Policy Gradient Algorithms

PPO、GRPO的理论基础

策略梯度

与Reinforce算法

东川路第一可爱猫猫虫

主要内容

感谢粉丝大佬
熊大之光
的充电

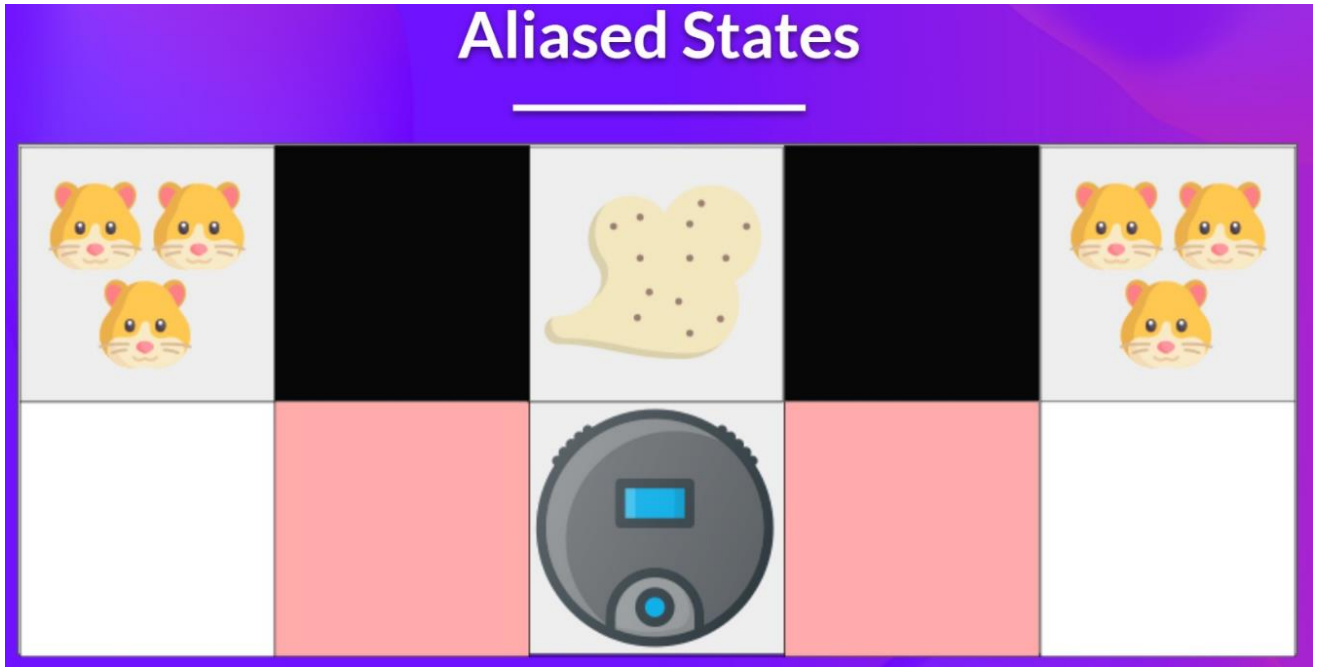
- Value-based方法遇到的问题
- 问题的本质
- 如何解决
- policy-based方法
- 策略梯度方法
- 梯度下降与梯度上升
- 策略梯度定理
- Reinforce算法

