多智能体系统与强化学习

主讲人：高阳、杨林、杨天培

https://reinforcement-learning-2025.github.io/

第四讲：策略梯度

从确定性策略转向随机策略

杨　林

大 纲

策略梯度

REINFORCE方法与方差问题

演员-评论家算法

确定性策略梯度算法

大 纲

策略梯度

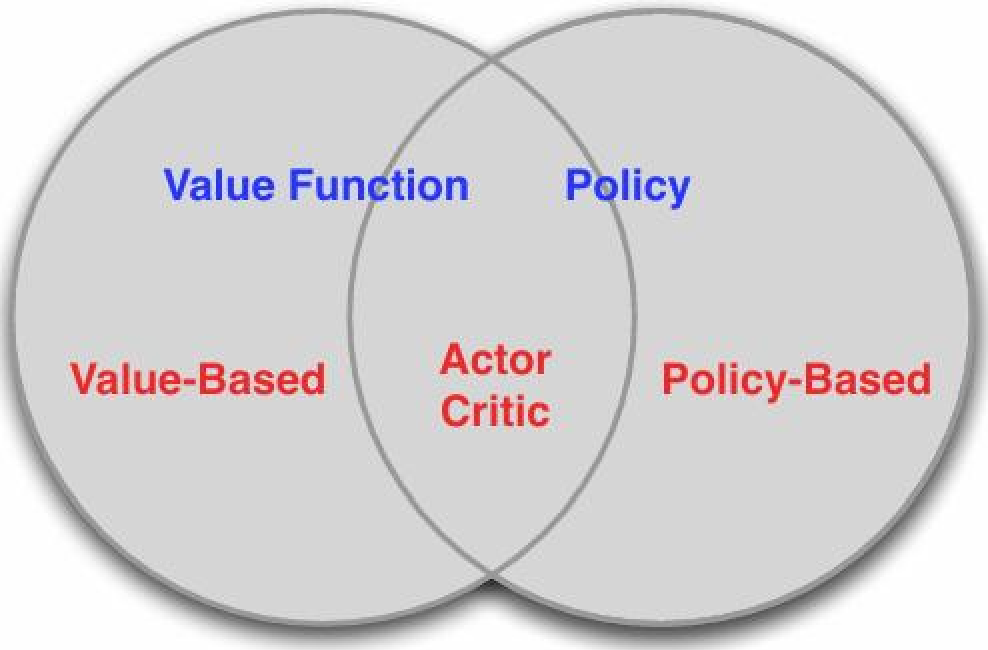
REINFORCE方法与方差问题

演员-评论家算法

确定性策略梯度算法

前瞻

- **□ 基于值的方法(上节课)**
  - ✓ 学习值函数
  - ✓ 用值函数生成策略（例如：ε-贪婪策略）
- **□ 基于策略的方法**
  - ✓ 没有值函数
  - ✓ 学习策略
- **□ 演员-评论家方法**
  - ✓ 学习值函数
  - ✓ 学习策略



回顾

- 时序差分学习
  - ✓ 时序差分学习学习动作价值函数：Sarsa
  - ✓ 时序差分学习学习最优动作价值函数：（值表型）Q-learning

Repeat (for each step of episode):

Choose a from s using policy derived from Q (e.g., ϵ-greedy)

Take action a, observe r, s'

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$$

$s \leftarrow s'$;

- 函数逼近
  - ✓ 带函数逼近的Sarsa
  - ✓ 带函数逼近的Q-learning → DQN

由更新值表变为更新逼近函数的参数！

$$\Delta \boldsymbol{w} = \alpha\left(r_{t+1} + \gamma \max \hat{Q}(s_{t+1}, a, w) - \hat{Q}(s_t, a_t, \boldsymbol{w})\right)\nabla_{\boldsymbol{w}}\hat{Q}(s_t, a_t, \boldsymbol{w})$$

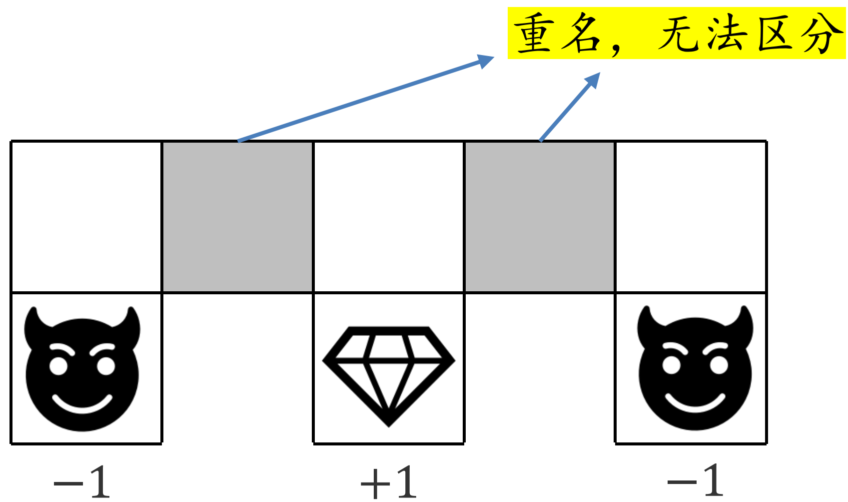
回顾

- 函数逼近
  - ✓ 计算效率提升：函数逼近的梯度更新（如反向传播）比遍历表格更高效，尤其在大规模分布式系统中
  - ✓ 泛化能力：函数逼近能自动从数据中学习状态之间的抽象特征，使相似状态共享参数化模型的权重

