

从AC到A2C

Actor-Critic算法

东川路第一可爱猫猫虫



主要内容

- Reinforce算法的问题
- 基于Q值的Actor-Critic算法
- Advantage Actor-Critic算法 (A2C)
- On-policy Off-policy

Reinforce算法的问题

- 方差大

无偏性的必然结果

它没有估算回报，采样的是每条轨迹的真实回报

- 机器学习里的偏差与方差

欠拟合：高偏差 过拟合：高方差

- 强化学习里的偏差与方差

Q-learning

Reinforce算法

Reinforce算法的问题

- 必须等轨迹结束才能更新

必须等一个episode结束才能求return
要求任务是有限步的

- 如果任务是无限步的

就无法使用Reinforce算法



Actor-Critic算法

- 用价值函数来替代MonteCarlo采样的总return, 指导策略更新
- Actor
 - 选动作
 - 是一个由 θ 参数化的策略函数 $\pi_\theta(a|s)$
- Critic
 - 评价动作的好坏
 - 用评估结果指导actor改进策略
- 如果Critic用Q值来计算误差
则定义为基于Q值的Actor-Critic算法

基于Q值的Actor-Critic算法

- Critic为价值网络
学习Q值，逼近真实的 $Q_{\pi}(s, a)$
- Q-learning
 - TD target: $r_t + \gamma \cdot \max Q(s_{t+1}, a'; w)$
 - TD error: $r_t + \gamma \cdot \max Q(s_{t+1}, a'; w) - Q(s_t, a_t; w)$
- 基于Q值的AC
 - TD target: $r_t + \gamma \cdot q(s_{t+1}, a_{t+1}; w)$
 - TD error: $r_t + \gamma \cdot q(s_{t+1}, a_{t+1}; w) - Q(s_t, a_t; w)$



QAC的流程

- Actor观测当前状态，按照当前策略随机执行一个动作
- Agent从环境获取即时反馈
- Actor按照当前的策略网络随机采样一个动作
- Critic用当前价值网络计算：
 - 当前状态动作对的估计价值 \hat{q}_t
 - 下一个状态动作对的估计价值 \hat{q}_{t+1}
- 计算TD Target: $\hat{y}_t = r_t + \gamma \cdot \hat{q}_{t+1}$
- TD Error: $\delta_t = \hat{q}_t - \hat{y}_t$
- 梯度下降，更新参数

On-policy Off-policy

- on-policy: the target and the behavior polices are the same
 产出数据的策略和用这批数据做更新的策略是同一个
 智能体一边用某个策略与环境互动、产生行动，一边直接用
 这些自己刚做的行动和反馈来更新这个策略
- off-policy: the learning is from the data off the target policy
 产出数据的策略和用这批数据做更新的策略不是同一个
 学习时只关注最优可能的行动反馈，或者直接用其他策略的
 数据来更新目标策略
- Actor-Critic属于on-policy

Advantage Actor-Critic算法

- A2C
- 通过优势函数 $A(s, a) = Q(s, a) - V(s)$
状态 s 下选动作 a , 比状态 s 的平均动作价值好多少
- 用 $r_t + \gamma V_\omega(s_{t+1})$ 来近似 Q
预测误差可以写成
$$\delta(t) = r_t + \gamma V_\omega(s_{t+1}) - V_\omega(s_t)$$
 $\delta(t)$ 为正, 说明当前 action 不错
 $\delta(t)$ 为负, 说明当前 action 不太行

Critic的目标：让自己的预测更准确

- Critic不断修正自己的估计

来让 $\delta(t)$ 最小化

这意味着 $V_\omega(s_t)$ 可以更精确表示从状态 s_t 出发的实际回报
批评家的评价越准确，演员的动作调整才越正确

- 如何让 $\delta(t)$ 最小化？

求损失函数

求梯度

梯度下降

损失函数求梯度

- 平方损失误差

$$L(\omega) = \frac{1}{2} (r_t + \gamma V_\omega(s_{t+1}) - V_\omega(s_t))^2$$

$$\nabla_\omega L(\omega) = -(r_t + \gamma V_\omega(s_{t+1}) - V_\omega(s_t)) \nabla_\omega V_\omega(s_t)$$

- 接下来使用梯度下降来更新参数

$$\theta = \theta + \alpha_\theta \sum_t \delta_t \nabla_\theta \log \pi_\theta(a_t | s_t)$$

$\nabla_\theta \log \pi_\theta(a_t | s_t)$ 衡量的是

参数 θ 调整时动作选择概率的变化率

