第七章 策略梯度方法

本章的内容是策略学习 (Policy-Based Reinforcement Learning) 以及策略梯度 (Policy Gradient)。策略学习的意思是通过求解一个优化问题，学出最优策略函数或它的近似（比如策略网络）。第 7.1 节描述策略网络。第 7.2 节把策略学习描述成一个最大化的问题。第 7.3 节推导策略梯度。第 7.4 和 7.5 节用不同的方法近似策略梯度，得到两种训练策略网络的方法——REINFORCE 和 Actor-Critic。本章介绍的 REINFORCE 和 Actor-Critic 只是帮助大家理解算法而已，实际效果并不好。在实践中不建议用本章原始的方法，而应该用下一章的方法。

7.1 策略网络

本章考虑离散动作空间，比如 $\mathcal{A} = \{左, 右, 上\}$。策略函数 π 是个条件概率质量函数：

$$\pi(a \mid s) \ \triangleq\ \mathbb{P}(A = a \mid S = s).$$

策略函数 π 的输入是状态 s 和动作 a，输出是一个 0 到 1 之间的概率值。举个例子，把超级玛丽游戏当前屏幕上的画面作为 s，策略函数会输出每个动作的概率值：

$$\begin{aligned}\pi(左 \mid s) &= 0.5,\\ \pi(右 \mid s) &= 0.2,\\ \pi(上 \mid s) &= 0.3.\end{aligned}$$

如果我们有这样一个策略函数，我们就可以拿它控制智能体。每当观测到一个状态 s，就用策略函数计算出每个动作的概率值，然后做随机抽样，得到一个动作 a，让智能体执行 a。

怎么样才能得到这样一个策略函数呢？当前最有效的方法是用神经网络 $\pi(a|s;\boldsymbol{\theta})$ 近似策略函数 $\pi(a|s)$。神经网络 $\pi(a|s;\boldsymbol{\theta})$ 被称为策略网络。$\boldsymbol{\theta}$ 表示神经网络的参数；一开始随机初始化 $\boldsymbol{\theta}$，随后利用收集的状态、动作、奖励去更新 $\boldsymbol{\theta}$。

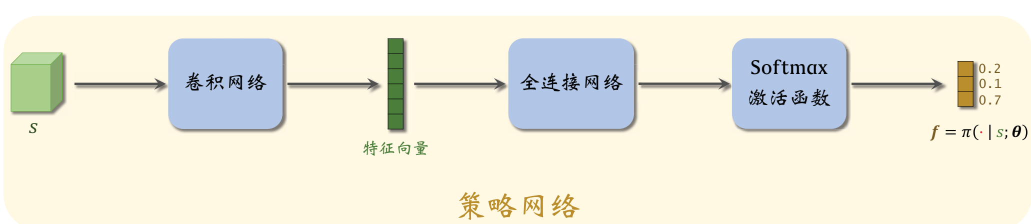

图 7.1: 策略网络 $\pi(a|s;\boldsymbol{\theta})$ 的神经网络结构。输入是状态 s，输出是动作空间 $\mathcal{A}$ 中每个动作的概率值。

策略网络的结构如图 7.1 所示。策略网络的输入是状态 s。在 Atari 游戏、围棋等应用中，状态是张量（比如图片），那么应该如图 7.1 所示用卷积网络处理输入。在机器人控制等应用中，状态 s 是向量，它的元素是多个传感器的数值，那么应该把卷积网络换成

全连接网络。策略网络输出层的激活函数是 Softmax，因此输出的向量（记作 $\boldsymbol{f}$）所有元素都是正数，而且相加等于 1。动作空间 $\mathcal{A}$ 的大小是多少，向量 $\boldsymbol{f}$ 的维度就是多少。在超级玛丽的例子中，$\mathcal{A} = \{左, 右, 上\}$，那么 $\boldsymbol{f}$ 就是 3 维的向量，比如 $\boldsymbol{f} = [0.2, 0.1, 0.7]$。$\boldsymbol{f}$ 描述了动作空间 $\mathcal{A}$ 上的离散概率分布，$\boldsymbol{f}$ 每个元素对应一个动作：

$$
\begin{aligned}
f_1 &= \pi\big(左 \,|\, s\big) = 0.2, \\
f_2 &= \pi\big(右 \,|\, s\big) = 0.1, \\
f_3 &= \pi\big(上 \,|\, s\big) = 0.7.
\end{aligned}
$$

7.2 策略学习的目标函数

为了推导策略学习的目标函数，我们需要先复习回报和价值函数。回报 U_t 是从 t 从时刻开始的所有奖励之和。U_t 依赖于 t 时刻开始的所有状态和动作：

$$S_t, A_t, \quad S_{t+1}, A_{t+1}, \quad S_{t+2}, A_{t+2}, \quad \cdots$$

在 t 时刻，U_t 是随机变量，它的不确定性来自于未来未知的状态和动作。动作价值函数的定义是：

$$Q_\pi\big(s_t, a_t\big) \;=\; \mathbb{E}\Big[U_t \,\Big|\, S_t = s_t, A_t = a_t\Big].$$

条件期望把 t 时刻状态 s_t 和动作 a_t 看做已知观测值，把 $t+1$ 时刻后的状态和动作看做未知变量，并消除这些变量。状态价值函数的定义是

$$V_\pi\big(s_t\big) \;=\; \mathbb{E}_{A_t \sim \pi(\cdot|s_t;\boldsymbol{\theta})}\Big[Q_\pi\big(s_t, A_t\big)\Big].$$

状态价值既依赖于当前状态 s_t，也依赖于策略网络 π 的参数 $\boldsymbol{\theta}$。

- 当前状态 s_t 越好，则 $V_\pi(s_t)$ 越大，也就是回报 U_t 的期望越大。例如，在超级玛丽游戏中，如果玛丽奥已经接近终点（也就是说当前状态 s_t 很好），那么回报的期望就会很大。
- 策略 π 越好（即参数 $\boldsymbol{\theta}$ 越好），那么 $V_\pi(s_t)$ 也会越大。例如，从同一起点出发打游戏，高手（好的策略）的期望回报远高于初学者（差的策略）。

如果一个策略很好，那么对于所有的状态 S，状态价值 $V_\pi(S)$ 的均值应当很大。因此我们定义目标函数：

$$\boxed{J(\boldsymbol{\theta}) \;=\; \mathbb{E}_S\Big[V_\pi(S)\Big].}$$

这个目标函数排除掉了状态 S 的因素，只依赖于策略网络 π 的参数 $\boldsymbol{\theta}$；策略越好，则 $J(\boldsymbol{\theta})$ 越大。所以策略学习可以描述为这样一个优化问题：

$$\boxed{\max_{\boldsymbol{\theta}} \; J(\boldsymbol{\theta}).}$$

我们希望通过对策略网络参数 $\boldsymbol{\theta}$ 的更新，使得目标函数 $J(\boldsymbol{\theta})$ 越来越大，也就意味着策略网络越来越强。想要求解最大化问题，显然可以用梯度上升更新 $\boldsymbol{\theta}$，使得 $J(\boldsymbol{\theta})$ 增大。设当前策略网络的参数为 $\boldsymbol{\theta}_{\text{now}}$。做梯度上升更新参数，得到新的参数 $\boldsymbol{\theta}_{\text{new}}$：

$$\boldsymbol{\theta}_{\text{new}} \;\leftarrow\; \boldsymbol{\theta}_{\text{now}} + \beta \cdot \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_{\text{now}}).$$

此处的 β 是学习率，需要手动调。上面的公式就是训练策略网络的基本想法，其中的梯度

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_{\text{now}}) \;\triangleq\; \frac{\partial\, J(\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{now}}}$$

被称作策略梯度。策略梯度可以写成下面定理中的期望形式。之后的算法推导都要基于这个定理，并对其中的期望做近似。

定理 7.1. 策略梯度定理（不严谨的表述）

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbb{E}_S\left[\mathbb{E}_{A\sim\pi(\cdot|S;\boldsymbol{\theta})}\left[\frac{\partial \ln \pi(A|S;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot Q_\pi(S,A)\right]\right].$$

