什么是人工智能？

一阶谓词逻辑有什么特点，产生式有什么特点

搜索求解策略（宽度优先算法，深度优先算法，启发式）

人工神经网络（不会是画个网络给你）：神经元，单个神经元的结构，前馈型人工神经网络什么特色特点，反馈型人工神经网络什么特色特点

基础遗传算法一般基础流程

双种群遗传算法特点

什么是人工智能

一阶谓词逻辑的特点

产生式的特点

搜索求解策略——宽度优先算法

搜索求解策略——深度优先算法

搜索求解策略——启发式

什么是人工神经网络

什么是神经元

单个神经元的结构

前馈型人工神经网络的特色特点

反馈型人工神经网络的特色特点

基础遗传算法的一般基础流程

双种群遗传算法特点

**·什么是人工智能**

用人工的方法在机器（计算机）上实现的智能；或者说是人们使机器具有类似于人的智能。

**·一阶谓词逻辑（表达式）的特点**

***优点：***

1. 自然性：谓词逻辑是一种接近自然语言的形式语言，用它表示的知识比较容易理解。
2. 精确性：谓词逻辑是二值逻辑，其谓词公式的真值只有“真”与“假”，因此可用它表示精确的知识，并可保证演绎推理所得结论的精确性。
3. 严密性：谓词逻辑具有严格的形式定义及推理规则，利用这些推理规则及有关定理证明技术可从已知事实推出新的事实，或证明所作的假设。
4. 容易实现：用谓词逻辑表示的知识可以比较容易地转换为计算机的内部形式，易于模块化，便于对知识进行增加、删除及修改。

***局限性：***

1. 不能表示不确定的知识：谓词逻辑只能表示精确性的知识，不能表示不精确、模糊性的知识，但人类的知识不同程度地具有不确定性，这就使得它表示知识的范围受到了限制。
2. 组合爆炸：在其推理过程中，随着事实数目的增大及盲目地使用推理规则，有可能形成组合爆炸。目前，在这一方面做了大量的研究工作，出现了一些比较有效的方法，如定义一个过程或启发式控制策略来选取合适的规则等。
3. 效率低：用谓词逻辑表示知识时，其推理是根据形式逻辑进行的，把推理与知识的语义割裂开来，这就使得推理过程冗长，降低了系统的效率。

**·产生式的特点**

***主要优点：***

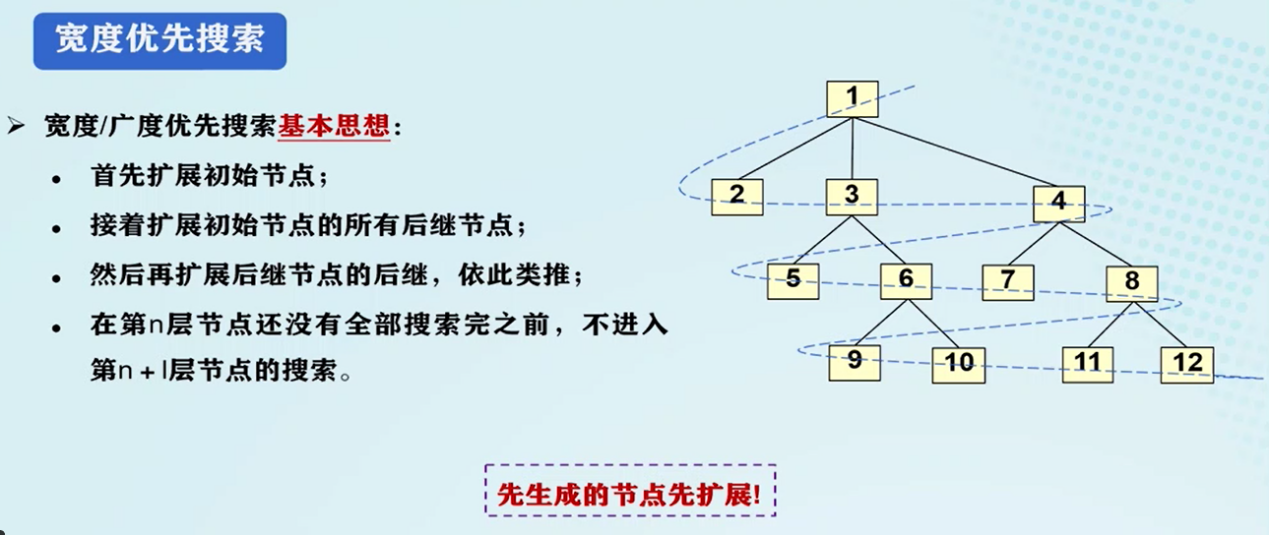
1. 自然性：产生式表示法用“如果……，则……的形式表示知识，这是人们常用的一种表达因果关系的知识表示形式，既直观、自然，又便于进行推理。正是由于这一原因，才使得产生式表示法成为人工智能中最重要且应用最多的一种知识表示方法。
2. 模块性：产生式是规则库中最基本的知识单元，它们同推理机构相对独立，而且每条规则都具有相同的形式。这就便于对其进行模块化处理，为知识的增、删、改带来了方便，为规则库的建立和扩展提供了可管理性。
3. 有效性：产生式表示法既可表示确定性知识，又可表示不确定性知识；既有利于表示启发式知识，又可方便地表示过程性知识。目前已建造成功的专家系统大部分是用产生式来表达其过程性知识的。
4. 清晰性：产生式有固定的格式。每一条产生式规则都由前提与结论(操作)这两部分组成，而且每一部分所含的知识量都比较少。这既便于对规则进行设计,又易于对规则库中知识的一致性及完整性进行检测。

