



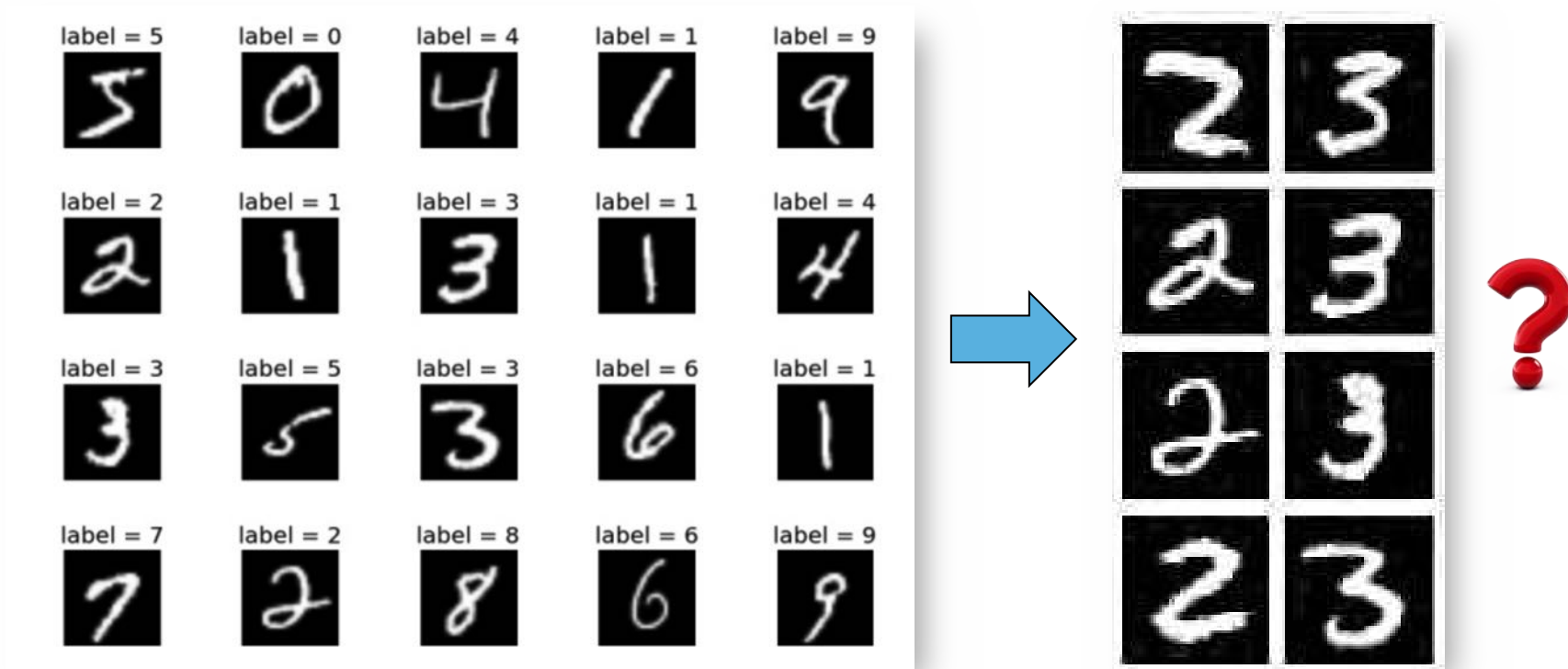
人工智能与信息社会

基于仿生算法的智能系统：神经网络2

2019.10.28

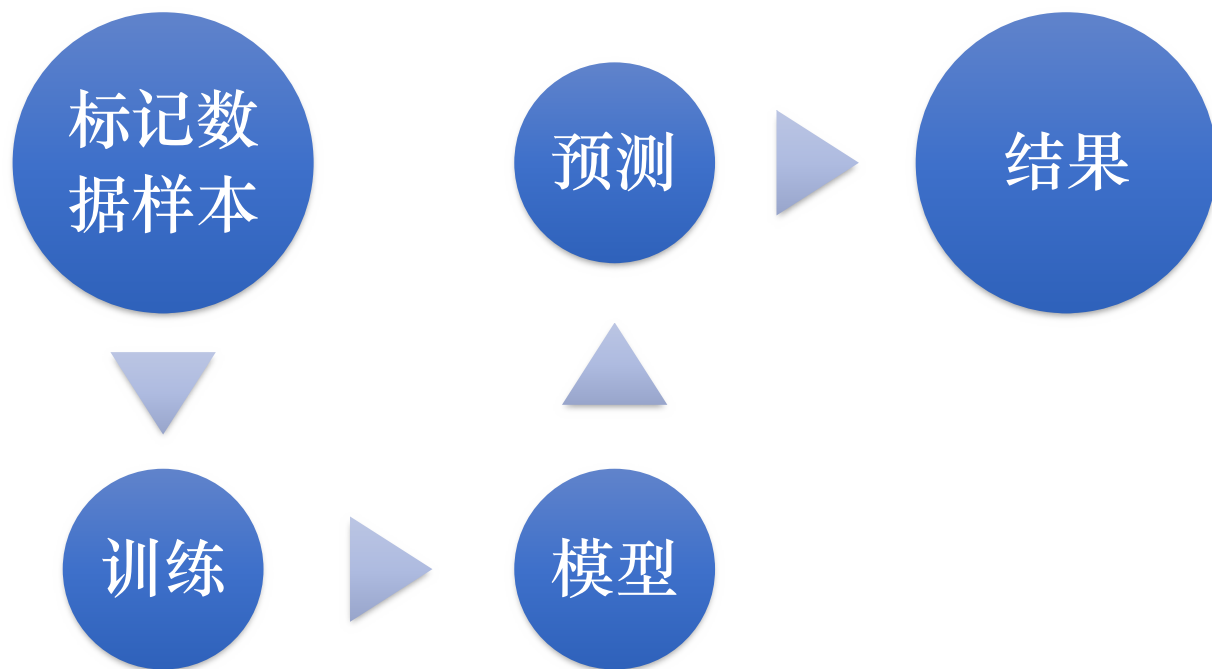
监督学习

- 利用标注好信息的样本，经过训练得到一个模型，可以用来预测新的样本



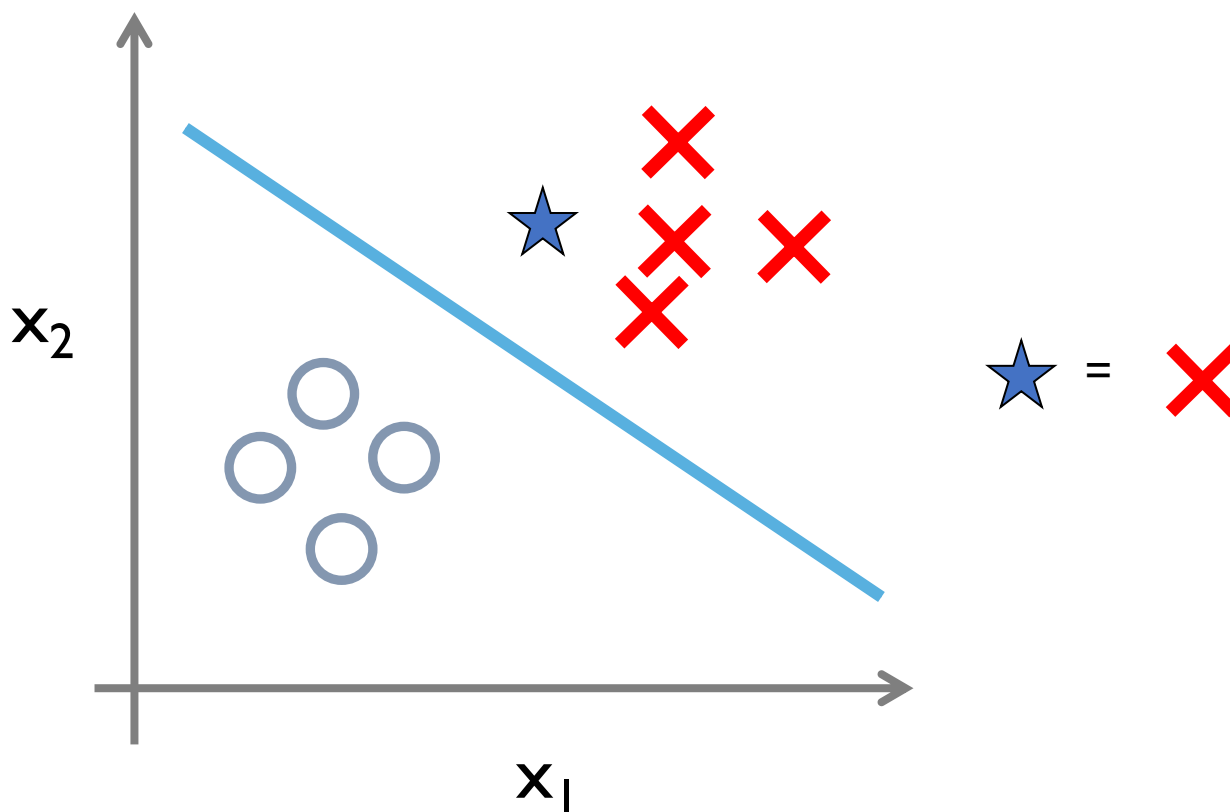
监督学习

- 利用标注好信息的样本，经过训练得到一个模型，可以用来预测新的样本



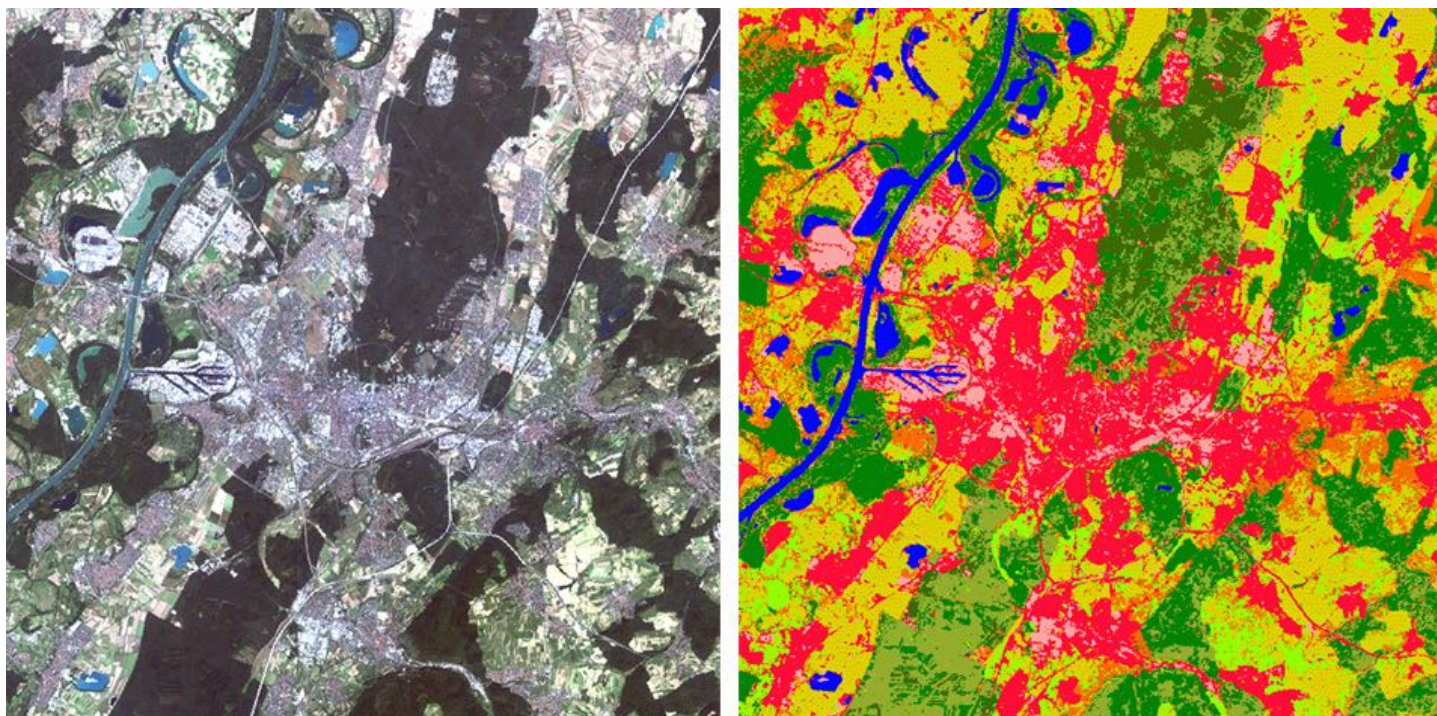
分类

- 当新来一个数据时，可以自动预测所属类型



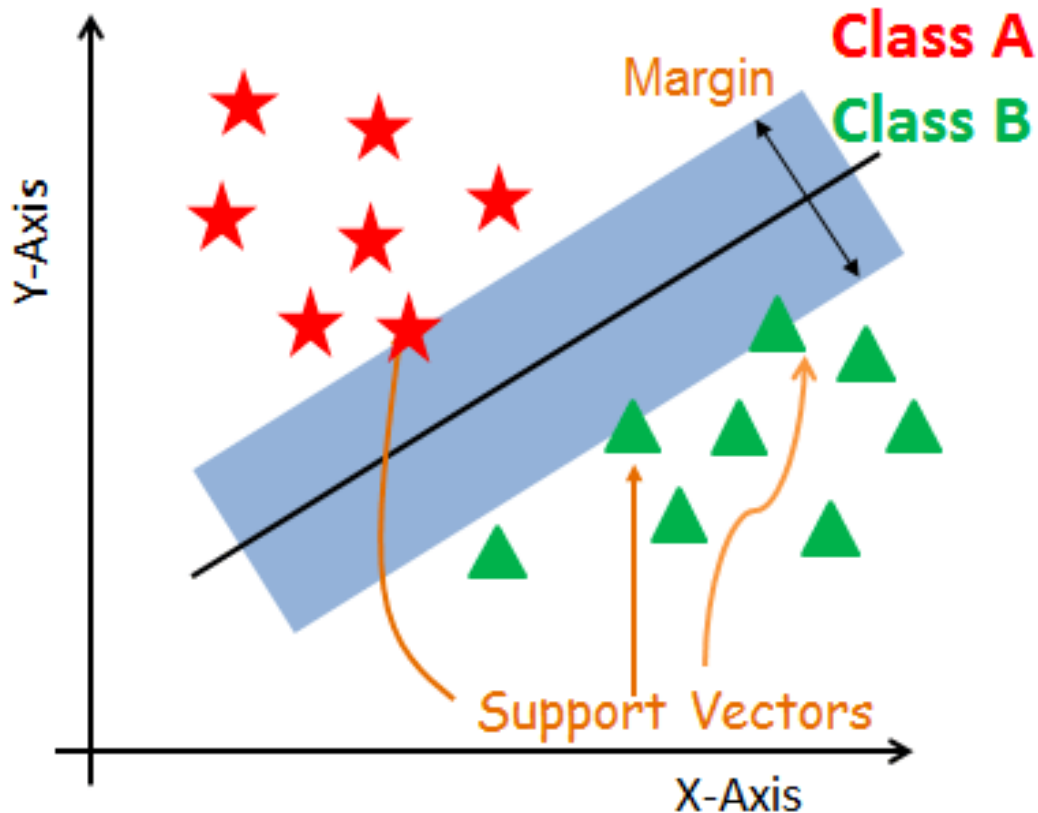
分类的应用

- 对于一幅遥感影像，对其中的部分水体、农田、建筑做好标记，通过监督分类方法就能得到其余的水体、农田、建筑。



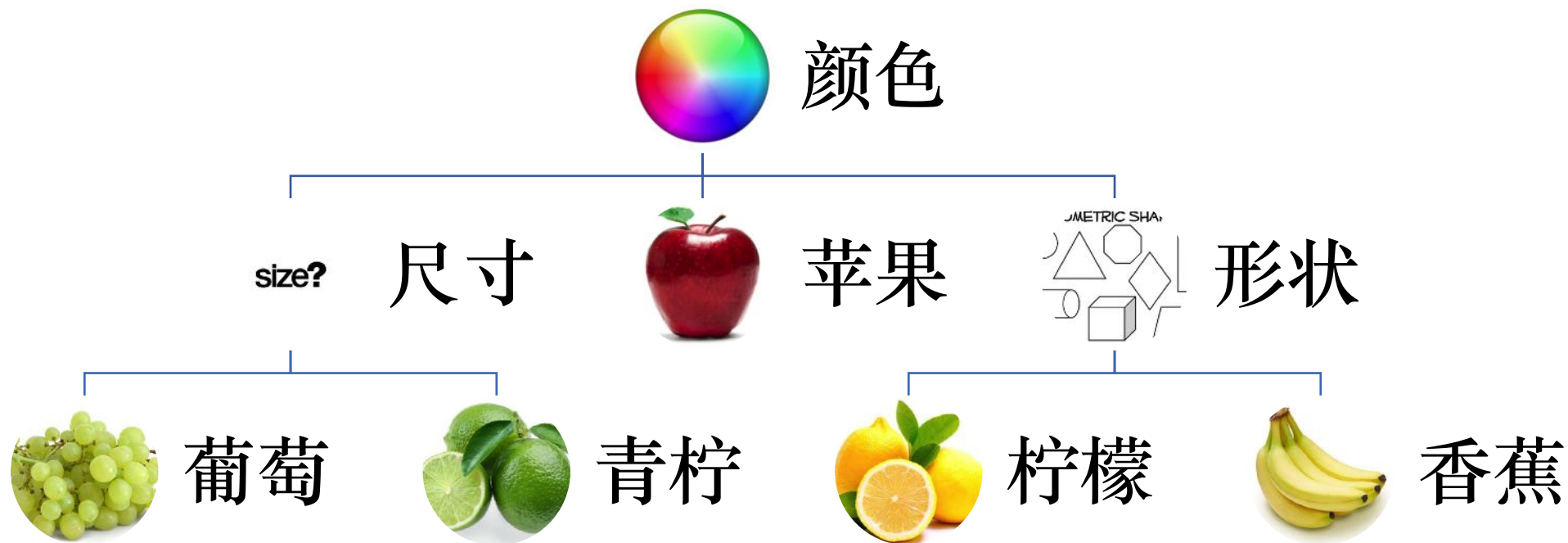
分类（预测结果是离散值）相关的方法

- 支持向量机：寻找最大化样本间隔的边界



