强化学习基础篇（一）强化学习入门

本文主要基于David Silver的强化学习基础课程进行总结回归梳理强化学习的基础知识。 主要基于的课本来自Richard.S.Sutton以及Andrew G.Barto的《Reinforcement Learning》第二版。同时有由俞凯翻译的中译本。

这里主要关注两个方面，一方面是基础知识，另一方面是基础算法代码的实现。代码实现上，如果需要选择pytorch以及tensorflow同时实现。简单的场景直接使用Numpy实现。

1. 机器学习与强化学习

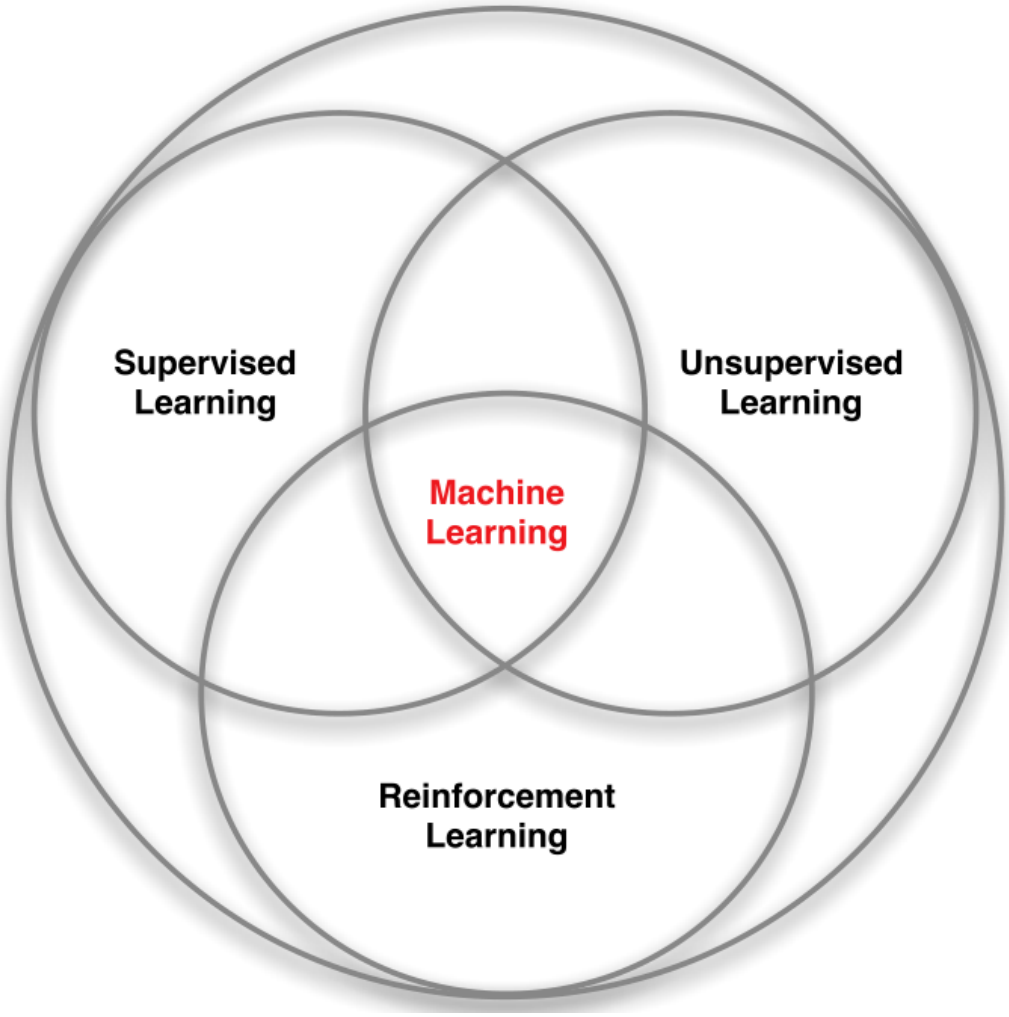
1.1 机器学习的分支领域

机器学习可以分为几个分支领域，监督学习（Supervised Learning），无监督学习（Unsupervised Learning）以及强化学习（Reinforcement Learning）。

监督学习（Supervised Learning）是从外部监督者提供的带标注训练集中进行学习，每一个样本都是关于情景与标注（label）的描述。标注即为在当前场景下，系统应当采取的正确动作，也可以将其看做为对当前情景进行分类的所属类别标签。采用这种学习方式是为了让系统能够具备推断或泛化能力，能够响应不同的情景并做出正确的动作选择，即目标是在情景未出现在训练数据中的情况下也能够做出正确的判断。