***主要缺点：***

1. 效率不高：在产生式系统求解问题的过程中，首先要用产生式的前提部分与综合数据库中的已知事实进行匹配，从规则库中选出可用的规则，此时选出的规则可能不止一个，这就需要按一定的策略进行“冲突消解”，然后把选中的规则启动执行。因此，产生式系统求解问题的过程是一个反复进行“匹配—冲突消解—执行”的过程。鉴于规则库一般都比较庞大，而匹配又是一件十分费时的工作，因此其工作效率不高，而且大量的产生式规则容易引起组合爆炸。
2. 不能表达具有结构性的知识：产生式适合于表达具有因果关系的过程性知识，但对具有结构关系的知识却无能为力，它不能把具有结构关系的事物间的区别与联系表示出来。后面介绍的框架表示法可以解决这方面的问题。因此，产生式表示法除了可以独立作为一种知识表示模式外，还经常与其他表示法结合起来表示特定领域的知识。例如，在专家系统PROSPECTOR中用产生式与语义网络相结合，在Alkins中把产生式与框架表示法结合起来，等等。

***适合产生式表达的知识：***

1. 由许多相对独立的知识元组成的领域知识，彼此间关系不密切，不存在结构关系，如化学反应方面的知识。
2. 具有经验性及不确定性的知识，而且相关领域中对这些知识没有严格、统一的理论，如医疗诊断、故障诊断等方面的知识。
3. 领域问题的求解过程可被表示为一系列相对独立的操作，而且每个操作可被表示为一条或多条产生式规则。

**·搜索求解策略——宽（广）度优先算法**

宽度优先搜索使用队列（queue）来实现

1. 将初始结点s放入open表中
2. 判断目前的open表是否为空，如果为空则失败退出，否则进行下一步，将open表的第一个节点n移到close表中
3. 扩展节点n，将n的后继节点放入open表的末端，并提供指向n节点的指针
4. 判断n的后继节点是否有目标节点，如果有则成功退出，否则转第二步。一直循环直到停止

宽度优先搜索使用队列（queue）来实现，整个过程也可以看做一个倒立的树形：

1. 把根节点放到队列的末尾。
2. 每次从队列的头部取出一个元素，查看这个元素所有的下一级元素，把它们放到队列的末尾。并把这个元素记为它下一级元素的前驱。
3. 找到所要找的元素时结束程序。
4. 如果遍历整个树还没有找到，结束程序。

