

第五章 行为学派

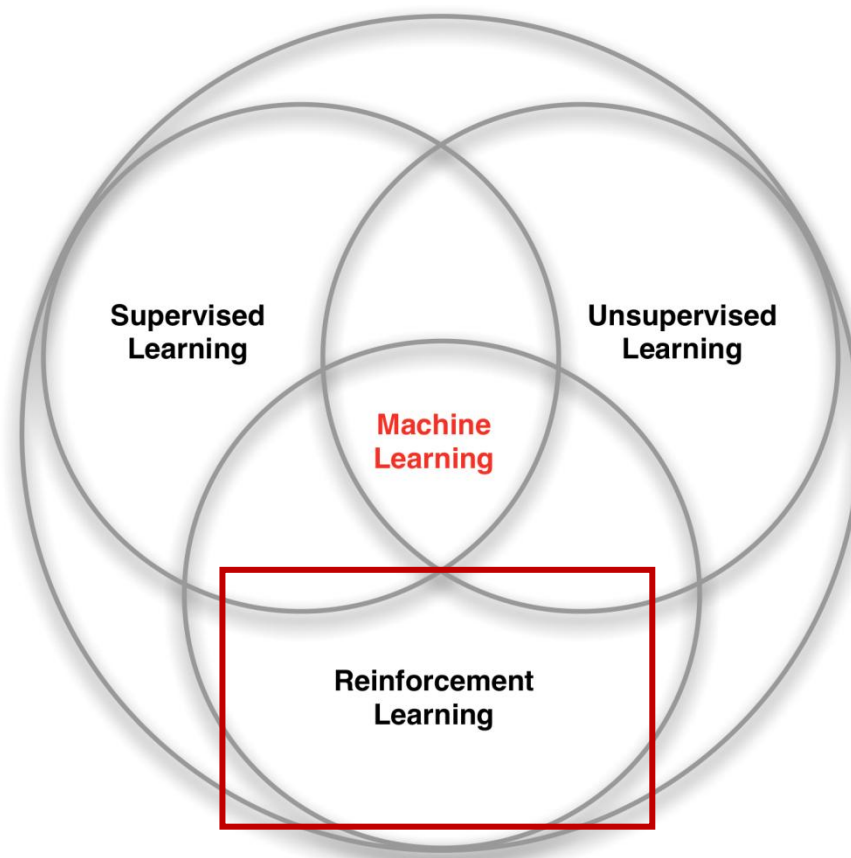
智能优化、强化学习

机器学习与强化学习

- 机器学习的分类：

监督学习
例：SVM

无监督学习
例：主元分析



机器学习与强化学习

- 强化学习（Reinforcement Learning, RL）的主体：智能体（Agent）
- 强化学习与其他机器学习范式的区别：
 - 没有监督，只有奖励信号
 - 延迟反馈，而非瞬时结果
 - 智能体与环境的互动（动态特性）
 - 机器的动作影响了它接下来获取的数据
 - 时序的重要作用（使用序列训练数据，而非独立同分布数据）

强化学习

- 强化学习的机制由来
- 基本模型和核心概念
- 强化学习的若干关键问题
- 经典强化学习：Q学习
- 深度强化学习
 - 深度Q网络
 - 其他方法

动物学习理论

最优控制理论



强化学习

动物学习实验

- 效果法则 (Law of Effect)

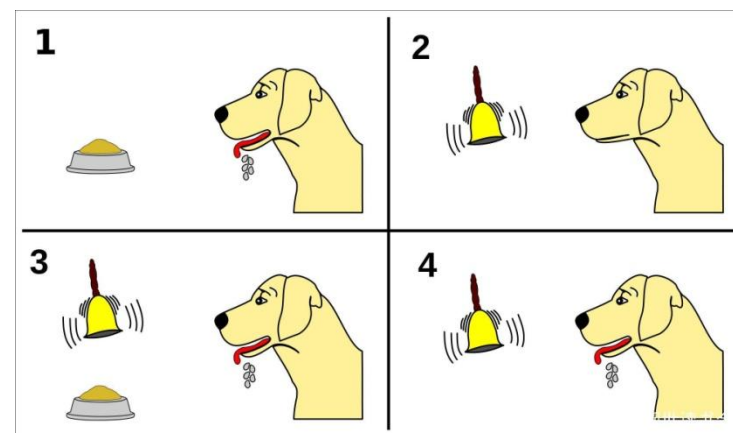
- “在其他条件相同的情况下，对同一情况作出的几种反应中，如果伴随着动物意志的满足感发生在其中某些反应的期间，或者紧随其后，那么这些反应将与该情况建立更牢固的联系，因此，当这种情况再次发生时，这些反应发生的可能性会提高；如果在其他条件相同的情况下，动物意志的不适感发生在某些反应期间，或者紧随其后，那么这些反应与该情况的联系将被削弱，因此，当这种情况再次发生时，这些反应发生的可能性会降低。满足感或不适感的程度越大，联系加强或削弱的程度就越大。” [桑代克, 1911, 《动物智慧》]

行为主义心理学的主要原理

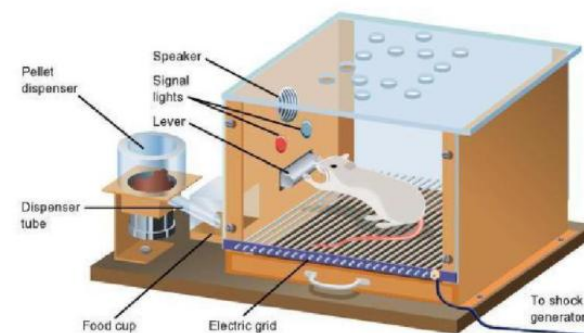
动物学习实验

- 经典的动物（包含人）**条件反射**。
“条件响应(conditioned response)的幅度和时效，会随着条件刺激和非条件刺激之间的偶然性(contingency)发生**变化**” [巴甫洛夫, 1927年]
- 操作条件反射**（或工具性条件反射）：人类和动物学习行为以获得奖励（obtain **rewards**）和避免惩罚（avoid **punishments**）的过程 [斯金纳, 1938]。

Remark: 强化指任何形式的条件反射，既可以是正面的（奖励）也可以是负面的（惩罚）



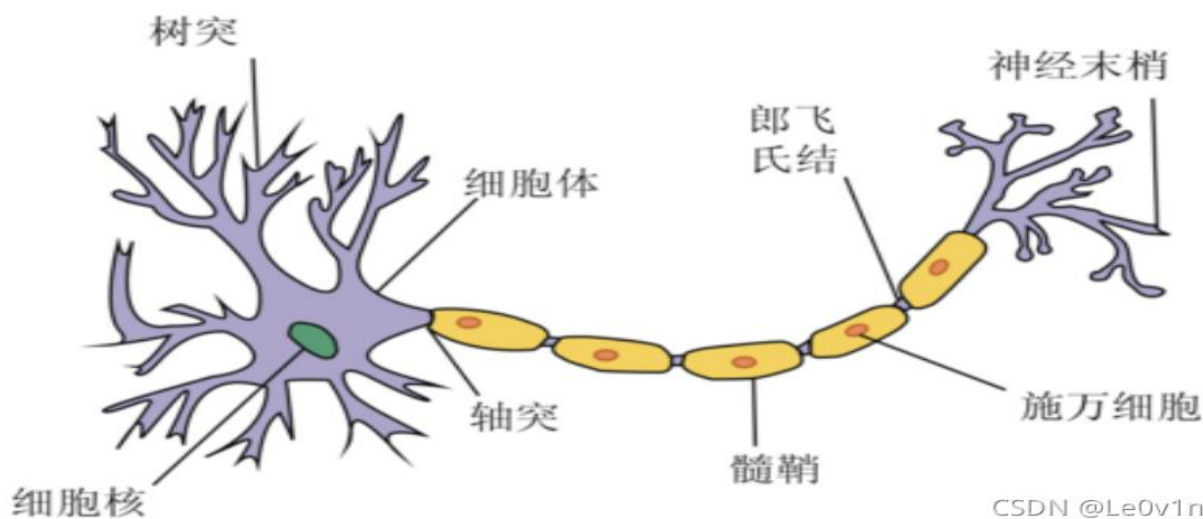
斯金纳的研究



行为主义心理学的主要原理

计算神经科学

- **赫布（Hebbian）学习**：通过共同激活神经元，来强化它们之间的突触权重，从而发展模型的形式。"如果先激活一个神经元，然后马上激活另一个神经元，它们就会连在一起"。[赫布, 1961]。

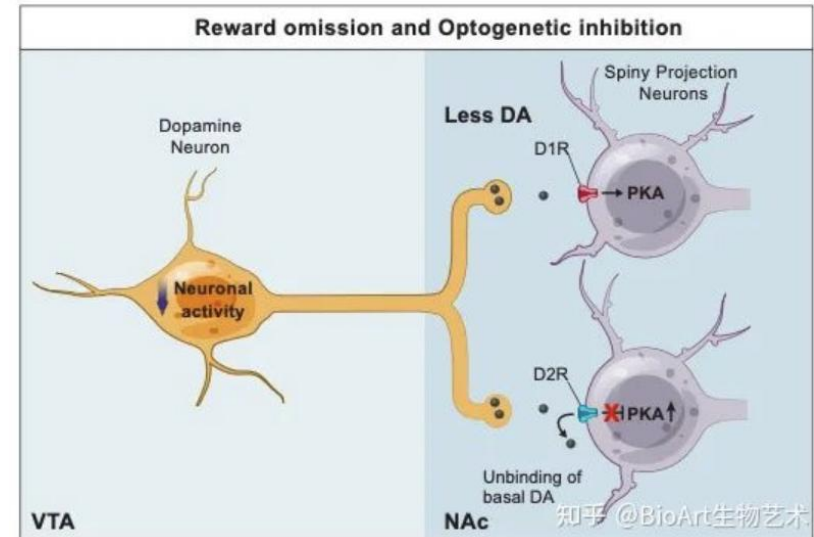
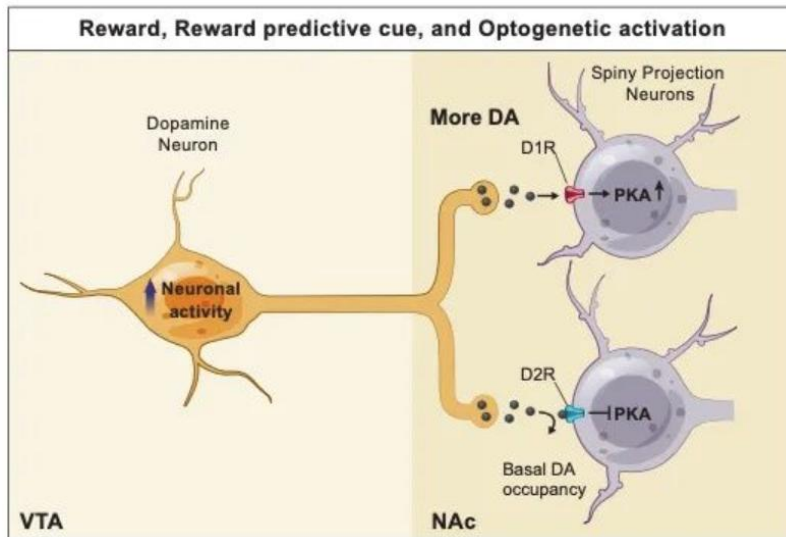


