ACTOR-CRITIC

Algorithm
Practice

Diego Klabjan
Professor, Industrial Engineering and Management Sciences

Northwestern | McCORMICK SCHOOL OF ENGINEERING

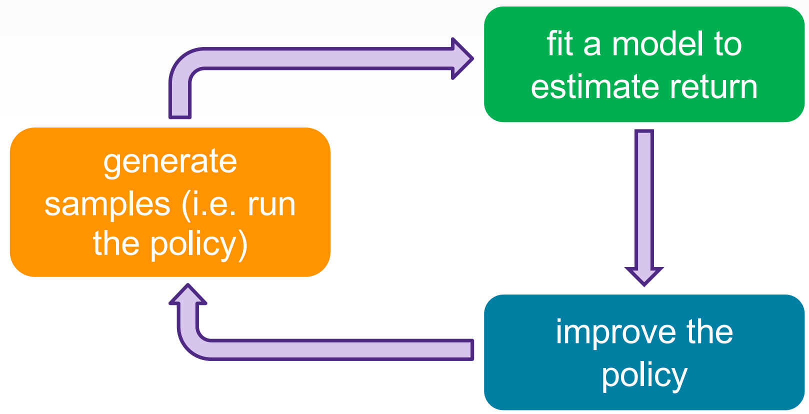
Recap: Policy Gradients

REINFORCE algorithm:

1. sample $\{\tau^i\}$ from $\pi_\theta(a_t|s_t)$ (run the policy)
2. $\nabla_\theta J(\theta) \approx \sum_i \left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta\left(a_t^i \middle| s_t^i\right) \left(\sum_{t'=t}^{T} r(s_{t'}^i, a_{t'}^i) \right) \right)$
3. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

$$\hat{Q}^\pi(x_t, u_t) = \sum_{t'=t}^{T} r(x_{t'}, u_{t'})$$

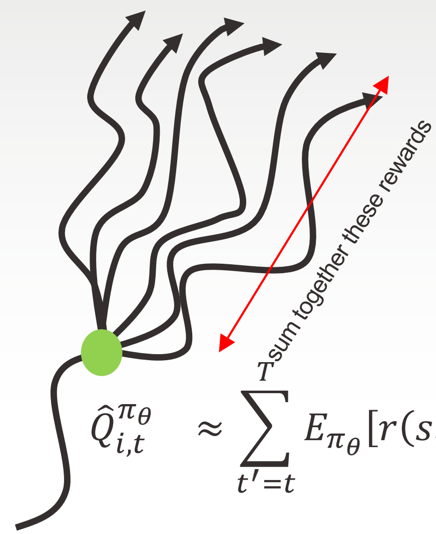
$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta\left(a_t^i \middle| s_t^i\right) \underbrace{\hat{Q}_{i,t}^{\pi_\theta}}_{\text{“reward to go”}}$$



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

Improving Policy Gradient

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_t^i|s_t^i)\underbrace{\sum_{t'=t}^{T} r(s_{t'}^i, a_{t'}^i)}_{\text{“reward to go”}\ \hat{Q}_{i,t}^{\pi_\theta}}$$



$$\hat{Q}_{i,t}^{\pi_\theta} \approx \sum_{t'=t}^{T} E_{\pi_\theta}[r(s_{t'}, a_{t'})|s_t, a_t]$$

$\hat{Q}_{i,t}^{\pi_\theta}$: estimate of expected reward if we take action a_t^i in state s_t^i

$Q^{\pi_\theta}(s_t, a_t) = \sum_{t'=t}^{T} E_{\pi_\theta}[r(s_{t'}, a_{t'})|s_t, a_t]$: true *expected* reward-to-go

