# 高级机器学习

# 强化学习
# (reinforcement learning)

# 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

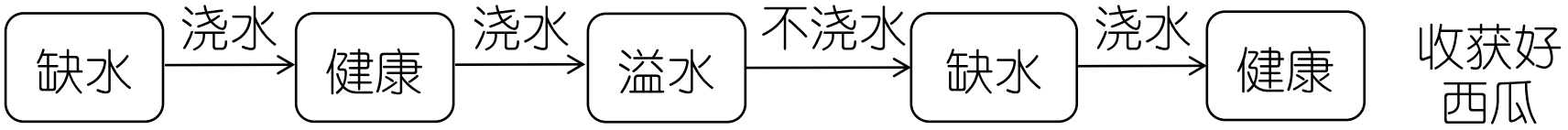

INNOVATIONS IN
The microbiome
nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
LEARNING CURVE
Self-taught AI software attains human-level performance in video games
PAGES 486 & 529
EPIDEMIOLOGY
SHARE DATA IN OUTBREAKS
Forge open access to sequences and more
PAGE 477
COSMOLOGY
A GIANT IN THE EARLY UNIVERSE
A supermassive black hole at a redshift of 6.3
PAGES 490 & 512
QUANTUM PHYSICS
TELEPORTATION FOR TWO
Transferring two properties of a single photon
PAGES 491 & 516
NATURE.COM/NATURE
26 February 2015
Vol. 518, No. 7540


nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO
CONSERVATION
SONGBIRDS À LA CARTE
Illegal harvest of millions of Mediterranean birds
PAGE 452
RESEARCH ETHICS
SAFEGUARD TRANSPARENCY
Don't let openness backfire on individuals
PAGE 459
POPULAR SCIENCE
WHEN GENES GOT 'SELFISH'
Dawkins's calling card 40 years on
PAGE 462
NATUREASIA.COM
28 January 2016
Vol. 529, No. 7587

# Atari(雅达利游戏)

# 例子：瓜农种西瓜

种下瓜苗后：(仅考虑浇水和不浇水两个动作，不考虑施肥、除草等 )

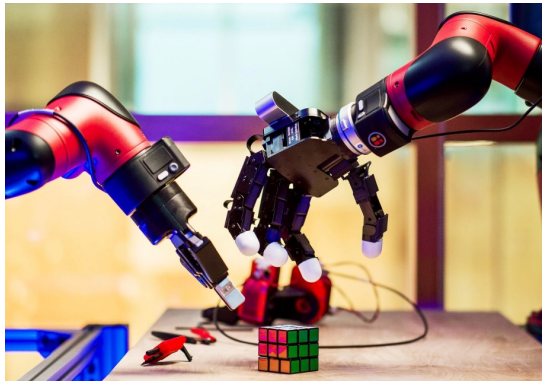



- 多步决策过程
- 过程中包含状态、动作、反馈 (奖赏) 等
- 需多次种瓜，在过程中不断摸索，才能总结出较好的种瓜策略

抽象该过程：序贯决策任务(Sequential Decision Making)

# 两种人工智能任务类型

- 预测型任务
  - 根据数据特征预测对应标注（有监督学习）
  - 根据输入数据输出所需新的数据（无监督学习）
- 决策型任务
  - 在动态环境中采取行动（强化学习）
    - 转变到新的状态
    - 获得即时奖励
    - 随着时间的推移最大化累计奖励
    - Learning from interaction in a trial-and-error manner

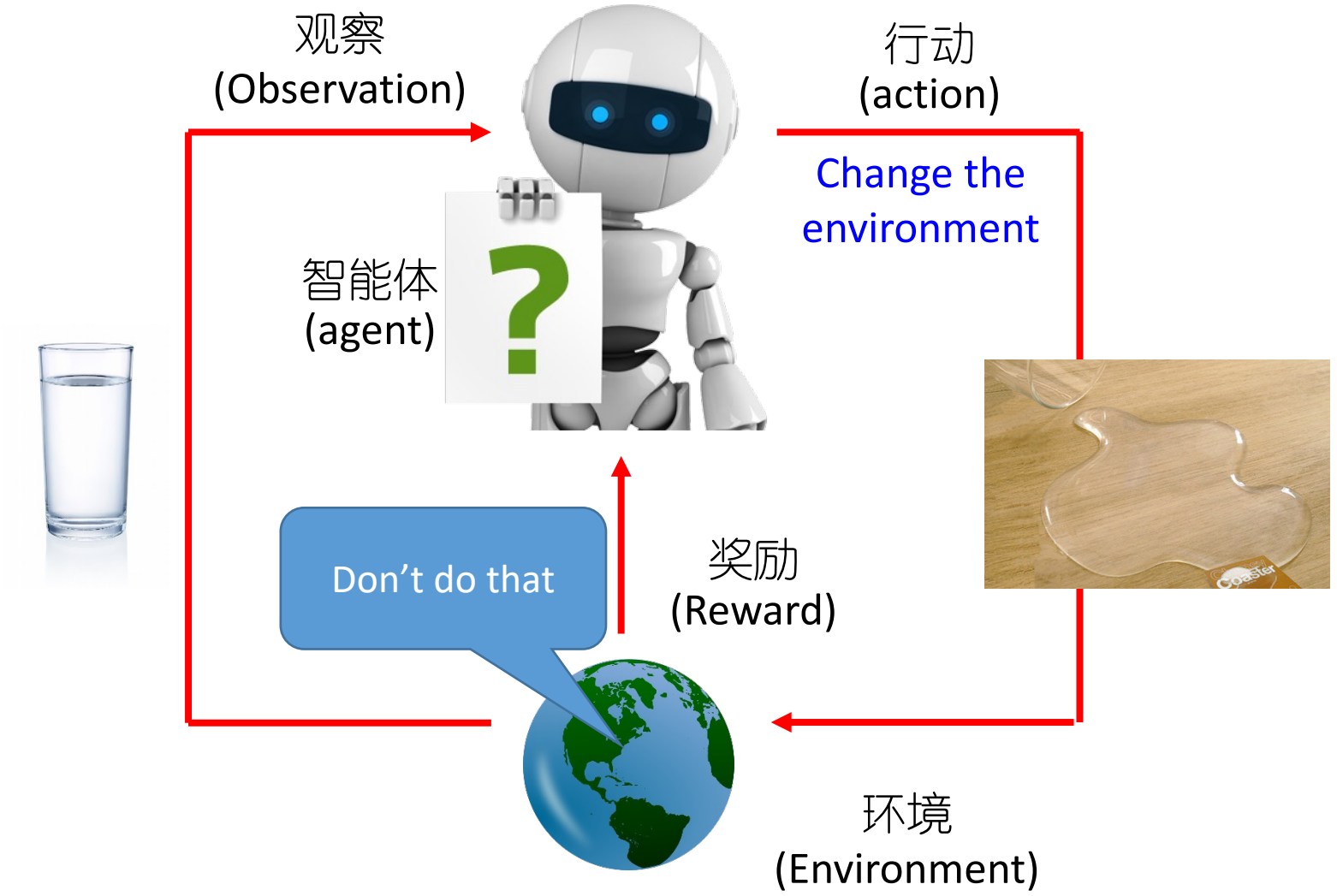


# 决策和预测的不同

- 预测仅产生信号，不考虑环境的改变
  - 预测的信号是否用、怎么用，不需要考虑
- 决策下达到环境中，直接改变环境
  - 未来发展随之改变
  - “做决策，担责任”
- 序贯决策
  - 决策者序贯地做出一个个决策，并接续看到新的观测，直到最终任务结束

