强化学习

吴飞

浙江大学计算机学院

提 纲

1、 强化学习定义：马尔科夫决策过程

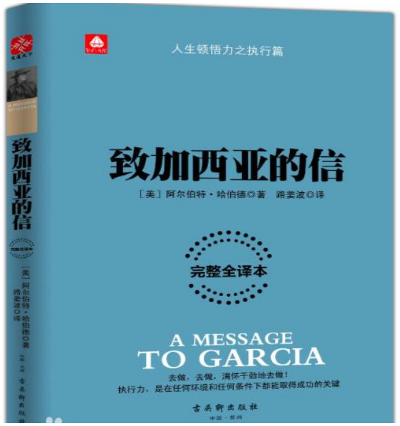
2、 强化学习中策略优化与策略评估

3、 强化学习求解：Q-Learning

4、 深度强化学习：深度学习+强化学习

《西游记》、《致加西亚的信》与强化学习

- 人生没有预先写好的剧本，与生活这一环境交互和成长
- 序列学习：取经路上花费十四年、给加西亚送信开销九天



唐太宗送别唐三藏于长安城：

宁恋本乡一捻土，莫爱他乡万两金

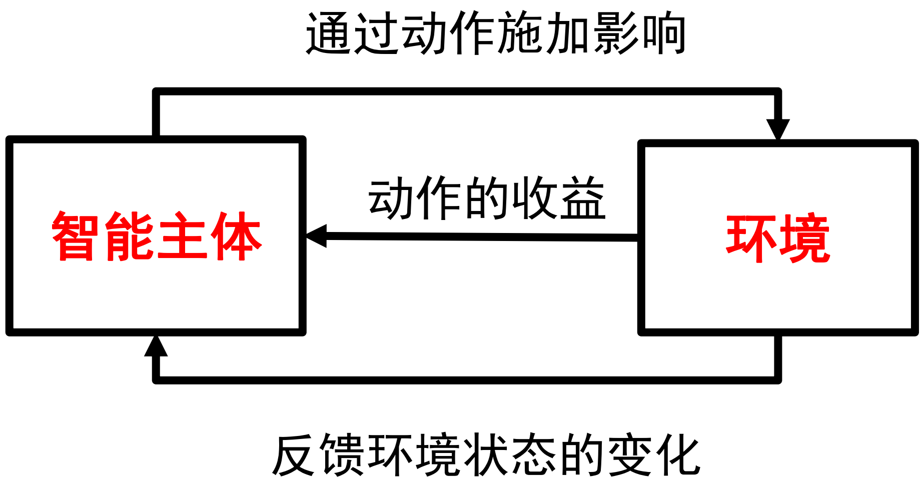
如果有一个人能够寻找到加西亚将军的话，他就是罗文

强化学习：在与环境交互之中进行学习

生活中常见的学习过程

- 人通过动作对环境产生影响 （向前走一步）
- 环境向人反馈状态的变化 （撞到了树上）
- 人估计动作得到的收益 （疼痛）
- 更新做出动作的策略 （下次避免向有树这一障碍的方向前进）

强化学习模仿了这个过程，在智能主体与环境的交互中，学习能最大化收益的行动模式



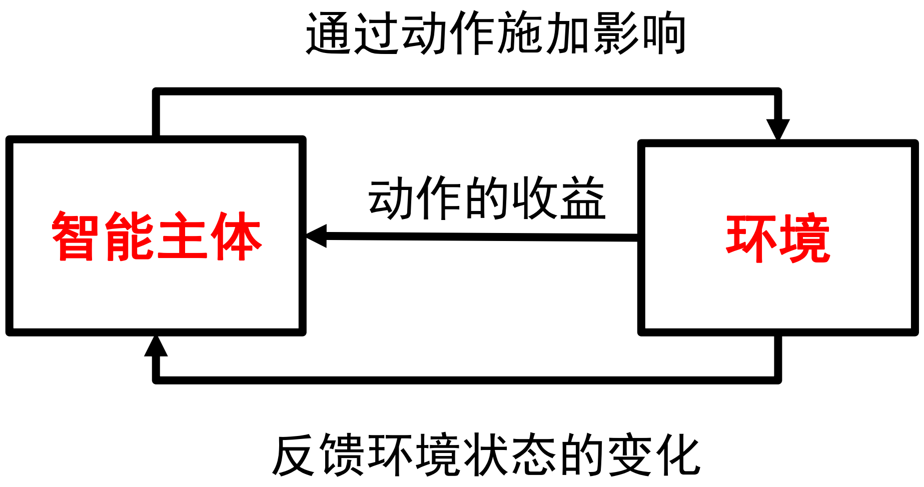
强化学习中的概念

智能主体（agent）

- 按照某种策略（policy），根据当前的状态（state）选择合适的动作（action）
- 状态指的是智能主体对环境的一种解释
- 动作反映了智能主体对环境主观能动的影响，动作带来的收益称为奖励（reward）
- 智能主体可能知道也可能不知道环境变化的规律

环境（environment）

- 系统中智能主体以外的部分
- 向智能主体反馈状态和奖励
- 按照一定的规律发生变化



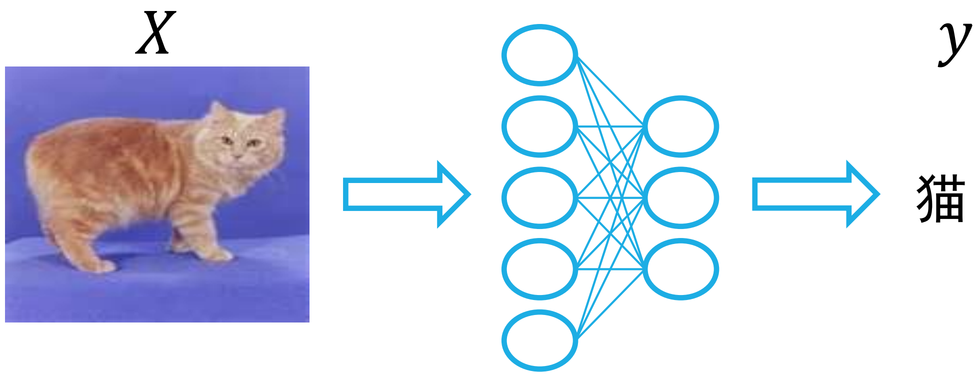
机器学习的不同类型

有监督学习

从数据X和标签y中学习映射$f: X \mapsto y$

X：图像、文本、音频、视频等

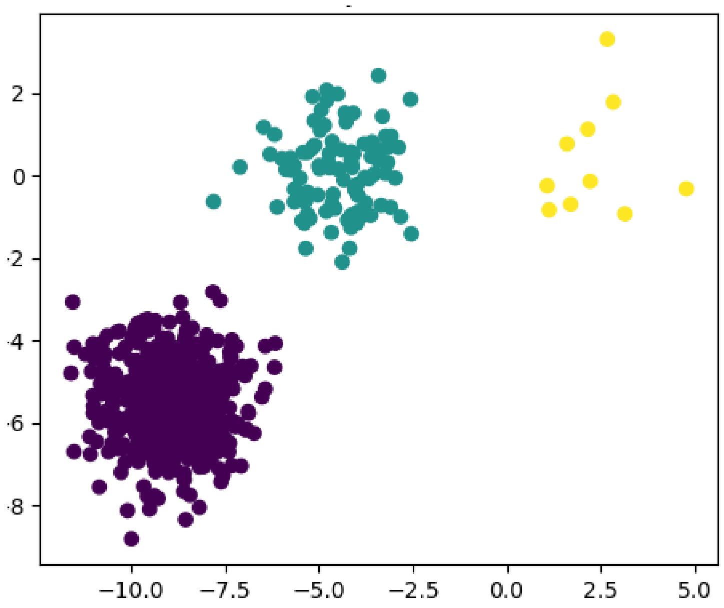
y：连续或离散的标量、结构数据



分类问题

无监督学习

寻找数据X中存在的结构和模式



聚类问题

强化学习的特点

	有监督学习	无监督学习	强化学习
学习依据	基于监督信息	基于对数据结构的假设	基于评价（evaluative）
数据来源	一次性给定	一次性给定	在交互中产生（interactive）
决策过程	单步（one-shot）	无	序列（sequential）
学习目标	样本到语义标签的映射	同一类数据的分布模式	选择能够获取最大收益的的状态到动作的映射

强化学习的特点

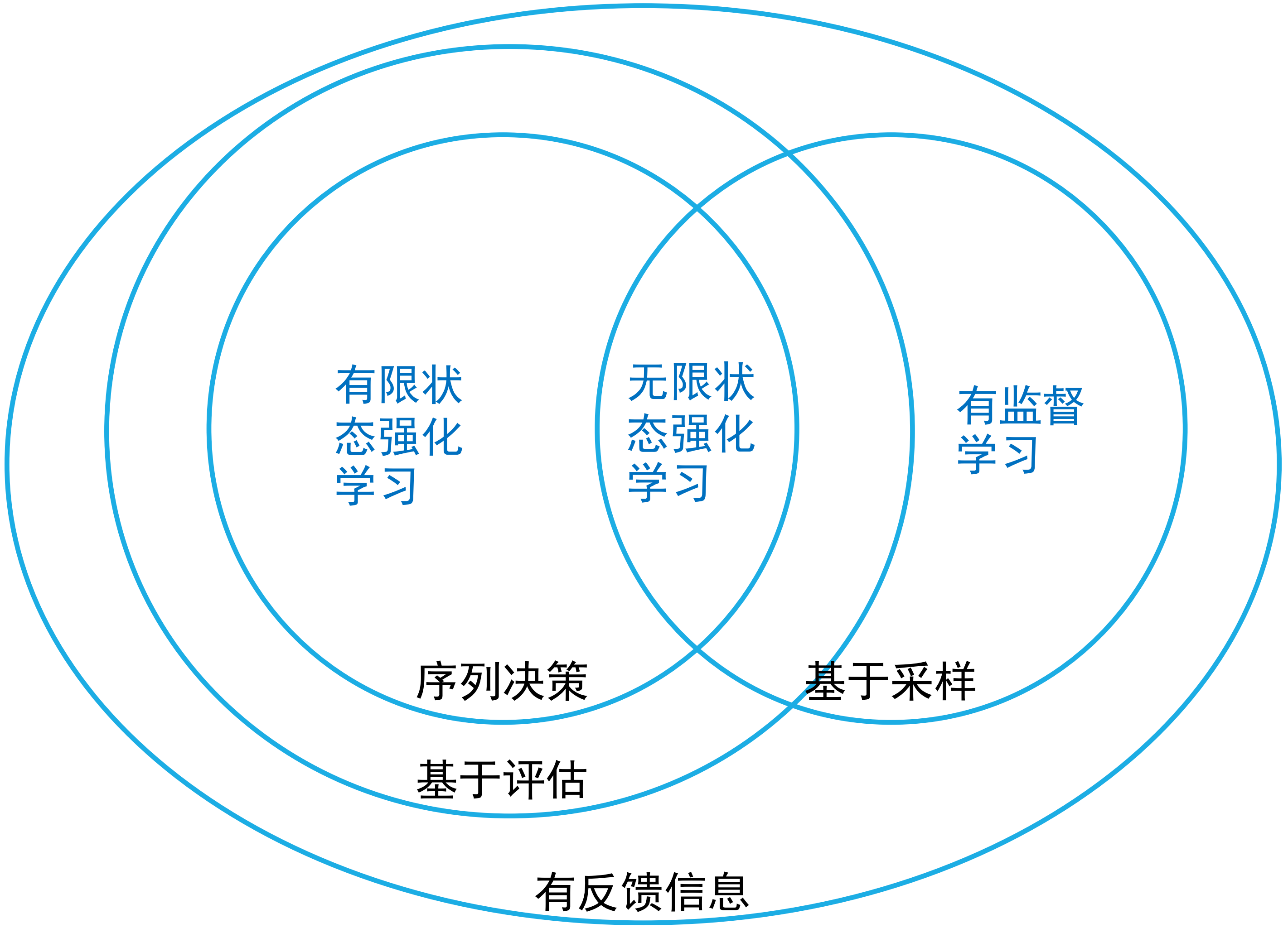
- **基于评估：**强化学习利用环境评估当前策略，以此为依据进行优化
- **交互性：**强化学习的数据在与环境的交互中产生
- **序列决策过程：**智能主体在与环境的交互中需要作出一系列的决策，这些决策往往是前后关联的

