Actor-Critic Methods

Chen-Yu Wei

Review: Full-Information Policy Learning in MDPs

$$\theta_{k+1} = \operatorname*{argmax}_{\theta} \left(\underbrace{V^{\pi_\theta}(\rho) - V^{\pi_{\theta_k}}(\rho)}_{} - \frac{1}{\eta} D(\theta, \theta_k) \right)$$

$$\approx \sum_{s,a} d_\rho^{\pi_{\theta_k}}(s)\big(\pi_\theta(a|s) - \pi_{\theta_k}(a|s)\big) Q^{\pi_{\theta_k}}(s,a) = \mathbb{E}_{(s_i,a_i) \sim \pi_{\theta_k}} \left[\frac{\pi_\theta(a_i|s_i) - \pi_{\theta_k}(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \boxed{Q^{\pi_{\theta_k}}(s_i, a_i)} \right]$$

$$\approx \underbrace{(\theta - \theta_k)^\top \sum_{s,a} d_\rho^{\pi_{\theta_k}}(s) \left(\nabla_\theta \pi_\theta(a|s) \Big|_{\theta=\theta_k} \right) Q^{\pi_{\theta_k}}(s,a)}_{}$$

$$= \mathbb{E}_{(s_i,a_i)} \left[\frac{\nabla_\theta \pi_\theta(a_i|s_i)|_{\theta=\theta_k}}{\pi_{\theta_k}(a_i|s_i)} \boxed{Q^{\pi_{\theta_k}}(s_i, a_i)} \right]$$

PG/NPG: Estimate them using the empirical sum of reward in the trajectory (i.e., Monte Carlo estimator)

We can also use other estimators to balance bias and variance

Actor-Critic Methods

Use value function approximation to estimate $Q^{\pi_{\theta_k}}(s_i, a_i)$ or $A^{\pi_{\theta_k}}(s_i, a_i)$

Use $V_\phi(s)$: $\approx V^{\pi_{\theta_k}}(s)$ $\quad \min_\phi \mathbb{E}_{(s,r,s')\sim\pi_{\theta_k}}\left[\left(V_\phi(s) - r - \gamma V_{\phi_k}(s')\right)^2\right]$

Use $Q_\phi(s, a)$: $\approx Q^{\pi_{\theta_k}}(s,a)$ $\quad \min_\phi \mathbb{E}_{(s,a,r,s',a')\sim\pi_{\theta_k}}\left[\left(Q_\phi(s,a) - r - \gamma Q_{\phi_k}(s',a')\right)^2\right]$

Possible estimators for $A^{\pi_{\theta_k}}(s, a)$:

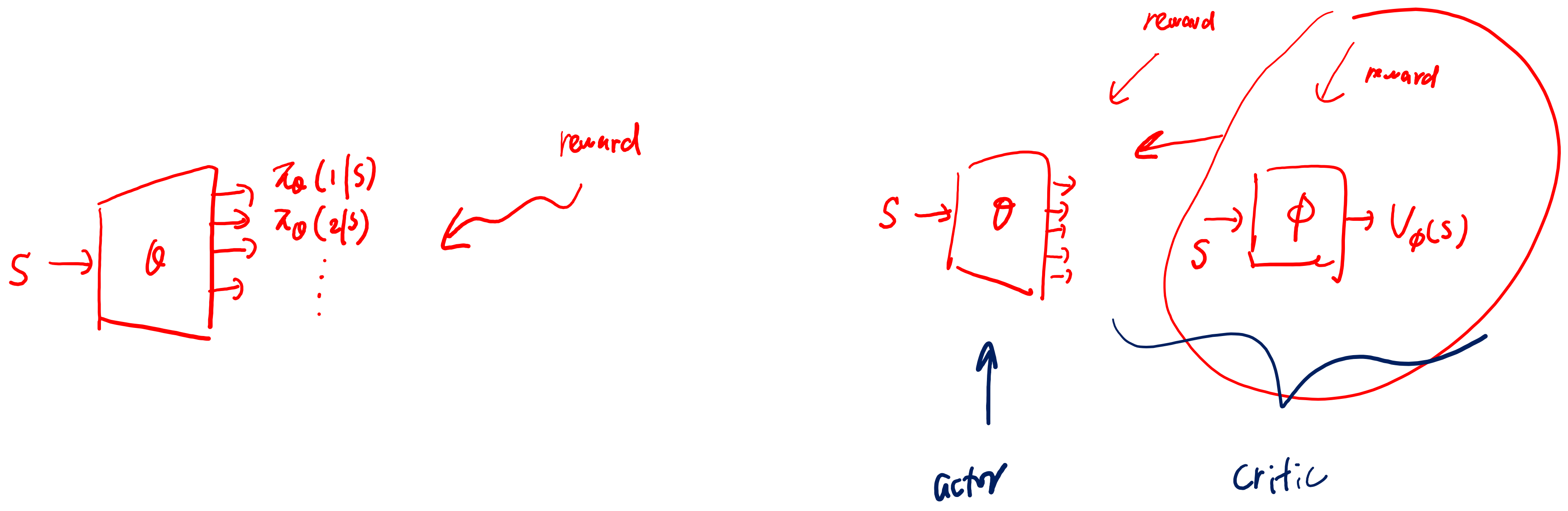
Let $(s_1, a_1, r_1, s_2, a_2, r_2 \ldots)$ be a trajectory starting from $s_1 = s, a_1 = a$

