



人工智能基础与进阶

计算机与人工智能

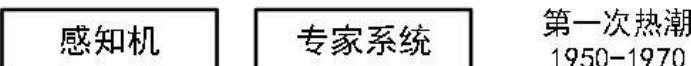
周 越

上海交通大学

计算机与AI 理论基础



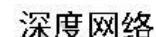
人工智能的 技术发展



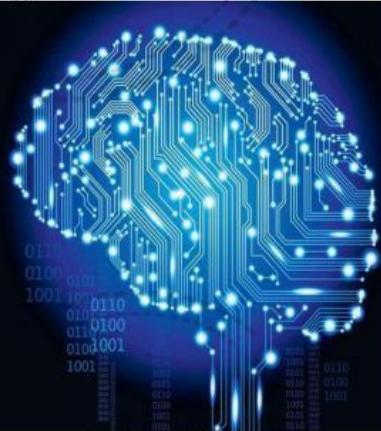
符号主义, 早期推理系统, 早期神经网络, 专家系统



基于统计的方法, 连接主义快速发展



目录 content



第一节 数理逻辑的基础—布尔代数

第二节 “用机器计算”的基础—可计算理论

第三节 人工智能之父——图灵

第四节 计算机

第五节 简单的智能

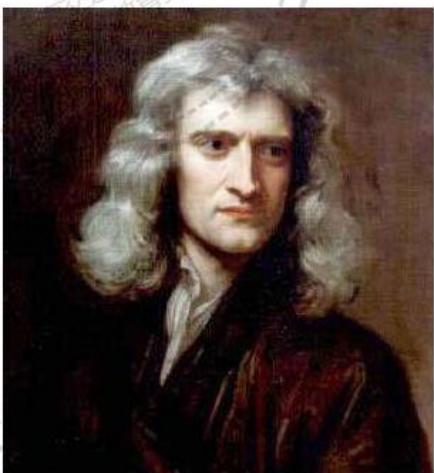
第六节 动物识别的专家系统

第七节 二进制与位运算

第一节

数理逻辑的基础——布尔代数

数理逻辑的基础——布尔代数



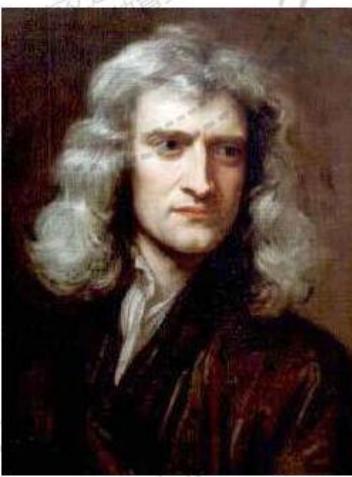
莱布尼茨
(1646—1716年)
德国哲学家、数学家
发明几何微积分

数理逻辑的创始人，十七世纪提出数理逻辑的指导思想

- ✓ 建立一种“普通的符号语言”，语言符号是表意而不是拼音，每个符号表达一个概念；
- ✓ 应该是一个“思维的演算”，演算就是符号作运算，在数量和思维方面都起作用。

数理逻辑又称符号逻辑、理论逻辑。它既是数学的一个分支，也是逻辑学的一个分支。是用数学方法研究逻辑或形式逻辑的学科。其研究对象是对证明和计算这两个直观概念进行符号化以后的形式系统。数理逻辑是数学基础的一个不可缺少的组成部分。

数理逻辑的基础——布尔代数



莱布尼茨
(1646—1716年)
德国哲学家、数学家
发明几何微积分

第一阶段 1679

数字代替概念以其演算
五条演算规则：

- ab是ba（交换律）
- a是aa（重言律）
- a是a（同一原则）
- ab是a或ab是b（化简原则）
- 如a是b且b是c，则a是c

四个定理：

- 如a是b且a是c，则a是bc
- 如a是bc，则a是b且a是c
- 如a是b，则ac是bc
- 如a是b且c是d，则ac是bd

第二阶段 1686

等式符号作系词符号

A=BY

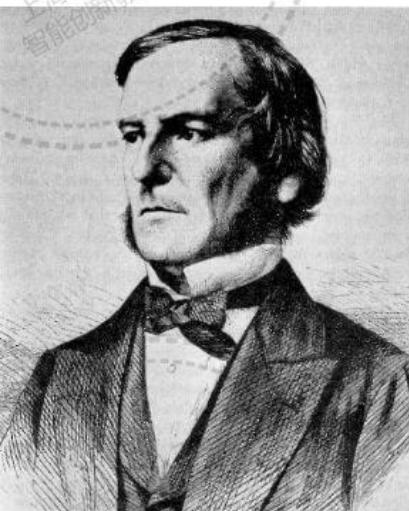
表述全称肯定命题 (Y为一未确定的系数，用以修饰B而使B成为A的一部分)
双重否定之为肯定，即
非非A=A

第三阶段 1690

最有价值的十四个基本命题：

- A=A+A “+” 表示逻辑相乘
- 如A=B且B=C，则A=C
- 如A=B且B≠C，则A ≠ C
- 如A=B且B<C，则A < C
- 如A=B且C<B，则C < A
- 如A=B且C=D，则A+C < B+D
- 如A=B，则A+C = B+C
- 如A<B，则A+C < B+C
- 如A+B+A，则B < A
- 如B < A，则A+B=A
- 如A < B且B < C，则A < C
- 如A < B且B < A，则A=B
- 如A < C且B < C，则A+B < C
- 如A < B且C < D，则A+C < B+D
- 如B包含在A中且C包括除去内容B之外的整个A的内容，则A-B=C

数理逻辑的基础——布尔代数



乔治·布尔
(1815 — 1864年)
英国数学家

让人想起：布尔型变量 布尔运算

1847年，布尔出版了《逻辑的数学分析》，这是他对符号逻辑诸多贡献中的第一次。建立了“布尔代数”，并创造一套符号系统，利用符号来表示逻辑中的各种概念。

1854年出版《思维规律的研究》，这是他最著名的著作。在这本书中布尔介绍了现在以他名字命名的布尔代数。

数理逻辑的基础——布尔代数



乔治·布尔
(1815 — 1864年)
英国数学家

布尔用数学方法研究逻辑问题，成功地建立了逻辑演算

用等式表示判断，把推理看作等式的变换

这种变换的有效性不依赖人们对符号的解释，只依赖于符号的组合规律

这一逻辑理论人们常称它为布尔代数

很多计算机语言中将逻辑运算称为布尔运算，将其结果称为布尔值

数理逻辑的基础——布尔代数



乔治·布尔
(1815 — 1864年)
英国数学家

布尔代数的伟大在于，把数学和逻辑结合在了一起。

乔治·布尔在符号逻辑运算领域的研究成果为历史上计算机的发展和当今的人工智能革命打下了**坚实基础**，而人工智能将带领人类社会生产力再一次实现飞跃性发展。

数理逻辑的基础——布尔代数

A
B



A
B



A
B



A
B



A
B



A
B



逻辑器件的电路符号

A



A



A



A



A
B



A
B



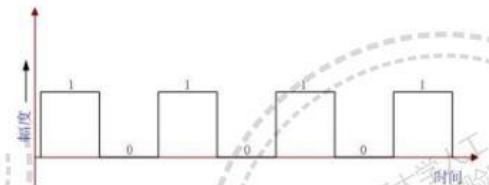
- 1884年，德国数学家**弗雷格**出版了《算术的基础—对数概念的逻辑数学研究》一书，在书中引入量词的符号，使得数理逻辑的符号系统更加完备

