强化学习与博弈论
Reinforcement Learning and Game Theory

陈 旭

计算机学院

中山大學
SUN YAT-SEN UNIVERSITY

Policy-Based Reinforcement Learning

Policy Function Approximation

Policy Function $\pi(a|s)$

- Policy function $\pi(a|s)$ is a probability density function (PDF).
- It takes state s as input.
- It output the probabilities for all the actions, e.g.,

$$\pi(\text{left}|s) = 0.2,$$
$$\pi(\text{right}|s) = 0.1,$$
$$\pi(\text{up}|s) = 0.7.$$

- Randomly sample action a random drawn from the distribution.

Can we directly learn a policy function $\pi(a|s)$?

- If there are only a few states and actions, then yes, we can.
- Draw a table (matrix) and learn the entries.

	Action a_1	Action a_2	Action a_3	Action a_4	$\cdots$
State s_1					
State s_2					
State s_3					
$\vdots$					

Can we directly learn a policy function $\pi(a|s)$?

- If there are only a few states and actions, then yes, we can.
- Draw a table (matrix) and learn the entries.
- What if there are too many (or infinite) states or actions?

	Action a_1	Action a_2	Action a_3	Action a_4	$\cdots$
State s_1					
State s_2					
State s_3					
$\vdots$					

Policy Network $\pi(a|s;\boldsymbol{\theta})$

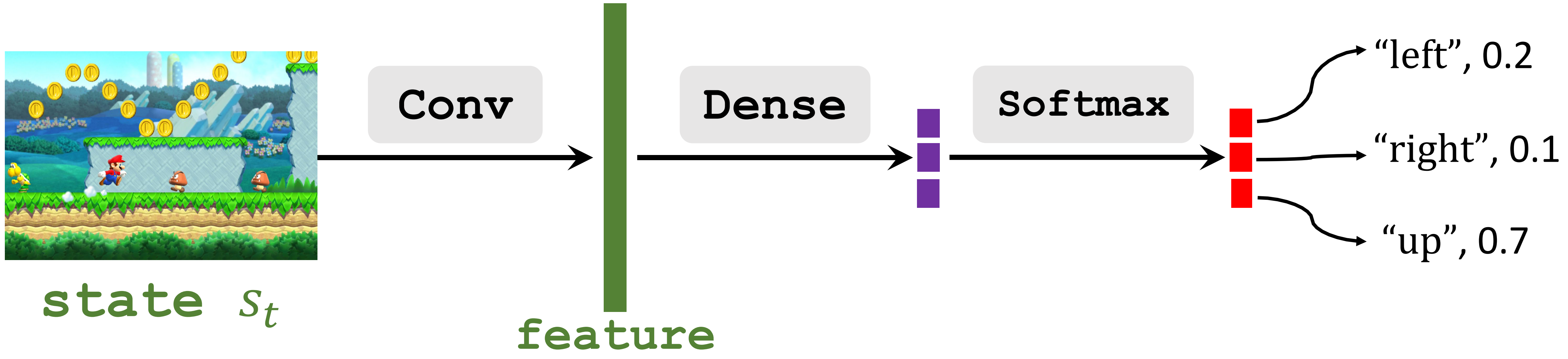
Policy network: Use a neural net to approximate $\pi(a|s)$.

- Use policy network $\pi(a|s;\boldsymbol{\theta})$ to approximate $\pi(a|s)$.
- $\boldsymbol{\theta}$: trainable parameters of the neural net.

Policy Network $\pi(a|s;\boldsymbol{\theta})$

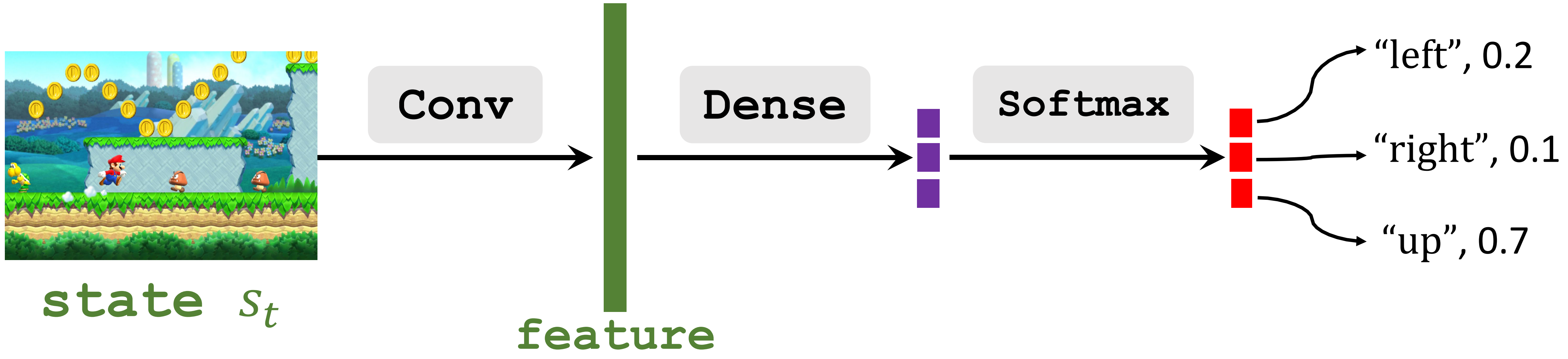
Policy network: Use a neural net to approximate $\pi(a|s)$.

- Use policy network $\pi(a|s;\boldsymbol{\theta})$ to approximate $\pi(a|s)$.
- $\boldsymbol{\theta}$: trainable parameters of the neural net.



Policy Network $\pi(a|s;\boldsymbol{\theta})$

- $\sum_{a \in \mathcal{A}} \pi(a|s;\boldsymbol{\theta}) = 1.$
- Here, $\mathcal{A} = \{\text{"left"}, \text{"right"}, \text{"up"}\}$ is the set all actions.
- That is why we use softmax activation.



State-Value Function Approximation

Action-Value Function

Definition: Discounted return.

- $U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \gamma^3 \cdot R_{t+3} + \cdots$

- The return depends on actions $A_t, A_{t+1}, A_{t+2}, \cdots$ and states $S_t, S_{t+1}, S_{t+2}, \cdots$
- Actions are random: $\mathbb{P}[A = a \mid S = s] = \pi(a|s)$. (Policy function.)
- States are random: $\mathbb{P}[S' = s'|S = s, A = a] = p(s'|s, a)$. (State transition.)

Action-Value Function

Definition: Discounted return.

