Lecture 10: Policy Optimization I

Ziyu Shao

School of Information Science and Technology
ShanghaiTech University

April 30, 2025

Outline

Outline

Value-based RL versus Policy-based RL

- In previous lectures, we focused on value-based RL and had value function approximation with parameters $\mathbf{w}$

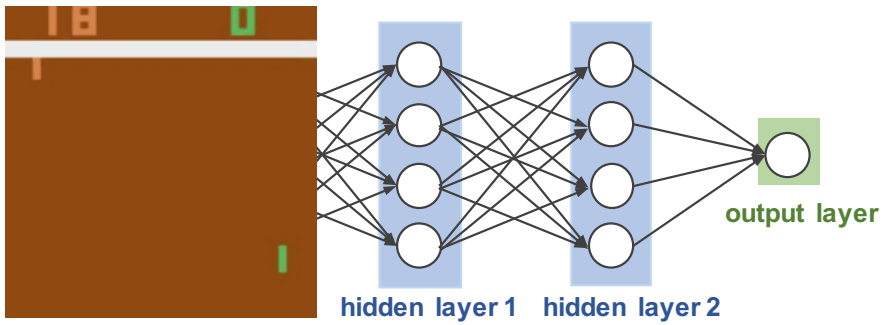
$$V_{\mathbf{w}}(s) \approx v^{\pi}(s)$$
$$q_{\mathbf{w}}(s,a) \approx q^{\pi}(s,a)$$

- Then a policy is generated directly from the value function:
  - using ϵ-greedy or greedy
- Instead, we can also approximate the policy function with parameters θ:

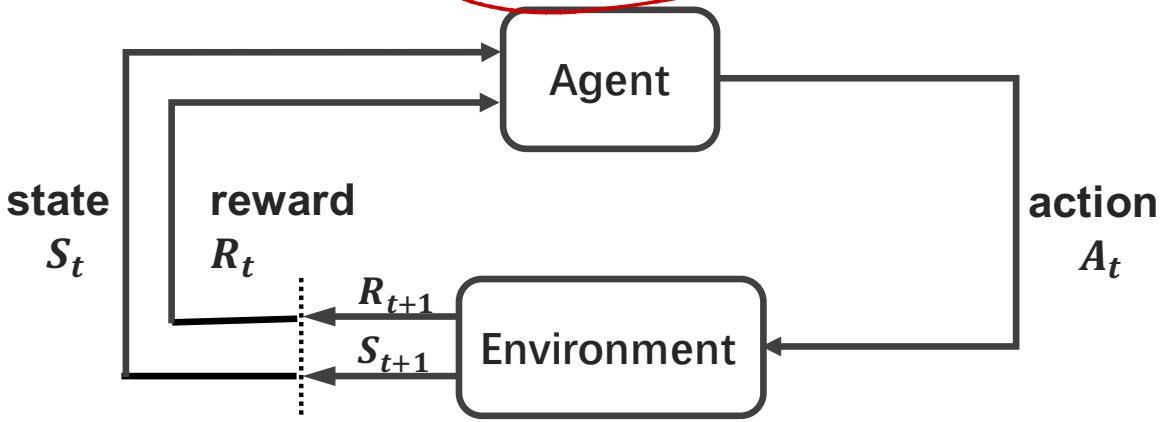
$$\pi_{\theta}(s,a) \approx \pi(a|s)$$

Policy-based RL

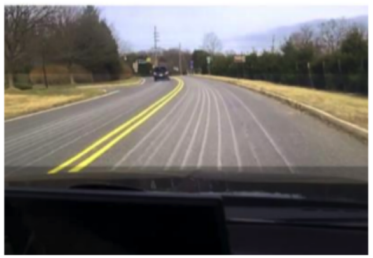
- Making the action is all we care, then let's optimize the policy directly!



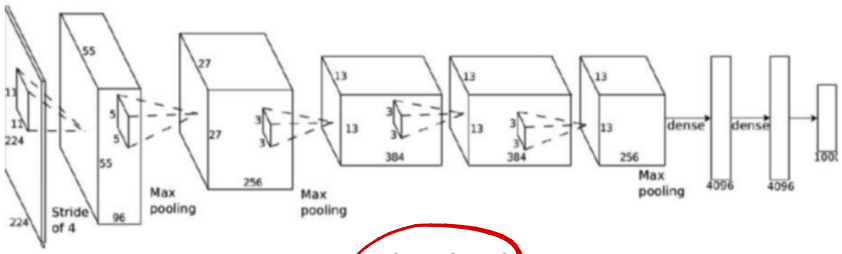
$$\pi_\theta(A_t|S_t)$$



Policy-based RL



$\mathbf{s}_t$



$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$

$\mathbf{a}_t$

$\mathbf{s}_t$

$\mathbf{a}_t$

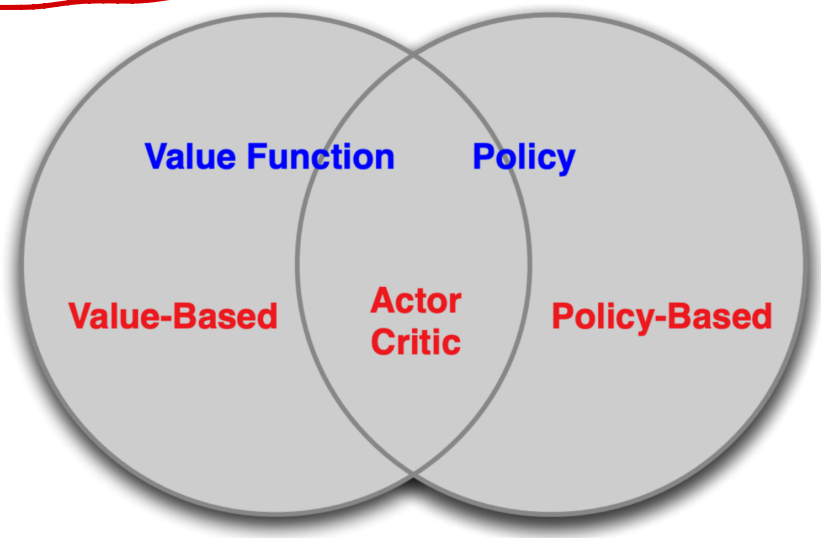
training data

supervised learning

$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$

Value-based RL versus Policy-based RL

- Value-based RL
  - to learn value function
  - implicit policy based on the value function
- Policy-based RL
  - No need to learn value function
  - to learn policy directly
- Actor-critic
  - to learn both policy and value function



