Actor-Critic Methods

Shusen Wang

Value-Based Methods   **Actor-Critic Methods**   **Policy-Based Methods**
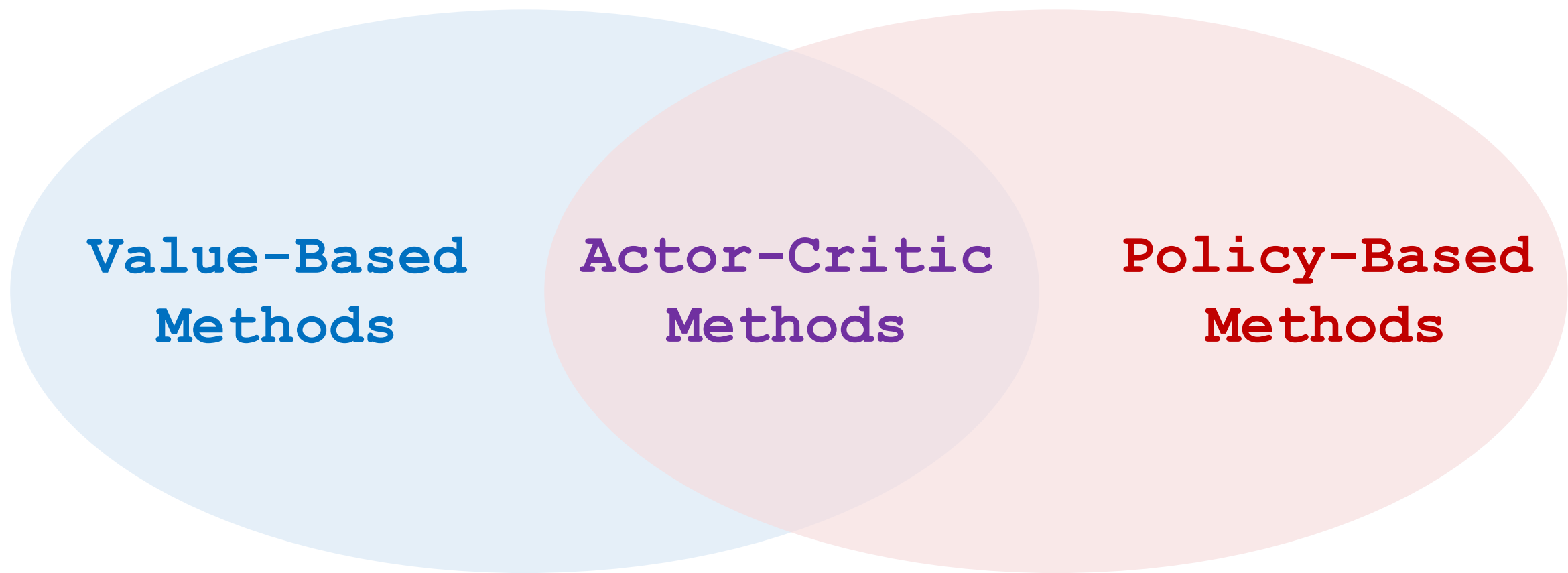
Value Network and Policy Network

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.

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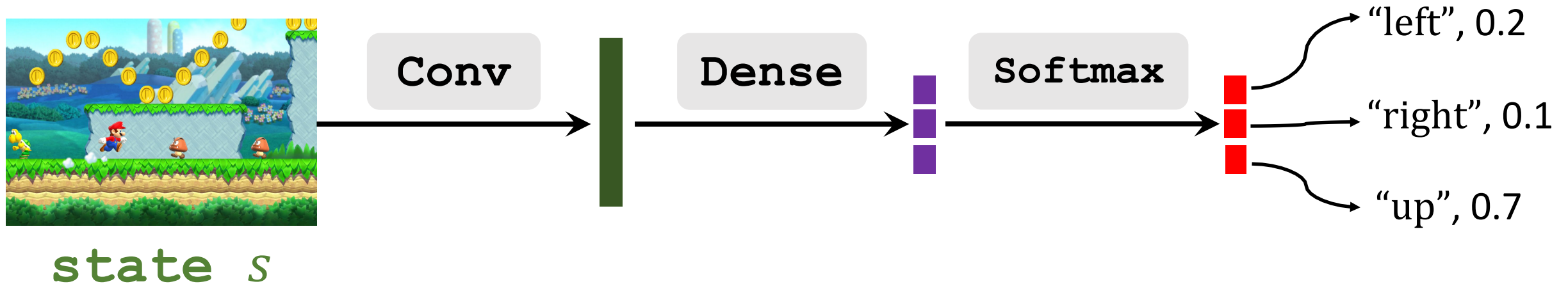
Policy network (actor):

