Policy Gradient: Reinforce

Recap: two definitions of MDPs

$$\mathcal{M} = \{P, r, \gamma, \mu, S, A\}$$

where $s_0 \sim \mu$

Objective: $J(\pi) := \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h)\right]$

Recap: two definitions of MDPs

$$\mathcal{M} = \{P, r, \gamma, \mu, S, A\}$$

where $s_0 \sim \mu$

$$\text{Objective: } J(\pi) := \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h)\right]$$

$$\mathcal{M} = \{P, r, H, \mu, S, A\}$$

where $s_0 \sim \mu$

$$\text{Objective: } J(\pi) := \mathbb{E}\left[\sum_{h=0}^{H-1} r(s_h, a_h)\right]$$

Recap: value-based learning

Deep Q network (DQN)

$$\theta \Leftarrow \theta - \eta \frac{1}{|\mathcal{B}|} \sum_{s,a,r,s' \in \mathcal{B}} \left(Q_\theta(s,a) - r - \gamma \max_{a'} Q_{\tilde{\theta}}(s',a') \right) \nabla_\theta Q_\theta(s,a)$$

Δ

Recap: value-based learning

Deep Q network (DQN)

$$\theta \Leftarrow \theta - \eta \frac{1}{|\mathscr{B}|} \sum_{s,a,r,s' \in \mathscr{B}} \left(Q_\theta(s,a) - r - \gamma \max_{a'} Q_{\tilde{\theta}}(s',a') \right) \nabla_\theta Q_\theta(s,a)$$

Avg over a mini-batch
sampled from a replay buffer

$$\begin{bmatrix} \vdots \\ (s, a, r, s') \\ \vdots \end{bmatrix}$$

Recap: value-based learning

Deep Q network (DQN)

$\tilde{\theta}$

$$\theta \Leftarrow \theta - \eta \underbrace{\frac{1}{|\mathscr{B}|} \sum_{s,a,r,s' \in \mathscr{B}}} \left(Q_\theta(s,a) - r - \gamma \max_{a'} Q_{\tilde{\theta}}(s',a') \right) \nabla_\theta Q_\theta(s,a)$$

Avg over a mini-batch sampled from a replay buffer

Target network being updated once every C steps of gradient updates

Recap: value-based learning

Issues of Q-learning

1. In worst case, it **cannot** even guarantee convergence when function approximator is used

$Q_\theta \Rightarrow TQ_{\tilde{\theta}}$

2-layer

Q-Target

Two-layer

$\| Q_\theta - TQ_\theta \|_\infty = 0$

Recap: value-based learning

Issues of Q-learning

1. In worst case, it **cannot** even guarantee convergence when function approximator is used

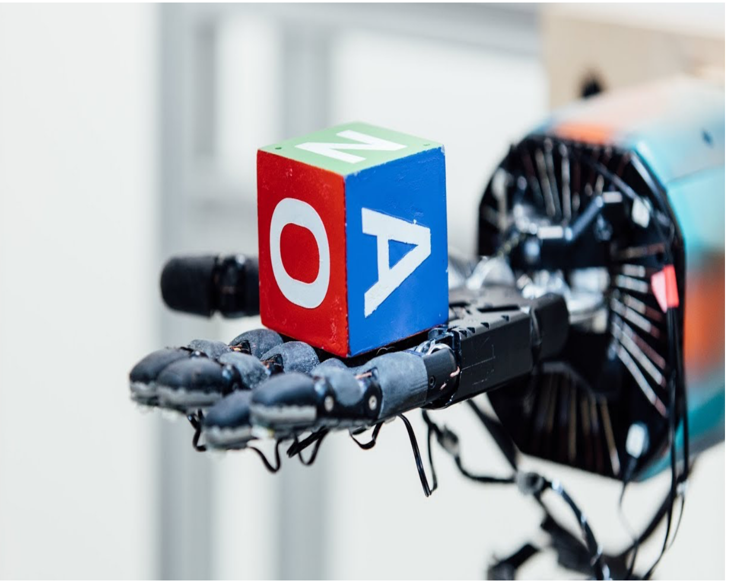
2. Q-learning is not a direct approach — It does not **directly** optimize the ultimate objective: maximizing expected total reward

$$\arg\max_a Q_\theta(s, a) \approx \pi^*$$

Today

Reinforce: policy gradient algorithm that can directly optimizes the ultimate objective

Policy gradient is very popular in practice



Robotics

ChatGPT

Generative AI

Outline for today

1. Warm up: computing gradient using importance weighting

2. Policy Gradient formulation

Warm Up: Importance Weighting

$\max_{\theta} J(\theta)$

$$J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right]$$

Warm Up: Importance Weighting

$$J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] \qquad \nabla_\theta J(\theta) = \underline{\nabla_\theta \mathbb{E}_{x\sim P_\theta} f(x)}$$

$$= \nabla_\theta \int_\theta P_\theta(x)\, f(x)\, d\theta$$

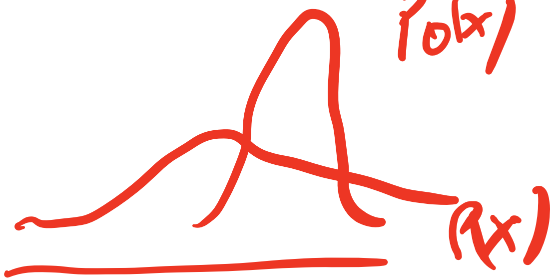
$$= \int_\theta \nabla_\theta P_\theta(x)\, f(x)\, d\theta$$