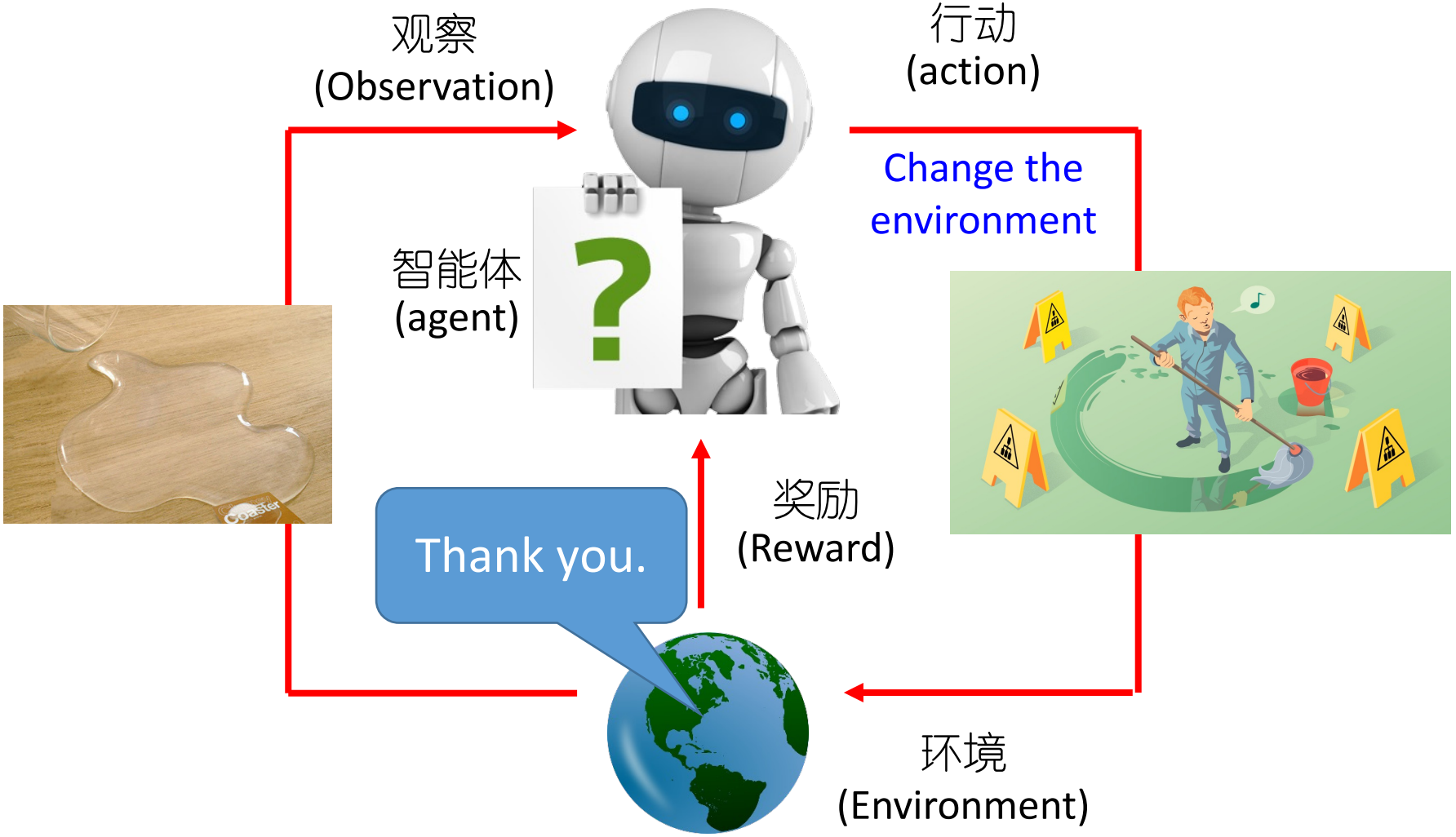
绝大多数序贯决策问题，可以用强化学习来解

# 强化学习


观察
(Observation)
行动
(action)
Change the environment
智能体
(agent)
?
Don’t do that
奖励
(Reward)
环境
(Environment)

