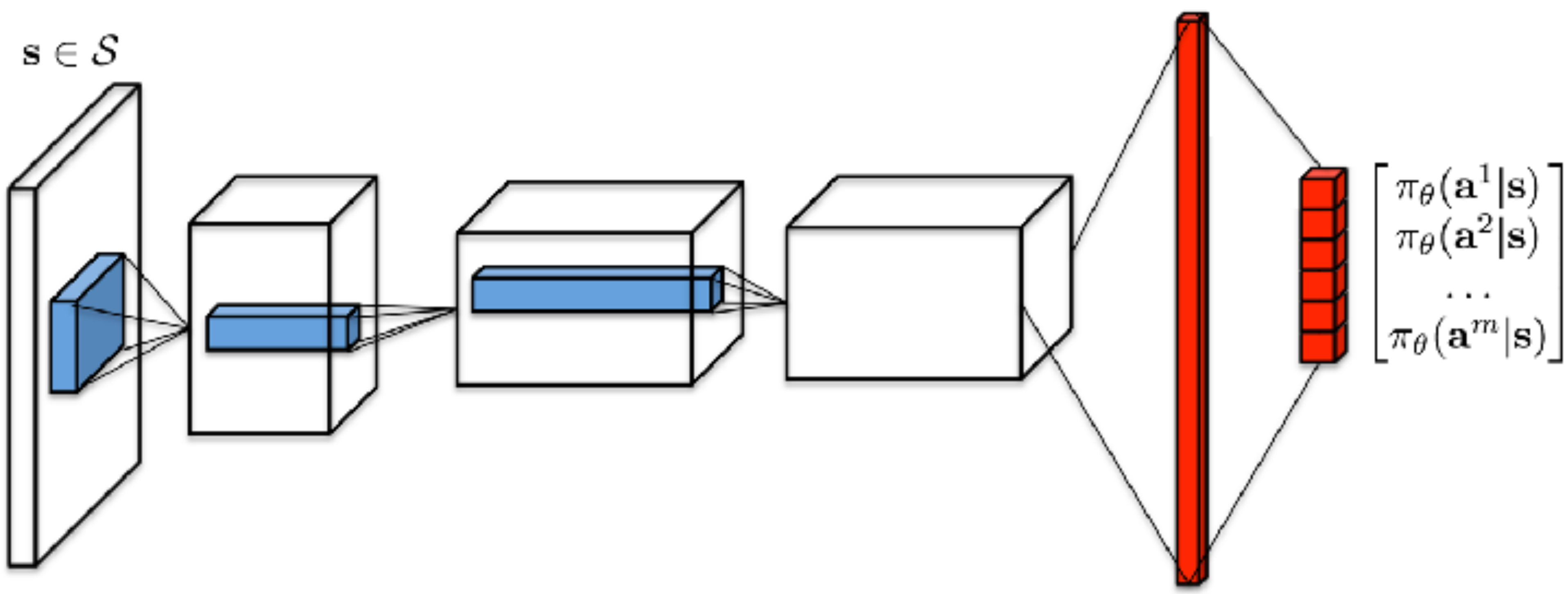
Policy gradient deep reinforcement learning algorithms

