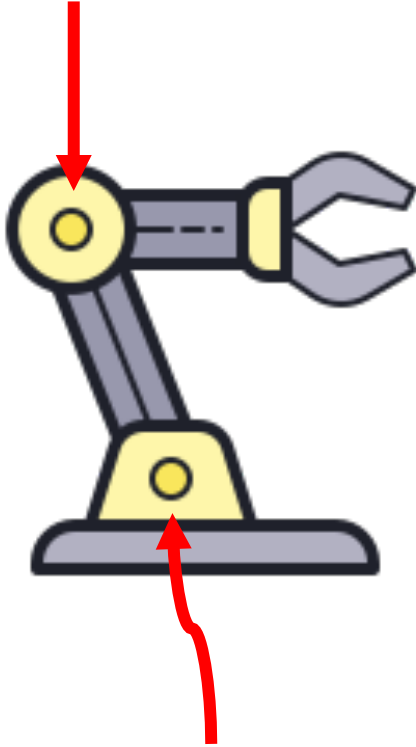


Deterministic Policy Gradient (DPG)

Shusen Wang

Continuous Action Space

$$a_1 \in [0^\circ, 360^\circ]$$



$$a_2 \in [0^\circ, 180^\circ]$$

- The action space \mathcal{A} is a subset of \mathbb{R}^2 .
- The action space \mathcal{A} is continuous:
$$\mathcal{A} = [0^\circ, 360^\circ] \times [0^\circ, 180^\circ].$$
- Actions are 2-dim vectors.

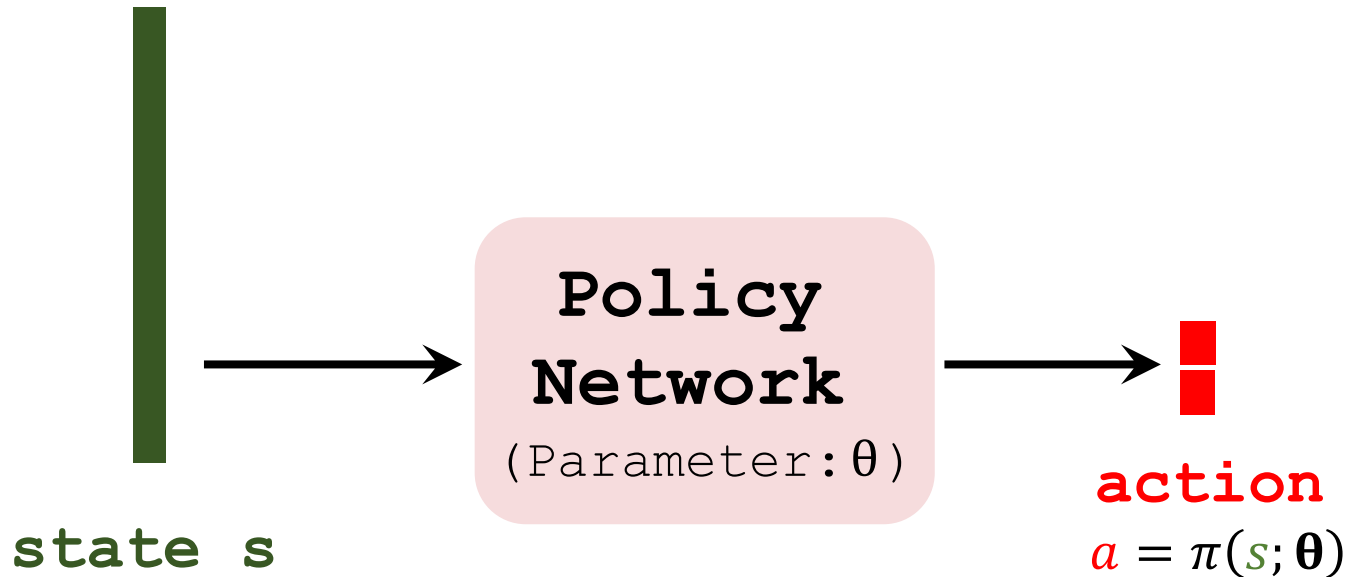
Deterministic Policy Gradient (DPG)

Reference

- Silver et al. [Deterministic Policy Gradient Algorithms](#). In *ICML*, 2014.
- Lillicrap et al. [Continuous control with deep reinforcement learning](#). In *ICLR*, 2016.