$$Q_\phi(s_1, a_1) - \mathbb{E}_{a'\sim\pi_{\theta_k}(\cdot|s)}[Q_\phi(s_1, a')]$$

λ

$r_1 + \gamma V_\phi(s_2) - V_\phi(s_1)$	$r_1 + \gamma Q_\phi(s_2, a_2) - \mathbb{E}_{a'\sim\pi_{\theta_k}(\cdot\|s)}[Q_\phi(s_1, a')]$
$r_1 + \gamma r_2 + \gamma^2 V_\phi(s_3) - V_\phi(s_1)$	$r_1 + \gamma r_2 + \gamma^2 Q_\phi(s_3, a_3) - \mathbb{E}_{a'\sim\pi_{\theta_k}(\cdot\|s)}[Q_\phi(s_1, a')]$
$\vdots$	$\vdots$

Pure Policy-Based Methods vs. Actor-Critic Methods



Actor-Critic with Q_ϕ

(find π^*)

(given π)

(off-policy)

Q-learning: $Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left[r + \max_{a'} Q(s',a')\right]$ $\Rightarrow Q^*$

TD-learning: $Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left[r + \sum_{a'} \pi(a'|s')Q(s',a')\right]$ $\Rightarrow Q^\pi$

(on-policy)

For $k = 1, 2, \ldots$

Use π_{θ_k} to collect n trajectories

$$\left(s_1^{(1)}, a_1^{(1)}, r_1^{(1)}, \cdots, s_{\tau_1}^{(1)}, a_{\tau_1}^{(1)}, r_{\tau_1}^{(1)}\right), \ldots\ldots, \left(s_1^{(n)}, a_1^{(n)}, r_1^{(n)}, \cdots, s_{\tau_n}^{(n)}, a_{\tau_n}^{(n)}, r_{\tau_n}^{(n)}\right)$$

Define

$$g = \frac{1}{n}\sum_{i=1}^{n}\sum_{h=1}^{\tau_n} \frac{\nabla_\theta\, \pi_\theta\left(a_h^{(i)} \middle| s_h^{(i)}\right)\Big|_{\theta=\theta_k}}{\pi_{\theta_k}\left(a_h^{(i)} \middle| s_h^{(i)}\right)} \underbrace{Q_{\phi_k}\left(s_h^{(i)}, a_h^{(i)}\right)}_{\approx\, Q^{\pi_{\theta_k}}} \quad \text{or} \quad \frac{1}{n}\sum_{i=1}^{n}\sum_{h=1}^{\tau_n}\sum_{a} \nabla_\theta\, \pi_\theta\left(a \middle| s_h^{(i)}\right)\Big|_{\theta=\theta_k} Q_{\phi_k}\left(s_h^{(i)}, a\right)$$

Perform updates

$$\theta_{k+1} \leftarrow \theta_k + \eta g \qquad \phi_{k+1} \leftarrow \phi_k - \lambda \nabla_\phi \sum_{i=1}^{n}\sum_{h=1}^{\tau_n}\left(Q_\phi\left(s_h^{(i)}, a_h^{(i)}\right) - r_h^{(i)} - \gamma Q_{\phi_k}\left(s_{h+1}^{(i)}, a_{h+1}^{(i)}\right)\right)^2\Bigg|_{\phi=\phi_k}$$