Warm Up: Importance Weighting

$$J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] \qquad \nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{x \sim P_\theta} f(x)$$

$\rho(x)$

Suppose that I have a sampling distribution ρ, s.t., $\max_x P_\theta(x)/\rho(x) < \infty$



Warm Up: Importance Weighting

$$J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] \qquad \nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{x \sim P_\theta} f(x)$$

Suppose that I have a sampling distribution ρ, s.t., $\max_x P_\theta(x)/\rho(x) < \infty$

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{x \sim P_\theta} f(x) = \nabla_\theta \mathbb{E}_{x \sim \rho} \frac{P_\theta(x)}{\rho(x)} f(x)$$

$$= \nabla_\theta \int_x \cancel{\rho(x)} \cdot \frac{P_\theta(x)}{\cancel{\rho(x)}} f(x)$$

Warm Up: Importance Weighting

$$J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] \qquad \nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{x \sim P_\theta} f(x)$$

Suppose that I have a sampling distribution ρ, s.t., $\max_x P_\theta(x)/\rho(x) < \infty$

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{x \sim P_\theta} f(x) = \nabla_\theta \mathbb{E}_{x \sim \rho} \frac{P_\theta(x)}{\rho(x)} f(x) \quad = \mathbb{E}_{x \sim \rho} \frac{\nabla_\theta P_\theta(x)}{\rho(x)} f(x)$$

$x_1 \cdots x_n \overset{\text{i.i.d}}{\sim} \rho$

$$\frac{1}{n} \sum_{i=1}^{n} \frac{\nabla_\theta P_\theta(x_i)}{\rho(x_i)} f(x_i)$$

Warm Up: Importance Weighting

$$J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] \qquad \nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{x \sim P_\theta} f(x)$$

Suppose that I have a sampling distribution ρ, s.t., $\max_x P_\theta(x)/\rho(x) < \infty$

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{x \sim P_\theta} f(x) = \nabla_\theta \mathbb{E}_{x \sim \rho} \frac{P_\theta(x)}{\rho(x)} f(x) = \mathbb{E}_{x \sim \rho} \frac{\nabla_\theta P_\theta(x)}{\rho(x)} f(x) \approx \underline{\frac{1}{N} \sum_{i=1}^{N} \frac{\nabla_\theta P_\theta(x_i)}{\rho(x_i)} f(x_i)}$$

Unbiased Estimate of $\nabla_\theta J(\theta)$

Warm Up: Importance Weighting

$$J(\theta) = \mathbb{E}_{x\sim P_\theta}\left[f(x)\right] \qquad \nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{x\sim P_\theta} f(x)$$

Suppose that I have a sampling distribution ρ, s.t., $\max_x P_\theta(x)/\rho(x) < \infty$

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{x\sim P_\theta} f(x) = \nabla_\theta \mathbb{E}_{x\sim\rho} \frac{P_\theta(x)}{\rho(x)} f(x) \;= \mathbb{E}_{x\sim\rho} \frac{\nabla_\theta P_\theta(x)}{\rho(x)} f(x) \;\approx \frac{1}{N}\sum_{i=1}^{N} \frac{\nabla_\theta P_\theta(x_i)}{\rho(x_i)} f(x_i)$$

θ_0

To compute gradient at θ_0: $\nabla_\theta J(\theta_0)$ (in short of $\nabla_\theta J(\theta)\,|_{\theta=\theta_0}$)

Warm Up: Importance Weighting

$$J(\theta) = \mathbb{E}_{x\sim P_\theta}\left[f(x)\right] \qquad \nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{x\sim P_\theta} f(x)$$

Suppose that I have a sampling distribution ρ, s.t., $\max_x P_\theta(x)/\rho(x) < \infty$

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{x\sim P_\theta} f(x) = \nabla_\theta \mathbb{E}_{x\sim\rho} \frac{P_\theta(x)}{\rho(x)} f(x) = \mathbb{E}_{x\sim\rho} \frac{\nabla_\theta P_\theta(x)}{\rho(x)} f(x) \approx \frac{1}{N}\sum_{i=1}^{N} \frac{\nabla_\theta P_\theta(x_i)}{\rho(x_i)} f(x_i)$$

To compute gradient at θ_0: $\nabla_\theta J(\theta_0)$ (in short of $\nabla_\theta J(\theta)|_{\theta=\theta_0}$)

We can set sampling distribution $\rho = P_{\theta_0}$

$$\mathbb{E}_{x\sim P_{\theta_0}} \frac{\nabla_\theta P_\theta(x)|_{\theta=\theta_0}}{P_{\theta_0}(x)} f(x) = \mathbb{E}_{x\sim P_{\theta_0}} \left[\nabla_\theta \ln \pi_\theta(x)|_{\theta=\theta_0} f(x)\right]$$

