强化学习

第七讲：策略梯度

教师：赵冬斌　朱圆恒　张启超

中国科学院大学
中国科学院自动化研究所

April 29, 2021

上次回顾

- 函数逼近器
- 动态规划方法 + 函数逼近器
- 无模型预测方法 + 函数逼近器
- 无模型控制方法 + 函数逼近器

上节课基于价值逼近器 RL 的优缺点

- 共同的特点:
  - 学习价值函数逼近器
  - 策略由价值逼近器提取
  - 适用于有限动作集的 MDPs 问题

上节课基于价值逼近器 RL 的优缺点

- 共同的特点:
  - 学习价值函数逼近器
  - 策略由价值逼近器提取
  - 适用于有限动作集的 MDPs 问题
- 缺点同样明显:
  - 策略是确定性的 (greedy or ϵ-greedy), 无法表示随机策略

- 疑问: 前面课程不是提到确定性策略足够表示最优策略 $\pi^* = \arg\max Q^*(s,a)$ 了吗?

[1] Gordon, G. J. (1995). Stable function approximation in dynamic programming. *Machine Learning Proceedings 1995* (pp. 261–268).

- 疑问：前面课程不是提到确定性策略足够表示最优策略 $\pi^* = \arg\max Q^*(s, a)$ 了吗？
- 前提：在价值函数等于最优价值函数时，贪心策略是最优策略

[1] Gordon, G. J. (1995). Stable function approximation in dynamic programming. *Machine Learning Proceedings 1995* (pp. 261–268).

- 疑问: 前面课程不是提到确定性策略足够表示最优策略 $\pi^* = \arg\max Q^*(s, a)$ 了吗?
- 前提: 在价值函数等于最优价值函数时, 贪心策略是最优策略
- 基于逼近器时, 逼近器和真实函数之间始终存在逼近误差
- e.g. 在使用收缩逼近器下, approximate VI 收敛到 $\hat{V}$, 它和最优 V^* 之间的误差表示成 $\varepsilon = \|\hat{V} - V^*\|_\infty$

[1] Gordon, G. J. (1995). Stable function approximation in dynamic programming. *Machine Learning Proceedings 1995* (pp. 261–268).

- 疑问: 前面课程不是提到确定性策略足够表示最优策略 $\pi^* = \arg\max Q^*(s, a)$ 了吗?
- 前提: 在价值函数等于最优价值函数时, 贪心策略是最优策略
- 基于逼近器时, 逼近器和真实函数之间始终存在逼近误差
- e.g. 在使用收缩逼近器下, approximate VI 收敛到 $\hat{V}$, 它和最优 V^* 之间的误差表示成 $\varepsilon = \| \hat{V} - V^* \|_\infty$
- 由 $\hat{V}$ 提取的贪心策略 $\hat{\pi} = \mathcal{G}(\hat{V})$, 对应的策略价值 $V_{\hat{\pi}}$
- $V_{\hat{\pi}}$ 和 V^* 之间的误差满足[1]:

$$\| V_{\hat{\pi}} - V^* \|_\infty \leq \frac{2\gamma}{1-\gamma} \varepsilon$$

[1] Gordon, G. J. (1995). Stable function approximation in dynamic programming. *Machine Learning Proceedings 1995* (pp. 261–268).

- 疑问: 前面课程不是提到确定性策略足够表示最优策略 $\pi^* = \arg\max Q^*(s, a)$ 了吗?
- 前提: 在价值函数等于最优价值函数时, 贪心策略是最优策略
- 基于逼近器时, 逼近器和真实函数之间始终存在逼近误差
- e.g. 在使用收缩逼近器下, approximate VI 收敛到 $\hat{V}$, 它和最优 V^* 之间的误差表示成 $\varepsilon = \|\hat{V} - V^*\|_\infty$
- 由 $\hat{V}$ 提取的贪心策略 $\hat{\pi} = \mathcal{G}(\hat{V})$, 对应的策略价值 $V_{\hat{\pi}}$
- $V_{\hat{\pi}}$ 和 V^* 之间的误差满足[1]:

$$\|V_{\hat{\pi}} - V^*\|_\infty \leq \frac{2\gamma}{1-\gamma}\varepsilon$$

- γ 通常取接近 1 的值,
  - $\gamma = 0.9, \frac{2\gamma}{1-\gamma} = 18; \quad \gamma = 0.95, \frac{2\gamma}{1-\gamma} = 38$
- 贪心策略和最优策略之间的价值误差被放大

[1] Gordon, G. J. (1995). Stable function approximation in dynamic programming. *Machine Learning Proceedings 1995* (pp. 261–268).

- 有的时候随机策略比确定策略更有利

- 有的时候随机策略比确定策略更有利

- 石头 -剪子 -布游戏
- 如果只玩一局，每种策略都只有 1/3 的概率赢

- 有的时候随机策略比确定策略更有利

- 石头 -剪子 -布游戏
- 如果只玩一局，每种策略都只有 1/3 的概率赢
- 但如果是一直重复的玩，
  - 一个确定性的策略很容易被对手发觉并针对
  - 一个均匀随机的策略是最优的 (即纳什均衡解)

上节课基于价值逼近器 RL 的优缺点

- 共同的特点:
  - 学习价值函数逼近器
  - 策略由价值逼近器提取
  - 适用于有限动作集的 MDPs 问题
- 缺点同样明显:
  - 策略是确定性的 (greedy or ϵ-greedy), 无法表示随机策略

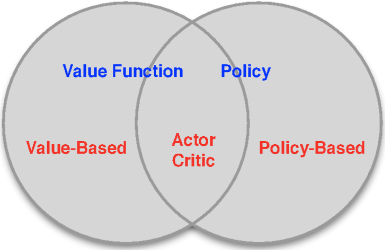
上节课基于价值逼近器 RL 的优缺点

- 共同的特点:
  - 学习价值函数逼近器
  - 策略由价值逼近器提取
  - 适用于有限动作集的 MDPs 问题
- 缺点同样明显:
  - 策略是确定性的 (greedy or ϵ-greedy), 无法表示随机策略
  - 价值函数逼近器的误差往往会导致贪心策略和最优策略之间 更大的误差
  - 难以应对 大规模动作集或连续动作空间

- 基于策略 RL 的好处:
  - 更好的收敛性 (至少有当前的策略保底, 每次都有提升的话, 回报可以稳定上升)
  - 有效解决大规模动作集或连续动作空间问题
  - 能够学习随机策略
- 基于策略 RL 的缺点:
  - 通常只能收敛到局部最优解, 而不是全局最优解
  - 对一个策略评估时通常会费时费力, 而且方差较大

基于价值和基于策略的 RL

- 基于价值 RL
  - 定义 价值逼近器
  - 隐式的策略 (e.g. ϵ-贪心策略)
- 基于策略 RL
  - 定义 策略逼近器
  - 没有价值函数
- Actor-Critic
  - 定义 价值逼近器
  - 定义 策略逼近器
  - 价值逼近器辅助策略逼近器训练



基于策略的强化学习

策略逼近器

- 与价值函数一样，用参数化的逼近器近似策略
- 根据问题类型，可以有多种形式的策略逼近器

- 与价值函数一样，用参数化的逼近器近似策略
- 根据问题类型，可以有多种形式的 策略逼近器

1 对有限动作集：表示给定状态下选择某一动作的概率

$$p(a|s) = \pi(a|s, \theta)$$

- e.g. softmax 策略，基于特征的线性组合表示每个动作被选择的概率

$$\pi(a|s, \theta) \propto e^{\phi^T(s,a)\theta}$$

- 也可以用非线性逼近器替代上述表示，e.g. 神经网络的输入层对应 s，输出层是对应 a 的 softmax 层
- 本节课我们从这种类型的策略入手

策略逼近器

- 与价值函数一样，用参数化的逼近器近似策略
- 根据问题类型，可以有多种形式的策略逼近器

2 对连续动作空间：表示给定状态下选择动作的概率分布，常见的是高斯分布

$$a \sim \mathcal{N}(\mu(s,\theta), \Sigma)$$

$$a \sim \mathcal{N}(\mu(s,\theta), \Sigma(s,\theta))$$

- e.g. 用状态特征的线性组合表示均值

$$\mu(s,\theta) = \phi^T(s)\theta$$

- 同样可以用非线性的如神经网络逼近器输出 $\mu(s)$
- 可以用固定方差，也可以使用参数化的方差表示

策略逼近器

- 与价值函数一样，用参数化的逼近器近似策略
- 根据问题类型，可以有多种形式的 策略逼近器

3 对连续动作空间：表示给定状态下确定性的动作

$$a = \pi(s, \theta)$$

- e.g. 特征线性组合

$$\pi(s, \theta) = \phi^T(s)\theta$$

- 也可以用神经网络替代

策略优化目标

- 在确定好策略的参数化表示后，我们要定义策略的优化目标
- 不同的场景对应不同的优化目标

策略优化目标

- 在确定好策略的参数化表示后，我们要定义策略的优化目标
- 不同的场景对应不同的优化目标

---

1. episodic 场景，每个轨迹都是有限固定步长，从同一初始状态 s_0 或分布 $s_0 \sim d$ 出发，优化目标是代表整个轨迹的奖励和

$$J_0(\theta) = \mathbb{E}_{\pi_\theta}[G(s_0)] = \mathbb{E}_{\pi_\theta}[r_1 + r_2 + \cdots + r_T]$$

$$J_0(\theta) = \sum_{s_0} d(s_0)\mathbb{E}_{\pi_\theta}[G(s_0)]$$

   - e.g. 单一重复的机械臂任务，给生产线上的汽车安装零件

策略优化目标

- 在确定好策略的参数化表示后，我们要定义策略的优化目标
- 不同的场景对应不同的优化目标

---

2 连续运行场景，优化目标是在状态空间上的 平均回报

$$J_{avg}(\theta) = \sum_{s} d_{\pi_\theta}(s) V_{\pi_\theta}(s)$$

- 其中 $V_{\pi_\theta}(s)$ 是使用策略 π_θ 时在状态 s 下智能体获得的期望回报

$$V_{\pi_\theta}(s) = \mathbb{E}_{\pi_\theta}[G(s)] = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s\right]$$

