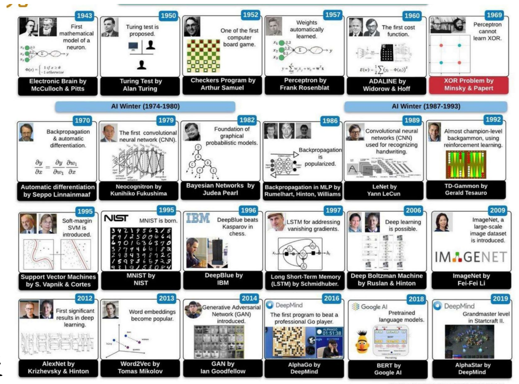


绪论：发展简史

①纪元前：亚里士多德（逻辑与AI）；莱布尼茨；图灵；ENIAC；M-P模型

②原爆点：达特茅斯会议 1956年

主要学派

人工智能学派	主要思想	代表方法
连接主义	利用数学模型来研究人类认知的方法，用神经元的连接机制实现人工智能	神经网络、SVM
符号主义	认知就是通过对有意义的表示符号进行推导计算，并将学习视为逆向演绎，主张用显式的公理和逻辑体系搭建人工智能系统	专家系统，知识图谱，决策树等
行为主义 + 演化主义	以控制论及感知-动作型控制系统原理模拟行为以复现人类智能	强化学习等
	对生物进化进行模拟，使用遗传算法和遗传编程	遗传算法等

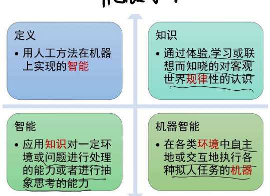
主要研究与典型应用

符号主义专家系统，用于分析质谱仪与细菌感染诊断系统

跳棋程序，机器定理证明，机器翻译

Hopfield网络 连接主义

机器学习



核心能力	处理内容
计算能力	超强存储和快速计算的能力，使海量数据操作成为可能
感知能力	使机器具备感知能力，可以将非结构化的数据结构化，并与用户互动。
认知能力	让机器像人一样，有理解能力、归纳能力、推理能力，有运用知识的能力。

AGENT-ENVIRONMENT范式

- 简单/模型反射式智能体 ⇒ 信息充足，直接求解
- 基于目标/效用智能体 ⇒ 信息较不足，对问题抽象 需定义目标作用 探索 回溯
- 学习智能体 ⇒ 信息不足，进行机器归纳 建立问题 机器学习

问题解决的符号语言

问题：从一个实例描述集合到集合的映射

$T: I \to O$

$i \in I$（$(i,O) \in I \times O$）称 i 为问题T的实例

I有穷则T有穷，$O = T(i)$ 中的T对应 i 的解

若程序P有穷，把T的任意实例 i 输入P，总有输出 $O = T(i)$，则P解决了问题

系统
- ABS（algorithm）：对问题有全面认识 任意有穷问题都有算法 但可能性过多
- SBS（search）：通过探查回溯寻优 利用部分知识解决问题 认识不完整，有片段
- CBS（Case）：对一组（实例，结果）集合归纳，自动推广 机器学习

问题可计算完备 / 算法长度有穷 / 求解时间合理 } 搜索状态爆炸，规则集准 / 寻优不针对问题，针对规则使用

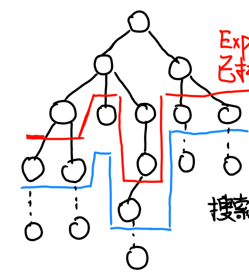
求解思路 （寻路问题）
- 基于算法 数学规划 全面认识 各路径长
- 基于搜索 搜索算法 认识片段 图+边
- 基于实例 机器学习 无知识 地图

搜索 图搜索
- ①扩展Frontier节点
- ②检查目标节点是否在子节点中 有，得到解
- ③无，按策略选某子节点作为Frontier
- ④重复①至无可操作节点

～基本问题
- ①是否一定有解 Completeness
- ②是否为最优解 Optimality
- ③时空复杂性如何 Complexity
- ④是否陷入循环 Goal

～过程：当前状态 $\xrightarrow[\text{选择操作}]{\text{操作算子集}}$ 新状态 $\xrightarrow{\text{满足}}$ 含解路径（不满足）

