强化学习2022
第14节

涉及知识点：
多智能体强化学习、多智能体强化学习入门、
多智能体强化学习进阶

多智能体强化学习

张伟楠 – 上海交通大学

课程大纲

强化学习基础部分

1. 强化学习、探索与利用
2. MDP和动态规划
3. 值函数估计
4. 无模型控制方法
5. 规划与学习
6. 参数化的值函数和策略
7. 深度强化学习价值方法
8. 深度强化学习策略方法

强化学习前沿部分

9. 基于模型的深度强化学习
10. 模仿学习
11. 离线强化学习
12. 参数化动作空间
13. 目标导向的强化学习
14. 多智能体强化学习
15. 强化学习大模型
16. 技术交流与回顾

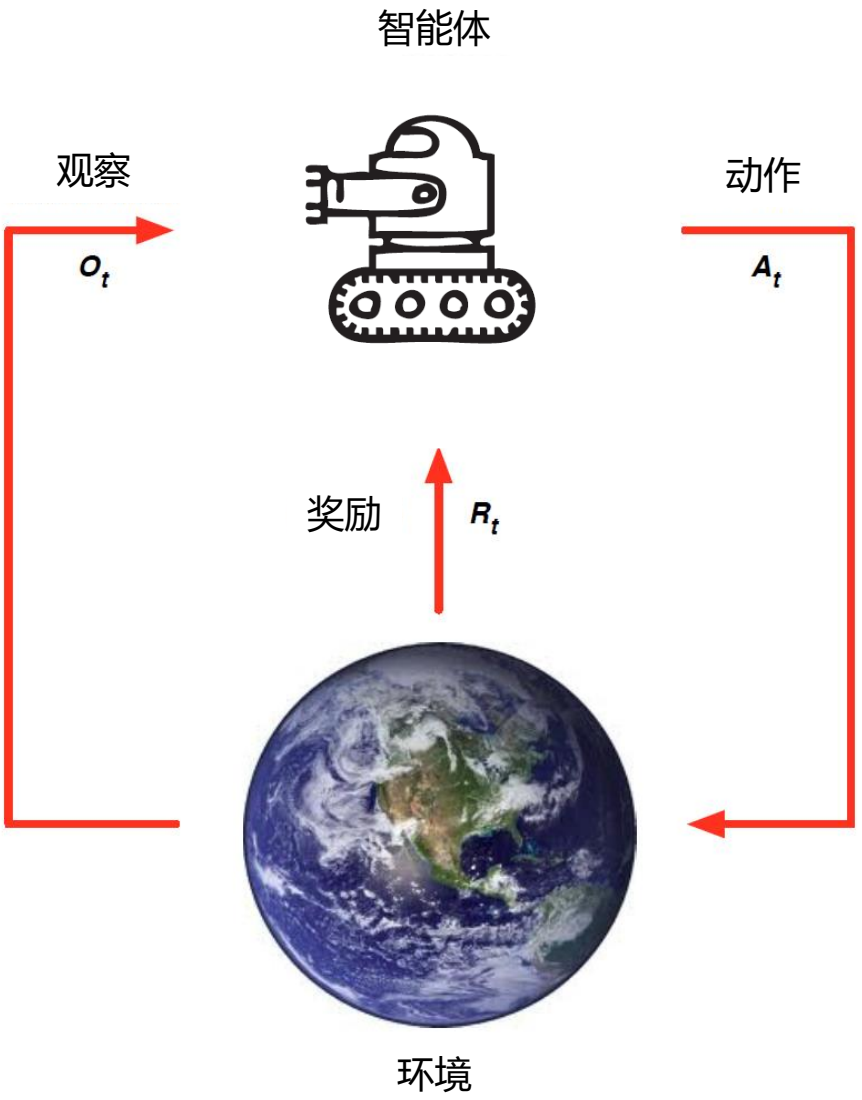
多智能体强化学习

张伟楠 – 上海交通大学

强化学习

在与环境的交互过程中学习

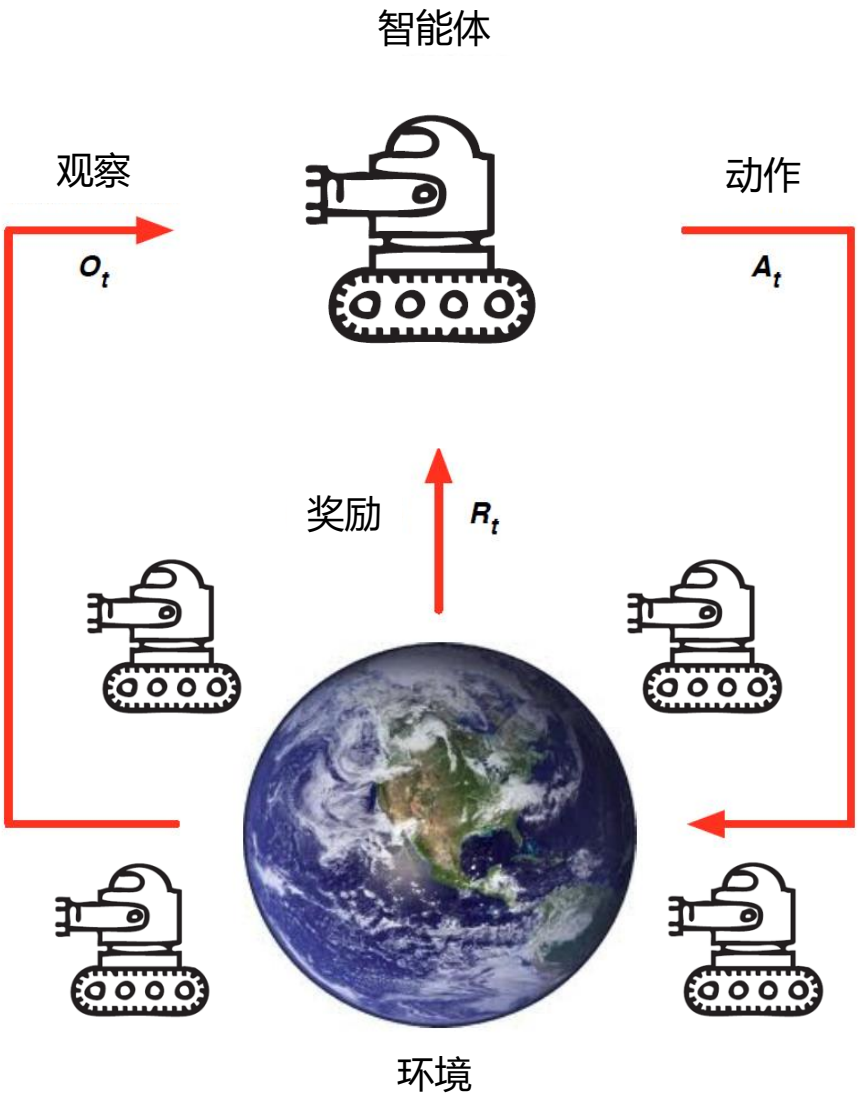
- 智能体
  - 感知来自环境的观测结果 O_t
  - 采取动作作用于环境 A_t
  - 从环境获得奖励信号 R_t
- 通常来讲，环境是稳态的(stationary)
  - 也即是环境的状态迁移是随机的，但随机分布是不变的



