



Reinforcement Learning

Lecture 21: Extensions to Policy Search

4 Nov, 2025

Instructor: Raunak P. Bhattacharyya



ScAI

YARDI SCHOOL OF ARTIFICIAL INTELLIGENCE
INDIAN INSTITUTE OF TECHNOLOGY DELHI

Recap

- Control variates, baseline, optimal baseline
- Use Q instead of G
- Advantage Estimation

Actor-Critic

Critic Updates action-value function parameters w

Actor Updates policy parameters θ , in direction suggested by critic

$$\nabla_{\theta} J(\theta) \approx \mathbb{E}_{\pi_{\theta}} [\nabla_{\theta} \log \pi_{\theta}(s, a) Q_w(s, a)]$$

$$\Delta \theta = \alpha \nabla_{\theta} \log \pi_{\theta}(s, a) Q_w(s, a)$$

Advantage computation

$$A(s_t, a_t) = Q_w(s_t, a_t) - V_v(s_t)$$

$$Q(s_t, a_t) = \mathbb{E}[r_{t+1} + \gamma V(s_{t+1})]$$

$$A(s_t, a_t) = r_{t+1} + \gamma V_v(s_{t+1}) - V_v(s_t)$$

Actor-Critic

Sample $\{s_i, a_i\}$ from $\pi_\theta(a|s)$

Fit $\hat{V}_\phi(s)$ to sampled reward sums

How to do this?

Evaluate $\hat{A}^\pi(s_i, a_i) = r(s_i, a_i) + \hat{V}_\phi(s'_i) - \hat{V}_\phi(s_i)$

Approximation here

$$\nabla_\theta J(\theta) \simeq \sum_i \nabla_\theta \log \pi_\theta(a_i|s_i) \hat{A}^\pi(s_i, a_i)$$

Improve policy by $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

QAC: Q Actor-Critic

- Simple actor-critic algorithm based on action-value critic
- Using linear value fn approx. $Q_w(s, a) = \phi(s, a)^\top w$
 - Critic Updates w by linear TD(0)
 - Actor Updates θ by policy gradient

function QAC

 Initialise s, θ

 Sample $a \sim \pi_\theta$

for each step **do**

 Sample reward $r = \mathcal{R}_s^a$; sample transition $s' \sim \mathcal{P}_{s,\cdot}^a$.

 Sample action $a' \sim \pi_\theta(s', a')$

$\delta = r + \gamma Q_w(s', a') - Q_w(s, a)$

$\theta = \theta + \alpha \nabla_\theta \log \pi_\theta(s, a) Q_w(s, a)$

$w \leftarrow w + \beta \delta \phi(s, a)$

$a \leftarrow a', s \leftarrow s'$

end for

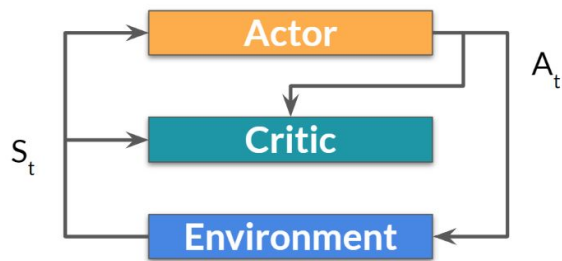
end function

Different Flavours of PG methods

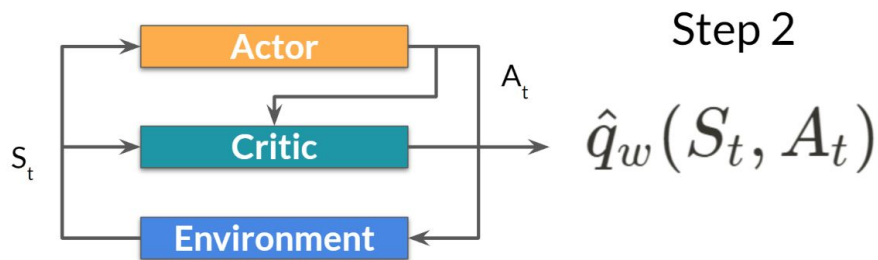
$\nabla_{\theta} J(\theta) = \mathbb{E}_{\pi_{\theta}} [\nabla_{\theta} \log \pi_{\theta}(s, a) G_t]$	REINFORCE
$= \mathbb{E}_{\pi_{\theta}} [\nabla_{\theta} \log \pi_{\theta}(s, a) Q^w(s, a)]$	Q Actor-Critic
$= \mathbb{E}_{\pi_{\theta}} [\nabla_{\theta} \log \pi_{\theta}(s, a) A^w(s, a)]$	Advantage Actor-Critic
$= \mathbb{E}_{\pi_{\theta}} [\nabla_{\theta} \log \pi_{\theta}(s, a) \delta]$	TD Actor-Critic

