REINFORCE with Baseline

Shusen Wang

Policy Gradient with Baseline

Value Functions

- Discounted return:

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

- Action-value function:

$$Q_\pi(s_t, a_t) = \mathbb{E}[U_t \mid s_t, a_t].$$

- State-value function:

$$V_\pi(s_t) = \mathbb{E}_A[Q_\pi(s_t, A) \mid s_t].$$

Policy Gradient with Baseline

- Use policy network, $\pi(a|s;\boldsymbol{\theta})$, for controlling the agent.
- State-value function:

$$V_\pi(s) = \mathbb{E}_{A\sim\pi}[Q_\pi(s,A)]$$

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

- Policy gradient with baseline:

$$\frac{\partial\, V_\pi(s)}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{A\sim\pi}\left[\frac{\partial \ln \pi(A\mid s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \big(Q_\pi(s,A) - V_\pi(s_t)\big)\right].$$

Policy Gradient with Baseline

Policy gradient with baseline:

$$\frac{\partial\, V_\pi(s_t)}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{A_t \sim \pi}\left[\frac{\partial \ln \pi(A_t \mid s_t; \boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \left(Q_\pi(s_t, A_t) - V_\pi(s_t)\right)\right].$$

Policy Gradient with Baseline

Policy gradient with baseline:

$$\frac{\partial\, V_\pi(s_t)}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{A_t \sim \pi}\left[\frac{\partial \ln \pi(A_t \mid s_t;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \big(Q_\pi(s_t, A_t) - V_\pi(s_t)\big)\right].$$

$$= \mathbf{g}(A_t)$$

Policy Gradient with Baseline

Policy gradient with baseline:

$$\frac{\partial\, V_\pi(s_t)}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{A_t \sim \pi}\left[\frac{\partial \ln \pi(A_t \mid s_t; \boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \big(Q_\pi(s_t, A_t) - V_\pi(s_t)\big)\right].$$

$$= \mathbf{g}(A_t)$$

- Randomly sample $a_t \sim \pi(\cdot \mid s_t; \boldsymbol{\theta})$.
- Then $\mathbf{g}(a_t)$ is an unbiased estimation of the policy gradient.

Approximations

Policy gradient with baseline:

$$\frac{\partial\, V_\pi(s_t)}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{A_t \sim \pi}\left[\frac{\partial \ln \pi(A_t \mid s_t;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \big(Q_\pi(s_t, A_t) - V_\pi(s_t)\big)\right].$$

$$= \mathbf{g}(A_t)$$

Stochastic policy gradient with baseline:

$$\mathbf{g}(a_t) = \frac{\partial \ln \pi(a_t|s_t;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \big(Q_\pi(s_t, a_t) - V_\pi(s_t)\big).$$

Approximations

Stochastic policy gradient with baseline:

$$\mathbf{g}(a_t) \;=\; \frac{\partial \ln \pi(a_t|s_t;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \big(Q_\pi(s_t, a_t) - V_\pi(s_t)\big).$$

Approximations

Stochastic policy gradient with baseline:

$$\mathbf{g}(a_t) \;=\; \frac{\partial \ln \pi(a_t|s_t;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \big(Q_\pi(s_t, a_t) - V_\pi(s_t)\big)$$

Approximations

Stochastic policy gradient with baseline:

$$\mathbf{g}(a_t) \;=\; \frac{\partial \ln \pi(a_t|s_t;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \left(Q_\pi(s_t, a_t) - V_\pi(s_t)\right).$$

- Recall that $Q_\pi(s_t, a_t) = \mathbb{E}[U_t \mid s_t, a_t]$.
- Monte Carlo approximation to $Q_\pi(s_t, a_t) \approx u_t$ (REINFORCE):
  - Observing the trajectory: $s_t, a_t, r_t, s_{t+1}, a_{t+1}, r_{t+1}, \cdots, s_T, a_T, r_T$.
  - Compute return: $u_t = \sum_{i=t}^{T} \gamma^{i-t} \cdot r_i$.
  - u_t is unbiased Monte Carlo estimate of $Q_\pi(s_t, a_t)$.

Approximations

Stochastic policy gradient with baseline:

$$\mathbf{g}(a_t) = \frac{\partial \ln \pi(a_t|s_t;\boldsymbol{\theta})}{\partial \, \boldsymbol{\theta}} \cdot \left(Q_\pi(s_t, a_t) - V_\pi(s_t)\right)$$

- Approximate $V(s;\boldsymbol{\theta})$ by the value network, $v(s;\mathbf{w})$.