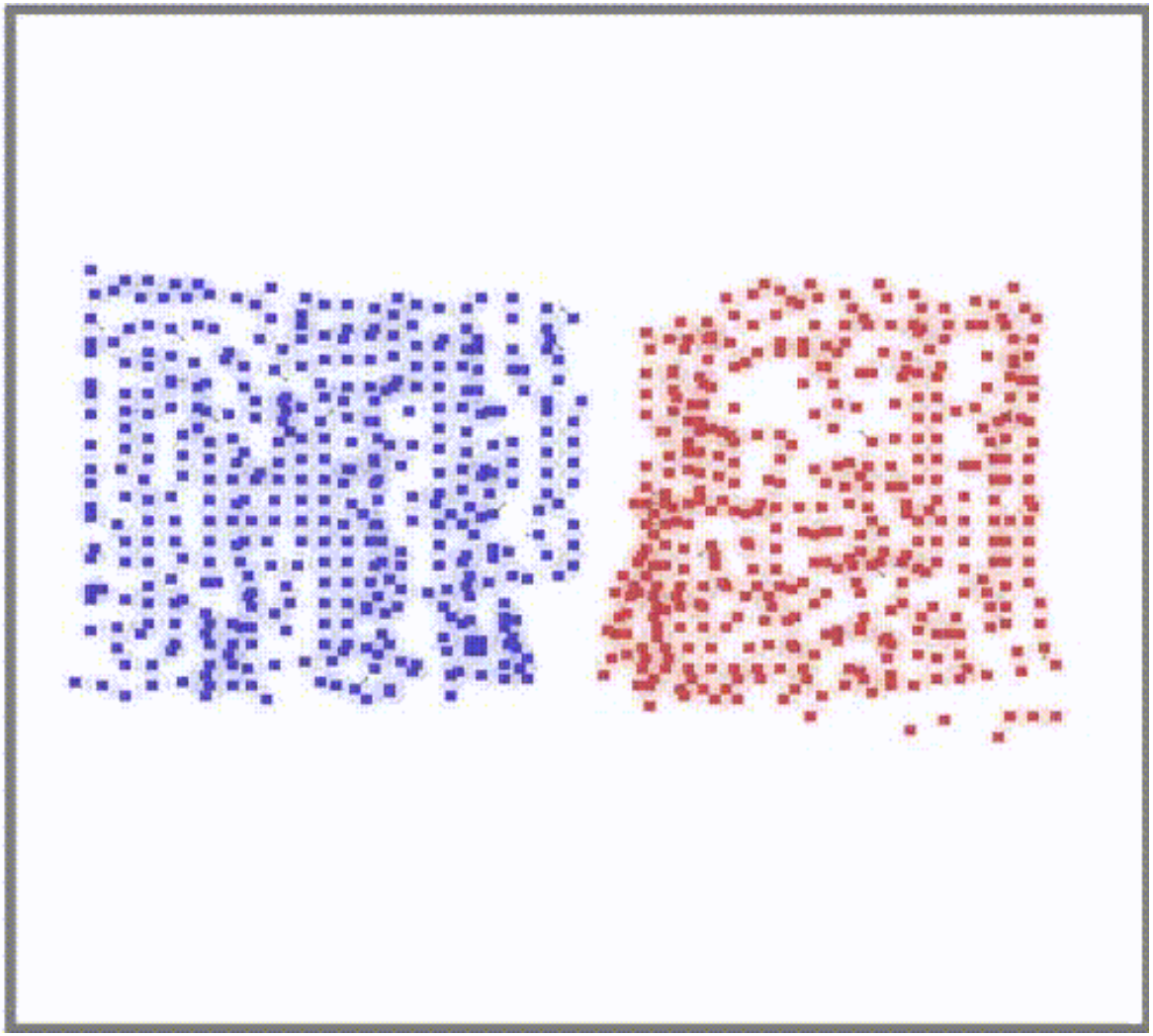
多智能体强化学习

在与环境的交互过程中学习

- 环境包含有不断进行学习和更新的其他智能体
- 在任何一个智能体的视角下，环境是非稳态的（non-stationary）
  - 环境迁移的分布会发生改变



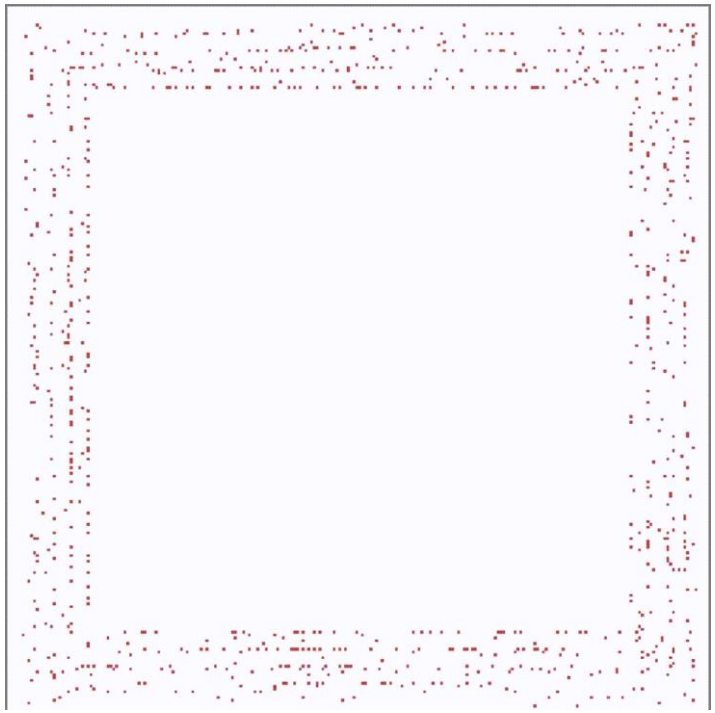
例1：对战游戏

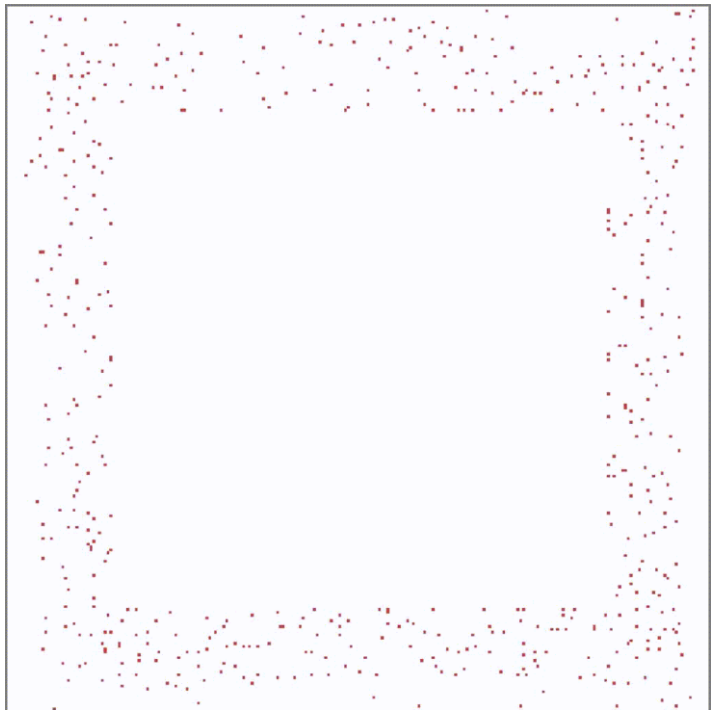


Lianmin Zheng, Weinan Zhang et al. Magent: a many-agent reinforcement learning platform for artificial collective intelligence. NIPS17 & AAAI18.

例2：队伍排列

- 让一整支智能体队伍排列成特定的样式





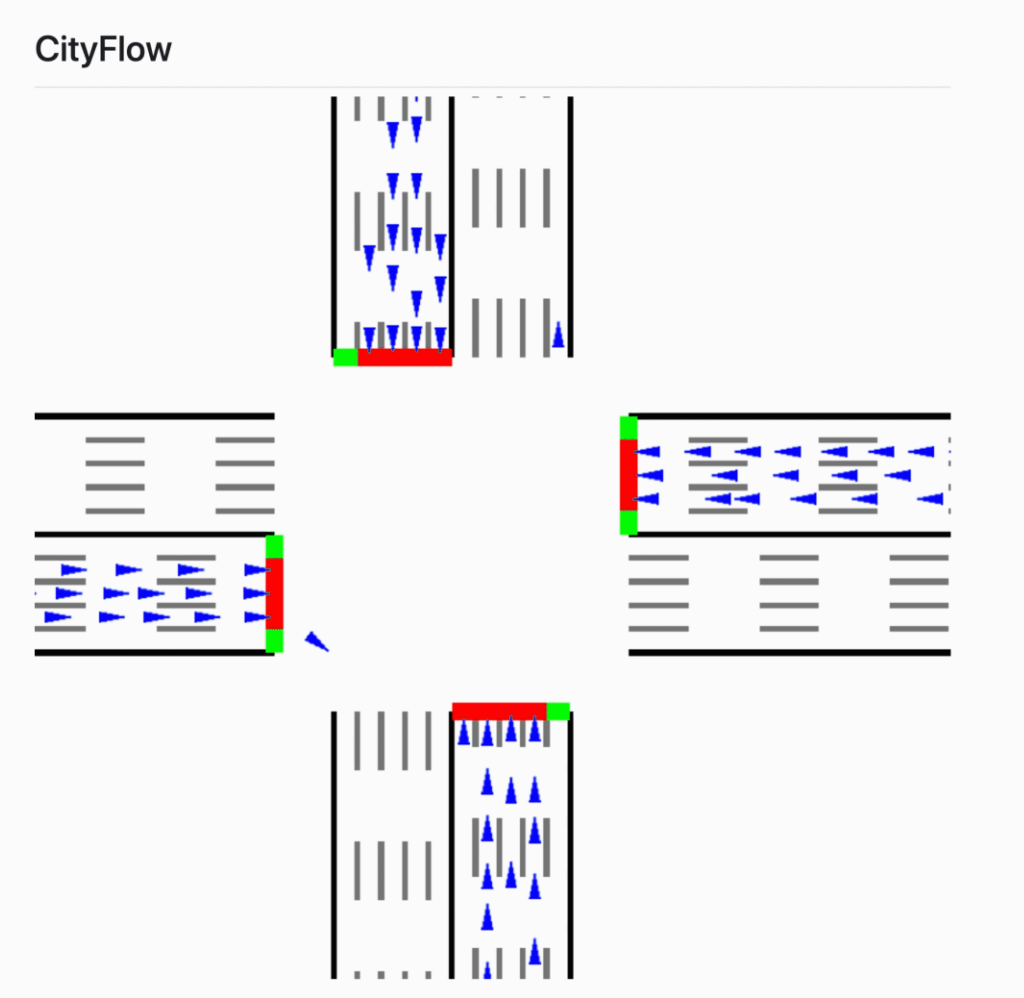
Lianmin Zheng, Weinan Zhang et al. Magent: a many-agent reinforcement learning platform for artificial collective intelligence. NIPS17 & AAAI18.

