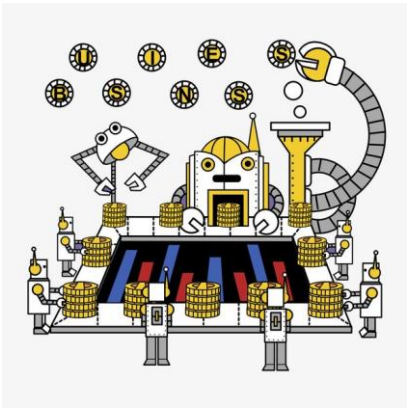
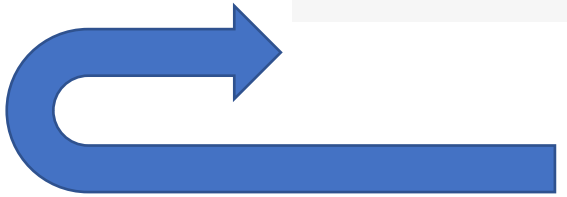
具身智能-04

刘华平

2024年3月12日

背景





Actions

Actions

Actions

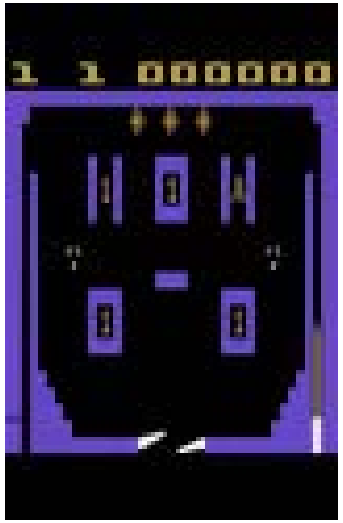
Labels

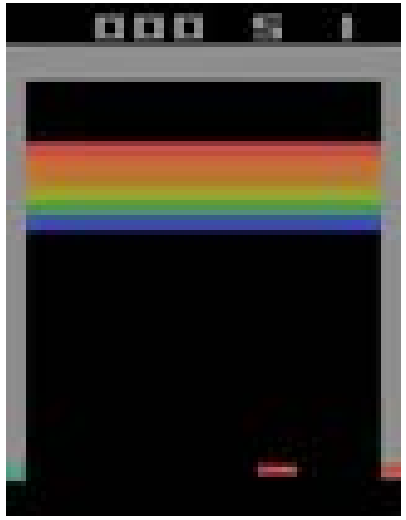
Images

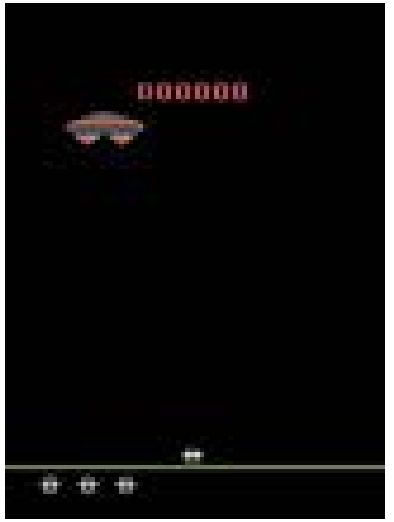
Texts

强化学习——深度Q学习

➢ 深度Q网络（DQN）







Goal：

Complete the game with the highest score.

State：

Raw pixel inputs of the game state

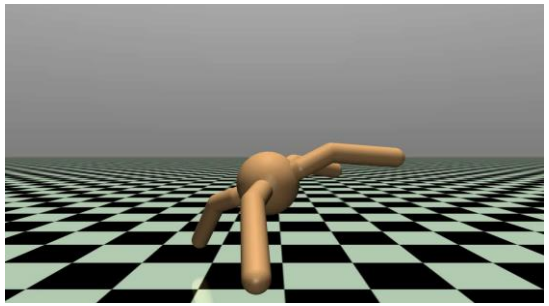
Action：

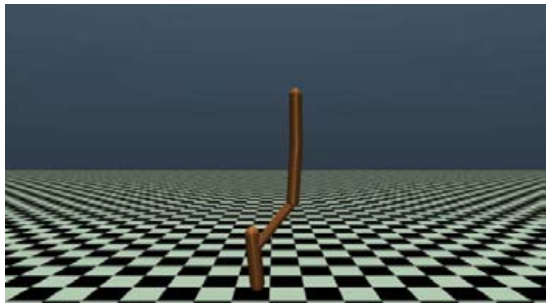
Game controls e.g. Left, Right, Up, Down

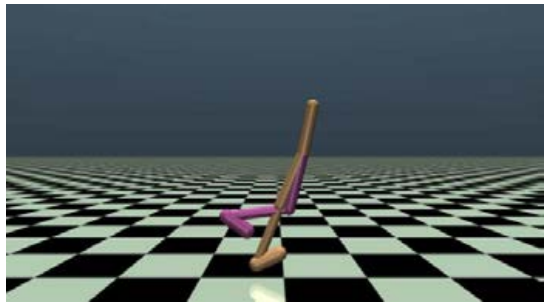
Reward：

Score increase/decrease at each time step.

➢ 深度Q网络（DQN）的局限







Robot Locomotion

Goal：

Make the robot move forward.

State：

Angle and position of the joints

Action：

Torques applied on joints

Reward：

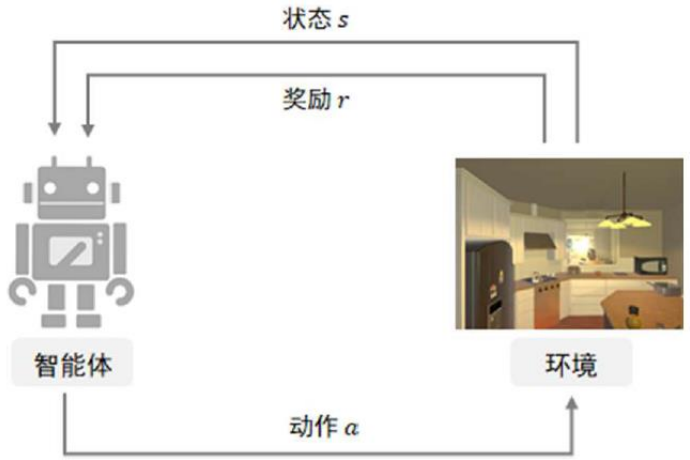
1 at each time step upright + forward movement.

- 背景
- 强化学习：Q学习
- 强化学习：策略梯度

强化学习——回顾

➢ 强化学习

Reinforcement learning solves a particular kind of problem where decision making is sequential, and the goal is long-term, such as game playing, robotics, resource management, or logistics.



- $\mathcal{S}$ 为关于智能体状态的集合
- $\mathcal{A}$ 为关于智能体动作的集合
- T 为状态转移函数，定义为：$T(s'|s,a)=\text{Prob}\left[s_{t+1}=s'|s_t=s,a_t=a\right]$，其中 $s_t\in\mathcal{S}$ 与 $a_t\in\mathcal{A}$ 分别代表 t 时刻的状态与动作取值，而 $s_{t+1}\in\mathcal{S}$ 则代表 $t+1$ 时刻的状态取值。这个概率转化关系是 Markov 决策过程的具体体现。即只要 t 时刻的状态 s_t 与动作 a_t 可知，则不再需要所有的历史信息就可以确定 $t+1$ 时刻的状态取值 s_{t+1} 取值。通俗地说就是：$t+1$ 时刻的状态取值 s_{t+1} 仅与 t 时刻的状态 s_t 与动作 a_t 有关，而与之前的信息无关。此外，即使状态 s_t 与动作 a_t 已经确定了，s_{t+1} 也是可以在状态集合 $\mathcal{S}$ 中随机取值的。状态转移函数显然满足：$\sum_{s'\in\mathcal{S}}T(s'|s,a)=1$。
- r 为一标量，称为立即奖励，用于表示在状态 s 下，采取动作 a，并转移到状态 s' 的情况下，环境给智能体反馈的奖励。这一奖励反映了智能体与环境的交互，其核心是一个函数映射：$r:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}$。也可具体展开记为 $r(s,a,s')$。注意我们也可以进一步定义 $r(s,a)=\mathbb{E}_{s'\sim T(s,a)}\left(r(s,a,s')\right)$，用于刻画转移到不同状态的平均奖励。
- $\gamma\in[0,1)$ 为一标量，称为遗忘因子，用于刻画奖励函数的累计效应。具体作用将在后面介绍。

$$\langle S,A,T,r,\gamma\rangle$$

$$\max_{\pi}\ \mathbb{E}_{\tau\sim p_{\pi}(\tau)}\left\{\sum_{t=0}^{+\infty}\gamma^{t}r(s_t,a_t,s_{t+1})\right\}$$

强化学习——回顾

➢ Q学习

$$\max_{\pi} \quad \mathbb{E}_{\tau \sim p_\pi(\tau)} \left\{ \sum_{t=0}^{+\infty} \gamma^t r(s_t, a_t, s_{t+1}) \right\}$$

$$V^\pi(s) = \mathrm{E}_{\tau \sim p_\pi(\tau)} \left\{ \sum_{t=0}^{+\infty} \gamma^t r(s_t, a_t, s_{t+1}) \mid s_0 = s \right\}$$

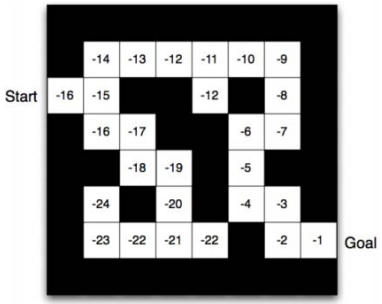
$$Q^\pi(s,a) = \mathbb{E}_{s' \sim T(s'|s,a)} \left(r(s,a,s') + \gamma \cdot V^\pi(s') \right)$$

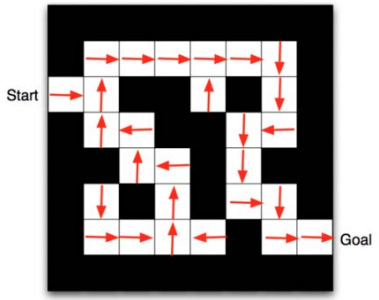
$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} \quad \mathbb{E}_{s' \sim T(s'|s,a)} \left(r(s,a,s') + \gamma \cdot V^*(s') \right)$$

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} Q^*(s,a)$$

$$V(s) = \mathbb{E}_{s' \sim T(s'|s,a)} \left(r(s,a,s') + \gamma V(s') \right)$$

$$Q(s_t, a_t) = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1})$$





强化学习——策略优化

➢ 问题描述

离散 $a_t = \pi(s_t)$

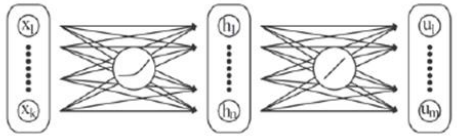
连续 $\pi(a_t \mid s_t)$

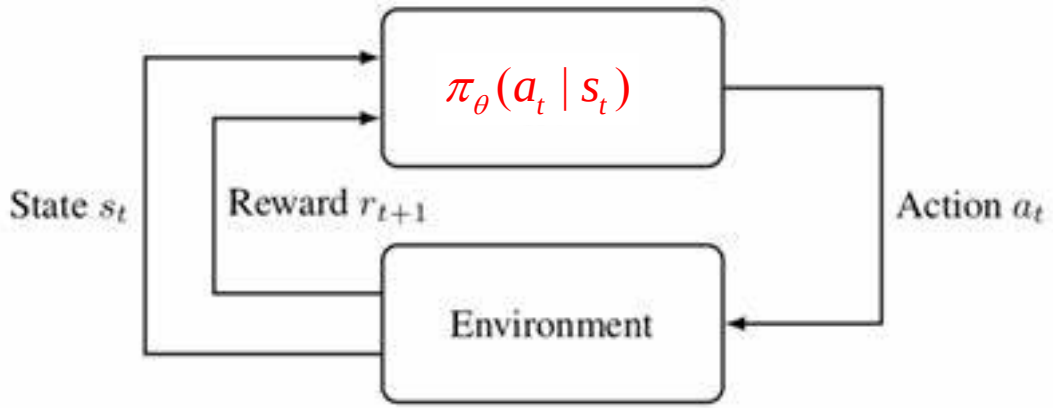
$$\pi(a \mid s) = \delta\left(\arg\max_{a \in \mathcal{A}} Q(s,a)\right)$$

$\pi(a_t \mid s_t)$

强化学习——策略优化

➢ 问题描述





$$\max_{\pi} \ \mathbb{E}_{\tau \sim p_\pi(\tau)} \left\{ \sum_{t=0}^{+\infty} \gamma^t r(s_t, a_t, s_{t+1}) \right\}$$

直接

$$V(s_t)$$

$$Q(s_t, a_t)$$

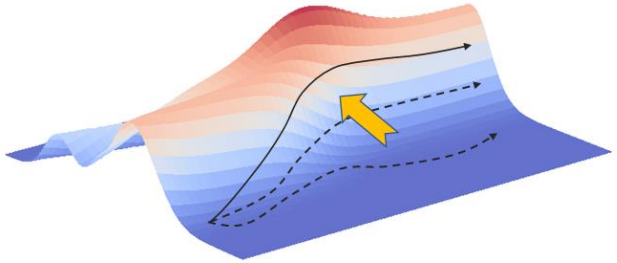
$$\tau = \{s_0, a_1; s_1, a_1; s_2, a_2; \ldots\}$$

$$J(\pi) = \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) \right]$$

强化学习——策略优化

➢ 策略梯度学习方法

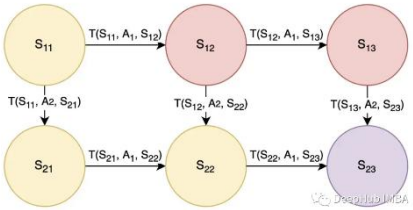
$$\max J(\pi) = \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) \right]$$



$$\tau = \{s_0, a_1; s_1, a_1; s_2, a_2; \ldots\}$$

- 优化空间

$$p_\pi(\tau) = p_0(s_0) \prod_{t=0}^{\infty} \pi(a_t \mid s_t) T(s_{t+1} \mid s_t, a_t)$$



- 优化变量

$$\theta^* = \arg\max_{\theta} \quad J(\pi_\theta)$$

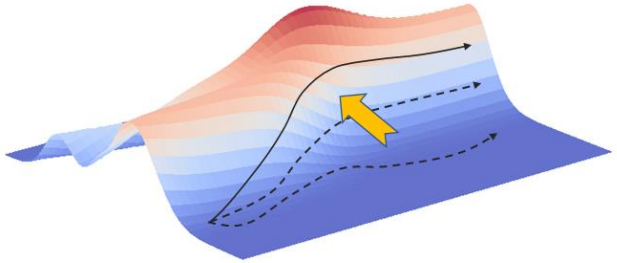
- 优化手段

$$\nabla_\theta J(\pi_\theta)$$

强化学习——策略优化

➢ 策略梯度学习方法

$$J(\pi)=\mathbb{E}_{\tau\sim p_{\pi}(\tau)}\left[\sum_{t=0}^{\infty}\gamma^{t}r(s_t,a_t,s_{t+1})\right]$$



$$\tau=\{s_0,a_1;s_1,a_1;s_2,a_2;\ldots\}$$

$$R(\tau)=\sum_{t=0}^{\infty}\gamma^{t}r(s_t,a_t,s_{t+1})$$

累计奖励

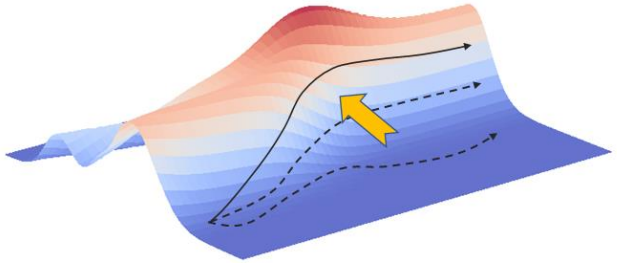
$$J(\pi_\theta)=\mathbb{E}_{\tau\sim p_{\pi_\theta}(\tau)}\left(R(\tau)\right)=\int_{\tau}R(\tau)\cdot p_{\pi_\theta}(\tau)d\tau$$

数学期望

强化学习——策略优化

➢ 策略梯度学习方法

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim p_{\pi_\theta}(\tau)}\left(R(\tau)\right) = \int_\tau R(\tau) \cdot p_{\pi_\theta}(\tau) d\tau$$



$$\tau = \{s_0, a_1; s_1, a_1; s_2, a_2; \ldots\}$$

$$\nabla_\theta J(\pi_\theta) = \int_\tau R(\tau) \cdot \nabla_\theta p_{\pi_\theta}(\tau) d\tau$$

为了写成数学期望形式

$$\nabla_\theta J(\pi_\theta) = \int_\tau R(\tau) \cdot \frac{\nabla_\theta p_{\pi_\theta}(\tau)}{p_{\pi_\theta}(\tau)} \cdot p_{\pi_\theta}(\tau) d\tau$$

$$= \int_\tau R(\tau) \cdot \nabla_\theta \log p_{\pi_\theta}(\tau) \cdot p_{\pi_\theta}(\tau) d\tau$$

对数-导数技巧

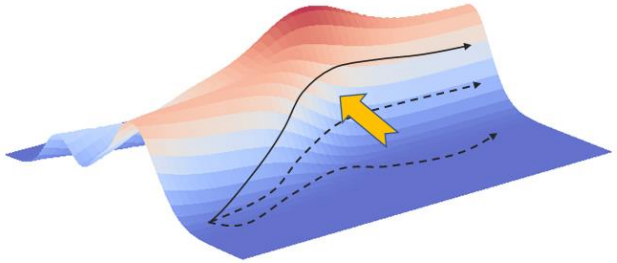
$$\nabla f(x) = f(x) \nabla \log f(x)$$

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim p_{\pi_\theta}(\tau)}\left(R(\tau) \cdot \nabla_\theta \log p_{\pi_\theta}(\tau)\right)$$

强化学习——策略优化

➢ 策略梯度学习方法

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim p_{\pi_\theta}(\tau)} \left(R(\tau) \cdot \nabla_\theta \log p_{\pi_\theta}(\tau) \right)$$



$$\tau = \{s_0, a_1; s_1, a_1; s_2, a_2; \ldots\}$$

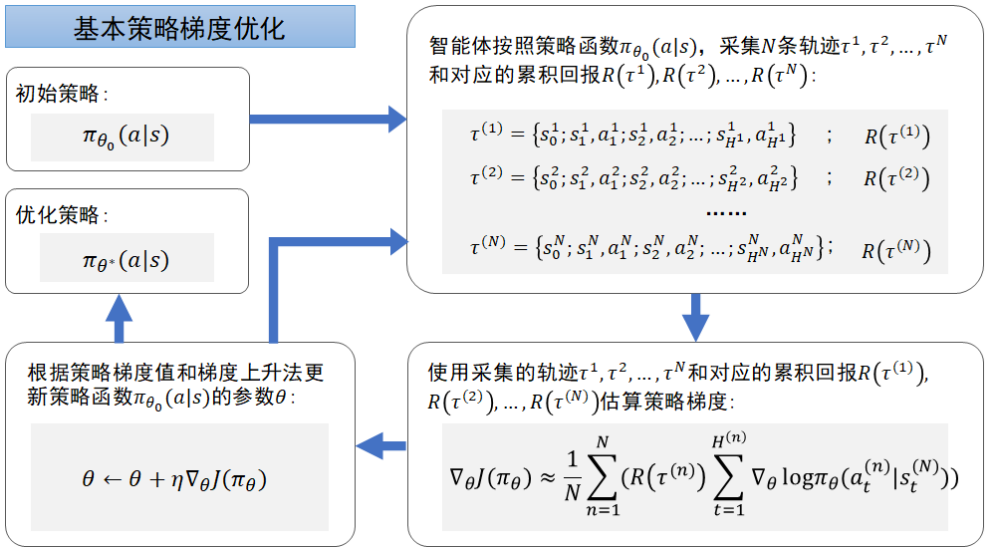
$$\nabla_\theta \log p_{\pi_\theta}(\tau) = \sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t \mid s_t) \quad \longleftarrow \quad p_\pi(\tau) = p_0(s_0) \prod_{t=0}^{\infty} \pi(a_t \mid s_t) T(s_{t+1} \mid s_t, a_t)$$

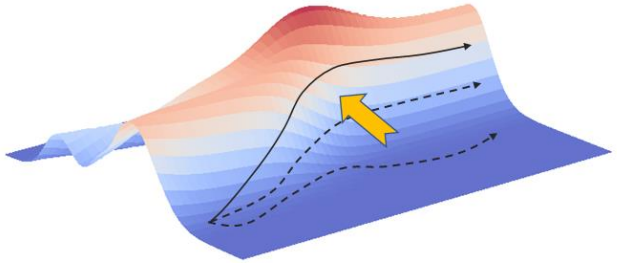
$$\nabla_\theta J(\pi_\theta) \approx \frac{1}{N} \sum_{n=1}^{N} \left(R(\tau^{(n)}) \cdot \sum_{t=0}^{H^{(n)}} \nabla_\theta \log \pi_\theta(a_t^{(n)} \mid s_t^{(n)}) \right)$$

$$\nabla_\theta J(\pi_\theta) \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{t=0}^{H^{(n)}} R(\tau^{(n)}) \nabla_\theta \log \pi_\theta(a_t^{(n)} \mid s_t^{(n)})$$

强化学习——策略优化

➢ 策略梯度学习方法





$$\tau=\{s_0,a_1;s_1,a_1;s_2,a_2;\dots\}$$

算法 3.4：基本策略梯度算法

目标：最优策略参数 θ^*

Step 1：初始化：令 $k=0$。初始化 θ 为随机参数。

Step 2：$k=k+1$。

Step 3：让智能体按照该策略完整地运行 N 个回合

利用策略分布 $p_{\pi_\theta}(\tau)$ 采集 N 条轨迹 $\tau^{(n)}\sim p_{\pi_\theta}(\tau)$，记作：

$\tau^{(n)}=\{s_0^{(n)},a_0^{(n)};s_1^{(n)},a_1^{(n)};s_2^{(n)},a_2^{(n)};\dots;s_{H^{(n)}}^{(n)},a_{H^{(n)}}^{(n)}\}$，$n=1,2,\dots,N$

Step 4：计算梯度 $\nabla_\theta J(\pi_\theta)\approx\frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{H^{(n)}}R(\tau^{(n)})\nabla_\theta\log\pi_\theta(a_t^{(n)}\mid s_t^{(n)})$

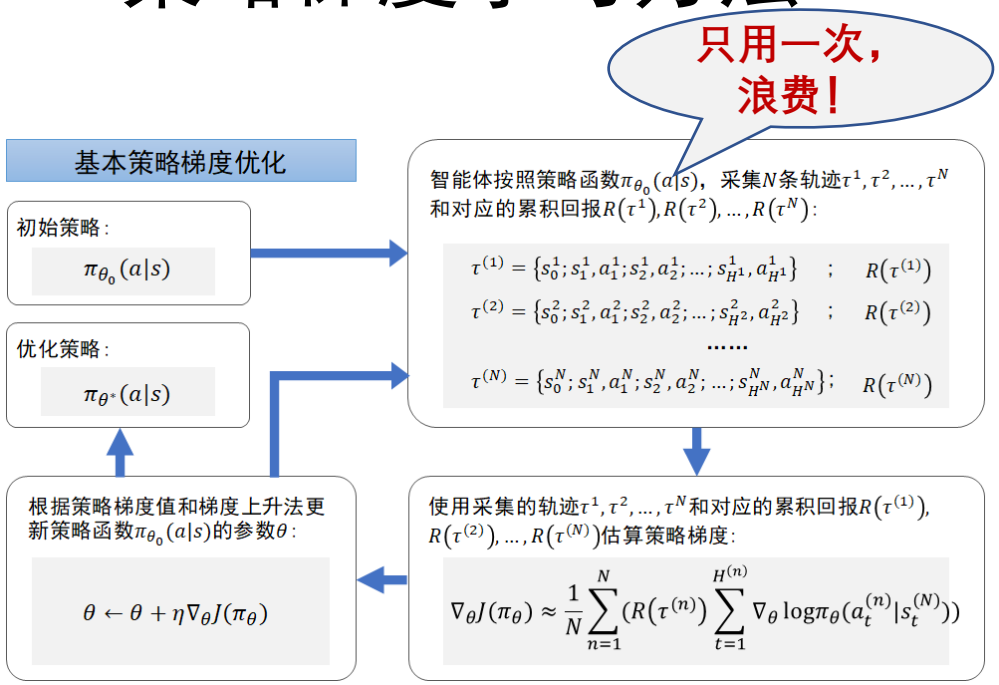
Step 5：更新参数 $\theta\leftarrow\theta+\eta\nabla_\theta J(\pi_\theta)$

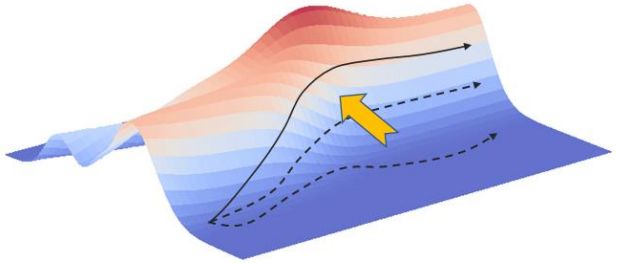
Step 6：如果 $k=\text{MAX\_ITE}$，令 $\theta^*=\theta$，退出循环；

否则，返回 Step 2。

强化学习——策略优化

➢ 策略梯度学习方法





$$\tau=\{s_0,a_1;s_1,a_1;s_2,a_2;\ldots\}$$

算法 3.4：基本策略梯度算法

目标：最优策略参数θ^*

Step 1：初始化：令$k=0$。初始化θ为随机参数。

Step 2：$k=k+1$。

Step 3：让智能体按照该策略完整地运行N个回合

利用策略分布$p_{\pi_\theta}(\tau)$采集N条轨迹$\tau^{(n)}\sim p_{\pi_\theta}(\tau)$，记作：

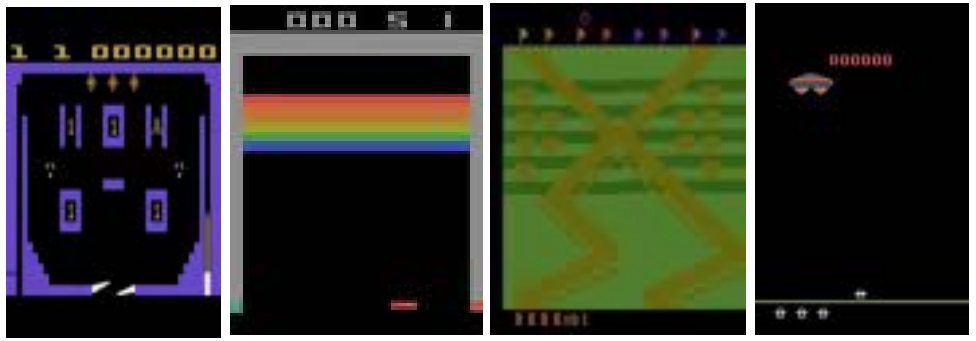
$\tau^{(n)}=\{s_0^{(n)},a_0^{(n)};s_1^{(n)},a_1^{(n)};s_2^{(n)},a_2^{(n)};\ldots;s_{H^{(n)}}^{(n)},a_{H^{(n)}}^{(n)}\}$，$n=1,2,\ldots,N$