注：现实中常见的强化学习问题往往还具有奖励滞后，基于采样的评估等特点

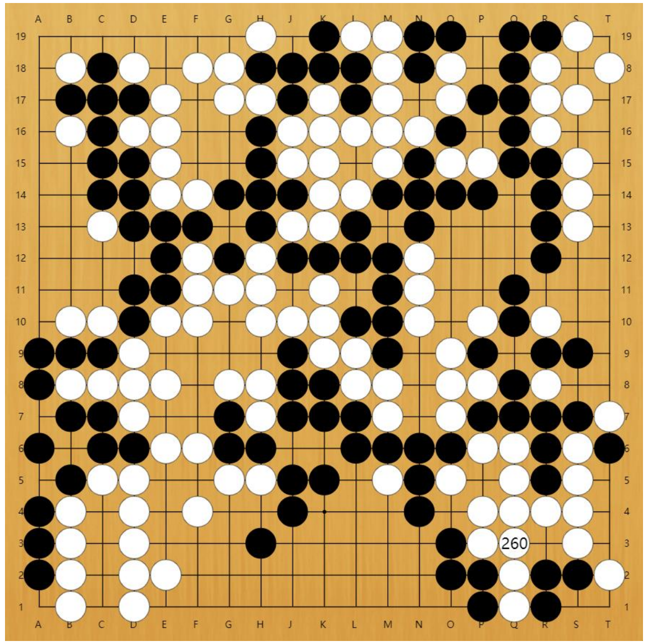
强化学习的特点

根据以下特点直观定位强化学习

- **有**/无可靠的反馈信息
- **基于评估**/基于监督信息
- **序列决策**/单步决策
- **基于采样**/基于穷举



强化学习的应用

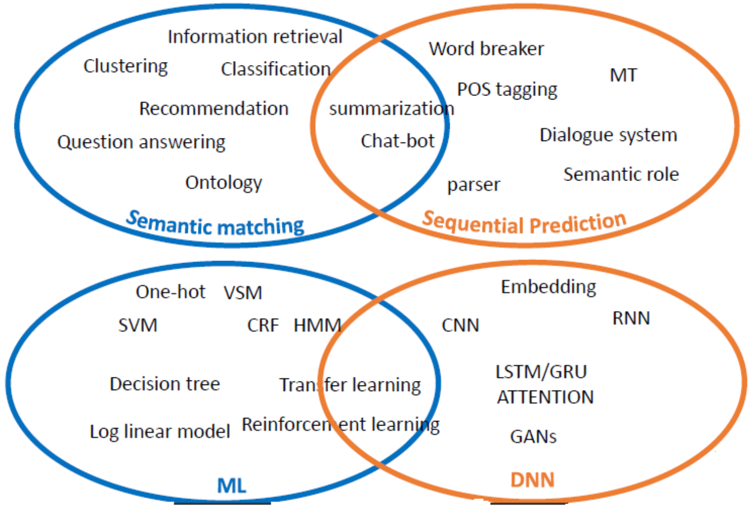


围棋游戏

注：AlphaGo的三大法宝：

- 深度学习(感知棋面)
- 强化学习（自我博弈）
- 蒙特卡洛树搜索（采样学习）

机器人运动：learning to learn

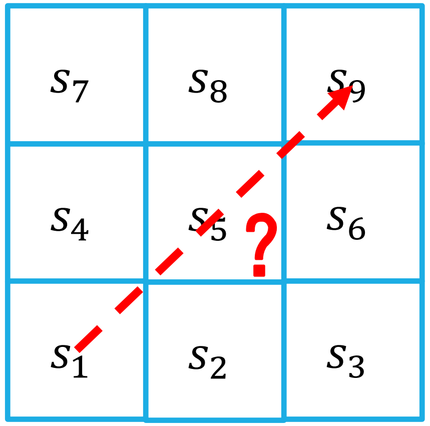


自然语言理解

强化学习示例

（序列优化）问题：

- 在下图网格中，假设有一个机器人位于s_1，其每一步只能向上或向右移动一格，跃出方格会被惩罚（且游戏停止）
- 如何使用强化学习找到一种策略，使机器人从s_1到达s_9？



刻画解该问题的因素

智能主体	迷宫机器人
环境	3×3方格
状态	机器人当前时刻所处方格
动作	每次移动一个方格
奖励	到达s_9时给予奖励；越界时给予惩罚

离散马尔可夫过程（Discrete Markov Process）

- 一个随机过程实际上是一列随时间变化的随机变量，其中当时间是离散量时，一个随机过程可以表示为$\{X_t\}_{t=0,1,2,\cdots}$，其中每个X_t都是一个随机变量，这被称为离散随机过程
- 马尔可夫链（Markov Chain）：满足马尔可夫性（Markov Property）的离散随机过程，也被称为离散马尔科夫过程。

$$Pr(X_{t+1}=x_{t+1}|X_0=x_0,X_1=x_1,\cdots,X_t=x_t)=Pr(X_{t+1}=x_{t+1}|X_t=x_t)$$

$$Pr(X_{t+1}=x_{t+1}|X_t=x_t,X_{t-1}=x_{t-1})$$

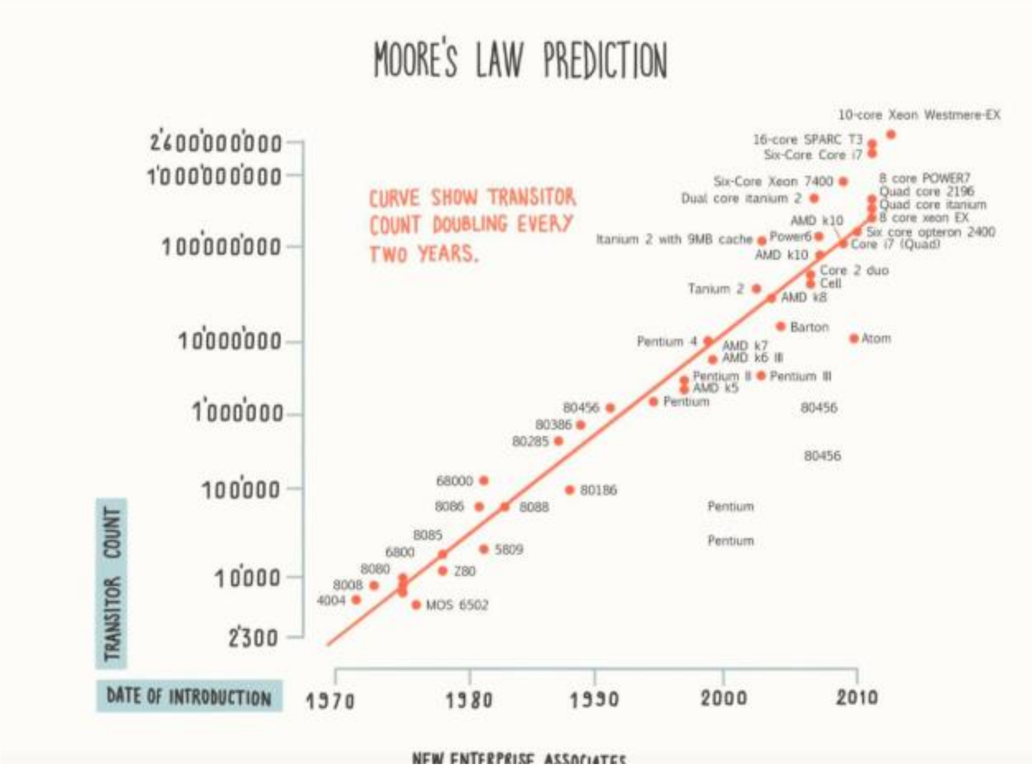
$t+1$时刻状态仅与t时刻状态相关

二阶马尔科夫链

$t+1$时刻状态与t和$t-1$时刻状态相关

离散马尔可夫过程（Discrete Markov Process）

生活中的马尔科夫链



摩尔定律
集成电路元器件数目(今天|1年半前)

上级法院 穿给 下级法院

传给

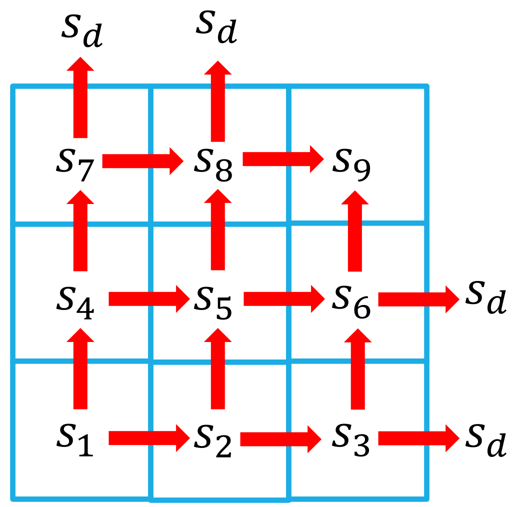
自然语言理解中n元文法(n为2表示前后相邻单词相互依赖)
(n-gram grammar)