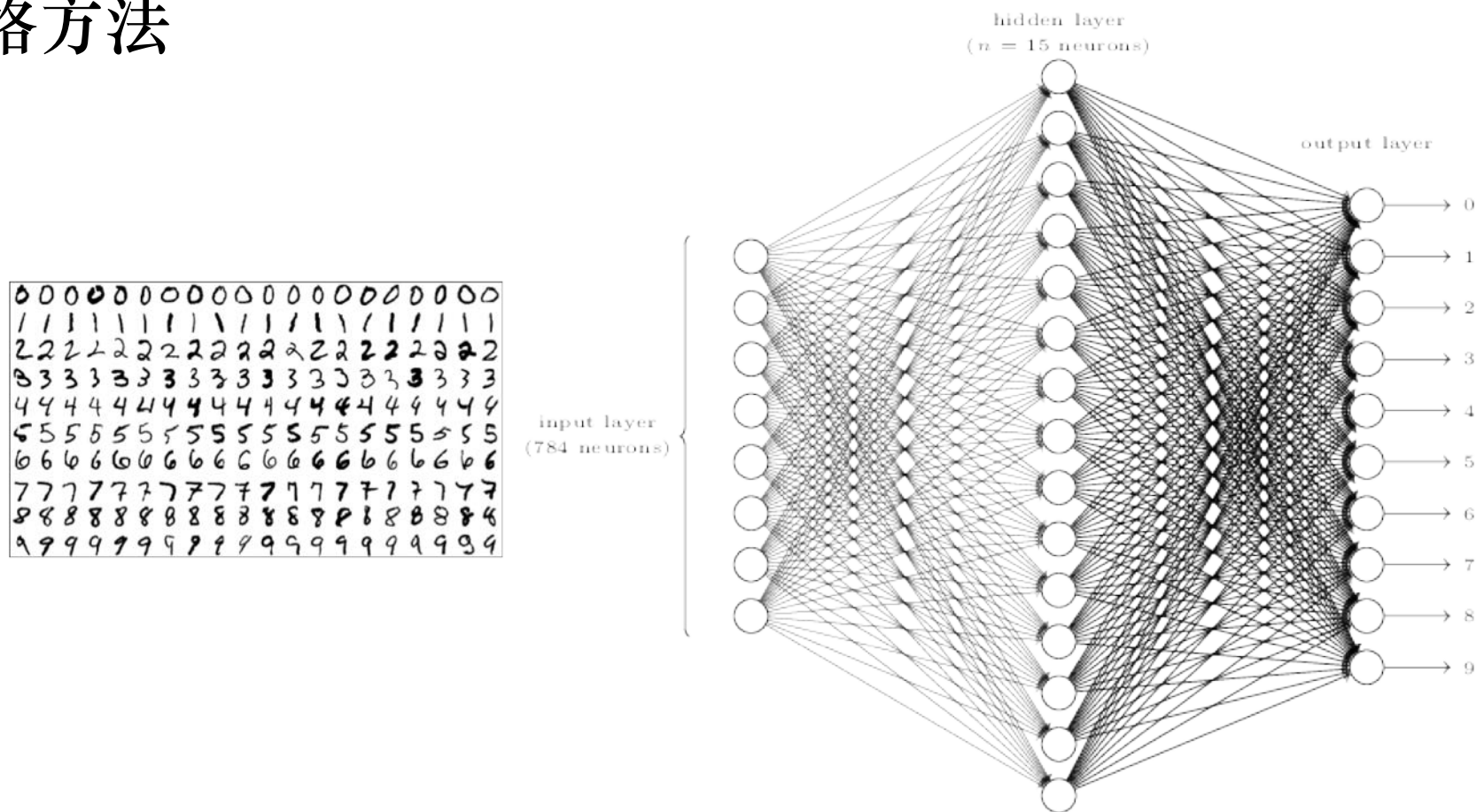
分类（预测结果是离散值）相关的方法

- 分类决策树



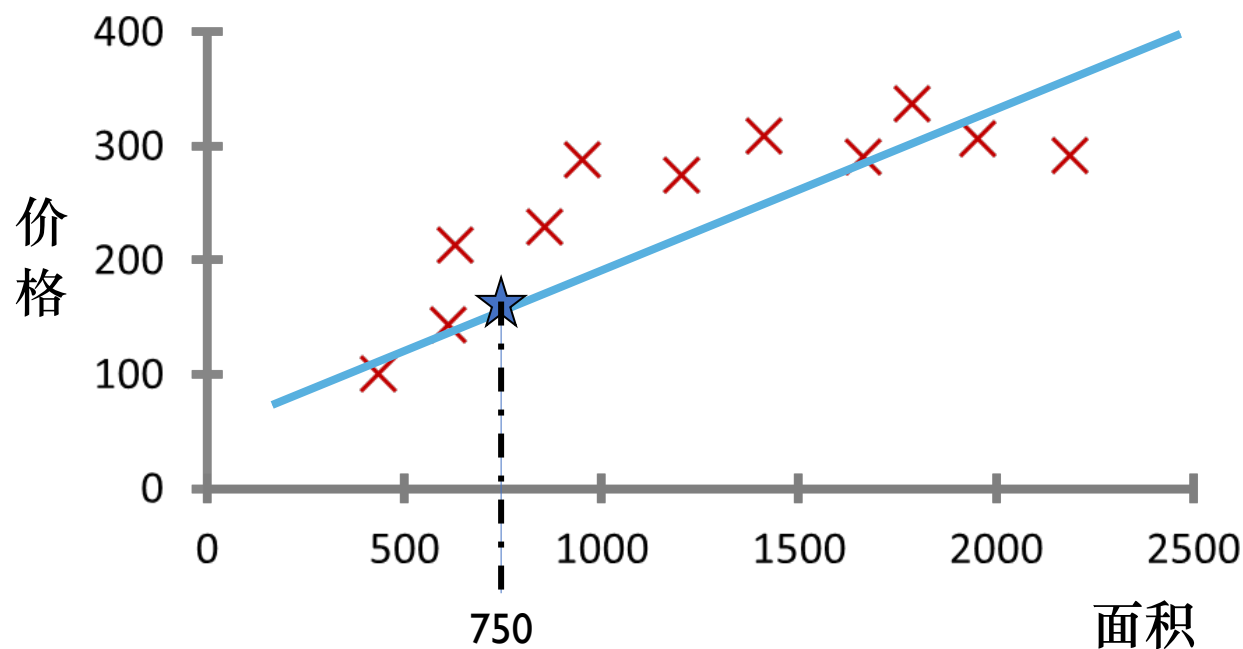
分类（预测结果是离散值）相关的方法

- 神经网络方法



回归

- 直线拟合（最小二乘法）
 - 我们希望通过已有的训练数据学习一个模型，当新来一个面积数据时，可以自动预测出销售价格



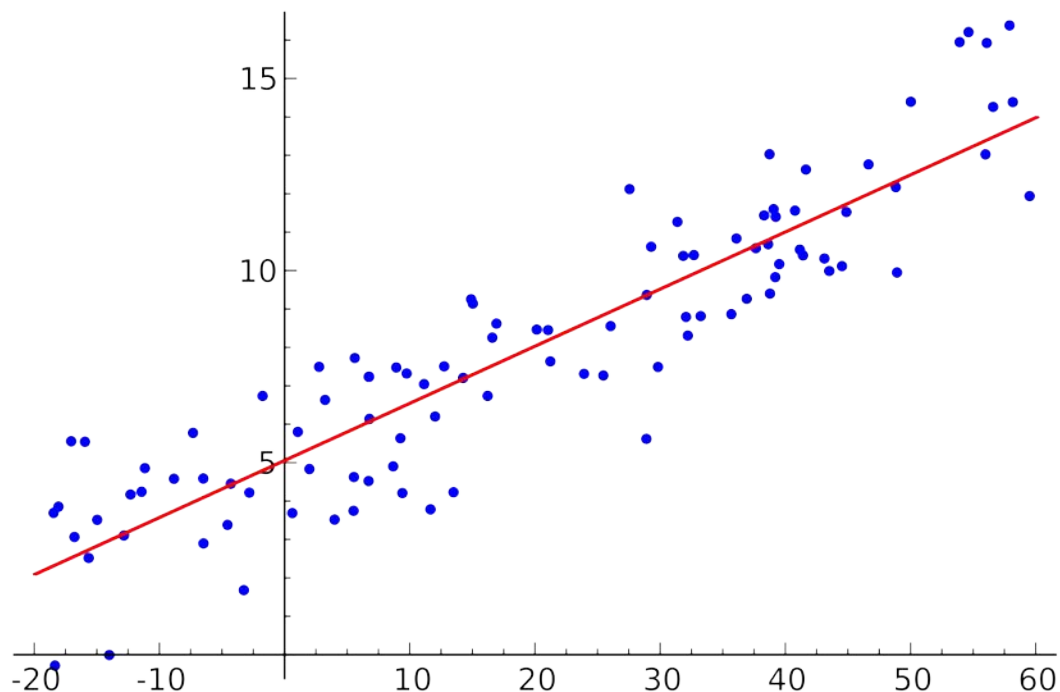
回归的应用

- 人脸好看程度评分。
- 通过标记分数的图片得出回归模型
- 输入新的图片就能得出分数。

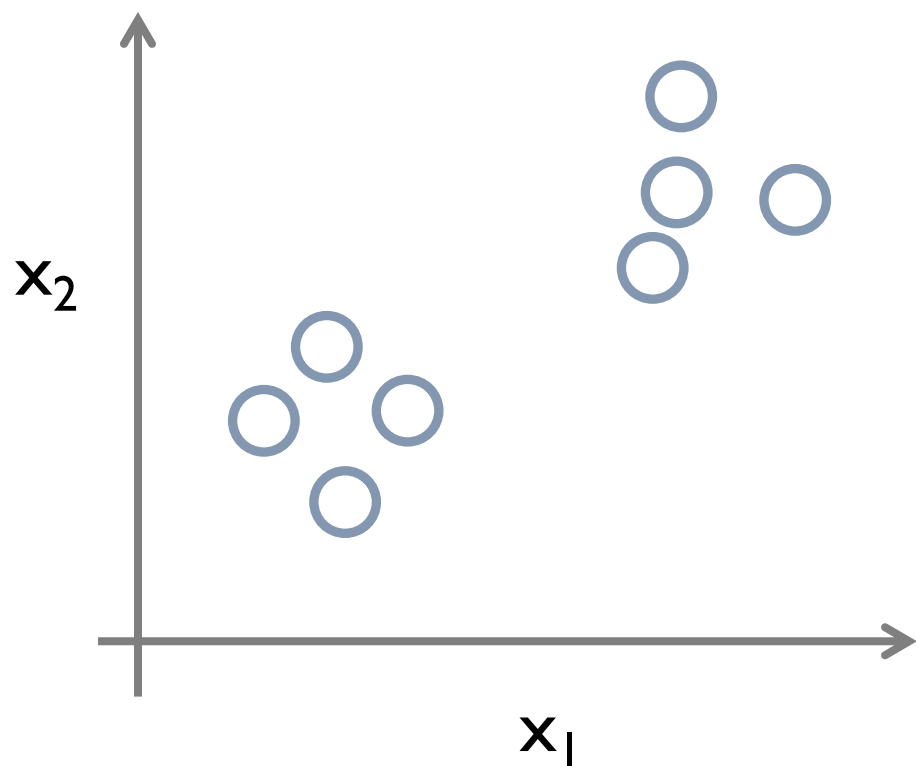


回归（预测结果是连续值） 相关的方法

- 线性回归
 - 在平面上拟合线性函数
- 最邻近方法
 - 使用最相似的训练样本来预测新样本值
- 神经网络方法

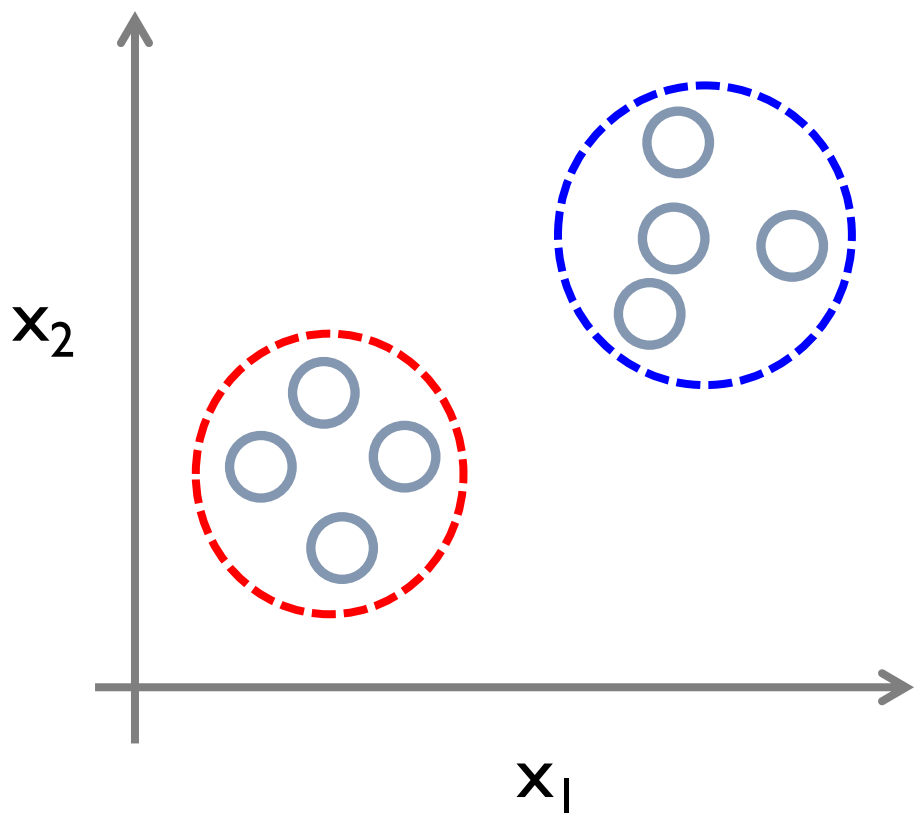


非监督学习



- 所有数据只有特征向量**没有标签**
- 但是可以发现这些数据呈现出**聚群**的结构