Advantage Actor-Critic (A2C) = PG + V_ϕ

For $k = 1,\ 2, \ldots$

Use π_{θ_k} to collect n trajectories

$$\left(s_1^{(1)}, a_1^{(1)}, r_1^{(1)}, \cdots, s_{\tau_1}^{(1)}, a_{\tau_1}^{(1)}, r_{\tau_1}^{(1)}\right), \ldots\ldots, \left(s_1^{(n)}, a_1^{(n)}, r_1^{(n)}, \cdots, s_{\tau_n}^{(n)}, a_{\tau_n}^{(n)}, r_{\tau_n}^{(n)}\right)$$

Define

$$= \frac{\nabla_\theta \pi_\theta\left(a_h^{(i)} \middle| s_h^{(i)}\right)\Big|_{\theta=\theta_k}}{\pi_{\theta_k}\left(a_h^{(i)} \middle| s_h^{(i)}\right)}$$

$$\approx A^{\pi_k}\left(s_h^{(i)}, a_h^{(i)}\right)$$

$$\mathbb{E}_{s,a}(\cdot) = \sum d_\rho^{\pi_{\theta_k}}(s)\, \nabla_\theta \pi_\theta(a|s)\, A^{\pi_k}(s,a)$$

$$g = \frac{1}{n}\sum_{i=1}^{n}\sum_{h=1}^{\tau_n} \nabla_\theta \log \pi_\theta\left(a_h^{(i)} \middle| s_h^{(i)}\right)\Big|_{\theta=\theta_k} \left(r_h^{(i)} + \gamma V_{\phi_k}\left(s_{h+1}^{(i)}\right) - V_{\phi_k}\left(s_h^{(i)}\right)\right)$$

or any other advantage estimator in the previous slide

Perform updates

$$V_\phi \approx V^{\pi_{\theta_k}}.$$

$$\theta_{k+1} \leftarrow \theta_k + \eta g \qquad \phi_{k+1} \leftarrow \phi_k - \lambda \nabla_\phi \frac{1}{n}\sum_{i=1}^{n}\sum_{h=1}^{\tau_n}\left(V_\phi\left(s_h^{(i)}\right) - r_h^{(i)} - \gamma V_{\phi_k}\left(s_{h+1}^{(i)}\right)\right)^2\Bigg|_{\phi=\phi_k}$$

Mnih et al., Asynchronous Methods for Deep Reinforcement Learning. 2016.

Proximal Policy Optimization (PPO) = NPG + V_ϕ

For $k = 1,\ 2, \ldots$

Use π_{θ_k} to collect n trajectories

$$\left(s_1^{(1)}, a_1^{(1)}, r_1^{(1)}, \cdots, s_{\tau_1}^{(1)}, a_{\tau_1}^{(1)}, r_{\tau_1}^{(1)}\right), \ldots \ldots, \left(s_1^{(n)}, a_1^{(n)}, r_1^{(n)}, \cdots, s_{\tau_n}^{(n)}, a_{\tau_n}^{(n)}, r_{\tau_n}^{(n)}\right)$$

Perform updates

$\approx A^{\pi_k}(s,a)$

or any other advantage estimator in the previous slide

$$\theta_{k+1} \leftarrow \underset{\theta}{\operatorname{argmax}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \sum_{h=1}^{\tau_n} \frac{\pi_\theta\left(a_h^{(i)} \middle| s_h^{(i)}\right)}{\pi_{\theta_k}\left(a_h^{(i)} \middle| s_h^{(i)}\right)} \left(r_h^{(i)} + \gamma V_{\phi_k}\left(s_{h+1}^{(i)}\right) - V_{\phi_k}\left(s_h^{(i)}\right)\right) - \frac{1}{\eta} \frac{1}{n} \sum_{i=1}^{n} \sum_{h=1}^{\tau_n} \mathrm{KL}\left(\pi_\theta\left(\cdot \middle| s_h^{(i)}\right), \pi_{\theta_k}\left(\cdot \middle| s_h^{(i)}\right)\right) \right\}$$

$$\phi_{k+1} \leftarrow \phi_k - \lambda \nabla_\phi \frac{1}{n} \sum_{i=1}^{n} \sum_{h=1}^{\tau_n} \left(V_\phi\left(s_h^{(i)}\right) - r_h^{(i)} - \gamma V_{\phi_k}\left(s_{h+1}^{(i)}\right)\right)^2 \Bigg|_{\phi=\phi_k}$$

Schulman et al., Proximal Policy Optimization Algorithms. 2017.