Warm Up: Importance Weighting

$$J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] \qquad \nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{x \sim P_\theta} f(x)$$

Suppose that I have a sampling distribution ρ, s.t., $\max_x P_\theta(x)/\rho(x) < \infty$

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{x \sim P_\theta} f(x) = \nabla_\theta \mathbb{E}_{x \sim \rho} \frac{P_\theta(x)}{\rho(x)} f(x) = \mathbb{E}_{x \sim \rho} \frac{\nabla_\theta P_\theta(x)}{\rho(x)} f(x) \approx \frac{1}{N} \sum_{i=1}^{N} \frac{\nabla_\theta P_\theta(x_i)}{\rho(x_i)} f(x_i)$$

To compute gradient at θ_0: $\nabla_\theta J(\theta_0)$ (in short of $\nabla_\theta J(\theta)|_{\theta=\theta_0}$)

We can set sampling distribution $\rho = P_{\theta_0}$

$$\nabla_\theta J(\theta_0) = \mathbb{E}_{x \sim P_{\theta_0}} \left[\nabla_\theta \ln P_{\theta_0}(x) \cdot f(x) \right]$$

$x \sim P_{\theta_0}$

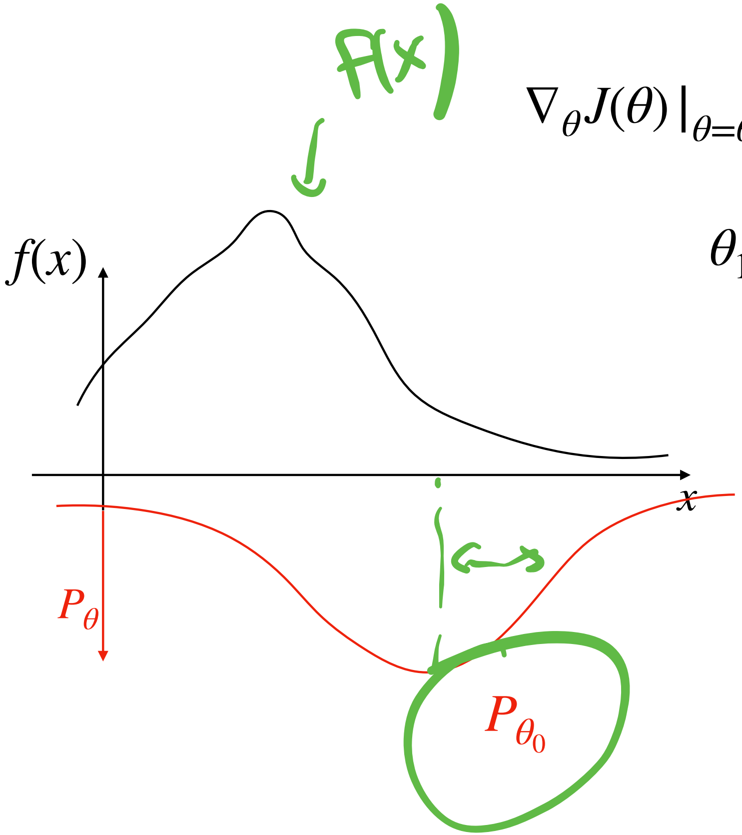
$\nabla_\theta \ln P_{\theta_0}(x) \cdot f(x)$

Warm Up

$$\nabla_\theta J(\theta)\,|_{\theta=\theta_0} = \mathbb{E}_{x\sim P_{\theta_0}} \nabla_\theta \ln P_{\theta_0}(x)\, f(x)$$

$$\mathbb{E}_{x\sim P_\theta} f(x)$$

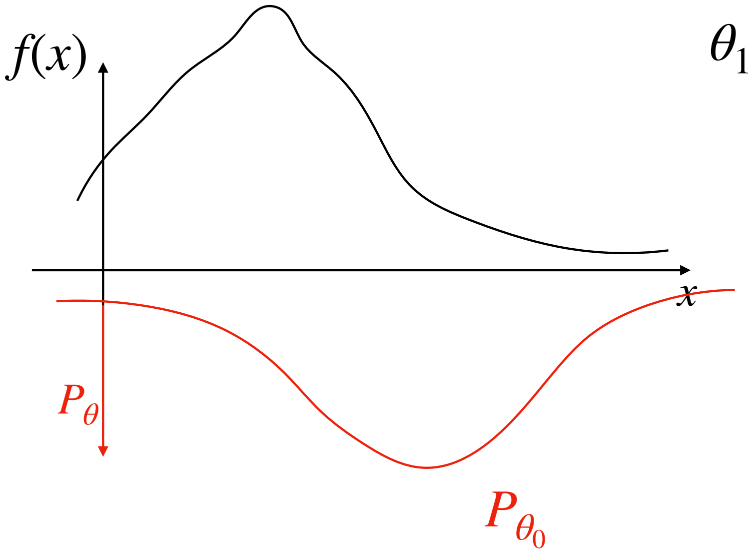
$$\theta_1 = \theta_0 + \eta \nabla_\theta J(\theta_0)$$