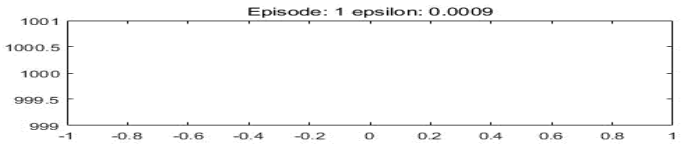
Step 4：计算梯度$\nabla_\theta J(\pi_\theta)\approx\frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{H^{(n)}}R(\tau^{(n)})\nabla_\theta\log\pi_\theta(a_t^{(n)}|s_t^{(n)})$

Step 5：更新参数$\theta\leftarrow\theta+\eta\nabla_\theta J(\pi_\theta)$

Step 6：如果$k=$MAX_ITE，令$\theta^*=\theta$，退出循环；

否则，返回 Step 2。



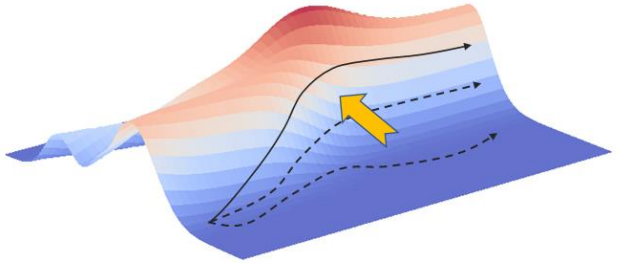


强化学习——策略梯度学习

➢ 策略梯度学习方法改进

$$J(\pi_\theta)=\mathbb{E}_{\tau\sim p_{\pi\theta}(\tau)}\left(R(\tau)\right)=\int_\tau R(\tau)\cdot p_{\pi_\theta}(\tau)d\tau$$

$$\nabla_\theta J(\pi_\theta)=\mathbb{E}_{\tau\sim p_{\pi\theta}(\tau)}\left(R(\tau)\cdot\nabla_\theta \log p_{\pi_\theta}(\tau)\right)$$



$$\tau=\{s_0,a_1;s_1,a_1;s_2,a_2;\ldots\}$$

权重$R(\tau)$直接影响梯度的更新量

- **水涨船高**：如果计算所得的轨迹奖励数都较大，则对应参数的更新量也较大，优化过程的波动会比较大。
- **通货膨胀**：如果奖励值都是正，那么策略优化过程中各动作出现的概率都会上升，即使是效果不好的动作，其概率也会得到提升（只不过提升的幅度小一点）。这使得好的动作与差的动作之间难以拉开差距。

↓

剔除掉冗余、公共、不必要的部分

强化学习——策略梯度学习

➢ REINFORCE

$$J(\pi)=\mathbb{E}_{\tau\sim p_\pi(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t,s_{t+1})\right]$$

值函数

$$J(\pi_\theta)=\mathbb{E}_{s\sim p_0(s)}\left(V^{\pi_\theta}(s)\right)$$

V函数与Q函数的关系 $V^{\pi_\theta}(s)=\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(Q^{\pi_\theta}(s,a)\right)$

$$J(\pi_\theta)=\mathbb{E}_{s\sim p_0(s)}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(Q^{\pi_\theta}(s,a)\right)$$

$\nabla_\theta J(\pi_\theta)=?$

策略梯度定理

$$\nabla_\theta J(\pi_\theta)=\mathbb{E}_{s\sim p_0(s)}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(Q^{\pi_\theta}(s,a)\cdot\nabla_\theta\log\pi_\theta(a\,|\,s)\right)$$

$$g_\theta(s,a)=Q^{\pi_\theta}(s,a)\cdot\nabla_\theta\log\pi_\theta(a\,|\,s)$$

$$\nabla_\theta J(\pi_\theta)=\mathbb{E}_{s\sim p_0(s)}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(g_\theta(s,a)\right)$$

强化学习——策略梯度学习

➢ REINFORCE

$$g_\theta(s,a) = Q^{\pi_\theta}(s,a) \cdot \nabla_\theta \log \pi_\theta(a \mid s)$$

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim p_0(s)} \mathbb{E}_{a\sim \pi_\theta(\cdot|s)} \left(g_\theta(s,a) \right)$$

可以将 $g_\theta(s,a)$作为$\nabla_\theta J(\pi_\theta)$的无偏估计。

强化学习——策略梯度学习

➢ REINFORCE

$$g_\theta(s,a) = Q^{\pi_\theta}(s,a) \cdot \nabla_\theta \log \pi_\theta(a \mid s)$$

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim p_0(s)} \mathbb{E}_{a\sim \pi_\theta(\cdot|s)} \left(g_\theta(s,a) \right)$$

可以将 $g_\theta(s,a)$作为$\nabla_\theta J(\pi_\theta)$的无偏估计。

由于 $Q^{\pi_\theta}(s,a)$并不知道，因此需要对其做近似估计。

RENFORCE算法就是利用实际获得的奖励函数来估计

$$Q^{\pi}(s_t,a_t) = \mathbb{E}\left(\sum_{t'=t}^{+\infty} \gamma^{t'-t} r(s_{t'}, a_{t'}, s_{t'+1}) \middle| s_t, a_t \right)$$

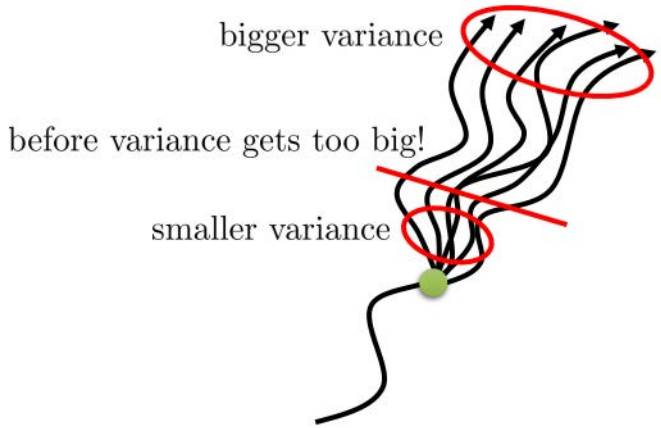
强化学习——策略梯度学习

➢ REINFORCE：水涨船高

$$g_\theta(s,a) = Q^{\pi_\theta}(s,a)\cdot\nabla_\theta \log \pi_\theta(a\,|\,s)$$

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim p_0(s)}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(g_\theta(s,a)\right)$$

可以将 $g_\theta(s,a)$作为$\nabla_\theta J(\pi_\theta)$的无偏估计。



由于$Q^{\pi_\theta}(s,a)$并不知道，因此需要对其做近似估计。

RENFORCE算法就是利用实际获得的奖励函数来估计

$$Q^{\pi}(s_t,a_t) = \mathbb{E}\left(\sum_{t'=t}^{+\infty}\gamma^{t'-t}r(s_{t'},a_{t'},s_{t'+1})\,\middle|\,s_t,a_t\right)$$

$$\bar{R}_t(\tau) = \sum_{t'=t}^{+\infty}\gamma^{t'-t}r(s_{t'},a_{t'},s_{t'+1})$$

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim p_{\pi_\theta}(\tau)}\left(R(\tau)\cdot\nabla_\theta \log p_{\pi_\theta}(\tau)\right)$$

$$\bar{R}_t(\tau)\cdot\nabla_\theta \log\pi_\theta(a_t\,|\,s_t) \longrightarrow \nabla_\theta J(\pi_\theta)$$ 无偏估计

强化学习——策略梯度学习

➢ REINFORCE：通货膨胀

策略梯度定理

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim p_0(s)}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(Q^{\pi_\theta}(s,a)\cdot\nabla_\theta \log\pi_\theta(a\,|\,s)\right)$$

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim p_0(s)}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left((Q^{\pi_\theta}(s,a)-b)\cdot\nabla_\theta \log\pi_\theta(a\,|\,s)\right)$$

其中*b*为与动作*a*无关的基线函数

- 引入这一基线的目的相当于将所有权重都降低一个公共值
- 对于小于基线b的Q函数，可以得到负的权重，从而显著拉开不同动作权重的差距，进一步减小估计的方差。
- 在REINFORCE算法中，一般可以将基线函数取为：

$$b = \mathbb{E}_{\tau\sim p_{\pi_\theta}(\tau)}\left(\sum_{t'=t}^{\infty}\gamma^{t'-t}r(s_{t'},a_{t'},s_{t'+1})\right)$$

求证：$\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(b\cdot\nabla_\theta\log\pi_\theta(a\,|\,s)\right)=0$

证明：由于 b 为与动作 a 无关，我们可以将上式写为

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(b\cdot\nabla_\theta\log\pi_\theta(a\,|\,s)\right)=b\cdot\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(\nabla_\theta\log\pi_\theta(a\,|\,s)\right)$$

按照数学期望的定义，我们有：

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(\nabla_\theta\log\pi_\theta(a\,|\,s)\right)=\sum_a\pi_\theta(a\,|\,s)\nabla_\theta\log\pi_\theta(a\,|\,s)$$

按照“对数-导数技巧”，我们有 $\pi_\theta(a\,|\,s)\nabla_\theta\log\pi_\theta(a\,|\,s)=\nabla_\theta\pi_\theta(a\,|\,s)$。从而可得：

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(\nabla_\theta\log\pi_\theta(a\,|\,s)\right)=\sum_a\nabla_\theta\pi_\theta(a\,|\,s)=\nabla_\theta\sum_a\pi_\theta(a\,|\,s)$$

考虑到全概率 $\sum_a\pi_\theta(a\,|\,s)=1$，可得 $\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(\nabla_\theta\log\pi_\theta(a\,|\,s)\right)=0$。因而可得：

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(b\cdot\nabla_\theta\log\pi_\theta(a\,|\,s)\right)=0$$

强化学习——策略梯度学习

➢ REINFORCE

算法 3.4：基本策略梯度算法

目标：最优策略参数 θ^*

Step 1：初始化：令 $k=0$。初始化 θ 为随机参数。

Step 2：$k=k+1$。

Step 3：让智能体按照该策略完整地运行 N 个回合

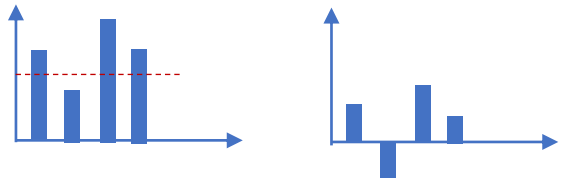
利用策略分布 $p_{\pi_\theta}(\tau)$ 采集 N 条轨迹 $\tau^{(n)} \sim p_{\pi_\theta}(\tau)$，记作：

$$\tau^{(n)} = \{s_0^{(n)}, a_0^{(n)}; s_1^{(n)}, a_1^{(n)}; s_2^{(n)}, a_2^{(n)}; \ldots; s_{H^{(n)}}^{(n)}, a_{H^{(n)}}^{(n)}\}，\ n = 1, 2, \ldots, N$$

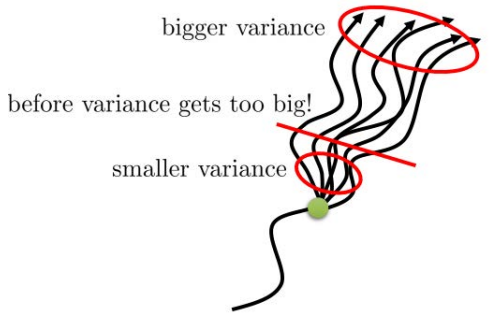
Step 4：计算梯度 $\nabla_\theta J(\pi_\theta) \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{H^{(n)}} R(\tau^{(n)})\nabla_\theta \log \pi_\theta(a_t^{(n)} \mid s_t^{(n)})$

Step 5：更新参数 $\theta \leftarrow \theta + \eta\nabla_\theta J(\pi_\theta)$

Step 6：如果 $k = \text{MAX\_ITE}$，令 $\theta^* = \theta$，退出循环；
否则，返回 Step 2。



通货膨胀（增加区分度）



水涨船高
（忘记过去）

算法 3.5：REINFORCE 算法

目标：最优策略参数 θ^*

Step 1：初始化：令 $k=0$。初始化 θ 为随机参数。

Step 2：$k=k+1$。

Step 3：让智能体按照该策略完整地运行 N 个回合

利用策略分布 $p_{\pi_\theta}(\tau)$ 采集 N 条轨迹 $\tau^{(n)} \sim p_{\pi_\theta}(\tau)$，记作：

$$\tau^{(n)} = \{s_1^{(n)}, a_1^{(n)}; s_2^{(n)}, a_2^{(n)}; \ldots; s_{H^{(n)}}^{(n)}, a_{H^{(n)}}^{(n)}\}，\ n = 1, 2, \ldots, N$$

Step 4：计算 $\bar{R}_t^{(n)} = \sum_{t'=t}^{H^{(n)}} \gamma^{t'-t} r(s_{t'}^{(n)}, a_{t'}^{(n)}, s_{t'+1}^{(n)})$

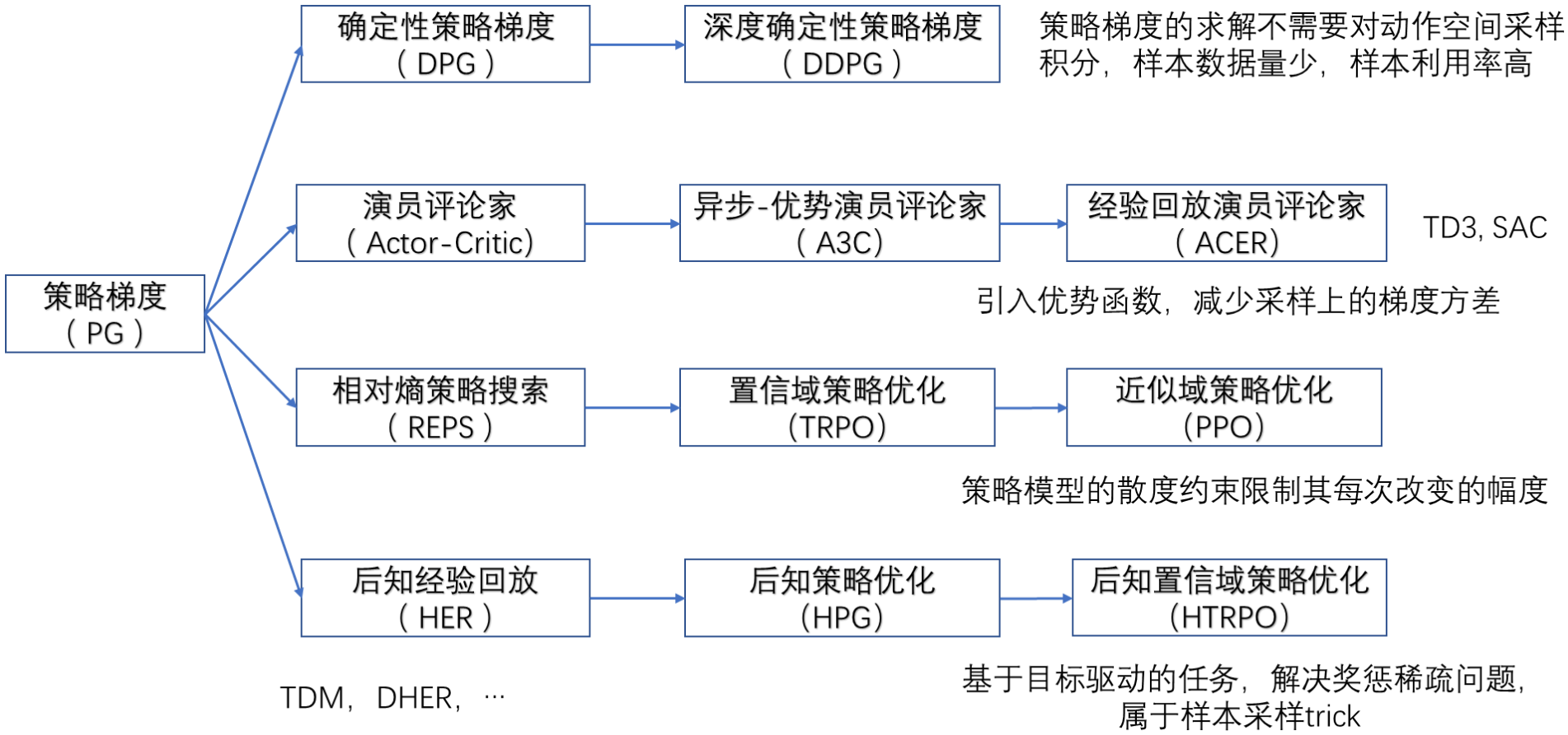
Step 5：拟合基线：$b_t = \frac{1}{N}\sum_{n=1}^{N} R_t^{(n)}$

Step 6：计算梯度 $\nabla_\theta J(\pi_\theta) \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{H^{(n)}} (\bar{R}_t^{(n)} - b_t)\nabla_\theta \log \pi_\theta(a_t^{(n)} \mid s_t^{(n)})$

Step 7：更新参数 $\theta \leftarrow \theta + \eta\nabla_\theta J(\pi_\theta)$

Step 8：如果 $k = \text{MAX\_ITE}$，令 $\theta^* = \theta$，退出循环；
否则，返回 Step 2。

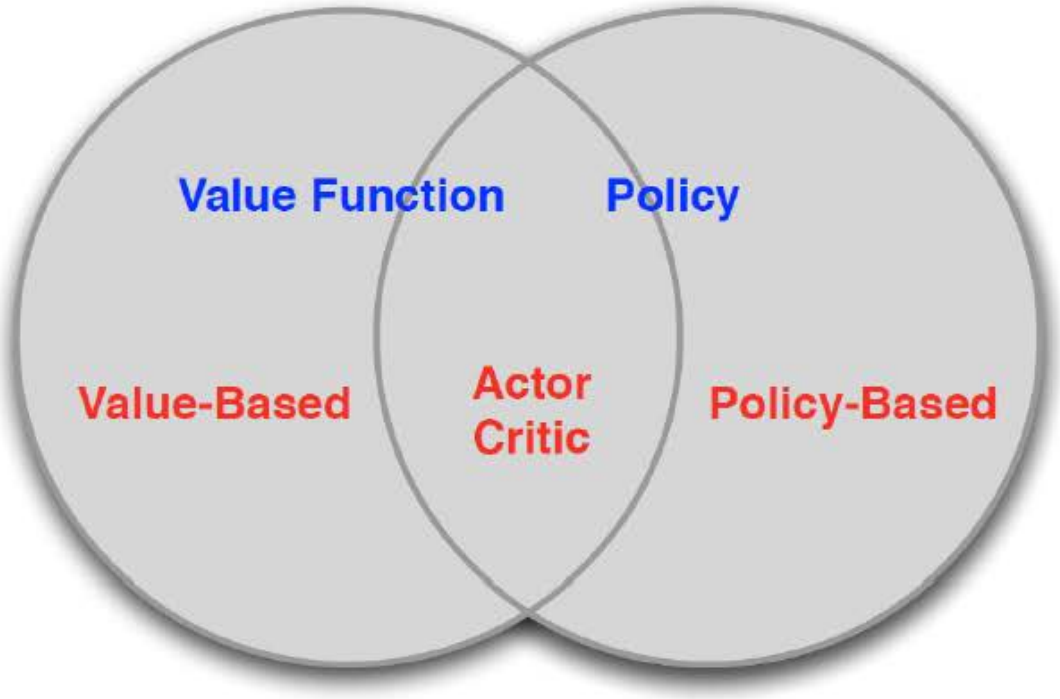
强化学习——策略梯度学习


策略梯度
（PG）
确定性策略梯度
（DPG）
深度确定性策略梯度
（DDPG）
策略梯度的求解不需要对动作空间采样积分，样本数据量少，样本利用率高
演员评论家
（Actor-Critic）
异步-优势演员评论家
（A3C）
经验回放演员评论家
（ACER）
TD3, SAC
引入优势函数，减少采样上的梯度方差
相对熵策略搜索
（REPS）
置信域策略优化
（TRPO）
近似域策略优化
（PPO）
策略模型的散度约束限制其每次改变的幅度
后知经验回放
（HER）
后知策略优化
（HPG）
后知置信域策略优化
（HTRPO）
TDM，DHER，…
基于目标驱动的任务，解决奖惩稀疏问题，
属于样本采样trick

强化学习——策略梯度学习

➢ Actor-Critic

- Value Based
  - Learning Value Function
  - Implicit Policy (e.g., ε-greedy)
- Policy Based
  - No Value Function
  - Directly learning policy
- Actor-Critic
  - Learning Value Function
  - Learning Policy



强化学习——Actor-Critic

➢ 基本Actor-Critic

- **REINFORCE**

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim p_0(s)}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(Q^{\pi_\theta}(s,a)\cdot\nabla_\theta \log\pi_\theta(a\mid s)\right)$$

$$Q^{\pi}(s_t,a_t) = \mathbb{E}\left(\sum_{t'=t}^{+\infty}\gamma^{t'-t}r(s_{t'},a_{t'},s_{t'+1})\middle| s_t,a_t\right)$$

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim p_0(s)}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(Q^{\pi_\theta}(s,a)\cdot\nabla_\theta \log\pi_\theta(a\mid s)\right)$$

- **Actor-Critic**

$$\hat{Q}_\omega(s,a)\approx Q^{\pi_\theta}(s,a)$$

$$\nabla_\theta J(\pi_\theta) = \hat{Q}_\omega(s_t,a_t)\cdot\nabla_\theta \log\pi_\theta(a_t\mid s_t)$$

强化学习——Actor-Critic

➢ 基本Actor-Critic

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim p_0(s)}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left(Q^{\pi_\theta}(s,a)\cdot\nabla_\theta \log\pi_\theta(a\mid s)\right)$$

$$\hat{Q}_\omega(s,a)\approx Q^{\pi_\theta}(s,a) \qquad \nabla_\theta J(\pi_\theta)=\hat{Q}_\omega(s_t,a_t)\cdot\nabla_\theta\log\pi_\theta(a_t\mid s_t)$$

算法 3.6：Actor-Critic 策略梯度算法

目标：最优策略参数 θ^*，最优值函数参数 ω^*

Step 1：初始化：令 $k=0$，$t=0$，初始化 θ、ω 为随机参数。

Step 2：$k=k+1$。

Step 3：$t=t+1$，让智能体运行策略 π_θ 一步，获得四元组 $\{s_t,a_t,r_t,s_{t+1}\}$。

Step 4：计算 $\nabla_\theta J(\pi_\theta)=\hat{Q}_\omega(s_t,a_t)\cdot\nabla_\theta\log\pi_\theta(a_t\mid s_t)$

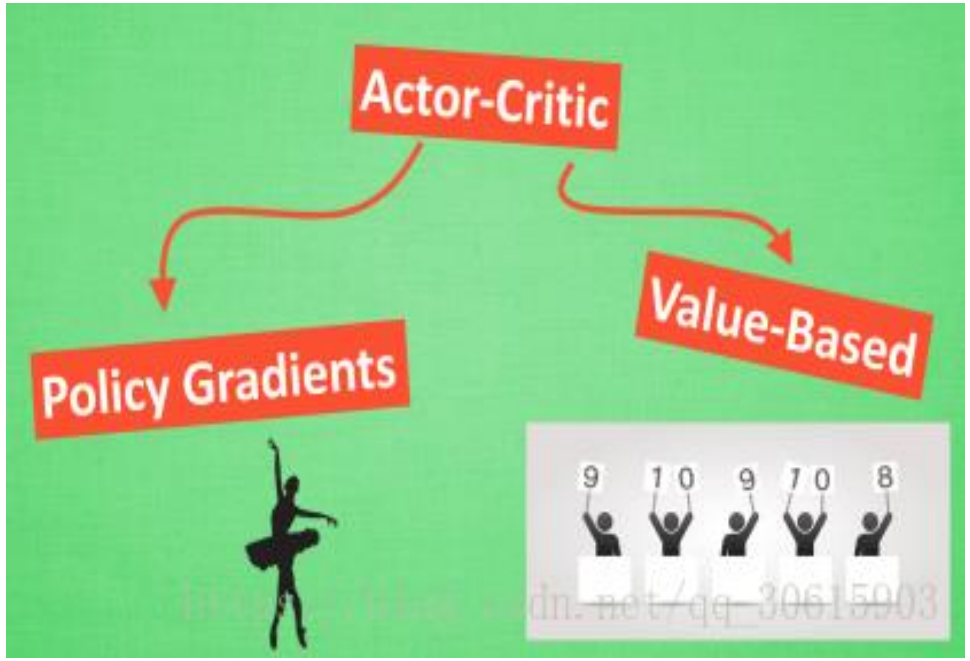
Step 5：计算 $\nabla_\omega L(\omega)=\left(\hat{Q}_\omega(s_t,a_t)-\left(r_t+\gamma\cdot\hat{Q}_\omega(s_{t+1},a_{t+1})\right)\right)\cdot\nabla_\omega\hat{Q}_\omega(s_t,a_t)$

Step 6：更新策略参数 $\theta\leftarrow\theta+\eta_\theta\cdot\nabla_\theta J(\pi_\theta)$

Step 7：更新策略参数 $\omega\leftarrow\omega-\eta\nabla_\omega L(\omega)$

Step 8：如果 t 不是终止时刻，返回 Step 3。

Step 9：如果 $k=$MAX_ITE，令 $\theta^*=\theta$，$\omega^*=\omega$，退出循环；
否则，返回 Step 2。



强化学习——Actor-Critic

➢ 改进Actor-Critic

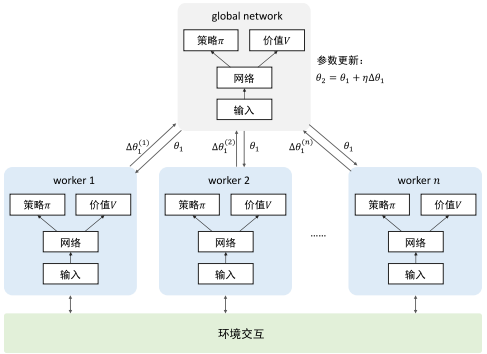
A2C

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim p_0(s)}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left((Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s))\cdot\nabla_\theta \log\pi_\theta(a\mid s)\right)$$

$$\hat{g}_\theta(s,a) = \left(\hat{Q}_\omega(s,a) - V^{\pi_\theta}(s)\right)\cdot\nabla_\theta \log\pi_\theta(a\mid s)$$

