Policy gradient theorem的证明

如今，强化学习基本都采用参数化的神经网络来学习一个策略，而神经网络一般是通过梯度下降法或者各种变种来优化的，因此，获取累积回报关于策略的梯度至关重要。本节会给大家推导策略梯度的表达式，并介绍实际训练中是如何采样近似该表达式的。

这里我们首先额外引入一下动作价值函数的定义

$$Q\left(s_t, a_t\right)=E_{s_{t+1}, a_{t+1}, \ldots}\left[\sum_{l=0}^{\infty} \gamma^l r^{t+l}\right]$$

即在状态 s_t 下采用动作 a_t 后，后续动作服从策略 π 的情况下的累积期望回报，其中 $\gamma \in (0,1)$ 是折扣因子。

接着，我们将策略梯度计算过程详细展开如下：

$$\begin{aligned}
&\nabla_\theta J(\theta) \\
&=\nabla_\theta V\left(s_0\right) \\
&=\nabla\left[\sum_{a_0} \pi\left(a_0 \mid s_0\right) Q_\pi\left(s_0, a_0\right)\right] \\
&=\sum_{a_0}\left[\nabla \pi\left(a_0 \mid s_0\right) Q_\pi\left(s_0, a_0\right)+\pi\left(a_0 \mid s_0\right) \nabla Q_\pi\left(s_0, a_0\right)\right] \\
&=\sum_{a_0}\left[\nabla \pi\left(a_0 \mid s_0\right) Q_\pi\left(s_0, a_0\right)+\pi\left(a_0 \mid s_0\right) \nabla \sum_{s_1, r_1} p\left(s_1, r_1 \mid s_0, a_0\right)\left(r_1+\gamma V\left(s_1\right)\right)\right] \\
&=\sum_{a_0} \nabla \pi\left(a_0 \mid s_0\right) Q_\pi\left(s_0, a_0\right)+\sum_{a_0} \pi\left(a_0 \mid s_0\right) \sum_{s_1} p\left(s_1 \mid s_0, a_0\right) \cdot \gamma \nabla V\left(s_1\right) \\
&=\sum_{a_0} \nabla \pi\left(a_0 \mid s_0\right) Q_\pi\left(s_0, a_0\right) \\
&+\sum_{a_0} \pi\left(a_0 \mid s_0\right) \sum_{s_1} p\left(s_1 \mid s_0, a_0\right) \cdot \gamma \sum_{a_1} \nabla \pi\left(a_1 \mid s_1\right) Q_\pi\left(s_1, a_1\right) \\
&+\sum_{a_0} \pi\left(a_0 \mid s_0\right) \sum_{s_1} p\left(s_1 \mid s_0, a_0\right) \cdot \gamma \sum_{a_1} \pi\left(a_1 \mid s_1\right) \sum_{s_2} p\left(s_2 \mid s_1, a_1\right) \gamma \nabla V\left(s_2\right) \\
&=\sum_{a_0} \nabla \pi\left(a_0 \mid s_0\right) Q_\pi\left(s_0, a_0\right) \\
&+\sum_{a_0} \pi\left(a_0 \mid s_0\right) \sum_{s_1} p\left(s_1 \mid s_0, a_0\right) \cdot \gamma \sum_{a_1} \nabla \pi\left(a_1 \mid s_1\right) Q_\pi\left(s_1, a_1\right)+\cdots \\
&=\sum_{s_0} \operatorname{Pr}\left(s_0 \rightarrow s_0, 0, \pi\right) \sum_{a_0} \nabla \pi\left(a_0 \mid s_0\right) \gamma^0 Q_\pi\left(s_0, a_0\right) \\
&+\sum_{s_1} \operatorname{Pr}\left(s_0 \rightarrow s_1, 1, \pi\right) \sum_{a_1} \nabla \pi\left(a_1 \mid s_1\right) \gamma^1 Q_\pi\left(s_1, a_1\right)+\cdots
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{s_0} Pr\ (s_0 \to s_0, 0, \pi) \sum_{a_0} \pi(a_0 \mid s_0) \left[\gamma^0 Q_\pi(s_0, a_0) \nabla \log \pi(a_0 \mid s_0)\right] \\
&+ \sum_{s_1} Pr\ (s_0 \to s_1, 1, \pi) \sum_{a_1} \pi(a_1 \mid s_1) \left[\gamma^1 Q_\pi(s_1, a_1) \nabla \log \pi(a_1 \mid s_1)\right] + \cdots \\
&= \sum_{t=0}^{\infty} \sum_{s_t} Pr\ (s_0 \to s_t, t, \pi) \sum_{a_t} \pi(a_t \mid s_t) \left[\gamma^t Q_\pi(s_t, a_t) \nabla \log \pi(a_t \mid s_t)\right]
\end{aligned}
$$

其中 $Pr\ (s_0 \to s_t, t, \pi)$ 代表：从状态 s_0 出发，且按照策略 π 与环境交互（rollout），在 t 时刻到达状态 s_t 的概率。