- d_{π_θ} 是 MDPs 在策略 $\pi(\theta)$ 下状态的静态分布，满足

$$d_{\pi_\theta}(s') = \sum_{s} d_{\pi_\theta}(s) \sum_{a} \pi_\theta(a|s) \mathcal{P}(s'|s, a)$$

- e.g. 绕地卫星的控制，长期保持固定的高度

策略优化目标

- 在确定好策略的参数化表示后，我们要定义策略的优化目标
- 不同的场景对应不同的优化目标

---

3 在有的连续运行场景，优化目标会选择**每步的平均奖励**

$$J_{avR}(\theta) = \sum_s d_{\pi_\theta}(s) \sum_a \pi_\theta(a|s)\mathcal{R}(s, a)$$

- e.g. 德州扑克，关注平均每局能赢的机率和金额
- 这节讲我们主要考虑场景 1 和场景 2 的策略目标

策略优化

- 给定策略逼近器 $\pi(\theta)$ 和策略目标后 $J(\theta)$，我们可以使用 无梯度 (gradient free) 的优化算法解决 优化问题：找到使 $J(\theta)$ 最大化的 θ
  - 爬山算法 Hill Climbing
  - 单纯型算法 Simplex/amoeba/Nelder Mead
  - 遗传算法 Genetic algorithm
  - 交叉熵算法 Cross-Entropy method (CEM)
  - 协方差矩阵自适应算法 Covariance Matrix Adaptation (CMA)
- 无梯度优化的优势:
  - 适用于任何形式的策略逼近器，甚至是不可微的逼近器
  - 很容易实现并行计算，加快学习速度
- 缺陷:
  - 计算量大，数据利用率低
  - 把问题看成黑箱，没有考虑 MDPs 问题的 时间连贯特性 (temporal structure)

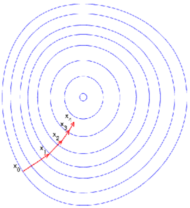
- 也可以使用基于梯度的优化方法
  - 梯度下降法 Gradient Descent
  - 共轭梯度法 Conjugate Gradient
  - 拟牛顿法 Quasi-Newton
- 相比于无梯度优化, 梯度方法数据利用率高
- 而且有效利用 MDPs 的时间连贯特性
- 本课只考虑基于梯度的策略优化算法

有限差分策略梯度

策略梯度

- 目标 $J(\theta)$ 是关于策略参数 θ 的函数
- 策略梯度: $J(\theta)$ 对 θ 的梯度

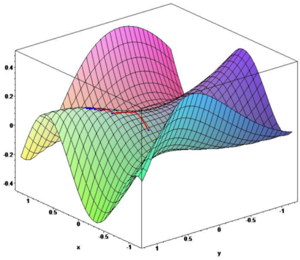
$$\nabla_\theta J(\theta) = \begin{pmatrix} \frac{\partial J(\theta)}{\partial \theta_1} \\ \vdots \\ \frac{\partial J(\theta)}{\partial \theta_n} \end{pmatrix}$$



- 策略梯度算法根据梯度上升方向调整 θ, 找到 $J(\theta)$ 的局部最大点

$$\Delta\theta = \alpha \nabla_\theta J(\theta)$$

其中 α 是更新步长



- 找到稳定的梯度更新量是策略梯度 RL 算法的关键

有限差分法 Finite Difference

- 有限差分法是求解函数梯度的一种简便有效的数值方法

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

- 有限差分法是求解函数梯度的一种简便有效的数值方法

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

- 对策略参数 θ 的每个维度 $k = 1, \ldots, n$, 在 θ_k 上增加一个微小的扰动, 用有限差分近似策略目标梯度

$$\frac{\delta J(\theta)}{\delta \theta_k} \approx \frac{J(\theta + \epsilon \mathbf{u}_k) - J(\theta)}{\epsilon}$$

其中 $\mathbf{u}_k$ 代表第 k 个元素等于 1, 其它等于 0 的单位向量

有限差分法 Finite Difference

- 有限差分法是求解函数梯度的一种简便有效的数值方法

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

- 对策略参数 θ 的每个维度 $k = 1, \ldots, n$, 在 θ_k 上增加一个微小的扰动, 用有限差分近似策略目标梯度

$$\frac{\delta J(\theta)}{\delta \theta_k} \approx \frac{J(\theta + \epsilon \mathbf{u}_k) - J(\theta)}{\epsilon}$$

其中 $\mathbf{u}_k$ 代表第 k 个元素等于 1, 其它等于 0 的单位向量

- 为了计算策略目标, 对于 episodic 问题可以用 m 次轨迹回报的均值作为目标 $J(\theta)$ 的无偏估计

$$J(\theta) = \mathbb{E}_{\pi_\theta}[G(s_0)] \approx \frac{1}{m} \sum_{l=1}^{m} \left(r_1^{(l)} + r_2^{(l)} + \cdots + r_T^{(l)} \right)$$

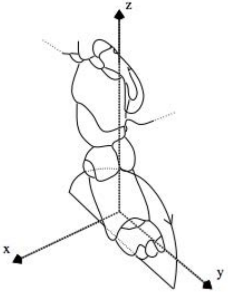
- 根据 n 次的结果构造 n 维的梯度向量, 更新参数

$$\tilde{\nabla}_\theta J(\theta) = \left[\frac{\delta J(\theta)}{\delta\theta_1}, \ldots, \frac{\delta J(\theta)}{\delta\theta_n}\right]^T$$

$$\theta \leftarrow \theta + \alpha\tilde{\nabla}_\theta J(\theta)$$

- 缺点:
  - 计算量大: 每个策略梯度要对 n+1 个策略评估, 每次评估需要生成 m 条轨迹
  - 方差大: 用整条轨迹的回报近似目标
- 优点:
  - 简单: 数值方法, 不需要分析任何梯度
  - 有时很有效: 主要针对低维策略参数空间

使用有限差分策略梯度法训练 AIBO 走路[2]



- 目标: 训练 AIBO 使它走的很快 (机器人足球比赛需求)
- 利用有限差分策略梯度法训练行走策略参数
- 根据机器人实地行走时间评估策略性能

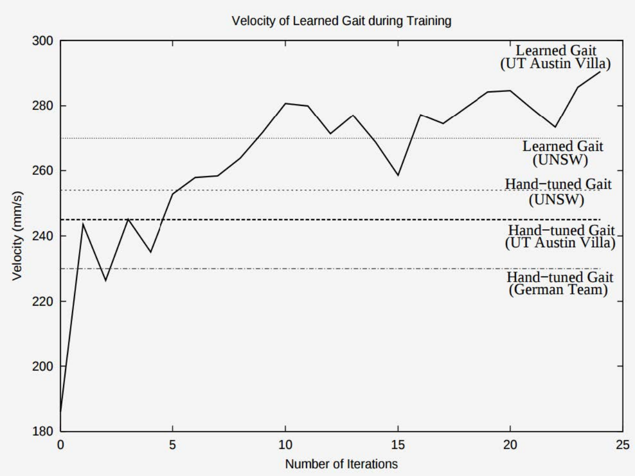
[2] Kohl, N., & Stone, P. (2004). Policy gradient reinforcement learning for fast quadrupedal locomotion. *ICRA'04*, 3:2619–2624, 2004

- AIBO 行走策略是个开环策略
- 没有状态, 只是选择一组参数定义一个椭圆, AIBO 的脚沿着椭圆轨迹移动即可向前行走
- 一共是 12 个连续的参数
  - The front locus (3 parameters: height, x-pos., y-pos.)
  - The rear locus (3 parameters)
  - Locus length
  - Locus skew multiplier in the x-y plane (for turning)
  - The height of the front of the body
  - The height of the rear of the body
  - The time each foot takes to move through its locus
  - The fraction of time each foot spends on the ground

- 每次计算策略梯度时，根据当前策略 $\pi = \{\theta_1, \ldots, \theta_N\}, (N = 12)$，随机生成 15 个扰动的策略 $R_1, R_2, \ldots, R_t, (t = 15)$
- 其中 $R_i = \{\theta_1 + \Delta_1, \ldots, \theta_N + \Delta_N\}$，每个 Δ_j 是从 $+\epsilon_j$, 0, $-\epsilon_j$ 中随机选一个
- 每个参数维度可以有多个策略结果近似相应的梯度
- 同时在 3 个 AIBOs 上对每个策略进行评估，取均值
- 每次迭代耗时大约 $7\frac{1}{2}$ 分钟

"All of the policy evaluations took place on actual robots... only human intervention required during an experiment involved replacing discharged batteries ... about once an hour."