Advantage Actor-Critic

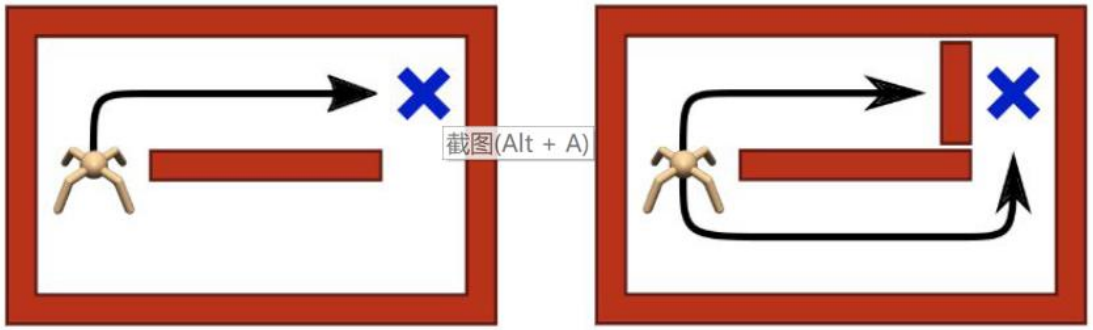
A3C



Asynchronous Advantage Actor-Critic

强化学习——Actor-Critic

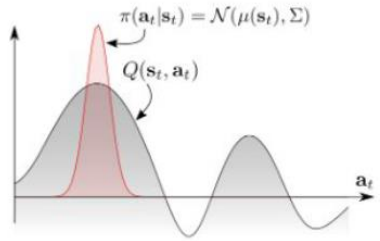
- Soft Actor-Critic

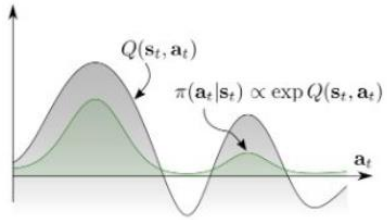


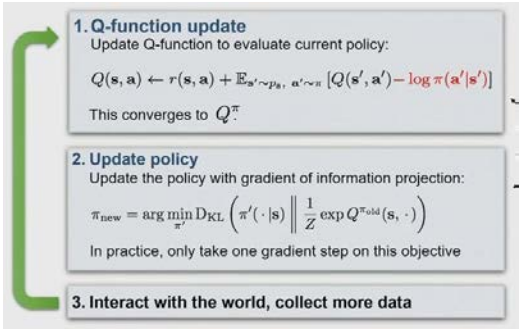
$$\sum_t \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\rho_\pi}\left[r(\mathbf{s}_t,\mathbf{a}_t) + \alpha\mathcal{H}(\pi(\cdot|\mathbf{s}_t))\right]$$

Entropy of a RV x

$$H(P) = \mathop{\mathrm{E}}_{x\sim P}[-\log P(x)]$$







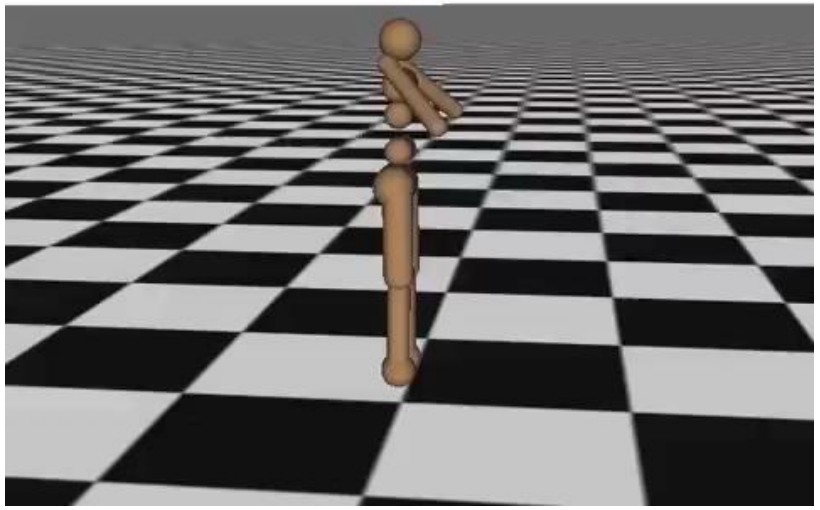
强化学习——Actor-Critic

$$\sum_t \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\rho_\pi}\left[r(\mathbf{s}_t,\mathbf{a}_t)+\alpha\mathcal{H}(\pi(\cdot|\mathbf{s}_t))\right]$$

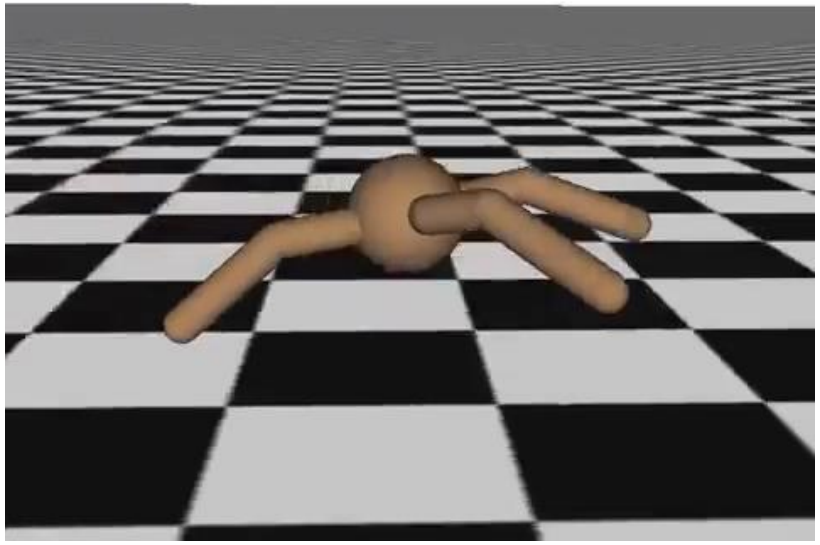
Entropy of a RV x

$$H(P)=\mathop{\mathrm{E}}_{x\sim P}[-\log P(x)]$$

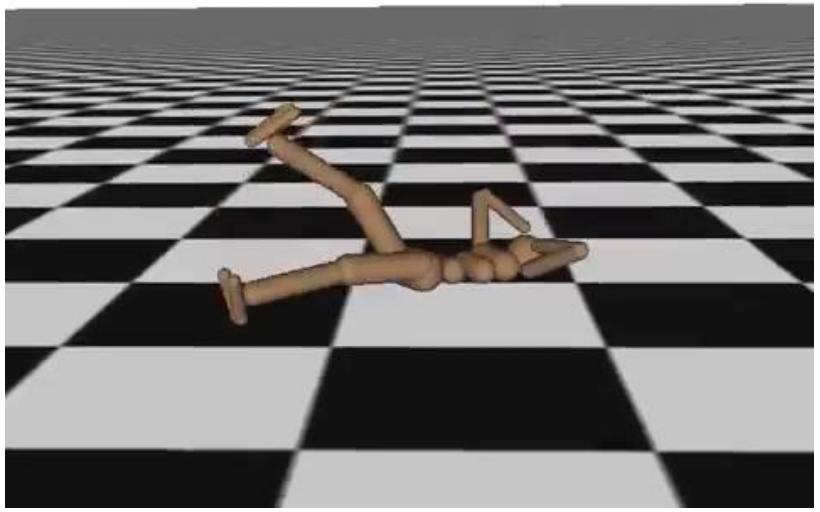
Iteration 0



Iteration 80



Iteration 10



强化学习——PPO

➢ TRPO：Trust Region Policy Optimization

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim p_0(s)}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left((Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s))\cdot\nabla_\theta \log\pi_\theta(a\mid s)\right)$$

优势函数 $A^{\pi_\theta}(s_t,a_t) = Q^{\pi_\theta}(s_t,a_t) - V^{\pi_\theta}(s_t)$

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim p_{\pi_\theta}(\tau)}\left(A^{\pi_\theta}(s_t,a_t)\cdot\sum_{t=1}^{\infty}\nabla_\theta \log\pi_\theta(a_t\mid s_t)\right)$$

$$J(\pi_\theta) = J(\pi_{\theta_{\text{old}}}) + \mathbb{E}_{\tau\sim p_{\pi_\theta}(\tau)}\left(\sum_{t=0}^{\infty}\gamma^t A^{\pi_{\theta_{\text{old}}}}(s_t,a_t)\right)$$

考虑到 $A^{\pi}(s_t,a_t) = \mathbb{E}_{s'\sim P(s'|s,a)}\left(r(s,a,s') + \gamma V^{\pi}(s') - V^{\pi}(s)\right)$，则有：

$$\mathbb{E}_{\tau\sim p_{\pi_\theta}(\tau)}\left(\sum_{t=0}^{\infty}\gamma^t A^{\pi_{\theta_{\text{old}}}}(s_t,a_t)\right) = \mathbb{E}_{\tau\sim p_{\pi_\theta}(\tau)}\left(\sum_{t=0}^{\infty}\gamma^t\left(r(s_t,a_t,s_{t+1}) + \gamma V^{\pi_{\theta_{\text{old}}}}(s_{t+1}) - V^{\pi_{\theta_{\text{old}}}}(s_t)\right)\right)$$

$$= \mathbb{E}_{\tau\sim p_{\pi_\theta}(\tau)}\left(\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t,s_{t+1}) - V^{\pi_{\theta_{\text{old}}}}(s_0)\right)$$

$$= \mathbb{E}_{\tau\sim p_{\pi_\theta}(\tau)}\left(\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t,s_{t+1})\right) - \mathbb{E}_{s_0\sim p(s_0)}\left(V^{\pi_{\theta_{\text{old}}}}(s_0)\right)$$

$$= J(\pi_\theta) - J(\pi_{\theta_{\text{old}}})$$

强化学习——PPO

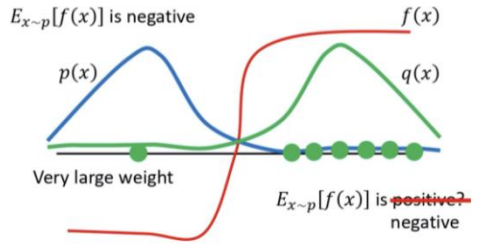
➢ TRPO

$$J(\pi_\theta) = J(\pi_{\theta_{\text{old}}}) + \mathbb{E}_{\tau \sim p_{\pi_\theta}(\tau)} \left(\sum_{t=0}^{\infty} \gamma^t A^{\pi_{\theta_{\text{old}}}}(s_t, a_t) \right)$$

重要性采样

$$\mathbb{E}_{\tau \sim p_{\pi_\theta}(\tau)} \left(\sum_{t=0}^{\infty} \gamma^t A^{\pi_{\theta_{\text{old}}}}(s_t, a_t) \right)$$

$$E_{x\sim p}[f(x)] = E_{x\sim q}[f(x)\frac{p(x)}{q(x)}]$$



$$\max L_{\pi_{\theta_{\text{old}}}}(\pi_\theta) = \mathbb{E}_{\tau \sim p_{\pi_{\theta_{\text{old}}}}(\tau)} \left(\sum_{t=0}^{\infty} \gamma^t \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} A^{\pi_{\theta_{\text{old}}}}(s_t, a_t) \right)$$

$$\mathbb{E}_{\tau \sim p_{\pi_{\theta_{\text{old}}}}(\tau)} \left(KL(\pi_{\theta_{\text{old}}}(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s)) \right) \le \delta$$

KL 散度：

- KL 散度是 Kullback-Leibler Divergence 的通俗化称谓，也叫做相对熵（Relative Entropy）。它是用于衡量两个概率分布差异的常用度量，但并是一种真正的距离度量（因为不满足对称性）。
- 概率 $P(x)$和 $Q(x)$的 KL 散度计算如下：

$$\text{KL}(P|Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

强化学习——PPO

- PPO：Proximal Policy Optimization

$$\max_{\pi_\theta} \quad L_{\pi_{\theta_{\text{old}}}}(\pi_\theta)$$

$$\text{s.t.} \quad \mathbb{E}_{\tau \sim p_{\pi_{\theta_{\text{old}}}}(\tau)}\left(KL(\pi_{\theta_{\text{old}}}(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s))\right) \le \delta$$

$$\rho_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$$

$$L_{\pi_{\theta_{\text{old}}}}(\pi_\theta) = \mathbb{E}_{\tau \sim p_{\pi_{\theta_{\text{old}}}}(\tau)}\left(\sum_{t=0}^{\infty} \gamma^t \rho_t(\theta) A^{\pi_{\theta_{\text{old}}}}(s_t, a_t)\right)$$

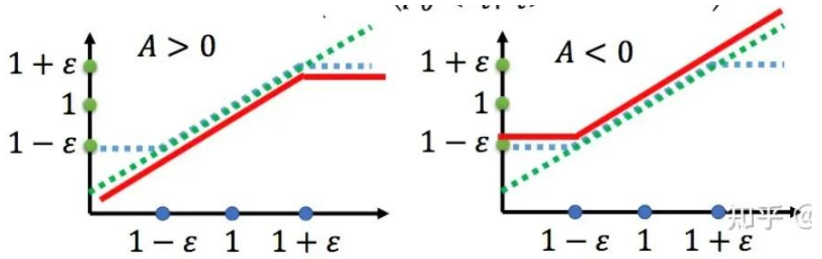
强化学习——PPO

➢ PPO

$$L_{\pi_{\theta_{\text{old}}}}(\pi_\theta)=\mathbb{E}_{\tau\sim p_{\pi_{\theta_{\text{old}}}}(\tau)}\left(\sum_{t=0}^{\infty}\gamma^t\rho_t(\theta)A^{\pi_{\theta_{\text{old}}}}(s_t,a_t)\right)$$

$$L_{\pi_{\theta_{\text{old}}}}(\pi_\theta)=\mathbb{E}_{\tau\sim p_{\pi_{\theta_{\text{old}}}}(\tau)}\left(\sum_{t=0}^{\infty}\gamma^t\ \min\left\{(\rho_t(\theta)A^{\pi_{\theta_{\text{old}}}}(s_t,a_t),\text{clip}(\rho_t(\theta),1-\varepsilon,1+\varepsilon)A^{\pi_{\theta_{\text{old}}}}(s_t,a_t)\right\}\right)$$

$$\text{clip}(\rho_t(\theta),1-\varepsilon,1+\varepsilon)=\begin{cases}\rho_t(\theta) & 1-\varepsilon\le\rho_t(\theta)\le 1+\varepsilon\\ 1-\varepsilon & \rho_t(\theta)<1-\varepsilon\\ 1+\varepsilon & \rho_t(\theta)>1+\varepsilon\end{cases}$$



PPO相当于加了两个进一步的约束

- 一方面，将两个策略之间的比例$\rho_t(\theta$限制在1附近的区间
- 另一方面，通过min算子也进一步保证了策略之间的相似度。

强化学习——PPO

Trust Region Policy Optimization

John Schulman JOSCHU@EECS.BERKELEY.EDU
Sergey Levine SLEVINE@EECS.BERKELEY.EDU
Philipp Moritz PCMORITZ@EECS.BERKELEY.EDU
Michael Jordan JORDAN@CS.BERKELEY.EDU
Pieter Abbeel PABBEEL@CS.BERKELEY.EDU
University of California, Berkeley, Department of Electrical Engineering and Computer Sciences

Abstract

We describe an iterative procedure for optimizing policies, with guaranteed monotonic improvement. By making several approximations to the theoretically-justified procedure, we develop a practical algorithm, called Trust Region Policy Optimization (TRPO). This algorithm is similar to natural policy gradient methods and is effective for optimizing large nonlinear policies such as neural networks. Our experiments demonstrate its robust performance on a wide variety of tasks: learning simulated robotic swimming, hopping, and walking gaits; and playing Atari games using images of the screen as input. Despite its approximations that deviate from the theory, TRPO tends to give monotonic improvement, with little tuning of hyperparameters.

1 Introduction

Most algorithms for policy optimization can be classified into three broad categories: (1) policy iteration methods, which alternate between estimating the value function under the current policy and improving the policy (Bertsekas, 2005); (2) policy gradient methods, which use an estimator of the gradient of the expected return (total reward) obtained from sample trajectories (Peters & Schaal, 2008a) (and which, as we later discuss, have a close connection to policy iteration); and (3) derivative-free optimization methods, such as the cross-entropy method (CEM) and covari- (CMA), which treat the return as a be optimized in terms of the policy rincz, 2006).

e stochastic optimization methods MA are preferred on many prob- hieve good results while being sim- l implement. For example, while

International Conference on Machine 2015. JMLR: W&CP volume 37. Copy- s).

Tetris is a classic benchmark problem for approximate dynamic programming (ADP) methods, stochastic optimization methods are difficult to beat on this task (Gabillon et al., 2013). For continuous control problems, methods like CMA have been successful at learning control policies for challenging tasks like locomotion when provided with hand-engineered policy classes with low-dimensional parameterizations (Wampler & Popović, 2009). The inability of ADP and gradient-based methods to consistently beat gradient-free random search is unsatisfying, since gradient-based optimization algorithms enjoy much better sample complexity guarantees than gradient-free methods (Nemirovski, 2005). Continuous gradient-based optimization has been very successful at learning function approximators for supervised learning tasks with huge numbers of parameters, and extending their success to reinforcement learning would allow for efficient training of complex and powerful policies.

In this article, we first prove that minimizing a certain surrogate objective function guarantees policy improvement with non-trivial step sizes. Then we make a series of approximations to the theoretically-justified algorithm, yielding a practical algorithm, which we call trust region policy optimization (TRPO). We describe two variants of this algorithm: first, the *single-path* method, which can be applied in the model-free setting; second, the *vine* method, which requires the system to be restored to particular states, which is typically only possible in simulation. These algorithms are scalable and can optimize nonlinear policies with tens of thousands of parameters, which have previously posed a major challenge for model-free policy search (Deisenroth et al., 2013). In our experiments, we show that the same TRPO methods can learn complex policies for swimming, hopping, and walking, as well as playing Atari games directly from raw images.

2 Preliminaries

Consider an infinite-horizon discounted Markov decision process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, \rho_0, \gamma)$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the transition probability distri-

Proximal Policy Optimization Algorithms

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov
OpenAI
{joschu, filip, prafulla, alec, oleg}@openai.com

Abstract

We propose a new family of policy gradient methods for reinforcement learning, which alternate between sampling data through interaction with the environment, and optimizing a "surrogate" objective function using stochastic gradient ascent. Whereas standard policy gradient methods perform one gradient update per data sample, we propose a novel objective function that enables multiple epochs of minibatch updates. The new methods, which we call proximal policy optimization (PPO), have some of the benefits of trust region policy optimization (TRPO), but they are much simpler to implement, more general, and have better sample complexity (empirically). Our experiments test PPO on a collection of benchmark tasks, including simulated robotic locomotion and Atari game playing, and we show that PPO outperforms other online policy gradient methods, and overall strikes a favorable balance between sample complexity, simplicity, and wall-time.

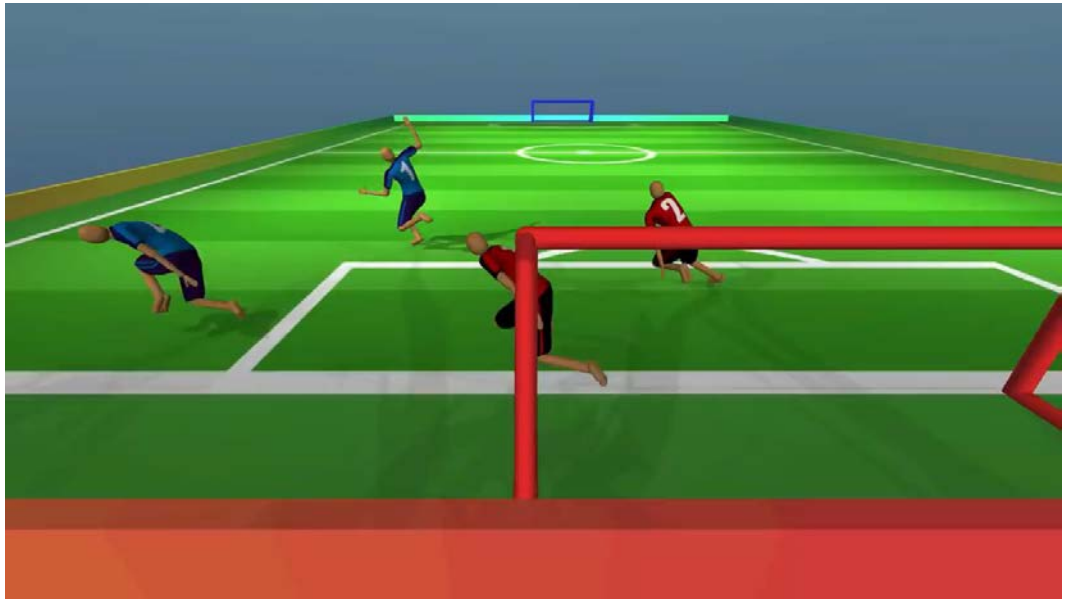
1 Introduction

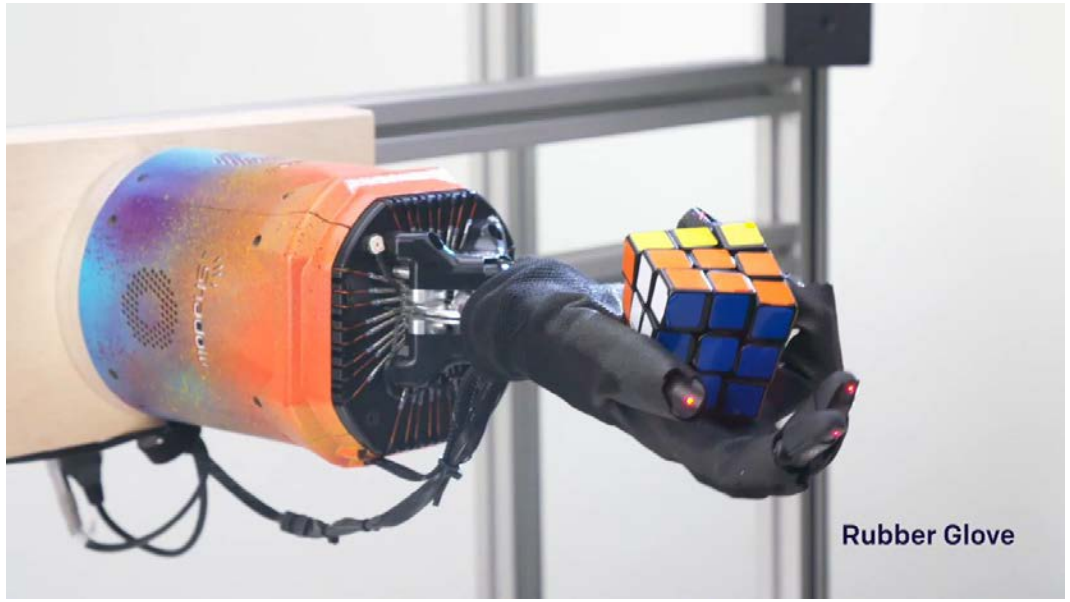
In recent years, several different approaches have been proposed for reinforcement learning with neural network function approximators. The leading contenders are deep Q-learning [Mni+15], "vanilla" policy gradient methods [Mni+16], and trust region / natural policy gradient methods [Sch+15b]. However, there is room for improvement in developing a method that is scalable (to large models and parallel implementations), data efficient, and robust (i.e., successful on a variety of problems without hyperparameter tuning). Q-learning (with function approximation) fails on many simple problems[1] and is poorly understood, vanilla policy gradient methods have poor data effiency and robustness; and trust region policy optimization (TRPO) is relatively complicated, and is not compatible with architectures that include noise (such as dropout) or parameter sharing (between the policy and value function, or with auxiliary tasks).

This paper seeks to improve the current state of affairs by introducing an algorithm that attains the data efficiency and reliable performance of TRPO, while using only first-order optimization. We propose a novel objective with clipped probability ratios, which forms a pessimistic estimate (i.e., lower bound) of the performance of the policy. To optimize policies, we alternate between sampling data from the policy and performing several epochs of optimization on the

Our experiments compare the performance of various different versions of the su tive, and find that the version with the clipped probability ratios performs best. We PPO to several previous algorithms from the literature. On continuous control tas better than the algorithms we compare against. On Atari, it performs significantly b of sample complexity) than A2C and similarly to ACER though it is much simpler.

[1] While DQN works well on game environments like the Arcade Learning Environment [Bel+ action spaces, it has not been demonstrated to perform well on continuous control benchmarks OpenAI Gym [Bro+16] and described by Duan et al. [Dua+16].




