Tutorial on Reinforcement Learning

Policy Gradient Methods

Chunwei Xing

Table of Contents

Table of Contents

MDPs and Policies

- We consider a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$,
- $\mathcal{S}$ and $\mathcal{A}$ are state and action spaces, $P(s' \mid s, a)$ is the transition kernel, $r(s, a)$ is the expected immediate reward, and $\gamma \in (0, 1)$ is the discount factor.
- A trajectory $\tau = (s_0, a_0, s_1, a_1, \dots)$ is sampled from $p_\theta(\tau) = p(s_0) \prod_{t \geq 0} \pi_\theta(a_t \mid s_t) P(s_{t+1} \mid s_t, a_t)$.
- A stochastic policy $\pi_\theta(a \mid s)$ is parameterized by $\theta \in \mathbb{R}^d$.
- We assume bounded rewards $|r| \leq R_{\max}$ and differentiable policies with $\pi_\theta(a \mid s) > 0$ on support.

$$J(\pi_\theta) \triangleq \mathbb{E}_{\tau \sim p_\theta} \Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \Big] = \mathbb{E}_{\tau \sim p_\theta} \Big[R(\tau) \Big]. \tag{1}$$

Value Functions

$$V^{\pi}(s) = \mathbb{E}_{\pi}\Big[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s\Big], \tag{2}$$

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\Big[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s,\ a_0 = a\Big], \tag{3}$$

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s). \tag{4}$$

- The discounted performance objective in terms of the value functions is

$$J(\pi_\theta) \triangleq \mathbb{E}_{\tau\sim p_\theta}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t)\Big] = \mathbb{E}_{s\sim\mu(s)}\Big[V^{\pi}(s)\Big]. \tag{5}$$

- The policy gradient using the advantage function is

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim p_\theta}\Big[\sum_{t=0}^{\infty} \gamma^t A^{\pi}(s_t,a_t)\nabla_\theta \log \pi_\theta(a_t \mid s_t)\Big]. \tag{6}$$

Table of Contents

Univariate Gaussian Policy

Consider a scalar Gaussian policy

$$a \sim \pi_\theta(\cdot \mid s) = \mathcal{N}(\mu_\theta(s), \sigma^2), a \in \mathbb{R}, s \in \mathbb{R}^n$$

Problem

Derive the policy gradient for a Gaussian policy.

Solution:

$$\log \pi_\theta(a \mid s) = -\tfrac{1}{2}\left(\frac{(a-\mu)^2}{\sigma^2} + \log(2\pi\sigma^2)\right), \qquad \mu = \mu_\theta(s).$$

Gradients w.r.t. μ and σ:

$$g_\mu := \frac{\partial}{\partial \mu} \log \pi_\theta(a \mid s) = \frac{a-\mu}{\sigma^2}$$

$$g_\sigma := \frac{\partial}{\partial \sigma} \log \pi_\theta(a \mid s) = \frac{(a-\mu)^2 - \sigma^2}{\sigma^3}$$

Gaussian Policy Update Rules

By chain rule, gradients w.r.t. network parameters θ:

$$\nabla_\theta \log \pi_\theta(a \mid s) = g_\mu \nabla_\theta \mu_\theta(s)$$

Update policy parameters by gradient ascent:

$$\begin{aligned}
\theta_{k+1} &= \theta_k + \alpha \nabla_\theta \log \pi_\theta(a \mid s) A^\pi(s,a) && (7)\\
&= \theta_k + \alpha \left[\frac{a-\mu}{\sigma^2} A^\pi(s,a)\right] \nabla_\theta \mu_\theta(s) && (8)\\
&= \theta_k + \alpha\, g_\mu\, A^\pi(s,a)\, \nabla_\theta \mu_\theta(s), && (9)\\
\sigma_{k+1} &= \sigma_k + \beta \left[\frac{(a-\mu)^2}{\sigma^3} - \frac{1}{\sigma}\right] A^\pi(s,a) && (10)\\
&= \sigma_k + \beta\, g_\sigma\, A^\pi(s,a), && (11)
\end{aligned}$$

Comments on the Policy Gradient

- **Score function**: $\nabla_\theta \log \pi_\theta(a|s) = \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}$
- **Zero expectation**: $\mathbb{E}_{a\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s)] = 0$
- **Exploration control**: When advantage is large, σ increases for more exploration
- **Convergence**: When the advantage is positive, the gradient is large when the action is far from the mean, and the gradient is close to zero if the action is close to the mean.

Problem: Gaussian Policy Gradient

Problem

Derive the policy gradient when the policy is parameterized by the log-standard-deviation $\omega \triangleq \log \sigma$, and discuss the stability of the learning.

Problem

For $r(a) = -(a - a^*)^2$, show gradient ascent on μ converges to a^* when σ is fixed.

Additional Problems

Penn
University of Pennsylvania

Problem

Derive the policy gradient when the state is not fully observed. What practical changes to make in the algorithm?

Problem

Derive the policy gradient when the action is delayed by N time step. What practical changes to make in the algorithm?

Discrete Policy Parametrizations

Direct parametrization:

$$\pi_\theta(a \mid s) = \theta_{s,a}$$

where $\theta_{s,a} \geq 0$ and $\sum_{a \in \mathcal{A}} \theta_{s,a} = 1$.

Softmax policy:

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

Log-linear policy:

$$\pi_\theta(a \mid s) = \frac{\exp(\theta \cdot \phi(s,a))}{\sum_{a' \in \mathcal{A}} \exp(\theta \cdot \phi(s,a'))}$$

Neural softmax policy:

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

Score Function for Discrete Policy

Score function for direct parametrization:

$$\nabla_{\theta_{s',a'}} \log \pi_\theta(a \mid s) = \frac{1}{\theta_{s,a}} \cdot \mathbf{1}_{a'=a,s'=s}$$

where $\mathbf{1}_{a'=a,s'=s}$ is the indicator function for action a' and state s'.