- 美国数学家**皮尔斯**，他也在著作中引入了逻辑符号

现代数理逻辑最基本的理论基础逐步形成，成为一门独立的学科

20世纪30年代，逻辑代数在电路系统上获得应用

数理逻辑的基础——布尔代数



20世纪初电子管技术的突破使得电子电路蓬勃发展，产生了无线电通信和早期的无线电广播和电视。

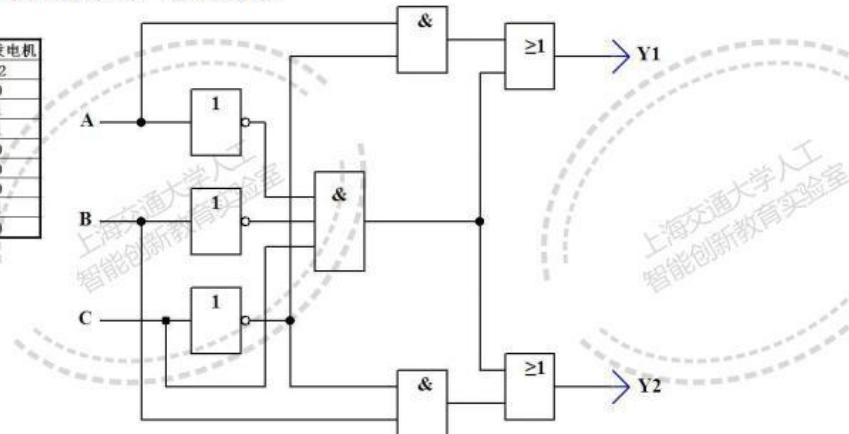
数字逻辑电路随之出现，它是一种传递和处理离散信号的、按照二进制、实现数字信号逻辑运算和处理的电路。

数字信号指自变量是离散的、因变量也是离散的信号，这种信号的自变量用整数表示，因变量用有限数字中的一个数字来表示。在计算机中，数字信号的大小常用有限位的二进制数表示。

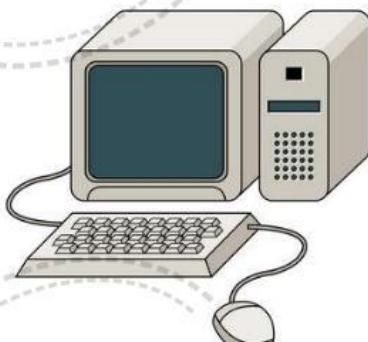
1车间	2车间	3车间	第一发电机	第二发电机
A	B	C	Y1	Y2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	0
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

$$Y_1 = ABC + \bar{A}BC + \bar{ABC} + \bar{ABC} + A\bar{C}$$

$$Y_2 = \bar{ABC} + \bar{ABC} + \bar{ABC} + \bar{ABC} + B\bar{C}$$



“用机器计算”的基础—可计算理论



左图是什么？

通俗回答电脑，正确回答“计算机”，突出computer的原义

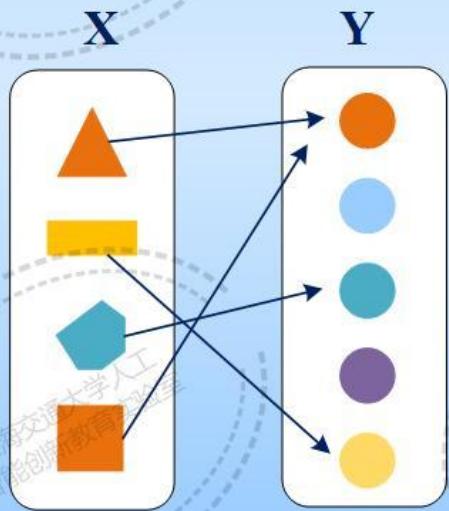
什么是计算机？

计算机的定义是能高速计算的机器，可以进行数值计算、逻辑计算，存储记忆。

计算机是计算**函数**的机器。

函数是什么？

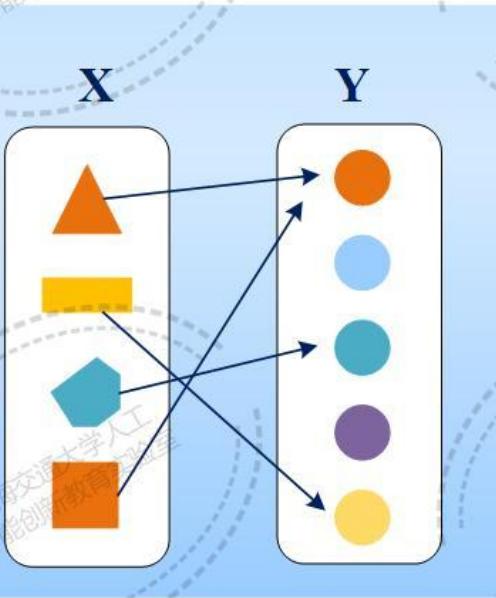
“用机器计算”的基础—可计算理论



函数的定义？

近代定义是给定一个数集A，假设其中的元素为x，对A中的元素x施加对应法则f，记作 $f(x)$ ，得到另一数集B，假设B中的元素为y，则y与x之间的等量关系可以用 $y = f(x)$ 表示，函数概念含有三个要素：定义域A、值域C和对应法则f。其中**核心是对应法则 f**，它是函数关系的本质特征。

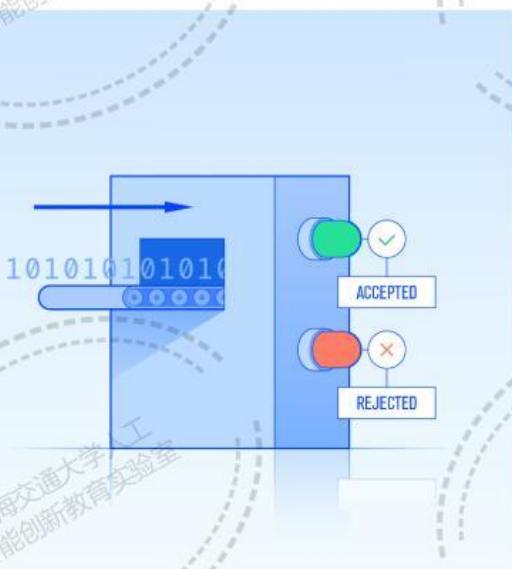
“用机器计算”的基础—可计算理论



数学上的函数定义使我们能够得出一些有趣的结论，例如存在不可计算的函数（即无法解决的问题如停机问题，理发师悖论等，可作扩展内容）：

也就是说，并不是每个函数都可以被描述为一个“算法”。

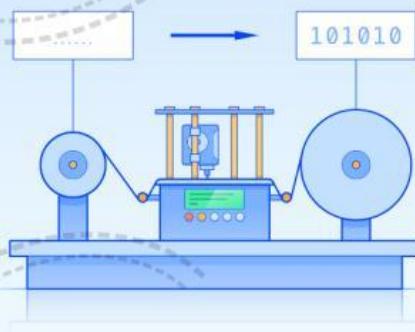
“用机器计算”的基础—可计算理论



1936年，数理逻辑学者提出了计算模型的问题，借以研究**每个问题是否都有解**。关于这一系列问题的理论统称为计算理论（theory of computation），主要包括算法、算法学、计算复杂性理论、可计算性理论、自动机理论和形式语言理论等。

图灵机模型就是**可计算性理论**（computability theory）的产物之一。

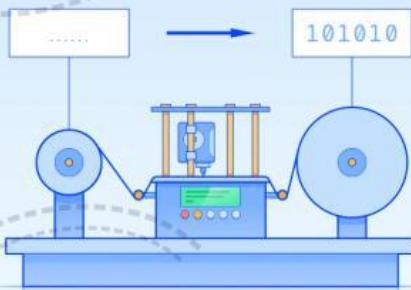
“用机器计算”的基础—可计算理论



可计算性理论的研究方法，是通过建立数学模型，模拟某问题的计算过程，从而精确区分哪些问题是可计算的，哪些是不可计算的。

计算过程即执行算法的过程，可计算性理论的重要课题之一，是将算法这一直观概念精确化，途径之一是通过定义**抽象计算机**，把算法看作抽象计算机的程序，某一问题能在抽象计算机上通过算法求解，则该问题是可计算的，反之则为不可计算。