例3：去中心化的游戏人工智能（AI）

- 为复杂的集体游戏智能设计多智能体通信和协同学习的算法

Peng, Peng, et al. "Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games." *NIPS workshop 2017.*

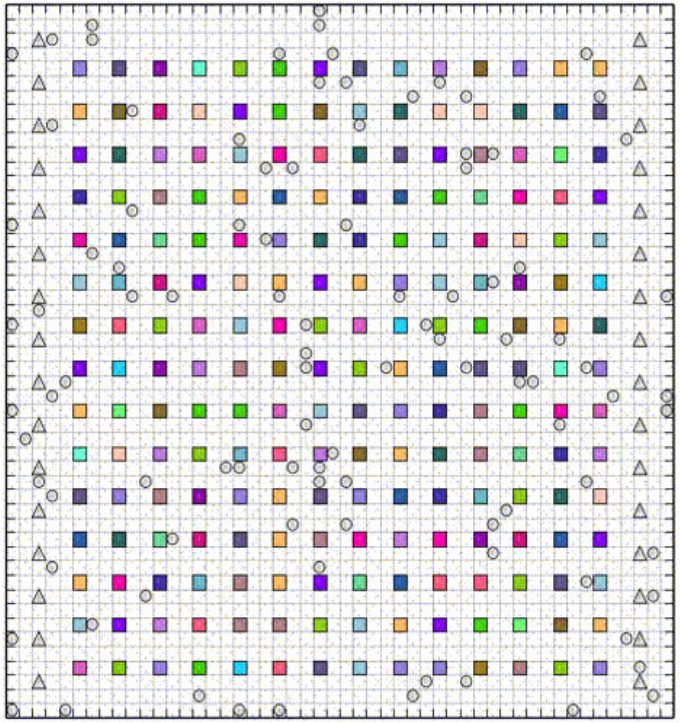
例4：城市大脑模拟器



- 设计
  - 车辆路由策略
  - 交通灯控制
  - 车队管理及出租车派车

Huichu Zhang, Weinan Zhang et al. CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario. WWW 2019.

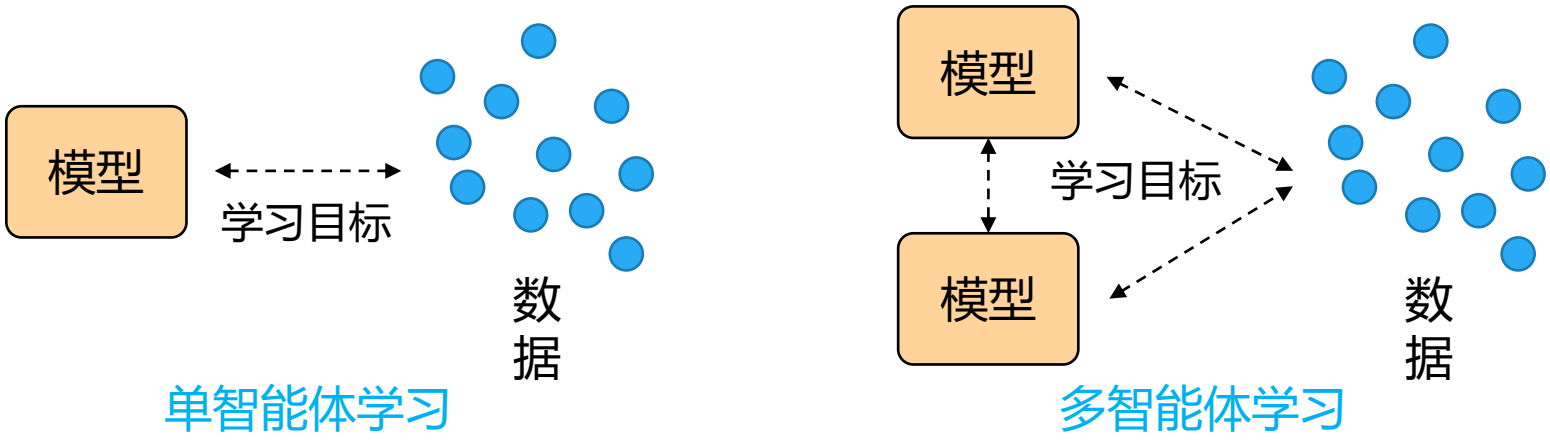
例5：分拣机器人



Haifeng Zhang, Weinan Zhang et al. Layout design for intelligent warehouse by evolution with fitness approximation. IEEE Access 2019.

多智能体学习的难点

- 多智能体学习（MAL）从原理上来讲更加困难
  - 智能体不仅要与环境进行交互，还要相互之间进行交互



- 假如把其他智能体考虑成环境的一部分从而能够使用单智能体的Q学习算法，是否可行?
  - 这种做法破坏了理论上的收敛保证，使学习不稳定
    - 即，一个智能体策略的改变会影响其智能体的策略，反之亦然