通过上述的推导，我们就得到了**无限长时间步**下的策略梯度的表达式，对于**有限长时间步**的环境，我们可以做一个简单的转化，把它变成无限长，从而同样适用上述公式。假设时间步长度为 T ，对于所有可能出现在最后一步的状态 s_{T-1} ，我们定义：

1. 从 s_{T-1} 出发，不论采取什么动作，一定会跳转到一个虚拟的吸收态 s_T ，并返回奖励值0。
2. 从 s_T 出发，不论采取什么动作，一定会跳转回这个虚拟的吸收态 s_T ，并返回奖励值0。

由此将有限长的时间步扩展到了无限长，因为环境会陷入到 s_T 的死循环中。

不过，上式实际上很难优化，要求遍历整个状态空间和时间步空间。具体来说，该式要求计算每个时间步上到达每个状态的概率。一方面，这在**计算成本上是无法容忍**的；另一方面，我们在绝大多数情况下，**无法获得环境的转移概率**，因此无法计算特定时间步下整个状态空间上的概率分布。

那怎么办，我们可以用 Monte Carlo 方法，通过采样来逼近上面的策略梯度公式。这里先把上式转化为期望的形式：

$$
\begin{aligned}
\text{上式} &= \sum_{t=0}^{\infty} \sum_{s_t} Pr\ (s_0 \to s_t, t, \pi) \sum_{a_t} \pi(a_t \mid s_t) \left[\gamma^t Q_\pi(s_t, a_t) \nabla \log \pi(a_t \mid s_t)\right] \\
&= \sum_{t=0}^{\infty} E_{s_t} \sum_{a_t} \pi(a_t \mid s_t) \left[\gamma^t Q_\pi(s_t, a_t) \nabla \log \pi(a_t \mid s_t)\right] \\
&= \sum_{t=0}^{\infty} E_{s_t} E_{a_t} \left[\gamma^t Q_\pi(s_t, a_t) \nabla \log \pi(a_t \mid s_t)\right] \\
&= \sum_{t=0}^{\infty} E_{s_t, a_t} \left[\gamma^t Q_\pi(s_t, a_t) \nabla \log \pi(a_t \mid s_t)\right] \\
&= E_{s_0, a_0, s_1, a_1, \cdots} \sum_{t=0}^{\infty} \left[\gamma^t Q_\pi(s_t, a_t) \nabla \log \pi(a_t \mid s_t)\right] \\
&= E_\tau \sum_{t=0}^{\infty} \left[\gamma^t Q_\pi(s_t, a_t) \nabla \log \pi(a_t \mid s_t)\right]
\end{aligned}
$$

其中 $\tau = [s_0,\ a_0,\ s_1, a_1,\ \cdots]$ 是按照策略 π rollout 出来的状态动作的轨迹。可以看出，将 $\gamma^t Q_\pi(s_t, a_t) \nabla \log \pi(a_t \mid s_t)$ 这一项，先在时间步 t 上求和，再关于轨迹 τ 取期望，就得到了策略

梯度。至此，Monte Carlo方法就可以很简单地结合进来，我们先将将 E_τ 替换为采样 N 条轨迹 $\left[\tau^1, \cdots, \tau^N\right]$ 。并定义其中第 n 条轨迹为 $\tau^n=\left\langle s_0^n, a_0^n, r_0^n, \cdots, s_{T_n-1}^n, a_{T_n-1}^n, r_{T_n-1}^n\right\rangle$ ，轨迹长度为 T_n 。最后对结果取平均：

$$\begin{aligned} & E_\tau \sum_{t=0}^{\infty}\left[\gamma^t Q_\pi\left(s_t, a_t\right) \nabla \log \pi\left(a_t \mid s_t\right)\right] \\ = & \frac{1}{N} \sum_{n=1}^N \sum_{t=0}^{T_n-1}\left[\gamma^t Q_\pi\left(s_t^n, a_t^n\right) \nabla \log \pi\left(a_t^n \mid s_t^n\right)\right] \end{aligned}$$

注意到 $Q_\pi\left(s_t^n, a_t^n\right)=E_{s_{t+1}^n, a_{t+1}^n, s_{t+2}^n, a_{t+2}^n, \cdots \mid s_t^n, a_t^n}\left[\sum_{l=t}^{T_n-1} \gamma^l\ r_l^n\right]$ ，因此从期望角度二者也是可以替换的。

当我们的算法没有显式地估计 $Q_\pi\left(s_t^n, a_t^n\right)$ 时，可以定义 $G_t\left(\tau^n\right)=\sum_{l=t}^{T_n-1} \gamma^l\ r_l^n$ （即最朴素的策略梯度），并用它替换 $Q_\pi\left(s_t^n, a_t^n\right)$ ，另外再将式中的 γ^t 省略掉（是一种更简便的近似），就得到了实际使用的策略梯度公式：

$$\frac{1}{N} \sum_{n=1}^N \sum_{t=0}^{T_n-1}\left[G_t\left(\tau^n\right) \nabla \log \pi\left(a_t^n \mid s_t^n\right)\right]$$

P.S. 策略梯度相关的代码示例可以参考我们的“算法-代码”注解文档