Policy gradients and the log-derivative trick

Ten problems on differentiating an expectation through its distribution: the log-derivative trick, the zero-mean score and why baselines are free, the softmax policy gradient in closed form and a gradient step with numbers, the variance of the one-sample estimator and the optimal baseline, the Gaussian policy's scores, trajectories with the dynamics dropping out, reward-to-go, the entropy bonus and the importance-sampled surrogate, with worked solutions and the mistakes that move the gradient inside the expectation, differentiate with respect to the action, or weight actions by past rewards.

BEFORE YOU START

A policy gradient method improves a stochastic policy by gradient ascent on the expected reward, and the whole difficulty is that the parameters sit inside the distribution the expectation is taken over, not inside the function being averaged. One identity moves them: the gradient of a probability is the probability times the gradient of its logarithm. Everything in this family of methods, from REINFORCE to the clipped objectives of modern policy optimization, is that identity plus variance reduction. These ten problems derive the log-derivative trick, show why any constant baseline leaves the gradient unchanged, work out the softmax policy gradient in closed form and take one step with numbers, compute the variance of the one-sample estimator and the baseline that minimizes it, derive the Gaussian policy's score functions, extend the trick to trajectories where the environment's dynamics drop out, justify reward-to-go, differentiate an entropy bonus and end with the importance-sampled surrogate objective that lets old samples be reused. The five mistakes at the end are the ones that produce an update that still runs: the gradient moved inside the expectation, the score taken with respect to the action, a baseline that depends on the action, actions weighted by the rewards before them, and a softmax gradient with its mean term dropped.

- A policy is a distribution $p_\theta(a)$ over actions a with parameters θ: either a finite set $a \in \{1, \dots, K\}$ or a real number. A reward $r(a)$ does not depend on θ. The objective is $J(\theta) = \mathbb{E}_{a \sim p_\theta}[r(a)]$, which is maximized.
- The score function is $\nabla_\theta \log p_\theta(a)$, and the log-derivative identity is $\nabla_\theta p_\theta(a) = p_\theta(a)\, \nabla_\theta \log p_\theta(a)$, the chain rule on $\log$ rearranged.
- The softmax policy over K actions has $\theta \in \mathbb{R}^K$ and $p_\theta(a) = e^{\theta_a} / \sum_b e^{\theta_b}$; from the softmax page, $\partial p_a / \partial \theta_k = p_a(\delta_{ak} - p_k)$, with $\delta_{ak} = 1$ if $a = k$ and 0 otherwise. e_a is the a-th standard basis vector, $\mathbf{1}$ the all-ones vector, p the vector of probabilities and r the vector of rewards, so $J = p^\top r$. $\odot$ is the elementwise product.
- The Gaussian policy has $a \sim \mathcal{N}(\mu, \sigma^2)$ with $\theta = (\mu, \sigma)$; $\varepsilon \sim \mathcal{N}(0, 1)$ has $\mathbb{E}[\varepsilon] = \mathbb{E}[\varepsilon^3] = 0$ and $\mathbb{E}[\varepsilon^2] = 1$.

- A trajectory is $\tau = (s_0, a_0, s_1, a_1, \dots, s_{T-1}, a_{T-1})$ in an environment with start distribution $p(s_0)$, policy $\pi_\theta(a \mid s)$, transitions $p(s' \mid s, a)$ and rewards $r_t = r(s_t, a_t)$; the return is $R(\tau) = \sum_{t=0}^{T-1} r_t$ and $J(\theta) = \mathbb{E}_\tau[R(\tau)]$, the expectation over trajectories generated by π_θ.
- The entropy of the softmax policy is $H(p) = -\sum_a p_a \log p_a$ (the entropy page). An estimator $\hat{g}$ of a gradient is unbiased if $\mathbb{E}[\hat{g}] = \nabla_\theta J$; its variance is $\mathbb{E}\|\hat{g}\|^2 - \|\nabla_\theta J\|^2$, the trace of its covariance (the variance page).

Builds on: The softmax Jacobian

Problems

1. For a finite action set, show that $\nabla_\theta J = \mathbb{E}_{a \sim p_\theta}\big[r(a)\, \nabla_\theta \log p_\theta(a)\big]$. difficulty ·

2. Show that $\mathbb{E}_{a \sim p_\theta}[\nabla_\theta \log p_\theta(a)] = 0$, and deduce that for any constant b, $\mathbb{E}_{a \sim p_\theta}\big[(r(a) - b)\, \nabla_\theta \log p_\theta(a)\big] = \nabla_\theta J$. difficulty ·

3. For the softmax policy, show that $\nabla_\theta \log p_\theta(a) = e_a - p$ and that $\nabla_\theta J = p \odot (r - J\mathbf{1})$ with $J = p^\top r$. Show that the entries of $\nabla_\theta J$ sum to zero, and say why they must. difficulty ··

4. Take $K = 3$, $\theta_0 = (0, 0, 0)$ and $r = (1, 0, -1)$. Compute $J(\theta_0)$ and $\nabla_\theta J$ there, then $\theta_1 = \theta_0 + \eta \nabla_\theta J$ with $\eta = 1$, and the policy and expected reward at θ_1 to four decimal places. difficulty ··

