



南京大學
NANJING UNIVERSITY

人工智能导论

强化学习 (reinforcement learning)

郭兰哲

南京大学 智能科学与技术学院

Homepage: www.lamda.nju.edu.cn/guolz

Email: guolz@nju.edu.cn

大纲

□ 强化学习简介

□ 多臂老虎机

□ 有模型学习

□ 无模型学习

□ 模仿学习

□ 深度强化学习

强化学习

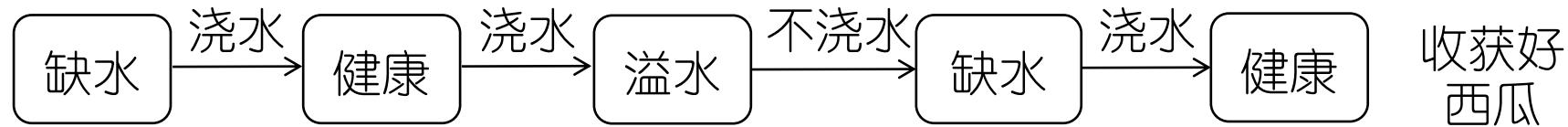


Atari (雅达利游戏)



例子：瓜农种西瓜

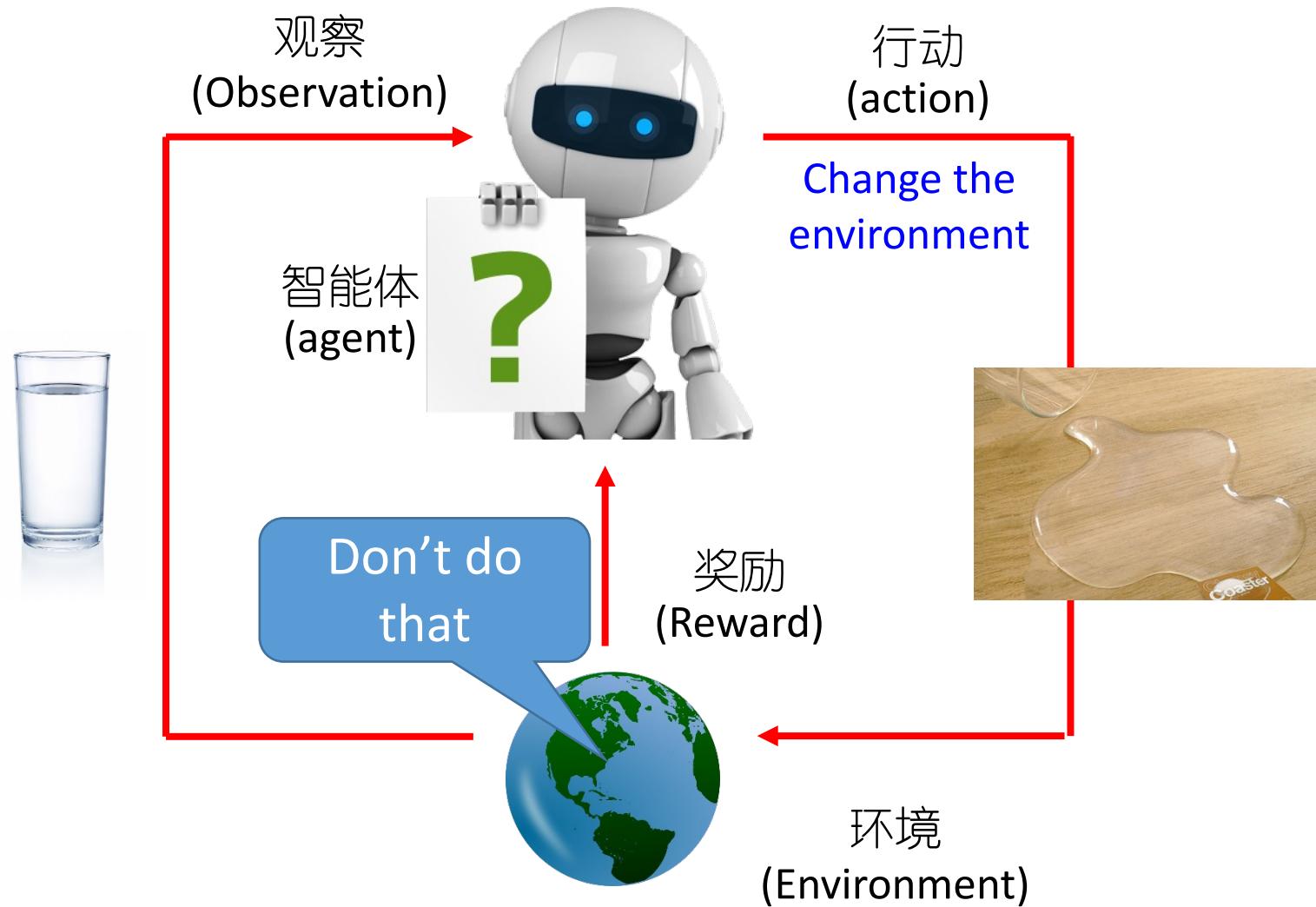
种下瓜苗后：(为简便，仅考虑浇水和不浇水两个动作，不考虑施肥、除草等)