♡

注 上面是策略梯度定理是不严谨的表述，尽管大多数论文和书籍使用这种表述。严格地讲，这个定理只有在“状态 S 服从马尔科夫链的稳态分布 $d(\cdot)$”这个假设下才成立。定理中的等号其实是不对的，期望前面应该有一项系数 $1+\gamma+\cdots+\gamma^{n-1}=\frac{1-\gamma^n}{1-\gamma}$，其中 γ 是折扣率，n 是一局游戏的长度。策略梯度定理应该是：

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1-\gamma^n}{1-\gamma} \cdot \mathbb{E}_{S\sim d(\cdot)}\left[\mathbb{E}_{A\sim\pi(\cdot|S;\boldsymbol{\theta})}\left[\frac{\partial \ln \pi(A|S;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot Q_\pi(S,A)\right]\right].$$

在实际应用中，系数 $\frac{1-\gamma^n}{1-\gamma}$ 无关紧要，可以忽略掉。其原因是做梯度上升的时候，系数 $\frac{1-\gamma^n}{1-\gamma}$ 会被学习率 β 吸收。

7.3 策略梯度定理的证明

策略梯度定理是策略学习的关键所在。本节的内容是证明策略梯度定理。尽管本节数学较多，但还是建议读者认真读完第 7.3.1 小节，理解策略梯度简化的推导。第 7.3.2 小节是策略梯度定理完整的证明。由于完整证明较为复杂，大多数教材中不涉及这部分内容，本书也不建议读者理解、掌握完整证明，除非读者从事强化学习科研。

7.3.1 简化的证明

把策略网络 $\pi(a\,|\,s;\,\boldsymbol{\theta})$ 看做动作的概率质量函数（或概率密度函数）。状态价值函数 $V_\pi(s)$ 可以写成：

$$\begin{aligned}
V_\pi(s) &= \mathbb{E}_{A\sim\pi(\cdot|s;\boldsymbol{\theta})}\Big[Q_\pi(s,A)\Big] \\
&= \sum_{a\in\mathcal{A}} \pi\big(a\,\big|\,s;\,\boldsymbol{\theta}\big)\cdot Q_\pi\big(s,a\big).
\end{aligned}$$

状态价值 $V_\pi(s)$ 关于 $\boldsymbol{\theta}$ 的梯度可以写作：

$$\begin{aligned}
\frac{\partial\, V_\pi(s)}{\partial\,\boldsymbol{\theta}} &= \frac{\partial}{\partial\,\boldsymbol{\theta}}\sum_{a\in\mathcal{A}} \pi\big(a\,\big|\,s;\,\boldsymbol{\theta}\big)\cdot Q_\pi\big(s,a\big) \\
&= \sum_{a\in\mathcal{A}} \frac{\partial\,\pi(a|s;\,\boldsymbol{\theta})\cdot Q_\pi(s,a)}{\partial\,\boldsymbol{\theta}}.
\end{aligned} \tag{7.1}$$

上面第二个等式把求导放入连加里面；等式成立的原因是求导的对象 $\boldsymbol{\theta}$ 与连加的对象 a 不同。回忆一下链式法则：设 $z=f(x)\cdot g(x)$，那么

$$\frac{\partial\, z}{\partial\, x} = \frac{\partial\, f(x)}{\partial\, x}\cdot g(x) + f(x)\cdot\frac{\partial\, g(x)}{\partial\, x}.$$

应用链式法则，公式 (7.1) 中的梯度可以写作：

$$\begin{aligned}
\frac{\partial\, V_\pi(s)}{\partial\,\boldsymbol{\theta}} &= \sum_{a\in\mathcal{A}} \frac{\partial\,\pi(a|s;\boldsymbol{\theta})}{\partial\,\boldsymbol{\theta}}\cdot Q_\pi\big(s,a\big) + \sum_{a\in\mathcal{A}} \pi\big(a\,\big|\,s;\,\boldsymbol{\theta}\big)\cdot\frac{\partial\, Q_\pi(s,a)}{\partial\,\boldsymbol{\theta}} \\
&= \sum_{a\in\mathcal{A}} \frac{\partial\,\pi(a|s;\boldsymbol{\theta})}{\partial\,\boldsymbol{\theta}}\cdot Q_\pi\big(s,a\big) + \underbrace{\mathbb{E}_{A\sim\pi(\cdot|s;\boldsymbol{\theta})}\left[\frac{\partial\, Q_\pi(s,A)}{\partial\,\boldsymbol{\theta}}\right]}_{\text{设为 } x}.
\end{aligned}$$

上面公式最右边一项 x 的分析非常复杂，此处不具体分析了。由上面的公式可得：

$$\begin{aligned}
\frac{\partial\, V_\pi(s)}{\partial\,\boldsymbol{\theta}} &= \sum_{A\in\mathcal{A}} \frac{\partial\,\pi(A|S;\boldsymbol{\theta})}{\partial\,\boldsymbol{\theta}}\cdot Q_\pi\big(S,A\big) + x \\
&= \sum_{A\in\mathcal{A}} \pi\big(A\,\big|\,S;\,\boldsymbol{\theta}\big)\cdot\underbrace{\frac{1}{\pi(A\,|\,S;\,\boldsymbol{\theta})}\cdot\frac{\partial\,\pi(A\,|\,S;\,\boldsymbol{\theta})}{\partial\,\boldsymbol{\theta}}}_{\text{等于 } \partial\ln\pi(A\,|\,S;\boldsymbol{\theta})\;/\;\partial\boldsymbol{\theta}}\cdot Q_\pi\big(S,A\big) + x.
\end{aligned}$$

上面第二个等式成立的原因是添加的两个红色项相乘等于一。公式中用下花括号标出的项等于 $\frac{\partial\,\ln\pi(A\,|\,S;\boldsymbol{\theta})}{\partial\,\boldsymbol{\theta}}$。由此可得

$$\begin{aligned}
\frac{\partial\, V_\pi(s)}{\partial\,\boldsymbol{\theta}} &= \sum_{A\in\mathcal{A}} \pi\big(A\,\big|\,S;\,\boldsymbol{\theta}\big)\cdot\frac{\partial\,\ln\pi(A\,|\,S;\,\boldsymbol{\theta})}{\partial\,\boldsymbol{\theta}}\cdot Q_\pi\big(S,A\big) + x \\
&= \mathbb{E}_{A\sim\pi(\cdot\,|\,S;\boldsymbol{\theta})}\left[\frac{\partial\,\ln\pi(A\,|\,S;\,\boldsymbol{\theta})}{\partial\,\boldsymbol{\theta}}\cdot Q_\pi\big(S,A\big)\right] + x.
\end{aligned} \tag{7.2}$$

公式中红色标出的 $\pi(A|S;\boldsymbol{\theta})$ 被看做概率质量函数，因此连加可以写成期望的形式。根据目标函数的定义 $J(\boldsymbol{\theta}) = \mathbb{E}_S[V_\pi(S)]$ 可得

$$\begin{aligned}\frac{\partial\, J(\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} &= \mathbb{E}_S\left[\frac{\partial\, V_\pi(S)}{\partial\, \boldsymbol{\theta}}\right] \\ &= \mathbb{E}_S\left[\mathbb{E}_{A\sim\pi(\cdot\,|\,S;\boldsymbol{\theta})}\left[\frac{\partial\, \ln\pi(A\,|\,S;\,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(S,A)\right]\right] + \mathbb{E}_S[x].\end{aligned}$$

不严谨的证明通常忽略掉 x，于是得到定理 7.1。在下一小节中，我们给出严格的证明。除非读者对强化学习的数学很感兴趣，否则没必要阅读下一小节。

7.3.2 完整的证明

本小节给出策略梯度定理的严格数学证明。首先证明几个引理，最后用引理证明策略梯度定理。引理 7.2 分析梯度 $\frac{\partial V_\pi(s)}{\partial \boldsymbol{\theta}}$，并把它递归地表示为 $\frac{\partial V_\pi(S')}{\partial \boldsymbol{\theta}}$ 的期望，其中 S' 是下一时刻的状态。