“用机器计算”的基础—可计算理论



通常把那些存在算法计算其值的函数叫做**可计算函数** (computable function)。其定义为：能够在抽象计算机上编出程序计算其值的函数。可计算函数，是可计算理论的基本研究对象。

这一时期产生的用于计算**可计算函数**的计算模型主要包括: Turing 机、递归函数、 λ 演算、POST 系统、正则算法，其中前三者模型在数学上等价。

人工智能之父—图灵



阿兰·图灵
(1912—1954年)
英国数学家、逻辑学家，
被称为计算机科学之父，
人工智能之父

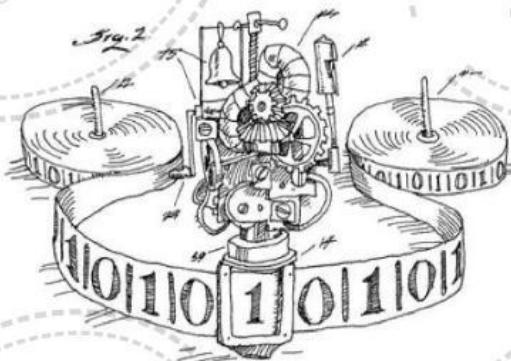
他提出的图灵机模型为现代计算机的逻辑工作方式奠定了基础；

图灵实验是一种用于判定机器是否具有智能的试验方法；

图灵在科学、特别在数理逻辑和计算机科学方面，取得了举世瞩目的成就，一些科学成果构成了现代计算机技术的基础；

图灵的名字与计算机、人工智能密不可分，故冠名计算机领域的最高奖项——图灵奖。

人工智能之父—图灵：图灵机



图灵机

1936年，图灵在他的“论可计算数及其在判定问题中的应用”中提出了一种抽象的计算模型——图灵机(Turing machine)。

他的思路是将人们使用纸笔进行数学运算的过程进行抽象，由一个虚拟的机器替代人类进行数学运算。

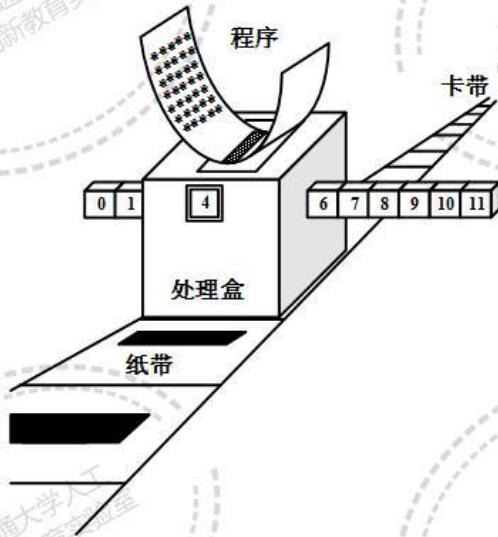
他把运算过程看作下列两种简单的动作：

- 1、在纸上写上或擦除某个符号；
- 2、把注意力从纸的一个位置移动到另一个位置。

而在每个阶段，人要决定下一步的动作，依赖于：

- (1) 此人当前所关注的纸上某个位置的符号；
- (2) 此人当前思维的状态。

人工智能之父—图灵：图灵机



假想该机器有一条无限长的纸带，纸带分成了一个一个的小方格，每个方格有不同的颜色。有一个机器头在纸带上移来移去。

机器头有一组内部状态，还有一些固定的程序。在每个时刻，机器头都要从当前纸带上读入一个方格信息，然后结合自己的内部状态查找程序表，根据程序输出信息到纸带方格上，并转换自己的内部状态，然后进行移动。

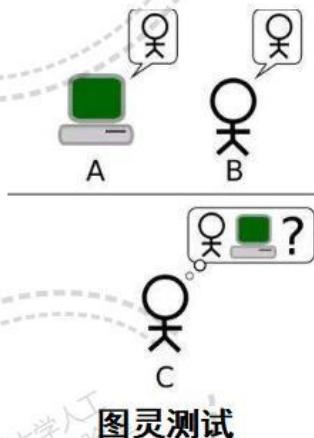
这种自动机器后来被人们称为“图灵机”。

邱奇-图灵论题(The Church-Turing thesis):

“一切可计算过程都可以用图灵机模拟”

人工智能之父—图灵：图灵测试

上海交通大学人工
智能创新教育实验室



图灵测试

1950年，图灵发表了一篇划时代的论文《计算机与智能》，文中预言了创造出具有真正智能的机器的可能性。

由于注意到“智能”这一概念难以确切定义，他提出了著名的图灵测试：如果一台机器能够与人类展开对话（通过电传设备）而不能被辨别出其机器身份，那么称这台机器具有智能。

“到2000年，人类应该可以用10GB的计算机设备，制造出可以在5分钟的问答中骗过30%成年人的人工智能。”

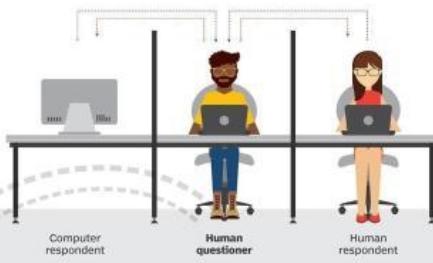
目前的几次公开测试结果显示，我们远远落后此预言。

人工智能之父—图灵：图灵测试

Turing test

During the Turing test, the human questioner asks a series of questions to both respondents. After the specified time, the questioner tries to decide which terminal is operated by the human respondent and which terminal is operated by the computer.

■ QUESTION TO RESPONDENTS ■ ANSWERS TO QUESTIONER



- 观察者通过控制打字机向两个测试对象通话，其中一个是人，另一个是机器。
- 要求观察者不断提出各种问题，从而辨别回答者是人还是机器。理论上一切人类可回答的问题机器都应可以识别并模拟作答，人类无法回答的问题，机器应模拟无法回答。
- 图灵的机器智能思想无疑是人工智能的直接起源之一。

人工智能之父—图灵：图灵测试

近期公开的图灵测试：

- 2014年6月8日，一台名叫尤金·古斯特曼的计算机成功让人类相信它是一个13岁的男孩，成为有史以来首台通过图灵测试的计算机（它并不是超级计算机，也不是电脑，而是一个聊天机器人，是一个电脑程序，或“算法”）
- 2015年11月，《Science》刊载了一项研究，人工智能能像人类一样学习藏文并创造相似字符，结果表明该系统能够迅速学会写陌生的文字，同时还能识别出非本质特征
- 2018年，谷歌的Duplex人工智能语音技术“部分通过图灵测试”，即超过30%的被试者不能分辨对话者是机器或是人。