Rubber Glove

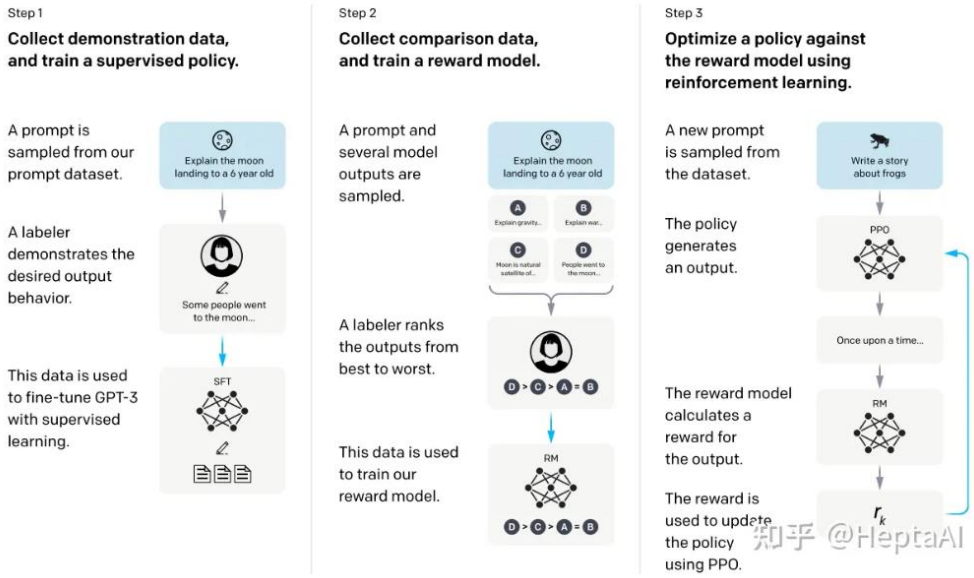
强化学习

➢ GPT

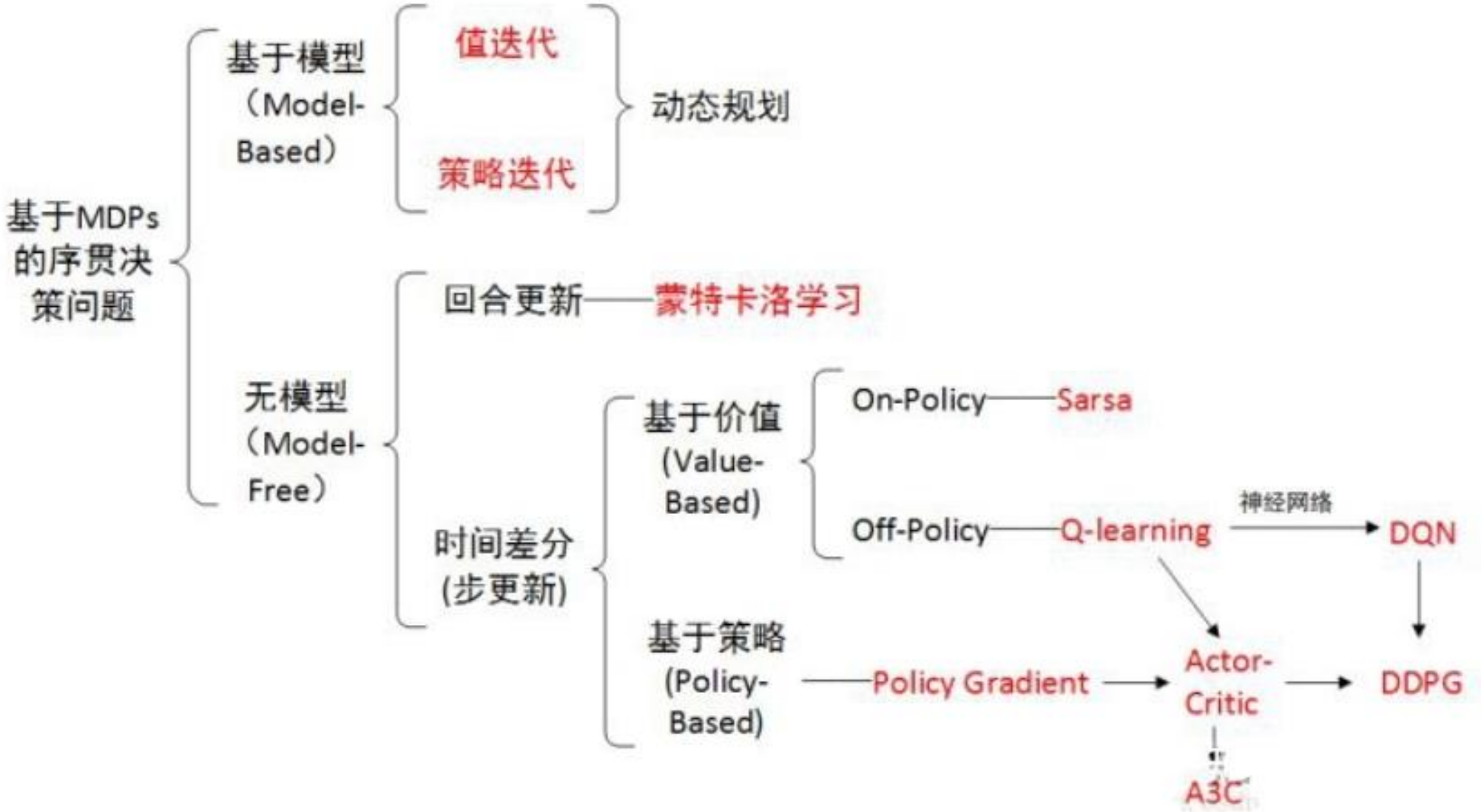
- GPT-1用的是无监督预训练+有监督微调。
- GPT-2用的是纯无监督预训练。
- GPT-3沿用了GPT-2的纯无监督预训练，但是数据大了好几个量级。
- InstructGPT在GPT-3上用强化学习做微调，内核模型为PPO-ptx。
- ChatGPT沿用了InstructGPT，但是数据大了好几个量级。

Training language models to follow instructions with human feedback

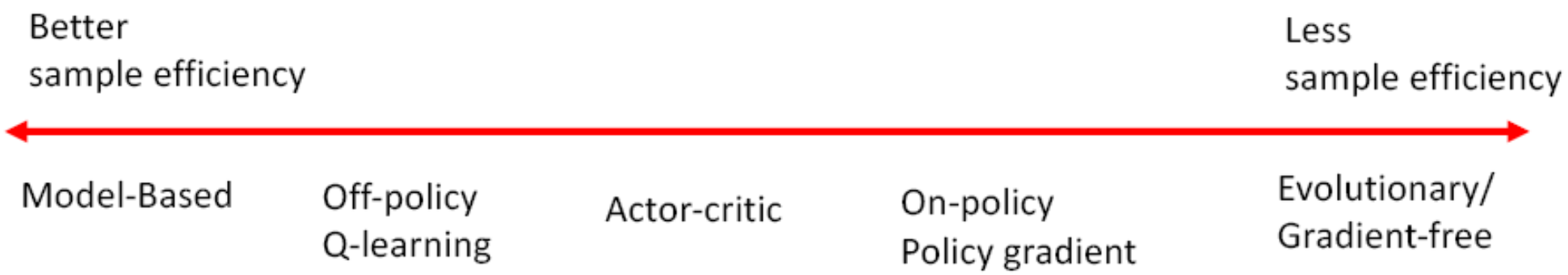
Long Ouyang* Jeff Wu* Xu Jiang* Diogo Almeida* Carroll L. Wainwright*
Pamela Mishkin* Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray
John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens
Amanda Askell† Peter Welinder Paul Christiano*†
Jan Leike* Ryan Lowe*
OpenAI



强化学习——总结


基于MDPs的序贯决策问题
基于模型（Model-Based）
值迭代
策略迭代
动态规划
无模型（Model-Free）
回合更新——蒙特卡洛学习
时间差分(步更新)
基于价值(Value-Based)
On-Policy——Sarsa
Off-Policy——Q-learning
神经网络
DQN
基于策略(Policy-Based)
Policy Gradient
Actor-Critic
DDPG
A3C

Reinforcement Learning Algorithms



Model-Based

- Learn the model of the world, then plan using the model
- Update model often
- Re-plan often

Value-Based

- Learn the state or state-action value
- Act by choosing best action in state
- Exploration is a necessary add-on

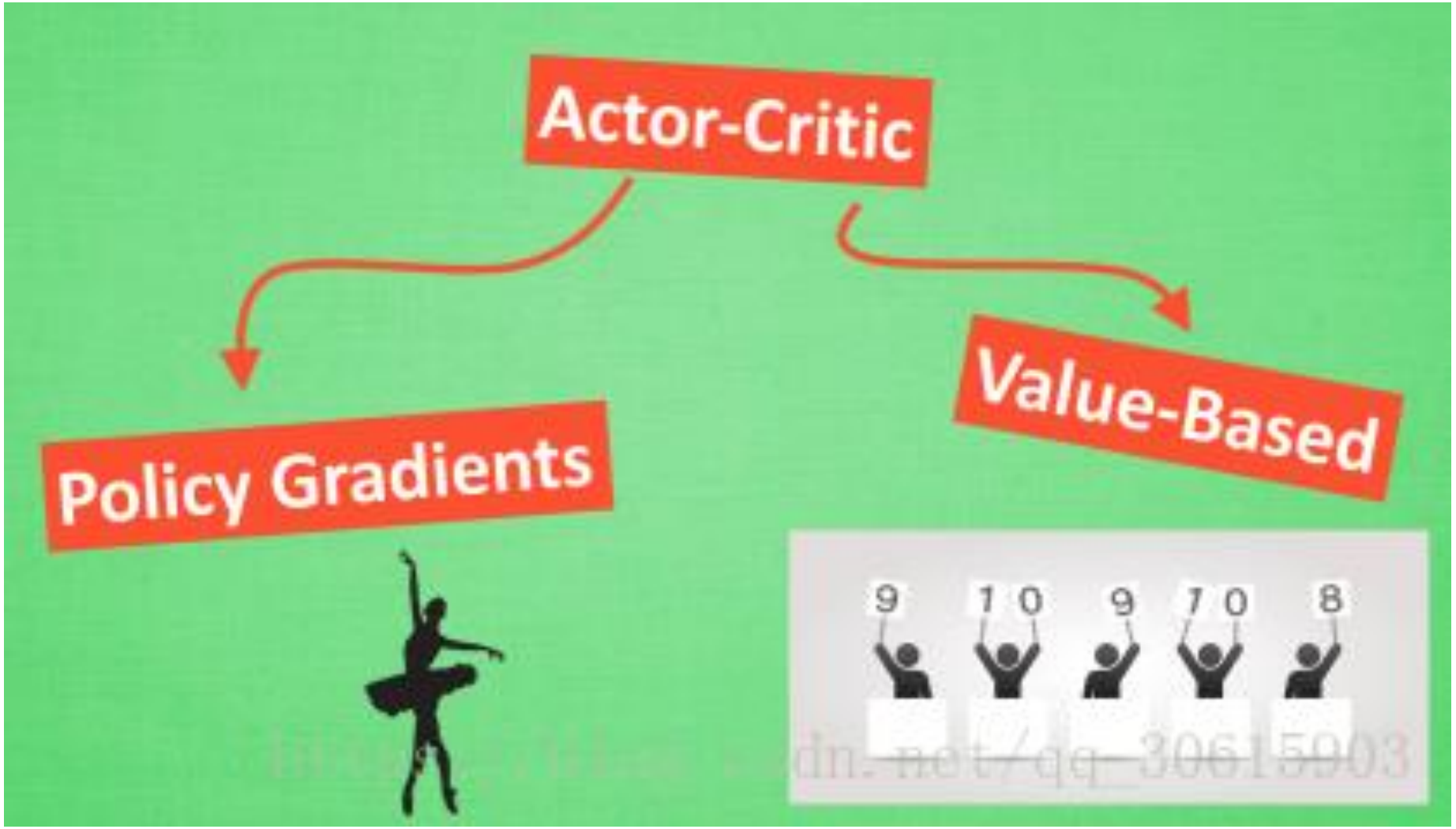
Policy-based

- Learn the stochastic policy function that maps state to action
- Act by sampling policy
- Exploration is baked in

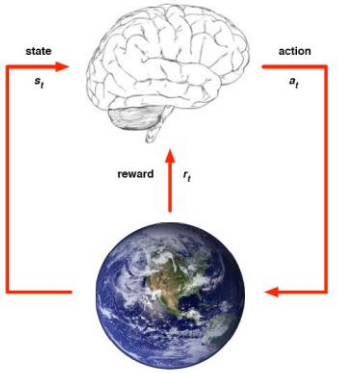
强化学习——总结

- 强化学习是智能系统从环境到行为映射的学习，以使奖励信号(强化信号)函数值最大
- 强化学习不同于连接主义学习中的监督学习，主要表现在教师信号上，强化学习中由环境提供的强化信号是对产生动作的好坏作一种评价(通常为标量信号)，而不是告诉强化学习系统如何去产生正确的动作。
- 由于外部环境提供的信息很少，智能体必须靠自身的经历进行学习。通过这种方式，智能体在行动-评价的环境中获得知识，改进行动方案以适应环境。

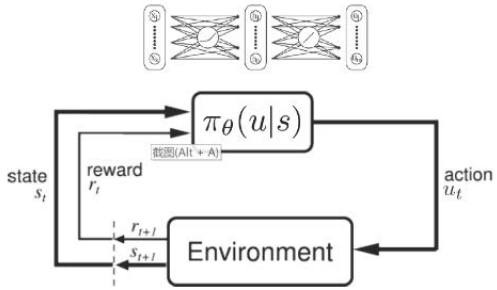
强化学习——总结



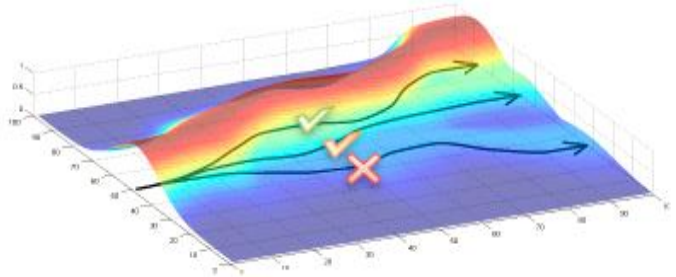
强化学习——总结



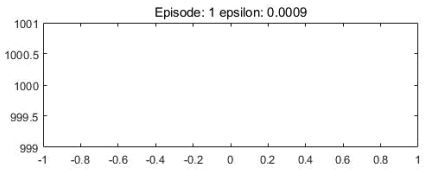
$$a = \arg\max_{a' \in actions} Q(s, a')$$

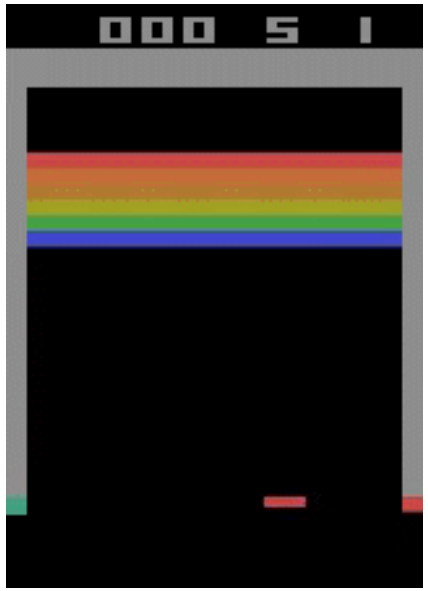


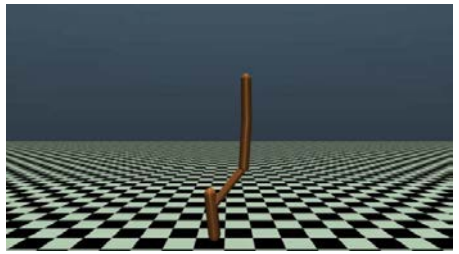
$$\max_{\theta} \quad \mathrm{E}[\sum_{t=0}^{H} R(s_t)|\pi_\theta]$$

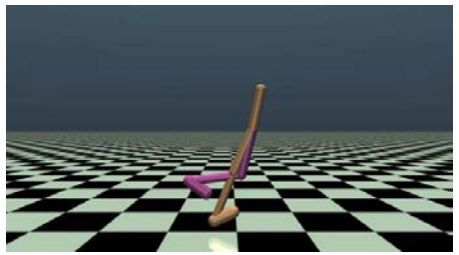


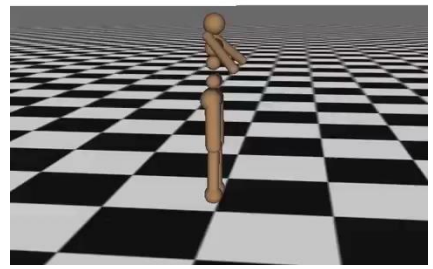
$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau;\theta) R(\tau)$$

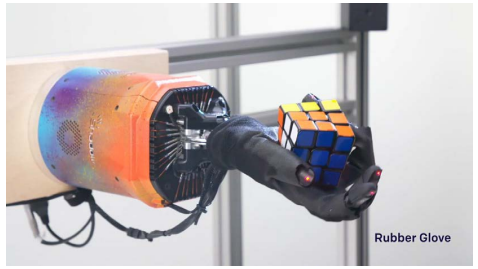












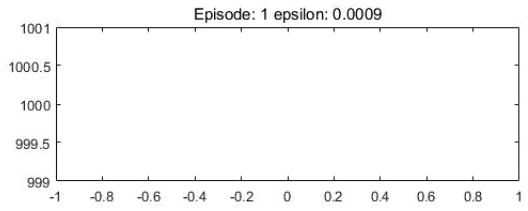
强化学习——总结

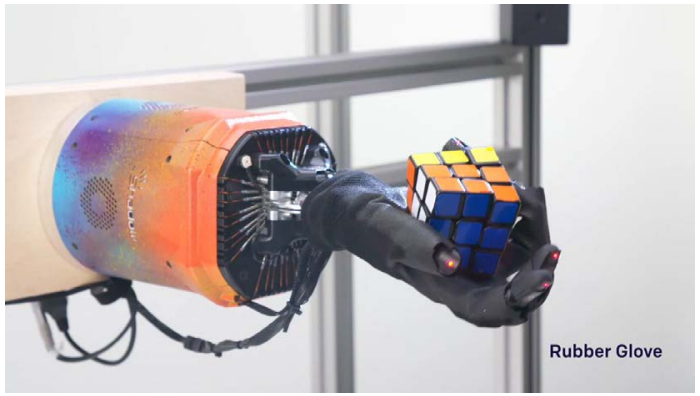
Reinforcement Learning: Learning policies guided by sparse rewards, e.g., win or not the game.

- Good: simplest, cheapest form of supervision
- Bad: High sample complexity

Where is it successful so far?

- in simulation, where we can afford a lot of trials, easy to parallelize
- not in robotic systems:
  1. action execution takes long
  2. we cannot afford to fail
  3. safety concerns



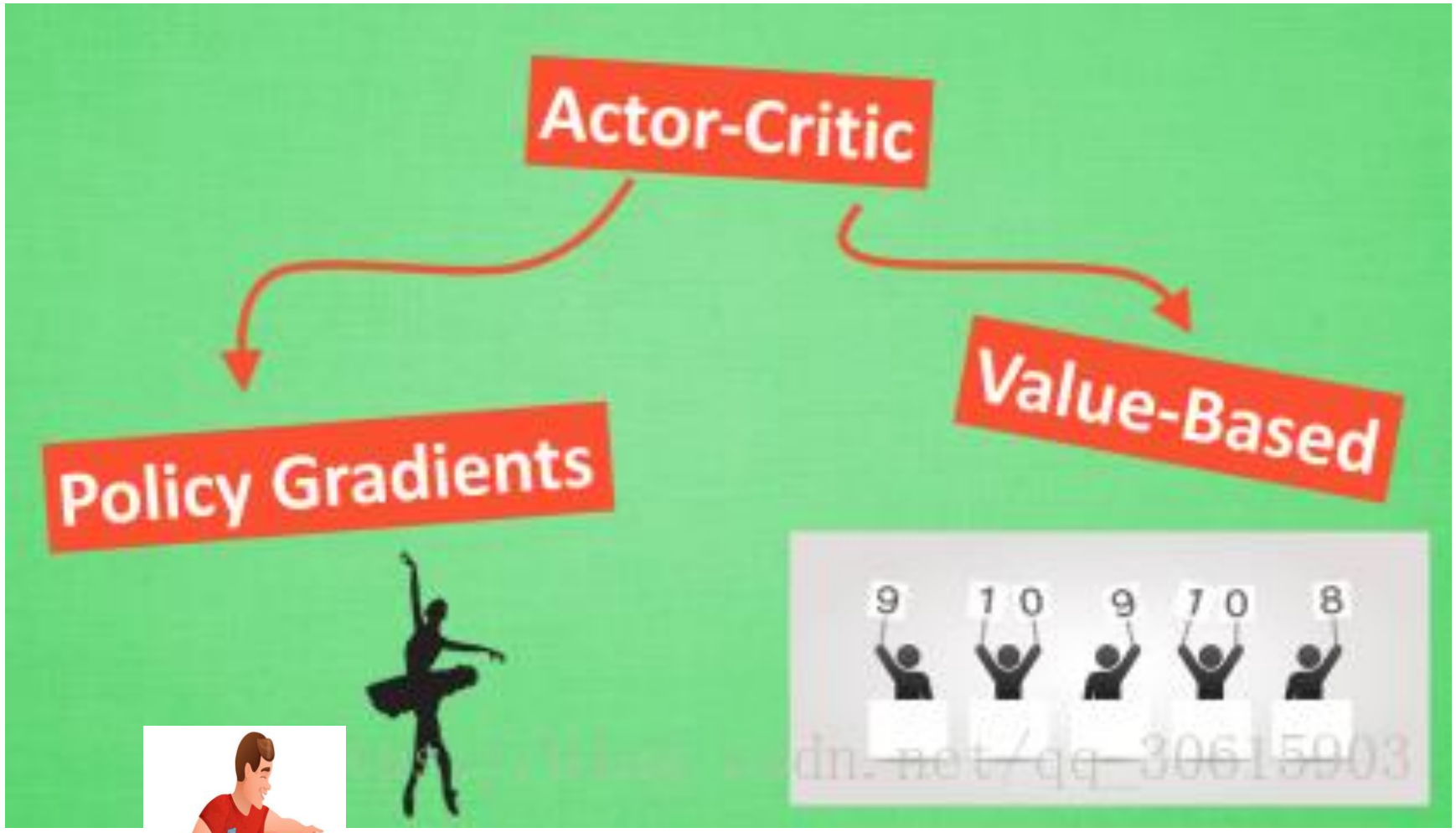


强化学习——总结

- Ideally we want dense in time rewards to closely guide the agent closely along the way.

- Who will supply those shaped rewards?
  - We will manually design them: *“cost function design by hand remains one of the black arts’ of mobile robotics, and has been applied to untold numbers of robotic systems”*
  - We will learn them from demonstrations: *“rather than having a human expert tune a system to achieve desired behavior, the expert can demonstrate desired behavior and the robot can tune itself to match the demonstration”*

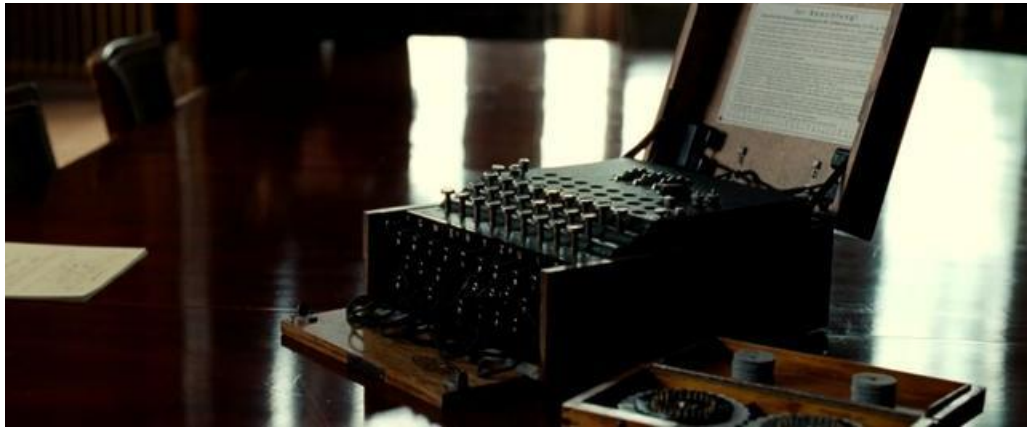
强化学习——总结



Expert

- **模仿学习**
  - ❑ 示教与模仿
  - ❑ 模仿学习问题定义
  - ❑ 监督式行为克隆
  - ❑ 逆强化学习
  - ❑ 对抗式生成模仿学习

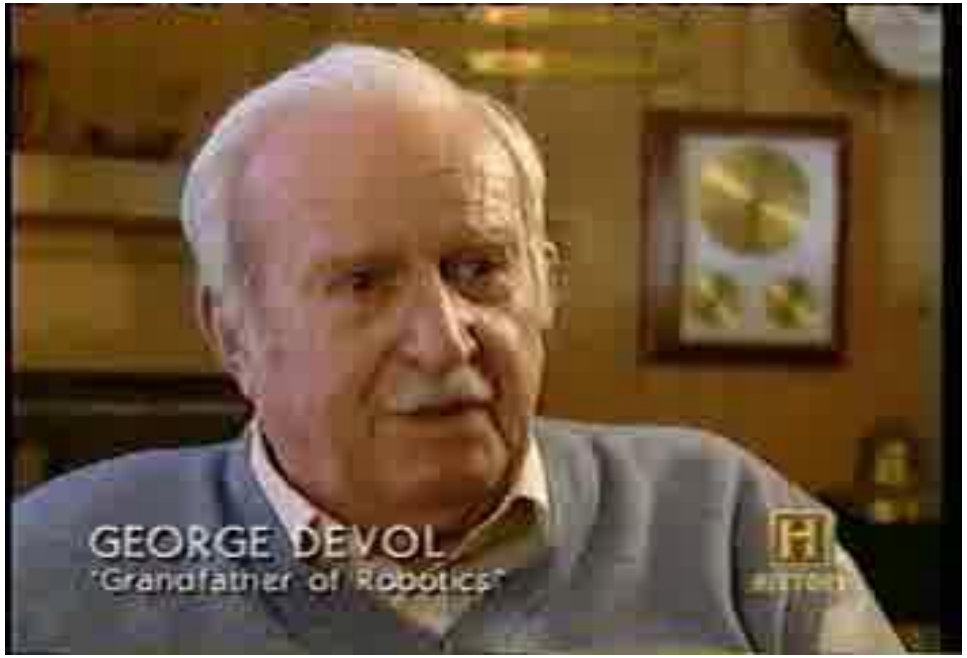
示教与模仿



“既然用人海战术破译密码既耗时费力又没有成效，那我们为什么不能制造一个比人脑运算还快的机器呢？”

示教与模仿

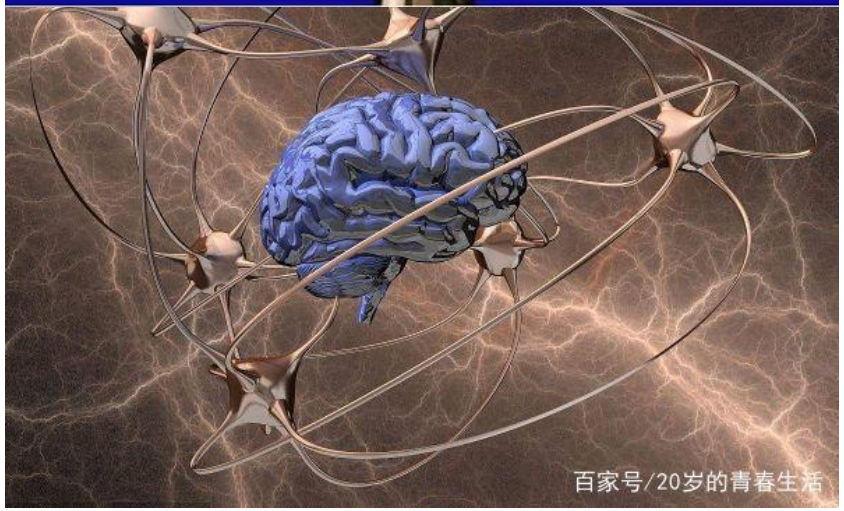
1954年，美国戴沃尔最早提出了工业机器人的概念，并申请了专利。该专利的要点是借助伺服技术控制机器人的关节，利用人手对机器人进行动作示教，机器人能实现动作的记录和再现。这就是所谓的示教再现机器人。现有的机器人基本都采用这种控制方式。

