强化学习2022
第6节

涉及知识点：
参数化值函数近似、状态值函数与状态-动作值、
函数近似、策略梯度、Actor-Critic

价值和策略近似逼近方法

课程回顾

基于模型的动态规划

- 值迭代 $V(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s') V(s')$
- 策略迭代 $\pi(s) = \arg\max_{a \in A} \sum_{s' \in S} P_{sa}(s') V(s')$

无模型的强化学习

- 在线策略蒙特卡洛 $V(s_t) \leftarrow V(s_t) + \alpha(G_t - V(s_t))$
- 在线策略时序差分 $V(s_t) \leftarrow V(s_t) + \alpha\left(r_{t+1} + \gamma V(s_{t+1}) - V(s_t)\right)$
- 在线策略时序差分 SARSA学习

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

- 离线策略时序差分 Q-学习

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t))$$

参数化价值函数

本课程中解决的关键问题

- 之前所有模型的做法都是基于创建一个查询表，在表中维护状态值函数$V(s)$或状态-动作值函数$Q(s,a)$

- 当处理大规模马尔可夫决策过程（MDP）时，即：
  - 状态或者状态-动作空间非常大
  - 连续的状态或动作空间

  是否仍然需要为每一个状态维护$V(s)$或为每个状态-动作对维护$Q(s,a)$?
  - 例如
    - 围棋博弈（10^{170}的状态空间）
    - 直升机，自动驾驶汽车（连续的状态空间）

主要内容

- 大规模马尔可夫决策过程的解决方法
  - 对状态/动作进行离散化或分桶
  - 构建参数化的值函数估计

目录

Contents

01

对状态/动作进行离散化

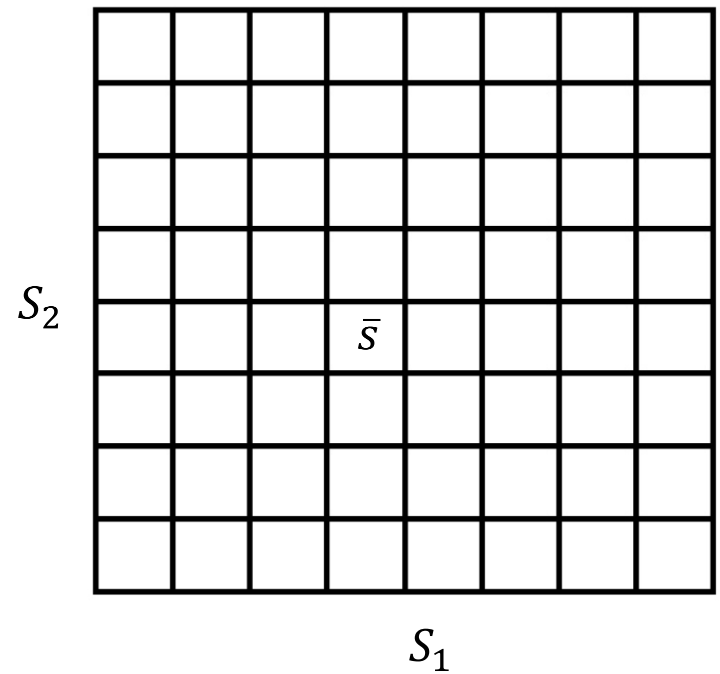
离散化连续马尔可夫决策过程

- 对于连续状态马尔可夫决策过程，我们可以对状态空间进行离散化

- 例如，如果用2维连续值 (s_1, s_2)表示状态，可以使用网格对状态空间进行切分从而转化为离散的状态值
- 记离散的状态值为$\bar{s}$
- 离散化的马尔可夫决策过程可以表示为：

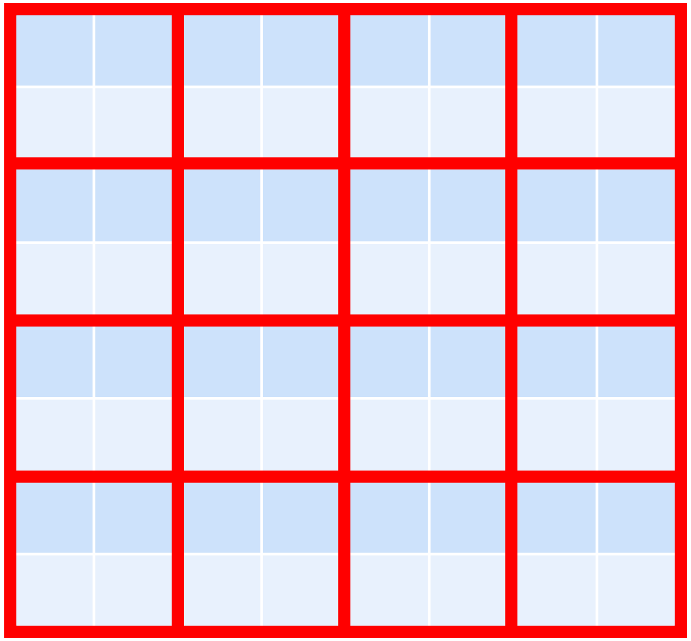
$$(\bar{S}, A, \{P_{\bar{s}a}\}, \gamma, R)$$

- 这样一来，就能够使用前述方法求解马尔可夫决策过程



对大型马尔可夫决策过程分桶

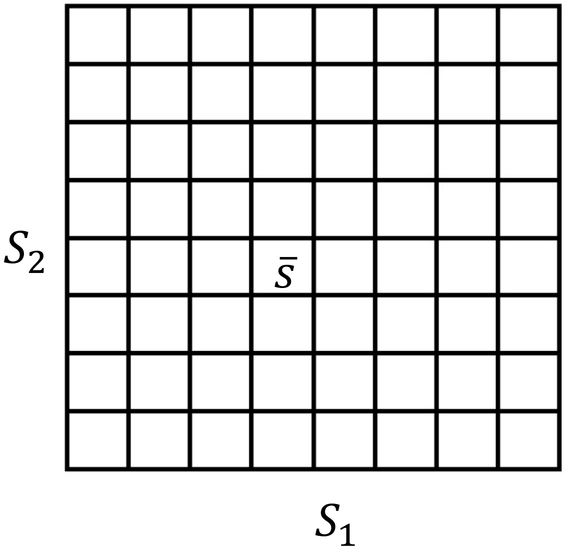
- 对于一个大型的离散状态马尔可夫决策过程，我们可以对状态值进一步分桶以进行采样聚合
  - 使用先验知识将相似的离散状态归类到一起
    - 例如，利用根据先验知识抽取出来的状态特征对状态进行聚类