5. The one-sample estimator with baseline b is $\hat{g}_b = (r(a) - b)(e_a - p)$ with $a \sim p$. Show that $\mathbb{E}\|\hat{g}_b\|^2 = \sum_a p_a (r_a - b)^2 \|e_a - p\|^2$, that the variance-minimizing baseline is $b^* = \dfrac{\sum_a p_a \|e_a - p\|^2 \, r_a}{\sum_a p_a \|e_a - p\|^2}$, and that $b^* = J$ when p is uniform. difficulty ···

6. For the Gaussian policy $a \sim \mathcal{N}(\mu, \sigma^2)$, compute $\partial_\mu \log p(a)$, $\partial_\sigma \log p(a)$ and $\partial_{\log \sigma} \log p(a)$. For $r(a) = -(a - a^*)^2$, show that $J = -\big((\mu - a^*)^2 + \sigma^2\big)$ and that $\mathbb{E}\big[r(a)\, \partial_\mu \log p(a)\big] = \partial J / \partial \mu = -2(\mu - a^*)$. difficulty ··

7. For trajectories, show that $\nabla_\theta \log p_\theta(\tau) = \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)$ and hence that $\nabla_\theta J = \mathbb{E}_\tau\Big[R(\tau) \sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\Big]$. difficulty ··

8. Show that $\mathbb{E}_\tau\big[r_{t'}\, \nabla_\theta \log \pi_\theta(a_t \mid s_t)\big] = 0$ for $t' < t$, and hence that $\nabla_\theta J = \mathbb{E}_\tau\Big[\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, \hat{G}_t\Big]$ with the reward-to-go $\hat{G}_t = \sum_{t' \geq t} r_{t'}$. difficulty ···

9. For the softmax policy, show that $\nabla_\theta H(p) = -p \odot (\log p + H\mathbf{1})$, and that this is the log-derivative trick applied to the θ-dependent reward $-\log p_\theta(a)$. difficulty ··

10. Actions are drawn from a behavior policy $q(a) > 0$ rather than from p_θ. Show that $\nabla_\theta J = \mathbb{E}_{a \sim q}\Big[\dfrac{p_\theta(a)}{q(a)}\, r(a)\, \nabla_\theta \log p_\theta(a)\Big]$, and that this is also the gradient of the surrogate $L(\theta) = \mathbb{E}_{a \sim q}\Big[\dfrac{p_\theta(a)}{q(a)}\, r(a)\Big]$. difficulty ···

Answers

1. $\nabla_\theta J = \mathbb{E}_{a\sim p_\theta}\big[r(a)\,\nabla_\theta \log p_\theta(a)\big]$
2. $\mathbb{E}_{a\sim p_\theta}[\nabla_\theta \log p_\theta(a)] = 0$, so $\mathbb{E}\big[(r(a)-b)\,\nabla_\theta \log p_\theta(a)\big] = \nabla_\theta J$ for every constant b
3. $\nabla_\theta \log p_\theta(a) = e_a - p$; $\nabla_\theta J = p \odot (r - J\mathbf{1})$, whose entries sum to zero
4. At θ_0: $J = 0$ and $\nabla_\theta J = (\tfrac{1}{3}, 0, -\tfrac{1}{3})$; $\theta_1 = (\tfrac{1}{3}, 0, -\tfrac{1}{3})$ with $p \approx (0.4484, 0.3213, 0.2302)$ and $J \approx 0.2182$
5. $\mathbb{E}\|\hat g_b\|^2 = \sum_a p_a (r_a - b)^2 \|e_a - p\|^2$; $b^* = \dfrac{\sum_a p_a \|e_a - p\|^2\, r_a}{\sum_a p_a \|e_a - p\|^2}$; for uniform p, $b^* = J$
6. $\partial_\mu \log p(a) = \dfrac{a-\mu}{\sigma^2}$, $\partial_\sigma \log p(a) = \dfrac{(a-\mu)^2 - \sigma^2}{\sigma^3}$, $\partial_{\log\sigma} \log p(a) = \dfrac{(a-\mu)^2}{\sigma^2} - 1$; $J = -\big((\mu - a^*)^2 + \sigma^2\big)$ and $\mathbb{E}\big[r(a)\,\partial_\mu \log p(a)\big] = -2(\mu - a^*) = \partial J/\partial \mu$
7. $\nabla_\theta \log p_\theta(\tau) = \sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)$; $\nabla_\theta J = \mathbb{E}_\tau\Big[R(\tau) \sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\Big]$
8. $\mathbb{E}_\tau\big[r_{t'}\,\nabla_\theta \log \pi_\theta(a_t \mid s_t)\big] = 0$ for $t' < t$; $\nabla_\theta J = \mathbb{E}_\tau\Big[\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\,\hat G_t\Big]$ with $\hat G_t = \sum_{t' \ge t} r_{t'}$
9. $\nabla_\theta H(p) = -p \odot (\log p + H\mathbf{1}) = \mathbb{E}_{a\sim p}\big[(-\log p_\theta(a) - H)(e_a - p)\big]$
10. $\nabla_\theta J = \mathbb{E}_{a\sim q}\Big[\dfrac{p_\theta(a)}{q(a)}\, r(a)\, \nabla_\theta \log p_\theta(a)\Big] = \nabla_\theta\, \mathbb{E}_{a\sim q}\Big[\dfrac{p_\theta(a)}{q(a)}\, r(a)\Big]$

Worked solutions

Problem 1

> For a finite action set, show that $\nabla_\theta J = \mathbb{E}_{a\sim p_\theta}\big[r(a)\,\nabla_\theta \log p_\theta(a)\big]$.