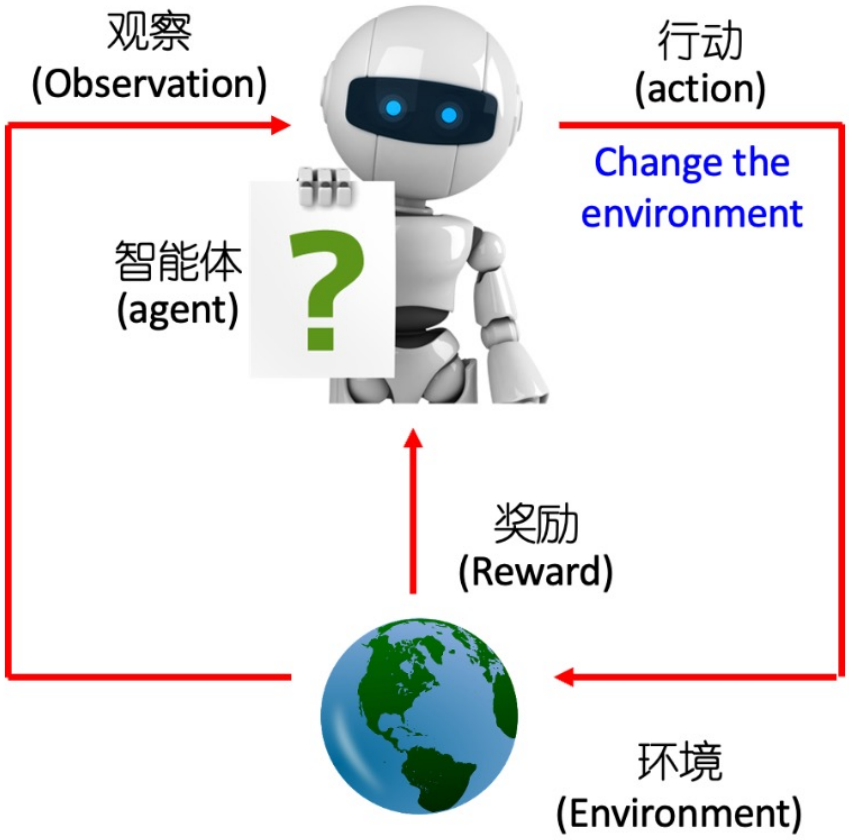
# 强化学习

智能体从交互中学习得到最大化累计奖励的行动



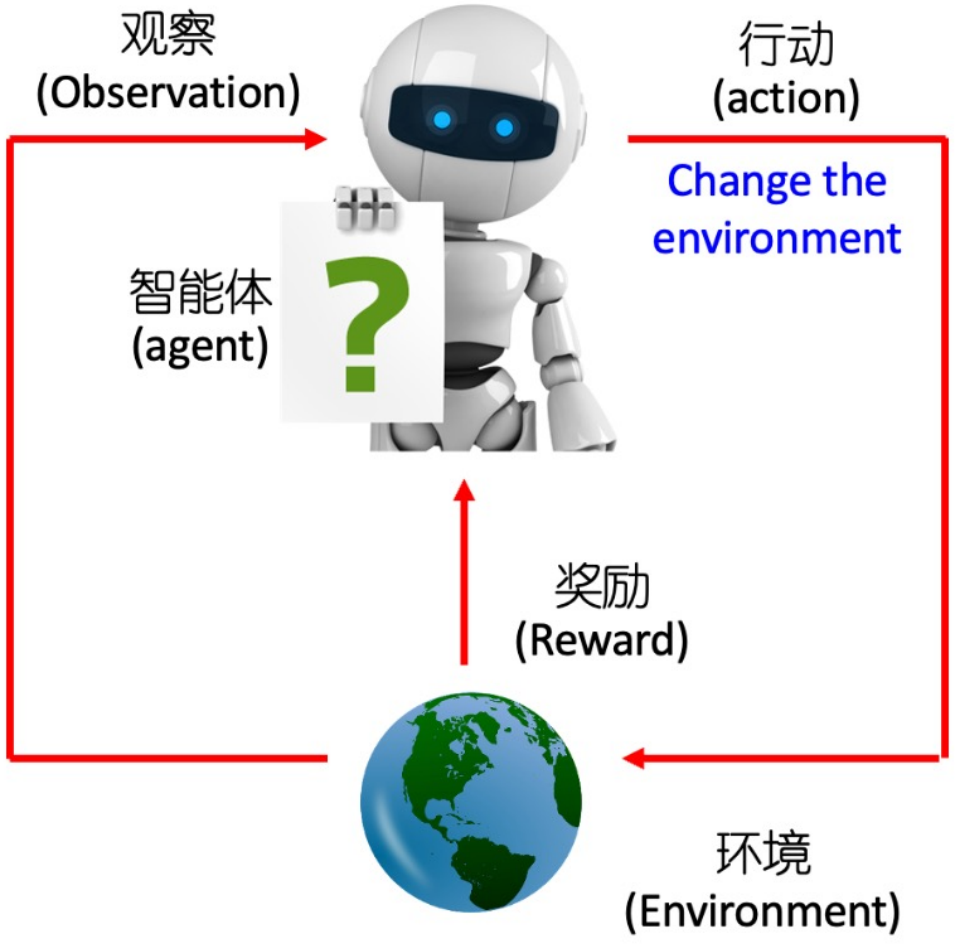
# 强化学习的定义

通过与环境不断交互来实现目标的计算方法




- 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

# 强化学习的交互过程

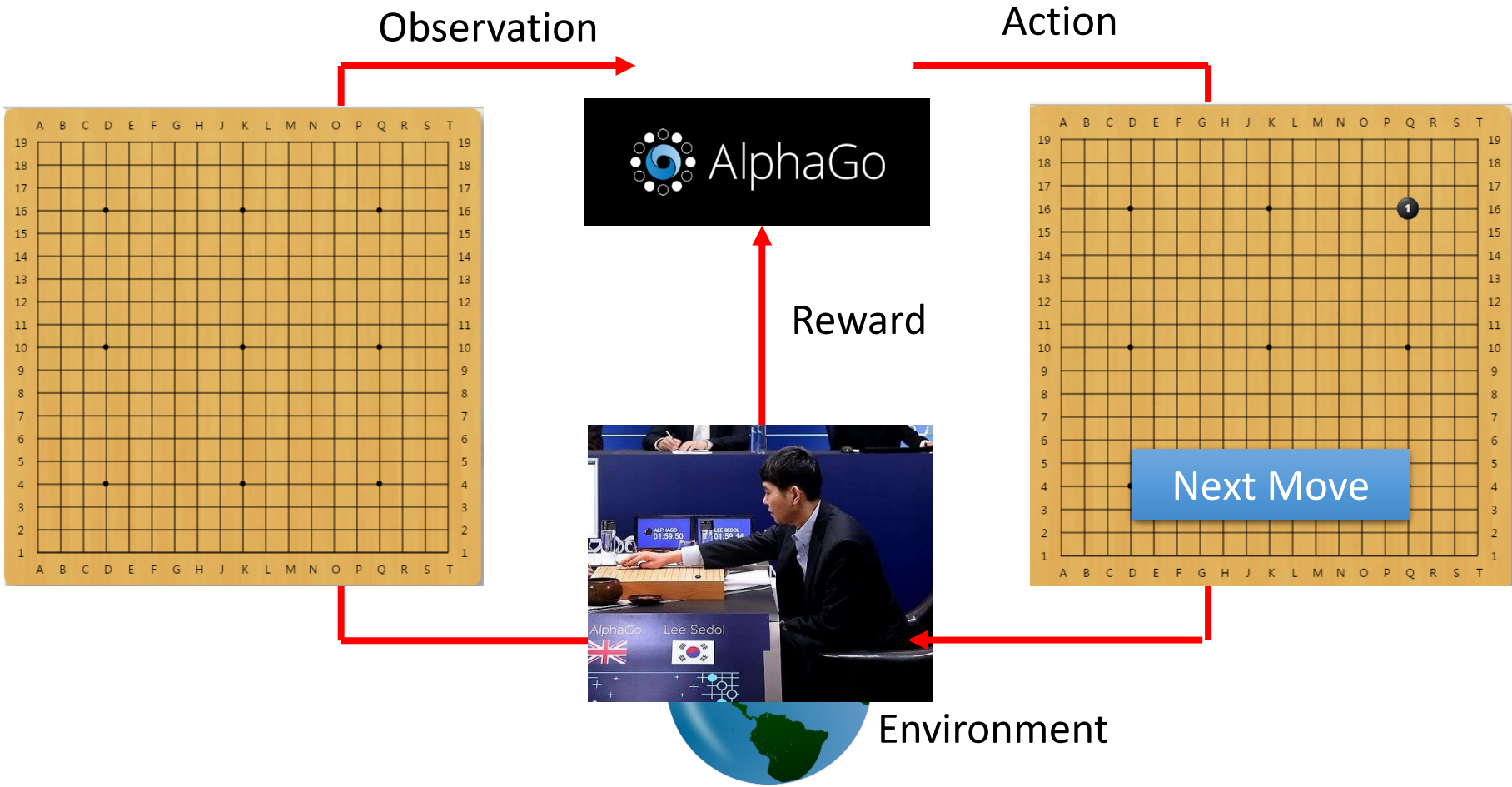



- 在每一步 $t$，智能体：
  - 获得观察 $O_t$
  - 获得奖励 $R_t$
  - 执行行动 $A_t$
- 环境：
  - 获得行动 $A_t$
  - 给出观察 $O_{t+1}$
  - 给出奖励 $R_{t+1}$
- $t$ 在环境这一步增加

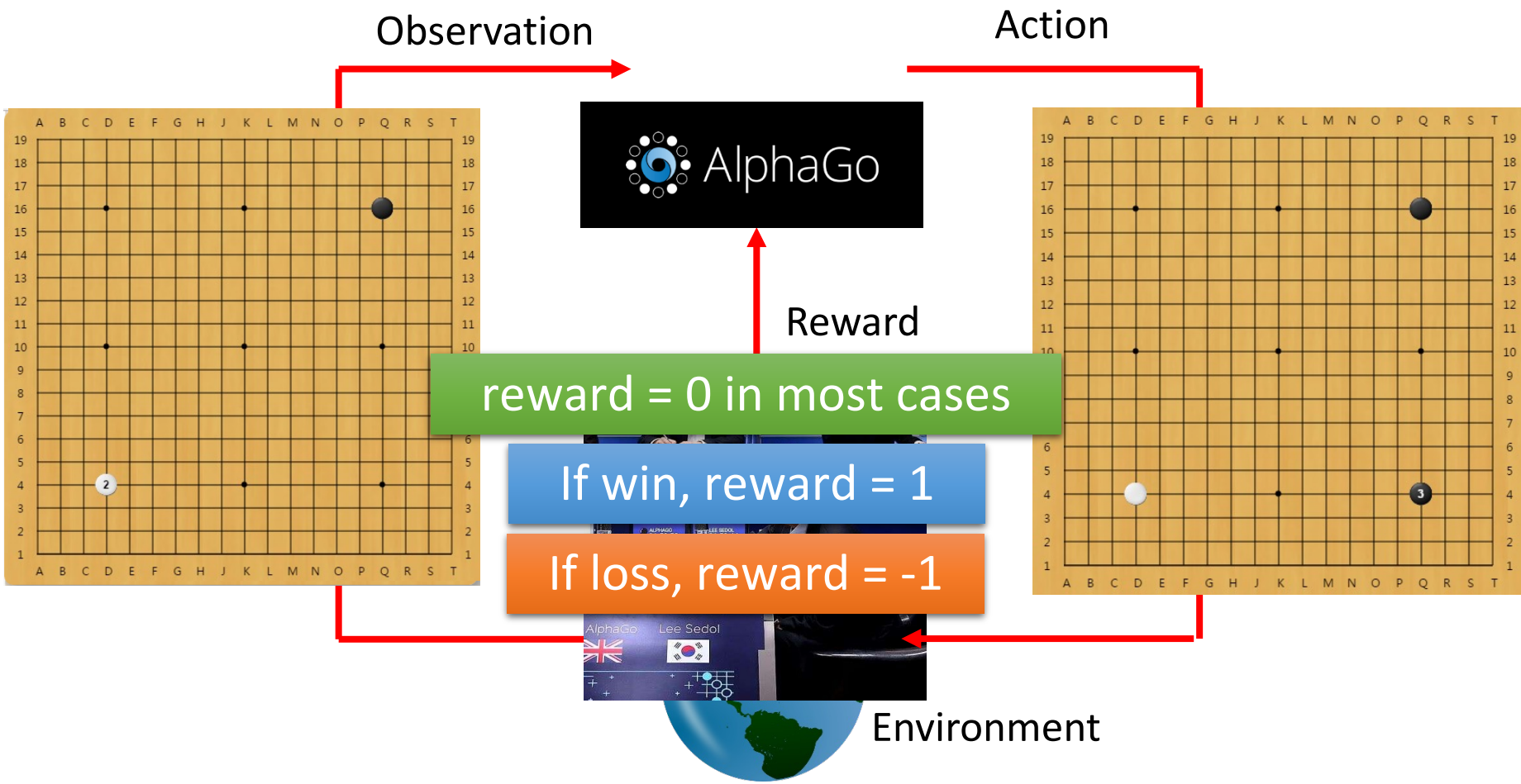
# 强化学习的关键要素

- 智能体所处的环境$E$
  - 例如，种西瓜任务中，环境是西瓜生长的自然世界
- 状态的集合$X$：$x \in X$是智能体感知到的环境的描述
  - *西瓜长势的描述*
- 智能体所能采取的动作集合$A$
  - *浇水、施肥等*
- 策略(policy) π：$X \to A$（或π：$X \times A \to \mathbb{R}$）
  - *例如，瓜苗状态是缺水时，返回动作浇水*
- 状态转移概率$P$：$X \times A \times X \to \mathbb{R}$
  - 瓜苗当前状态缺水，选择动作浇水，有一定概率恢复健康，也有一定概率无法恢复
- 奖励函数：$R: X \times A \times X \mapsto \mathbb{R}$ *(或*$R$：$X \times X \to \mathbb{R}$)
  - 瓜苗健康对应奖赏+1，瓜苗凋零对应奖赏-10

