WDS

DataScience@Web

第三部分

策略梯度

汇报人：庄玥

2021.07.20

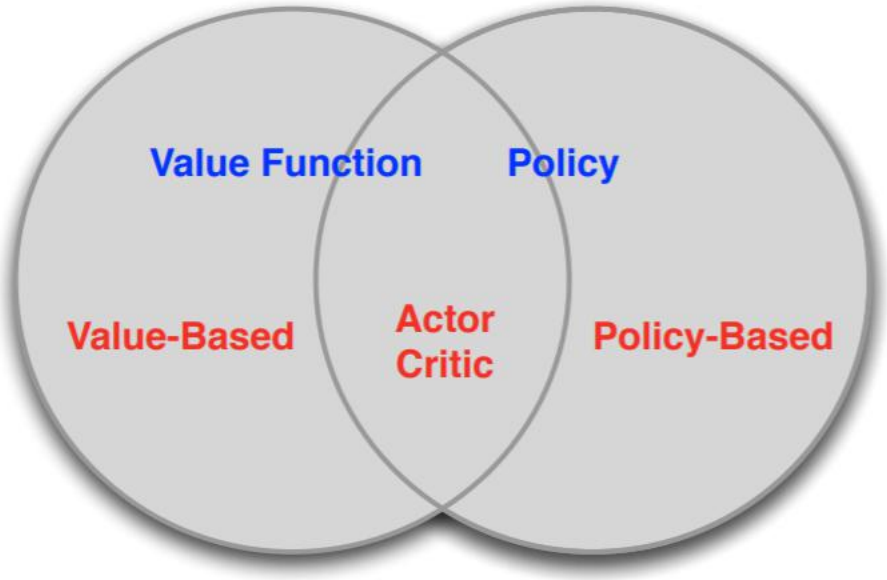
2021WDS暑期讨论班

目录

目录

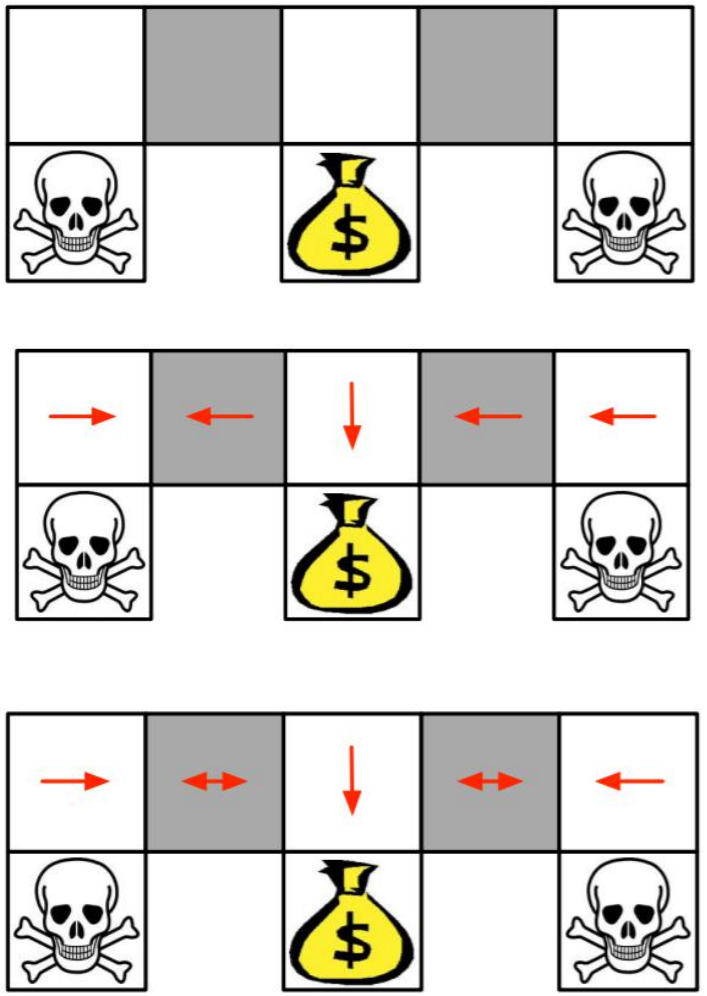
引言

- Value Based
  - 显示学习价值函数
  - 隐含策略
- Policy Based
  - 没有价值函数
  - 直接学习策略
- Actor-Critic
  - 学习价值函数
  - 学习策略