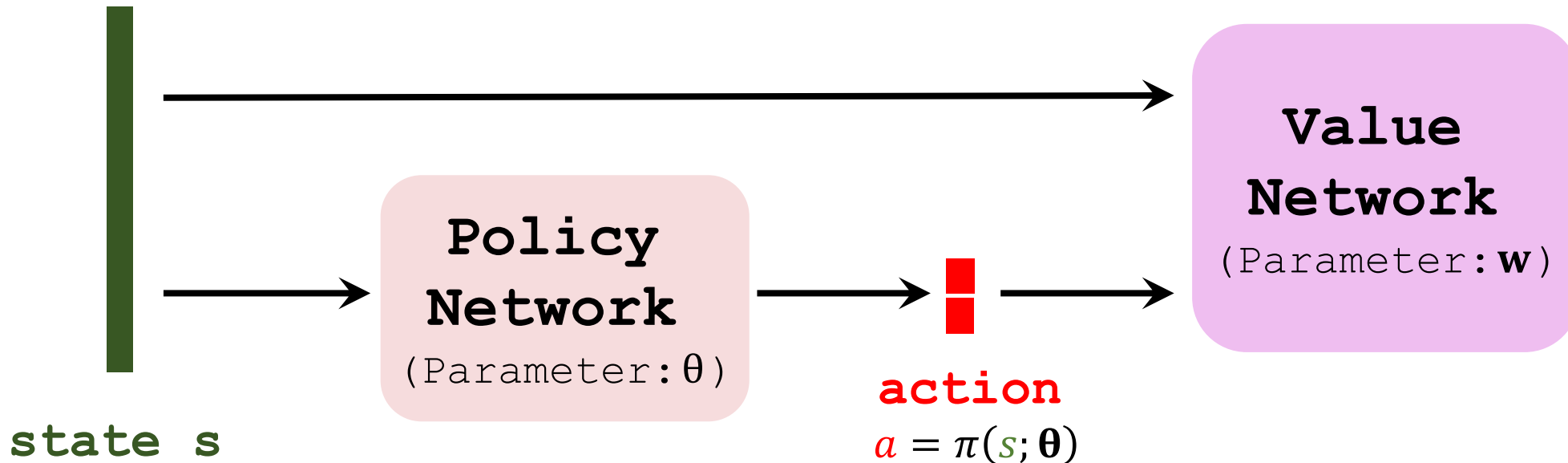
Deterministic Actor-Critic

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



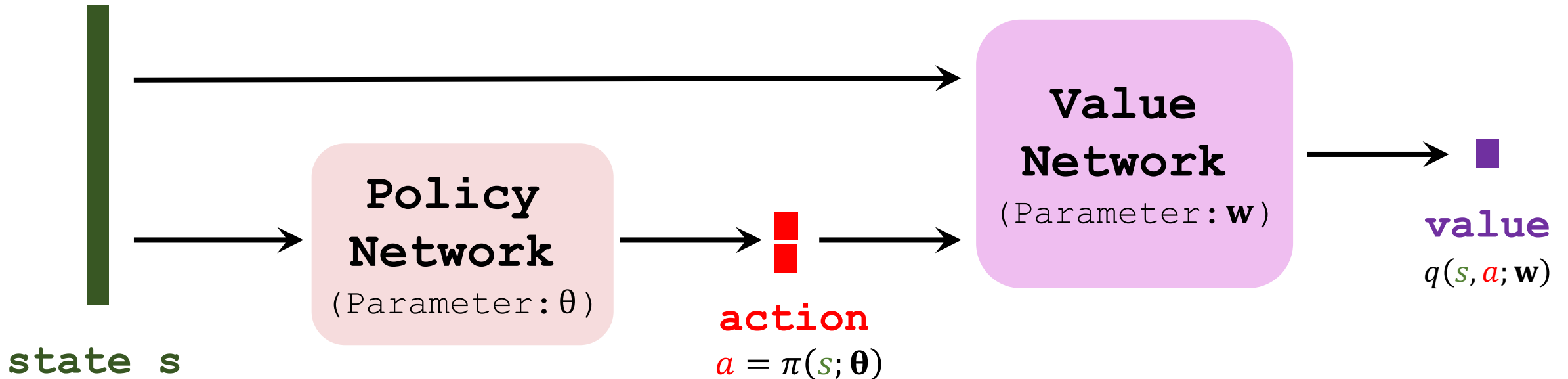
Deterministic Actor-Critic

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



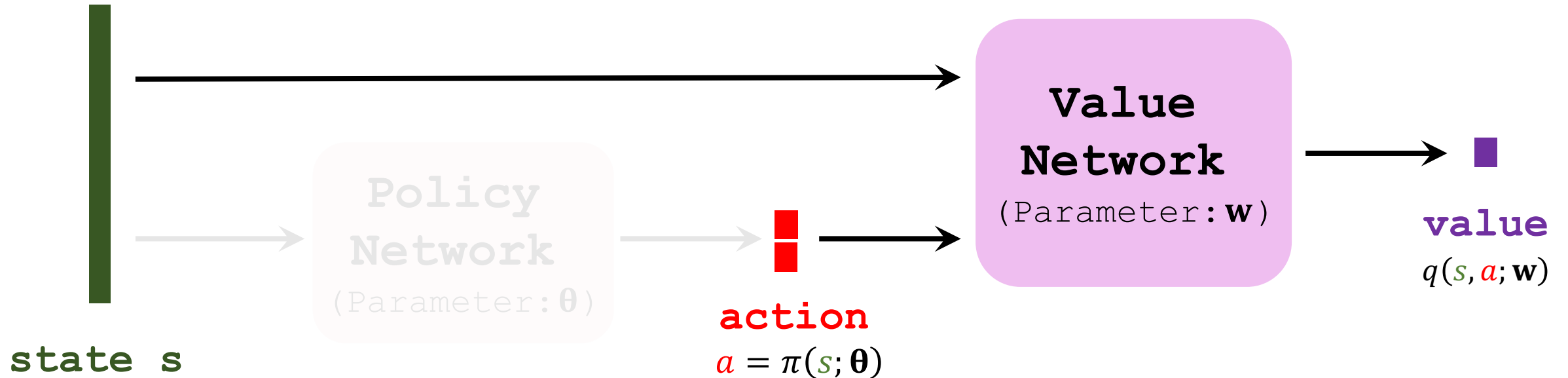
Deterministic Actor-Critic

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