引理 7.2. 递归公式

$$\frac{\partial\, V_\pi(s)}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{A\sim\pi(\cdot|s;\boldsymbol{\theta})}\left[\frac{\partial\, \ln\pi(A|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,A) + \gamma\cdot \mathbb{E}_{S'\sim p(\cdot|s,A)}\left[\frac{\partial\, V_\pi(S')}{\partial\, \boldsymbol{\theta}}\right]\right].$$

♡

证明 设奖励 R 和新状态 S' 是在智能体执行动作 A 之后由环境给出的。新状态 S' 的概率密度函数是状态转移函数 $p(S'|S,A)$。设奖励 R 是 S、A、S' 三者的函数，因此可以将其记为 $R(S,A,S')$。由贝尔曼方程可得：

$$\begin{aligned}Q_\pi(s,a) &= \mathbb{E}_{S'\sim p(\cdot|s,a)}\Big[R(s,a,S') + \gamma\cdot V_\pi(s')\Big] \\ &= \sum_{s'\in\mathcal{S}} p(s'|s,a)\cdot\Big[R(s,a,s') + \gamma\cdot V_\pi(s')\Big] \\ &= \sum_{s'\in\mathcal{S}} p(s'|s,a)\cdot R(s,a,s') + \gamma\cdot\sum_{s'\in\mathcal{S}} p(s'|s,a)\cdot V_\pi(s').\end{aligned} \tag{7.3}$$

在观测到 s、a、s' 之后，$p(s'|s,a)$ 和 $R(s,a,s')$ 都与策略网络 π 无关，因此

$$\frac{\partial}{\partial\, \boldsymbol{\theta}}\Big[p(s'\,|\,s,a)\cdot R(s,a,s')\Big] = 0. \tag{7.4}$$

由公式 (7.3) 与 (7.4) 可得：

$$\begin{aligned}\frac{\partial\, Q_\pi(s,a)}{\partial\, \boldsymbol{\theta}} &= \underbrace{\sum_{s'\in\mathcal{S}}\frac{\partial}{\partial\, \boldsymbol{\theta}}\Big[p(s'\,|\,s,a)\cdot R(s,a,s')\Big]}_{\text{等于零}} + \gamma\cdot\sum_{s'\in\mathcal{S}}\frac{\partial}{\partial\, \boldsymbol{\theta}}\Big[p(s'\,|\,s,a)\cdot V_\pi(s')\Big] \\ &= \gamma\cdot\sum_{s'\in\mathcal{S}} p(s'\,|\,s,a)\cdot\frac{\partial\, V_\pi(s')}{\partial\, \boldsymbol{\theta}} \\ &= \gamma\cdot\mathbb{E}_{S'\sim p(\cdot|s,a)}\left[\frac{\partial\, V_\pi(S')}{\partial\, \boldsymbol{\theta}}\right].\end{aligned} \tag{7.5}$$

由上一小节的公式 (7.2) 可得：

$$\begin{aligned}&\frac{\partial\, V_\pi(s)}{\partial\, \boldsymbol{\theta}} \\ &= \mathbb{E}_{A\sim\pi(\cdot|S;\boldsymbol{\theta})}\left[\frac{\partial\, \ln\pi(A|S;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}\cdot Q_\pi(S,A)\right] + \mathbb{E}_{A\sim\pi(\cdot|S;\boldsymbol{\theta})}\left[\frac{\partial\, Q_\pi(s,a)}{\partial\, \boldsymbol{\theta}}\right].\end{aligned} \tag{7.6}$$

结合公式 (7.5)、(7.6) 可得引理 7.2 □

引理 7.3. 策略梯度的连加形式

设 $\boldsymbol{g}(s,a;\boldsymbol{\theta}) \triangleq Q_\pi(s,a)\cdot\frac{\partial\ln\pi(a|s;\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}$。设一局游戏在第 n 步之后结束。那么

$$\begin{aligned}\frac{\partial J(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} \;=\;& \mathbb{E}_{S_1,A_1}\Big[\boldsymbol{g}(S_1,A_1;\boldsymbol{\theta})\Big] \\ &+\gamma\cdot\mathbb{E}_{S_1,A_1,S_2,A_2}\Big[\boldsymbol{g}(S_2,A_2;\boldsymbol{\theta})\Big] \\ &+\gamma^2\cdot\mathbb{E}_{S_1,A_1,S_2,A_2,S_3,A_3}\Big[\boldsymbol{g}(S_3,A_3;\boldsymbol{\theta})\Big] \\ &+\cdots \\ &+\gamma^{n-1}\cdot\mathbb{E}_{S_1,A_1,S_2,A_2,S_3,A_3,\cdots S_n,A_n}\Big[\boldsymbol{g}(S_n,A_n;\boldsymbol{\theta})\Big]\Big].\end{aligned}$$

♡

证明 设 S、A 为当前状态和动作，S' 为下一个状态。引理 7.2 证明了下面的结论：

$$\frac{\partial V_\pi(S)}{\partial\boldsymbol{\theta}} \;=\; \mathbb{E}_A\Bigg[\underbrace{\frac{\partial\ln\pi(A|S;\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\cdot Q_\pi(S,A)}_{\text{定义为 }\boldsymbol{g}(S,A;\boldsymbol{\theta})}+\gamma\cdot\mathbb{E}_{S'}\bigg[\frac{\partial V_\pi(S')}{\partial\boldsymbol{\theta}}\bigg]\Bigg].$$

这样我们可以把 $\frac{\partial V_\pi(S_1)}{\partial\boldsymbol{\theta}}$ 写成递归的形式：

$$\frac{\partial V_\pi(S_1)}{\partial\boldsymbol{\theta}} \;=\; \mathbb{E}_{A_1}\Big[\boldsymbol{g}(S_1,A_1;\boldsymbol{\theta})\Big] \;+\; \gamma\cdot\mathbb{E}_{A_1,S_2}\bigg[\frac{\partial V_\pi(S_2)}{\partial\boldsymbol{\theta}}\bigg]. \tag{7.7}$$

同理，$\frac{\partial V_\pi(S_2)}{\partial\boldsymbol{\theta}}$ 可以写成

$$\frac{\partial V_\pi(S_2)}{\partial\boldsymbol{\theta}} \;=\; \mathbb{E}_{A_2}\Big[\boldsymbol{g}(S_2,A_2;\boldsymbol{\theta})\Big] \;+\; \gamma\cdot\mathbb{E}_{A_2,S_3}\bigg[\frac{\partial V_\pi(S_3)}{\partial\boldsymbol{\theta}}\bigg]. \tag{7.8}$$

把等式 (7.8) 插入等式 (7.7)，得到

$$\begin{aligned}\frac{\partial V_\pi(S_1)}{\partial\boldsymbol{\theta}} \;=\;& \mathbb{E}_{A_1}\Big[\boldsymbol{g}(S_1,A_1;\boldsymbol{\theta})\Big] \\ &+\gamma\cdot\mathbb{E}_{A_1,S_2,A_2}\Big[\boldsymbol{g}(S_2,A_2;\boldsymbol{\theta})\Big] \\ &+\gamma^2\cdot\mathbb{E}_{A_1,S_2,A_2,S_3}\bigg[\frac{\partial V_\pi(S_3)}{\partial\boldsymbol{\theta}}\bigg].\end{aligned}$$

按照这种规律递归下去，可得：

$$\begin{aligned}\frac{\partial V_\pi(S_1)}{\partial\boldsymbol{\theta}} \;=\;& \mathbb{E}_{A_1}\Big[\boldsymbol{g}(S_1,A_1;\boldsymbol{\theta})\Big] \\ &+\gamma\cdot\mathbb{E}_{A_1,S_2,A_2}\Big[\boldsymbol{g}(S_2,A_2;\boldsymbol{\theta})\Big] \\ &+\gamma^2\cdot\mathbb{E}_{A_1,S_2,A_2,S_3,A_3}\Big[\boldsymbol{g}(S_3,A_3;\boldsymbol{\theta})\Big] \\ &+\cdots \\ &+\gamma^{n-1}\cdot\mathbb{E}_{A_1,S_2,A_2,S_3,A_3,\cdots S_n,A_n}\Big[\boldsymbol{g}(S_n,A_n;\boldsymbol{\theta})\Big] \\ &+\gamma^{n}\cdot\mathbb{E}_{A_1,S_2,A_2,S_3,A_3,\cdots S_n,A_n,S_{n+1}}\bigg[\underbrace{\frac{\partial V_\pi(S_{n+1})}{\partial\boldsymbol{\theta}}}_{\text{等于零}}\bigg].\end{aligned}$$