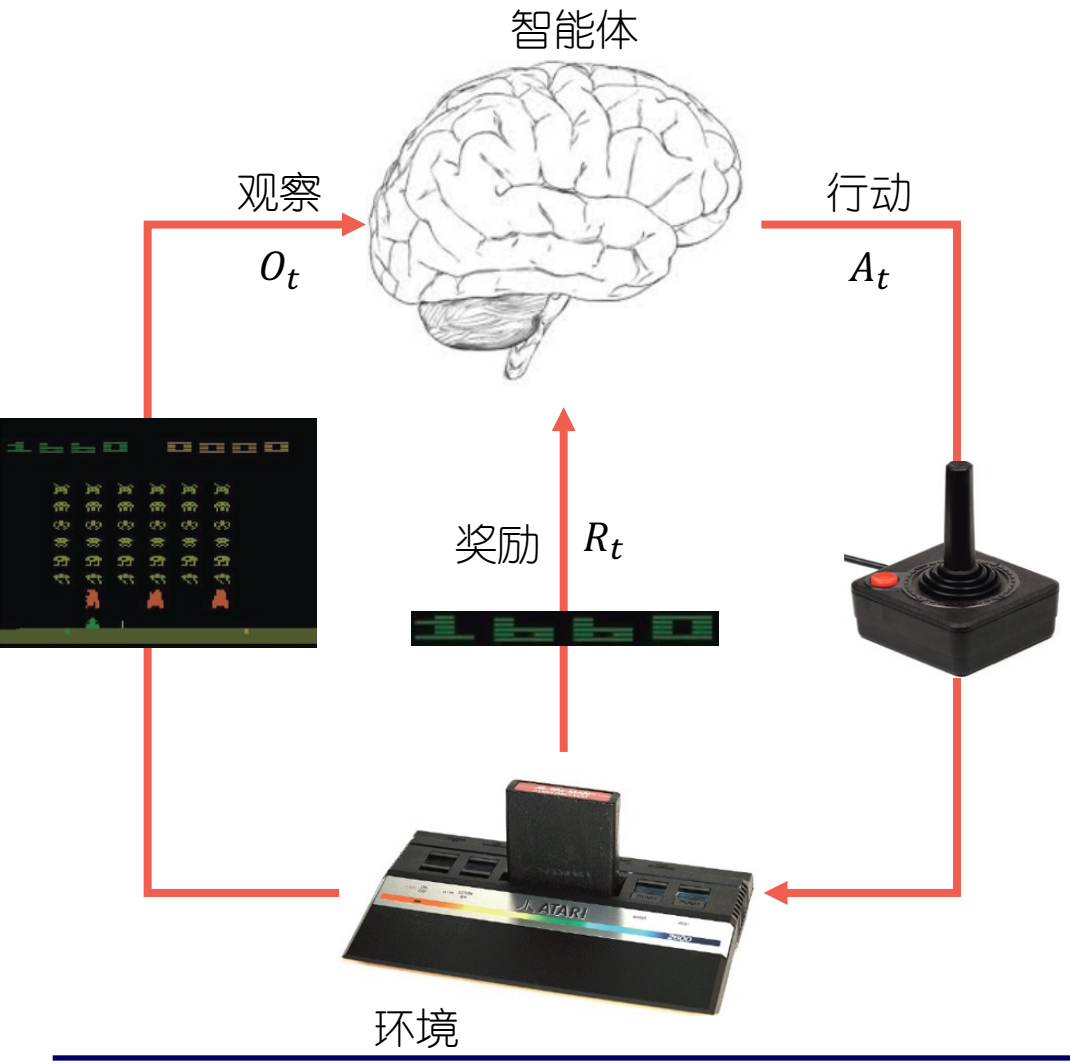
# 强化学习-围棋


Observation
Action
AlphaGo
Reward
Next Move
AlphaGo
Lee Sedol
Environment

# 强化学习-围棋


Observation
Action
AlphaGo
Reward
reward = 0 in most cases
If win, reward = 1
If loss, reward = -1
AlphaGo
Lee Sedol
Environment

# 强化学习-游戏

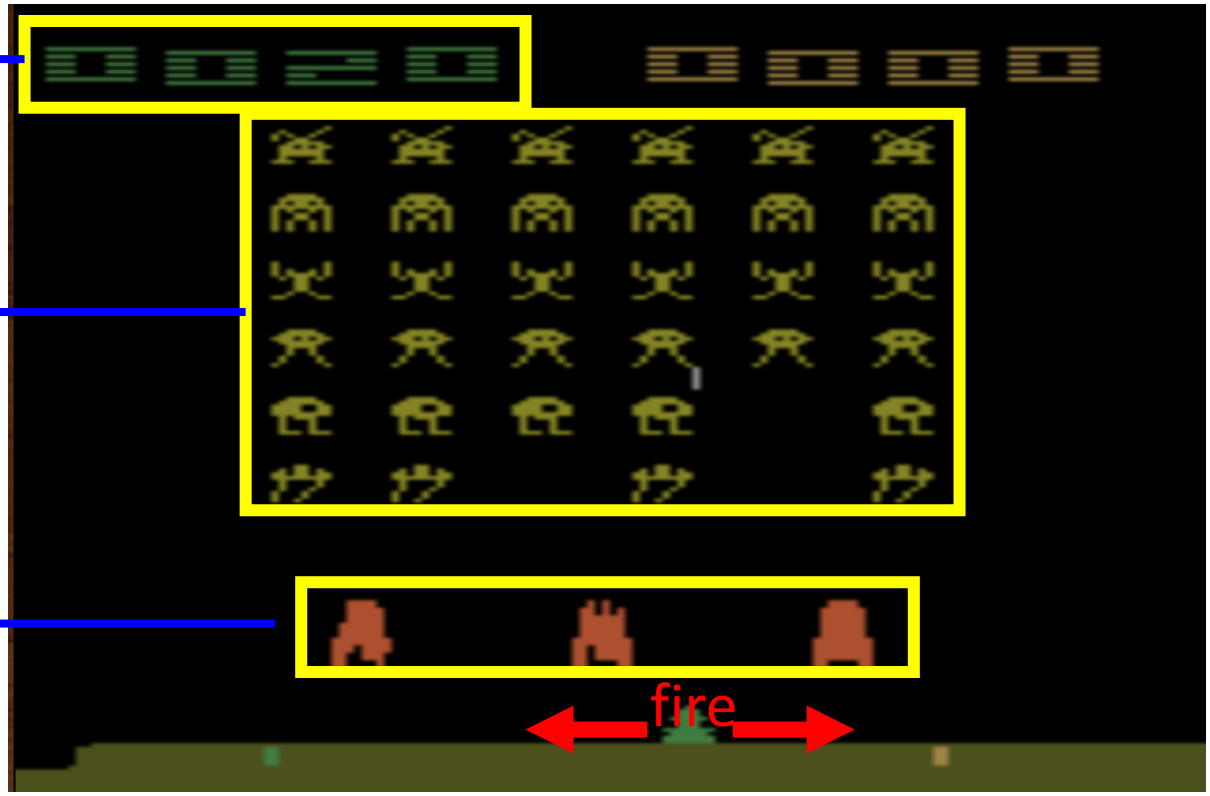



- 游戏规则未知
- 从交互游戏中进行学习
- 在操纵杆上选择行动并查看分数和像素画面

# 强化学习-游戏

- Space invader

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

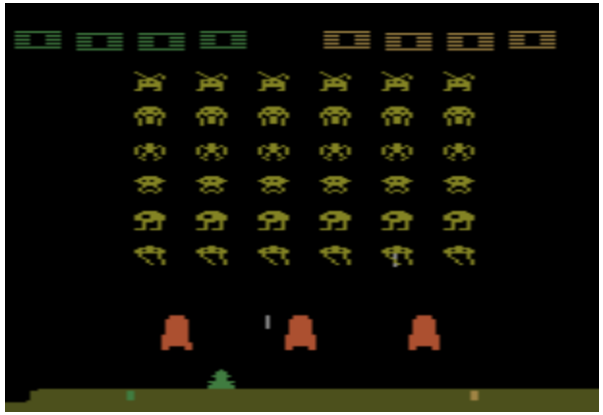


# 强化学习-游戏

Start with observation $s_1$

Observation $s_2$

Observation $s_3$



Obtain reward $r_1 = 0$

Action $a_1$: "right"

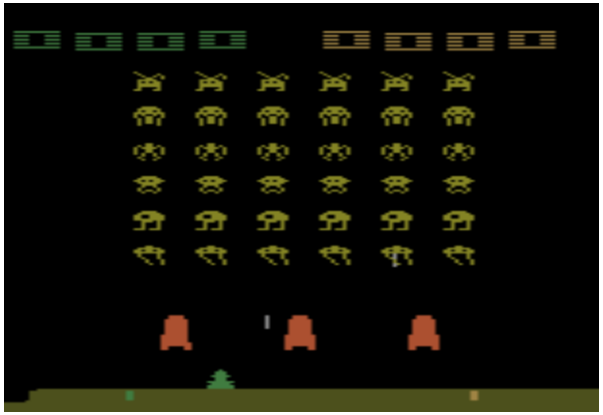
Obtain reward $r_2 = 5$

Action $a_2$: "fire" (kill an alien)

# 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***.

agent学习每个episode
最大化奖励的行动

# 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

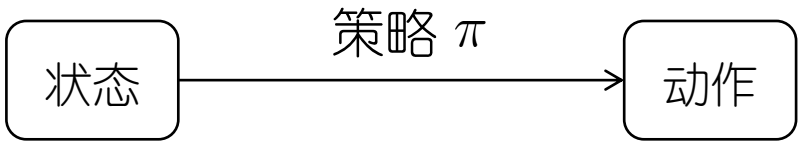
Learning from experience

First move → …… many moves …… → Win!

(Two agents play with each other.)

# 强化学习的特点


- 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 ⟷

Dynamic Environment

Agent不同，交互出的数据也不同！

# 强化学习应用案例：无人驾驶小车


Alex Kendall et.al, Learning to Drive in a Day. ICRA 2019: 8248-8254
https://www.youtube.com/watch?v=eRwTbRtnT1I

# 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

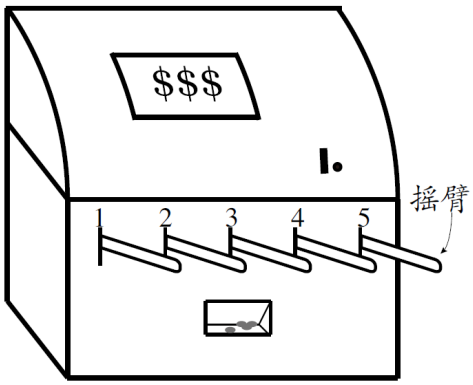
- Sutton’s book:
  - http://incompleteideas.net/sutton/book/the-book.html
- David Silver’s Lecture:
  - https://www.davidsilver.uk/teaching/
- 动手学强化学习：
  - https://hrl.boyuai.com/chapter/intro
- OpenAI GYM:
  - https://github.com/openai/gym
  - https://openai.com/blog/universe/