Value-based方法遇到的问题

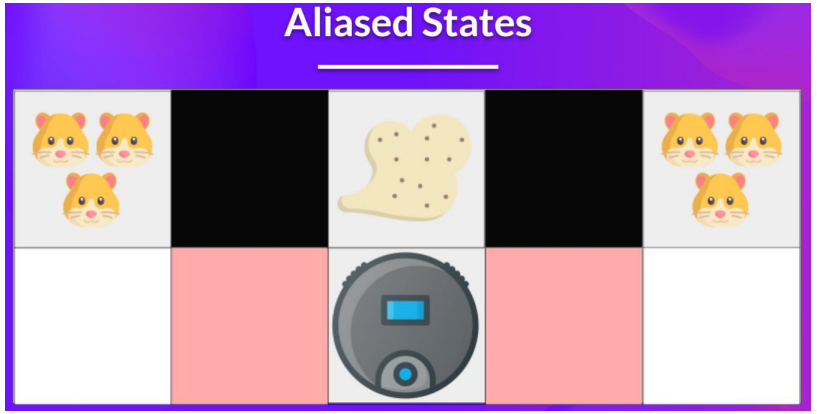
- Q-learning等基于value的方法里

  我们最小化损失函数来逼近Q函数

  这里的策略是隐式的，是基于值函数的



吸尘器避仓鼠找灰尘



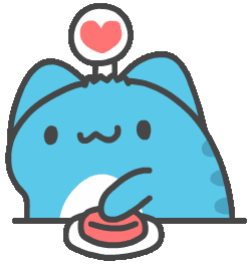
- 两个红色区域里，智能体的感知态是相同的
- Q (s,a) 的输入是智能体的感知态s，而非环境的真实状态
- 所有共享同一感知态s的真实状态（如 S_1和 S_2），其动作a的Q值会被混合计算

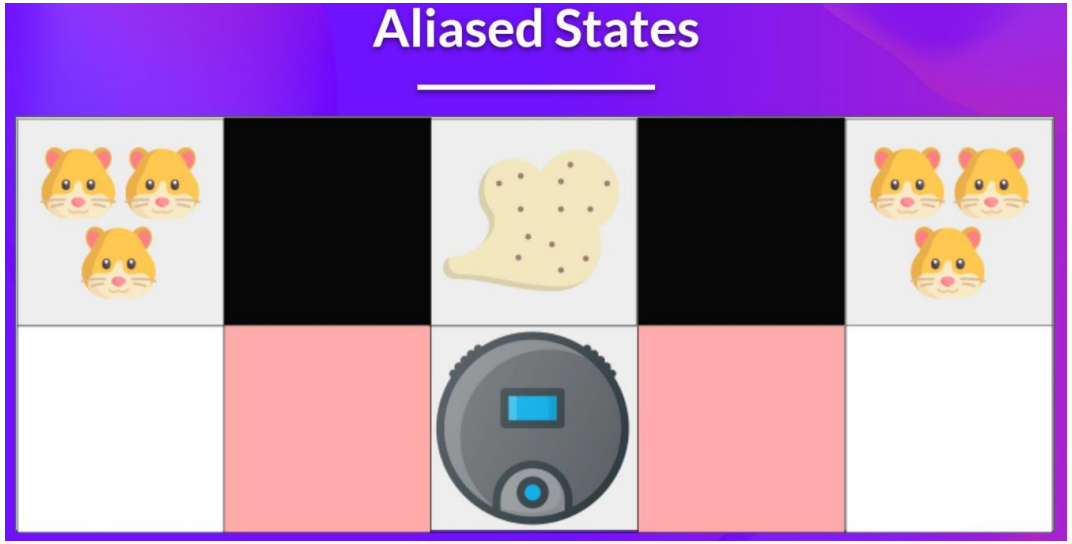
  perceptual aliasing 混淆感知态

  智能体无法为S_1和S_2分别学习左 / 右动作的价值
- S_1和S_2的经验会混合更新S_alias的Q 值
- 值函数方法的策略是 从 Q 值推导的副产品，本质是准确定性策略，无法在 S_alias 下稳定输出随机动作