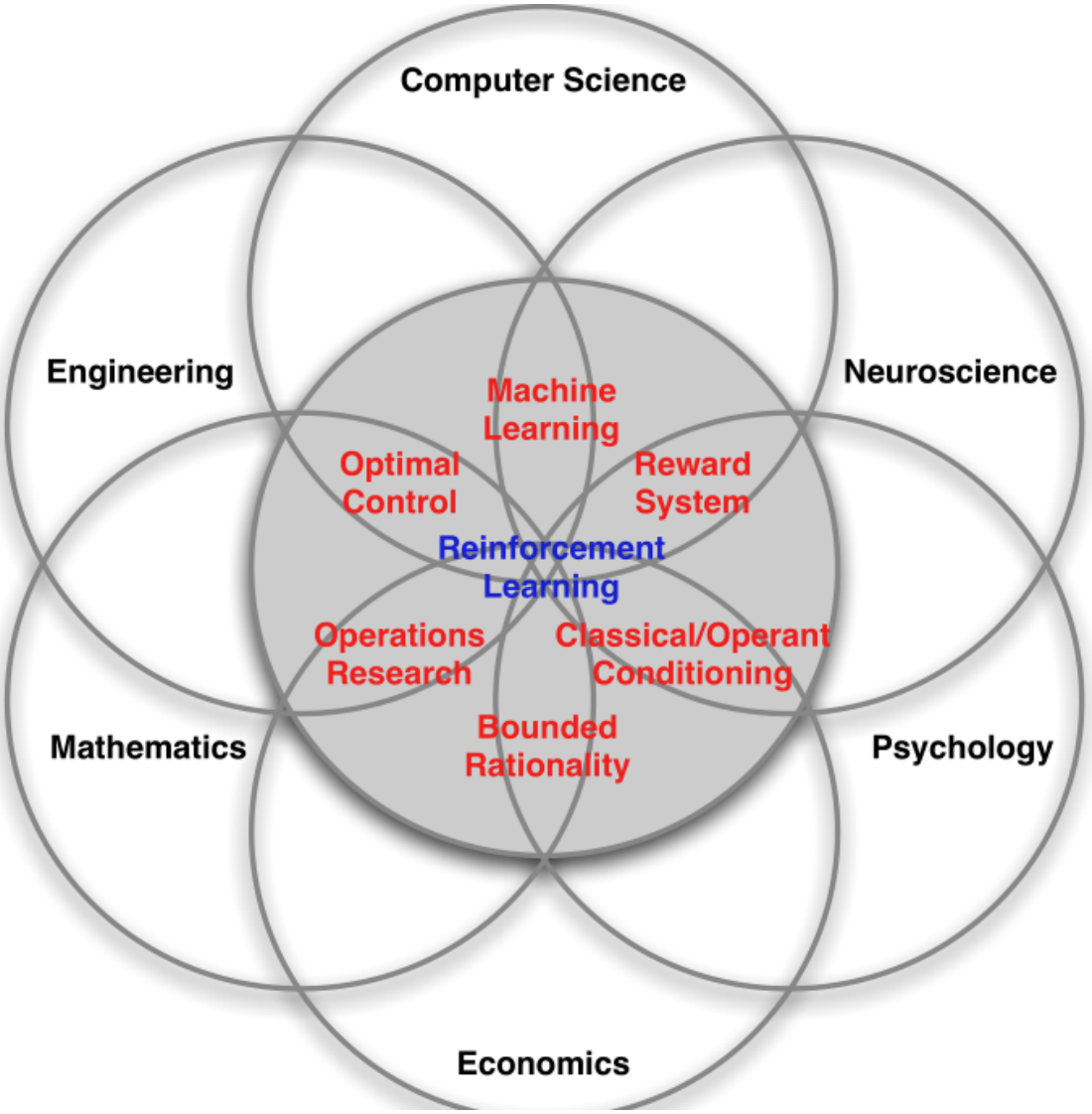
无监督学习（Unsupervised Learning） 是一个典型的寻找未标注数据中隐含结构的过程。强化学习有时候会被认为是一种无监督学习的方式，但是他们是有区别的。 强化学习的主要目的是最大化收益信号，而不是寻找数据的隐含结构。虽然无监督学习通过智能体寻找隐含结构对强化学习很有意义，但是这并不能解决最大化收益信号的问题。

所以**强化学习（Reinforcement Learning）**是在监督学习与无监督学习之外的第三种机器学习范式。他是学习这样一个问题"What to do,how to map situations to actions, so to maximize a numerical reward singal"。即考虑如何才将当前的情景映射为动作，以最大化数值化的收益信号。
这三种学习范式的关系如下所示：



1.2 强化学习在多学科中的应用

强化学习多学科中都有着有效的应用，他们在不同学科中强化学习的方法与思想可能有着不同的专业名称。比如在计算机科学中属于机器学习（Machine Learning），在工程领域属于最优化控制（Optimal Control），在神经科学领域属于奖励系统（Reward System），在心理学领域中属于经典/操作性条件反射（Classical/Operant Conditioning）,在经济学领域属于有限理性（bounded rationality）理论，在运筹学中属于也有相应的数学基础。这些交叉领域相关的关系如下图所示：



1.3 强化学习的主要特征

强化学习区别于其他机器学习范式的主要几点原因是：

a. 强化学习没有监督信号，只有收益信号（reward signal）

b. 强化学习得到得到回馈不是即时的，是有时延的反馈。

c. 强化学习中使用的数据不是独立同分布，时序数据对强化学习非常重要。

d. 智能体在当前场景下做出的决策，将对未来将会受到的数据序列产生直接的影响。

1.4 强化学习的应用实例

这里简单说一些：

a. 直升飞机进行特技表演，这是吴恩达在斯坦福做出来的东西，没有特定程式的控制，飞机自己学习特技表演，仅仅依靠奖励来提升自己
b. AlphaGo的自我对弈，就是强化学习，通过胜负这个奖励信号，不断提升自己的棋力
c. Tesauro的TD-Gammon程序在西洋双陆棋（BackGammon）的出色表现。
d. 使用强化学习控制能源站，训练机器人走路，玩视频游戏等等。

2. 强化学习中的几个基本概念

2.1 奖励（Reward）

奖励R_t是一个变量的回馈信号，他表明了智能体在时间步t时表现如何。 智能体训练的目的即最大化这个累积的奖励R_t。 强化学习最基础的假设即奖励假设

> Reward Hypothesis
>
> All goals can be described by the maximisation of expected cumulative reward.
>
> 奖励假设
>
> 所有的目标可以由最大化期望累计奖励来描述