- 本质是一个相似的类型会聚集在一起。

聚类



- 把这些没有标签的数据分成一个一个**组合**，就是聚类 (Clustering)。

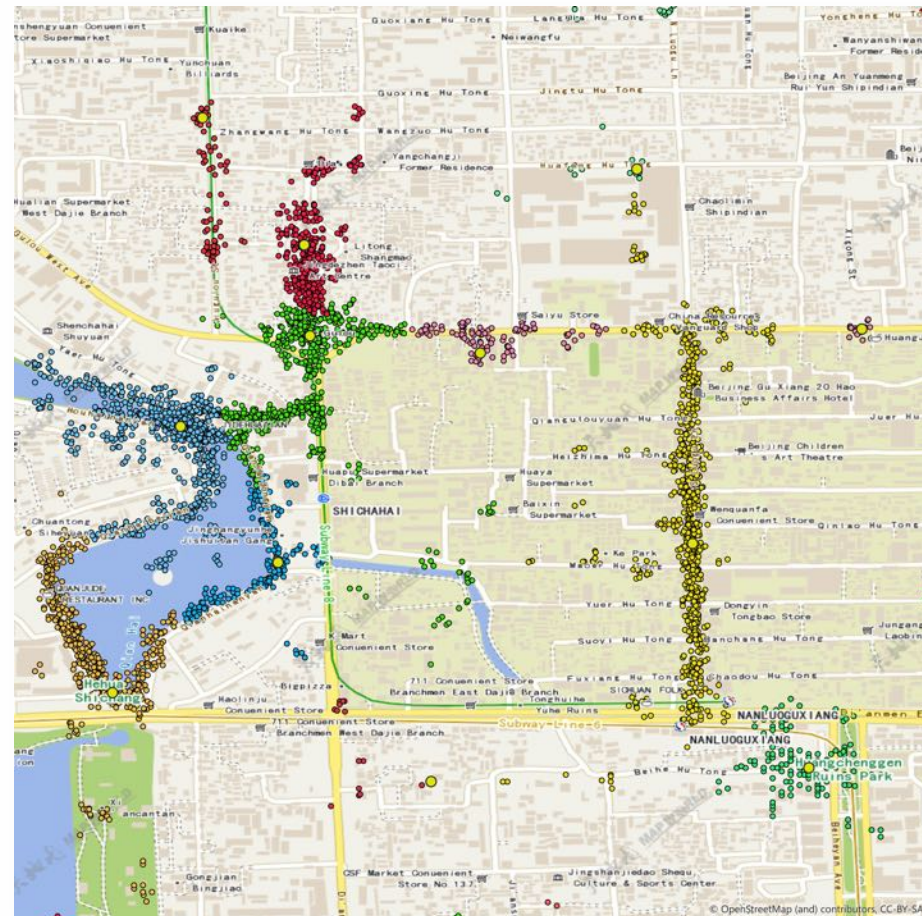
聚类案例：Google新闻

- 每天会搜集大量的新闻
- 然后把它们全部聚类
- 会自动分成几十个不同的组
（比如商业，科技，体育.....），
- 每个组内新闻都具有相似的内容结构



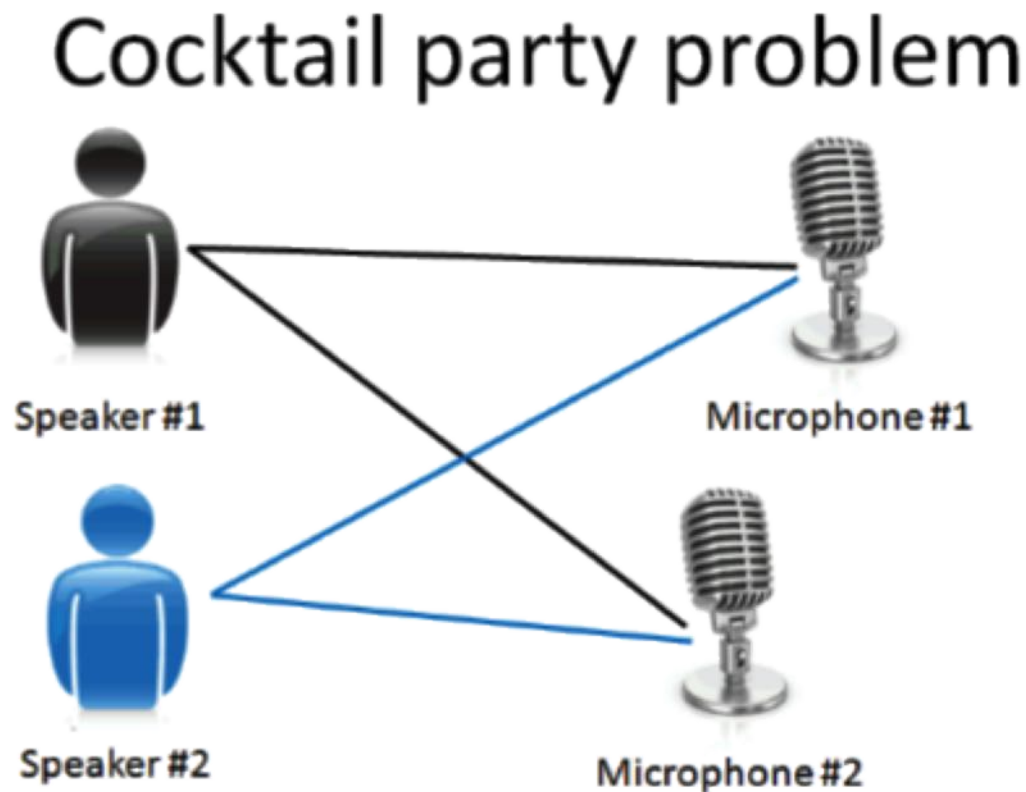
聚类案例：景区提取

- 对大量游客的微博定位点进行聚类
- 能够自动提取出不同的景点的位置分布



聚类案例：鸡尾酒会问题

- 在一个满是人的房间中，人们都在互相对话
- 我们记录下房间中的声音
- 利用非监督学习算法就能够识别房间中某一个人所说的话。



人工智能学玩游戏

- 让人工智能学会玩游戏是一项吸引人眼球的事情。
- 在棋牌类游戏或者FPS类游戏中，提供一个高性能的AI能增加游戏的挑战性；
- 而在另一些游戏中，比如模拟人生和Minecraft等，需要一个能够优化游戏体验的AI。

