第八讲：策略梯度算法

主讲人 陈达贵

清华大学自动化系

在读硕士

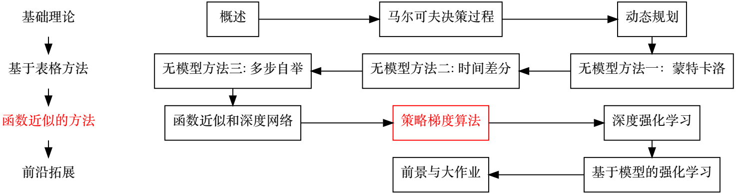
深蓝学院

shenlanxueyuan.com

强化学习理论与实践

目录

章节目录


基础理论
基于表格方法
函数近似的方法
前沿拓展
概述
马尔可夫决策过程
动态规划
无模型方法一：蒙特卡洛
无模型方法二: 时间差分
无模型方法三: 多步自举
函数近似和深度网络
策略梯度算法
深度强化学习
基于模型的强化学习
前景与大作业

本章目录

基于策略的强化学习

- 在过去的课程中我们讲述了基于值函数的方法
- 上一节中，使用了带参数 w 的函数去近似值函数

$$V_w(s) \approx V^\pi(s)$$
$$Q_w(s,a) \approx Q^\pi(s,a)$$

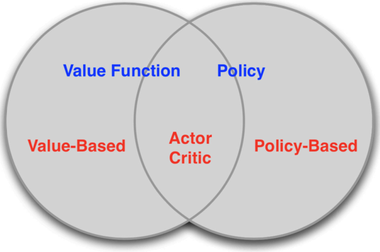
- 策略是从值函数中导出的
  - 使用贪婪的方法导出最优策略
  - 使用 ε 贪婪的方法导出行为策略
- 我们直接参数化**策略**

$$\pi_\theta(a|s) = \mathbb{P}[a|s,\theta]$$

- 这里仍然考虑无模型的方法

强化学习分类

- 基于值函数的方法
  - 学习值函数
  - 用值函数导出策略
- 基于策略的方法
  - 没有值函数
  - 学习策略
- Actor-Critic
  - 学习值函数
  - 学习策略

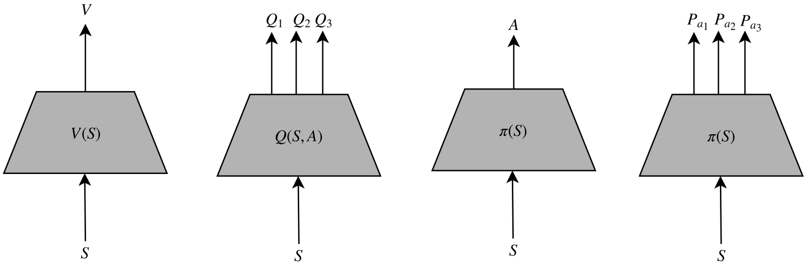


为什么要使用策略梯度算法?

基于值函数方法的局限性

- 针对确定性策略
- 策略退化
- 难以处理高维度的状态/动作空间
  - 不能处理连续的状态/动作空间
- 收敛速度慢

策略模型的建模方式



策略梯度算法的优缺点

优点

- 更好的收敛性
- 能够有效地处理高维和连续的动作空间
- 能够学到随机策略
- 不会导致策略退化

缺点

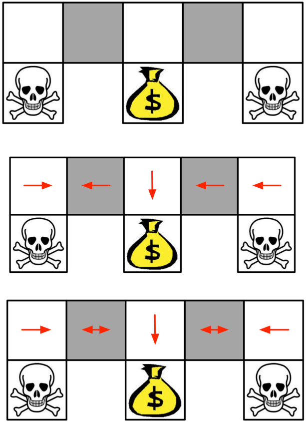
- 更容易收敛到局部最优值
- 难以评价一个策略，而且评价的方差较大

随机策略

石头剪刀布

- 两个人玩“石头剪刀布”
- 如果是一个确定性策略
  - 则很容易输掉游戏
- 一个均匀分布的随机策略才是最优的 (满足纳什均衡)

