

Report RL Tutorial 3

Antoine Gorceix

November 2025

1 Deriving the Policy Gradient Theorem

1.1 Exercise 1

Here we study the discounted state (ergodic occupancy) distribution

$$d^\pi(s) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t p_{\theta,t}(s), \quad (1)$$

where $p_{\theta,t}(s)$ is the marginal distribution of the state after t steps under policy π_θ .

(i) Let $p(s_{i+1} | a_i, s_i)$ denote the environment transition and $\pi_\theta(a_i | s_i)$ the policy. The joint over (s_{i+1}, a_i) given s_i factorizes as

$$p_\theta(s_{i+1}, a_i | s_i) = \pi_\theta(a_i | s_i) p(s_{i+1} | s_i, a_i). \quad (2)$$

Marginalizing actions yields the *induced state kernel* (a Markov kernel on states):

$$p_\theta(s_{i+1} | s_i) = \int \pi_\theta(a_i | s_i) p(s_{i+1} | s_i, a_i) da_i \quad (\text{sum over } a_i \text{ in the discrete case}). \quad (3)$$

(ii) Let $p_0(s_0)$ be the initial state distribution and let $p_\theta(s' | s)$ denote the induced state kernel in (3). The joint over the length t state sequence factorizes as

$$p_{\theta,t}(s_t, s_{t-1}, \dots, s_0) = p_0(s_0) \prod_{i=0}^{t-1} p_\theta(s_{i+1} | s_i). \quad (4)$$

Marginalizing over (s_{t-1}, \dots, s_0) gives

$$p_{\theta,t}(s) = \int \cdots \int p_0(s_0) \prod_{i=0}^{t-1} p_\theta(s_{i+1} | s_i), ds_{t-1} \cdots ds_0 \quad \text{with } s_t = s \quad \text{use sums in the discrete case.} \quad (5)$$

Equivalently the forward recursion reads

$$p_{\theta,0}(s) = p_0(s), \quad p_{\theta,t+1}(s') = \int p_\theta(s' | s), p_{\theta,t}(s), ds \quad \text{use sums in the discrete case.} \quad (6)$$

(iii) Discounted occupancy and a probabilistic view: For a finite horizon N , define the (unnormalized) discounted occupancy

$$p_{N,\theta}(s) = \sum_{t=0}^{N-1} \gamma^t p_{\theta,t}(s). \quad (7)$$

This is *not* a probability measure since its total mass is

$$\int p_{N,\theta}(s) ds = \sum_{t=0}^{N-1} \gamma^t = \frac{1-\gamma^N}{1-\gamma} \neq 1. \quad (8)$$

A convenient probabilistic interpretation arises by treating the time index t as a random variable with a (truncated) geometric prior

$$p_N(t) = \frac{\gamma^t}{\sum_{k=0}^{N-1} \gamma^k}, \quad t = 0, \dots, N-1, \quad (9)$$

and taking the joint $p_{N,\theta}(s, t) = p_\theta(s | t) p_N(t)$ with $p_\theta(s | t) = p_{\theta,t}(s)$. Marginalizing out t gives the normalized mixture

$$\tilde{p}_{N,\theta}(s) = \sum_{t=0}^{N-1} p_\theta(s | t) p_N(t) = \frac{1}{\sum_{k=0}^{N-1} \gamma^k} \sum_{t=0}^{N-1} \gamma^t p_{\theta,t}(s). \quad (10)$$

Letting $N \rightarrow \infty$ yields $\sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}$ and therefore

$$\lim_{N \rightarrow \infty} \tilde{p}_{N,\theta}(s) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t p_{\theta,t}(s) \equiv d^\pi(s), \quad (11)$$

which is a bona fide probability distribution. Intuitively, d^π is the distribution of the state when we (i) roll out policy π_θ from p_0 and (ii) sample a random time $T \sim \text{Geom}(1-\gamma)$ on $\{0, 1, 2, \dots\}$, then look at S_T .