$\frac{r(a)\mathbb{1}(a_t=a)}{p_t(a)}$ $\frac{(r(a)-1)}{p_t(a)}\mathbb{1}(a_t=a)$

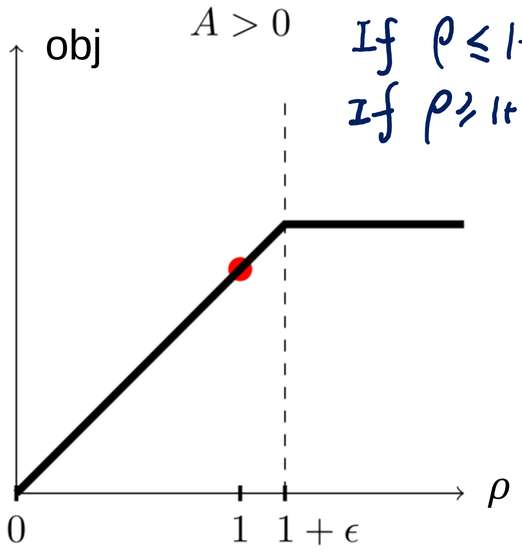
Additional Technique 1: Clipped Objective (for PPO)

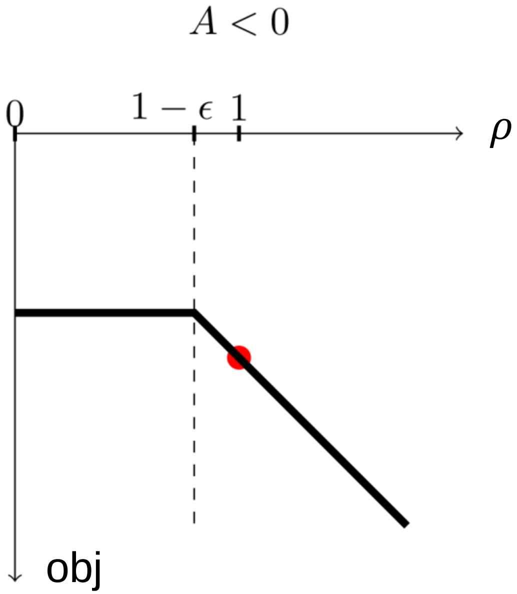
$$\rho := \frac{\pi_\theta\left(a_h^{(i)} \middle| s_h^{(i)}\right)}{\pi_{\theta_k}\left(a_h^{(i)} \middle| s_h^{(i)}\right)} \qquad A := \left(r_h^{(i)} + \gamma V_{\phi_k}\left(s_{h+1}^{(i)}\right) - V_{\phi_k}\left(s_h^{(i)}\right)\right)$$

$\text{clip}_{[1-\varepsilon,1+\varepsilon]}(\rho) = \min(\max(\rho, 1-\varepsilon), 1+\varepsilon)$

Instead of using ρA as the objective, use $\min\{\rho A,\ \text{clip}_{[1-\epsilon,1+\epsilon]}(\rho)A\}$

If $\rho \in [1-\varepsilon, 1+\varepsilon] \Rightarrow \rho A$
If $\rho \le 1-\varepsilon \Rightarrow \rho A$
If $\rho \ge 1+\varepsilon \Rightarrow (1+\varepsilon)A$





If $\rho \in (1-\varepsilon, 1+\varepsilon] \Rightarrow \rho A$
If $\rho \le 1-\varepsilon \Rightarrow (1-\varepsilon)A$ (strange case)
If $\rho \ge 1+\varepsilon \Rightarrow \rho A$

algorithm	avg. normalized score
No clipping or penalty	-0.39
Clipping, $\epsilon = 0.1$	0.76
Clipping, $\epsilon = 0.2$	**0.82**
Clipping, $\epsilon = 0.3$	0.70
Adaptive KL $d_{\text{targ}} = 0.003$	0.68
Adaptive KL $d_{\text{targ}} = 0.01$	0.74
Adaptive KL $d_{\text{targ}} = 0.03$	0.71
Fixed KL, $\beta = 0.3$	0.62
Fixed KL, $\beta = 1.$	0.71
Fixed KL, $\beta = 3.$	0.72
Fixed KL, $\beta = 10.$	0.69

Schulman et al., Proximal Policy Optimization Algorithms. 2017.