- $U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \gamma^3 \cdot R_{t+3} + \cdots$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}\left[U_t \mid S_t = s_t, A_t = a_t\right].$

The expectation is taken w.r.t.
actions $A_{t+1}, A_{t+2}, A_{t+3}, \cdots$
and states $S_{t+1}, S_{t+2}, S_{t+3}, \cdots$

State-Value Function

Definition: Discounted return.

- $U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \gamma^3 \cdot R_{t+3} + \cdots$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}\left[U_t | S_t = s_t, A_t = a_t\right].$

Definition: State-value function.

- $V_\pi(s_t) = \mathbb{E}_A\left[Q_\pi(s_t, A)\right]$

Integrate out action $A \sim \pi(\cdot | s_t)$.

State-Value Function

Definition: Discounted return.

- $U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \gamma^3 \cdot R_{t+3} + \cdots$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}\left[U_t \mid S_t = s_t, A_t = a_t\right].$

Definition: State-value function.

- $V_\pi(s_t) = \mathbb{E}_A\left[Q_\pi(s_t, A)\right] = \sum_a \pi(a \mid s_t) \cdot Q_\pi(s_t, a).$

Integrate out action $A \sim \pi(\cdot \mid s_t)$.

Policy-Based Reinforcement Learning

Definition: State-value function.

- $V_\pi(s_t) = \mathbb{E}_A\left[Q_\pi(s_t, A)\right] = \sum_a \pi(a|s_t) \cdot Q_\pi(s_t, a)$.

Policy-Based Reinforcement Learning

Definition: State-value function.

- $V_\pi(s_t) = \mathbb{E}_A\left[Q_\pi(s_t, A)\right] = \sum_a \pi(a|s_t) \cdot Q_\pi(s_t, a)$.

Approximate state-value function.

- Approximate policy function $\pi(a|s_t)$ by policy network $\pi(a|s_t; \boldsymbol{\theta})$.

Policy-Based Reinforcement Learning

Definition: State-value function.

- $V_\pi(s_t) = \mathbb{E}_A\,[Q_\pi(s_t, A)] = \sum_a \pi(a|s_t) \cdot Q_\pi(s_t, a)$.

Approximate state-value function.

- Approximate policy function $\pi(a|s_t)$ by policy network $\pi(a|s_t; \boldsymbol{\theta})$.
- Approximate value function $V_\pi(s_t)$ by:

$$V(s_t; \boldsymbol{\theta}) = \sum_a \pi(a|s_t; \boldsymbol{\theta}) \cdot Q_\pi(s_t, a).$$

Policy-Based Reinforcement Learning

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy-based learning: Learn $\boldsymbol{\theta}$ that maximizes $J(\boldsymbol{\theta}) = \mathbb{E}_S[V(S;\boldsymbol{\theta})]$.

Policy-Based Reinforcement Learning

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy-based learning: Learn $\boldsymbol{\theta}$ that maximizes $J(\boldsymbol{\theta}) = \mathbb{E}_S[V(S;\boldsymbol{\theta})]$.

How to improve $\boldsymbol{\theta}$? Policy gradient ascent!

- Observe state s.
- Update policy by: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \beta \cdot \boxed{\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}}$.

Policy gradient

Policy Gradient

Reference

- Sutton and others: Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, 2000.

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \frac{\partial \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)}{\partial\, \boldsymbol{\theta}}$

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $$\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \frac{\partial \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)}{\partial\, \boldsymbol{\theta}}$$

$$= \sum_a \frac{\partial \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)}{\partial\, \boldsymbol{\theta}}$$

Push derivative inside the summation

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $$\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \frac{\partial \sum_a \pi(a|s;\boldsymbol{\theta})\cdot Q_\pi(s,a)}{\partial\, \boldsymbol{\theta}}$$
$$= \sum_a \frac{\partial \pi(a|s;\boldsymbol{\theta})\cdot Q_\pi(s,a)}{\partial\, \boldsymbol{\theta}}$$
$$= \sum_a \frac{\partial \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a)$$

Pretend Q_π is independent of $\boldsymbol{\theta}$.
(It may not be true.)

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \sum_a \frac{\partial \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a)$

Policy Gradient

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $$\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \sum_a \boxed{\frac{\partial \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}} \, Q_\pi(s,a)$$
$$= \sum_a \boxed{\pi(a|s;\boldsymbol{\theta}) \cdot \frac{\partial \log \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}} \cdot Q_\pi(s,a)$$

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $$\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \sum_a \frac{\partial \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a)$$
$$= \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot \boxed{\frac{\partial \log \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}} \cdot Q_\pi(s,a)$$

- Chain rule: $\frac{\partial \log[\pi(\boldsymbol{\theta})]}{\partial \boldsymbol{\theta}} = \frac{1}{\pi(\boldsymbol{\theta})} \cdot \frac{\partial\, \pi(\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$.

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \sum_a \frac{\partial \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a)$

$$= \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot \frac{\partial \log \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a)$$

- Chain rule: $\frac{\partial \log[\pi(\boldsymbol{\theta})]}{\partial \boldsymbol{\theta}} = \frac{1}{\pi(\boldsymbol{\theta})} \cdot \frac{\partial\, \pi(\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$.
- $\Rightarrow$ $\pi(\boldsymbol{\theta}) \cdot \frac{\partial \log[\pi(\boldsymbol{\theta})]}{\partial \boldsymbol{\theta}} = \pi(\boldsymbol{\theta}) \cdot \frac{1}{\pi(\boldsymbol{\theta})} \cdot \frac{\partial\, \pi(\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \sum_a \frac{\partial \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a)$

$$= \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot \frac{\partial \log \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a)$$

- Chain rule: $\frac{\partial \log[\pi(\boldsymbol{\theta})]}{\partial \boldsymbol{\theta}} = \frac{1}{\pi(\boldsymbol{\theta})} \cdot \frac{\partial\, \pi(\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$.
- → $\pi(\boldsymbol{\theta}) \cdot \frac{\partial \log[\pi(\boldsymbol{\theta})]}{\partial \boldsymbol{\theta}} = \cancel{\pi(\boldsymbol{\theta})} \cdot \cancel{\frac{1}{\pi(\boldsymbol{\theta})}} \cdot \frac{\partial\, \pi(\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$.