1.2 Exercise 2

Throughout, let $\gamma \in (0, 1)$, $p_0(s)$ be the initial-state distribution, $\pi_\theta(a | s)$ a differentiable policy, and

$$J(\theta) = \mathbb{E}_{p(\tau)} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right] = \int p_0(s_0) \int \pi_\theta(a_0 | s_0) Q^\pi(s_0, a_0) da_0 ds_0, \quad (12)$$

where $Q^\pi(s, a)$ is the action-value under π . We assume standard regularity so we may interchange ∇_θ with integrals/sums and that rewards do not depend on θ .

(i)

$$\begin{aligned} \nabla_\theta J(\theta) &= \nabla_\theta \int p_0(s_0) \int \pi_\theta(a_0 | s_0) Q^\pi(s_0, a_0) da_0 ds_0 \\ &= \int p_0(s_0) \int \nabla_\theta [\pi_\theta(a_0 | s_0) Q^\pi(s_0, a_0)] da_0 ds_0 \\ &= \int p_0(s_0) \int \left[\underbrace{\nabla_\theta \pi_\theta(a_0 | s_0) Q^\pi(s_0, a_0)}_{\text{policy term}} + \underbrace{\pi_\theta(a_0 | s_0) \nabla_\theta Q^\pi(s_0, a_0)}_{Q\text{-term}} \right] da_0 ds_0. \end{aligned} \quad (13)$$

(ii) Let the Bellman equation be

$$Q^\pi(s_i, a_i) = r(s_i, a_i) + \gamma \int p(s_{i+1} | s_i, a_i) \int \pi_\theta(a_{i+1} | s_{i+1}) Q^\pi(s_{i+1}, a_{i+1}) da_{i+1} ds_{i+1}. \quad (14)$$

Assuming r and p do not depend on θ , differentiate w.r.t. θ and use the product rule inside the inner integral:

$$\begin{aligned} \nabla_\theta Q^\pi(s_i, a_i) &= \gamma \int p(s_{i+1} | s_i, a_i) \int \left[\nabla_\theta \pi_\theta(a_{i+1} | s_{i+1}) Q^\pi(s_{i+1}, a_{i+1}) \right. \\ &\quad \left. + \pi_\theta(a_{i+1} | s_{i+1}) \nabla_\theta Q^\pi(s_{i+1}, a_{i+1}) \right] da_{i+1} ds_{i+1}. \end{aligned} \quad (15)$$

Now integrate both sides against the start-state distribution and the policy at $t = 0$:

$$I_0 \triangleq \int p_0(s_0) \int \pi_\theta(a_0 | s_0) \nabla_\theta Q^\pi(s_0, a_0) da_0 ds_0. \quad (16)$$

Plug (15) into I_0 and swap integrals (Tonelli/Fubini). Using

$$p_{\theta,1}(s_1) = \int p_0(s_0) \int \pi_\theta(a_0 | s_0) p(s_1 | s_0, a_0) da_0 ds_0, \quad (17)$$

we obtain

$$I_0 = \gamma \int p_{\theta,1}(s_1) \int \nabla_\theta \pi_\theta(a_1 | s_1) Q^\pi(s_1, a_1) da_1 ds_1 + \gamma I_1, \quad (18)$$

where

$$I_t \triangleq \int p_{\theta,t}(s) \int \pi_\theta(a | s) \nabla_\theta Q^\pi(s, a) da ds. \quad (19)$$

Applying the same substitution to I_1, I_2, \dots yields the recursion

$$I_t = \gamma \int p_{\theta,t+1}(s) \int \nabla_\theta \pi_\theta(a | s) Q^\pi(s, a) da ds + \gamma I_{t+1}. \quad (20)$$