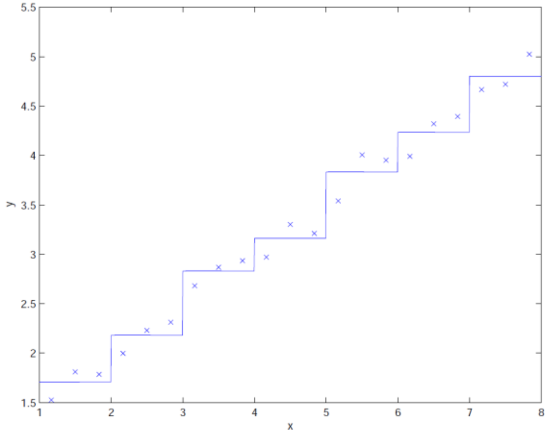


离散化/分桶

- 优点
  - 操作简洁直观
  - 高效
  - 在处理许多问题时能够达到较好效果
- 缺点
  - 过于简单地表示价值函数V
  - 可能为每个离散区间假设一个常数值
  - 维度灾难

$$S = R^n \Rightarrow \bar{S} = \{1, \dots, k\}^n$$





02

参数化
价值函数

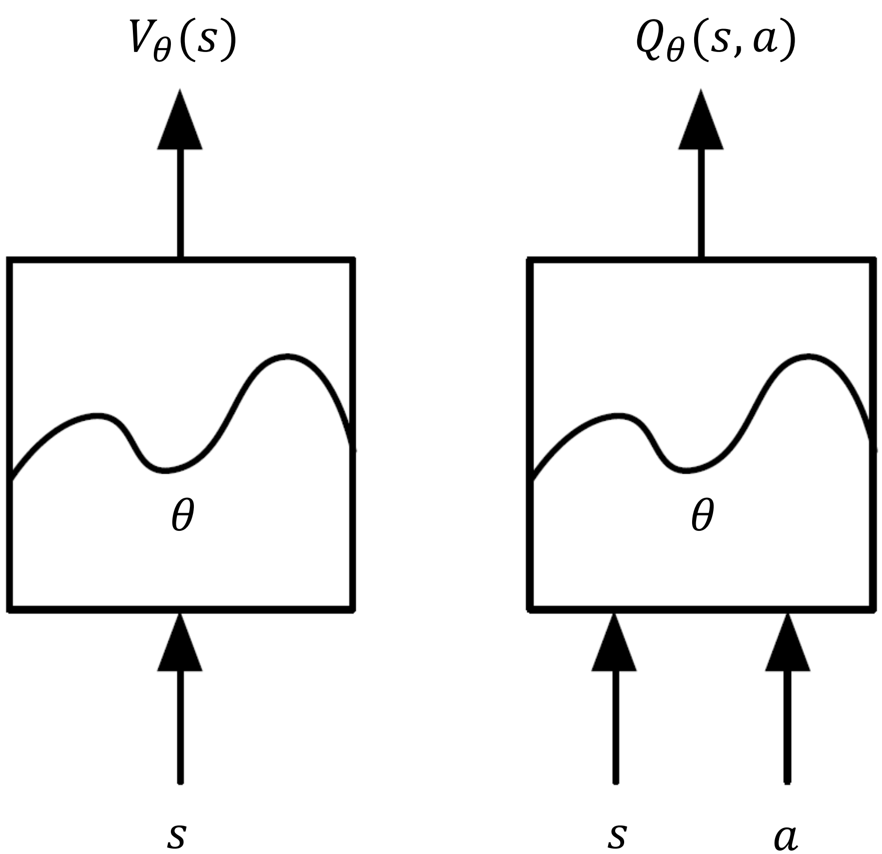
参数化值函数近似

- 构建参数化（可学习的）函数来近似值函数

$$V_\theta(s) \simeq V^\pi(s)$$
$$Q_\theta(s,a) \simeq Q^\pi(s,a)$$

  - θ是近似函数的参数，可以通过强化学习进行更新
  - 参数化的方法将现有可见的状态泛化到没有见过的状态上

值函数近似的主要形式



- 一些函数近似
  - （一般的）线性模型
  - 神经网络
  - 决策树
  - 最近邻
  - 傅立叶/小波基底
- 可微函数
  - （一般的）线性模型
  - 神经网络
- 我们希望模型适合在非稳态的、非独立同分布的数据上训练
  - 因此参数化模型比树模型更适合

基于随机梯度下降（SGD）的值函数近似

- 目标：找到参数向量θ最小化值函数近似值与真实值之间的均方误差

$$J(\theta)=\mathbb{E}_\pi[\frac{1}{2}\left(V^\pi(s)-V_\theta(s)\right)^2]$$

- 误差减小的梯度方向

$$-\frac{\partial J(\theta)}{\partial\theta}=\mathbb{E}_\pi[\left(V^\pi(s)-V_\theta(s)\right)\frac{\partial V_\theta(s)}{\partial\theta}]$$

- 单次采样进行随机梯度下降

$$\begin{aligned}\theta &\leftarrow \theta-\alpha\frac{\partial J(\theta)}{\partial\theta}\\ &=\theta+\alpha\left(V^\pi(s)-V_\theta(s)\right)\frac{\partial V_\theta(s)}{\partial\theta}\end{aligned}$$

特征化状态

- 用一个特征向量表示状态

$$x(s) = \begin{bmatrix} x_1(s) \\ \vdots \\ x_k(s) \end{bmatrix}$$

- 以直升机控制问题为例
  - 3D位置
  - 3D速度（位置的变化量）
  - 3D加速度（速度的变化量）

价值函数近似算法

目录

Contents

01

状态值函数近似

线性状态值函数近似

- 用特征的线性组合表示价值函数

$$V_\theta(s) = \theta^{\mathrm{T}} x(s)$$

- 目标函数是参数θ的二次函数

$$J(\theta) = \mathbb{E}_\pi\left[\frac{1}{2}\left(V^\pi(s) - \theta^{\mathrm{T}} x(s)\right)^2\right]$$

- 因而随机梯度下降能够收敛到全局最优解上

$$\begin{aligned}\theta &\leftarrow \theta - \alpha \frac{\partial J(\theta)}{\partial \theta} \\ &= \theta + \underbrace{\alpha}_{\text{步长}} \underbrace{\left(V^\pi(s) - V_\theta(s)\right)}_{\text{预测误差}} \underbrace{x(s)}_{\text{特征值}}\end{aligned}$$

蒙特卡洛状态值函数近似

$$\theta \leftarrow \theta + \alpha\left(V^{\pi}(s) - V_{\theta}(s)\right)x(s)$$

- 我们用$V^{\pi}(s)$表示真实的目标价值函数
- 在“训练数据”上运用监督学习对价值函数进行预测

$$\langle s_1, G_1 \rangle, \langle s_2, G_2 \rangle, \dots, \langle s_T, G_T \rangle$$

- 对于每个数据样本$\langle s_t, G_t \rangle$

$$\theta \leftarrow \theta + \alpha\left(G_t - V_{\theta}(s)\right)x(s_t)$$

- 蒙特卡洛预测至少能收敛到一个局部最优解
  - 在价值函数为线性的情况下可以收敛到全局最优

时序差分状态值函数近似

$$\theta \leftarrow \theta + \alpha\big(V^{\pi}(s) - V_{\theta}(s)\big)x(s)$$

- 时序差分算法的目标$r_{t+1} + \gamma V_{\theta}(s_{t+1})$是真实目标价值$V_{\pi}(s_t)$的有偏采样

- 在“训练数据”上运用监督学习

$$\langle s_1, r_2 + \gamma V_{\theta}(s_2)\rangle, \langle s_2, r_3 + \gamma V_{\theta}(s_3)\rangle, \ldots, \langle s_T, r_T\rangle$$

- 对于每个数据样本$\langle s_t, r_{t+1} + \gamma V_{\theta}(s_{t+1})\rangle$

$$\theta \leftarrow \theta + \alpha\big(r_{t+1} + \gamma V_{\theta}(s_{t+1}) - V_{\theta}(s)\big)x(s_t)$$

- 线性情况下时序差分学习（接近）收敛到全局最优解

02

状态-动作值函数近似

状态-动作值函数近似

- 对动作-状态值函数进行近似

$$Q_\theta(s,a) \simeq Q^\pi(s,a)$$

- 最小均方误差

$$J(\theta) = \mathbb{E}_\pi\left[\frac{1}{2}\left(Q^\pi(s,a) - Q_\theta(s,a)\right)^2\right]$$

- 在单个样本上进行随机梯度下降

$$\begin{aligned}\theta &\leftarrow \theta - \alpha \frac{\partial J(\theta)}{\partial \theta} \\ &= \theta + \alpha\big(Q^\pi(s,a) - Q_\theta(s,a)\big)\frac{\partial Q_\theta(s,a)}{\partial \theta}\end{aligned}$$