以上进展显示在某些特定领域，算法可以让机器模拟人类行为并以假乱真

人工智能之父—图灵：人工智能学科的诞生

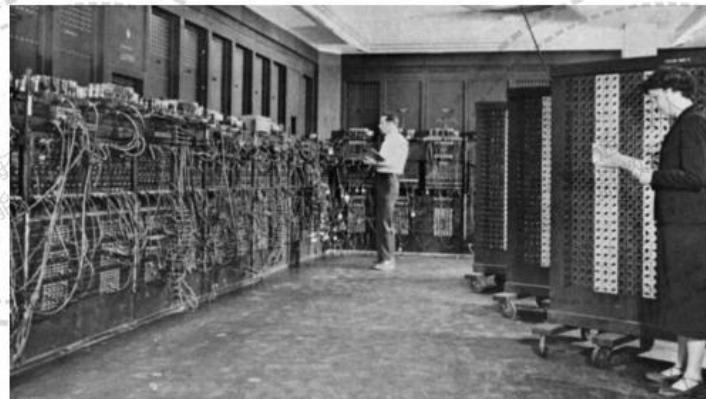
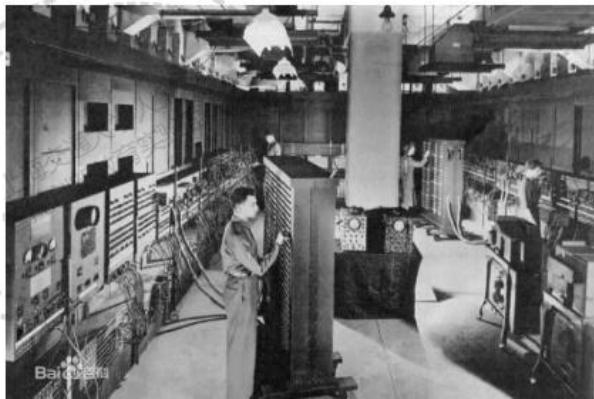
- 1956年8月，在美国汉诺斯小镇的达特茅斯学院中，约翰·麦卡锡（John McCarthy）、马文·闵斯基（Marvin Minsky）、克劳德·香农（Claude Shannon）、艾伦·纽厄尔（Allen Newell）、赫伯特·西蒙（Herbert Simon）等科学家正聚在一起，讨论着一个完全不食人间烟火的主题：用机器来模仿人类学习以及其他方面的智能。
- 会议足足开了两个月的时间，虽然大家没有达成普遍的共识，但是却为会议讨论的内容起了一个名字：**人工智能**。

计算机的基本框架和计算智能的理念已经基本形成
1956年达特茅斯会议水到渠成地达成普遍共识，
使得这一年成为了人工智能元年。

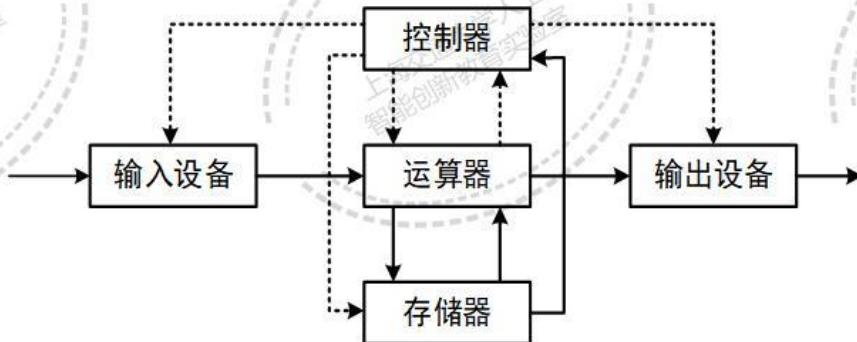


计算机—电子计算机的诞生

- 世界上第一台电子计算机是阿塔纳索夫-贝瑞计算机（Atanasoff-Berry Computer），1937年开始设计，不可编程，仅仅设计用于求解线性方程组，并在1942年成功进行了测试。
- 冯·诺依曼主持设计的ENIAC是世界第二台电子计算机，第一台通用电子计算机，它是图灵完备的电子计算机，能够重新编程，解决各种计算问题。
- 冯·诺依曼同时设计的EDVAC首次使用二进制而不是十进制，EDVAC是第一台现代意义的通用计算机，其体系结构一直延续至今



计算机—计算机结构



现代通用计算机模型

所有程序都基于机器语言运行；机器语言是一串二进制数字（0和1）构成的语言，是计算机的元件能直接阅读的语言。

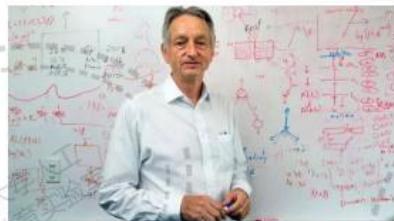
一般程序是由高级语言编写，然后在编译的过程中，被一个编译器/解释器转译为机器语言，从而得以执行；也可以用汇编语言进行编程，汇编语言在机器语言上进行了改进，以单词代替了0和1。

汇编语言实际上是一个记号。

人工智能技术发展



马文·明斯基(Marvin Minsky)



杰弗里·辛顿 (Geoffrey Hinton)

早期科学家就如何应对人工智能展开了辩论。一些人，比如有影响力的学者马文·明斯基(Marvin Minsky)，喜欢采用自上而下的方法：用控制人类行为的规则为计算机编程。

其他人则更喜欢自下而上的方法：比如模拟大脑，学习新行为。随着时间的推移，明斯基的观点占据了主导地位，他和麦卡锡一起从美国政府那里获得了大量资金。

现在所处的这个时期，我们主要采取的方法是基于统计的方法，不需要把人类的思维过程模拟出一套规则来教给计算机，我们可以在一个大的数量集里面来训练计算机，让它自己找到规律从而完成人工智能遇到的问题。

联结主义：感知机，神经网络（如BP神经网络，SOM自组织映射网络）

符号学派：专家系统

行为主义学派：机器人

人工智能技术发展—感知机

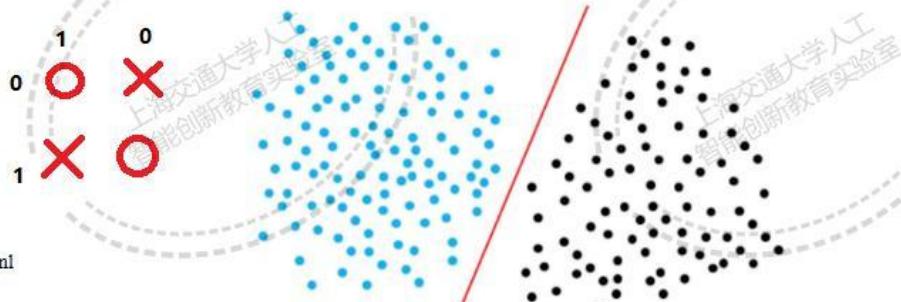
- 假设在一个平台上有很多的男孩女孩，感知机的模型就是尝试找到一条直线，能够把所有的男孩和女孩隔离开。放到三维空间或者更高维的空间，感知机的模型就是尝试找到一个超平面，能够把所有的二元类别隔离开。
- 右图是理想情况，一条直线，完美将平面上的蓝点和黑点分成两类，但大多数情况下，无法做到100%分类正确。那就意味着类别线性不可分，也就意味着感知机模型不适合你的数据的分类。
- 什么叫非线性问题？

异或逻辑，英文为exclusive OR，缩写成xor。其运算法则为：

$$a \oplus b = (\neg a \wedge b) \vee (a \wedge \neg b)$$

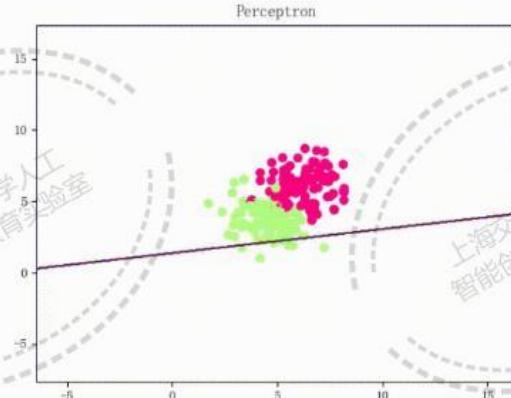
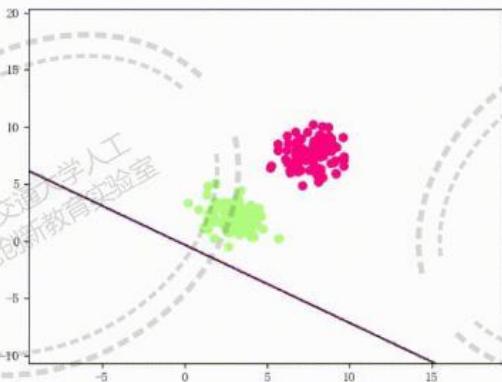
如果a、b两个值不相同，则异或结果为1。如果a、b两个值相同，异或结果为0。