问题的本质

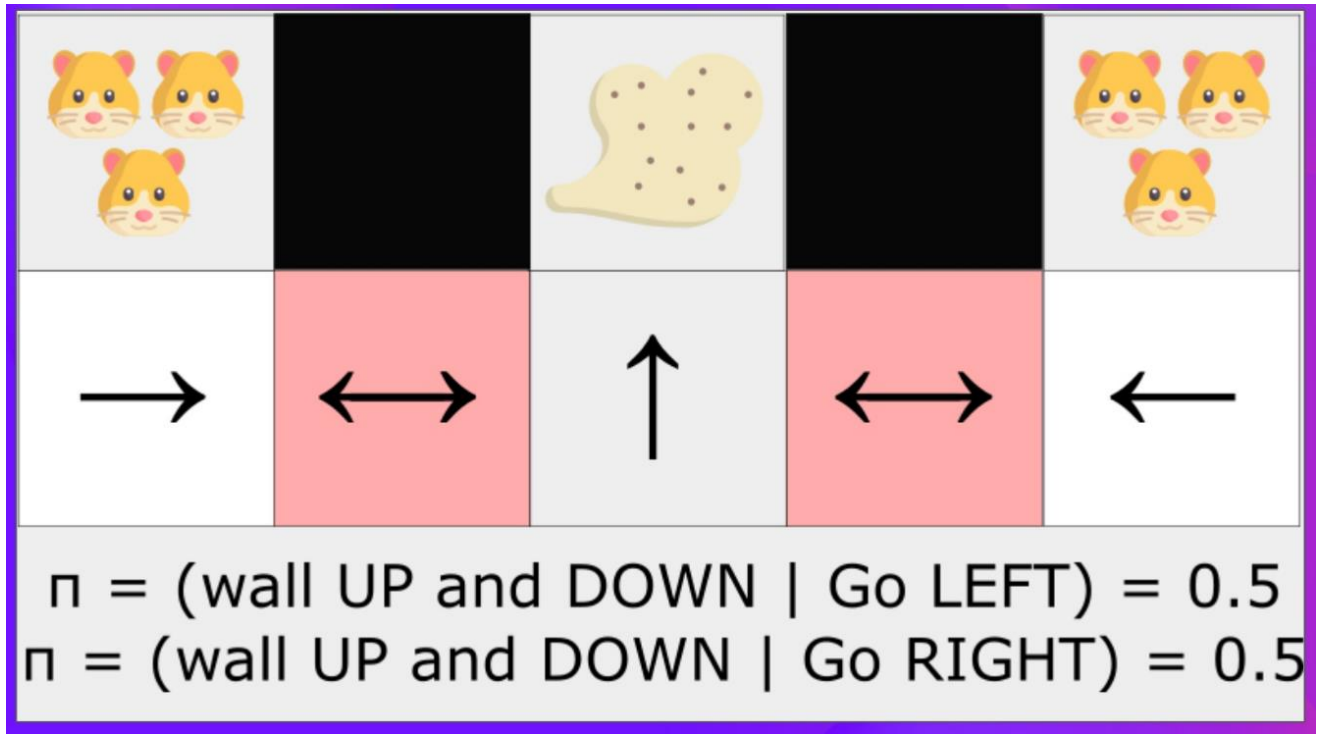
- Q 值与感知态的强绑定
  - 无法学习随机策略
  - 无法区分感知相同但需求不同的状态
  - 确定性（准确定性）策略无法应对非对称