CSDN @Le0v1n

计算神经科学

- 多巴胺和基底核模型：与运动控制和决策有直接联系
[铜谷贤治, 1999]

– Remark: 强化代表了多巴胺（和惊喜）的作用。



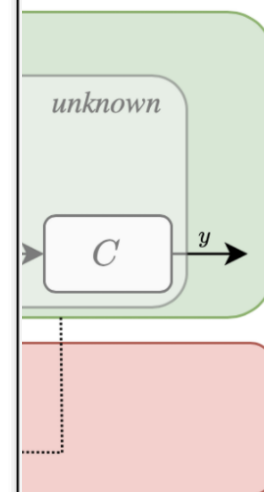
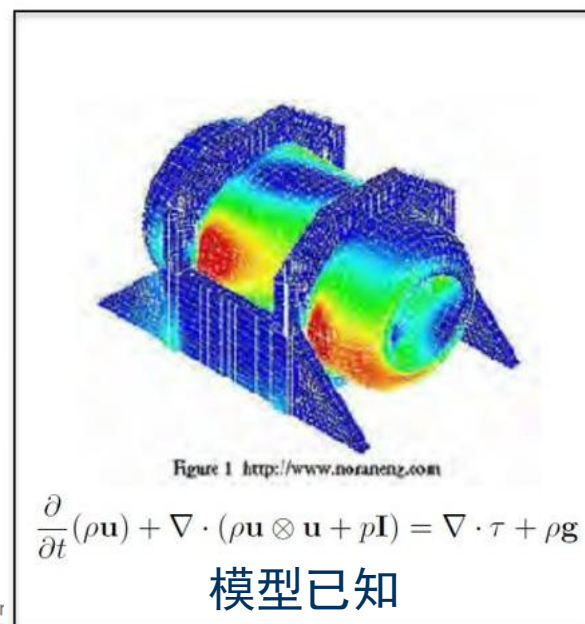
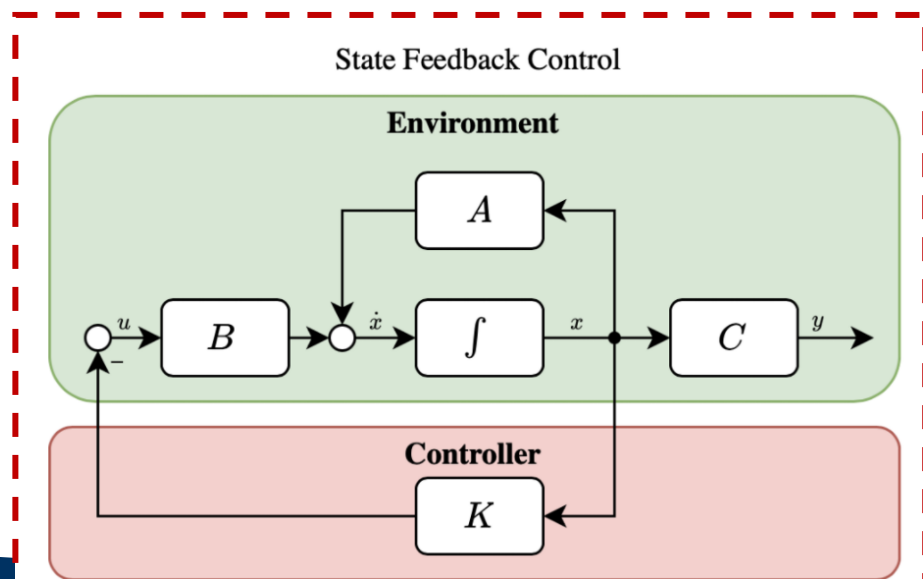
计算神经科学

- **情绪理论**：关于情感过程如何影响决策过程的模型 [达马西奥，1994]。



最优控制与强化学习

- **最优控制**：以优化方法的形式框架，求取连续时间控制问题中的最优控制策略。
 - 假设模型已知（Model-based，如左图）
 - **动态规划**是求解最优控制问题的一种经典方法

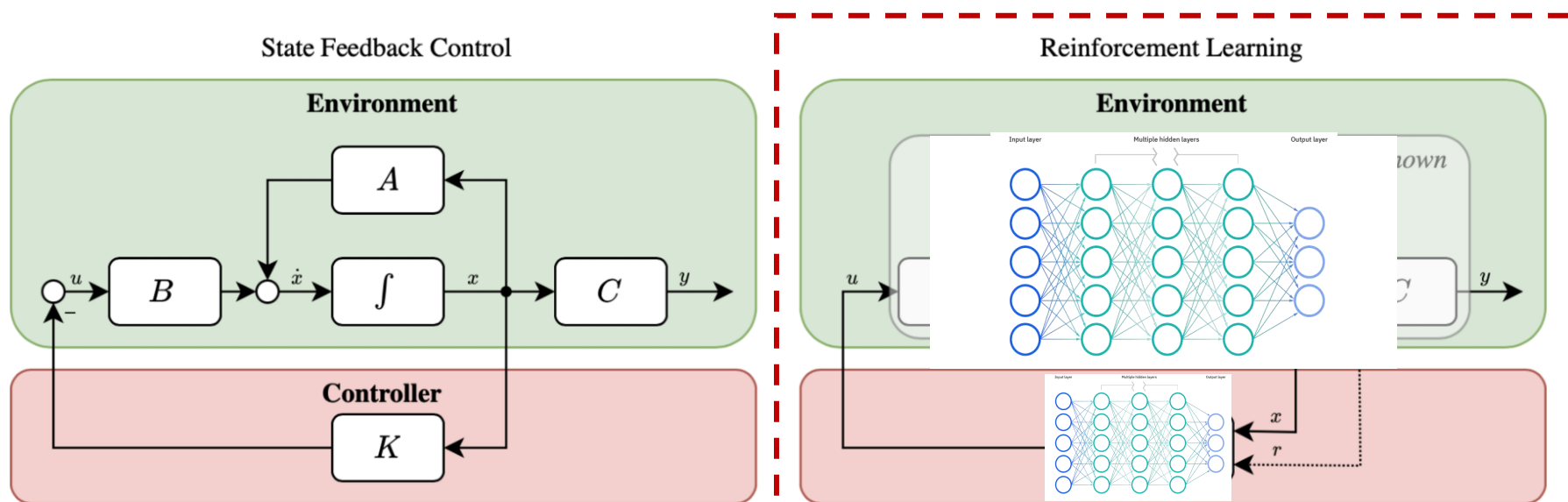


Classic control is more similar to reinforcement learning than it is, that in control we assume to know the underlying system dynamics, whereas in reinforcement we do not.

(Image by author)

最优控制与强化学习

- **强化学习**：通过与未知和不确定（如随机）环境的直接交互（试错）学习一种行为策略，使长期的奖励总和（延迟奖励）最大化。
 - 模型通常未知（Model-free，如右图）



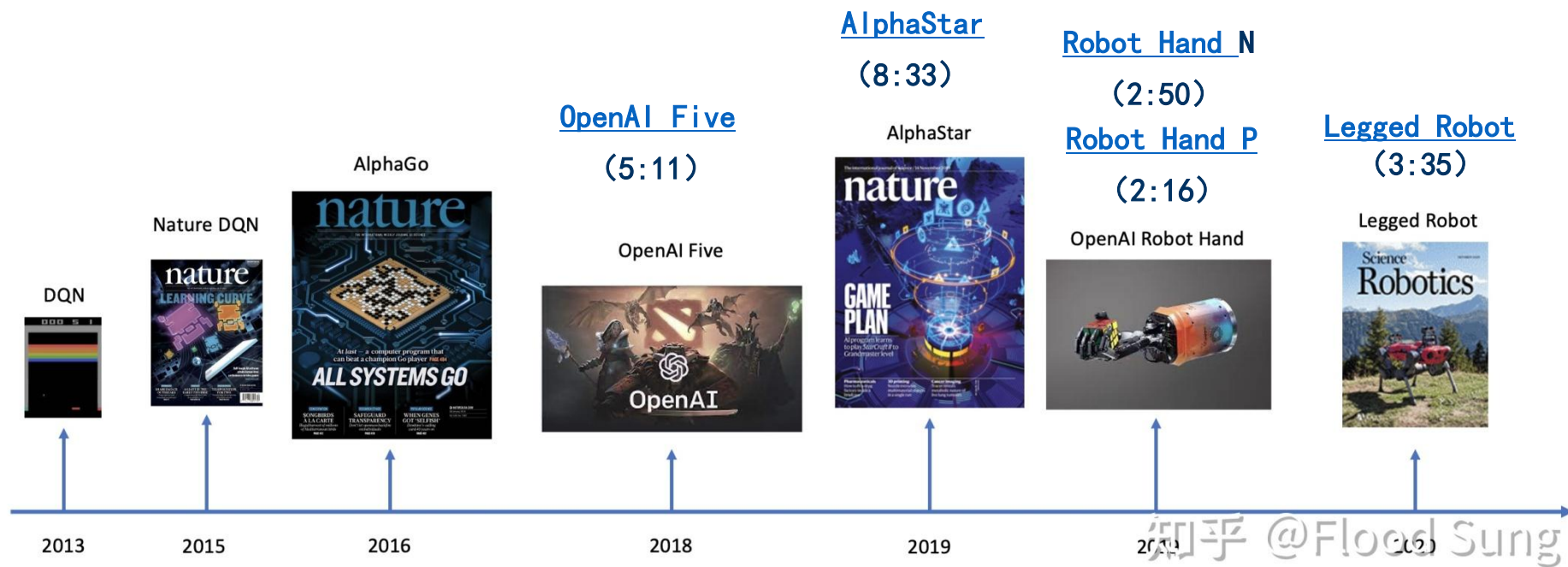
Classic control is more similar to reinforcement learning than one might think. However, the key distinction between them is, that in control we assume to know the underlying system dynamics, whereas in reinforcement we do not.

(Image by author)

强化学习的过去*

- 计算机下棋程序 [香农, 1950 (论文1988)].
- 神经模拟强化系统理论 [明斯基, 1954].
- 利用跳棋游戏进行机器学习的研究 [Samuel, 1959].
- 试错 (井字棋) [米基, 1961].
- 自适应控制实验 (单连杆倒立摆) [米基 和 钱伯斯, 1968].
- 惩罚/奖励: 在自适应阈值系统中与批判者一起学习 (神经网络) [威德罗 等, 1973].
- 联想搜索网络, 强化学习联想记忆 [Barto等, 1981].
- 强化学习中的时间分数分配" (时差学习) [Sutton, 1984].
- 延迟奖励的学习 (Q学习) [Watkins, 1989].
- 时间差分法与TD-Gammon [Tesauro, 1995].