随机策略



- 假设灰色区域是部分观测的
- 因此两个灰色区域是等价的
- 确定性策略会导致两个灰色区域有相同的动作
- 即便使用 ε 的贪婪策略，也会导致获得长时间的徘徊
- 最佳的策略是以 0.5 的概率选择动作
- 很多时候我们需要一个确定分布的随机动作

策略退化

- 真实的最优值函数会导致真实的最优策略
- 然而近似的最优值函数可能导致完全不同的策略

例子

- 假设有两个动作，A 和 B, 其中动作 A 的真实 Q 值为 0.5001，动作 B 的真实 Q 值为 0.4999
- 假设对 B 的估计准确无误
- 如果对 A 的 Q 值估计为 0.9999，误差很大，但是导出的最优动作是正确的
- 如果对 A 的 Q 值估计为 0.4998，误差很小，但是导出的最优动作是错误的

策略退化

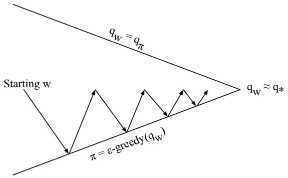
- 真实的最优值函数会导致真实的最优策略
- 然而近似的最优值函数可能导致完全不同的策略
- 使用函数近似时，也会产生策略退化

例子

- 包含两个状态：$\{A, B\}$
- 假设特征是一维的：$A:2, B:1$
- 如果最优的策略 π^* 是使 B 的 V 值比 A 大，那么使用函数近似时，参数 ω 应该是负值
- 为了逼近真实的值函数（假设 >0），那么 ω 应该是正值
- 值函数越准确，策略越差

收敛性对比

- 基于值函数的方法
  - 收敛慢. 需要对 V(or Q) 和 π 交替优化
  - 方差小
- 策略梯度方法
  - 收敛快. 直接对 π 进行优化
  - 方差大



目录

策略梯度目标函数

- 用一个参数 θ 建模策略 $\pi_\theta(s, a)$，如何寻找最优的参数 θ?
- 值函数近似时，优化的目标是使值函数的输出接近目标值
- 如何评价一个策略 π_θ 的好坏？
- 一种定义方法，使用初始状态的值函数（对于片段性任务）

$$J_1(\theta) = V^{\pi_\theta}(s_1) = \mathbb{E}_{\pi_\theta}[v_1]$$

- 策略优化问题就变成了：找 θ 使得最大化 $J_1(\theta)$
- 解此类问题有两大类算法：基于梯度的和不基于梯度的
- 本文主要是关注基于梯度的算法

数值法求梯度

- 目标函数：$J_1(\theta)$
- 策略模型：$\pi_\theta(s, a)$
- 怎么求 $\nabla_\theta J_1$?

数值梯度法

- 对于 θ 的每一个维度 $k \in [1, n]$
  - 通过给 θ 的第 k 维加入一点扰动 ε
  - 然后估计对第 k 维的偏导数

$$\frac{\partial J(\theta)}{\partial \theta_k} \approx \frac{J(\theta + \varepsilon u_k) - J(\theta)}{\varepsilon}$$

  - 其中 u_k 是单位向量，第 k 维是 1，其他均为 0

- 每次求 θ 的梯度需要计算 n 次
- 简单，噪声大，效率低
- 有时很有效，对任意策略均适用，甚至策略不可微的情况也适用

策略梯度算法

- 已有策略模型：$\pi_\theta(a|s)$
  - 策略模型可微分，即我们能求 $\nabla_\theta \pi_\theta$
- 策略梯度算法的出发点:
  - 找到一种合适的目标函数 J, 满足:
    - 最大化 J 相当于最大化期望回报值
    - 且能够建立 $\nabla_\theta J$ 与 $\nabla_\theta \pi_\theta$ 的关系.
  - 可以不需要知道 J 的具体形式，关键是计算 $\nabla_\theta J$

策略梯度的推导

参考自 https://media.nips.cc/Conferences/2016/Slides/6198-Slides.pdf

轨迹

用 τ 表示每次仿真的状态-行为序列 $S_0, A_0, \ldots, S_T, A_T$，每一个轨迹代表了强化学习的一个样本。轨迹的回报：

$$R(\tau) = \sum_{t=0}^{T} \gamma^t R(s_t, a_t)$$

用 $\mathbb{P}(\tau;\theta)$ 表示轨迹 τ 出现的概率，强化学习的目标函数可表示为

$$U(\theta) = \mathbb{E}\left(\sum_{t=0}^{T} \gamma^t R(s_t, a_t); \pi_\theta\right) = \sum_{\tau} \mathbb{P}(\tau;\theta) R(\tau)$$