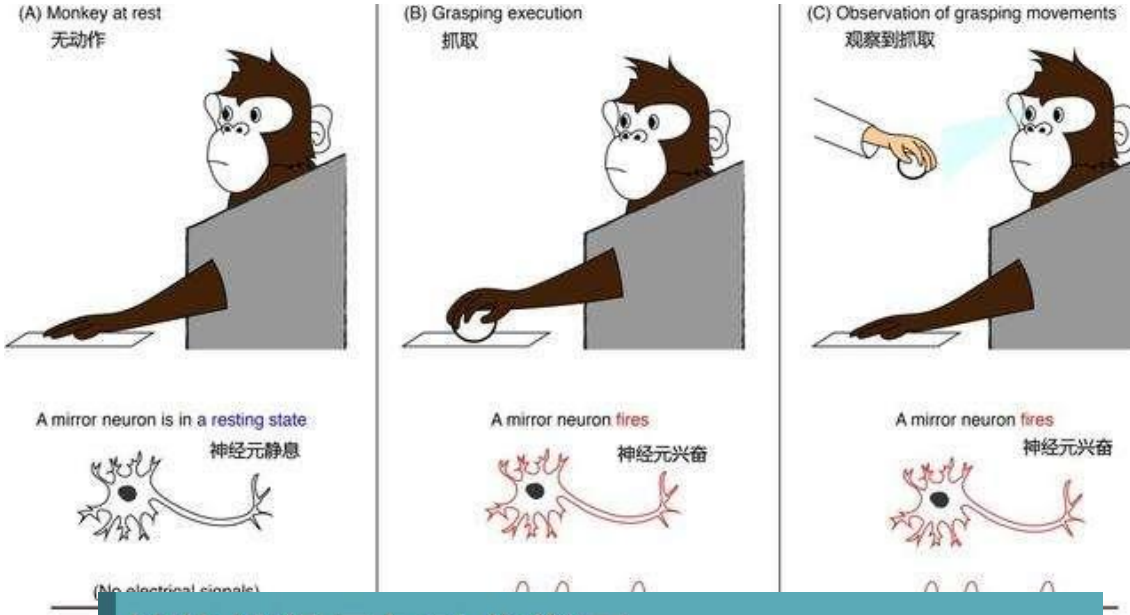


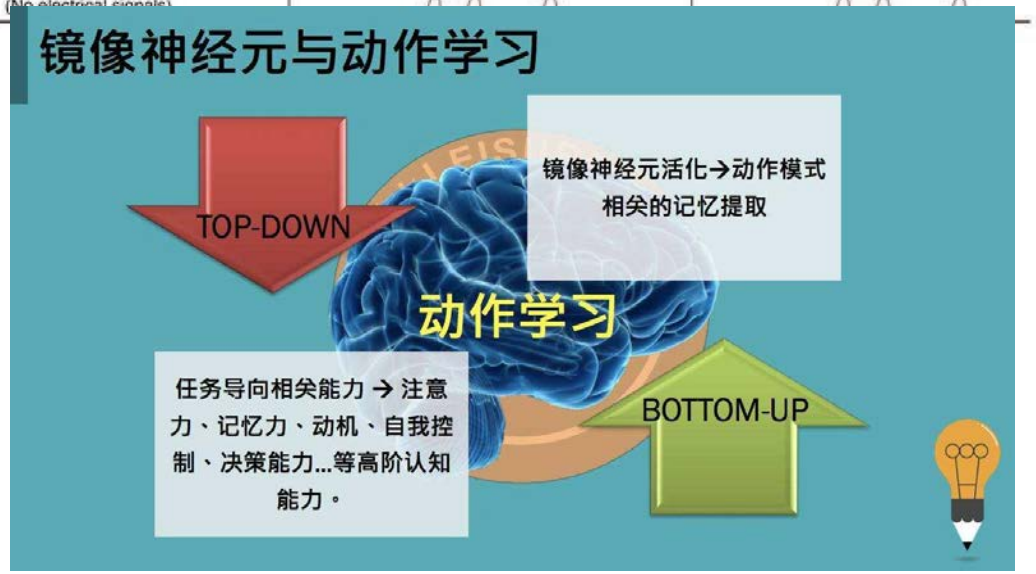
示教与模仿

示教与模仿

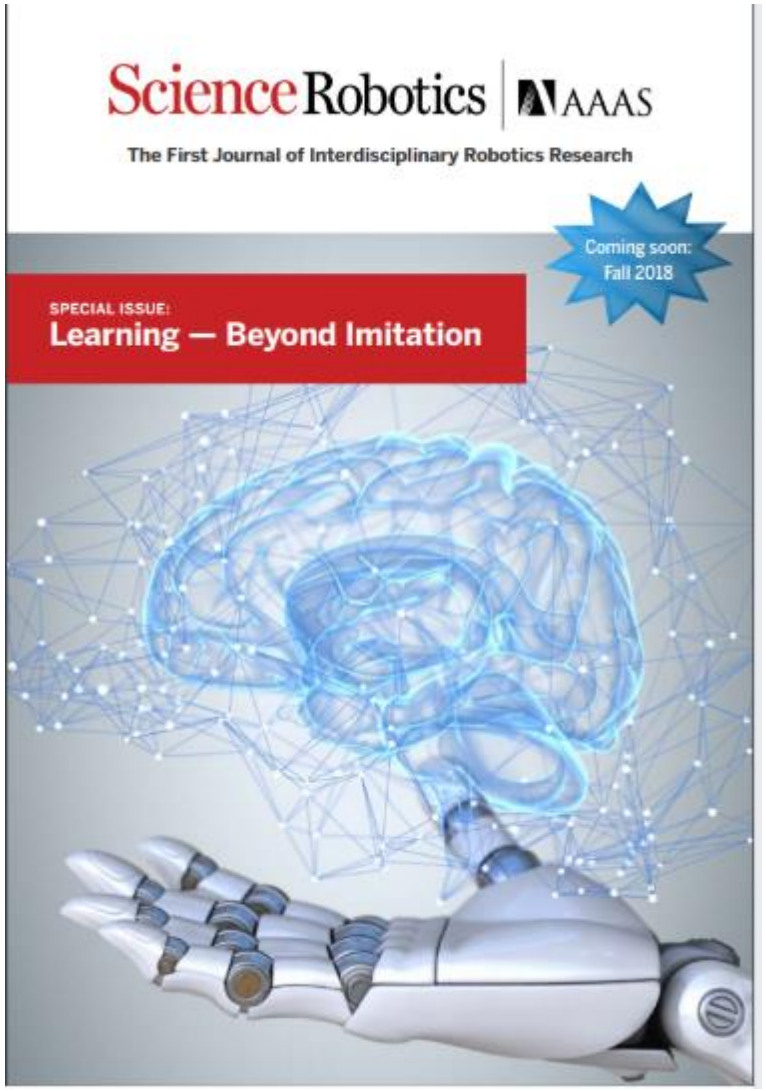

“脑中之镜”-镜像神经元
动作观察
观察者
做动作者

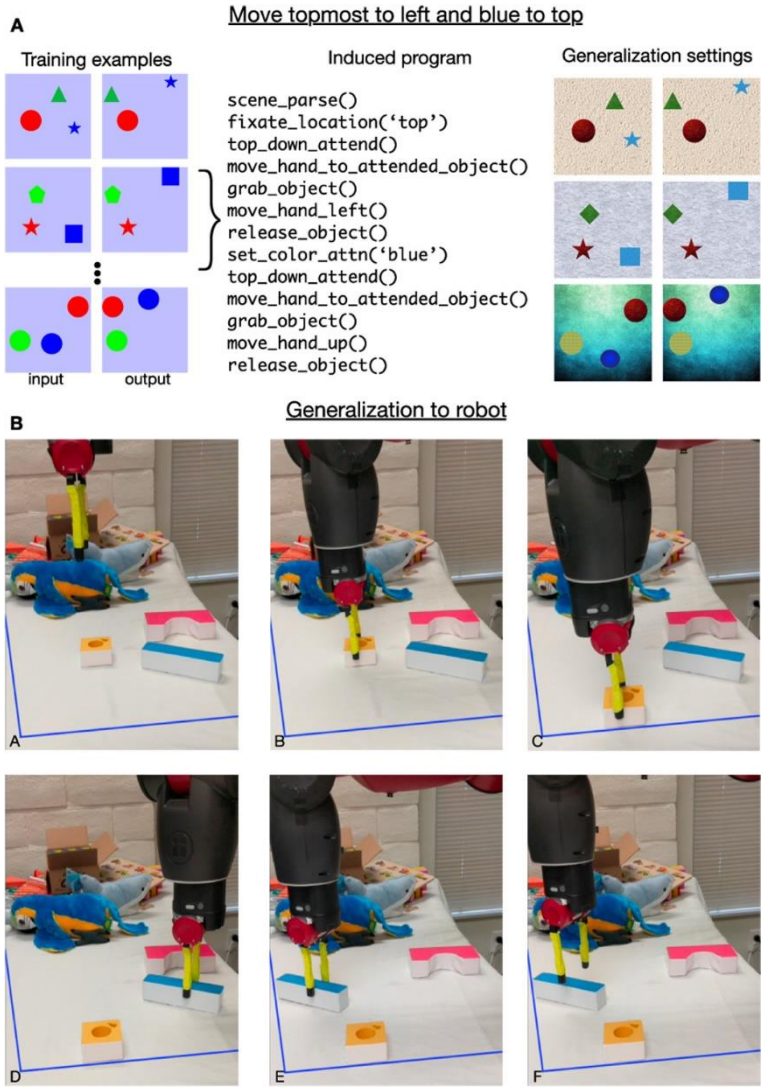

百家号/20岁的青春生活


(A) Monkey at rest
无动作
(B) Grasping execution
抓取
(C) Observation of grasping movements
观察到抓取
A mirror neuron is in a resting state
神经元静息
A mirror neuron fires
神经元兴奋
A mirror neuron fires
神经元兴奋


镜像神经元与动作学习
TOP-DOWN
镜像神经元活化→动作模式
相关的记忆提取
动作学习
任务导向相关能力 → 注意力、记忆力、动机、自我控制、决策能力...等高阶认知能力。
BOTTOM-UP

示教与模仿


Science Robotics
AAAS
The First Journal of Interdisciplinary Robotics Research
Coming soon:
Fall 2018
SPECIAL ISSUE:
Learning — Beyond Imitation


A
Move topmost to left and blue to top
Training examples
Induced program
Generalization settings
scene_parse()
fixate_location('top')
top_down_attend()
move_hand_to_attended_object()
grab_object()
move_hand_left()
release_object()
set_color_attn('blue')
top_down_attend()
move_hand_to_attended_object()
grab_object()
move_hand_up()
release_object()
input
output
B
Generalization to robot
A
B
C
D
E
F

示教与模仿

➢ 观察学习理论

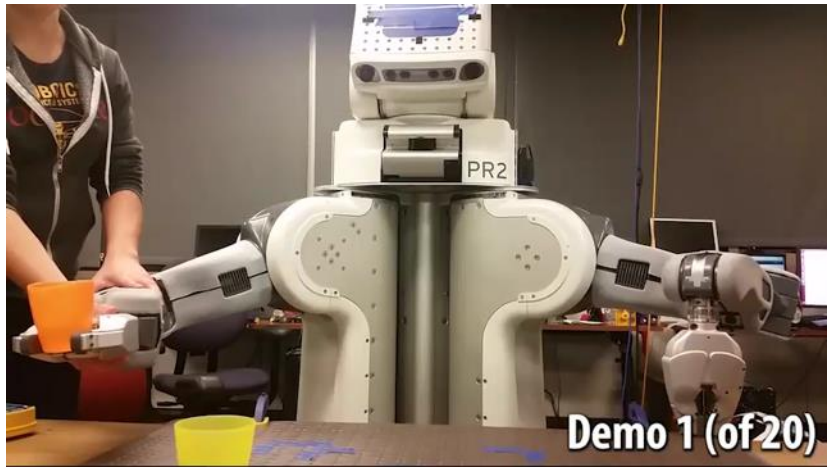
- 观察学习是指，个体仅仅是在观察到他人的行为被强化或被惩罚后，就会在后来或者做出类似行为，或者抑制该行为。
- 班杜拉在实验室对孩子观察学习的能力进行了经典演示：在看过一个成人榜样对大型塑料娃娃拳打脚踢之后，与未目睹过攻击榜样的控制组儿童想比，实验组儿童表现出了更高频率的攻击性行为！

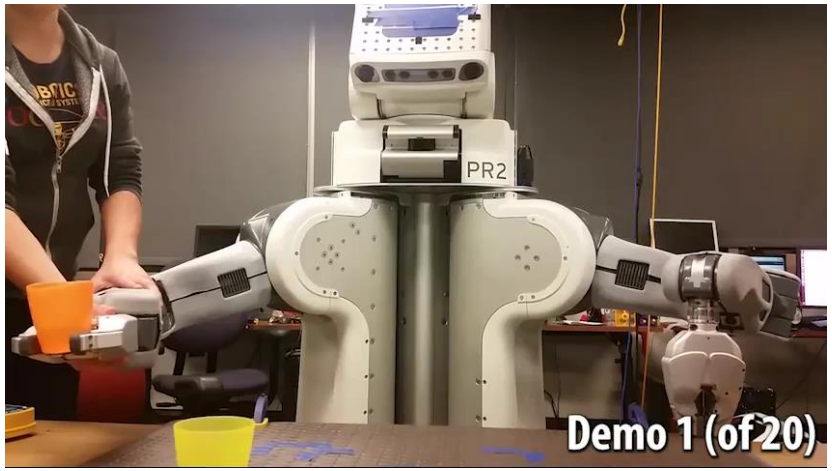
阿尔伯特·班杜拉（Albert Bandura，1925年-2021年7月28日），新行为主义的主要代表人物之一，社会学习理论的创始人。阿尔伯特·班杜拉，美国当代著名心理学家，斯坦福大学心理学系约丹讲座教授。

示教与模仿

What is Imitation Learning?

Supervised Learning VS **Robot learning/Imitation Learning**





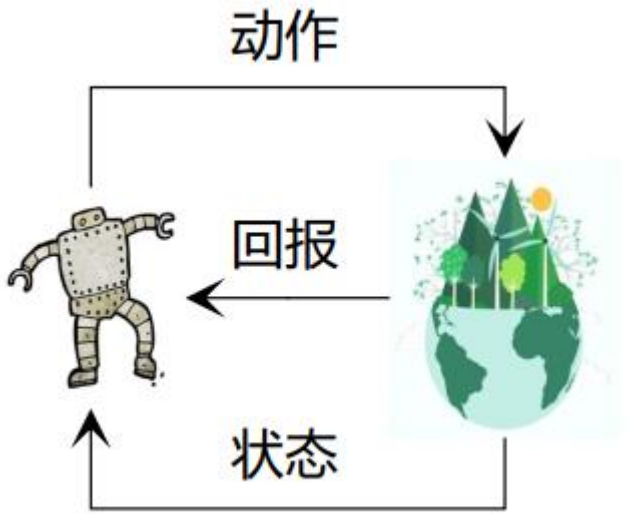
This is a cup!

静态问题

How to pour water!

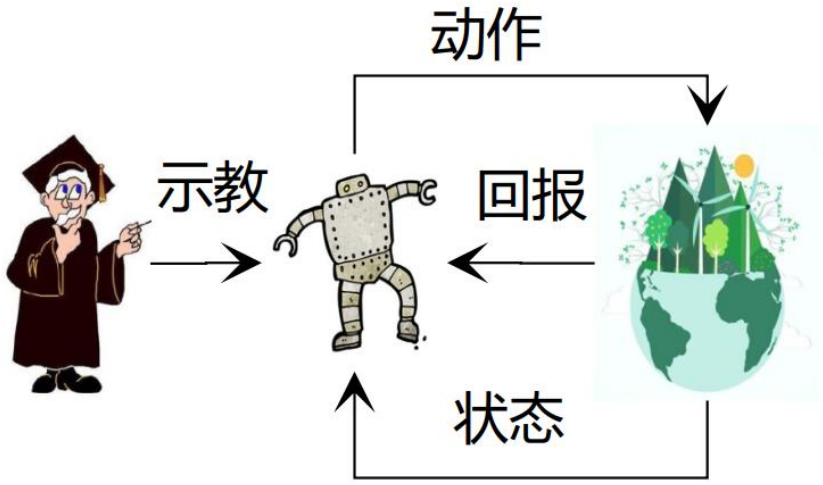
动态，多步决策问题

示教与模仿



强化学习

- 设计合适奖惩函数具有挑战性
- 长周期任务奖惩函数一般稀疏
- 奖惩函数难以区分行为的好坏

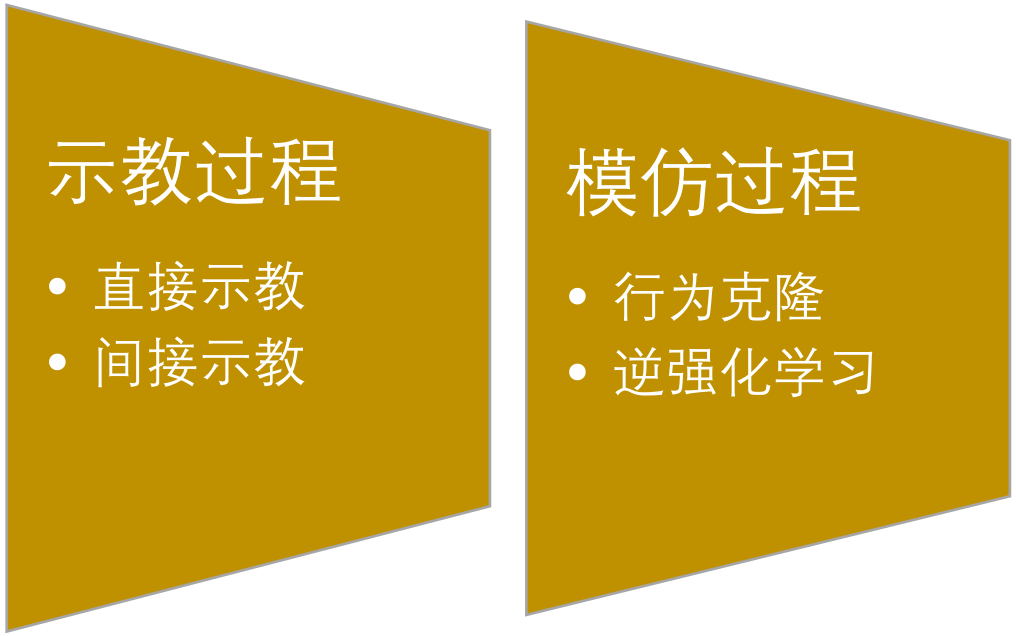


模仿学习

- 加速策略学习训练
- 不依赖于奖惩函数训练
- 一定准则下恢复奖惩函数

示教与模仿

➢ 模仿学习组成部分



第一阶段：示教过程，提供专家演示数据

第二阶段：模仿过程，离线或在线学习模仿策略

示教与模仿

➢ 示教过程—操作场景

（a）拖拽操作示教

（b）VR遥操作示教

直接示教：

- **人机耦合**
- **示教低效**
- **示教信息完备**

间接示教：

- **人机解耦**
- **示教低效**
- **示教信息完备**

Zhang et al. 2018. Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation

示教与模仿

- **示教过程—运动场景**

间接观测示教：

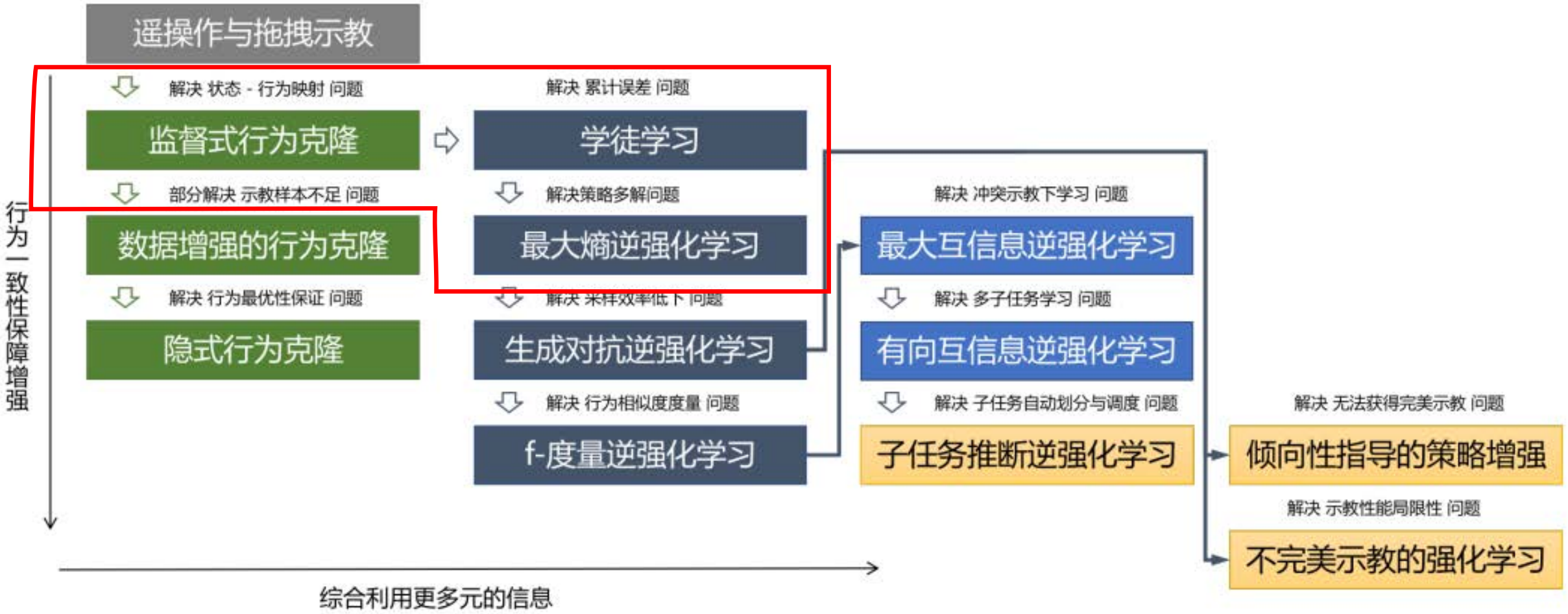
人机解耦

示教高效

示教信息不完备

Peng et al. Deepmimic 2018

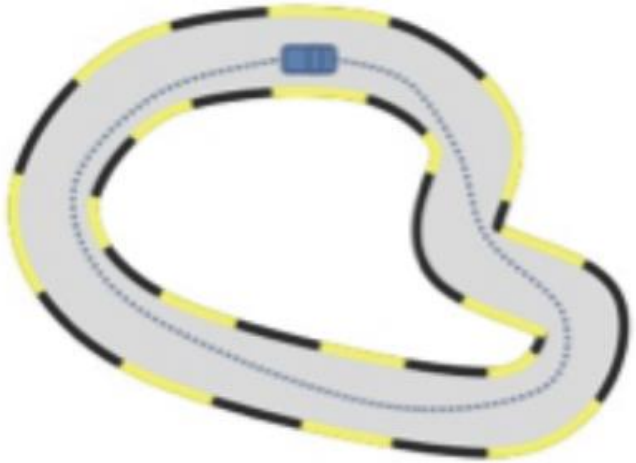
示教与模仿


遥操作与拖拽示教
解决 状态 - 行为映射 问题
监督式行为克隆
部分解决 示教样本不足 问题
数据增强的行为克隆
解决 行为最优性保证 问题
隐式行为克隆
解决 累计误差 问题
学徒学习
解决策略多解问题
最大熵逆强化学习
解决 采样效率低下 问题
生成对抗逆强化学习
解决 行为相似度度量 问题
f-度量逆强化学习
解决 冲突示教下学习 问题
最大互信息逆强化学习
解决 多子任务学习 问题
有向互信息逆强化学习
解决 子任务自动划分与调度 问题
子任务推断逆强化学习
解决 无法获得完美示教 问题
倾向性指导的策略增强
解决 示教性能局限性 问题
不完美示教的强化学习
行为一致性保障增强
综合利用更多元的信息

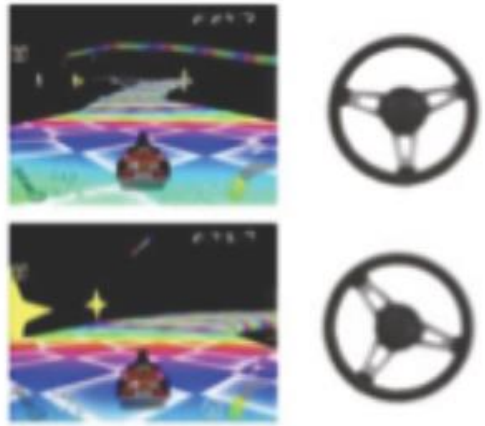
- 示教与模仿
- 模仿学习问题定义
- 监督式行为克隆
- 逆强化学习
- 对抗式生成模仿学习