Actor-Critic Framework

- Actor: a policy maker that chooses an action given the state
- Critic: the process that determines the contribution (cost or reward) from a policy, from which we can compute a value function
- Actor–critic framework: the interaction of making decisions and updating the value function
- An updating process with two time scales
  - one for the inner iteration to evaluate a policy (critic)
  - and one for the outer iteration where the policy is updated (actor)

Advantages of Policy-based RL

- **Advantages**:
  - Better convergence properties: guaranteed convergence to a local maximum (worst case) or a global maximum (best case)
  - More effective in high-dimensional action space
  - Can learn stochastic policies, while value-based RL can't
- **Disadvantages**:
  - typically converges to a local optimum
  - evaluating a policy is inefficient (in terms of sample efficiency) with high variance

Different Schools of Reinforcement Learning

- Value-based RL
  - Typical applications: games(Go, Starcraft, Texas Hold'em, et al.)
  - Representative algorithms: DQN(Deep Q-learning with Neural Network)
  - Representative person: Richard Sutton, David Silver(from DeepMind)
- Policy-based RL
  - Typical applications: LLM & Robotics
  - Representative algorithms: PPO(Proximal Policy Optimization)
  - Representative person: Pieter Abbeel, Sergey Levine, John Schulman, from OpenAI & Berkeley

Outline

Two Types of Policies

- Deterministic: given a state s, the policy returns the action $a = \pi(s)$ to take
- Stochastic: given a state s, the policy returns an action a with a probability $\pi(a|s)$ to take (e.g., 40% chance to turn left, 60% chance to turn right), equivalently sampling an action $a \sim \pi(\cdot|s)$.

$A \sim \pi(\cdot|s)$

$P(A=a) = \pi(a|s)$

Example: Rock-Paper-Scissors

- Two-player game
- What is the best policy?
  - A deterministic policy is easily beaten
  - A uniform random policy is optimal (Nash equilibrium)

Score Function

- Assume policy π_θ is differentiable whenever it is no-zero
- Given s, a, we can compute the gradient $\nabla_\theta \pi_\theta(s, a)$
- Famous trick

$$\begin{aligned}\nabla_\theta \pi_\theta(s, a) &= \pi_\theta(s, a) \frac{\nabla_\theta \pi_\theta(s, a)}{\pi_\theta(s, a)} \\ &= \pi_\theta(s, a) \nabla_\theta \log \pi_\theta(s, a)\end{aligned}$$

- The score function is $\nabla_\theta \log \pi_\theta(s, a)$

Example in Discrete Action Spaces: Softmax Policy

- Simple policy model: weight actions using linear combination of features $\phi(s,a)^T\theta$
- Probability of action is proportional to the exponentiated weight

$$\pi_\theta(s,a) \propto e^{\phi(s,a)^\top \theta}$$

$$\pi_\theta(s,a) = \frac{e^{\phi(s,a)^T\theta}}{\sum_{a'} e^{\phi(s,a')^T\theta}}$$

- The score function is

$$\begin{aligned}\nabla_\theta \log \pi_\theta(s,a) &= \phi(s,a) - \sum_{a\in\mathcal{A}} \pi_\theta(s,a)\phi(s,a) \\ &= \phi(s,a) - \mathbb{E}_{A\sim\pi_\theta(\cdot|s)}[\phi(s,A)]\end{aligned}$$

Example in Continuous Action Spaces: Gaussian Policy

- In continuous action spaces, a Gaussian policy is natural
- Mean is a linear combination of state features $\mu(s) = \phi(s)^T\theta$
- Variance may be fixed σ^2
- Policy is Gaussian, $A \sim \mathcal{N}(\mu(s), \sigma^2)$
- The score function is

$$\nabla_\theta \log \pi_\theta(s, a) = \frac{(a - \mu(s))\phi(s)}{\sigma^2}$$

Optimizing Policy Value

- Policy-based RL is an optimization problem that find θ to maximize an object function $J(\theta)$
- If $J(\theta)$ is differentiable, we can use gradient-based methods:
  - gradient ascend
  - conjugate gradient
  - quasi-newton
- Some derivative-free black-box optimization methods:
  - Cross-entropy method (CEM)
  - Hill climbing
  - Evolution algorithm

Policy Optimization using Derivative-free Methods

- Sometimes we cannot compute the derivative, i.e., $\nabla_\theta J(\theta)$
- Derivative free methods:
  - Cross Entropy Method (CEM)
  - Finite Difference

Cross-Entropy Method

- $\theta^* = \arg\max J(\theta)$
- Treat $J(\theta)$ as a black box score function (not differentiable)

Algorithm 1 CEM for black-box function optimization

for iter $i = 1$ to N **do**
  $\mathcal{C} = \{\}$
  for parameter set $e = 1$ to N **do**
    sample $\theta^{(e)} \sim P_{\mu^{(i)}}(\theta)$
    execute roll-outs under $\theta^{(e)}$ to evaluate $J(\theta^{(e)})$
    store $(\theta^e, J(\theta^{(e)}))$ in $\mathcal{C}$
  end for
  $\mu^{(i+1)} = \arg\max_\mu \sum_{k \in \hat{\mathcal{C}}} \log P_\mu(\theta^{(k)})$
  where $\hat{\mathcal{C}}$ are the top 10% of $\mathcal{C}$ ranked by $J(\theta^{(e)})$
end for