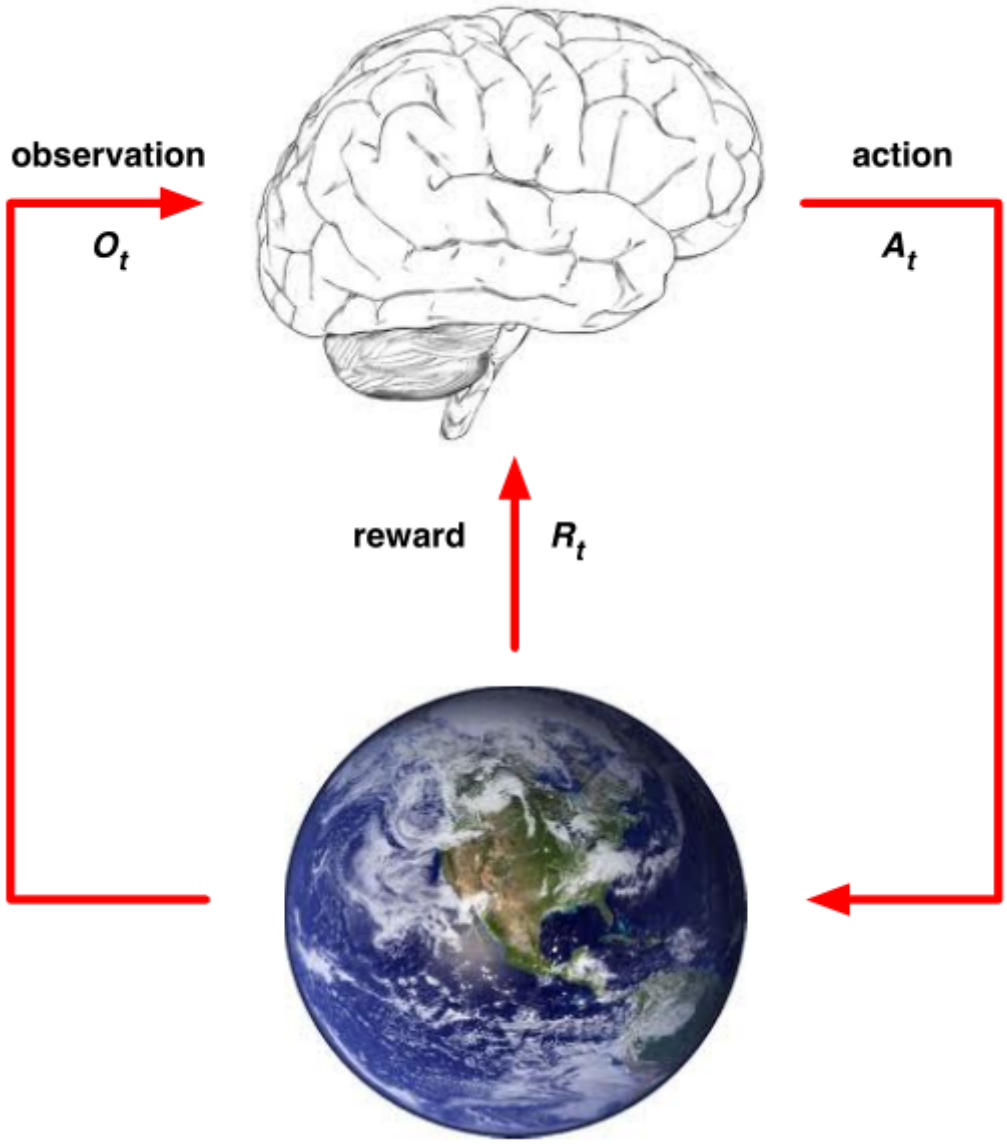
奖励的设定是个很重要的问题，比如在训练一个类人机器人走路的任务中，可以设定机器人向前移动即可增加奖励r,如果摔倒即扣除奖励r。在训练智能体做Atari视频游戏任务中，其奖励可以是视频游戏中本身的分数。

2.2 序贯决策(Sequential Decision Making)

序贯决策的目的还是在于最大化未来的总回报。智能体在进行训练任务中可能有很长的动作序列， 其最终回报不是即时的，会有一定的时延。甚至在一些场景之下，智能体必须牺牲一些即时回报，以获得长期的回报。

2.3 智能体与环境（Agent and Enviroment）

智能体和环境之间的交互如下图所示：



在每个时间步t，智能体从环境接收观测信号O_t以及量化的回报R_t， 并执行动作A_t。环境在每个时间步即接收动作A_t,产生由动作执行后生成的新的观测状态O_{t+1}与回报R_{t+1}。

2.4 状态（state）

这里有个特定的概念即历史（history），在强化学习中是指序列化的观测状态，动作以及回报。即：

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

这里包含了从迭代开始到时间步t的所有观测变量。并且在时间步t之后的所有观测都依赖于当前的history。智能体基于当前的历史选择动作，并从环境中接收新的观测与回报。

与历史(history)这个概念有所区别的是状态(State)， 状态是主要用于智能体如何进行下一步决策，他不是一个序列，而是从历史产生的映射状态，即$S_t = f(H_t)$。

状态需要考虑到三种状态：环境状态（Enviroment State）， 智能体状态（Agent State）， 信息状态（Information States）。

环境状态S_t^e（Enviroment State)

环境状态是对环境的私有表示，他是一些测试智能体任务中用来挑选下一步观察和奖励的数据。环境State并不总是对智能体可见，即使可见，也可能包含一些对任务无关的信息

智能体状态S_t^a（ Agent State）

智能体状态是智能体的内部表示，他包含智能体用来挑选下一步动作的信息，智能体State是我们强化学习算法所需要的主要状态。 他可以是history的任何函数：$S_t^a = f(H_t)$,

信息状态 （Information State)

信息状态是包含历史中所有有用的信息，也称为Markov State，马尔科夫状态。既然叫马尔科夫状态，也就是说，下一个状态只依赖于当前状态。
马尔科夫性质其定义为：

> 一个状态S_t具有马尔科夫性质，其充分必要条件为：
>
> $$P[S_{t+1}|S_t] = P[S_{t+1}|S_1, \dots, S_t]$$

即未来的状态在给定当前状态的情况下，是独立于历史状态的。所有的历史数据已经充分反映在了当前的状态S_t当中。

$$H_{1:t} \rightarrow S_t \rightarrow H_{t+1:\infty}$$

一旦我们知道了当前状态，我们即可丢弃所有的历史数据。当前状态是对未来数据的充分统计量。与此同时，环境状态S_t^e与历史H_t都是具有马尔可夫性。

2.5 环境的可观测性

有两种环境的分类，一种是完全可观测环境，另一种是部分可观测环境。

完全可观测（Fully Observable Enviroments)

在完全可观测环境之中，智能体可以直接获取到环境的状态， 即$O_t = S_t^a = S_t^e$。这种场景之下智能体状态，环境状态，信息状态完全一致。这种场景就是马尔科夫决策过程MDP(Markov decision process)。

部分可观测(Partially Observable Enviroments)

在其他一些场景之下，智能体不能直接观察环境，比如机器人的摄像头不能告诉他具体的位置，卡牌游戏不知道别人的牌，只知道已经打出的牌。此时智能体状态（agent state ）不等于 环境状态（environment state），所以这只是一个 partially observable Markov decision process(POMDP)，部分可观察马尔科夫决策过程。在这种情况下，智能体没法参考环境状态，必须构建自己的状态表示，比如以下几种情况