问题描述

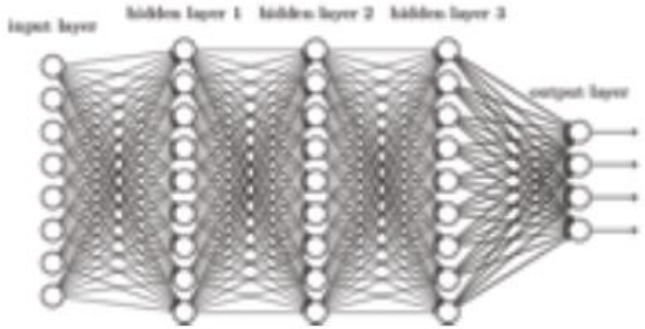
Expert Demonstrations



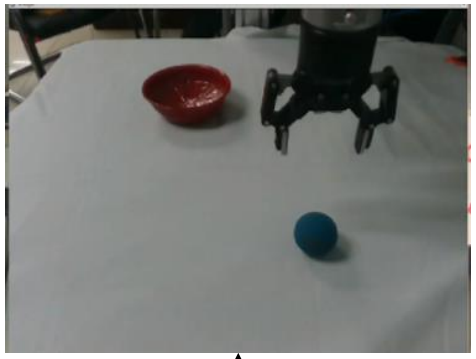
State/Action Pairs

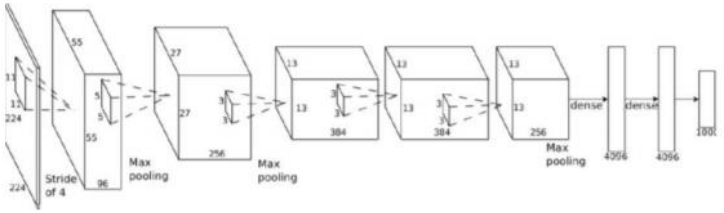


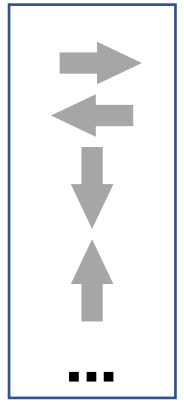
Learning



问题描述



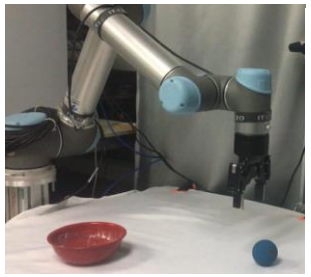




$\pi(a_t|s_t)$

s_t

a_t



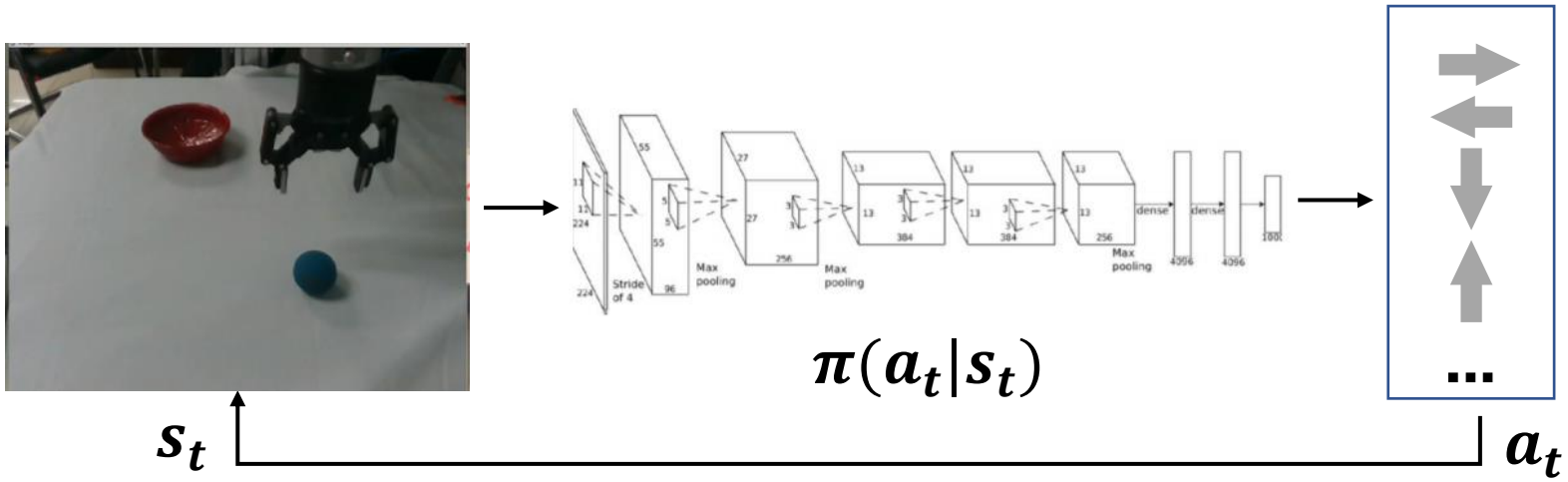
o_t

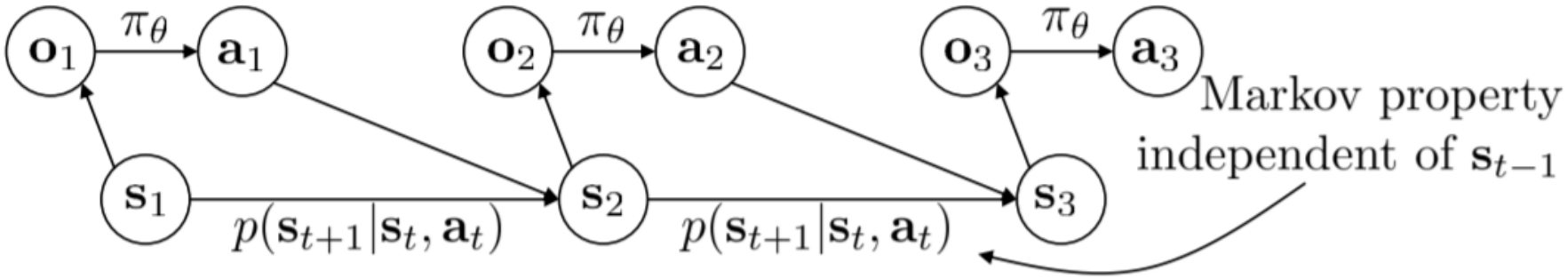
s_t

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy

$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

问题描述


$\pi(a_t|s_t)$
...
s_t
a_t

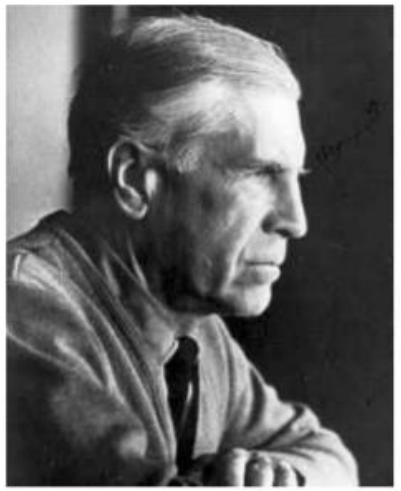

$\mathbf{o}_1$
π_θ
$\mathbf{a}_1$
$\mathbf{o}_2$
π_θ
$\mathbf{a}_2$
$\mathbf{o}_3$
π_θ
$\mathbf{a}_3$
$\mathbf{s}_1$
$\mathbf{s}_2$
$\mathbf{s}_3$
$p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
$p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
Markov property
independent of $\mathbf{s}_{t-1}$

问题描述

$\mathbf{s}_t$ – state
$\mathbf{a}_t$ – action

$\mathbf{x}_t$ – state
$\mathbf{u}_t$ – action управление

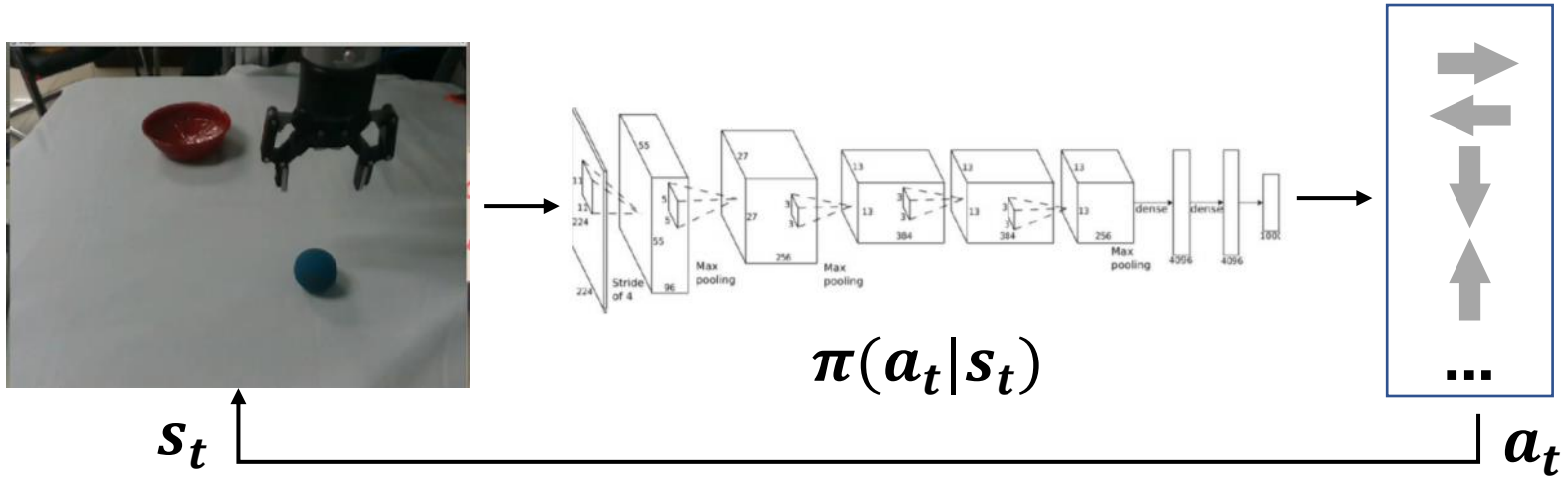
Richard Bellman

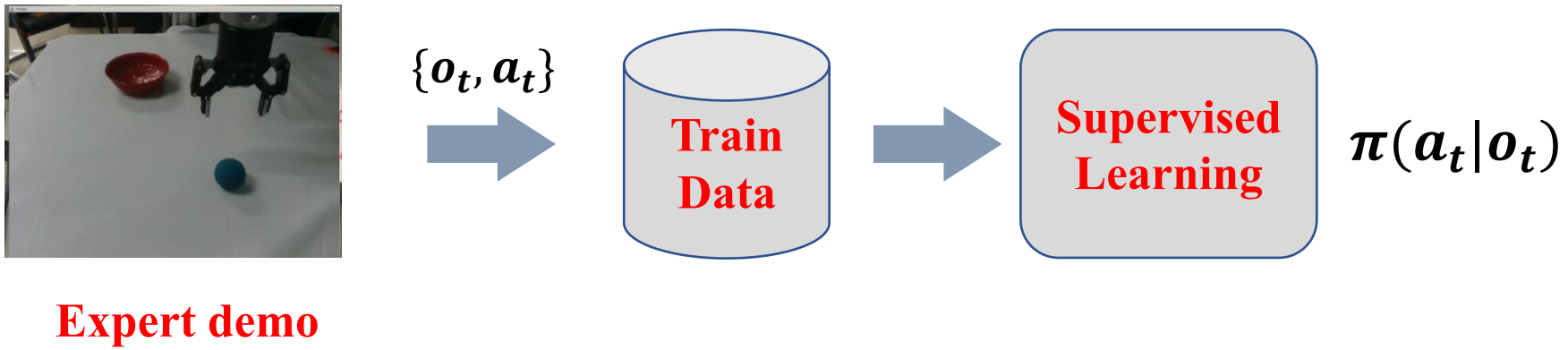


Lev Pontryagin

- 示教与模仿
- 模仿学习问题定义
- 监督式行为克隆
- 逆强化学习
- 对抗式生成模仿学习

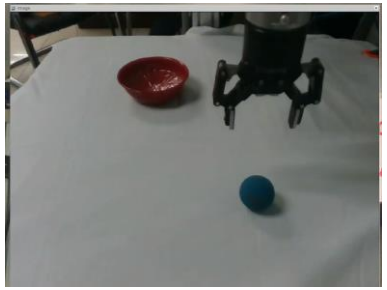
监督式行为克隆





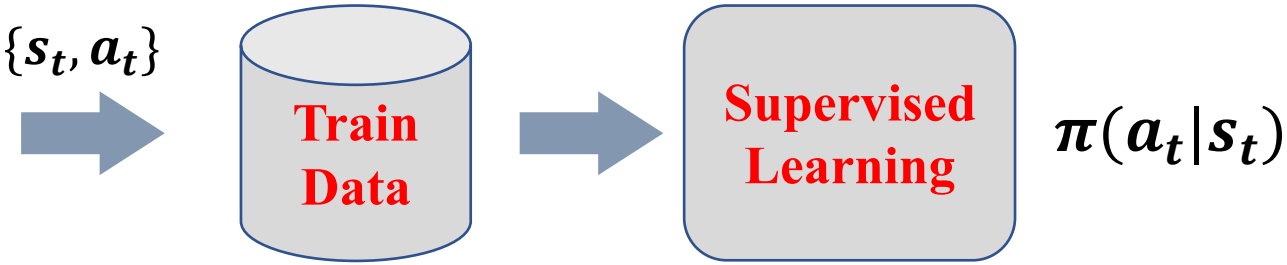
Behavior Clone

监督式行为克隆



Expert demo

$\{s_t, a_t\}$



$\pi(a_t|s_t)$

Behavior Clone

$$\tau_E^{(n)} = \{s_0^{(n)}, a_0^{(n)}; s_1^{(n)}, a_1^{(n)}; s_2^{(n)}, a_2^{(n)}; \ldots; s_{H^{(n)}}^{(n)}, a_{H^{(n)}}^{(n)}\}$$

$$\max_{\theta} \quad \mathrm{E}_{(s,a)\sim\mathcal{D}} \log \pi_\theta(a \mid s)$$

$$\min_{\theta} \quad \sum_{n=1}^{N} \sum_{t=0}^{H^{(n)}} \left\| \pi_\theta(s_t^{(n)}) - a_t^{(n)} \right\|_2^2$$

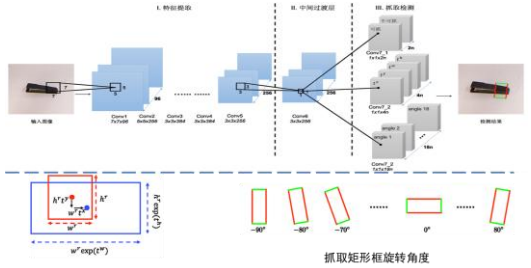
监督式行为克隆

➢ 示教过程—操作场景

通过轨迹规划，自监督采集专家数据

o_t: 每帧图像（由外置深度相机采集）

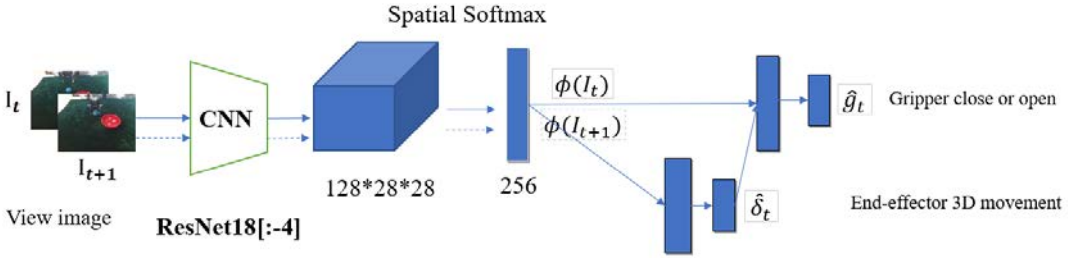
a_t: 机械臂末端的相对运动与机械爪的开合



$$L(g,a,t) = L_{grp}(g,\hat{g}) + \lambda_1 \hat{g} L_{ang}(a,\hat{a}) + \lambda_2 \hat{g} L_{reg}(t,\hat{t})$$

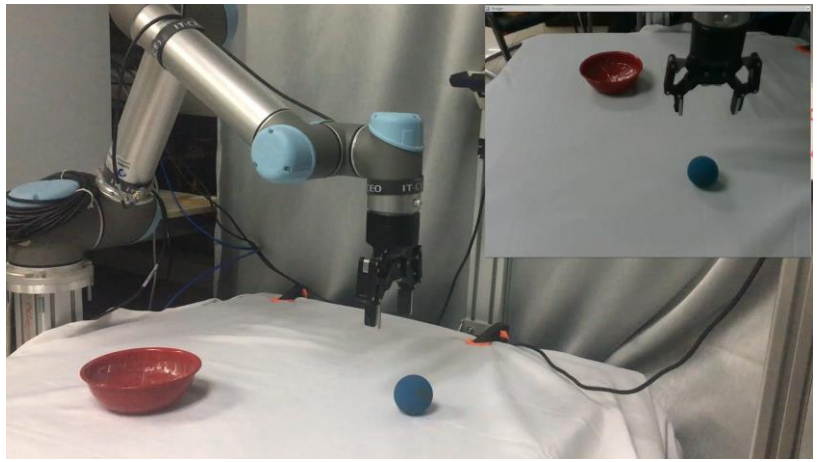
Expert Demo: Self-place → Motion-planning → Self-pick

专家训练样本



$$L = \| \hat{\delta}_t - \delta_t \|_2 + \lambda * L(\hat{g}_t, g_t)$$

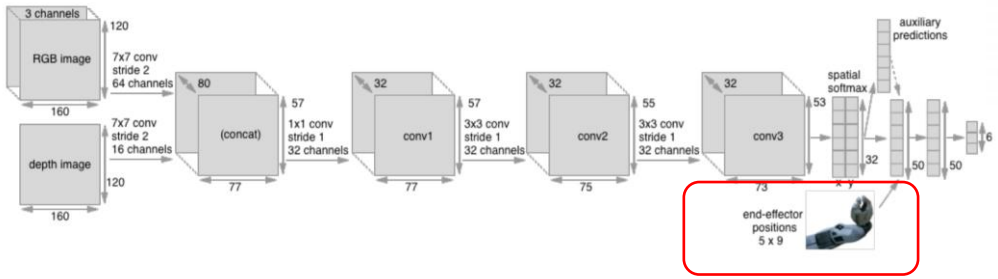
双路回归网络



UR5+Robotiq pick-and-place manipulation application

监督式行为克隆

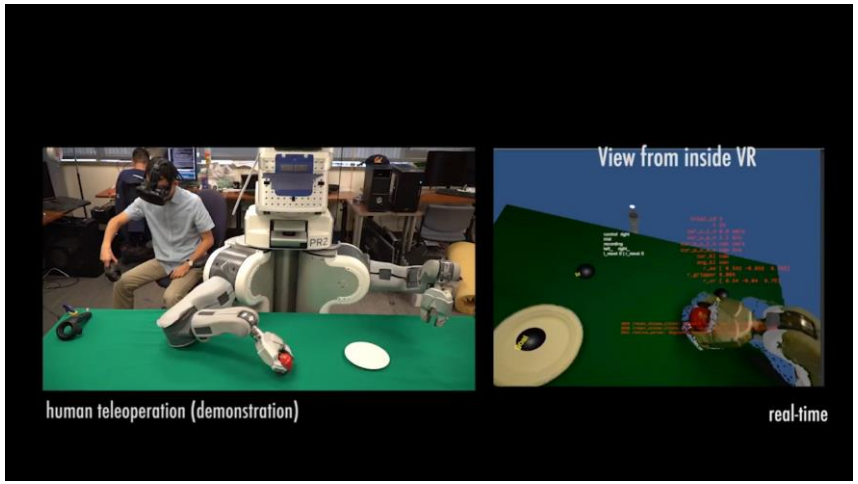
- 示教过程—操作场景

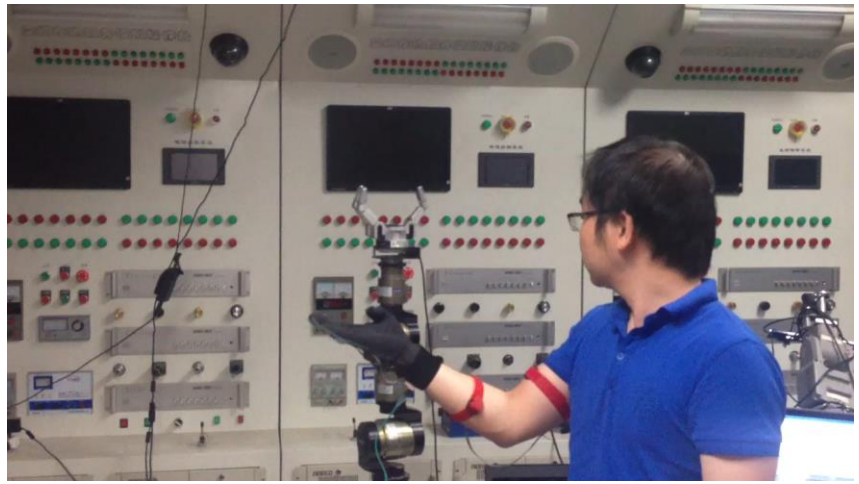


$$\mathcal{L}_{aux}^{(a)} = ||\mathrm{NN}(f_t; \theta_{aux}^{(a)}) - s_t^{(a)}||_2^2$$

Auxiliary supervised

arm position



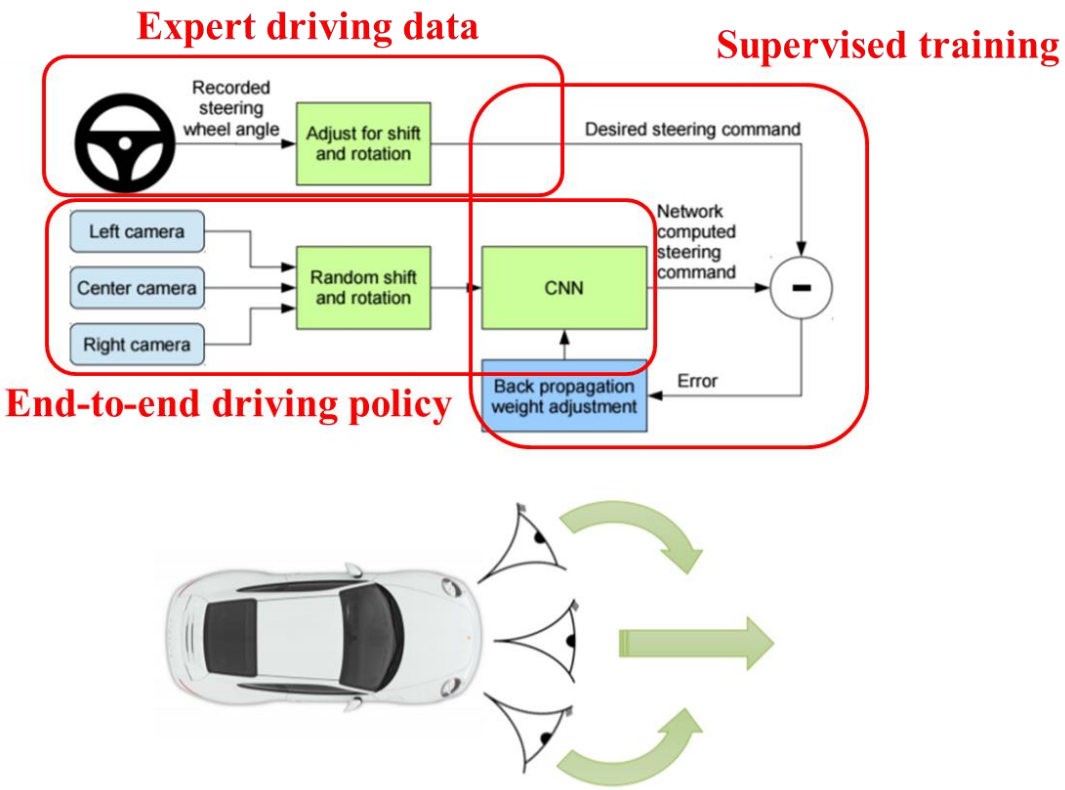


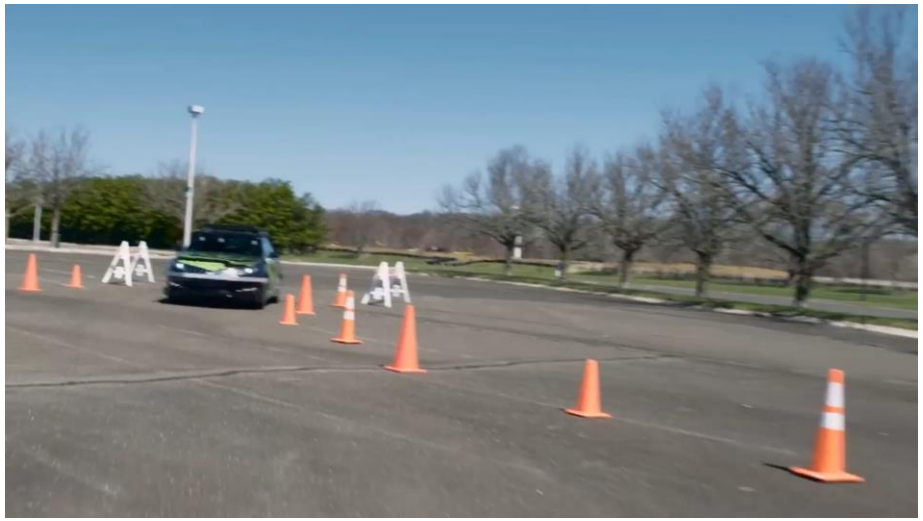
More complex policy for robot grasp manipulation

Zhang et al. 2018. Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation

监督式行为克隆

➢ 示教过程—自动驾驶





Bojarski M, Del Testa D, Dworakowski D, et al. End to end learning for self-driving cars[J]. arXiv preprint arXiv:1604.07316, 2016. **NVIDIA**

监督式行为克隆

➢ 存在问题

- ❖ 累积误差（compounding error）
- ❖ 数据分配“漂移”（distributional drift）
- ❖ 非马尔科夫行为（Non-Markovian behavior）
- ❖ 多模态专家示教行为（Multimodal behavior）

监督式行为克隆

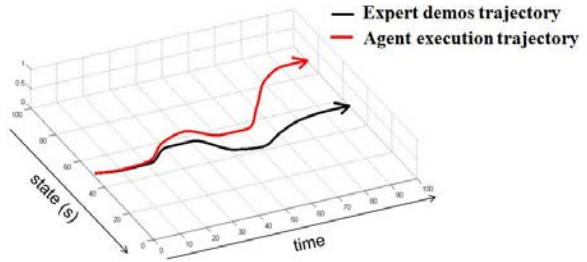
- 存在问题：累计误差

error at time t with probability ε

E[Total errors] ≤ εT

error at time t with probability ε

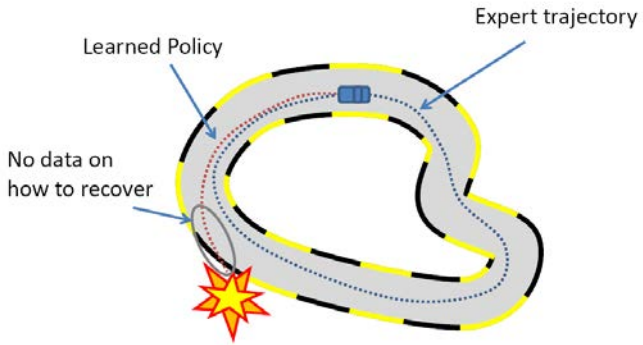
E[Total errors] ≤ ε(T + (T-1) + (T-2) + …+ 1) ∝ εT²



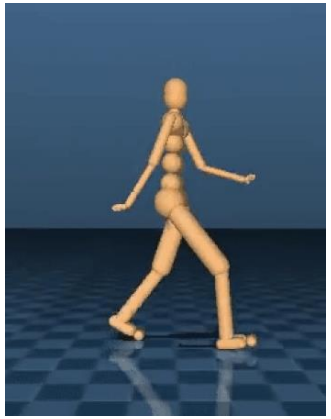
监督式行为克隆

- 存在问题：数据分布“飘移”

$$P_{\pi^*}(\mathbf{o}_t) \neq P_{\pi_\theta}(\mathbf{o}_t)$$

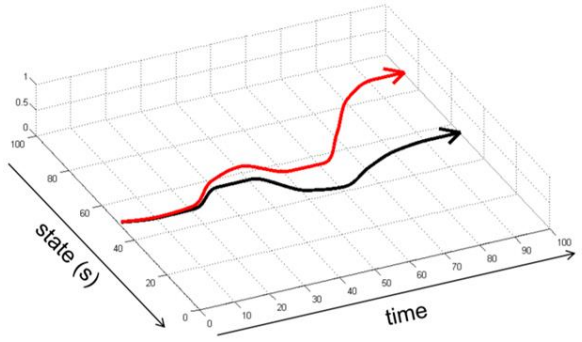


	supervised learning	supervised learning + control (NAIVE)
train	(x,y) ~ D	$\mathbf{o}_t \sim P_{\pi^*}(\mathbf{o}_t)$
test	(x,y) ~ D	$\mathbf{o}_t \sim P_{\pi_\theta}(\mathbf{o}_t)$