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

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



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 $t + 1$:

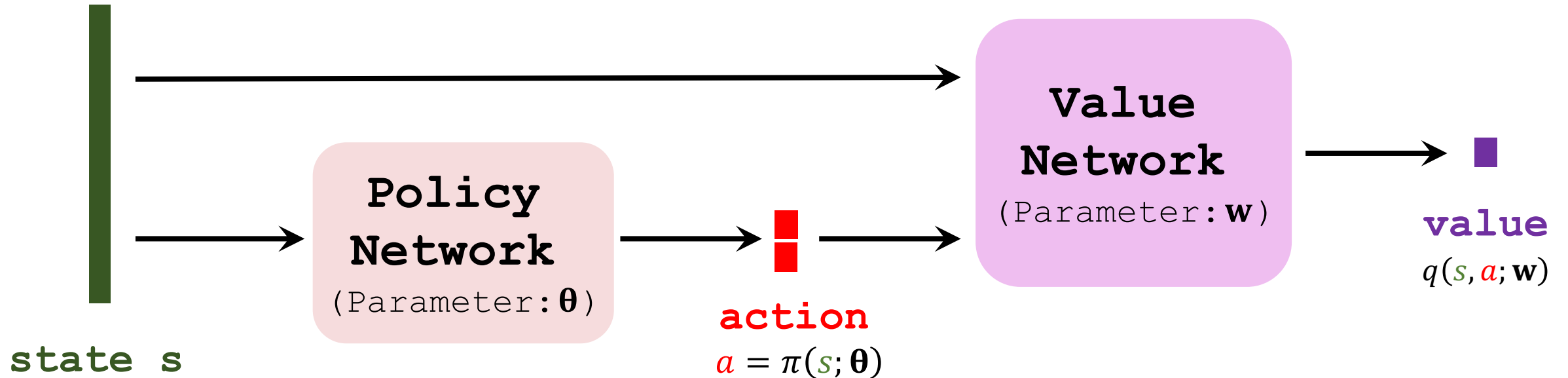
$$q_{t+1} = q(s_{t+1}, a'_{t+1}; \mathbf{w}), \text{ where } a'_{t+1} = \pi(s_{t+1}; \boldsymbol{\theta}).$$