Oleksii Hrinchuk

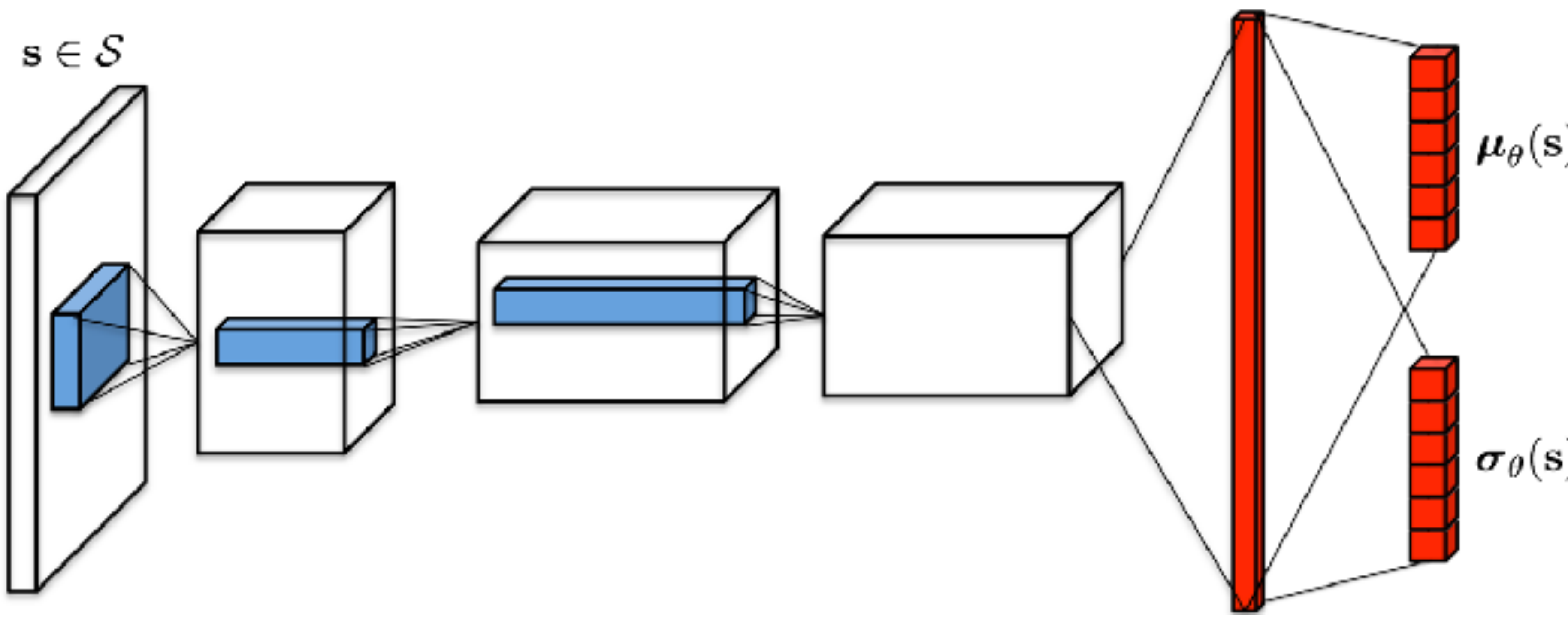
Policy gradient

- Parametric policy approximation $\pi_\theta(\mathbf{a}|\mathbf{s})$
- Discrete case: categorical distribution



Policy gradient

- Parametric policy approximation $\pi_\theta(\mathbf{a}|\mathbf{s})$
- Continuous case: parameters of some continuous probability distribution, e.g. Gaussian



$$\pi_\theta(\mathbf{a}|\mathbf{s}) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_\theta(\mathbf{s})|}} \exp\left(-\frac{1}{2}(\mathbf{a} - \boldsymbol{\mu}_\theta(\mathbf{s}))^\top \Sigma_\theta^{-1}(\mathbf{s})(\mathbf{a} - \boldsymbol{\mu}_\theta(\mathbf{s}))\right)$$

Policy gradient

- Distribution over trajectories $\tau = (\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \dots, \mathbf{s}_T)$

$$p_\theta(\tau) = p(\mathbf{s}_0) \prod_{t=0}^{T-1} \pi_\theta(\mathbf{a}_t | \mathbf{s}_t) p(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t)$$

- Optimization problem

$$J(\theta) = \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} r(\mathbf{s}_t, \mathbf{a}_t) \right] = \mathbb{E}_\tau \left[r(\tau) \right] \to \max_\theta$$

$$\theta \leftarrow \theta + \nabla_\theta J(\theta), \quad \nabla_\theta J(\theta) = ?$$

Direct policy differentiation

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_\tau \left[r(\tau) \right] = \nabla_\theta \int p_\theta(\tau) r(\tau) d\tau$$

$$= \int \nabla_\theta p_\theta(\tau) r(\tau) d\tau = \int p_\theta(\tau) \nabla_\theta \log p_\theta(\tau) r(\tau) d\tau$$

$$\boxed{\nabla_\theta \log p_\theta(\tau) = \frac{\nabla_\theta p_\theta(\tau)}{p_\theta(\tau)} \quad \Rightarrow \quad \nabla_\theta p_\theta(\tau) = p_\theta(\tau) \nabla_\theta \log p_\theta(\tau)}$$