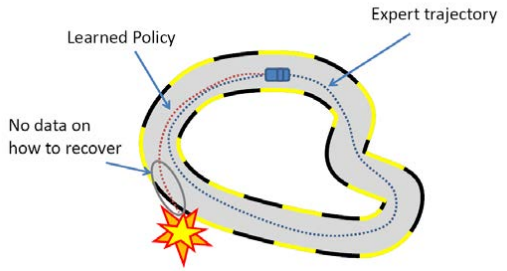
监督式行为克隆

累积误差



数据分配“漂移”

$$P_{\pi^*}(\mathbf{o}_t) \neq P_{\pi_\theta}(\mathbf{o}_t)$$

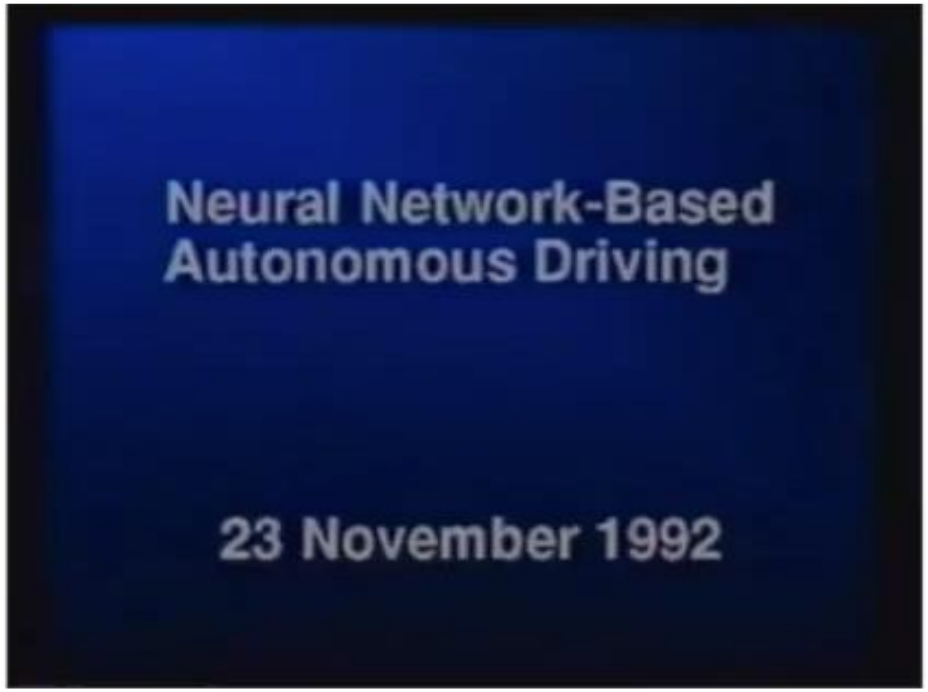


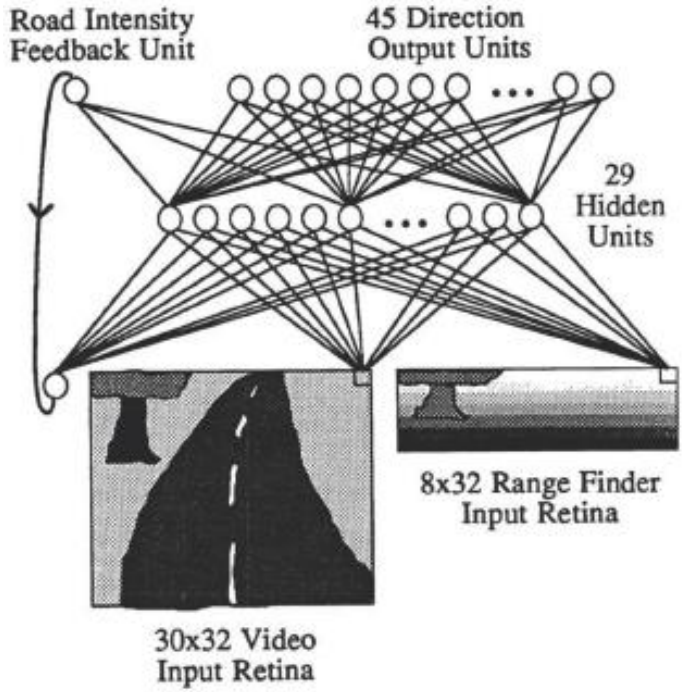
By augmenting the expert demonstration trajectories.
This means: add examples in expert demonstration trajectories to cover the states/observations points where the agent will land when trying out its own policy. How?

1. By generating synthetic data in simulation
2. By collecting additional data via clever hardware
3. By interactively querying the experts in additional datapoints

监督式行为克隆

- By generating synthetic data in simulation



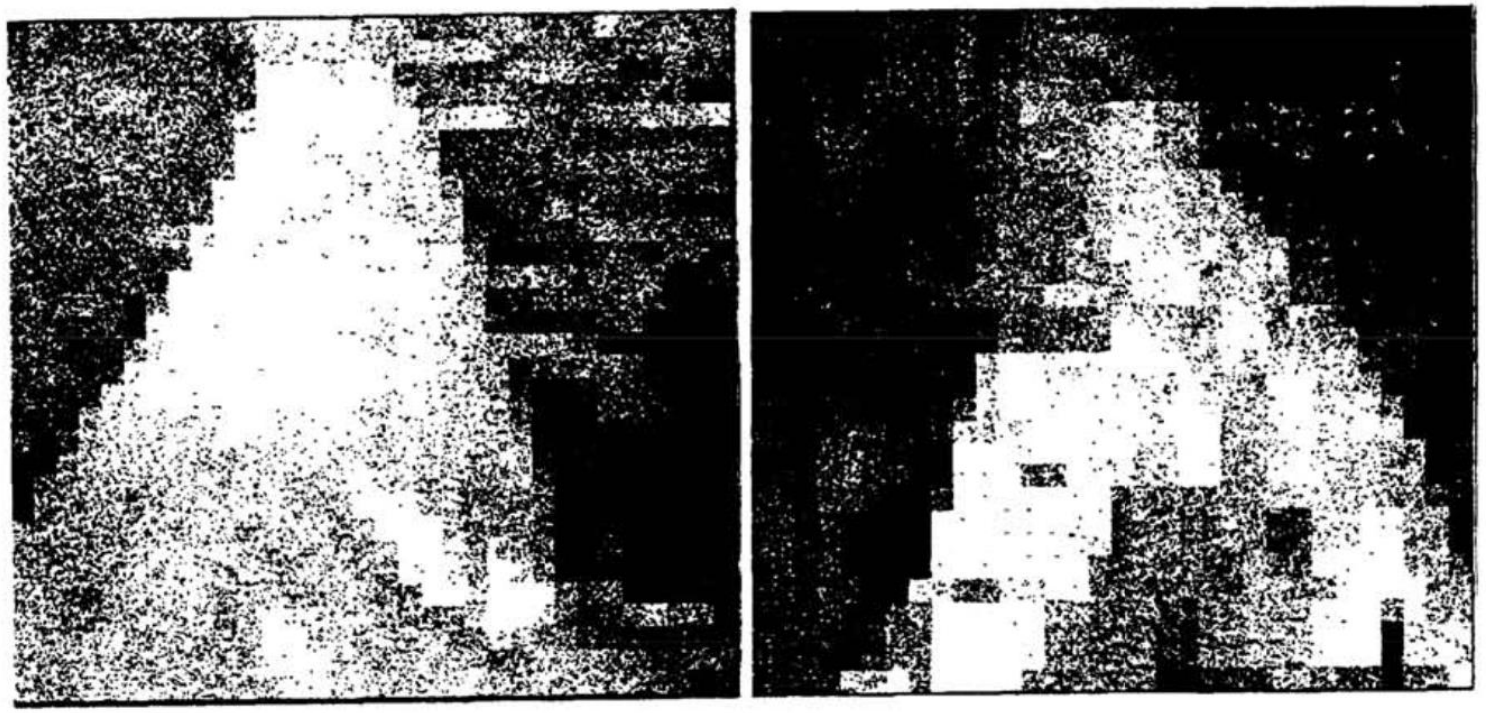


ALVINN 1989

“In addition, the network must not solely be shown examples of accurate driving, but also how to recover (i.e. return to the road center) once a mistake has been made. Partial initial training on a variety of simulated road images should help eliminate these difficulties and facilitate better performance.”
ALVINN: An autonomous Land vehicle in a neural Network”, Pomerleau 1989

监督式行为克隆

- By generating synthetic data in simulation

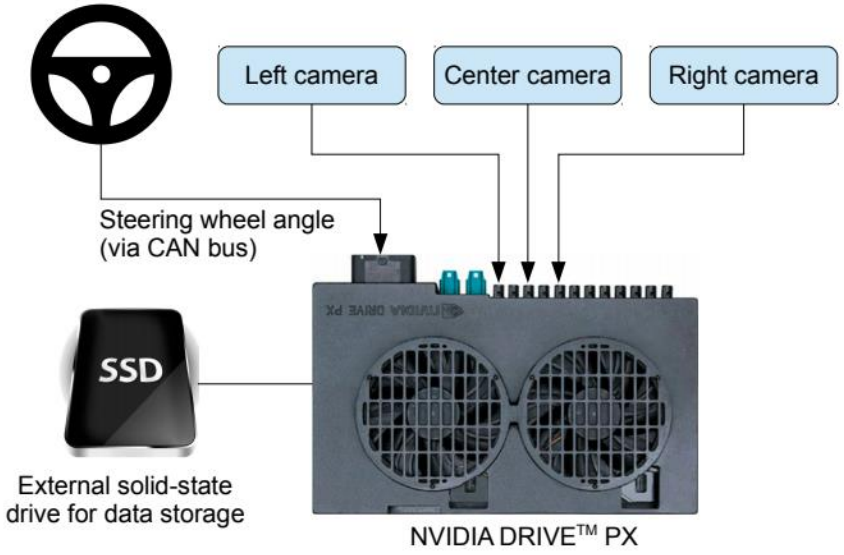


Real Road Image　　Simulated Road Image

- Use of image simulator to generate images of how the road looks like when the vehicle deviates slightly from its trajectory.
- Simulating the images too longer than training the network

监督式行为克隆

- By generating synthetic data via clever hardware

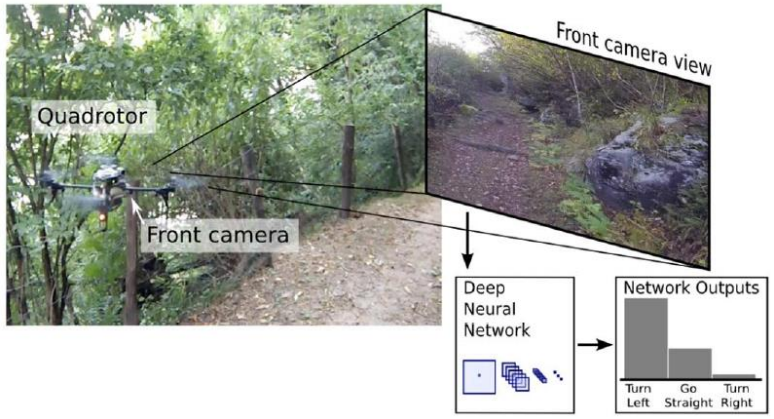


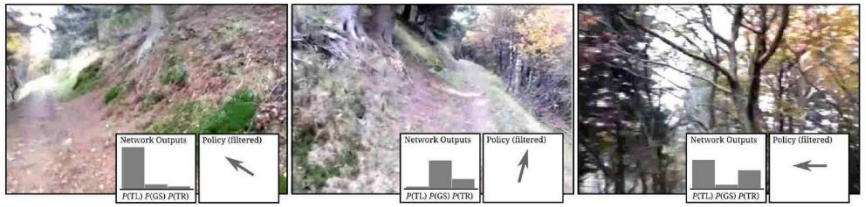
"DAVE-2 was inspired by the pioneering work of Pomerleau [6] who in 1989 built the Autonomous Land Vehicle in a Neural Network (ALVINN) system. Training with data from only the human driver is not sufficient. The network must learn how to recover from mistakes. …",

- End to End Learning for Self-Driving Cars , Bojarski et al. 2016

监督式行为克隆

- By generating synthetic data via clever hardware

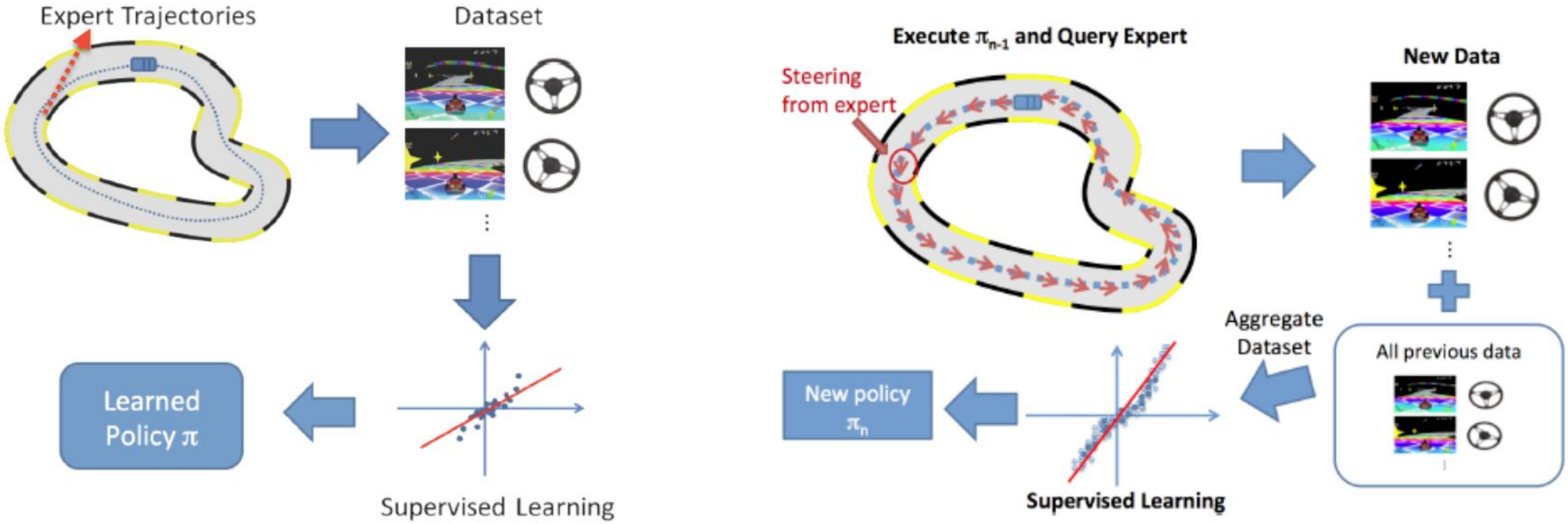




- A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots. Giusti et al.

监督式行为克隆

- By interactively querying the experts in additional datapoints



This assumes you can actively access an expert during training!

监督式行为克隆

- By interactively querying the experts in additional datapoints

算法 3.8：DAgger 算法

目标：最优策略参数 θ^*

Step 1：采集专家轨迹 $\left\{\tau_E^{(n)}\right\}_{n=1}^{N}$。

Step 2：从专家轨迹 $\left\{\tau_E^{(n)}\right\}_{n=1}^{N}$ 中随机提取 M 个数据对：$\mathcal{D}=\left\{\left(s_i^*, a_i^*\right)\right\}_{i=1}^{M}$

Step 3：初始化：令 $k=0$，

Step 4：$k=k+1$。

Step 5：利用监督式学习算法，通过求解优化问题求解参数

$$\theta^{(k)}=\underset{\theta}{\operatorname{argmax}} \sum_{(s,a)\in\mathcal{D}} \log \pi_\theta(a \mid s)$$

Step 6：如果 $k=$ MAX_ITE，令 $\theta^*=\theta^{(k)}$，退出循环。

Step 7：在环境中执行 $\pi_{\theta^{(k)}}$，得到新的数据样本集 $\left\{s_j^{(\text{new})}\right\}_{j=1}^{K}$

Step 8：通过向专家咨询，对集合 $\left\{s_j^{(\text{new})}\right\}_{j=1}^{K}$ 中的每个元素 $s_j^{(\text{new})}$ 逐一给出对应的动作 $a_j^{(new)}$，

从而形成新的数据集 $\mathcal{D}^{(\text{new})}=\left\{\left(s_j^{(\text{new})}, a_j^{(\text{new})}\right)\right\}_{j=1}^{K}$

Step 9：数据聚合：$\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}^{(\text{new})}$

Step 10：返回 Step 3。

Ross S, Gordon G, Bagnell D. A reduction of imitation learning and structured prediction to no-regret online learning[C] //Proceedings of the fourteenth international conference on artificial intelligence and statistics. 2011: 627-635.

监督式行为克隆

- 存在问题：非马尔科夫行为

$$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$$

behavior depends only on current observation

$$\pi_\theta(\mathbf{a}_t|\mathbf{o}_1, \ldots, \mathbf{o}_t)$$

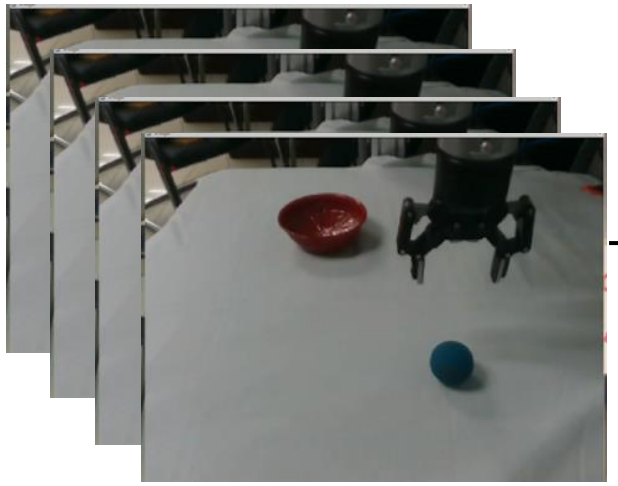
behavior depends on all past observations

If we see the same thing twice, we do the same thing twice, regardless of what happened before

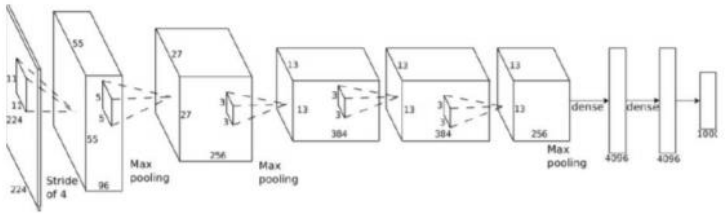
Often very unnatural for human demonstrators

监督式行为克隆

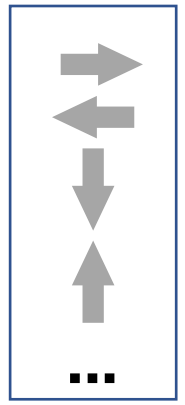
- 存在问题：非马尔科夫行为



s_t



$\pi(a_t|s_t)$

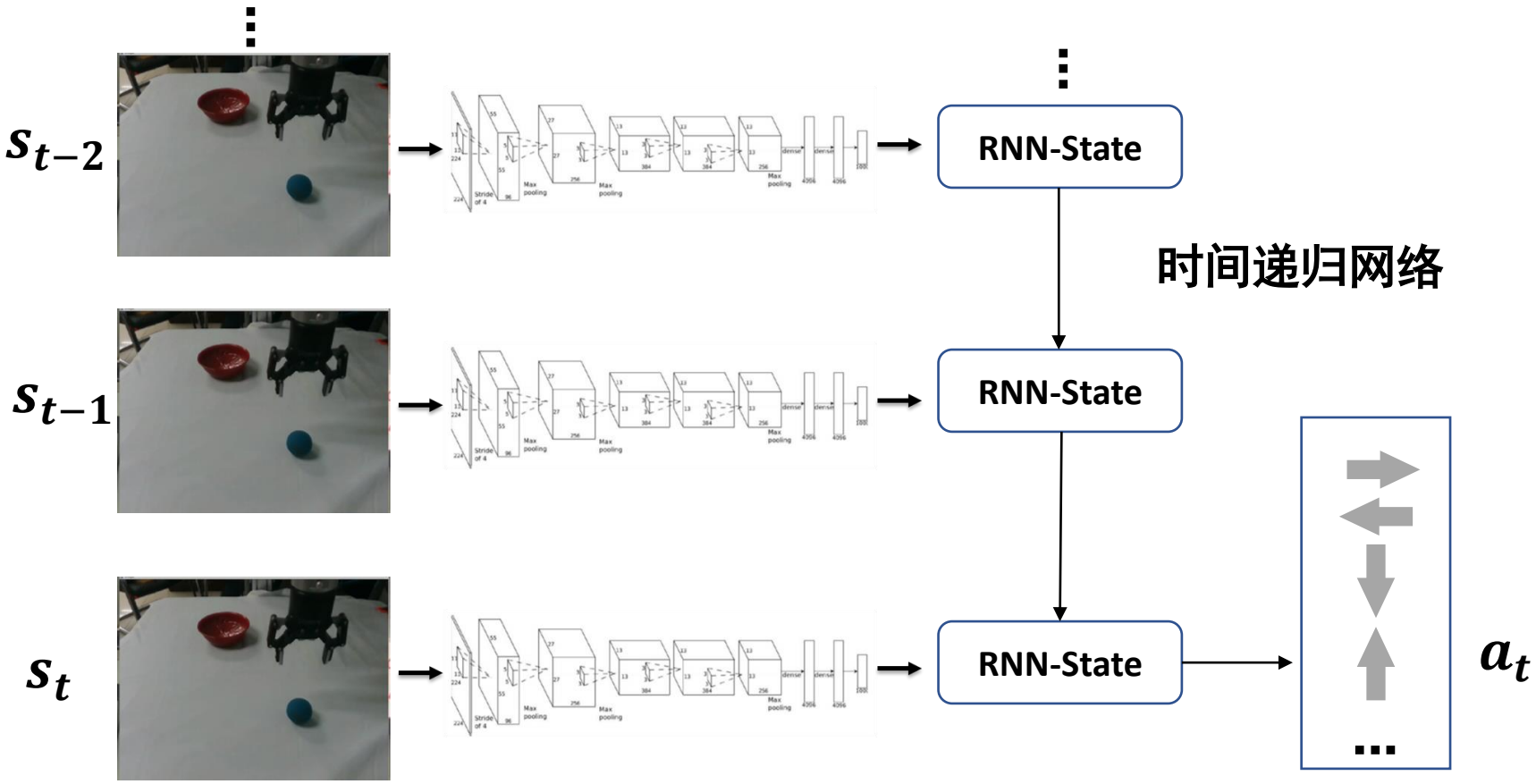


a_t

之前的多帧图像作为输入

监督式行为克隆

- 存在问题：非马尔科夫行为

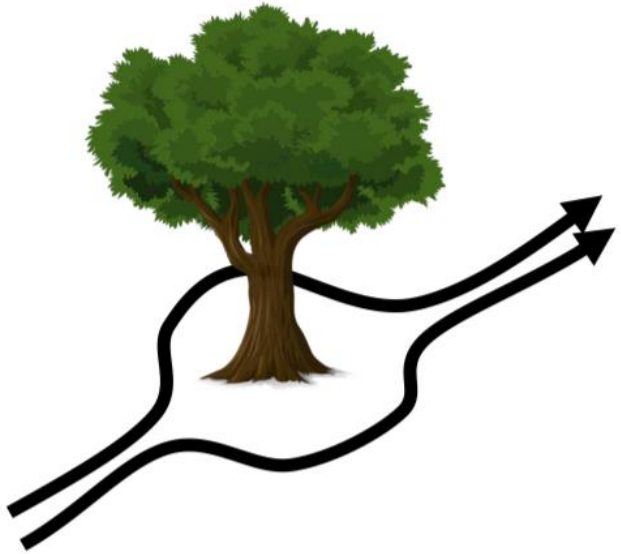


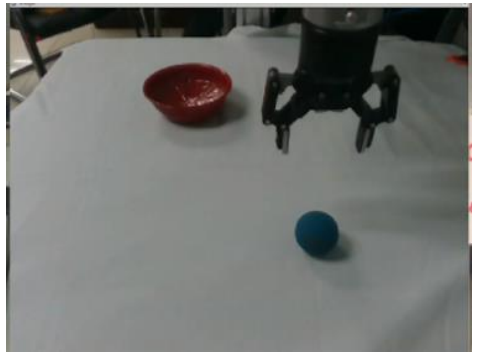
监督式行为克隆

- 存在问题：非马尔科夫行为

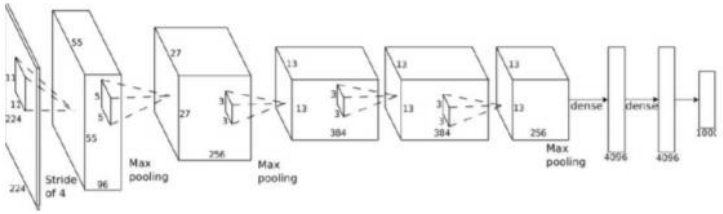
监督式行为克隆

- 存在问题：多模态专家示教

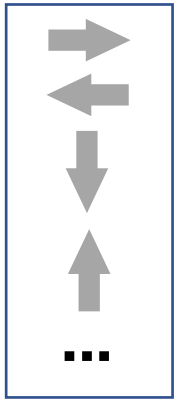




s_t



$\pi(a_t|s_t)$



a_t

Policy Confusion

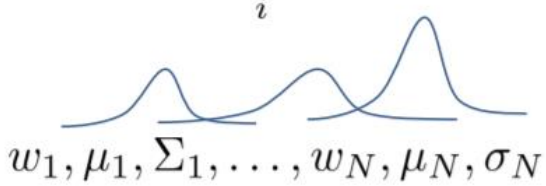
在同一状态下对应不同动作

监督式行为克隆

➢ 存在问题：多模态专家示教

1. **Output mixture of Gaussians**
2. **Latent variable models**
3. **Autoregressive discretization**

$$\pi(\mathbf{a}|\mathbf{o}) = \sum_i w_i \mathcal{N}(\mu_i, \Sigma_i)$$



$$w_1, \mu_1, \Sigma_1, \ldots, w_N, \mu_N, \sigma_N$$

- Conditional variational autoencoder
- Normalizing flow/realNVP
- Stein variational gradient descent

监督式行为克隆

➢ 存在问题：多模态专家示教

Input: sequence of context x_t

- e.g., noisy player detection

State: $s_t = (x_{t:t-K}, a_{t-1:t-K})$

- recent context and actions

Goal: learn $\pi(s_t) \rightarrow a_t$

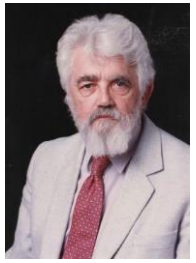
- Imitate expert

- Le et al., ICML ‘16

监督式行为克隆

符号主义

Marvin Lee Minsky
(1927-2016)



John McCarthy
(1927-2011)

强化学习不是真的人工智能，因为它缺乏符号表示和逻辑推理，无法处理抽象和复杂的问题

联接主义

Hilton
（1947-）

Yann LeCun

强化学习是真的人工智能，因为它模仿了人类大脑的神经网络，可以通过自适应和学习来解决各种问题

行为主义

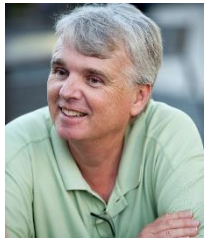
Brooks
(1954-)

Hans Moravec

强化学习是真的人工智能，因为它关注了智能体的行为和环境的交互，通过反馈和探索实现自主和适应

统计主义

Judea Pearl
（1936-）

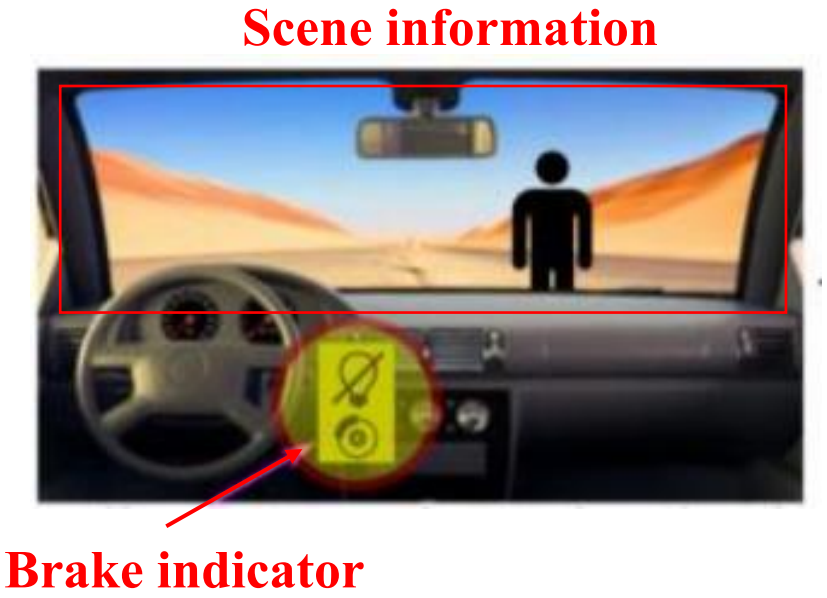


David Heckerman

强化学习不是真的人工智能，因为它缺乏概率模型和因果推断，无法处理不确定性和复杂性

监督式行为克隆

- 存在问题：因果性？



Causal Confusion in state representation

De Haan P, Jayaraman D, Levine S. Causal confusion in imitation learning[C]// NeurIPS2019: 11693-11704.