- 多步决策过程
- 过程中包含状态、动作、反馈(奖赏)等
- 需多次种瓜，在过程中不断摸索，才能总结出较好的种瓜策略

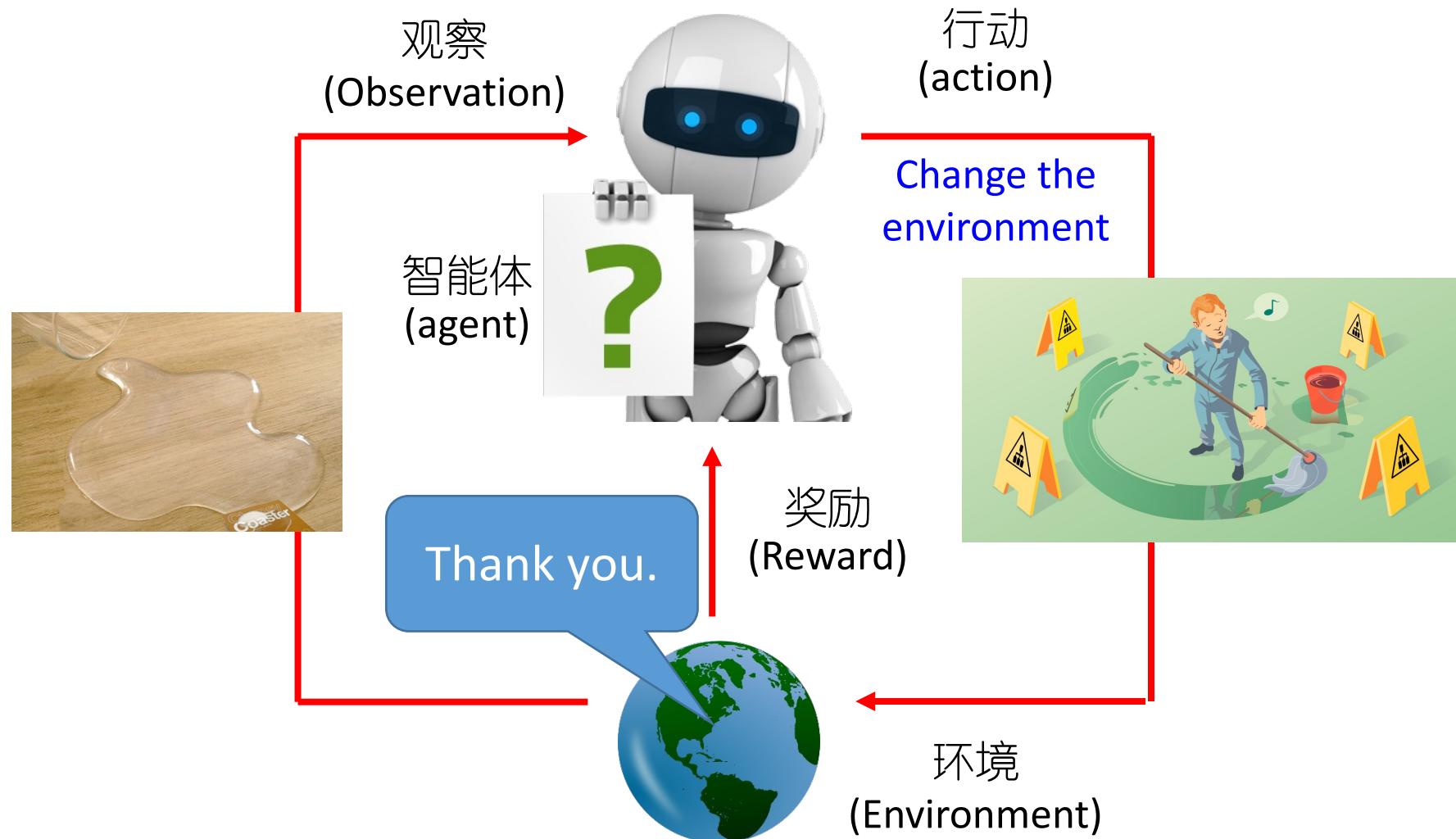
抽象该过程：**强化学习(reinforcement learning)**

强化学习



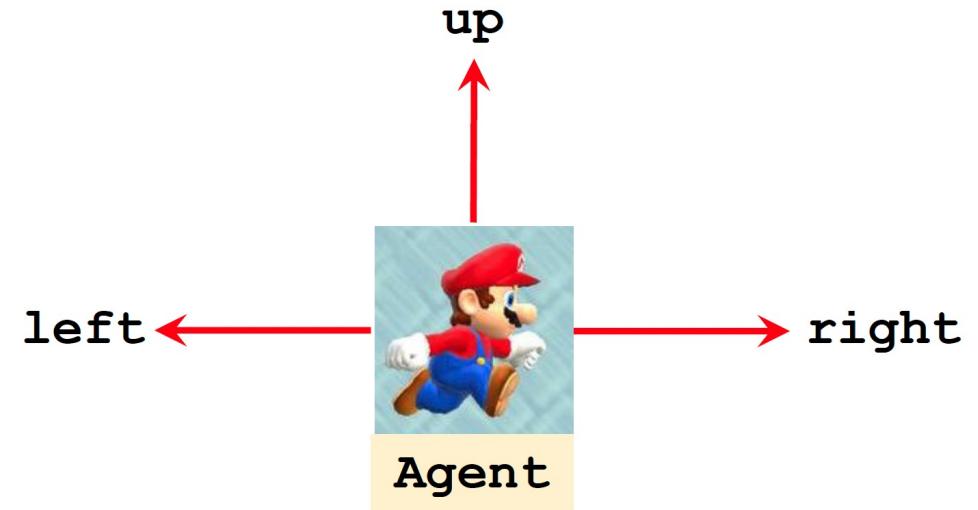
强化学习

智能体从交互中学习得到最大化累计奖励的行动



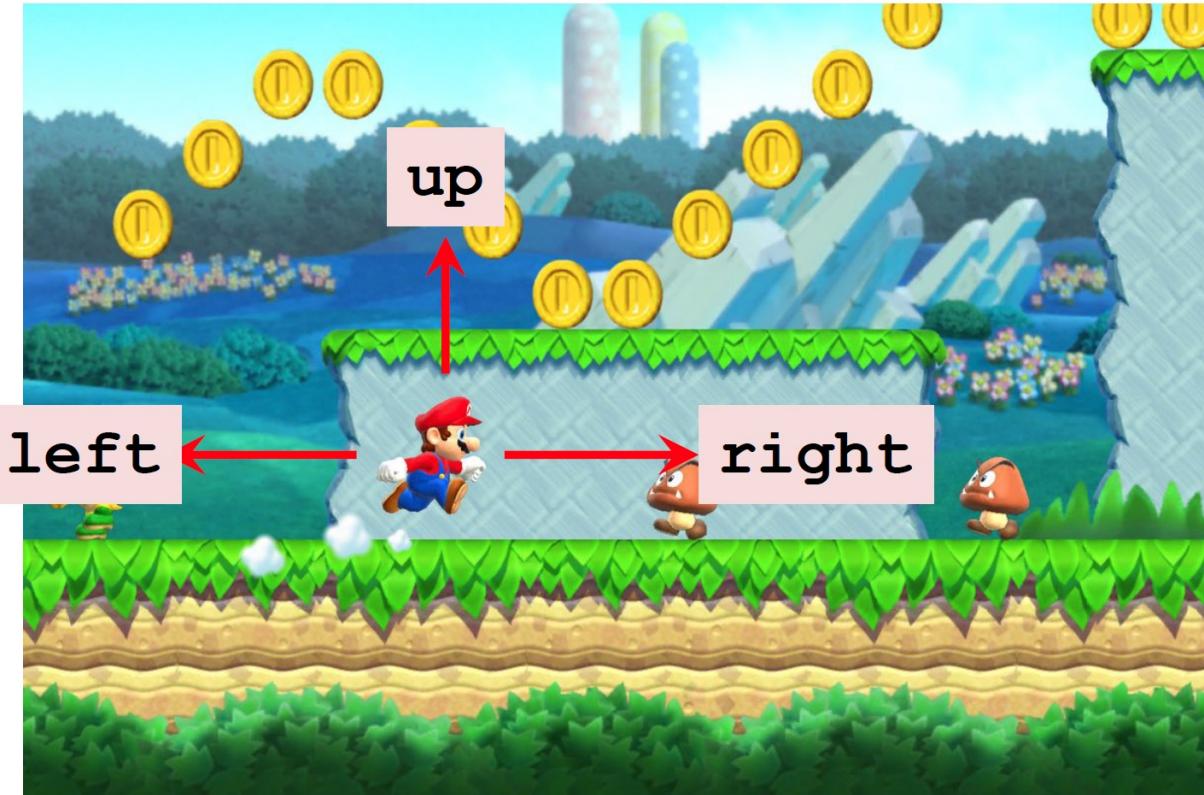
强化学习的关键要素

- 状态(State) : $s \in S$ 是对当前环境的描述
 - 种西瓜任务中，西瓜长势的描述
 - 围棋任务中，当前的局面
 - 超级马里奥中，当前的游戏画面
 - ...
- 动作(Action): $a \in A$ 对智能体采取的行动
 - 种西瓜任务中，浇水、施肥等
 - 围棋任务中，选择一个位置落子
 - 超级马里奥中，向左、向右、向上
 - ...



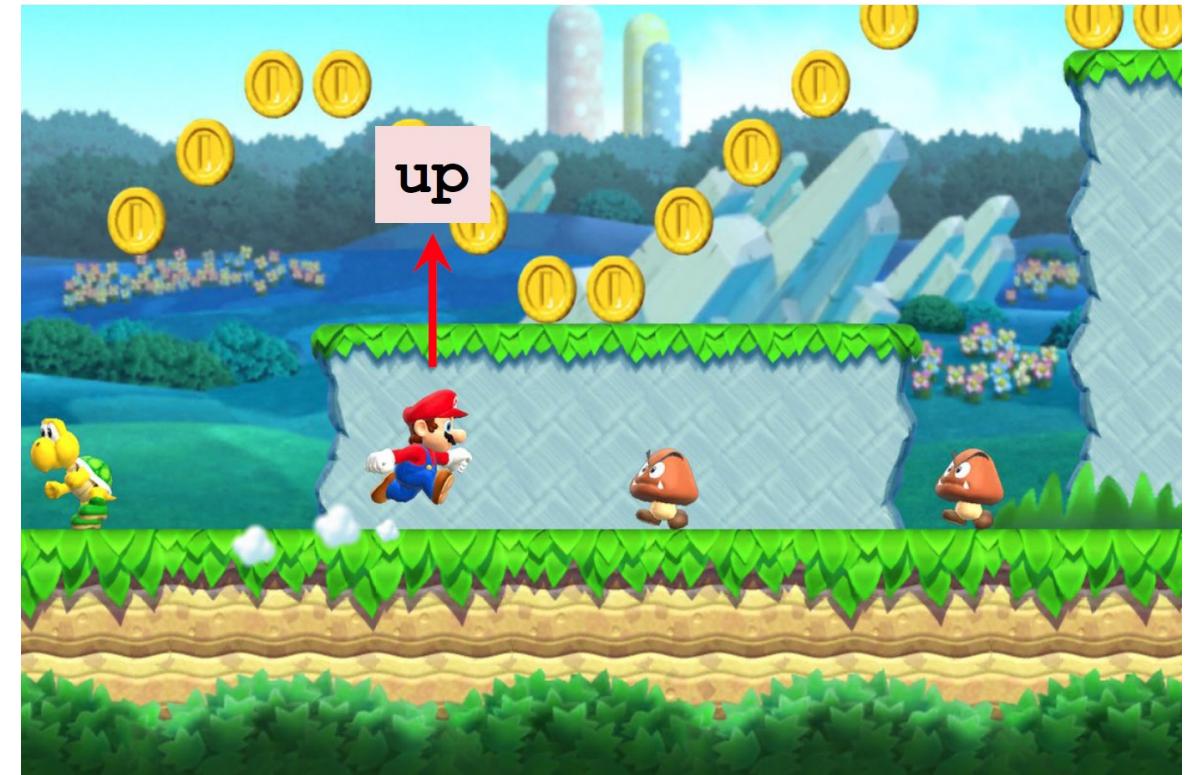
强化学习的关键要素

- 策略(Policy) π ：智能体如何根据观察到的状态做决策，即从动作空间中选择一个动作
- 强化学习最终学得的就是一个策略函数
- 确定性策略： $\pi: S \rightarrow A$
- 随机性策略： $\pi: S \times A \rightarrow \mathbb{R}$