Additional Technique 2: Entropy Bonus

In the objective of policy update, add a bonus term

$$H(\pi_\theta(\cdot\,|s)) = \sum_a \pi_\theta(a|s)\ln\frac{1}{\pi_\theta(a|s)}$$

For PPO:

$$\underset{\theta}{\operatorname{argmax}}\left\{\frac{1}{n}\sum_{i=1}^{n}\sum_{h=1}^{\tau_n}\frac{\pi_\theta\left(a_h^{(i)}\middle|s_h^{(i)}\right)}{\pi_{\theta_k}\left(a_h^{(i)}\middle|s_h^{(i)}\right)}A_h^{(i)} \;-\frac{1}{\eta}\frac{1}{n}\sum_{i=1}^{n}\sum_{h=1}^{\tau_n}\mathrm{KL}\left(\pi_\theta\left(\cdot\middle|s_h^{(i)}\right),\pi_{\theta_k}\left(\cdot\middle|s_h^{(i)}\right)\right)\; +c\frac{1}{n}\sum_{i=1}^{n}\sum_{h=1}^{\tau_n}\underbrace{H\left(\pi_\theta\left(\cdot\middle|s_h^{(i)}\right)\right)}_{-\,\mathrm{KL}\left(\pi_\theta\left(\cdot\middle|s_h^{(i)}\right),\,\pi_{\mathrm{unif}}\left(\cdot\middle|s_h^{(i)}\right)\right)}\right\}$$

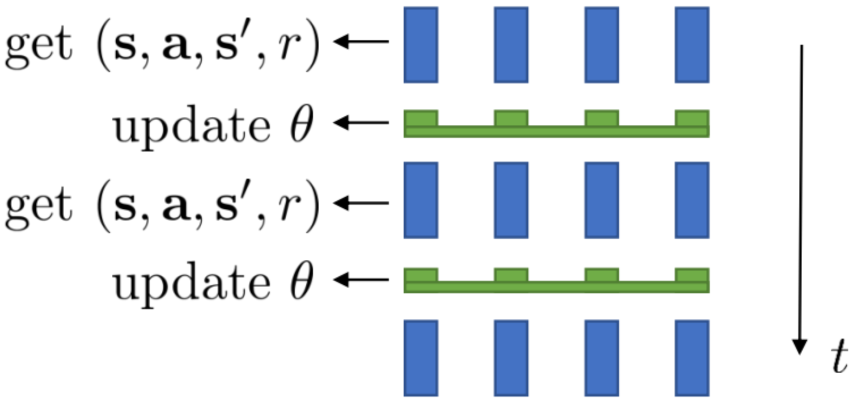
For A2C:

$$g = \frac{1}{n}\sum_{i=1}^{n}\sum_{h=1}^{\tau_n}\nabla_\theta \log\pi_\theta\left(a_h^{(i)}\middle|s_h^{(i)}\right)\Big|_{\theta=\theta_k}A_h^{(i)} + c\nabla_\theta\frac{1}{n}\sum_{i=1}^{n}\sum_{h=1}^{\tau_n}H\left(\pi_\theta\left(\cdot\middle|s_h^{(i)}\right)\right)$$

Additional Technique 3: Parallel Sample Collection

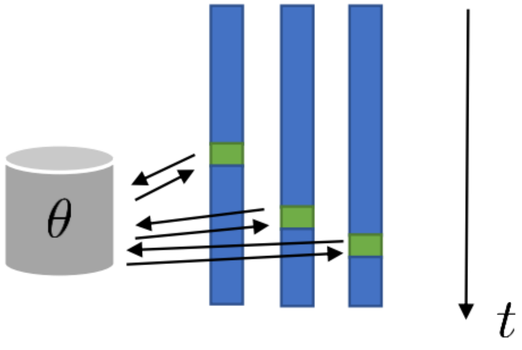
A2C

synchronized parallel actor-critic



A3C .

asynchronous parallel actor-critic



[Levine CS285 Lecture 6]