策略梯度方法的优劣势

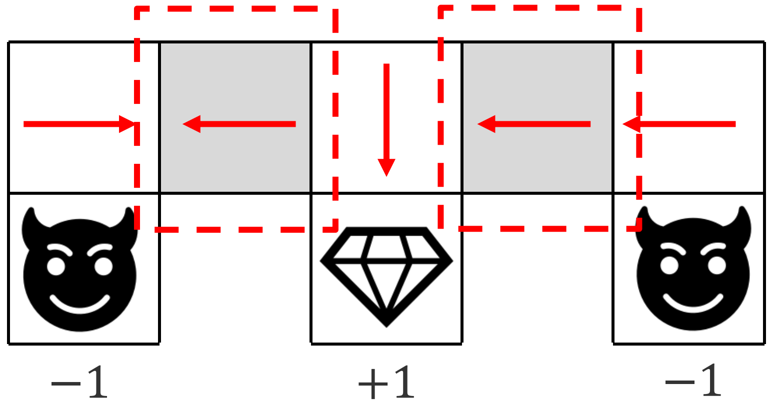
- 优势：
  - ✓ 能够有效地处理高维和连续动作空间
  - ✓ 能够学习随机(Stochastic)策略
  - ✓ 更好的收敛性
- 劣势：
  - ✓ 更容易收敛到局部最优而非全局最优
  - ✓ 策略评估通常效率低下且方差较大

Aliased Gridworld：重名的格子世界



- 环境设定如下：
  - ✓ 动作空间为方向(上，左，下，右)，特征$\phi(s,a)=\mathbf{1}(s=\text{在空白处}, a=\text{向右})$
  - ✓ 基于值的方法使用状态-动作值函数$Q_\theta(s,a)=f(\phi(s,a),\ \theta)$作为策略
  - ✓ 基于策略的方法直接使用参数化策略$\pi_\theta(s,a)=g(\phi(s,a),\ \theta)$

Aliased Gridworld：重名的格子世界



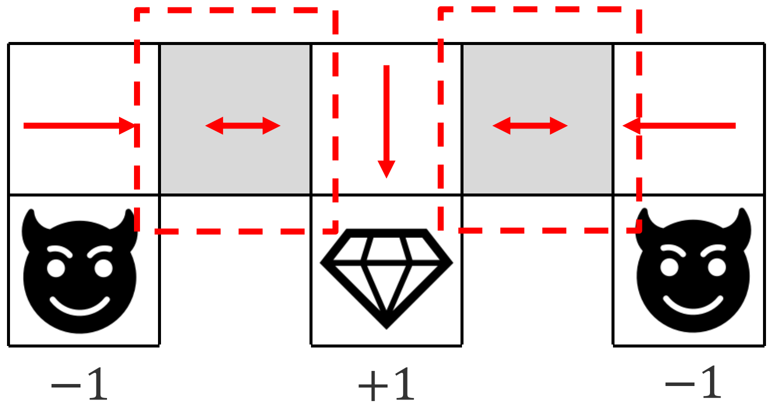
基于值的方法：

- ✓ 学习到一个最优的**确定性策略**，但无法区分灰色状态，其策略假定为

$$a = \arg\max_{a} Q_\theta(\text{灰色}, a) = \text{向左}$$

- ✓ 当使用ε-贪心策略时，Value-based方法将在走廊上徘徊很长时间
- ✓ 在上述状态上都会陷入永远找不到钻石情况

Aliased Gridworld：重名的格子世界



- 基于策略的方法：

✓ 学习到一个**随机(Stochastic)的策略，**其策略为

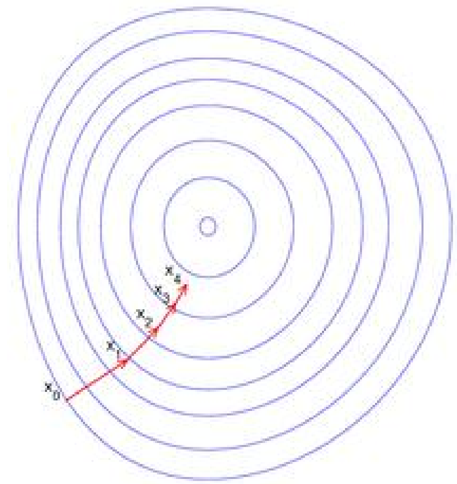
$$a \sim \pi_\theta,\ \pi_\theta(\text{灰色, 向左}) = \pi_\theta(\text{灰色, 向右}) = 0.5$$

✓ 将以较高的概率通过少量步数找到钻石

✓ 基于策略的方法能学习到最优的随机策略

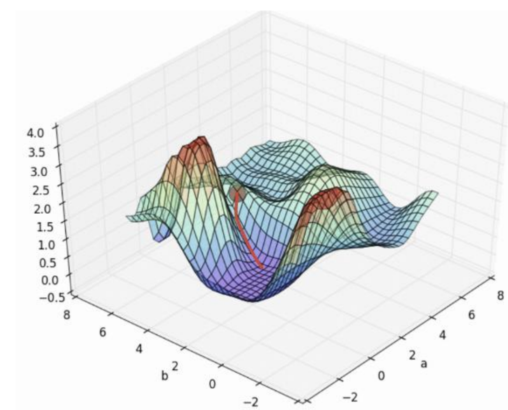
策略梗度

- 定义任意策略目标函数$J(\theta)$
- 最大目标函数的方向为正梯度方向



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

- $\nabla_\theta J(\theta)$定义为策略梯度



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

α为一个步长参数

策略梯度

- 目标：找到最优随机策略 $\pi_\theta(s,a)$，最大化收益 $J(\theta)$

✓ 初始状态收获的期望(回合性episodic环境的任务)

$$J_1(\theta) = V^{\pi_\theta}(s_1) = \mathbb{E}_{\pi_\theta}[V_1]$$

✓ 无明确初始状态，可定义平均价值(连续环境的任务)

$$J_{avV}(\theta) = \sum_s d^{\pi_\theta}(s) V^{\pi_\theta}(s)$$

✓ 或者定义为每一时间步的平均奖励

$$J_{avR}(\theta) = \sum_s d^{\pi_\theta}(s) \sum_a \pi_\theta(s,a) R_s^a = \lim_{n\to\infty} \frac{1}{n} \mathbb{E}[r_{t+1} + r_{t+2} + \ldots r_{t+n}]$$