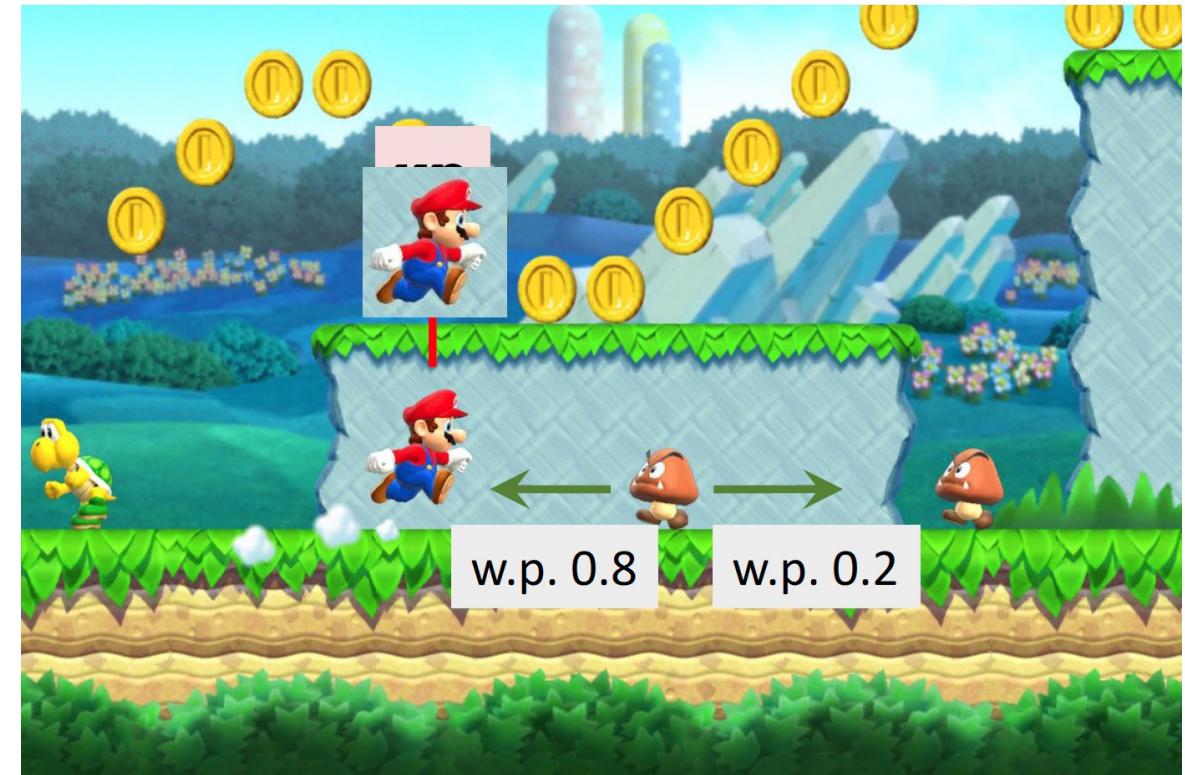
强化学习的关键要素

- 状态转移(state transition) $P: S \times A \times S \rightarrow \mathbb{R}$
- 智能体从当前 t 时刻状态 s_t 转移到下一个时刻的状态 s_{t+1}



强化学习的关键要素

- 状态转移(state transition) $P: S \times A \times S \rightarrow \mathbb{R}$
- 智能体从当前 t 时刻状态 s_t 转移到下一个时刻的状态 s_{t+1}



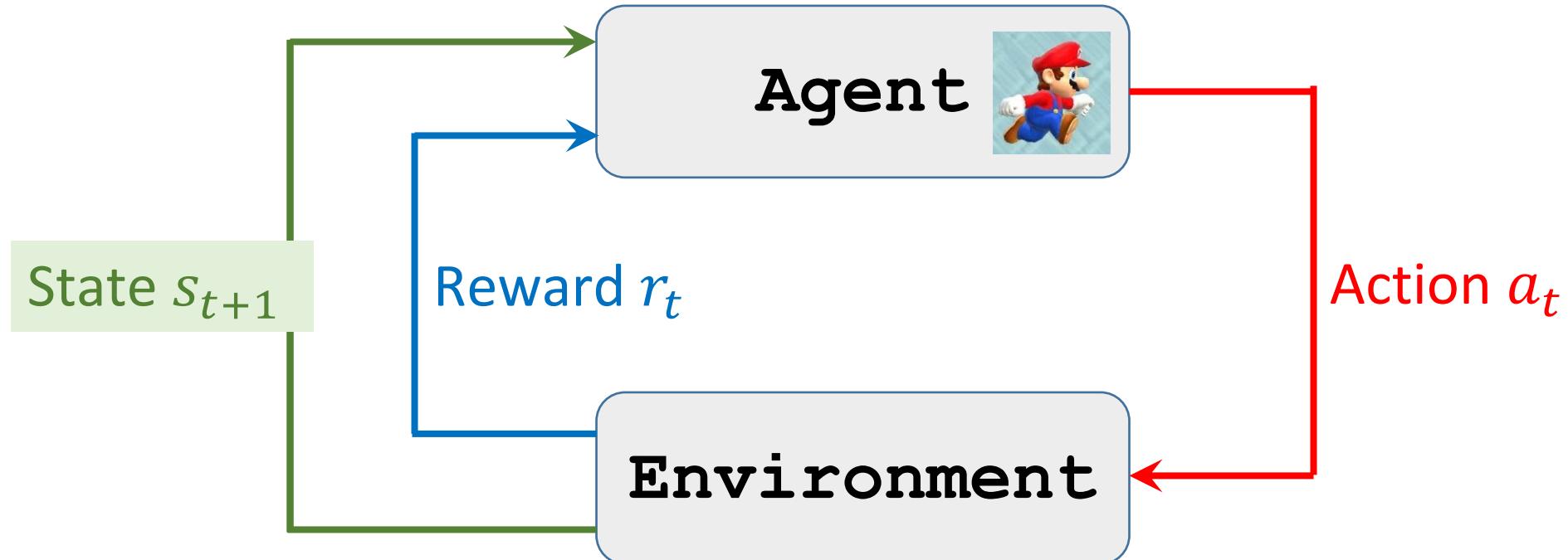
强化学习的关键要素

- 奖励(Reward) $R: S \times A \times S \mapsto \mathbb{R}$ (或 $R: S \times S \rightarrow \mathbb{R}$)
- 智能体执行一个动作之后，环境返回给智能体一个数值
- 通常需要人来定义
- 奖励函数的好坏影响强化学习的结果



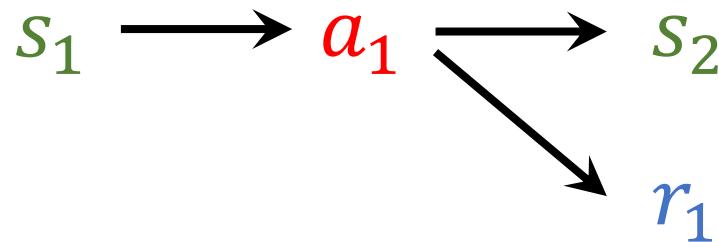
- 收集硬币： $R=+1$
- 被板栗仔撞到： $R=-1000$
- 通关： $R=+1000$
- 无事发生： $R=0$