Warm Up

$$\nabla_\theta J(\theta)\,|_{\theta=\theta_0} = \mathbb{E}_{x\sim P_{\theta_0}} \nabla_\theta \ln P_{\theta_0}(x)\cdot f(x)$$

$$\theta_1 = \theta_0 + \eta \nabla_\theta J(\theta_0)$$

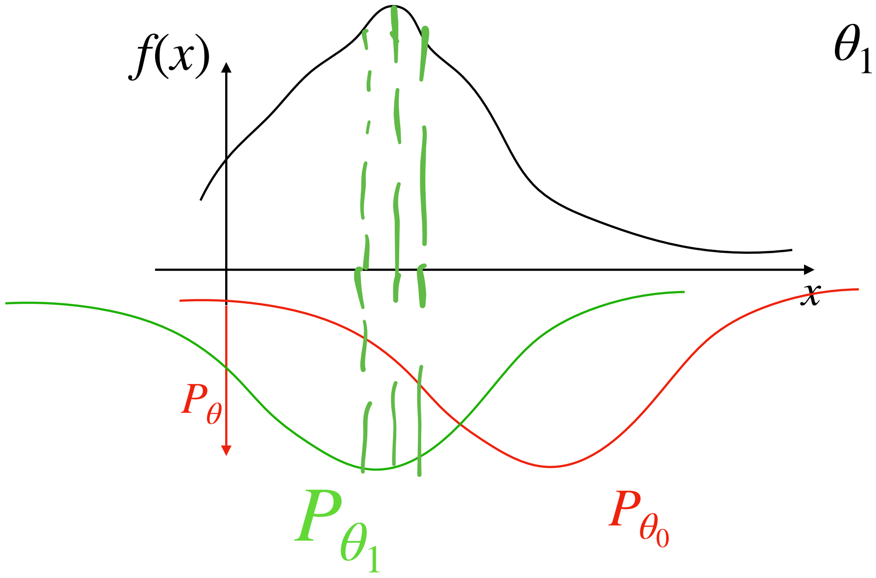


Q: how would this distribution move during the update?

Warm Up

$$\nabla_\theta J(\theta)\,|_{\theta=\theta_0} = \mathbb{E}_{x\sim P_{\theta_0}} \nabla_\theta \ln P_{\theta_0}(x)\cdot f(x)$$

$$\theta_1 = \theta_0 + \eta\,\nabla_\theta J(\theta_0)$$

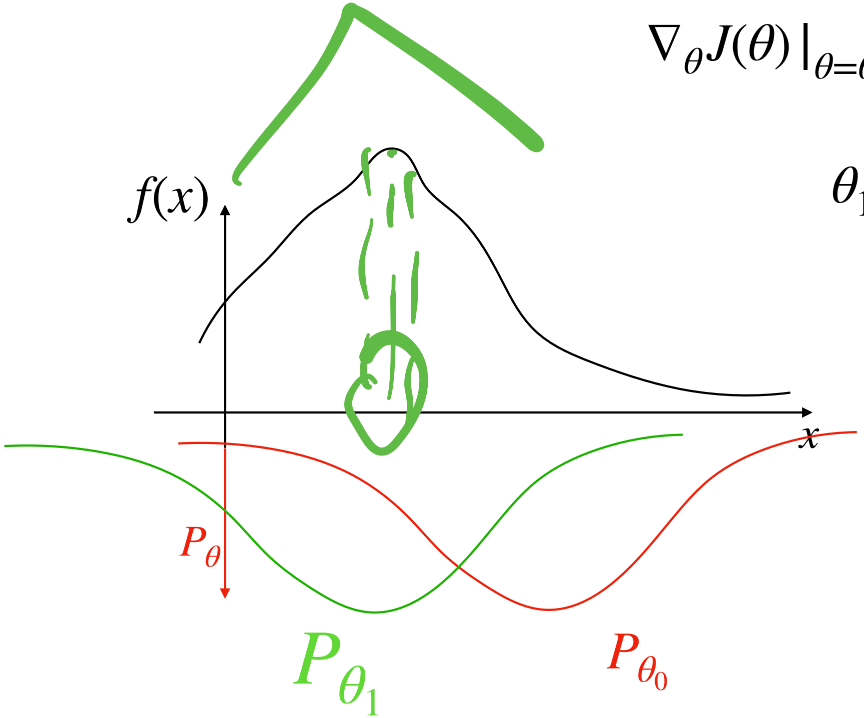


Q: how would this distribution move during the update?

Warm Up

$$\nabla_\theta J(\theta)\,|_{\theta=\theta_0} = \mathbb{E}_{x\sim P_{\theta_0}} \nabla_\theta \ln P_{\theta_0}(x) \cdot f(x)$$

$$\theta_1 = \theta_0 + \eta \nabla_\theta J(\theta_0)$$



Update distribution (via updating θ) such that P_θ has higher probability mass at regions where $f(x)$ is larger

Q: how would this distribution move during the update?