有限差分法计算策略梯度

- 策略不可导时的近似方法(次梯度)
  - ✓ 对每一个维度$k \in [1, n]$，对参数θ的第k个分量θ_k求目标函数的偏导数
  - ✓ 通过对θ的第k个分量震荡一个微小的量ε，记为$\theta_{+\varepsilon u_k}$

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

解析法计算策略梯度

- 标准似然函数

$$L(\theta) = \prod_{i=1}^{n} f_i(y_i|\theta)$$

- 对数似然函数

$$F(\theta) = \sum_{i=1}^{n} \ln f_i(y_i|\theta)$$

- 评分函数

$$u(\theta) = \frac{\partial}{\partial\theta} F(\theta) = \sum_{i=1}^{n} \frac{\ln f_i(y_i|\theta)}{\partial\theta} = \sum_{i=1}^{n} \frac{1}{f_i(y_i|\theta)} \frac{\partial f_i(y_i|\theta)}{\partial\theta}$$

解析法计算策略梯度

- 考虑到一般情况，将值函数作为优化目标

$$J(\theta)=\mathbb{E}_{s\sim d^{\pi_\theta}(s)}[V^{\pi_\theta}(s)]$$

回顾： $$V^{\pi_\theta}(s)=\mathbb{E}_{a\sim\pi_\theta(s,a)}[Q^{\pi_\theta}(s,a)]=\sum_a \pi_\theta(s,a)Q^{\pi_\theta}(s,a)$$

当前状态值函数为当前采取动作下状态-动作值函数的期望

- 假设策略π_θ在非零时可导（策略空间连续）

前提： $$\nabla_\theta \pi_\theta(s,a)=\pi_\theta(s,a)\frac{\nabla_\theta \pi_\theta(s,a)}{\pi_\theta(s,a)}=\pi_\theta(s,a)\nabla_\theta \log \pi_\theta(s,a)$$

解析法计算策略梯度

简要推导

$$\nabla_\theta V^{\pi_\theta}(s) = \frac{\partial}{\partial\theta}\sum_a \pi_\theta(s,a)Q^{\pi_\theta}(s,a) = \sum_a \frac{\partial \pi_\theta(s,a)Q^{\pi_\theta}(s,a)}{\partial\theta}$$

①链式法则

$$= \sum_a \frac{\partial \pi_\theta(s,a)}{\partial\theta} Q^{\pi_\theta}(s,a) + \sum_a \frac{\partial Q^{\pi_\theta}(s,a)}{\partial\theta}\pi_\theta(s,a)$$

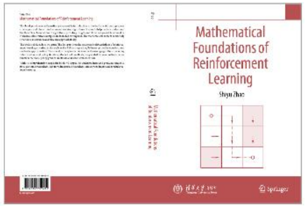
$$= \sum_a \frac{\partial \pi_\theta(s,a)}{\partial\theta} Q^{\pi_\theta}(s,a) + \underbrace{\mathbb{E}_{a\sim\pi_\theta(s,a)}\left[\frac{\partial Q^{\pi_\theta}(s,a)}{\partial\theta}\right]}_{x}$$

②x项可进一步展开本节不做具体分析，在有些教材中作为固定值

③可导前提

$$= \sum_a \pi_\theta(s,a)\nabla_\theta \log\pi_\theta(s,a)\, Q^{\pi_\theta}(s,a) + x$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim d^{\pi_\theta}(s) a\sim\pi_\theta(s,a)}[\nabla_\theta \log\pi_\theta(s,a) Q^{\pi_\theta}(s,a)] + \mathbb{E}_{a\sim\pi_\theta(s,a)}[x]$$



策略梯度定理

定理

✓ 对任意可微的策略 $\pi_\theta(s,a)$，任意策略的目标函数 $J = J_1, J_{avR}, J_{avV}$，其策略梯度是

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\underbrace{\nabla_\theta \log \pi_\theta(s,a)}_{\text{被命名为得分函数（score function）}} Q^{\pi_\theta}(s,a)]$$

✓ 用随机梯度代替真实的梯度（环境模型未知）：

$$\text{随机梯度} = \nabla_\theta \log \pi_\theta(s_t,a_t)\, Q^{\pi_\theta}(s_t,a_t)$$

常见的策略分布类型

Softmax策略分布

- ✓ 使用描述特征$\phi(s,a)$ 与参数θ的线性组合来权衡一个行为发生的几率

$$a \sim \pi_\theta(s,a) = \frac{e^{\phi(s,a)^\top\theta}}{\sum_b e^{\phi(s,b)^\top\theta}} \Rightarrow \nabla_\theta \log \pi_\theta(s,a) = \phi(s,a) - \mathbb{E}_{\pi_\theta}[\phi(s,\cdot)]$$

对应的得分函数

高斯策略分布(处理连续分布)