$J(\theta) = \sum_a p_\theta(a)\, r(a).$	The expectation over a finite set is a weighted sum.
$\nabla_\theta J = \sum_a r(a)\, \nabla_\theta p_\theta(a).$	Differentiate term by term; $r(a)$ does not depend on θ.
$\nabla_\theta p_\theta(a) = p_\theta(a)\, \nabla_\theta \log p_\theta(a).$	The chain rule gives $\nabla \log p = \nabla p / p$; multiply through by $p > 0$.
$\nabla_\theta J = \sum_a p_\theta(a)\, r(a)\, \nabla_\theta \log p_\theta(a) = \mathbb{E}_{a\sim p_\theta}\big[r(a)\,\nabla_\theta \log p_\theta(a)\big].$	Substitute; a sum weighted by p_θ is an expectation under p_θ.
$\nabla_\theta J = \mathbb{E}_{a\sim p_\theta}\big[r(a)\,\nabla_\theta \log p_\theta(a)\big]$	The gradient of an expectation whose distribution depends on θ is again an expectation under that distribution, so it can be estimated by sampling: draw a, compute $r(a)\nabla_\theta \log p_\theta(a)$. It uses only the value $r(a)$, never a derivative of r, so the reward can be a game score or any black box, and a can be discrete; that is what separates it from the ELBO page's reparameterization, which needs ∇r. For a continuous a the sum becomes an integral and the argument is unchanged (Problem 6). Mistake 1 is the gradient moved inside instead.

Problem 2

> Show that $\mathbb{E}_{a\sim p_\theta}[\nabla_\theta \log p_\theta(a)] = 0$, and deduce that for any constant b, $\mathbb{E}_{a\sim p_\theta}\big[(r(a) - b)\,\nabla_\theta \log p_\theta(a)\big] = \nabla_\theta J$.

$\mathbb{E}[\nabla_\theta \log p_\theta(a)] = \sum_a p_\theta(a)\, \nabla_\theta \log p_\theta(a) = \sum_a \nabla_\theta p_\theta(a).$	Problem 1's identity read from right to left.

$\sum_a \nabla_\theta p_\theta(a) = \nabla_\theta \sum_a p_\theta(a) = \nabla_\theta 1 = 0.$	A finite sum of gradients is the gradient of the sum, and the probabilities sum to 1 for every θ.
$\mathbb{E}\big[(r(a) - b)\,\nabla_\theta \log p_\theta(a)\big] = \mathbb{E}\big[r(a)\,\nabla_\theta \log p_\theta(a)\big] - b\,\mathbb{E}[\nabla_\theta \log p_\theta(a)] = \nabla_\theta J - 0.$	Linearity, then Problem 1 and step 2.
$\mathbb{E}_{a\sim p_\theta}[\nabla_\theta \log p_\theta(a)] = 0$, **so** $\mathbb{E}\big[(r(a) - b)\,\nabla_\theta \log p_\theta(a)\big] = \nabla_\theta J$ **for every constant** b	The score has mean zero because probabilities sum to one, differentiated. For the softmax, $\sum_a p_a(e_a - p) = p - p = 0$ directly. A baseline leaves the mean of the estimator alone and changes only its variance, which Problem 5 minimizes; it must not depend on the action drawn (Mistake 3).

Problem 3

> For the softmax policy, show that $\nabla_\theta \log p_\theta(a) = e_a - p$ and that $\nabla_\theta J = p \odot (r - J\mathbf{1})$ with $J = p^\top r$. Show that the entries of $\nabla_\theta J$ sum to zero, and say why they must.

$\log p_\theta(a) = \theta_a - \log \sum_b e^{\theta_b}.$	The log of the softmax.
$\dfrac{\partial \log p_\theta(a)}{\partial \theta_k} = \delta_{ak} - \dfrac{e^{\theta_k}}{\sum_b e^{\theta_b}} = \delta_{ak} - p_k$, so $\nabla_\theta \log p_\theta(a) = e_a - p.$	The derivative of the log-sum-exp is the softmax, as on the softmax page.
$\nabla_\theta J = \sum_a p_a r_a (e_a - p) = \sum_a p_a r_a e_a - \Big(\sum_a p_a r_a\Big) p = p \odot r - Jp.$	Problem 1 with step 2; $\sum_a p_a r_a e_a$ is the vector with entries $p_a r_a$.
$p \odot r - Jp = p \odot (r - J\mathbf{1}).$	Entry k is $p_k r_k - J p_k = p_k(r_k - J)$.
$\mathbf{1}^\top \nabla_\theta J = \sum_k p_k r_k - J \sum_k p_k = J - J = 0.$	$\sum_k p_k = 1$.