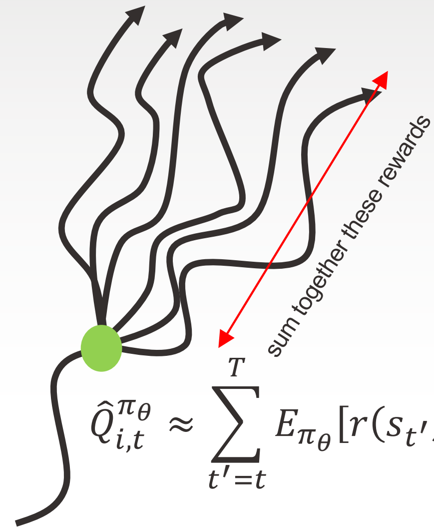
$$V^{\pi_\theta}(s_t) = E_{a_t \sim \pi_\theta(a_t|s_t)}[Q^{\pi_\theta}(s_t, a_t)]$$

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_t^i|s_t^i)\left(Q^{\pi_\theta}(s_t^i, a_t^i) - V^{\pi_\theta}(s_t^i)\right)$$

Baseline

$Q^{\pi_\theta}(s_t, a_t) = \sum_{t'=t}^{T} E_{\pi_\theta}[r(s_{t'}, a_{t'})|s_t, a_t]$: true *expected* reward-to-go

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_t^i|s_t^i)\left(Q^{\pi_\theta}(s_t^i, a_t^i) - V^{\pi_\theta}(s_t^i)\right)$$



$$\hat{Q}_{i,t}^{\pi_\theta} \approx \sum_{t'=t}^{T} E_{\pi_\theta}[r(s_{t'}, a_{t'})|s_t, a_t]$$

$$b_t = \frac{1}{N}\sum_i Q^{\pi_\theta}(s_t^i, a_t^i)$$

Average what?

$$V^{\pi_\theta}(s_t) = E_{a_t \sim \pi_\theta(a_t|s_t)}[Q^{\pi_\theta}(s_t, a_t)]$$

State and Q Value Functions

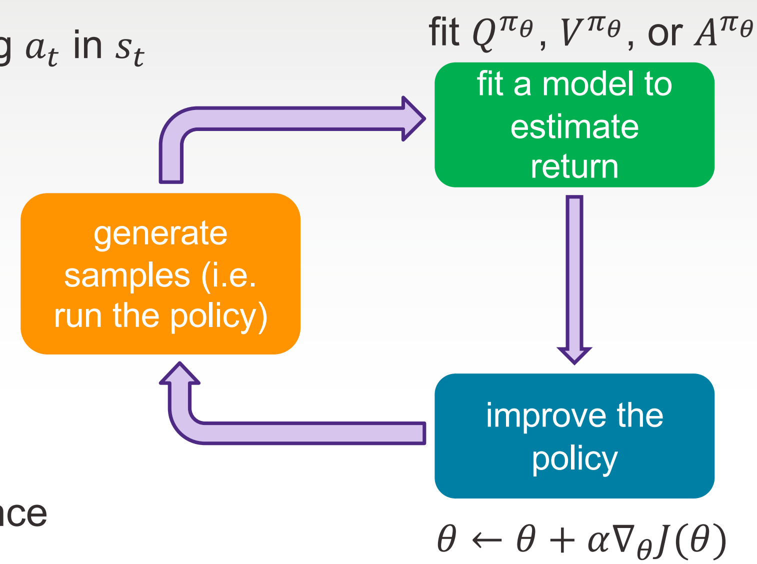
$Q^{\pi_\theta}(s_t, a_t) = \sum_{t'=t}^{T} E_{\pi_\theta}[r(s_{t'}, a_{t'}) | s_t, a_t]$: total reward from taking a_t in s_t

$V^{\pi_\theta}(s_t) = E_{a_t \sim \pi_\theta(a_t|s_t)}[Q^{\pi_\theta}(s_t, a_t)]$: total reward from s_t

$A^{\pi_\theta}(s_t, a_t) = Q^{\pi_\theta}(s_t, a_t) - V^{\pi_\theta}(s_t)$: how much better a_t is

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t^i | s_t^i) A^{\pi_\theta}(s_t^i, a_t^i)$$

the better this estimate, the lower the variance



$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t^i | s_t^i) \left(\underline{\sum_{t'=t}^{T} r(s_{i,t'}, a_{i,t'}) - b} \right)$$

unbiased, but high variance single-sample estimate

Value Function Fitting

$Q^{\pi_\theta}(s_t, a_t) = \sum_{t'=t}^{T} E_{\pi_\theta}[r(s_{t'}, a_{t'})|s_t, a_t]$