# 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

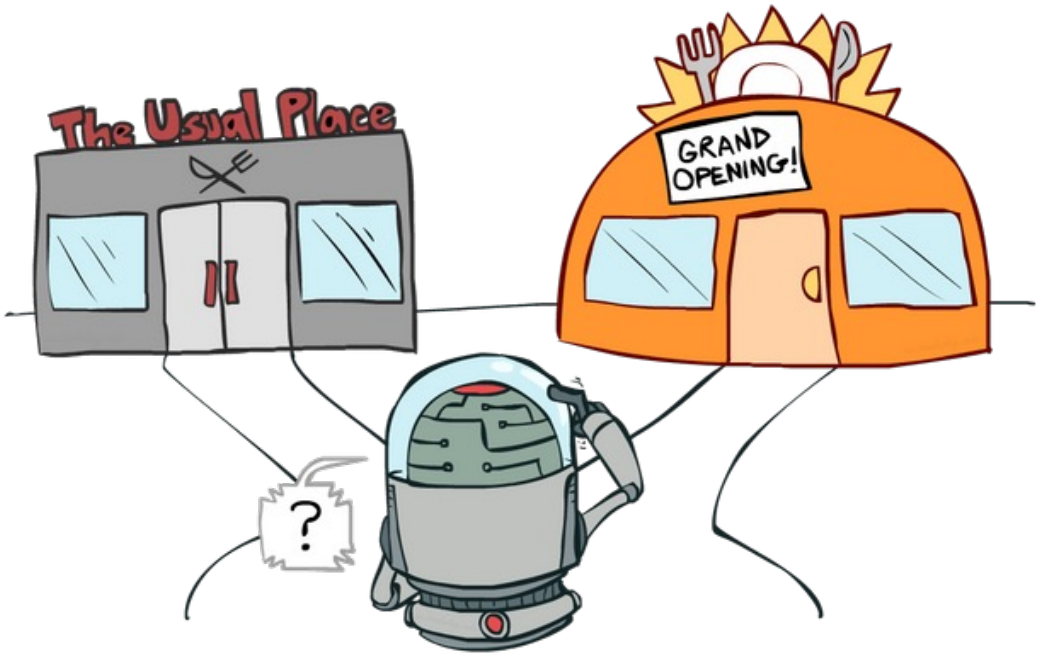
- K-摇臂赌博机（K-Armed Bandit）
  - 只有一个状态，K个动作
  - 每个摇臂的奖赏服从某个期望未知的分布
  - 执行有限次动作
  - 最大化累积奖赏




- 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 $\epsilon$-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

# 策略探索的一些原则

- $\epsilon$-贪心
  - 以$\epsilon$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$(\epsilon, \tau)$，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

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

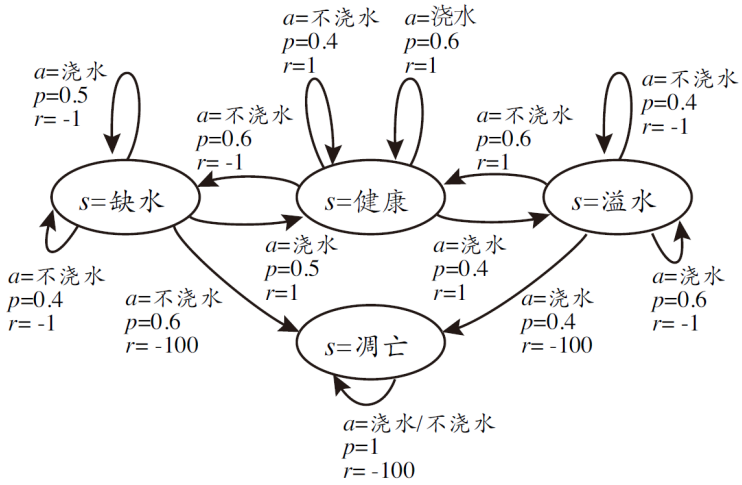
- 马尔可夫决策过程（Markov Decision Process，MDP）
  - 提供了一套为在结果部分随机、部分在决策者的控制下的决策过程建模的数学框架

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

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

# 强化学习对应了马尔可夫四元组

- 强化学习对应了四元组：$(S, A, P, R)$




- 强化学习的目标

  机器通过在环境中不断尝试从而学到一个策略$\pi$，

  使得长期执行该策略后得到的累积奖赏最大

  T步累积奖赏：$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} r_t\right]$

  $\gamma$折扣累积奖赏：$\mathbb{E}[\sum_{t=0}^{+\infty} \gamma^t r_t]$

# 有模型学习

- MDP的动态性如下所示：
  - 从状态$x_0$开始，智能体选择某个动作$a_0 \in A$，智能体得到奖励$R(x_0, a_0)$
  - 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

$$x_0 \xrightarrow{a_0, R(x_0,a_0)} x_1 \xrightarrow{a_1, R(x_1,a_1)} x_2 \xrightarrow{a_2, R(x_2,a_2)} x_3 \quad \cdots$$

  - 智能体的总回报为

$$R(x_0, a_0) + \gamma R(x_1, a_1) + \gamma^2 R(x_2, a_2) + \cdots$$

- 有模型学习(model-based learning)：
  - 假设$S, A, P, R$均已知
  - 方便起见，假设状态空间和动作空间均为有限的

# 有模型学习

- 目标：选择能够最大化累积奖励期望的动作

$$\mathbb{E}[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots]$$

  - $\gamma \in [0,1]$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
    - 以金融为例，今天的$1比明天的$1更有价值

- 给定一个特定的策略 $\pi(s): S \to A$，即在状态 $s$ 下采取动作 $a = \pi(s)$

- 给策略$\pi$定义价值函数：状态值函数和状态动作值函数

$$V^{\pi}(s) = \mathbb{E}[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots | s_0 = s]$$

$$Q(s,a) = \mathbb{E}[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots | s_0 = s, a_0 = a]$$

# 策略评估

- 价值函数

$$V^{\pi}(s) = \mathbb{E}[R(s_0) + \underbrace{\gamma R(s_1) + \gamma^2 R(s_2) + \cdots}_{\gamma V^{\pi}(s_1)} | s_0 = s, \pi]$$

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

立即奖励　状态转移

时间折扣　下一个状态的价值

Bellman等式

# 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

$$V^*(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s') V^*(s')$$

最优Bellman等式

- 最优策略

$$\pi^*(s) = \arg\max_{a \in A} \sum_{s' \in S} P_{sa}(s') V^*(s')$$

- 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

# 价值迭代和策略迭代

- 价值函数和策略相关

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

$$\pi(s) = \arg\max_{a \in A} \sum_{s' \in S} P_{sa}(s') V^{\pi}(s')$$

- 可以对最优价值函数和最优策略执行迭代更新
  - 价值迭代
  - 策略迭代

# 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

1. 对每个状态$s$，初始化 $V(s)=0$
2. 重复以下过程直到收敛 {

对每个状态，更新

$$V(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s') V(s')$$

}

# 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

1. 随机初始化策略 $\pi$
2. 重复以下过程直到收敛 {

a) 让$V := V^{\pi}$

b) 对每个状态，更新

$$\pi(s) = arg \max_{a \in A} \sum_{s' \in S} P_{sa}(s')V(s')$$

}

# 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）

# 大纲

- 强化学习简介
- 多臂老虎机
- 有模型学习
- 免模型学习
- 模仿学习
- 深度强化学习

# 免模型强化学习(Model-free RL)

- 在现实问题中，通常没有明确地给出状态转移和奖励函数
  - 例如，我们仅能观察到部分片段（episodes）

Episode 1: $s_0^{(1)} \xrightarrow[R(s_0)^{(1)}]{a_0^{(1)}} s_1^{(1)} \xrightarrow[R(s_1)^{(1)}]{a_1^{(1)}} s_2^{(1)} \xrightarrow[R(s_2)^{(1)}]{a_2^{(1)}} s_3^{(1)} \dots s_T^{(1)}$

Episode 2: $s_0^{(2)} \xrightarrow[R(s_0)^{(2)}]{a_0^{(2)}} s_1^{(2)} \xrightarrow[R(s_1)^{(2)}]{a_1^{(2)}} s_2^{(2)} \xrightarrow[R(s_2)^{(2)}]{a_2^{(2)}} s_3^{(2)} \dots s_T^{(2)}$

- 模型无关的强化学习直接从经验中学习值（value）和策略（policy），而无需构建马尔可夫决策过程模型（MDP）
- 关键步骤：（1）估计值函数；（2）优化策略

# 免模型强化学习(Model-free RL)

- 在有模型的强化学习中，值函数能够通过动态规划计算获得

$$V^{\pi}(s) = \mathbb{E}[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots | s_0 = s, \pi]$$
$$= R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s') V^{\pi}(s')$$

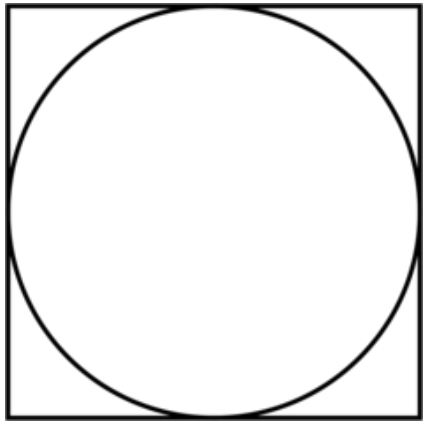
- 在免模型的强化学习中：
  - 我们无法直接获得 $P_{sa}$ 和 $R$
  - 但是，我们拥有一系列采样轨迹

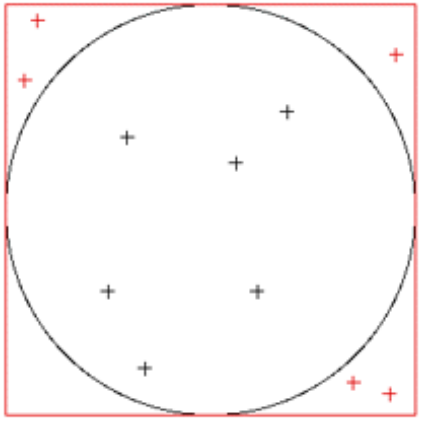
Episode 1: $s_0^{(1)} \xrightarrow[R(s_0)^{(1)}]{a_0^{(1)}} s_1^{(1)} \xrightarrow[R(s_1)^{(1)}]{a_1^{(1)}} s_2^{(1)} \xrightarrow[R(s_2)^{(1)}]{a_2^{(1)}} s_3^{(1)} \ldots s_T^{(1)}$