现代强化学习研究里程碑



(深度) 强化学习能做什么？

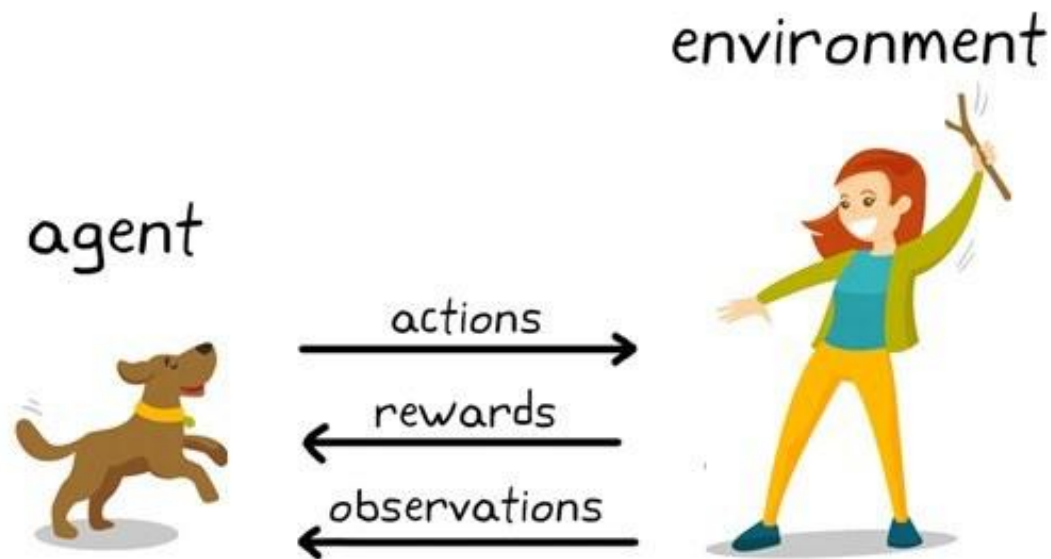
强化学习

- 强化学习的机制由来
- 基本模型和核心概念
- 强化学习的若干关键问题
- 经典强化学习：Q学习
- 深度强化学习
 - 深度Q网络
 - 其他方法

强化学习要素与基本模型

- 智能体与环境

- 强化学习要素



- 环境
- 智能体
- 状态 s
- 动作 a
- 奖励 r

智能体与环境的交互



◆在 t 时刻，智能体：

- 收到观察 O_t
- 收到标量的奖励 R_t
- 执行动作 A_t

◆环境：

- 收到动作 A_t
- 释放观察 O_{t+1}
- 释放标量奖励 R_{t+1}

$$O_1, R_1, A_1, \dots, A_{t-1}, O_t, R_t$$

要素：奖励

- 奖励 R_t 是一种标量的反馈信号；
- 反映了 t 时刻，智能体 Action 的好坏；
- 智能体的任务：最大化累积奖励（Cumulative reward）。



回报（Return） $G_t = R_{t+1} + R_{t+2} + R_{t+3} + \dots$ 吃到的鸡腿数量最大化

强化学习基于奖励假设：

➤ 所有的目标都可以描述为某种期望的累积奖励的最大化。

你同意吗？



Ref

[1] Silver David, Singh Satinder, Precup Doina, Sutton Richard S.. **Reward Is Enough**[J]. Artificial Intelligence, 2021(prepublish)

要素：奖励-例子

- 机器人运动控制：
 - 获得正向(+)奖励：跟随了预定轨迹
 - 获得负向(-)奖励：发生碰撞
- 控制发电站
 - 获得正向(+)奖励：正常发出电力
 - 获得负向(-)奖励：违反安全约束
- 玩电子游戏
 - 获得正向(+)奖励：获得高积分
 - 获得负向(-)奖励：死亡

要素：状态

- **历史 (History)**：观察 (Observation)、动作 (Action) 和奖励 (Reward) 的序列

$$H_t = O_1, R_1, A_1, \dots, A_{t-1}, O_t, R_t$$

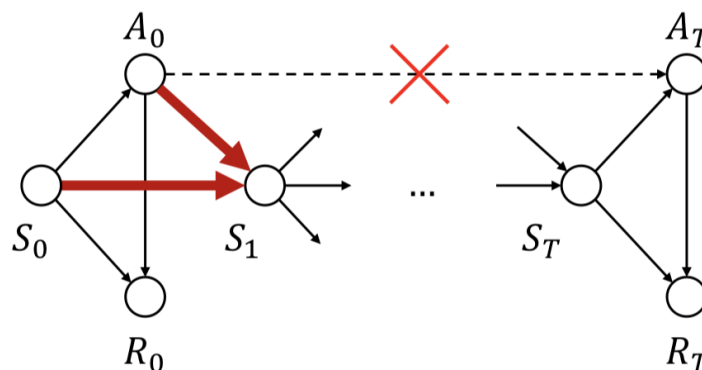
- 历史包括了截止到 t 时刻所有能观察到的变量
- 例如：机器人全部运动传感器的数据流
- 根据历史做决定 → **需要所有历史信息么？ Usually No**
- **状态 (State)**：决定下一步要做什么所需要的信息
 - 状态是历史的函数： $S_t = f(H_t)$ （即从历史中提取必要信息）

状态的马尔科夫性质

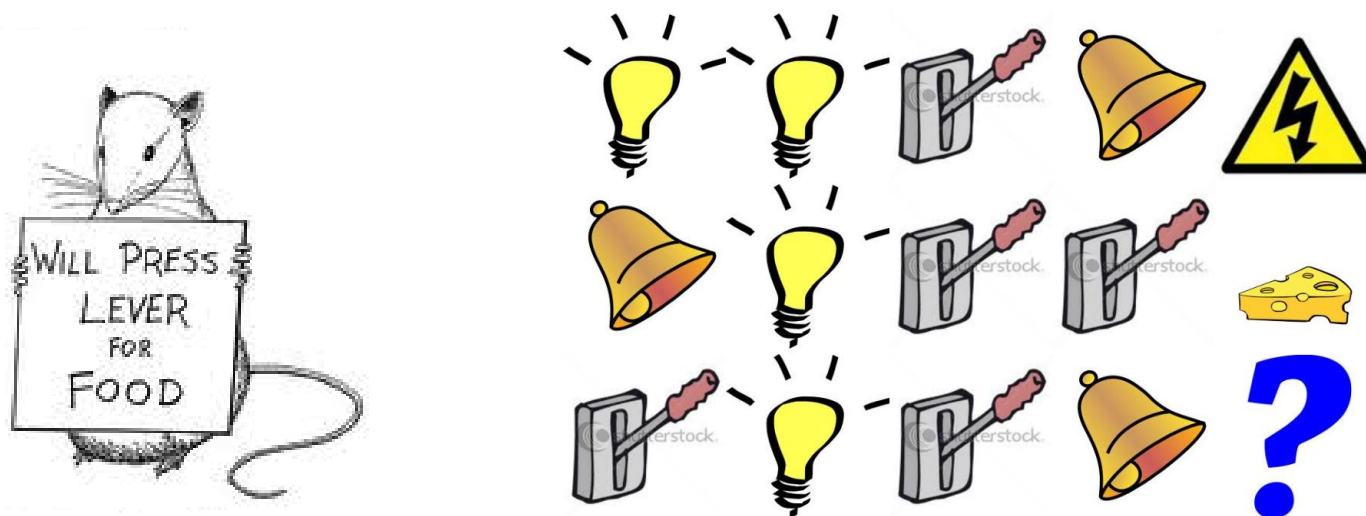
- 信息状态（Information State）：某状态包含了历史中的全部有用信息。→ 马尔科夫状态（Markov State）
- 状态 S_t 具备马尔科夫性质，当且仅当满足下式：

$$P[S_{t+1}|S_t, A_t] = P[S_{t+1}|S_t, \dots, S_1, A_t]$$
 - 过去包含在现在中，而未来由现在决定；
 - [在给定现在状态时，它与过去状态（即该过程的历史路径）是条件独立的]
 - 状态确定后，可以无需考虑历史。