多智能体强化学习入门

张伟楠 – 上海交通大学

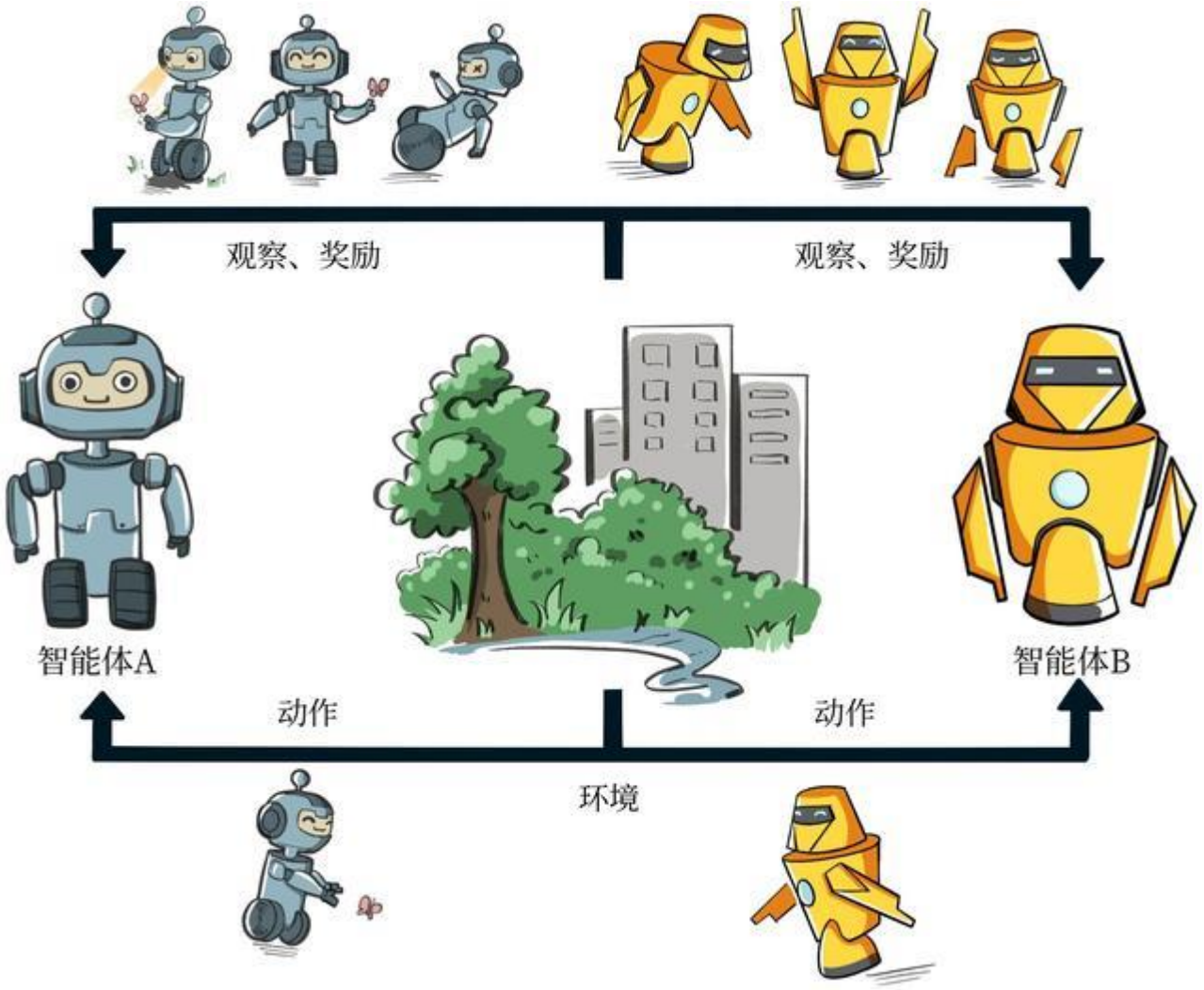
目录

Contents

01

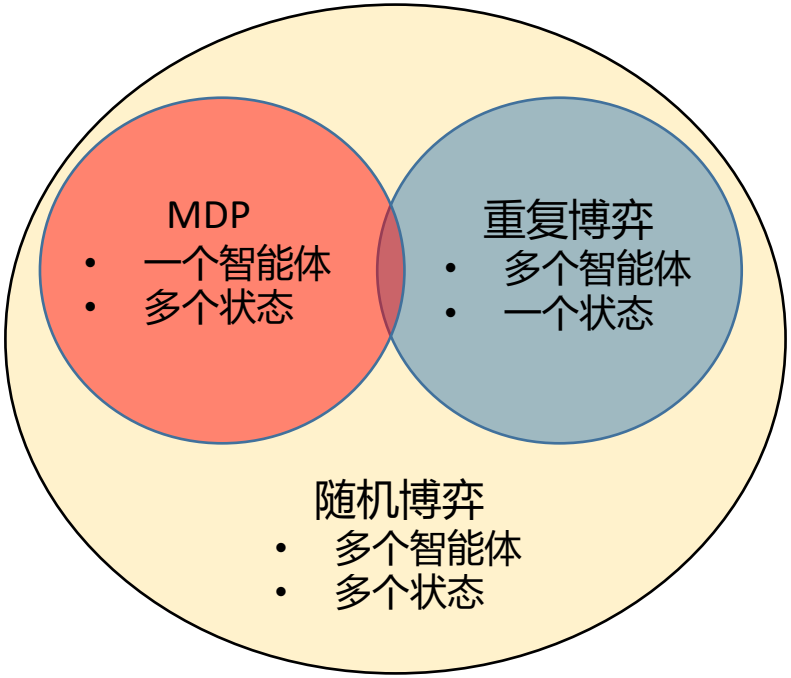
问题定义

多智能体强化学习



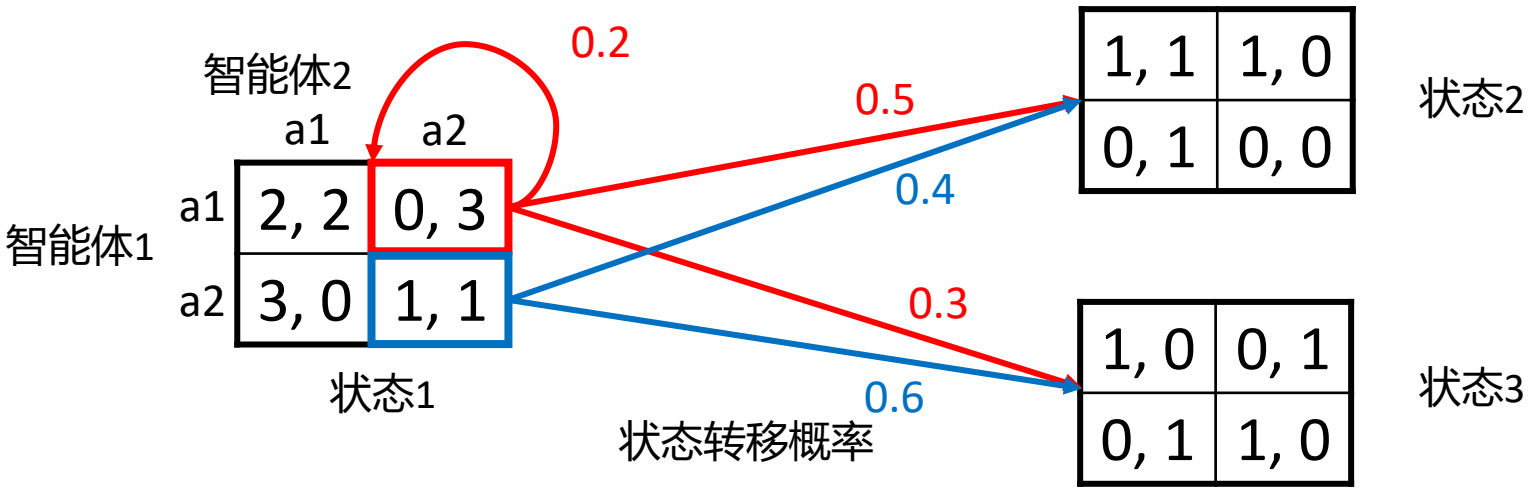
序贯决策的三种问题

- 在博弈论视角下，序贯决策的三种问题
  1. 马尔可夫决策过程
     - 一个智能体
     - 多个状态
  2. 重复博弈
     - 多个智能体
     - 一个状态
  3. 随机博弈（又称马尔可夫博弈）
     - 多个智能体
     - 多个状态
- 当智能体能看到整个环境的状态时，多智能体强化学习任务属于随机博弈



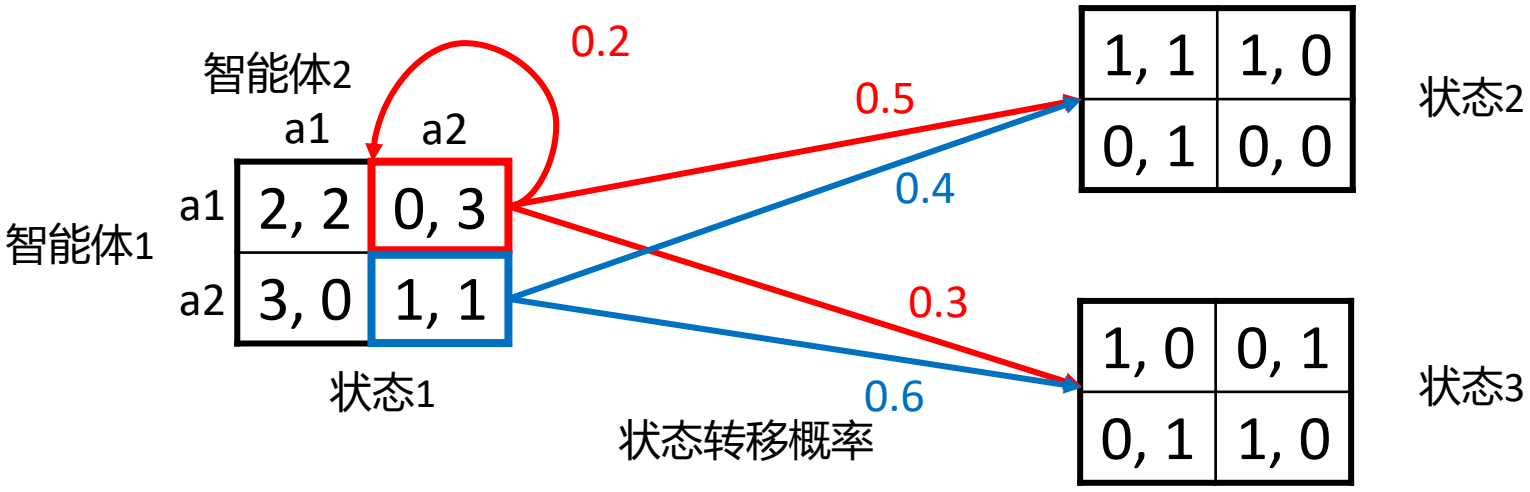
随机博弈

- 一个随机博弈（又称马尔可夫博弈）包含多个多个智能体和多个状态
  - 在表格型强化学习的设置下，每个状态对应一个博弈表格
  - 每一个时间步，所有智能体在当前博弈表格中同时选择自己的动作
  - 根据智能体共同决定的博弈表格单元分配给智能体奖励
  - 博弈根据状态和联合动作转移到下一个状态，对应另一个博弈表格
  - 一般情况下，奖励值随时间步衰减



随机博弈问题定义

- 一个随机博弈可以被定义为元组$(N, \mathcal{S}, \mathcal{A}, \mathcal{R}, p, \gamma)$
  - N是智能体的数目
  - $\mathcal{S}$ 是所有智能体的状态集合
  - $\mathcal{A} = A_1 \times A_2 \times \cdots \times A_N$ 是所有智能体的动作集合
  - $\mathcal{R} = r_1 \times r_2 \times \cdots \times r_N$ 是所有智能体的奖励函数集合
  - $p: \mathcal{S} \times \mathcal{A} \to \Omega(\mathcal{S})$ 是环境转移的概率，其中$\Omega(\mathcal{S})$在$\mathcal{S}$的分布集合
  - $\gamma \in (0,1)$ 是奖励随时间步的衰减因子



随机博弈问题定义

- 对于一个随机博弈中的每一个智能体 i，其策略表示为

$$\pi_i: \mathcal{S} \rightarrow \Omega(A_i)$$ ← A_i上的概率分布集合

- 所有智能体的联合策略可表示为 $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_N]$
- 所有智能体的联合动作可表示为 $\boldsymbol{a} = [a_1, a_2, \ldots, a_N]$