线性状态-动作值函数近似

- 用特征向量表示状态-动作对

$$x(s,a)=\begin{bmatrix} x_1(s,a) \\ \vdots \\ x_k(s,a) \end{bmatrix}$$

- 线性情况下，参数化后Q函数

$$Q_\theta(s,a)=\theta^{\mathrm{T}}x(s,a)$$

- 利用随机梯度下降更新

$$\begin{aligned}\theta &\leftarrow \theta-\alpha\frac{\partial J(\theta)}{\partial\theta} \\ &=\theta+\alpha\left(Q^\pi(s,a)-\theta^{\mathrm{T}}x(s,a)\right)x(s,a)\end{aligned}$$

时序差分状态-动作值函数近似

$$\theta \leftarrow \theta + \alpha\left(Q^{\pi}(s,a) - Q_{\theta}(s,a)\right)\frac{\partial Q_{\theta}(s,a)}{\partial \theta}$$

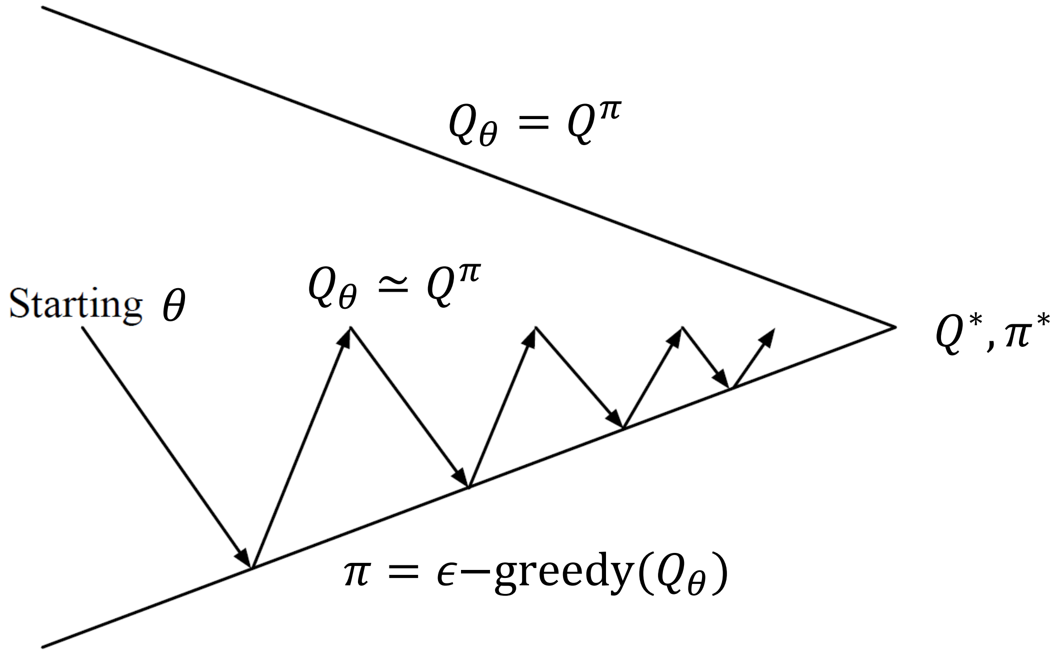
- 对于蒙特卡洛学习，目标是累计奖励G_t

$$\theta \leftarrow \theta + \alpha\left(G_t - Q_{\theta}(s,a)\right)\frac{\partial Q_{\theta}(s,a)}{\partial \theta}$$

- 对于时序差分学习，目标是$r_{t+1} + \gamma Q_{\theta}(s_{t+1}, a_{t+1})$

$$\theta \leftarrow \theta + \alpha\left(r_{t+1} + \gamma Q_{\theta}(s_{t+1}, a_{t+1}) - Q_{\theta}(s,a)\right)\frac{\partial Q_{\theta}(s,a)}{\partial \theta}$$

时序差分状态-动作值函数近似



- 策略评估：近似策略评估$Q_\theta \simeq Q^\pi$
- 策略改进：ϵ-贪心策略提升

时序差分学习参数更新过程

- 对于$TD(0)$，时序差分学习的目标是
  - 状态值函数

$$\theta \leftarrow \theta + \alpha\big(V^{\pi}(s_t) - V_{\theta}(s)\big)\frac{\partial V_{\theta}(s_t)}{\partial \theta}$$
$$= \theta + \alpha\big(r_{t+1} + \gamma V_{\theta}(s_{t+1}) - V_{\theta}(s)\big)\frac{\partial V_{\theta}(s_t)}{\partial \theta}$$

  - 动作-状态值函数

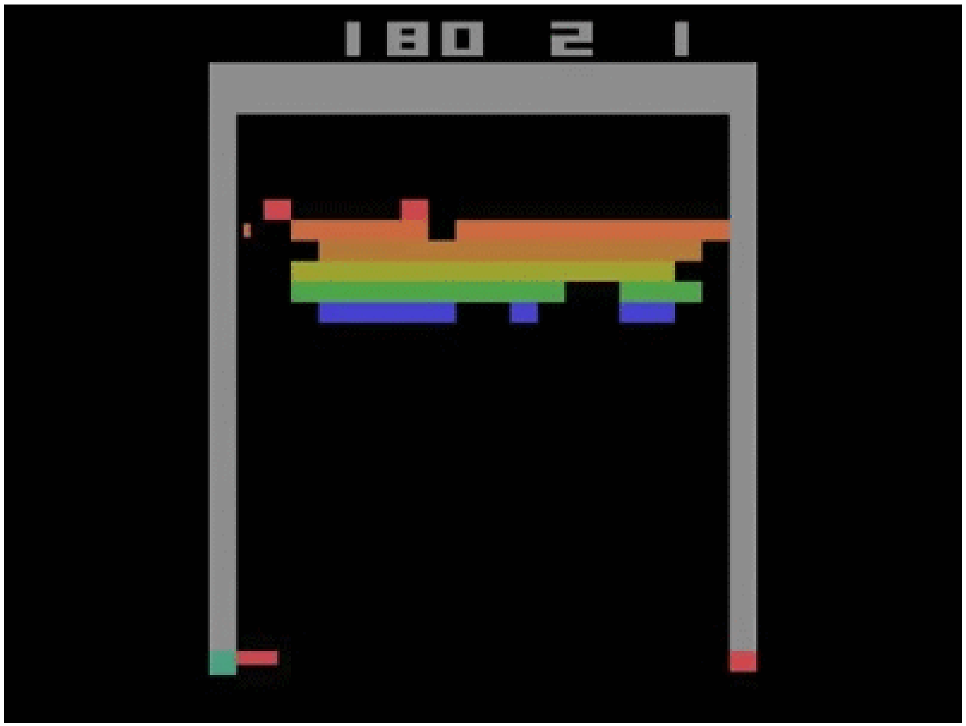
$$\theta \leftarrow \theta + \alpha\big(Q^{\pi}(s,a) - Q_{\theta}(s,a)\big)\frac{\partial Q_{\theta}(s,a)}{\partial \theta}$$
$$= \theta + \alpha\big(r_{t+1} + \gamma Q_{\theta}(s_{t+1},a_{t+1}) - Q_{\theta}(s,a)\big)\frac{\partial Q_{\theta}(s,a)}{\partial \theta}$$