- 强化学习基本模型

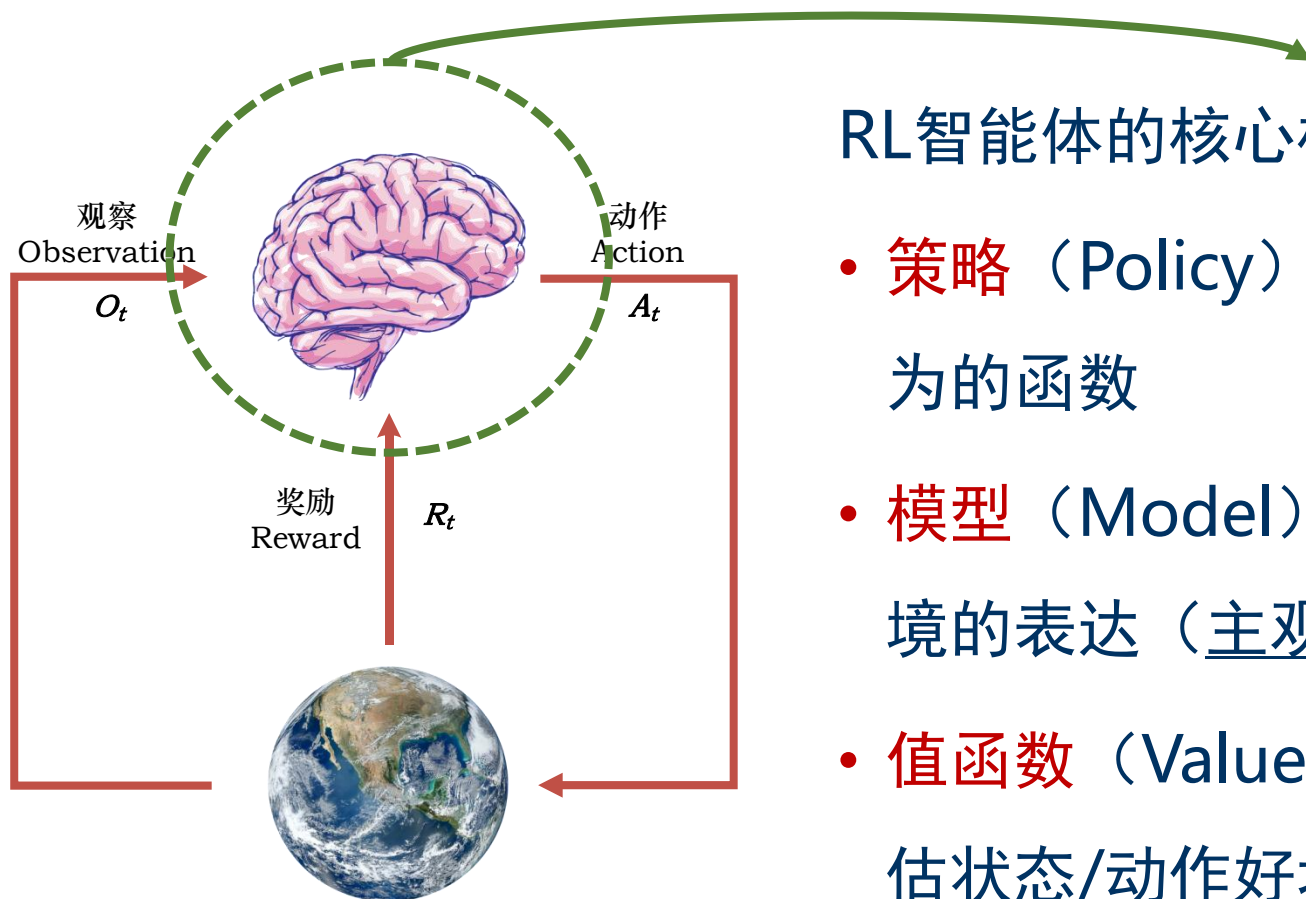


小鼠实验例子-状态的提取



- 如果智能体状态 = 最后三个信号的序列？
- 如果智能体状态 = 统计出现的灯、铃铛和拉杆的数量？
- 如果智能体状态 = 完整的序列？

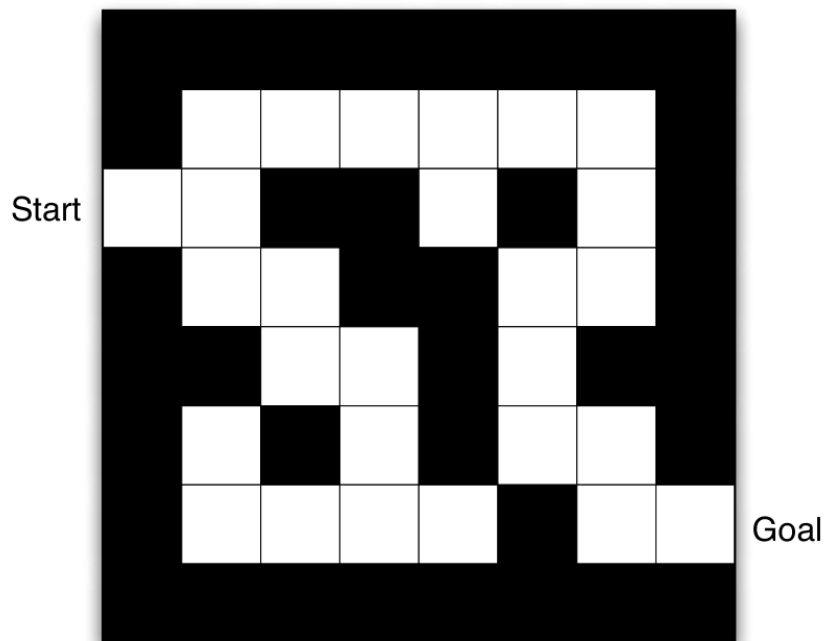
智能体学习的核心构成



RL智能体的核心构成：

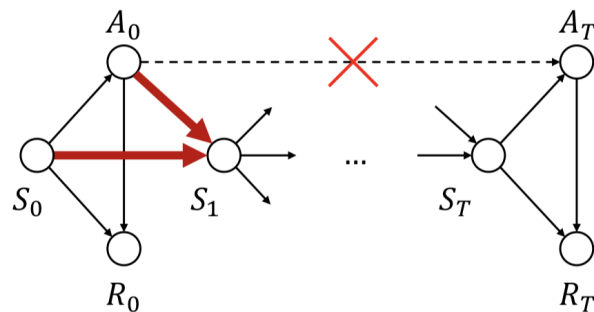
- **策略** (Policy)：决定智能体行为的函数
- **模型** (Model)：智能体对环境的表达 (主观认识)
- **值函数** (Value function)：评估状态/动作好坏的函数

以迷宫为例



- 奖励：每走一步-1
- 动作：东、南、西、北四方向
(用上、下、左、右箭头表示)
- 状态：智能体的位置

策略 (Policy)



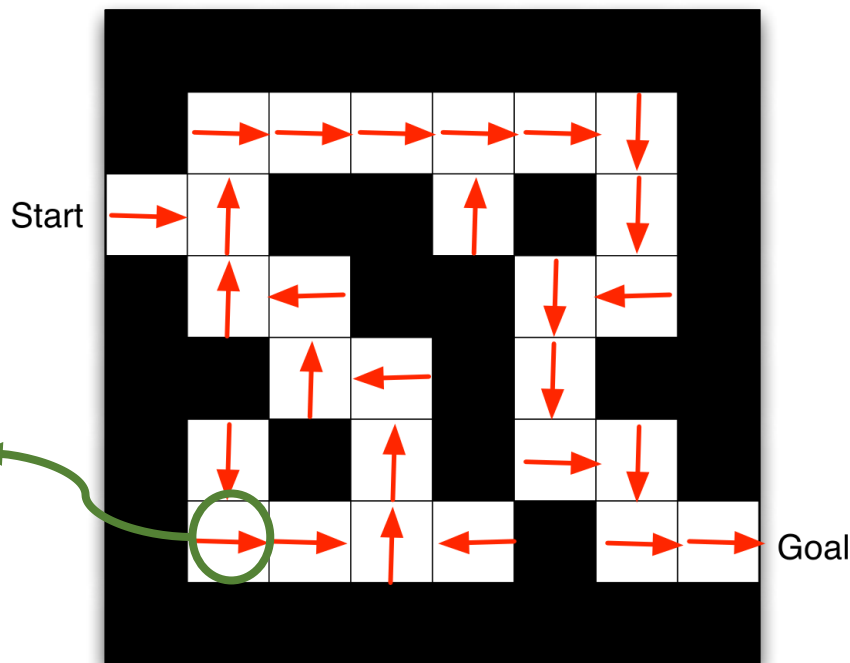
- 策略 (Policy) 代表了智能体的行为函数
- 表示从状态到动作的映射，比如：

- 确定性策略： $a = \pi(s)$
- 随机型策略（处于状态 s 时采取动作 a 的概率）

$$\pi(a|s) = P[A_t = a | S_t = s]$$

所有箭头表示在每个状态（位置）下的确定性策略 $\pi(s)$

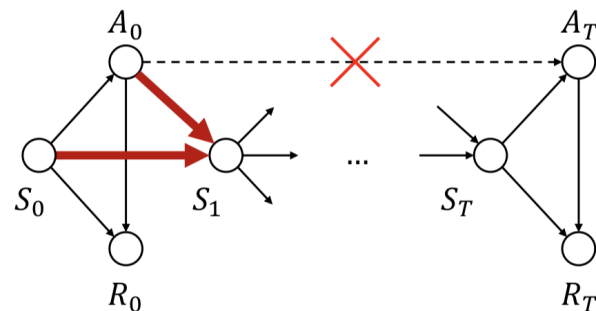
若agent处在该位置，则根据策略 $\pi(s)$ ，下一个动作是向右走



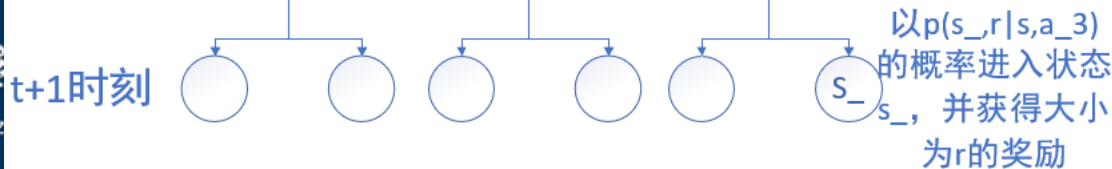
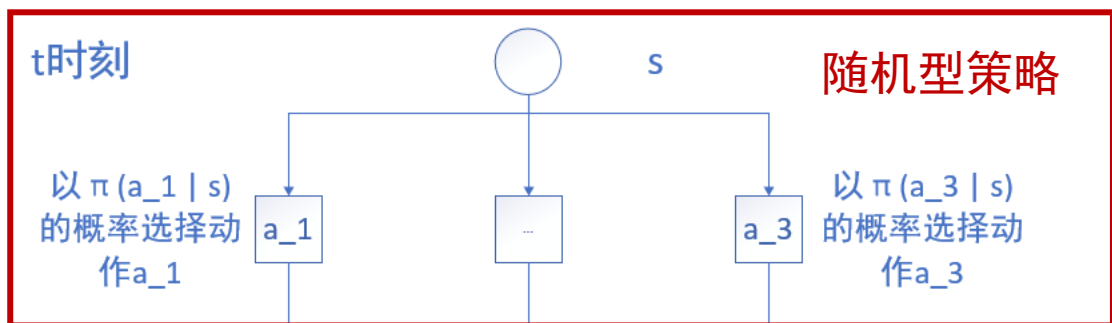
华中科技大学

HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

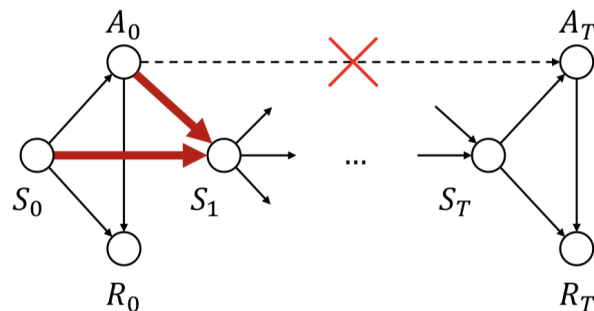
策略 (Policy)



- 策略 (Policy) 代表了智能体的行为函数
 - 表示从状态到动作的映射，比如：
 - 确定性策略: $a = \pi(s)$
 - 随机型策略 (处于状态 s 时采取动作 a 的概率)
- $$\pi(a|s) = P[A_t = a | S_t = s]$$



模型 (Model)



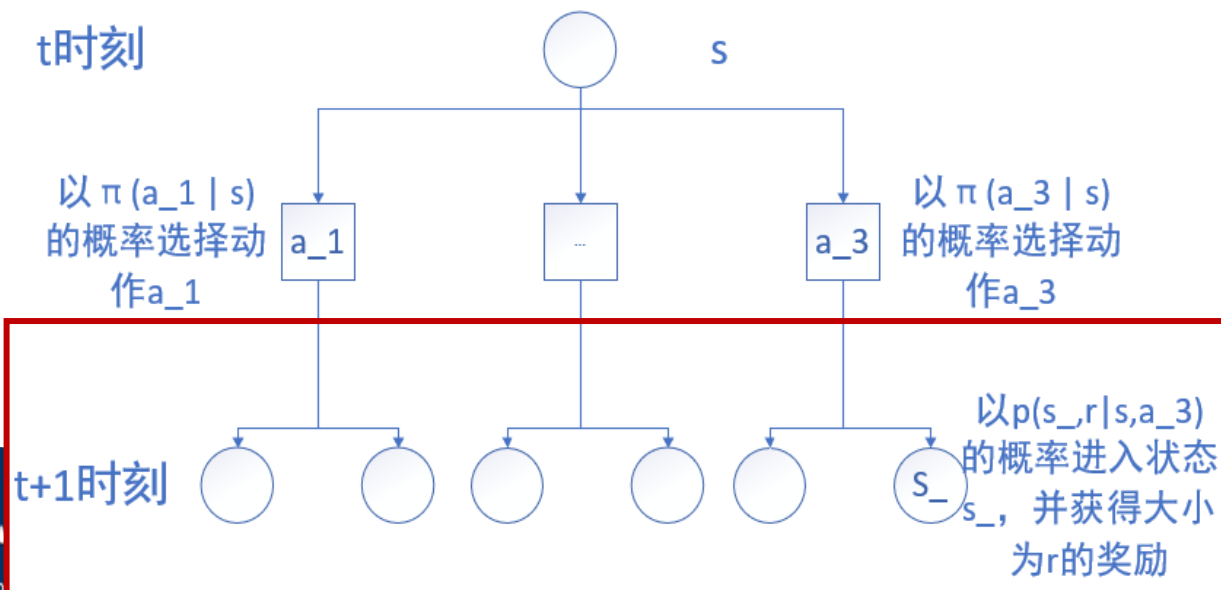
- **模型**：预测环境接下来如何变化

➤ 转移 (Transition) 模型： \mathcal{P} 预测下一个状态

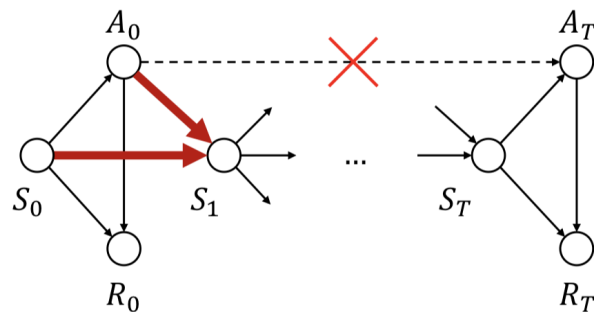
$$\mathcal{P}_{ss'}^a = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$$