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \sum_a \boxed{\frac{\partial \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}} Q_\pi(s,a)$

$$= \sum_a \boxed{\pi(a|s;\boldsymbol{\theta}) \cdot \frac{\partial \log \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}} \cdot Q_\pi(s,a)$$

- Chain rule: $\frac{\partial \log[\pi(\boldsymbol{\theta})]}{\partial \boldsymbol{\theta}} = \frac{1}{\pi(\boldsymbol{\theta})} \cdot \frac{\partial\, \pi(\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$.
- $\Rightarrow$ $\pi(\boldsymbol{\theta}) \cdot \frac{\partial \log[\pi(\boldsymbol{\theta})]}{\partial \boldsymbol{\theta}} = \cancel{\pi(\boldsymbol{\theta})} \cdot \frac{1}{\cancel{\pi(\boldsymbol{\theta})}} \cdot \frac{\partial\, \pi(\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$.

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $$\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \sum_a \frac{\partial \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a)$$
$$= \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot \frac{\partial \log \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a)$$
$$= \mathbb{E}_A\left[\frac{\partial \log \pi(A|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,A)\right].$$

The expectation is taken w.r.t. the random variable $A \sim \pi(\cdot\,|s;\boldsymbol{\theta})$.

Policy Gradient

Policy gradient:

$$\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{A\sim\pi(\cdot|s;\boldsymbol{\theta})}\left[\frac{\partial \log \pi(A|s,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s, A)\right].$$

Calculate Policy Gradient

Policy Gradient: $$\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{A\sim\pi(\cdot|s;\boldsymbol{\theta})}\left[\frac{\partial \log \pi(A|s,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s, A)\right].$$

Calculate Policy Gradient

Policy Gradient: $\frac{\partial V(s;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbb{E}_{A\sim\pi(\cdot|s;\boldsymbol{\theta})}\left[\frac{\partial \log \pi(A|s,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot Q_\pi(s, A)\right].$

1. Randomly sample an action $\hat{a}$ according to $\pi(\cdot\,|s;\boldsymbol{\theta})$.

Calculate Policy Gradient

Policy Gradient: $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{A\sim\pi(\cdot|s;\boldsymbol{\theta})}\left[\boxed{\frac{\partial \log \pi(A|s,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s, A)}\right].$

1. Randomly sample an action $\hat{a}$ according to $\pi(\cdot\,|s;\boldsymbol{\theta})$.
2. Calculate $\boxed{\mathbf{g}(\hat{a}, \boldsymbol{\theta}) = \frac{\partial \log \pi(\hat{a}|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s, \hat{a}).}$

- By the definition of $\mathbf{g}$, $\mathbb{E}_A[\mathbf{g}(A, \boldsymbol{\theta})] = \frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$.
- $\mathbf{g}(\hat{a}, \boldsymbol{\theta})$ is an unbiased estimate of $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$.

Calculate Policy Gradient

Policy Gradient: $$\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{A\sim\pi(\cdot|s;\boldsymbol{\theta})}\left[\frac{\partial \log \pi(A|s,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s, A)\right].$$

1. Randomly sample an action $\hat{a}$ according to $\pi(\cdot\,|s;\boldsymbol{\theta})$.
2. Calculate $\mathbf{g}(\hat{a}, \boldsymbol{\theta}) = \frac{\partial \log \pi(\hat{a}|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s, \hat{a})$.
3. Use $\mathbf{g}(\hat{a}, \boldsymbol{\theta})$ as an approximation to the policy gradient $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$.

Update policy network using policy gradient

Algorithm

1. Observe the state s_t.
2. Randomly sample action a_t according to $\pi(\cdot\,|s_t; \boldsymbol{\theta}_t)$.

Algorithm

1. Observe the state s_t.
2. Randomly sample action a_t according to $\pi(\cdot \mid s_t; \boldsymbol{\theta}_t)$.
3. Compute $q_t \approx Q_\pi(s_t, a_t)$ (some estimate).
4. Differentiate policy network: $\mathbf{d}_{\theta,t} = \frac{\partial \log \pi(a_t|s_t,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \mid_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$.

Algorithm

1. Observe the state s_t.
2. Randomly sample action a_t according to $\pi(\cdot \mid s_t; \boldsymbol{\theta}_t)$.
3. Compute $q_t \approx Q_\pi(s_t, a_t)$ (some estimate).
4. Differentiate policy network: $\mathbf{d}_{\theta,t} = \frac{\partial \log \pi(a_t | s_t, \boldsymbol{\theta})}{\partial \, \boldsymbol{\theta}} \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$.
5. (Approximate) policy gradient: $\mathbf{g}(a_t, \boldsymbol{\theta}_t) = q_t \cdot \mathbf{d}_{\theta,t}$.
6. Update policy network: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \beta \cdot \mathbf{g}(a_t, \boldsymbol{\theta}_t)$.

Algorithm

1. Observe the state s_t.
2. Randomly sample action a_t according to $\pi(\cdot \mid s_t; \boldsymbol{\theta}_t)$.
3. $\boxed{\text{Compute } q_t \approx Q_\pi(s_t, a_t) \text{ (some estimate).}}$ How?
4. Differentiate policy network: $\mathbf{d}_{\theta,t} = \frac{\partial \log \pi(a_t \mid s_t, \boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$.
5. (Approximate) policy gradient: $\mathbf{g}(a_t, \boldsymbol{\theta}_t) = q_t \cdot \mathbf{d}_{\theta,t}$.
6. Update policy network: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \beta \cdot \mathbf{g}(a_t, \boldsymbol{\theta}_t)$.

Algorithm

1. Observe the state s_t.
2. Randomly sample action a_t according to $\pi(\cdot \mid s_t; \boldsymbol{\theta}_t)$.
3. $\boxed{\text{Compute } q_t \approx Q_\pi(s_t, a_t) \text{ (some estimate).}}$ How?

Option 1: REINFORCE.

- Play the game to the end and generate the trajectory:

$$s_1, a_1, r_1, \; s_2, a_2, r_2, \quad \cdots \quad , s_T, a_T, r_T.$$

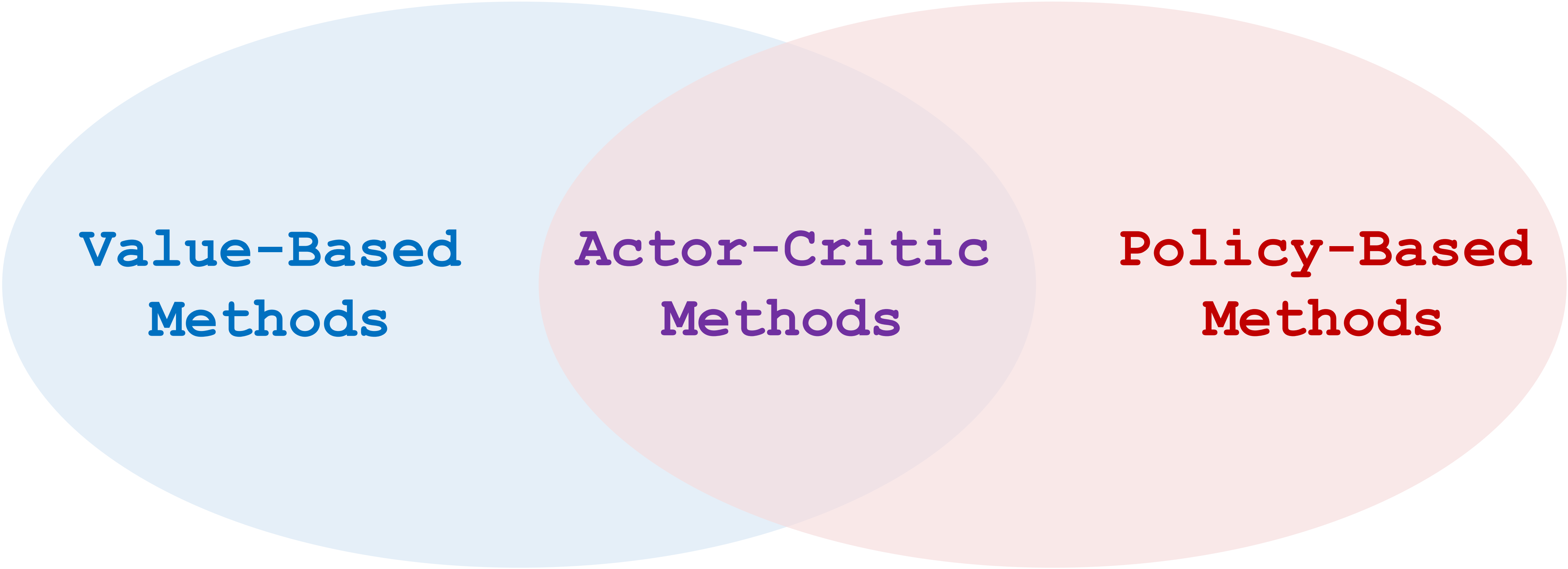
- Compute the discounted return $u_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$, for all t.
- Since $Q_\pi(s_t, a_t) = \mathbb{E}[U_t]$, we can use u_t to approximate $Q_\pi(s_t, a_t)$.
- ➔ Use $q_t = u_t$.

Algorithm

1. Observe the state s_t.
2. Randomly sample action a_t according to $\pi(\cdot \mid s_t; \boldsymbol{\theta}_t)$.
3. $\boxed{\text{Compute } q_t \approx Q_\pi(s_t, a_t) \text{ (some estimate).}}$ How?

Option 2: Approximate Q_π using a neural network.

- This leads to the actor-critic method.


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

Value Network and Policy Network

State-Value Function Approximation

Definition: State-value function.

- $V_\pi(s) = \sum_a \pi(a|s) \cdot Q_\pi(s, a)$.

State-Value Function Approximation

Definition: State-value function.

- $V_\pi(s) = \sum_a \pi(a|s) \cdot Q_\pi(s, a)$.

Policy network (actor):

- Use neural net $\pi(a|s; \boldsymbol{\theta})$ to approximate $\pi(a|s)$.
- $\boldsymbol{\theta}$: trainable parameters of the neural net.

State-Value Function Approximation

Definition: State-value function.

- $V_\pi(s) = \sum_a \pi(a|s) \cdot Q_\pi(s, a)$.

Policy network (actor):

- Use neural net $\pi(a|s; \boldsymbol{\theta})$ to approximate $\pi(a|s)$.
- $\boldsymbol{\theta}$: trainable parameters of the neural net.

Value network (critic):

- Use neural net $q(s, a; \mathbf{w})$ to approximate $Q_\pi(s, a)$.
- $\mathbf{w}$: trainable parameters of the neural net.

State-Value Function Approximation

Definition: State-value function.

- $V_\pi(s) = \sum_a \pi(a|s) \cdot Q_\pi(s, a) \approx \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot q(s, a; \mathbf{w})$.

Policy network (actor):

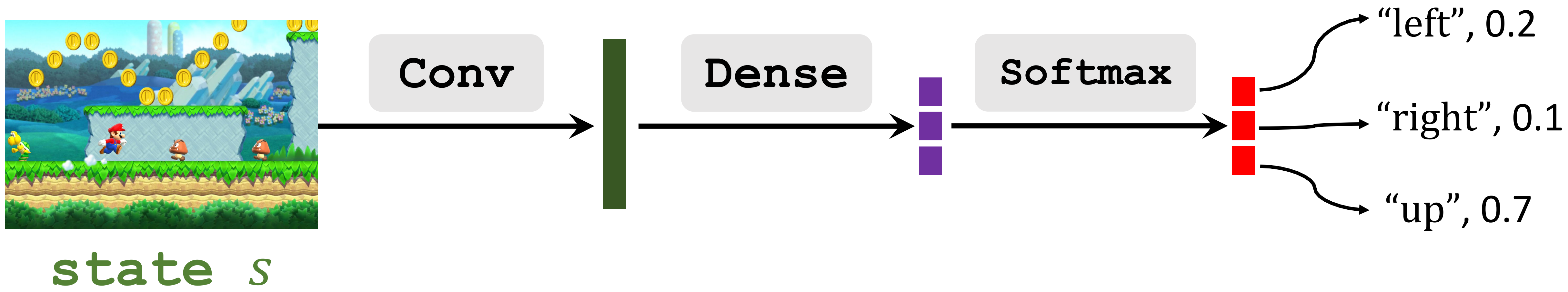
- Use neural net $\pi(a|s;\boldsymbol{\theta})$ to approximate $\pi(a|s)$.
- $\boldsymbol{\theta}$: trainable parameters of the neural net.

Value network (critic):

- Use neural net $q(s, a; \mathbf{w})$ to approximate $Q_\pi(s, a)$.
- $\mathbf{w}$: trainable parameters of the neural net.