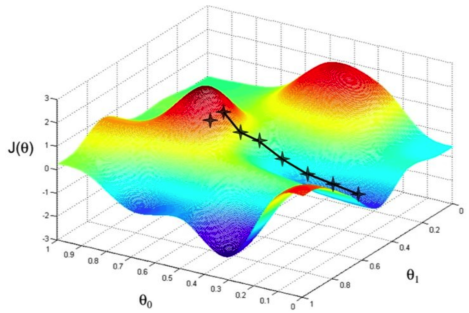
Approximate Gradients by Finite Difference

- To evaluate policy gradient of $\pi_\theta(s, a)$
- For each dimension $k \in [1, n]$
  - estimate kth partial derivative of objective function by perturbing θ by a small amount ϵ in kth dimension

$$\frac{\partial J(\theta)}{\partial \theta_k} \approx \frac{J(\theta + \epsilon u_k) - J(\theta)}{\epsilon}$$

    where u_k is unit vector with 1 in kth component, 0 else where
- uses n evaluations to compute policy gradient in total n dimensions
- though noisy and inefficient, but works for arbitrary policies, even if policy is not differentiable.

Our Focus: Policy Optimization using Gradient (Policy Gradient)



- Consider a function $J(\theta)$ to be any policy objective function
- Goal is to find parameter θ^* that maximizes $J(\theta)$ by ascending the gradient of the policy, w.r.t parameter θ

$$\Delta\theta = \alpha \nabla_\theta J(\theta)$$

- Adjust θ in the direction of the gradient, where α is step-size
- Define the gradient of $J(\theta)$ to be

$$\nabla_\theta J(\theta) = (\frac{\partial J(\theta)}{\partial \theta_1}, \frac{\partial J(\theta)}{\partial \theta_2}, \ldots, \frac{\partial J(\theta)}{\partial \theta_n})^T$$

Outline

Policy Optimization

- For simplicity, we begin from the episodic MDP with length T and discount factor $\gamma = 1$
- Can easily extend to the continuing & infinite horizon case
- Given a policy function π_θ, we sample a random trajectory $\Psi \sim P_\theta$, and its value is denoted as ψ, a state-action-reward trajectory of one episode with length T:

$$\psi = (s_0, a_0, r_1, \ldots, s_{T-1}, a_{T-1}, r_T, s_T)$$

- Corresponding PMF/PDF of Ψ is denoted as $P_\Psi(\psi) \triangleq P_\theta(\psi)$:

$$\begin{aligned} P_\theta(\psi) &= \mu(s_0)\pi_\theta(a_0|s_0)p(s_1|s_0, a_0)\cdots p(s_T|s_{T-1}, a_{T-1}) \\ &= \mu(s_0)\prod_{t=0}^{T-1}\pi_\theta(a_t|s_t)p(s_{t+1}|s_t, a_t) \end{aligned}$$

where $\mu(\cdot)$ is the initial state distribution.

Policy Optimization

- Corresponding accumulated reward along the trajectory ψ is

$$R(\psi) = \sum_{t=1}^{T} r_t.$$

- We want to maximize the expectation of accumulated reward, thus the object function of policy optimization is

$$J(\theta) = \mathbb{E}_{\Psi \sim P_\theta}[R(\Psi)] = \sum_{\psi} P_\theta(\psi) R(\psi)$$

- Policy optimization is $\max_\theta J(\theta)$
- The goal of policy-based RL is to find the θ^*:

$$\theta^* = \arg\max_\theta \mathbb{E}_{\Psi \sim P_\theta}[R(\Psi)] = \arg\max_\theta \sum_{\psi} P_\theta(\psi) R(\psi)$$

Policy Gradient

- Gradient Ascent Method
- Take the gradient with respect to θ:

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \sum_\psi P_\theta(\psi) R(\psi) \\
&= \sum_\psi (\nabla_\theta P_\theta(\psi)) R(\psi) \\
&= \sum_\psi \frac{P_\theta(\psi)}{P_\theta(\psi)} (\nabla_\theta P_\theta(\psi)) R(\psi) \\
&= \sum_\psi P_\theta(\psi) R(\psi) \frac{\nabla_\theta P_\theta(\psi)}{P_\theta(\psi)} \\
&= \sum_\psi P_\theta(\psi) R(\psi) \nabla_\theta \log P_\theta(\psi)
\end{aligned}$$

Policy Gradient

- Policy gradient:

$$\begin{aligned}\nabla_\theta J(\theta) &= \sum_\psi P_\theta(\psi) R(\psi) \nabla_\theta \log P_\theta(\psi) \\ &= \mathbb{E}_{\Psi \sim P_\theta} \left[R(\Psi) \nabla_\theta \log P_\theta(\Psi) \right]\end{aligned}$$

- Approximate with empirical estimate for m trajectories $\psi_1, \ldots, \psi_m$ under policy π_θ:

$$\nabla_\theta J(\theta) \approx \frac{1}{m} \sum_{i=1}^{m} R(\psi_i) \nabla_\theta \log P_\theta(\psi_i)$$

Decomposing the Trajectories

- Decompose $\nabla_\theta \log P_\theta(\psi)$

$$\nabla_\theta \log P_\theta(\psi) = \nabla_\theta \log \left[\mu(s_0) \prod_{t=0}^{T-1} \pi_\theta(a_t|s_t) p(s_{t+1}|s_t, a_t) \right]$$

$$= \nabla_\theta \left[\log \mu(s_0) + \sum_{t=0}^{T-1} (\log \pi_\theta(a_t|s_t) + \log p(s_{t+1}|s_t, a_t)) \right]$$

$$= \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)$$

- Policy gradients are now model-free!

$$\nabla_\theta \log P_\theta(\psi_i) = \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t^i|s_t^i)$$

$$\nabla_\theta J(\theta) \approx \frac{1}{m} \sum_{i=1}^{m} R(\psi_i) \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t^i|s_t^i)$$