对目标函数的几点说明

$$U(\theta) = \mathbb{E}\left(\sum_{t=0}^{T} \gamma^t R(s_t, a_t); \pi_\theta\right) = \sum_{\tau} \mathbb{P}(\tau; \theta) R(\tau)$$

- 强化学习的目标是

$$\max_{\theta} U(\theta) = \max \sum_{\tau} \mathbb{P}(\tau; \theta) R(\tau)$$

- 不同的策略 π_θ 影响了不同轨迹的出现的概率
- 在一个固定的环境中，轨迹的 $R(\tau)$ 是稳定的

求解 $\nabla_\theta U(\theta)$

如何求解 $\nabla_\theta U(\theta)$?

- $\mathbb{P}(\tau;\theta)$ 未知
- 无法用一个可微分的数学模型直接表达 $U(\theta)$

策略梯度解决的问题是，即使未知 $U(\theta)$ 的具体形式，也能求其梯度。包括两种角度

- 似然率的角度
- 重要性采样的角度

从似然率的角度

$$
\begin{aligned}
\nabla_\theta U(\theta) &= \nabla_\theta \sum_\tau \mathbb{P}(\tau;\theta)R(\tau) = \sum_\tau \nabla_\theta \mathbb{P}(\tau;\theta)R(\tau) \\
&= \sum_\tau \frac{\mathbb{P}(\tau;\theta)}{\mathbb{P}(\tau;\theta)} \nabla_\theta \mathbb{P}(\tau;\theta)R(\tau) = \sum_\tau \mathbb{P}(\tau;\theta) \frac{\nabla_\theta \mathbb{P}(\tau;\theta)R(\tau)}{\mathbb{P}(\tau;\theta)} \\
&= \sum_\tau \mathbb{P}(\tau;\theta)\nabla_\theta \log \mathbb{P}(\tau;\theta)R(\tau) = \mathbb{E}_\tau \left[\nabla_\theta \log \mathbb{P}(\tau;\theta)R(\tau)\right]
\end{aligned}
$$

为什么要推导成这样的形式？

- $\mathbb{P}(\tau|\theta)$ 可以通过 $\pi(a|s)$ 的模型表达 (后面会证明)
- $R(\tau)$ 可以通过采样的方式估计
- 期望符号 $\mathbb{E}$ 可以通过经验平均去估算

利用当前策略 π_θ 采样 m 条轨迹，使用经验平均来估计梯度

$$
\nabla_\theta U(\theta) \approx \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \mathbb{P}(\tau;\theta)R(\tau)
$$

从重要性采样的角度

对于参数的更新 $\theta_{\text{old}} \to \theta$, 我们使用参数 θ_{old} 产生的数据去评估参数 θ 的回报期望，由重要性采样得到:

$$U(\theta) = \sum_{\tau} \mathbb{P}(\tau|\theta_{\text{old}}) \frac{\mathbb{P}(\tau;\theta)}{\mathbb{P}(\tau|\theta_{\text{old}})} R(\tau) = \mathbb{E}_{\tau\sim\theta_{\text{old}}}\left[\frac{\mathbb{P}(\tau|\theta)}{\mathbb{P}(\tau|\theta_{\text{old}})} R(\tau)\right]$$

此时导数变成了

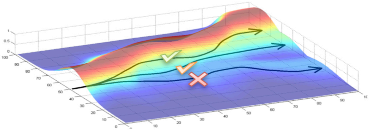
$$\nabla_\theta U(\theta) = E_{\tau\sim\theta_{\text{old}}}\left[\frac{\nabla_\theta \mathbb{P}(\tau|\theta)}{\mathbb{P}(\tau|\theta_{\text{old}})} R(\tau)\right]$$