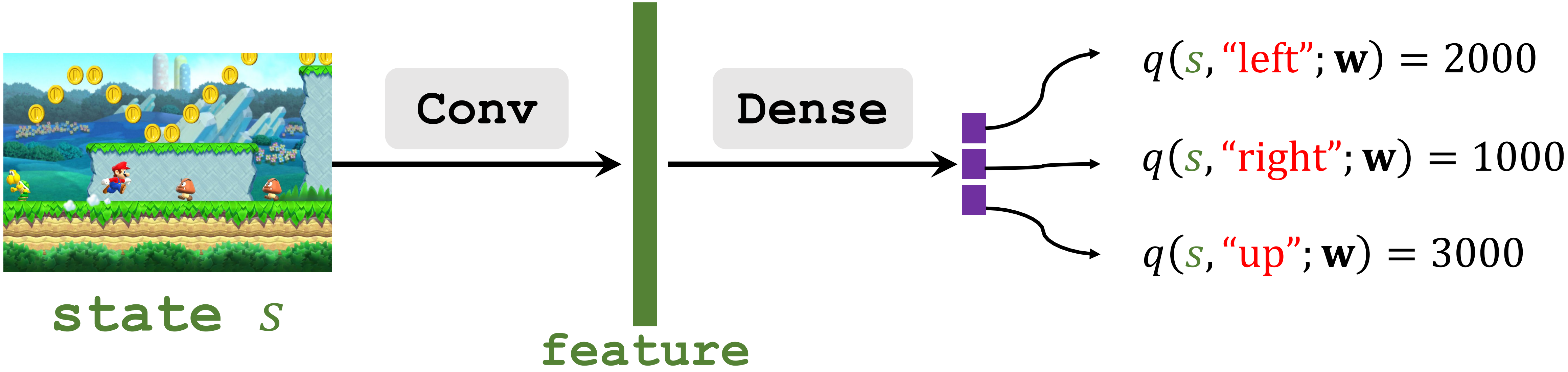
Policy Network (Actor): $\pi(a|s;\boldsymbol{\theta})$

- Input: state s, e.g., a screenshot of Super Mario.
- Output: probability distribution over the actions.
- Let $\mathcal{A}$ be the set all actions, e.g., $\mathcal{A} = \{\text{“left”}, \text{“right”}, \text{“up”}\}$.
- $\sum_{a\in\mathcal{A}} \pi(a|s;\boldsymbol{\theta}) = 1$. (That is why we use softmax activation.)



Value Network (Critic): $q(s, a; \mathbf{w})$

- Inputs: state s.
- Output: action-values of all the actions.



Actor-Critic Method

policy network (actor)

value network (critic)

Train the Neural Networks

Train the networks

Definition: State-value function approximated using neural networks.

- $V(s; \boldsymbol{\theta}, \mathbf{w}) = \sum_a \pi(a|s; \boldsymbol{\theta}) \cdot q(s, a; \mathbf{w})$.

Training: Update the parameters $\boldsymbol{\theta}$ and $\mathbf{w}$.

Train the networks

Definition: State-value function approximated using neural networks.

- $V(s; \boldsymbol{\theta}, \mathbf{w}) = \sum_a \pi(a|s; \boldsymbol{\theta}) \cdot q(s, a; \mathbf{w})$.

Training: Update the parameters $\boldsymbol{\theta}$ and $\mathbf{w}$.

- Update policy network $\pi(a|s; \boldsymbol{\theta})$ to increase the state-value $V(s; \boldsymbol{\theta}, \mathbf{w})$.
  - Actor gradually performs better.
  - Supervision is purely from the value network (critic).

Train the networks

Definition: State-value function approximated using neural networks.

- $V(s; \boldsymbol{\theta}, \mathbf{w}) = \sum_a \pi(a|s; \boldsymbol{\theta}) \cdot q(s, a; \mathbf{w})$.

Training: Update the parameters $\boldsymbol{\theta}$ and $\mathbf{w}$.

- Update policy network $\pi(a|s; \boldsymbol{\theta})$ to increase the state-value $V(s; \boldsymbol{\theta}, \mathbf{w})$.
  - Actor gradually performs better.
  - Supervision is purely from the value network (critic).
- Update value network $q(s, a; \mathbf{w})$ to better estimate the return.
  - Critic's judgement becomes more accurate.
  - Supervision is purely from the rewards.

Train the networks

Definition: State-value function approximated using neural networks.

- $V(s; \boldsymbol{\theta}, \mathbf{w}) = \sum_a \pi(a|s; \boldsymbol{\theta}) \cdot q(s, a; \mathbf{w})$.

Training: Update the parameters $\boldsymbol{\theta}$ and $\mathbf{w}$.

1. Observe the state s_t.
2. Randomly sample action a_t according to $\pi(\cdot\,|s_t; \boldsymbol{\theta}_t)$.
3. Perform a_t and observe new state s_{t+1} and reward r_t.
4. Update $\mathbf{w}$ (in value network) using temporal difference (TD).
5. Update $\boldsymbol{\theta}$ (in policy network) using policy gradient.

Update value network q using TD

- Compute $q(s_t, a_t; \mathbf{w}_t)$ and $q(s_{t+1}, a_{t+1}; \mathbf{w}_t)$.
- TD target: $y_t = r_t + \gamma \cdot q(s_{t+1}, a_{t+1}; \mathbf{w}_t)$.

Update value network q using TD

- Compute $q(s_t, a_t; \mathbf{w}_t)$ and $q(s_{t+1}, a_{t+1}; \mathbf{w}_t)$.
- TD target: $y_t = r_t + \gamma \cdot q(s_{t+1}, a_{t+1}; \mathbf{w}_t)$.

- Loss: $L(\mathbf{w}) = \frac{1}{2}\left[q(s_t, a_t; \mathbf{w}) - y_t\right]^2$.
- Gradient descent: $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \frac{\partial\, L(\mathbf{w})}{\partial\, \mathbf{w}} \Big|_{\mathbf{w}=\mathbf{w}_t}$.

Update policy network π using policy gradient

Definition: State-value function approximated using neural networks.

- $V(s;\boldsymbol{\theta},\mathbf{w}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot q(s,a;\mathbf{w})$.

Policy gradient: Derivative of $V(s_t;\boldsymbol{\theta},\mathbf{w})$ w.r.t. $\boldsymbol{\theta}$.

- Let $\mathbf{g}(a,\boldsymbol{\theta}) = \frac{\partial \log \pi(a|s,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot q(s_t,a;\mathbf{w})$.
- $\frac{\partial V(s;\boldsymbol{\theta},\mathbf{w}_t)}{\partial \boldsymbol{\theta}} = \mathbb{E}_A[\mathbf{g}(A,\boldsymbol{\theta})]$.

Update policy network π using policy gradient

Definition: State-value function approximated using neural networks.

- $V(s;\boldsymbol{\theta},\mathbf{w}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot q(s,a;\mathbf{w})$.

Policy gradient: Derivative of $V(s_t;\boldsymbol{\theta},\mathbf{w})$ w.r.t. $\boldsymbol{\theta}$.