➤ 奖励模型： \mathcal{R} 预测下一个瞬时奖励

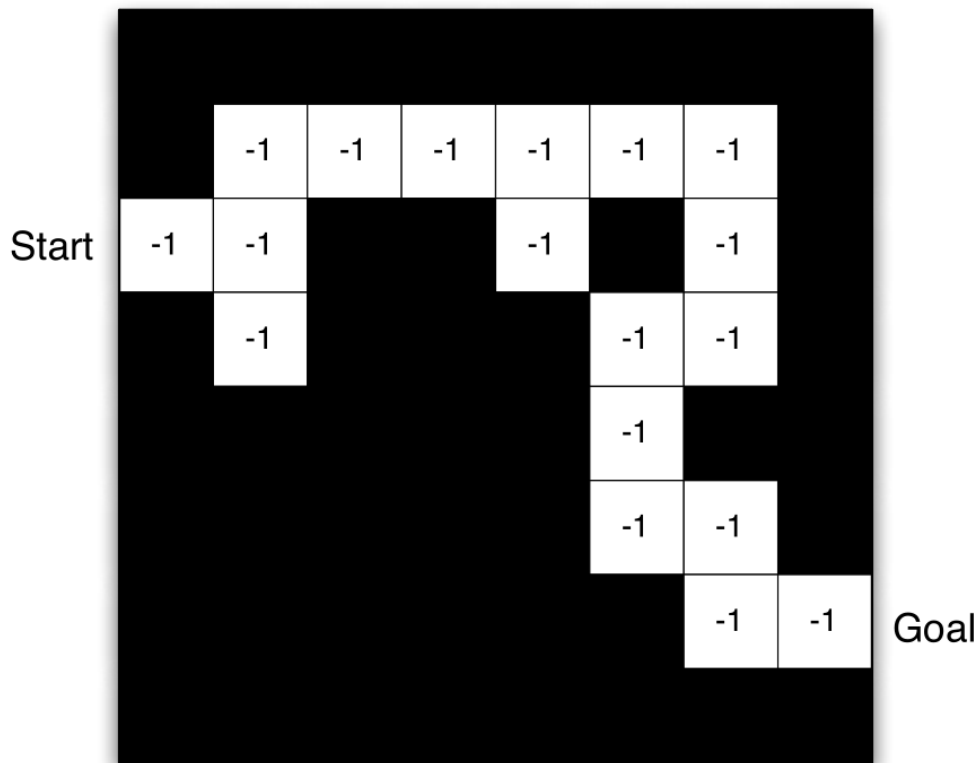
$$\mathcal{R}_s^a = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$$



模型（以迷宫为例）



- 格子的排列表示了转移模型 $\mathcal{P}_{ss'}^a$
- 数字表示了每个状态 s 对应的瞬时奖励 \mathcal{R}_s^a
 - 本例中对所有的动作 a 都相同



值函数 (Value Function)

- 值函数：对未来的回报期望 (Return)

策略 π 下agent
处于状态 s 时，
未来的回报期
望

$$V_{\pi}(s) = E_{\pi}[G_t | S_t = s]$$

$$= E_{\pi}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s]$$

依赖于策略

折扣因子

- 可用来衡量状态的好坏 (根据未来回报期望)
- 动作选择以值函数为依据 (贪婪)
 - 状态值函数: $V_{\pi}(s) = E_{\pi}[G_t | S_t = s]$
 - 动作值函数: $Q_{\pi}(s, a) = E_{\pi}[G_t | S_t = s, A_t = a]$
- RL目标: $V^*(s) = \max_{\pi} V_{\pi}(s)$ 或 $Q^*(s, a) = \max_{\pi} Q_{\pi}(s, a)$

值函数 (Value Function)

- 回报 (Return) 可写作递归形式:

$$G_t = R_{t+1} + \gamma G_{t+1}$$

- 值函数也可写作递归形式: 贝尔曼方程

$$V_{\pi}(s) = E_{\pi} [R_{t+1} + \gamma G_{t+1} | S_t = s]$$

状态值函数

$$= E_{\pi} [R_{t+1} + \gamma V_{\pi}(S_{t+1}) | S_t = s]$$

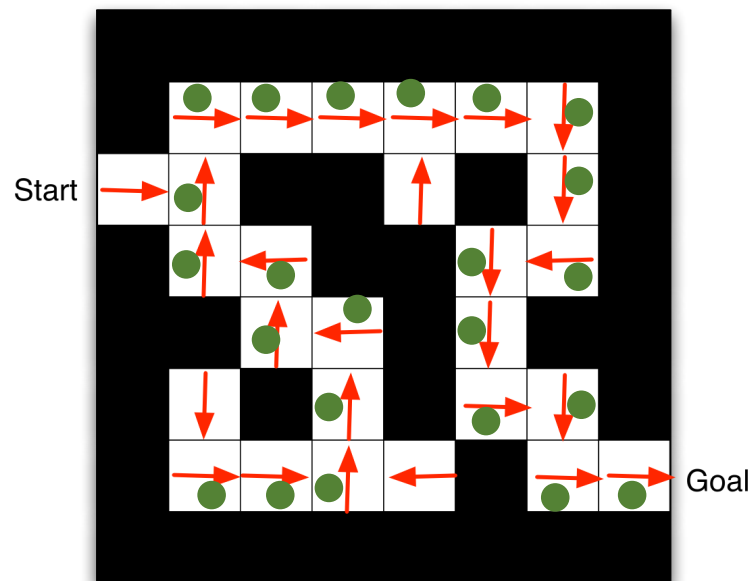
动作值函数 $Q_{\pi}(s, a) = E_{\pi} [R_{t+1} + \gamma Q_{\pi}(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]$

状态和动作值函数关系

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

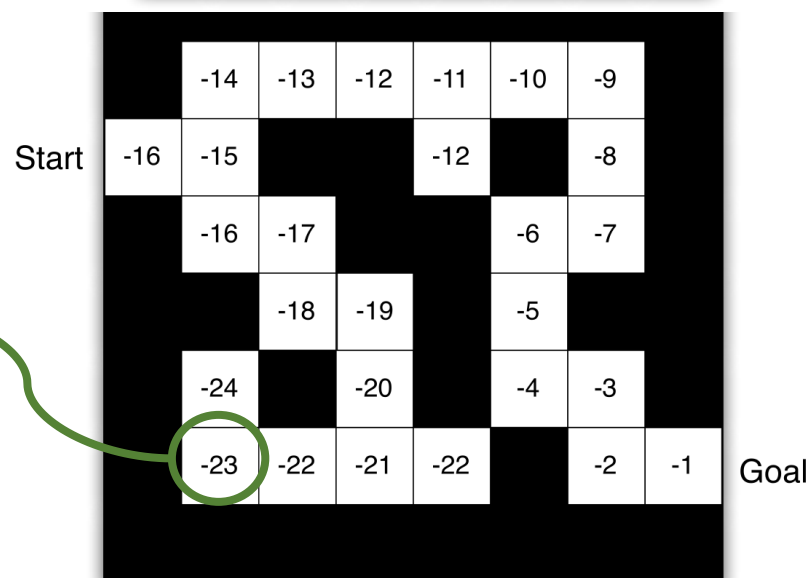


值函数 (Value Function)



数字表示在每个状态（位置）下的值函数 $V_{\pi}(s)$

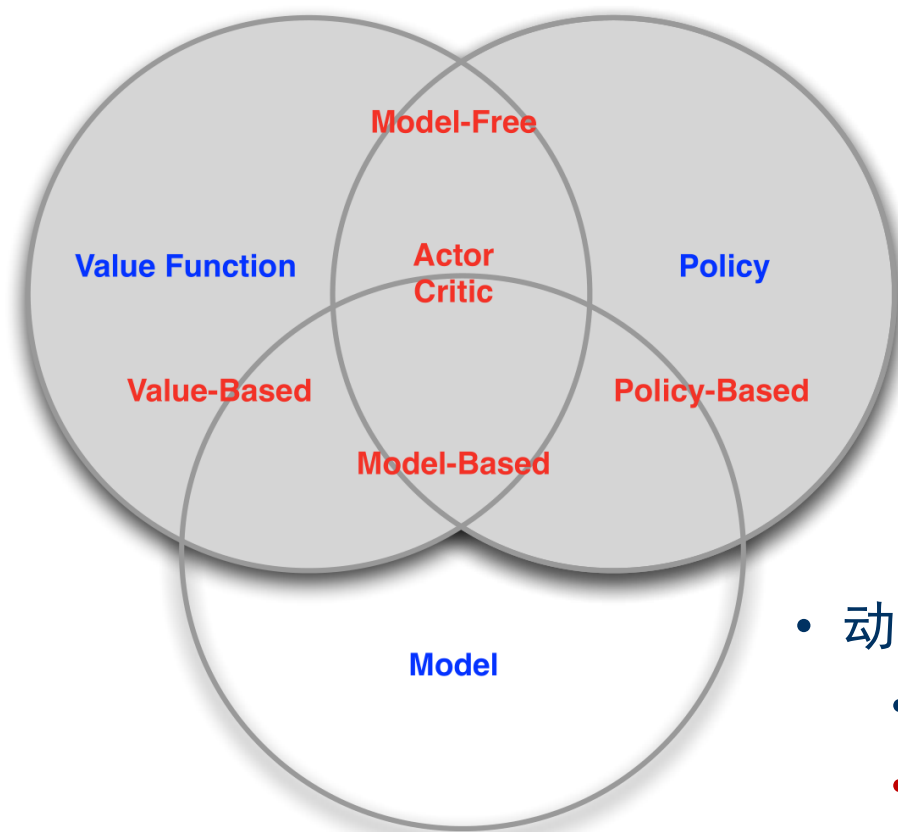
若agent处在该位置，则根据策略 $\pi(s)$ ，得到该状态下的值函数 $V_{\pi}(s)$ 为-23



RL智能体分类

- 基于值函数的
 - 在值函数空间中搜索
 - 策略隐式表达（基于值函数选择动作）
- 基于策略的
 - 在策略空间中搜索
 - 没有值函数
- 混合型策略
 - 有策略显式表达（动作网络）
 - 有值函数（评价网络）
 - Actor Critic
- 无模型的（Model-free）
 - 基于策略和/或值函数
 - 没有模型
- 基于模型的（Model-based）
 - 基于策略和/或值函数
 - 有模型

RL智能体分类



- 强化学习（环境未知）：
 - 模型常常靠学出来
 - 需要估计值函数
 - Value-based
 - 或直接优化策略
 - Policy-based
- 动态规划（环境已知）
 - 不需要学习模型
 - 动态规划求解

