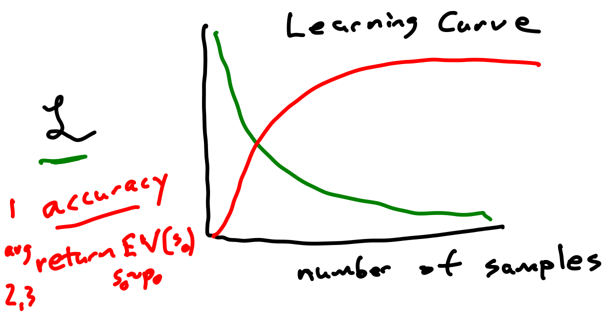
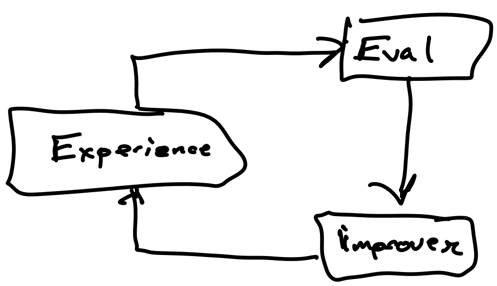



# Actor - Critic Deep RL

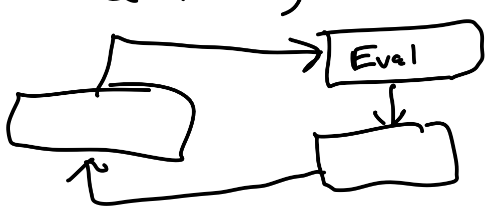
Last time: Policy Grad




$$\hat{Q} = \sum r_t \leftarrow \text{high variance}$$

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$$

Before: Q-learning




Q-network params

$$\phi \leftarrow \phi + \alpha \frac{\partial Q}{\partial \phi}\left(Q_\phi - r - \gamma \max_{a'} Q_{\phi'}(s', a')\right)$$

$$\pi(a|s) = \begin{cases} 1-\varepsilon & \text{if } a = \arg\max Q_\phi(s,a) \\ \frac{1}{|A|-1} & \text{o.w.} \end{cases}$$

Actor - Critic

↑ policy ↑ Value fn

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_i \sum_t \nabla_\theta \log \pi_\theta(a|s) \underbrace{\left(Q(\ \ ) - b\right)}_{A^\pi(s,a)}$$

know $Q^\pi(s,a)$

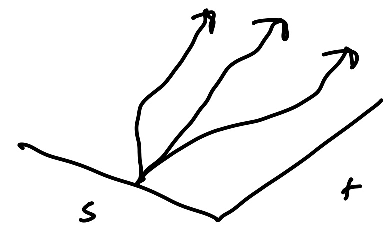
$$V^\pi(s) = E\left[Q(s,a) \mid \pi(a|s)\right]$$

$$A^\pi(s,a) = Q^\pi(s,a) - V(s)$$

$Q_\phi$, $V_\phi$, or $A_\phi$ ?     $A_\phi$ - use it directly

$V_\phi(s)$

$$Q(s,a) = r(s,a) + \gamma E[V(s')]$$
$$\approx r(s,a) + \gamma V(s')$$
$$A(s,a) \approx r(s,a) + \gamma V(s') - V(s)$$




targets for $V(\cdot)$

- $\{(s_{i,t}, \sum_{t'=t}^{T} r(s_{i,t'}, a_{i,t'}))\}$   Monte Carlo Target
- $\{(s_{i,t}, r(s_{i,t}, a_{i,t}) + \gamma V_\phi(s_{i,t+1}))\}$   "Bootstrapped" target

Batch Actor-Critic Algorithm

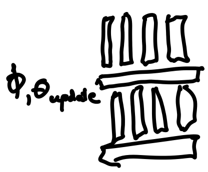
loop
Collect data by running $\pi_\theta$
fit $V_\phi^{\pi_\theta}$
evaluate $A^\pi(s,a) = r(s,a) + \gamma V_\phi^\pi(s') - V_\phi^\pi(s)$
$\nabla_\theta J(\theta) = \sum_i \nabla_\theta \log \pi_\theta(a|s) \hat{A}(s,a)$
$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

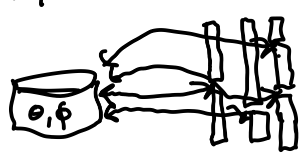
Online

$\longrightarrow$ get $(s, a, s', r)$
update $\phi$ using $r + \gamma V_\phi(s')$
⋮

---

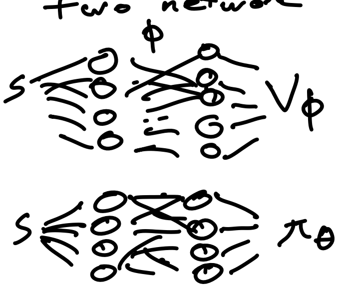
A3C   Asynchronous Advantage Actor-Critic