Approximations

Approximate policy gradient with baseline:

$$\frac{\partial\, V_\pi(s_t)}{\partial\, \boldsymbol{\theta}} \approx \mathbf{g}(a_t) \approx \frac{\partial \ln \pi(a_t|s_t;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \big(u_t - v(s_t;\mathbf{w})\big).$$

- Three approximations:

  1. Approximate expectation using one sample, a_t. (Monte Carlo.)
  2. Approximate $Q_\pi(s_t, a_t)$ by u_t. (Another Monte Carlo.)
  3. Approximate $V_\pi(s)$ by the value network, $v(s;\mathbf{w})$.

Summary of Approximations

$$\frac{\partial\, V_\pi(s_t)}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{A_t \sim \pi}\left[\frac{\partial\, \ln \pi(A_t \mid s_t; \boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \big(Q_\pi(s_t, A_t) - V_\pi(s_t)\big)\right].$$

$$\mathbf{g}(a_t) = \frac{\partial\, \ln \pi(a_t | s_t; \boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \big(Q_\pi(s_t, a_t) - V_\pi(s_t)\big).$$

Summary of Approximations

$$\frac{\partial\, V_\pi(s_t)}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{A_t \sim \pi}\left[\frac{\partial \ln \pi(A_t \mid s_t; \boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \big(Q_\pi(s_t, A_t) - V_\pi(s_t)\big)\right].$$

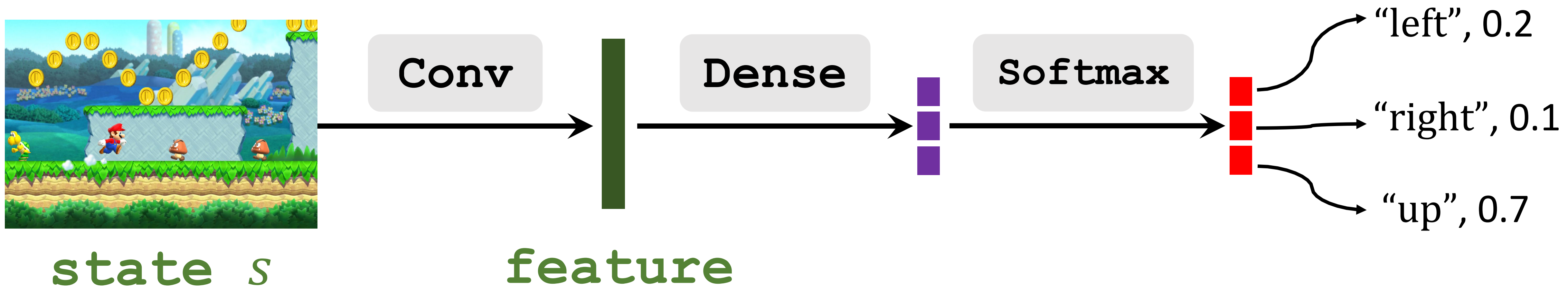
$$\mathbf{g}(a_t) = \frac{\partial \ln \pi(a_t|s_t;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \big(Q_\pi(s_t, a_t) - V_\pi(s_t)\big).$$

$$\mathbf{g}(a_t) \approx \frac{\partial \ln \pi(a_t|s_t;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \big(u_t - v(s_t; \mathbf{w})\big).$$