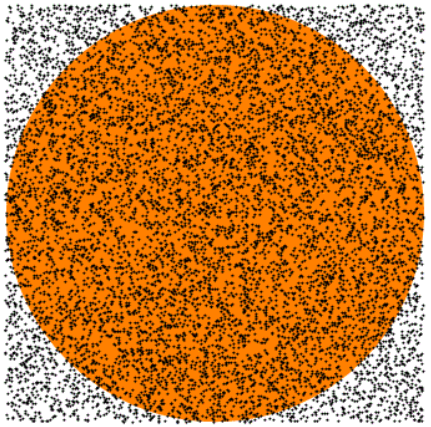
Episode 2: $s_0^{(2)} \xrightarrow[R(s_0)^{(2)}]{a_0^{(2)}} s_1^{(2)} \xrightarrow[R(s_1)^{(2)}]{a_1^{(2)}} s_2^{(2)} \xrightarrow[R(s_2)^{(2)}]{a_2^{(2)}} s_3^{(2)} \ldots s_T^{(2)}$

# 蒙特卡洛方法

- 蒙特卡洛方法（Monte-Carlo methods）是一类广泛的计算算法。
  - 依赖于重复随机抽样来获得数值结果
- 例如，计算圆的面积



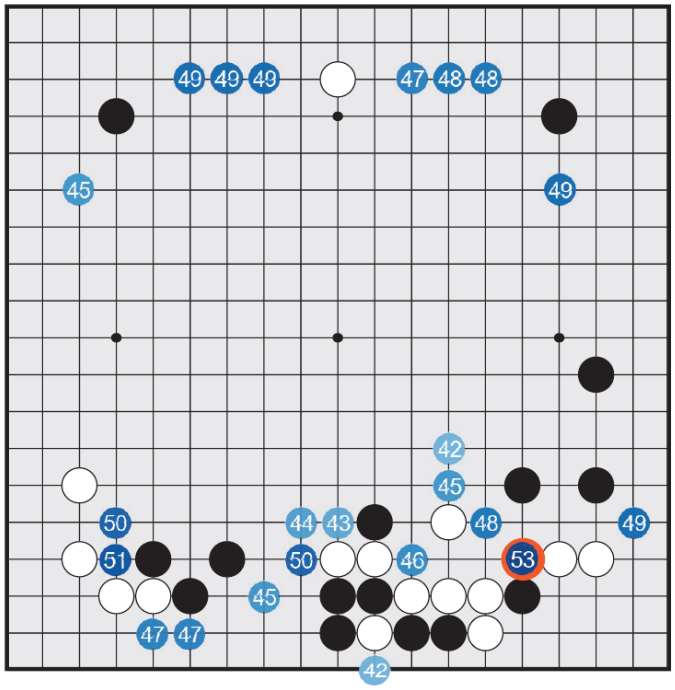


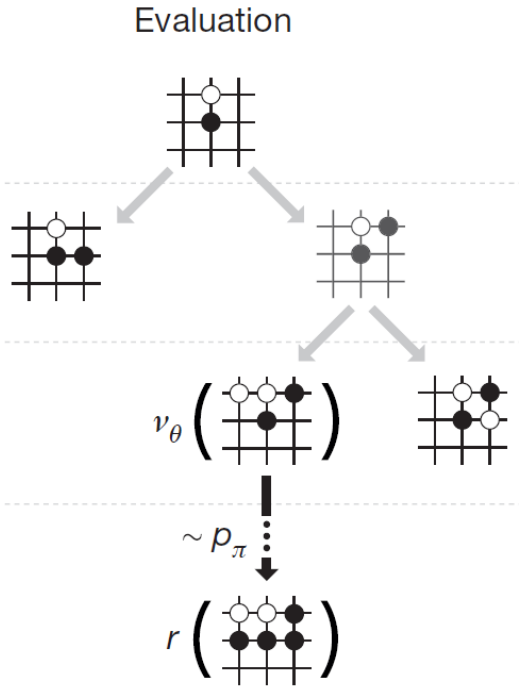


$$\text{Circle Surface} = \text{Square Surface} \times \frac{\#\text{points in circle}}{\#\text{points in total}}$$

# 蒙特卡洛方法

- 围棋对弈：估计当前状态下的胜率




$$\text{Win Rate}(s) = \frac{\#\text{win simulation cases started from } s}{\#\text{simulation cases started from } s \text{ in total}}$$

# 蒙特卡蒙特卡洛价值估计洛方法

- 目标：从策略$\pi$下的经验片段学习 $V^{\pi}$

$$s_0^{(i)} \xrightarrow[R_1^{(i)}]{a_0^{(i)}} s_1^{(i)} \xrightarrow[R_2^{(i)}]{a_1^{(i)}} s_2^{(i)} \xrightarrow[R_3^{(i)}]{a_2^{(i)}} s_3^{(i)} \ldots s_T^{(i)} \sim \pi$$

- 累计奖励是总折扣奖励

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots \gamma^{T-1} R_T$$

- 值函数是期望累计奖励

$$\begin{aligned} V^{\pi}(s) &= \mathbb{E}[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots | s_0 = s, \pi] \\ &= \mathbb{E}[G_t | s_t = s, \pi] \\ &\simeq \frac{1}{N} \sum_{i=1}^{N} G_t^{(i)} \end{aligned}$$

  - 使用策略$\pi$从状态$s$采样$N$个片段
  - 计算平均累计奖励

- 蒙特卡洛策略评估使用经验均值累计奖励而不是期望累计奖励

# 蒙特卡洛价值估计方法

- 实现：使用策略$\pi$采样片段

$$s_0^{(i)} \xrightarrow[R_1^{(i)}]{a_0^{(i)}} s_1^{(i)} \xrightarrow[R_2^{(i)}]{a_1^{(i)}} s_2^{(i)} \xrightarrow[R_3^{(i)}]{a_2^{(i)}} s_3^{(i)} \dots s_T^{(i)} \sim \pi$$

- 在一个片段中的每个时间步长$t$的状态都被访问
  - 增量计数器 $N(s) \leftarrow N(s) + 1$
  - 增量总累计奖励 $S(s) \leftarrow S(s) + G_t$
  - 价值被估计为累计奖励的均值 $V(s) = S(s)/N(s)$
  - 由大数定率有

$$V(s) \rightarrow V^{\pi}(s) \text{ as } N(s) \rightarrow \infty$$

# 时序差分学习

- 蒙特卡罗强化学习的缺点：低效
  - 求平均时以“批处理式”进行
  - 在一个完整的采样轨迹完成后才对状态-动作值函数进行更新

- 克服缺点的办法：
  - 时序差分 (temporal difference，TD) 学习

# 增量蒙特卡洛更新

- 每个片段结束后逐步更新$V(s)$

- 对于每个状态$S_t$ 和对应累计奖励$G_t$

$$N(S_t) \leftarrow N(S_t) + 1$$
$$V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}\big(G_t - V(S_t)\big)$$

- 对于非稳定的问题（即，环境会随时间发生变化），我们可以跟踪一个现阶段的平均值（即，不考虑过久之前的片段）

$$V(S_t) \leftarrow V(S_t) + \alpha\big(G_t - V(S_t)\big)$$

# 时序差分学习

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = R_{t+1} + \gamma V(S_{t+1})$$

$$V(S_t) \leftarrow V(S_t) + \alpha\big(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\big)$$

观测值　　对未来的猜测

- 时序差分方法直接从经验片段中进行学习
- 时序差分更新当前预测值使之接近估计累计奖励（非真实值）：

  $R_{t+1} + \gamma V(S_{t+1})$

# 时序差分学习

- 从知道什么是好的：估计$V^{\pi}(S_t)$

$$V(S_t) \leftarrow V(S_t) + \alpha\big(G_t - V(S_t)\big)$$

$$V(S_t) \leftarrow V(S_t) + \alpha\big(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\big)$$