- AIBO 行走速度沿着迭代次数的变化曲线
- 明显超过手工编码和其它学习算法的策略



解析法策略梯度

- 现在我们解析地计算策略梯度
- 假定策略 π_θ 关于参数 θ 是可微的
- 而且 $\nabla_\theta \pi_\theta(s, a)$ 是已知的

- 现在我们解析地计算策略梯度
- 假定策略 π_θ 关于参数 θ 是可微的
- 而且 $\nabla_\theta \pi_\theta(s,a)$ 是已知的
- 对 episodic 问题的一条轨迹 $\tau = (s_0, a_0, r_1, s_1, \ldots, s_{T-1}, a_{T-1}, r_T, s_T)$
- 轨迹 τ 的奖励和 (i.e. $\gamma = 1$ 的回报): $G = \sum_{t=0}^{T-1} r_{t+1}$

- 现在我们解析地计算策略梯度
- 假定策略 π_θ 关于参数 θ 是可微的
- 而且 $\nabla_\theta \pi_\theta(s,a)$ 是已知的
- 对 episodic 问题的一条轨迹
  $\tau = (s_0, a_0, r_1, s_1, \ldots, s_{T-1}, a_{T-1}, r_T, s_T)$
- 轨迹 τ 的奖励和 (i.e. $\gamma = 1$ 的回报): $G = \sum_{t=0}^{T-1} r_{t+1}$
- 策略目标对应上述奖励和在所有可能轨迹上的期望

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T-1} r_{t+1}\right] = \sum_{\tau} p(\tau;\theta) G(\tau)$$

其中 $p(\tau;\theta)$ 代表智能体使用策略 π_θ 时产生轨迹 τ 的概率

- 目标 $J(\theta)$ 对参数 θ 的梯度等于

$$\nabla_\theta J(\theta) = \sum_\tau \nabla_\theta p(\tau;\theta) G(\tau) = \sum_\tau p(\tau;\theta) \nabla_\theta \log p(\tau;\theta) G(\tau)$$

其中 $\nabla_\theta \log p(\tau;\theta)$ 称为似然比 (likelihood ratio)

- 轨迹 τ 的概率

$$p(\tau;\theta) = p(s_0) \prod_{t=0}^{T-1} \pi_\theta(a_t|s_t) \mathcal{P}(s_{t+1}|s_t, a_t)$$

$$\Rightarrow \log p(\tau;\theta) = \log p(s_0) + \sum_{t=0}^{T-1} \Big(\log \pi_\theta(a_t|s_t) + \log \mathcal{P}(s_{t+1}|s_t, a_t) \Big)$$

- 似然比的梯度

$$\nabla_\theta \log p(\tau;\theta) = \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)$$

解析的策略梯度

- 策略梯度

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \sum_\tau p(\tau;\theta)\left(\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)\right)\left(\sum_{t=0}^{T-1} r_{t+1}\right) \\
&= \mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)\right)\left(\sum_{t=0}^{T-1} r_{t+1}\right)\right]
\end{aligned}
$$

- 回顾：用样本近似期望回报：

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T-1} r_{t+1}\right] \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T-1} r_{t+1}^{i}$$

- 同样采样 N 条轨迹，用样本近似策略梯度

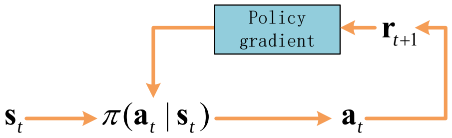
$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=0}^{T-1}\nabla_\theta \log \pi_\theta(a_t|s_t)\right)\left(\sum_{t=0}^{T-1} r_{t+1}\right)$$

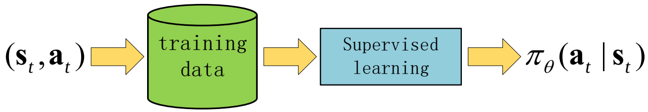
- 只涉及策略逼近器的梯度 $\nabla_\theta \log \pi_\theta(s, a)$，不包含模型 $\mathcal{P}$ 和初始状态分布 $p(s_0)$（无模型方法）
- $\nabla_\theta \log \pi_\theta(s, a)$ 也称为得分函数 score function

策略梯度和最大似然 Maximum Likelihood

策略梯度: $$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=0}^{T-1}\nabla_\theta \log \pi_\theta(a_t|s_t)\right)\left(\sum_{t=0}^{T-1} r_{t+1}\right)$$

最大似然: $$\nabla_\theta J_{ML}(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=0}^{T-1}\nabla_\theta \log \pi_\theta(a_t|s_t)\right)$$





- 基于样本的解析策略梯度是无偏的, 但方差巨大, 不适合实际应用
- 下面我们将从以下几点提高策略梯度方法的实用性
  - 时间连贯特性 temporal structure
  - 价值函数 value function
  - 优势函数 advantage function
  - 自然梯度 natural gradient

REINFORCE 算法

时间连贯性

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\sum_{t=0}^{T-1}\nabla_\theta \log \pi_\theta(a_t|s_t)\right)\left(\sum_{t=0}^{T-1} r_{t+1}\right)\right]$$

$$= \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T-1}\left(\nabla_\theta \log \pi_\theta(a_t|s_t)\sum_{t'=0}^{T-1} r_{t'+1}\right)\right]$$

($t' < t$ 时刻之前的奖励与 t 时刻的策略无关,
所以对 t 时刻的梯度更新也没有任何影响)

时间连贯性

$$\begin{aligned}\nabla_\theta J(\theta) =&\mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\sum_{t=0}^{T-1}\nabla_\theta \log \pi_\theta(a_t|s_t)\right)\left(\sum_{t=0}^{T-1} r_{t+1}\right)\right]\\ =&\mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T-1}\left(\nabla_\theta \log \pi_\theta(a_t|s_t)\sum_{t'=0}^{T-1} r_{t'+1}\right)\right]\\ =&\mathbb{E}\left[\sum_{t=0}^{T-1}\left(\nabla_\theta \log \pi_\theta(a_t|s_t)\sum_{\color{red}{t'=t}}^{T-1} r_{\color{red}{t'+1}}\right)\right]\end{aligned}$$

- 当前时刻的策略梯度使用当前时刻的回报 $G_t = \sum_{t'=t}^{T-1} r_{t'+1}$

$$\begin{aligned}\nabla_\theta J(\theta) =&\mathbb{E}\left[\sum_{t=0}^{T-1}\left(\nabla_\theta \log \pi_\theta(a_t|s_t)\sum_{t'=t}^{T-1} r_{t'+1}\right)\right]\\ \approx&\frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T-1}\left(\nabla_\theta \log \pi_\theta(a_t|s_t)\,G_t\right)\end{aligned}$$

- 同样可以把上述策略梯度公式扩展到连续运行的场景

$$
\begin{aligned}
\nabla_\theta J(\theta) =& \mathbb{E}\left[\sum_{t=0}^{\infty}\left(\nabla_\theta \log \pi_\theta(a_t|s_t)\sum_{t'=t}^{\infty}\gamma^{t'-t}r_{t'+1}\right)\right] \\
\approx& \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{\infty}\left(\nabla_\theta \log \pi_\theta(a_t|s_t)\, G_t\right)
\end{aligned}
$$

- 其中 $G_t = \sum_{t'=t}^{\infty}\gamma^{t'-t}r_{t'+1}$, $0 < \gamma < 1$
- 实际算法运行时不可能产生无穷长的轨迹，可以截断地计算回报 $G_t = \sum_{t'=t}^{t+T}\gamma^{t'-t}r_{t'+1}$，只要 T 足够大