引言

- 基于值函数的缺点
  - 无法很好的处理连续空间的动作问题
  - 不同的两个状态在我们建模后拥有相同的特征描述



引言

■ 基于值函数的缺点

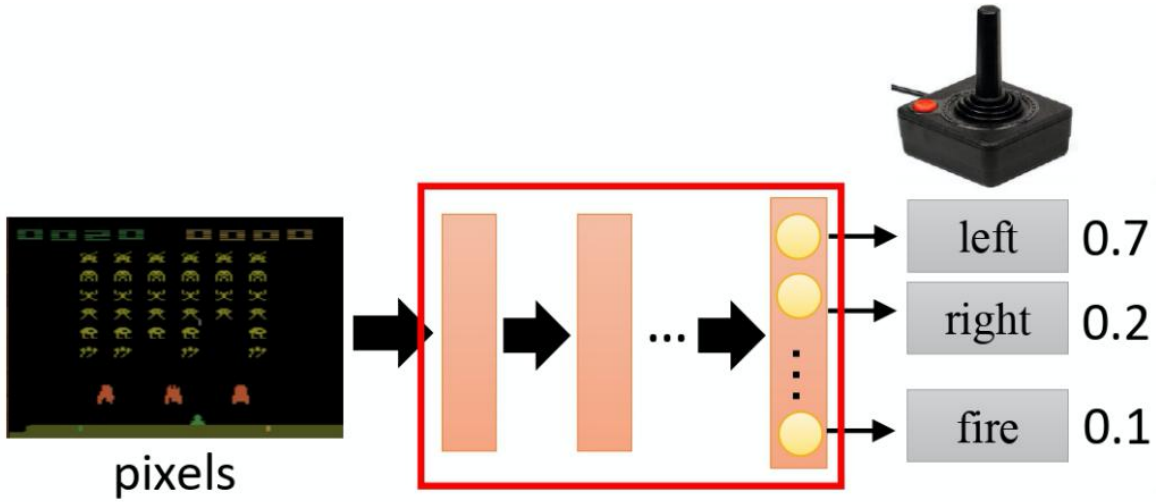
- 无法很好的处理连续空间的动作问题
- 不同的两个状态在我们建模后拥有相同的特征描述
- 无法解决随机策略问题