$\nabla_\theta \log p_\theta(a) = e_a - p$; $\nabla_\theta J = p \odot (r - J\mathbf{1})$, **whose entries sum to zero**	An action's logit rises when its reward is above the current average J and falls when below, at a rate proportional to its probability, so a rarely chosen good action moves slowly, which is the exploration problem in one line. The entries must sum to zero because adding a constant to every logit leaves the softmax unchanged (the softmax page's shift invariance): J is constant along $\mathbf{1}$, so its gradient is orthogonal to $\mathbf{1}$. Mistake 5 breaks this.

Problem 4

> Take $K = 3$, $\theta_0 = (0, 0, 0)$ and $r = (1, 0, -1)$. Compute $J(\theta_0)$ and $\nabla_\theta J$ there, then $\theta_1 = \theta_0 + \eta \nabla_\theta J$ with $\eta = 1$, and the policy and expected reward at θ_1 to four decimal places.

$p = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ and $J = \frac{1}{3}(1 + 0 - 1) = 0$.	The softmax of equal logits is uniform.
$\nabla_\theta J = p \odot (r - J\mathbf{1}) = \frac{1}{3}(1, 0, -1)$.	Problem 3 with $J = 0$.
$\theta_1 = \theta_0 + \eta \nabla_\theta J = (\frac{1}{3}, 0, -\frac{1}{3})$.	Gradient ascent on J with $\eta = 1$.
$e^{1/3} \approx 1.3956$, $e^0 = 1$, $e^{-1/3} \approx 0.7165$, summing to 3.1121, so $p \approx (0.4484, 0.3213, 0.2302)$.	The softmax at θ_1.
$J \approx 0.4484 - 0.2302 = 0.2182$.	$p^\top r$ with $r_2 = 0$.

At θ_0: $J = 0$ and $\nabla_\theta J = (\frac{1}{3}, 0, -\frac{1}{3})$; $\theta_1 = (\frac{1}{3}, 0, -\frac{1}{3})$ with $p \approx (0.4484, 0.3213, 0.2302)$ and $J \approx 0.2182$	One step moved probability from the worst action to the best and raised J from 0 to 0.218; the middle action, whose reward equals the mean, did not move. Repeating the steps sends $p \to e_1$ and $J \to 1$, ever more slowly, because the entry for action 3 is $p_3(r_3 - J)$ and $p_3 \to 0$. The exact gradient here is what a REINFORCE step estimates from a sample: drawing $a = 1$ gives $(1 - 0)(e_1 - p) = (\frac{2}{3}, -\frac{1}{3}, -\frac{1}{3})$, and the p-weighted average of the three possible samples is the gradient above.

Problem 5

> The one-sample estimator with baseline b is $\hat{g}_b = (r(a) - b)(e_a - p)$ with $a \sim p$. Show that $\mathbb{E}\|\hat{g}_b\|^2 = \sum_a p_a (r_a - b)^2 \|e_a - p\|^2$, that the variance-minimizing baseline is $b^* = \frac{\sum_a p_a \|e_a - p\|^2 r_a}{\sum_a p_a \|e_a - p\|^2}$, and that $b^* = J$ when p is uniform.

$\mathbb{E}\|\hat{g}_b\|^2 = \sum_a p_a \, (r_a - b)^2 \, \|e_a - p\|^2.$	The expectation over the finite set; the scalar $(r_a - b)$ comes out of the squared norm as its square.
$\mathrm{Var}(\hat{g}_b) = \mathbb{E}\|\hat{g}_b\|^2 - \|\nabla_\theta J\|^2$, and $\nabla_\theta J$ does not depend on b, so minimizing the variance over b is minimizing $\mathbb{E}\|\hat{g}_b\|^2$.	Problem 2: the mean of $\hat{g}_b$ is $\nabla_\theta J$ for every b.
$\frac{d}{db}\mathbb{E}\|\hat{g}_b\|^2 = -2\sum_a p_a (r_a - b)\|e_a - p\|^2 = 0$, so $b \sum_a p_a \|e_a - p\|^2 = \sum_a p_a \|e_a - p\|^2 \, r_a$.	Differentiate the square and set to zero.
$b^* = \frac{\sum_a p_a \|e_a - p\|^2 r_a}{\sum_a p_a \|e_a - p\|^2}$, and the second derivative $2\sum_a p_a \|e_a - p\|^2 > 0$ makes it the minimum.	A convex quadratic in b has its minimum at its stationary point.