- ✓ 对应的行为从高斯分布

$$a \sim \mathcal{N}(\phi(s,a)^\top\theta, \sigma^2) \Rightarrow \nabla_\theta \log \pi_\theta(s,a) = \frac{(a - \phi(s,a)^\top\theta)\phi(s)}{\sigma^2}$$

对应的得分函数

大 纲

策略梯度

REINFORCE方法与方差问题

演员-评论家算法

确定性策略梯度算法

回顾：策略梯度定理

定理

- ✓ 对任意可微的策略 $\pi_\theta(s,a)$，任意策略的目标函数 $J = J_1, J_{avR}, J_{avV}$，其策略梯度是

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)\, \underline{Q^{\pi_\theta}(s,a)}]$$

状态动作值函数未知，如何估计？

REINFORCE：蒙特卡罗策略梯度

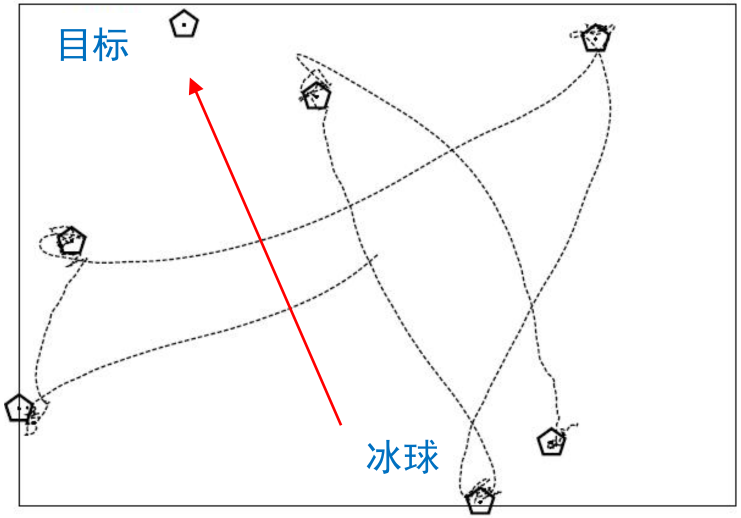
REINFORCE的思想

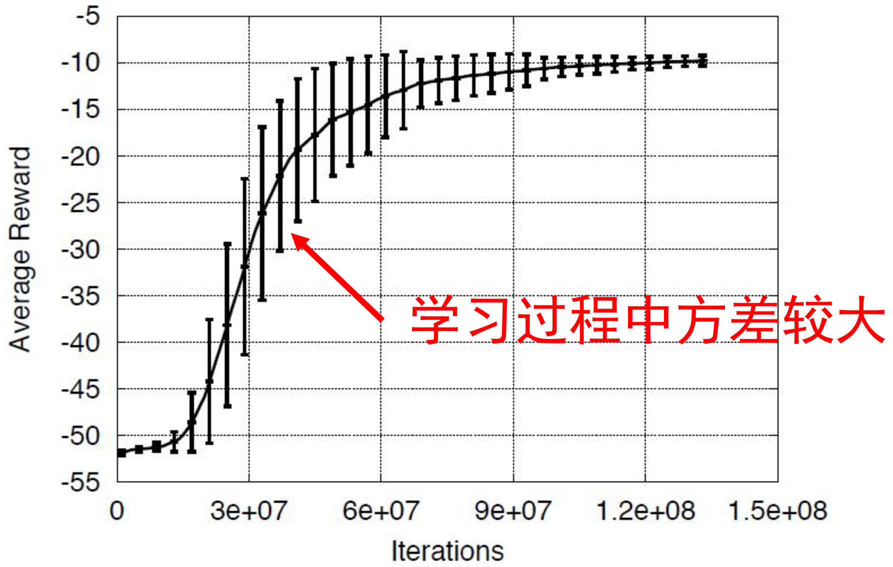
- ✓ 直接使用回报G_t作为$Q^{\pi_\theta}(s,a)$的无偏估计

$$\Delta\theta_t = \alpha \nabla_\theta \log \pi_\theta(s_t, a_t)\, G_t$$

算法：	REINFORCE算法伪代码
	Initialize： 策略参数θ
	for 序列$e = 1 \rightarrow E$ **do**
	用当前策略π_θ采样轨迹$\{s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_T, a_T, r_T\}$
	计算当前轨迹每个时刻往后的回报$\sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$记为$G_t$
	更新策略参数$\theta = \theta + \alpha \sum_t^T G_t \nabla_\theta \log \pi_\theta(s_t, a_t)$
	end for