REINFORCE 算法

定义策略逼近器 π_θ, 初始化参数 θ, 初始状态分布 d

repeat

  根据 d 和 π_θ 采样 N 条轨迹 $\tau_i = \{s_0^i, a_0^i, r_1^i, s_1^i, \ldots\}$

  for all $i = 1, \ldots, N$ **do**

    for all $t = 0, 1, \ldots$ **do**

      计算回报 $G_t^i = r_{t+1}^t + \gamma r_{t+2}^i + \ldots$

      计算更新量 $\Delta\theta = \Delta\theta + \alpha \nabla_\theta \log \pi_\theta(s_t^i, a_t^i) G_t^i$

    end for

  end for

  更新策略参数 $\theta \leftarrow \theta + \Delta\theta$

until 达到一定迭代次数或策略无明显的提升

REINFORCE 算法

定义策略逼近器 π_θ, 初始化参数 θ, 初始状态分布 d

repeat

根据 d 和 π_θ 采样 N 条轨迹 $\tau_i = \{s_0^i, a_0^i, r_1^i, s_1^i, \dots\}$

for all $i = 1, \dots, N$ **do**

for all $t = 0, 1, \dots$ **do**

计算回报 $G_t^i = r_{t+1}^t + \gamma r_{t+2}^i + \dots$

计算更新量 $\Delta\theta = \Delta\theta + \alpha \nabla_\theta \log \pi_\theta(s_t^i, a_t^i) G_t^i$

end for

end for

更新策略参数 $\theta \leftarrow \theta + \Delta\theta$

until 达到一定迭代次数或策略无明显的提升

- 也称为蒙特卡洛策略梯度算法 (episodic)

举例: 直流电机

$$s_{t+1} = \mathcal{P}(s_t, a_t) = \begin{bmatrix} 1 & 0.0049 \\ 0 & 0.9540 \end{bmatrix} s_t + \begin{bmatrix} 0.0021 \\ 0.8505 \end{bmatrix} a_t$$

- 二维连续状态空间 $\alpha \in [-\pi, \pi] \times \dot{\alpha} \in [-16\pi, 16\pi]$
- 有限动作集 $\mathcal{A} = \{-10, 0, 10\}$
- 奖励函数

$$r_{t+1} = \mathcal{R}(s_t, a_t) = -c_1 s_t(1)^2 - c_2 s_t(2)^2 - c_3 a_t^2,$$
$$c_1 = 5, c_2 = 0.01, c_3 = 0.01$$

- 折扣因子 $\gamma = 0.95$

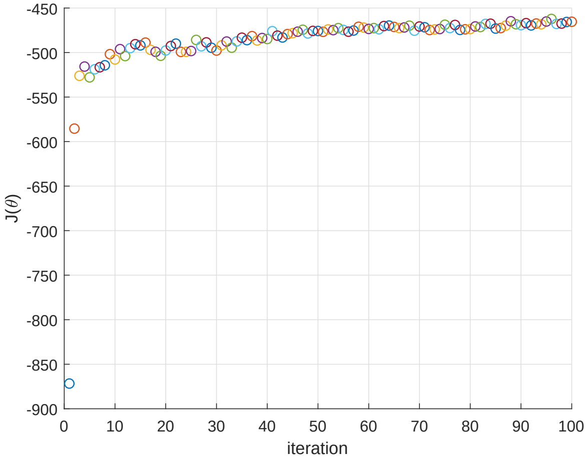
- 使用线性逼近器定义策略

$$\pi(s,a) = \frac{e^{\mathbf{x}^T(s,a)\theta}}{\sum_i e^{\mathbf{x}^T(s,a_i)\theta}}$$

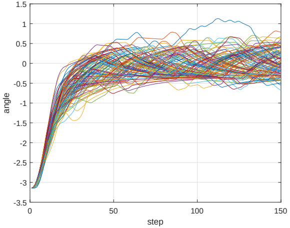
- 其中特征向量定义成 $\mathbf{x}(s,a) = \Big[[\phi_1(s),\phi_2(s),\dots]\mathcal{I}(a=\mathcal{A}_1),[\phi_1(s),\dots]\mathcal{I}(a=\mathcal{A}_2),[\phi_1(s),\dots]\mathcal{I}(a=\mathcal{A}_3)\Big]^T$
- $\phi_i(s)$ 是状态高斯核函数，中心点 $c_i = [c_{i,1}, c_{i,2}]^T$，协方差矩阵 $B_i = \begin{bmatrix} \sigma_{i,1}^2 & 0 \\ 0 & \sigma_{i,2}^2 \end{bmatrix}$
  - $c_{i,1}$ 在角度 $[-\pi,\pi]$ 范围均匀选取 9 个点
  - $c_{i,2}$ 在角速度 $[-16\pi,16\pi]$ 范围均匀选取 9 个点
  - $\sigma_{i,1} = \frac{2\pi}{8}$
  - $\sigma_{i,2} = \frac{32\pi}{8}$
- 特征向量元素一共有 9*9*3=243 个

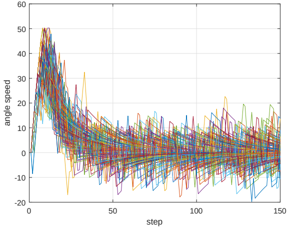
- 每次生成 100 条轨迹, 初始状态固定 $s_0 = [-\pi, 0]^T$
- 每条轨迹长 300 步, 但只使用前 150 个时刻的数据计算策略梯度
  - 保证计算第 150 个时刻回报时后序轨迹仍有 150 步
- 策略更新学习率 $\alpha = 0.05$

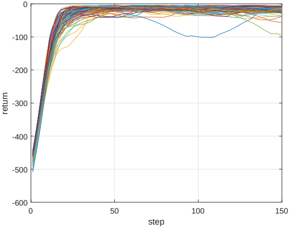
■ 学习过程中平均 $G(s_0)$ 随策略更新的变化趋势



- 100 次更新后在策略作用下生成的 100 条轨迹中状态，动作，回报曲线







Actor-Critic

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T-1}\left(\nabla_\theta \log \pi_\theta(a_t|s_t)\, G_t\right)$$

- 轨迹的回报具有明显的高方差，使用样本回报计算的策略梯度面临相同的困扰
- $Q(s_t, a_t)$ 是在初始 (s_t, a_t) 下 $G_t = r_{t+1} + \gamma r_{t+2} + \ldots$ 的期望
- 使用更稳定的 Q 函数能够降低策略梯度的方差

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T-1}\left(\nabla_\theta \log \pi_\theta(a_t|s_t)\, Q(s_t, a_t)\right)$$

- 还可以扩展到更为广泛的策略目标上

策略梯度定理

策略梯度定理[3]

对任意可微的策略 $\pi_\theta(s,a)$，在任意形式的策略目标 $J = J_0$ (episodic return)，J_{avg} (average return)，或 J_{avR} (average reward per time step) 下，策略梯度都满足

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a) Q_{\pi_\theta}(s,a)\right]$$

[3]Sutton, R.S., McAllester, D.A., Singh, S.P., & Mansour, Y. (2000). Policy Gradient Methods for Reinforcement Learning with Function Approximation. *NIPS'99*, 1057–1063, 2000

For the start-state formulation:

$$\begin{aligned}
\frac{\partial V^\pi(s)}{\partial\theta} &\stackrel{\text{def}}{=} \frac{\partial}{\partial\theta}\sum_a \pi(s,a)Q^\pi(s,a) \qquad \forall s\in\mathcal{S} \\
&= \sum_a\left[\frac{\partial\pi(s,a)}{\partial\theta}Q^\pi(s,a)+\pi(s,a)\frac{\partial}{\partial\theta}Q^\pi(s,a)\right] \\
&= \sum_a\left[\frac{\partial\pi(s,a)}{\partial\theta}Q^\pi(s,a)+\pi(s,a)\frac{\partial}{\partial\theta}\left[\mathcal{R}_s^a+\sum_{s'}\gamma\mathcal{P}_{ss'}^a V^\pi(s')\right]\right] \\
&= \sum_a\left[\frac{\partial\pi(s,a)}{\partial\theta}Q^\pi(s,a)+\pi(s,a)\sum_{s'}\gamma\mathcal{P}_{ss'}^a\frac{\partial}{\partial\theta}V^\pi(s')\right] \qquad (7) \\
&= \sum_x\sum_{k=0}^\infty \gamma^k Pr(s\to x,k,\pi)\sum_a\frac{\partial\pi(x,a)}{\partial\theta}Q^\pi(x,a),
\end{aligned}$$

after several steps of unrolling (7), where $Pr(s\to x,k,\pi)$ is the probability of going from state s to state x in k steps under policy π. It is then immediate that

$$\begin{aligned}
\frac{\partial\rho}{\partial\theta} &= \frac{\partial}{\partial\theta}E\left\{\sum_{t=1}^\infty\gamma^{t-1}r_t \,\middle|\, s_0,\pi\right\} = \frac{\partial}{\partial\theta}V^\pi(s_0) \\
&= \sum_s\sum_{k=0}^\infty\gamma^k Pr(s_0\to s,k,\pi)\sum_a\frac{\partial\pi(s,a)}{\partial\theta}Q^\pi(s,a) \\
&= \sum_s d^\pi(s)\sum_a\frac{\partial\pi(s,a)}{\partial\theta}Q^\pi(s,a). \qquad \text{Q.E.D.}
\end{aligned}$$