强化学习的交互过程



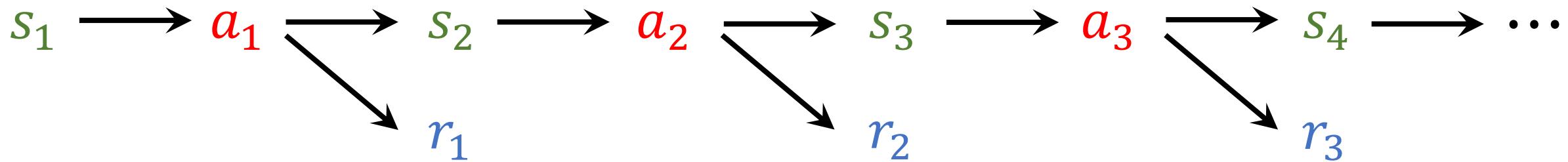
强化学习的交互过程

- 观察到状态 s_t , 根据策略 $\pi(s_t)$ 选择动作 a_t
- 执行动作 a_t , 状态转移至 s_{t+1} , 获得奖励 r_t



强化学习的交互过程

- 观察到状态 s_t , 根据策略 $\pi(s_t)$ 选择动作 a_t
- 执行动作 a_t , 状态转移至 s_{t+1} , 获得奖励 r_t



- 轨迹(trajectory): 智能体观测到的所有状态、动作、奖励的序列

$s_1, a_1, r_1, s_2, a_2, r_2, \dots, s_n, a_n, r_n$

什么是好的策略

- 回报(return)：从当前时刻开始到本回合结束的所有奖励总和

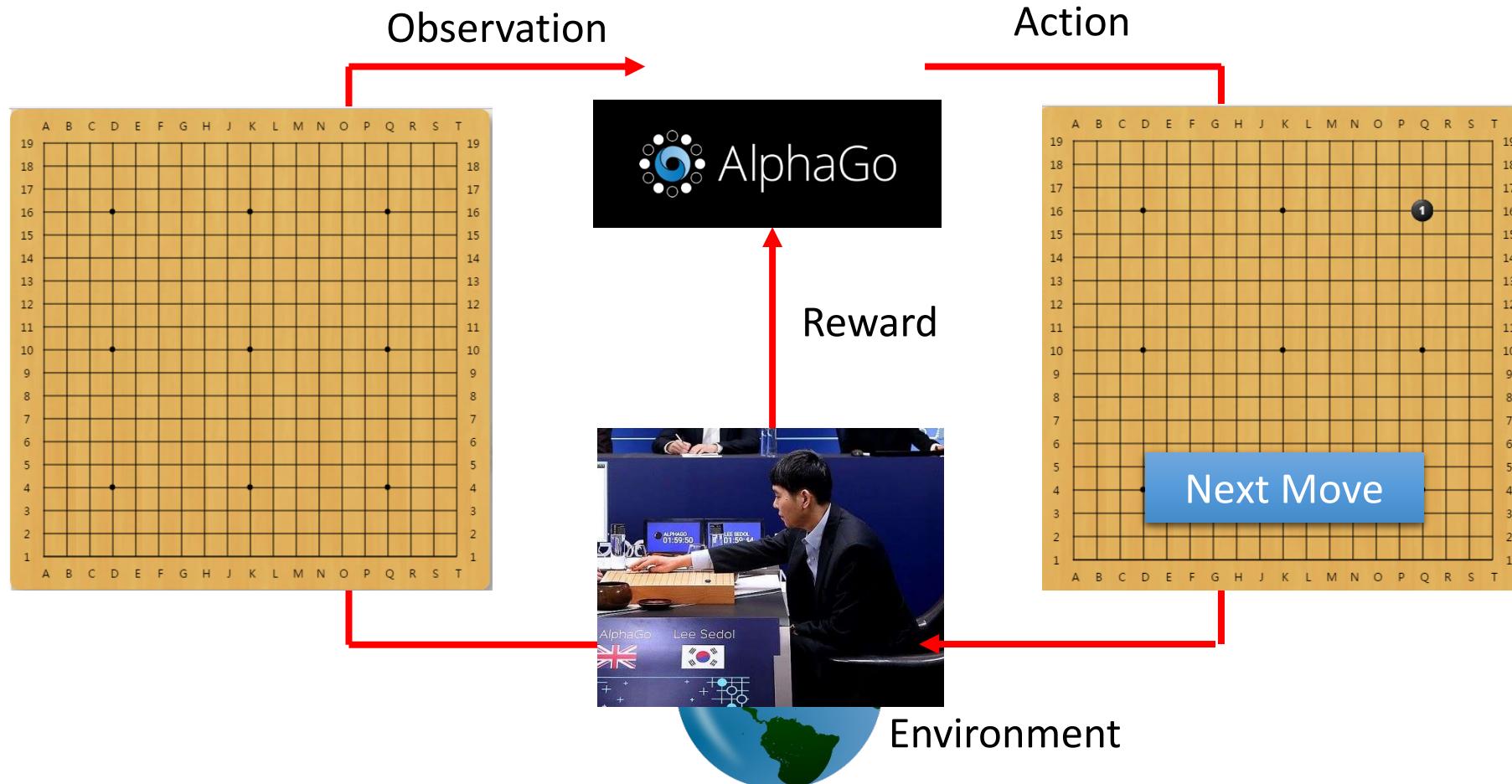
$$G_t = R_t + R_{t+1} + R_{t+2} + R_{t+3} + \dots$$

- 强化学习的目标是最大化回报，即累计奖励，而不是最大化当前奖励（好比下棋的时候目标是赢得比赛，而不是吃掉对方的棋子）
- 通常考虑折扣汇报： $\gamma \in [0,1]$ 是未来奖励的折扣因子，使得和未来奖励相比起来智能体更重视即时奖励