Unrolling for T steps gives

$$I_0 = \sum_{t=1}^T \gamma^t \int p_{\theta,t}(s) \int \nabla_\theta \pi_\theta(a | s) Q^\pi(s, a) da ds + \gamma^{T+1} I_{T+1}. \quad (21)$$

Since $\gamma < 1$ and I_{T+1} is bounded (e.g. $|Q^\pi| \leq R_{\max}/(1 - \gamma)$), the last term vanishes as $T \rightarrow \infty$. Therefore,

$$\int p_0(s_0) \int \pi_\theta(a_0 | s_0) \nabla_\theta Q^\pi(s_0, a_0) da_0 ds_0 = \sum_{t=1}^{\infty} \gamma^t \int p_{\theta,t}(s) \int \nabla_\theta \pi_\theta(a | s) Q^\pi(s, a) da ds. \quad (22)$$

(iii) Combine (i) and (ii) to obtain the policy-gradient theorem: From (13),

$$\nabla_\theta J(\theta) = \int p_0(s_0) \int \nabla_\theta \pi_\theta(a_0 | s_0) Q^\pi(s_0, a_0) da_0 ds_0 + \int p_0(s_0) \int \pi_\theta(a_0 | s_0) \nabla_\theta Q^\pi(s_0, a_0) da_0 ds_0. \quad (23)$$

Replacing the second term in (23) using (22) (and noting $p_{\theta,0} = p_0$) gives

$$\begin{aligned} \nabla_\theta J(\theta) &= \int p_{\theta,0}(s) \int \nabla_\theta \pi_\theta(a | s) Q^\pi(s, a) da ds + \sum_{t=1}^{\infty} \gamma^t \int p_{\theta,t}(s) \int \nabla_\theta \pi_\theta(a | s) Q^\pi(s, a) da ds \\ &= \sum_{t=0}^{\infty} \gamma^t \int p_{\theta,t}(s) \int \nabla_\theta \pi_\theta(a | s) Q^\pi(s, a) da ds. \end{aligned} \quad (24)$$

2 Implementation and Actor Critic

2.1 Exercise 5

Throughout, let $d^\pi(s)$ denote the (discounted) state visitation distribution under π_θ , and assume standard regularity so we may swap ∇_θ with integrals. Recall the policy-gradient form (cf. Eq. (24) in your notes):

$$\nabla_\theta J(\theta) = \int d^\pi(s) \int \nabla_\theta \pi_\theta(a | s) Q^\pi(s, a) da ds. \quad (25)$$

(i) Baselines do not change the gradient: Let $b : \mathcal{S} \rightarrow \mathbb{R}$ be any bounded function that does not depend on the action a (it may depend on s and even on θ). Consider

$$\begin{aligned} \int d\pi(s) \int \nabla_\theta \pi_\theta(a | s) (Q^\pi(s, a) - b(s)) da ds &= \int d\pi(s) \int \nabla_\theta \pi_\theta(a | s) Q^\pi(s, a) da ds \\ &\quad - \int d\pi(s) b(s) \int \nabla_\theta \pi_\theta(a | s) da ds. \end{aligned} \quad (26)$$

The second term in (26) is zero by normalization of the policy:

$$\begin{aligned} \int d\pi(s) b(s) \int \nabla_\theta \pi_\theta(a | s) da ds &= \int d\pi(s) b(s) \nabla_\theta \left[\int \pi_\theta(a | s) da \right] ds \\ &= \int d\pi(s) b(s) \nabla_\theta 1 ds = 0. \end{aligned} \quad (27)$$

Plugging (27) into (26) yields

$$\int d\pi(s) \int \nabla_\theta \pi_\theta(a | s) Q^\pi(s, a) da ds = \int d\pi(s) \int \nabla_\theta \pi_\theta(a | s) (Q^\pi(s, a) - b(s)) da ds, \quad (28)$$

which proves that any bounded, action-independent baseline $b(s)$ can be subtracted without changing the policy gradient.