Model-Free Policy Gradient

- Policy gradients are now model-free!

$$\nabla_\theta \log P_\theta(\Psi) = \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(A_t|S_t)$$

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \mathbb{E}_{\Psi\sim P_\theta}\left[R(\Psi)\right] \\
&= \mathbb{E}_{\Psi\sim P_\theta}\left[R(\Psi)\nabla_\theta \log P_\theta(\Psi)\right] \\
&= \mathbb{E}_{\Psi\sim P_\theta}\left[R(\Psi)\sum_{t=0}^{T-1}\nabla_\theta \log \pi_\theta(A_t|S_t)\right]
\end{aligned}$$

Score Function Gradient Estimator

- Consider the generic form of $R(\psi)\nabla_\theta \log P_\theta(\psi)$ as

$$\nabla_\theta \mathbb{E}_{X\sim p(x;\theta)}[f(X)] = \mathbb{E}_X[f(X)\nabla_\theta \log p(X;\theta)]$$

- Given sample x, $f(x)$ measures how good the sample x is.
- The direction of $f(x)\nabla_\theta \log p(x;\theta)$ pushes up the log probability of the sample, in proportion to how good it is

Comparison to Maximum Likelihood

- Maximum likelihood Estimation(MLE): given m trajectories $\psi_1, \ldots, \psi_m$ independently sampled from distribution $P_\theta(\cdot)$
- The corresponding log-likelihood function is

$$L(\theta) = \log \prod_{i=1}^{m} P_\theta(\psi_i) = \sum_{i=1}^{m} \log P_\theta(\psi_i)$$

- Thus

$$\arg\max_\theta L(\theta) = \arg\max_\theta \frac{1}{m} L(\theta)$$

- Let

$$J_{ML}(\theta) = \frac{1}{m} L(\theta) = \frac{1}{m} \sum_{i=1}^{m} \log P_\theta(\psi_i)$$

- Then

$$\nabla_\theta J_{ML}(\theta) = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log P_\theta(\psi_i).$$

Comparison to Maximum Likelihood

- Policy gradient:

$$\nabla_\theta J(\theta) \approx \frac{1}{m} \sum_{i=1}^{m} R(\psi_i) \nabla_\theta \log P_\theta(\psi_i)$$

- Interpretation: policy gradients try to increase probability of trajectories with positive accumulated rewards (trail and error method)
- Maximum likelihood:

$$\nabla_\theta J_{ML}(\theta) = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log P_\theta(\psi_i)$$

- Interpretation: MLE try to maximize the probability of trajectories

Large Variance of Policy Gradient

- We have the following approximate update

$$\nabla_\theta J(\theta) \approx \frac{1}{m} \sum_{i=1}^{m} R(\psi_i) \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t^i | s_t^i)$$

- Unbiased but very noisy
- Three fixes:
  - Use temporal causality
  - Include a baseline
  - Include a critic

Outline

Reduce Variance of Policy Gradient using Causality

- We have

$$\nabla_\theta J(\theta) = \mathbb{E}_{\Psi \sim P_\theta} \left[\sum_{t=0}^{T-1} R(\Psi) \nabla_\theta \log \pi_\theta(A_t|S_t) \right]$$

$$\nabla_\theta J(\theta) \approx \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{T-1} R(\psi_i) \nabla_\theta \log \pi_\theta(a_t^i|s_t^i)$$

- This is not reasonable since at any time-step t, the weight of score function $\nabla_\theta \log \pi_\theta(A_t|S_t)$ is the same $R(\Psi) = \sum_{t=1}^{T} R_t$.
- Causality: policy at time t cannot affect reward at time t' ($R_{t'}$) when $t' \leq t$
- Intuition: any time-step t, the weight of score function $\nabla_\theta \log \pi_\theta(A_t|S_t)$ should be the "reward to go function" $\sum_{t'=t+1}^{T} R_{t'}$, this is exactly the return function G_t.

Reduce Variance of Policy Gradient using Causality

- Previously

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\Psi \sim P_\theta}[R(\Psi)] = \mathbb{E}_{\Psi \sim P_\theta}\left[R(\Psi) \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(A_t|S_t)\right]$$

- Since $R(\Psi) = \sum_{t=1}^{T} R_t$, we have

$$\nabla_\theta \mathbb{E}_{\Psi \sim P_\theta}\left[\sum_{t=1}^{T} R_t\right] = \mathbb{E}_{\Psi \sim P_\theta}\left[\left(\sum_{t=1}^{T} R_t\right)\left(\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(A_t|S_t)\right)\right]$$

- We now consider a random trajectory with a single final reward term $R_{t'}$: $t' = T$, and reward $R_t = 0, 1 \le t \le t' - 1$, then we have

$$\nabla_\theta \mathbb{E}_{\Psi \sim P_\theta}[R_{t'}] = \mathbb{E}_{\Psi \sim P_\theta}\left[R_{t'}\left(\sum_{t=0}^{t'-1} \nabla_\theta \log \pi_\theta(A_t|S_t)\right)\right]$$

Reduce Variance of Policy Gradient using Causality

- Now for any t', we have

$$\nabla_\theta \mathbb{E}_{\psi \sim P_\theta}\left[R_{t'}\right] = \mathbb{E}_{\psi \sim P_\theta}\left[R_{t'}\left(\sum_{t=0}^{t'-1} \nabla_\theta \log \pi_\theta(A_t|S_t)\right)\right]$$

- Then we have

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \mathbb{E}_{\psi \sim P_\theta}\left[\sum_{t'=1}^{T} R_{t'}\right] = \sum_{t'=1}^{T} \nabla_\theta \mathbb{E}_{\psi \sim P_\theta}\left[R_{t'}\right] \\
&= \sum_{t'=1}^{T} \mathbb{E}_{\psi \sim P_\theta}\left[R_{t'}\left(\sum_{t=0}^{t'-1} \nabla_\theta \log \pi_\theta(A_t|S_t)\right)\right] \\
&= \mathbb{E}_{\psi \sim P_\theta}\left[\sum_{t'=1}^{T} R_{t'}\left(\sum_{t=0}^{t'-1} \nabla_\theta \log \pi_\theta(A_t|S_t)\right)\right]
\end{aligned}$$