- 基于联合策略$\boldsymbol{\pi}$，智能体 i 在状态 s 下的价值为

$$V_i^{\boldsymbol{\pi}}(s) = \sum_{t=0}^{\infty} \gamma^t \, \mathbb{E}_{\boldsymbol{\pi},p}[r_i^t | s_0 = s]$$

- 基于联合策略$\boldsymbol{\pi}$，智能体 i 在状态行动对 $(s, \boldsymbol{a})$ 下的价值为

$$Q_i^{\boldsymbol{\pi}}(s, \boldsymbol{a}) = r_i(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s' \sim p}[V_i^{\boldsymbol{\pi}}(s')]$$

多智能体强化学习的方法分类

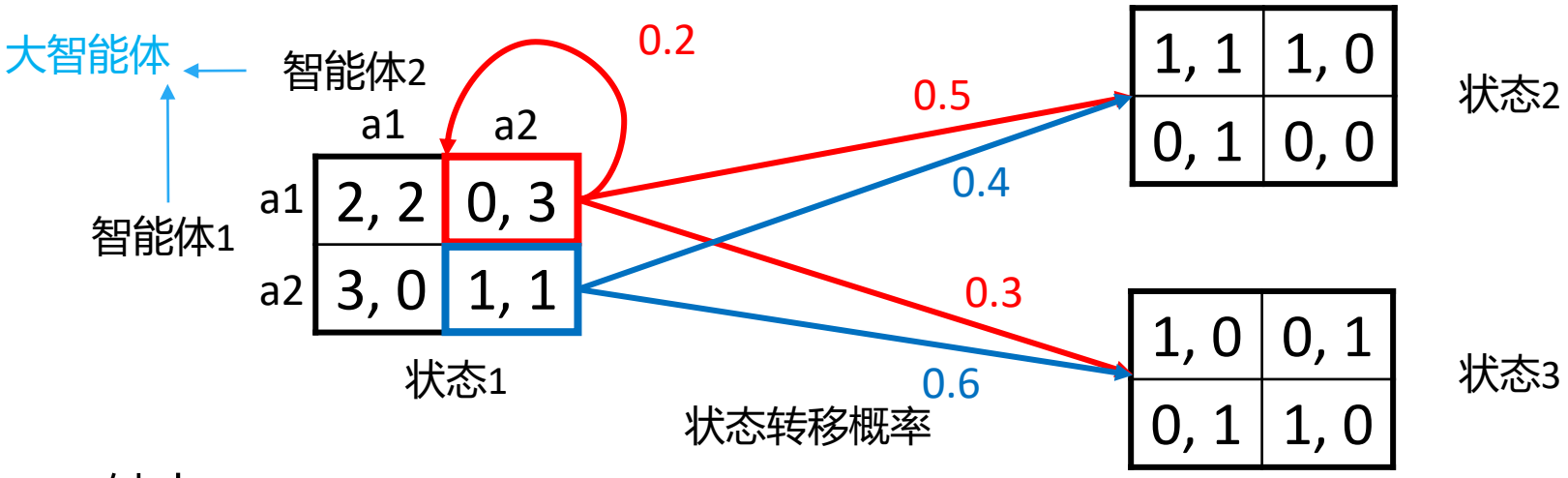
- 完全中心化方法（fully centralized）
  - 将多个智能体进行决策当作一个超级智能体在做决策，也即是把所有智能体的状态聚合在一起当作一个全局的超级状态，把所有智能体的动作连起来作为一个联合动作
  - 优点：环境是稳态；缺点：复杂度高
- 完全去中心化方法（fully decentralized）
  - 假设每个智能体都在自身的环境中独立地进行学习，不考虑其他智能体的改变。完全去中心化方法直接对每个智能体用一个单智能体强化学习算法来学习
  - 优点：简单好实现；缺点：环境非稳态，很可能不收敛
- 中心化训练去中心化执行 （centralized training with decentralized execution，即CTDE）
  - 在训练的时候使用一些单个智能体看不到的全局信息而达到更好的训练效果，而在执行时不使用这些信息，每个智能体完全根据自己的策略直接行动，以达到去中心化执行的效果
  - 特点：介于完全中心化方法和完全去中心化方法之间

02

完全中心化方法

完全中心化方法1：整体智能体

- 对于合作任务，可以将N个智能体看成一个大的智能体，每次在具体的状态s下，直接选择联合动作$\boldsymbol{a}$，以每个智能体获得的奖励之和作为此大智能体的奖励



- 缺点：
  1. 动作空间太大，以至于状态转移和奖励函数的复杂度太高，往往需要极大量的数据和算力
  2. 无法处理非合作的任务，例如对抗任务

完全中心化方法2：纳什Q学习

- 给定联合策略 $\boldsymbol{\pi}$，其智能体 i 的价值函数为

$$V_i^{\boldsymbol{\pi}}(s) = V_i(s;\boldsymbol{\pi}) = \sum_{t=0}^{\infty} \gamma^t \, \mathbb{E}_{\boldsymbol{\pi},p}[r_i^t | s_0 = s]$$

- 因此，针对智能体 i 优化 $V_i^{\boldsymbol{\pi}}(s)$ 依赖于联合策略 $\boldsymbol{\pi}$

- 在随机博弈中的纳什均衡（Nash Equilibrium）可以由一个联合策略 $\boldsymbol{\pi}^*$来表示

$$\boldsymbol{\pi}^* = [\pi_1^*, \pi_2^*, \ldots, \pi_N^*]$$

如果没有任何一个智能体有动机去进一步修改它的策略

$$V_i^{\boldsymbol{\pi}^*}(s) = V_i(s;\boldsymbol{\pi}^*) = V_i(s;\pi_i^*, \boldsymbol{\pi}_{-i}^*) \geq V_i(s;\pi_i, \boldsymbol{\pi}_{-i}^*) \text{ for } \forall \pi_i$$

$$\boldsymbol{\pi}_{-i}^* = [\pi_1^*, \pi_2^*, \ldots, \pi_{i-1}^*, \pi_{i+1}^*, \ldots, \pi_N^*]$$

Hu, Junling, and Michael P. Wellman. Nash Q-learning for general-sum stochastic games. JMLR 2003.

完全中心化方法2：纳什Q学习

- 给定纳什策略$\boldsymbol{\pi}^*$，其纳什价值函数为

$$V_{\text{Nash}}(s) = \left[V_1^{\boldsymbol{\pi}^*}(s), V_2^{\boldsymbol{\pi}^*}(s), \ldots, V_N^{\boldsymbol{\pi}^*}(s)\right]$$

- 考虑到智能体 i 在状态行动对 $(s, \boldsymbol{a})$ 下的价值为

$$Q_i^{\boldsymbol{\pi}}(s, \boldsymbol{a}) = r_i(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s' \sim p}[V_i^{\boldsymbol{\pi}}(s')]$$

- 纳什Q学习不算进行以下两部操作，直到收敛
  1. 基于当前每个状态的Q值表，求解纳什均衡$\boldsymbol{\pi}^*$，以及每个状态的V_{Nash}
  2. 基于纳什均衡价值V_{Nash}，使用上式更新Q值表

- 作为一种完全中心化方法，纳什Q学习仍然有以下缺点
  1. 非常高的计算复杂度
  2. 无法处理非合作的博弈场景

Hu, Junling, and Michael P. Wellman. Nash Q-learning for general-sum stochastic games. JMLR 2003.

完全中心化方法2：纳什Q学习

Initialize $Q(s, a)$ arbitrarily

Initialize s

loop

$a_i \leftarrow$ probabilistic outcome of Nash policy derived from $Q(s, a)$, for player i {Mixed with exploration policy}

Take action a_i, observe reward r, next state s' and the joint action of other players a_{-i}

for $i = 1 \ldots n$ **do**

$$Q_i(s, \langle a_i, a_{-i} \rangle) \leftarrow Q_i(s, \langle a_i, a_{-i} \rangle) + \alpha\big(r_i + \gamma V_i(s') - Q_i(s, \langle a_i, a_{-i} \rangle)\big)$$

