Policy Gradient: Theory

Thuong Dang

November 2025

Contents

1 Overview

In the previous project we showed that Q-learning can be interpreted as a fixed-point search algorithm. However, in the gridworld example, this approach has a clear limitation: if we change the grid configuration, then the environment changes and the Bellman fixed point changes as well. In order to train and act effectively on many different grids, we need a more general method.

One such method is *policy gradient* [1, 2]. Policy gradient methods form one of the core foundations of modern reinforcement learning (e.g. actor–critic methods, PPO, and many others).

The goal is conceptually simple. We wish to learn a policy that selects an action a given the current state s, written

$$a \sim \pi(a \mid s).$$

If π_θ is represented by a neural network with parameters θ, then at evaluation time we select the greedy action

$$a^* = \arg\max_a \pi_\theta(a \mid s).$$

To train the neural network, we need an *objective*. Unlike supervised learning, reinforcement learning has no ground-truth labels. Instead, the agent must maximize long-term reward. In the discounted setting, this is

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

where the notation $\mathbb{E}_{\tau\sim\pi_\theta}$ means *expectation over trajectories generated by following policy* π_θ. Formally, it is defined to be

$$\mathbb{E}_{\tau\sim\pi_\theta}[f(\tau)] = \sum_{\text{all possible } \tau} P(\tau \mid \theta) f(\tau)$$

Trajectories and the probability $P(\tau \mid \theta)$ will be discussed in Section 3. And our task is to compute $\nabla_\theta J(\theta)$ so that we can update the policy parameters by gradient ascent. In Section 4, we will prove the policy gradient theorem for discounted reward setting and derive REINFORCE. In Section 5 we will discuss the average reward setting for the policy gradient theorem.

2 Markov assumptions

To make the derivation precise, we list the assumptions required by the MDP model and by the policy parameterization.

2.1 Environment assumptions (MDP)

The environment is a Markov decision process. In particular:

1. The initial state s_0 is drawn from a distribution that depends only on the environment, not on the policy:

$$\nabla_\theta P(s_0) = 0.$$

2. The transition dynamics depend only on the current state and action and are independent of the policy parameters:

$$P(s_{t+1} \mid s_0, a_0, \ldots, s_t, a_t) = P(s_{t+1} \mid s_t, a_t), \qquad \nabla_\theta P(s_{t+1} \mid s_t, a_t) = 0.$$

2.2 Policy assumptions (automatic from the neural network)

Because our policy is represented by a neural network $\pi_\theta(a \mid s)$, it automatically satisfies:

1. **Markov policy:** the action distribution depends only on the current state:
$$P(a_t \mid s_0, a_0, \ldots, s_t) = \pi_\theta(a_t \mid s_t).$$

2. **Time-homogeneous (stationary) policy:** the policy does not change with time:
$$\pi_\theta(a_t = a \mid s_t = s) = \pi_\theta(a \mid s) \qquad \text{for all } t.$$

Together, these assumptions correspond exactly to the standard MDP framework and the usual policy parameterizations used in deep reinforcement learning.

3 Trajectories and probabilities

A trajectory is a complete history of interaction between an agent and an environment
$$\tau = (s_0, a_0, s_1, a_1, ...),$$
where s_t is the state at time t, a_t is the action taken at time t.

Everytime we execute a policy, we obtain different trajectory, because

- The initial state s_0 may be random
- The policy π_θ may be stochastic $a_t \sim \pi_\theta(. \mid s_t)$
- The environment transition are stochastic $s_{t+1} \sim P(. \mid s_t, a_t)$