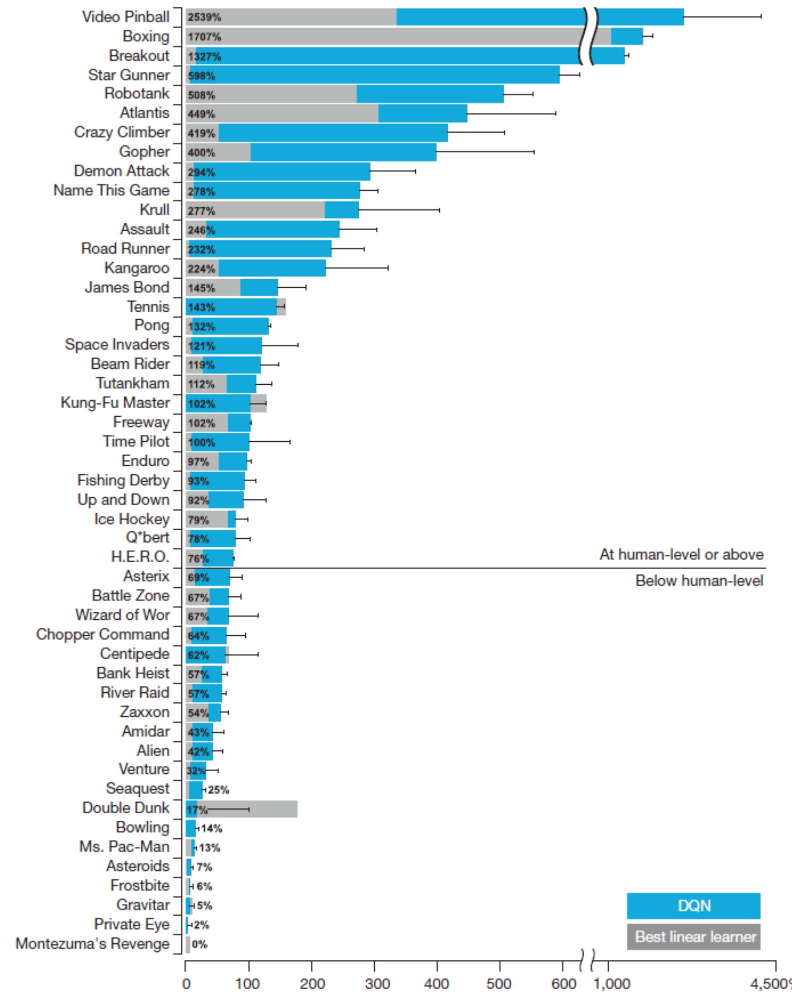
- 虽然θ在时序差分学习的目标中出现，但是我们并不需要计算目标函数的梯度。想想这是为什么？

03

案例分析

案例分析：Deep Q-Network (DQN)



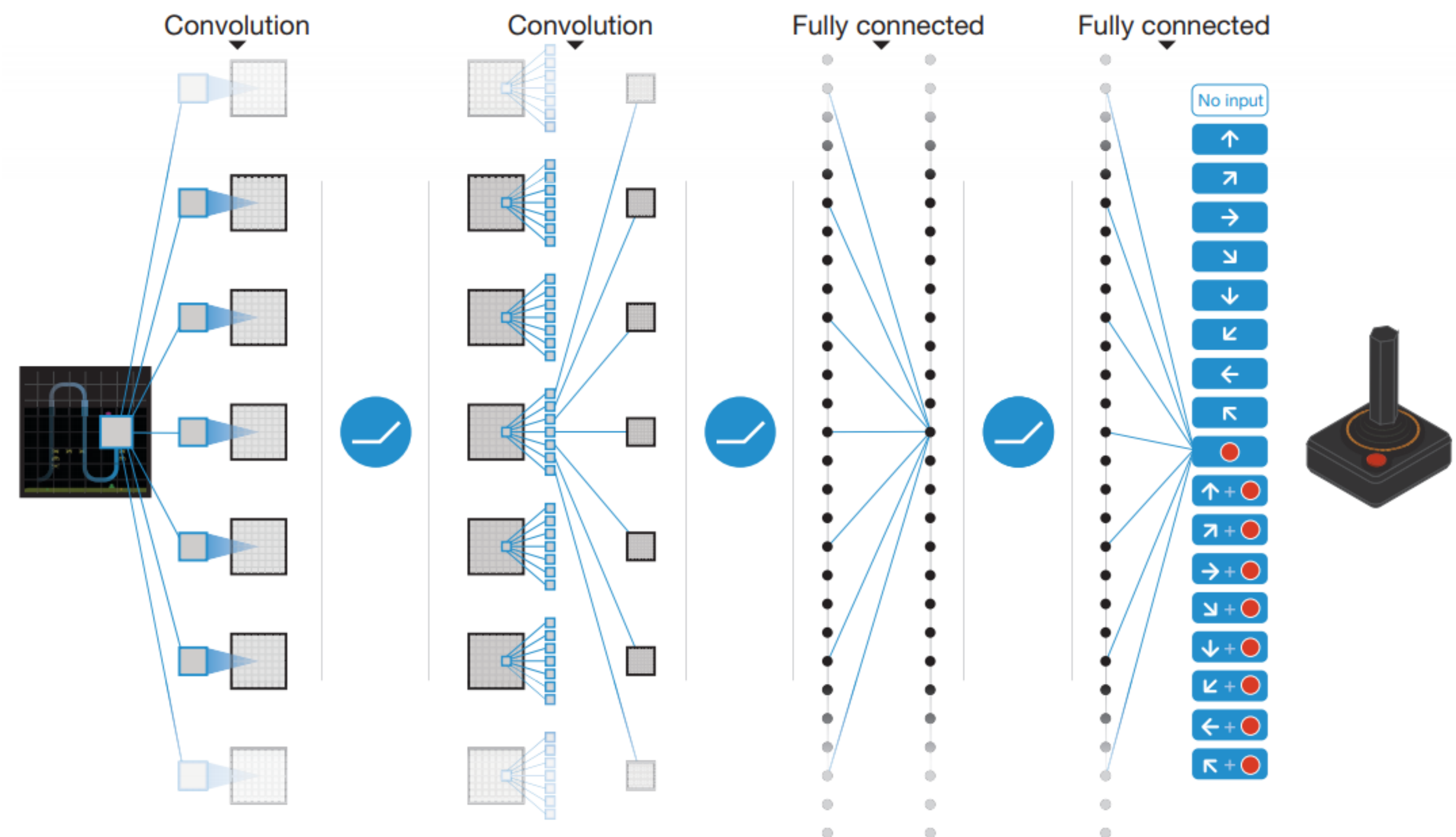


Volodymyr Mnih, Koray Kavukcuoglu, David Silver et al. Playing Atari with Deep Reinforcement Learning.NIPS 2013 workshop.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver et al. Human-level control through deep reinforcement learning. Nature 2015.

案例分析：Deep Q-Network (DQN)

- 使用深度神经网络表示Q函数



Volodymyr Mnih, Koray Kavukcuoglu, David Silver et al. Playing Atari with Deep Reinforcement Learning.NIPS 2013 workshop.
Volodymyr Mnih, Koray Kavukcuoglu, David Silver et al. Human-level control through deep reinforcement learning. Nature 2015.

案例分析：Deep Q-Network (DQN)

- 第i轮迭代中更新的Q-学习损失函数

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim U(D)}\left[\Big(\underbrace{r + \gamma \max_{a'} Q(s',a';\theta_i^-)}_{\text{目标Q值}} - \underbrace{Q(s,a;\theta_i)}_{\text{预测Q值}}\Big)^2\right]$$

- θ_i是第i轮迭代中将要更新的网络参数
  - 通过标准的反向传播算法进行更新
- θ_i^-是目标网络参数
  - 仅在θ_i每更新C步后进行更新
- $(s,a,r,s')\sim U(D)$：样本从经验池D中均匀抽样
  - 这样做可以避免在近期经验上过拟合

Volodymyr Mnih, Koray Kavukcuoglu, David Silver et al. Playing Atari with Deep Reinforcement Learning.NIPS 2013 workshop.
Volodymyr Mnih, Koray Kavukcuoglu, David Silver et al. Human-level control through deep reinforcement learning. Nature 2015.