当 $\theta = \theta_{\text{old}}$ 时，我们得到当前策略的导数:

$$\nabla_\theta U(\theta)|_{\theta=\theta_{\text{old}}} = \mathbb{E}_{\tau\sim\theta_{\text{old}}}\left[\frac{\nabla_\theta \mathbb{P}(\tau|\theta)|_{\theta_{\text{old}}}}{\mathbb{P}(\tau|\theta_{\text{old}})} R(\tau)\right] = \mathbb{E}_{\tau\sim\theta_{\text{old}}}\left[\nabla_\theta \mathbb{P}(\tau|\theta)|_{\theta_{\text{old}}} R(\tau)\right]$$

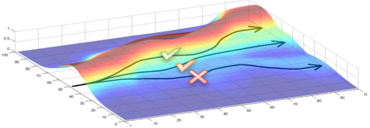
似然率梯度的理解

$$\nabla_\theta U(\theta) \approx \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau;\theta) R(\tau)$$



- $\nabla_\theta \log \mathbb{P}(\tau;\theta)$ 是轨迹 τ 的概率随参数 θ 变化最陡的方向
  1. 沿正方向，轨迹出现的概率会变大
  2. 沿负方向，轨迹出现的概率会变小
- $R(\tau)$ 控制了参数更新的方向和步长，正负决定了方向，大小决定了增大 (减小) 的幅度

似然率梯度的理解



策略梯度

- 增大了高回报轨迹出现的概率，回报值越大增加越多
- 减少了低回报轨迹出现的概率，回报值越小减少越多

注意到似然率梯度只是改变轨迹出现的概率，而没有尝试去改变轨迹

将轨迹分解成状态和动作

轨迹的似然率的表达 (链式法则):

$$\mathbb{P}(\tau^{(i)};\theta) = \prod_{t=0}^{T} \mathbb{P}(s_{t+1}^{(i)}|s_t^{(i)}, a_t^{(i)}) \cdot \pi_\theta(a_t^{(i)}|s_t^{(i)})$$

由于状态转移概率 $\mathbb{P}(s_{t+1}^{(i)}|s_t^{(i)}, a_t^{(i)})$ 中不包含参数 θ，因此求导的过程可以消掉，所以

$$\begin{aligned}
\nabla_\theta \log \mathbb{P}(\tau^{(i)};\theta) &= \nabla_\theta \log \left[\prod_{t=0}^{T} \mathbb{P}(s_{t+1}^{(i)}|s_t^{(i)}, a_t^{(i)}) \cdot \pi_\theta(a_t^{(i)}|s_t^{(i)})\right] \\
&= \nabla_\theta \left[\sum_{t=0}^{T} \log \mathbb{P}(s_{t+1}^{(i)}|s_t^{(i)}, a_t^{(i)}) + \sum_{t=0}^{T} \log \pi_\theta(a_t^{(i)}|s_t^{(i)})\right] \\
&= \nabla_\theta \left[\sum_{t=0}^{T} \log \pi_\theta(a_t^{(i)}|s_t^{(i)})\right] = \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t^{(i)}|s_t^{(i)})
\end{aligned}$$

似然率梯度估计

根据之前的推导，我们可以在仅有可微分的策略模型 π_θ 的情况下，求得 $\nabla_\theta U(\theta)$

$$\hat{\eta} = \frac{1}{m}\sum_{i=1}^{m}\nabla_\theta \log \mathbb{P}(\tau^{(i)};\theta)R(\tau^{(i)})$$

这里

$$\nabla_\theta \log \mathbb{P}(\tau^{(i)};\theta) = \sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t^{(i)}|s_t^{(i)})$$

$\hat{\eta}$ 是 $\nabla_\theta U(\theta)$ 的无偏估计

$$\mathbb{E}\left[\hat{\eta}\right] = \nabla_\theta U(\theta)$$

目录

减少方差

- 方差大
- 如果所有的 $R(\tau)$ 都是正的，那么所有轨迹出现的概率都会增加

我们可以通过以下的方法去减少方差

- 引入基线 (baseline)
- 修改回报函数
- Actor-Critic 方法
- 优势函数
- ...