Therefore, a trajectory τ is a *random variable*. Using the chain rule of probability, we have

$$P(\tau \mid \theta) = P(s_0, a_0, s_1, a_1, ...) =$$

$$= P(s_0)P(a_0 \mid s_0)P(s_1 \mid s_0, a_0)P(a_1 \mid s_0, a_0, s_1)...$$

By using Markov's assumptions from Section 2, we have

$$P(a_t \mid s_0, a_0, ..., s_t) = \pi_\theta(a_t \mid s_t),$$

and

$$P(s_{t+1} \mid s_0, a_0, ..., s_t, a_t) = P(s_{t+1} \mid s_t, a_t),$$

we obtain

$$P(\tau \mid \theta) = P(s_0) \prod_{t=0}^{\infty} \pi_\theta(a_t \mid s_t) P(s_{t+1} \mid s_t, a_t)$$

4 Policy Gradient Theorem: Discounted Return Setting

We consider a Markov decision process (MDP) with transition dynamics $P(s_{t+1} \mid s_t, a_t)$ that are independent of the policy parameters θ. The policy $\pi_\theta(a \mid s)$ is assumed to be a stationary (time-homogeneous) Markov policy: it depends only on the current state and is independent of t.

4.1 Discounted performance objective

Fix a discount factor $\gamma \in [0, 1)$. The discounted performance objective is defined as

$$J_\gamma(\theta) := \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right].$$

Under bounded rewards $\mid r(s, a) \mid \leq R_{\max}$, the discounted sum is absolutely convergent:

$$\mid \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid \ \leq \frac{R_{\max}}{1 - \gamma}.$$

Let $G_t(\tau) = \sum_{k=t}^{\infty} \gamma^{k-t} r(s_k, a_k)$ denote the discounted reward-to-go at time t.

4.2 Trajectory form of the gradient

Writing $P(\tau \mid \theta)$ for the trajectory distribution under π_θ, we have

$$J_\gamma(\theta) = \sum_\tau P(\tau \mid \theta)\, G_0(\tau), \qquad \nabla_\theta J_\gamma(\theta) = \sum_\tau \nabla_\theta P(\tau \mid \theta)\, G_0(\tau).$$

Using the log-derivative trick,

$$\nabla_\theta P(\tau \mid \theta) = P(\tau \mid \theta)\, \nabla_\theta \log P(\tau \mid \theta),$$

and factorizing the trajectory distribution (Section 3),

$$P(\tau \mid \theta) = P(s_0) \prod_{t=0}^{\infty} \pi_\theta(a_t \mid s_t)\, P(s_{t+1} \mid s_t, a_t),$$

we obtain

$$\nabla_\theta \log P(\tau \mid \theta) = \sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t \mid s_t),$$

because $P(s_0)$ and $P(s_{t+1} \mid s_t, a_t)$ do not depend on θ (Section 2). Thus,

$$\nabla_\theta J_\gamma(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, G_0(\tau)\right].$$

Lemma 1 (Causality)**.** *For any $k < t$, past rewards do not contribute to the policy gradient:*

$$\mathbb{E}_{\tau\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \cdot r(s_k, a_k)\right] = 0.$$

Proof. The key is that $\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\nabla_\theta \log \pi_\theta(a|s)] = 0$ for any state s. To see this:

$$\begin{aligned}
\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\nabla_\theta \log \pi_\theta(a|s)] &= \sum_a \pi_\theta(a|s)\nabla_\theta \log \pi_\theta(a|s) \\
&= \sum_a \nabla_\theta \pi_\theta(a|s) \\
&= \nabla_\theta \sum_a \pi_\theta(a|s) \\
&= \nabla_\theta 1 = 0.
\end{aligned}$$

Since $r(s_k, a_k)$ with $k < t$ is determined before a_t is sampled, we can condition on the history and factor out the past reward, leaving only the zero expectation above. □

By the causality lemma, we may replace $G_0(\tau)$ with $G_t(\tau)$:

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

This is the REINFORCE estimator [1].