- 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.)

state s

Conv → Dense → Softmax →

"left", 0.2
"right", 0.1
"up", 0.7

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

- Inputs: state s and action a.

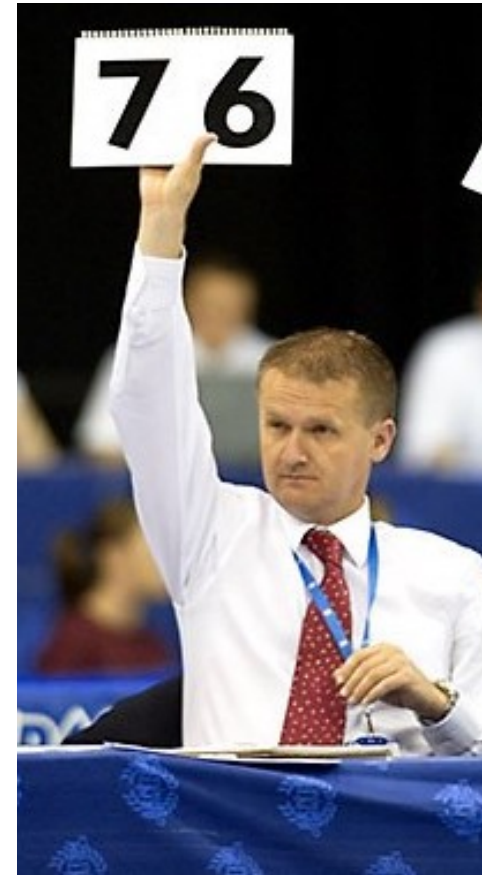- Output: approximate action-value (scalar).
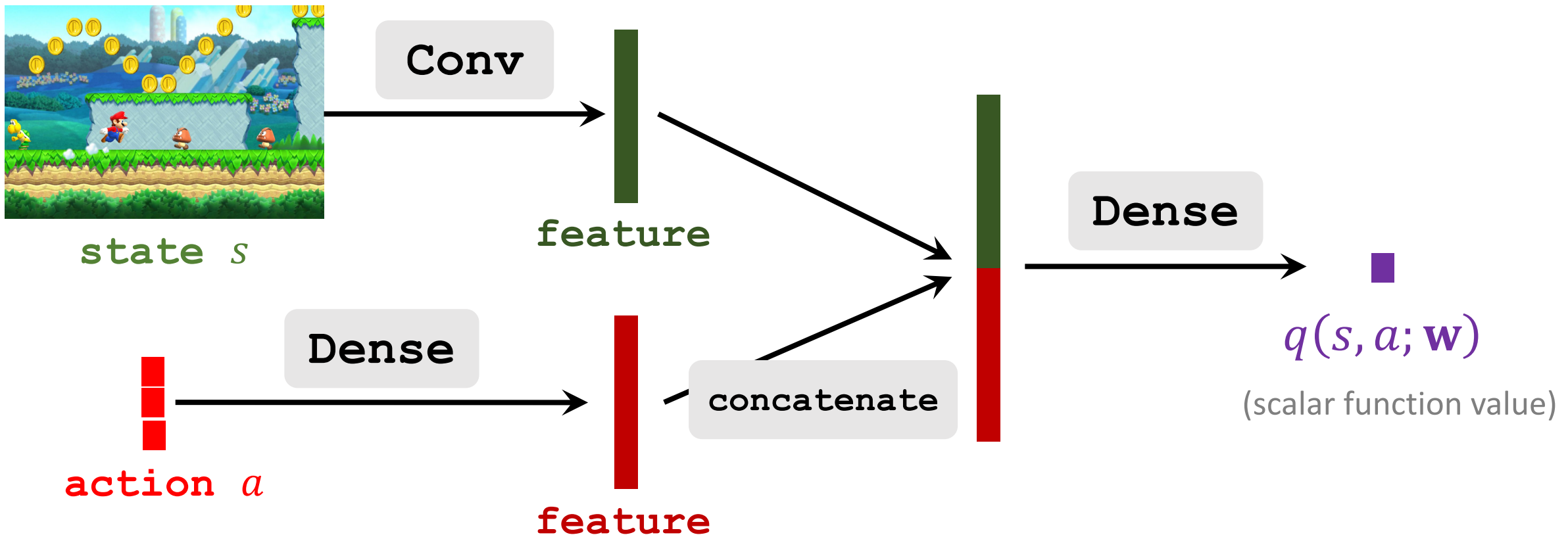
Actor-Critic Method