如何解决

- 不再学习值函数，直接显式学习策略（policy-based)
- 这样就可以学习到随机的策略，而不会被一直卡住



policy-based

- policy-based方法

  直接优化策略

  如何实现：将策略参数化$\pi_\theta(s) = P[A|s;\theta]$

  通过优化θ，来让策略$\pi_\theta(s)$变好

  这样我们可以得到一个随机的策略

  输出一个动作的概率分布

  定义一个目标函数$J(\theta)$

  $J(\theta)$代表的是预期的累计奖励

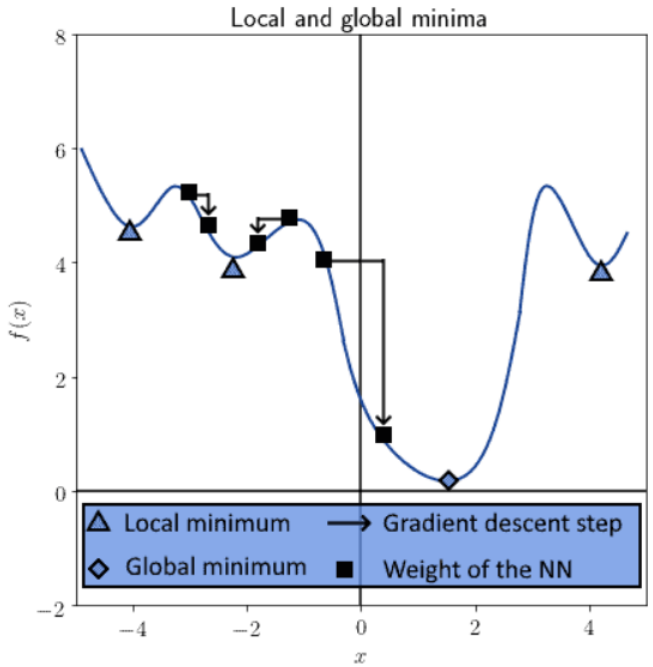
  接下来，我们让$J(\theta)$最大，就找到了最优策略

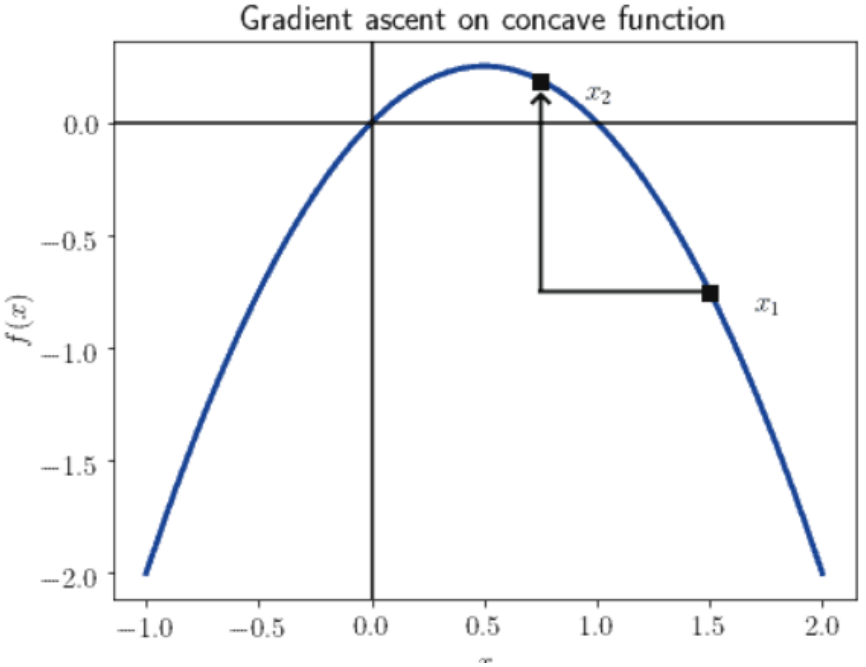
策略梯度方法

- 通过梯度上升的方法让$J(\theta)$最大

  称为策略梯度

- 梯度下降与梯度上升





策略梯度方法

- 我们有了一个参数化的策略$\pi_\theta(s) = P[A|s;\theta]$
- 我们定义一个目标函数$J(\theta)$
- $J(\theta) = \Sigma_\tau P(\tau;\theta)R(\tau)$

  τ代表一个trajectory轨迹
- 要用梯度上升的方法来maximize $J(\theta)$，我们要先对$J(\theta)$求梯度
- $\nabla_\theta J(\theta) = \nabla_\theta \Sigma_\tau P(\tau;\theta)R(\tau)$

  $= \Sigma_\tau \nabla_\theta P(\tau;\theta)\, R(\tau)$

  $= \Sigma_\tau \frac{\nabla_\theta P(\tau;\theta)}{P(\tau;\theta)} P(\tau;\theta)R(\tau)$

$$\Sigma_\tau \frac{\nabla_\theta P(\tau;\theta)}{P(\tau;\theta)} P(\tau;\theta) R(\tau)$$