Puck World：冰球世界



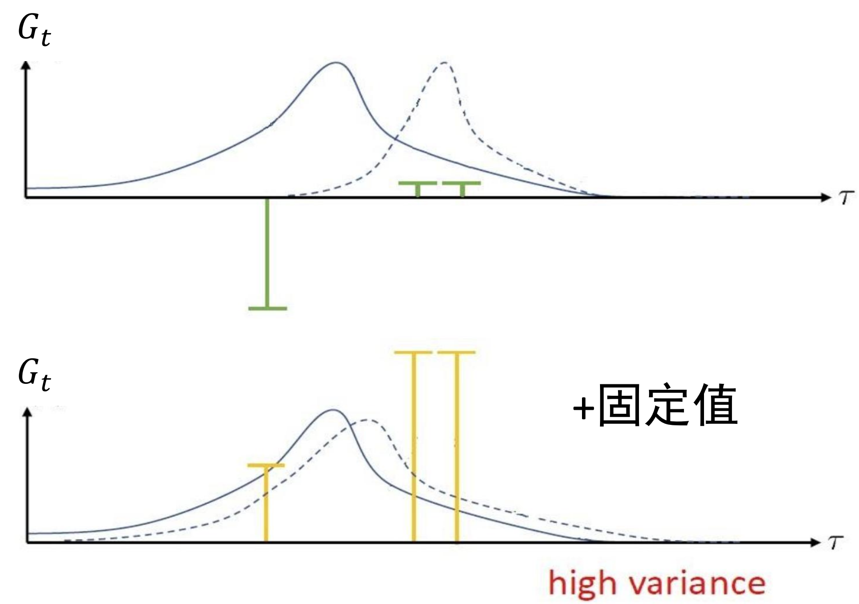


- 状态空间：个体观察自己的位置，速度以及目标物体的位置
- 动作空间：上、下、左、右四个方向加速和不操作
- 环境动力学：将个体的动作转化为其速度和位置的变化。目标物体出现位置随机，且每30秒时间更新位置。
- 奖励：奖励值的大小基于个体与目标物体之间的距离，距离越小奖励越大
- 使用蒙特卡罗策略梯度进行策略的训练