离散马尔可夫过程：机器人移动问题

$MP = \{S, Pr\}$可用来刻画该问题

- 随机变量序列$\{S_t\}_{t=0,1,2,\cdots}$，其中S_t表示机器人第t步的位置，每个随机变量S_t的取值范围为$S = \{s_1, s_2, \cdots, s_9, s_d\}$
- 状态转移概率$Pr(S_{t+1}|S_t)$满足马尔可夫性。它的一种取值如图中箭头所示，每个箭头对应0.5的转移概率

这个模型不能体现机器人能动性，缺乏与环境进行进行交互的手段（如目标优化方式等）！！！



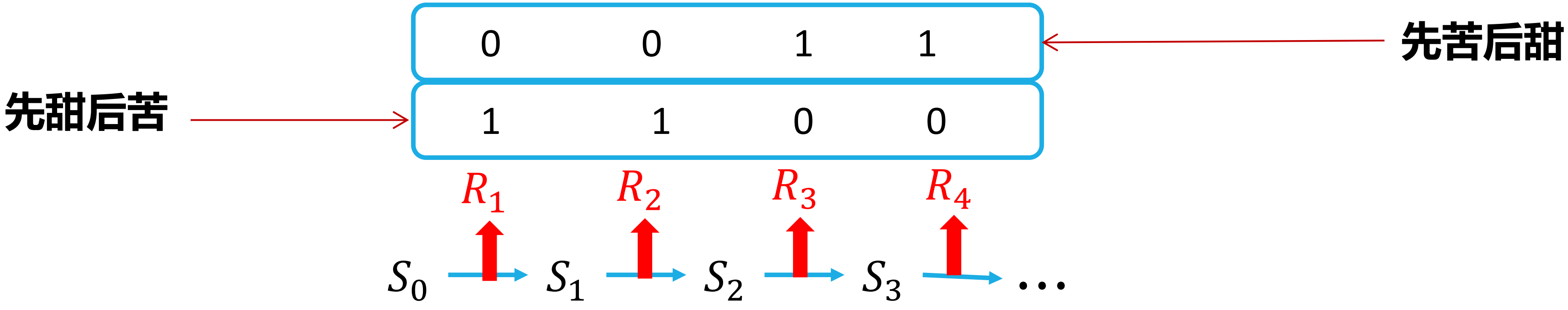
S集合（即状态空间）可为无限的，如用经纬度坐标表示现实中机器人的位置。

马尔可夫奖励过程（Markov Reward Process）：引入奖励

为了在序列决策中对目标进行优化，在马尔可夫随机过程框架中加入了奖励机制：

- 奖励函数$R: S \times S \mapsto \mathbb{R}$，其中$R(S_t, S_{t+1})$描述了从第$t$步状态转移到第 $t+1$步状态所获得奖励
- 在一个序列决策过程中，不同状态之间的转移产生了一系列的奖励$(R_1, R_2, \cdots)$，其中R_{t+1}为$R(S_t, S_{t+1})$的简便记法。
- 引入奖励机制，这样可以衡量任意序列的优劣，即对序列决策进行评价。

问题：给定两个因为状态转移而产生的奖励序列(1,1,0,0)和(0,0,1,1)，哪个序列决策更好？



马尔可夫奖励过程（Markov Reward Process）

问题：给定两个因为状态转移而产生的奖励序列(1,1,0,0)和(0,0,1,1)，哪个奖励序列更好？

为了比较不同的奖励序列，**定义反馈（return），用来反映累加奖励：**

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

其中折扣系数（discount factor）$\gamma \in [0, 1]$

反馈值反映了某个时刻后所得到累加奖励，当衰退系数小于1时，越是遥远的未来对累加反馈的贡献越少

假设 $\gamma = 0.99$

(1,1,0,0)：$G_0 = 1 + 0.99 \times 1 + 0.99^2 \times 0 + 0.99^3 \times 0 = 1.99$

(0,0,1,1)：$G_0 = 0 + 0.99 \times 0 + 0.99^2 \times 1 + 0.99^3 \times 1 = 1.9504$

先苦后甜	0	0	1	1
先甜后苦	1	1	0	0
	R_1	R_2	R_3	R_4

$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow \ldots$

马尔可夫奖励过程（Markov Reward Process）

使用离散马尔可夫过程描述机器人移动问题

- 随机变量序列$\{S_t\}_{t=0,1,2,\cdots}$：S_t表示机器人第t步的位置，每个随机变量S_t的取值范围为$S = \{s_1, s_2, \cdots, s_9, s_d\}$
- 状态转移概率：$Pr(S_{t+1}|S_t)$满足马尔可夫性

⇦ 马尔可夫过程

- 定义奖励函数$R(S_t, S_{t+1})$：从S_t到S_{t+1}所获得奖励，其取值如图中所示
- 定义衰退系数：$\gamma \in [0, 1]$

	-1		
0	0	1	
0	0	0	-1
0	0	0	

综合以上信息，可用$MRP = \{S, Pr, R, \gamma\}$来刻画马尔科夫奖励过程

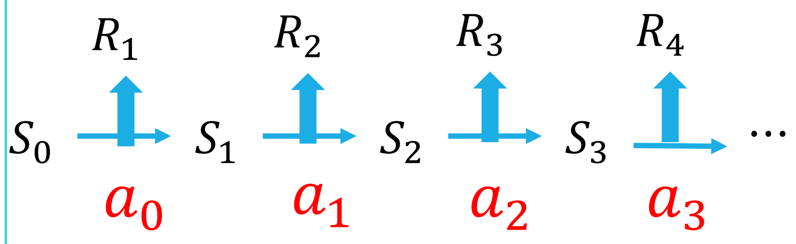
这个模型不能体现机器人能动性，仍然缺乏与环境进行进行交互的手段

马尔可夫决策过程（Markov Decision Process）：引入动作

在强化学习问题中，智能主体与环境交互过程中可自主决定所采取的动作，不同**动作**会对环境产生不同影响，为此：

- 定义智能主体能够采取的动作集合为A
- 由于不同的动作对环境造成的影响不同，因此状态转移概率定义为$Pr(S_{t+1}|S_t, a_t)$，其中$a_t \in A$为第t步采取的动作
- 奖励可能受动作的影响，因此修改奖励函数为$R(S_t, a_t, S_{t+1})$

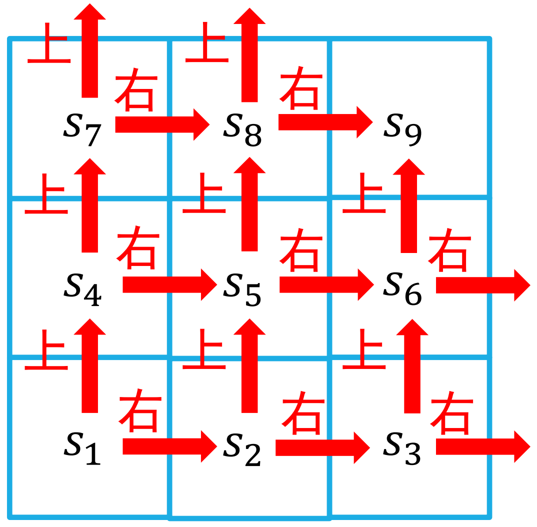
> - 动作集合A可以是有限的，也可以是无限的
> - 状态转移可是确定（deterministic）的，也可以是随机概率性（stochastic）的。
> - 确定状态转移相当于发生从S_t到S_{t+1}的转移概率为1



马尔可夫决策过程（Markov Decision Process）

使用离散马尔可夫过程描述机器人移动问题

- 随机变量序列$\{S_t\}_{t=0,1,2,\cdots}$：S_t表示机器人第t步所在位置（即状态），每个随机变量S_t的取值范围为$S=\{s_1,s_2,\cdots,s_9,s_d\}$
- 动作集合：$A=\{$上，右$\}$
- 状态转移概率$Pr(S_{t+1}|S_t,a_t)$：满足马尔可夫性，其中$a_t\in A$。状态转移如图所示。
- 奖励函数：$R(S_t,a_t,S_{t+1})$
- 衰退系数：$\gamma\in[0,1]$



综合以上信息，可通过$MDP=\{S,A,Pr,R,\gamma\}$来刻画马尔科夫决策过程

马尔可夫决策过程（Markov Decision Process）

- 马尔可夫决策过程$MDP = \{S, A, Pr, R, \gamma\}$是刻画强化学习中环境的标准形式
- 马尔可夫决策过程可用如下序列来表示：

$$S_0 \xrightarrow[a_0]{R_1} S_1 \xrightarrow[a_1]{R_2} S_2 \xrightarrow[a_2]{R_3} S_3 \xrightarrow[a_3]{R_4} \cdots$$

马尔科夫过程中产生的状态序列称为轨迹(trajectory)，可如下表示

$$(S_0, a_0, R_1, S_1, a_1, R_2, \cdots, S_T)$$

- 轨迹长度可以是无限的，也可以有终止状态S_T。有终止状态的问题叫做分段的（即存在回合的）（episodic），否则叫做持续的（continuing）

分段问题中，一个从初始状态到终止状态的完整轨迹称为一个片段或回合（episode）。如围棋对弈中一个胜败对局为一个回合。

马尔可夫决策过程（Markov Decision Process）

在机器人移动问题中：状态、行为、衰退系数、起始/终止状态、反馈、状态转移概率矩阵的定义如下

$S = \{s_1, s_2, \cdots, s_9, s_d\}$

$A = \{上，右\}$

$\gamma = 0.99$

起始状态：$S_0 = s_1$

终止状态:$S_T \in \{s_9, s_d\}$

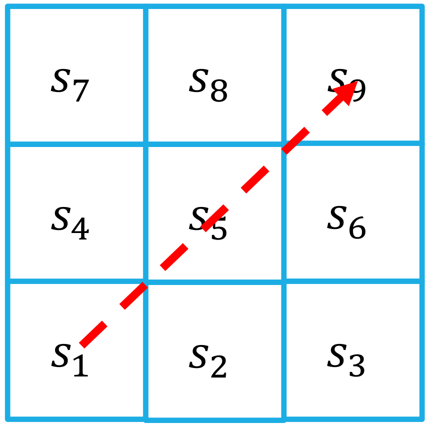
$$R(S_t, a_t, S_{t+1}) = \begin{cases} 1\ ,如果 S_{t+1} = s_9 \\ -1, 如果 S_{t+1} = s_d \\ 0\ ,其他情况 \end{cases}$$