Log derivative trick

Direct policy differentiation

$$\begin{aligned}\nabla_\theta J(\theta) &= \nabla_\theta \mathbb{E}_\tau\left[r(\tau)\right] = \nabla_\theta \int p_\theta(\tau) r(\tau) d\tau \\ &= \int \nabla_\theta p_\theta(\tau) r(\tau) d\tau = \int p_\theta(\tau) \nabla_\theta \log p_\theta(\tau) r(\tau) d\tau \\ &= \mathbb{E}_\tau\left[\nabla_\theta \log p_\theta(\tau) r(\tau)\right] \\ &= \mathbb{E}_\tau\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \sum_{t=0}^{T-1} r(\mathbf{s}_t, \mathbf{a}_t)\right]\end{aligned}$$

$$\begin{aligned}\nabla_\theta \log p_\theta(\tau) &= \cancel{\nabla_\theta \log p(\mathbf{s}_0)} + \nabla_\theta \sum_{t=0}^{T-1} \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \\ &\quad + \cancel{\nabla_\theta \sum_{t=0}^{T-1} \log p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)} = \nabla_\theta \sum_{t=0}^{T-1} \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)\end{aligned}$$

Estimating gradients in practice

- In practice we estimate expectations by averaging over samples

1. Sample N trajectories $\{(\mathbf{s}_{i,0}, \mathbf{a}_{i,0}, \mathbf{s}_{i,1}, \ldots, \mathbf{s}_{i,T})\}_{i=1}^{N}$

2. For each trajectory calculate

$$\hat{\nabla}_\theta J_i(\theta) = \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) \sum_{t=0}^{T-1} r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})$$

3. Estimate gradient as $\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \hat{\nabla}_\theta J_i(\theta)$

Causality variance reduction

- Objective gradient

$$\nabla_\theta J(\theta) = \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \sum_{t'=0}^{T-1} r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) \right]$$

- Causality: policy at time t' can not affect reward at time t where $t < t'$

$$\nabla_\theta J(\theta) = \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \sum_{t'=t}^{T-1} r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) \right]$$

Reward-to-go

$$\nabla_\theta J(\theta) = \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) Z^\pi(\mathbf{s}_t, \mathbf{a}_t) \right]$$

Baseline variance reduction

- Subtracting state-dependent baseline does not change the expectation

$$\nabla_\theta J(\theta) = \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) Z^\pi(\mathbf{s}_t, \mathbf{a}_t) \right]$$

$$\nabla_\theta J(\theta) = \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \left(Z^\pi(\mathbf{s}_t, \mathbf{a}_t) - b(\mathbf{s}_t) \right) \right]$$

Advantage Actor-Critic

- Single-sample estimate of the objective gradient

$$\hat{\nabla}_\theta J(\theta) = \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \left(Z^\pi(\mathbf{s}_t, \mathbf{a}_t) - b(\mathbf{s}_t)\right)$$

Advantage Actor-Critic

- Single-sample estimate of the objective gradient

$$\hat{\nabla}_\theta J(\theta) \approx \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \left(Q(\mathbf{s}_t, \mathbf{a}_t) - b(\mathbf{s}_t) \right)$$

Advantage Actor-Critic

- Single-sample estimate of the objective gradient

$$\hat{\nabla}_\theta J(\theta) \approx \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \left(Q(\mathbf{s}_t, \mathbf{a}_t) - V(\mathbf{s}_t) \right)$$

Advantage Actor-Critic

- Single-sample estimate of the objective gradient

$$\hat{\nabla}_\theta J(\theta) \approx \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) {\color{red} A(\mathbf{s}_t, \mathbf{a}_t)}$$

Advantage Actor-Critic

- Single-sample estimate of the objective gradient

$$\hat{\nabla}_\theta J(\theta) \approx \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) A(\mathbf{s}_t, \mathbf{a}_t)$$

- Idea: in addition to fitting policy $\pi_\theta(\mathbf{a}|\mathbf{s})$, fit advantage function estimator $A_\phi(\mathbf{s}, \mathbf{a})$ to reduce variance