4.3 State–action form

Define the discounted action-value function

$$Q_\gamma^\pi(s,a) = \mathbb{E}_\pi[G_t \mid s_t = s,\, a_t = a].$$

Then,

$$\nabla_\theta J_\gamma(\theta) = \sum_{t=0}^{\infty} \sum_s P_\pi(s_t = s) \sum_a \pi_\theta(a \mid s)\, \nabla_\theta \log \pi_\theta(a \mid s)\, Q_\gamma^\pi(s,a).$$

Using $\pi_\theta(a \mid s)\nabla_\theta \log \pi_\theta(a \mid s) = \nabla_\theta \pi_\theta(a \mid s)$, we obtain

$$\nabla_\theta J_\gamma(\theta) = \sum_{t=0}^{\infty} \sum_s \sum_a \gamma^t P_\pi(s_t = s)\, \nabla_\theta \pi_\theta(a \mid s)\, Q_\gamma^\pi(s, a).$$

It is convenient to introduce the *discounted state-visitation distribution*

$$d_\gamma^\pi(s) := (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t P_\pi(s_t = s),$$

which is well-defined without any assumption. We then obtain the *discounted policy gradient theorem*

$$\nabla_\theta J_\gamma(\theta) = \frac{1}{1 - \gamma} \sum_s d_\gamma^\pi(s) \sum_a \nabla_\theta \pi_\theta(a \mid s)\, Q_\gamma^\pi(s, a).$$

4.4 Monte Carlo approximation of the action-value function.

In the discounted policy gradient theorem, the exact gradient involves the action-value function

$$Q_\gamma^\pi(s, a) = \mathbb{E}_\pi[G_t \mid s_t = s,\, a_t = a], \qquad G_t = \sum_{k=t}^{\infty} \gamma^{k-t} r(s_k, a_k).$$

In practice, the expectation defining Q_γ^π is not computed analytically. Instead, we approximate $Q_\gamma^\pi(s_t, a_t)$ by a *Monte Carlo sample*:

$$Q_\gamma^\pi(s_t, a_t) \approx G_t(\tau) := \sum_{k=t}^{T-1} \gamma^{k-t} r(s_k, a_k),$$

where T is the terminal time of the episode. This estimator is *unbiased*:

$$\mathbb{E}_{\tau \sim \pi_\theta}[G_t(\tau) \mid s_t, a_t] = Q_\gamma^\pi(s_t, a_t).$$

Substituting G_t into the trajectory form of the gradient yields the REINFORCE estimator

$$\nabla_\theta J_\gamma(\theta) \approx \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, G_t(\tau),$$

which is the *update formula* that can be implemented for training. In this sense, Monte Carlo return estimates play the role of sampled action-values in the policy gradient.

5 Policy Gradient Theorem: Average Reward Setting

The average-reward policy gradient theorem was developed by Sutton et al. [2]. The proof for this case is very similar to the discounted reward setting, but there is one fundamental difference: in order to derive the policy gradient theorem, we need an assumption that the Markov chain generated by the policy is *ergodic*, so that the *stationary distribution* exists. In this set-up, The average-reward objective is

$$J(\theta) := \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T-1} r(s_t, a_t)\right].$$

For each finite T, define the finite-horizon average return on a trajectory τ by

$$R_T(\tau) := \frac{1}{T}\sum_{t=0}^{T-1} r(s_t, a_t).$$

Then

$$J(\theta) = \lim_{T\to\infty} \mathbb{E}_{\tau\sim\pi_\theta}[R_T(\tau)] = \lim_{T\to\infty} \sum_{\tau} P_T(\tau \mid \theta) R_T(\tau),$$

where $P_T(\tau \mid \theta)$ is the probability of observing a length-T trajectory under policy π_θ.