策略梯度的方差问题

◻ 方差直观解释

- ✓ 实线：真实的策略分布
- ✓ 虚线：基于三个样本（绿色和黄色）经过策略梯度更新后的策略分布
- ✓ 通过梯度更新，使得策略更倾向于后两个样本轨迹



$$\nabla_\theta J(\theta) = \nabla_\theta \log \pi_\theta(s_t, a_t)\, G_t$$

回顾：参数更新

样本奖励波动，导致实际更新后的策略分布差异大，导致学习不稳定

大 纲

策略梯度

REINFORCE方法与方差问题

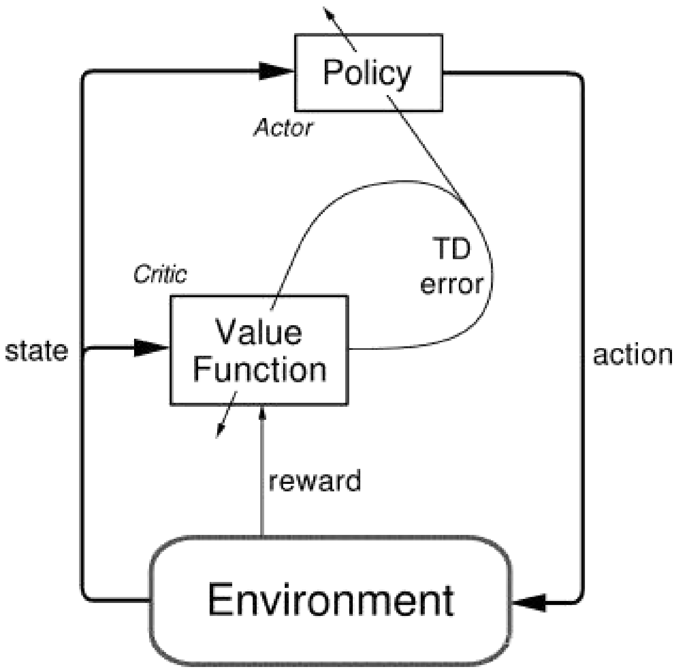
演员-评论家算法

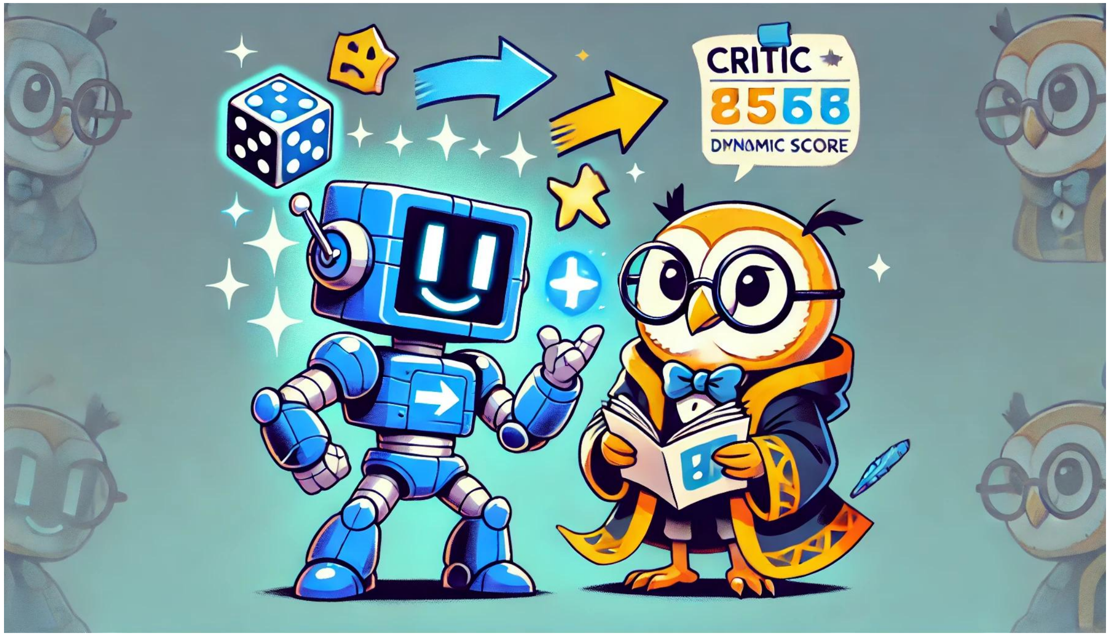
确定性策略梯度算法

AC：演员-评论家算法

Actor-Critic的思想

- ✓ REINFORCE策略梯度方法：方差来源于使用蒙特卡罗采样的不确定性
- ✓ 为什么不建立一个可训练的值函数Q_w来减少过大的方差？





AC：演员-评论家算法

Actor-Critic的思想

- ✓ Critic：采取TD更新状态-动作值函数参数
- ✓ Actor：根据Critic值估计策略梯度更新策略参数

算法： Actor-Critic（QAC）算法伪代码

Initialize：Actor参数θ，Critic参数w

for 序列 $e = 1 \rightarrow E$ **do**

  用当前策略π_θ采样轨迹$\{s_1, a_1, r_1, s_2, a_2, r_2, \cdots\}$

  为每一步数据计算$\delta_t = r_t + \gamma Q_w(s_{t+1}, a_{t+1}) - Q_w(s_t, a_t)$

  更新Actor参数$\theta = \theta + \beta \sum_t \nabla_\theta \log \pi_\theta(s_t, a_t)\, Q_w(s_t, a_t)$

  更新Critic参数$w = w + \alpha \sum_t \delta_t \nabla_w Q_w(s_t, a_t)$ *（回顾第三讲内容）*

end for

回顾：线性状态-动作值函数估计

- 与预测算法一致，需要使用目标值替换未知真实值$Q^{\pi}(s,a)$
  - ✓ **MC**，使用回报G_t

$$\Delta \boldsymbol{w} = \alpha\left(G_t - \widehat{Q}(s_t, a_t, \boldsymbol{w})\right)\nabla_{\boldsymbol{w}}\widehat{Q}(s_t, a_t, \boldsymbol{w})$$

  - ✓ **TD(0)**，使用目标值$r_{t+1} + \gamma\widehat{Q}(s_{t+1}, a_{t+1}, w)$

$$\Delta \boldsymbol{w} = \alpha\left(r_{t+1} + \gamma\widehat{Q}(s_{t+1}, a_{t+1}, w) - \widehat{Q}(s_t, a_t, \boldsymbol{w})\right)\nabla_{\boldsymbol{w}}\widehat{Q}(s_t, a_t, \boldsymbol{w})$$

时序差分用
来得到梯度！

  - ✓ **前向视角TD(λ)**，使用λ –回报G_t^{λ}

$$\Delta \boldsymbol{w} = \alpha\left(G_t^{\lambda} - \widehat{Q}(s_t, a_t, \boldsymbol{w})\right)\nabla_{\boldsymbol{w}}\widehat{Q}(s_t, a_t, \boldsymbol{w})$$

  - ✓ **后向视角TD(λ)**，类似

$$\delta_t = r_{t+1} + \gamma\widehat{Q}(s_{t+1}, a_{t+1}, \boldsymbol{w}) - \widehat{Q}(s_t, a_t, \boldsymbol{w}),\ E_t = \gamma\lambda E_{t-1} + \boldsymbol{x}(s_t)$$

$$\boldsymbol{\Delta w} = \alpha\delta_t E_t$$

演员-评论家的偏差问题

- 根据Critic值估计策略梯度时会引入偏差
- 有偏差的策略梯度可能无法找到正确的解
- 因此，需要我们谨慎地选择值函数$Q_w(s,a)$的估计方法：
  - ✓避免引入任何偏差
  - ✓遵循准确的策略梯度

如何形式化地保证？

演员-评论家的偏差问题

- 如果值函数估计满足以下两个条件：
  - ✓ 值函数估计器与策略估计兼容（一致性原则）：

$$\nabla_w Q_w(s,a) = \nabla_\theta \log\pi_\theta(s,a)$$

  - ✓ 值函数参数w使得均方误差最小化（最小化原则）：

$$\varepsilon = \mathbb{E}_{\pi_\theta}[(Q^{\pi_\theta}(s,a) - Q_w(s,a)^2]$$

- 那么，策略梯度就是精确的：

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

估计的Q值可直接带入策略梯度

演员-评论家的偏差问题

- 简单证明：
  - ✓ 如果w是的平均方差最小，那么ε关于w的梯度为O，有：

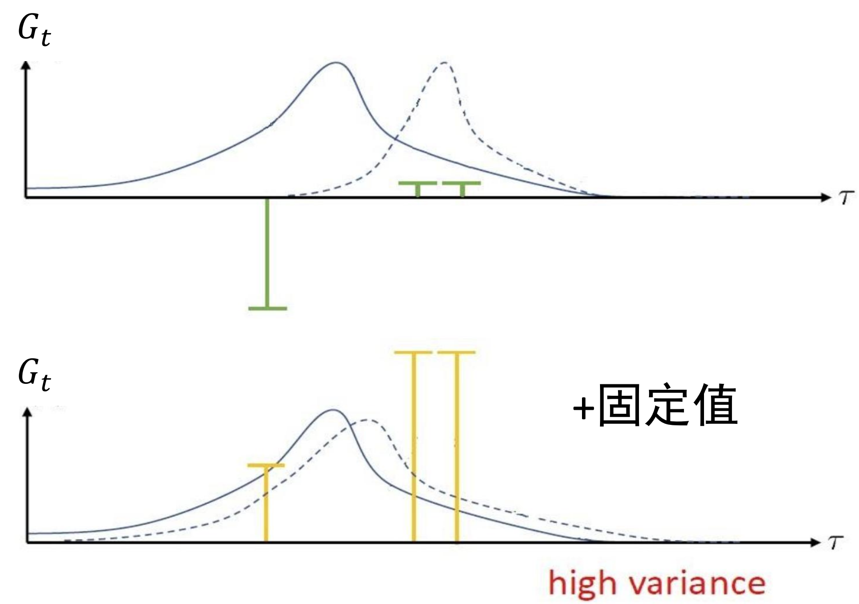
$$
\begin{aligned}
\nabla_w \varepsilon &= 0 \\
\mathbb{E}_{\pi_\theta}\left[(Q^\theta(s,a) - Q_w(s,a))\nabla_w Q_w(s,a)\right] &= 0 \\
\mathbb{E}_{\pi_\theta}\left[(Q^\theta(s,a) - Q_w(s,a))\nabla_\theta \log \pi_\theta(s,a)\right] &= 0 \\
\mathbb{E}_{\pi_\theta}\left[Q^\theta(s,a)\nabla_\theta \log \pi_\theta(s,a)\right] &= \mathbb{E}_{\pi_\theta}\left[Q_w(s,a)\nabla_\theta \log \pi_\theta(s,a)\right]
\end{aligned}
$$