策略梯度

参数化策略

- 我们能够将策略参数化

$$\pi_\theta(a|s)$$

策略可以是确定性的

$$a = \pi_\theta(s)$$

也可以是随机的

$$\pi_\theta(a|s) = P(a|s;\theta)$$

  - θ 是策略的参数
  - 将可见的已知状态泛化到未知的状态上
  - 在本课程中我们主要讨论的是模型无关的强化学习

基于策略的强化学习

优点

- 具有更好的收敛性质
- 在高维度或连续的动作空间中更有效
  - 最重要的因素：基于值函数的方法，通常需要取最大值
- 能够学习出随机策略

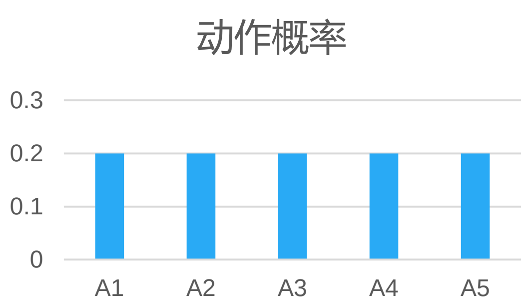
缺点

- 通常会收敛到局部最优而非全局最优
- 评估一个策略通常不够高效并具有较大的方差（variance）

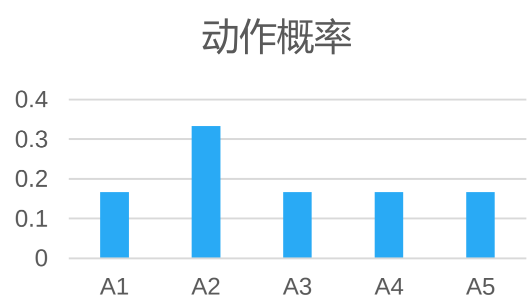
策略梯度

- 对于随机策略$\pi_\theta(a|s) = P(a|s;\theta)$
- 直觉上我们应该
  - 降低带来较低价值/奖励的动作出现的概率
  - 提高带来较高价值/奖励的动作出现的概率
- 一个离散动作空间维度为5的例子

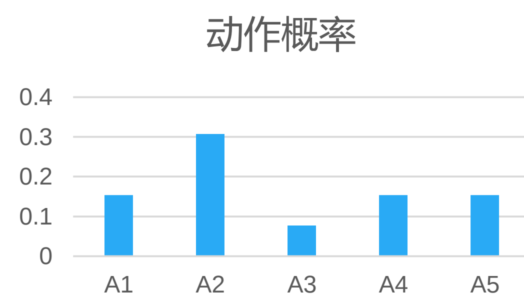
1. 初始化θ



3. 根据策略梯度更新θ



5. 根据策略梯度更新θ



2. 采取动作A2
观察到正的奖励

4. 采取动作A3
观察到负的奖励

单步马尔可夫决策过程中的策略梯度

- 考虑一个简单的单步马尔可夫决策过程
  - 起始状态为$s \sim d(s)$
  - 决策过程在进行一步决策后结束，获得奖励值为r_{sa}
- 策略的价值期望

$$J(\theta) = \mathbb{E}_{\pi_\theta}[r] = \sum_{s \in S} d(s) \sum_{a \in A} \pi_\theta(a|s) r_{sa}$$

$$\frac{\partial J(\theta)}{\partial \theta} = \sum_{s \in S} d(s) \sum_{a \in A} \frac{\partial \pi_\theta(a|s)}{\partial \theta} r_{sa}$$

似然比 (Likelihood Ratio)

- 似然比利用下列特性

$$\frac{\partial \pi_\theta(a|s)}{\partial\theta} = \pi_\theta(a|s)\frac{1}{\pi_\theta(a|s)}\frac{\partial \pi_\theta(a|s)}{\partial\theta} = \pi_\theta(a|s)\frac{\partial \log\pi_\theta(a|s)}{\partial\theta}$$

- 所以策略的价值期望可以写成

$$J(\theta) = \mathbb{E}_{\pi_\theta}[r] = \sum_{s\in S} d(s)\sum_{a\in A}\pi_\theta(a|s)r_{sa}$$

$$\frac{\partial J(\theta)}{\partial\theta} = \sum_{s\in S} d(s)\sum_{a\in A}\frac{\partial \pi_\theta(a|s)}{\partial\theta}r_{sa} = \sum_{s\in S} d(s)\sum_{a\in A}\pi_\theta(a|s)\frac{\partial \log\pi_\theta(a|s)}{\partial\theta}r_{sa} = \mathbb{E}_{\pi_\theta}\left[\frac{\partial \log\pi_\theta(a|s)}{\partial\theta}r_{sa}\right]$$

这一结果可以通过从d(s)中采样状态s和从πθ中采样动作a来近似估计

策略梯度定理

- 策略梯度定理把似然比的推导过程泛化到多步马尔可夫决策过程
  - 用长期的价值函数 $Q^{\pi_\theta}(s,a)$代替前面的瞬时奖励 r_{sa}
- 策略梯度定理涉及
  - 起始状态目标函数 J_1，平均奖励目标函数 J_{avR}，和平均价值目标函数 J_{avV}

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

$$\frac{\partial J(\theta)}{\partial \theta} = \mathbb{E}_{\pi_\theta}\left[\frac{\partial \log \pi_\theta(a|s)}{\partial \theta} Q^{\pi_\theta}(s,a)\right]$$

详细证明过程请参考:

1. Rich Sutton’s Reinforcement Learning: An Introduction (2nd Edition)第13章
2. 动手学强化学习策略梯度的附录
   https://hrl.boyuai.com/chapter/2/%E7%AD%96%E7%95%A5%E6%A2%AF%E5%BA%A6%E7%AE%97%E6%B3%95/

蒙特卡洛策略梯度（REINFORCE)

- 利用随机梯度上升更新参数
- 利用策略梯度定理
- 利用累计奖励值G_t作为$Q^{\pi_\theta}(s,a)$的无偏采样

$$\Delta\theta_t = \alpha \frac{\partial \log \pi_\theta(a_t|s_t)}{\partial\theta} G_t$$

- REINFORCE算法

initialize θ arbitrarily

for each episode $\{s_1, a_1, r_2, \ldots, s_{T-1}, a_{T-1}, r_T\} \sim \pi_\theta$ do

  for $t = 1$ to $T-1$ do

    $$\theta \leftarrow \theta + \alpha \frac{\partial}{\partial\theta} \log \pi_\theta(a_t|s_t) G_t$$