Advantage Actor-Critic

- Single-sample estimate of the objective gradient

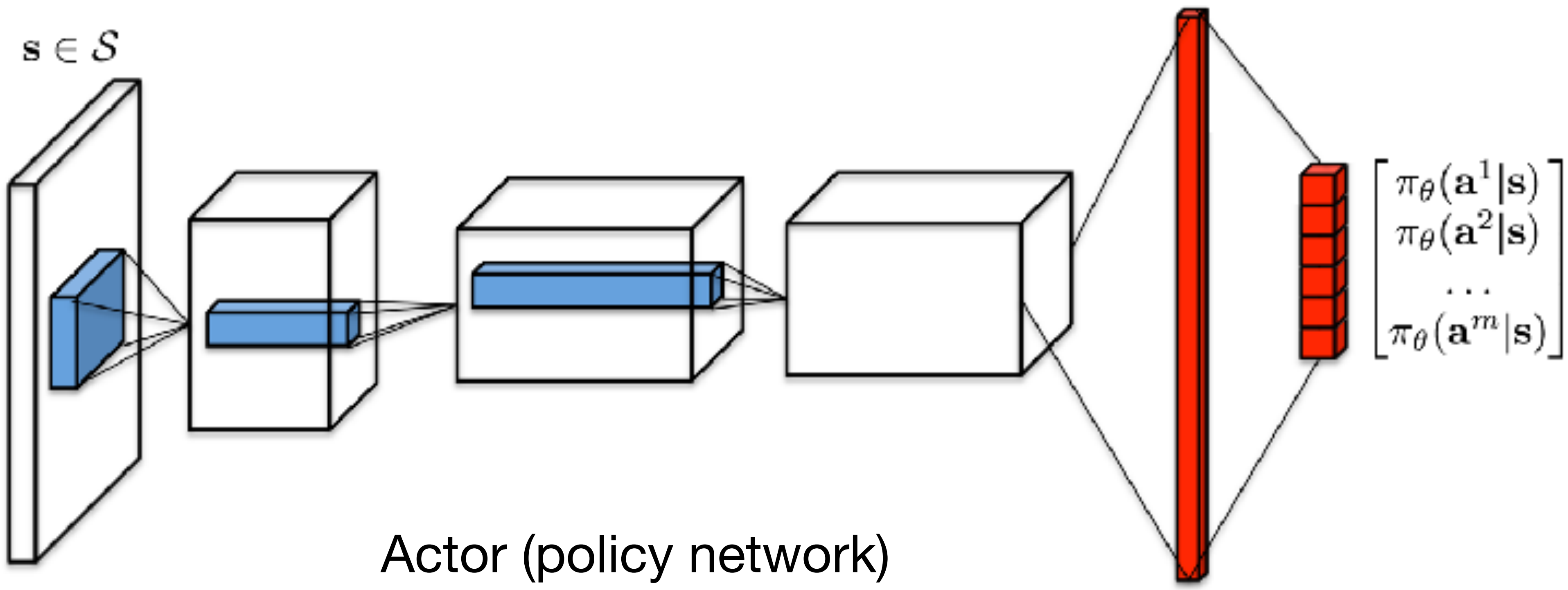
$$\hat{\nabla}_\theta J(\theta) \approx \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) A(\mathbf{s}_t, \mathbf{a}_t)$$

- Idea: in addition to fitting policy $\pi_\theta(\mathbf{a}|\mathbf{s})$, fit advantage function estimator $A_\phi(\mathbf{s}, \mathbf{a})$ to reduce variance

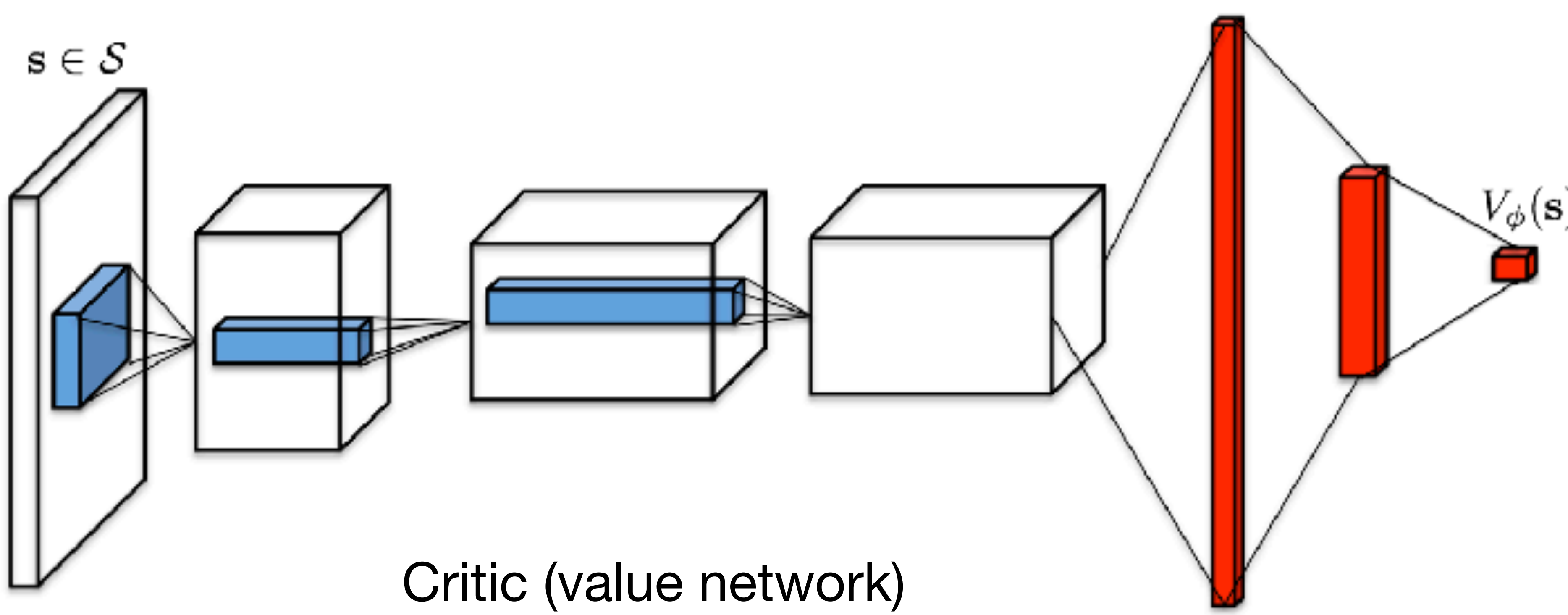
- In practice, people usually fit value function $V_\phi(\mathbf{s}_t)$