$Pr(S_{t+1}|S_t, a_t = 右)$

S_t \ S_{t+1}	s_1	s_2	s_3		s_9	s_d
s_1	0	1	0	…	0	0
s_2	0	0	1	…	0	0
s_3	0	0	0	…	0	1
	…			…		…
s_8	0	0	0	…	1	0
s_9	0	0	0	…	0	1

$Pr(S_{t+1}|S_t, a_t = 上)$

S_t \ S_{t+1}	s_1	s_4	s_7		s_9	s_d
s_1	0	1	0	…	0	0
s_4	0	0	1	…	0	0
s_7	0	0	0	…	0	1
	…			…		…
s_6	0	0	0	…	1	0
s_9	0	0	0	…	0	1



如何从起始状态到终止状态？

马尔可夫决策过程（Markov Decision Process）中的策略学习

马尔可夫决策过程$MDP=\{S,A,Pr,R,\gamma\}$对环境进行了描述，那么**智能主体如何与环境交互而完成任务？需要进行策略学习**

对环境中各种因素的说明	
已知的：S,A,R,γ	不一定已知的：Pr
观察到的：$(S_0,a_0,R_1,S_1,a_1,R_2,\cdots,S_T)$	

策略函数：

- 策略函数$\pi:S\times A\mapsto[0,1]$，其中$\pi(s,a)$的值表示在状态$s$下采取动作$a$的概率。
- 策略函数的输出可以是确定的，即给定s情况下，只有一个动作a使得概率$\pi(s,a)$取值为1。对于确定的策略，记为$a=\pi(s)$。

马尔可夫决策过程（Markov Decision Process）中的策略学习

由马尔可夫性，未来的状态和奖励只与当前状态相关，与t无关。因此t取任意值该等式均成立，如“逢山开路,遇水搭桥”。

如何进行策略学习：一个好的策略是在当前状态下采取了一个行动后，该行动能够在未来收到最大化的反馈：

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

为了对策略函数π进行评估，定义

- **价值函数（Value Function）** $V: S \mapsto \mathbb{R}$，其中$V_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s]$，即在第$t$步状态为$s$时，按照策略$\pi$行动后在未来所获得反馈值的期望
- **动作-价值函数(Action-Value Function)** $q: S \times A \mapsto \mathbb{R}$，其中$q_\pi(s,a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a]$ 表示在第t步状态为s时，按照策略π采取动作a后，在未来所获得反馈值的期望

这样，策略学习转换为如下优化问题：

寻找一个最优策略π^*，对任意$s \in S$使得$V_{\pi^*}(s)$值最大

价值函数与动作-价值函数的关系：对策略进行评估

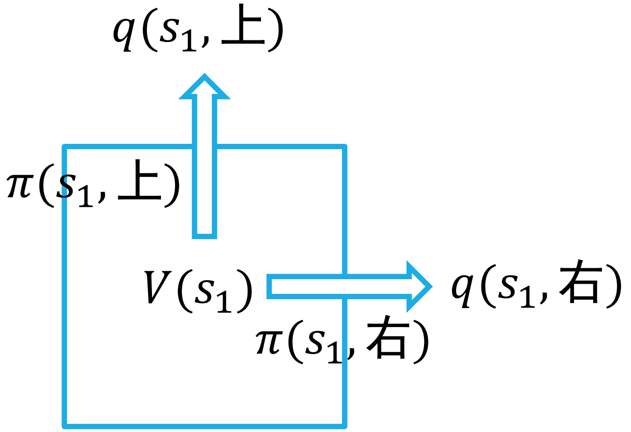
$$
\begin{aligned}
V_\pi(s) &= \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots | S_t = s] \\
&= \mathbb{E}_{a\sim\pi(s,\cdot)}[\mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots | S_t = s, A_t = a]] \\
&= \sum_{a\in A} \pi(s,a) q_\pi(s,a)
\end{aligned}
$$

$$
\begin{aligned}
q_\pi(s,a) &= \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots | S_t = s, A_t = a] \\
&= \mathbb{E}_{s'\sim Pr(\cdot|s,a)}\left[R(s,a,s') + \gamma \mathbb{E}_\pi[R_{t+2} + \gamma R_{t+3} + \cdots | S_{t+1} = s']\right] \\
&= \sum_{s'\in S} Pr(s'|s,a)\,[R(s,a,s') + \gamma V_\pi(s')]
\end{aligned}
$$

价值函数与动作-价值函数的关系：以状态s_1的计算为例

$$V_\pi(s) = \sum_{a \in A} \pi(s,a) q_\pi(s,a)$$

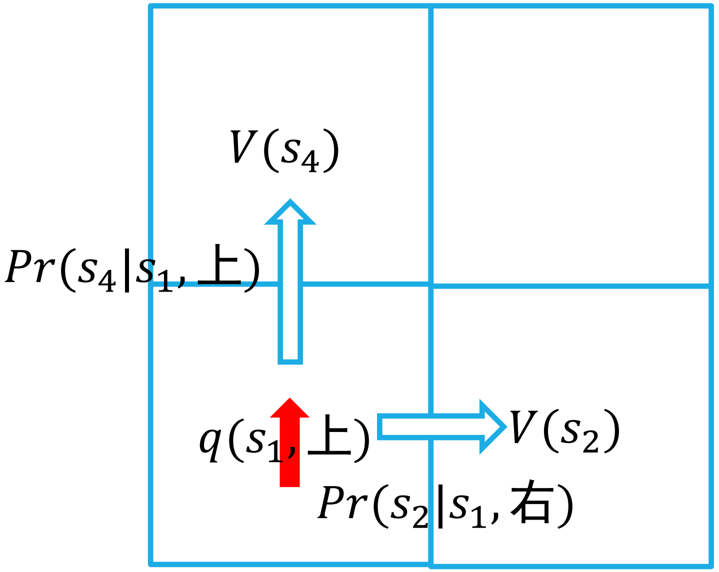
$$V_\pi(s_1) = \pi(s_1, 上) q_\pi(s_1, 上) + \pi(s_1, 右) q_\pi(s_1, 右)$$



不同动作下的反馈累加

$$q_\pi(s,a) = \sum_{s' \in S} Pr(s'|s,a)\,[R(s,a,s') + \gamma V_\pi(s')]$$

$$q_\pi(s_1, 上) = Pr(s_4|s_1, 上)[R(s_1, 上, s_4) + \gamma V_\pi(s_4)]$$



动作确定时状态转移后的反馈结果

贝尔曼方程（Bellman Equation）：刻画了价值函数和行动-价值函数自身以及两者相互之间的递推关系

$$V_\pi(s) = \sum_{a \in A} \pi(s,a) q_\pi(s,a) \qquad q_\pi(s,a) = \sum_{s' \in S} Pr(s'|s,a)\left[R(s,a,s') + \gamma V_\pi(s')\right]$$

将右式带入左式，得到价值函数的贝尔曼方程

$$V_\pi(s) = \sum_{a \in A} \pi(s,a) \sum_{s' \in S} Pr(s'|s,a)\left[R(s,a,s') + \gamma V_\pi(s')\right]$$

将左式带入右式，得到行动-价值函数的贝尔曼方程

$$q_\pi(s,a) = \sum_{s' \in S} Pr(s'|s,a)\left[R(s,a,s') + \gamma \sum_{a' \in A} \pi(s',a') q_\pi(s',a')\right]$$

将利用贝尔曼方程进行策略评估，进而进行策略优化

提 纲

1、 强化学习定义：马尔科夫决策过程

2、 强化学习中的策略优化与策略评估

3、 强化学习求解：Q-Learning

4、 深度强化学习：深度学习+强化学习

强化学习的问题与求解

强化学习的问题定义：给定马尔可夫决策过程$MDP = \{S, A, Pr, R, \gamma\}$

寻找一个最优策略π^*，对任意$s \in S$使得$V_{\pi^*}(s)$值最大

强化学习求解：在策略优化和策略评估的交替迭代中优化参数

策略优化，根据V_π优化π

价值函数
状态根据策略函数所采取行动后得到的反馈期望

$$\begin{tikzcd} V \arrow[r, bend left] & \pi \arrow[l, bend left] \end{tikzcd}$$

策略函数
每个状态所采取的行动

策略评估，根据π计算V_π

强化学习的求解方法

- **基于价值（Value-based）的方法**
  - 对价值函数进行建模和估计，以此为依据制订策略
- **基于策略（Policy-based）的方法**
  - 对策略函数直接进行建模和估计，优化策略函数使反馈最大化
- **基于模型（Model-based）的方法**
  - 对环境的运作机制建模，然后进行规划（planning）等

基于价值的求解方法

第一部分：策略优化（Policy Improvement）

策略优化定理：

对于确定的策略π和π'，如果对于任意状态$s \in S$

$$q_\pi\big(s, \pi'(s)\big) \geq q_\pi(s, \pi(s))$$

那么对于任意状态$s \in S$，有

$$V_{\pi'}(s) \geq V_\pi(s)$$

即策略π'不比π差

注意，不等式左侧的含义是只在当前这一步将动作修改为$\pi'(s)$，未来的动作仍然按照π的指导进行

因此给定当前策略π、价值函数V_π和行动-价值函数q_π时，可如下构造新的策略π'，只要π'满足如下条件：

$$\pi'(s) = \mathrm{argmax}_a q_\pi(s, a) \quad （对于任意s \in S）$$

π'便是对π的一个改进

基于价值的求解方法

第一部分：**策略优化（Policy Improvement）**

给定当前策略π、价值函数V_π和行动-价值函数q_π时，可如下构造新的策略π'，π'要满足如下条件：

$$\pi'(s) = \mathrm{argmax}_a q_\pi(s,a) \quad （对于任意s \in S）$$

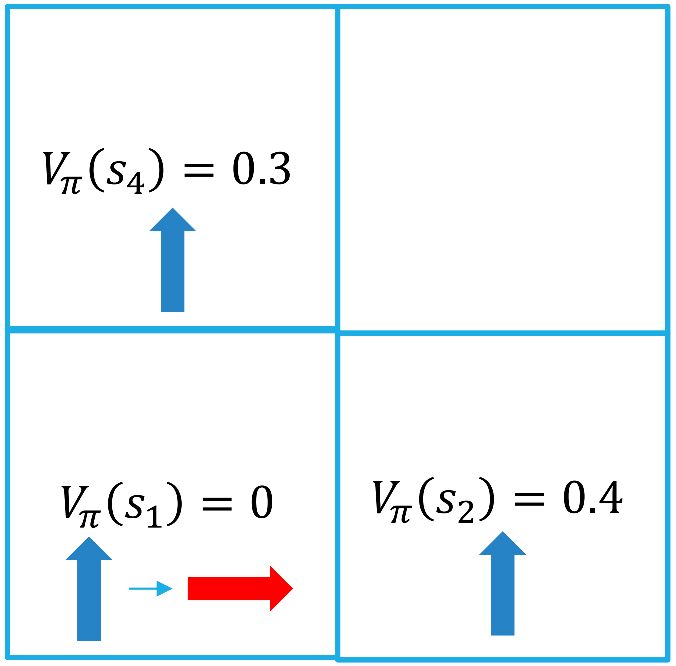
$$q_\pi(s,a) = \sum_{s' \in S} Pr(s'|s,a)\,[R(s,a,s') + \gamma V_\pi(s')]$$