  end for

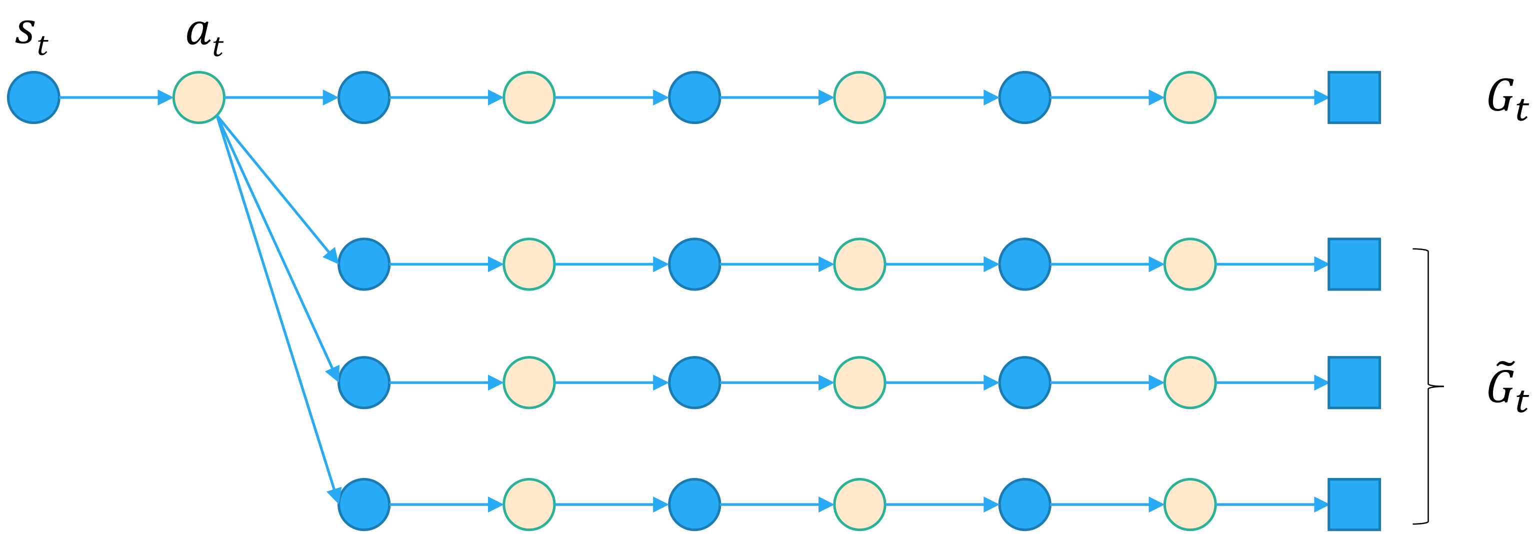
end for

return θ

蒙特卡洛策略梯度（REINFORCE）

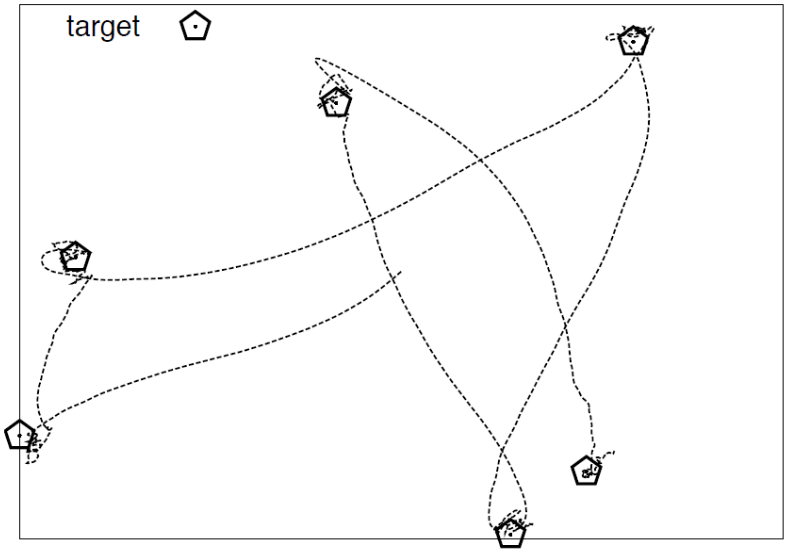
$$\Delta\theta_t = \alpha \frac{\partial \log \pi_\theta(a_t|s_t)}{\partial \theta} G_t$$

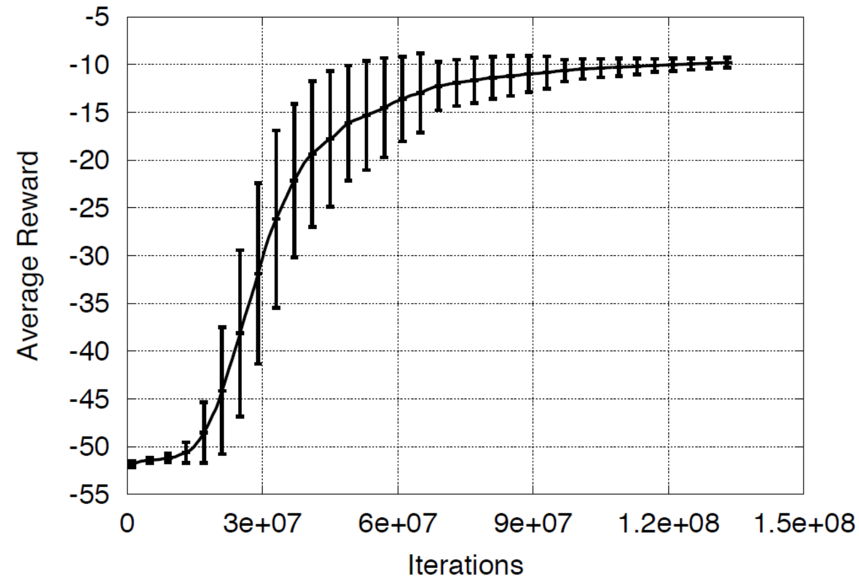
- 可通过多次roll-out的G_t平均值来逼近$Q(s_t, a_t)$



$$\tilde{G}_t = \frac{1}{N}\sum_{i=1}^{n} G_t^{(i)}$$

Puck World 冰球世界示例





- 连续的动作对冰球施加较小的力
- 冰球接近目标可以得到奖励
- 目标位置每30秒重置一次
- 使用蒙特卡洛策略梯度方法训练策略

Softmax随机策略

- Softmax策略是一种非常常用的随机策略

$$\pi_\theta(a|s) = \frac{e^{f_\theta(s,a)}}{\sum_{a'} e^{f_\theta(s,a')}}$$

  - 式中，$f_\theta(s,a)$是用θ参数化的状态-动作对得分函数，可以预先定义

- 其对数似然的梯度是

$$\begin{aligned}\frac{\partial \log \pi_\theta(a|s)}{\partial \theta} &= \frac{\partial f_\theta(s,a)}{\partial \theta} - \frac{1}{\sum_{a'} e^{f_\theta(s,a')}} \sum_{a''} e^{f_\theta(s,a'')} \frac{\partial f_\theta(s,a'')}{\partial \theta} \\ &= \frac{\partial f_\theta(s,a)}{\partial \theta} - \mathbb{E}_{a' \sim \pi_\theta(a'|s)}\left[\frac{\partial f_\theta(s,a')}{\partial \theta}\right]\end{aligned}$$

Softmax随机策略

- Softmax策略是一种非常常用的随机策略

$$\pi_\theta(a|s) = \frac{e^{f_\theta(s,a)}}{\sum_{a'} e^{f_\theta(s,a')}}$$

  - 式中，$f_\theta(s,a)$是用θ参数化的状态-动作对得分函数，可以预先定义

- 举线性得分函数为例，则有

$$f_\theta(s,a) = \theta^{\mathrm{T}} x(s,a)$$

$$\begin{aligned}\frac{\partial \log \pi_\theta(a|s)}{\partial \theta} &= \frac{\partial f_\theta(s,a)}{\partial \theta} - \mathbb{E}_{a' \sim \pi_\theta(a'|s)}\left[\frac{\partial f_\theta(s,a')}{\partial \theta}\right] \\ &= x(s,a) - \mathbb{E}_{a' \sim \pi_\theta(a'|s)}[x(s,a')]\end{aligned}$$