- 基于$V$函数，如何做好行动？

$$\pi(s) = \arg\max_{a \in A} \sum_{s' \in S} P_{sa}(s')V(s')$$

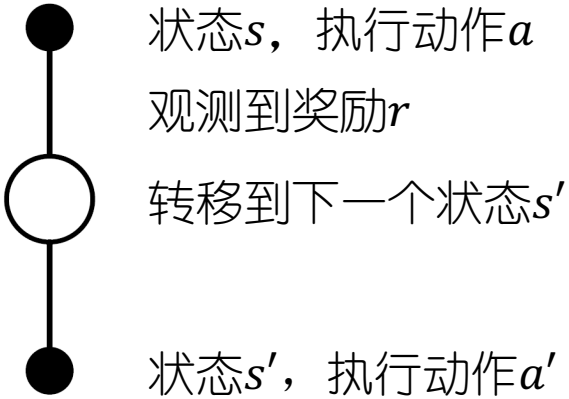
需要知道环境模型

- 基于$Q$函数如何选择好的行动？

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

# SARSA

- 对于当前策略执行的每个（状态–动作–奖励–状态–动作）元组

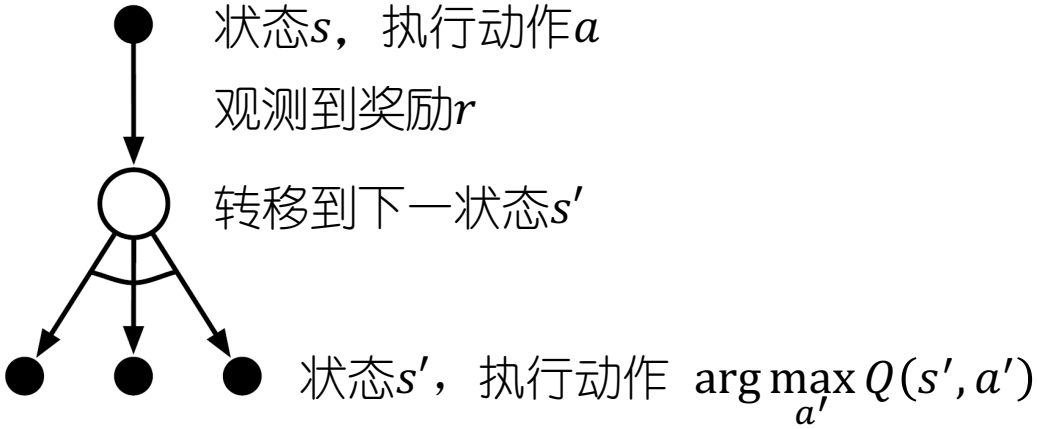



- SARSA更新状态–动作值函数为

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma Q(s',a') - Q(s,a))$$

# Q-Learning

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

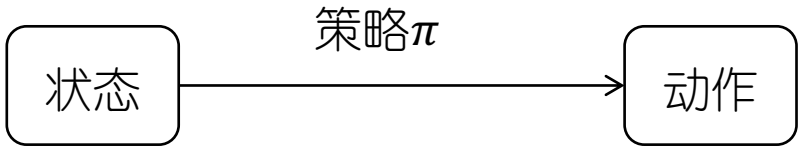


# 大纲

- 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

# 模仿学习(Imitation Learning)

- 模仿学习：直接模仿人类专家的状态-动作对来学习策略
  - 相当于告诉机器在什么状态下应该选择什么动作
  - 引入了监督信息来学习策略

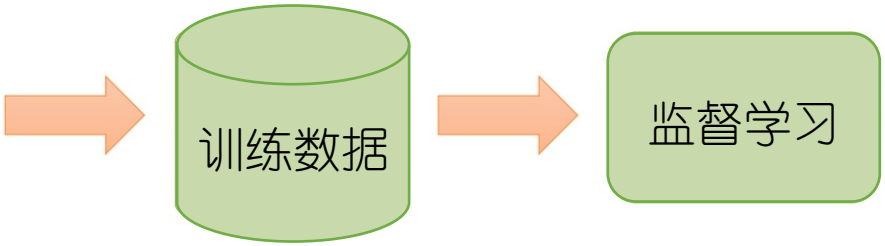


# 模仿学习(Imitation Learning)

- 直接模仿学习
  - 利用专家的决策轨迹，构造训练数据：状态作为特征，动作作为标记
  - 利用数据集，使用分类/回归算法即可学得策略
  - 将学得的策略作为初始策略
  - 策略改进，从而获得更好的策略

$\mathbf{o}_t$

$\mathbf{a}_t$




$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

# 模仿学习(Imitation Learning)

- 强化学习任务中，设计合理的符合应用场景的奖赏函数往往相当困难
- 缓解方法：从人类专家提供的范例数据中反推出奖赏函数
- 逆强化学习 (inverse reinforcement learning)
  - 基本思想：寻找某种奖赏函数使得范例数据是最优的，然后即可使用这个奖赏函数来训练策略

# 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

# 深度强化学习

AI = Deep Learning + Reinforcement Learning


Deep Reinforcement Learning

# 深度强化学习

$Q_\theta(s,a)$

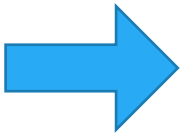
| | $a_1$ | $a_2$ |
|---|---|---|
| $s_1$ | 3.2 | 4.1 |
| $s_2$ | 0.8 | 1.2 |
| $s_3$ | 2.3 | -0.3 |

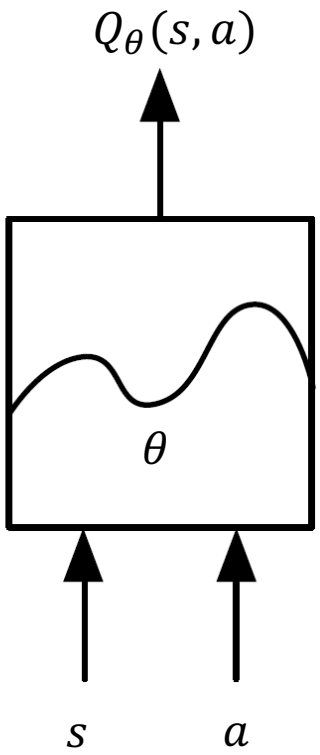
价值函数表格

$\pi_\theta(a|s)$

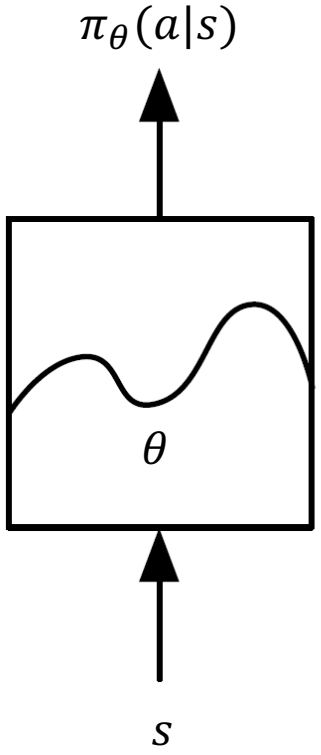
| | $a_1$ | $a_2$ |
|---|---|---|
| $s_1$ | 0.2 | 0.8 |
| $s_2$ | 0.4 | 0.6 |
| $s_3$ | 0.9 | 0.1 |

策略表格






价值函数近似




策略近似

- 假如我们直接使用深度神经网络建立这些近似函数呢？
- 深度强化学习！

# 深度强化学习

- 2012年AlexNet在ImageNet比赛中大幅度领先对手获得冠军
- 2013年12月，第一篇深度强化学习论文出自NIPS 2013 Reinforcement Learning Workshop

**Playing Atari with Deep Reinforcement Learning**

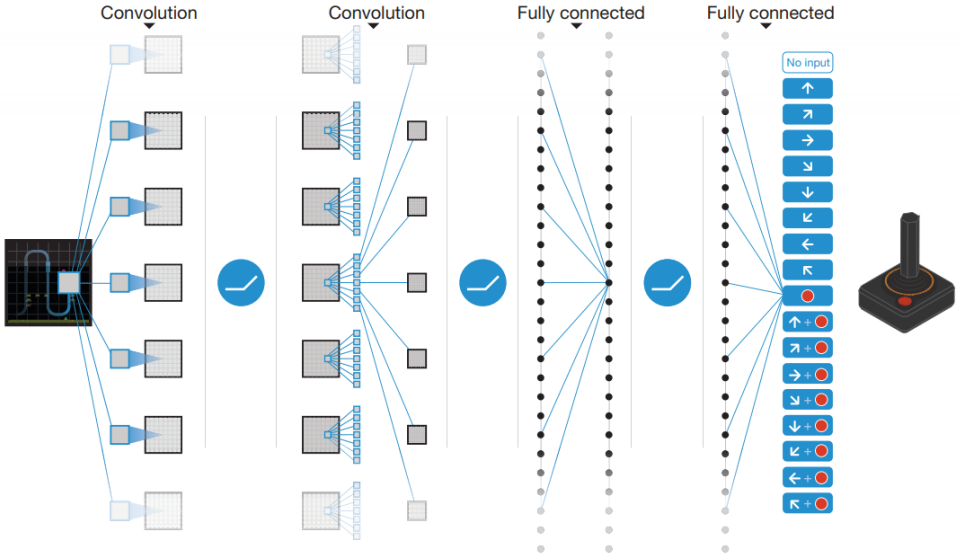
**Volodymyr Mnih** **Koray Kavukcuoglu** **David Silver** **Alex Graves** **Ioannis Antonoglou**

**Daan Wierstra** **Martin Riedmiller**

DeepMind Technologies

{vlad,koray,david,alex.graves,ioannis,daan,martin.riedmiller} @ deepmind.com

# 深度强化学习

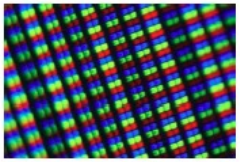

$$\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}_{s,a\sim\rho(\cdot);s'\sim\mathcal{E}}\left[\left(r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) - Q(s, a; \theta_i)\right) \nabla_{\theta_i} Q(s, a; \theta_i)\right]$$

Q函数的参数通过神经网络反向传播学习

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

# 深度强化学习

标准（传统）
计算机视觉



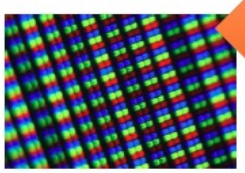
features
(e.g. HOG)

mid-level features
(e.g. DPM)

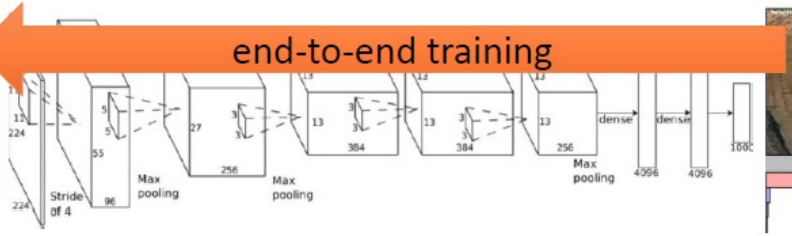
Felzenszwalb '08

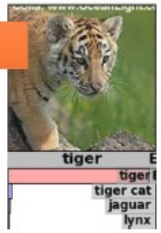
classifier
(e.g. SVM)

深度学习

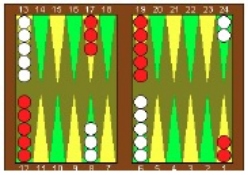


end-to-end training

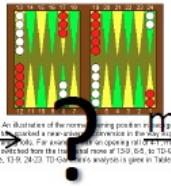



标准（传统）
强化学习



features



?