目录

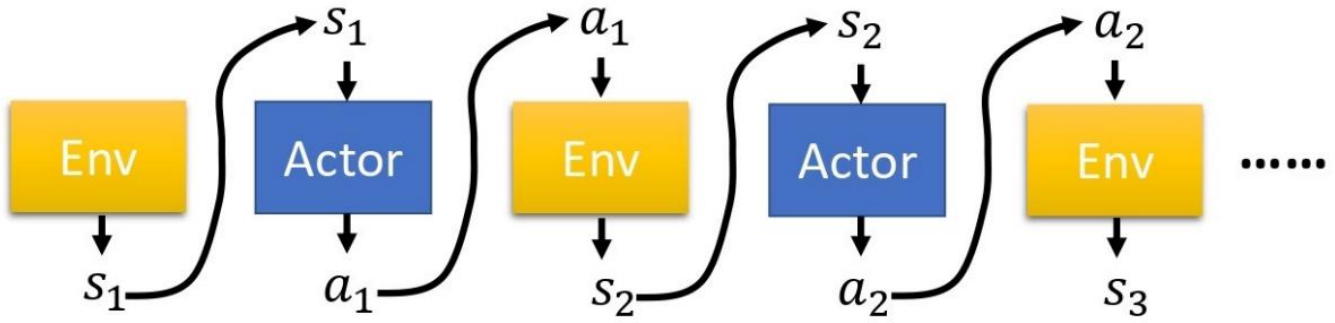
策略梯度

■ 策略的优化

- 目标：给定一个带有参数θ的策略 $\pi_\theta(s,a)$，寻找最佳的参数θ
- 如何衡量策略的优劣呢？



策略梯度



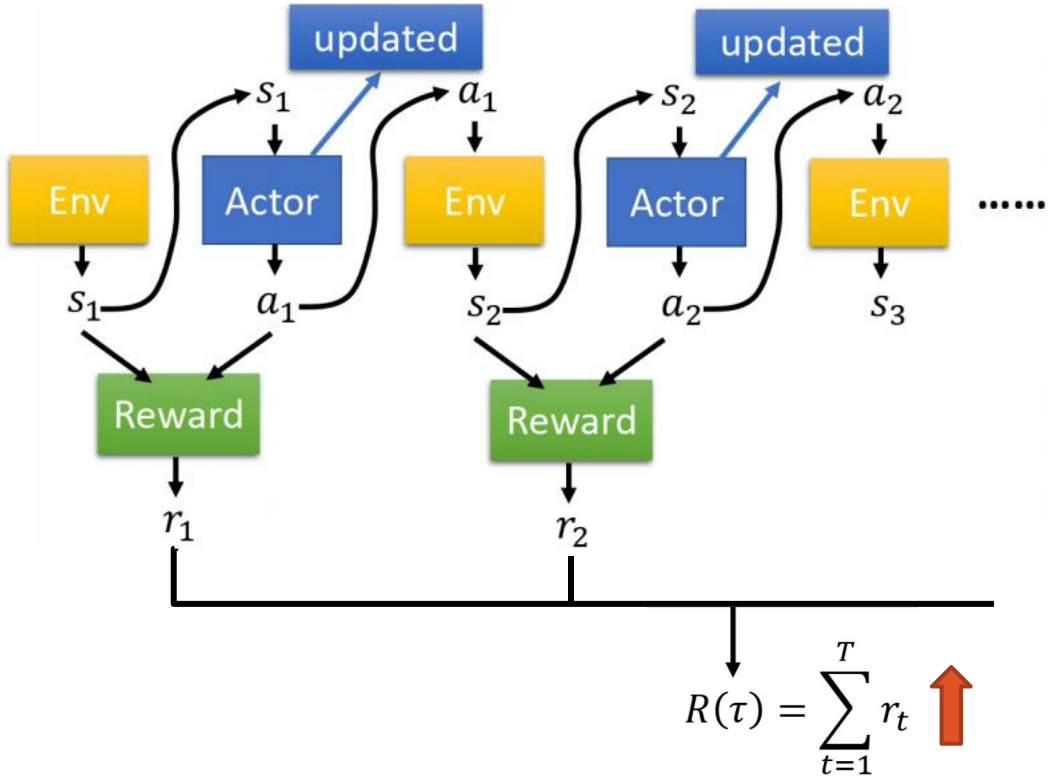
$$\text{Trajectory } \tau = \{s_1, a_1, s_2, a_2, \cdots, s_t, a_t\}$$

$$p_\theta(\tau) = p(s_1)\, p_\theta(a_1|s_1)\, p(s_2|s_1, a_1)\, p_\theta(a_2|s_2)\, p(s_3|s_2, a_2) \cdots$$

$$= p(s_1) \prod_{t=1}^{T} p_\theta(a_t|s_t)\, p(s_{t+1}|s_t, a_t)$$

Actor　　Environment

策略梯度



$$\bar{R}_\theta = \sum_\tau R(\tau) p_\theta(\tau) = E_{\tau \sim p_\theta(\tau)}[R(\tau)]$$

梯度上升 (gradient ascent)

策略梯度

$$\bar{R}_\theta = \sum_\tau R(\tau) p_\theta(\tau) = E_{\tau \sim p_\theta(\tau)}[R(\tau)]$$

$$\nabla \bar{R}_\theta = \sum_\tau R(\tau) \nabla p_\theta(\tau)$$

$$= \sum_\tau R(\tau) p_\theta(\tau) \frac{\nabla p_\theta(\tau)}{p_\theta(\tau)}$$