Remark. Equivalently, using $\nabla_\theta \pi_\theta = \pi_\theta \nabla_\theta \log \pi_\theta$,

$$\mathbb{E}_{s \sim d^\pi, a \sim \pi_\theta(\cdot | s)} [\nabla_\theta \log \pi_\theta(a | s) b(s)] = \mathbb{E}_{s \sim d^\pi} \left[b(s) \nabla_\theta \int \pi_\theta(a | s) da \right] = 0,$$

giving the same result.

(ii) Actor–critic methods improve variance and sample efficiency

Variance reduction:

Because $\mathbb{E}_{a \sim \pi_\theta(\cdot | s)} [A^\pi(s, a)] = 0$ for every s , the factor multiplying $\nabla_\theta \log \pi_\theta(a | s)$ is centered, which reduces the conditional variance of the estimator without changing its mean. In fact, for estimators of the form $g(s, a) = \nabla_\theta \log \pi_\theta(a | s) (Q^\pi(s, a) - b(s))$, the per-state variance is minimized by a baseline close to $b^*(s) = V^\pi(s)$. Thus using $b(s) = V^\pi(s)$ (or an approximation of it) yields an almost minimum-variance gradient.

In actor–critic, we do not have V^π exactly and instead use a learned critic $V_\phi \approx V^\pi$ together with a one-step bootstrapped advantage

$$\hat{A}_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t) \quad (\text{TD error } \delta_t), \quad (29)$$

so the actor update is

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a_t | s_t) \hat{A}_t. \quad (30)$$

The TD target in (29) replaces a long, high-variance Monte Carlo return by a low-variance bootstrap; consequently the policy-gradient estimator has substantially lower variance.

Sample efficiency: Actor–critic improves sample efficiency for several reasons

1. **Bootstrapping from the critic.** The critic is updated from *single transitions* (s_t, a_t, r_t, s_{t+1}) , e.g.

$$\phi \leftarrow \phi - \beta \nabla_\phi (r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t))^2, \quad (31)$$

so learning proceeds at every step instead of waiting for full-trajectory returns. Although the target uses one successor state, $V_\phi(s_{t+1})$ is a learned estimate that aggregates many past samples, letting each transition influence many states with fewer new trajectories.

2. **Data reuse.** Because the critic is trained as a supervised model, the same batch of transitions can support many gradient steps and—in many variants—off-policy learning with replay, extracting substantially more information per collected sample.
3. **Generalization.** Parametric critics V_ϕ or Q_ω share information across similar states/actions, reducing how many distinct states must be visited to obtain useful return estimates.
4. **Faster return propagation.** Bootstrapped (TD / TD(λ)) updates propagate reward information backward across time over repeated passes on the same data, spreading credit more quickly than Monte Carlo methods and thereby lowering the number of fresh rollouts needed.

2.2 Exercise 6

(i) Entropy bonus encouraging exploration:

We rewrite Eq. (4) and recognize that

$$\mathbb{E}_{a \sim \pi_\theta(\cdot | s)}[-\log \pi_\theta(a | s)] = H(\pi_\theta(\cdot | s)),$$

the Shannon entropy. Hence, for $\alpha > 0$,

$$J(\theta) = \mathbb{E}_{s \sim p_0} \left[\mathbb{E}_{a \sim \pi_\theta(\cdot | s)} [A_\omega(a, s)] + \alpha H(\pi_\theta(\cdot | s)) \right]. \quad (32)$$

Maximizing J therefore *rewards* policies with higher entropy, i.e., more spread-out action probabilities.

For a discrete action set \mathcal{A} , $H(\pi(\cdot | s))$ is strictly concave in π and is maximized by the uniform policy. Using a Lagrange multiplier λ for the simplex constraint $\sum_a \pi(a | s) = 1$,

$$\frac{\partial}{\partial \pi(a | s)} \left[-\sum_{a'} \pi(a' | s) \log \pi(a' | s) + \lambda \left(\sum_{a'} \pi(a' | s) - 1 \right) \right] = -(1 + \log \pi(a | s)) + \lambda = 0,$$

so $\log \pi(a | s) = \lambda - 1$ for all a , implying $\pi(a | s) = 1/|\mathcal{A}|$.