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 $\mathbf{g}(a, \boldsymbol{\theta}) = \dfrac{\partial \log \pi(a|s, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot q(s_t, a; \mathbf{w})$.

- $\dfrac{\partial V(s; \boldsymbol{\theta}, \mathbf{w}_t)}{\partial \boldsymbol{\theta}} = \mathbb{E}_A[\mathbf{g}(A, \boldsymbol{\theta})]$.

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 $\mathbf{g}(a, \boldsymbol{\theta}) = \dfrac{\partial \log \pi(a|s, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot q(s_t, a; \mathbf{w})$.

- $\dfrac{\partial V(s; \boldsymbol{\theta}, \mathbf{w}_t)}{\partial \boldsymbol{\theta}} = \mathbb{E}_A[\mathbf{g}(A, \boldsymbol{\theta})]$.

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: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \beta \cdot \mathbf{g}(a, \boldsymbol{\theta}_t)$.
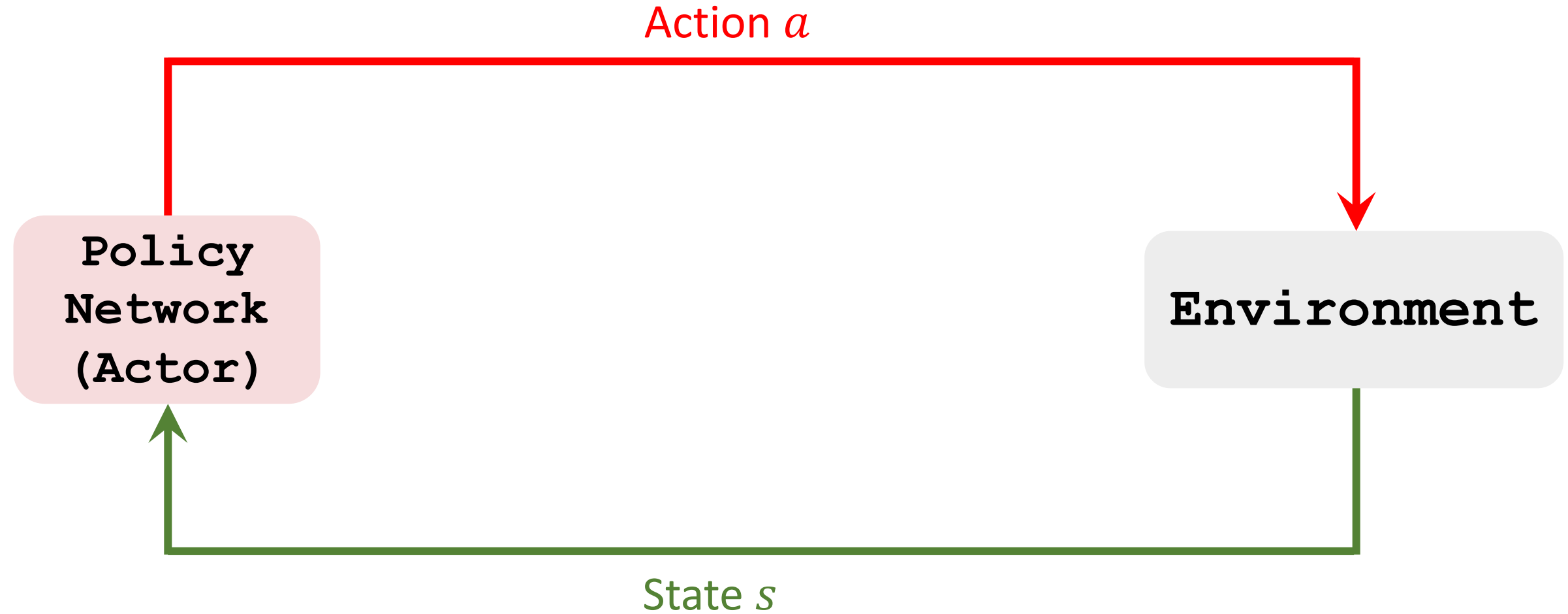
Actor-Critic Method

policy network (actor)

value network (critic)