假设当前价值函数在右图中给出，策略用箭头表示，对状态s_1而言：

注意这个问题里状态转移是确定的

$$q_\pi(s_1, 上) = 0 + 0.99 \times 0.3 = 0.297$$

$$q_\pi(s_1, 右) = \underbrace{\boxed{0}}_{R(s_1,右,s_2)} + \underbrace{\boxed{0.99}}_{\gamma} \times \underbrace{\boxed{0.4}}_{V_\pi(s_2)} = 0.396$$



因此根据策略优化原则$\pi'(s) = \mathrm{argmax}_a q_\pi(s,a)$

更新状态s_1策略 $\pi'(s_1) = \mathrm{argmax}_a q_\pi(s,a) = 右$

第二部分：策略评估(Policy Evaluation)

通过迭代计算贝尔曼方程进行策略评估

- 动态规划
- 蒙特卡洛采样
- 时序差分（Temporal Difference）

基于价值的求解方法

第二部分：策略评估(Policy Evaluation)

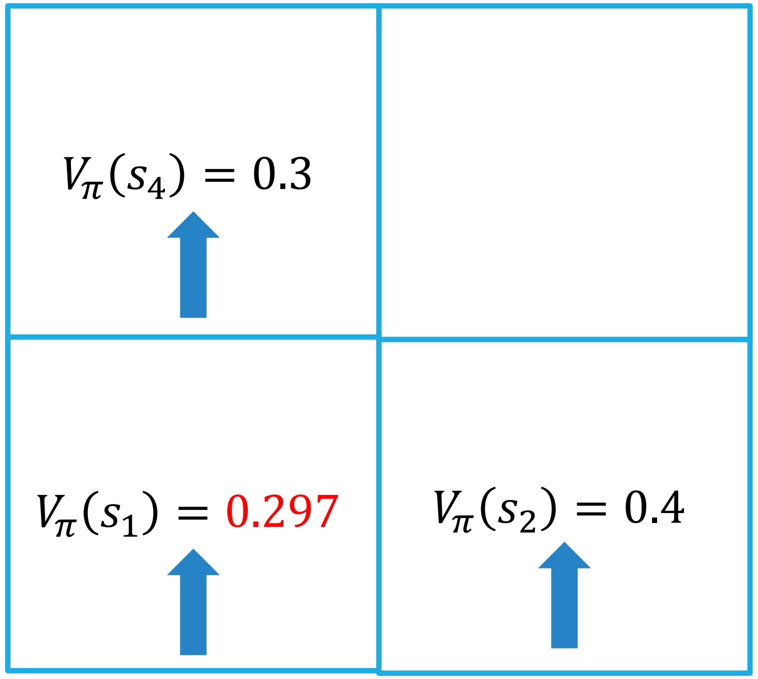
基于动态规划的价值函数更新：使用迭代的方法求解贝尔曼方程组

初始化V_π函数
循环
  枚举$s \in S$
    $V_\pi(s) \leftarrow \sum_{a\in A} \pi(s,a) \overset{q_\pi(s,a)}{\boxed{\sum_{s'\in S} Pr(s'|s,a)\,[R(s,a,s') + \gamma V_\pi(s')]}}$
直到V_π收敛

更新$V_\pi(s_1)$的值：

$$q_\pi(s_1,\text{上}) = 1 \times (0 + 0.99 \times 0.3) + 0 \times (0 + 0.99 \times 0.4)$$
$$+ \ldots = 0.297$$

$$V_\pi(s_1) = 1 \times q_\pi(s_1,\text{上}) + 0 \times q_\pi(s_1,\text{右}) = 0.297$$



动态规划法的缺点：1) 智能主体需要事先知道状态转移概率;2) 无法处理状态集合大小无限的情况

基于价值的求解方法

第二部分：策略评估(Policy Evaluation)

基于蒙特卡洛采样的价值函数更新

选择不同的起始状态，按照当前策略π采样若干轨迹，记它们的集合为D
枚举$s \in S$
    计算D中s每次出现时对应的反馈$G_1, G_2, \cdots, G_k$
    $V_\pi(s) \leftarrow \frac{1}{k}\sum_{i=1}^{k} G_i$

假设按照当前策略可样得到以下两条轨迹

$$(s_1, s_4, s_7, s_8, s_9)$$
$$(s_1, s_2, s_3, s_d)$$

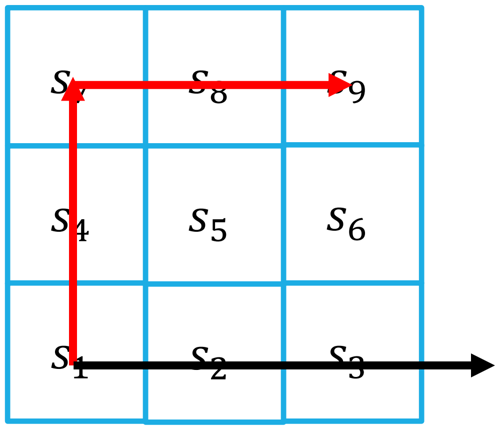
s_1对应的反馈值分别为

$$0 + \gamma \times 0 + \cdots + \gamma^3 \times 1 = 0.970$$
$$0 + \gamma \times 0 + \gamma^2 \times (-1) = -0.980$$

因此估计

$$V(s_1) = \frac{1}{2}(0.970 - 0.980) = -0.005$$

如果是确定的策略，每个起点只会产生一种轨迹



基于价值的求解方法

第二部分：**策略评估(Policy Evaluation)**

基于蒙特卡洛采样的价值函数更新

选择不同的起始状态，按照当前策略π采样若干轨迹，记它们的集合为D
枚举$s \in S$
    计算D中s每次出现时对应的反馈$G_1, G_2, \cdots, G_k$
    $V_\pi(s) \leftarrow \frac{1}{k}\sum_{i=1}^{k} G_i$

蒙特卡洛采样法的优点

- 智能主体不必知道状态转移概率
- 容易扩展到无限状态集合的问题中

蒙特卡洛采样法的缺点

- 状态集合比较大时，一个状态在轨迹可能非常稀疏，不利于估计期望
- 在实际问题中，最终反馈需要在终止状态才能知晓，导致反馈周期较长

基于价值的求解方法

第二部分：策略评估(Policy Evaluation)

基于时序差分（Temporal Difference）的价值函数更新

初始化V_π函数
循环
　　初始化s为初始状态
　　循环
　　　　$a \sim \pi(s,\cdot)$
　　　　执行动作a，观察奖励R和下一个状态s'
　　　　更新$V_\pi(s) \leftarrow V_\pi(s) + \alpha[\underline{R(s,a,s') + \gamma V_\pi(s')} - \underline{V_\pi(s)}]$
　　　　$s \leftarrow s'$
　　直到s是终止状态
直到V_π收敛

- 根据贝尔曼方程$V_\pi(s) = \mathbb{E}_{a\sim\pi(s,\cdot),s'\sim Pr(\cdot|s,a)}[R(s,a,s') + \gamma V_\pi(s')]$
- 利用蒙特卡洛采样的思想，通过采样a和s'来估计期望
- $R(s,a,s') + \gamma V_\pi(s')$是对$V_\pi(s)$的一个估计值
- 部分更新$V_\pi(s)$的值：$V_\pi(s) \leftarrow (1-\alpha)\underbrace{V_\pi(s)}_{\text{过去的价值函数值}} + \underbrace{\alpha[R(s,a,s') + \gamma V_\pi(s')]}_{\text{学习得到的价值函数值}}$

基于价值的求解方法

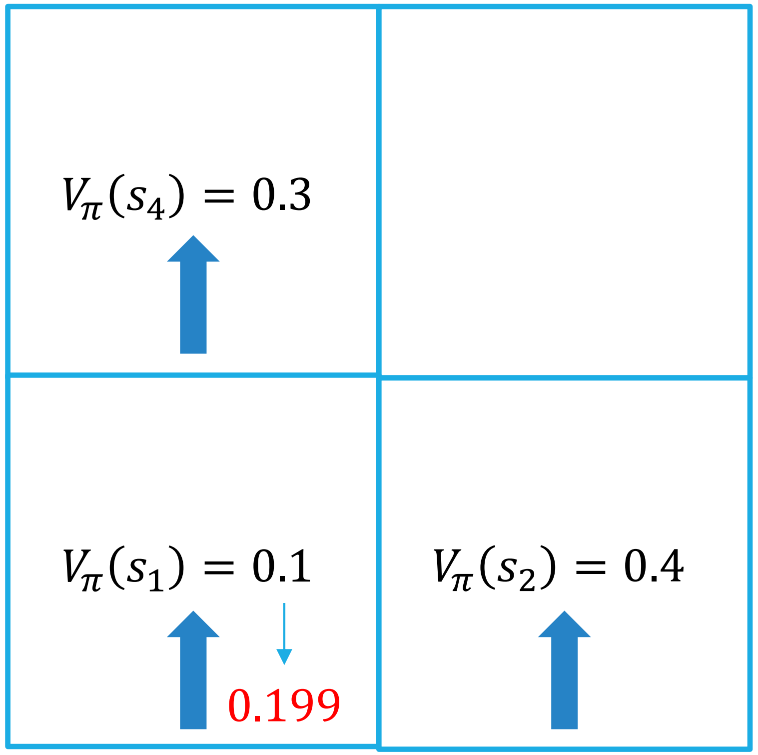
第二部分：策略评估(Policy Evaluation)

基于时序差分（Temporal Difference）的价值函数更新

初始化V_π函数
循环
　　初始化s为初始状态
　　循环
　　　　$a \sim \pi(s,\cdot)$
　　　　执行动作a，观察奖励R和下一个状态s'
　　　　更新$V_\pi(s) \leftarrow V_\pi(s) + \alpha[R(s,a,s') + \gamma V_\pi(s') - V_\pi(s)]$
　　　　$s \leftarrow s'$
　　直到s是终止状态
直到V_π收敛