Actor-Critic

- 利用上节课的内容，定义一个 Q 函数逼近器 $Q_{\mathbf{w}}(s,a)$，称为 Critic
- 相应的策略逼近器 $\pi_\theta(s,a)$ 称为 Actor
- 训练 Critic 对 Actor 进行评估，同时基于 Critic 训练 Actor
  - Critic 更新价值函数的权重，从而近似当前策略的价值
    (可以使用任何一种基于样本对价值逼近器更新的 RL 算法)
  - Actor 更新策略的参数，基于样本的形式

$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(s,a) Q_{\mathbf{w}}(s,a)$$

(有偏的，因为 $Q_{\mathbf{w}}$ 不完全等于 Q_{π_θ})

基于 TD 更新的 Actor-Critic 算法

定义 Q 函数逼近器 $Q_{\mathbf{w}}(s,a)$，策略逼近器 $\pi_\theta(s,a)$，初始化 $\mathbf{w}$，θ，$s_t = s_0$，$t = 0$

repeat

  采样动作 $a_t \sim \pi_\theta(s_t, a_t)$ 并执行，观测 r_{t+1}, s_{t+1}

  $\delta = r_{t+1} + \gamma Q_{\mathbf{w}}(s_{t+1}, a_{t+1}) - Q_{\mathbf{w}}(s_t, a_t)$

  $\mathbf{w} \leftarrow \mathbf{w} + \beta \delta \nabla_{\mathbf{w}} Q_{\mathbf{w}}(s_t, a_t)$

  $\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) Q_{\mathbf{w}}(s_t, a_t)$

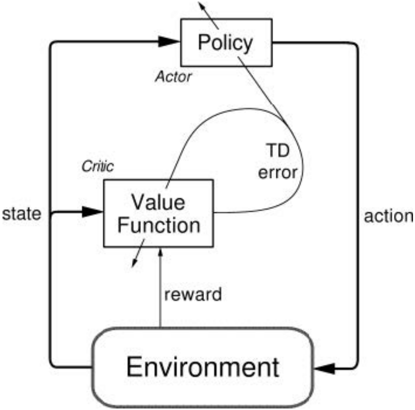
  $t \leftarrow t + 1$

until

基于 TD 更新的 Actor-Critic 算法

定义 Q 函数逼近器 $Q_{\mathbf{w}}(s,a)$，策略逼近器 $\pi_\theta(s,a)$，初始化 $\mathbf{w}$，θ，$s_t = s_0$，$t = 0$
repeat
  采样动作 $a_t \sim \pi_\theta(s_t, a_t)$ 并执行，观测 r_{t+1}, s_{t+1}
  $\delta = r_{t+1} + \gamma Q_{\mathbf{w}}(s_{t+1}, a_{t+1}) - Q_{\mathbf{w}}(s_t, a_t)$
  $\mathbf{w} \leftarrow \mathbf{w} + \beta\delta\nabla_{\mathbf{w}} Q_{\mathbf{w}}(s_t, a_t)$
  $\theta \leftarrow \theta + \alpha\nabla_\theta \log \pi_\theta(s_t, a_t) Q_{\mathbf{w}}(s_t, a_t)$
  $t \leftarrow t + 1$
until

- 存在两个学习率：Actor 学习率 α，Critic 学习率 β
- 因为 Actor 的策略梯度是基于 Critic 对当前 Actor 的策略评估，通常 β 选的比 α 大些，让 Critic 学习的快些，Actor 变化的慢些



- Actor: decide which action to take
- Critic: tells the actor how good its action was and how it should adjust

–from Sutton & Barto, 1998

- 价值逼近器始终是存在逼近误差的 $\varepsilon = \|Q_{\mathbf{w}} - Q_{\pi}\|$
- 那么基于价值逼近器的策略梯度是否会导致策略参数向错误的方向更新?

兼容 (Compatible) 的价值函数逼近

定理

如果 Actor-Critic 算法中 Q 逼近器和策略逼近器满足如下两个条件:

1. Q 逼近器和策略逼近器之间是兼容的

$$\nabla_{\mathbf{w}} Q_{\mathbf{w}}(s, a) = \nabla_\theta \log \pi_\theta(s, a)$$

2. Q 逼近器的权重等于真实 Q_{π_θ} 的最小均方误差解

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathbb{E}_{\pi_\theta}\left[\left(Q^{\pi_\theta}(s, a) - Q_{\mathbf{w}}(s, a)\right)^2\right]$$

那么基于 Q 逼近器的策略梯度是准确的

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s, a) Q_{\mathbf{w}^*}(s, a)\right]$$

证明

- 如果 Q 逼近器的权重使均方误差最小, 那么均方误差关于权重的梯度一定等于 0

$$
\begin{aligned}
0 &= \nabla_{\mathbf{w}} \mathbb{E}_{\pi_\theta}\left[(Q_{\pi_\theta}(s,a) - Q_{\mathbf{w}}(s,a))^2\right] && \text{条件 1} \\
0 &= \mathbb{E}_{\pi_\theta}\left[\left(Q_{\pi_\theta}(s,a) - Q_{\mathbf{w}}(s,a)\right) \nabla_{\mathbf{w}} Q_{\mathbf{w}}(s,a)\right] \\
&= \mathbb{E}_{\pi_\theta}\left[\left(Q_{\pi_\theta}(s,a) - Q_{\mathbf{w}}(s,a)\right) \nabla_\theta \log \pi_\theta(s,a)\right] && \text{条件 2}
\end{aligned}
$$

$$
\Rightarrow \mathbb{E}_{\pi_\theta}[Q_{\pi_\theta}(s,a) \nabla_\theta \log \pi_\theta(s,a)] = \mathbb{E}_{\pi_\theta}[Q_{\mathbf{w}}(s,a) \nabla_\theta \log \pi_\theta(s,a)]
$$

- 满足两个条件的 Q 逼近器可以用于计算精确的策略梯度

$$
\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)\, Q_{\mathbf{w}}(s,a)]
$$

证明

- 如果 Q 逼近器的权重使均方误差最小，那么均方误差关于权重的梯度一定等于 0

$$
\begin{aligned}
0 &= \nabla_{\mathbf{w}} \mathbb{E}_{\pi_\theta}\left[(Q_{\pi_\theta}(s,a) - Q_{\mathbf{w}}(s,a))^2\right] && \text{条件 1} \\
0 &= \mathbb{E}_{\pi_\theta}\left[\left(Q_{\pi_\theta}(s,a) - Q_{\mathbf{w}}(s,a)\right) \nabla_{\mathbf{w}} Q_{\mathbf{w}}(s,a)\right] \\
&= \mathbb{E}_{\pi_\theta}\left[\left(Q_{\pi_\theta}(s,a) - Q_{\mathbf{w}}(s,a)\right) \nabla_\theta \log \pi_\theta(s,a)\right] && \text{条件 2}
\end{aligned}
$$

$$
\Rightarrow \mathbb{E}_{\pi_\theta}[Q_{\pi_\theta}(s,a)\nabla_\theta \log \pi_\theta(s,a)] = \mathbb{E}_{\pi_\theta}[Q_{\mathbf{w}}(s,a)\nabla_\theta \log \pi_\theta(s,a)]
$$

- 满足两个条件的 Q 逼近器可以用于计算精确的策略梯度

$$
\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) Q_{\mathbf{w}}(s,a)]
$$

- 理论性的定理：限制了 Q 逼近器的结构，而且真实的 Q 函数是未知的，无法计算最小均方误差的解

策略梯度引入基准

基准 Baseline

- 回顾: episodic 策略梯度

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\sum_{t=0}^{T-1}\nabla_\theta \log \pi_\theta(s_t, a_t)\right)\left(\sum_{t=0}^{T-1} r_{t+1}\right)\right]$$

- 在回报上减去一个 baseline b 是不改变梯度的

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\sum_{t=0}^{T-1}\nabla_\theta \log \pi_\theta(s_t, a_t)\right)\left(\sum_{t=0}^{T-1} r_{t+1} - {\color{red}b}\right)\right]$$

基准 Baseline

- 回顾: episodic 策略梯度

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(s_t, a_t)\right)\left(\sum_{t=0}^{T-1} r_{t+1}\right)\right]$$

- 在回报上减去一个 baseline b 是不改变梯度的

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(s_t, a_t)\right)\left(\sum_{t=0}^{T-1} r_{t+1} - b\right)\right]$$

- 证明:

$$\begin{aligned}\mathbb{E}_{\tau\sim\pi_\theta}[\nabla_\theta \log \pi(\tau) b] &= \sum_\tau p(\tau)\nabla_\theta \log p(\tau) b = \sum_\tau \nabla_\theta p(\tau) b \\ &= b\nabla_\theta \sum_\tau p(\tau) = b\nabla_\theta 1 = 0\end{aligned}$$

- 对连续问题的策略梯度同样可以使用 baseline $b(s_t)$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\Big[\nabla_\theta \log \pi_\theta(s_t, a_t)\big(\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'+1} - b(s_t)\big)\Big]$$

- 对连续问题的策略梯度同样可以使用 baseline $b(s_t)$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s_t, a_t)\left(\sum_{t'=t}^{\infty}\gamma^{t'-t} r_{t'+1} - b(s_t)\right)\right]$$