Actor-Critic Method

Action a

Policy Network (Actor)

Environment

State s

Actor-Critic Method
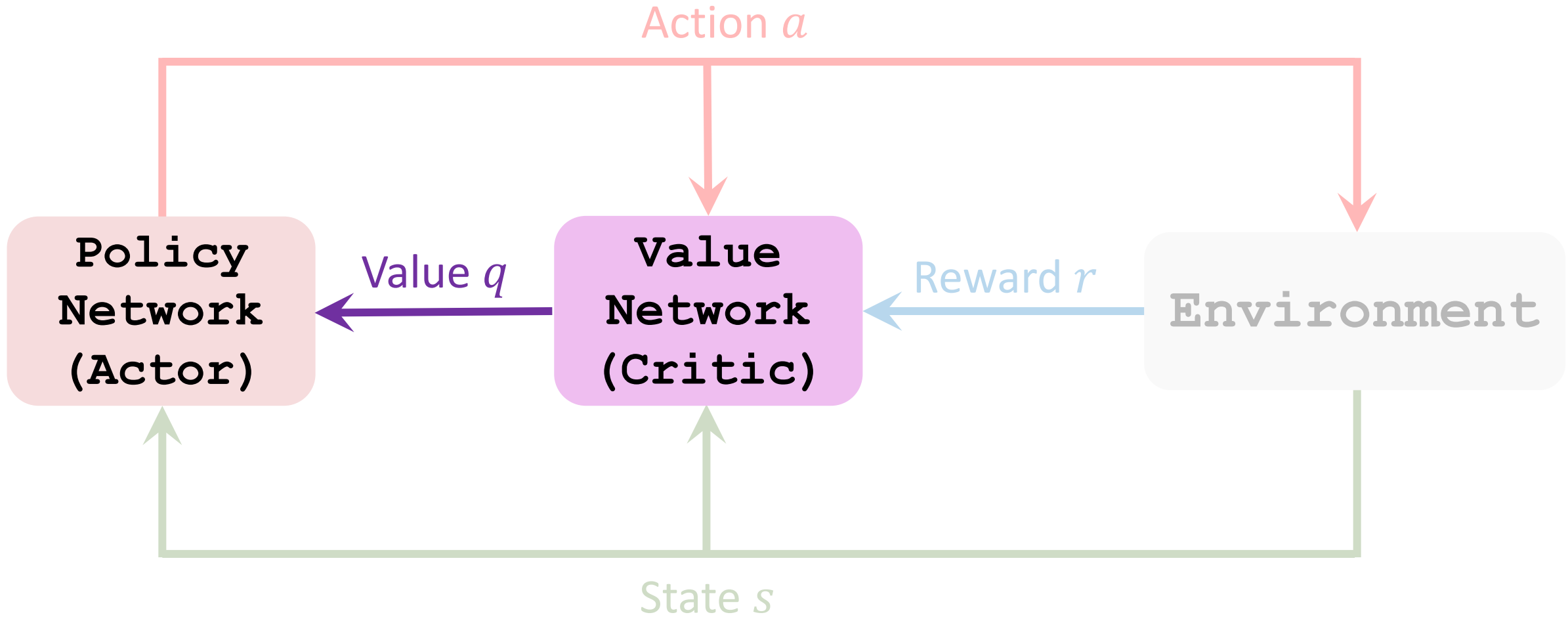