- Let $\mathbf{g}(a,\boldsymbol{\theta}) = \frac{\partial \log \pi(a|s,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot q(s_t,a;\mathbf{w})$.
- $\frac{\partial\, V(s;\boldsymbol{\theta},\mathbf{w}_t)}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_A[\mathbf{g}(A,\boldsymbol{\theta})]$.

Algorithm: Update policy network using stochastic policy gradient.

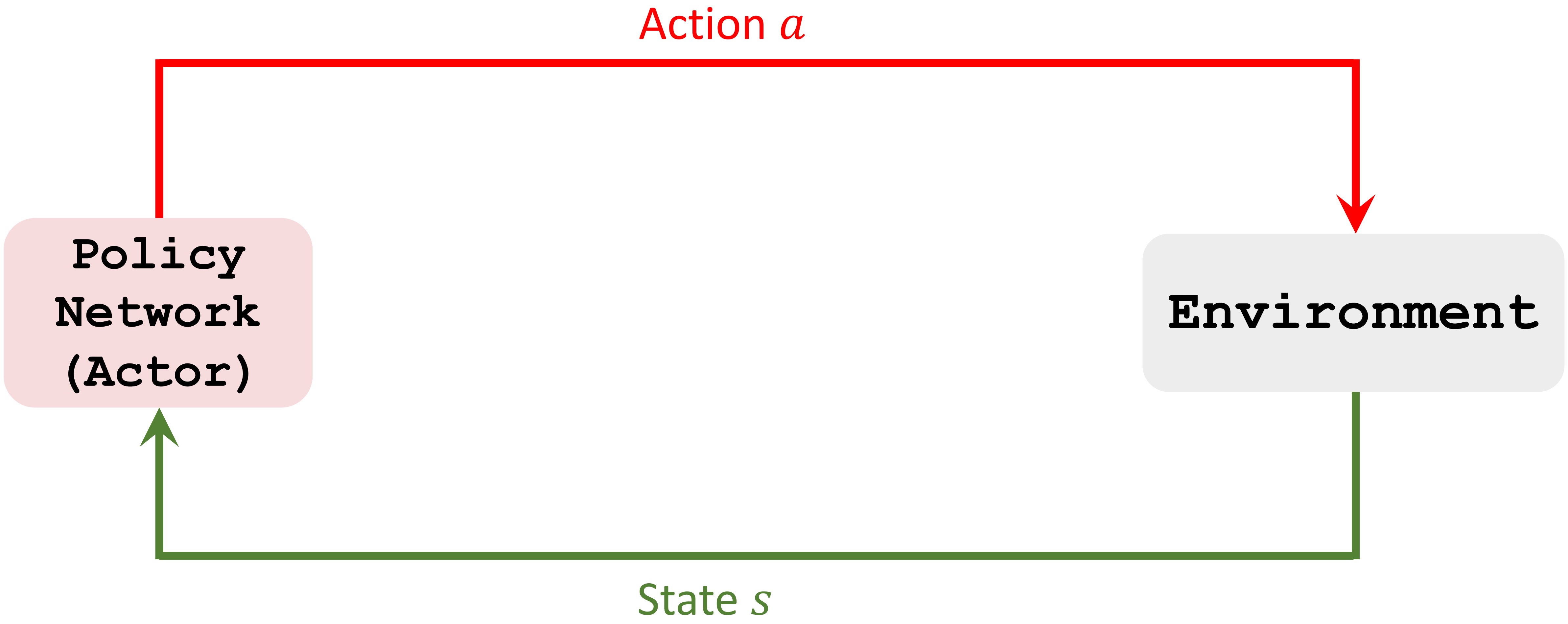
- Random sampling: $a \sim \pi(\cdot\,|s_t;\boldsymbol{\theta}_t)$. (Thus $\mathbf{g}(a,\boldsymbol{\theta})$ is unbiased.)
- Stochastic gradient ascent: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \beta \cdot \mathbf{g}(a,\boldsymbol{\theta}_t)$.

Actor-Critic Method

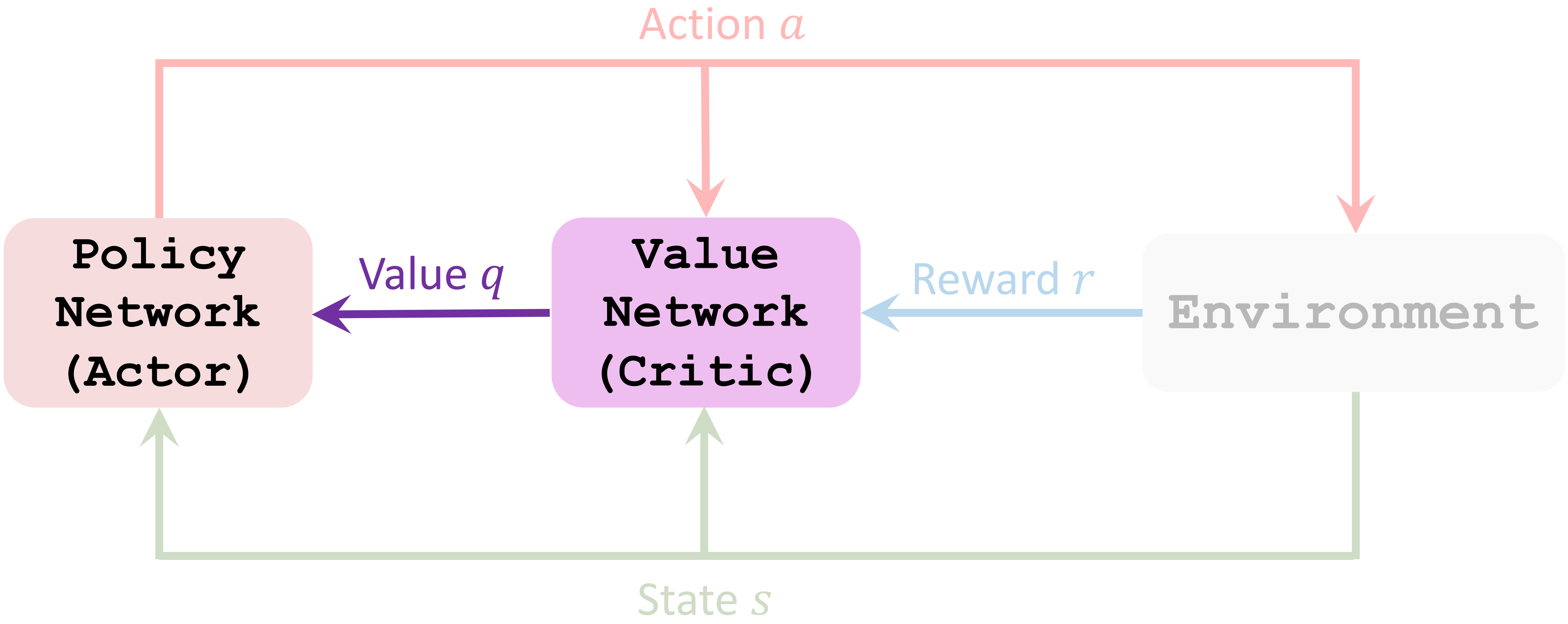
policy network (actor)

value network (critic)