$$A_\phi(\mathbf{s}_t, \mathbf{a}_t) \approx r(\mathbf{s}_t, \mathbf{a}_t) + \gamma V_\phi(\mathbf{s}_{t+1}) - V_\phi(\mathbf{s}_t)$$

Advantage Actor-Critic

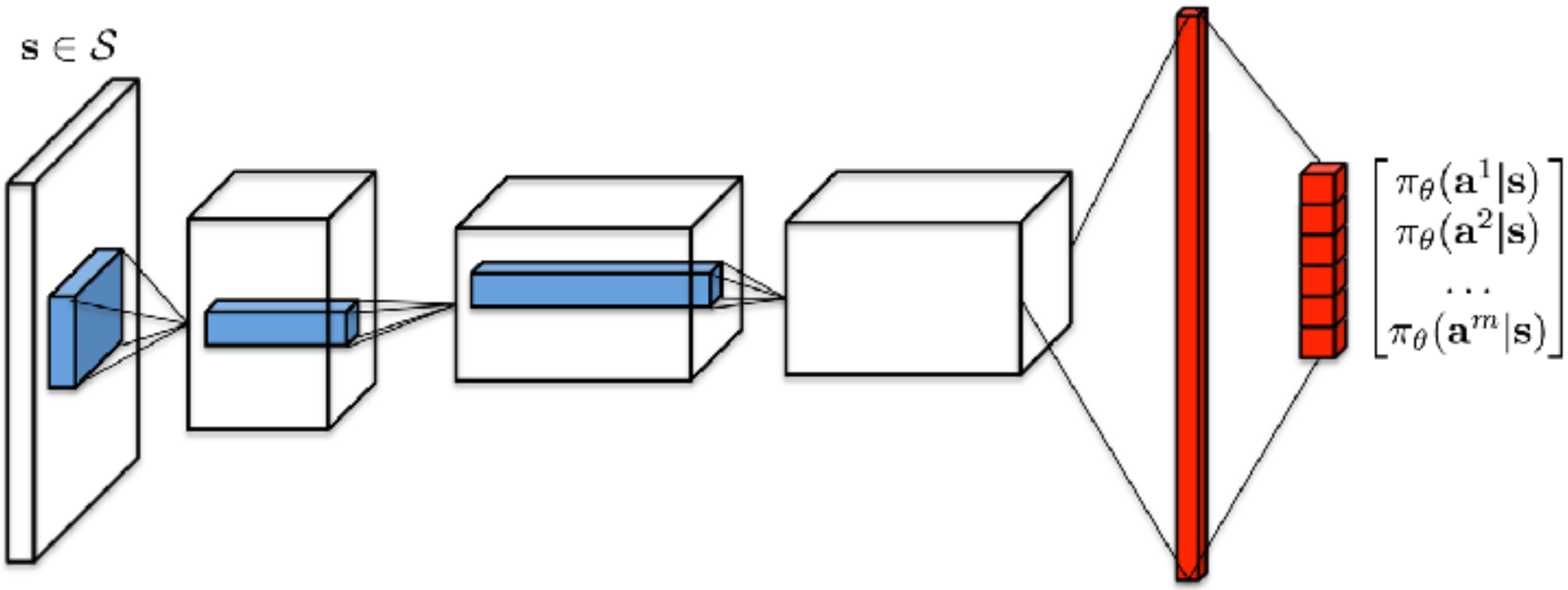


Actor (policy network)