- 证明:

$$\begin{aligned}
&\mathbb{E}_{\tau\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s_t, a_t) b(s_t)\right] \\
=&\mathbb{E}_{s_{0:t}, a_{0:t-1}}\left[\mathbb{E}_{a_t}\left[\mathbb{E}_{s_{t+1:T}, a_{t+1:T-1}}[\nabla_\theta \log \pi_\theta(s_t, a_t) b(s_t)]\right]\right] \\
=&\mathbb{E}_{s_{0:t}, a_{0:t-1}}\left[b(s_t)\mathbb{E}_{a_t}[\nabla_\theta \log \pi_\theta(s_t, a_t)]\right] \\
=&\mathbb{E}_{s_{0:t}, a_{0:t-1}}\left[b(s_t)\sum_{a_t}\pi_\theta(s_t, a_t)\frac{\nabla_\theta \pi_\theta(s_t, a_t)}{\pi_\theta(s_t, a_t)}\right] \\
=&\mathbb{E}_{s_{0:t}, a_{0:t-1}}\left[b(s_t)\nabla_\theta\sum_{a_t}\pi_\theta(s_t, a_t)\right] \\
=&\mathbb{E}_{s_{0:t}, a_{0:t-1}}[b(s_t)\cdot 0] = 0
\end{aligned}$$

- 减去一个 baseline 对策略梯度在期望上是无偏的, 那么对 方差 是否有影响?

- 减去一个 baseline 对策略梯度在期望上是无偏的，那么对 方差 是否有影响?
- 方差: $\mathrm{Var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2$
- 以 episodic 策略梯度为例:
$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\nabla_\theta \log p(\tau)(G(\tau) - b)\right]$
- 策略梯度方差:

$$\begin{aligned}
\mathrm{Var} =& \mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\nabla_\theta \log p(\tau)(G(\tau)-b)\right)^2\right] \\
&- \mathbb{E}_{\tau\sim\pi_\theta}\left[\nabla_\theta \log p(\tau)(G(\tau)-b)\right]^2 \\
=& \mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\nabla_\theta \log p(\tau)\right)^2 (G(\tau)-b)^2\right] \\
&- \mathbb{E}_{\tau\sim\pi_\theta}\left[\nabla_\theta \log p(\tau) G(\tau)\right]^2
\end{aligned}$$

- 前一部分是关于 b 的 二次型，后一部分与 b 无关

最佳基准 Best Baseline

- 使 episodic 策略梯度方差最小的 baseline:

$$\frac{\mathrm{dVar}}{\mathrm{d}b} = -2\mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\nabla_\theta \log p(\tau)\right)^2 G(\tau)\right] + 2b\mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\nabla_\theta \log p(\tau)\right)^2\right] = 0$$

$$\Rightarrow b^* = \frac{\mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\nabla_\theta \log p(\tau)\right)^2 G(\tau)\right]}{\mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\nabla_\theta \log p(\tau)\right)^2\right]}$$

最佳基准 Best Baseline

- 使 episodic 策略梯度方差最小的 baseline:

$$\frac{\mathrm{dVar}}{\mathrm{d}b} = -2\mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\nabla_\theta \log p(\tau)\right)^2 G(\tau)\right] + 2b\mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\nabla_\theta \log p(\tau)\right)^2\right] = 0$$

$$\Rightarrow b^* = \frac{\mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\nabla_\theta \log p(\tau)\right)^2 G(\tau)\right]}{\mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\nabla_\theta \log p(\tau)\right)^2\right]}$$

- best baseline 是加权的期望回报, 权重等于轨迹似然比的平方
- best baseline 计算过于复杂, 更常用的是

$$b = \mathbb{E}_{\tau\sim\pi_\theta}[G(\tau)] = V_0$$

- 对连续问题的策略梯度同样可以分析得到 best baseline 等于

$$b^*(s_t) = \frac{\mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\nabla_\theta \log \pi(s_t, a_t)\right)^2 \sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'+1}\right]}{\mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\nabla_\theta \log \pi(s_t, a_t)\right)^2\right]}$$

- 为了方便计算，通常选择

$$b(s_t) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'+1}\right] = V_{\pi_\theta}(s_t)$$

优势函数 Advantage Function

- 使用 V 函数作为 baseline 的 (无偏) 策略梯度

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)\big(Q_{\pi_\theta}(s,a) - V_{\pi_\theta}(s)\big)]$$

- 其中 $A_{\pi_\theta}(s,a) = Q_{\pi_\theta}(s,a) - V_{\pi_\theta}(s)$ 又称为优势函数
- 使用优势函数不改变策略梯度的期望，同时可以有效降低梯度的方差

- 我们可以分别构造两个逼近器 $Q_{\mathbf{w}}(s,a)$, $V_{\mathbf{v}}(s,a)$, 定义两组权重 $\mathbf{w}$, $\mathbf{v}$
- 用已有的算法更新两个价值逼近器, 用它们计算优势函数 (e.g. MC 方法)

$$
\begin{aligned}
Q_{\mathbf{w}}(s,a) \leftarrow & fit\Big(\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'+1} | s_t = s, a_t = a\Big) \\
V_{\mathbf{v}}(s) \leftarrow & fit\Big(\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'+1} | s_t = s\Big) \\
A(s,a) = & Q_{\mathbf{w}}(s,a) - V_{\mathbf{v}}(s) \\
\nabla_\theta J(\theta) = & \nabla_\theta \log \pi_\theta(s,a) A(s,a)
\end{aligned}
$$

- 注意 $Q_\pi(s,a) = \mathbb{E}[\mathcal{R}_s^a + \gamma V_\pi(s')]$
- 我们可以只构造 V 函数逼近器 $V_\mathbf{w}(s)$, 用 $r + \gamma V_\mathbf{w}(s')$ 近似相应的 $Q(s,a)$
- 优势函数由 TD 误差近似

$$A(s,a) = \delta = r + \gamma V_\mathbf{w}(s') - V_\mathbf{w}(s)$$

- 只需要训练一个价值逼近器, e.g. TD 方法

Advantage Actor-Critic 算法

定义 V 函数逼近器 $V_{\mathbf{w}}(s)$，策略逼近器 $\pi_\theta(s, a)$，初始化 $\mathbf{w}$，θ，$s_t = s_0$，$t = 0$

repeat

  采样动作 $a_t \sim \pi_\theta(s_t, a_t)$ 并执行，观测 r_{t+1}, s_{t+1}

  $\delta = r_{t+1} + \gamma V_{\mathbf{w}}(s_{t+1}) - V_{\mathbf{w}}(s_t)$

  $\mathbf{w} \leftarrow \mathbf{w} + \beta \delta \nabla_{\mathbf{w}} V_{\mathbf{w}}(s_t)$

  $\theta \leftarrow \theta + \alpha \delta \nabla_\theta \log \pi_\theta(s_t, a_t)$

  $t \leftarrow t + 1$

until

多种 Critic 训练方式 (上节课内容)

- 基于 MC 方法，更新目标是回报 G_t

$$\Delta \mathbf{w} = \beta(G_t - V_{\mathbf{w}}(s))\nabla_{\mathbf{w}} V_{\mathbf{w}}(s)$$

- 基于 TD(0) 方法，更新目标是 TD 目标 $r + \gamma V_{\mathbf{w}}(s')$

$$\Delta \mathbf{w} = \beta(r + \gamma V_{\mathbf{w}}(s') - V_{\mathbf{w}}(s))\nabla_{\mathbf{w}} V_{\mathbf{w}}(s)$$

- 对前向 TD(λ)，目标是 λ-回报 G_t^{λ}

$$\Delta \mathbf{w} = \beta(G_t^{\lambda} - V_{\mathbf{w}}(s))\nabla_{\mathbf{w}} V_{\mathbf{w}}(s)$$

- 对后向 TD(λ)，使用资格迹

$$\begin{aligned} \delta_t &= r_{t+1} + \gamma V_{\mathbf{w}}(s_{t+1}) - V_{\mathbf{w}}(s_t) \\ e_t &= \gamma\lambda e_{t-1} + \nabla_{\mathbf{w}} V_{\mathbf{w}}(s_t) \\ \Delta \mathbf{w} &= \beta\delta_t e_t \end{aligned}$$

- 线性价值逼近器 $V_{\mathbf{w}}(s) = \mathbf{x}^T(s_t)\mathbf{w}$ 还可以使用 LSMC 或 LSTD

$$\min_{\mathbf{w}} \sum_{t=1}^{T} (G_t - \mathbf{x}^T(s_t)\mathbf{w})^2 \qquad \min_{\mathbf{w}} \sum_{t=1}^{T} (r_{t+1} + \gamma\mathbf{x}^T(s_{t+1})\mathbf{w} - \mathbf{x}^T(s_t)\mathbf{w})^2$$