$V^{\pi_\theta}(s_t) = E_{a_t \sim \pi_\theta(a_t|s_t)}[Q^{\pi_\theta}(s_t, a_t)]$

$A^{\pi_\theta}(s_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi_\theta}(s_t)$

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t^i|s_t^i) A^{\pi_\theta}(s_t^i, a_t^i)$$

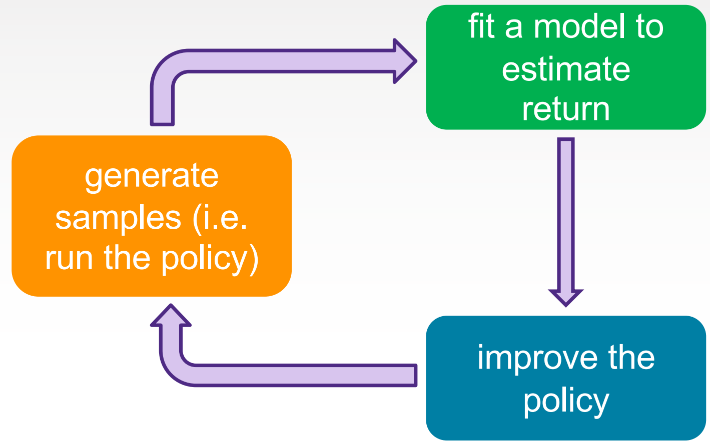
Neural network to fit Q^{π_θ}, V^{π_θ}, A^{π_θ}?

$Q^{\pi_\theta}(s_t, a_t) = r(s_t, a_t) + \underline{E_{s_{t+1} \sim p(s_{t+1}|s_t, a_t)}[V^{\pi_\theta}(s_{t+1})]}$

$A^{\pi_\theta}(s_t, a_t) \approx r(s_t, a_t) + V^{\pi_\theta}(s_{t+1}) - V^{\pi_\theta}(s_t)$

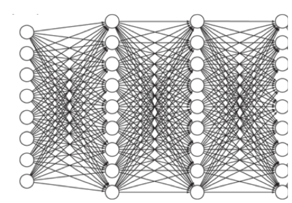
Let's just fit $V^{\pi}(s)$!

fit Q^{π_θ}, V^{π_θ}, or A^{π_θ}



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

s



$\hat{V}^{\pi_\theta}(s)$

parameters ϕ

Policy Evaluation

$V^{\pi_\theta}(s_t) = \sum_{t'=t}^{T} E_{\pi_\theta}[r(s_{t'}, a_{t'})|s_t]$

$J(\theta) = E_{s_1 \sim p(s_1)}[V^{\pi_\theta}(s_1)]$

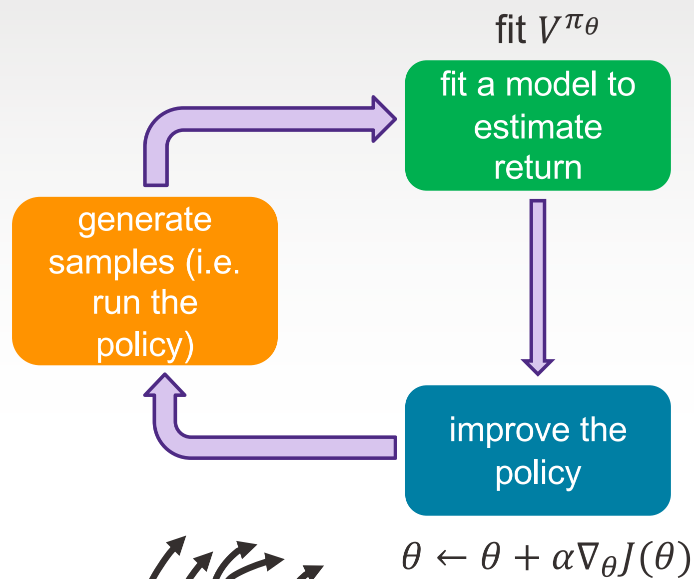
How can we perform policy evaluation?