Actor-Critic Summary

PG $\longrightarrow$ A2C

NPG $\longrightarrow$ PPO

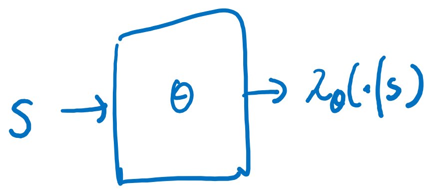
$s \to \boxed{\theta} \to \pi(\cdot|s)$

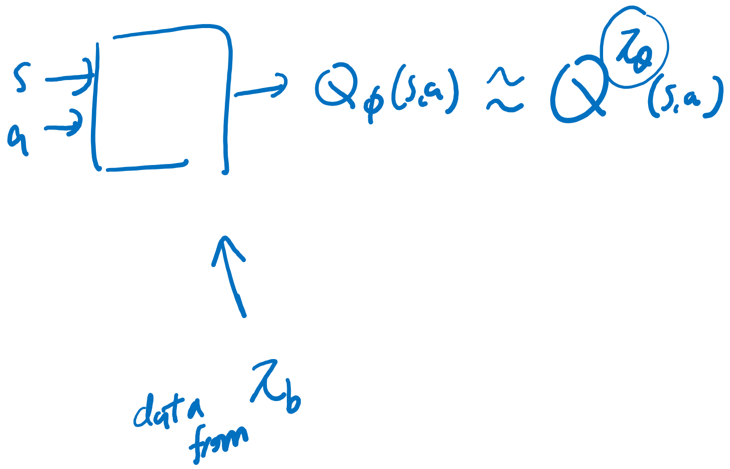
$s \to \boxed{\theta} \to \pi(\cdot|s)$ $\quad s \to \boxed{\phi} \to V_\phi(s)$

$s, a \to \boxed{\phi} \to Q_\phi(s,a)$

Off-Policy Actor-Critic

- Leveraging **off-policy evaluation** → allow reusing data





Review: Full-Information Policy Learning in MDPs

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}} \left(\underbrace{V^{\pi_\theta}(\rho) - V^{\pi_{\theta_k}}(\rho)} - \frac{1}{\eta} D(\theta, \theta_k) \right)$$

$$\approx \sum_{s,a} d_\rho^{\pi_{\theta_k}}(s) \left(\pi_\theta(a|s) - \pi_{\theta_k}(a|s) \right) Q^{\pi_{\theta_k}}(s,a)$$

$$\approx (\theta - \theta_k)^\top \sum_{s,a} d_\rho^{\pi_{\theta_k}}(s) \left(\nabla_\theta \pi_\theta(a|s) \Big|_{\theta=\theta_k} \right) Q^{\pi_{\theta_k}}(s,a)$$

Use any off-policy policy evaluation methods to find ϕ_k such that $Q_{\phi_k}(s,a) \approx Q^{\pi_{\theta_k}}(s,a)$

Suppose that our (s_i, a_i) samples are obtained from $\hat{\pi}$

Off-Policy Actor-Critic

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}} \left(\underbrace{V^{\pi_\theta}(\rho) - V^{\pi_{\theta_k}}(\rho)} - \frac{1}{\eta} D(\theta, \theta_k) \right) \quad // \quad \sum_s d_\rho^{\hat{\pi}}(s) \cdot \frac{d_\rho^{\pi_{\theta_k}}(s)}{d_\rho^{\hat{\pi}}(s)} \sum_a \cdots$$

$$\approx \sum_{s,a} d_\rho^{\pi_{\theta_k}}(s) \left(\pi_\theta(a|s) - \pi_{\theta_k}(a|s)\right) {\color{red} Q_{\phi_k}(s,a)} \qquad = \mathbb{E}_{s\sim\hat{\pi}}\left[\cancel{\frac{d_\rho^{\pi_{\theta_k}}(s)}{d_\rho^{\hat{\pi}}(s)}} \sum_a \left(\pi_\theta(a|s) - \pi_{\theta_k}(a|s)\right) {\color{red} Q_{\phi_k}(s,a)}\right]$$

$$\approx (\theta - \theta_k)^\top \sum_{s,a} d_\rho^{\pi_{\theta_k}}(s) \left(\nabla_\theta \pi_\theta(a|s)\Big|_{\theta=\theta_k} \right) {\color{red} Q_{\phi_k}(s,a)} \qquad = (\theta - \theta_k)^\top \mathbb{E}_{s\sim\hat{\pi}}\left[\cancel{\frac{d_\rho^{\pi_{\theta_k}}(s)}{d_\rho^{\hat{\pi}}(s)}} \sum_a \nabla_\theta \pi_\theta(a|s)\Big|_{\theta=\theta_k} {\color{red} Q_{\phi_k}(s,a)}\right]$$

Use any off-policy policy evaluation methods to find ϕ_k such that ${\color{red} Q_{\phi_k}(s,a) \approx Q^{\pi_{\theta_k}}(s,a)}$

Suppose that our (s_i, a_i) samples are obtained from $\hat{\pi}$

[Zhang et al. Global Optimality and Finite Sample Analysis of Softmax Off-Policy Actor Critic under State Distribution Mismatch. 2022.]