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

- Update: $\mathbf{w} \leftarrow \mathbf{w} - \alpha \cdot \delta_t \cdot \frac{\partial q(s_t, a_t; \mathbf{w})}{\partial \mathbf{w}}$.

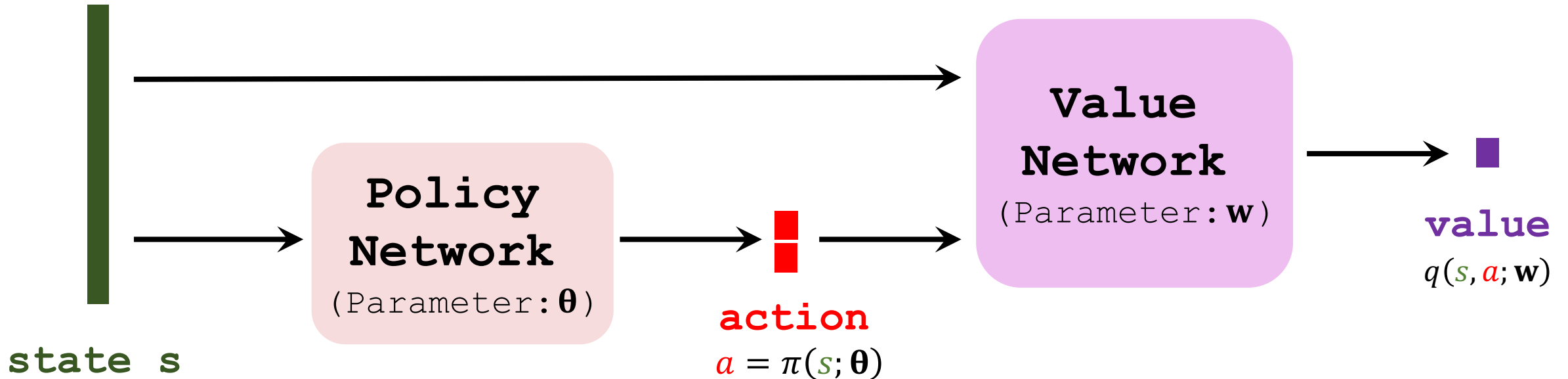
Updating Policy Network by DPG

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



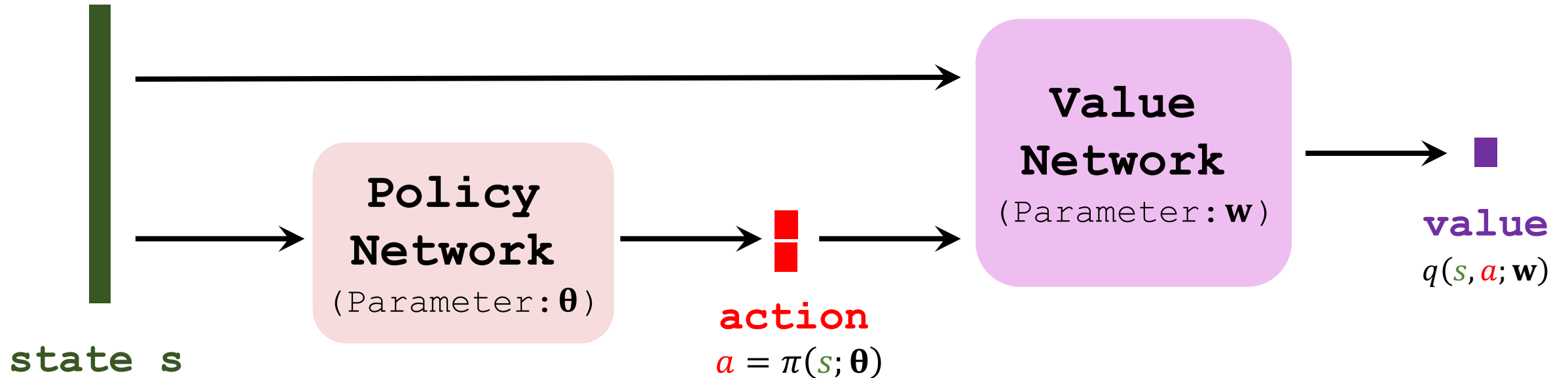