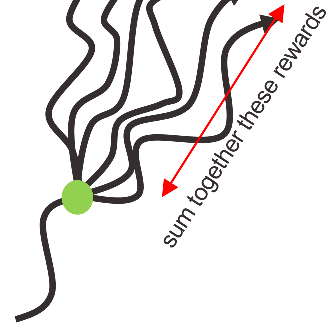
Monte Carlo policy evaluation

$$V^{\pi_\theta}(s_t) \approx \sum_{t'=t}^{T} r(s_{t'}, a_{t'})|s_t$$

$$V^{\pi_\theta}(s_t) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t'=t}^{T} r(s_{t'}^i, a_{t'}^i)|s_t$$

(requires us to reset the simulator to given state)





Monte Carlo Evaluation

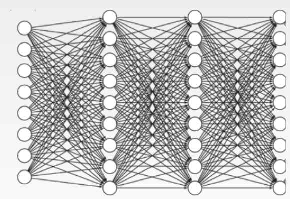
- Need trajectories using given state at *t*

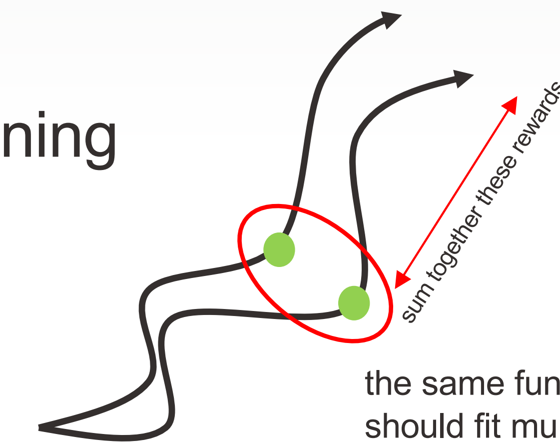
$$V^{\pi_\theta}(s_t) \approx \sum_{t'=t}^{T} r(s_{t'}, a_{t'})|s_t$$

  - Hard to obtain
- Instead sample trajectories from beginning
  - Record reward from *t* onwards

$$V^{\pi_\theta}(s_t) \approx \frac{1}{N}\sum_{i=1}^{N} \underbrace{\sum_{t'=t}^{T} r(s_{t'}^i, a_{t'}^i)}_{y_i}$$

training data: $\left\{\left(s_t^i, \sum_{t'=t}^{T} r(\, s_{t'}^i, s_{t'}^i)\right)\right\}$





the same function should fit multiple samples!

Monte Carlo Evaluation

- Generate trajectories from initial state
- Capture
  - State at time *t*
  - Reward to the end

training data: $\left\{\left(s^i, y_i = \sum_{t'=t}^{T} r\left(s_{t'}^i, s_{t'}^i\right)\right)\right\}$

supervised regression: $\mathcal{L}(\phi) = \frac{1}{2}\sum_i \left\| \hat{V}_\phi^{\pi_\theta}\left(s^i\right) - y_i \right\|^2$

Improvement

Ideal target: $y_{i,t} = \sum_{t'=t}^{T} E_{\pi_\theta}\left[r(s_{t'}, a_{t'})\middle|s_t^i\right] \approx r\left(s_t^i, a_t^i\right) + V^{\pi_\theta}\left(s_{t+1}^i\right) \approx r\left(s_t^i, a_t^i\right) + \underbrace{\hat{V}_\phi^{\pi_\theta}\left(s_{t+1}^i\right)}$

Directly use previous fitted value function!

Training data: $\left\{\left(s_t^i, r\left(s_t^i, a_t^i\right) + \underbrace{\hat{V}_{\phi_{old}}^{\pi_\theta}\left(s_{t+1}^i\right)}\right)\right\}$