多种 Actor 训练方式

- 策略梯度有多种等价的形式

$$\begin{aligned}\nabla_\theta J(\theta) =& \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) G_t] \\ =& \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) Q_{\mathbf{w}}(s,a)] \\ =& \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) A(s,a)]\end{aligned}$$

- 每一种都可以基于样本实现随机梯度训练

$$\begin{aligned}\Delta\theta =& \alpha \nabla_\theta \log \pi_\theta(s_t,a_t)\left(\sum_{t'=t}^{T-1} r_{t'+1}\right) \\ =& \alpha \nabla_\theta \log \pi_\theta(s_t,a_t) Q_{\mathbf{w}}(s_t,a_t) \\ =& \alpha \nabla_\theta \log \pi_\theta(s_t,a_t) A(s_t,a_t)\end{aligned}$$

多种 Advantage 形式 (1)

- 学习两个价值逼近器, $Q_{\mathbf{w}}(s,a)$ 和 $V_{\mathbf{v}}(s)$, 基于两者的差

$$A(s_t,a_t) = Q_{\mathbf{w}}(s_t,a_t) - V_{\mathbf{v}}(s_t)$$

- 学习一个 V 逼近器 $V_{\mathbf{w}}(s)$, 基于 TD 目标

$$A(s_t,a_t) = r_{t+1} + \gamma V_{\mathbf{w}}(s_{t+1}) - V_{\mathbf{w}}(s_t)$$

- 基于回报

$$A(s_t,a_t) = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{T-t-1} r_T - V_{\mathbf{w}}(s_t)$$

- 基于 n-步 TD 目标

$$A(s_t,a_t) = r_{t+1} + \gamma r_{t+2} + \ldots \gamma^{n-1} r_{t+n} + \gamma^n V_{\mathbf{w}}(s_{t+n}) - V_{\mathbf{w}}(s_t)$$

多种 Advantage 形式 (2)

- 基于 λ-回报 (前向)

$$
\begin{aligned}
A(s_t, a_t) =&(1-\lambda)(r_{t+1} + \gamma V_{\mathbf{w}}(s_{t+1})) \\
&+ (1-\lambda)\lambda(r_{t+1} + \gamma r_{t+2} + \gamma^2 V_{\mathbf{w}}(s_{t+2})) \\
&+ (1-\lambda)\lambda^2(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 V_{\mathbf{w}}(s_{t+3})) \\
&\vdots \\
&- V_{\mathbf{w}}(s_t)
\end{aligned}
$$

- Advantage 的资格迹 g (后向)

$$
\begin{aligned}
\delta =& r_{t+1} + \gamma V_{\mathbf{w}}(s_{t+1}) - V_{\mathbf{w}}(s_t) && \text{(TD 误差)} \\
g \leftarrow& \gamma\lambda g + \nabla_\theta \log \pi_\theta(s_t, a_t) && \text{(Advantage 资格迹)} \\
\theta \leftarrow& \theta + \alpha\delta g && \text{(策略更新量)}
\end{aligned}
$$

自然梯度

- 目前我们使用的策略梯度是欧式距离上的最速上升方向 steepest ascent direction

$$\frac{\nabla_\theta J(\theta)}{\|\nabla_\theta J(\theta)\|} = \lim_{\varepsilon \to 0} \frac{1}{\epsilon} \underset{d \text{ s.t. } \|d\| \leq \epsilon}{\arg\max} \; J(\theta + d)$$

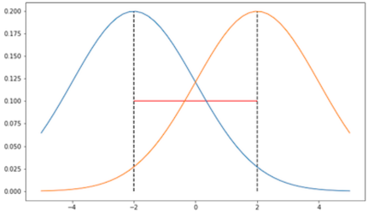
  - 最速上升方向指的是在参数 θ 欧式距离小于 ϵ 的邻域，找到一个向量 d，使目标函数在新参数 $\theta + d$ 上是最大化
  - 定义邻域使用的是欧式距离

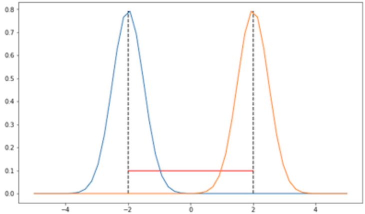
- 目前我们使用的策略梯度是欧式距离上的最速上升方向 steepest ascent direction

$$\frac{\nabla_\theta J(\theta)}{\|\nabla_\theta J(\theta)\|} = \lim_{\varepsilon \to 0} \frac{1}{\epsilon} \underset{d \text{ s.t. } \|d\| \leq \epsilon}{\arg\max} J(\theta + d)$$

  - 最速上升方向指的是在参数 θ 欧式距离小于 ϵ 的邻域，找到一个向量 d，使目标函数在新参数 $\theta + d$ 上是最大化
  - 定义邻域使用的是欧式距离
- 实际上目标函数 J 的变化是由于策略概率分布 π 的变化
- 参数空间上欧式距离的变化能否代表概率分布的变化?

举例





- 考虑高斯分布，蓝色均值 -2，红色均值 +2)
- 方差不一样：上图方差 2，下图方差 0.5
- 上下两图中蓝色和红色高斯分布在欧式距离上都等于 4
- 但是从分布上是明显不同的：上图两个分布距离近，下图两个分布距离

KL 散度

- 概率分布空间上，描述两个分布的距离通常使用 KL 散度

$$\mathrm{KL}[p(x|\theta)||p(x|\theta')] = \mathbb{E}_{p(x|\theta)}\left[\log \frac{p(x|\theta)}{p(x|\theta')}\right]$$

- 两个分布越相似，重合度越高，KL 散度越低
- 因此目标函数的优化变成了在分布空间上，以 KL 散度作为距离度量寻找最速上升方向

$$d^* = \underset{d \text{ s.t. } \mathrm{KL}[p_\theta||p_{\theta+d}] = c}{\arg\max} J(\theta + d)$$

- 将带约束的优化问题写成 Lagrangian 形式 (负号是由于 max 变成了 min)

$$d^* = \arg\min_d -J(\theta + d) + \lambda(\mathrm{KL}[p_\theta || p_{\theta+d}] - c)$$

- 使用泰勒展式

$$J(\theta + d) \approx J(\theta) + \nabla_\theta J(\theta)^T d$$

$$\begin{aligned}\mathrm{KL}[p_\theta || p_{\theta+d}] \approx & \mathrm{KL}[p_\theta || p_\theta] + \left(\nabla_{\theta'} \mathrm{KL}[p_\theta || p_{\theta'}]\big|_{\theta'=\theta}\right)^T d \\ & + \frac{1}{2} d^T \left(\nabla^2_{\theta'} \mathrm{KL}[p_\theta || p_{\theta'}]\big|_{\theta'=\theta}\right) d\end{aligned}$$

- 其中

$$\nabla_{\theta'} \mathrm{KL}[p_\theta || p_{\theta'}]\big|_{\theta'=\theta} = 0$$

$$\nabla^2_{\theta'} \mathrm{KL}[p_\theta || p_{\theta'}]\big|_{\theta'=\theta} = \mathbb{E}_{p(x|\theta)}\left[\nabla^2_\theta \log p(x|\theta)\right]$$

- 最后一项是 negative expected Hesseian of log likelihood, 在概率统计里等价于 Fisher Information Matrix

$$\mathrm{F} = \mathbb{E}_{p(x|\theta)}\left[\nabla_\theta \log p(x|\theta) \nabla_\theta \log p(x|\theta)^T\right]$$

$$\Rightarrow \arg\min_{d} -J(\theta) - \nabla_\theta J(\theta)^T d + \frac{1}{2}\lambda d^T \mathrm{F} d - \lambda c$$

- 为了解优化问题，对上式求关于 d 的偏导并设为 0

$$\begin{aligned} 0 &= -\nabla_\theta J(\theta) + \lambda \mathrm{F} d \\ d^* &= \frac{1}{\lambda}\mathrm{F}^{-1}\nabla_\theta J(\theta) \end{aligned}$$

- 所以最优梯度方向 (自然梯度 Natural Gradient) 是

$$\tilde{\nabla}_\theta J(\theta) = \mathrm{F}^{-1}\nabla_\theta J(\theta)$$

- 对策略梯度 RL

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)\, Q(s,a)] \\
\mathrm{F} &= \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) \nabla_\theta \log \pi_\theta(s,a)^T]
\end{aligned}
$$

- 对策略梯度 RL

$$\begin{aligned}\nabla_\theta J(\theta) =& \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s, a) Q(s, a)] \\ \mathrm{F} =& \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s, a) \nabla_\theta \log \pi_\theta(s, a)^T]\end{aligned}$$

- 回顾: 如果使用 兼容的线性 Q 函数逼近器

$$Q_{\mathbf{w}}(s, a) = \mathbf{x}(s, a)^T \mathbf{w}, \qquad \nabla_{\mathbf{w}} Q_{\mathbf{w}}(s, a) = \nabla_\theta \log \pi_\theta(s, a)$$