A	B	P
0	0	0
0	1	1
1	0	1
1	1	0



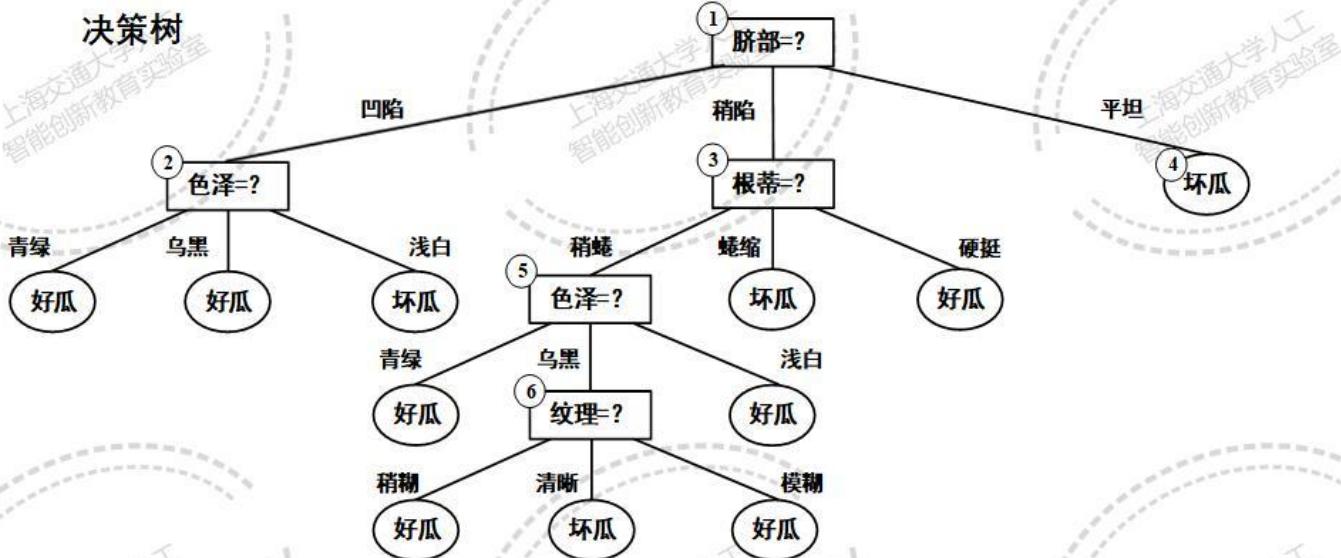
人工智能技术发展—感知机

- 左图是可线性分类问题的求解过程
- 右图是不可线性分类问题的求解过程
- 感知机无法解决异或XOR问题的缺陷导致了第一次人工神经网络研究的衰退。
- 感知机模型在大多数时候泛化能力不强。泛化能力通俗来讲就是指学习到的模型对未知数据的预测能力。



早期人工智能—树与森林

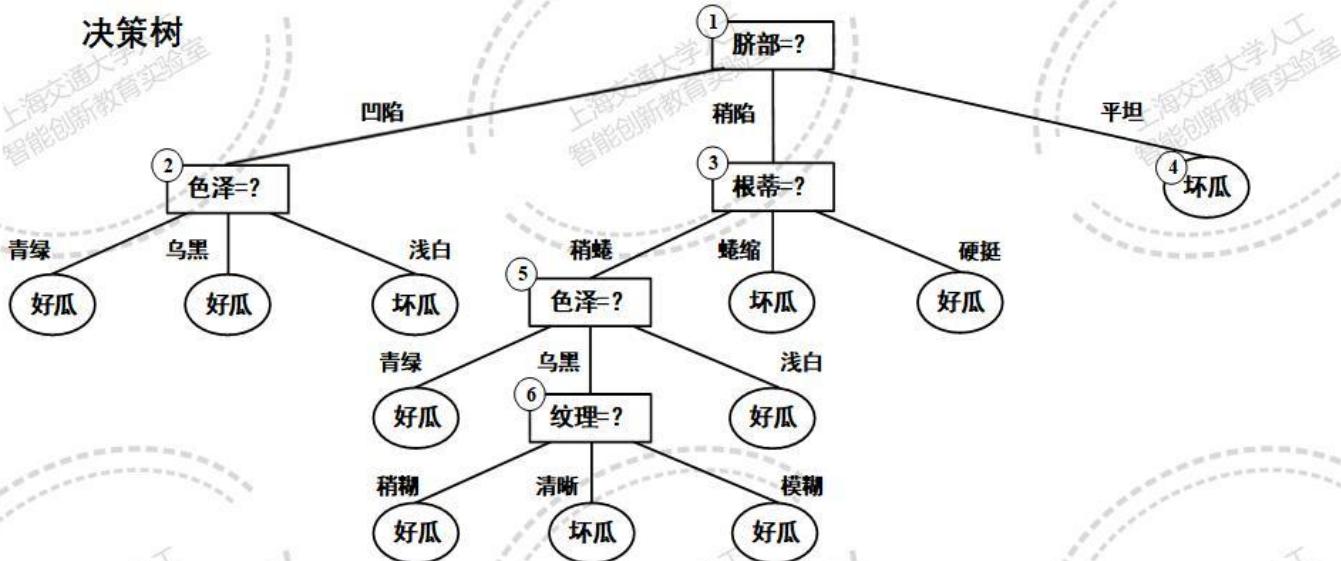
决策树



决策树(Decision Tree) 机器学习中是一个预测模型，仅有单一输出；
决策树学习采用的是自顶向下的递归方法，其基本思想是以信息熵为度量；
构造一棵熵值下降最快的树，执行过程主要由归纳和修剪组成；
决策树构成了其他算法的基础，比如boosting和随机森林

早期人工智能—树与森林

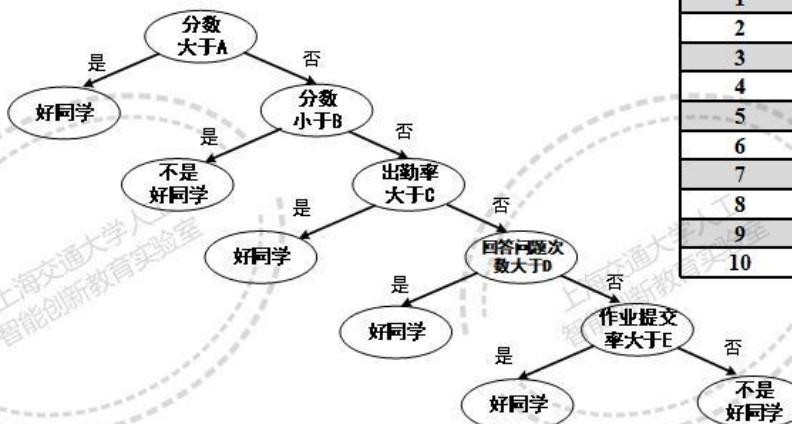
决策树



决策树是一种十分常用的分类方法，需要监管学习：给出一堆样本，依据样本的属性和分类结果，通过学习这些样本得到一个决策树，该决策树能够对新的数据给出的分类。决策树的生成算法有ID3, C4.5和C5.0等。