棋类游戏

- 从国际象棋到围棋，AI已经在棋类游戏中战胜了人类。



其他游戏

- DeepMind研发星际争霸2的AI
- OpenAI研发DOTA2的AI

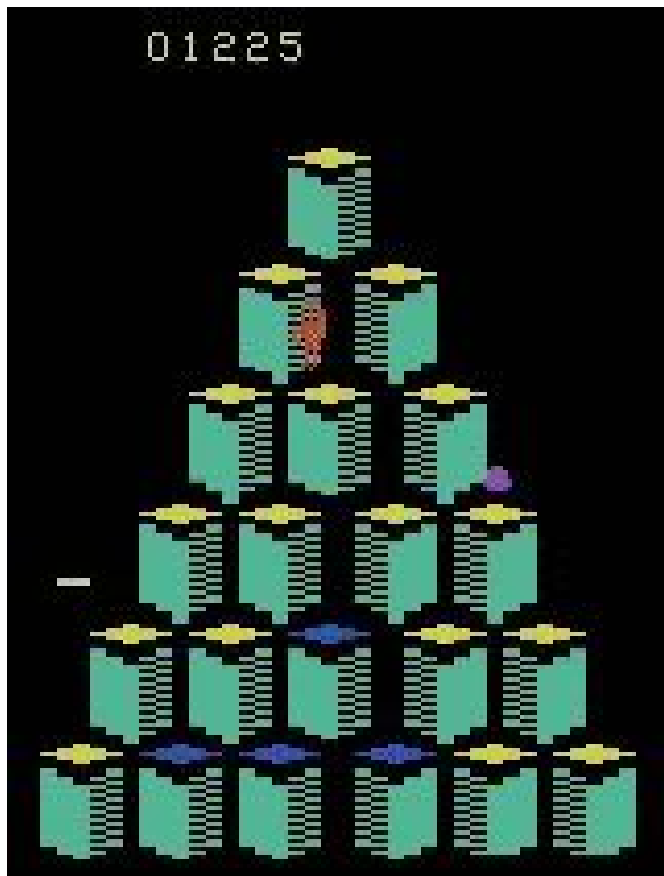


人工智能学玩游戏



- AI目前可以很好的掌握一些经典的小游戏
- 通过短时间的学习就能快速上手，有着优异表现。

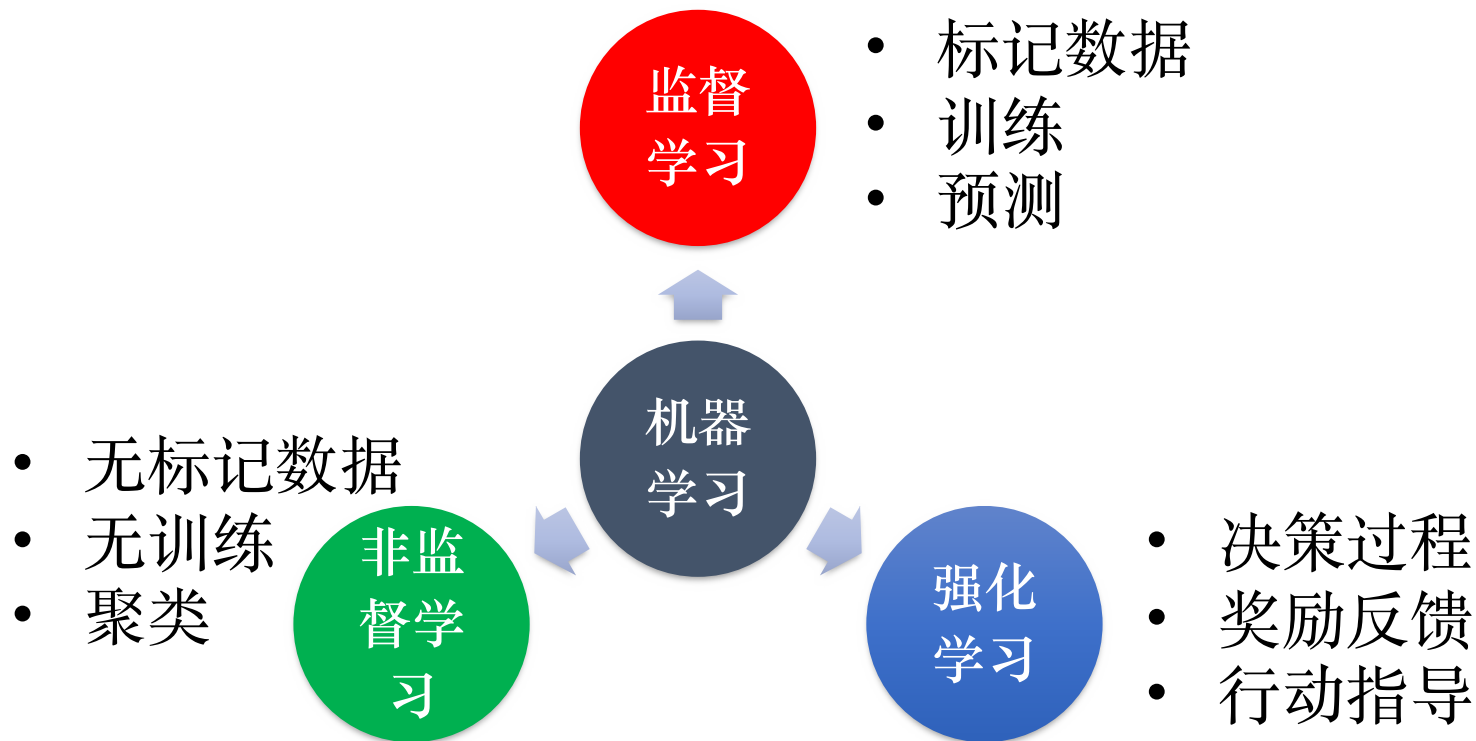
人工智能学玩游戏



- 甚至还会利用游戏中的 bug,
- 无需继续玩下一关就能令分数快速增长。

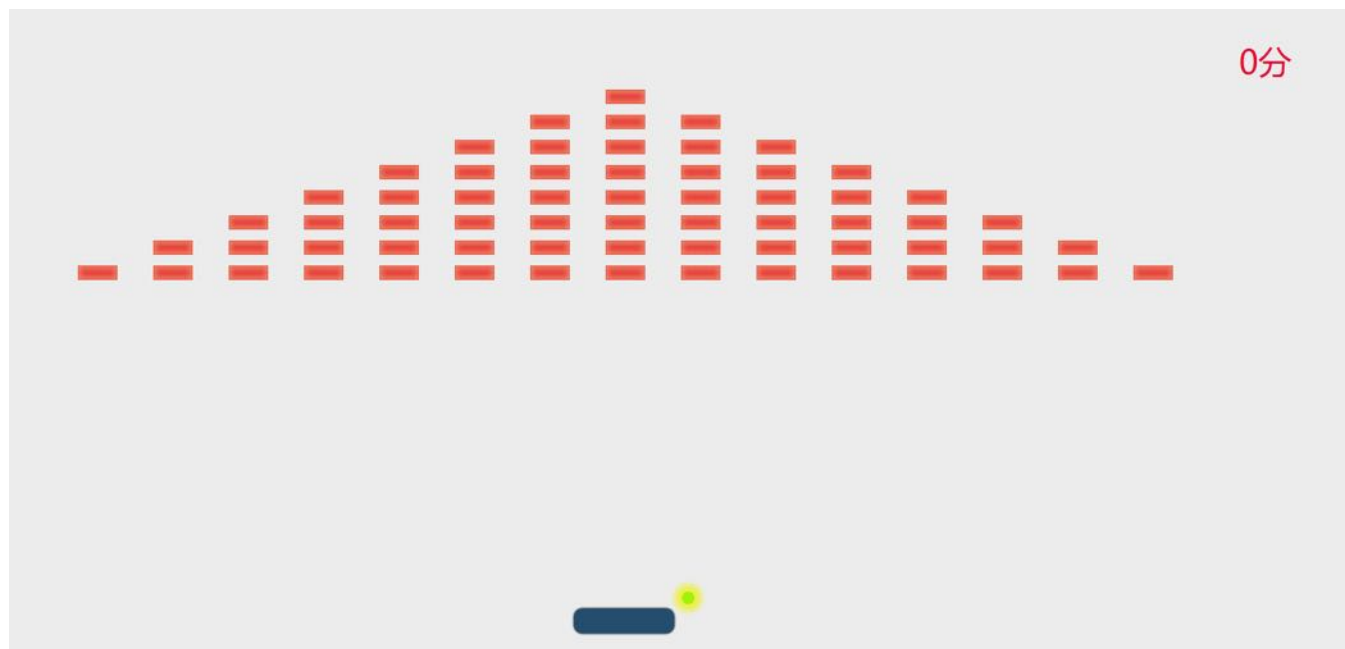
强化学习(Reinforcement Learning)

- 控制一个在某环境中的主体，通过与环境的互动来改善主体的行为。

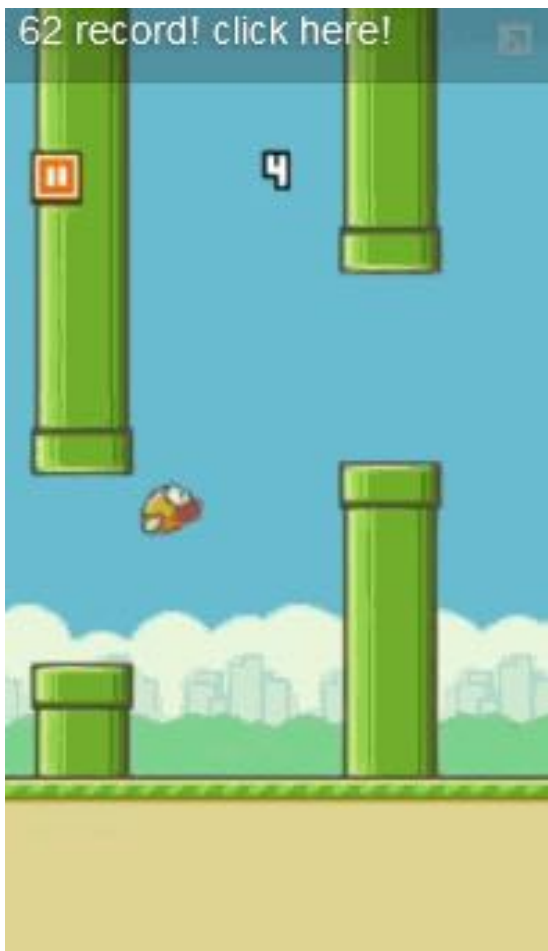


强化学习：符合学习玩游戏情形

- 通常游戏都是玩家控制一些角色
- 根据游戏画面的反馈来不断调整动作，
- 从而达到既定的目标（高分或者胜利）

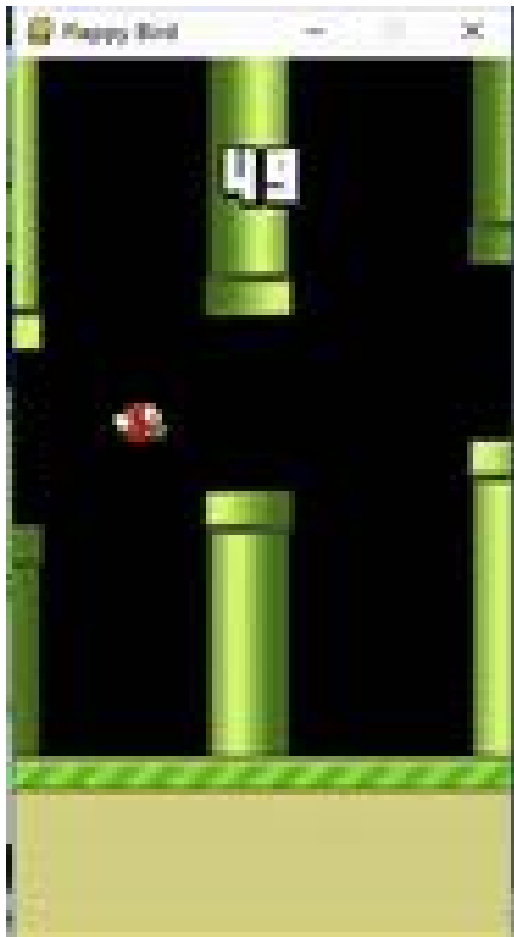


Flappy Bird



- 一款2013年5月发布的游戏，2014年1月，此游戏成为iTunes最受欢迎免费应用软件。
- 玩家操控小鸟飞行且避开绿色的管道
- 如果小鸟碰到了障碍物，游戏就会结束。每当小鸟飞过一组管道，玩家就会获得一分。

Flappy Bird



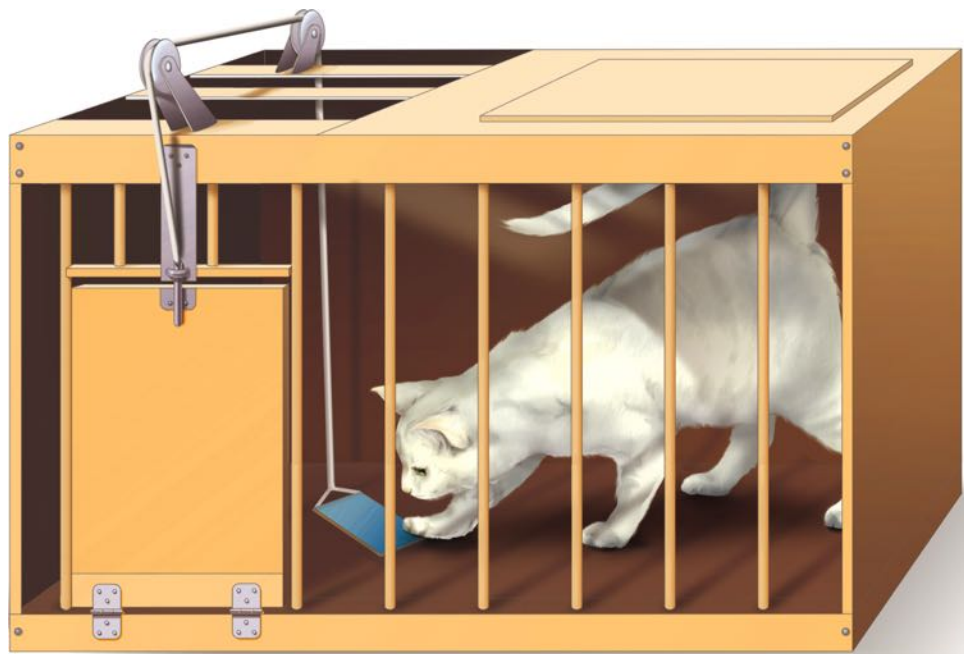
- 类似人的手眼配合学习
- 仅通过分析游戏时的截屏图像
- AI通过强化学习学会了玩 Flappy Bird

爱德华·桑代克



- 美国心理学家
- 现代教育心理学之父
- 心理学行为主义代表人物之一
- 提出试错式学习理论

机关盒子 (puzzle box)



- 将饿猫关入笼中
- 笼外放一条鱼
- 饿猫急于冲出笼门去吃笼外鱼，
- 但是要想打开笼门
- 饿猫必须一气完成若干个机关。

效果律 (law of effect)



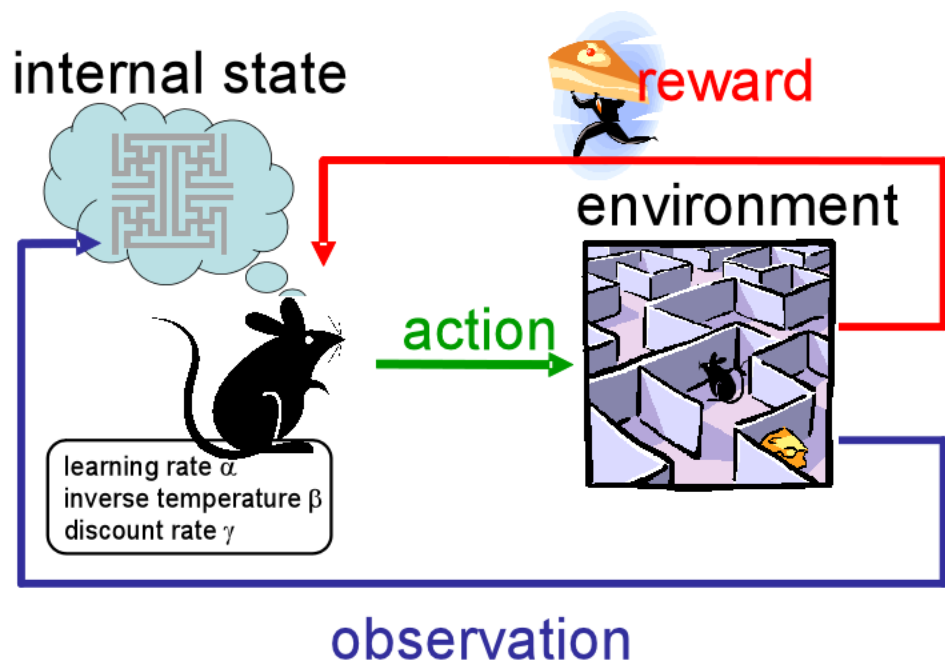
- 紧接着有利后果的行为更有可能再次发生。
 - 被老师称赞的工作或行为，你会继续保持。
- 不良后果的行为不太可能再次发生。
 - 如果你上课迟到并错过重要内容，之后就会吸取教训。

试错式学习 (trail and error)

- 猫的学习是经过多次的试误，由刺激情境与正确反应之间形成的联结所构成的。
- 人的学习的过程也是一种渐进的尝试错误的过程。在这个过程中，无关的错误的反应逐渐减少，而正确的反应最终形成。



强化学习 (reinforcement learning)