上式中最后一项等于零，原因是游戏在 n 时刻后结束，而 $n+1$ 时刻之后没有奖励，所以 $n+1$ 时刻的回报和价值都是零。最后，由上面的公式和

$$\frac{\partial J(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} \;=\; \mathbb{E}_{S_1}\bigg[\frac{\partial V_\pi(S_1)}{\partial\boldsymbol{\theta}}\bigg]$$

可得引理 7.3. □

稳态分布： 想要严格证明策略梯度定理，需要用到马尔科夫链 (Markov Chain) 的稳态分布 (Stationary Distribution)。设状态 s' 是这样得到的：$s \to a \to s'$。回忆一下，状态转移函数 $p(s'|s,a)$ 是一个概率密度函数。设 $d(s)$ 是状态 s 的概率密度函数。那么状态 s' 的边缘分布是

$$\tilde{d}(s') \;=\; \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} p(s'|s,a)\cdot\pi(a|s;\boldsymbol{\theta})\cdot d(s).$$

如果 $\tilde{d}(\cdot)$ 与 $d(\cdot)$ 是相同的概率密度函数，即 $d'(s)=d(s), \forall s\in\mathcal{S}$，则意味着马尔科夫链达到稳态，而 $d(\cdot)$ 就是稳态时的概率密度函数。

引理 7.4

设 $d(\cdot)$ 是马尔科夫链稳态时的概率密度函数。那么对于任意函数 $f(S')$，

$$\mathbb{E}_{S\sim d(\cdot)}\Big[\mathbb{E}_{A\sim\pi(\cdot|S;\boldsymbol{\theta})}\Big[\mathbb{E}_{S'\sim p(\cdot|s,A)}\Big[f(S')\Big]\Big]\Big] \;=\; \mathbb{E}_{S'\sim d(\cdot)}\Big[f(S')\Big].$$

♡

证明 把引理中的期望写成连加的形式：

$$\begin{aligned}
&\mathbb{E}_{S\sim d(\cdot)}\Big[\mathbb{E}_{A\sim\pi(\cdot|S;\boldsymbol{\theta})}\Big[\mathbb{E}_{S'\sim p(\cdot|s,A)}\Big[f(S')\Big]\Big]\Big]\\
=\;&\sum_{s\in\mathcal{S}} d(s)\sum_{a\in\mathcal{A}}\pi(a|s;\boldsymbol{\theta})\sum_{s'\in\mathcal{S}} p(s'|s,a)\cdot f(s')\\
=\;&\sum_{s'\in\mathcal{S}} f(s')\underbrace{\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} p(s'|s,a)\cdot\pi(a|s;\boldsymbol{\theta})\cdot d(s)}_{\text{等于 } d(s')}.
\end{aligned}$$

上面等式最右边标出的项等于 $d(s')$，这是根据稳态分布的定义得到的。于是有

$$\begin{aligned}
\mathbb{E}_{S\sim d(\cdot)}\Big[\mathbb{E}_{A\sim\pi(\cdot|S;\boldsymbol{\theta})}\Big[\mathbb{E}_{S'\sim p(\cdot|s,A)}\Big[f(S')\Big]\Big]\Big] \;&=\; \sum_{s'\in\mathcal{S}} f(s')\cdot d(s')\\
&=\; \mathbb{E}_{S'\sim d(\cdot)}\Big[f(S')\Big].
\end{aligned}$$

由此可得引理 7.4 □

定理 7.5. 策略梯度定理（严谨的表述）

设目标函数为 $J(\boldsymbol{\theta})=\mathbb{E}_{S\sim d(\cdot)}\big[V_\pi(S)\big]$，设 $d(s)$ 为马尔科夫链稳态分布的概率密度函数。那么

$$\frac{\partial\, J(\boldsymbol{\theta})}{\partial\,\boldsymbol{\theta}} \;=\; \Big(1+\gamma+\gamma^2+\cdots+\gamma^{n-1}\Big)\cdot\mathbb{E}_{S\sim d(\cdot)}\bigg[\mathbb{E}_{A\sim\pi(\cdot|S;\boldsymbol{\theta})}\bigg[\frac{\partial\,\ln\pi(A|S;\boldsymbol{\theta})}{\partial\,\boldsymbol{\theta}}\cdot Q_\pi(S,A)\bigg]\bigg].$$

♡

证明 设初始状态 S_1 服从马尔科夫链的稳态分布，它的概率密度函数是 $d(S_1)$。对于所有的 $t=1,\cdots,n$，动作 A_t 根据策略网络抽样得到：

$$A_t \;\sim\; \pi\big(\,\cdot\,\big|\,S_t;\,\boldsymbol{\theta}\big),$$

新的状态 S_{t+1} 根据状态转移函数抽样得到：

$$S_{t+1} \;\sim\; p\big(\,\cdot\,\big|\,S_t,A_t\big).$$

对于任意函数 f，反复应用引理 7.4 可得：

$$
\begin{aligned}
& \mathbb{E}_{S_1\sim d}\Big\{\mathbb{E}_{A_1\sim\pi,S_2\sim p}\Big\{\mathbb{E}_{A_2,S_3,A_3,S_4,\cdots,A_{t-1},S_t}\Big[f\big(S_t\big)\Big]\Big\}\Big\} \\
= \;& \mathbb{E}_{S_2\sim d}\Big\{\mathbb{E}_{A_2,S_3,A_3,S_4,\cdots,A_{t-1},S_t}\Big[f\big(S_t\big)\Big]\Big\} \qquad \text{（由引理 7.4 得出）} \\
= \;& \mathbb{E}_{S_2\sim d}\Big\{\mathbb{E}_{A_2\sim\pi,S_3\sim p}\Big\{\mathbb{E}_{A_3,S_4,A_4,S_5,\cdots,A_{t-1},S_t}\Big[f\big(S_t\big)\Big]\Big\}\Big\} \\
= \;& \mathbb{E}_{S_3\sim d}\Big\{\mathbb{E}_{A_3,S_4,A_4,S_5,\cdots,A_{t-1},S_t}\Big[f\big(S_t\big)\Big]\Big\} \qquad \text{（由引理 7.4 得出）} \\
& \vdots \\
= \;& \mathbb{E}_{S_{t-1}\sim d}\Big\{\mathbb{E}_{A_{t-1}\sim\pi,S_t\sim p}\Big\{f\big(S_t\big)\Big\}\Big\} \\
= \;& \mathbb{E}_{S_t\sim d}\Big\{f\big(S_t\big)\Big\}. \qquad \text{（由引理 7.4 得出）}
\end{aligned}
$$

设 $\boldsymbol{g}(s,a;\boldsymbol{\theta}) \triangleq Q_\pi(s,a)\cdot\frac{\partial \ln \pi(a|s;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$。设一局游戏在第 n 步之后结束。由引理 7.3 与上面的公式可得：