- 使用完整的历史信息，$S_t^a = H_t$
- 坚信环境状态：$S_t^a = (P[S_t^e = s^1], \dots, P[S_t^e = s^n]$
- 使用循环神经网络: $S_t^a = \delta(S_{t-1}^a W_s + O_t W_o)$

3. 强化学习要素

除了奖励，智能体和环境之外，强化学习系统也会有如下三个核心要素，包括策略（Policy），值函数（Value Function）以及模型（Model）。

3.1策略(Policy)

策略定义了智能体在特定时间的行为方式，即，策略是环境状态到动作的映射。策略可能是确定的策略，也可能是随机的策略，随机策略有助于探索未知的奖励。策略可能是一个简单的函数或查询表格，也可能是涉及大量计算的神经网络。策略本身是可以决定行为的，因此策略是强化学习智能体的核心。一般来说，策略可能是环境所在状态和智能体所采取的动作的随机函数。

确定性策略可以表示为：$a = \pi(s)$

随机策略可以表示为：$\pi(a|s) = P[A_t = a|S_t = s]$

3.2 值函数（Value Function)

之前所说的奖励，即收益信号，他表明了在即时状态下什么是好的，而值函数（Value Function）表示了从长远角度看什么是好的。简单地说，一个状态的价值是一个智能体从这个状态开始，对将来累积的总收益的期望。尽管收益决定了环境状态的直接、即时、内在的吸引力，但价值表示了接下来所有可能状态的长期期望。智能体通过值函数进行动作的选择。

$$v_\pi = E_\pi[R_{t_1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots | S_t = s]$$

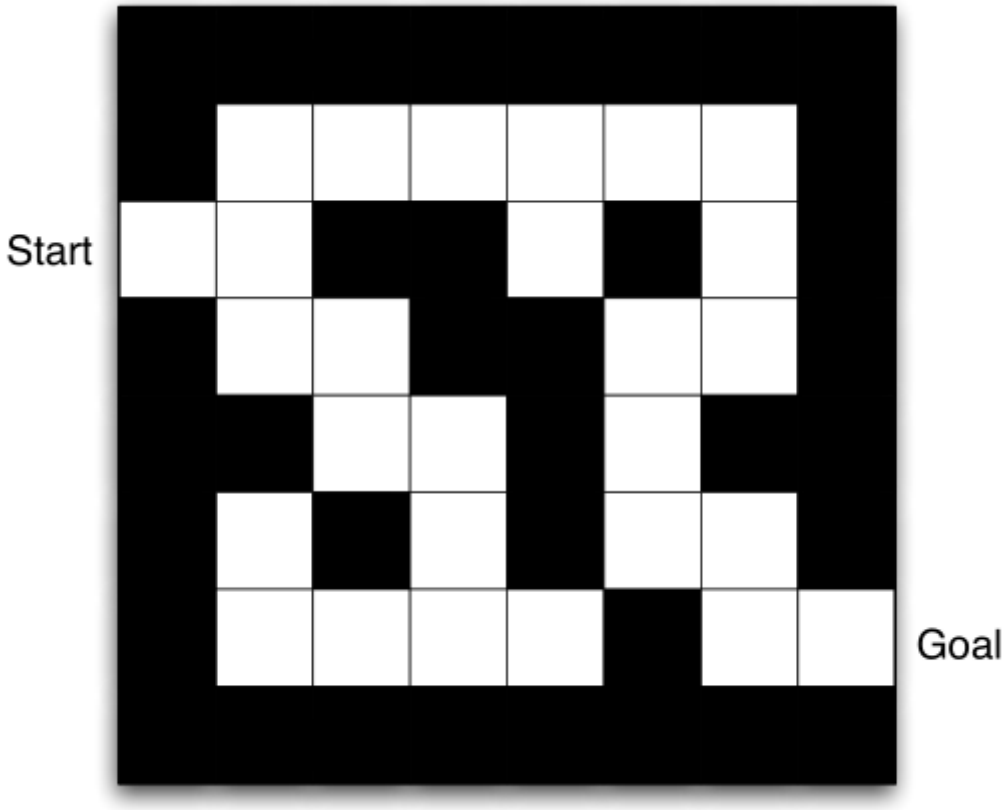
3.3 模型（Model)

智能体对环境建立的模型是另一个重要的要素，这是一种对环境反映模式的模拟。智能体根据模型对外部环境进行推断。例如，给定一个状态和动作，模型就可以预测外部环境的下一个状态和收益。环境模型可以被用来做规划。