Reduce Variance of Policy Gradient using Causality

- Then we have

$$\nabla_\theta J(\theta) = \mathbb{E}_{\Psi\sim P_\theta}\left[\sum_{t'=1}^{T} R_{t'}\left(\sum_{t=0}^{t'-1}\nabla_\theta \log \pi_\theta(A_t|S_t)\right)\right]$$

$$= \mathbb{E}_{\Psi\sim P_\theta}\left[\sum_{t=0}^{T-1}\nabla_\theta \log \pi_\theta(A_t|S_t)\left(\sum_{t'=t+1}^{T} R_{t'}\right)\right]$$

$$= \mathbb{E}_{\Psi\sim P_\theta}\left[\sum_{t=0}^{T-1}\nabla_\theta \log \pi_\theta(A_t|S_t) G_t\right]$$

Reduce Variance of Policy Gradient using Causality

- Therefore we have

$$\nabla_\theta J(\theta) = \mathbb{E}_{\Psi \sim P_\theta}\left[\sum_{t=0}^{T-1}\left(G_t \cdot \nabla_\theta \log \pi_\theta(A_t|S_t)\right)\right]$$

- $G_t = \sum_{t'=t+1}^{T} R_{t'}$ is the return for a trajectory at time step t, i.e, total accumulated reward starting from time step t.
- Causality: policy at time t' cannot affect reward at time t when $t < t'$
- Then we can have the following estimated update

$$\nabla_\theta J(\theta) \approx \frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{T-1} G_t^{(i)} \cdot \nabla_\theta \log \pi_\theta(a_t^i|s_t^i)$$

Reduce Variance of Policy Gradient using Causality

- Another View

$$\nabla_\theta J(\theta) = \mathbb{E}_{\psi \sim P_\theta} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(A_t|S_t) \left(\sum_{t'=t+1}^{T} R_{t'} \right) \right]$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\psi \sim P_\theta} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(A_t|S_t) \left(\sum_{t'=1}^{T} R_{t'} \right) \right]$$

- Therefore we have the equality

$$0 = \mathbb{E}_{\psi \sim P_\theta} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(A_t|S_t) \left(\sum_{t'=1}^{t} R_{t'} \right) \right]$$

- We can prove the above equality using another method

Reduce Variance of Policy Gradient using Causality

- Notation $H_t = \sum_{t'=1}^{t} R_{t'}$
- Given t:

$$
\begin{aligned}
&\mathbb{E}_{\psi \sim P_\theta}\left[\nabla_\theta \log \pi_\theta(A_t|S_t)\left(\sum_{t'=1}^{t} R_{t'}\right)\right] \\
&= \mathbb{E}_{\psi \sim P_\theta}\left[\nabla_\theta \log \pi_\theta(A_t|S_t) H_t\right] \\
&= \mathbb{E}_{S_{0:t}, A_{0:t-1}}\left[\mathbb{E}_{S_{t+1:T}, A_{t:T-1}}\left[\nabla_\theta \log \pi_\theta(A_t|S_t) H_t | S_{0:t}, A_{0:t-1}\right]\right] \\
&= \mathbb{E}_{S_{0:t}, A_{0:t-1}}\left[\mathbb{E}_{S_{t+1:T}, A_{t:T-1}}\left[\nabla_\theta \log \pi_\theta(A_t|S_t) | S_{0:t}, A_{0:t-1}\right] H_t\right] \\
&= \mathbb{E}_{S_{0:t}, A_{0:t-1}}\left[\mathbb{E}_{S_{t+1:T}, A_{t:T-1}}\left[\nabla_\theta \log \pi_\theta(A_t|S_t) | S_t\right] H_t\right] \\
&= \mathbb{E}_{S_{0:t}, A_{0:t-1}}\left[\mathbb{E}_{A_t \sim \pi_\theta(\cdot|S_t)}\left[\nabla_\theta \log \pi_\theta(A_t|S_t)\right] H_t\right] \\
&= \mathbb{E}_{S_{0:t}, A_{0:t-1}}\left[0 \cdot H_t\right] \\
&= 0
\end{aligned}
$$

Reduce Variance of Policy Gradient using Causality

- Where

$$
\begin{aligned}
&\mathbb{E}_{A_t \sim \pi_\theta(\cdot|S_t)} \left[\nabla_\theta \log \pi_\theta(A_t|S_t) \right] \\
&= \sum_{a_t} \pi_\theta(a_t|S_t) \nabla_\theta \log \pi_\theta(a_t|S_t) \\
&= \sum_{a_t} \pi_\theta(a_t|S_t) \frac{\nabla_\theta \pi_\theta(a_t|S_t)}{\pi_\theta(a_t|S_t)} \\
&= \sum_{a_t} \nabla_\theta \pi_\theta(a_t|S_t) \\
&= \nabla_\theta \left(\sum_{a_t} \pi_\theta(a_t|S_t) \right) \\
&= \nabla_\theta 1 \\
&= 0
\end{aligned}
$$

Reduce Variance of Policy Gradient using Causality

- Then

$$
\begin{aligned}
&\mathbb{E}_{\psi \sim P_\theta}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(A_t|S_t)\left(\sum_{t'=1}^{t} R_{t'}\right)\right] \\
&= \mathbb{E}_{\psi \sim P_\theta}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(A_t|S_t) H_t\right] \\
&= \sum_{t=0}^{T-1} \mathbb{E}_{\psi \sim P_\theta}\left[\nabla_\theta \log \pi_\theta(A_t|S_t) H_t\right] \\
&= \sum_{t=0}^{T-1} 0 \\
&= 0
\end{aligned}
$$