假设$\alpha = 0.5$，更新$V_\pi(s_1)$的值：
从$\pi(s_1,\cdot)$ 中采样得到动作$a =$ 上
从$Pr(\cdot\,|s_1,$上$)$中采样得到下一步状态$s' = s_4$

$$V_\pi(s_1) \leftarrow V_\pi(s_1) + \alpha[R(s_1, 上, s_4) + \gamma V_\pi(s_4) - V_\pi(s_1)]$$
$$= 0.1 + 0.5 \times [0 + 0.99 \times 0.3 - 0.1] = 0.199$$



提 纲

1、 强化学习定义：马尔科夫决策过程

2、 强化学习中的策略优化与策略评估

3、 强化学习求解：Q-Learning

4、 深度强化学习：深度学习+强化学习

基于价值的求解方法

第一部分和第二部分结合：**策略优化与策略评估结合**

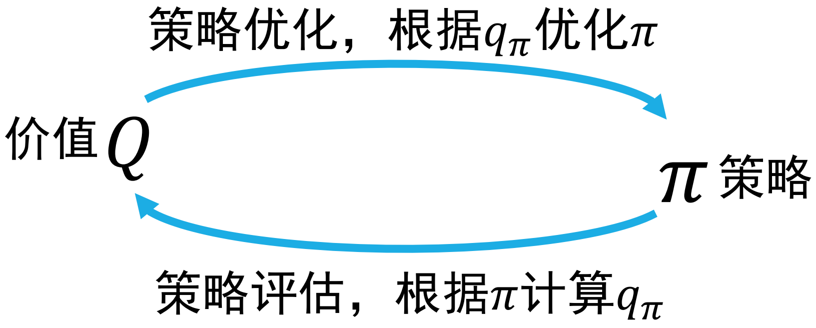
基于时序差分的方法 – Q学习（Q-Learning）[Q: quality]

初始化q_π函数
循环
　　初始化s为初始状态
　　循环
　　　　$\cancel{a \sim \pi(s,\cdot)}$ ➡ $a = \boxed{\operatorname{argmax}_{a'}} q_\pi(s,a')$
　　　　执行动作a，观察奖励R和下一个状态s'
　　　　~~更新~~$\cancel{V_\pi(s) \leftarrow V_\pi(s) + \alpha[R + \gamma V_\pi(s') - V_\pi(s)]}$
　　➡ 更新$q_\pi(s,a) \leftarrow q_\pi(s,a) + \alpha\left[R + \gamma \boxed{\max_{a'}} q_\pi(s',a') - q_\pi(s,a)\right]$
　　　　$s \leftarrow s'$
　　直到s是终止状态
直到q_π收敛

策略优化：$\pi'(s) = \operatorname{argmax}_a q_\pi(s,a)$

$$q_\pi(s,a) \leftarrow (1-\alpha)q_\pi(s,a) + \alpha\left[R + \gamma \max_{a'} q_\pi(s',a')\right]$$

- 基于价值的方法不直接对策略建模，因此策略优化在采样和更新两步中之max操作上得以间接体现
- 在同一次循环中策略评估和策略优化交替进行
- 由于策略优化要求计算动作-价值函数q，因此Q学习直接利用q函数的贝尔曼方程进行更新



基于价值的方法 – Q 学习

s_d	s_7	s_8	s_9
	s_4	s_5	s_6
	s_1	s_2	s_3

初始化q_π函数
循环
    初始化s为初始状态
    循环
        $a = \mathrm{argmax}_{a'} q_\pi(s,a')$
        执行动作a，观察奖励R和下一个状态s'
        更新$q_\pi(s,a) \leftarrow q_\pi(s,a) + \alpha\left[R + \gamma \max_{a'} q_\pi(s',a') - q_\pi(s,a)\right]$
        $s \leftarrow s'$
    直到s是终止状态
直到q_π收敛

初始化q_π函数

在右图中，${}^{a}/_{b}$表示$q_\pi(s,上) = a, q_\pi(s,右) = b$

所有终止状态的q函数值设为${}^{0}/_{0}$，其余状态可随机初始化，此处设${}^{0.2}/_{0}$

初始化s，s的值在右图中用黑框框出

${}^{0}/_{0}$	${}^{0.2}/_{0}$	${}^{0.2}/_{0}$	${}^{0}/_{0}$
	${}^{0.2}/_{0}$	${}^{0.2}/_{0}$	${}^{0.2}/_{0}$
	$\boxed{{}^{0.2}/_{0}}$	${}^{0.2}/_{0}$	${}^{0.2}/_{0}$

基于价值的方法 – Q 学习

s_d	s_7	s_8	s_9
	s_4	s_5	s_6
	s_1	s_2	s_3

初始化q_π函数
循环
  初始化s为初始状态
  循环
    $a = \mathrm{argmax}_{a'} q_\pi(s, a')$
    执行动作a，观察奖励R和下一个状态s'
    更新$q_\pi(s,a) \leftarrow q_\pi(s,a) + \alpha\left[R + \gamma \max_{a'} q_\pi(s',a') - q_\pi(s,a)\right]$
    $s \leftarrow s'$
  直到s是终止状态
直到q_π收敛

$a = \mathrm{argmax}_{a'} q_\pi(s_1, a') =$ 上

$R = 0$，$s' = s_4$

$$q_\pi(s_1, 上) \leftarrow 0.2 + 0.5 \times [0 + 0.99 \times \max\{0, 0.2\} - 0.2] = 0.199$$

$s \leftarrow s_4$

$^0/_0$	$^{0.2}/_0$	$^{0.2}/_0$	$^0/_0$
	$\boxed{^{0.2}/_0}$	$^{0.2}/_0$	$^{0.2}/_0$
	$^{0.199}/_0$	$^{0.2}/_0$	$^{0.2}/_0$

基于价值的方法 – Q 学习

s_d

s_7	s_8	s_9
s_4	s_5	s_6
s_1	s_2	s_3

初始化q_π函数
循环
　　初始化s
　　循环
　　　　$a = \text{argmax}_{a'} q_\pi(s, a')$
　　　　执行动作a，观察奖励R和下一个状态s'
　　　　更新$q_\pi(s,a) \leftarrow q_\pi(s,a) + \alpha \left[R + \gamma \max_{a'} q_\pi(s',a') - q_\pi(s,a)\right]$
　　　　$s \leftarrow s'$
　　直到s是终止状态
直到q_π收敛

$^0/_0$

$\boxed{^{0.2}/_0}$	$^{0.2}/_0$	$^0/_0$
$^{0.199}/_0$	$^{0.2}/_0$	$^{0.2}/_0$
$^{0.199}/_0$	$^{0.2}/_0$	$^{0.2}/_0$

$a = \text{argmax}_{a'} q_\pi(s_4, a') = 上$

$R = 0$，$s' = s_7$

$$q_\pi(s_4, 上) \leftarrow 0.2 + 0.5 \times [0 + 0.99 \times \max\{0,0.2\} - 0.2] = 0.199$$

$s \leftarrow s_7$

基于价值的方法 – Q学习

s_d

s_7	s_8	s_9
s_4	s_5	s_6
s_1	s_2	s_3

初始化q_π函数
循环
  初始化s
  循环
    $a = \text{argmax}_{a'} q_\pi(s, a')$
    执行动作a，观察奖励R和下一个状态s'
    更新$q_\pi(s,a) \leftarrow q_\pi(s,a) + \alpha\left[R + \gamma \max\limits_{a'} q_\pi(s',a') - q_\pi(s,a)\right]$
    $s \leftarrow s'$
  直到s是终止状态
直到q_π收敛

$a = \text{argmax}_{a'} q_\pi(s_7, a') =$ 上
$R = -1$，$s' = s_d$
$q_\pi(s_7, \text{上}) \leftarrow 0.2 + 0.5 \times [-1 + 0.99 \times \max\{0,0\} - 0.2] = -0.4$
$s \leftarrow s_d$

因为s_d是终止状态，因此一个片段（episode）结束

$^{0}/_{0}$

$^{-0.4}/_{0}$	$^{0.2}/_{0}$	$^{0}/_{0}$
$^{0.199}/_{0}$	$^{0.2}/_{0}$	$^{0.2}/_{0}$
$^{0.199}/_{0}$	$^{0.2}/_{0}$	$^{0.2}/_{0}$

基于价值的方法 – Q学习

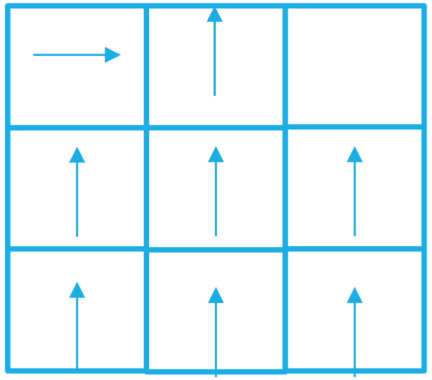
q函数

$-0.4/0$	$0.2/0$	$0/0$
$0.199/0$	$0.2/0$	$0.2/0$
$0.199/0$	$0.2/0$	$0.2/0$

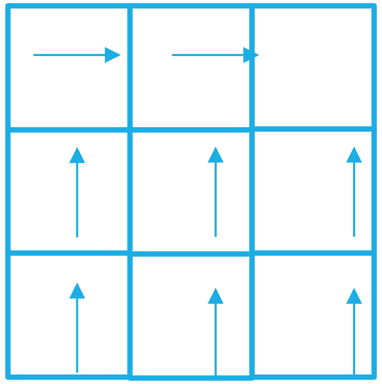
$-0.4/0.099$	$-0.4/0$	$0/0$
$0.100/0$	$0.2/0$	$0.2/0$
$0.198/0$	$0.2/0$	$0.2/0$

$-0.4/0.050$	$-0.4/0.5$	$0/0$
$0.099/0$	$0.2/0$	$0.2/0$
$0.148/0$	$0.2/0$	$0.2/0$