Image taken from CMU CS10703 lecture slides

Actor-Critic Process: Visuals

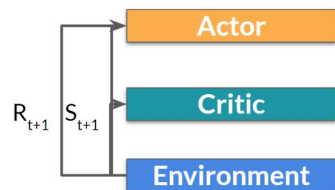


Step 1



Step 2

Actor-Critic Process: Visuals

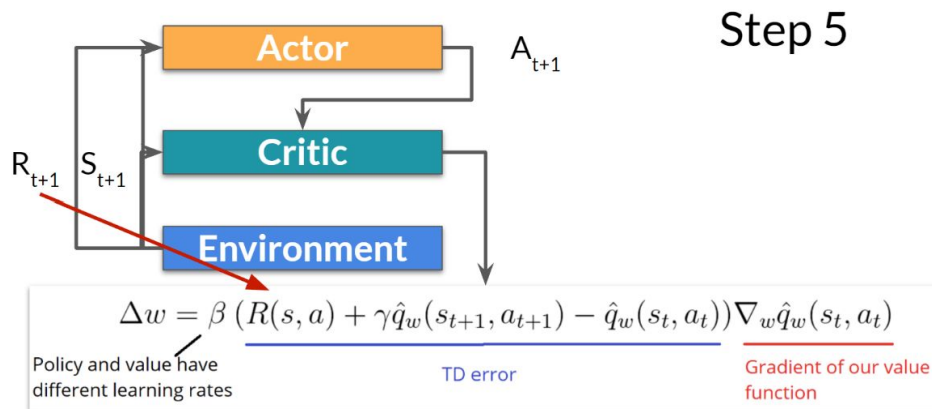


Step 3

Step 4

$$\Delta\theta = \alpha \nabla_{\theta}(\log\pi_{\theta}(s, a)) \hat{q}_w(s, a)$$

Change in policy parameters (weights) Action value estimate



Example

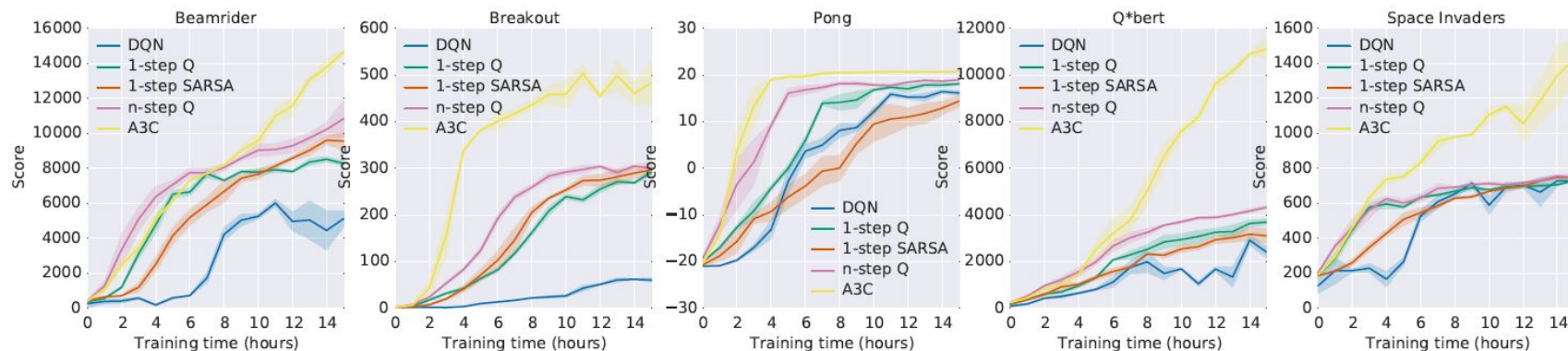


Labyrinth, A3C paper, Source: [Youtube](#)



[Source: Youtube](#)

Actor-Critic Performance



Method	Training Time	Mean	Median
DQN	8 days on GPU	121.9%	47.5%
Gorila	4 days, 100 machines	215.2%	71.3%
D-DQN	8 days on GPU	332.9%	110.9%
Dueling D-DQN	8 days on GPU	343.8%	117.1%
Prioritized DQN	8 days on GPU	463.6%	127.6%
A3C, FF	1 day on CPU	344.1%	68.2%
A3C, FF	4 days on CPU	496.8%	116.6%
A3C, LSTM	4 days on CPU	623.0%	112.6%

Advent of Policy Gradient Methods

- Oriented towards finding deterministic policies
- Arbitrary change in action value can cause it to be selected/not selected
 - Not converge
- Instead of approx value func and then deterministic pol, direct stochastic policy

Policy Gradient Methods for Reinforcement Learning with Function Approximation

Richard S. Sutton, David McAllester, Satinder Singh, Yishay Mansour
AT&T Labs – Research, 180 Park Avenue, Florham Park, NJ 07932

Why Policy Optimisation instead of Value Function

	Policy Optimization	Dynamic Programming
Conceptually:	Optimize what you care about	Indirect, exploit the problem structure, self-consistency
Empirically:	More compatible with rich architectures (including recurrence) More versatile More compatible with auxiliary objectives	More compatible with exploration and off-policy learning More sample-efficient when they work

Extensions: Motivation



[Source](#)



[Source](#)

Plan

- Off-policy and surrogate adv. obj.
- Algorithmic instantiation