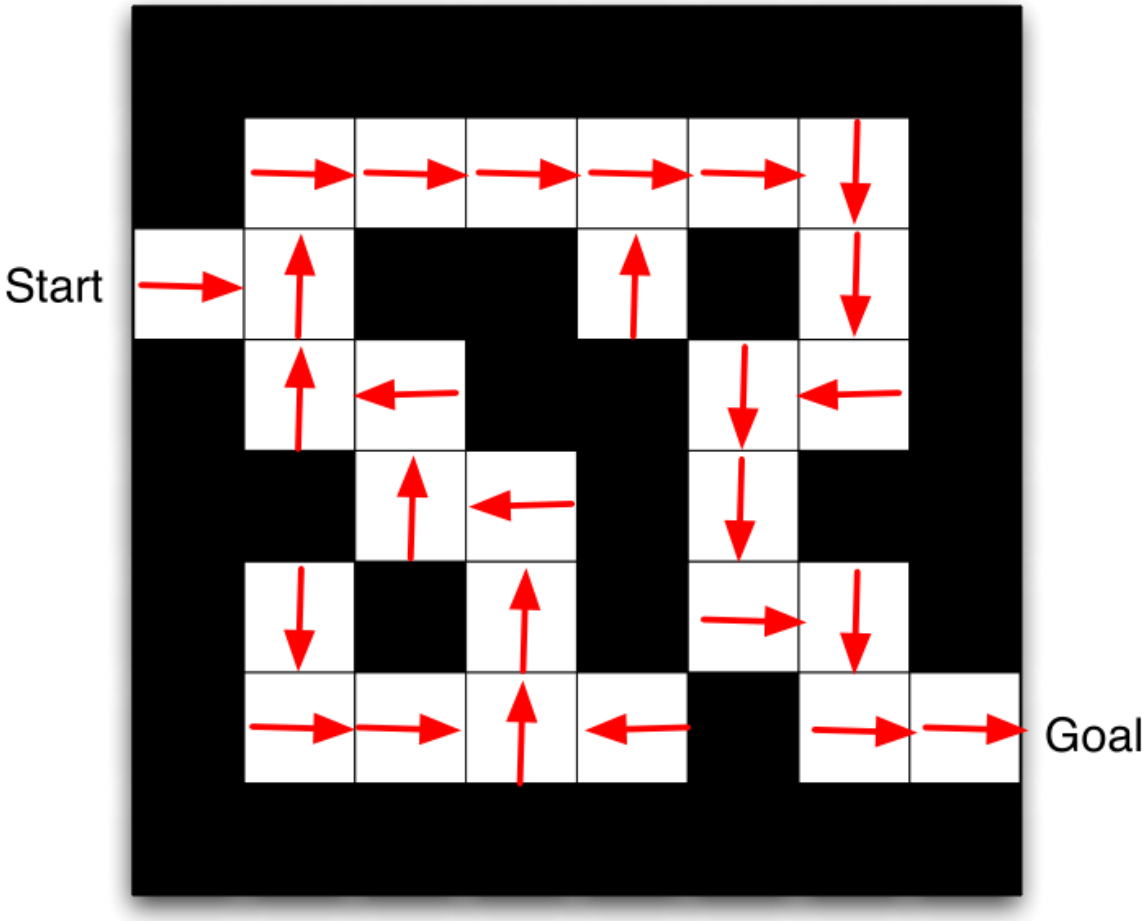
模型可以如下表示：

$$P^a_{ss'} = P[S_{t+1} = s' | S_t = s, A_t = a]$$
$$R^a_s = E[R_{t+1} | S_t = s, A_t = a]$$

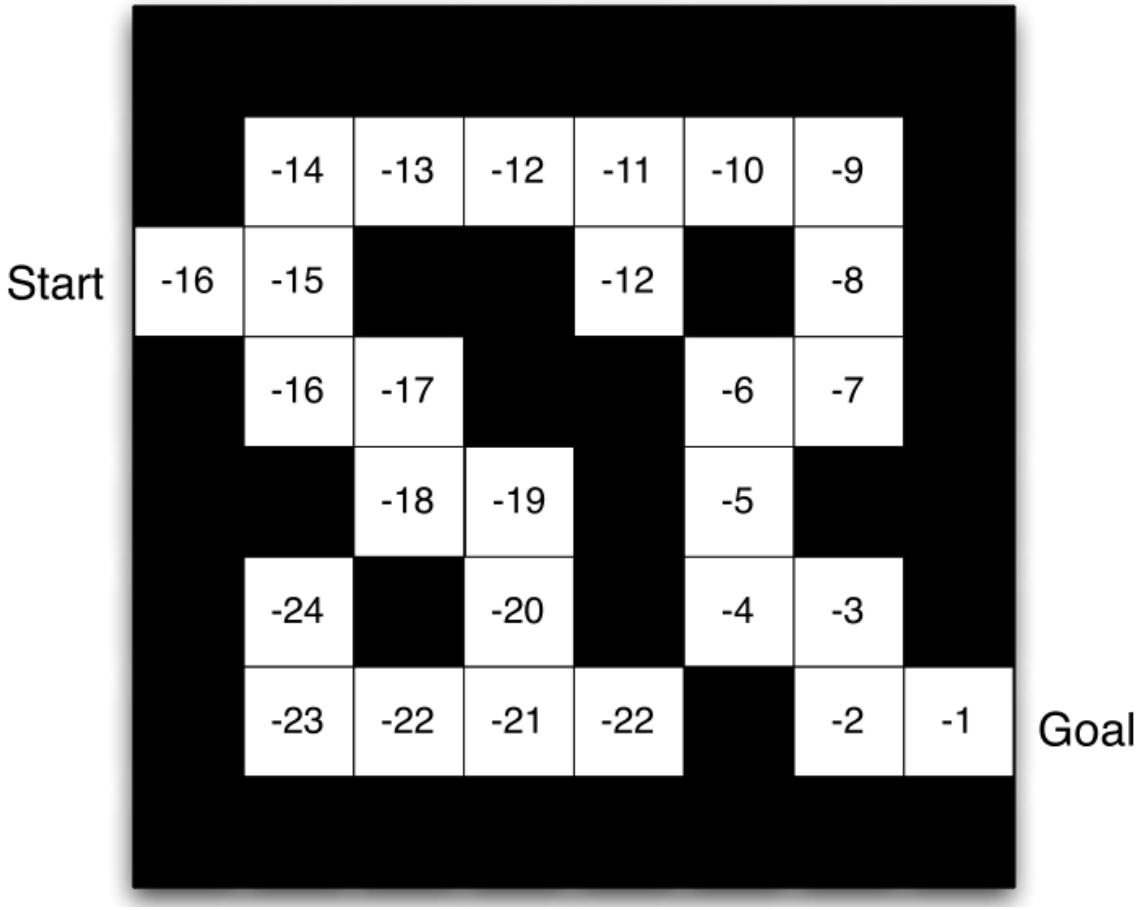
下面这个例子展示在一个迷宫游戏中什么是策略，值函数以及模型。在迷宫环境中我们设定奖励为每个时间步为-1。可执行的动作为（N,E,S,W），状态即为智能体的位置。