For uniform p, $\|e_a - p\|^2 = \left(1 - \frac{1}{K}\right)^2 + (K-1)\frac{1}{K^2} = \dfrac{K-1}{K}$ for every a, so the weights are equal and $b^* = \sum_a p_a r_a = J$.	One entry of $e_a - p$ is $1 - \frac{1}{K}$ and the other $K-1$ are $-\frac{1}{K}$; $(K-1)^2 + (K-1) = (K-1)K$.
$\mathbb{E}\|\hat{g}_b\|^2 = \sum_a p_a (r_a - b)^2 \|e_a - p\|^2$; $b^* = \dfrac{\sum_a p_a \|e_a - p\|^2 r_a}{\sum_a p_a \|e_a - p\|^2}$; **for uniform** p, $b^* = J$	The best baseline is a weighted mean of the rewards, weighted toward actions with large scores. The mean reward J, or a running estimate of it, which in reinforcement learning is the value function, is the usual choice: exactly optimal for a uniform policy and close otherwise. At Problem 4's θ_0, $b = J = 0$ gives $\mathbb{E}\|\hat{g}\|^2 = \frac{4}{9}$ against $\frac{10}{9}$ for $b = 1$, with the same mean.

Problem 6

> For the Gaussian policy $a \sim \mathcal{N}(\mu, \sigma^2)$, compute $\partial_\mu \log p(a)$, $\partial_\sigma \log p(a)$ and $\partial_{\log\sigma} \log p(a)$. For $r(a) = -(a - a^*)^2$, show that $J = -\big((\mu - a^*)^2 + \sigma^2\big)$ and that $\mathbb{E}\big[r(a)\,\partial_\mu \log p(a)\big] = \partial J/\partial\mu = -2(\mu - a^*)$.

$\log p(a) = -\frac{1}{2}\log(2\pi) - \log\sigma - \dfrac{(a-\mu)^2}{2\sigma^2}$.	The Gaussian log-density.
$\partial_\mu \log p(a) = \dfrac{a-\mu}{\sigma^2}$ and $\partial_\sigma \log p(a) = -\dfrac{1}{\sigma} + \dfrac{(a-\mu)^2}{\sigma^3} = \dfrac{(a-\mu)^2 - \sigma^2}{\sigma^3}$.	Chain rule on $(a-\mu)^2$, whose inner derivative in μ is -1; power rule on σ^{-2} and the derivative of $-\log\sigma$.
$\partial_{\log\sigma} \log p(a) = \sigma\,\partial_\sigma \log p(a) = \dfrac{(a-\mu)^2}{\sigma^2} - 1$.	$d/d(\log\sigma) = \sigma\, d/d\sigma$, since $d\sigma/d\log\sigma = \sigma$.
$J = \mathbb{E}[-(a-a^*)^2] = -\big(\operatorname{Var}(a) + (\mathbb{E}[a] - a^*)^2\big) = -\big((\mu-a^*)^2 + \sigma^2\big)$, so $\partial J/\partial\mu = -2(\mu - a^*)$ and $\partial J/\partial\sigma = -2\sigma$.	The variance page's $\mathbb{E}[u^2] = \operatorname{Var}(u) + (\mathbb{E}[u])^2$ with $u = a - a^*$.

With $a = \mu + \sigma\varepsilon$: $\mathbb{E}\left[r(a)\dfrac{a-\mu}{\sigma^2}\right] = -\dfrac{1}{\sigma}\mathbb{E}\left[\left((\mu - a^*) + \sigma\varepsilon\right)^2\varepsilon\right] = -\dfrac{1}{\sigma}\mathbb{E}\left[(\mu - a^*)^2\varepsilon + 2(\mu - a^*)\sigma\varepsilon^2 + \sigma^2\varepsilon^3\right] = -2(\mu - a^*)$.	$(a-\mu)/\sigma^2 = \varepsilon/\sigma$; expand the square; $\mathbb{E}[\varepsilon] = \mathbb{E}[\varepsilon^3] = 0$ and $\mathbb{E}[\varepsilon^2] = 1$.
$\partial_\mu \log p(a) = \dfrac{a-\mu}{\sigma^2}$, $\partial_\sigma \log p(a) = \dfrac{(a-\mu)^2 - \sigma^2}{\sigma^3}$, $\partial_{\log\sigma} \log p(a) = \dfrac{(a-\mu)^2}{\sigma^2} - 1$; $J = -\left((\mu - a^*)^2 + \sigma^2\right)$ **and** $\mathbb{E}\left[r(a)\,\partial_\mu \log p(a)\right] = -2(\mu - a^*) = \partial J/\partial\mu$	The score in μ says that an action which landed above the mean pushes the mean up, weighted by its reward, and the score in σ widens the policy when actions farther than σ from the mean are rewarded; the check confirms $\mathbb{E}[r\,\partial_\sigma \log p] = -2\sigma$ too. The reparameterized route of the ELBO page would use $r'(a) = -2(a - a^*)$, the derivative of the reward, which an agent usually cannot evaluate; the log-derivative trick needs only values of r, at the cost of variance (30 against 4 in the ELBO page's Problem 6). Parameterizing by $\log\sigma$ keeps $\sigma > 0$ and makes the score scale-free. Mistake 2 differentiates in a instead of μ.

Problem 7

> For trajectories, show that $\nabla_\theta \log p_\theta(\tau) = \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)$ and hence that $\nabla_\theta J = \mathbb{E}_\tau\left[R(\tau)\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right]$.