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

- 以金融为例，今天的\$1比明天的\$1更有价值

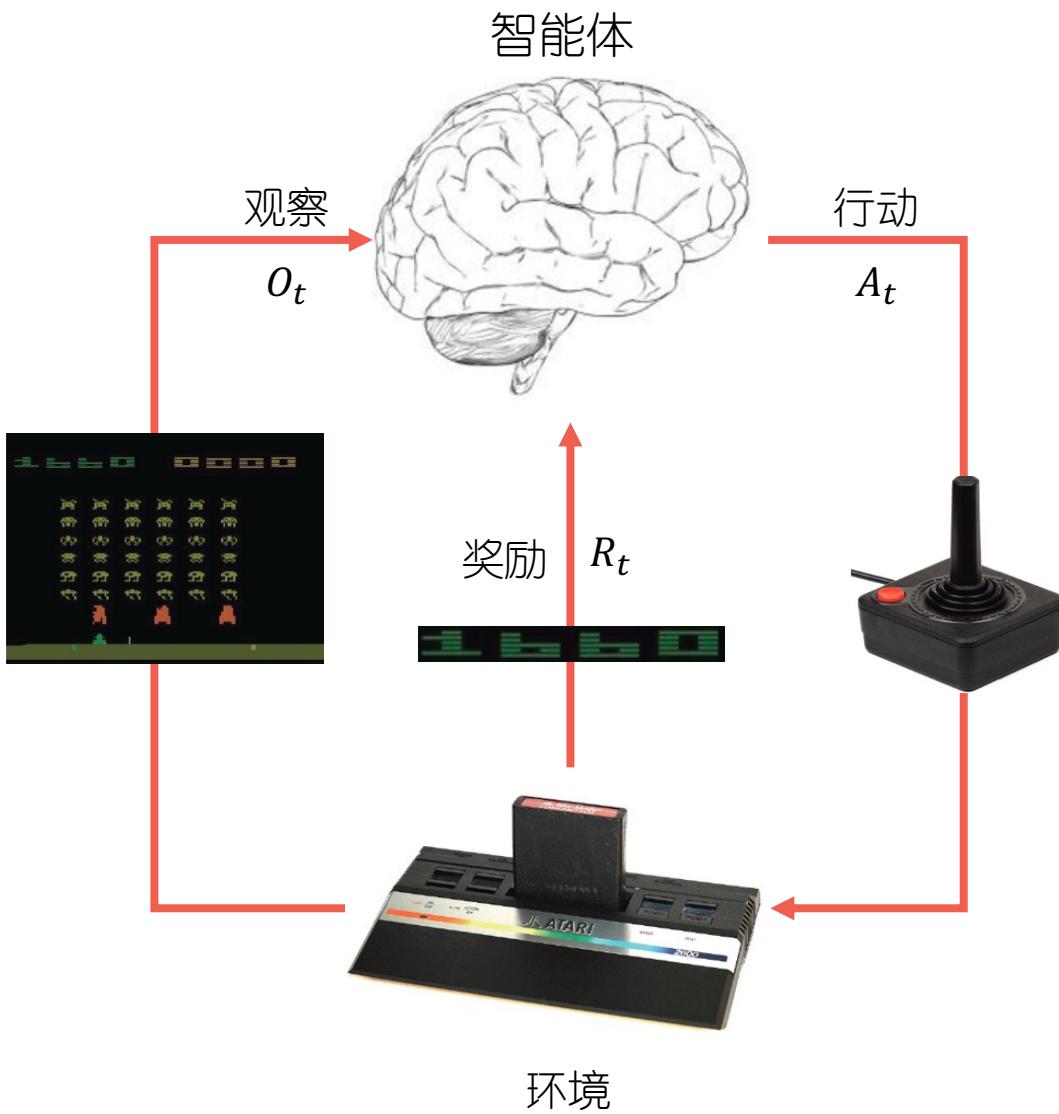
强化学习-围棋



强化学习-围棋



强化学习-游戏

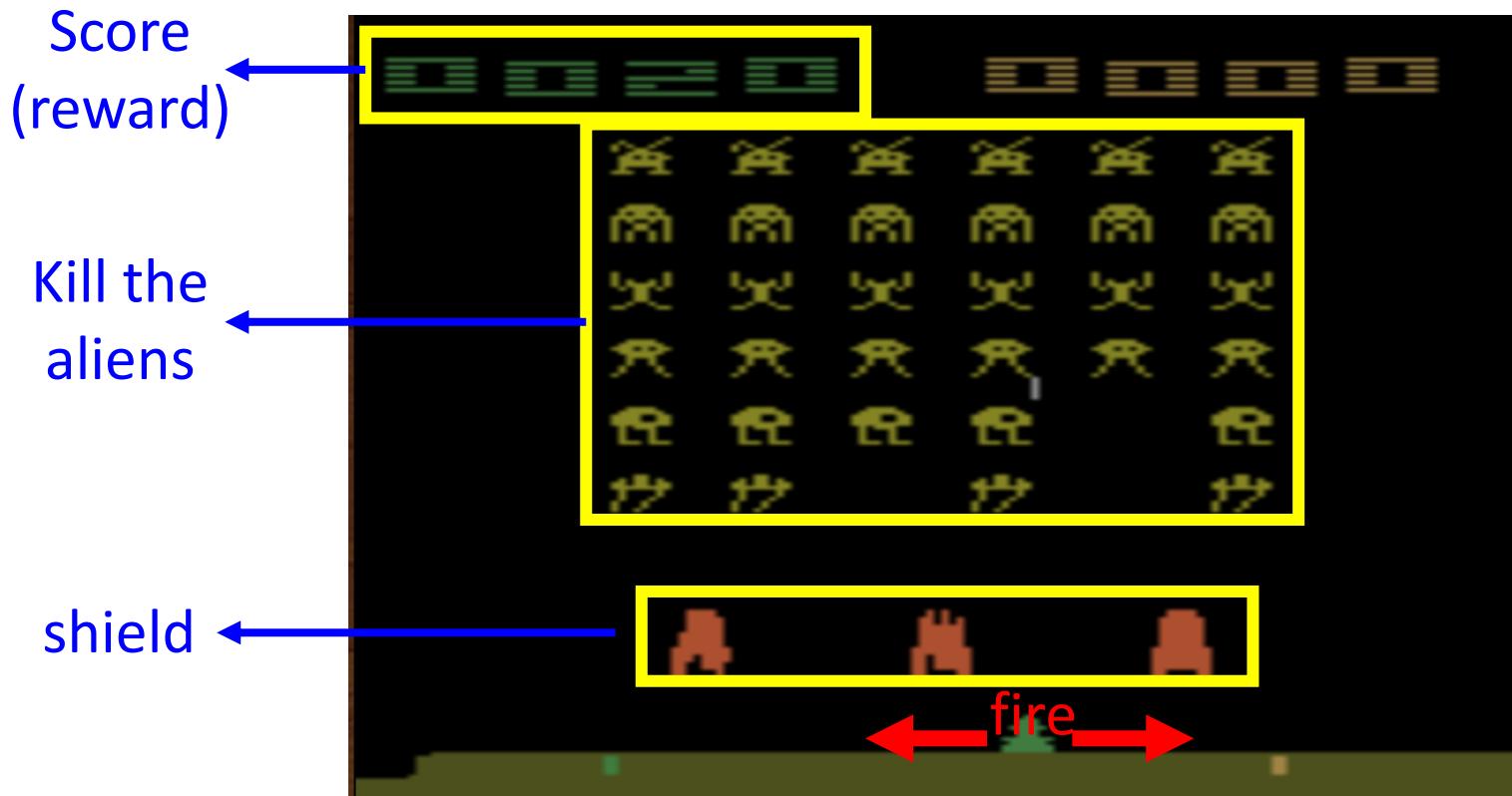


- 游戏规则未知
- 从交互游戏中进行学习
- 在操纵杆上选择行动并查看分数和像素画面

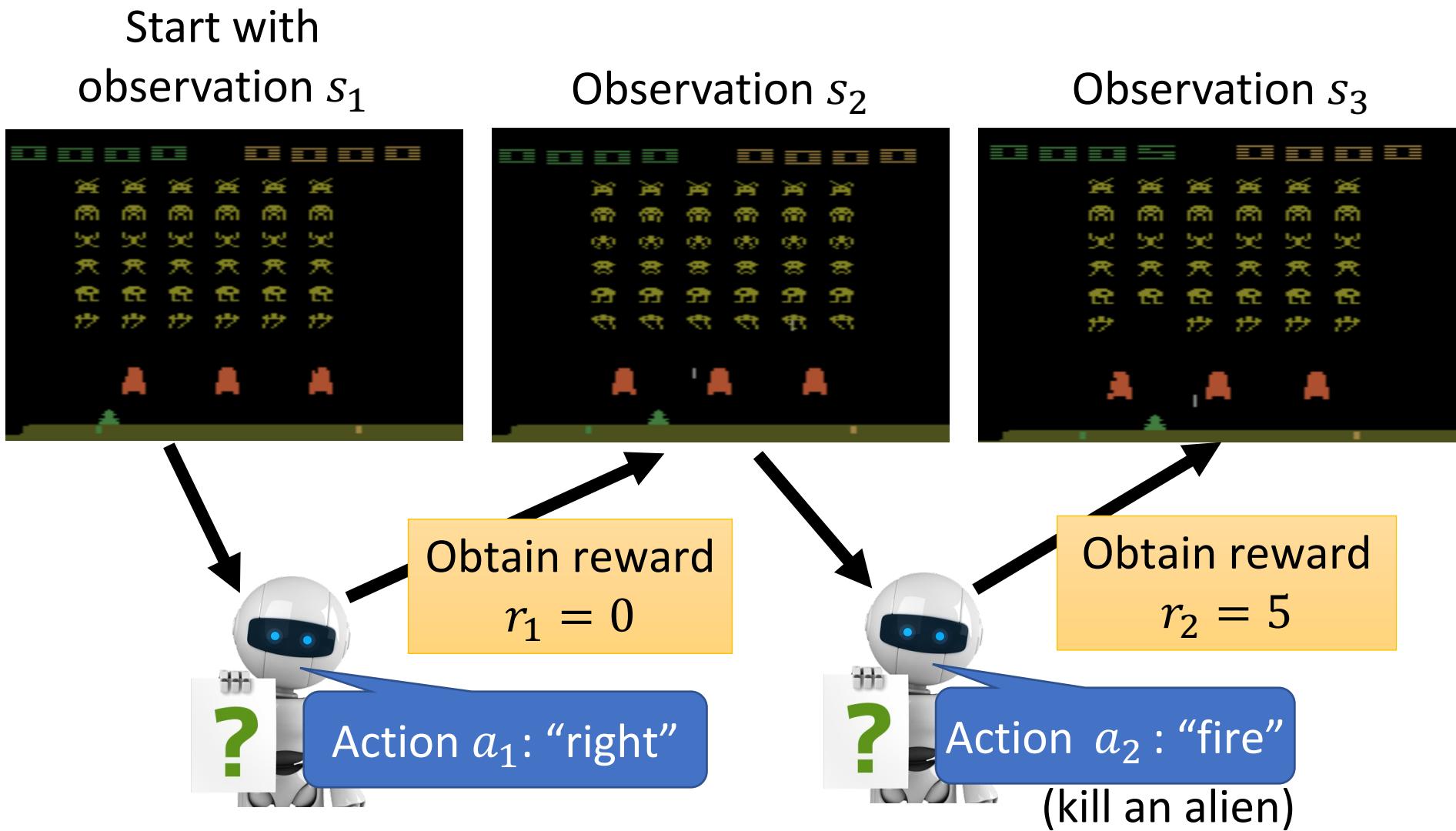
强化学习-游戏

- Space invader

Termination: all the aliens are killed,
or your spaceship is destroyed.



强化学习-游戏



强化学习-游戏

Start with
observation s_1



Observation s_2



Observation s_3



After many turns



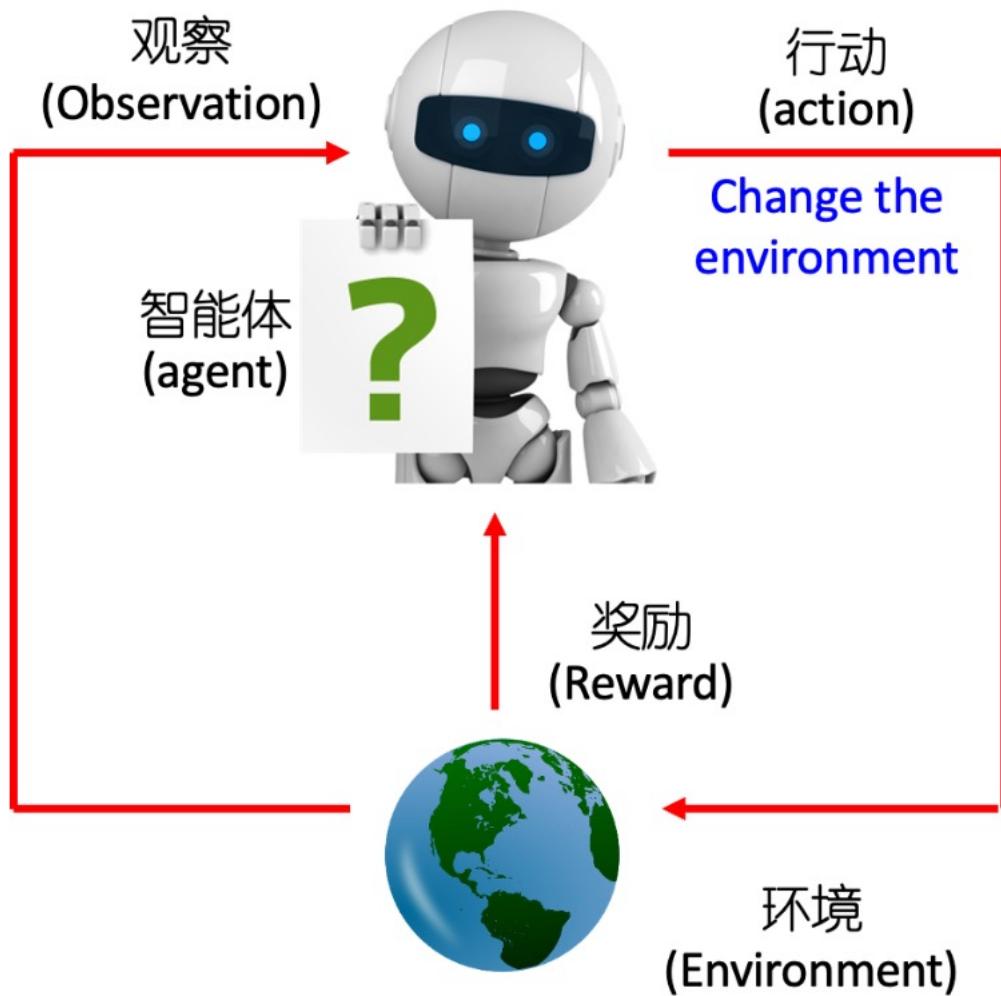
Action a_T

Obtain reward r_T

This is an *episode*.