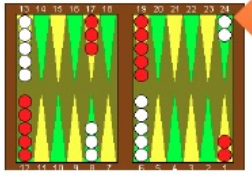
more features

?

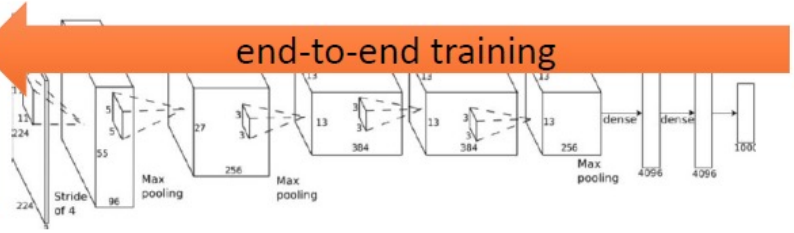
linear policy
or value func.

action

深度强化学习

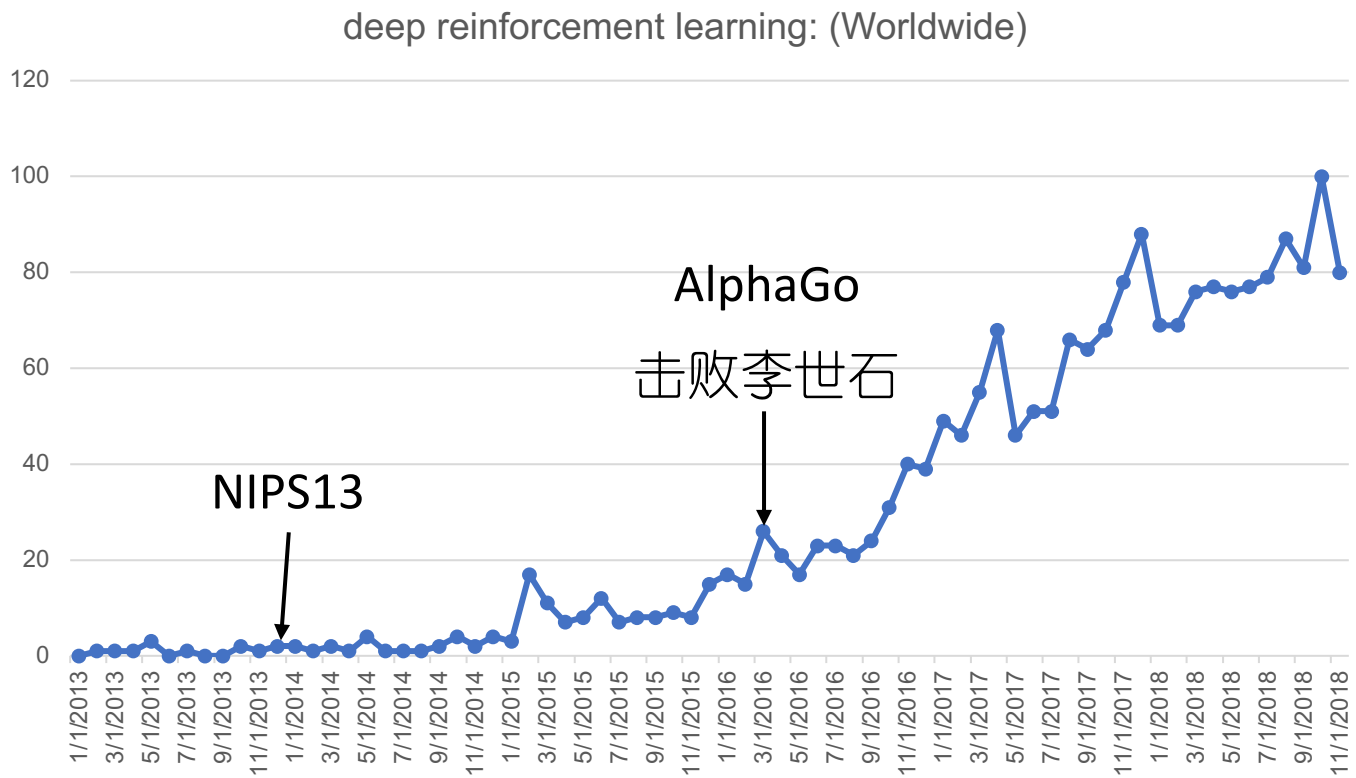


end-to-end training




action

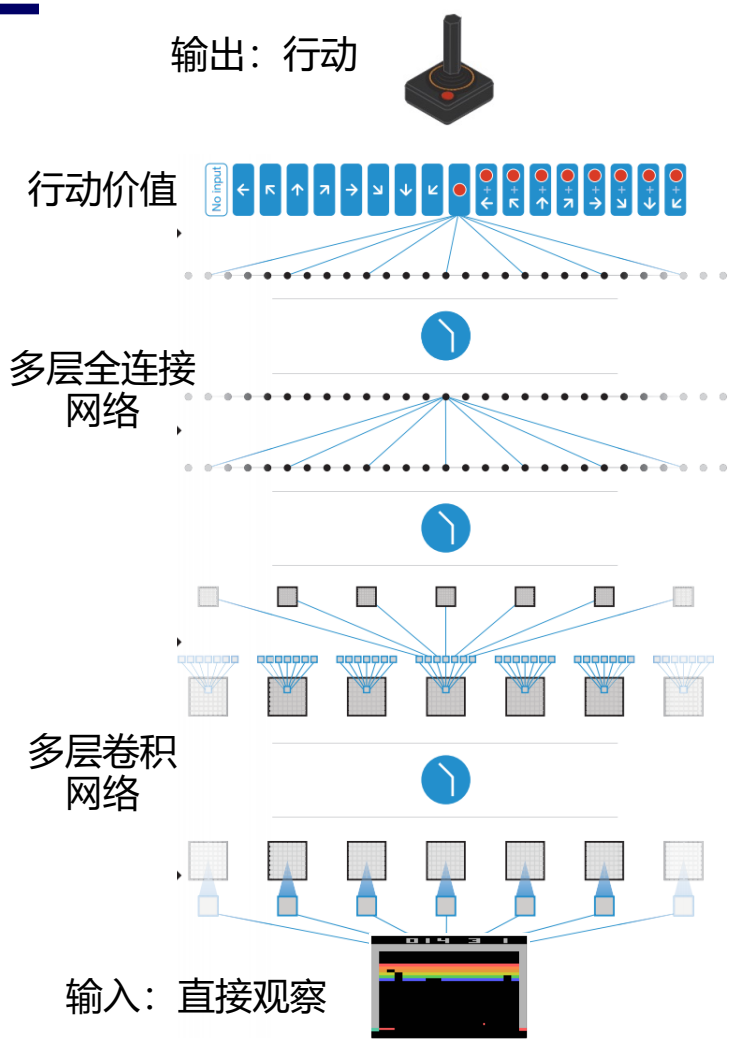
# 深度强化学习的趋势




Google搜索中词条“深度强化学习（deep reinforcement learning）”的趋势

# 深度强化学习

- 将深度学习（DL）和强化学习（RL）结合在一起会发生什么？
  - 价值函数和策略变成了深度神经网络
  - 相当高维的参数空间
  - 难以稳定地训练
  - 容易过拟合
  - 需要大量的数据
  - 需要高性能计算
  - CPU（用于收集经验数据）和GPU（用于训练神经网络）之间的平衡
  - …
- 这些新的问题促进着深度强化学习算法的创新



# 深度强化学习的研究前沿


基于模拟模型的强化学习

- 模拟器的无比重要性

目标策动的层次化强化学习

- 长程任务的中间目标是桥梁的基石

模仿学习

- 无奖励信号下跟随专家做策略学习

多智能体强化学习

- 分散式、去中心化的人工智能

# 深度强化学习的落地场景

- 无人驾驶
- 游戏AI
- 交通灯调度
- 网约车派单
- 组合优化
- 推荐搜索系统
- 数据中心节能优化
- 对话系统
- 机器人控制
- 路由选路
- 工业互联网场景
- ...

# 小结

- 强化学习：序列决策任务，两个特点：延迟反馈、agent的动作会影响环境
- 有模型学习：MDP模型已知
- 无模型学习：利用采样去评估价值函数和价值动作函数，了解SARSA和Q-Learning的基本思想
- 模仿学习：模仿人类专家的“状态-动作对”，或者基于人类专家数据反推奖赏函数
- 深度强化学习：利用神经网络建模值函数