监督式行为克隆

➢ 缺陷

实质上可能只是学到一种简单的对应关系，但并没有学到更深层次的知识。而且在学习过程中只是一味地模仿行为本身，并不保证是否能真正地优化奖励函数。这使得模型针对新的场景泛化性会比较差。此外，由于在状态中可能包括不同维度的信息，在学习过程中，有可能会找到错误的特征，从而拟合出错误的对应关系。

Content

- 示教与模仿
- 模仿学习问题定义
- 监督式行为克隆
- 逆强化学习
- 对抗式生成模仿学习

逆强化学习

- 动机

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim p_{\pi_\theta}(\tau)}\left(R(\tau)\right)$$

$$\pi_\theta^* = \arg\max_{\pi_\theta} \mathbb{E}_{\tau \sim p_{\pi_\theta}(\tau)}\left(R(\tau)\right)$$

- 很多时候我们很难设计出合理的奖励函数
- 通过给定专家轨迹来学习出这样的奖励函数

逆强化学习

➢ 基本准则

学徒学习：由专家采集到样本的累计回报应该比任何非专家采集的样本的累计回报要高

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim p_{\pi_\theta}(\tau)}\left(R(\tau)\right)$$

$$J(\pi_E) > J(\pi)$$

- **奖励函数的表示**

$$J(\pi_E) > J(\pi)$$

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim p_{\pi_\theta}(\tau)}\left(R(\tau)\right)$$

参数化 $r(s,a,s') = \vartheta^T \varphi(s,a,s')$

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim p_{\pi_\theta}(\tau)}\left(\sum_{t=0}^{\infty} \gamma^t \vartheta^T \varphi(s_t,a_t,s_{t+1})\right) = \vartheta^T \cdot \mathbb{E}_{\tau \sim p_{\pi_\theta}(\tau)}\left(\sum_{t=0}^{\infty} \gamma^t \varphi(s_t,a_t,s_{t+1})\right)$$

逆强化学习

➢ 学徒学习

$$J(\pi_\theta)=\mathbb{E}_{\tau\sim p_{\pi_\theta}(\tau)}\left(\sum_{t=0}^{\infty}\gamma^t\vartheta^T\varphi(s_t,a_t,s_{t+1})\right)=\vartheta^T\cdot\mathbb{E}_{\tau\sim p_{\pi_\theta}(\tau)}\left(\sum_{t=0}^{\infty}\gamma^t\varphi(s_t,a_t,s_{t+1})\right)$$

$$\mu(\pi_\theta)=\mathbb{E}_{\tau\sim p_{\pi_\theta}(\tau)}\left(\sum_{t=0}^{\infty}\gamma^t\varphi(s_t,a_t,s_{t+1})\right)$$

$$J(\pi_\theta)=\vartheta^T\mu(\pi_\theta)$$

$$\tau_E^{(n)}=\{s_0^{(n)},a_0^{(n)};s_1^{(n)},a_1^{(n)};s_2^{(n)},a_2^{(n)};\ldots;s_{H^{(n)}}^{(n)},a_{H^{(n)}}^{(n)}\}$$

$$\mu(\pi_E)\approx\frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{H^{(n)}}\gamma^t\varphi(s_t^{(n)},a_t^{(n)},s_{t+1}^{(n)})$$

$$\begin{aligned}&\max_{\delta,\vartheta}\quad\delta\\&\text{s.t.}\quad\vartheta^T\mu(\pi_E)\geq\vartheta^T\mu(\pi)+\delta\\&\qquad\|\vartheta\|_2\leq 1\end{aligned}$$

逆强化学习

➢ 学徒学习算法

算法 3.9：学徒学习算法

目标：最优奖励估计参数 ϑ^*

Step 1：采集专家轨迹 $\left\{\tau_E^{(n)}\right\}_{n=1}^{N}$，估计 $\mu(\pi_E)$

Step 2：初始化：令 $k=0$，随机产生一个策略 $\pi^{(k)}$，计算 $\mu(\pi^{(k)})$

Step 3：$k=k+1$。

Step 4：训练得到估计的奖励函数系数

$$\max_{\delta,\vartheta} \quad \delta$$

$$\text{s.t.} \quad \vartheta^T\mu(\pi_E) \geq \vartheta^T\mu(\pi)+\delta \qquad \pi \in \{\pi^{(0)},\ldots,\pi^{(k-1)}\}$$

$$\|\vartheta\|_2 \leq 1$$

Step 5：基于估计的 ϑ^T，构造奖励函数 $r_\vartheta(s,a)$，并利用强化学习算法学习最优策略 $\pi^{(k)}$

Step 6：如果 $\delta<\varepsilon$，则退出循环。

否则，返回 Step 3。

逆强化学习

- 学徒学习算法

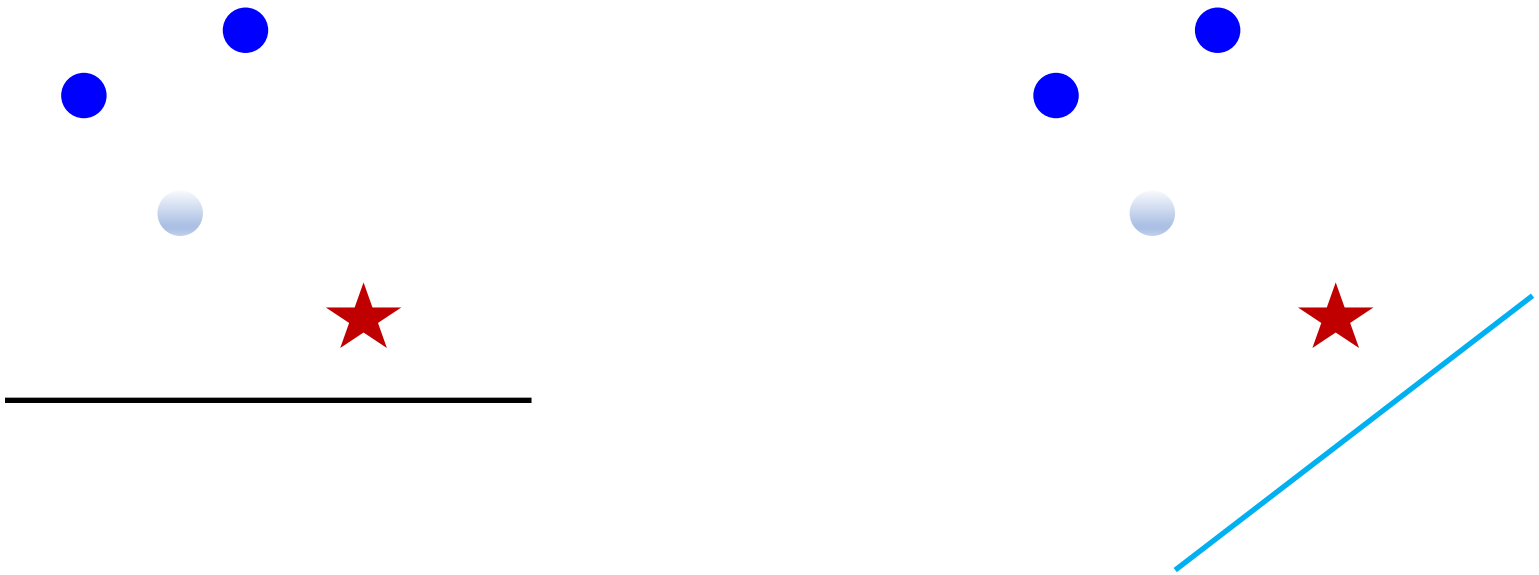
Enables robotic helicopters to teach themselves to fly difficult stunts by watching other helicopters perform the same maneuvers. The result is an autonomous helicopter than can fly dazzling stunts on its own.

➢ 前沿方法：对抗式生成模仿学习

$$RL(c) = \arg\min_{\pi \in \Pi} -H(\pi) + E_{\pi}[c(s,a)]$$

$$IRL(\pi) = \max_{c \in C} \left(\min_{\pi \in \Pi} -H(\pi) + E_{\pi}[c(s,a)] - E_{\pi_E}[c(s,a)] \right)$$

high cost to other policies　　low cost to expert policy



逆强化学习

➢ 前沿方法：对抗式生成模仿学习

$$\min_{\pi} \max_{D} E_{\pi}\left[\log\left(D(s,a)\right)\right] + E_{\pi_E}\left[\log\left(1 - D(s,a)\right)\right] - \lambda H(\pi)$$

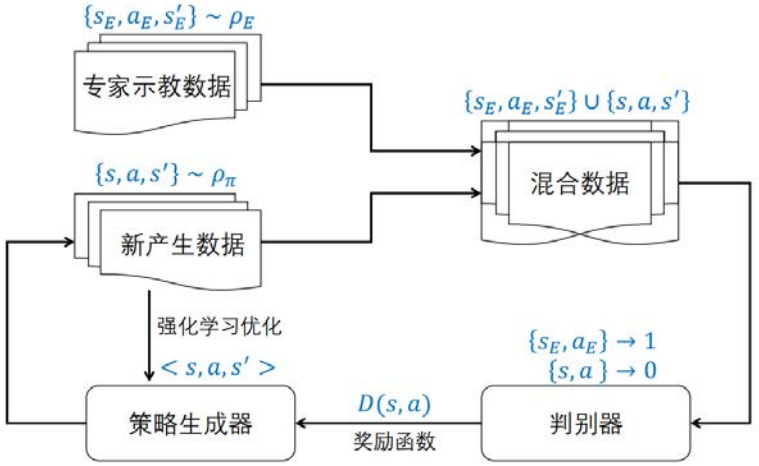
high cost to other policies

low cost to expert policy

Minimize expect of cost

Regular

GAIL: 一种对抗式生成模仿学习方法，将逆强化学习问题与对抗式生成网络结合，引入判别器区分专家与非专家样本，绕开复杂行为轨迹难以采样的问题，是模仿学习的里程碑工作

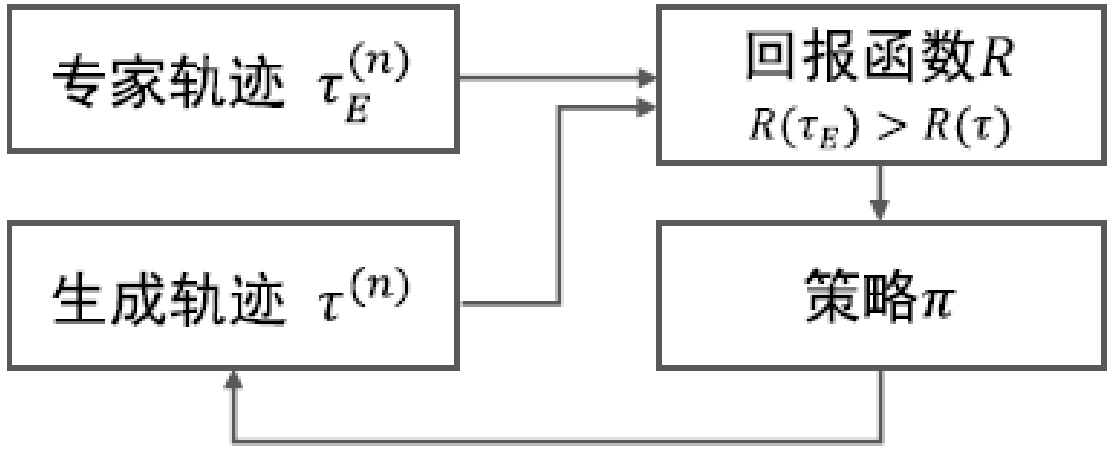


Ho, Jonathan, and Stefano Ermon. "Generative adversarial imitation learning." NIPS. 2016.

总结

专家轨迹：$\tau_E^{(n)} = \{s_1^{(n)}, a_1^{(n)}; s_2^{(n)}, a_2^{(n)}; \ldots; s_{H^{(n)}}^{(n)}, a_{H^{(n)}}^{(n)}\}$

行为克隆



学徒学习

逆强化学习

➢ 观测模仿学习

	直接示教	间接示教
示教效率	低	高
示教成本	高	低
示教安全性	人机耦合	人机解耦

1. 示教过程灵活
2. 高自由度技能
3. 海量示教视频

机器人模仿操作

自然示教演示

方法	优化目标函数
行为克隆 BC[92]	$\mathbb{E}_{\rho^{\exp}(s,a)}\left[D_{\mathrm{KL}}\pi^{\exp}(a\|s)\|\|\pi(a\|s)\right] = -\mathbb{E}_{\rho^{\exp}(s,a)}\left[\log \pi(a\|s)\right] + C$
DAgger[94]	在第 $n+1$ 次迭代时：$\mathbb{E}_{\rho^{\mathrm{agg}1:n}(s)}\left[\mathrm{KL}\left(\pi^{\exp}(a\|s)\|\|\pi(a\|s)\right)\right]$
AIRL[107]	$\mathrm{KL}(\rho^{\pi}(s,a)\|\|\rho^{\exp}(s,a)) = -\mathbb{E}_{\rho^{\pi}(s,a)}\left[\log \rho^{\exp}(s,a)\right] - \mathcal{H}(\rho^{\pi}(s,a))$
GAIL[26]	$D_{\mathrm{JS}}(\rho^{\pi}(s,a)\|\|\rho^{\exp}(s,a)) - \lambda\mathcal{H}^{\mathrm{causal}}(\pi)$
FAIRL[108]	$\mathrm{KL}(\rho^{\exp}(s,a)\|\|\rho^{\pi}(s,a)) = -\mathbb{E}_{\rho^{\exp}(s,a)}\left[\log \rho^{\pi}(s,a)\right] - \mathcal{H}(\rho^{\exp}(s,a))$
对称 f-散度[26]	$D_{f\text{-symm}}(\rho^{\pi}(s,a)\|\|\rho^{\exp}(s,a)) - \lambda\mathcal{H}^{\mathrm{causal}}(\pi)$
f-MAX[108]	$D_f(\rho^{\pi}(s,a)\|\|\rho^{\exp}(s,a))$
PWIL[109]	$D_{\mathcal{W}_1}(\hat{\rho}^{\pi}(s,a), \hat{\rho}^{\exp}(s,a))$
SIL[110]	$D_{\mathcal{W}_s^{\beta}}(\rho^{\pi}(s,a), \rho^{\exp}(s,a))_{c_w}$
GWIL[111]	$D_{\mathcal{GW}}(\pi,\pi') = \mathcal{GW}\left((S_E \times A_E, d_E, \rho_{\pi_E}), (S_A \times A_A, d_A, \rho_{\pi_A})\right)$

- 提出了理论完备的观测模仿学习，在现有的模仿学习框架下，利用逆动力学差异度，理论上刻画了模仿学习与观测模仿学习的差异。
- 推导出基于最大熵准则下逆动力学差异度的优化上界，为实际求解该问题提供高效解决方案。

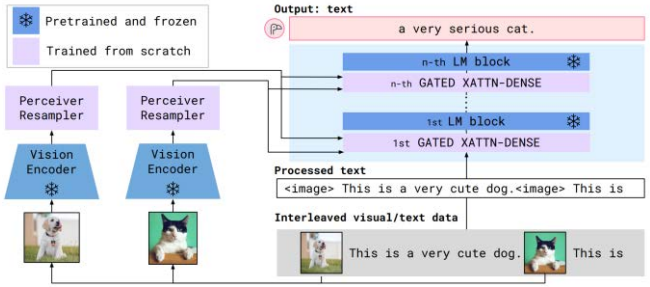
$$\underbrace{D_f(\rho_\pi(a|s,s')||\rho_E(a|s,s'))}_{\text{IDD}} = \underbrace{D_f(\rho_\pi(s,a)||\rho_E(s,a))}_{\text{GAIL}} - \underbrace{D_f(\rho_\pi(s,s')||\rho_E(s,s'))}_{\text{GAIfO}}$$

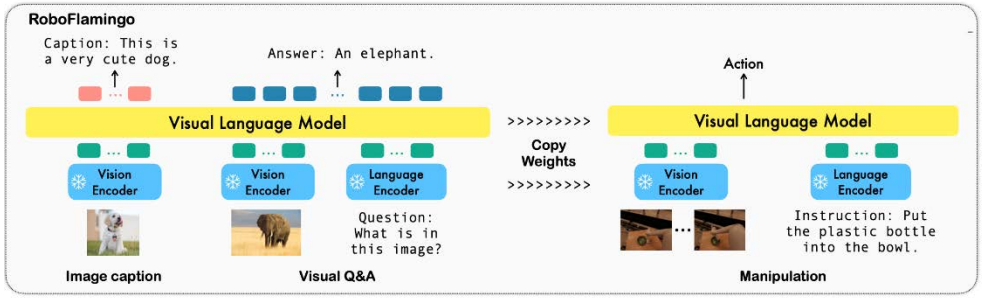
$$\boxed{\text{IDD}} := D_f(\rho_\pi(a|s,s')||\rho_E(a|s,s')) \leq -\mathcal{H}_\pi(s,a) + const.$$

- Imitation Learning from Observations by Minimizing Inverse Dynamics Disagreement. NeurIPS 2019 (Oral)

前沿工作

➢ 基于多模态基础模型的模仿学习

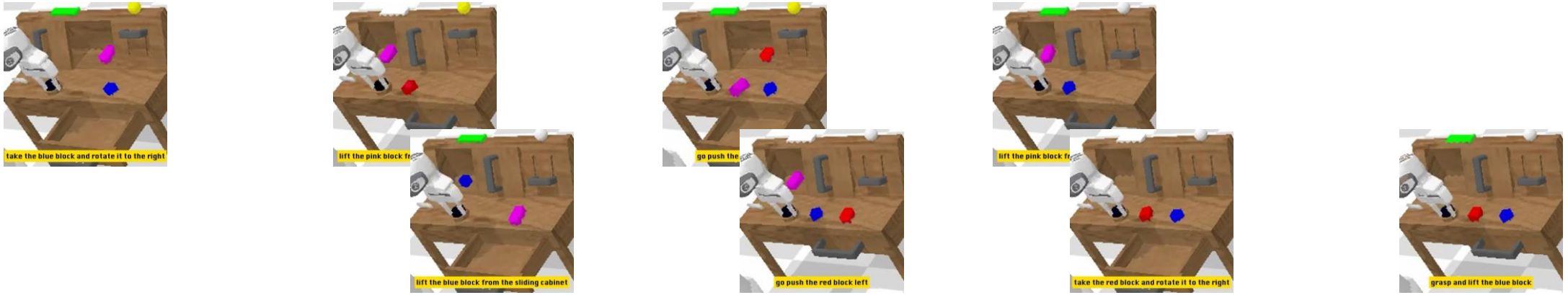




👉RoboFlamingo 在各种setting下达到了最佳性能, e.g., 任务平均成功长度（average sequence of completing 5 tasks） 从 3.06 提升到了4.06.

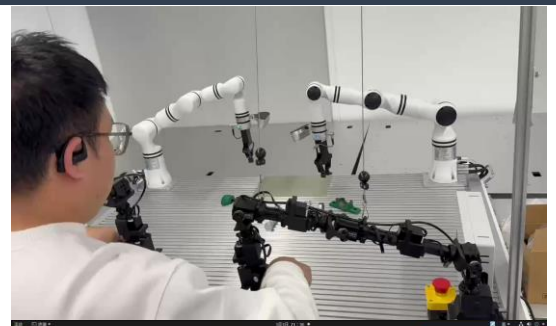
👉**只用了1% 带语言标注的数据, RoboFlamingo 超过了其他baseline.**

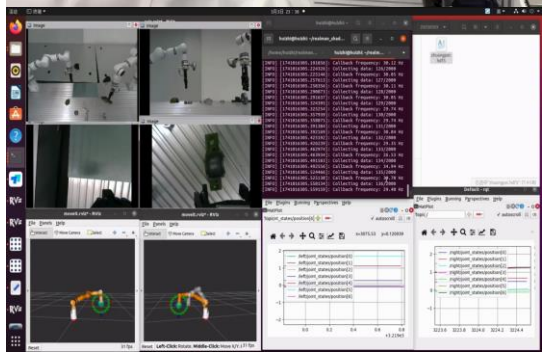
👉RoboFlamingo 在**未见过的场景中**显著提升了成功率和任务平均成功长度



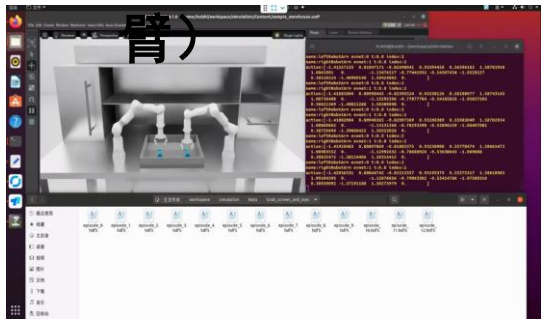
- RoboFlamingo: Vision-Language Foundation Models as Effective Robot Imitators, ICLR, 2024, Spotlight

模仿学习系统



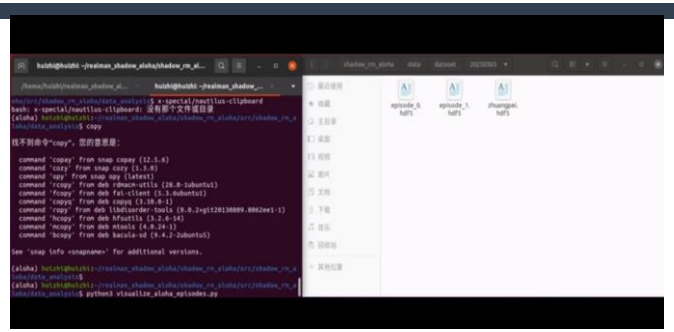


（真实机械臂）

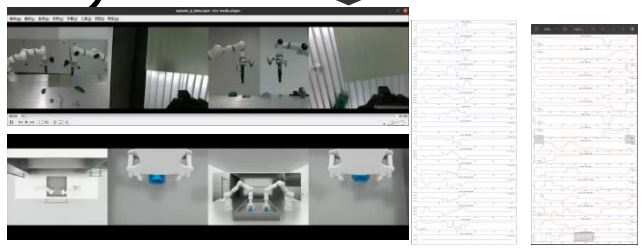


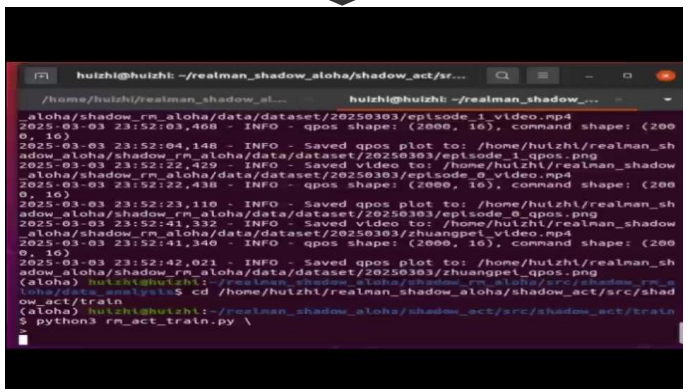
（仿真环境）

数据采集

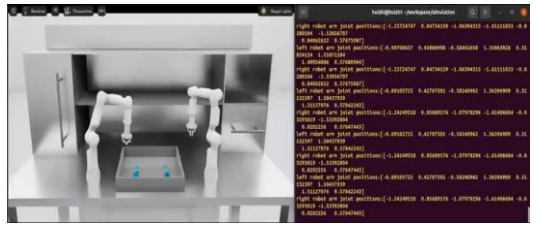


数据可视化（视频+曲线）

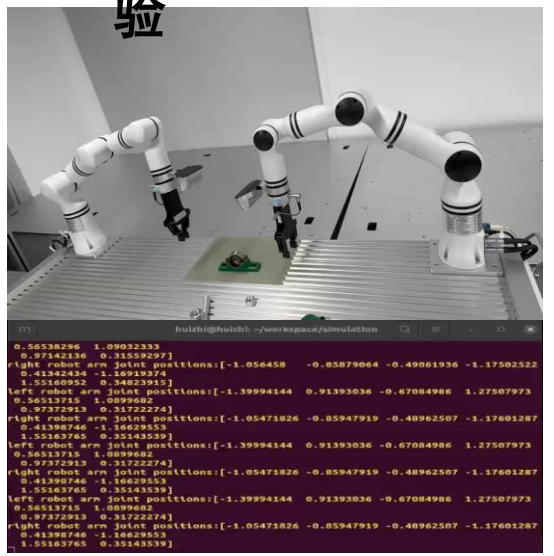




模型训练



仿真环境试验



模型本地推理