Previous value of ϕ

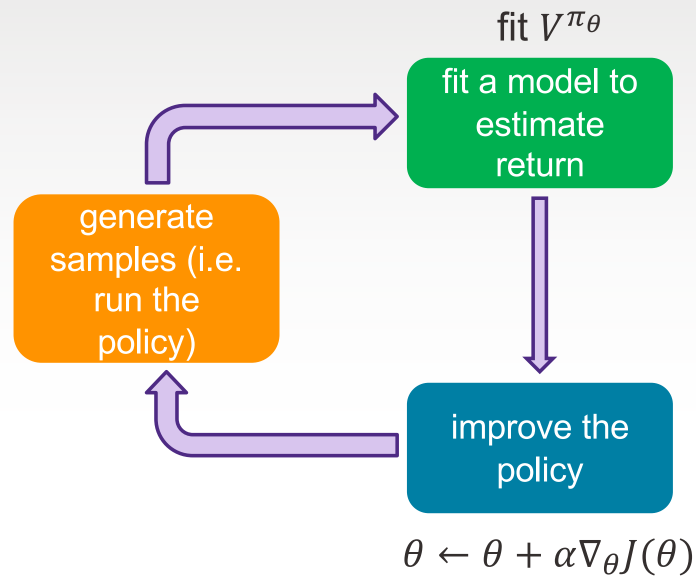
Supervised regression: $\mathcal{L}(\phi) = \frac{1}{2}\sum_i \left\|\hat{V}_\phi^{\pi_\theta}(s_i) - y_i\right\|^2$

- Lower accuracy than starting at time t in state s_t
- Variance reduced
  - Do not reset in each iteration

Actor-critic Algorithm

Batch actor-critic algorithm:

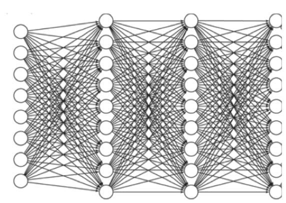
1. sample $\{s_t^i, a_t^i\}$ from $\pi_\theta(a|s)$ // several trajectories
2. fit $\hat{V}_\phi^{\pi_\theta}(s)$ to samples reward sums
3. evaluate $\hat{A}^{\pi_\theta}(s_t^i, a_t^i) = r(s_t^i, a_t^i) + \hat{V}_\phi^{\pi_\theta}(s_t^{i\prime}) - \hat{V}_\phi^{\pi_\theta}(s_t^i)$
4. $\nabla_\theta J(\theta) \approx \Sigma_i \Sigma_t \nabla_\theta \log \pi_\theta\,(a_t^i|s_t^i) \hat{A}^{\pi_\theta}(s_t^i, a_t^i)$
5. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$



$$y_{i,t} = r(s_t^i, a_t^i) + \hat{V}_\phi^{\pi_\theta}(s_t^{i\prime} = s_{t+1}^i)$$

$$\mathcal{L}(\phi) = \frac{1}{2} \sum_i \left\| \hat{V}_\phi^{\pi_\theta}(s_i) - y_i \right\|^2$$

s



$\hat{V}^{\pi_\theta}(s)$
parameters ϕ

Actor-Critic Enhancement

- When T large
  - $\hat{V}_{\phi}^{\pi_\theta}(s_t^{i'})$ tend to grow
- Modified advantage

$$\hat{A}^{\pi_\theta}(s_t^i, a_t^i) = r(s_t^i, a_t^i) + \gamma \hat{V}_{\phi}^{\pi_\theta}(s_t^{i'}) - \hat{V}_{\phi}^{\pi_\theta}(s_t^i)$$

- γ unrelated to the discount factor
  - Set to value close to 1
    - For example 0.99
  - Use mostly for infinite time horizon

Discount Factors for Policy Gradient

option 1: $\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_t^i|s_t^i)\left(\sum_{t'=t}^{T}\gamma^{t'-t}r(s_{t'}^i, a_{t'}^i)\right)$

option 2: $\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_t^i|s_t^i)\right)\left(\sum_{t=1}^{T}\gamma^{t-1}r(s_{t'}^i, a_{t'}^i)\right)$

Not the same!