$$= \sum_\tau R(\tau) p_\theta(\tau) \nabla \log p_\theta(\tau)$$

$$= E_{\tau \sim p_\theta(\tau)} [R(\tau) \nabla \log p_\theta(\tau)]$$

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

策略梯度

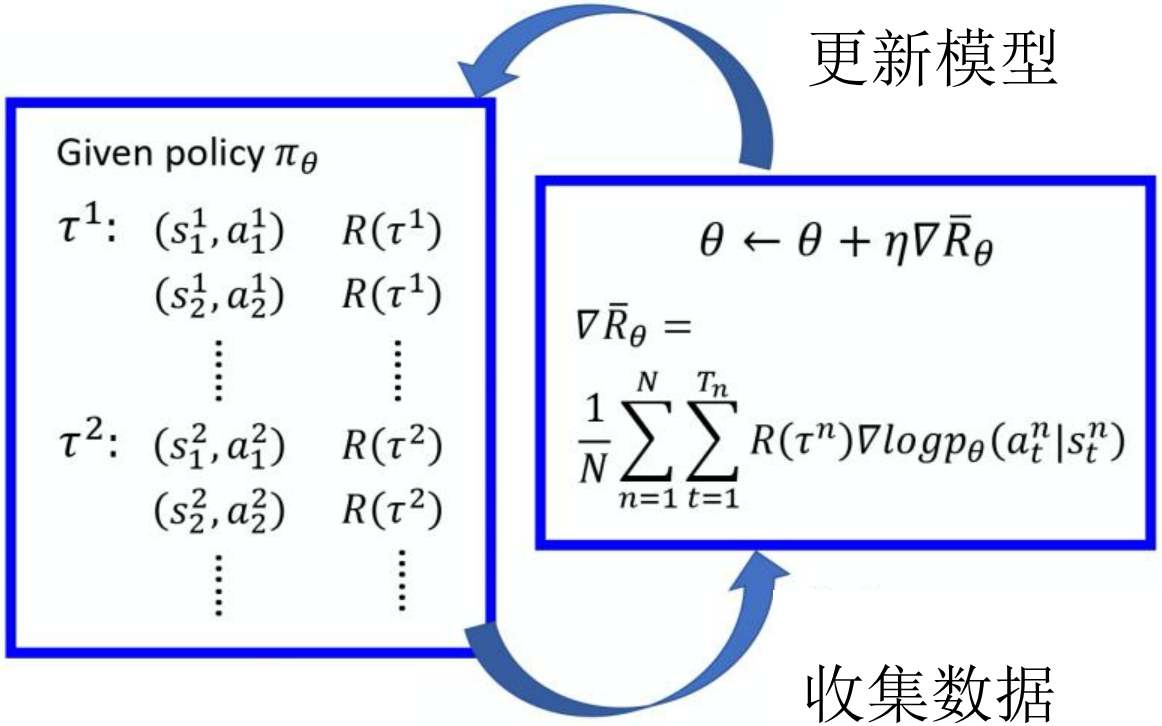
$$\nabla \log p(s_1) = 0 \qquad \nabla \sum_{t=1}^{T} \log p(s_{t+1}|s_t, a_t) = 0$$

$$\nabla \log p_\theta(\tau) = \nabla \left(\cancel{\log p(s_1)} + \sum_{t=1}^{T} \log p_\theta(a_t|s_t) + \sum_{t=1}^{T} \cancel{\log p(s_{t+1}|s_t, a_t)} \right)$$

$$E_{\tau \sim p_\theta(\tau)} \left[R(\tau) \nabla \log p_\theta(\tau) \right] \approx \frac{1}{N} \sum_{n=1}^{N} R\left(\tau^n\right) \nabla \log p_\theta\left(\tau^n\right)$$

$$= \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} R\left(\tau^n\right) \nabla \log p_\theta\left(a_t^n \mid s_t^n\right)$$

策略梯度



- 采集数据只使用一次——近段策略优化（PPO）

策略梯度

■ 公式改进

- 奖励往往都是正值 —— 引入baseline

$$\nabla \bar{R}_\theta = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} R(\tau^n)\nabla \log p_\theta(a_t^n \mid s_t^n)$$

$$\nabla \bar{R}_\theta = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} (R(\tau^n)-b)\nabla \log p_\theta(a_t^n \mid s_t^n)$$

$$b \approx E[R(\tau)]$$

目录

REINFORCE算法

■ PolicyGradeint算法

- 蒙特卡洛MC —— REINFORCE

$$\nabla \bar{R}_\theta \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} G_t^n \nabla \log \pi_\theta(a_t^n \mid s_t^n)$$

- 时序差分TD —— Actor-Critic

$$\nabla \bar{R}_\theta \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} Q^n(s_t^n, a_t^n) \nabla \log \pi_\theta(a_t^n \mid s_t^n)$$