  - ✓ 由上式， $Q^{\pi_\theta}(s,a)$可以使用 $Q_w(s,a)$代替

回顾：策略梯度的方差问题

方差可视化解释

- ✓ 实线：真实的策略分布
- ✓ 虚线：基于三个样本（绿色和黄色）经过策略梯度更新后的策略分布
- ✓ 通过梯度更新，使得策略更倾向于后两个样本轨迹



$$\nabla_\theta J(\theta) = \nabla_\theta \log \pi_\theta(s_t, a_t)\, G_t$$

回顾：参数更新

由于**样本奖励信号差异过大**
导致的更新策略差异过大

使用基线的方法

利用$\nabla_\theta \pi_\theta(s,a) = \pi_\theta(s,a)\nabla_\theta \log \pi_\theta(s,a)$

- 目的：通过减去基线使奖励信号分布均匀：$Q^{\pi_\theta}(s,a) - B(s)$
- 当$B(s)$与动作无关时，可在不改变期望值的情况下减少方差，

$$\mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)\, B(s)] = \sum_{s\in\mathcal{S}} d^{\pi_\theta}(s) \sum_{a} \nabla_\theta \pi_\theta(s,a) B(s)$$

$$= \sum_{s\in\mathcal{S}} d^{\pi_\theta} B(s)\, \nabla_\theta \sum_{a\in\mathcal{A}} \pi_\theta(s,a) = 0$$

- 一个好的基线是状态价值函数$B(s) = V^{\pi_\theta}(s)$
- 可以使用优势函数$A^{\pi_\theta}(s,a)$重写策略梯度（绝对值变为相对值）

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

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

优势函数估计—方法一

- 优势函数能显著减少方差，但是需要通过Critic同时估计$V^{\pi}(s)$和$Q^{\pi}(s,a)$，例如：
  - ✓ 使用两组函数估计器和参数向量(v,w)

$$V_v(s) \approx V^{\pi_\theta}(s)$$

$$Q_w(s,a) \approx Q^{\pi_\theta}(s,a)$$

$$A(s,a) = Q_w(s,a) - V_v(s)$$

- 并通过时差(TD)学习等方法来更新这两个值函数

优势函数估计—方法二

- 对于真实值函数 $V_{\pi_\theta}(s)$，时序差分(TD)误差 δ_{π_θ} 为

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

- δ^{π_θ}是优势函数的无偏估计, TD error的期望就是优势函数

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

优势函数估计—方法二

- 因此，可以使用时间差分误差来计算策略梯度

$$\nabla_\theta J(\theta) = E_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)\delta^{\pi_\theta}]$$

- 在实际应用中，我们可以使用时间差分误差来估计：

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

- 这种方法只需要一组参数v

优势函数估计—方法二

- 对于多步累积情况，可以参考TD(λ)
- 对于前向视角，策略梯度更新如下

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

- 对于后向视角，策略梯度更新如下

$$\delta_V = r_{t+1} + \gamma V_v(s_{t+1}) - V_v(s_t)$$

$$e_{t+1} = \lambda e_t + \nabla_\theta \log \pi_\theta(s_t, a_t)$$

$$\Delta\theta = \alpha\delta e_t$$

回顾：评论家更新方法

- 评论家更新方式和上一节值更新方法相似，更新方式如下
  - ✓ **蒙特卡罗方法**（MC），使用回报G_t

$$\Delta \boldsymbol{w} = \alpha\left(G_t - \hat{V}_{\boldsymbol{w}}(s_t)\right)\nabla_{\boldsymbol{w}}\hat{V}_{\boldsymbol{w}}(s_t)$$

  - ✓ **时差学习**TD(0)，使用目标值$R_{t+1} + \gamma\hat{V}(s_{t+1}, \boldsymbol{w})$

$$\Delta \boldsymbol{w} = \alpha\left(R_{t+1} + \gamma\hat{V}_{\boldsymbol{w}}(s_{t+1}) - \hat{V}_{\boldsymbol{w}}(s_t)\right)\nabla_{\boldsymbol{w}}\hat{V}_{\boldsymbol{w}}(s_t)$$

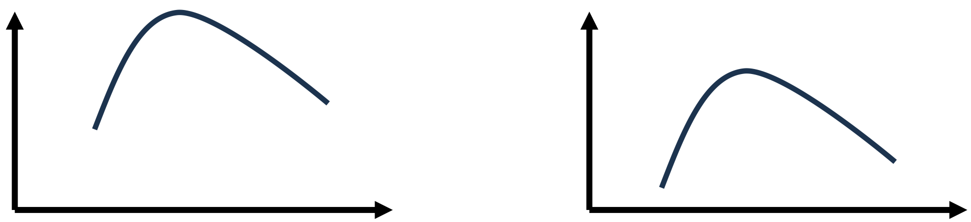
  - ✓ **时差学习**TD(λ)，使用λ –回报G_t^{λ}

$$\Delta \boldsymbol{w} = \alpha\left(G_t^{\lambda} - \hat{V}_{\boldsymbol{w}}(s_t)\right)\nabla_{\boldsymbol{w}}\hat{V}_{\boldsymbol{w}}(s_t)$$