**宽度优先搜索的性质：**

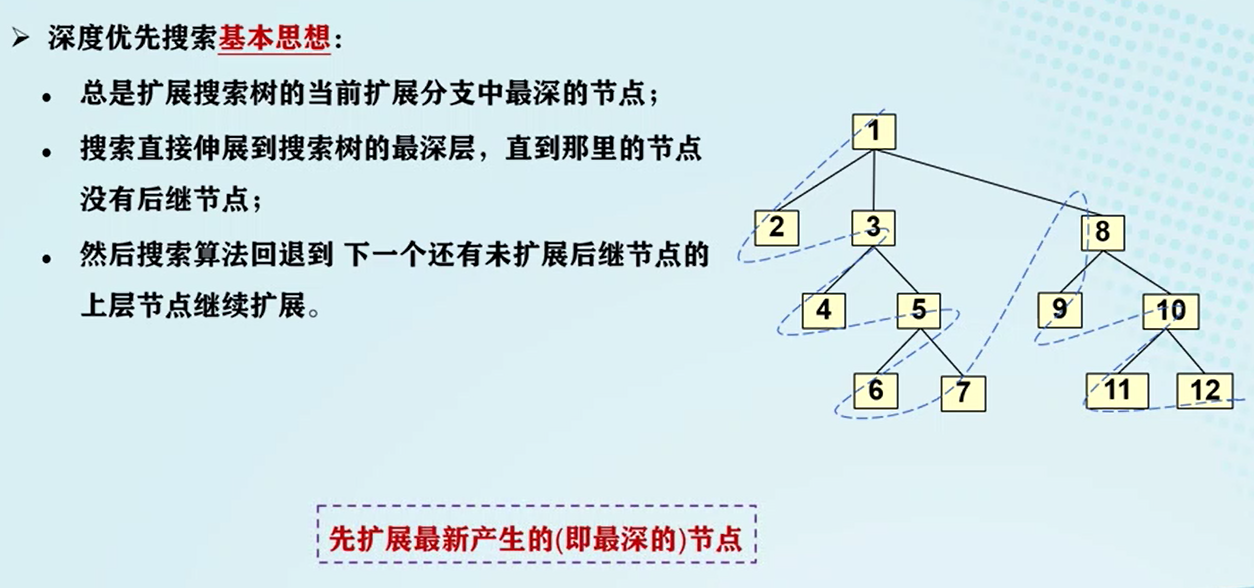
属于图搜索

新扩展的节点排在open表的末端

当问题有解时，一定能找到解

方法与问题无关，具有通用性

效率较低

**·搜索求解策略——深度优先算法**

深度优先搜索用栈（stack）来实现

1. 将初始结点s放入open表中
2. 判断目前的open表是否为空，如果为空则失败退出，否则进行下一步，将open表的第一个节点n移到close表中
3. 判断节点n的深度是否等于最大深度，如果是返回第二步，否则进行下一步
4. 扩展节点n，将n的后继节点放入open表的前端，并提供指向n节点的指针
5. 判断n的后继节点是否有目标节点，如果有则成功退出，否则转第二步。一直循环直到停止

深度优先搜索用栈（stack）来实现，整个过程可以想象成一个倒立的树形：

1. 把根节点压入栈中。
2. 每次从栈中弹出一个元素，搜索所有在它下一级的元素，把这些元素压入栈中。并把这个元素记为它下一级元素的前驱。
3. 找到所要找的元素时结束程序。
4. 如果遍历整个树还没有找到，结束程序。