end for

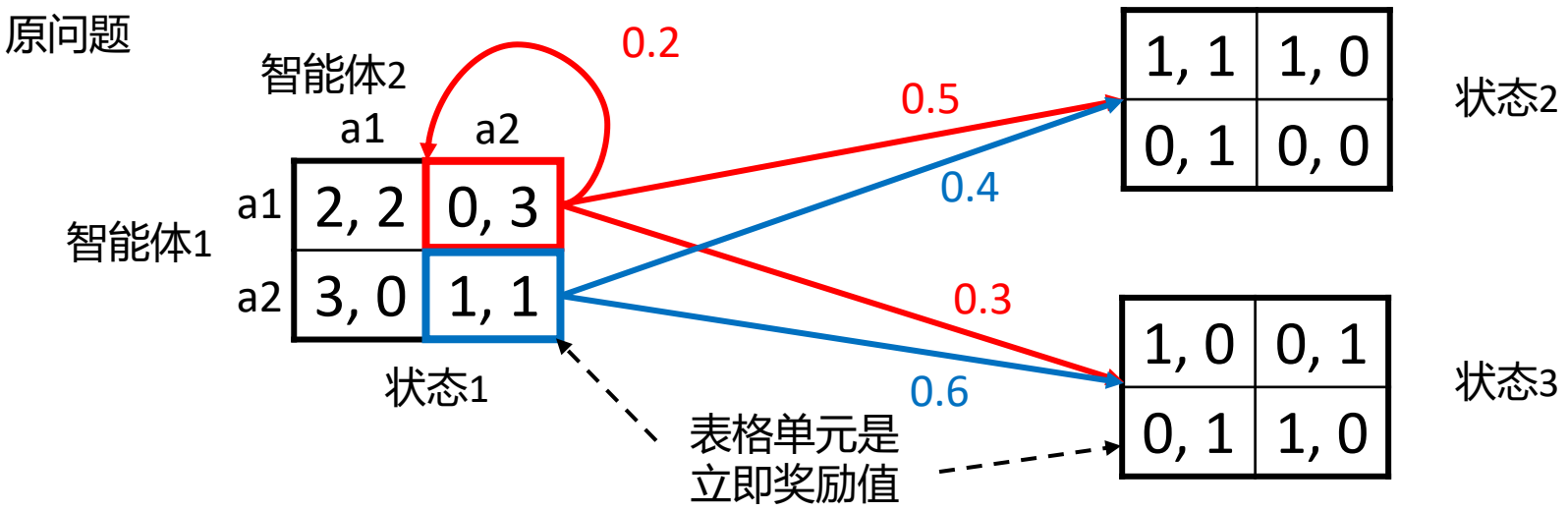
where $V(s) = Nash([Q(s, a)])$ 代替Q学习中的 $\max_a$ 操作

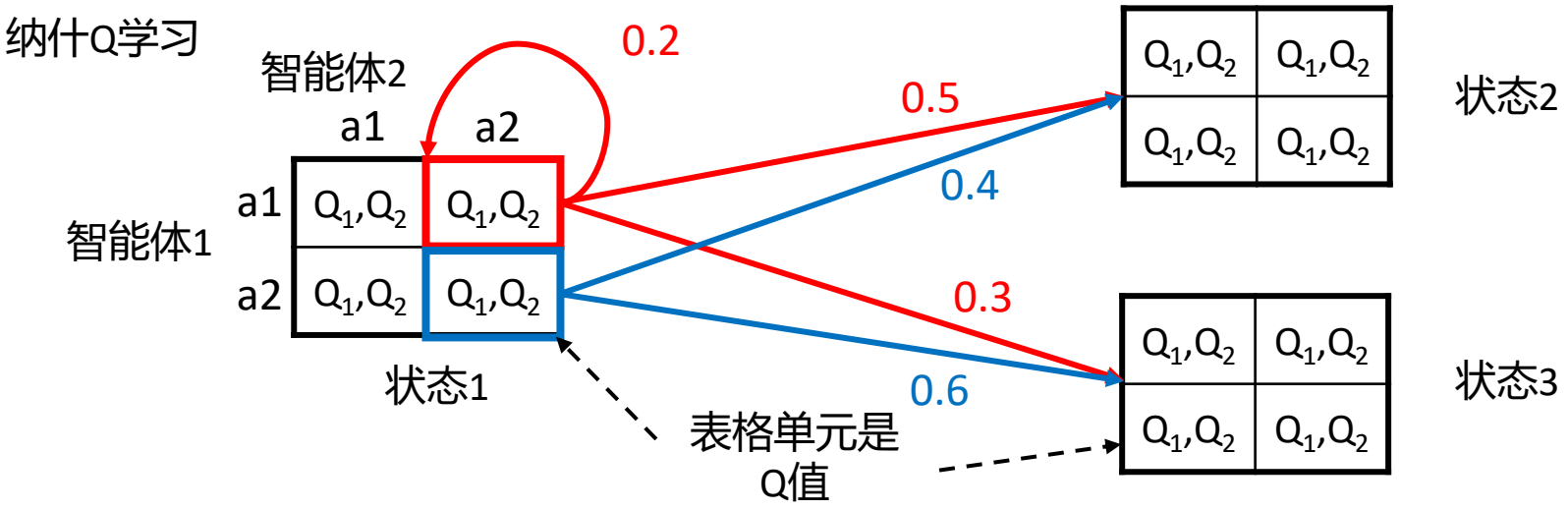
$s \leftarrow s'$

end loop

Hu, Junling, and Michael P. Wellman. Nash Q-learning for general-sum stochastic games. JMLR 2003.

完全中心化方法2：纳什Q学习





03

完全去中心化方法

完全去中心化方法（fully decentralized）

- 假设每个智能体都在自身的环境中独立地进行学习，不考虑其他智能体的改变。完全去中心化方法直接对每个智能体用一个单智能体强化学习算法来学习
  - 优点：简单好实现
  - 缺点：环境非稳态，很可能不收敛

- 一个随机博弈可以被定义为元组$(N, \mathcal{S}, \mathcal{A}, \mathcal{R}, p, \gamma)$
  - N是智能体的数目
  - $\mathcal{S} = S_1 \times S_2 \times \cdots \times S_N$ 是所有智能体的状态集合
  - $\mathcal{A} = A_1 \times A_2 \times \cdots \times A_N$ 是所有智能体的动作集合
  - $\mathcal{R} = r_1 \times r_2 \times \cdots \times r_N$ 是所有智能体的奖励函数集合
  - $p: \mathcal{S} \times \mathcal{A} \to \Omega(\mathcal{S})$ 是环境转移的概率，其中$\Omega(\mathcal{S})$在$\mathcal{S}$的分布集合
  - $\gamma \in (0,1)$ 是奖励随时间步的衰减因子

完全去中心化方法可以让每个智能体的局部感知状态不同

完全去中心化方法（fully decentralized）

- 假设每个智能体都在自身的环境中独立地进行学习，不考虑其他智能体的改变。完全去中心化方法直接对每个智能体用一个单智能体强化学习算法来学习
  - 优点：简单好实现
  - 缺点：环境非稳态，很可能不收敛

- 独立Q学习
  - 对智能体i，建设其他智能体策略是不变（但未知）的，直接做Q学习

$$Q_i(s,a_i) \leftarrow Q_i(s,a_i) + \alpha(r(s,a_i,\boldsymbol{a}_{-i}) + \gamma \max_{a_i'} Q_i(s',a_i') - Q_i(s,a_i))$$