Intuition from the gradient. Let $h(\pi) = -\sum_a \pi(a | s) \log \pi(a | s)$. Then

$$\frac{\partial h}{\partial \pi(a | s)} = -(1 + \log \pi(a | s)).$$

If the policy is very confident about some action a^* (i.e., $\pi(a^* | s) \approx 1$), then $\log \pi(a^* | s) \approx 0$ and $\frac{\partial h}{\partial \pi(a^* | s)} \approx -1 < 0$: ascending J decreases $\pi(a^* | s)$ and increases probabilities of other actions—*encouraging exploration*. When all actions have equal probability, $\pi(a | s) = 1/|\mathcal{A}|$ for all a , the entropy is maximal and the entropy term exerts no further push; exploration is already high.

Thus the $-\alpha \log \pi_\theta(a | s)$ bonus prevents premature collapse to a near-deterministic policy and systematically promotes trying alternative actions until the advantage term provides strong evidence.

(ii) Implementation results:

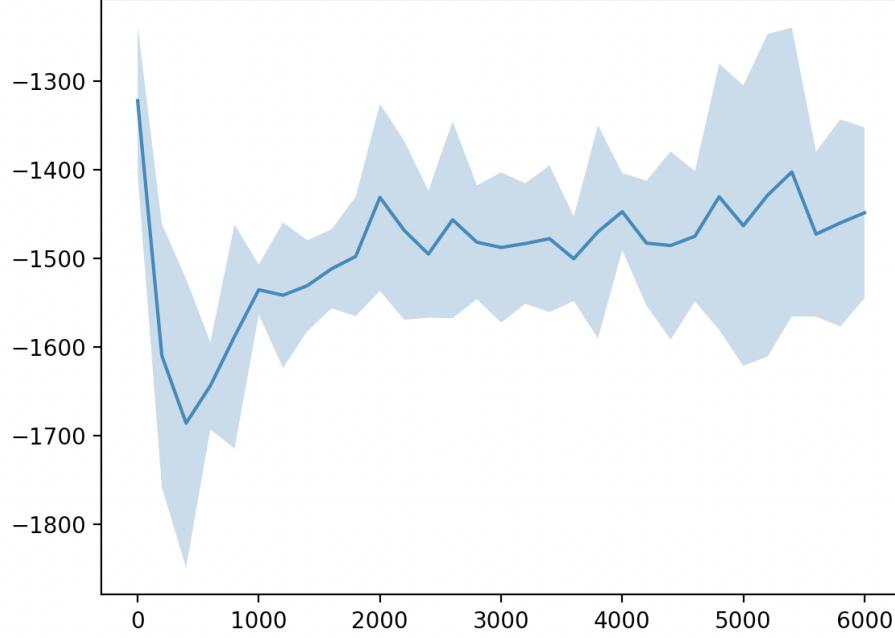


Figure 1: Results of 5 trials without the entropy term

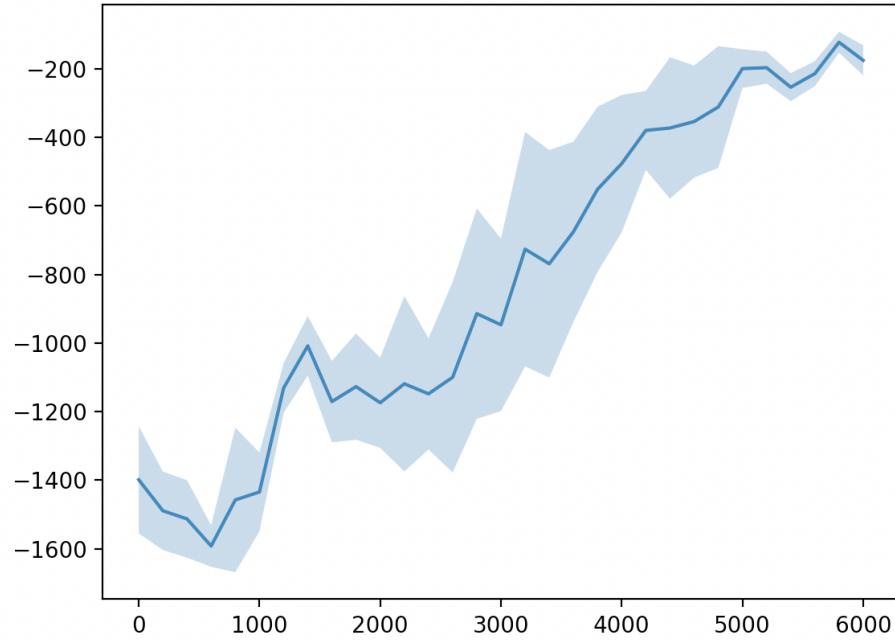


Figure 2: Results of 5 trials with the entropy term

Observations:

- **No entropy (Fig. 1):** Training shows an early dip and then hovers around a poor plateau (roughly -1.6×10^3 to -1.4×10^3) with growing variance across seeds. This suggests premature collapse to a near-deterministic, sub-optimal policy and weak

exploration.

- **With entropy (Fig. 2):** After the same initial dip, returns improve steadily throughout training, reaching about -2×10^2 by the end. The band widens mid-training (more exploration) and then narrows as policies converge, indicating both faster learning and greater stability across runs.

2.3 Exercise 7

(i) Greedy policies amplify approximation bias: With function approximation, we can write

$$Q_\omega(s, a) = Q^\star(s, a) + \varepsilon(s, a),$$