- 复合函数求梯度（求导）

$$\nabla_x log f(x) = \frac{\nabla_x f(x)}{f(x)}$$

- $\frac{\nabla_\theta P(\tau;\theta)}{P(\tau;\theta)} = \nabla_\theta \log P(\tau;\theta)$
- $\nabla_\theta J(\theta) = \Sigma_\tau P(\tau;\theta) \nabla_\theta \log P(\tau;\theta) R(\tau)$
- 期望的定义

$$E[X] = \Sigma_{所有可能结果\tau} P(\tau) X(\tau)$$

- $P(\tau;\theta)$是轨迹τ发生的概率 $\nabla_\theta \log P(\tau;\theta) R(\tau)$是每条轨迹对应的一个值

$$\nabla_\theta J(\theta) = \Sigma_\tau \, P(\tau;\theta) \, \nabla_\theta \log P(\tau;\theta) \, R(\tau)$$

- 本质上是一个期望
- $\nabla_\theta J(\theta) = \Sigma_\tau \, P(\tau;\theta) \, \nabla_\theta \log P(\tau;\theta) \, R(\tau)$

  $= E_{\tau \sim P(\tau;\theta)} \Sigma_\tau \nabla_\theta \log P(\tau;\theta) \, R(\tau)$
- 我们可以通过采样来近似
- 理论依据：大数定律

  从独立同分布的样本里采样，样本均值会收敛到理论的期望
- $\nabla_\theta J(\theta) = \frac{1}{m} \Sigma_{i=1}^{m} \nabla_\theta log P\left(\tau^{(i)};\theta\right) R\left(\tau^{(i)}\right)$

  其中$\tau^{(i)}$表示采样出来的一条轨迹

$\nabla_\theta log P\left(\tau^{(i)};\theta\right)$

- $P(\tau;\theta) = \prod_{t=0} P(s_{t+1}|s_t;a_t)\pi_\theta(a_t|s_t)$

$$P(\tau^{(i)};\theta) = \mu(s_0)\prod_{t=0}^{H} P\left(s_{t+1}^{(i)} \mid s_t^{(i)}, a_t^{(i)}\right)\pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right)$$

$$\nabla_\theta \log P(\tau^{(i)};\theta) = \nabla_\theta \log\left[\mu(s_0)\prod_{t=0}^{H} P\left(s_{t+1}^{(i)} \mid s_t^{(i)}, a_t^{(i)}\right)\pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right)\right]$$

$$\nabla_\theta \log P(\tau^{(i)};\theta) = \nabla_\theta\left[\log\mu(s_0) + \sum_{t=0}^{H}\log P\left(s_{t+1}^{(i)} \mid s_t^{(i)}, a_t^{(i)}\right) + \sum_{t=0}^{H}\log\pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right)\right]$$

$$\nabla_\theta \log\mu(s_0) + \nabla_\theta\sum_{t=0}^{H}\log P\left(s_{t+1}^{(i)} \mid s_t^{(i)}, a_t^{(i)}\right) + \nabla_\theta\sum_{t=0}^{H}\log\pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right)$$

$$\nabla_\theta \log P(\tau^{(i)};\theta) = \nabla_\theta\sum_{t=0}^{H}\log\pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right)$$

$$\nabla_\theta J(\theta) = \hat{g} = \frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{H}\nabla_\theta \log\pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right) R(\tau^{(i)})$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log\pi_\theta(a_t|s_t)R(\tau)]$$

策略梯度定理 $\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(a_t|s_t) R(\tau)]$

- 基于策略梯度定理

  Reinforce算法（蒙特卡洛策略梯度）
- 用当前策略走一条完整的轨迹
- 倒推计算这条轨迹中每个时间步的累计回报
- 相乘得到一条轨迹的梯度估计
- 沿着梯度上升的方向更新策略 $\theta \leftarrow \theta + \alpha \hat{g}$
- 重复上述步骤，直到策略收敛