总结：偏差与方差

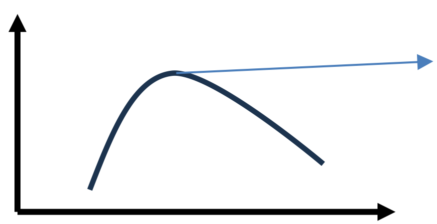
以凸函数的优化为例:

- 方差：函数绝对值（回报）使得返回函数的梯度差别过大



使用Q时左图比右图在策略更新时拥有绝对值更大的梯度

- 偏差：存在一个期望非零的梯度$\nabla_\theta \log \pi_\theta(s_t, a_t) \cdot e(s_t, a_t)$



接近最优点
时梯度不为
0！

大 纲

策略梯度

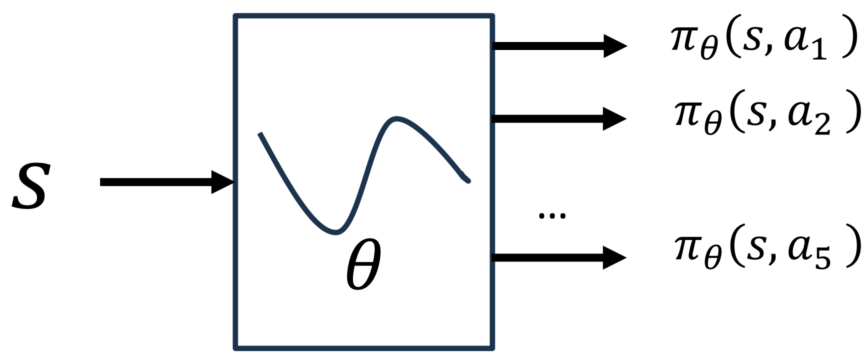
REINFORCE方法与方差问题

演员-评论家算法

确定性策略梯度算法

确定性策略梯度（DPG）

- 回顾策略梯度算法都要求$\pi_\theta(s,a)$都是大于0的,也就是随机性的策略，可否采用确定性的策略？好处是什么？



如果某个状态下输出的是
无限个或连续的动作？

确定性策略梯度（DPG）

- 定义确定性的策略

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

- 定义目标函数

$$J(\theta) = \mathbb{E}\left[v_\mu(s)\right] = \sum_s d_0(s) v_\mu(s)$$

- $d_0(s)$满足$\sum_s d_0(s) = 1$

确定性策略梯度（DPG）

- 计算梯度

$$\nabla_\theta J(\theta) = \sum_s \rho_\mu(s) \nabla_\theta \mu(s,\theta) (\nabla_a Q(s,a))|_{a=\mu(s)}$$

- 上述梯度的计算并不依赖于a_t,因此并不一定是online的
- 梯度更新：

$$\theta_{t+1} = \theta_t + \alpha_\theta \sum_s \rho_\mu(s) \nabla_\theta \mu(s,\theta) (\nabla_a Q(s,a))|_{a=\mu(s)}$$
$$= \theta_t + \alpha_\theta \mathbb{E}_{S\sim\rho_\mu}[\nabla_\theta \mu(S,\theta)(\nabla_a Q(S,a))|_{a=\mu(S)}]$$

确定性策略梯度（DPG）

算法：	Deterministic A-C 算法伪代码

Initialize：给定的行为策略$\beta(a|s)$，Actor参数θ，Critic参数w,

for each t **do**

生成行动a_t,观察r_{t+1}, s_{t+1}

计算时序差分：

$$\delta_t = r_{t+1} + \gamma Q_{w_t}(s_{t+1}, \mu(s_{t+1}, \theta_t)) - Q_{w_t}(s_t, a_t)$$

更新Critic参数:

$$w_{t+1} = w_t + \alpha_w \delta_t \sum_t \nabla_w Q_{w_t}(s_t, a_t)$$

更新Actor参数:

$$\theta_{t+1} = \theta_t + \alpha_\theta \left[\nabla_\theta \mu(s_t, \theta_t)(\nabla_a Q_{w_{t+1}}(s_t, a,))|_{a=\mu(s_t)}\right]$$

end for

策略梯度算法总结

- 一般策略梯度有以下几种表达形式：

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)\, G_t] && \textit{REINFORCE} \\
&= \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)\, Q^w(s,a)] && \textit{Q Actor-Critic} \\
&= \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)\, A(s,a)] && \textit{Advantage Actor-Critic} \\
&= \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)\, \delta] && \textit{TD Actor-Critic} \\
&= \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)\, \delta_e] && \textit{TD}(\lambda)\ \textit{Actor-Critic} \\
&= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \mu(s,\theta)(\nabla_a Q(s,a))|_{a=\mu(s)}\right] && \textit{DPG}
\end{aligned}
$$

- 每种形式均对应一种随机梯度上升算法
- 评论家通过策略评估方法（如MC或TD学习）来估计$Q^\pi(s,a)$、$A^\pi(s,a)$或者$V^\pi(s)$

思考和讨论

1. 基于值和基于策略算法的区别是什么？
2. 方差和偏差的含义和区别是什么？
3. QAC和优势函数的区别是什么？

谢 谢 ！