Outline for today

✓ 1. Warm up: computing gradient using importance weighting

2. Policy Gradient formulation

Policy Gradient: Examples of Policy Parameterization (discrete actions)

Recall that we consider parameterized policy $\pi_\theta(\cdot \mid s) \in \Delta(A), \forall s$

$\nabla_\theta \ln \pi_\theta(a|s)$

1. Softmax Policy for discrete MDPs:

$\theta_{s,a} \in \mathbb{R}, \forall s, a \in S \times A$

Policy Gradient: Examples of Policy Parameterization (discrete actions)

Recall that we consider parameterized policy $\pi_\theta(\cdot \mid s) \in \Delta(A), \forall s$

1. Softmax Policy for discrete MDPs:

$\theta_{s,a} \in \mathbb{R}, \forall s, a \in S \times A$

$$\pi_\theta(a \mid s) = \frac{\exp(\theta_{s,a})}{\sum_{a'} \exp(\theta_{s,a'})}$$

$\nabla_\theta \pi_\theta(a \mid s)$

Policy Gradient: Examples of Policy Parameterization (discrete actions)

Recall that we consider parameterized policy $\pi_\theta(\cdot \mid s) \in \Delta(A), \forall s$

1. Softmax Policy for discrete MDPs:

$$\theta_{s,a} \in \mathbb{R}, \forall s, a \in S \times A$$

$$\pi_\theta(a \mid s) = \frac{\exp(\theta_{s,a})}{\sum_{a'} \exp(\theta_{s,a'})}$$

2. Softmax linear Policy

Policy Gradient: Examples of Policy Parameterization (discrete actions)

Recall that we consider parameterized policy $\pi_\theta(\cdot \mid s) \in \Delta(A), \forall s$

1. Softmax Policy for discrete MDPs:

$$\theta_{s,a} \in \mathbb{R}, \forall s, a \in S \times A$$

$$\pi_\theta(a \mid s) = \frac{\exp(\theta_{s,a})}{\sum_{a'} \exp(\theta_{s,a'})}$$

2. Softmax linear Policy

Feature vector $\phi(s, a) \in \mathbb{R}^d$, and parameter $\theta \in \mathbb{R}^d$

Policy Gradient: Examples of Policy Parameterization (discrete actions)

Recall that we consider parameterized policy $\pi_\theta(\cdot \mid s) \in \Delta(A), \forall s$

1. Softmax Policy for discrete MDPs:

$$\theta_{s,a} \in \mathbb{R}, \forall s, a \in S \times A$$

$$\pi_\theta(a \mid s) = \frac{\exp(\theta_{s,a})}{\sum_{a'} \exp(\theta_{s,a'})}$$

2. Softmax linear Policy

Feature vector $\phi(s, a) \in \mathbb{R}^d$, and parameter $\theta \in \mathbb{R}^d$

$$\pi_\theta(a \mid s) = \frac{\exp(\theta^\top \phi(s, a))}{\sum_{a'} \exp(\theta^\top \phi(s, a'))}$$

$\theta^T \phi(s,a)$

$\Leftrightarrow f_\theta(s,a)$

Policy Gradient: Examples of Policy Parameterization (discrete actions)

Recall that we consider parameterized policy $\pi_\theta(\cdot \mid s) \in \Delta(A), \forall s$

1. Softmax Policy for discrete MDPs:

$$\theta_{s,a} \in \mathbb{R}, \forall s, a \in S \times A$$

$$\pi_\theta(a \mid s) = \frac{\exp(\theta_{s,a})}{\sum_{a'} \exp(\theta_{s,a'})}$$

2. Softmax linear Policy

Feature vector $\phi(s, a) \in \mathbb{R}^d$, and parameter $\theta \in \mathbb{R}^d$

$$\pi_\theta(a \mid s) = \frac{\exp(\theta^\top \phi(s, a))}{\sum_{a'} \exp(\theta^\top \phi(s, a'))}$$

3. Neural Policy:

Policy Gradient: Examples of Policy Parameterization (discrete actions)

Recall that we consider parameterized policy $\pi_\theta(\cdot \mid s) \in \Delta(A), \forall s$

1. Softmax Policy for discrete MDPs:

$$\theta_{s,a} \in \mathbb{R}, \forall s, a \in S \times A$$

$$\pi_\theta(a \mid s) = \frac{\exp(\theta_{s,a})}{\sum_{a'} \exp(\theta_{s,a'})}$$

2. Softmax linear Policy

Feature vector $\phi(s, a) \in \mathbb{R}^d$, and parameter $\theta \in \mathbb{R}^d$

$$\pi_\theta(a \mid s) = \frac{\exp(\theta^\top \phi(s, a))}{\sum_{a'} \exp(\theta^\top \phi(s, a'))}$$

3. Neural Policy:

Neural network $f_\theta : S \times A \mapsto \mathbb{R}$

Policy Gradient: Examples of Policy Parameterization (discrete actions)

Recall that we consider parameterized policy $\pi_\theta(\cdot \mid s) \in \Delta(A), \forall s$