- 自然策略梯度变成

$$\begin{aligned}\nabla_\theta J(\theta) =& \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s, a) \nabla_\theta \log \pi_\theta(s, a)^T \mathbf{w}] \\ =& \mathrm{F}\mathbf{w} \\ \tilde{\nabla}_\theta J(\theta) =& \mathbf{w}\end{aligned}$$

- i.e. Actor 更新方向等于 Critic 的权重

确定型 Actor-Critic

- 以上的策略梯度算法主要针对有限动作集的 MDPs 问题
    - 策略定义的是动作的选择概率: $a \sim \pi_\theta(s)$
    - 策略梯度是价值乘上似然比的梯度:
      $\nabla_\theta J(\theta) = \mathbb{E}[Q(s,a)\nabla_\theta \log \pi_\theta(s,a)]$

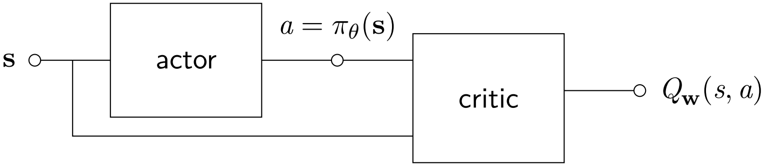
- 以上的策略梯度算法主要针对有限动作集的 MDPs 问题
  - 策略定义的是动作的选择概率: $a \sim \pi_\theta(s)$
  - 策略梯度是价值乘上似然比的梯度:
    $\nabla_\theta J(\theta) = \mathbb{E}[Q(s,a)\nabla_\theta \log \pi_\theta(s,a)]$
- 连续动作空间的 MDPs, 策略可以是确定性的: $a = \pi_\theta(s)$
- 动作的好坏直接可以由 Q 函数反映: $Q(s, \pi_\theta(s))$
- 为了找到使 Q 函数最大的动作, 我们可以利用动作的连续性和梯度反传, 令策略梯度等于

$$\nabla_\theta Q(s, \pi_\theta(s)) = \nabla_a Q(s,a)\Big|_{a=\pi_\theta(s)} \nabla_\theta \pi_\theta(s)$$

确定性 Actor-Critic

- 使用逼近器构造 Critic 近似 Q 函数 $Q_{\mathbf{w}}(s, a)$
  - 使用如 MC, TD, TD(λ), etc 训练权重 $\mathbf{w}$
- 使用逼近器构造 Actor 近似连续动作的策略 $a = \pi_\theta(s)$

$$\nabla_\theta Q_{\mathbf{w}}(s, \pi_\theta(s)) = \nabla_a Q_{\mathbf{w}}(s, a)\Big|_{a=\pi_\theta(s)} \nabla_\theta \pi_\theta(s)$$



Deterministic Policy Gradient 算法 (TD)

定义 Q 函数逼近器 $Q_{\mathbf{w}}(s,a)$，策略逼近器 $\pi_\theta(s,a)$，初始化 $\mathbf{w}$，θ，$s_t=s_0$，$t=0$

repeat

  采样动作 $a_t=\pi_\theta(s_t)+\mathcal{N}_t$ 并执行，观测 r_{t+1},s_{t+1}

  计算 TD 误差：$\delta=r_{t+1}+\gamma Q_{\mathbf{w}}(s_{t+1},\pi_\theta(s_{t+1}))-Q_{\mathbf{w}}(s_t,a_t)$

  更新 Critic：$\mathbf{w}\leftarrow\mathbf{w}+\beta\delta\nabla_{\mathbf{w}}Q_{\mathbf{w}}(s_t,a_t)$

  更新 Actor：$\theta\leftarrow\theta+\alpha\nabla_a Q_{\mathbf{w}}(s_t,a)\big|_{a=\pi_\theta(s_t)}\nabla_\theta\pi_\theta(s_t)$

  $t\leftarrow t+1$

until

Deterministic Policy Gradient 算法 (TD)

定义 Q 函数逼近器 $Q_{\mathbf{w}}(s,a)$，策略逼近器 $\pi_\theta(s,a)$，初始化 $\mathbf{w}$, θ, $s_t = s_0$, $t = 0$

repeat

  采样动作 $a_t = \pi_\theta(s_t) + \mathcal{N}_t$ 并执行，观测 r_{t+1}, s_{t+1}

  计算 TD 误差：$\delta = r_{t+1} + \gamma Q_{\mathbf{w}}(s_{t+1}, \pi_\theta(s_{t+1})) - Q_{\mathbf{w}}(s_t, a_t)$

  更新 Critic: $\mathbf{w} \leftarrow \mathbf{w} + \beta\delta\nabla_{\mathbf{w}} Q_{\mathbf{w}}(s_t, a_t)$

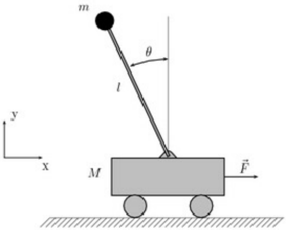
  更新 Actor: $\theta \leftarrow \theta + \alpha \nabla_a Q_{\mathbf{w}}(s_t, a)\big|_{a=\pi_\theta(s_t)} \nabla_\theta \pi_\theta(s_t)$

  $t \leftarrow t + 1$

until

- $\mathcal{N}_t$ 是 exploration noise (e.g. 高斯噪声)，避免参数陷入局部解

举例: 小车倒立摆问题[4]



$$\frac{d^2\theta}{dt^2} = \frac{g\sin\theta + \cos\theta[-F - ml\dot{\theta}^2\sin\theta + \mu_c \mathrm{sgn}(\dot{x})] - \frac{\mu_p\dot{\theta}}{ml}}{l\left(\frac{4}{3} - \frac{m\cos^2\theta}{m_c+m}\right)}$$

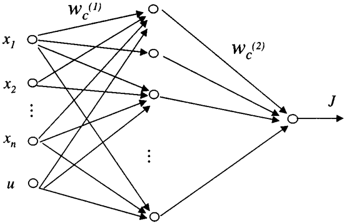
$$\frac{d^2x}{dt^2} = \frac{F + ml[\dot{\theta}^2\sin\theta - \ddot{\theta}\cos\theta] - \mu_c \mathrm{sgn}(\dot{x})}{m_c + m}$$

[4] Si, J., & Wang, Y. T. (2001). Online learning control by association and reinforcement. IEEE Transactions on Neural networks, 12(2), 264-276.

问题建立

- $s = [x, \theta, \dot{x}, \dot{\theta}]^T$
- $\mathcal{A} = u = \{+10, -10\}$
- $\mathcal{R}(s) = \begin{cases} 0, & \text{if } |x| \leq 2.4 \text{ and } |\theta| \leq 12° \\ -1, & \text{otherwise} \end{cases}$ (目标: 顶点稳定)
- $\gamma = 0.95$

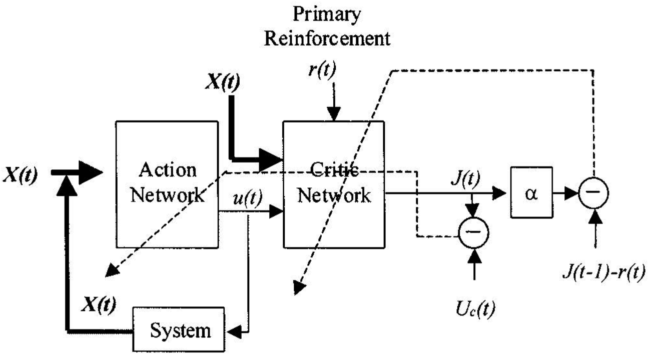
- Critic NN 结构



  - 三层网络：输入层，隐含层，输出层
  - 输入：状态，动作
  - 隐含层结点：6 个
  - 输出层：$Q(s, a)$

- Actor NN 结构
  - 与 Critic 类似，只不过输入层只包含状态
  - 输出层：$\pi(s)$

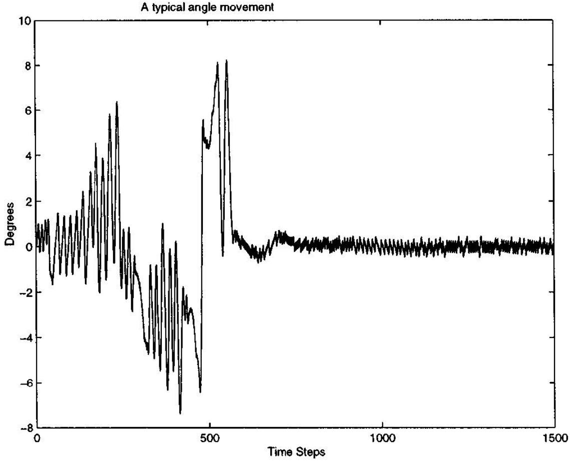
■ 系统框图



■ 如果 Action network 输出大于零, $u=+10$; 否则 $u=-10$

实验结果

- 摆杆角度控制曲线



总结

- 基于策略的强化学习
- 有限差分策略梯度
- 解析法策略梯度
- REINFORCE 算法
- Actor-Critic
- 策略梯度引入基准
- 自然梯度
- 确定型 Actor-Critic