$$
\begin{aligned}
\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \;=\;& \mathbb{E}_{S_1,A_1}\Big[\boldsymbol{g}(S_1,A_1;\boldsymbol{\theta})\Big] \\
& + \gamma\cdot\mathbb{E}_{S_1,A_1,S_2,A_2}\Big[\boldsymbol{g}(S_2,A_2;\boldsymbol{\theta})\Big] \\
& + \gamma^2\cdot\mathbb{E}_{S_1,A_1,S_2,A_2,S_3,A_3}\Big[\boldsymbol{g}(S_3,A_3;\boldsymbol{\theta})\Big] \\
& + \cdots \\
& + \gamma^{n-1}\cdot\mathbb{E}_{S_1,A_1,S_2,A_2,S_3,A_3,\cdots S_n,A_n}\Big[\boldsymbol{g}(S_n,A_n;\boldsymbol{\theta})\Big]\Big] \\
=\;& \mathbb{E}_{S_1\sim d(\cdot)}\Big\{\mathbb{E}_{A_1\sim\pi(\cdot|S_1;\boldsymbol{\theta})}\Big[\boldsymbol{g}(S_1,A_1;\boldsymbol{\theta})\Big]\Big\} \\
& + \gamma\cdot\mathbb{E}_{S_2\sim d(\cdot)}\Big\{\mathbb{E}_{A_2\sim\pi(\cdot|S_2;\boldsymbol{\theta})}\Big[\boldsymbol{g}(S_2,A_2;\boldsymbol{\theta})\Big]\Big\} \\
& + \gamma^2\cdot\mathbb{E}_{S_3\sim d(\cdot)}\Big\{\mathbb{E}_{A_3\sim\pi(\cdot|S_3;\boldsymbol{\theta})}\Big[\boldsymbol{g}(S_3,A_3;\boldsymbol{\theta})\Big]\Big\} \\
& + \cdots \\
& + \gamma^{n-1}\cdot\mathbb{E}_{S_n\sim d(\cdot)}\Big\{\mathbb{E}_{A_n\sim\pi(\cdot|S_n;\boldsymbol{\theta})}\Big[\boldsymbol{g}(S_n,A_n;\boldsymbol{\theta})\Big]\Big\} \\
=\;& \Big(1+\gamma+\gamma^2+\cdots+\gamma^{n-1}\Big)\cdot\mathbb{E}_{S\sim d(\cdot)}\Big\{\mathbb{E}_{A\sim\pi(\cdot|S;\boldsymbol{\theta})}\Big[\boldsymbol{g}(S,A;\boldsymbol{\theta})\Big]\Big\}.
\end{aligned}
$$

由此可得定理 7.5。 □

7.3.3 近似策略梯度

先复习一下前两小节的内容。策略学习可以表述为这样一个优化问题：

$$\max_{\boldsymbol{\theta}} \left\{ J(\boldsymbol{\theta}) \ \triangleq \ \mathbb{E}_S\Big[V_\pi(S)\Big] \right\}.$$

求解这个最大化问题最简单的算法就是梯度上升：

$$\boldsymbol{\theta} \ \leftarrow \ \boldsymbol{\theta} + \beta \cdot \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}).$$

其中的 $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ 是策略梯度。策略梯度定理证明：

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \ = \ \mathbb{E}_S\Big[\mathbb{E}_{A\sim\pi(\cdot|S;\boldsymbol{\theta})}\Big[Q_\pi(S,A) \ \cdot \ \nabla_{\boldsymbol{\theta}} \ln \pi\big(A \,\big|\, S;\, \boldsymbol{\theta}\big)\Big]\Big].$$

解析求出这个期望是不可能的，因为我们并不知道状态 S 概率密度函数；即使我们知道 S 的概率密度函数，能够通过连加或者定积分求出期望，我们也不愿意这样做，因为连加或者定积分的计算量非常大。

回忆一下，第 2 章介绍了期望的蒙特卡洛近似，可以将这种方法用来近似策略梯度中的期望。每次从环境中观测到一个状态 s，它相当于随机变量 S 的观测值。然后再根据当前的策略网络（策略网络的参数必须是最新的）随机抽样得出一个动作：

$$a \ \sim \ \pi\big(\ \cdot \ \big|\, s;\, \boldsymbol{\theta}\big).$$

计算随机梯度：

$$\boxed{\boldsymbol{g}(s,a;\boldsymbol{\theta}) \ \triangleq \ Q_\pi(s,a) \ \cdot \ \nabla_{\boldsymbol{\theta}} \ln \pi\big(a \,\big|\, s;\, \boldsymbol{\theta}\big).}$$

很显然，$\boldsymbol{g}(s,a;\boldsymbol{\theta})$ 是策略梯度 $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ 的无偏估计：

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \ = \ \mathbb{E}_S\Big[\mathbb{E}_{A\sim\pi(\cdot|S;\boldsymbol{\theta})}\Big[\boldsymbol{g}(S,A;\boldsymbol{\theta})\Big]\Big].$$

于是我们得到下面的结论：

结论 7.1

随机梯度 $\boldsymbol{g}(s,a;\boldsymbol{\theta}) \triangleq Q_\pi(s,a) \cdot \nabla_{\boldsymbol{\theta}} \ln \pi(a|s;\boldsymbol{\theta})$ 是策略梯度 $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ 的无偏估计。♣

应用上述结论，我们可以做随机梯度上升来更新 $\boldsymbol{\theta}$，使得目标函数 $J(\boldsymbol{\theta})$ 逐渐增长：

$$\boldsymbol{\theta} \ \leftarrow \ \boldsymbol{\theta} \ + \ \beta \cdot \boldsymbol{g}(s,a;\boldsymbol{\theta}).$$

此处的 β 是学习率，需要手动调。但是这种方法仍然不可行，我们计算不出 $\boldsymbol{g}(s,a;\boldsymbol{\theta})$，原因在于我们不知道动作价值函数 $Q_\pi(s,a)$。在后面两节中，我们用两种方法对 $Q_\pi(s,a)$ 做近似：一种方法是 REINFORCE，用实际观测的回报 u 近似 $Q_\pi(s,a)$；另一种方法是 Actor-Critic，用神经网络 $q(s,a;\boldsymbol{w})$ 近似 $Q_\pi(s,a)$。

7.4 REINFORCE

策略梯度方法用策略梯度 $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ 更新策略网络参数 $\boldsymbol{\theta}$，从而增大目标函数。上一节中，我们推导出策略梯度 $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ 的无偏估计，即下面的随机梯度：

$$\boxed{\boldsymbol{g}(s, a; \boldsymbol{\theta}) \ \triangleq \ Q_\pi(s, a) \ \cdot \ \nabla_{\boldsymbol{\theta}} \ln \pi\big(a \,\big|\, s;\, \boldsymbol{\theta}\big).}$$

但是其中的动作价值函数 Q_π 是未知的，导致无法直接计算 $\boldsymbol{g}(s, a; \boldsymbol{\theta})$。REINFORCE 进一步对 Q_π 做蒙特卡洛近似，把它替换成回报 u。REINFORCE 属于策略梯度方法。

7.4.1 REINFORCE 的简化推导

设一局游戏有 n 步，一局中的奖励记作 $R_1, \cdots, R_n$。回忆一下，t 时刻的折扣回报定义为：

$$U_t \ = \ \sum_{k=t}^{n} \gamma^{k-t} \cdot R_k.$$

而动作价值定义为 U_t 的条件期望：

$$Q_\pi(s_t, a_t) \ = \ \mathbb{E}\Big[U_t \,\Big|\, S_t = s_t, A_t = a_t\Big].$$

我们可以用蒙特卡洛近似上面的条件期望。从时刻 t 开始，智能体完成一局游戏，观测到全部奖励 $r_t, \cdots, r_n$，然后可以计算出 $u_t = \sum_{k=t}^{n} \gamma^{k-t} \cdot r_k$。因为 u_t 是随机变量 U_t 的观测值，所以 u_t 是上面公式中期望的蒙特卡洛近似。在实践中，可以用 u_t 代替 $Q_\pi(s_t, a_t)$，那么随机梯度 $\boldsymbol{g}(s_t, a_t; \boldsymbol{\theta})$ 可以近似成

$$\boxed{\tilde{\boldsymbol{g}}(s_t, a_t; \boldsymbol{\theta}) \ = \ u_t \ \cdot \ \nabla_{\boldsymbol{\theta}} \ln \pi\big(a_t \,\big|\, s_t;\, \boldsymbol{\theta}\big).}$$

$\tilde{\boldsymbol{g}}$ 是 $\boldsymbol{g}$ 的无偏估计，所以也是策略梯度 $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ 的无偏估计；$\tilde{\boldsymbol{g}}$ 也是一种随机梯度。

我们可以用反向传播计算出 $\ln \pi$ 关于 $\boldsymbol{\theta}$ 的梯度，而且可以实际观测到 u_t，于是我们可以实际计算出随机梯度 $\tilde{\boldsymbol{g}}$ 的值。有了随机梯度的值，我们可以做随机梯度上升更新策略网络参数 $\boldsymbol{\theta}$：

$$\boldsymbol{\theta} \ \leftarrow \ \boldsymbol{\theta} \ + \ \beta \cdot \tilde{\boldsymbol{g}}(s_t, a_t; \boldsymbol{\theta}). \tag{7.9}$$