Actor-Critic + Replay Buffer

For $k = 1,\ 2, \ldots$

Collect samples using π_{θ_k}, and place them in the replay buffer

Sample a batch $\{(s_i, a_i, r_i, s_i')\}_{i=1}^n$ from replay buffer

Define

$$g = \frac{1}{n}\sum_{i=1}^{n}\sum_{a}\nabla_\theta \pi_\theta(a|s_i)\Big|_{\theta=\theta_k} Q_{\phi_k}(s_i, a)$$

Note: not using a_i here

Perform updates

$$\theta_{k+1} \leftarrow \theta_k + \eta g$$

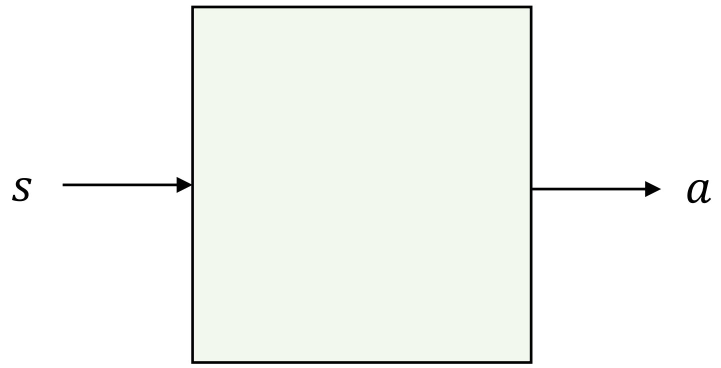
Off-policy TD → unstable (more on this later)

$$\phi_{k+1} \leftarrow \phi_k - \lambda \nabla_\phi \frac{1}{n}\sum_{i=1}^{n}\left(Q_\phi(s_i, a_i) - r_i - \gamma\, \mathbb{E}_{a' \sim \pi_{\theta_k}(\cdot|s_i')}\left[Q_\phi(s_i', a')\right]\right)^2\Bigg|_{\phi=\phi_k}$$

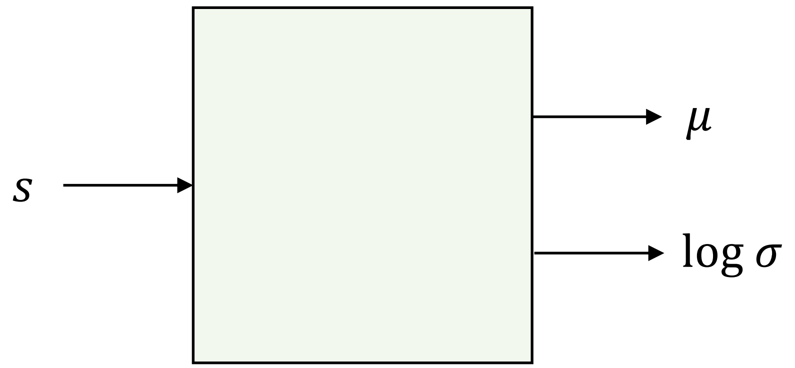
Dealing with Continuous Action Sets

Review: Linear Bandits and One-Point Gradient Estimator

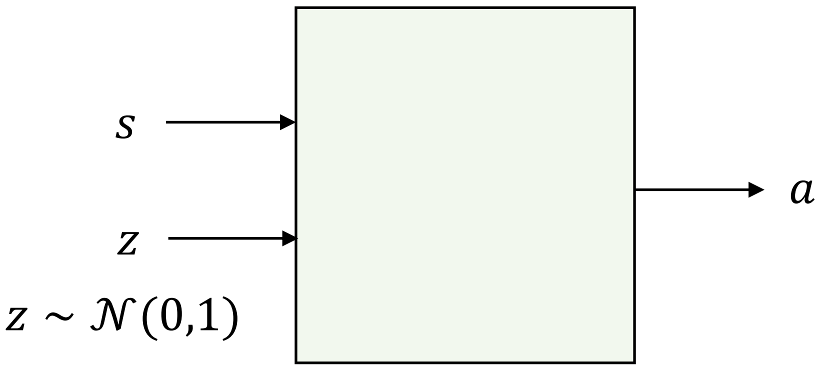
Policy Network for Continuous Action Sets



Policy Network for Continuous Action Sets



Explicitly models $\pi_\theta(a|s)$



Implicitly modeling $\pi_\theta(a|s)$

Option 1: making σ part of policy parameters

Option 2: making σ a hyper-parameters (decreases over time)

can sample from it, but do not know the function $\pi_\theta(\cdot\,|s)$

A2C / PPO with Continuous Action Sets

$$g = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \log \pi_\theta(a_i|s_i)\Big|_{\theta=\theta_k} A_i$$

$$\theta_{k+1} \leftarrow \underset{\theta}{\operatorname{argmax}} \left\{ \frac{1}{n}\sum_{i=1}^{n} \frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} A_i - \frac{1}{\eta}\frac{1}{n}\sum_{i=1}^{n} \mathrm{KL}\left(\pi_\theta(\cdot\,|s_i), \pi_{\theta_k}(\cdot\,|s_i)\right) \right\}$$