Problem

Derive the gradient for a discrete action policy, i.e., softmax policy, log-linear policy, and neural softmax policy.

Policy Gradient Expressions

The policy gradient can be expressed in three equivalent forms:

- **REINFORCE expression:**

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim p_\theta} \left[R(\tau) \left(\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t|s_t) \right) \right]$$

- **Action value expression:**

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim p_\theta} \left[\sum_{t=0}^{\infty} \gamma^t Q^{\pi_\theta}(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t|s_t) \right]$$

- **Baseline expression:**

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim p_\theta} \left[\sum_{t=0}^{\infty} \gamma^t [Q^{\pi_\theta}(s_t, a_t) - b(s_t)] \nabla_\theta \log \pi_\theta(a_t|s_t) \right]$$

Baseline Invariance Theorem

Theorem (Baseline Invariance)

For any function $b : \mathcal{S} \to \mathbb{R}$, the estimator $\mathbb{E}_{\tau \sim p_\theta}\left[\sum_{t=0}^{\infty} \gamma^t (G_t - b(s_t)) \nabla \log \pi(a \mid s)\right]$ *is unbiased.*

Proof.

It's sufficient to show that $\mathbb{E}_{a \sim \pi(\cdot|s)}[\nabla \log \pi(a \mid s)\, b(s)] = 0$.

$$\begin{aligned}
\mathbb{E}_{a \sim \pi(\cdot|s)}[\nabla \log \pi(a \mid s)\, b(s)] &= b(s) \cdot \mathbb{E}_{a \sim \pi(\cdot|s)}[\nabla \log \pi(a \mid s)] \\
&= b(s) \cdot \sum_a \pi(a \mid s) \nabla \log \pi(a \mid s) \\
&= b(s) \cdot \sum_a \pi(a \mid s) \frac{\nabla \pi(a \mid s)}{\pi(a \mid s)} \\
&= b(s) \cdot \nabla \sum_a \pi(a \mid s) = b(s) \cdot \nabla 1 = 0.
\end{aligned}$$

Problems

Problem

Since actions at time t cannot affect past rewards, thus one may replace $R(\tau)$ in the REINFORCE expression by the return-to-go

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t, \quad G_t \triangleq \sum_{k=t}^{\infty} \gamma^{k-t} r_k, \tag{12}$$

$$\Rightarrow \quad \nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim p_\theta} \Big[\sum_{t \geq 0} \gamma^t G_t \nabla \log \pi_\theta(a_t \mid s_t) \Big]. \tag{13}$$

Problem

Derive the action-value expression of the policy gradient from:

$$J(\pi_\theta) = \mathbb{E}_{s_0 \sim \mu(s_0)} \left[V^\pi(s_0) \right], \quad \mu(s_0) \triangleq \textit{initial state distribution}$$

Table of Contents

Monte Carlo Estimation

Consider the expectation

$$F(\theta) = \mathbb{E}_{\xi \sim p(\xi)}[f(\theta, \xi)]$$

The gradient is

$$\nabla_\theta F(\theta) = \nabla_\theta \int f(\theta, \xi) p(\xi) d\xi \tag{14}$$

$$= \int \nabla_\theta f(\theta, \xi) p(\xi) d\xi \tag{15}$$

$$= \mathbb{E}_{\xi \sim p(\xi)}[\nabla_\theta f(\theta, \xi)] \tag{16}$$

Unbiased gradient estimators:

$$\hat{\nabla}_\theta F(\theta) = \nabla_\theta f(\theta, \xi), \quad \text{where } \xi \sim p(\xi) \tag{17}$$

$$\hat{\nabla}_\theta F(\theta) = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta f(\theta, \xi_i), \quad \text{where } \xi_1, \ldots, \xi_n \sim p(\xi) \tag{18}$$

REINFORCE: Monte Carlo Policy Gradient Method

The policy gradient theorem states:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim p_\theta} \left[\sum_{t=0}^{\infty} \gamma^t G_t \nabla_\theta \log \pi_\theta(a_t|s_t) \right]$$

In practice, REINFORCE approximates this expectation with Monte Carlo samples:

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

where each trajectory $(s_0^{(i)}, a_0^{(i)}, r_0^{(i)}, \ldots)$ is a rollout from the environment.
It estimates the policy gradient using returns from **a full trajectory**, without bootstrapping or temporal-difference updates.

Algorithm: REINFORCE

Algorithm 1 REINFORCE: Monte Carlo Policy Gradient Method

Initialize θ, baseline b (e.g., running average or V-critic)
for episodes **do**
    Sample trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$ under π_θ
    for $t = 0, \ldots, T-1$ **do**
        Compute return-to-go $G_t \leftarrow \sum_{k=t}^{\infty} \gamma^{k-t} r_k$
        Compute advantage $A_t \leftarrow G_t - b(s_t)$
        Compute policy gradient $g_t \leftarrow \gamma^t A_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)$
        Update $\theta \leftarrow \theta + \alpha\, g_t$
    end for
    Update baseline b
end for

Demo

Table of Contents

References

- R. S. Sutton and A. G. Barto (2018). *Reinforcement Learning: An Introduction*, 2nd ed.
- R. J. Williams (1992). *Simple statistical gradient-following algorithms for connectionist RL (REINFORCE)*.
- N. He (2024). *Foundations of Reinforcement Learning*. ETH course.
- L. Weng (2018). *A (Long) Peek into Reinforcement Learning*. Blog post.

Thank you!