsymbol{\theta}} \cdot \frac{\partial\, q(s,a;\mathbf{w})}{\partial\, a}$.

Updating policy and value networks

- Policy network makes a decision: $a = \pi(s; \boldsymbol{\theta})$.
- Update policy network by DPG: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \beta \cdot \frac{\partial\, a}{\partial\, \boldsymbol{\theta}} \cdot \frac{\partial\, q(s,a;\mathbf{w})}{\partial\, a}$.
- Value network computes $q_t = q(s, a; \mathbf{w})$.
- Target networks, $\pi(s; \boldsymbol{\theta}^-)$ and $q(s, a; \mathbf{w}^-)$, compute q_{t+1}.

Updating policy and value networks

- Policy network makes a decision: $a = \pi(s; \boldsymbol{\theta})$.
- Update policy network by DPG: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \beta \cdot \frac{\partial\, a}{\partial\, \boldsymbol{\theta}} \cdot \frac{\partial\, q(s,a;\mathbf{w})}{\partial\, a}$.
- Value network computes $q_t = q(s, a; \mathbf{w})$.
- Target networks, $\pi(s; \boldsymbol{\theta}^-)$ and $q(s, a; \mathbf{w}^-)$, compute q_{t+1}.
- TD error: $\delta_t = q_t - (r_t + \gamma \cdot q_{t+1})$.
- Update value network by TD: $\mathbf{w} \leftarrow \mathbf{w} - \alpha \cdot \delta_t \cdot \frac{\partial\, q(s,a;\mathbf{w})}{\partial \mathbf{w}}$.

Updating target networks

- Set a hyper-parameter $\tau \in (0, 1)$.
- Update the target networks by weighted averaging:

$$\mathbf{w}^{-} \;\leftarrow\; \tau \cdot \mathbf{w} + (1 - \tau) \cdot \mathbf{w}^{-}.$$

$$\boldsymbol{\theta}^{-} \;\leftarrow\; \tau \cdot \boldsymbol{\theta} + (1 - \tau) \cdot \boldsymbol{\theta}^{-}.$$

Improvements

- Target networks.
- Experience replay.
- Multi-step TD target.

Stochastic Policy VS Deterministic Policy

	Stochastic Policy	Deterministic Policy
Policy:	$\pi(a \| s; \boldsymbol{\theta})$	$\pi(s; \boldsymbol{\theta})$

	Stochastic Policy	Deterministic Policy
Policy:	$\pi(a \mid s; \boldsymbol{\theta})$	$\pi(s; \boldsymbol{\theta})$
Output:	Probability distribution over the action space	Action a

	Stochastic Policy	Deterministic Policy
Policy:	$\pi(a \mid s; \boldsymbol{\theta})$	$\pi(s; \boldsymbol{\theta})$
Output:	Probability distribution over the action space	Action a
Control:	Randomly sample an action from the distribution	Directly use the output, a

	Stochastic Policy	Deterministic Policy
Policy:	$\pi(a \mid s; \boldsymbol{\theta})$	$\pi(s; \boldsymbol{\theta})$
Output:	Probability distribution over the action space	Action a
Control:	Randomly sample an action from the distribution	Directly use the output, a
Application:	Mostly discrete control	Continuous control

Thank you!