根据上述推导，我们得到了训练策略网络的方法，这种方法叫做 REINFORCE。

7.4.2 训练流程

当前策略网络的参数是 $\boldsymbol{\theta}_{\text{now}}$。REINFORCE 执行下面的步骤对策略网络的参数做一次更新：

1. 用策略网络 $\boldsymbol{\theta}_{\text{now}}$ 控制智能体从头开始玩一局游戏，得到一条轨迹 (Trajectory)：

$$s_1, a_1, r_1, \quad s_2, a_2, r_2, \quad \cdots, \quad s_n, a_n, r_n.$$

2. 计算所有的回报：

$$u_t \ = \ \sum_{k=t}^{n} \gamma^{k-t} \cdot r_k, \qquad \forall\, t = 1, \cdots, n.$$

3. 用 $\{(s_t, a_t)\}_{t=1}^{n}$ 作为数据，做反向传播计算：

$$\nabla_{\boldsymbol{\theta}} \ln \pi\big(a_t \,\big|\, s_t;\, \boldsymbol{\theta}_{\text{now}}\big), \qquad \forall\, t = 1, \cdots, n.$$

4. 做随机梯度上升更新策略网络参数：

$$\boldsymbol{\theta}_{\text{new}} \;\leftarrow\; \boldsymbol{\theta}_{\text{now}} \;+\; \beta \,\cdot\, \sum_{t=1}^{n} \gamma^{t-1} \,\cdot\, \underbrace{u_t \,\cdot\, \nabla_{\boldsymbol{\theta}} \ln \pi\big(a_t \,\big|\, s_t;\, \boldsymbol{\theta}_{\text{now}}\big)}_{\text{即随机梯度 } \tilde{\boldsymbol{g}}(s_t, a_t; \boldsymbol{\theta}_{\text{now}})}.$$

注 在算法最后一步中，随机梯度前面乘以系数 γ^{t-1}。读者可能会好奇，为什么需要这个系数呢？原因是这样的：前面 REINFORCE 的推导是简化的，而非严谨的数学推导；按照我们简化的推导，不应该乘以系数 γ^{t-1}。下一小节做严格的数学推导，得出的 REINFORCE 算法需要系数 γ^{t-1}。读者只要知道这个事实就行了，不必读懂下一小节的数学推导。

注 REINFORCE 是一种**同策略** (On-Policy) 方法，要求行为策略 (Behavior Policy) 与目标策略 (Target Policy) 相同，两者都必须是策略网络 $\pi(a|s;\boldsymbol{\theta}_{\text{now}})$，其中 $\boldsymbol{\theta}_{\text{now}}$ 是策略网络当前的参数。所以经验回放不适用于 REINFORCE。

7.4.3 REINFORCE 严格的推导

第 7.4.1 小节对策略梯度做近似，推导出 REINFORCE 方法。那种推导是简化过的，帮助读者理解 REINFORCE 算法，但实际上那种推导并不够严谨。本小节做严格的数学推导，对策略梯度做近似，得出真正的 REINFORCE 方法。建议对数学证明不感兴趣的读者跳过本小节。

根据定义，$\boldsymbol{g}(s, a; \boldsymbol{\theta}) \triangleq Q_\pi\big(s, a\big) \cdot \nabla_{\boldsymbol{\theta}} \ln \pi\big(a \,\big|\, s;\, \boldsymbol{\theta}\big)$。引理 7.3 把策略梯度 $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ 表示成期望的连加：

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \;\; = \;\; & \mathbb{E}_{S_1, A_1}\Big[\boldsymbol{g}(S_1, A_1; \boldsymbol{\theta})\Big] \\
& + \gamma \cdot \mathbb{E}_{S_1, A_1, S_2, A_2}\Big[\boldsymbol{g}(S_2, A_2; \boldsymbol{\theta})\Big] \\
& + \gamma^2 \cdot \mathbb{E}_{S_1, A_1, S_2, A_2, S_3, A_3}\Big[\boldsymbol{g}(S_3, A_3; \boldsymbol{\theta})\Big] \\
& + \cdots \\
& + \gamma^{n-1} \cdot \mathbb{E}_{S_1, A_1, S_2, A_2, S_3, A_3, \cdots, S_n, A_n}\Big[\boldsymbol{g}(S_n, A_n; \boldsymbol{\theta})\Big]\Big].
\end{aligned} \tag{7.10}$$

我们可以对期望做蒙特卡洛近似。首先观测到第一个状态 $S_1 = s_1$。然后用最新的策略网络 $\pi(a|s;\boldsymbol{\theta}_{\text{now}})$ 控制智能体与环境交互，观测到到轨迹

$$s_1, a_1, r_1, \;\; s_2, a_2, r_2, \;\; \cdots, \;\; s_n, a_n, r_n.$$

对公式 (7.10) 中的期望做蒙特卡洛近似，得到：

$$\nabla_{\boldsymbol{\theta}} J\big(\boldsymbol{\theta}_{\text{now}}\big) \;\; \approx \;\; \boldsymbol{g}\big(s_1, a_1; \boldsymbol{\theta}_{\text{now}}\big) + \gamma \cdot \boldsymbol{g}\big(s_2, a_2; \boldsymbol{\theta}_{\text{now}}\big) + \cdots + \gamma^{n-1} \cdot \boldsymbol{g}\big(s_n, a_n; \boldsymbol{\theta}_{\text{now}}\big).$$

进一步把 $\boldsymbol{g}\big(s_t, a_t; \boldsymbol{\theta}_{\text{now}}\big) \triangleq Q_\pi\big(s_t, a_t\big) \cdot \nabla_{\boldsymbol{\theta}} \ln \pi\big(a_t \,\big|\, s_t;\, \boldsymbol{\theta}_{\text{now}}\big)$ 中的 $Q_\pi(s_t, a_t)$ 替换成 u_t，那么 $\boldsymbol{g}(s_t, a_t; \boldsymbol{\theta}_{\text{now}})$ 就被近似成为

$$\boldsymbol{g}(s_t, a_t; \boldsymbol{\theta}_{\text{now}}) \;\;\approx\;\; u_t \,\cdot\, \nabla_{\boldsymbol{\theta}} \,\ln \pi\big(a_t \,\big|\, s_t;\, \boldsymbol{\theta}_{\text{now}}\big).$$

经过上述两次近似，策略梯度被近似成为下面的随机梯度

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_{\text{now}}) \approx \sum_{t=1}^{n} \gamma^{t-1} \cdot u_t \cdot \nabla_{\boldsymbol{\theta}} \ln \pi(a_t|s_t;\boldsymbol{\theta}_{\text{now}}).$$

这样就得到了 REINFORCE 算法的随机梯度上升公式：

$$\boldsymbol{\theta}_{\text{new}} \leftarrow \boldsymbol{\theta}_{\text{now}} + \beta \cdot \sum_{t=1}^{n} \gamma^{t-1} \cdot u_t \cdot \nabla_{\boldsymbol{\theta}} \ln \pi\big(a_t \,\big|\, s_t;\, \boldsymbol{\theta}_{\text{now}}\big).$$

7.5 Actor-Critic

策略梯度方法用策略梯度 $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ 更新策略网络参数 $\boldsymbol{\theta}$，从而增大目标函数。第 7.2 节推导出策略梯度 $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ 的无偏估计，即下面的随机梯度：

$$\boxed{\boldsymbol{g}(s,a;\boldsymbol{\theta}) \ \triangleq \ Q_\pi\big(s,a\big) \ \cdot \ \nabla_{\boldsymbol{\theta}} \ln \pi\big(a \,\big|\, s;\, \boldsymbol{\theta}\big).}$$

但是其中的动作价值函数 Q_π 是未知的，导致无法直接计算 $\boldsymbol{g}(s,a;\boldsymbol{\theta})$。上一节的 REINFORCE 用实际观测的回报近似 Q_π，本节的 Actor-Critic 方法用神经网络近似 Q_π。