人工智能技术发展—决策树-例

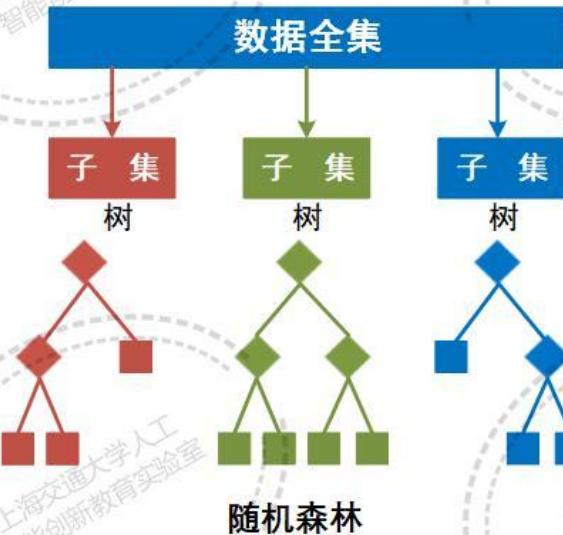
- 给出一组数据，一共有十个样本（学生数量），每个样本有分数，出勤率，回答问题次数，作业提交率四个属性，最后判断这些学生是否是好学生。最后一列给出了人工分类结果。
- 设定决策树为二叉树，学习上表的数据，就是确定下A, B, C, D, E的具体值。



学生编号	分数	出勤率	回答问题次数	作业提交率	是否好学生
1	99	80%	5	90%	是
2	89	100%	6	100%	是
3	69	100%	7	100%	否
4	50	60%	8	70%	否
5	95	70%	9	80%	否
6	98	60%	10	80%	是
7	92	65%	11	100%	是
8	91	80%	12	85%	是
9	85	80%	13	95%	是
10	85	91%	14	98%	是

人工智能技术发展——随机森林

上海交通大学人工
智能创新教育实验室



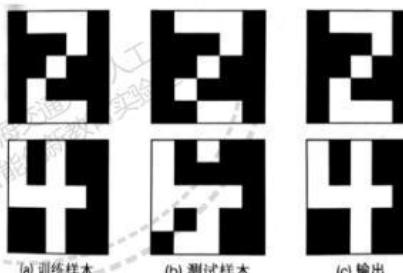
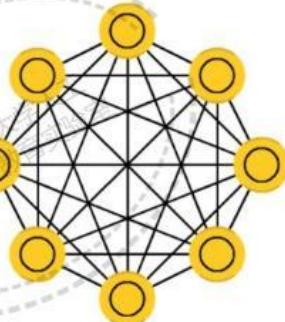
随机森林就是通过集成学习的思想将多棵树集成的一种算法，它的基本单元是决策树

每棵决策树都是一个分类器， N 棵树会有 N 个分类结果，随机森林集成了所有的分类投票结果，将投票次数最多的类别指定为最终的输出

随机森林是集成学习的一个子类

人工智能技术发展—Hopfield网络

- Hopfield神经网络是一种单层互相全连接的反馈型神经网络
- 每个神经元既是输入也是输出，网络中的每一个神经元都将自己的输出通过连接权传送给所有其它神经元，同时又都接收所有其它神经元传递过来的信息。
- 当需要记忆的模式之间较为相似，或者需要记忆的模式太多时，Hopfield神经网络就不能正确地辨别模式。这种相互干扰、不能准确记忆的情况成为串扰(crosstalk)。

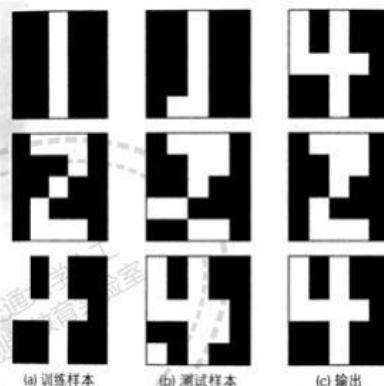


Hopfield网络的应用

记住训练样本的输入
↓
自联想记忆
↓
对测试样本去噪

人工智能技术发展—玻尔兹曼机

- 如果发生串扰或陷入局部最优解，Hopfield神经网络就不能正确地辨别模式，如图。
- 玻尔兹曼机（Boltzmann Machine）则可以通过让每个单元按照一定的概率分布发生状态变化，来避免陷入局部最优解。



<https://www.cnblogs.com/shona/p/10640110.html>

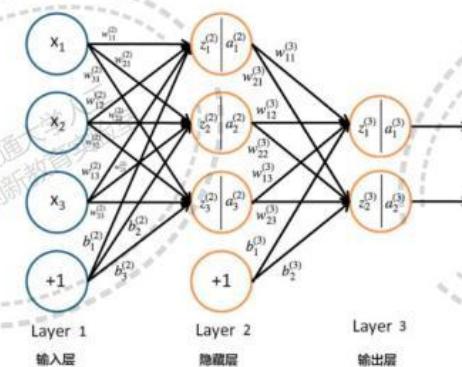
$$\begin{cases} p(x_i = 1 | u_i) = \frac{\exp\left(\frac{x}{KT}\right)}{1 + \exp\left(\frac{x}{KT}\right)} \\ p(x_i = 0 | u_i) = \frac{1}{1 + \exp\left(\frac{x}{KT}\right)} \end{cases}$$

- $T (>0)$ 表示温度系数，当 T 趋近于无穷时，无论 u_i 取值如何， x_i 等于 1 或 0 的概率都是 $1/2$ ，这种状态称为稳定状态。
- 温度系数越大，跳出局部最优解的概率越高；但是温度系数增大时，获得能量函数极小值的概率就会降低。
- 反之，温度系数减小时，虽然获得能量函数极小值的概率增加了，但是玻尔兹曼机需要经历较长时间才能达到稳定状态。

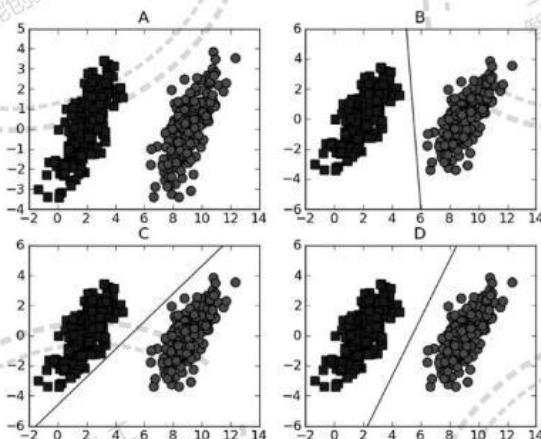
- 玻尔兹曼机是 Hopfield 网络的改良，目的是为了避免陷入局部最优，但是却增加了庞大的计算量，需要进一步优化采样。

人工智能技术发展—神经网络

- BP神经网络是一种多层的前馈神经网络，其主要的特点是：信号是前向传播的，而误差是反向传播的。
- 过程主要分为两个阶段，第一阶段是信号的前向传播，从输入层经过隐含层，最后到达输出层；第二阶段是误差的反向传播，从输出层到隐含层，最后到输入层，依次调节隐含层到输出层的权重和偏置，输入层到隐含层的权重和偏置。
- BP神经网络的优点是具有很强的非线性映射能力和柔性的网络结构。网络的中间层数、各层的神经元个数可根据具体情况任意设定，并且随着结构的差异其性能也有所不同。但是BP神经网络也存在学习速度慢，容易陷入局部极小值，网络层数、神经元个数的选择依赖个人经验，网络推广能力有限等问题。