Actor-Critic

REINFORCE 存在的问题

- 基于片段式数据的任务
  - 通常情况下，任务需要有终止状态，REINFORCE才能直接计算累计折扣奖励
- 低数据利用效率
  - 实际中，REINFORCE需要大量的训练数据
- 高训练方差（最重要的缺陷）
  - 从单个或多个片段中采样到的值函数具有很高的方差

Actor-Critic

- Actor-Critic的思想
  - REINFORCE策略梯度方法：使用蒙特卡洛采样直接估计(s_t, a_t)的值G_t
  - 为什么不建立一个可训练的值函数Q_Φ来完成这个估计过程?
- 演员（Actor）和评论家（Critic）

演员 $\pi_\theta(a|s)$

采取动作使评论家满意的策略

评论家 $Q_\Phi(s, a)$

学会准确估计演员策略所采取动作价值的值函数

Actor-Critic训练

- 评论家Critic: $Q_\Phi(s,a)$
  - 学会准确估计当前演员策略（actor policy）的动作价值

$$Q_\Phi(s,a) \simeq r(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a), a' \sim \pi_\theta(a'|s')}[Q_\Phi(s',a')]$$

- 演员Actor: $\pi_\theta(a|s)$
  - 学会采取使critic满意的动作

$$J(\theta) = \mathbb{E}_{s \sim p, \pi_\theta}[\pi_\theta(a|s) Q_\Phi(s,a)]$$

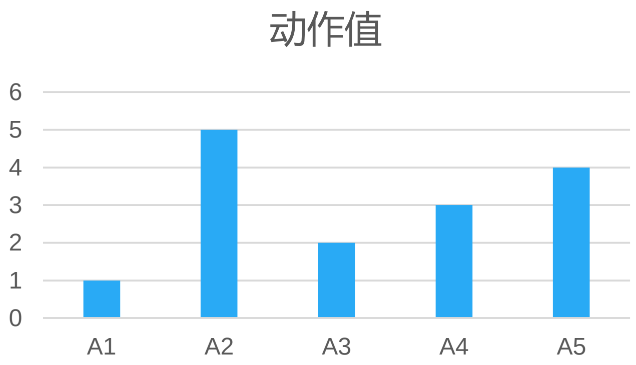
$$\frac{\partial J(\theta)}{\partial \theta} = \mathbb{E}_{\pi_\theta}\left[\frac{\partial \log \pi_\theta(a|s)}{\partial \theta} Q_\Phi(s,a)\right]$$

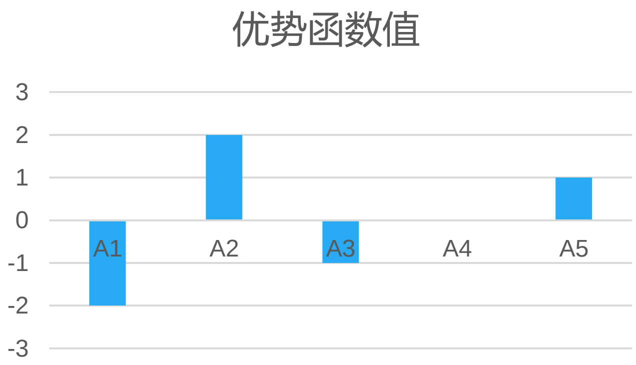
A2C： Advantageous Actor-Critic

- 思想：通过减去一个基线函数来标准化评论家的打分
  - 更多信息指导：降低较差动作概率，提高较优动作概率
  - 进一步降低方差

- 优势函数（Advantage Function）

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





A2C：Advantageous Actor-Critic

- 状态-动作值和状态值函数

$$
\begin{aligned}
Q^{\pi}(s,a) &= r(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a), a' \sim \pi_\theta(a'|s')}[Q_\Phi(s',a')] \\
&= r(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)}[V^{\pi}(s')]
\end{aligned}
$$

- 因此我们只需要拟合状态值函数来拟合优势函数

$$
\begin{aligned}
A^{\pi}(s,a) &= Q^{\pi}(s,a) - V^{\pi}(s) \\
&= r(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)}[V^{\pi}(s')] - V^{\pi}(s) \\
&\simeq r(s,a) + \gamma V^{\pi}(s') - V^{\pi}(s)
\end{aligned}
$$

采样下一个状态s'

价值和策略的近似逼近方法总结

- 价值和策略的近似逼近方法是强化学习技术从‘玩具’走向‘现实’的第一步，是深度强化学习的基础设置
- 参数化的价值函数和策略
- 通过链式法则，价值函数的参数可以被直接学习
- 通过likelihood-ratio方法，可以用advantage对策略的参数进行学习
- Actor-critic框架同时学习了价值函数和策略，通过价值函数的Q（或Advantage）估计，以策略梯度的方式更新策略参数

THANK YOU

策略梯度定理：平均奖励原则

- 平均奖励目标函数

$$J(\pi) = \lim_{n\to\infty} \frac{1}{n} \mathbb{E}[r_1 + r_2 + \cdots + r_n|\pi] = \sum_s d^\pi(s) \sum_a \pi(a|s) r(s,a)$$

$$Q^\pi(s,a) = \sum_{t=1}^{\infty} \mathbb{E}[r_t - J(\pi)|s_0 = s, a_0 = a, \pi]$$

$$\begin{aligned}
\frac{\partial V^\pi(s)}{\partial \theta} &\overset{\text{def}}{=} \frac{\partial}{\partial \theta} \sum_a \pi(a|s) Q^\pi(s,a), \quad \forall s \\
&= \sum_a \left[\frac{\partial \pi(a|s)}{\partial \theta} Q^\pi(s,a) + \pi(a|s) \frac{\partial}{\partial \theta} Q^\pi(s,a) \right] \\
&= \sum_a \left[\frac{\partial \pi(a|s)}{\partial \theta} Q^\pi(s,a) + \pi(a|s) \frac{\partial}{\partial \theta} \left(r(s,a) - J(\pi) + \sum_{s'} P_{ss'}^a V^\pi(s') \right) \right] \\
&= \sum_a \left[\frac{\partial \pi(a|s)}{\partial \theta} Q^\pi(s,a) + \pi(a|s) \left(-\frac{\partial J(\pi)}{\partial \theta} + \frac{\partial}{\partial \theta} \sum_{s'} P_{ss'}^a V^\pi(s') \right) \right]
\end{aligned}$$

$$\Rightarrow \frac{\partial J(\pi)}{\partial \theta} = \sum_a \left[\frac{\partial \pi(a|s)}{\partial \theta} Q^\pi(s,a) + \pi(a|s) \sum_{s'} P_{ss'}^a \frac{\partial V^\pi(s')}{\partial \theta} \right] - \frac{\partial V^\pi(s)}{\partial \theta}$$