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_t^i|s_t^i)\left(\sum_{t'=t}^{T}\gamma^{t'-1}r(s_{t'}^i, a_{t'}^i)\right)$$

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\gamma^{t-1}\nabla_\theta \log \pi_\theta(a_t^i|s_t^i)\left(\sum_{t'=t}^{T}\gamma^{t'-t}r(s_{t'}^i, a_{t'}^i)\right)$$

Mathematically equivalent

$$r(\tau) = \sum_{t'=1}^{T}\gamma^{t-1}r(s_{t'}^i, a_{t'}^i)$$

Discount Factor

- Mathematically correct
  - Option 2

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\gamma^{t-1}\nabla_\theta \log \pi_\theta(a_t^i|s_t^i)\left(\sum_{t'=t}^{T}\gamma^{t'-t} r(s_{t'}^i, a_{t'}^i)\right)$$

  - Discount factor part of input
- In practice option 1 used

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_t^i|s_t^i)\left(\sum_{t'=t}^{T}\gamma^{t'-t} r(s_{t'}^i, a_{t'}^i)\right)$$

  - Variance reduction
    - Scale down rewards
    - Factor as hyperparameter
      - Needs tuning

Online Actor-critic Algorithm

Batch actor-critic algorithm:

1. sample $\{s_t^i, a_t^i\}$ from $\pi_\theta(a|s)$
2. fit $\hat{V}_\phi^{\pi_\theta}(s)$ to samples reward sums
3. evaluate $\hat{A}^{\pi_\theta}(s_i, a_i) = r(s_i, a_i) + \gamma \hat{V}_\phi^{\pi_\theta}(s_i') - \hat{V}_\phi^{\pi_\theta}(s_i)$
4. $\nabla_\theta J(\theta) \approx \Sigma_i \nabla_\theta \log \pi_\theta\,(a_i|s_i) \hat{A}^{\pi_\theta}(s_i, a_i)$
5. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

Online actor-critic algorithm:

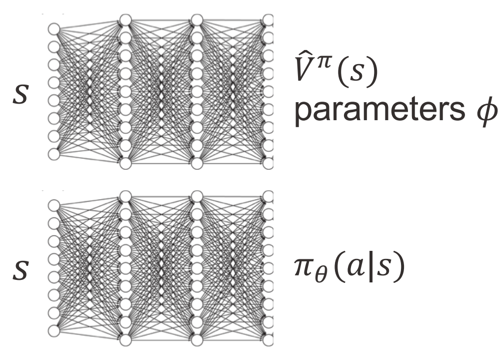
1. take action $a \sim \pi_\theta(a|s)$, get (s, a, s', r)
2. update $\hat{V}_\phi^{\pi_\theta}(s)$ using target $r + \gamma \hat{V}_\phi^{\pi_\theta}(s')$
3. evaluate $\hat{A}^{\pi_\theta}(s, a) = r(s, a) + \gamma \hat{V}_\phi^{\pi_\theta}(s') - \hat{V}_\phi^{\pi_\theta}(s)$
4. $\nabla_\theta J(\theta) \approx \nabla_\theta \log \pi_\theta\,(a|s) \hat{A}^{\pi_\theta}(s, a)$
5. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

Architecture Design

Online actor-critic algorithm:

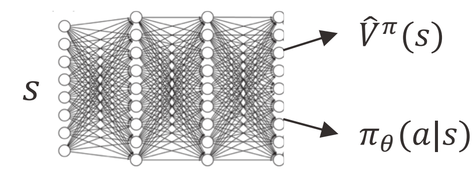
1. take action $a \sim \pi_\theta(a|s)$, get (s, a, s', r)
2. update $\hat{V}_\phi^{\pi_\theta}(s)$ using target $r + \gamma \hat{V}_\phi^{\pi_\theta}(s')$
3. evaluate $\hat{A}^{\pi_\theta}(s,a) = r(s,a) + \gamma \hat{V}_\phi^{\pi_\theta}(s') - \hat{V}_\phi^{\pi_\theta}(s)$
4. $\nabla_\theta J(\theta) \approx \nabla_\theta \log \pi_\theta(a|s) \hat{A}^{\pi_\theta}(s,a)$
5. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

two-network design



+ simple & stable
- no shared features between actor & critic

shared network design



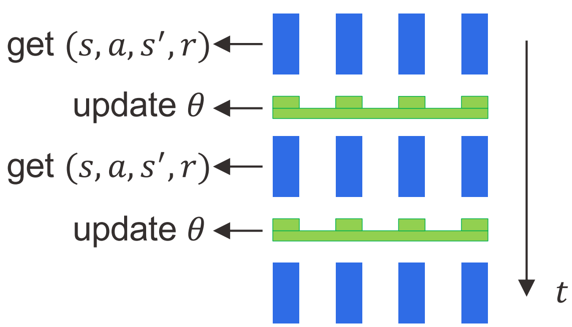
+ less prone to overfitting
- harder to train – different learning rate for each task