7.5.1 价值网络

Actor-Critic 方法中用一个神经网络近似动作价值函数 $Q_\pi(s,a)$，这个神经网络叫做“价值网络”，记为 $q(s,a;\boldsymbol{w})$，其中的 $\boldsymbol{w}$ 表示神经网络中可训练的参数。价值网络的输入是状态 s，输出是每个动作的价值。动作空间 $\mathcal{A}$ 中有多少种动作，那么价值网络的输出就是多少维的向量，向量每个元素对应一个动作。举个例子，动作空间是 $\mathcal{A}=\{\text{左},\text{右},\text{上}\}$，价值网络的输出是

$$\begin{aligned} q\big(s,\, \text{左};\, \boldsymbol{w}\big) &= 219, \\ q\big(s,\, \text{右};\, \boldsymbol{w}\big) &= -73, \\ q\big(s,\, \text{上};\, \boldsymbol{w}\big) &= 580. \end{aligned}$$

神经网络的结构见图 7.2。

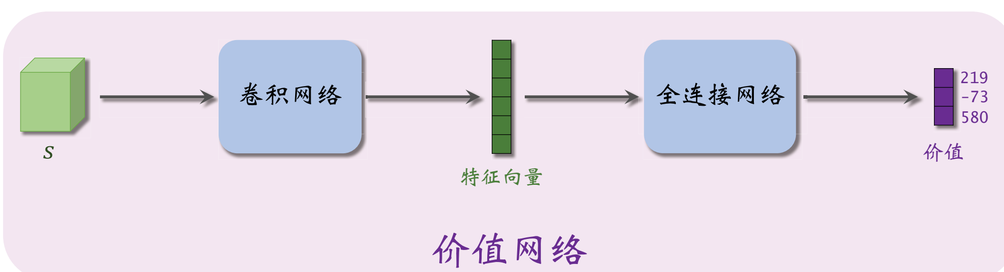

图 7.2: 价值网络 $q(s,a;\boldsymbol{w})$ 的结构。输入是状态 s；输出是每个动作的价值。

虽然价值网络 $q(s,a;\boldsymbol{w})$ 与之前学的 DQN 有相同的结构，但是两者的意义不同，训练算法也不同。

- 价值网络是对动作价值函数 $Q_\pi(s,a)$ 的近似。而 DQN 则是对最优动作价值函数 $Q_\star(s,a)$ 的近似。
- 对价值网络的训练使用的是 SARSA 算法，它属于同策略，不能用经验回放。对 DQN 的训练使用的是 Q 学习算法，它属于异策略，可以用经验回放。

7.5.2 算法的推导

Actor-Critic 翻译成“演员—评委”方法。策略网络 $\pi(a|s;\boldsymbol{\theta})$ 相当于演员，它基于状态 s 做出动作 a。价值网络 $q(s,a;\boldsymbol{\theta})$ 相当于评委，它给演员的表现打分，量化在状态 s 的情况下做出动作 a 的好坏程度。策略网络（演员）和价值网络（评委）的关系如图 7.3 所示。

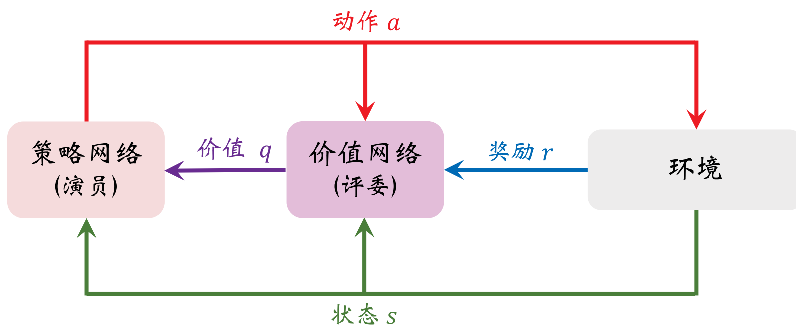

图 7.3: Actor-Critic 方法中策略网络（演员）和价值网络（评委）的关系图。

读者可能会对图 7.3 感到不解：为什么不直接把奖励 R 反馈给策略网络（演员），而要用价值网络（评委）这样一个中介呢？原因是这样的。策略学习的目标函数 $J(\boldsymbol{\theta})$ 是回报 U 的期望，而不是奖励 R 的期望；注意回报 U 和奖励 R 的区别。虽然能观测到当前的奖励 R，但是它对价值网络是毫无意义的；训练策略网络（演员）需要的是回报 U，也就是未来所有奖励的加权和。价值网络（评委）能够估算出回报 U 的期望，因此能帮助训练策略网络（演员）。

训练策略网络（演员）： 策略网络（演员）想要改进自己的演技，但是演员自己不知道什么样的表演才算更好，所以需要价值网络（评委）的帮助。在演员做出动作 a 之后，评委会打一个分数 $\widehat{q} \triangleq q(s,a;\boldsymbol{w})$，并把分数反馈给演员，帮助演员做出改进。演员利用当前状态 s，自己的动作 a，以及评委的打分 $\widehat{q}$，计算近似策略梯度，然后更新自己的参数 $\boldsymbol{\theta}$（相当于改变自己的技术）。通过这种方式，演员的表现越来越受评委的好评，于是演员的获得的评分 $\widehat{q}$ 越来越高。

训练策略网络的基本想法是用策略梯度 $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ 的近似来更新参数 $\boldsymbol{\theta}$。之前我们推导过策略梯度的无偏估计：

$$\boldsymbol{g}(s,a;\boldsymbol{\theta}) \quad \triangleq \quad Q_\pi\big(s,a\big) \ \cdot\ \nabla_{\boldsymbol{\theta}} \ln \pi\big(a \,\big|\, s;\ \boldsymbol{\theta}\big).$$

价值网络 $q(s,a;\boldsymbol{w})$ 是对动作价值函数 $Q_\pi(s,a)$ 的近似，所以把上面公式中的 Q_π 替换成价值网络，得到近似策略梯度：

$$\boxed{\widehat{\boldsymbol{g}}\big(s,a;\ \boldsymbol{\theta}\big) \quad \triangleq \quad \underbrace{q\big(s,a;\ \boldsymbol{w}\big)}_{\text{评委的打分}} \cdot \nabla_{\boldsymbol{\theta}} \ln \pi\big(a \,\big|\, s;\ \boldsymbol{\theta}\big).} \tag{7.11}$$

最后做梯度上升更新策略网络的参数：

$$\boxed{\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \beta \cdot \widehat{\boldsymbol{g}}(s, a; \boldsymbol{\theta}).} \tag{7.12}$$

注 用上述方式更新参数之后，会让评委打出的分数越来越高，原因是这样的。状态价值函数 $V_\pi(s)$ 可以近似成为：

$$v(s; \boldsymbol{\theta}) = \mathbb{E}_{A \sim \pi(\cdot|s; \boldsymbol{\theta})}\Big[q(s, A; \boldsymbol{w})\Big].$$

因此可以将 $v(s;\boldsymbol{\theta})$ 看做评委打分的均值。不难证明，公式 (7.11) 中定义的近似策略梯度 $\widehat{\boldsymbol{g}}(s, a; \boldsymbol{\theta})$ 的期望等于 $v(s;\boldsymbol{\theta})$ 关于 $\boldsymbol{\theta}$ 的梯度；

$$\nabla_{\boldsymbol{\theta}} v(s; \boldsymbol{\theta}) = \mathbb{E}_{A \sim \pi(\cdot|s; \boldsymbol{\theta})}\Big[\widehat{\boldsymbol{g}}(s, A; \boldsymbol{\theta})\Big].$$

因此，用公式 7.12 中的梯度上升更新 $\boldsymbol{\theta}$，会让 $v(s;\boldsymbol{\theta})$ 变大，也就是让评委打分的均值更高。

训练价值网络（评委）： 通过以上分析，我们不难发现上述训练策略网络（演员）的方法不是真正让演员表现更好，只是让演员更迎合评委的喜好而已。因此，评委的水平也很重要，只有当评委的打分 $\widehat{q}$ 真正反映出动作价值 Q_π，演员的水平才能真正提高。初始的时候，价值网络的参数 $\boldsymbol{w}$ 是随机的，也就是说评委的打分是瞎猜。可以用 SARSA 算法更新 $\boldsymbol{w}$，提高评委的水平。每次从环境中观测到一个奖励 r，把 r 看做是真相，用 r 来校准评委的打分。