正向 初始 数据驱动；逆向 目标 目的驱动；双向搜索：正逆同时至某处汇合



Explored closed表：已生成扩展 已找到从起始点到该点的最佳路径的那些点，

Frontier Open表：已扩展而子状态未扩展 还在寻找最优路径的那些点，

Unexplored 我们还没有预见到的点，

搜索过程：Unexplored → Frontier
Frontier → Explored
Goal → Explored，end

☆状态空间的图描述

对于某搜索问题
- 状态空间 S　目标验证 G(s)
- 初始空间 S_0　动作代价 $c(s,a,s')$
- 某状态下的动作集合 A(s)
- 转移模型 $s' = Result(s,a)$

解：能到目标状态的动作序列
最优解：所有解cost最低者

状态空间图（每状态仅一次）
- 节点抽象了环境状态
- 边表示了状态转移
- 目标检验表示为目标节点

表示搜索过程

搜索树（有向图）
- 节点抽象状态
- 边表示求解步骤
- 初始节点为根结点
- 状态到状态的算子序列 ⟺ 图中某路径

☆～策略

盲目搜索：无任何信息，按固定步骤（依次/随机调用操作算子）进行搜索
- DFS：优先扩展最新产生的节点，深度相等节点按生成次序 Open表：栈：需给出一个节点扩展最大深度
  - 扩展最深节点，使搜索沿着状态空间某条单一的路径从起始节点向下进行
  - 仅当搜索到达一个没有后裔的状态时，才考虑另一条替代的路径

  → 深度界限：合适；采取不断加大限制值方式反复进行至找到解

  ☆不能保证第一次搜索到某个状态时其路径是到这个状态的最短路径
- BFS：以接近起始节点程度为依据逐层扩展节点
  - 每次选择深度最浅节点首先扩展，搜索逐层进行
  - 高价搜索，有解存在必能找到

  所有子状态及子状态的后裔状态先于兄弟状态被搜索 多次得到同一状态应保留最短路径

  解决有权图问题 ⊂ 根到n的代价
- UCS 节点扩展选择以g(n)为依据，选最小者扩展 Dijkstra

启发式搜索：考虑特定领域可应用的知识，动态确定操作算子的调用（优先合适的）减少不必要操作，加速 → BFS最佳优先搜索 A^*（不一定最短，一定快于UCS）

设计启发函数h(n)，以估价函数f(n)大小重排OPEN表

- A^*完备
- A^*最优 ⇒ 条件
  - 启发函数可采纳性 $h(n) \le \boxed{h^*(n)}$（实际距离） 对树搜索成立
  - 启发函数一致性 $h(n) \le c(n,a,n') + h(n')$（单调性） 图搜索中$A^*$最优条件

*信息性：A^*中对∀state，$\exists h_1(n) \le h_2(n)$，称$h_2$信息性更多⇒h(n)↑（n→n'实际代价），$A^*$信息↑，$\boxed{工作量}$↓ $\boxed{计算量}$↑ 取舍

利用 启发式信息：用来简化搜索的有关问题领域的特性信息

启发式图搜索特点：重排OPEN表，选最有希望的节点扩展

启发式信息分类
- 陈述性启发信息 精确/精炼描述状态 用于选择扩展节点
- 过程性启发信息 构造操作算子 ～生成～
- 控制性启发信息 控制策略 ～删除～

估价函数：节点希望程度的量度

估价函数：初始节点经n节点达目标节点路径最小代价估计值 $f(n) = g(n) + h(n)$（实际代价 估计代价） 启发函数：与目标"接近程度"的启发式估计

h(n)比重↑加速，解不一定最优；h(n)比重↓工作量↑，解可能最优

☆八数码问题 h(n)选择
- 不在位数 h_1
- 各数码到目的位步数和 h_2
- 逆转数码数 h_3
- h_1+h_3 h_4

g(n)选择：节点n深度

适用情形
- 没有确定解（问题/数据模糊性）
- 状态数随搜索深度指数增长（状态空间大）

MIN-MAX $\boxed{博弈树}$
- 我方每一步使f(n)最大
- 敌方每一步使f(n)最小

→ 常过大
- 减少结点 α-β剪枝
- 对结点采样 蒙特卡洛树搜索。