Updating Policy Network by DPG

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



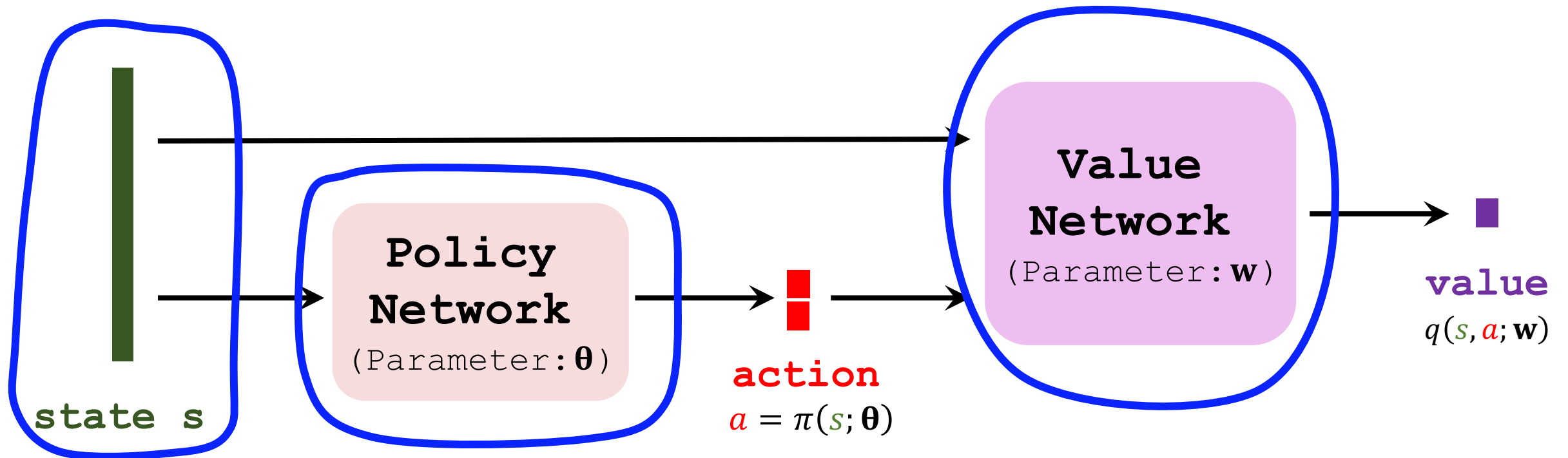
Updating Policy Network by DPG

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



Updating Policy Network by DPG

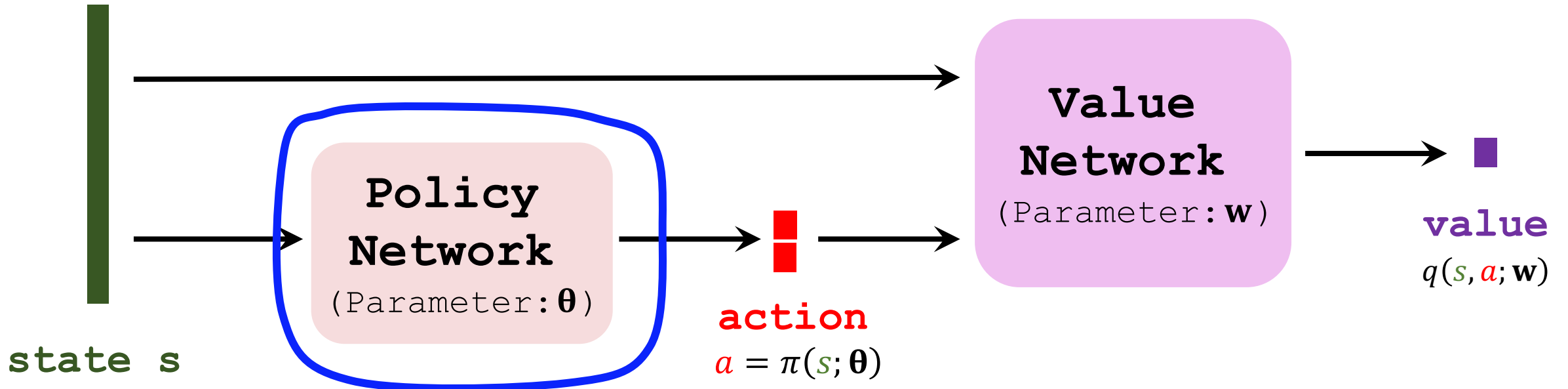