Actor-Critic Method: Update Actor
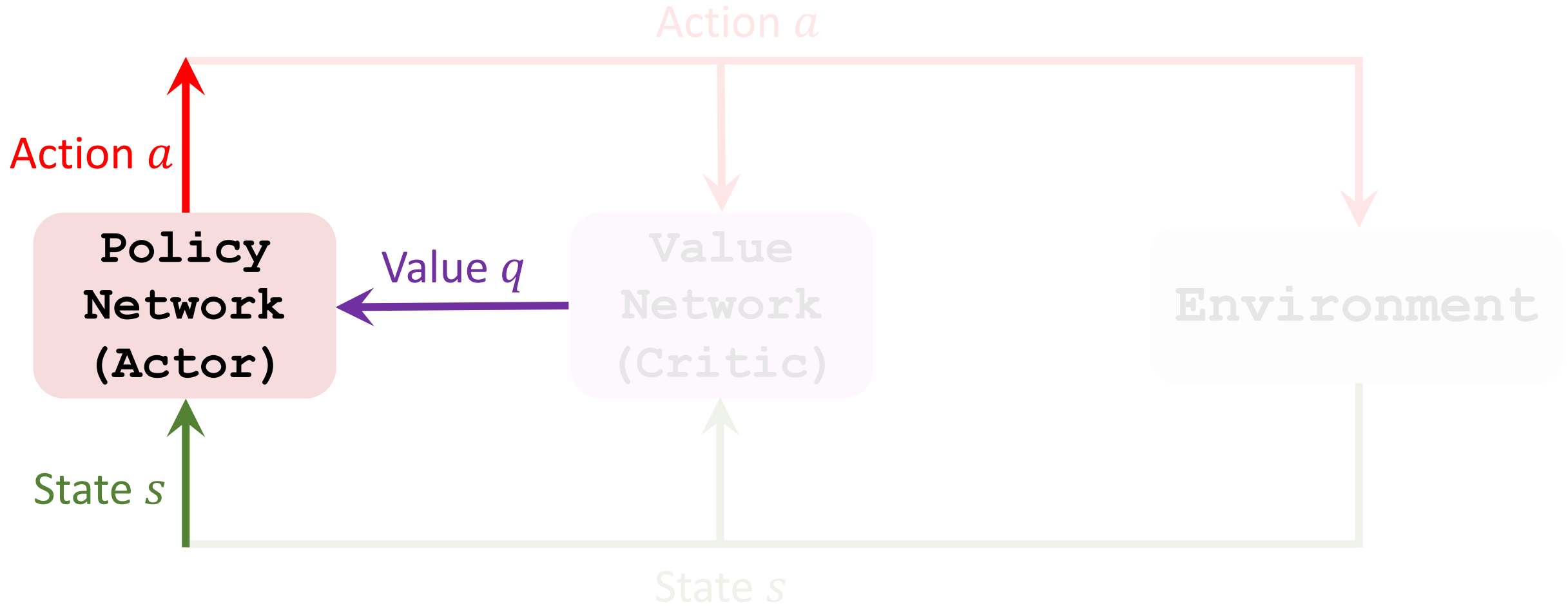
Actor-Critic Method: Update Critic
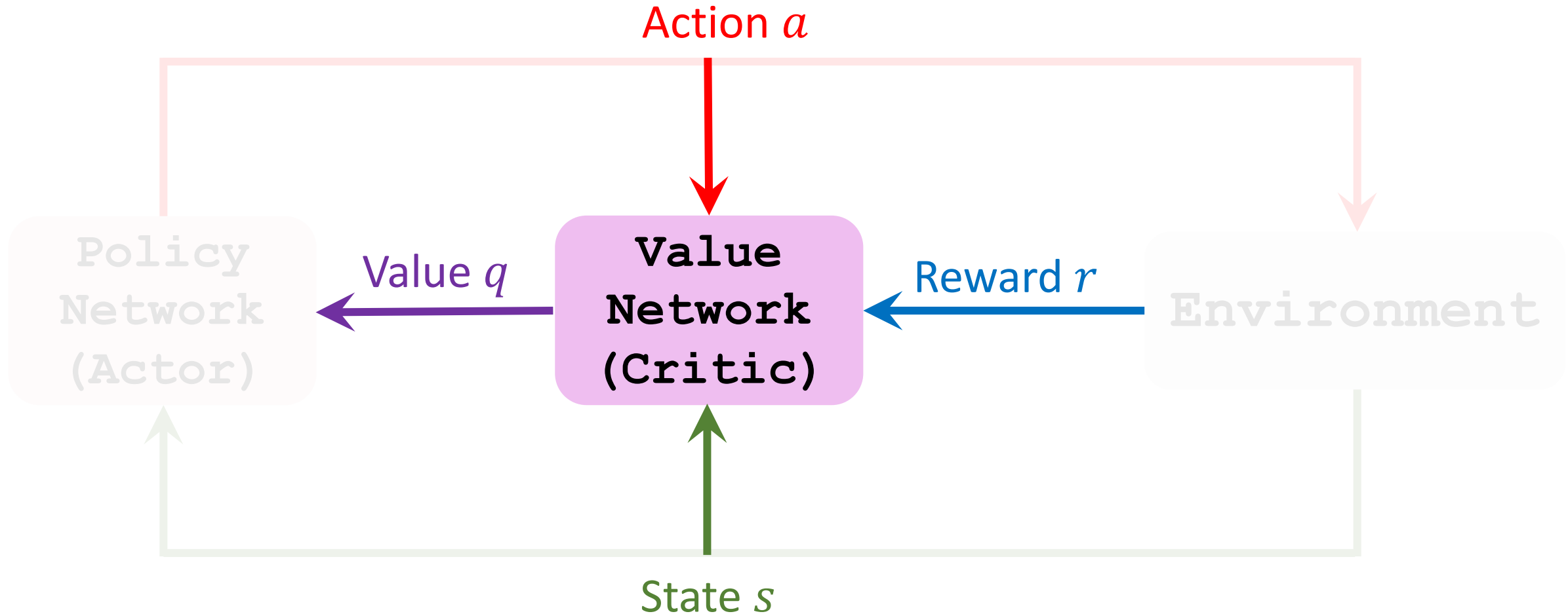
Action a