**深度优先搜索的性质：**

一般不能保证找到最优解

当深度限制不合理时，可能找不到解，可以将算法改为可变深度限制

最坏情况时，搜索空间等同于穷举

**·搜索求解策略——启发式**

启发式搜索是利用与问题有关的启发性信息，并以这些启发性信息指导的搜索的问题求解过程。

* 1. 需定义一个评价函数，对当前的搜索状态进行评估，找出一个最有希望的节点来扩展
  2. 重排open表，选择最有希望的节点加以扩展

种类：A、A\*算法

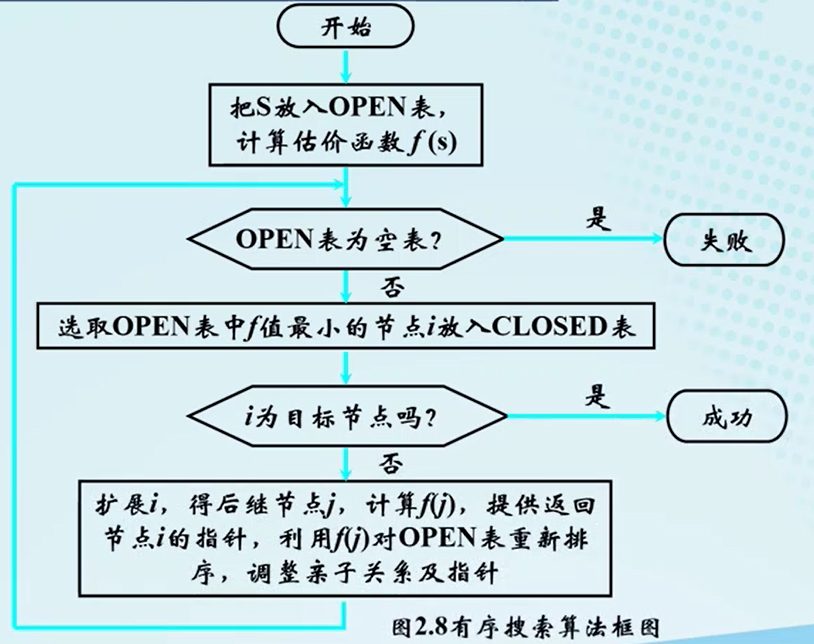
**A算法（局部择优，有序搜索）**

估价函数f(n)：估计节点“希望”程度的量度

f(n) = g(n) + h(n)

g(n)：从起始状态到当前状态n的代价。

h(n)：从当前状态n到目标的估计代价（启发函数）。

1. 局部择优搜索
   1. 把初始结点S0放入open表，计算f(S0)
   2. 如果open表为空，则问题无解，退出
   3. 把open表的第一个节点(记为节点n)取出，放入close表
   4. 考查节点n是否为目标节点，若是，则求得了问题的解，退出
   5. 若节点n不可扩展，则转第二步
   6. 扩展节点n，用估价函数f(x)**计算每个子节点的估价值，并按估价值从小到大的顺序依次放入open表的首部**，并为每一个子节点都配置指向父节点的指针，转第二步。
2. 有序搜索（最好优先搜索，全局择优，选择open表上具有**最小f值的节点**作为下一个要扩展的节点）

**A\*算法（隶属于A算法，其特点在于对估价函数的定义上）**

采用h\*(n)的下界h(n)为启发函数的A算法成为A\*算法

把open表中的节点按估价函数的值从小到大进行排序

f(n) = g(n) + h(n)

g\*(n)：从初始节点S0到任意节点n的一条最佳路径的状态

h\*(n)：从节点n到目标节点的一条最佳路径的代价

g(n)是对g\*(n)的估计，g(n)>= g\*(n)

h(n)是h\*(n)的下界，即对所有的n，h(n)<= h\*(n)

满足以上约束的估计函数定义的A算法就是A\*算法

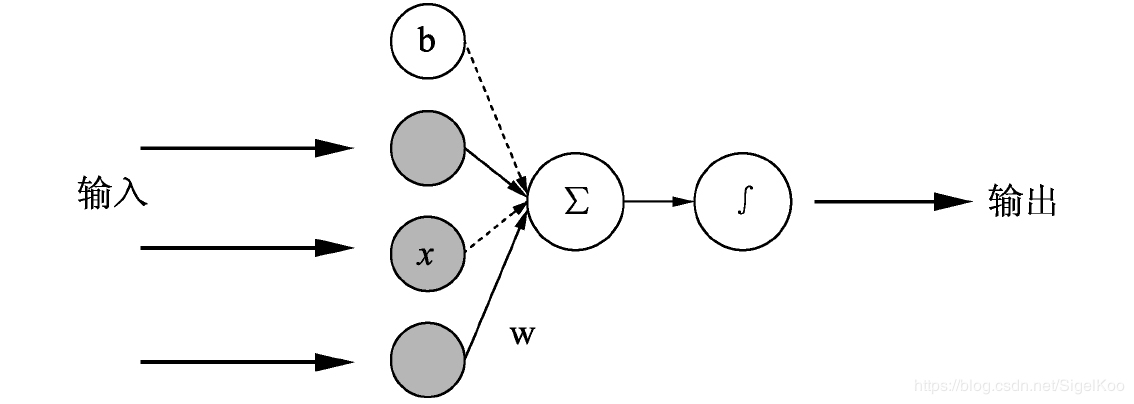
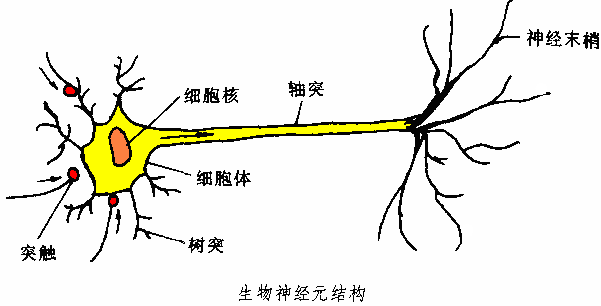
启发式搜索就是在状态空间中的搜索对每一个搜索的位置进行评估，得到最好的位置，再从这个位置进行搜索直到目标。这样可以省略大量无谓的搜索路径，提高了效率。在启发式搜索中，对位置的估价是十分重要的。采用了不同的估价可以有不同的效果。