Policy and Value Networks

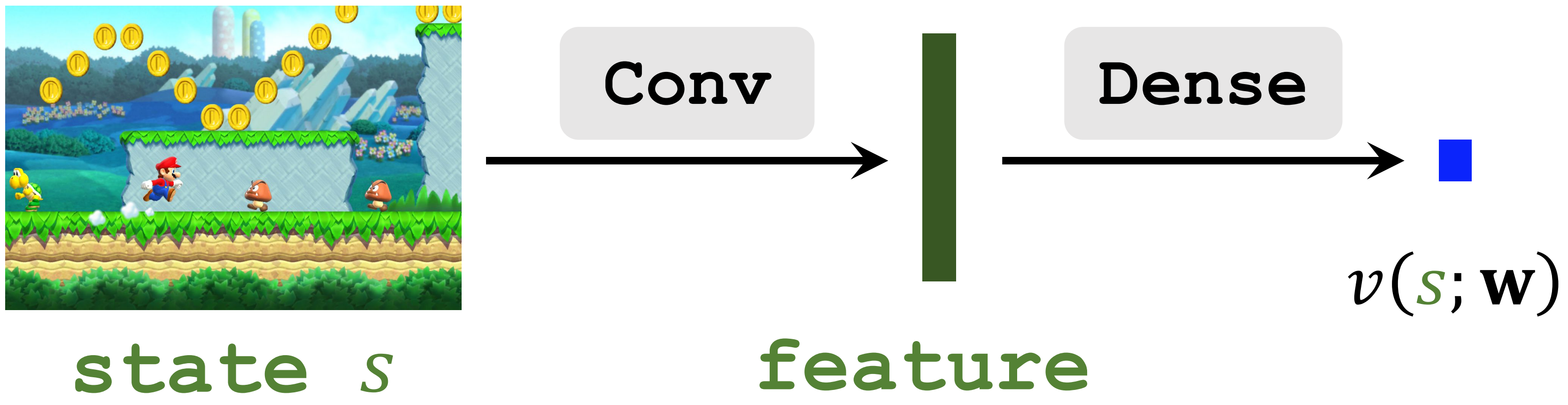
Policy Network

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

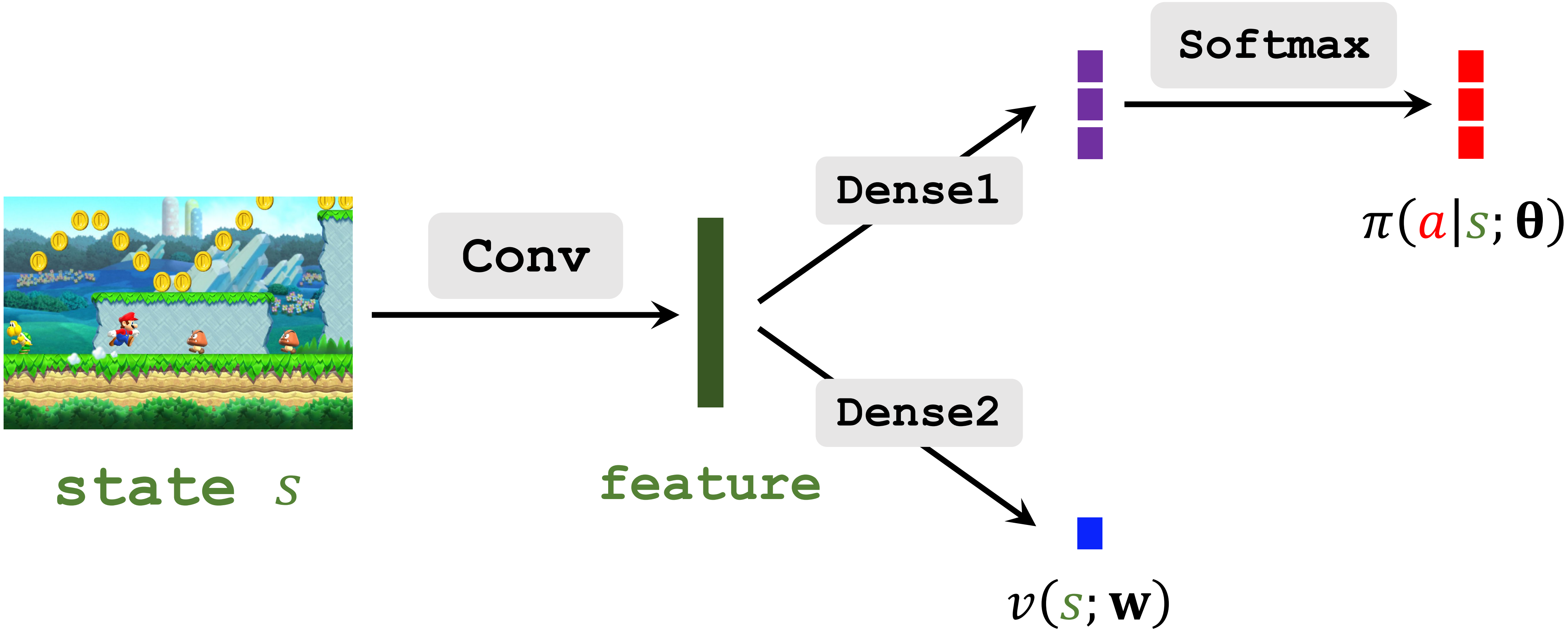


Value Network

Approximate state-value, $V_\pi(s)$, by value network, $v(s; \mathbf{w})$.