推导过程请参考Rich Sutton’s Reinforcement Learning: An Introduction (2nd Edition)第13章

策略梯度定理：平均奖励原则

- 目标函数

$$\frac{\partial J(\pi)}{\partial \theta} = \sum_{a} [\frac{\partial \pi(a|s)}{\partial \theta} Q^{\pi}(s,a) + \pi(a|s) \sum_{s'} P_{ss'}^{a} \frac{\partial V^{\pi}(s')}{\partial \theta}] - \frac{\partial V^{\pi}(s)}{\partial \theta}$$

$$\sum_{s} d^{\pi}(s) \frac{\partial J(\pi)}{\partial \theta} = \sum_{s} d^{\pi}(s) \sum_{a} \frac{\partial \pi(a|s)}{\partial \theta} Q^{\pi}(s,a) + \sum_{s} d^{\pi}(s) \sum_{a} \pi(a|s) \sum_{s'} P_{ss'}^{a} \frac{\partial V^{\pi}(s')}{\partial \theta} - \sum_{s} d^{\pi}(s) \frac{\partial V^{\pi}(s)}{\partial \theta}$$

$$\sum_{s} d^{\pi}(s) \sum_{a} \pi(a|s) \sum_{s'} P_{ss'}^{a} \frac{\partial V^{\pi}(s')}{\partial \theta} = \sum_{s} \sum_{a} \sum_{s'} d^{\pi}(s)\, \pi(a|s) P_{ss'}^{a} \frac{\partial V^{\pi}(s')}{\partial \theta}$$

$$= \sum_{s} \sum_{s'} d^{\pi}(s) \left(\sum_{a} \pi(a|s) P_{ss'}^{a} \right) \frac{\partial V^{\pi}(s')}{\partial \theta} = \sum_{s} \sum_{s'} d^{\pi}(s)\, P_{ss'} \frac{\partial V^{\pi}(s')}{\partial \theta}$$

$$= \sum_{s'} \left(\sum_{s} d^{\pi}(s) P_{ss'} \right) \frac{\partial V^{\pi}(s')}{\partial \theta} = \sum_{s'} d^{\pi}(s') \frac{\partial V^{\pi}(s')}{\partial \theta}$$

$$\Rightarrow \sum_{s} d^{\pi}(s) \frac{\partial J(\pi)}{\partial \theta} = \sum_{s} d^{\pi}(s) \sum_{a} \frac{\partial \pi(a|s)}{\partial \theta} Q^{\pi}(s,a) + \sum_{s'} d^{\pi}(s') \frac{\partial V^{\pi}(s')}{\partial \theta} - \sum_{s} d^{\pi}(s) \frac{\partial V^{\pi}(s)}{\partial \theta}$$

$$\Rightarrow \frac{\partial J(\pi)}{\partial \theta} = \sum_{s} d^{\pi}(s) \sum_{a} \frac{\partial \pi(a|s)}{\partial \theta} Q^{\pi}(s,a)$$

推导过程请参考Rich Sutton’s Reinforcement Learning: An Introduction (2nd Edition)第13章

策略梯度定理：起始价值原则

- 起始状态价值目标

$$J(\pi) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \middle| s_0, \pi\right]$$

$$Q^{\pi}(s,a) = \mathbb{E}\left[\sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k} \middle| s_t = s, a_t = a, \pi\right]$$

$$\begin{aligned}
\frac{\partial V^{\pi}(s)}{\partial \theta} &= \frac{\partial}{\partial \theta} \sum_a \pi(s,a) Q^{\pi}(s,a), \quad \forall s \\
&= \sum_a \left[\frac{\partial \pi(s,a)}{\partial \theta} Q^{\pi}(s,a) + \pi(s,a) \frac{\partial}{\partial \theta} Q^{\pi}(s,a)\right] \\
&= \sum_a \left[\frac{\partial \pi(s,a)}{\partial \theta} Q^{\pi}(s,a) + \pi(s,a) \frac{\partial}{\partial \theta} \left(r(s,a) + \sum_{s'} \gamma P_{ss'}^a V^{\pi}(s')\right)\right] \\
&= \sum_a \frac{\partial \pi(s,a)}{\partial \theta} Q^{\pi}(s,a) + \sum_a \pi(s,a) \gamma \sum_{s'} P_{ss'}^a \frac{\partial V^{\pi}(s')}{\partial \theta}
\end{aligned}$$

推导过程请参考Rich Sutton’s Reinforcement Learning: An Introduction (2nd Edition)第13章

策略梯度定理：起始价值原则

- 起始状态价值目标

$$\frac{\partial V^{\pi}(s)}{\partial\theta}=\sum_{a}\frac{\partial\pi(s,a)}{\partial\theta}Q^{\pi}(s,a)+\sum_{a}\pi(s,a)\gamma\sum_{s_1}P_{ss_1}^{a}\frac{\partial V^{\pi}(s_1)}{\partial\theta}$$

$$\sum_{a}\frac{\partial\pi(s,a)}{\partial\theta}Q^{\pi}(s,a)=\gamma^{0}\,\Pr(s\rightarrow s,0,\pi)\sum_{a}\frac{\partial\pi(s,a)}{\partial\theta}Q^{\pi}(s,a)$$

$$\sum_{a}\pi(s,a)\gamma\sum_{s_1}P_{ss_1}^{a}\frac{\partial V^{\pi}(s_1)}{\partial\theta}=\sum_{s_1}\sum_{a}\pi(s,a)\gamma\,P_{ss_1}^{a}\frac{\partial V^{\pi}(s_1)}{\partial\theta}$$

$$=\sum_{s_1}\gamma P_{ss_1}\frac{\partial V^{\pi}(s_1)}{\partial\theta}=\gamma^{1}\sum_{s_1}\Pr(s\rightarrow s_1,1,\pi)\frac{\partial V^{\pi}(s_1)}{\partial\theta}$$

$$\frac{\partial V^{\pi}(s_1)}{\partial\theta}=\sum_{a}\frac{\partial\pi(s_1,a)}{\partial\theta}Q^{\pi}(s_1,a)+\gamma^{1}\sum_{s_2}\Pr(s_1\rightarrow s_2,1,\pi)\frac{\partial V^{\pi}(s_2)}{\partial\theta}$$

推导过程请参考Rich Sutton’s Reinforcement Learning: An Introduction (2nd Edition)第13章

策略梯度定理：起始价值原则

- 起始状态价值目标

$$\frac{\partial V^{\pi}(s)}{\partial \theta} = \gamma^0 \Pr(s \to s, 0, \pi) \sum_a \frac{\partial \pi(s,a)}{\partial \theta} Q^{\pi}(s,a) + \gamma^1 \sum_{s_1} \Pr(s \to s_1, 1, \pi) \sum_a \frac{\partial \pi(s_1,a)}{\partial \theta} Q^{\pi}(s_1,a)$$

$$+\gamma^2 \sum_{s_1} \Pr(s \to s_1, 1, \pi) \sum_{s_2} \Pr(s_1 \to s_2, 1, \pi) \frac{\partial V^{\pi}(s_2)}{\partial \theta}$$

$$= \gamma^0 \Pr(s \to s, 0, \pi) \sum_a \frac{\partial \pi(s,a)}{\partial \theta} Q^{\pi}(s,a) + \gamma^1 \sum_{s_1} \Pr(s \to s_1, 1, \pi) \sum_a \frac{\partial \pi(s_1,a)}{\partial \theta} Q^{\pi}(s_1,a)$$

$$+\gamma^2 \sum_{s_2} \Pr(s \to s_2, 2, \pi) \frac{\partial V^{\pi}(s_2)}{\partial \theta}$$

$$= \sum_{k=0}^{\infty} \sum_x \gamma^k \Pr(s \to x, k, \pi) \sum_a \frac{\partial \pi(x,a)}{\partial \theta} Q^{\pi}(x,a) = \sum_x \sum_{k=0}^{\infty} \gamma^k \Pr(s \to x, k, \pi) \sum_a \frac{\partial \pi(x,a)}{\partial \theta} Q^{\pi}(x,a)$$

$$\Rightarrow \frac{\partial J(\pi)}{\partial \theta} = \frac{\partial V^{\pi}(s_0)}{\partial \theta} = \sum_s \sum_{k=0}^{\infty} \gamma^k \Pr(s_0 \to s, k, \pi) \sum_a \frac{\partial \pi(s,a)}{\partial \theta} Q^{\pi}(s,a) = \sum_s d^{\pi}(s) \sum_a \frac{\partial \pi(s,a)}{\partial \theta} Q^{\pi}(s,a)$$

推导过程请参考Rich Sutton’s Reinforcement Learning: An Introduction (2nd Edition)第13章