强化学习的定义

通过与环境不断交互来最大化累计奖励的计算方法



- 三个方面：
 - 感知：在某种程度上感知环境的状态
 - 行动：可以采取行动来影响状态或者达到目标
 - 目标：随着时间推移最大化累积奖励

强化学习的特点

- 监督学习：给定有标记样本

Learning from teacher



Next move:
“5-5”



Next move:
“3-3”

- 强化学习：没有有标记样本，通过执行动作之后反馈的奖赏来学习

First move → many moves → Win!
(Two agents play with each other.)

Learning from experience

强化学习的特点

- 监督学习：给定有标记样本



- 强化学习：没有有标记样本，通过执行动作之后反馈的奖赏来学习

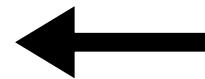


强化学习在某种意义上可以认为是具有“延迟标记信息”的监督学习

强化学习的特点

有监督、无监督学习

Model



Fixed Data

强化学习

Agent

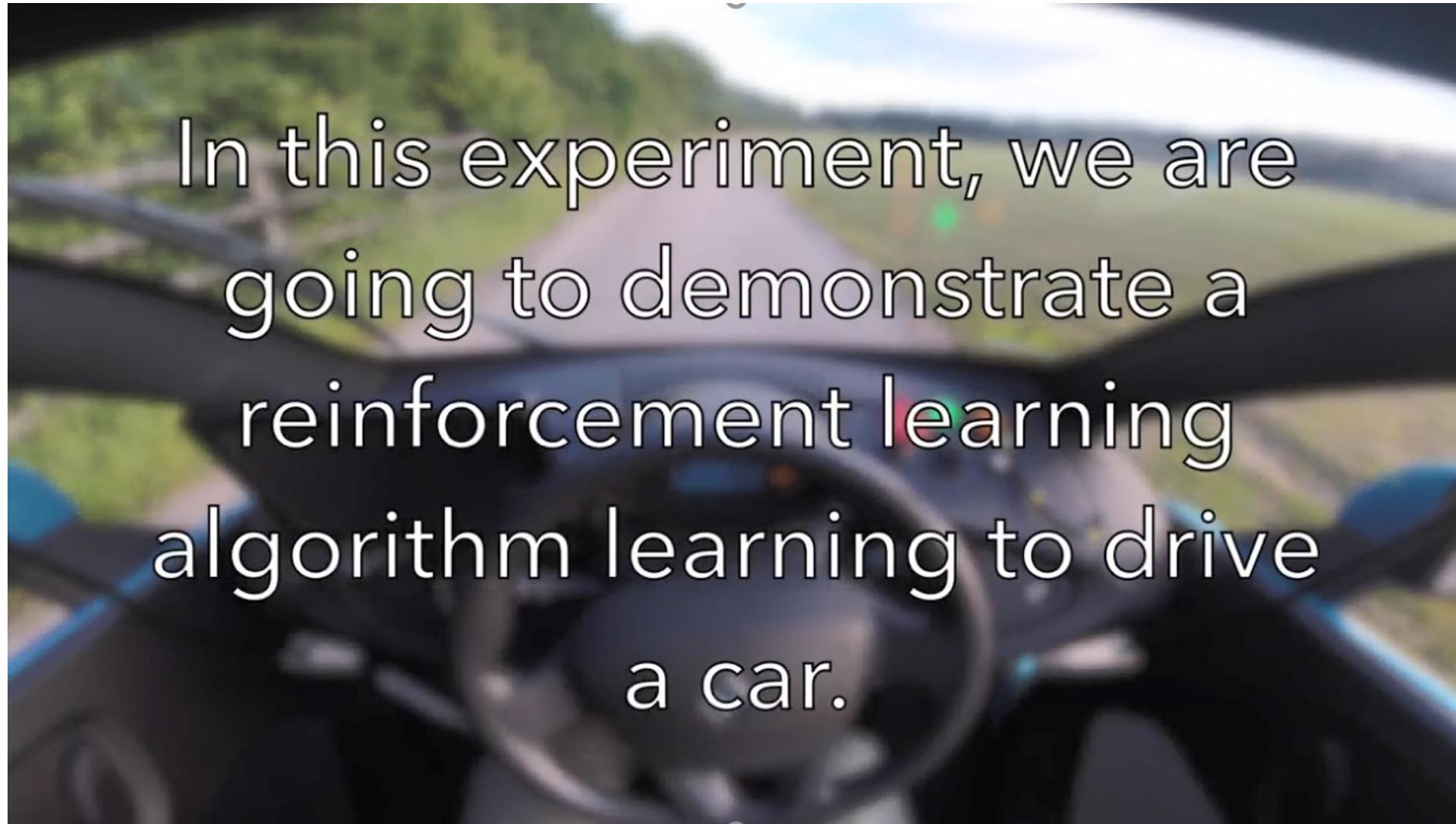


Agent不同，交互出
的数据也不同！



Dynamic Environment

强化学习应用案例：无人驾驶小车



In this experiment, we are going to demonstrate a reinforcement learning algorithm learning to drive a car.

更多应用

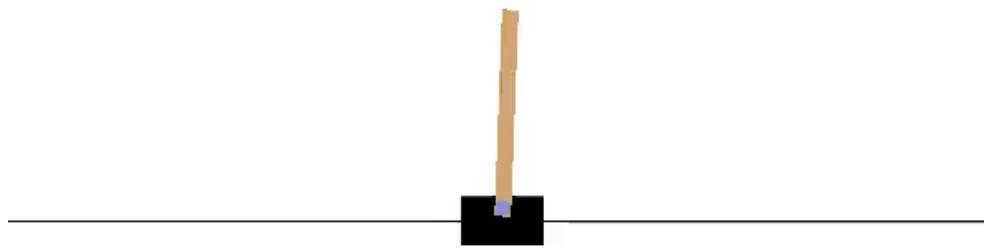
- Flying Helicopter
 - <https://www.youtube.com/watch?v=0JL04JJjocc>
- Driving
 - <https://www.youtube.com/watch?v=0xo1Ldx3L5Q>
- Robot
 - <https://www.youtube.com/watch?v=370cT-OAzzM>
- Text generation
 - <https://www.youtube.com/watch?v=pbQ4qe8EwLo>

扩展阅读

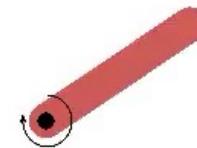
- Sutton's book:
 - <http://incompleteideas.net/sutton/book/the-book.html>
- David Silver's Lecture:
 - <https://www.davidsilver.uk/teaching/>
- 动手学强化学习：
 - <https://hrl.boyuai.com/chapter/intro>
- OpenAI GYM:
 - <https://github.com/openai/gym>
 - <https://gym.openai.com/>