1. Softmax Policy for discrete MDPs:

$$\theta_{s,a} \in \mathbb{R}, \forall s, a \in S \times A$$

$$\pi_\theta(a \mid s) = \frac{\exp(\theta_{s,a})}{\sum_{a'} \exp(\theta_{s,a'})}$$

2. Softmax linear Policy

Feature vector $\phi(s,a) \in \mathbb{R}^d$, and parameter $\theta \in \mathbb{R}^d$

$$\pi_\theta(a \mid s) = \frac{\exp(\theta^\top \phi(s,a))}{\sum_{a'} \exp(\theta^\top \phi(s,a'))}$$

3. Neural Policy:

Neural network $f_\theta : S \times A \mapsto \mathbb{R}$

$$\pi_\theta(a \mid s) = \frac{\exp(f_\theta(s,a))}{\sum_{a'} \exp(f_\theta(s,a'))}$$

Policy Gradient: Examples of Policy Parameterization (discrete actions)

Recall that we consider parameterized policy $\pi_\theta(\cdot \mid s) \in \underline{\Delta(A)}, \forall s$

1. Softmax Policy for discrete MDPs:

$$\theta_{s,a} \in \mathbb{R}, \forall s, a \in S \times A$$

$$\pi_\theta(a \mid s) = \frac{\exp(\theta_{s,a})}{\sum_{a'} \exp(\theta_{s,a'})}$$

2. Softmax linear Policy

Feature vector $\phi(s, a) \in \mathbb{R}^d$, and parameter $\theta \in \mathbb{R}^d$

$$\pi_\theta(a \mid s) = \frac{\exp(\theta^\top \phi(s, a))}{\sum_{a'} \exp(\theta^\top \phi(s, a'))}$$

3. Neural Policy:

Neural network $f_\theta : S \times A \mapsto \mathbb{R}$

$$\pi_\theta(a \mid s) = \frac{\exp(f_\theta(s, a))}{\sum_{a'} \exp(f_\theta(s, a'))}$$

In high level, think about π_θ as a classifier which has its parameters to be optimized

Derivation of Policy Gradient: REINFORCE

$$\tau = \{s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}\}$$

Derivation of Policy Gradient: REINFORCE

$\mu_0 \quad s_0 \sim \mu_0$

$a \sim \pi_\theta(\cdot | s)$

$s' \sim P(\cdot | s, a)$

$$\tau = \{s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}\}$$

What's the likelihood of trajectory τ under π_θ?

$\mu_0(s_0) \pi_\theta(a_0 | s_0) P(s_1 | s_0, a_0) \pi_\theta(a_1 | s_1)$

$P(s_2 | s_1, a_1)$

$\cdots P(s_{H-1} | s_{H-2}, a_{H-2})$

$\cdot \pi_\theta(a_{H-2} | s_{H-2})$

Derivation of Policy Gradient: REINFORCE

$$\tau = \{s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}\}$$

What's the likelihood of trajectory τ under π_θ?

$$\rho_\theta(\tau) = \mu(s_0)\pi_\theta(a_0 \mid s_0)P(s_1 \mid s_0, a_0)\pi_\theta(a_1 \mid s_1)\ldots$$

Derivation of Policy Gradient: REINFORCE

$$\tau = \{s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}\}$$

What's the likelihood of trajectory τ under π_θ?

$$\rho_\theta(\tau) = \mu(s_0)\pi_\theta(a_0 \mid s_0)P(s_1 \mid s_0, a_0)\pi_\theta(a_1 \mid s_1)\ldots$$

Rewrite the objective using the traj distribution:

$$J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \underbrace{\left[\sum_{h=0}^{H-1} r(s_h, a_h) \right]}_{R(\tau)}$$

Derivation of Policy Gradient: REINFORCE

$$\tau = \{s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}\}$$

What's the likelihood of trajectory τ under π_θ?

$$\rho_\theta(\tau) = \mu(s_0)\pi_\theta(a_0 \mid s_0)P(s_1 \mid s_0, a_0)\pi_\theta(a_1 \mid s_1)\ldots$$

Rewrite the objective using the traj distribution:

$$J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \underbrace{\left[\sum_{h=0}^{H-1} r(s_h, a_h)\right]}_{R(\tau)}$$

Let's apply the importance weighting trick!

$\nabla_\theta J(\theta)$

Derivation of Policy Gradient: REINFORCE

$$J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \underbrace{\left[\sum_{h=0}^{H-1} r(s_h, a_h) \right]}_{R(\tau)}$$

Let's say we are at θ_0, we want to compute gradient at θ_0, i.e., $\nabla_\theta J(\theta) \mid_{\theta=\theta_0}$

Derivation of Policy Gradient: REINFORCE

$$J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \underbrace{\left[\sum_{h=0}^{H-1} r(s_h, a_h) \right]}_{R(\tau)}$$

Let's say we are at θ_0, we want to compute gradient at θ_0, i.e., $\nabla_\theta J(\theta) \,|_{\theta=\theta_0}$

$$J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta} R(\tau) = \mathbb{E}_{\tau \sim \rho_{\theta_0}} \frac{\rho_\theta(\tau)}{\rho_{\theta_0}(\tau)} \cdot R(\tau)$$