Policy Network (Actor)

Value q

Value Network (Critic)

Reward r

Environment

State s

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 $\tilde{a}_{t+1}$!)

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

TD Target

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

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: $\mathbf{d}_{w,t} = \dfrac{\partial q(s_t, a_t; \mathbf{w})}{\partial \mathbf{w}} \Big|_{\mathbf{w}=\mathbf{w}_t}$.

7. Update value network: $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \delta_t \cdot \mathbf{d}_{w,t}$.

8. Differentiate policy network: $\mathbf{d}_{\theta,t} = \dfrac{\partial \log \pi(a_t|s_t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$.

9. Update policy network: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \beta \cdot q_t \cdot \mathbf{d}_{\theta,t}$.

Summary 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: $\mathbf{d}_{w,t} = \dfrac{\partial q(s_t, a_t; \mathbf{w})}{\partial \mathbf{w}} \Big|_{\mathbf{w} = \mathbf{w}_t}$.

7. Update value network: $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \delta_t \cdot \mathbf{d}_{w,t}$.

8. Differentiate policy network: $\mathbf{d}_{\theta,t} = \dfrac{\partial \log \pi(a_t | s_t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \Big|_{\boldsymbol{\theta} = \boldsymbol{\theta}_t}$.

9. Update policy network: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \beta \cdot q_t \cdot \mathbf{d}_{\theta,t}$.

Summary 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: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \beta \cdot \cancel{q_t} \cdot \mathbf{d}_{\theta,t}$.
   δ_t

Policy Gradient with Baseline

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

   $\underbrace{\phantom{(r_t + \gamma \cdot q_{t+1})}}_{\text{Baseline}}$

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: $\mathbf{d}_{\theta,t} = \frac{\partial \log \pi(a_t \mid s_t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \Big|_{\boldsymbol{\theta} = \boldsymbol{\theta}_t}$.

9. Update policy network: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \beta \cdot \cancel{q_t} \cdot \mathbf{d}_{\theta,t}$.

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

Roles of Actor and Critic

- 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.

Roles of Actor and Critic

- 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.

- Agent is controlled by policy network (actor): $a_t \sim \pi(\cdot \,|\, s_t; \boldsymbol{\theta})$.

- 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: $\dfrac{\partial V(s; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbb{E}_A \left[\dfrac{\partial \log \pi(A|s, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot q(s, A; \mathbf{w}) \right]$.

- 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.

Thank you!