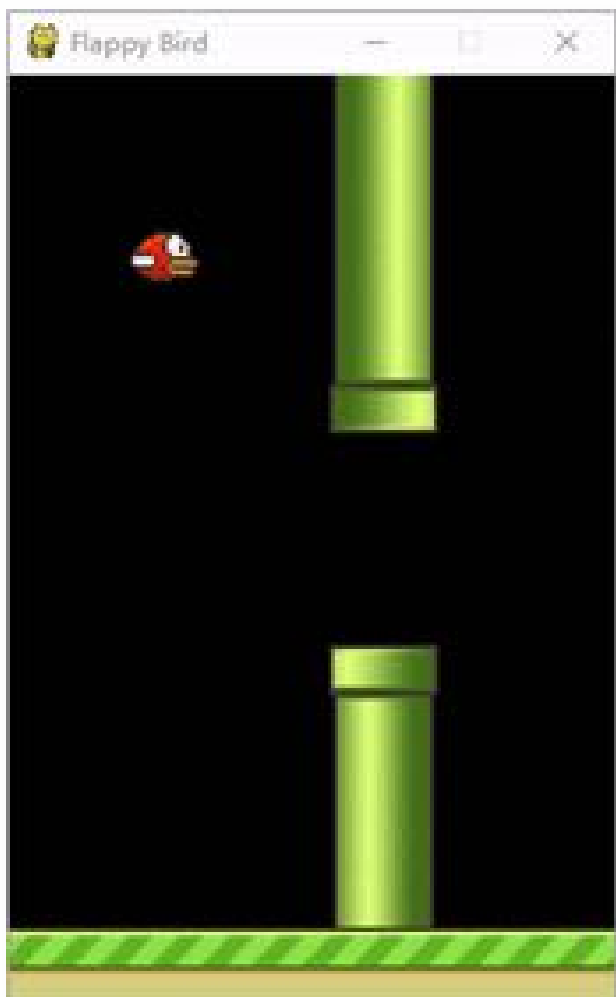
- 使得计算机能够像人一样通过不断试错式学习，完全自主掌握一项技能
- 不需要**借鉴**人类的经验
- 具有发展**强人工智能**潜力

Alpha Zero

- 利用试错式学习思想，自己跟自己不断对弈来提升水平
- 用这种通用的学习方式，在围棋、国际象棋、日本将棋等多个领域超越人类水平

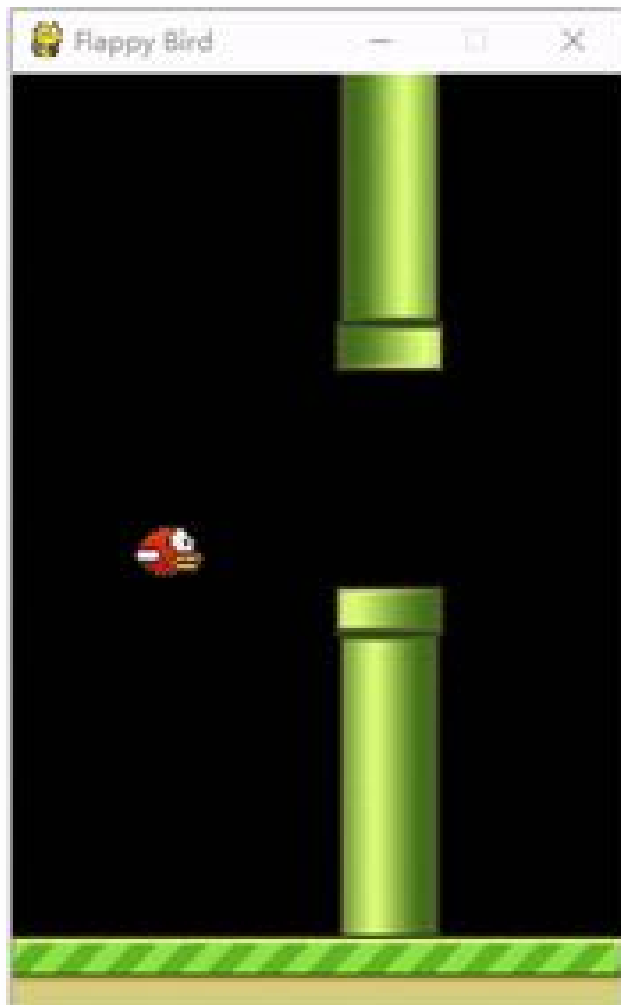


Flappy Bird



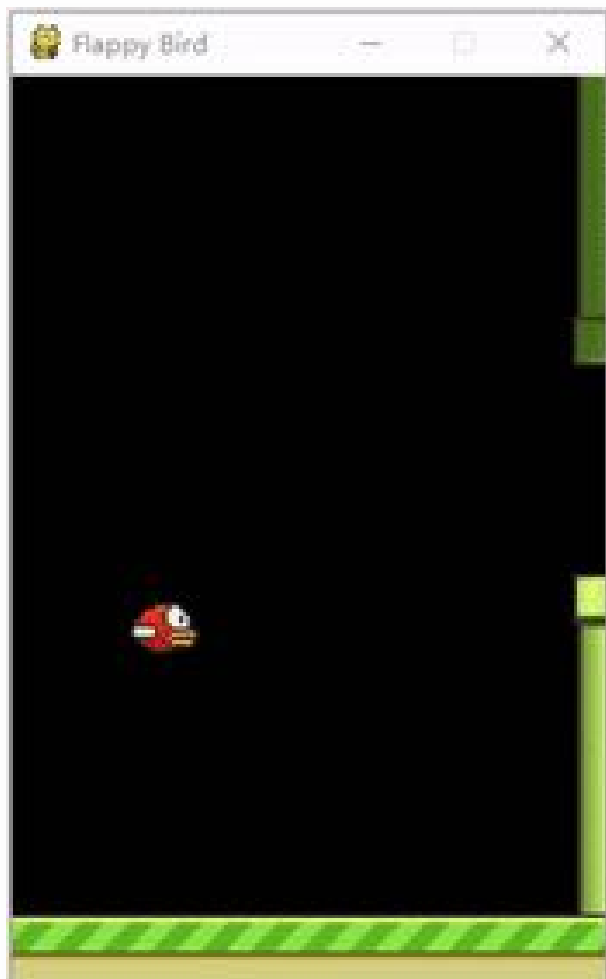
- 从零学习Flappy Bird

Flappy Bird



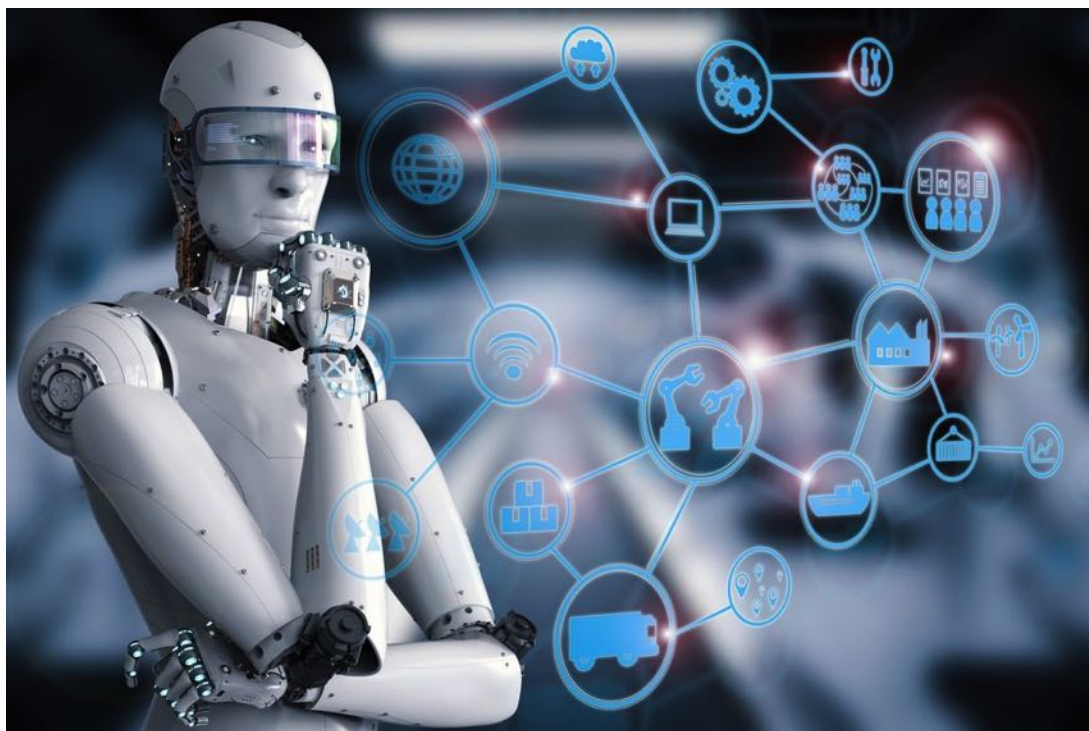
- 尝试10万次后，跨过第一根水管

Flappy Bird



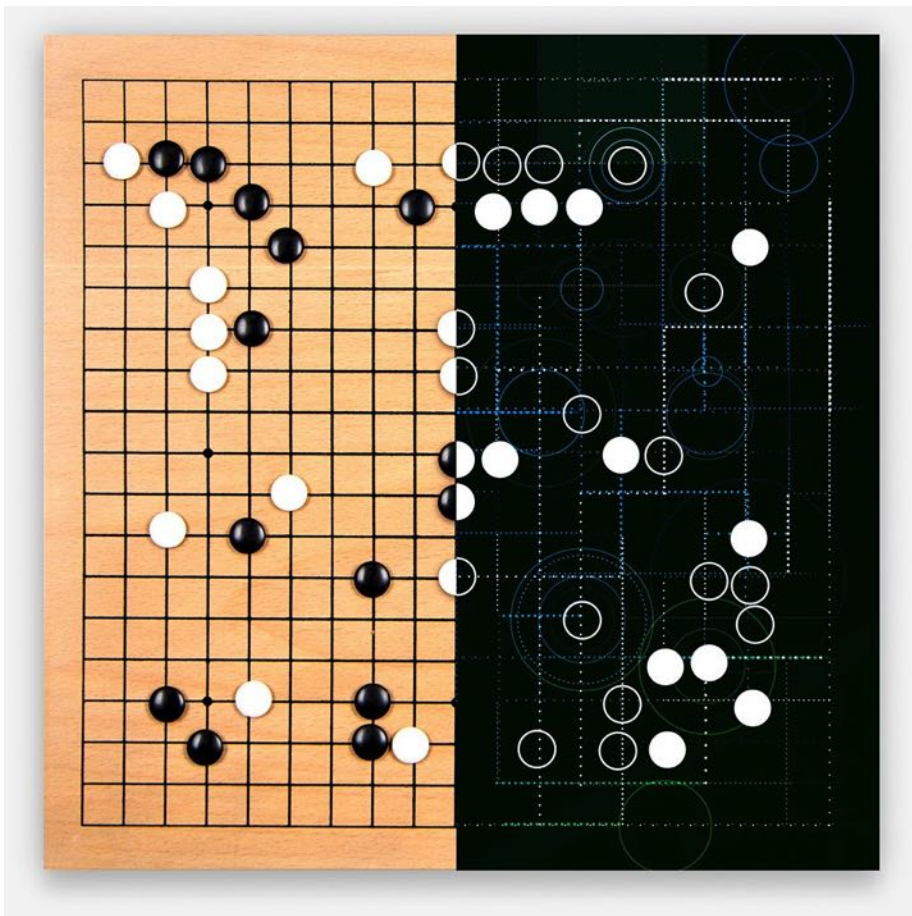
- 尝试150万次后，表现较高水平

强化学习的要素：主体Agent



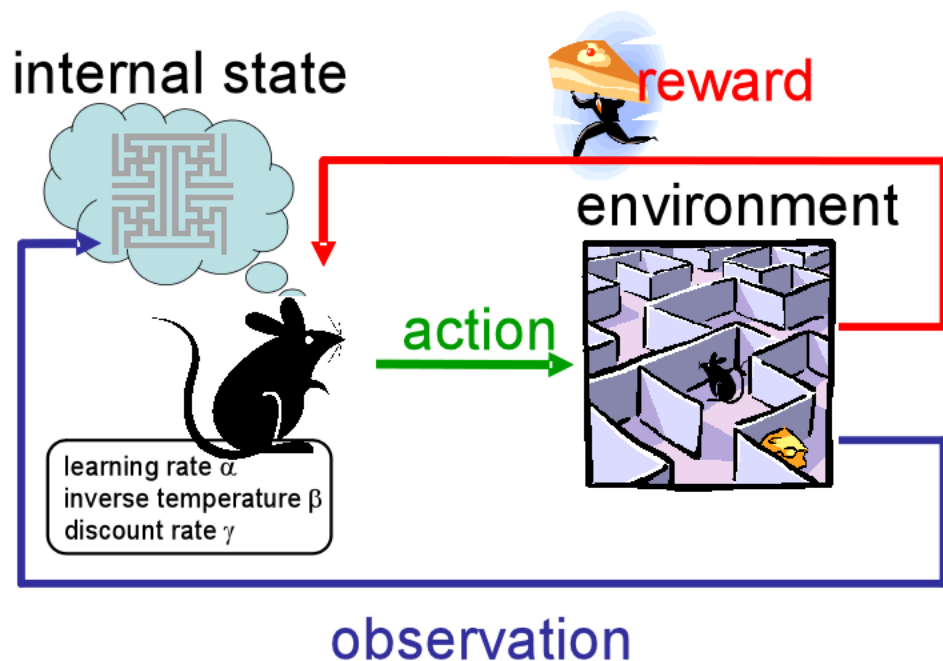
- 负责做出决策的实体。
 - 比如Alpha Go、人、玩flappy bird的AI。