$p_\theta(\tau) = p(s_0)\prod_{t=0}^{T-1}\pi_\theta(a_t \mid s_t)\prod_{t=0}^{T-2} p(s_{t+1} \mid s_t, a_t)$.	The chain rule of probability: each action depends on the current state through the policy and each next state on the current state and action through the dynamics.
$\log p_\theta(\tau) = \log p(s_0) + \sum_t \log \pi_\theta(a_t \mid s_t) + \sum_t \log p(s_{t+1} \mid s_t, a_t)$.	The log of a product.

$\nabla_\theta \log p_\theta(\tau) = \sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)$.	Only the policy factors contain θ; the start distribution and the dynamics do not.
$\nabla_\theta J = \mathbb{E}_\tau\big[R(\tau)\, \nabla_\theta \log p_\theta(\tau)\big] = \mathbb{E}_\tau\Big[R(\tau) \sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\Big]$.	Problem 1 with trajectories as the outcomes and $R(\tau)$ as the reward, then step 3.
$\nabla_\theta \log p_\theta(\tau) = \sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)$; $\nabla_\theta J = \mathbb{E}_\tau\Big[R(\tau) \sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\Big]$	The dynamics drop out: the estimator never needs $p(s' \mid s, a)$, which is why policy gradients are model-free. The check enumerates every trajectory of a two-state, two-action, three-step environment with a tabular softmax policy and compares the expectation with finite differences of J. This is REINFORCE; it weights every action by the whole return, including actions taken after a reward was already collected, which Problem 8 removes.

Problem 8

> Show that $\mathbb{E}_\tau\big[r_{t'}\, \nabla_\theta \log \pi_\theta(a_t \mid s_t)\big] = 0$ for $t' < t$, and hence that $\nabla_\theta J = \mathbb{E}_\tau\Big[\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, \hat{G}_t\Big]$ with the reward-to-go $\hat{G}_t = \sum_{t' \geq t} r_{t'}$.

Let $h_t = (s_0, a_0, \ldots, s_{t-1}, a_{t-1}, s_t)$ be the history up to the state at time t. Given h_t, $a_t \sim \pi_\theta(\cdot \mid s_t)$, so $\mathbb{E}\big[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \,\big|\, h_t\big] = \sum_a \pi_\theta(a \mid s_t) \nabla_\theta \log \pi_\theta(a \mid s_t) = 0$.	Problem 2 applied to the distribution $\pi_\theta(\cdot \mid s_t)$.
For $t' < t$, $r_{t'} = r(s_{t'}, a_{t'})$ is a function of h_t, so $\mathbb{E}\big[r_{t'} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\big] = \mathbb{E}\Big[r_{t'}\, \mathbb{E}\big[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \,\big|\, h_t\big]\Big] = \mathbb{E}[r_{t'} \cdot 0] = 0$.	Iterated expectations: condition on h_t, take $r_{t'}$ out of the inner expectation because it is fixed by h_t, then step 1.

$R(\tau)\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t) = \sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\Big(\sum_{t'<t} r_{t'} + \sum_{t'\geq t} r_{t'}\Big)$.	Split the return at t inside each term.
Taking expectations, every past term $r_{t'}$ with $t' < t$ contributes 0 by step 2, leaving $\nabla_\theta J = \mathbb{E}_\tau\Big[\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\sum_{t'\geq t} r_{t'}\Big]$.	Problem 7 and linearity.
$\mathbb{E}_\tau\big[r_{t'}\,\nabla_\theta \log \pi_\theta(a_t \mid s_t)\big] = 0$ **for** $t' < t$; $\nabla_\theta J = \mathbb{E}_\tau\Big[\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\,\hat{G}_t\Big]$ **with** $\hat{G}_t = \sum_{t'\geq t} r_{t'}$	An action cannot change rewards already received, and the algebra agrees: those terms are zero in expectation but not sample by sample, so dropping them keeps the estimate unbiased and usually lowers its variance, though not in every case: the same mechanism as Problem 2's baseline. A baseline may also be subtracted per state, $\hat{G}_t - b(s_t)$, because given s_t it is a constant with respect to a_t; that difference is the advantage. Weighting by the past rewards instead is Mistake 4. The check verifies both the zero and the equality on the enumerated environment.

Problem 9

> For the softmax policy, show that $\nabla_\theta H(p) = -p \odot (\log p + H\mathbf{1})$, and that this is the log-derivative trick applied to the θ-dependent reward $-\log p_\theta(a)$.

$\dfrac{\partial H}{\partial \theta_k} = -\sum_a \dfrac{\partial p_a}{\partial \theta_k}(\log p_a + 1)$.	Product rule on $p_a \log p_a$: its derivative is $(\log p_a + 1)\,\partial p_a$.
$\dfrac{\partial H}{\partial \theta_k} = -\sum_a p_a(\delta_{ak} - p_k)(\log p_a + 1) = -p_k(\log p_k + 1) + p_k \sum_a p_a(\log p_a + 1)$.	The softmax Jacobian; the δ_{ak} picks out $a = k$ and the p_k factors out of the rest.