人工智能技术发展—支持向量机



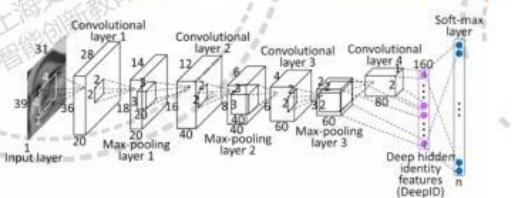
- 支持向量机是在所有知名的数据挖掘算法中最健壮，最准确的方法之一，它属于二分类算法，可以支持线性和非线性的分类。

假设在一个二维线性可分的数据集中，图一A所示，我们要找到一个超平面把两组数据分开，这时，我们认为线性回归的直线或逻辑回归的直线也能够做这个分类，这条直线可以是图一B中的直线，也可以是图一C中的直线，或者图一D中的直线，但哪条直线才最好呢，也就是说哪条直线能够达到最好的泛化能力呢？那就是一个能使两类之间的空间大小最大的一个超平面。

这个超平面在二维平面上看到的就是一条直线，在三维空间中就是一个平面...，离这个超平面最近的点就叫做支持向量，点到超平面的距离叫间隔。支持向量机就是要使超平面和支持向量之间的间隔尽可能的大，这样超平面才可以将两类样本准确的分开，而保证间隔尽可能的大就是保证我们的分类器误差尽可能的小，尽可能的具有泛化能力。

人工智能技术发展—深度学习

- 深度学习的基本思路是模仿大脑神经元之间的传递来处理信息。最显著的应用是计算机视觉和自然语言处理(NLP)领域。
- 1960年后，线性分类器的局限性开始被认识到，它只能将输入空间切分为非常简单的区域，即由一个超平面分离的两个半区间。对于像图像和语音识别这类问题，需要输入-输出函数对输入的非相关变化（位置的变化，方向变化，光照变化，语音的高音和低音变化）不敏感，而对类别敏感（如白狼和萨摩耶犬）。
- 为了让分类器更强大，可以使用广义非线性特征以及核函数方法。但广义特征（如高斯核函数）泛化能力差，常规的方法是手动设计的特征提取器，而这需要大量工程经验和领域专家才能完成。如果**好的特征**可以使用通过学习的方法自动学习得到，上述问题就可以避免，这是深度学习的核心优势。
- 性能表现方面，深度学习探索了神经网络的概率空间，与其他工具相比，深度学习算法更适合无监督和半监督学习，更适合强特征提取，也更适合于图像识别领域、文本识别领域、语音识别领域等。

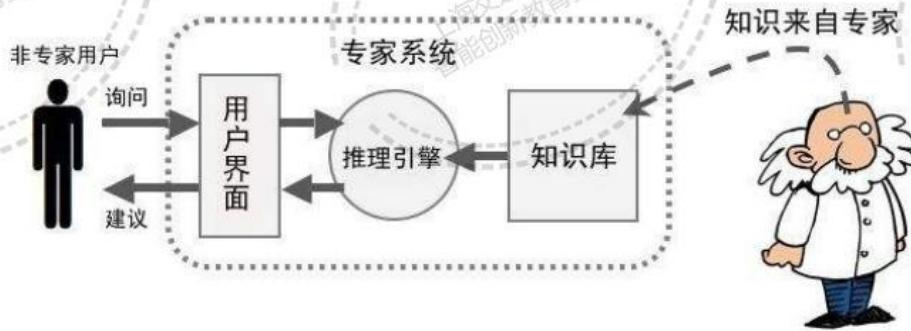


人工智能近期的发展

- 2011年，Watson参加智力问答节目。IBM开发的人工智能程序“沃森”（Watson）参加了一档智力问答节目并战胜了两位人类冠军。沃森存储了2亿页数据，能够将与问题相关的关键词从看似相关的答案中抽取出来。这一人工智能程序已被IBM广泛应用于医疗诊断领域。
- 2016~2017年，AlphaGo战胜围棋冠军。AlphaGo是由Google DeepMind开发的人工智能围棋程序，具有自我学习能力。它能够搜集大量围棋对弈数据和名人棋谱，学习并模仿人类下棋。DeepMind已进军医疗保健等领域。
- 2017年，深度学习大热。AlphaGoZero（第四代AlphaGo）在无任何数据输入的情况下，开始自学围棋3天后便以100:0横扫了第二版本的“旧狗”，学习40天后又战胜了在人类高手看来不可企及的第三个版本“大师”。
- 2019年1月25日凌晨，DeepMind与暴雪在网上直播星际争霸赛事，名为“AlphaStar”（阿尔法星际）的人工智能在与两位人类职业选手“TLO”和“MANA”的比赛中，均以5比0取胜。