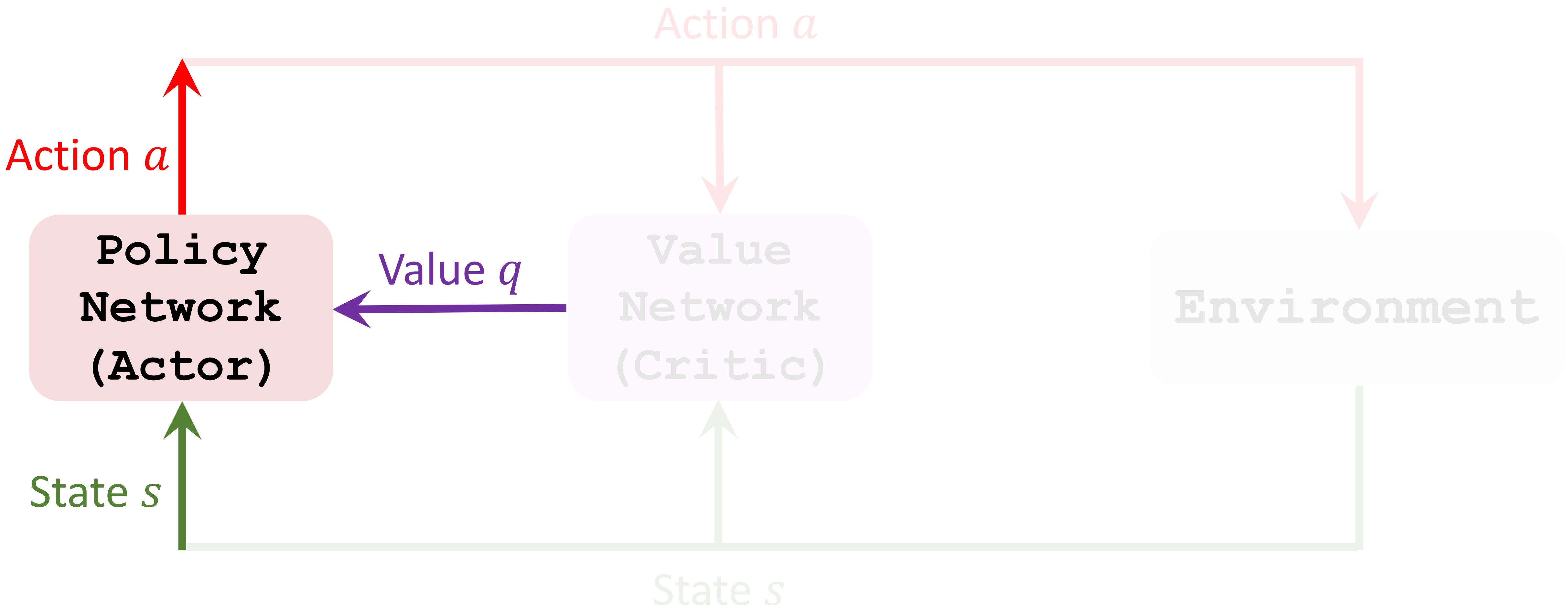
Actor-Critic Method



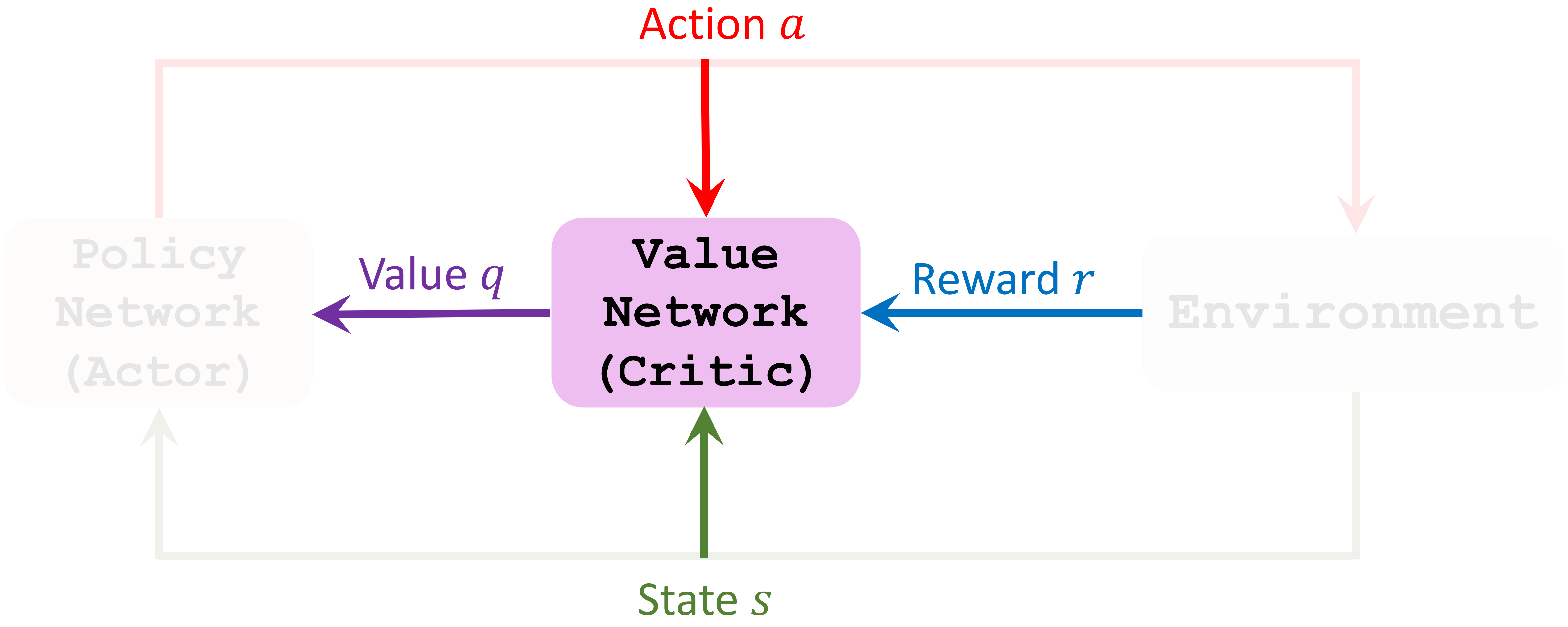
Actor-Critic Method



Actor-Critic Method: Update Actor



Actor-Critic Method: Update Critic



Summary of Algorithm

1. Observe state s_t and randomly sample $a_t \sim \pi(\cdot\,|s_t; \boldsymbol{\theta}_t)$.
2. Perform a_t; then environment gives new state s_{t+1} and reward r_t.
3. Randomly sample $\tilde{a}_{t+1} \sim \pi(\cdot\,|s_{t+1}; \boldsymbol{\theta}_t)$. (Do not perform $\tilde{a}_{t+1}$!)

Summary of Algorithm

1. Observe state s_t and randomly sample $a_t \sim \pi(\cdot \mid s_t; \boldsymbol{\theta}_t)$.
2. Perform a_t; then environment gives new state s_{t+1} and reward r_t.
3. Randomly sample $\tilde{a}_{t+1} \sim \pi(\cdot \mid s_{t+1}; \boldsymbol{\theta}_t)$. (Do not perform $\tilde{a}_{t+1}$!)
4. Evaluate value network: $q_t = q(s_t, a_t; \mathbf{w}_t)$ and $q_{t+1} = q(s_{t+1}, \tilde{a}_{t+1}; \mathbf{w}_t)$.
5. Compute TD error: $\delta_t = q_t - \underbrace{(r_t + \gamma \cdot q_{t+1})}_{\text{TD Target}}$.

Summary of Algorithm

1. Observe state s_t and randomly sample $a_t \sim \pi(\cdot\,|s_t;\boldsymbol{\theta}_t)$.
2. Perform a_t; then environment gives new state s_{t+1} and reward r_t.
3. Randomly sample $\tilde{a}_{t+1} \sim \pi(\cdot\,|s_{t+1};\boldsymbol{\theta}_t)$. (Do not perform $\tilde{a}_{t+1}$!)
4. Evaluate value network: $q_t = q(s_t, a_t; \mathbf{w}_t)$ and $q_{t+1} = q(s_{t+1}, \tilde{a}_{t+1}; \mathbf{w}_t)$.
5. Compute TD error: $\delta_t = q_t - (r_t + \gamma \cdot q_{t+1})$.
6. Differentiate value network: $\mathbf{d}_{w,t} = \frac{\partial q(s_t, a_t; \mathbf{w})}{\partial\, \mathbf{w}} \Big|_{\mathbf{w}=\mathbf{w}_t}$.
7. Update value network: $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \delta_t \cdot \mathbf{d}_{w,t}$.

Summary of Algorithm

1. Observe state s_t and randomly sample $a_t \sim \pi(\cdot\,|s_t;\boldsymbol{\theta}_t)$.
2. Perform a_t; then environment gives new state s_{t+1} and reward r_t.
3. Randomly sample $\tilde{a}_{t+1} \sim \pi(\cdot\,|s_{t+1};\boldsymbol{\theta}_t)$. (Do not perform $\tilde{a}_{t+1}$!)
4. Evaluate value network: $q_t = q(s_t, a_t; \mathbf{w}_t)$ and $q_{t+1} = q(s_{t+1}, \tilde{a}_{t+1}; \mathbf{w}_t)$.
5. Compute TD error: $\delta_t = q_t - (r_t + \gamma \cdot q_{t+1})$.
6. Differentiate value network: $\mathbf{d}_{w,t} = \frac{\partial q(s_t, a_t; \mathbf{w})}{\partial\, \mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}_t}$.