Online Actor-critic in Practice – Batch Version

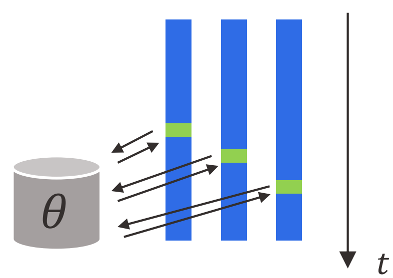
Online actor-critic algorithm:

1. take action $a \sim \pi_\theta(a|s)$, get (s, a, s', r)
2. update $\hat{V}_\phi^{\pi_\theta}(s)$ using target $r + \gamma \hat{V}_\phi^{\pi_\theta}(s')$
3. evaluate $\hat{A}^{\pi_\theta}(s, a) = r(s, a) + \gamma \hat{V}_\phi^{\pi_\theta}(s') - \hat{V}_\phi^{\pi_\theta}(s)$
4. $\nabla_\theta J(\theta) \approx \nabla_\theta \log \pi_\theta (a|s) \hat{A}^{\pi_\theta}(s, a)$
5. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

synchronized parallel actor-critic



asynchronous parallel actor-critic



Critics as State-dependent Baselines

Actor-critic:

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_t^i|s_t^i)\left(r\left(s_t^i, a_t^i\right) + \gamma \hat{V}_\phi^{\pi_\theta}\left(s_t^{i\prime}\right) - \hat{V}_\phi^{\pi_\theta}\left(s_t^i\right)\right)$$

\+ lower variance (due to critic)

\- not unbiased (if the critic is not perfect)

Policy gradient:

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_t^i|s_t^i)\left(\left(\sum_{t'=t}^{T}\gamma^{t'-t} r(s_{t'}^i, a_{t'}^i)\right) - b\right)$$

\+ no bias

\- higher variance (because single-sample estimate)

New variant - best of both worlds

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_t^i|s_t^i)\left(\left(\sum_{t'=t}^{T}\gamma^{t'-t} r(s_{t'}^i, a_{t'}^i)\right) - \hat{V}_\phi^{\pi_\theta}(s_t^i)\right)$$

\+ no bias

\+ lower variance (baseline is closer to rewards)

Average equals to advantage

Action-dependent Baselines for Online

$Q^{\pi}(s_t, a_t) = \sum_{t'=t}^{T} E_{\pi_\theta}[r(s_{t'}, a_{t'}) | s_t, a_t]$ $\quad V^{\pi}(s_t) = E_{a_t \sim \pi_\theta(a_t|s_t)}[Q^{\pi}(s_t, a_t)]$

$A^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t)$

$\hat{A}^{\pi}(s_t, a_t) = \sum_{t'=t}^{T} \gamma^{t'-t} r\left(s_{t'}^i, a_{t'}^i\right) - \hat{V}_{\phi}^{\pi}(s_t)$

+ no bias
- higher variance (because single-sample estimate)

$\hat{A}^{\pi}(s_t, a_t) = \sum_{t'=t}^{T} \gamma^{t'-t} r\left(s_{t'}^i, a_{t'}^i\right) - \hat{Q}_{\phi}^{\pi}(s_t, a_t)$

+ goes to zero in expectation if critic is correct
+ low variance
- biased [any function depending on s,a is biased]; incorrect

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta\left(a_t^i | s_t^i\right) \left(\hat{Q}_{i,t}^{\pi_\theta} - Q_{\phi}^{\pi_\theta}\left(s_t^i, a_t^i\right)\right) + \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta E_{a \sim \pi_\theta(a|s_t^i)} \left[Q_{\phi}^{\pi_\theta}\left(s_t^i, a\right)\right]$$