强化学习

- 强化学习的机制由来
- 基本模型和核心概念
- 强化学习的若干关键问题
- 经典强化学习：Q学习
- 深度强化学习
 - 深度Q网络
 - 其他方法

学习与规划

序贯决策问题中的两个基本问题：

- 强化学习Reinforcement Learning:
 - 没有关于环境的先验知识（开始时环境未知）
 - 智能体与环境互动
 - 智能体改进自己的策略
 - 通过试错方式
- 规划Planning:
 - 环境的模型已知
 - 智能体基于模型进行计算（不需与环境交互）
 - 智能体改进自己的策略
 - 通过推理式的搜索

雅达利游戏：规划

↓ 游戏运作方式**已知**（有模型）

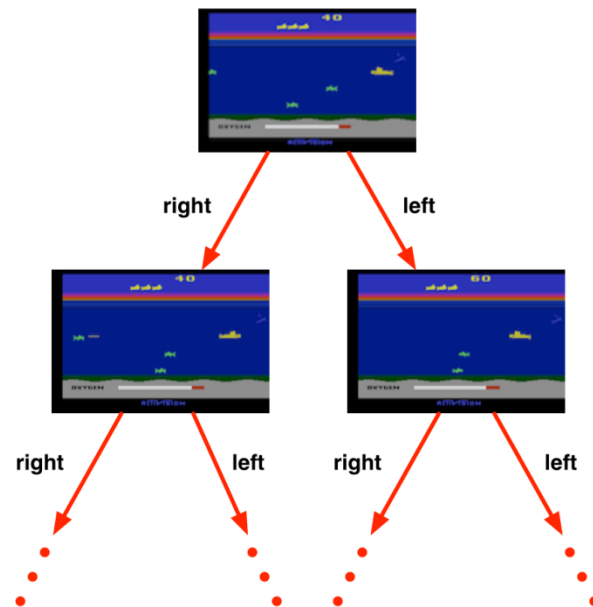
- 相当于可以向一个完美模型可供查询

↓ 如果在**状态s**下采取**动作a**

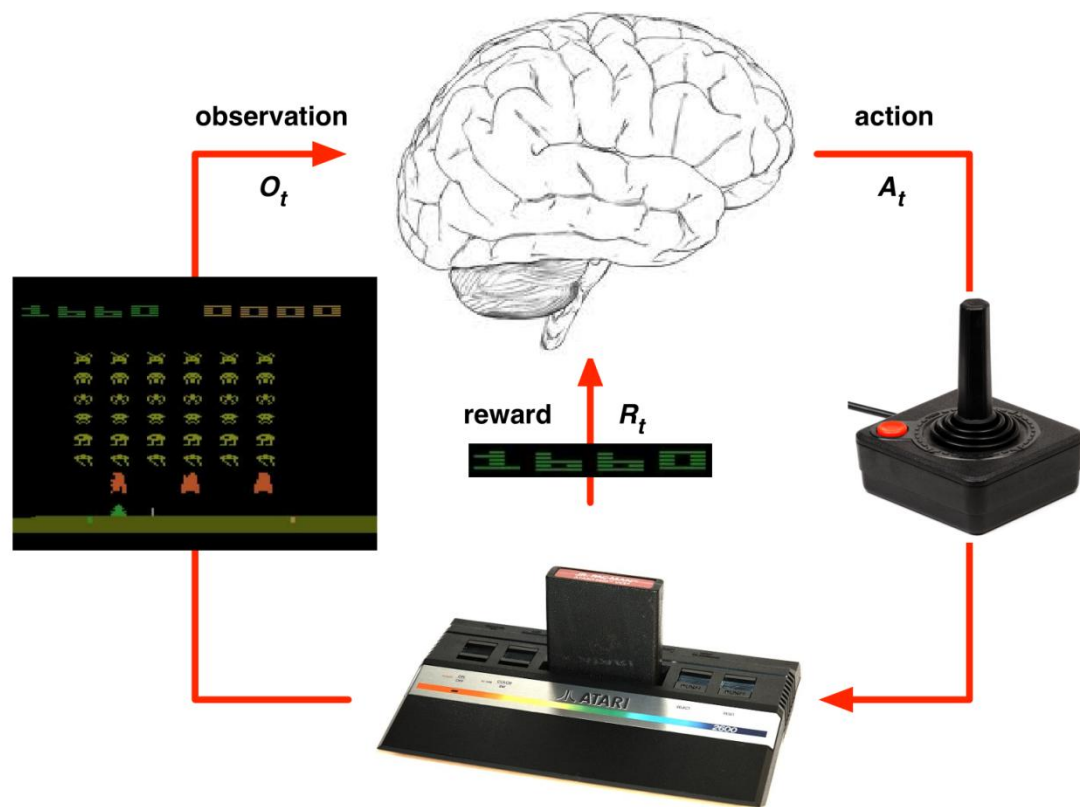
- 下一个状态是？
- 分数会变成？
- 统统已知！

↓ 能够在执行前找到最优策略

- 例：树搜索



雅达利游戏：强化学习



- 游戏运行方式未知（对于智能体，即环境未知）
- 通过游戏交互来学习
- 视角和动作方式与人类相同
 - 用摇杆进行动作，观察像素构成的屏幕，特别是屏幕上的分数

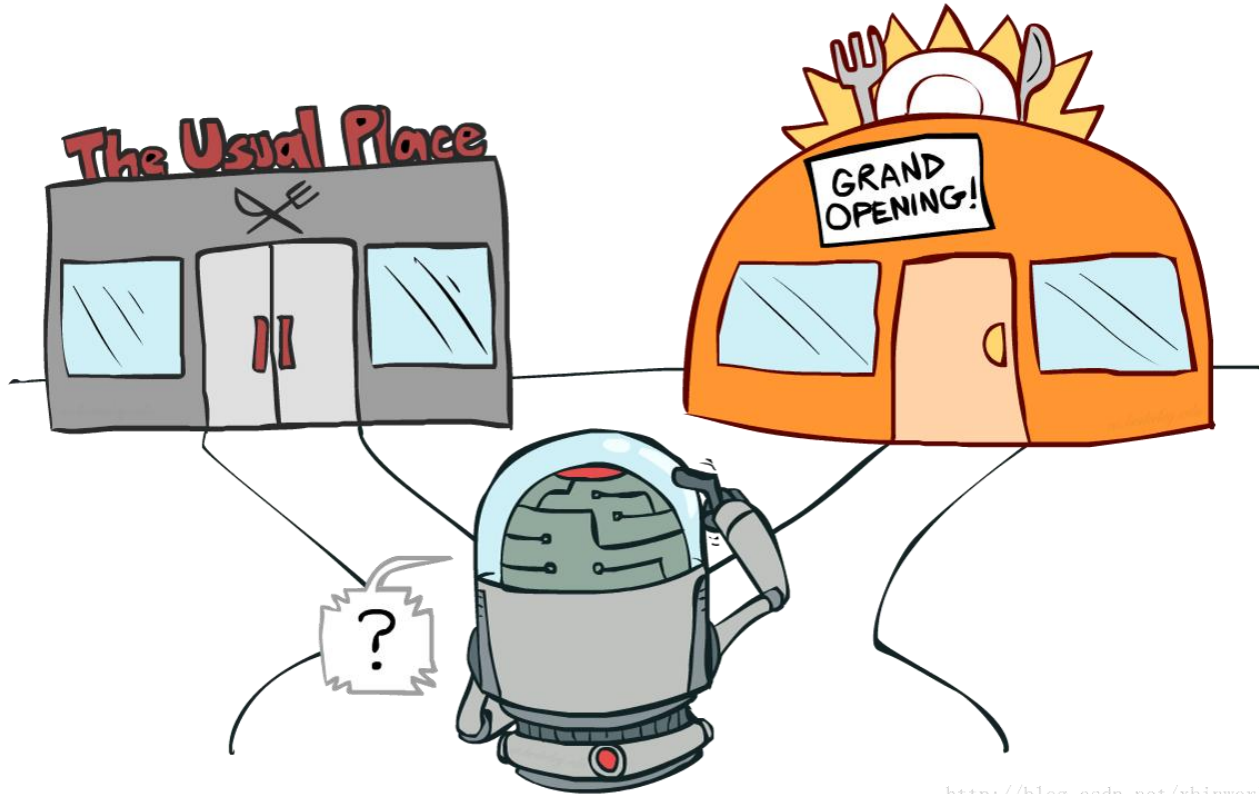
探索与利用

- 探索 (Exploration) 与 利用 (Exploitation)

- 强化学习是一种试错学习，在**试错**中：

- ❖ 智能体通过与环境互动的经验来学习
 - ❖ 智能体要尽可能多获得奖励/回报
 - ❖ 智能体要发现一个好的策略

探索与利用



<http://blog.csdn.net/xbinworld>

探索与利用

- 探索能帮助智能体发现更多关于环境的信息
- 利用能让智能体运用已知信息最大化奖励
- 探索和利用有着同等重要性！

强化学习

- 强化学习的机制由来
- 基本模型和核心概念
- 强化学习的若干关键问题
- 经典强化学习：Q学习
- 深度强化学习
 - 深度Q网络
 - 其他方法

经典强化学习：Q学习

Q学习（其中一种Value-based RL方法，同类：SARSA）

- 无模型的（Model-free）
- 基于动作值函数 $Q(s, a)$
- 通过与环境交互直接训练 $Q(s, a)$ ，找到 $Q^*(s, a)$ 和 $\pi^*(s)$

即时奖励已知

状态

Q-table initialised at zero

	UP	DOWN	LEFT	RIGHT
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0

After few episodes

	UP	DOWN	LEFT	RIGHT
0	0	0	0	0
1	0	0	0	0
2	0	2.25	2.25	0
3	0	0	5	0
4	0	0	0	0
5	0	0	0	0
6	0	5	0	0
7	0	0	2.25	0
8	0	0	0	0

Eventually

	UP	DOWN	LEFT	RIGHT
0	0	0	0.45	0
1	0	1.01	0	0
2	0	2.25	2.25	0
3	0	0	5	0
4	0	0	0	0
5	0	0	0	0
6	0	5	0	0
7	0	0	2.25	0
8	0	0	0	0


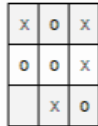
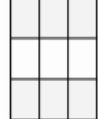
动作