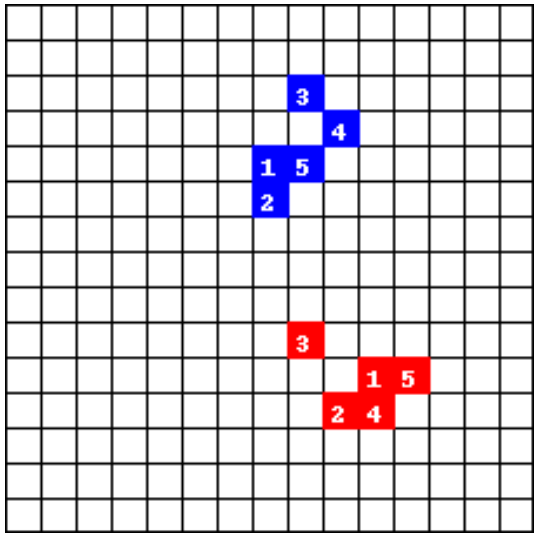
环境给出的奖励信号实际基于联合动作

- 独立PPO
  - 对智能体i，建设其他智能体策略是不变（但未知）的，直接做PPO学习

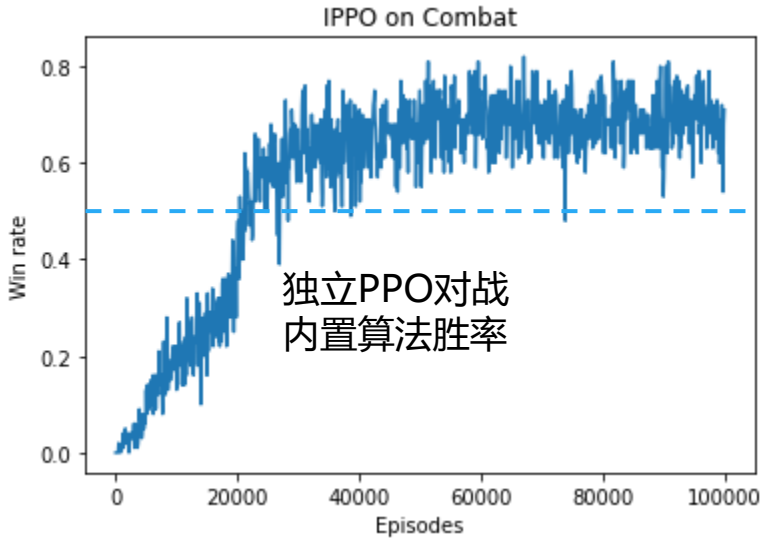
独立PPO实验

- 独立PPO（参数共享版）
  - 对智能体i，建设其他智能体策略是不变（但未知）的，直接做PPO学习
  - 团队所有智能体共享一套PPO参数

- 实验环境Combat
  - 二维的格子世界上进行两个队伍的对战模拟游戏



- 实验效果
  - 独立PPO算法对战内置算法
  - 攻击在范围内最近的敌人，如果攻击范围内没有敌人，则向敌人靠近



多智能体强化学习进阶

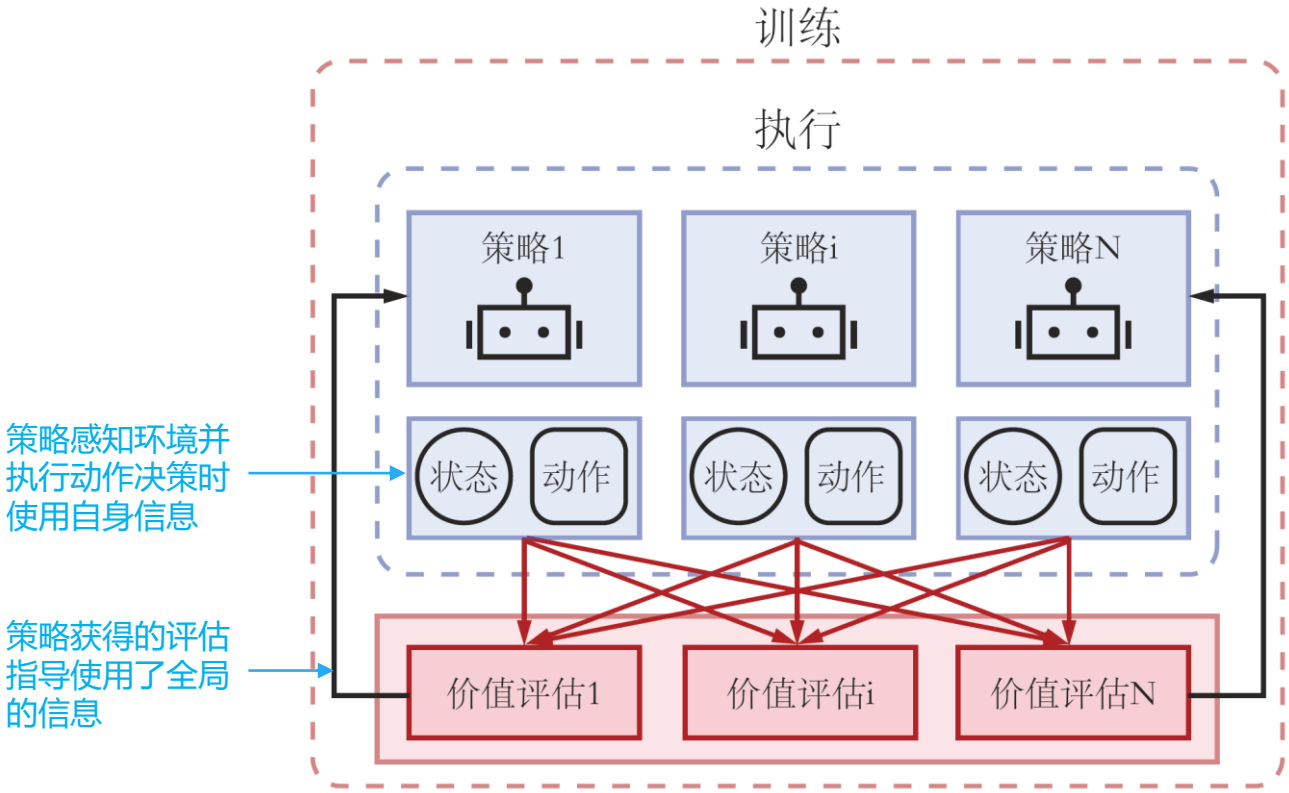
张伟楠 – 上海交通大学

多智能体强化学习的方法分类

- 完全中心化方法（fully centralized）
  - 将多个智能体进行决策当作一个超级智能体在做决策，也即是把所有智能体的状态聚合在一起当作一个全局的超级状态，把所有智能体的动作连起来作为一个联合动作
  - 优点：环境是稳态；缺点：复杂度高
- 完全去中心化方法（fully decentralized）
  - 假设每个智能体都在自身的环境中独立地进行学习，不考虑其他智能体的改变。完全去中心化方法直接对每个智能体用一个单智能体强化学习算法来学习
  - 优点：简单好实现；缺点：环境非稳态，很可能不收敛
- 中心化训练去中心化执行 （centralized training with decentralized execution，即CTDE）
  - 在训练的时候使用一些单个智能体看不到的全局信息而达到更好的训练效果，而在执行时不使用这些信息，每个智能体完全根据自己的策略直接行动，以达到去中心化执行的效果
  - 特点：介于完全中心化方法和完全去中心化方法之间

中心化训练去中心化执行（CTDE）

- 中心化训练去中心化执行（Centralized Training with Decentralized Execution）
  - 指在训练的时候使用一些单个智能体看不到的全局信息而达到更好的训练效果，而在执行时不使用这些信息，每个智能体完全根据自己的策略直接行动，以达到去中心化执行的效果。



CTDE问题建模

- CTDE的问题通常被建模为一个部分可观测马尔可夫博弈（partially observable Markov games）元组
  - N是智能体的数目
  - $\mathcal{S}$ 是所有智能体的全局状态集合
  - $\mathcal{A}$ 是所有智能体的联合动作集合
  - $p: \mathcal{S} \times \mathcal{A} \to \Omega(\mathcal{S})$ 是环境转移的概率，其中$\Omega(\mathcal{S})$在$\mathcal{S}$的分布集合
  - $\gamma \in (0,1)$ 是奖励随时间步的衰减因子
  - 对于每个智能体 i
    - O_i 是观测集合
    - A_i 是动作集合
    - $\pi_i: O_i \to \Omega(A_i)$
    - $r_i: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ 是奖励函数， r_i^t表示在t时间步获得的具体奖励信号
    - 目标是最大化其期望累积奖励

$$R_i = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_i^t\right]$$

CTDE算法：MADDPG

- MADDPG (Multi-Agent Deep Deterministic Policy Gradient)
  - 每个智能体 i 是一个actor-critic
  - Actor策略$\mu_i(a_i|o_i)$是自己独有的，只看到局部观测
  - Critic价值函数 $Q_i^{\mu}(\mathbf{x}, a_1, \ldots, a_N)$则是所有智能体贡献的，具有全局信息
  - $\mathbf{x} = (o_1, o_2, \ldots, o_N)$为所有智能体的观测
- 对于确定性策略来说，考虑N个连续策略μ_{θ_i}，其DDPG梯度公式为

$$\nabla_{\theta_i} J(\mu_i) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[\nabla_{\theta_i}\mu_i(o_i)\nabla_{a_i} Q_i^{\mu}(\mathbf{x}, a_1, \ldots, a_N)|_{a_i=\mu_i(o_i)}]$$

  - 其中$\mathcal{D}$为经验回放池，里面存储的数据单元为
$$(\mathbf{x}, \mathbf{x}', a_1, \ldots, a_N, r_1, \ldots, r_N)$$
- 对于中心化的critic价值函数$Q_{\omega_i}^{\mu}(\mathbf{x}, a_1, \ldots, a_N)$则通过时序差分来学习

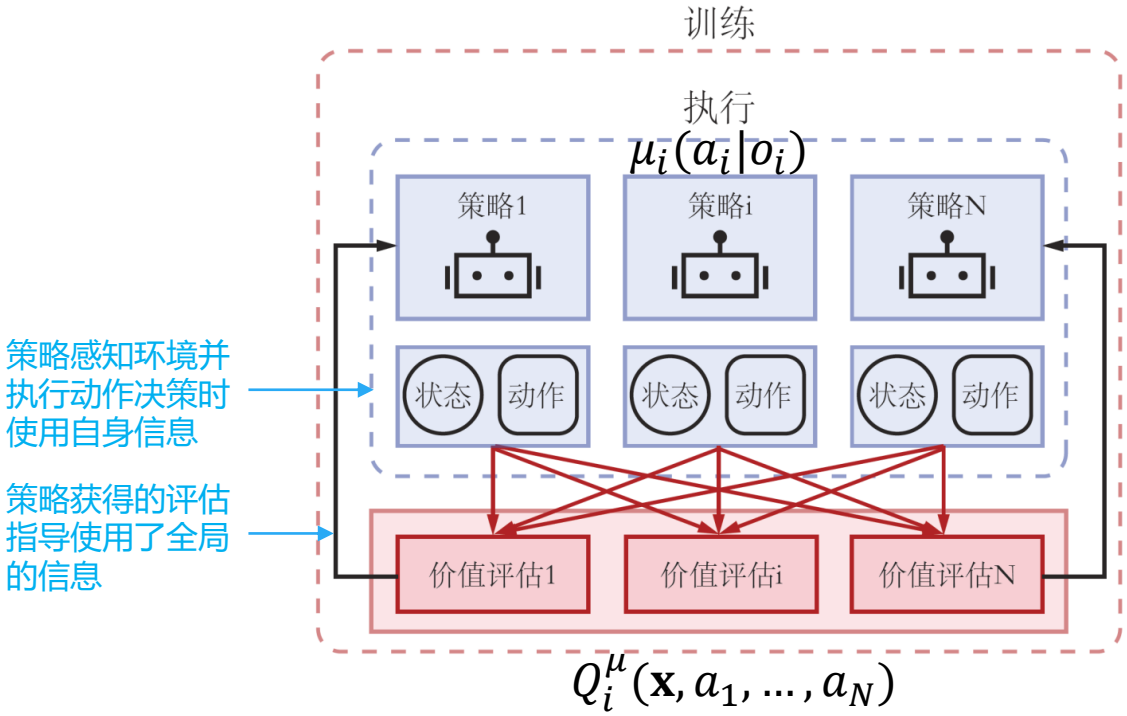
$$\mathcal{L}(\omega_i) = \mathbb{E}_{\mathbf{x},a,r,\mathbf{x}'}[(Q_i^{\mu}(\mathbf{x}, a_1, \ldots, a_N) - y)^2], \quad y = r_i + \gamma Q_i^{\mu'}(\mathbf{x}', a_1', \ldots, a_N')|_{a_j'=\mu_j'(o_j)}$$