REINFORCE: MC Policy Gradient Algorithm with Discount Factor $\gamma = 1$

- Since

$$\nabla_\theta J(\theta) = \mathbb{E}_{\Psi \sim P_\theta} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(A_t|S_t) G_t \right]$$

- Adopt SDG(stochastic gradient ascent) update: sample a trajectory $s_0, a_0, r_1, s_1, \ldots, r_T, s_T$

$$g_t = \sum_{t'=t+1}^{T} r_{t'} \tag{1}$$

$$\theta \leftarrow \theta + \alpha \sum_{t=0}^{T-1} g_t \nabla_\theta \log \pi_\theta(a_t|s_t) \tag{2}$$

- Loop for $t = 0, 1, \ldots, T-1$

$$\theta \leftarrow \theta + \alpha \cdot g_t \cdot \nabla_\theta \log \pi_\theta(a_t|s_t)$$

REINFORCE: MC Policy Gradient Algorithm with Discount Factor γ

- Consider the discount factor γ

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ (G_t)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

Outline

Reducing Variance using a Baseline

$$\nabla_\theta J(\theta) = \mathbb{E}_{\psi \sim P_\theta} \left[\sum_{t=0}^{T-1} G_t \cdot \nabla_\theta \log \pi_\theta(A_t|S_t) \right] \tag{3}$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\psi \sim P_\theta} \left[\sum_{t=0}^{T-1} (G_t - b(S_t)) \cdot \nabla_\theta \log \pi_\theta(A_t|S_t) \right] \tag{4}$$

- Baseline 1: $b(s)$ depends on state s, NOT depend on action a, can reduce variance, without changing the expectation.
- Baseline 2: $b(s) = b$ and b is a constant.

Reducing Variance using a Baseline

$$\nabla_\theta J(\theta) = \mathbb{E}_{\Psi \sim P_\theta} \left[\sum_{t=0}^{T-1} G_t \cdot \nabla_\theta \log \pi_\theta(A_t|S_t) \right] \tag{5}$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\Psi \sim P_\theta} \left[\sum_{t=0}^{T-1} (G_t - b(S_t)) \cdot \nabla_\theta \log \pi_\theta(A_t|S_t) \right] \tag{6}$$

- We can prove that subtracting a baseline is unbiased in expectation

$$\mathbb{E}_{\Psi \sim P_\theta} \left[\sum_{t=0}^{T-1} b(S_t) \cdot \nabla_\theta \log \pi_\theta(A_t|S_t) \right] = 0$$

Proof: Baseline 1

- Given t:

$$
\begin{aligned}
&\mathbb{E}_{\Psi\sim P_\theta}\left[\nabla_\theta \log \pi_\theta(A_t|S_t) b_{S_t}\right] \\
=&\mathbb{E}_{S_{0:t},A_{0:t-1}}\left[\mathbb{E}_{S_{t+1:T},A_{t:T-1}}\left[\nabla_\theta \log \pi_\theta(A_t|S_t) b_{S_t} | S_{0:t}, A_{0:t-1}\right]\right] \\
=&\mathbb{E}_{S_{0:t},A_{0:t-1}}\left[\mathbb{E}_{S_{t+1:T},A_{t:T-1}}\left[\nabla_\theta \log \pi_\theta(A_t|S_t) | S_{0:t}, A_{0:t-1}\right] b_{S_t}\right] \\
=&\mathbb{E}_{S_{0:t},A_{0:t-1}}\left[\mathbb{E}_{S_{t+1:T},A_{t:T-1}}\left[\nabla_\theta \log \pi_\theta(A_t|S_t) | S_t\right] b_{S_t}\right] \\
=&\mathbb{E}_{S_{0:t},A_{0:t-1}}\left[\mathbb{E}_{A_t\sim\pi_\theta(\cdot|S_t)}\left[\nabla_\theta \log \pi_\theta(A_t|S_t)\right] b_{S_t}\right] \\
=&\mathbb{E}_{S_{0:t},A_{0:t-1}}\left[0 \cdot b_{S_t}\right] \\
=&0
\end{aligned}
$$

Proof: Baseline 1

- Where

$$
\begin{aligned}
&\mathbb{E}_{A_t \sim \pi_\theta(\cdot|S_t)} \left[\nabla_\theta \log \pi_\theta(A_t|S_t) \right] \\
&= \sum_{a_t} \pi_\theta(a_t|S_t) \nabla_\theta \log \pi_\theta(a_t|S_t) \\
&= \sum_{a_t} \pi_\theta(a_t|S_t) \frac{\nabla_\theta \pi_\theta(a_t|S_t)}{\pi_\theta(a_t|S_t)} \\
&= \sum_{a_t} \nabla_\theta \pi_\theta(a_t|S_t) \\
&= \nabla_\theta \left(\sum_{a_t} \pi_\theta(a_t|S_t) \right) \\
&= \nabla_\theta 1 \\
&= 0
\end{aligned}
$$