Assuming we can interchange the limit and the gradient (by dominated convergence), we obtain

$$\nabla_\theta J(\theta) = \lim_{T\to\infty} \sum_{\tau} \nabla_\theta P_T(\tau \mid \theta) R_T(\tau).$$

Using the log-derivative trick,

$$\nabla_\theta P_T(\tau \mid \theta) = P_T(\tau \mid \theta)\, \nabla_\theta \log P_T(\tau \mid \theta).$$

For a trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots, s_{T-1}, a_{T-1})$,

$$P_T(\tau \mid \theta) = P(s_0) \prod_{t=0}^{T-1} \pi_\theta(a_t \mid s_t)\, P(s_{t+1} \mid s_t, a_t),$$

so

$$\log P_T(\tau \mid \theta) = \log P(s_0) + \sum_{t=0}^{T-1} \log \pi_\theta(a_t \mid s_t) + \sum_{t=0}^{T-1} \log P(s_{t+1} \mid s_t, a_t),$$

and since $P(s_0)$ and $P(s_{t+1} \mid s_t, a_t)$ do not depend on θ, we get

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

Thus,

$$\nabla_\theta J(\theta) = \lim_{T\to\infty} \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, R_T(\tau)\right].$$

Using the causality argument, we can replace $R_T(\tau)$ by the future return $G_t := \sum_{k=t}^{T-1} r(s_k, a_k)$ inside the expectation, obtaining

$$\nabla_\theta J(\theta) = \lim_{T\to\infty} \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, G_t\right].$$

Defining the action-value function

$$Q^\pi(s,a) := \mathbb{E}_\pi[G_t \mid s_t = s, a_t = a],$$

we get

$$\nabla_\theta J(\theta) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_s P_\pi(s_t = s) \sum_a \pi_\theta(a \mid s)\, \nabla_\theta \log \pi_\theta(a \mid s)\, Q^\pi(s,a).$$

Using $\pi_\theta(a \mid s)\, \nabla_\theta \log \pi_\theta(a \mid s) = \nabla_\theta \pi_\theta(a \mid s)$, we obtain

$$\nabla_\theta J(\theta) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_s \sum_a P_\pi(s_t = s)\, \nabla_\theta \pi_\theta(a \mid s)\, Q^\pi(s,a).$$

Assuming the Markov chain induced by π_θ is ergodic, the time-average state distribution converges:

$$d^\pi(s) := \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} P_\pi(s_t = s).$$

Exchanging the limit and the finite sums, we obtain the average-reward policy gradient:

$$\nabla_\theta J(\theta) = \sum_s d^\pi(s) \sum_a \nabla_\theta \pi_\theta(a \mid s)\, Q^\pi(s,a).$$

6 Why discounted-reward policy gradients are preferred

The average-reward objective is defined as

$$J_{\text{avg}}(\theta) = \lim_{T\to\infty} \frac{1}{T} \mathbb{E}_\pi\left[\sum_{t=0}^{T-1} r(s_t, a_t)\right].$$

To express this in state–action form, one requires the time-average state distribution

$$d^\pi(s) := \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} P_\pi(s_t = s),$$

which exists only when the Markov chain induced by the policy π_θ is *ergodic* (irreducible, aperiodic). More precisely, a Markov chain is *irreducible* if it is possible to get from any state to any other state. Moreover, it is *aperiodic* if it does not have a fixed cycle for returning to a state.

The irreducibility condition is often violated in practical environments. For example, in games such as Tic-Tac-Toe, Go and chess, we cannot get from the terminal state, when the game ends, to any other state. In such settings, it is not guaranteed that the long-run time average $\frac{1}{T}\sum_{t=0}^{T-1} P_\pi(s_t = s)$ converges, so $d^\pi(s)$ may not exist. The average-reward policy gradient theorem therefore requires assumptions that rarely hold in neural-network-based deep RL.

By contrast, the *discounted* objective

$$J_\gamma(\theta) = \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\Big], \qquad \gamma < 1,$$

always exists under bounded rewards and does not require any ergodicity assumption. The discounted state-visitation distribution

$$d^\pi_\gamma(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t P_\pi(s_t = s)$$

is always well-defined and automatically normalized. This makes the discounted policy gradient theorem robust and suitable for deep RL in practice.

References

[1] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256.

[2] Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12 (NIPS 1999)*, pages 1057–1063.