策略梯度（Policy Optimization）

一、策略梯度算法

- 策略一般记作 π
  - 假设我们使用 NN 网路做强化学习，那么策略就是一个网络，网络中有一组参数，用 θ 来代表 π 的参数
-

在一场游戏里面，我们把环境输出的 s 与演员输出的动作 a 全部组合起来，就是一个轨迹，即

$$\tau = \{s_1, a_1, s_2, a_2, \cdots, s_t, a_t\} \tag{4.1}$$

给定演员的参数 θ，我们可以计算某个轨迹 τ 发生的概率为

$$\begin{aligned} p_\theta(\tau) &= p(s_1)\, p_\theta(a_1|s_1)\, p(s_2|s_1, a_1)\, p_\theta(a_2|s_2)\, p(s_3|s_2, a_2) \cdots \\ &= p(s_1) \prod_{t=1}^{T} p_\theta(a_t|s_t)\, p(s_{t+1}|s_t, a_t) \end{aligned} \tag{4.2}$$

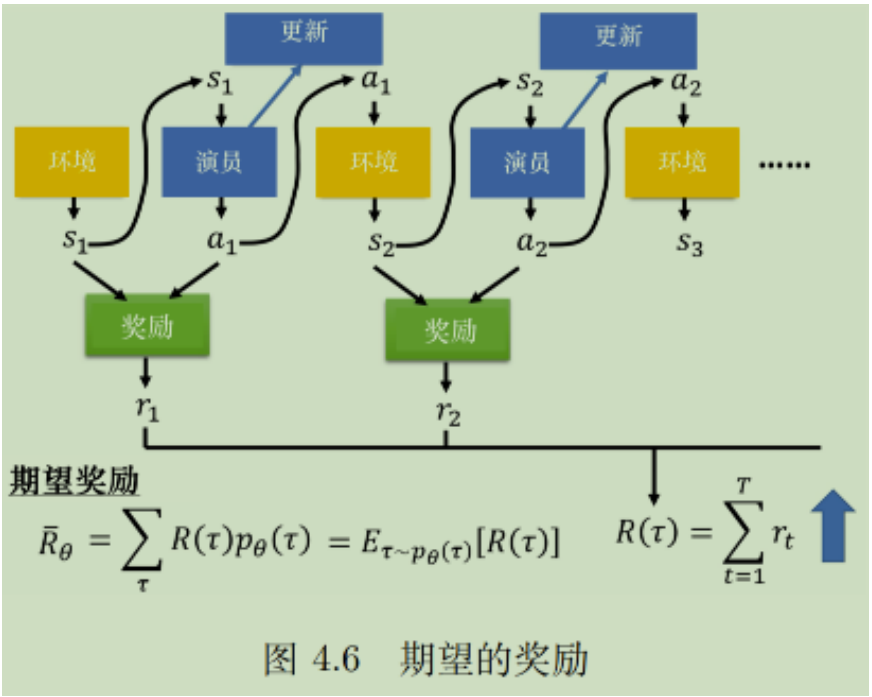
- 在强化学习里面，除了环境与演员以外，还有奖励函数。如图 4.6 所示，奖励函数根据在某一个状态采取的某一个动作决定这个动作可以得到的分数。对奖励函数输入 s1、a1，它会输出 r1；输入 s2、a2，奖励函数会输出 r2。我们把轨迹所有的奖励 r 都加起来，就得到了 R(τ)，其代表某一个轨迹 τ 的奖励
  - 

    图 4.6　期望的奖励
- 我们推导出来奖励的期望之后，就可以采用梯度上升的方法，获得最优策略参数，使得期望最大化

- 
  -
  -
- 过程描述
  - 用参数为 θ 的智能体与环境交互，也就是拿已经训练好的智能体先与环境交互，交互完以后，就可以得到大量游戏的数据
  - 把利用采样数据计算梯度
  - 更新模型
  - 丢掉数据
  - 重新采样
  - 循环上面的过程

二、策略梯度实现技巧

- 添加基线
  - 基准奖励不能总是非负的
  - $$\nabla \bar{R}_\theta \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \left(R\left(\tau^n\right)-b\right) \nabla \log p_\theta\left(a_t^n \mid s_t^n\right)$$
- 动作配分

- (s, a) pair 对不能永远是一样的权重，在不同的环境中，应该变化
- $$\nabla \bar{R}_\theta \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \left(\sum_{t'=t}^{T_n} r_{t'}^n - b \right) \nabla \log p_\theta \left(a_t^n \mid s_t^n \right)$$