Proof: Baseline 1

- Then

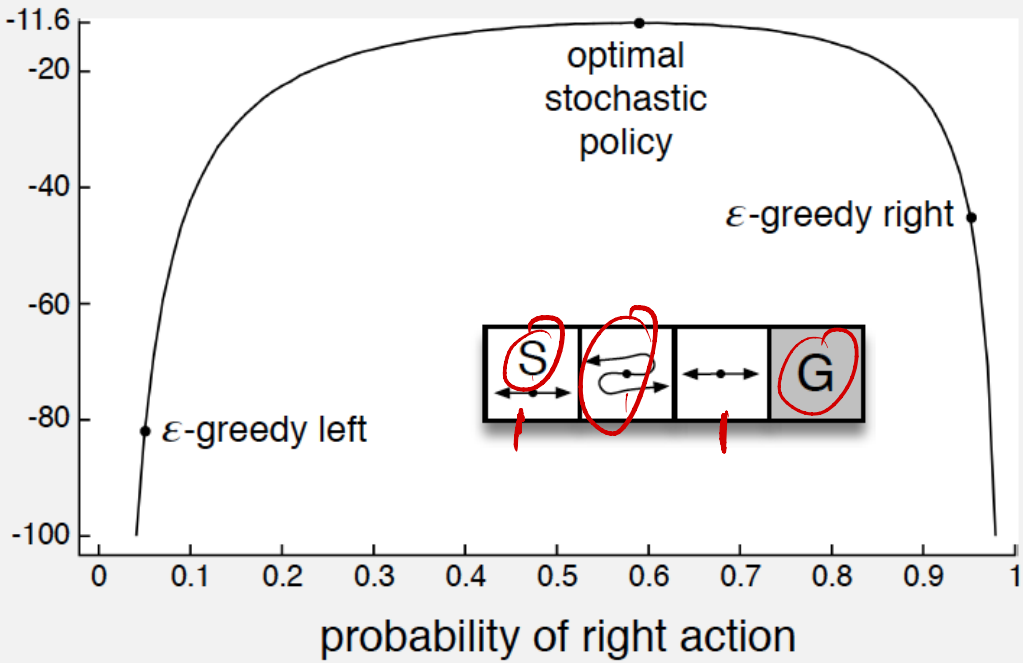
$$\begin{aligned}
&\mathbb{E}_{\psi\sim P_\theta}\left[\sum_{t=0}^{T-1}\nabla_\theta \log \pi_\theta(A_t|S_t)b_{S_t}\right] \\
&= \sum_{t=0}^{T-1}\mathbb{E}_{\psi\sim P_\theta}\left[\nabla_\theta \log \pi_\theta(A_t|S_t)b_{S_t}\right] \\
&= \sum_{t=0}^{T-1} 0 \\
&= 0
\end{aligned}$$

Proof: Baseline 2

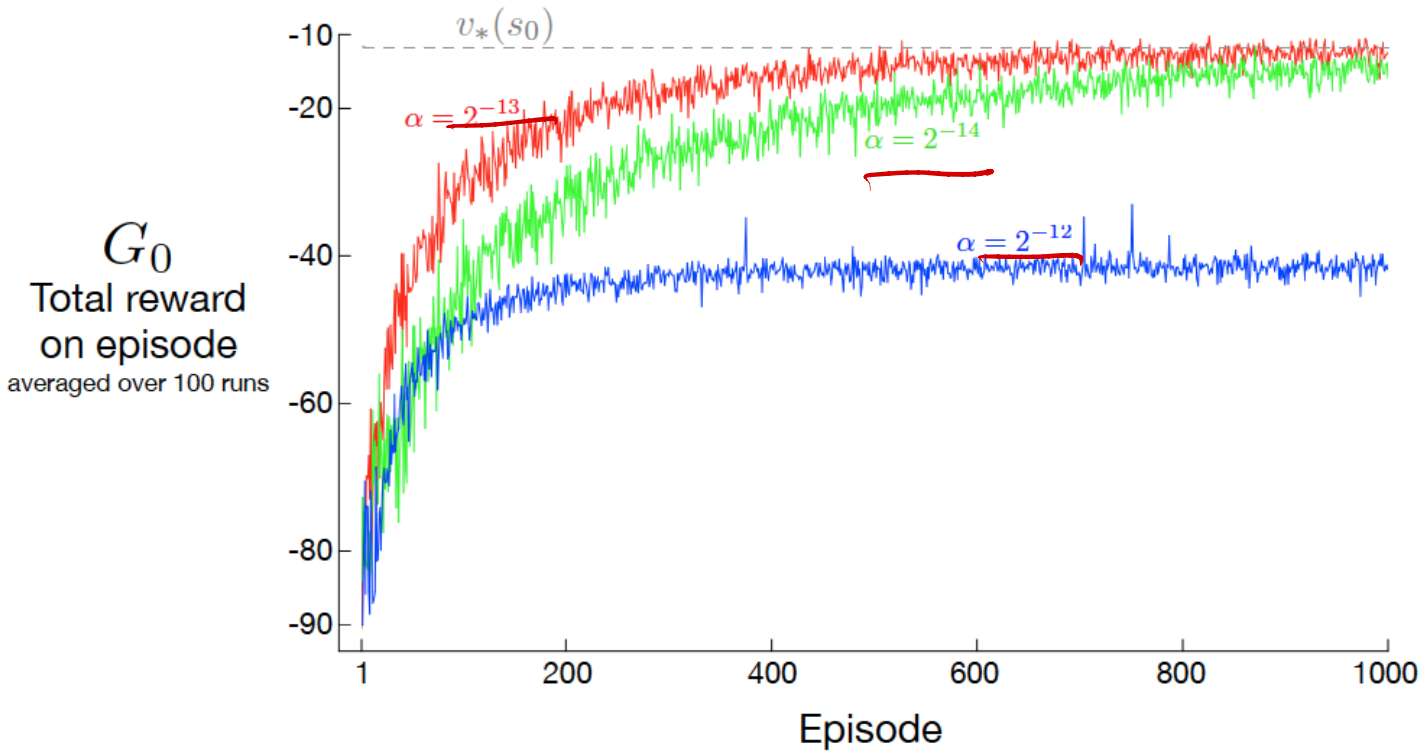
- Similar Proof (You can try it!)