ρ_{θ_0}

Derivation of Policy Gradient: REINFORCE

$$J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \underbrace{\left[\sum_{h=0}^{H-1} r(s_h, a_h) \right]}_{R(\tau)}$$

Let's say we are at θ_0, we want to compute gradient at θ_0, i.e., $\nabla_\theta J(\theta)\,|_{\theta=\theta_0}$

$$\nabla_\theta \quad J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta} R(\tau) \quad \nabla_\theta \quad = \mathbb{E}_{\tau \sim \rho_{\theta_0}} \left[\frac{\rho_\theta(\tau)}{\rho_{\theta_0}(\tau)} R(\tau) \right]$$

Derivation of Policy Gradient: REINFORCE

$$J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \underbrace{\left[\sum_{h=0}^{H-1} r(s_h, a_h) \right]}_{R(\tau)}$$

Let's say we are at θ_0, we want to compute gradient at θ_0, i.e., $\nabla_\theta J(\theta)\,|_{\theta=\theta_0}$

$$J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta} R(\tau) \quad = \mathbb{E}_{\tau \sim \rho_{\theta_0}} \left[\frac{\rho_\theta(\tau)}{\rho_{\theta_0}(\tau)} R(\tau) \right]$$

$$\nabla_\theta J(\theta)\,|_{\theta=\theta_0} = \mathbb{E}_{\tau \sim \rho_{\theta_0}} \left[\frac{\nabla_\theta \rho_\theta(\tau)\,|_{\theta=\theta_0}}{\rho_{\theta_0}(\tau)} R(\tau) \right]$$

Chain-Rule

Derivation of Policy Gradient: REINFORCE

$$J(\theta) = \mathbb{E}_{\tau\sim\rho_\theta(\tau)}\underbrace{\left[\sum_{h=0}^{H-1} r(s_h, a_h)\right]}_{R(\tau)}$$

Let's say we are at θ_0, we want to compute gradient at θ_0, i.e., $\nabla_\theta J(\theta)\,|_{\theta=\theta_0}$

$$J(\theta) = \mathbb{E}_{\tau\sim\rho_\theta} R(\tau) \quad = \mathbb{E}_{\tau\sim\rho_{\theta_0}}\left[\frac{\rho_\theta(\tau)}{\rho_{\theta_0}(\tau)} R(\tau)\right]$$

$$\nabla_\theta J(\theta)\,|_{\theta=\theta_0} = \mathbb{E}_{\tau\sim\rho_{\theta_0}}\left[\frac{\nabla_\theta \rho_\theta(\tau)\,|_{\theta=\theta_0}}{\rho_{\theta_0}(\tau)} R(\tau)\right] \quad = \mathbb{E}_{\tau\sim\rho_{\theta_0}}\left[\nabla_\theta \ln \rho_\theta(\tau)\,|_{\theta=\theta_0} R(\tau)\right]$$

Derivation of Policy Gradient: REINFORCE

$$\nabla_\theta J(\theta)\,|_{\theta=\theta_0} = \mathbb{E}_{\tau\sim\rho_{\theta_0}}\left[\nabla_\theta \ln \rho_\theta(\tau)\,|_{\theta=\theta_0} R(\tau)\right]$$

ρ_θ contains unknown transition, seems that we cannot compute $\nabla \ln \rho_\theta \ldots$

$$\rho_\theta(\tau) = \mu_0(s_0)\,\pi_\theta(a_0|s_0)\cdot P(s_1|s_0,a_0)\cdots$$

Derivation of Policy Gradient: REINFORCE

$$\nabla_\theta J(\theta)\,|_{\theta=\theta_0} = \mathbb{E}_{\tau\sim\rho_{\theta_0}}\left[\nabla_\theta \ln \rho_\theta(\tau)\,|_{\theta=\theta_0} R(\tau)\right]$$

ρ_θ contains unknown transition, seems that we cannot compute $\nabla \ln \rho_\theta \ldots$

Recall: $\rho_\theta(\tau) = \mu(s_0)\pi_\theta(a_0\,|\,s_0)P(s_1\,|\,s_0, a_0)\pi_\theta(a_1\,|\,s_1)\ldots$

Derivation of Policy Gradient: REINFORCE

$$\nabla_\theta J(\theta)\,|_{\theta=\theta_0} = \mathbb{E}_{\tau\sim\rho_{\theta_0}}\left[\nabla_\theta \ln \rho_\theta(\tau)\,|_{\theta=\theta_0} R(\tau)\right]$$

ρ_θ contains unknown transition, seems that we cannot compute $\nabla \ln \rho_\theta$…

Recall: $\rho_\theta(\tau) = \mu(s_0)\pi_\theta(a_0 \mid s_0)P(s_1 \mid s_0, a_0)\pi_\theta(a_1 \mid s_1)\dots$

In class excerise: can you plug in the definition of ρ_θ into the above expression, and see if you can simplify the gradient formulation further

$$\nabla_\theta \ln \rho_\theta(\tau) = \cancel{\nabla_\theta \ln \mu(s_0)} + \underline{\nabla_\theta \ln \pi_\theta(a_0|s_0)} + \cancel{\nabla_\theta \ln P(s_1|s_0,a_0)} \dots$$