Q值

Q学习：Q表

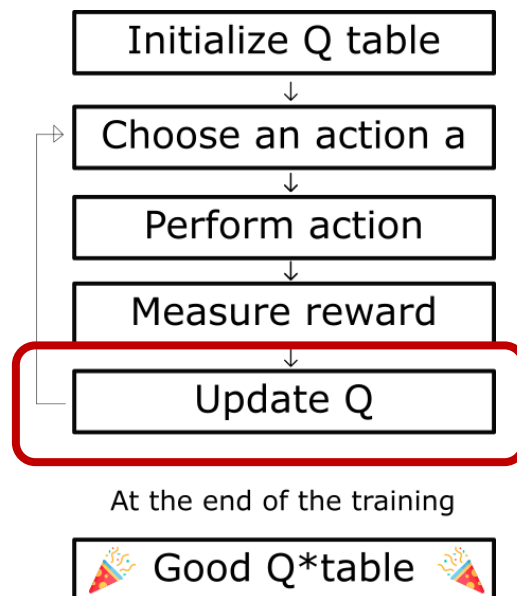
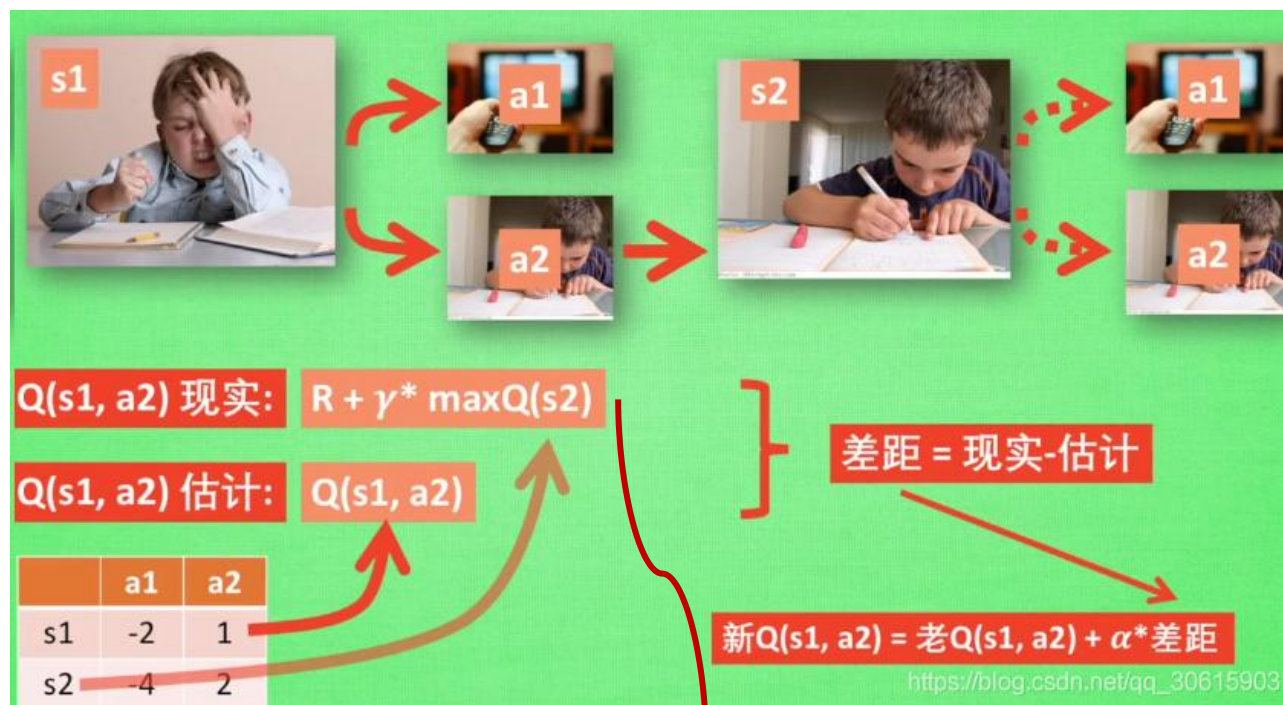


- Q表的行：状态
- Q表的列：动作
- 单元格：状态动作对的Q值
- 可先假设状态数量已知
 - Q表可拓展：添加行和列

Game State	Top Left	Top Middle	Top Right	Middle Left	Middle Middle	Middle Right	Bottom Left	Bottom Middle	Bottom Right
	N/A	0.5	N/A	N/A	N/A	N/A	0	0	N/A
	N/A	N/A	N/A	N/A	N/A	N/A	0.5	N/A	N/A
	0.3	0.5	0.3	0.5	0.7	0.3	0.3	0.5	0.3
...

三连棋游戏Q表

Q学习：更新Q值



$$\underbrace{newQ_{S,A}}_{\text{基于状态和}} = \underbrace{Q_{S,A}}_{\text{c当前Q值}} + \underbrace{\alpha}_{\text{学习效率}} \left(\underbrace{R_{S,A}}_{\text{基于状态和}} + \underbrace{\gamma}_{\text{折扣因子}} * \underbrace{\max_a Q'(s', a')}_{\text{在给定新的状态和行动下未来最大的奖励}} - Q_{S,A} \right)$$

基于状态和
 行动的新Q值

c当前Q值

学习效率

折扣因子

在给定新的状态和行动下未来最大的奖励

时序差分思想

值函数的近似表示

- 如果问题的状态集合规模大，Q表的存储和查询困难
 - 建立值函数的近似表示
- 引入一个状态值函数 $\hat{V}(s, w)$ ，由参数 w 描述，接受状态 s 作为输入，计算后得到状态 s 的值函数：

$$\hat{V}(s, w) \approx V_{\pi}(s)$$

- 类似地，引入动作值函数 $\hat{Q}(s, a, w)$ ：

$$\hat{Q}(s, a, w) \approx Q_{\pi}(s, a)$$

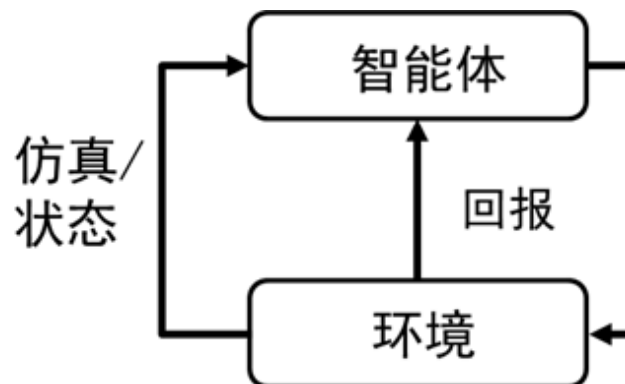
- 函数近似的方法：
 - 线性拟合，三次拟合，决策树，傅里叶变换...
 - 神经网络

强化学习

- 强化学习的机制由来
- 基本模型和核心概念
- 强化学习的若干关键问题
- 经典强化学习：Q学习
- 深度强化学习
 - 深度Q网络
 - 其他方法

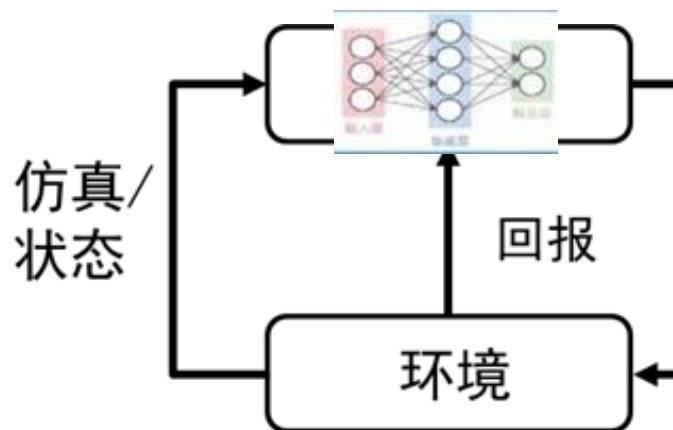
强化学习与深度强化学习

- 强化学习，一个框架：



- 深度强化学习：运用深度神经网络完成强化学习中的任务

深度强化学习
框架



强化学习：无人机控制 (4:20)



强化学习

- 强化学习的机制由来
- 基本模型和核心概念
- 强化学习的若干关键问题
- 经典强化学习：Q学习
- 深度强化学习
 - 深度Q网络
 - 其他方法

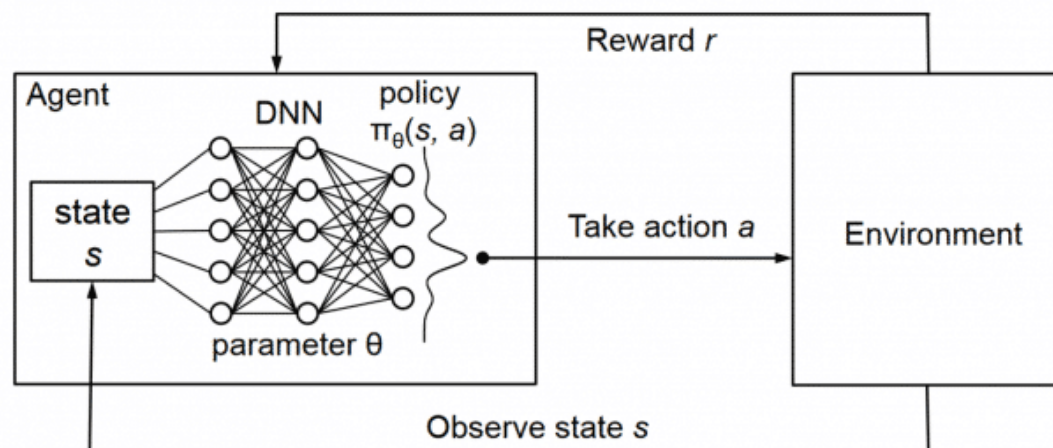
强化学习：雅达利游戏 (2:55)

Google Deepmind DQN playing
Atari Breakout

Setup:
NVIDIA GTX 690
i7-3770K - 16 GB RAM
Ubuntu 16.04 LTS
Google Deepmind DQN

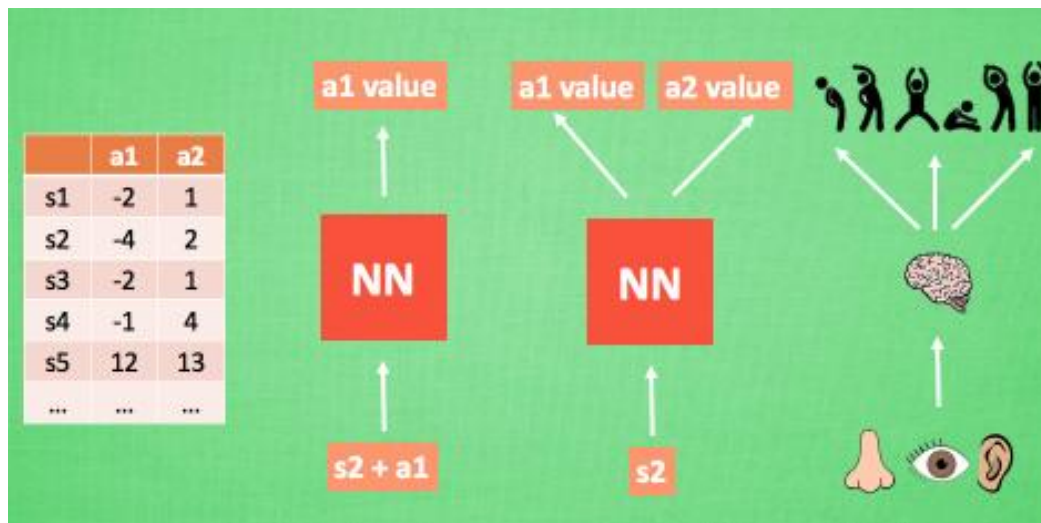
深度Q网络

- 深度Q网络（Deep Q Network, DQN）属于深度强化学习（DRL）的一种，它是深度学习与Q学习的结合体。
- Q表的局限性：当状态和行为的组合不可穷尽时，无法通过查表的方式选取最优的Action。
- DQN：用(深度)神经网络拟合Q表：
 - 转化为监督学习
 - 目标函数：Q函数