Example: Short Corridor with Switched Actions

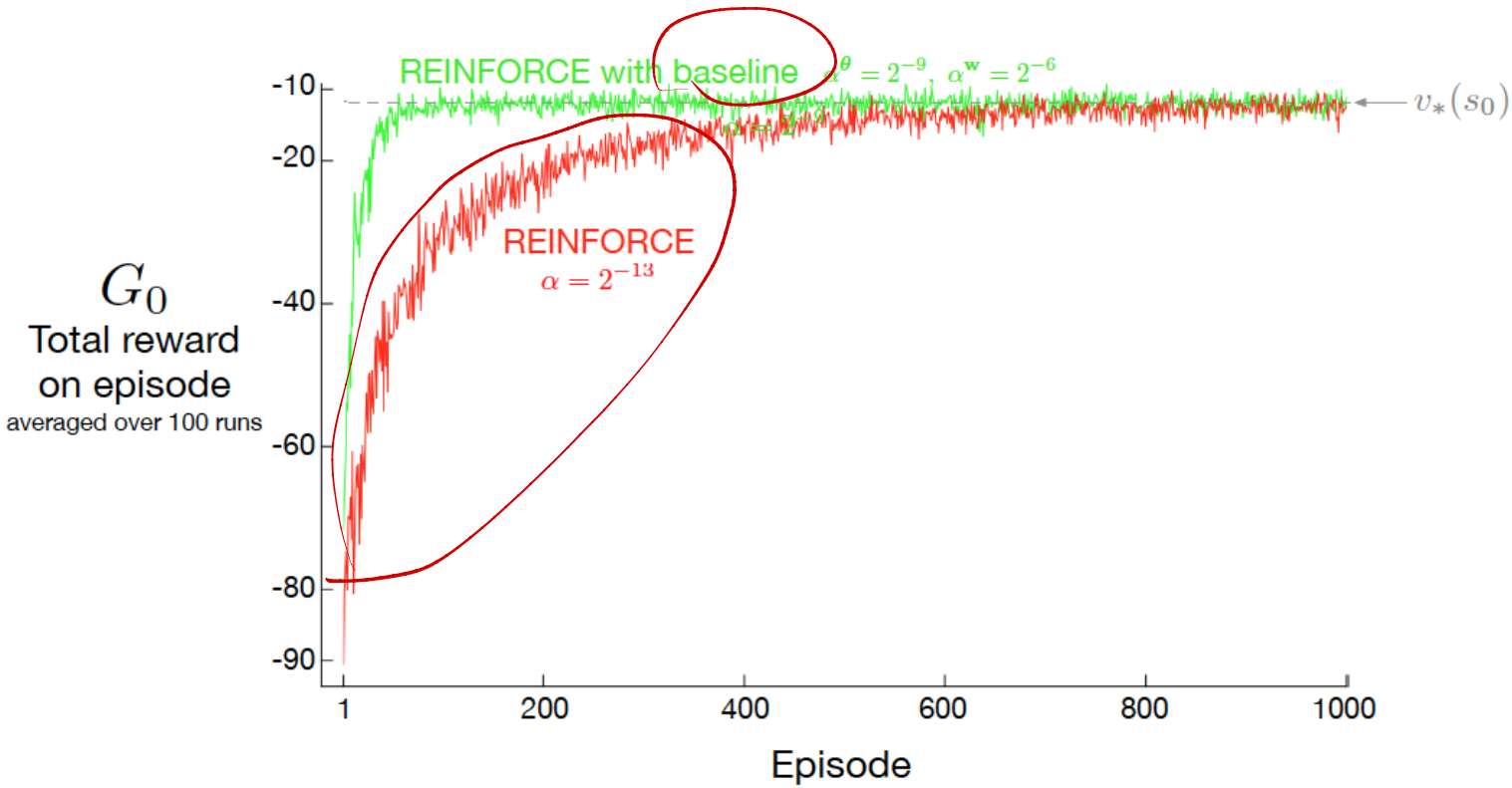
$$J(\boldsymbol{\theta}) = v_{\pi_{\boldsymbol{\theta}}}(\mathrm{S})$$



REINFORCE for Short Corridor with Switched Actions



REINFORCE with Baseline for Short Corridor with Switched Actions



Vanilla Policy Gradient Algorithm with Baseline

procedure POLICY GRADIENT(α)

Initialize policy parameters θ and baseline values $b(s)$ for all s, e.g. to 0

for iteration $= 1, 2, \ldots$ **do**

Collect a set of m trajectories by executing the current policy π_θ

for each time step t of each trajectory $\tau^{(i)}$ **do**

Compute the *return* $G_t^{(i)} = \sum_{t'=t}^{T-1} r_{t'}$

Compute the *advantage estimate* $\hat{A}_t^{(i)} = G_t^{(i)} - b(s_t)$

Re-fit the baseline to the empirical returns by updating $\mathbf{w}$ to minimize

$$\sum_{i=1}^{m} \sum_{t=0}^{T-1} \| b(s_t) - G_t^{(i)} \|^2$$

Update policy parameters θ using the policy gradient estimate $\hat{g}$

$$\hat{g} = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{T-1} \hat{A}_t^{(i)} \nabla_\theta \log \pi_\theta(a_t^{(i)} | s_t^{(i)})$$

with an optimizer like SGD ($\theta \leftarrow \theta + \alpha \cdot \hat{g}$) or Adam

return θ and baseline values $b(s)$

Practical Implementation of the Algorithm

- In practice we usually do not compute the gradients $\sum_t \hat{A}_t \nabla_\theta \log \pi_\theta(a_t|s_t)$ individually
- Instead, we accumulate data from current batch as

$$L(\theta) = \sum_t \hat{A}_t \log \pi_\theta(a_t|s_t;\theta)$$

- Then the policy gradient estimator $\hat{g} = \nabla_\theta L(\theta)$
- We also could have a joint loss with value function approximation as

$$L(\theta, \mathbf{w}) = \sum_t \left(\hat{A}_t \log \pi_\theta(a_t|s_t;\theta) - \|b(s_t) - \hat{R}_t\|^2 \right)$$

- Then solve this using auto diff

Outline

Main References

- Reinforcement Learning: An Introduction (second edition), R. Sutton & A. Barto, 2018.
- RL course slides from Richard Sutton, University of Alberta.
- RL course slides from David Silver, University College London.
- RL course slides from Sergey Levine, UC Berkeley
- RL course slides from Shusen Wang