- **Goal:** Increase $q(s, a; \mathbf{w})$, where $a = \pi(s; \theta)$.



Updating Policy Network by DPG

- **Goal:** Increase $q(s, a; w)$, where $a = \pi(s; \theta)$.

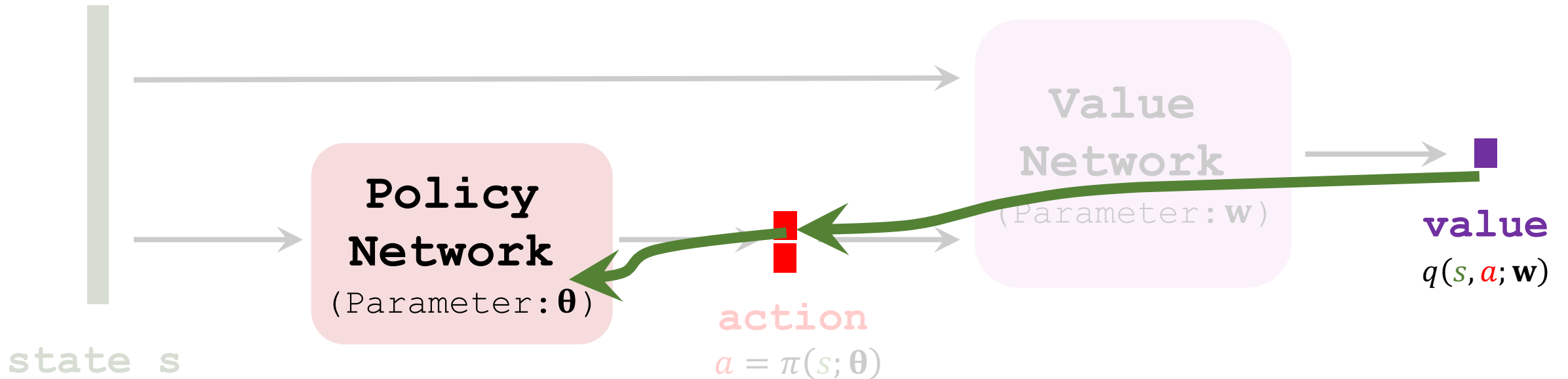
- DPG: $\mathbf{g} = \frac{\partial q(s, \pi(s; \theta); w)}{\partial \theta} = \frac{\partial a}{\partial \theta} \cdot \frac{\partial q(s, a; w)}{\partial a}$.