深度Q网络

- 神经网络的作用：



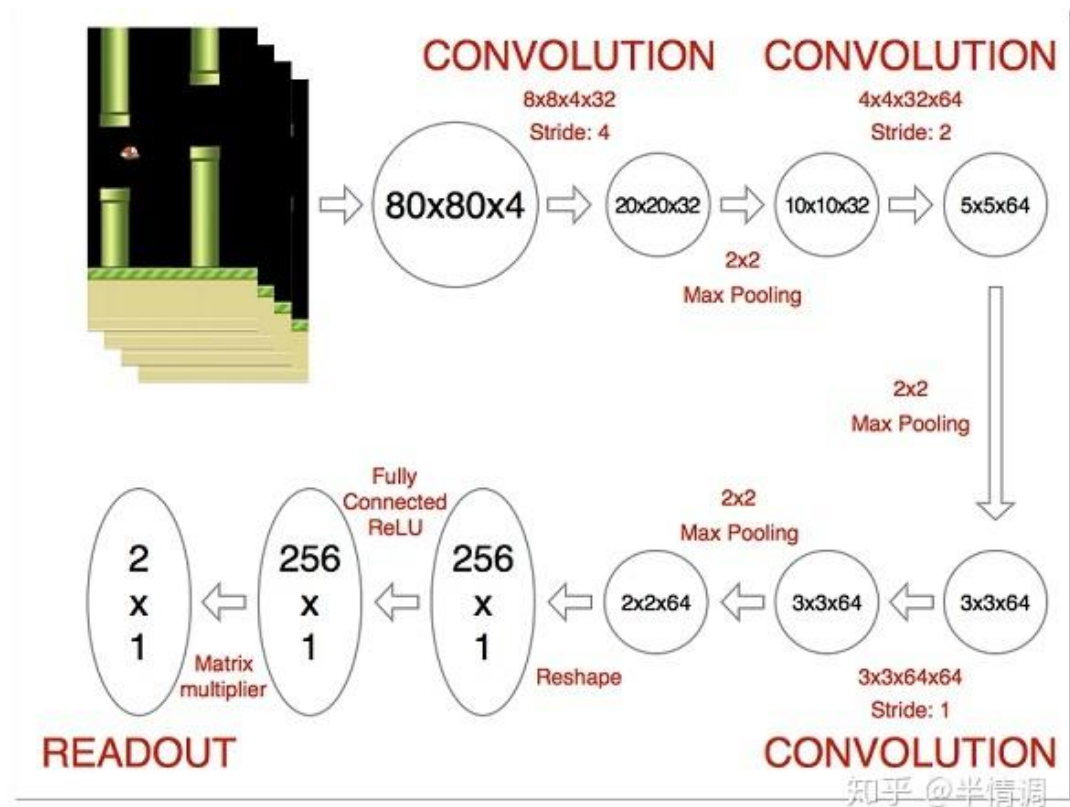
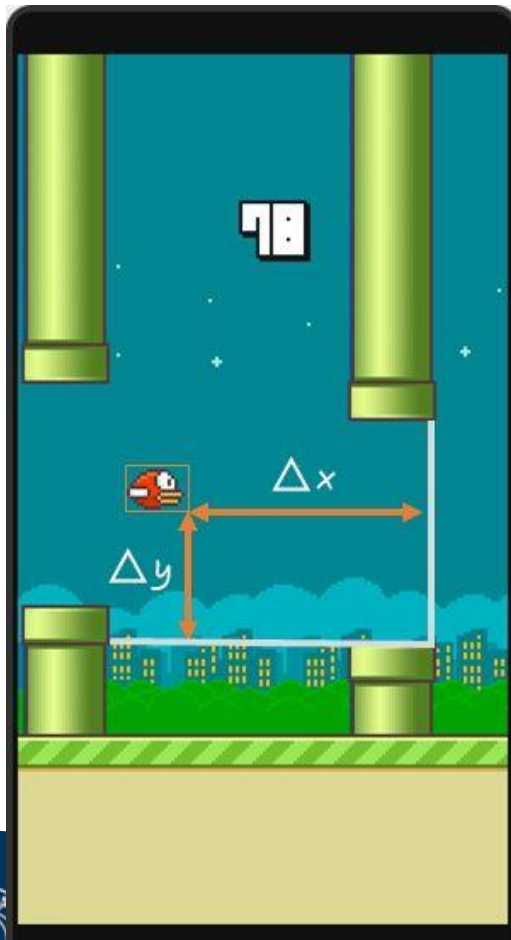
- 形式一：输入状态和动作，输出Q值
- 形式二：输入状态，输出所有动作的Q值 ←更多使用

Ref

[1] <https://mofanpy.com/tutorials/machine-learning/torch/intro-DQN/>

DQN: flappy bird

- 例子，DQN玩flappy bird [Deepmind 2015]:



知乎 @半情调

DQN: flappy bird

- 例子，DQN玩flappy bird：
 - 取 4 四帧游戏图像作为 state，输出每个 action 对应的 Q 值；
 - 网络输出相当于Q表的一行；
 - 采用三层卷积神经网络，不带池化层（保留对位置的敏感）

Layer	Input	Filter size	Stride	Num filters	Activation	Output
conv1	84x84x4	8x8	4	32	ReLU	20x20x32
conv2	20x20x32	4x4	2	64	ReLU	9x9x64
conv3	9x9x64	3x3	1	64	ReLU	7x7x64
fc4	7x7x64			512	ReLU	512
fc5	512			18	Linear	18

– Loss function:

$$L = \frac{1}{2} [r + \max_{a'} Q(s', a') - Q(s, a)]^2$$

深度Q网络

- 使用DQN模型代替Q表会遇到的问题：
 - 交互得到的序列存在一定的相关性：
 - ※监督学习要求样本独立同分布。
 - 交互数据的使用效率低：
 - ※迭代需要样本数量较多，样本获取靠交互。

经验回放

• 经验回放（Experience replay）：

- 收集样本：按照时间先后顺序存入结构中；
- 新的样本会覆盖时间上最久远的样本。

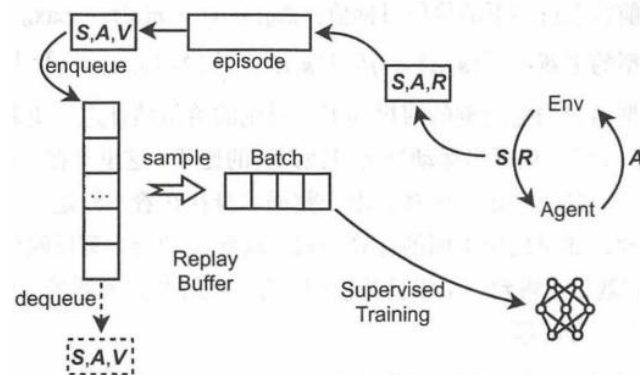
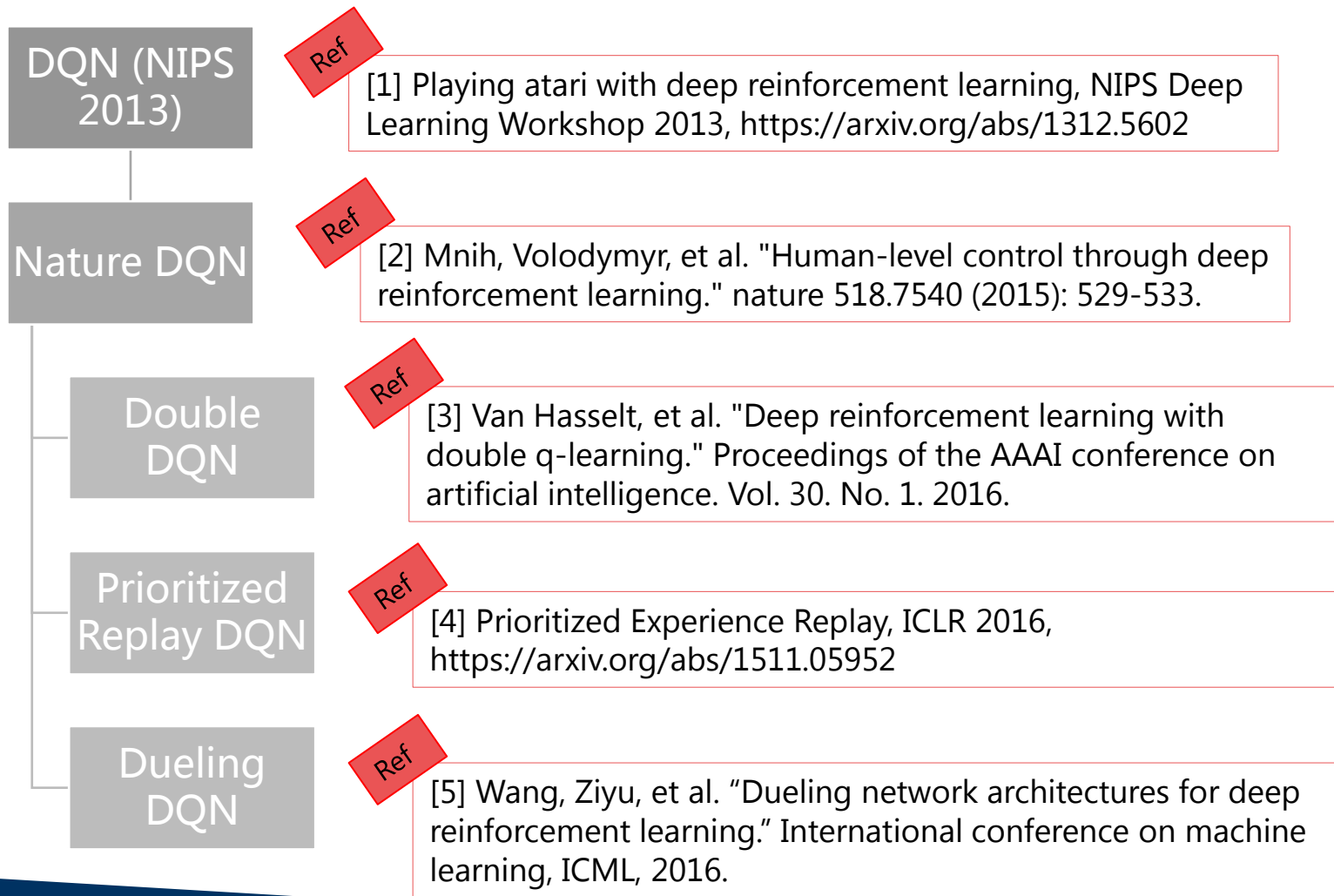


图 7-15 Replay Buffer 的结构图

- 采样样本：如果每次都取最新的样本，那么算法就和在线学习相差不多；一般来说，Replay Buffer会从缓存中均匀地随机采样一批样本进行学习。
- 好处：一批样本包含多条轨迹（多次交互），减小模型训练中的波动，稳定训练效果。

DQN发展历史



参考书目

- An Introduction to Reinforcement Learning, Sutton and Barto, 1998
 - MIT Press, 1998
 - <http://web.stanford.edu/class/psych209/Readings/SuttonBartoI-PRLBook2ndEd.pdf>
- Algorithms for Reinforcement Learning, Szepesvari, 2009
 - Morgan and Claypool, 2010
 - <https://www.semanticscholar.org/paper/Algorithms-for-Reinforcement-Learning-Szepesvari/e60f3c1cb857daa3233f2c5b17b6f111ff86698c>