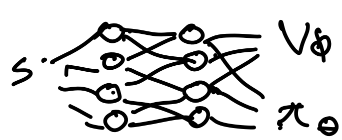

Design Choices: Network

two network




more data, easier to train

shared




less data, tricky to train

# Generalized Advantage Estimation GAE

$\hat{Q}$ ← high variance

$V_\phi$ ← biased ← especially in future

$$\hat{A}_n^\pi(s_t, a_t) = \sum_{t'=t}^{t+n} \gamma^{t'-t} r(s_t, a_t) - V_\phi^\pi(s_t) + \gamma^n V_\phi^\pi(s_{t+n})$$

$$\hat{A}_{GAE}^\pi = \sum w_n \hat{A}_n^\pi \qquad w_n \propto \lambda^{n-1}$$


$$= r_t + \gamma\left((1-\lambda) V_\phi(s_{t+1}) + \lambda\left(r_{t+1} + \gamma(1-\lambda) V_\phi(s_{t+2}) \ldots\right.\right.$$

$$= \sum_{t'=t}^{\infty} (\gamma\lambda)^{t'-t} \delta_{t'} \qquad \delta_{t'} = r(s_{t'}, a_{t'}) + \gamma V_\phi(s_{t'+1}) - V_\phi(s_{t'})$$

Eligibility Traces

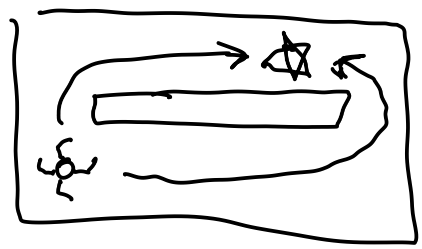
---

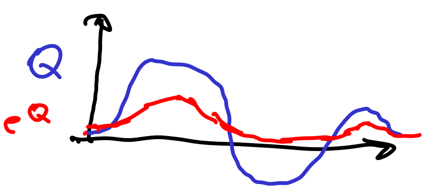
# Soft-Actor-Critic

$$\pi(a|s) \propto e^{Q(s,a)}$$

$$\pi^*_{MaxEnt} = \arg\max \left[\sum_t r_t + \beta H(\pi(a|s))\right]$$

entropy







$$\mathbb{E}_{s_t}\left[D_{KL}\left(\pi_\theta(a,s) \,\Big|\Big|\, \frac{Q(s,a)}{Z(s)}\right)\right]$$

Robust Convergence

---

Didn't Cover Model-Based Deep RL

1. Learn Dynamics Model
2. Use model Free RL / Planning / LQR

---

# How to Choose

Size S, A

**Sample Efficiency**

- how much experience does it take to get a good policy
- On Policy / off-policy

**Stability / Ease of Use**

Less Samples ⟷ More Samples

model-based shallow RL | model-based Deep RL | off-policy Q-learning | actor-critic | On-policy policy gradient | evolutionary gradient-free

Wall Clock time

With fast simulator, roughly reversed

slower ⟷ faster

more assumptions ⟷ less assumptions

linear/tabular | Markov/fully observable | partially observable $\pi(a|o)$

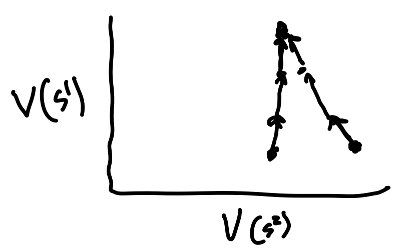
converge less reliably ⟷ Converge more reliably

Stability

- Converge?
- to what?
- every time?

$\|x\|^\infty = \max_i |x_i|$

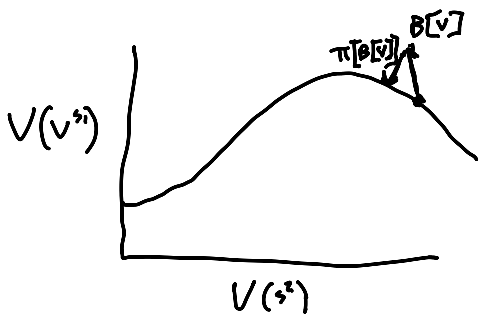
$S = \{s^1, s^2\}$




$V \leftarrow B[V]$

$V(s) \approx \lambda^T \beta(s_i)$

$\Pi[V] = \arg\min_\lambda \sum_{i=1}^{N} (\lambda^T \beta(s_i) - V(s_i))^2$




$\Pi[B[V]]$

$\| \ \|^\infty$

$\|f(x) - f(y)\| \leq \gamma \|x - y\|$

Contraction mapping

Deep Q-learning less stable

Policy Gradient more stable