Summary so far for Policy Gradients

We derived the most classic PG formulation:

$$\nabla J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \left[\left(\sum_{h=0}^{H} \underline{\nabla_\theta \ln \pi_\theta(a_h \mid s_h)} \right) R(\tau) \right]$$

Summary so far for Policy Gradients

We derived the most classic PG formulation:

$$\nabla J(\theta) = \underline{\mathbb{E}_{\tau \sim \rho_\theta(\tau)}} \left[\left(\sum_{h=0}^{H} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \right) R(\tau) \right]$$

Increase the likelihood of on trajectories with high total reward

$$\tau_1, \tau_2 \cdots \tau_n \overset{iid}{\sim} \rho_\theta(\tau)$$

$$\widehat{\nabla J(\theta)} = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_h \nabla_\theta \ln \pi_\theta(a_h^i \mid s_h^i) \, R(\tau_i) \right)$$

$$\theta \Leftarrow \theta + \eta \widehat{\nabla} J(\theta)$$

Further simplification on PG

$$\nabla J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \left[\left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \cdot \sum_{\tau=h}^{H-1} r(s_\tau, a_\tau) \right) \right]$$

Further simplification on PG

$$\nabla J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \left[\left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \cdot \sum_{\tau=h}^{H-1} r(s_\tau, a_\tau) \right) \right]$$

(Change action distribution at h only affects rewards later on…)

Further simplification on PG

$$\nabla J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \left[\left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \cdot \sum_{\tau=h}^{H-1} r(s_\tau, a_\tau) \right) \right]$$

(Change action distribution at h only affects rewards later on…)

Exercise:

Show this simplified version is equivalent to REINFORCE

Demo of REINFORCE on CartPole

REINFORCE works surprisingly well on generative models

We train LLMs to summarize reddit post

Example Post

SUBREDDIT: r/dogs
TITLE: [HELP] Not sure how to deal with new people/dogs and my big ole pup
POST: I have a three year old Dober/Pit mix named Romulus ("Rome" for short). I live with 3 other dogs: a 10 year old labrador, a 2 year old French Bulldog and a 8 year old maltese mix. The four of them get along just fine, Rome and the Frenchie are best best best best friends.   He isn't the best at meeting new people, but not ALWAYS....Then, the crux of the matter: I want to have a 4th of July party. Several people want to bring their dogs. I doubt I can say "no dogs allowed" and I don't want to let everyone else bring their dog and make mine stay at day care all day.
TL;DR:

Summary:

HOW do I introduce new people? HOW do I introduce new dogs? WHAT do I do about 4th of July??

REINFORCE works surprisingly well on generative models

Dataset Composition

- 210K Prompts total
  - 117K Prompts with *Human written summarizations*
  - 93K Prompts with *Human Preference Labels*

Post → Summary 1 ✓

Post → Summary 2 ✗

REINFORCE works surprisingly well on generative models

Dataset Composition

- 210K Prompts total
  - 117K Prompts with *Human written summarizations*
  - 93K Prompts with *Human Preference Labels*

Used to pre-train a reward model (details will come soon)

Using RL to optimize 7B size model on TL;DR

Model size	Algorithm	Winrate (↑)
6.9B	SFT	45.2 (±2.49)
	DPO	68.4 (±2.01)
	REINFORCE	70.7*
	PPO	77.6‡
	RLOO ($k = 2$)	74.2*
	RLOO ($k = 4$)	77.9*
	REBEL	**78.1** (±1.74)

* directly obtained from Ahmadian et al. (2024)

‡ directly obtained from Huang et al. (2024)

[REBEL: Reinforcement Learning via Regressing Relative Rewards, Neurips 2024]

Using RL to optimize 7B size model on TL;DR

Model size	Algorithm	Winrate (↑)
6.9B	SFT	45.2 (±2.49)
	DPO	68.4 (±2.01)
	REINFORCE	70.7*
	PPO	77.6‡
	RLOO ($k = 2$)	74.2*
	RLOO ($k = 4$)	77.9*
	REBEL	**78.1** (±1.74)

* directly obtained from Ahmadian et al. (2024)

‡ directly obtained from Huang et al. (2024)

Simple REINFORCE already achieves 70% win-rate over humans (evaluated by GPT4)

[REBEL: Reinforcement Learning via Regressing Relative Rewards, Neurips 2024]

Summary for today

1. Importance Weighting Trick

2. Policy Gradient:

REINFORCE (a direct application of our warm up example):

Summary for today

1. Importance Weighting Trick

2. Policy Gradient:

REINFORCE (a direct application of our warm up example):

$$\nabla J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \left[\left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h | s_h) \right) R(\tau) \right]$$

Summary for today

1. Importance Weighting Trick

2. Policy Gradient:

REINFORCE (a direct application of our warm up example):

$$\nabla J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \left[\left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h | s_h) \right) R(\tau) \right]$$

2. Use unbiased estimate of $\nabla_\theta J(\theta)$, SG ascent converges to local optimal policy