Critic (value network)

Actor (policy network)

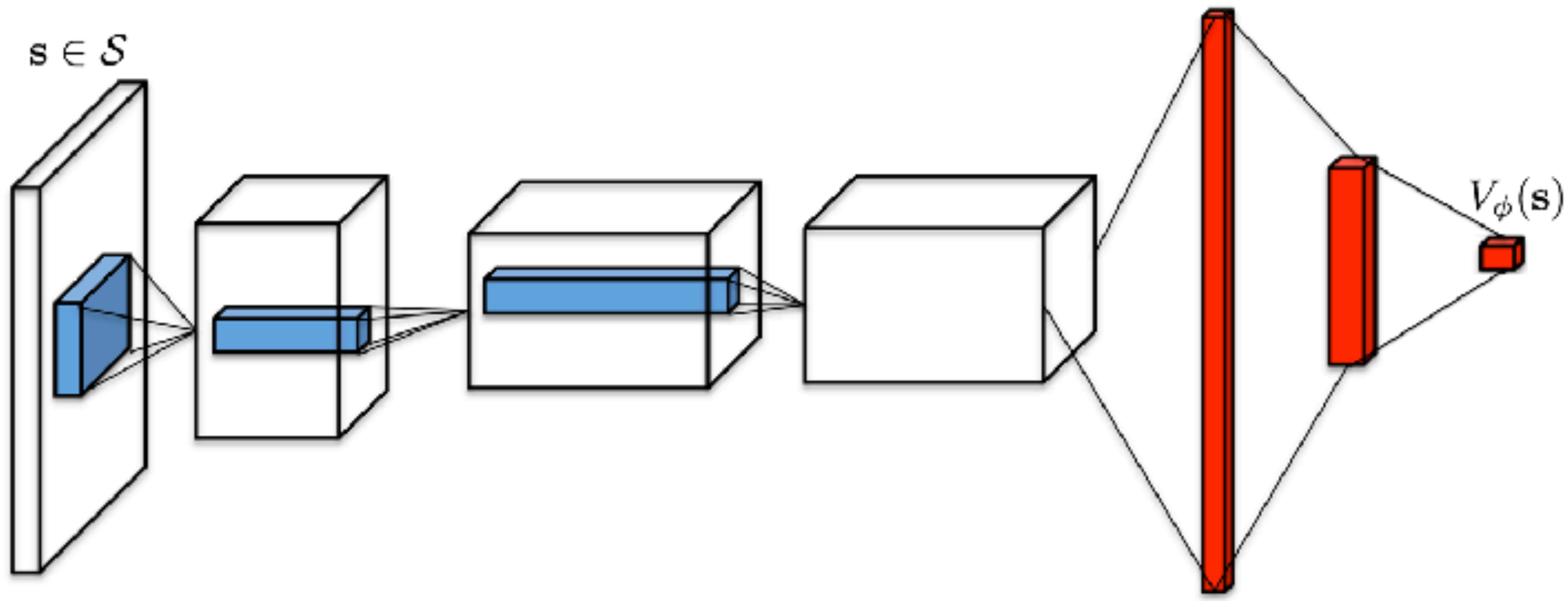


$$\mathcal{L}_\theta(\tau) \approx \sum_{t=1}^{T-1} \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \left[\mathbf{r}_t + \gamma V_\phi(\mathbf{s}_{t+1}) - V_\phi(\mathbf{s}_t)\right]$$

Pseudo-loss: cross entropy

$$\hat{\nabla}_\theta J(\theta) \approx \sum_{t=1}^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \left[\mathbf{r}_t + \gamma V_\phi(\mathbf{s}_{t+1}) - V_\phi(\mathbf{s}_t)\right]$$

Critic (value network)



$$\mathcal{L}_\phi(\mathbf{d}_t) = \left[V_\phi(\mathbf{s}_t) - (\mathbf{r}_t + \gamma V_\phi(\mathbf{s}_{t+1}))\right]^2$$

$$\hat{\nabla}_\phi \mathcal{L}_\phi(\mathbf{d}_t) \approx \nabla_\phi V_\phi(\mathbf{s}_t)\left[V_\phi(\mathbf{s}_t) - (\mathbf{r}_t + \gamma V_\phi(\mathbf{s}_{t+1}))\right]$$