REINFORCE算法

- 蒙特卡洛策略梯度

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

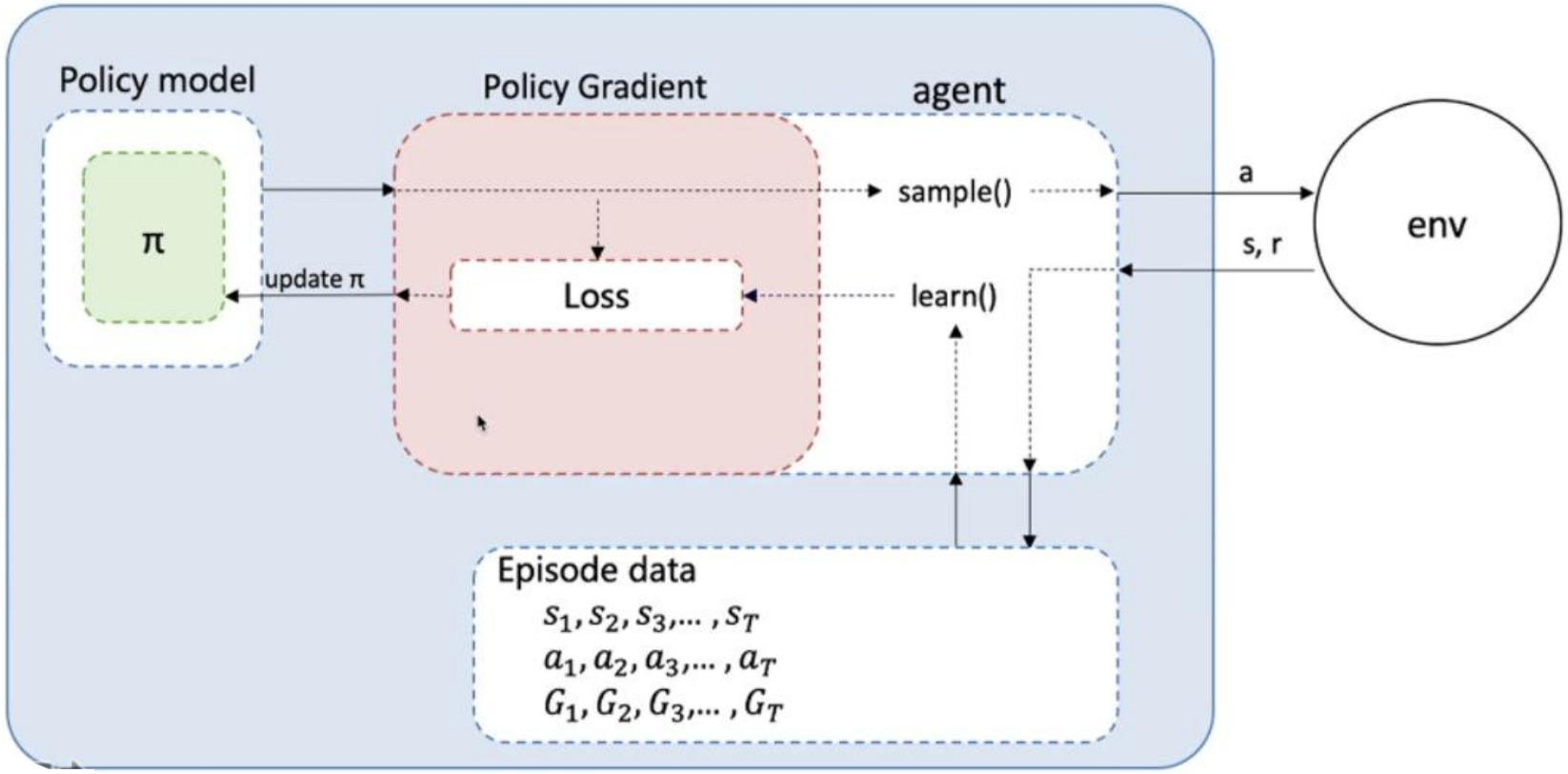
Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot,\boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ 未来总收益 (G_t)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\gamma^t G \nabla \ln \pi(A_t|S_t,\boldsymbol{\theta})$ 随机梯度上升

(s_1, a_1, G_1) (s_2, a_2, G_2) …… (s_T, a_T, G_T)

REINFORCE算法

■ REINFORCE流程图



REINFORCE算法

■ REINFORCE简单的扩展

- 方差较大 —— 加入不依赖于a的baseline

$$\sum_a b(s)\nabla\pi(a|s,\boldsymbol{\theta}) = b(s)\nabla\sum_a \pi(a|s,\boldsymbol{\theta}) = b(s)\nabla 1 = 0$$

$$\nabla J(\boldsymbol{\theta}) \propto \sum \mu(s) \sum \left(q_\pi(s,a) - b(s)\right) \nabla\pi(a|s,\boldsymbol{\theta})$$

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha\left(G_t - b\left(S_t\right)\right) \frac{\nabla\pi\left(A_t|S_t,\boldsymbol{\theta}_t\right)}{\pi\left(A_t|S_t,\boldsymbol{\theta}_t\right)}$$

目录

总结

- 策略梯度的优点
  - 基于策略的学习可能会具有更好的收敛性
  - 在高维度或连续状态空间使用基于策略的学习较为高效
  - 能够学习随机策略
- 策略梯度的缺点
  - 原始基于策略的学习有时候效率不高
  - 具有较高的变异性

Thanks!