强化学习的要素：环境（environment）



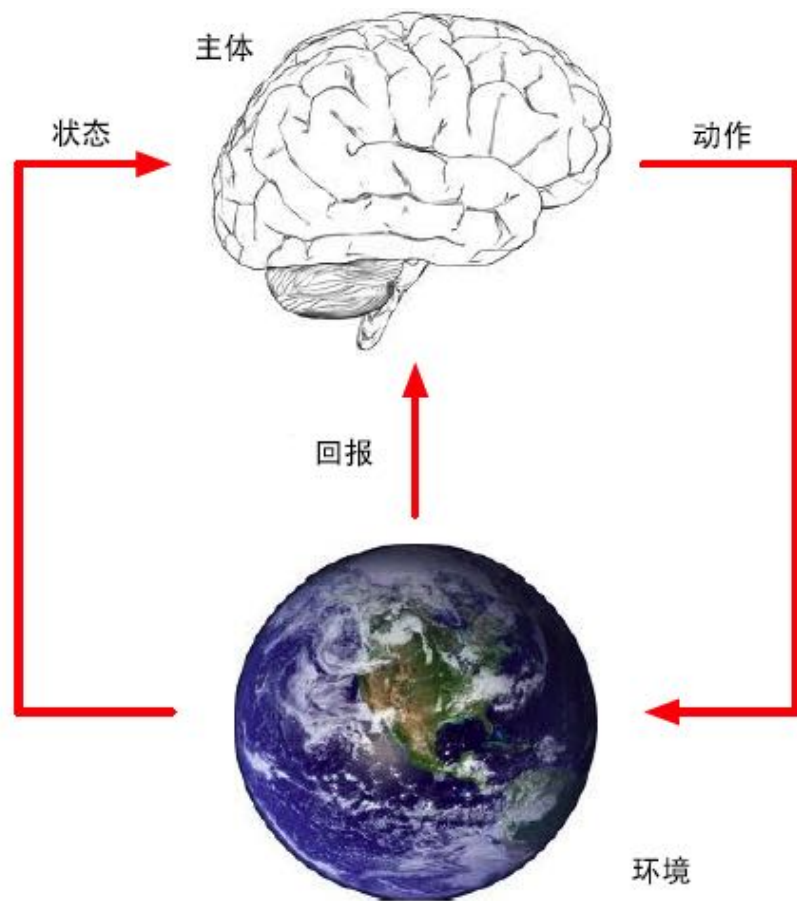
- 主体**存在**于环境之中；
- 主体观察环境**状态**；
- 主体的行为**作用**于环境；
- 并接受环境的**反馈**
 - 比如一个完整的游戏程序。

强化学习的要素



- 状态 (state)
 - 环境的状态不断发生变化。不同时刻的棋盘状况、游戏画面各不相同。
- 动作 (action)
 - 主体通过执行动作来改变环境的状态。
- 回报 (reward)
 - 环境状态改变之后会返回主体一个回报，主体可以根据回报来判断动作的好坏。

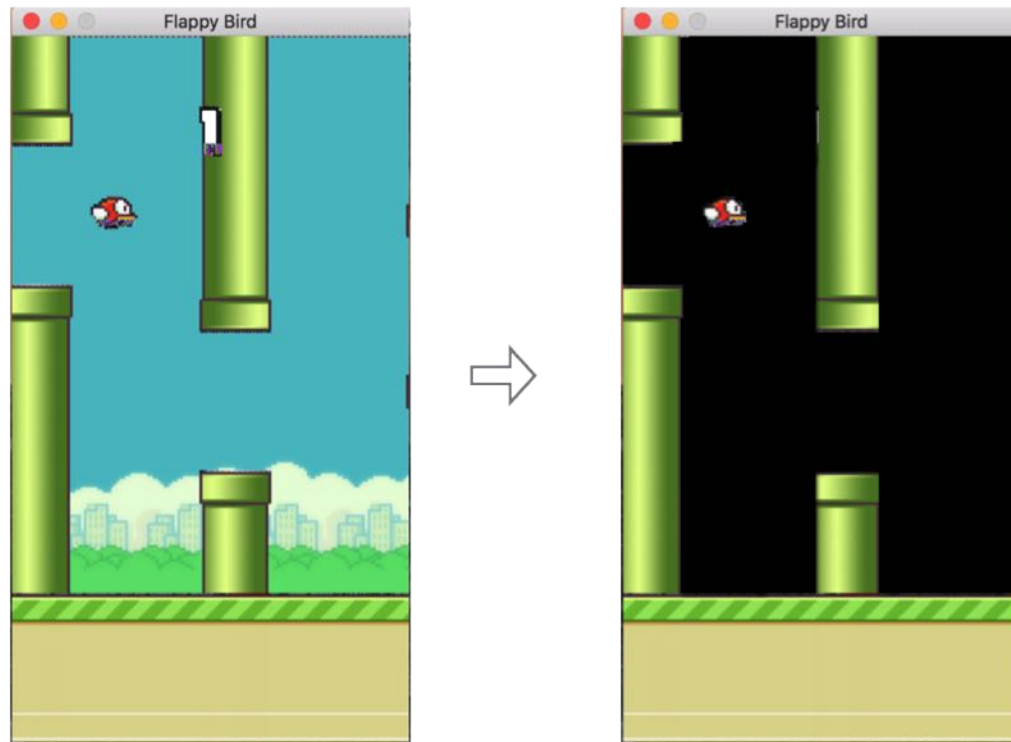
强化学习的主要流程



- 主体与环境不断地进行交互
- 产生多次尝试的经验
- 再利用这些经验去修改自身策略
- 经过大量迭代学习
- 最终获得最佳策略。

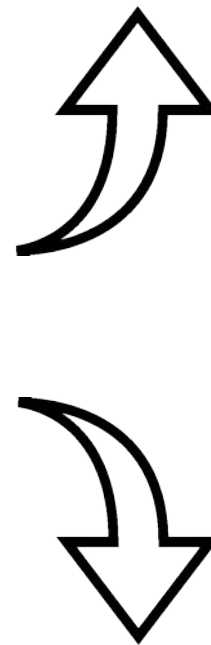
Flappy Bird: 状态

- 每一帧的画面都是一个状态
- 对画面进行简化
- 保留AI用于学习的关键信息



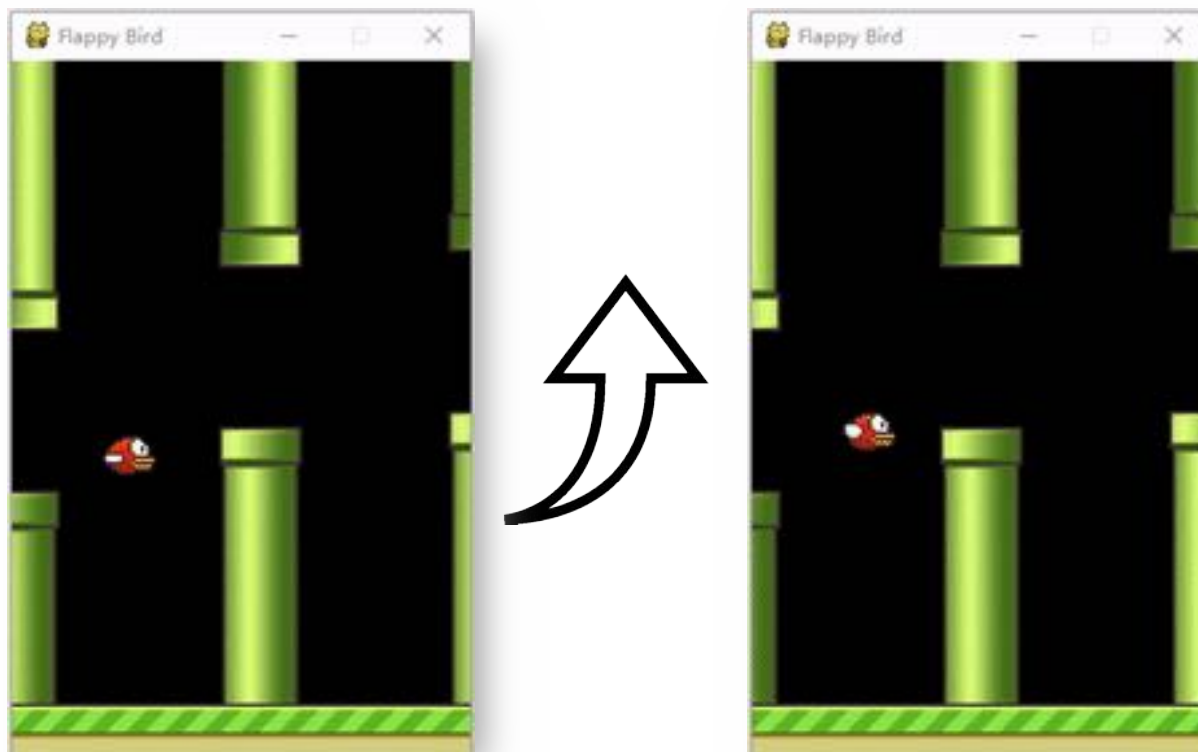
Flappy Bird: 动作

- 每个状态下都有两个可选择的动作
 - 让鸟往上跳 or
 - 什么都不做



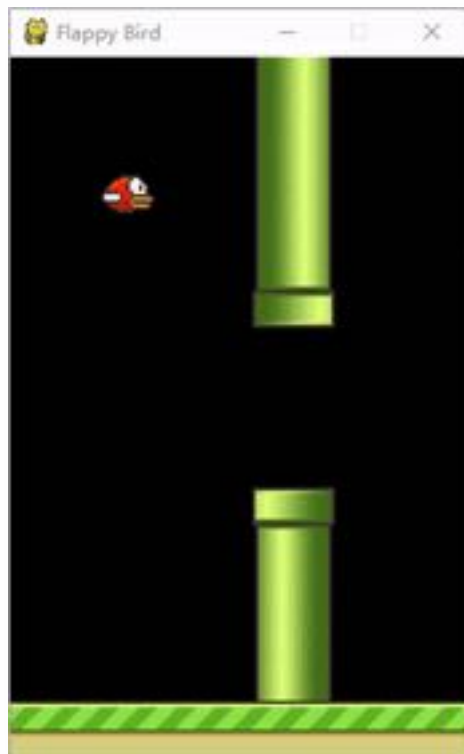
Flappy Bird: 动作

- 不同的动作会产生不同的新状态



Flappy Bird: 回报

- 主体在得到环境给的新状态时也会得到一个回报
- 简单情况下活着是1，死了是0

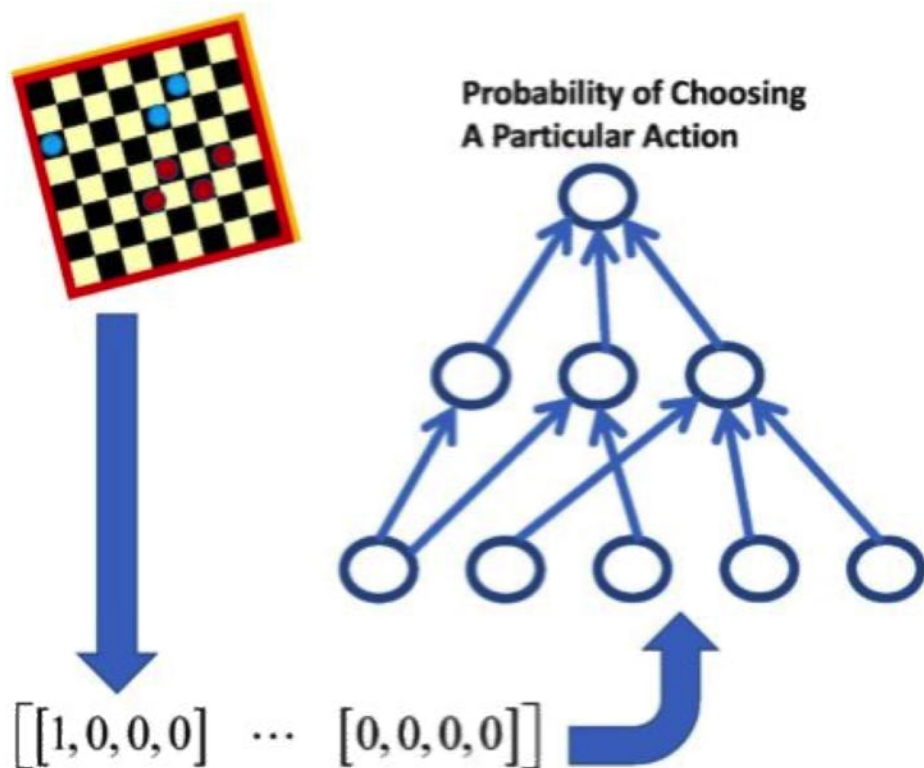


活着：1



死了：0

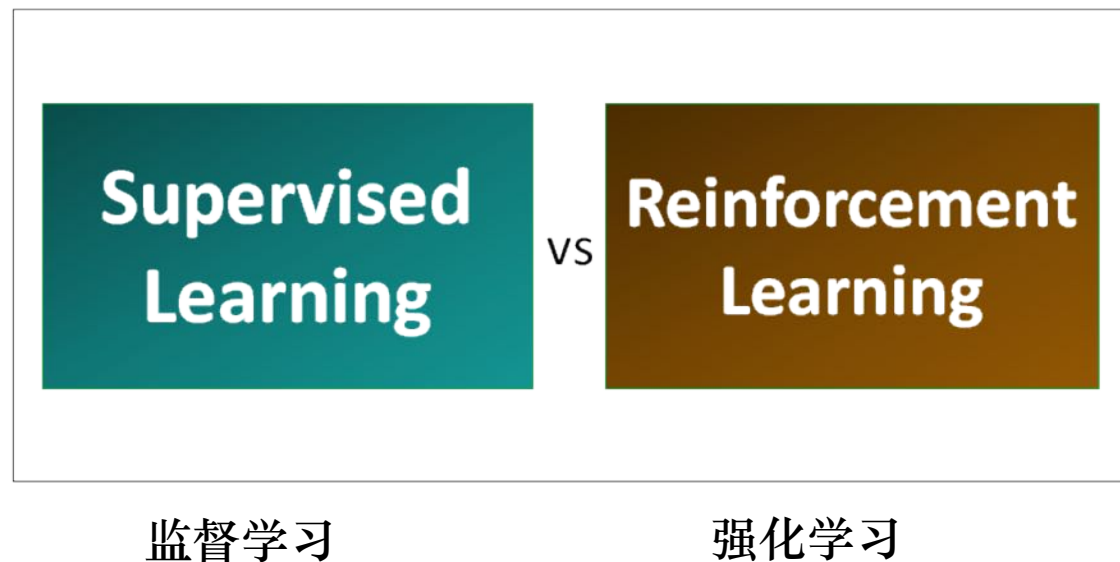
策略



- 从状态集（所有可能出现的状态）到动作集（所有可能采取的动作）的一个对应关系。
- 策略就是：某些状态下应该采取什么动作比较好
- 对于flappy bird游戏来说就是：
 - 在某个画面出现的时候
 - 是应该戳屏幕让鸟上飞
 - 还是什么都不做