第 5.1 节已经推导过 SARSA 算法，现在我们再回顾一下。在 t 时刻，价值网络输出

$$\widehat{q}_t = q(s_t, a_t; \boldsymbol{w}),$$

它是对动作价值函数 $Q_\pi(s_t, a_t)$ 的估计。在 $t+1$ 时刻，实际观测到 r_t, s_{t+1}, a_{t+1}，于是可以计算 TD 目标

$$\widehat{y}_t \triangleq r_t + \gamma \cdot q(s_{t+1}, a_{t+1}; \boldsymbol{w}),$$

它也是对动作价值函数 $Q_\pi(s_t, a_t)$ 的估计。由于 $\widehat{y}_t$ 部分基于实际观测到的奖励 r_t，我们认为 $\widehat{y}_t$ 比 $q(s_t, a_t; \boldsymbol{w})$ 更接近事实真相。所以把 $\widehat{y}_t$ 固定住，鼓励 $q(s_t, a_t; \boldsymbol{w})$ 去接近 $\widehat{y}_t$。SARSA 算法具体这样更新价值网络参数 $\boldsymbol{w}$。定义损失函数：

$$L(\boldsymbol{w}) \triangleq \frac{1}{2}\Big[q(s_t, a_t; \boldsymbol{w}) - \widehat{y}_t\Big]^2,$$

设 $\widehat{q}_t \triangleq q(s_t, a_t; \boldsymbol{w})$。损失函数的梯度是：

$$\nabla_{\boldsymbol{w}} L(\boldsymbol{w}) = \underbrace{(\widehat{q}_t - \widehat{y}_t)}_{\text{TD 误差 } \delta_t} \cdot \nabla_{\boldsymbol{w}}\, q(s_t, a_t; \boldsymbol{w}).$$

做一轮梯度下降更新 $\boldsymbol{w}$：

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha \cdot \nabla_{\boldsymbol{w}} L(\boldsymbol{w}).$$

这样更新 $\boldsymbol{w}$ 可以让 $q(s_t, a_t; \boldsymbol{w})$ 更接近 $\widehat{y}_t$。可以这样理解 SARSA：用观测到的奖励 r_t 来“校准”评委的打分 $q(s_t, a_t; \boldsymbol{w})$。

7.5.3 训练流程

最后概括 Actor-Critic 训练流程。设当前策略网络参数是 $\boldsymbol{\theta}_{\text{now}}$，价值网络参数是 $\boldsymbol{w}_{\text{now}}$。执行下面的步骤，将参数更新成 $\boldsymbol{\theta}_{\text{new}}$ 和 $\boldsymbol{w}_{\text{new}}$：

1. 观测到当前状态 s_t，根据策略网络做决策：$a_t \sim \pi(\cdot \,|\, s_t;\, \boldsymbol{\theta}_{\text{now}})$，并让智能体执行动作 a_t。
2. 从环境中观测到奖励 r_t 和新的状态 s_{t+1}。
3. 根据策略网络做决策：$\tilde{a}_{t+1} \sim \pi(\cdot|s_{t+1};\boldsymbol{\theta}_{\text{now}})$，但不让智能体执行动作 $\tilde{a}_{t+1}$。
4. 让价值网络打分：
$$\widehat{q}_t \;=\; q\big(s_t, a_t;\, \boldsymbol{w}_{\text{now}}\big) \qquad \text{和} \qquad \widehat{q}_{t+1} \;=\; q\big(s_{t+1}, \tilde{a}_{t+1};\, \boldsymbol{w}_{\text{now}}\big)$$
5. 计算 TD 目标和 TD 误差：
$$\widehat{y}_t \;=\; r_t + \gamma \cdot \widehat{q}_{t+1} \qquad \text{和} \qquad \delta_t \;=\; \widehat{q}_t - \widehat{y}_t .$$
6. 更新价值网络：
$$\boldsymbol{w}_{\text{new}} \;\leftarrow\; \boldsymbol{w}_{\text{now}} \;-\; \alpha \cdot \delta_t \cdot \nabla_{\boldsymbol{w}} q\big(s_t, a_t;\, \boldsymbol{w}_{\text{now}}\big).$$
7. 更新策略网络：
$$\boldsymbol{\theta}_{\text{new}} \;\leftarrow\; \boldsymbol{\theta}_{\text{now}} \;+\; \beta \cdot \widehat{q}_t \cdot \nabla_{\boldsymbol{\theta}} \ln \pi\big(a_t \,\big|\, s_t;\, \boldsymbol{\theta}_{\text{now}}\big).$$

7.5.4 用目标网络改进训练

第 6.2 节讨论了 Q 学习中的自举及其危害，以及用目标网络 (Target Network) 缓解自举造成的偏差。SARSA 算法中也存在自举——即用价值网络自己的估值 $\widehat{q}_{t+1}$ 去更新价值网络自己；我们同样可以用目标网络计算 TD 目标，从而缓解偏差。把目标网络记作 $q(s, a; \boldsymbol{w}^{-})$，它的结构与价值网络的结构相同，但是参数不同。使用目标网络计算 TD 目标，那么 Actor-Critic 的训练就变成了：

1. 观测到当前状态 s_t，根据策略网络做决策：$a_t \sim \pi(\cdot \,|\, s_t;\, \boldsymbol{\theta}_{\text{now}})$，并让智能体执行动作 a_t。
2. 从环境中观测到奖励 r_t 和新的状态 s_{t+1}。
3. 根据策略网络做决策：$\tilde{a}_{t+1} \sim \pi(\cdot \,|\, s_{t+1};\, \boldsymbol{\theta}_{\text{now}})$，但是不让智能体执行动作 $\tilde{a}_{t+1}$。
4. 让价值网络给 (s_t, a_t) 打分：
$$\widehat{q}_t \;=\; q\big(s_t, a_t;\, \boldsymbol{w}_{\text{now}}\big).$$
5. 让目标网络给 $(s_{t+1}, \tilde{a}_{t+1})$ 打分：
$$\widehat{q}_{t+1}^{-} \;=\; q\big(s_{t+1}, \tilde{a}_{t+1};\, \boldsymbol{w}_{\text{now}}^{-}\big).$$
6. 计算 TD 目标和 TD 误差：
$$\widehat{y}_t^{-} \;=\; r_t + \gamma \cdot \widehat{q}_{t+1}^{-} \qquad \text{和} \qquad \delta_t \;=\; \widehat{q}_t - \widehat{y}_t^{-} .$$
7. 更新价值网络：
$$\boldsymbol{w}_{\text{new}} \;\leftarrow\; \boldsymbol{w}_{\text{now}} \;-\; \alpha \cdot \delta_t \cdot \nabla_{\boldsymbol{w}} q\big(s_t, a_t;\, \boldsymbol{w}_{\text{now}}\big).$$

8. 更新策略网络：

$$\boldsymbol{\theta}_{\text{new}} \leftarrow \boldsymbol{\theta}_{\text{now}} + \beta \cdot \widehat{q}_t \cdot \nabla_{\boldsymbol{\theta}} \ln \pi\big(a_t \,\big|\, s_t; \boldsymbol{\theta}_{\text{now}}\big).$$

9. 设 $\tau \in (0,1)$ 是需要手动调的超参数。做加权平均更新目标网络的参数：

$$\boldsymbol{w}_{\text{new}}^{-} \leftarrow \tau \cdot \boldsymbol{w}_{\text{new}} + \big(1-\tau\big) \cdot \boldsymbol{w}_{\text{now}}^{-}.$$

第七章 相关文献

REINFORCE 方法由 Williams 在 1897 年提出 [118-119]。Actor-Critic 方法在 Barto 等人 1983 年的论文 [10] 中提出。很多论文分析过 Actor-Critic 方法的收敛，比如 [56, 14, 2, 15, 122]。策略梯度定理由 Marbach 和 Tsitsiklis 1999 年的论文 [67] 和 Sutton 等人 2000 年的论文 [98] 独立提出。