引入基线

首先要证明引入基线，不影响策略梯度

$$\nabla_\theta U(\theta) \approx \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \mathbb{P}(\tau;\theta) R(\tau) = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \mathbb{P}(\tau;\theta)\left(R(\tau) - b\right)$$

$$\begin{aligned}\mathbb{E}\left[\nabla_\theta \log \mathbb{P}(\tau;\theta) b\right] &= \sum_\tau \mathbb{P}(\tau;\theta)\nabla_\theta \log \mathbb{P}(\tau;\theta) b = \sum_\tau \mathbb{P}(\tau;\theta)\frac{\nabla_\theta \mathbb{P}(\tau;\theta) b}{\mathbb{P}(\tau;\theta)} \\ &= \sum_\tau \nabla_\theta \mathbb{P}(\tau;\theta) b = \nabla_\theta\left(\sum_\tau \mathbb{P}(\tau;\theta) b\right) = \nabla_\theta b = 0\end{aligned}$$

怎么选择基线

- 选择回报值函数的期望值

$$b = \mathbb{E}\left[R(\tau)\right] \approx \frac{1}{m}\sum_{i=1}^{m} R(\tau^{(i)})$$

- 最小方差

$$b = \frac{\sum_{i=1}^{m}\left[\left(\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t^{(i)}|s_t^{(i)})\right)^2 R(\tau^{(i)})\right]}{\sum_{i=1}^{m}\left[\left(\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t^{(i)}|s_t^{(i)})\right)^2\right]}$$

最小方差

最小方差

令 $X = \frac{1}{m}\nabla_\theta \log \mathbb{P}(\tau^{(i)};\theta)\left(R(\tau^{(i)}) - b\right)$，则方差为
$Var(X) = \mathbb{E}(X-\bar{X})^2 = \mathbb{E}\left[X^2\right] - \bar{X}^2$
方差最小时即导数，($\bar{X}$ 与 b 无关)

$$\frac{\partial Var(X)}{\partial b} = E\left(X\frac{\partial X}{\partial b}\right) = 0$$

$$b = \frac{\sum_{i=1}^m \left[\left(\sum_{t=0}^T \nabla_\theta \log \pi_\theta(a_t^{(i)}|s_t^{(i)})\right)^2 R(\tau^{(i)})\right]}{\sum_{i=1}^m \left[\left(\sum_{t=0}^T \nabla_\theta \log \pi_\theta(a_t^{(i)}|s_t^{(i)})\right)^2\right]}$$

修改回报函数

当前的估计值

$$\hat{\eta} = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \mathbb{P}(\tau;\theta)\left(R(\tau) - b\right)$$
$$= \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta\left(a_t^{(i)}|s_t^{(i)}\right)\right)\left(\sum_{t=0}^{T} R(s_t^{(i)}, a_t^{(i)}) - b\right)$$

- 将来的动作不依赖过去的奖励，因此我们可以修改回报值来降低方差

$$\frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{T}\left[\nabla_\theta \log \pi_\theta\left(a_t^{(i)}|s_t^{(i)}\right)\left(\sum_{k=t}^{T} R(s_k^{(i)}, a_k^{(i)}) - b(s_k^{(i)})\right)\right]$$

这里并没有考虑 γ 值

目录

实际更新算法

实际更新时

- 考虑单条轨迹

$$\hat{\eta} = \sum_{t=0}^{T} \left[\nabla_\theta \log \pi_\theta \left(a_t | s_t \right) \left(\sum_{k=t}^{T} \gamma^{k-t} R \left(s_k, a_k \right) \right) \right]$$

- 考虑单步更新

$$\hat{\eta}_t = \nabla_\theta \log \pi_\theta \left(a_t | s_t \right) \left(\sum_{k=t}^{T} \gamma^{k-t} R(s_k, a_k) \right)$$

MC 策略梯度 (REINFORCE)

- 使用梯度上升算法更新参数 θ
- 使用采样回报值 g_t 估计真实回报值

$$\Delta\theta_t = \alpha\nabla_\theta \log\pi_\theta\left(a_t|s_t\right) g_t$$

算法 1 REINFORCE 算法

```
1: 初始化 θ
2: for 每条轨迹 {s_0, a_0, r_1, ... s_T, a_T} ~ π_θ do
3:     for t = 0 to T do
4:         θ ← θ + α∇_θ log π_θ(a_t|s_t) g_t
5:     end for
6: end for
```

使用 Critic 函数减小方差

- 我们可以使用critic函数来估计回报值减小方差

$$Q_w(s_k, a_k) \approx \sum_{t=k}^{T} \left(\gamma^{t-k} R(s_k, a_k)\right)$$

- Actor-Critic 算法维持两个参数
  - **Critic** 更新 Q 函数的参数 w
  - **Actor** 使用 Critic 的方向更新策略参数 θ
- 近似策略梯度