目标：求得最佳策略

- 与手写数字识别不同，在强化学习中我们不关心把当前的状态分为什么类型，而是关心它能否**执行最佳动作**。



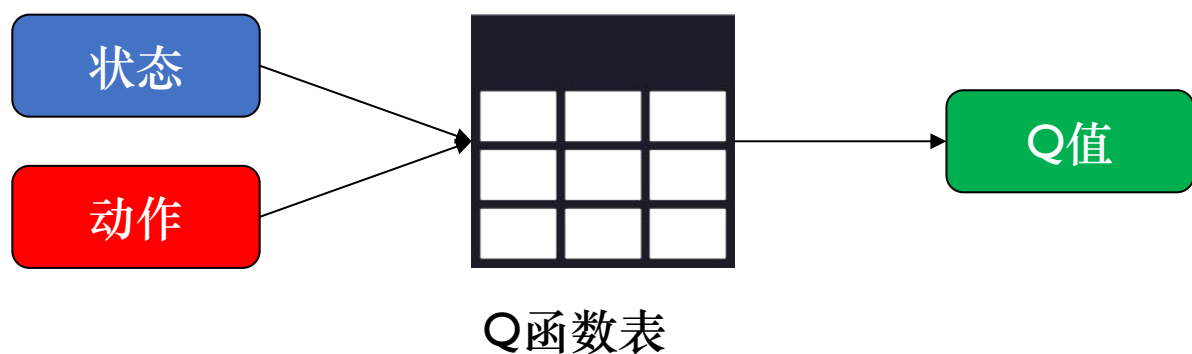
判断状态

- 状态值函数 V
 - 只和状态相关，用于对某个局面状态进行估值。
- 状态动作函数 Q
 - 和状态以及在该状态下采取的动作相关，用于对某个局面状态下采取某个动作进行估值。

Q-Learning

- 强化学习中一种常用算法。
- 基于状态动作函数 Q ，如果知道了某一状态下每个动作的估值，那么就可以选择估值最好的一个动作去执行了。

简单的Q函数表 (Q-Table)



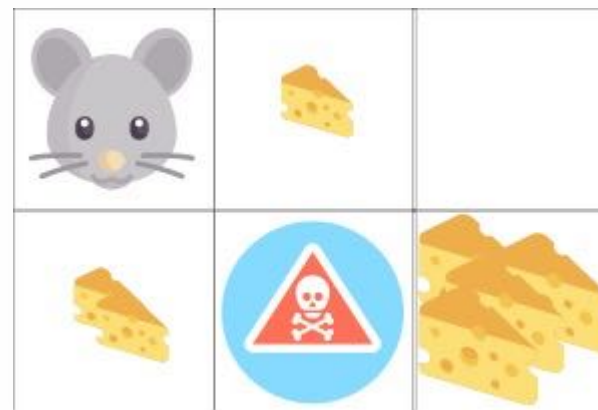
- Q函数表中行表示状态，列表表示动作，表中的值表示特定状态下执行某动作的评估值Q。
- 主体通过不断更新并查找该表，找到当前状态回报最高的动作执行。

简单的Q函数表（Q-Table）

- 示例

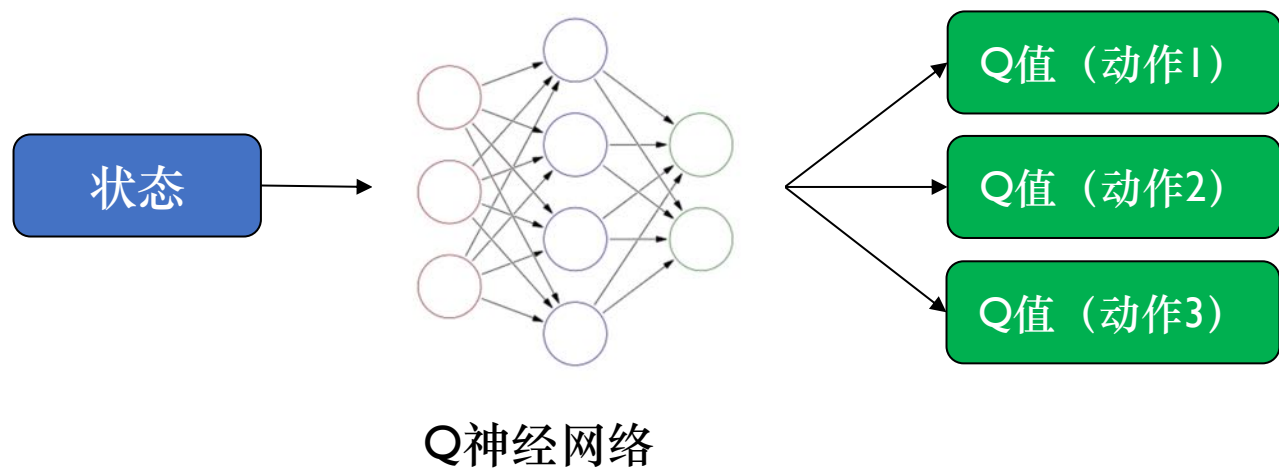
某个策略的Q函数表

状态\动作	上	下	左	右
开始点	0	20	0	10
一小块奶酪	0	-100	1	2
空白	0	100	10	0
两小块奶酪	5	0	0	-100
毒药	0	0	0	0
一堆奶酪（终点）	0	0	0	0



基于神经网络计算Q函数

- 对于复杂的状态，无法用表格表示，可使用神经网络对Q函数进行建模，其输入为状态，输出为各个动作的评估值。还是选取最高的动作执行。



总结

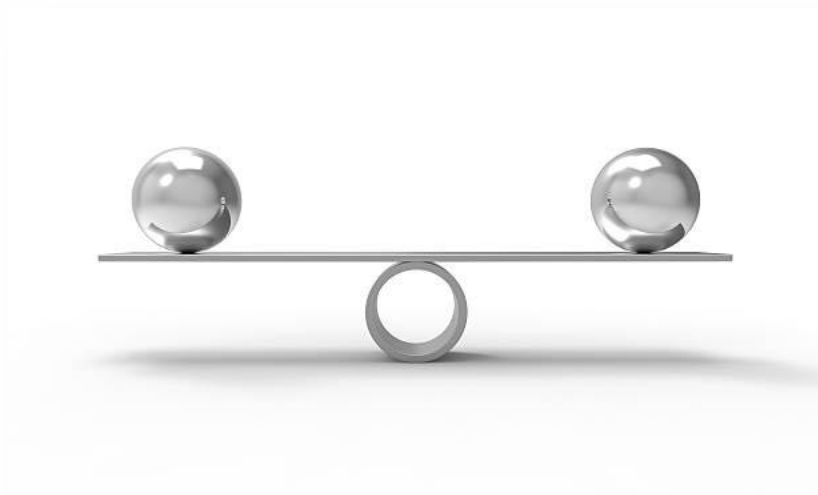
- Q-Learning算法通过学习获得一个状态动作函数（Q函数）
- 不直接决定主体该采取什么决策，而是提供一个估值参考。
- 如果Q函数较优，可以直接取最大价值来决定动作。

从零开始

- 刚开始时并不知道正确的策略以及Q函数应该是多少。
- 初始化一个随机Q函数，从零开始不断学习。



如何尝试



- 在Q函数不够准确的时候，每次尝试该如何选择动作？
- 涉及到探索 and 开发两者的平衡。

探索 (Exploration)



- 为了更好地学习最佳Q函数而尝试各种情况。
- 也就是说，应该选择**不同**的其它动作。
- “尝遍百草”

开发 (Exploitation)

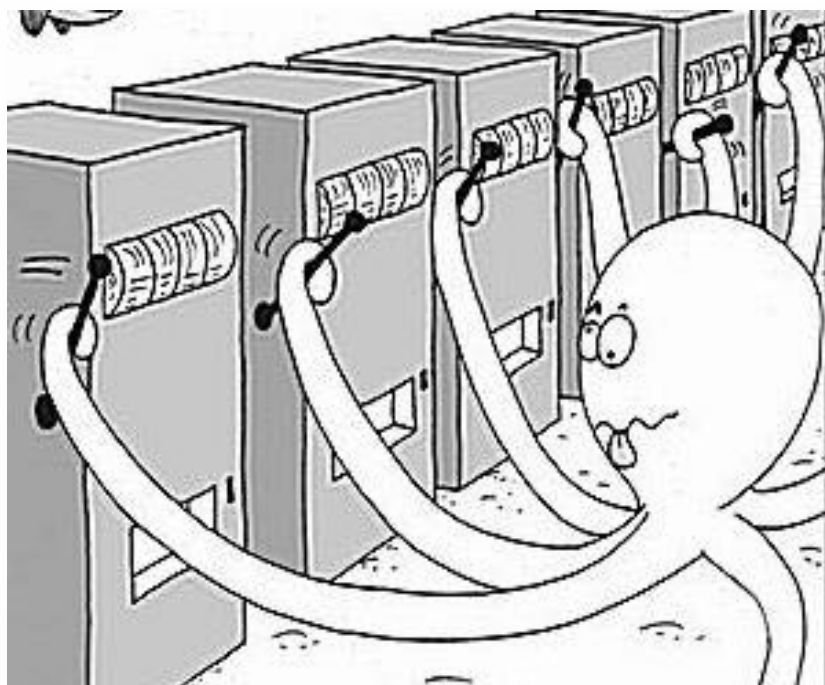
- 直接选择当前认为**最佳**的动作。
- 再进一步修改新状态下的Q值。



探索与开发

- 探索：随机的生成一个动作
 - 探索未知的动作会产生的效果，有利于更新Q值，获得更好的策略。
- 开发：根据当前的Q值计算出一个最优的动作（greedy）
 - 相对来说就不好更新出更好的Q值，但可以得到更好的测试效果用于判断算法是否有效。

ϵ -greedy策略

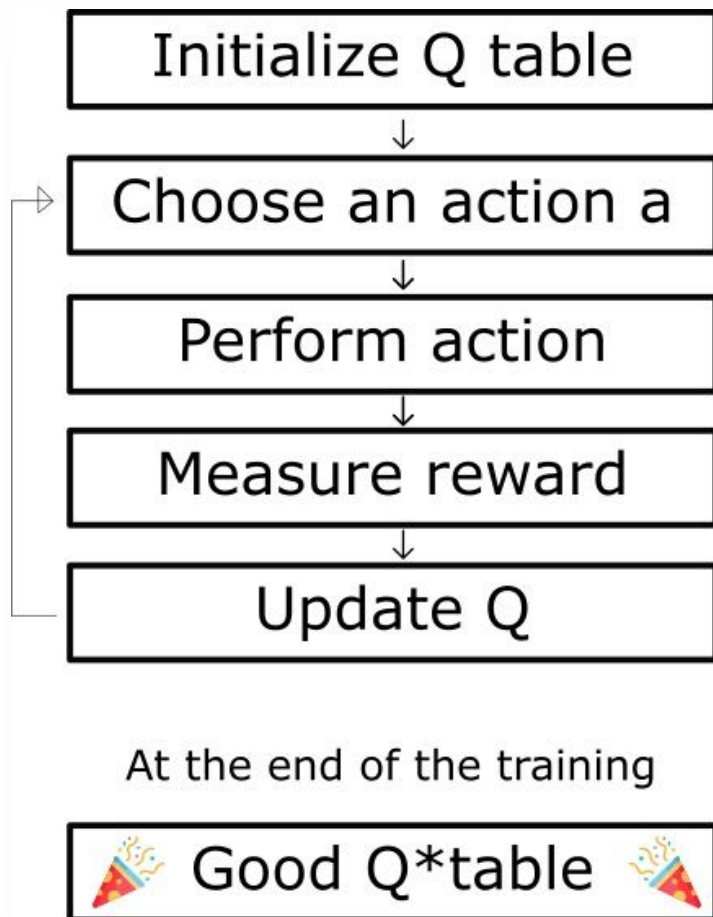


- 一种简单的平衡探索与开发的策略
- 有 ϵ 概率选取一个随机动作，剩下的情况依然选取Q值最大的动作。
 - ϵ 一般是一个很小的值，表示不断尝试的趋势。

ϵ -greedy策略

- 可以更改 ϵ 的值从而得到不同的探索 and 开发的比例。
 - 通常在刚开始学习时，可以稍微调大 (0.01)
 - 在Q函数逐渐优化后，可以调小 (0.001)
 - 甚至可以直接设为0，不再探索。