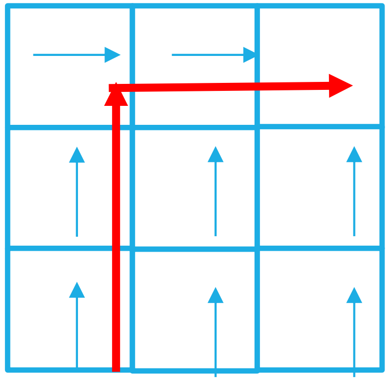
策略



第一个片段后



第二个片段后



第三个片段后

基于价值的方法 – Q学习

行百里者半于九十

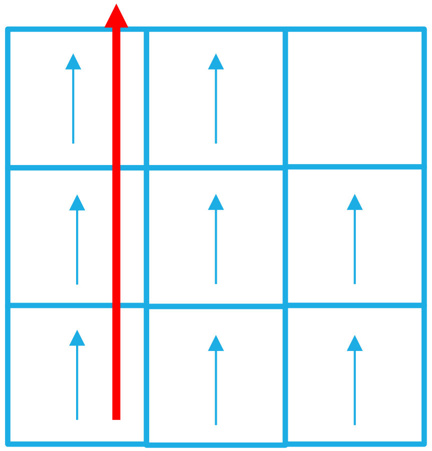
q函数

$^1/_{-2}$	$^1/_{-2}$	$^0/_0$
$^1/_{-2}$	$^1/_{-2}$	$^1/_{-2}$
$^1/_{-2}$	$^1/_{-2}$	$^1/_{-2}$

……

$^{-1}/_{-2}$	$^1/_{-2}$	$^0/_0$
$^{-0.990}/_{-2}$	$^1/_{-2}$	$^1/_{-2}$
$^{-0.980}/_{-2}$	$^1/_{-2}$	$^1/_{-2}$

策略



第n个片段后

- 如果q函数的初始化为$^1/_{-2}$，在模型收敛后，策略仍无法使得智能主体找到目标状态
- 这种情况并非个例，例如状态s_7往上走的期望反馈保持0.2、往右走的期望反馈保持0的条件下，将s_7往上走的惩罚值适当减少（在先前例子中，该惩罚值为-1），会得到类似的效果

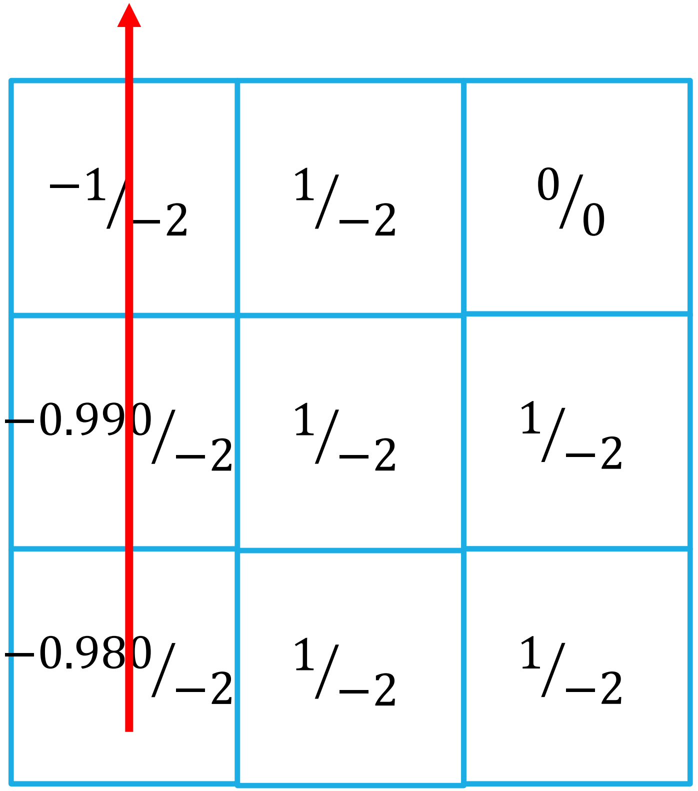
  问题：这种情况出现的原因为何？

策略学习中探索（exploration）与利用（exploitation）的平衡

问题：为何Q学习收敛到非最优策略？

观察每个片段的轨迹

- (s_1, s_4, s_7, s_d)
- (s_1, s_4, s_7, s_d)
- ……
- (s_1, s_4, s_7, s_d)



智能主体的策略（即按照动作-价值函数选择反馈最大的行为）始终不变，因此与环境交互的轨迹是固定的，过程中没有得到任何有关目标s_9的信息

- 外力：缺乏推动智能主体改变策略的外在因素
- 内因：智能主体缺乏从内部改变策略的动力

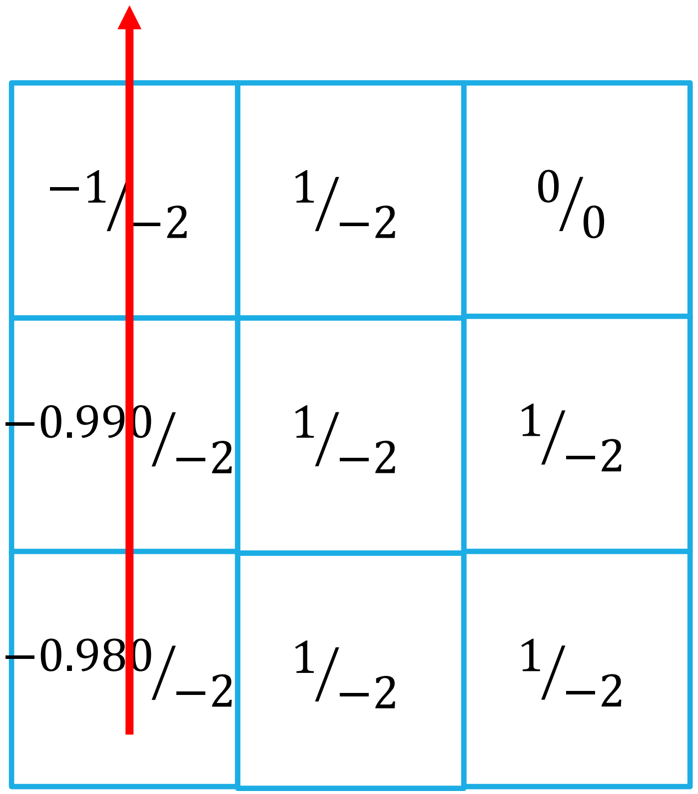
智能主体的“创新精神”：

- 根据目前已知的最优策略来选择动作，被称为**利用（exploitation）**
- 不根据当前策略而去尝试未知的动作被称为**探索（exploration）**

策略学习中探索（exploration）与利用（exploitation）的平衡

问题：为何Q学习收敛到非最优策略？

回答：算法中只有利用没有探索



探索与利用之间如何取得平衡

- 只利用而不探索
- 只探索而不利用（则训练过程完全没有意义 ）
- 大体上利用，偶尔探索

ϵ贪心（ϵ-greedy）策略

$$\epsilon\text{-}greedy_\pi(s)=\begin{cases}\text{argmax}_a q_\pi(s,a), & \text{以}1-\epsilon\text{的概率}\\ \text{随机的}a\in A\ , & \text{以}\epsilon\text{的概率}\end{cases}$$

ϵ贪心策略的解释：大体上遵循最优策略的决定，偶尔（以ϵ的小概率）进行探索

$\epsilon\text{-}greedy_\pi$策略是非确定的策略，严格来说应该写成概率形式，此处用了其简化表达

策略学习中探索（exploration）与利用（exploitation）的平衡

使用ϵ贪心策略的Q学习

初始化q_π函数
循环
　　初始化s为初始状态
　　循环
　　　　采样$a \sim \boxed{\epsilon - greedy_\pi(s)}$
　　　　执行动作a，观察奖励R和下一个状态s'
　　　　更新$q_\pi(s,a) \leftarrow q_\pi(s,a) + \alpha\left[R + \gamma \boxed{\max_{a'} q_\pi(s',a')} - q_\pi(s,a)\right]$
　　　　$s \leftarrow s'$
　　直到s是终止状态
直到q_π收敛

采样策略与更新策略不同

- 将动作采样从“确定地选取最优动作”改为“按照ϵ贪心策略选取动作”
- 更新时仍保持用max操作选取最佳策略。像这样更新时的目标策略与采样策略不同的方法，叫做离策略（off-policy）方法

基于价值的方法 – Q学习

***q*函数**

$^{1}/_{-2}$	$^{1}/_{-2}$	$^{0}/_{0}$
$^{1}/_{-2}$	$^{1}/_{-2}$	$^{1}/_{-2}$
$^{1}/_{-2}$	$^{1}/_{-2}$	$^{1}/_{-2}$

……

$^{-0.99}/_{0.78}$	$^{-1.00}/_{1}$	$^{0}/_{0}$
$^{0.12}/_{0.98}$	$^{0.99}/_{0.80}$	$^{1}/_{-2}$
$^{0.97}/_{0.24}$	$^{0.98}/_{-2}$	$\mathbf{^{1}/_{-2}}$

令$\epsilon = 0.1$，采用探索策略的Q学习，执行了100个片段后

- 学习得到的策略能够将智能主体导向目标s_9
- 仍有部分状态是没有被探索过的（s_3）
- 增大ϵ值有助于算法去探索这些未知状态

探索与利用
相互平衡策略

→	→	
→	↑	↑
↑	↑	↑

第100个片段后

提 纲

1、 强化学习定义：马尔科夫决策过程

2、 强化学习中策略优化与策略评估

3、 强化学习求解：Q-Learning

4、 深度强化学习：深度学习+强化学习

用神经网络拟合（行动）价值函数

- 状态数量太多时，有些状态可能始终无法采样到，因此对这些状态的q函数进行估计是很困难的
- 状态数量无限时，不可能用一张表（数组）来记录q函数的值

使用ϵ贪心策略的Q学习

初始化q_π函数
循环
　　初始化s为初始状态
　　循环
　　　　采样$a \sim \epsilon\text{-}greedy_\pi(s)$
　　　　执行动作a，观察奖励R和下一个状态s'
　　　　更新$q_\pi(s,a) \leftarrow q_\pi(s,a) + \alpha\left[R + \gamma \max_{a'} q_\pi(s',a') - q_\pi(s,a)\right]$
　　　　$s \leftarrow s'$
　　直到s是终止状态
直到q_π收敛