$\sum_a p_a(\log p_a + 1) = -H + 1$, so $\dfrac{\partial H}{\partial \theta_k} = -p_k(\log p_k + 1) + p_k(1 - H) = -p_k(\log p_k + H)$.	$\sum_a p_a \log p_a = -H$ and $\sum_a p_a = 1$; the $\pm p_k$ cancel.
With $r_\theta(a) = -\log p_\theta(a)$, $H = \mathbb{E}_{a\sim p_\theta}[r_\theta(a)]$ and $\nabla_\theta H = \mathbb{E}\big[r_\theta(a)\nabla_\theta \log p_\theta(a)\big] + \mathbb{E}\big[\nabla_\theta r_\theta(a)\big]$; the second term is $-\mathbb{E}[\nabla_\theta \log p_\theta(a)] = 0$ and the first is $\sum_a p_a(-\log p_a)(e_a - p) = -p \odot \log p + \Big(\sum_a p_a \log p_a\Big)p = -p \odot \log p - Hp$.	Product rule, because both the distribution and the reward depend on θ; Problem 2 for the second term; Problem 3's score for the first.
$\nabla_\theta H(p) = -p \odot (\log p + H\mathbf{1}) = \mathbb{E}_{a\sim p}\big[(-\log p_\theta(a) - H)(e_a - p)\big]$	The entropy gradient pushes down the logits of actions whose surprise $-\log p_a$ is below the average surprise H, the likely ones, and up the unlikely ones; it points toward the uniform policy, where it vanishes. Adding βH to the objective adds β times this to the update, or in sampled form adds the bonus $-\log p_\theta(a) - H$ to each sampled reward, with H acting as its own baseline. The entries sum to zero again, as every softmax gradient must.

Problem 10

> Actions are drawn from a behavior policy $q(a) > 0$ rather than from p_θ. Show that $\nabla_\theta J = \mathbb{E}_{a\sim q}\Big[\dfrac{p_\theta(a)}{q(a)}\, r(a)\, \nabla_\theta \log p_\theta(a)\Big]$, and that this is also the gradient of the surrogate $L(\theta) = \mathbb{E}_{a\sim q}\Big[\dfrac{p_\theta(a)}{q(a)}\, r(a)\Big]$.

$J(\theta) = \sum_a p_\theta(a)\, r(a) = \sum_a q(a)\, \dfrac{p_\theta(a)}{q(a)}\, r(a) = \mathbb{E}_{a\sim q}\big[\rho_\theta(a)\, r(a)\big]$ with $\rho_\theta(a) = p_\theta(a)/q(a)$.	Multiply and divide by $q(a) > 0$: importance sampling, the same move as the ELBO page's Problem 1.

$\nabla_\theta \rho_\theta(a) = \dfrac{\nabla_\theta p_\theta(a)}{q(a)} = \rho_\theta(a)\, \nabla_\theta \log p_\theta(a).$	q does not depend on θ; the log-derivative identity.
$\nabla_\theta J = \sum_a q(a)\, r(a)\, \nabla_\theta \rho_\theta(a) = \mathbb{E}_{a\sim q}\big[\rho_\theta(a)\, r(a)\, \nabla_\theta \log p_\theta(a)\big].$	Differentiate step 1 term by term, which is allowed because the weights $q(a)$ do not depend on θ; then step 2.
The right-hand side of step 1 is $L(\theta)$, so step 3 is also $\nabla_\theta L$: J and L are the same function of θ, written two ways.	Step 1 is an identity for every θ, not only at the θ that generated the samples.
$\nabla_\theta J = \mathbb{E}_{a\sim q}\Big[\dfrac{p_\theta(a)}{q(a)}\, r(a)\, \nabla_\theta \log p_\theta(a)\Big] = \nabla_\theta\, \mathbb{E}_{a\sim q}\Big[\dfrac{p_\theta(a)}{q(a)}\, r(a)\Big]$	This is how samples from an old policy are reused: the surrogate L has the right gradient everywhere, and differentiating the ratio in an automatic-differentiation framework produces the score-function estimator by itself, because $\nabla\rho = \rho\nabla \log p$. When $q = p_{\theta_{\text{old}}}$ and $\theta = \theta_{\text{old}}$ the ratios are 1 and the estimator is Problem 1's. Proximal methods replace r by an advantage and clip or constrain the ratios, because the estimator's variance grows with ρ_θ as p_θ drifts away from q.

Where this goes wrong

1. Moving the gradient inside the expectation

For a fixed distribution the gradient of an expectation is the expectation of the gradient, and that is the first rule to reach for.

$J(\theta) = \mathbb{E}_{a\sim p_\theta}[r(a)]$	Right so far: the objective.
"The gradient of an expectation is the expectation of the gradient."	The rule that causes the mistake: it holds when the distribution does not depend on θ, as after the ELBO page's reparameterization; here θ lives in the distribution.

$\nabla_\theta J = \mathbb{E}_{a\sim p_\theta}[\nabla_\theta r(a)] = 0$ ✗	$r(a)$ does not depend on θ, so the line says J is constant, while Problem 4 raises it from 0 to 0.218 in one step. The dependence on θ sits in the weights $p_\theta(a)$, and differentiating those is Problem 1: $\nabla_\theta J = \mathbb{E}[r(a)\nabla_\theta \log p_\theta(a)]$. When the action can be reparameterized and r is differentiable both routes exist and the ELBO page compares them; for a discrete action or a black-box reward only the score-function route does.