7. Update value network: $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \delta_t \cdot \mathbf{d}_{w,t}$.
8. Differentiate policy network: $\mathbf{d}_{\theta,t} = \frac{\partial \log \pi(a_t|s_t, \boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$.
9. Update policy network: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \beta \cdot q_t \cdot \mathbf{d}_{\theta,t}$.

Summary

Policy Network and Value Network

Definition: State-value function.

- $V_\pi(s) = \sum_a \pi(a|s) \cdot Q_\pi(s, a)$.

Definition: function approximation using neural networks.

- Approximate policy function $\pi(a|s)$ by $\pi(a|s; \mathbf{\theta})$ (actor).
- Approximate value function $Q_\pi(s, a)$ by $q(s, a; \mathbf{w})$ (critic).

Roles of Actor and Critic

During training

- Agent is controlled by policy network (actor): $a_t \sim \pi(\cdot\,|s_t; \boldsymbol{\theta})$.
- Value network q (critic) provides the actor with supervision.

Roles of Actor and Critic

During training

- Agent is controlled by policy network (actor): $a_t \sim \pi(\cdot \mid s_t; \boldsymbol{\theta})$.
- Value network q (critic) provides the actor with supervision.

After training

- Agent is controlled by policy network (actor): $a_t \sim \pi(\cdot \mid s_t; \boldsymbol{\theta})$.
- Value network q (critic) will not be used.

Training

Update the policy network (actor) by policy gradient.

- Seek to increase state-value: $V(s; \boldsymbol{\theta}, \mathbf{w}) = \sum_a \pi(a|s; \boldsymbol{\theta}) \cdot q(s, a; \mathbf{w})$.
- Compute policy gradient: $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_A \left[\frac{\partial \log \pi(A|s,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot q(s, A; \mathbf{w}) \right]$.
- Perform gradient ascent.

Training

Update the policy network (actor) by policy gradient.

- Seek to increase state-value: $V(s;\boldsymbol{\theta},\mathbf{w}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot q(s,a;\mathbf{w})$.
- Compute policy gradient: $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_A\left[\frac{\partial \log \pi(A|s,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot q(s,A;\mathbf{w})\right]$.
- Perform gradient ascent.

Update the value network (critic) by TD learning.

- Predicted action-value: $q_t = q(s_t, a_t; \mathbf{w})$.
- TD target: $y_t = r_t + \gamma \cdot q(s_{t+1}, a_{t+1}; \mathbf{w})$
- Gradient: $\frac{\partial (q_t - y_t)^2/2}{\partial\, \mathbf{w}} = (q_t - y_t) \cdot \frac{\partial\, q(s_t,a_t;\mathbf{w})}{\partial\, \mathbf{w}}$.
- Perform gradient descent.