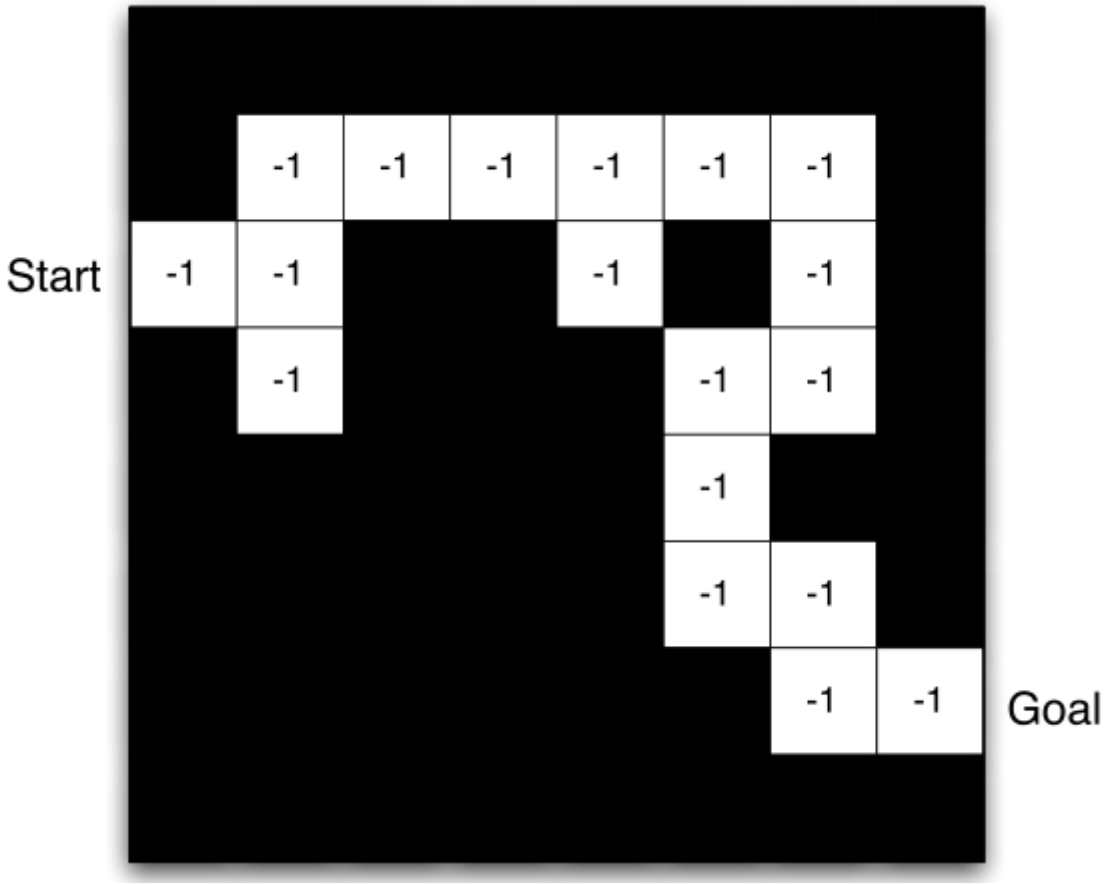
策略的表示在每个状态下采取的动作：



值函数表示为：



模型可以表示为：



这里要注意到智能体是有着独立的对环境的认知，这个对环境的认识不一定是完全与真实环境一模一样。这个认知中的环境包括执行动作将会对状态产生什么影响，以及在每个状态中可以获得什么样的奖励。认知中的模型完全可以是不完美的。