Use a critic without the bias (still unbiased), provided second term can be evaluated

Eligibility Traces

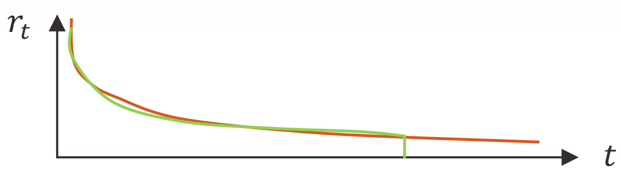
$\hat{A}_C^\pi(s_t, a_t) = r(s_t, a_t) + \gamma \hat{V}_\phi^\pi(s_{t+1}) - \hat{V}_\phi^\pi(s_t)$

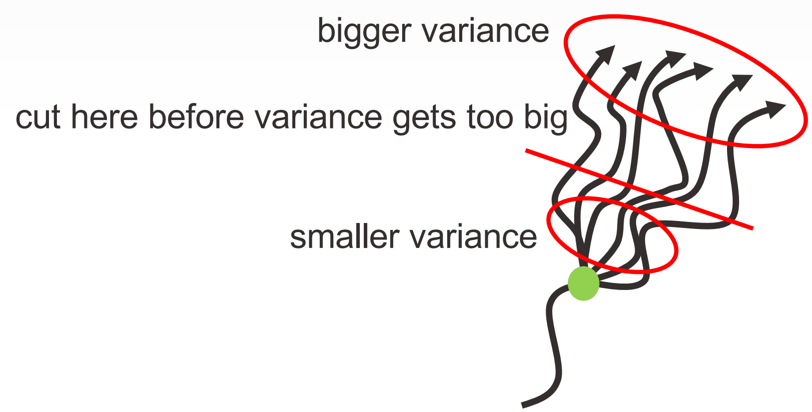
+ lower variance
- higher bias if value is wrong (it always is)

$\hat{A}_{MC}^\pi(s_t, a_t) = \sum_{t'=t}^{T} \gamma^{t'-t} r(s_{t'}, a_{t'}) - \hat{V}_\phi^\pi(s_t)$

+ no bias
- higher variance (because single-sample estimates)

Combine these two to control bias/variance tradeoff:

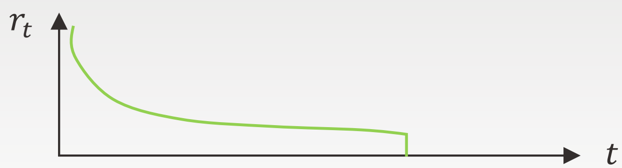




$\hat{A}_n^\pi(s_t, a_t) = \sum_{t'=t}^{t+n} \gamma^{t'-t} r(s_{t'}, a_{t'}) - \hat{V}_\phi^\pi(s_t) + \gamma^n \hat{V}_\phi^\pi(s_{t+n})$

Choosing $n > 1$ often works better

Generalized Advantage Estimation



Do it for many n and then weigh them

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

$$\hat{A}_{GAE}^\pi(s_t, a_t) = \sum_{n=1}^{\infty} w_n \hat{A}_n^\pi(s_t, a_t)$$

weighted combination of n-step returns

Mostly prefer cutting earlier (less variance)

$$w_n \propto \lambda^{n-1}$$

$$\hat{A}_{GAE}^\pi(s_t, a_t) = r(s_t, a_t) + \gamma((1-\lambda)\hat{V}_\phi^\pi(s_{t+1}) + \lambda(r(s_{t+1}, a_{t+1}) + \gamma\left((1-\lambda)\hat{V}_\phi^\pi(s_{t+2}) + \lambda r(s_{t+2}, a_{t+2}) + \ \ldots\right)$$

$$\hat{A}_{GAE}^\pi(s_t, a_t) = \sum_{t'=t}^{\infty} (\gamma\lambda)^{t'-t} \delta_{t'}$$

$$\delta_{t'} = r(s_{t'}, a_{t'}) + \gamma\hat{V}_\phi^\pi(s_{t'+1}) - \hat{V}_\phi^\pi(s_{t'})$$

similar effects as discount!