延迟更新的critic价值函数

Lowe, Ryan, et al. Multi-agent actor-critic for mixed cooperative-competitive environments. NIPS 2017.

CTDE算法：MADDPG

- MADDPG (Multi-Agent Deep Deterministic Policy Gradient)
  - 每个智能体 i 是一个actor-critic
  - Actor策略$\mu_i(a_i|o_i)$是自己独有的，只看到局部观测
  - Critic价值函数 $Q_i^{\mu}(\mathbf{x}, a_1, \ldots, a_N)$则是所有智能体贡献的，具有全局信息
  - $\mathbf{x} = (o_1, o_2, \ldots, o_N)$为所有智能体的观测



CTDE算法：MADDPG

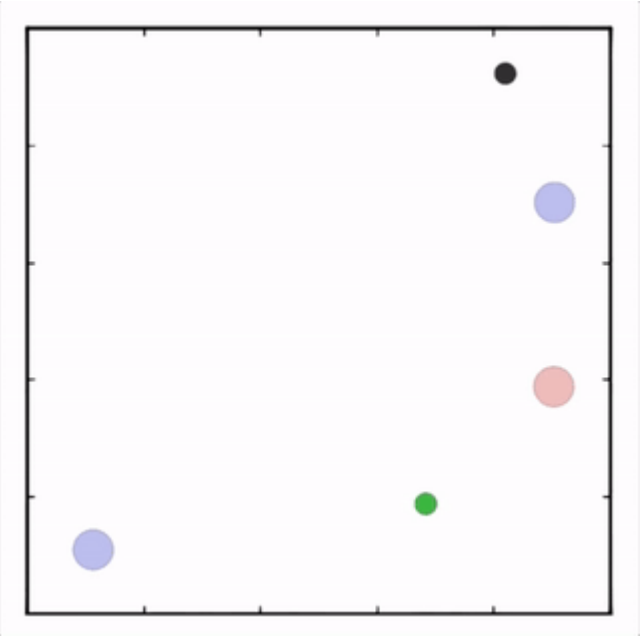
- MADDPG的算法流程如下：
  - 随机初始化每个智能体的Actor网络和Critic网络
  - **for** 序列 $e = 1 \to E$ **do**
  - 初始化一个随机过程N用于动作探索；
  - 获取所有智能体的初始观测 $\mathbf{x}$；
    - **for** $t = 1 \to T$ **do**：
      - 对于每个智能体 i，用当前的策略选择一个动作$a_i = \mu_{\theta_i}(o_i) + \mathcal{N}_t$；
      - 执行动作$a = (a_1, \ldots, a_N)$ 并且获得奖励r和新的观测$\mathbf{x}'$；
      - 把$(\mathbf{x}, a, r, \mathbf{x}')$ 存到经验回放池$\mathcal{D}$中；
      - 从$\mathcal{D}$中随机采样一些数据;
      - 对于每个智能体i，中心化训练Critic网络
      - 对于每个智能体i，训练自身的Actor网络
      - 对于每个智能体i，更新目标Actor网络和目标Critic网络
    - **end for**
  - **end for**

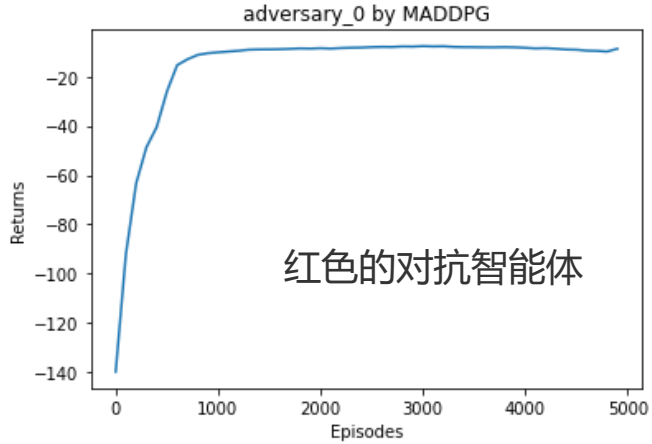
Lowe, Ryan, et al. "Multi-agent actor-critic for mixed cooperative-competitive environments." In NIPS 2017.

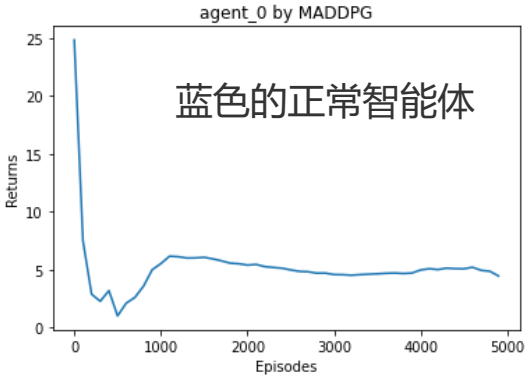
CTDE算法：MADDPG实验

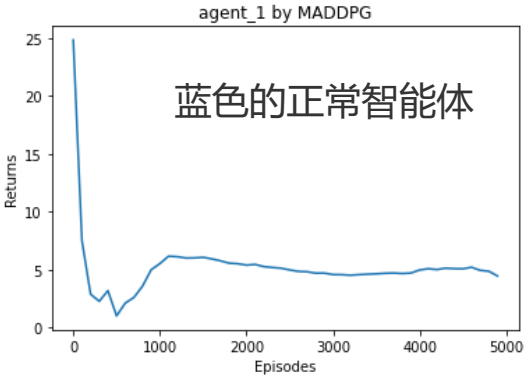
- OpenAI Multi-Agent Particle Environment (MPE) 环境

MPE对抗环境







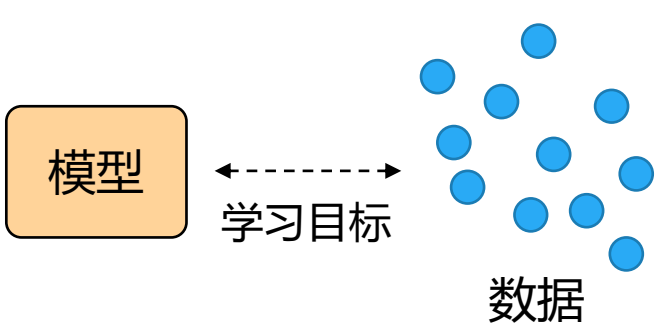


N个正常智能体知道哪一个是目标地点，但对抗智能体则不知道。正常智能体需要合作起来，分散去到不同的坐标点，这样来欺骗到对抗智能体。

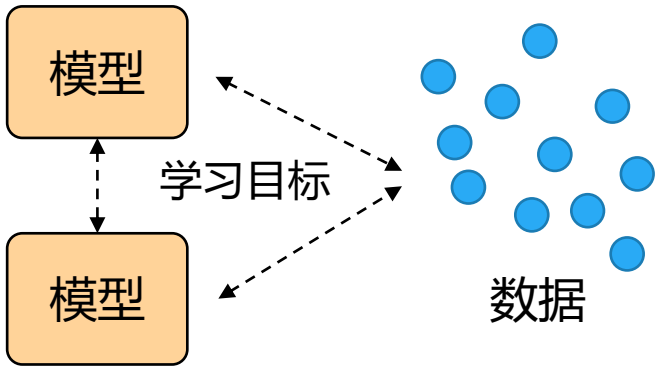
https://github.com/openai/multiagent-particle-envs

总结多智能体学习

- 多智能体学习强化学习（MARL）从原理上来讲更加困难
  - 智能体不仅要与环境进行交互，还要相互之间进行交互



单智能体学习



多智能体学习

- 多智能体强化学习的范式
  - 完全中心化（Nash Q-learning）、完全去中心化（IPPO）
  - 中心化训练去中心化执行（MADDPG）

THANK YOU