思路：将q函数参数化（parametrize），用一个非线性回归模型来拟合q函数，例如（深度）神经网络

- 能够用有限的参数刻画无限的状态
- 由于回归函数的连续性，没有探索过的状态也可通过周围的状态来估计

深度Q学习与梯度下降法

用深度神经网络拟合q函数

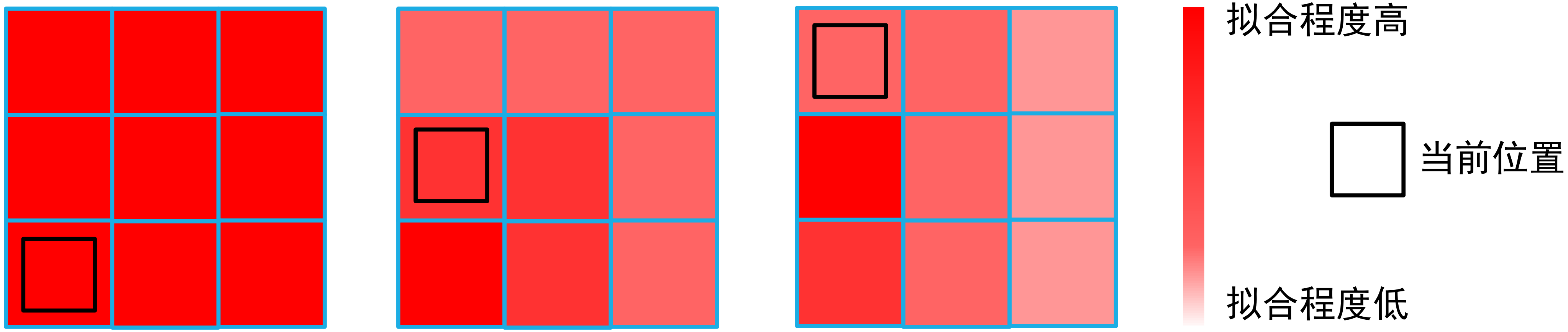
初始化q_π函数的参数θ
循环
    初始化s为初始状态
    循环
        采样$a \sim \epsilon\text{-}greedy_\pi(s;\theta)$
        执行动作a，观察奖励R和下一个状态s'
        损失函数$L(\theta) = \frac{1}{2}\left[R + \gamma \max_{a'} q_\pi(s',a';\theta) - q_\pi(s,a;\theta)\right]^2$
        根据梯度$\partial L(\theta)/\partial\theta$更新参数$\theta$
        $s \leftarrow s'$
    直到s是终止状态
直到q_π收敛

- 损失函数刻画了q的估计值$R + \gamma \max_{a'} q_\pi(s',a';\theta)$与当前值的平方误差
- 利用梯度下降法优化参数θ
- 如果用深度神经网络来拟合q函数，则**算法称为深度Q学习**

深度Q学习的两个不稳定因素

1. 相邻的样本来自同一条轨迹，样本之间相关性太强，集中优化相关性强的样本可能导致神经网络在其他样本上效果下降。

集中优化一条轨迹上的状态时，远离该轨迹的状态的估计值可能会发生较大偏离



2. 在损失函数中，q函数的值既用来估计目标值，又用来计算当前值。现在这两处的q函数通过θ有所关联，可能导致优化时不稳定

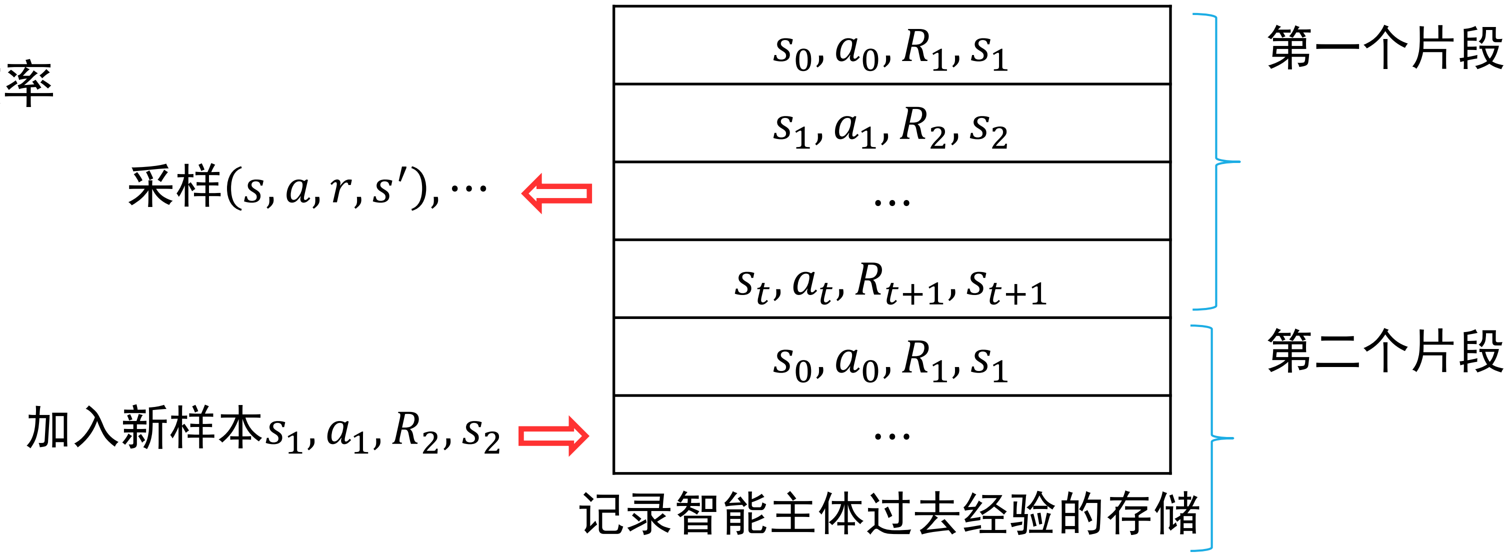
$$\frac{1}{2}\Big[\underbrace{R+\gamma\max_{a'} q_\pi(s',a';\theta)}_{\textbf{预测值}} - \underbrace{q_\pi(s,a;\theta)}_{\textbf{当前值}}\Big]^2$$

经验重现（Experience Replay）

- 相邻的样本来自同一条轨迹，样本之间相关性太强，集中优化相关性强的样本可能导致神经网络在其他样本上效果下降。

将过去的经验存储下来，每次将新的样本加入到存储中去，并从存储中采样一批样本进行优化

- 解决了样本相关性强的问题
- 重用经验，提高了信息利用的效率



目标网络（Target Network）

- 在损失函数中，q函数的值既用来估计目标值，又用来计算当前值。现在这两处的q函数通过θ有所关联，可能导致优化时不稳定

$$\frac{1}{2}\left[R+\gamma \max_{a'} \overset{\text{目标网络}}{\boxed{q_\pi(s',a';\theta^-)}} - q_\pi(s,a;\theta)\right]^2$$

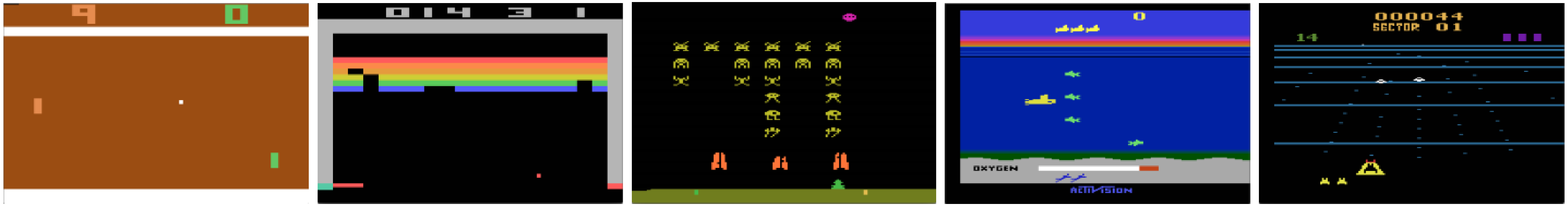
损失函数的两个q函数使用不同的参数计算

- 用于计算估计值的q使用参数θ^-计算，这个网络叫做目标网络
- 用于计算当前值的q使用参数θ计算
- 保持θ^-的值相对稳定，例如θ每更新多次后才同步两者的值

$$\theta^- \leftarrow \theta$$

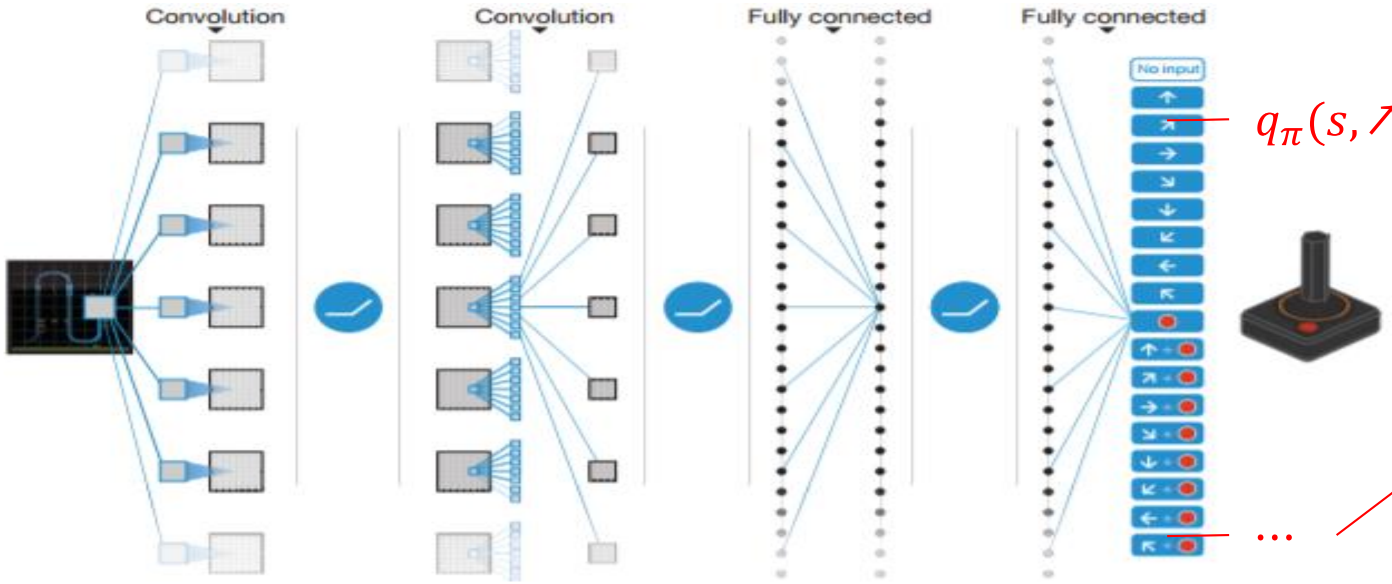
深度Q学习的应用实例

雅达利游戏



卷积神经网络

游戏画面



$q_{\pi}(s, \nearrow)$

…

q函数的回归值

q函数的学习模型

Mnih, Volodymyr, et al, Human-level control through deep reinforcement learning, Nature 518.7540 (2015)