$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(a|s) Q_w(s, a)$$

使用优势函数减小方差

优势函数

$$A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)$$

即通过 V 函数估计基线，用 Q 函数估计回报函数

$$\begin{aligned} V_v(s) &\approx V^{\pi_\theta}(s) \\ Q_w(s,a) &\approx Q^{\pi_\theta}(s,a) \\ A(s,a) &= Q_w(s,a) - V_v(s) \end{aligned}$$

近似策略梯度

$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(a|s) A(s,a)$$

使用 TD 误差替代优势函数

- 对于真实的值函数 $V^{\pi_\theta}(s)$，TD 误差为

$$\delta^{\pi_\theta} = r + \gamma V^{\pi_\theta}(s') - V^{\pi_\theta}(s)$$

- TD 误差是优势函数的无偏估计

$$\begin{aligned}
\mathbb{E}_{\pi_\theta}[\delta^{\pi_\theta}|s,a] &= \mathbb{E}_{\pi_\theta}[r + \gamma V^{\pi_\theta}(s')|s,a] - V^{\pi_\theta}(s) \\
&= Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s) \\
&= A^{\pi_\theta}(s,a)
\end{aligned}$$

- 使用 TD 误差来计算策略梯度

$$\nabla_\theta \log \pi_\theta(s,a)\delta^{\pi_\theta}$$

- 实际使用中，使用近似的 TD 误差

$$\delta_v = r + \gamma V_v(s') - V_v(s)$$

- 通过这样的方法，我们只需要一个 critic 参数 v

带资格迹的策略梯度

- 前向视角 TD(λ)，用 λ 回报值去估计优势函数

$$\Delta\theta = \alpha \left(G_t^{\lambda} - V_v(s_t) \right) \nabla_\theta \log \pi_\theta \left(a_t | s_t \right)$$

- 这里 $G_t^{\lambda} - V_v(s_t)$ 是优势函数的有偏估计
- 后向视角 TD(λ)

$$\begin{aligned}
\delta &= r_{t+1} + \gamma V_v(s_{t+1}) - V_v(s_t) \\
e_t &= \lambda e_{t-1} + \nabla_\theta \log \pi_\theta(a_t | s_t) \\
\Delta\theta &= \alpha \delta e_t
\end{aligned}$$

小结

- 策略梯度有多种形式

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) g_t\right] && \text{REINFORCE} \\
&= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) Q_w(s,a)\right] && \text{Q Actor-Critic} \\
&= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) A_{w,v}(s,a)\right] && \text{Advantage Actor-Critic} \\
&= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) \delta_v\right] && \text{TD Actor-Critic} \\
&= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) \delta_v e\right] && \text{TD}(\lambda)\text{ Actor-Critic}
\end{aligned}
$$

- 每种形式都能推导出随机梯度上升算法
- Critic 使用了策略评价 (MC 或 TD) 来估计 $Q^\pi(s,a), A^\pi(s,a)$ 或 $V^\pi(s)$

A2C

算法 2 A2C 算法

```
初始化 Actor π_θ 和 Critic V_v
repeat k = 1 2 3 ...
    初始化状态 s
    repeat t = 1, 2, 3, ...
        根据策略 π 采样动作 a
        执行动作 a，观察 r 和 s'
        计算 TD 误差 δ = r + γV_v(s') − V_v(s)
        更新 Critic: v ← v + βδ∇_v V(s)
        更新 Actor: θ ← θ + α∇_θ log π_θ(a|s)δ
    until 终止状态
until 收敛
```

repeat $k = 1\ 2\ 3\ \ldots$; **repeat** $t = 1, 2, 3, \ldots$

$\delta = r + \gamma V_v(s') - V_v(s)$

$v \leftarrow v + \beta\delta\nabla_v V(s)$

$\theta \leftarrow \theta + \alpha\nabla_\theta \log \pi_\theta(a|s)\delta$

目录

其他策略梯度算法

- 自然梯度算法
- 信赖域策略优化算法 (TRPO)
- 近端策略优化 (PPO)
- 确定性策略梯度算法 (DPG)
- ...