2. Score taken with respect to the action instead of the parameter

For a Gaussian the log-density is a function of $a - \mu$, and the derivative in the exponent is easy to take in the wrong variable.

$\log p(a) = -\log\sigma - \dfrac{(a-\mu)^2}{2\sigma^2} - \frac{1}{2}\log(2\pi)$	Right so far: Problem 6, step 1.
"The score is the derivative of the log-density."	The shortcut that causes the mistake: it is the derivative with respect to the parameter being learned, not with respect to the variable the density is over.
$\nabla_\theta \log p(a) = \partial_a \log p(a) = -\dfrac{a-\mu}{\sigma^2}$ ✗	The sign is wrong: $\partial_\mu \log p(a) = +\dfrac{a-\mu}{\sigma^2}$ (Problem 6), so Problem 6's estimate becomes $+2(\mu - a^*)$ and gradient ascent moves μ away from a^*. The two agree in size only because the density depends on $a - \mu$; for σ there is no ∂_a counterpart at all. The multivariate-Gaussian page's score $\nabla_x \log \mathcal{N}(x;\mu,\Sigma)$ is a derivative in x, used for sampling, which is where the name collides.

3. A baseline that depends on the action

Any constant can be subtracted from the reward, and the obvious thing to subtract is something informative about the action taken.

$\mathbb{E}\big[(r(a) - b)\nabla_\theta \log p_\theta(a)\big] = \nabla_\theta J$ for a constant b	Right so far: Problem 2.
"Subtract a better reference: $b(a)$, what this action usually earns."	The habit that causes the mistake: Problem 2's proof takes b outside the expectation over a, and a function of a cannot come out.
$\nabla_\theta J = \mathbb{E}\big[(r(a) - b(a))\nabla_\theta \log p_\theta(a)\big]$ for any function b ✗	By Problem 1, $\mathbb{E}[b(a)\nabla_\theta \log p_\theta(a)] = \nabla_\theta \mathbb{E}_{p_\theta}[b(a)]$, which is zero only when $\mathbb{E}_{p_\theta}[b(a)]$ does not depend on θ; with $b(a) = r(a)$ the line claims $\nabla_\theta J = 0$. The state-dependent baseline $b(s_t)$ of Problem 8 is allowed because, given s_t, it is a constant with respect to the action a_t that follows.

4. Weighting each action by the rewards before it

Credit runs backwards in time, and a running total of the rewards seen so far is what a cumulative sum in code produces.

$\nabla_\theta J = \mathbb{E}\Big[\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t) \sum_{t' \geq t} r_{t'}\Big]$	Right so far: Problem 8.
"Each action's weight is the return accumulated up to that point."	The habit that causes the mistake: a forward cumulative sum, where the reward-to-go is the cumulative sum taken from the end.

$\nabla_\theta J = \mathbb{E}\Big[\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t) \sum_{t' \le t} r_{t'}\Big]$ ✗	By Problem 8 the terms with $t' < t$ have expectation zero, so the line equals $\mathbb{E}\big[\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, r_t\big]$: each action is credited with its immediate reward only. The estimate is biased, not merely noisy; it ignores every delayed consequence, so an action with a small cost now and a large payoff later is pushed away. On the check's environment the two expectations differ.

5. Softmax policy gradient with the mean-reward term dropped

$\nabla_\theta J = p \odot (r - J\mathbf{1})$ contains a term that looks like a baseline, and baselines are optional.

$\nabla_\theta J = \mathbb{E}_{a\sim p}\big[r(a)(e_a - p)\big]$	Right so far: Problem 3, step 3.
"The $-p$ is a baseline term, so it can be left out."	The shortcut that causes the mistake: the $-Jp$ comes from the $-p$ inside the score $e_a - p$, which is part of the gradient, not from a subtracted baseline.
$\nabla_\theta J = p \odot r$ ✗	Its entries sum to $\sum_a p_a r_a = J$, but a function invariant under $\theta \to \theta + c\mathbf{1}$ has a gradient orthogonal to $\mathbf{1}$ (Problem 3). At Problem 4's θ_0 the line happens to be right because $J = 0$ there; at θ_1 it gives $(0.448, 0, -0.230)$ where the gradient is $(0.351, -0.070, -0.280)$. The middle action, whose reward sits below the new mean, is never pushed down, and the update drifts along $\mathbf{1}$, where it changes nothing.

More at eulerpad.app. This is one of 82 free practice sets with worked solutions at eulerpad.app/practice, alongside guides to the topics behind them.

EulerPad, the app, reads your handwritten math on iPad, iPhone and Android and checks it line by line. It's free to start.

This set is also Chapter 25 of *Deep Learning Math by Hand*, a 250-problem workbook with worked solutions: eulerpad.app/workbooks

eulerpad.app/practice/policy-gradient-and-the-log-derivative-trick