4. 强化学习算法分类

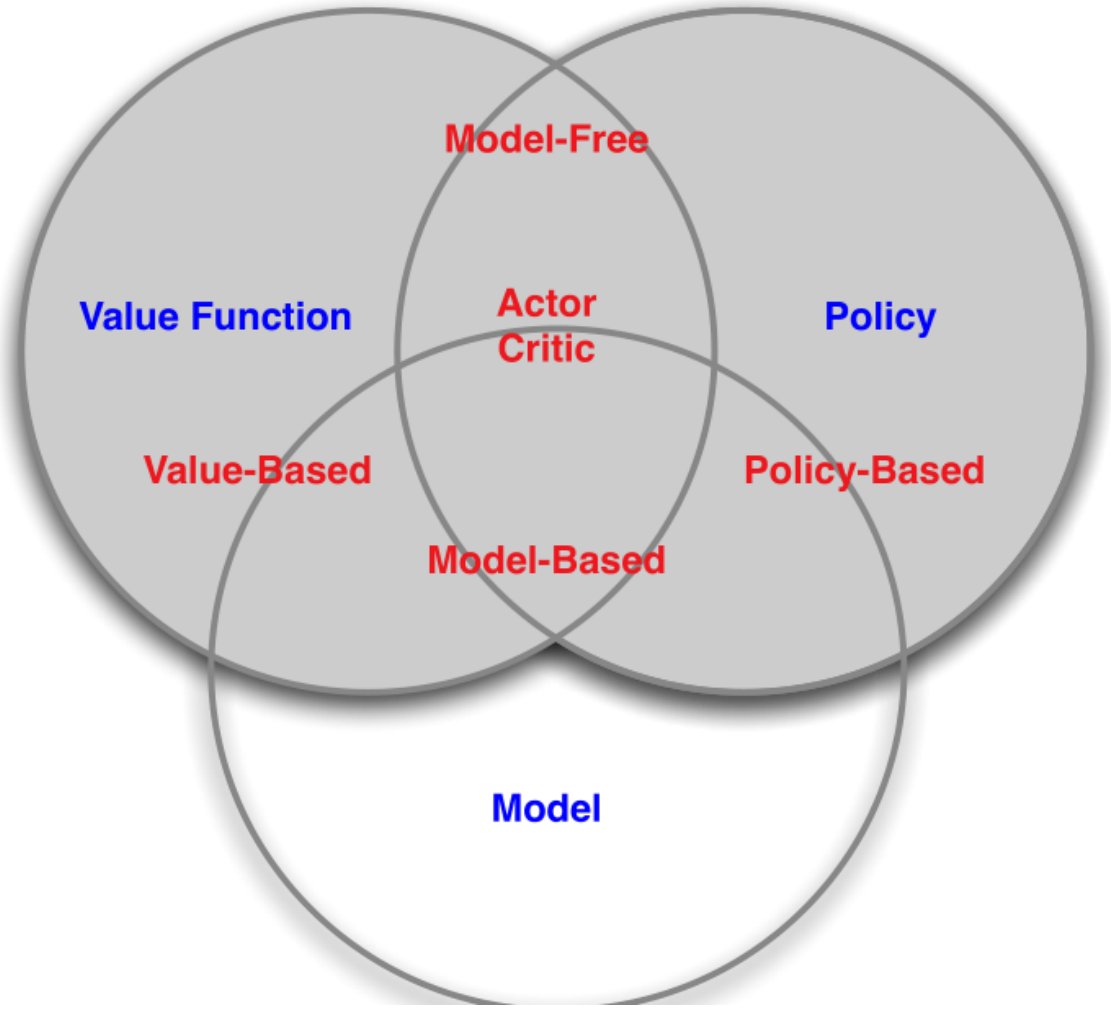
基于使用策略与使用值函数，我们可以将强化学习算法分为三大类：

- 基于值函数的强化学习算法
- 基于策略的强化学习算法
- Actor-Critic算法

基于是否使用模型，我们可以将强化学习算法分为两大类：

- 无模型算法（Model Free）
- 基于模型的强化学习算法（Model Based）

总结一下就是这张图：



5.强化学习中的几个基本问题

5.1 强化学习与规划（Reinforcement Learning and Planning）

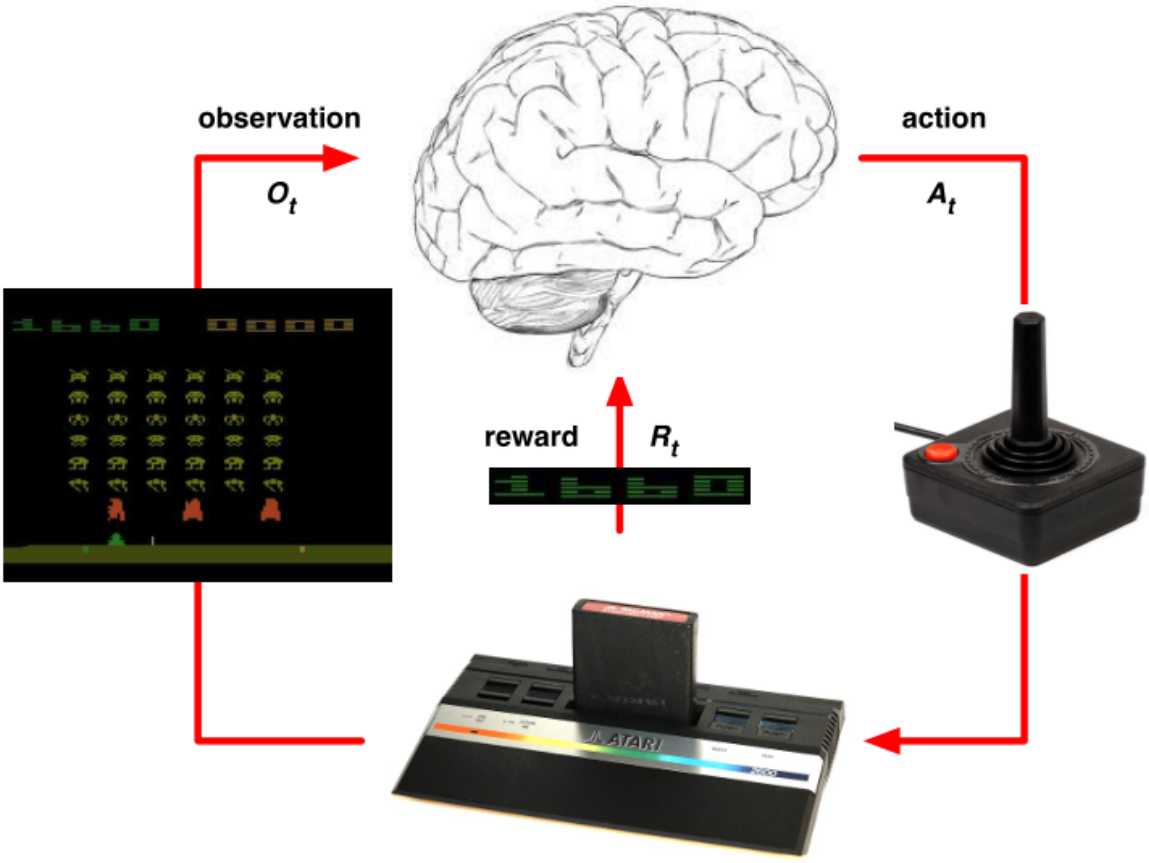
在序贯决策（Sequential Decision Making）中有两个基础性的问题，一个是强化学习(Reinforcement Learning），另一个是规划（Planning）。规划即时在真正经历之前，先考虑未来可能发生的各种情景，从而预先决定采取何种动作。他们的区别在于：
强化学习：