Recall: Actor-Critic without need for inverse weighting

Actor-critic with $Q_\phi(s,a)$ function approximation

For $k = 1,\ 2, \ldots$

Use π_{θ_k} to collect n trajectories

$$\left(s_1^{(1)}, a_1^{(1)}, r_1^{(1)}, \cdots, s_{\tau_1}^{(1)}, a_{\tau_1}^{(1)}, r_{\tau_1}^{(1)}\right), \ldots \ldots, \left(s_1^{(n)}, a_1^{(n)}, r_1^{(n)}, \cdots, s_{\tau_n}^{(n)}, a_{\tau_n}^{(n)}, r_{\tau_n}^{(n)}\right)$$

Define $$g = \frac{1}{n}\sum_{i=1}^{n}\sum_{h=1}^{\tau_n}\sum_{a} \nabla_\theta\, \pi_\theta\left(a \middle| s_h^{(i)}\right)\Big|_{\theta=\theta_k} Q_{\phi_k}\left(s_h^{(i)}, a\right)$$

Perform updates

$$\theta_{k+1} \leftarrow \theta_k + \eta g \qquad \phi_{k+1} \leftarrow \phi_k - \lambda \nabla_\phi \sum_{i=1}^{n}\sum_{h=1}^{\tau_n}\left(Q_\phi\left(s_h^{(i)}, a_h^{(i)}\right) - r_h^{(i)} - \gamma Q_{\phi_k}\left(s_{h+1}^{(i)}, a_{h+1}^{(i)}\right)\right)^2\Bigg|_{\phi=\phi_k}$$

Deterministic Policy Gradient Theorem

Deterministic Policy Gradient Algorithm

For $k = 1,\ 2, \ldots$

Use π_{θ_k} to collect n trajectories

$$\left(s_1^{(1)}, a_1^{(1)}, r_1^{(1)}, \cdots, s_{\tau_1}^{(1)}, a_{\tau_1}^{(1)}, r_{\tau_1}^{(1)}\right), \ldots \ldots, \left(s_1^{(n)}, a_1^{(n)}, r_1^{(n)}, \cdots, s_{\tau_n}^{(n)}, a_{\tau_n}^{(n)}, r_{\tau_n}^{(n)}\right)$$

Define $g = \frac{1}{n} \sum_{i=1}^{n} \sum_{h=1}^{\tau_n} \nabla_\theta Q_{\phi_k}\left(s_h^{(i)}, \pi_\theta\left(s_h^{(i)}\right)\right)\Big|_{\theta=\theta_k}$

Perform updates

$$\theta_{k+1} \leftarrow \theta_k + \eta g \qquad \phi_{k+1} \leftarrow \phi_k - \lambda \nabla_\phi \sum_{i=1}^{n} \sum_{h=1}^{\tau_n} \left(Q_\phi\left(s_h^{(i)}, a_h^{(i)}\right) - r_h^{(i)} - \gamma Q_{\phi_k}\left(s_{h+1}^{(i)}, a_{h+1}^{(i)}\right)\right)^2\Bigg|_{\phi=\phi_k}$$

Two Viewpoints for the Deterministic PG Algorithm

Deep Deterministic Policy Gradient (DDPG)

For $k = 1,\ 2, \ldots$

Use π_θ to collect samples and place them in replay buffer

Sample a batch $\{(s_i, a_i, r_i, s_i')\}_{i=1}^n$ from the replay buffer

$$\theta \leftarrow \theta + \eta \sum_{i=1}^{n} \nabla_\theta Q_\phi\big(s_i, \pi_\theta(s_i)\big)$$

$$\phi \leftarrow \phi - \lambda \nabla_\phi \sum_{i=1}^{n} \Big(Q_\phi(s_i, a_i) - r_i - \gamma Q_{\phi_{\text{tar}}}\big(s_i', \pi_{\theta_{\text{tar}}}(s_i')\big)\Big)^2$$

$$\theta_{\text{tar}} \leftarrow \tau\theta + (1-\tau)\theta_{\text{tar}}$$

$$\phi_{\text{tar}} \leftarrow \tau\phi + (1-\tau)\phi_{\text{tar}}$$

Twin Delayed DDPG (TD3)

Soft Actor-Critic (SAC)