Parameter Sharing



REINFORCE with Baseline

Updating the policy network

Approximate policy gradient with baseline:

$$\frac{\partial\, V_\pi(s_t)}{\partial\, \boldsymbol{\theta}} \approx \frac{\partial \ln \pi(a_t|s_t;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \left(u_t - v(s_t;\mathbf{w})\right).$$

- Update policy network by policy gradient ascent:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boxed{\beta} \cdot \frac{\partial \ln \pi(a_t\mid s_t;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \left(u_t - v(s_t;\mathbf{w})\right).$$

Updating the value network

- Recall $v(s_t; \mathbf{w})$ is an approximation to $V_\pi(s_t) = \mathbb{E}[U_t \mid s_t]$.

Updating the value network

- Recall $v(s_t; \mathbf{w})$ is an approximation to $V(s_t; \boldsymbol{\theta}) = \mathbb{E}[U_t \mid s_t]$.
- Encourage $v(s_t; \mathbf{w})$ to approach u_t by decreasing:

$$\underline{\delta_t} = \underline{u_t} - \underline{v(s_t; \mathbf{w})}.$$

Updating the value network

- Recall $v(s_t; \mathbf{w})$ is an approximation to $V(s_t; \boldsymbol{\theta}) = \mathbb{E}[U_t \mid s_t]$.
- Encourage $v(s_t; \mathbf{w})$ to approach u_t by decreasing:

$$\delta_t = u_t - v(s_t; \mathbf{w}).$$

- Gradient: $\frac{\partial\, \delta_t^2/2}{\partial\, \mathbf{w}} = -\delta_t \cdot \frac{\partial\, v(s_t;\mathbf{w})}{\partial\, \mathbf{w}}.$
- Gradient descent:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \cdot (-\delta_t) \cdot \frac{\partial\, v(s_t;\mathbf{w})}{\partial\, \mathbf{w}}.$$

Summary of Algorithm

- Play a game to the end and observe the trajectory:

$$s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_n, a_n, r_n \,.$$

- Compute $\underline{u_t = \sum_{i=t}^{T} \gamma^{i-t} \cdot r_i}$ and $\underline{\delta_t = u_t - v(s_t; \mathbf{w})}$.

Summary of Algorithm

- Play a game to the end and observe the trajectory:

$$s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_n, a_n, r_n \, .$$

- Compute $u_t = \sum_{i=t}^{T} \gamma^{i-t} \cdot r_i$ and $\delta_t = u_t - v(s_t; \mathbf{w})$.
- Update the policy network by:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \beta \cdot \delta_t \cdot \frac{\partial \ln \pi(a_t \mid s_t; \boldsymbol{\theta})}{\partial \, \boldsymbol{\theta}}.$$

- Update the value network by:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha \cdot \delta_t \cdot \frac{\partial \, v(s_t; \mathbf{w})}{\partial \, \mathbf{w}}.$$

Summary of Algorithm

- Play a game to the end and observe the trajectory:

$$\boxed{s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_n, a_n, r_n\ .}$$

- Compute $u_t = \sum_{i=t}^{T} \gamma^{i-t} \cdot r_i$ and $\delta_t = u_t - v(s_t; \mathbf{w})$.
- Update the policy network by:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \beta \cdot \delta_t \cdot \frac{\partial \ln \pi(a_t \mid s_t; \boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}.$$

- Update the value network by:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha \cdot \delta_t \cdot \frac{\partial\, v(s_t; \mathbf{w})}{\partial\, \mathbf{w}}.$$

Summary of Algorithm

- Play a game to the end and observe the trajectory:

$$s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_n, a_n, r_n \,.$$

- Compute $u_t = \sum_{i=t}^{T} \gamma^{i-t} \cdot r_i$ and $\delta_t = u_t - v(s_t; \mathbf{w})$.
- Update the policy network by:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \beta \cdot \delta_t \cdot \frac{\partial \ln \pi(a_t \mid s_t; \boldsymbol{\theta})}{\partial \, \boldsymbol{\theta}}.$$

- Update the value network by:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha \cdot \delta_t \cdot \frac{\partial \, v(s_t; \mathbf{w})}{\partial \, \mathbf{w}}.$$

Repeat this procedure for all t from 1 to n.

Thank you!