- 环境初始状态未知
- 智能体与环境进行实时交互
- 智能体会不断提升策略

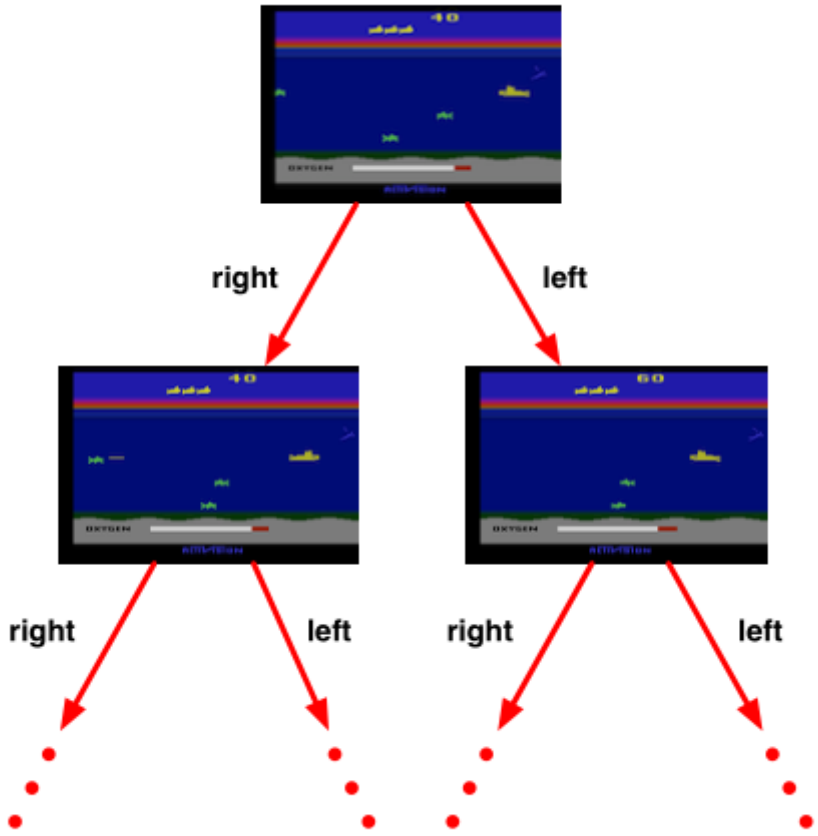
规划：

- 环境模型完全已知
- 智能体通过已知的模型进行所有运算，运算过程完全不需要与环境进行交互。
- 智能体会不断提升策略

以智能体执行视频游戏Atari训练任务为例，在强化学习中游戏规则是未知的，智能体在于游戏屏幕直接的交互过程中学习并通过输入的图像像素以及分数选择游戏手柄的动作。



但是在规划任务中，游戏规则是完全已知的，智能体可以随时查询模拟器获得所有信息，其中模拟器即游戏主机。在查询主机信息的过程中，主机会告诉智能体在状态s，下一个动作可能有哪些，以及在每个可能的动作产生的下一个状态会产生什么样的分数。规划的目的是要直接找到最优的策略。

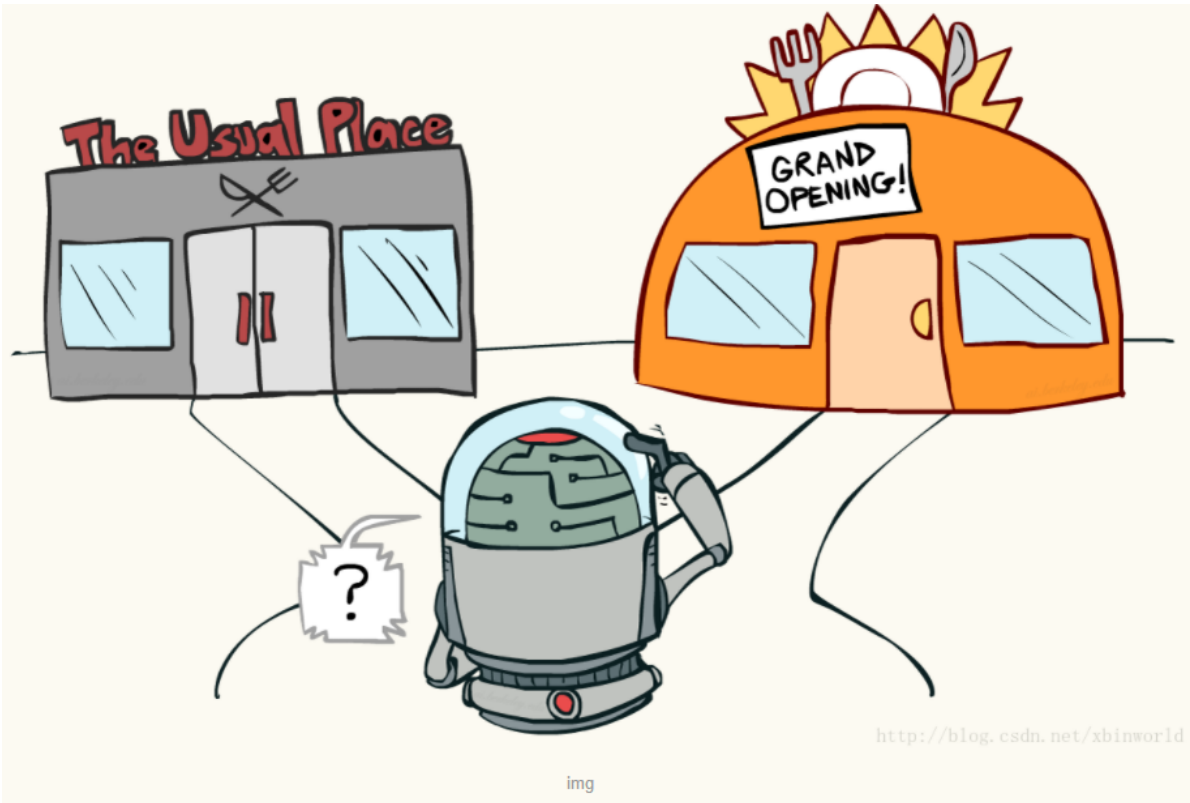


5.2. 探索与利用（Exploration and Explotiation）

探索与利用（Exploration and Explotiation）是强化学习的另一个基本问题。

假设有如下的场景：

> 假设你家附近有十个餐馆，到目前为止，你在八家餐馆吃过饭，知道这八家餐馆中最好吃的餐馆可以打8分，剩下的餐馆也许会遇到口味可以打10分的，也可能只有2分，如果为了吃到口味最好的餐馆，下一次吃饭你会去哪里？



img

所谓探索：是指做你以前从来没有做过的事情，以期望获得更高的回报。所谓利用：是指做你当前知道的能产生最大回报的事情。那么，你到底该去哪家呢？这就是探索-利用困境。

如果你是以每次的期望得分最高，那可能就是一直吃8分那家餐厅；但是你永远突破不了8分，不知道会不会吃到更好吃的口味。所以只有去探索未知的餐厅，才有可能吃到更好吃的，同时带来的风险就是也有可能吃到不和口味的食物。

这个例子只是用于找找感觉，后面会讨论多臂赌博机问题的时候再来多讨论一下。一般基本的最常用的随机策略为$\epsilon - greedy$策略（抖动策略）：

$$\pi(a|s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|A(s)|}, & if a = argmax_a Q(s,a) \\ \\ \frac{\epsilon}{|A(s)|}, & if a \neq argmax_a Q(s,a) \end{cases}$$

这个策略是比贪婪策略稍微复杂一点，意思是说，ϵ的概率随机选择任何一个动作，否则选择Q最大的动作。该策略称为抖动策略。

- 利用抖动策略的好处是：
  （1） 抖动策略计算容易，不需要复杂的计算公式。
  （2） 能保证充分探索所有状态。
- 当然相应的坏处是：
  （1） 需要大量探索，数据利用率低。
  （2） 需要无限长时间（取决于状态的数量以及ϵ大小）。

5.3 .预测与控制（Prediction and Control)

预测与控制的问题是另一个强化学习的基础问题，预测（Prediction ）即在给定策略的情况下对未来进行评估，控制（Control）即为了寻找到最优的策略，对未来进行最优化的过程。