Open AI Gym

经典控制问题



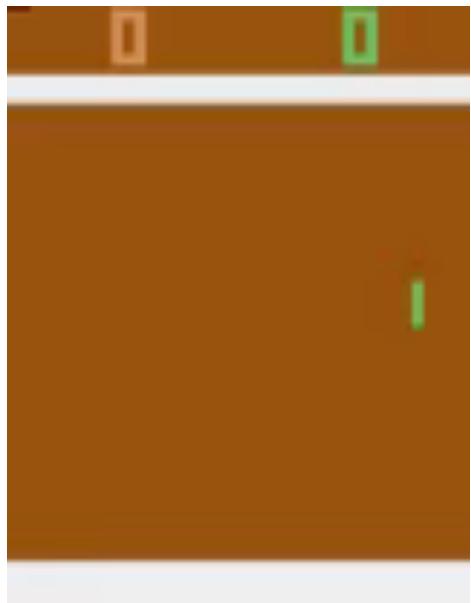
Cart Pole



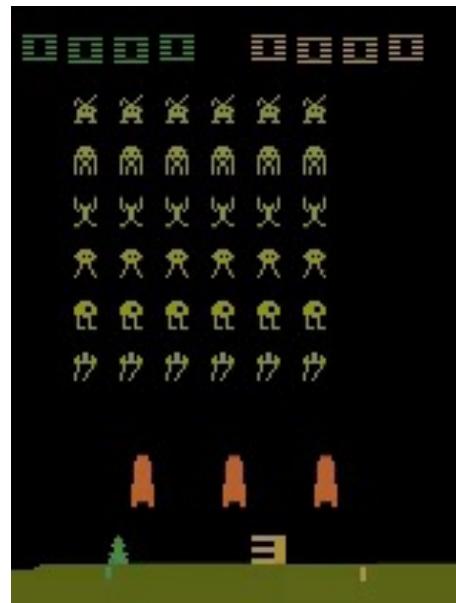
Pendulum

Open AI Gym

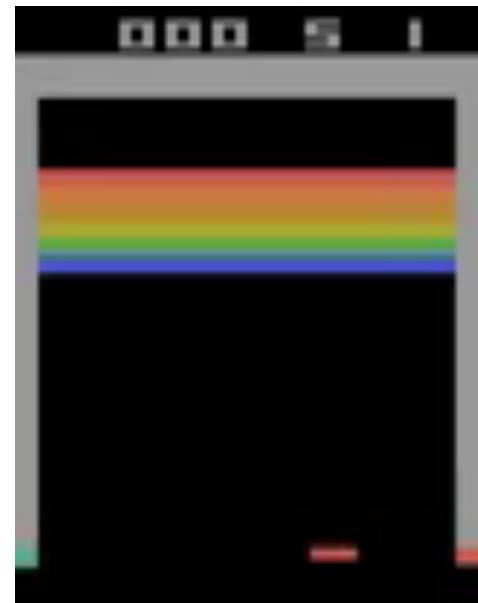
Atari游戏



Pong



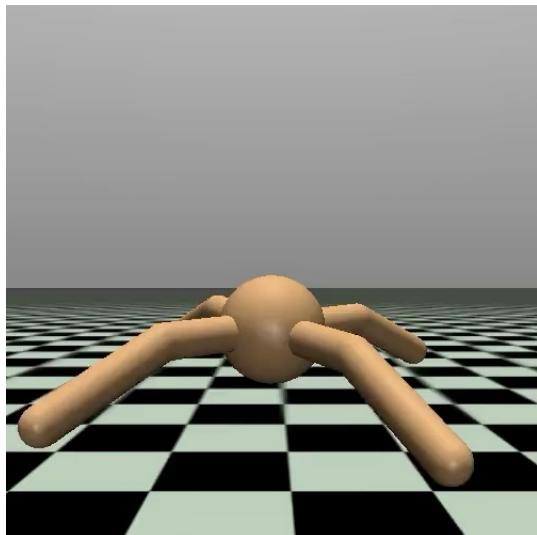
Space Invader



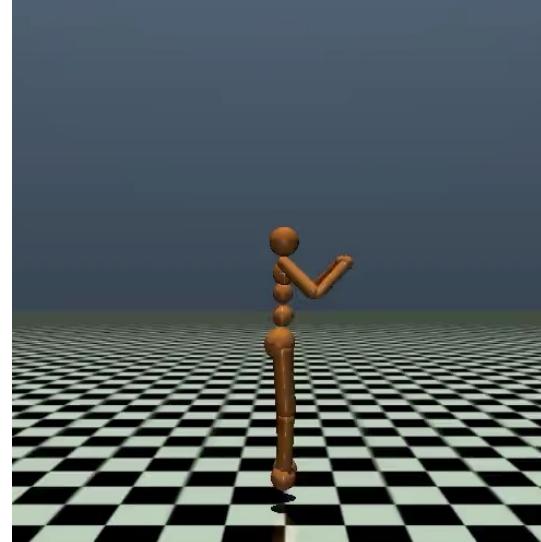
Breakout

Open AI Gym

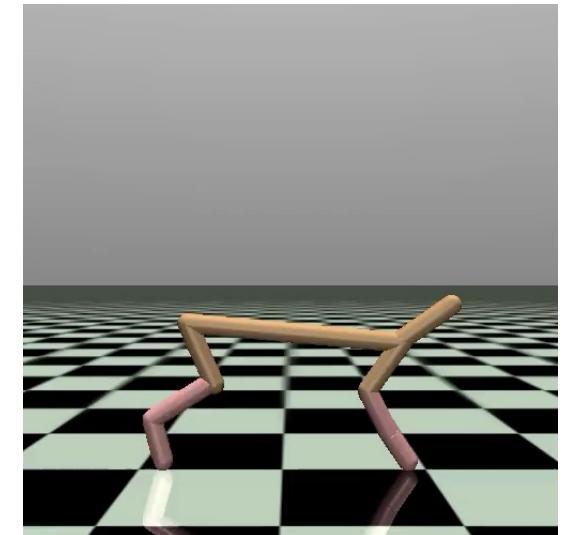
机器人的连续控制问题



Ant

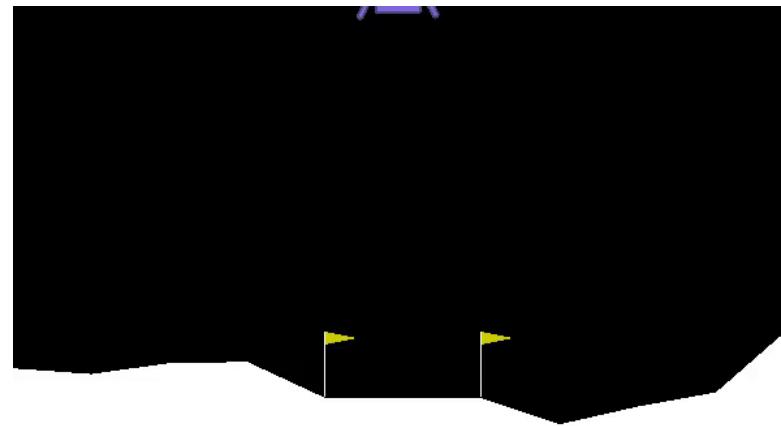
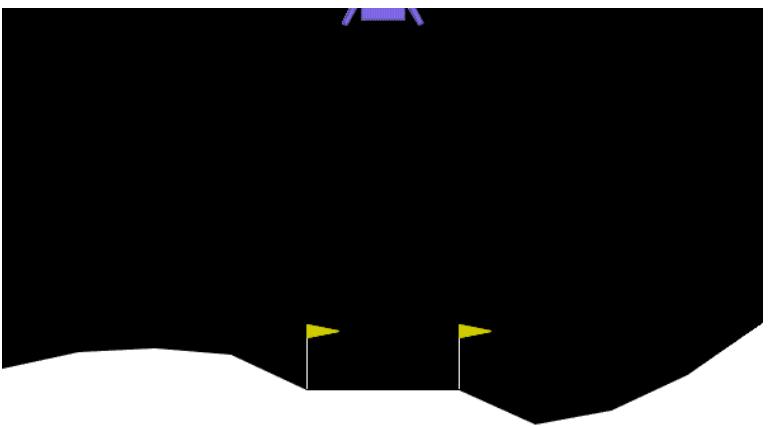


Humanoid



Half Cheetah

Open AI Gym



```
import gym
env = gym.make("LunarLander-v2", render_mode="human")
env.action_space.seed(42)

observation, info = env.reset(seed=42)

for _ in range(1000):
    observation, reward, terminated, truncated, info = env.step(env.action_space.sample())

    if terminated or truncated:
        observation, info = env.reset()

env.close()
```

大纲

□ 强化学习简介

□ 多臂老虎机

□ 有模型学习

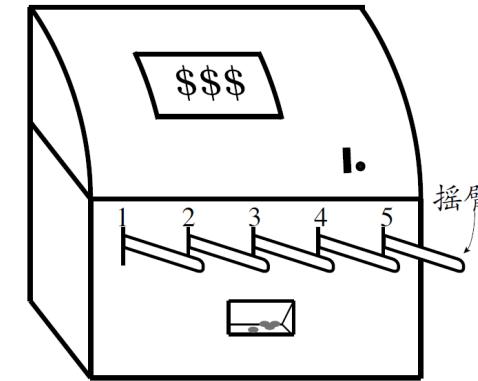
□ 无模型学习

□ 模仿学习

□ 深度强化学习

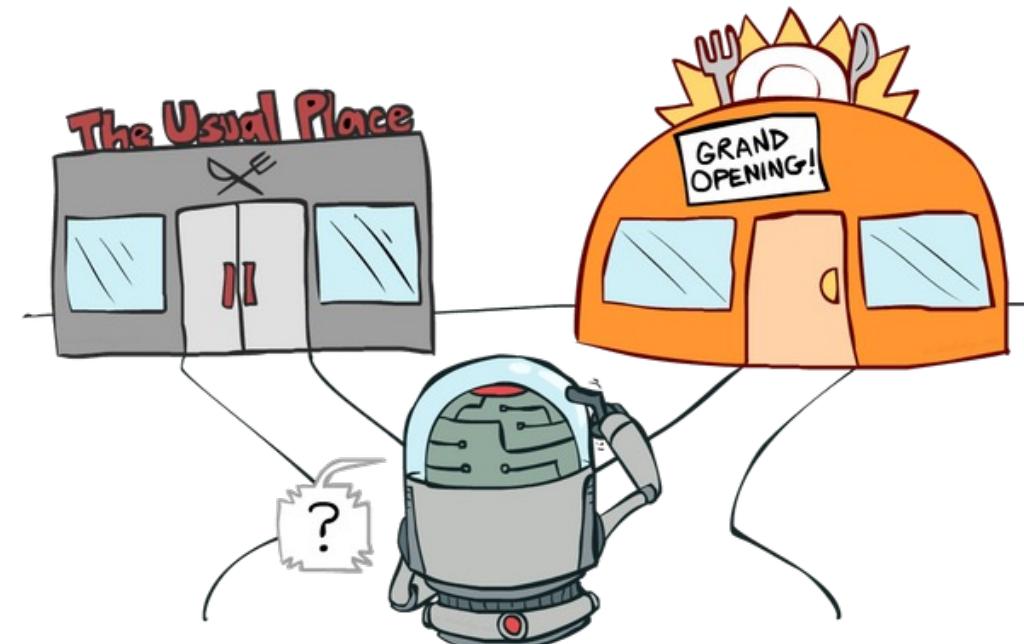
K-摇臂赌博机

- K-摇臂赌博机 (K-Armed Bandit)
 - 只有一个状态, K个动作
 - 每个摇臂的奖赏服从某个期望未知的分布
 - 执行有限次动作
 - 最大化累积奖赏
- 强化学习面临的主要困难：探索-利用窘境 (Exploration-Exploitation dilemma)
 - 探索(Exploration)：估计不同摇臂的优劣 (奖赏期望的大小)
 - 利用(Exploitation)：选择当前最优的摇臂