where ε is the approximation error. The greedy policy used by Q -learning is

$$\pi_\omega(a | s) = \delta\left(a = \arg \max_{a'} Q_\omega(s, a')\right).$$

Even if $\mathbb{E}[\varepsilon(s, a)] = 0$ for each fixed a , the selection step introduces a *maximization bias*:

$$\mathbb{E}\left[\max_a Q_\omega(s, a)\right] = \mathbb{E}\left[\max_a (Q^\star(s, a) + \varepsilon(s, a))\right] \geq \max_a Q^\star(s, a),$$

with strict inequality whenever the errors have nonzero variance. Hence the greedy action tends to be the one whose value is *overestimated* by the approximator. Because the induced policy is deterministic, the agent then commits to that action, gathers little or no data about alternatives, and the bootstrapped targets $r + \gamma \max_{a'} Q_\omega(s', a')$ propagate and reinforce the error. The result is over-optimistic value estimates, poor exploration, and potential divergence.

(ii) The optimality operator hides a hard inner maximization: The Bellman optimality operator for Q -learning is:

$$(\mathcal{T}^\star Q)(s, a) = r(s, a) + \gamma \mathbb{E}_{s' | s, a} \left[\max_{a'} Q(s', a') \right],$$

It requires computing $\max_{a'} Q_\omega(s', a')$ at every update. For continuous action spaces and general non-linear Q_ω (e.g., arbitrary neural networks), this is a nonconvex *global optimization* problem with no closed form. Practically, one must run a separate numerical optimizer per target, which is expensive and unreliable, and the $\arg \max$ is set-valued and non-differentiable, so gradients through the target are ill-behaved. This makes Q -learning cumbersome and unstable in continuous domains. Actor–critic methods avoid this inner maximization by learning a parametric actor $a = \mu_\theta(s)$ and using the differentiable target $r + \gamma Q_\omega(s', \mu_\theta(s'))$ instead.