Updating Policy Network by DPG

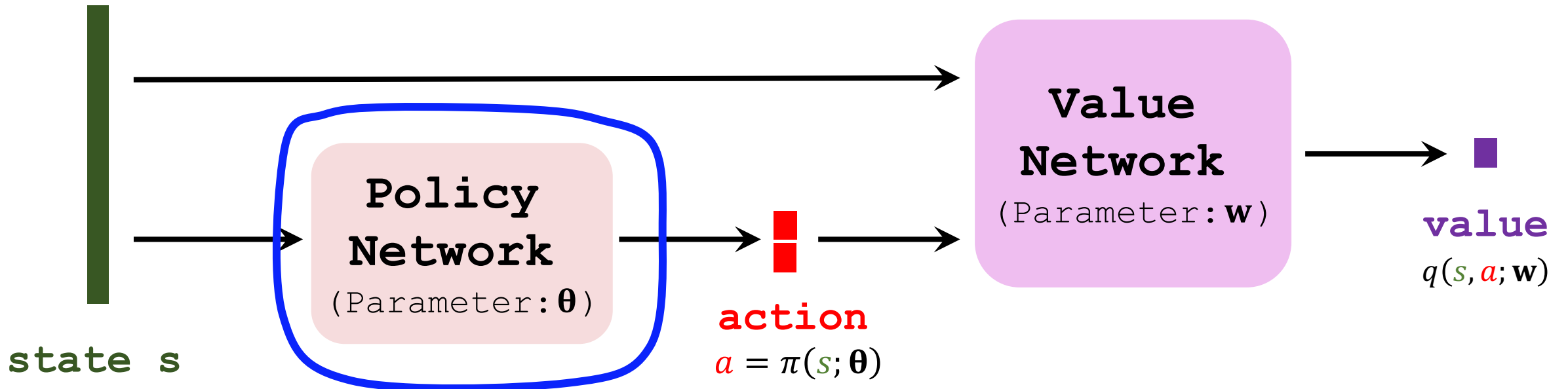
- **Goal:** Increase $q(s, a; \mathbf{w})$, where $a = \pi(s; \theta)$.

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



Updating Policy Network by DPG

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



Improvement: Using Target Networks

Bootstrapping

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

TD target

Bootstrapping

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



Bootstrapping: Using one's estimate to update oneself.

Bootstrapping

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

TD target

Basic idea: Compute the TD target using different networks.

Using target networks

- Value network makes prediction for time t :

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

- Target value network makes prediction for time $t + 1$:

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

Target value network

Target policy network

The same network structure, but different parameters.

Updating policy and value networks

- Policy network makes decision: $a = \pi(s; \theta)$.

- Updating policy network by DPG: $\theta \leftarrow \theta + \beta \cdot \frac{\partial a}{\partial \theta} \cdot \frac{\partial q(s, a; \mathbf{w})}{\partial a}$.

Updating policy and value networks

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

Stochastic Policy VS Deterministic Policy

Stochastic Policy

Deterministic Policy

Policy:

$$\pi(a|s; \theta)$$

$$\pi(s; \theta)$$

Stochastic Policy

Deterministic Policy

Policy:

$$\pi(\textcolor{red}{a}|\textcolor{green}{s}; \boldsymbol{\theta})$$

$$\pi(\textcolor{green}{s}; \boldsymbol{\theta})$$

Output:

Probability distribution
over the action space

Action $\textcolor{red}{a}$

Stochastic Policy

Deterministic Policy

Policy:

$$\pi(\textcolor{red}{a}|\textcolor{green}{s}; \boldsymbol{\theta})$$

$$\pi(\textcolor{green}{s}; \boldsymbol{\theta})$$

Output:

Probability distribution
over the action space

Action $\textcolor{red}{a}$

Control:

Randomly sample an action
from the distribution

Directly use
the output, $\textcolor{red}{a}$

Stochastic Policy

Deterministic Policy

Policy:

$$\pi(\textcolor{red}{a}|\textcolor{green}{s}; \boldsymbol{\theta})$$

$$\pi(\textcolor{green}{s}; \boldsymbol{\theta})$$

Output:

Probability distribution
over the action space

Action $\textcolor{red}{a}$

Control:

Randomly sample an action
from the distribution

Directly use
the output, $\textcolor{red}{a}$

Application:

Mostly discrete
control

Continuous
control

Thank you!