启发中的估价是用估价函数表示的，如：f(n) = g(n) + h(n)

其中f(n) 是节点n的估价函数，g(n)是在状态空间中从初始节点到n节点的实际代价，h(n)是从n到目标节点最佳路径的估计代价。在这里主要是h(n)体现了搜索的启发信息，因为g(n)是已知的。如果说详细点，g(n)代表了搜索的广度的优先趋势。但是当h(n) >> g(n)时，可以省略g(n)，而提高效率。

**·什么是人工神经网络**

人工神经网络(artificial neural networks, ANN)：模拟人脑神经系统的结构和功能，运用大量简单处理单元经广泛连接而组成的人工网络系统。

******·单个神经元的结构**

**·前馈型人工神经网络的特色特点**

  前馈神经网络（Feedforward Neural Network），简称前馈网络，是人工神经网络的一种。在此种神经网络中，各神经元从输入层开始，接收前一级输入，并输出到下一级，直至输出层。整个网络中无反馈，可用一个有向无环图表示。比如BP神经网络

前馈神经网络采用一种单向多层结构，其拓扑结构如图1所示。其中每一层包含若干个神经元，同一层的神经元之间没有互相连接，层间信息的传送只沿一个方向进行。其中第一层称为输入层。最后一层为输出层。中间为隐含层，简称隐层。隐层可以是一层。也可以是多层。

前馈神经网络结构简单，应用广泛，能够以任意精度逼近任意连续函数及平方可积函数．而且可以精确实现任意有限训练样本集。从系统的观点看，前馈网络是一种静态非线性映射．通过简单非线性处理单元的复合映射，可获得复杂的非线性处理能力。从计算的观点看．缺乏丰富的动力学行为。大部分前馈网络都是学习网络，其分类能力和模式识别能力一般都强于反馈网络。

前馈型神经网络取连续或离散变量，一般不考虑输出与输入在时间上的滞后效应，只表达输出与输入的映射关系；

前馈型神经网络的学习主要采用误差修正法（如BP算法），计算过程一般比较慢，收敛速度也比较慢；

**·反馈型人工神经网络的特色特点**

 反馈神经网络是一种反馈动力学系统。在这种网络中，每个神经元同时将自身的输出信号作为输入信号反馈给其他神经元，它需要工作一段时间才能达到稳定。比如Hopfield神经网络

反馈神经网络输出端会反馈到其输入端，在输入的激励下，其输出会产生不断的状态变化，这个反馈过程会一直反复进行。

反馈型神经网络可以用离散变量也可以用连续取值，考虑输出与输入之间在时间上的延迟，需要用动态方程来描述系统的模型。

而反馈型神经网络主要采用Hebb学习规则，一般情况下计算的收敛速度很快。

反馈网络也有类似于前馈网络的应用，并且在联想记忆和优化计算方面的应用更显特点。

**·什么是遗传算法**

遗传算法（GA）是一类借鉴生物界的进化规律（适者生存，优胜劣汰）演化而来的随机化搜索方法。

**·基础遗传算法的一般基础流程**

1. 使用随机方法或者其它方法，产生一个有N个染色体的初始群体 pop(1)，t:=1；
2. 对群体中的每一个染色体popi(t)，计算其适应值



1. 若满足停止条件，则算法停止；否则，以概率从pop(t)中随机选择一些染色体构成一个新种群
2. 以概率Pc进行交叉产生一些新的染色体，得到一个新的群体



1. 以一个较小的概率Pm使染色体的一个基因发生变异，形成mutpop(t+1)；t:=t+1，成为一个新的群体



返回 （2）。



**·双种群遗传算法特点**

双种群遗传算法：建立两个遗传算法群体，分别独立地运行复制、交叉、变异操作，同时当每一代运行结束以后，选择两个种群中的随机个体及最优个体分别交换。

双种群遗传算法是使用多种群同时进化，并交换种群之间优秀个体所携带的遗传信息，以打破种群内的平衡态达到更高的平衡态，有利于算法跳出局部最优。