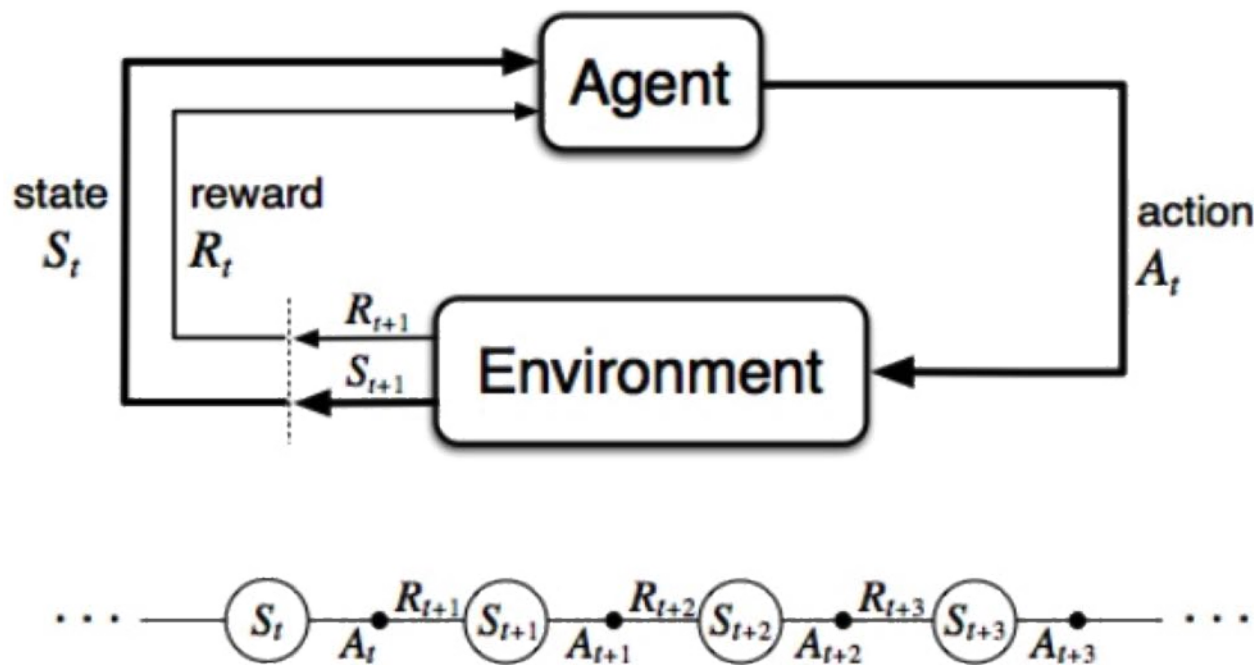
学习流程



- 初始化Q函数
- 不断重复每一局游戏
 - 选择动作
 - 得到回报
 - 更新Q函数
- 最终得到一个好的Q函数

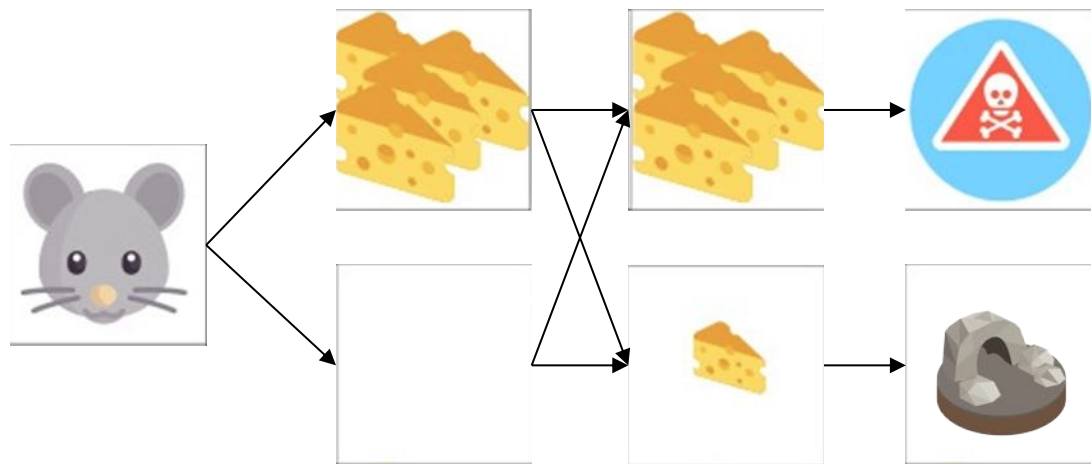
动作-状态序列

- 每一局游戏都是一个动作状态序列
- 下一个状态只和当前的状态+动作有关（马尔可夫性质）



长期回报

- 除了试错式搜索之外，强化学习的另一个重要的特点是回报的滞后性。
- 当前状态下的动作所产生的回报不仅取决于下一个状态，还取决于整个序列之后的每一个状态。



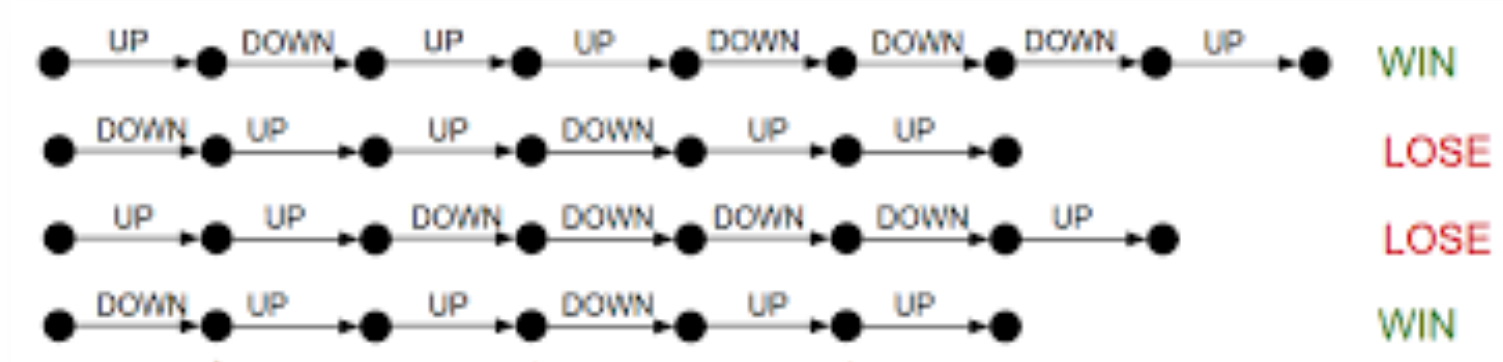
回报率

- 当前的动作对下一状态的影响是最直接的，对后续状态影响没那么直接。
- 某些动作产生的当前回报值比较高，但从长远来看，可能并没有那么高。
- 因此我们用一个回报率来平衡下一状态回报和更远状态回报。

$$0.9 \times \text{状态1} + 0.81 \times \text{状态2} + 0.729 \times \text{状态3} + \dots$$

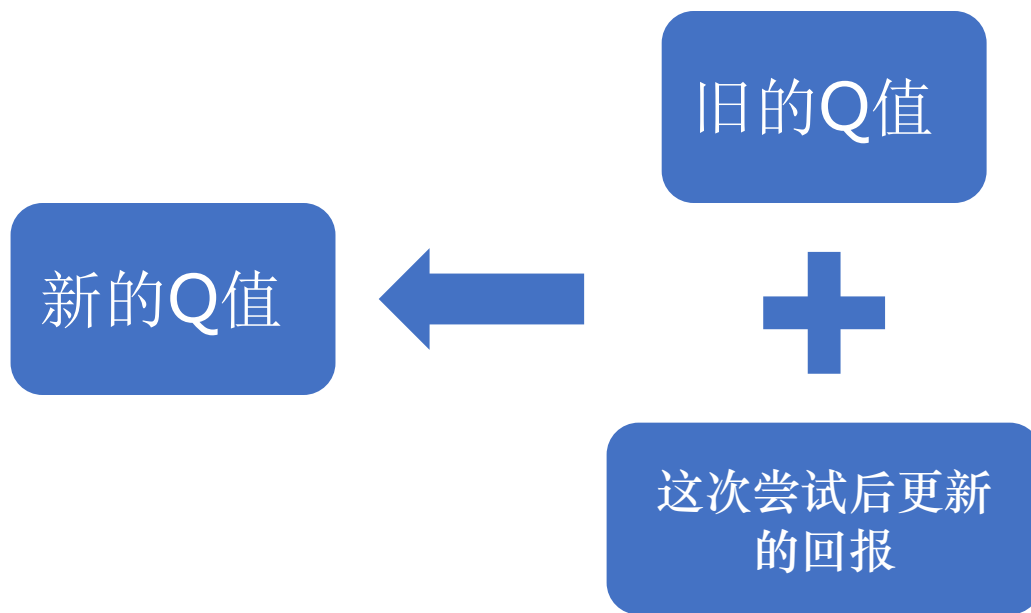
回报函数

- 每一次游戏会产生不同的状态动作序列，即每一次对后续状态的回报计算都不相同。
- 我们用后续状态的期望，即所有之后的序列的回报平均值作为回报函数。
- 回报函数值就是Q值。



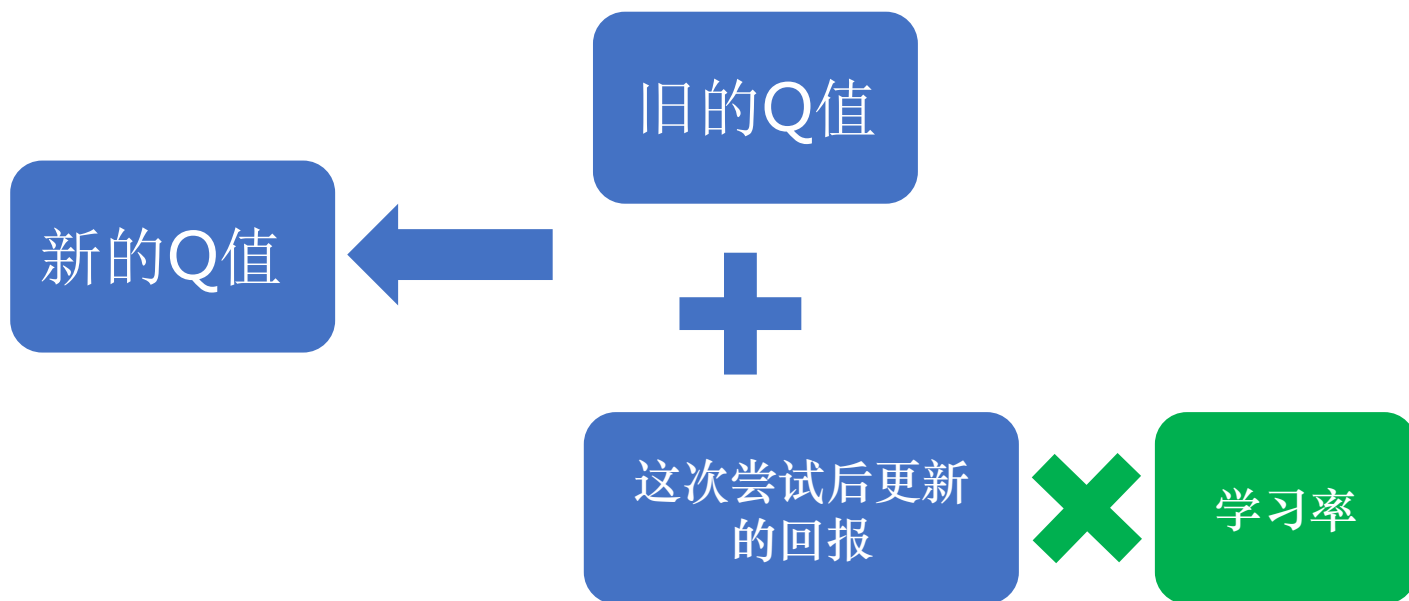
学习过程

- 每完成一局之后，就持续更新Q函数。
- 完成的局数越多，更新的次数就越多，结果也越准确。

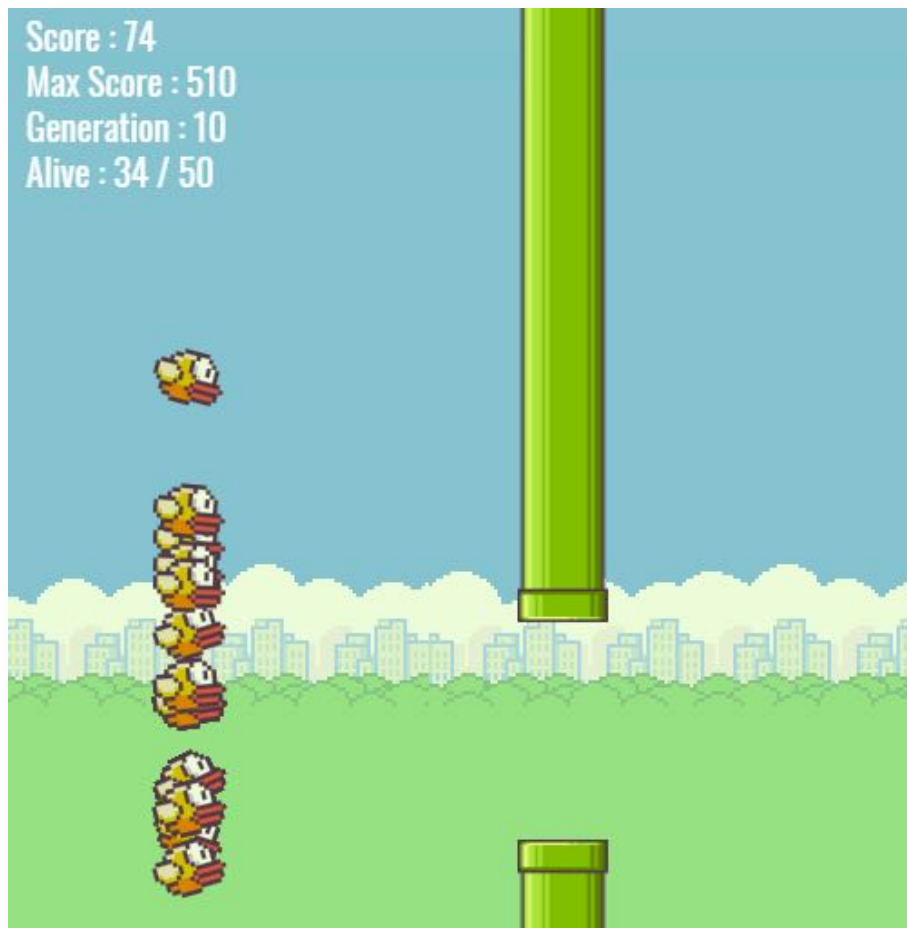


学习率

- 既要利用好已经学好的值，也要善于学习新的值。
- 这两者就通过学习率来平衡，一开始学习率可以大一些，最后稳定时学习率可以小一些。



熟能生巧



- 通过上述公式学习，在足够多的尝试之后
- AI所学到的状态动作值函数 Q 就能够达到一个较优的结果。
- 再根据这个 Q 函数来选择动作，就“熟能生巧”了！

强化学习归纳

- 归纳强化学习的几个要素
 - 归纳强化学习的流程
- 以Flappy Bird强化学习AI为例
 - 具体描述强化学习几个要素是什么？
 - 在Flappy Bird AI学习过程中，Q函数具体含义？
 - 如何通过学习得到优化的Q*函数？
- 井字棋游戏强化学习
 - 如果强化学习井字棋游戏AI，对应的几个要素具体是什么？
 - Q函数的具体含义？
 - 如何通过学习得到优化的Q*函数？