探索-利用窘境

- 基于目前策略获取已知最优收益还是尝试不同的决策
 - Exploitation：执行能够获得已知最优收益的决策
 - Exploration：尝试更多可能的决策，不一定会是最优收益



策略探索的一些原则

- 朴素方法 (Naïve Exploration)
 - 添加策略噪声 ϵ -greedy
- 积极初始化 (Optimistic Initialization)
- 基于不确定性的度量 (Uncertainty Measurement)
 - 尝试具有不确定收益的策略，可能带来更高的收益
- 概率匹配 (Probability Matching)
 - 基于概率选择最佳策略

K-摇臂赌博机

- 在探索与利用之间进行折中： ϵ -贪心和Softmax
- ϵ -贪心
 - 以 ϵ 的概率探索：均匀随机选择一个摇臂
 - 以 $1 - \epsilon$ 的概率利用：选择当前平均奖赏最高的摇臂
- Softmax：基于当前已知的摇臂平均奖赏来对探索与利用折中
 - 若某个摇臂当前的平均奖赏越大，则它被选择的概率越高
 - 概率分配使用Boltzmann分布：
$$P(k) = \frac{e^{\frac{Q(k)}{\tau}}}{\sum_{i=1}^K e^{\frac{Q(i)}{\tau}}}$$
- 两种算法都有一个折中参数 (ϵ, τ) ，算法性能孰好孰坏取决于具体应用问题

大纲

□ 强化学习简介

□ 多臂老虎机

□ 有模型学习

□ 无模型学习

□ 模仿学习

□ 深度强化学习

马尔可夫决策过程

□ 马尔可夫过程（Markov Process）是具有马尔可夫性质的随机过程

□ 状态 S_t 是马尔可夫的，当且仅当

$$\mathbb{P}[S_{t+1}|S_t] = \mathbb{P}[S_{t+1}|S_1, \dots, S_t]$$

□ 马尔可夫决策过程（Markov Decision Process, MDP）

- 提供了一套为在结果部分随机、部分在决策者的控制下的决策过程建模的数学框架

$$\mathbb{P}[S_{t+1}|S_t] = \mathbb{P}[S_{t+1}|S_1, \dots, S_t]$$

$$\mathbb{P}[S_{t+1}|S_t, A_t]$$

有模型学习

- 强化学习对应了马尔可夫四元组： (S, A, P, R)
- 有模型学习(model-based learning)：假设 S, A, P, R 均已知
- MDP模型的动态转换如下所示：
 - 从状态 s_0 开始，智能体选择某个动作 $a_0 \in A$ ，智能体得到奖励 r_0
 - MDP随机转移到下一个状态 $s_1 \sim P_{s_0 a_0}$
 - 这个过程不断进行

$$s_0 \xrightarrow{a_0, r_0} s_1 \xrightarrow{a_1, r_1} s_2 \xrightarrow{a_2, r_2} s_3 \dots$$

- 智能体的总回报为

$$r_0 + \gamma r_1 + \gamma^2 r_2 + \dots$$

有模型学习

- 目标：选择能够最大化累积奖励期望的策略 $\pi(s): S \rightarrow A$

$$\mathbb{E}[r_0 + \gamma r_1 + \gamma^2 r_2 + \dots]$$

- 三个步骤：
 - 策略评估
 - 策略改进
 - 策略迭代

策略评估

- 给策略 π 定义**价值函数**, 价值函数是状态(或状态-动作二元组)的函数, 用来评估当前智能体在给定状态(获给定状态与动作)下有多好, 这里的“好”是基于未来预期的累计奖励来定义的。
- 状态-动作值函数(State-Action value function)

$$Q^\pi(s_t, a_t) = \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots | s = s_t, a = a_t]$$

- 状态值函数(State value function)

$$V^\pi(s_t) = \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots | s = s_t]$$

$$V^\pi(s_t) = \sum_a \pi(s_t, a) \cdot Q^\pi(s_t, a)$$

策略评估

$$V^\pi(s) = \mathbb{E}[\underbrace{r_1 + \gamma r_2 + \gamma^2 r_3 + \dots}_{\gamma V^\pi(s_{t+1})} | s_0 = s]$$

$$= \sum_a \pi(s, a) \sum_{s' \in S} P_{sa}(s') (r_{sa}(s') + \gamma V^\pi(s'))$$

Bellman等式

$$Q^\pi(s, a) = \sum_{s' \in S} P_{sa}(s') (r_{sa}(s') + \gamma V^\pi(s'))$$

策略改进

- 最优策略对应的值函数称为**最优值函数**

$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

- 最优价值函数的Bellman等式

$$V^*(s) = \max_a \sum_{s' \in S} P_{sa}(s') (r_{sa}(s') + \gamma V^{\pi}(s')) = \max_a Q^{\pi*}(s, a)$$

最优Bellman等式

- 非最优策略的改进方式：将策略选择的动作改为当前最优的动作

$$\pi'(s) = \arg \max_{a \in A} Q^{\pi}(s, a)$$

策略迭代

- 对于一个动作空间和状态空间有限的MDP

$$|S| < \infty, |A| < \infty$$

- 策略迭代过程
 - 随机初始化策略 π
 - 重复以下过程直到收敛 {
 - 策略评估: $V := V^\pi$
 - 策略改进: 对每个状态, 更新

$$\pi(s) = \arg \max_{a \in A} Q^\pi(s, a)$$

每次改进策略后需要重新进行策略评估, 通常比较耗时

价值迭代

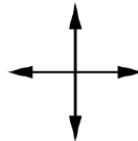
- 对于一个动作空间和状态空间有限的MDP

$$|S| < \infty, |A| < \infty$$

- 价值迭代过程
 - 对每个状态 s , 初始化 $V(s) = 0$
 - 重复以下过程直到收敛 {
 - 对每个状态, 更新

$$V(s) = \max_a \sum_{s' \in S} P_{sa}(s') (r_{sa}(s') + \gamma V^\pi(s'))$$
$$\}$$

策略评估：例子



	1	2	3
4	5	6	7
8	9	10	11
12	13	14	

- 非折扣MDP ($\gamma = 1$)
- 非终止状态：1, 2, ..., 14
- 两个终止状态（灰色方格）
- 如果动作指向所有方格以外，则这一步不动
- 奖励均为-1，直到到达终止状态
- 智能体的策略为均匀随机策略

$$\pi(n | \cdot) = \pi(e | \cdot) = \pi(s | \cdot) = \pi(w | \cdot) = 0.25$$

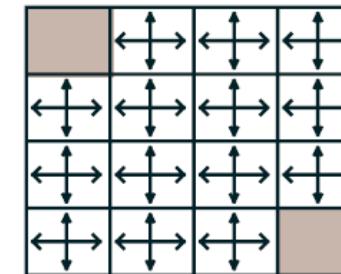
策略评估：例子

K=0

随机策略的 V_k

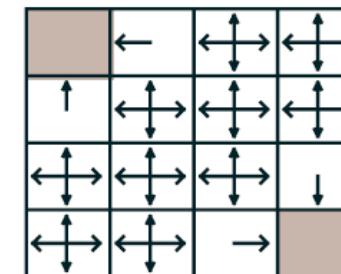
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

V_k 对应的贪心策略



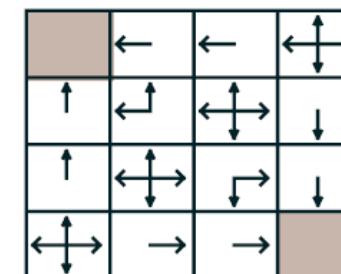
K=1

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0



K=2

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0



策略评估：例子

$K=3$

随机策略的 V_k

0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

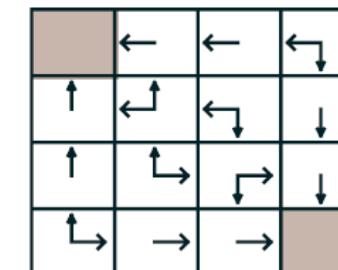
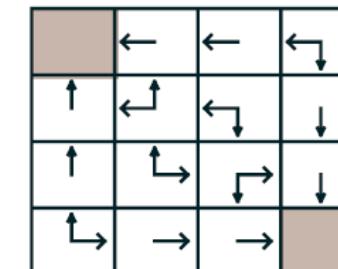
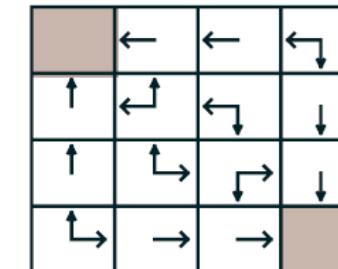
$K=10$

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0

$K=\infty$

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0

V_k 对应的贪心策略



$V := V^\pi$
最优策略

有模型学习

□ 有模型学习小结

- 强化学习任务可归结为**基于动态规划的寻优问题**
- 与监督学习不同，这里并未涉及到泛化能力，而是为每一个状态找到最好的动作

□ 问题：如果模型未知呢？

- 从“经验”中学习一个MDP模型
- 不学习MDP，从经验中直接学习价值函数和策略：
 - 模型无关的强化学习 (Model-free Reinforcement Learning)