早期人工智能—专家系统



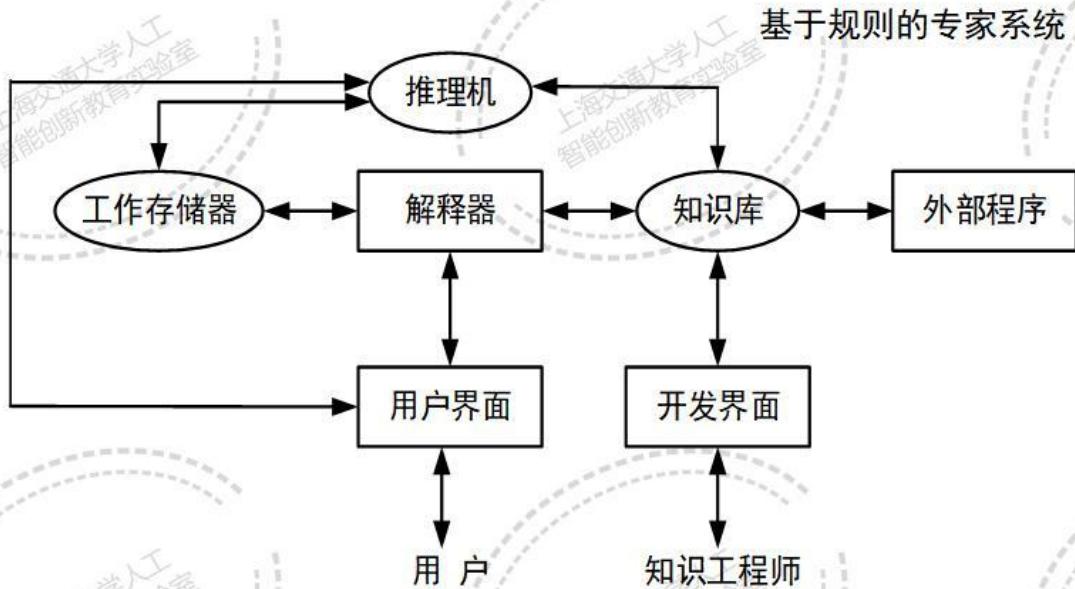
专家系统由模拟人类专家解决问题而得名

专家系统是一个具有大量专门知识与经验的程序系统

根据某领域一个或多个专家提供的知识和经验，进行推理和判断

模拟人类专家的决策过程解决复杂问题

早期人工智能—专家系统



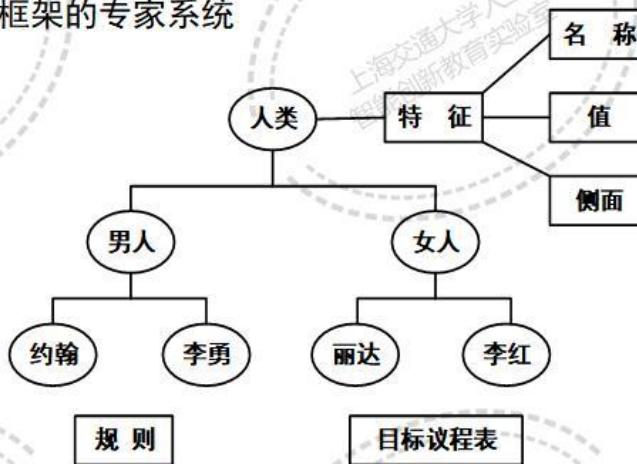
特点：IF...THEN...模型

即满足某项条件，就做什么事情

使用一套包含在知识库内的规则对工作存储器内的具体问题信息进行处理，通过推理机推断出新的信息

早期人工智能—专家系统

基于框架的专家系统



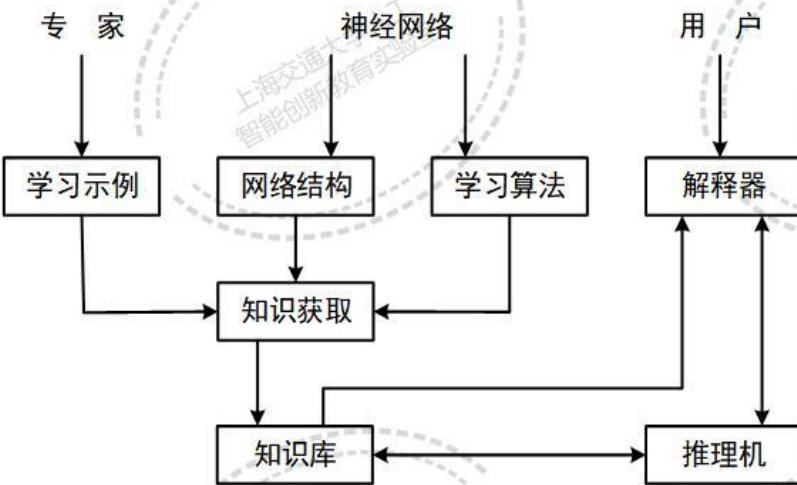
特点：面向对象模型，根据现实时间建模

设计基于框架的专家系统时，把整个问题和每件事想象为编织起来的事物

在辨识事物之后，寻找把这些事物组织起来的方法

早期人工智能—专家系统

基于模型的专家系统



特点：利用神经网络、遗传算法等智能计算方法，改进计算模型
神经网络专家系统的基本结构：

- 自动获取模块输入
- 组织并存储专家提供的学习实例
- 选定神经网络的结构
- 调用神经网络的学习算法，为知识库实现知识获取

早期人工智能—专家系统

- 新型专家系统
- 可以由多个专家系统组成，系统内部采用不同的推理判断方法
- 分布式专家系统：把一个专家系统的功能经分解以后分布到多个处理器上去并行地工作，提高系统处理效率
- 协同式专家系统：亦可称“群专家系统”，能综合若干个相近领域的或一个领域的多个方面的子专家系统互相协作共同解决一个更广领域问题的专家系统



动物识别的专家系统—制定规则库

```
( rules
  ((rule1
    (if (animal has hair))
    (then (animal is mammal)))
   ((rule2
     (if (animal gives milk))
     (then (animal is mammal)))
   .
   .
   .
   ((rule15
     (if (animal is bird))
     (animal flies well))
    (then (animal is albatross))))
```

若动物有毛发(F1)
则动物是哺乳动物(M1)

若动物有奶(F2)
则动物是哺乳动物(M1)

若动物是鸟(M4)
且善飞(F20)
则动物是信天翁(H6)

- 规则库由15条规则组成，规则名为rule1到rule15
- F标记的是初始事实或证据
- M标记的是中间结论
- H标记的是最终结论
- 到问题求解或没有可用规则为止

动物识别的专家系统—正向推理过程程序实现

1) 在rules中查找规则前件的全部条件在当前facts=(F13 F12 F3 F1)中的可用规则，首先找到规则R1，则把R1后件中不在facts中的结论M1添加到facts中，扩充facts为facts=(F13 F12 F3 F1 M1)

实际上，对facts=(F13 F12 F3 F1)还有一条可用规则R5，因为R5的前件F3也在当前facts中。但是，由前面提到的冲突消解策略——先选优先的规则，即若有多条可用规则，则按可用规则在规则库表rules中的顺序选择第一条可用规则

2) 对当前facts在rules中查找可用规则，仍然找到规则R1，但R1的后件结论M1已在facts中，因此不会执行规则R1。继续查找可用规则，找到规则R5，因为R5的后件结论M2不在当前的facts中，故执行R5，把R5不在facts中的结论M2添加到facts中，扩充facts为facts=(F13 F12 F3 F1 M1 M2)。

3) 对当前facts在rules中继续查找可用规则，规则R9的前件在facts中，因此R9是可用规则。而R9的后件结论H1不在当前的facts中，执行R9，把R9的结论H1扩充到facts中，使得facts=(F13 F12 F3 F1 M1 M2 H1)

4) 当前facts，在rules中找不到规则的前件所包含的全部条件在facts中且后件有不在facts中的结论的任何规则，至此，正向推理结束。

二进制位运算

- 用高低电平来表示不同状态的二状态系统，称为二进制系统
- 计算机采用的就是二进制系统
- 二进制系统中只有数字1和0
- 数字系统中，1和0两种状态的组合称为码，不同的组合用来表示计算机中的所有信息

二进制位运算一不同进制

上海交通大学人工
智能创新教育实验室

上海交通大学人工
智能创新教育实验室

上海交通大学人工
智能创新教育实验室

十进制：由数字0到9表示，逢十进一

60进制：时间上每60秒进位成1分钟

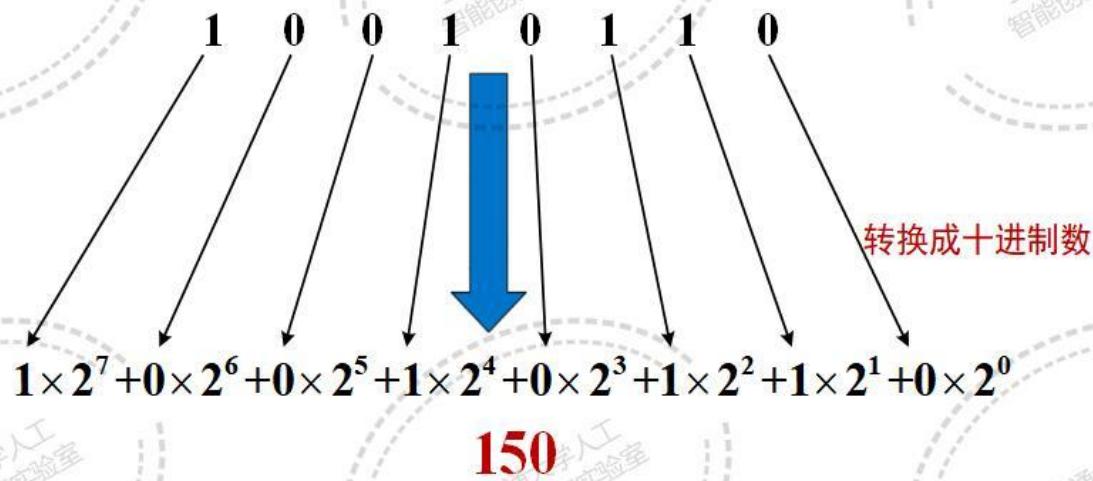
二进制：只有0和1，逢二进一

十六进制：数字0到9，字母A到F（或a到f）表示，其中A到F分别表示10到15，逢十六进一

N进制数：逢N进一



二进制位运算一不同进制数相互转换



二进制转换成十进制采用的是“按权相加法”

二进制数按权展开，然后十进制加法求和

二进制位运算一不同进制数相互转换

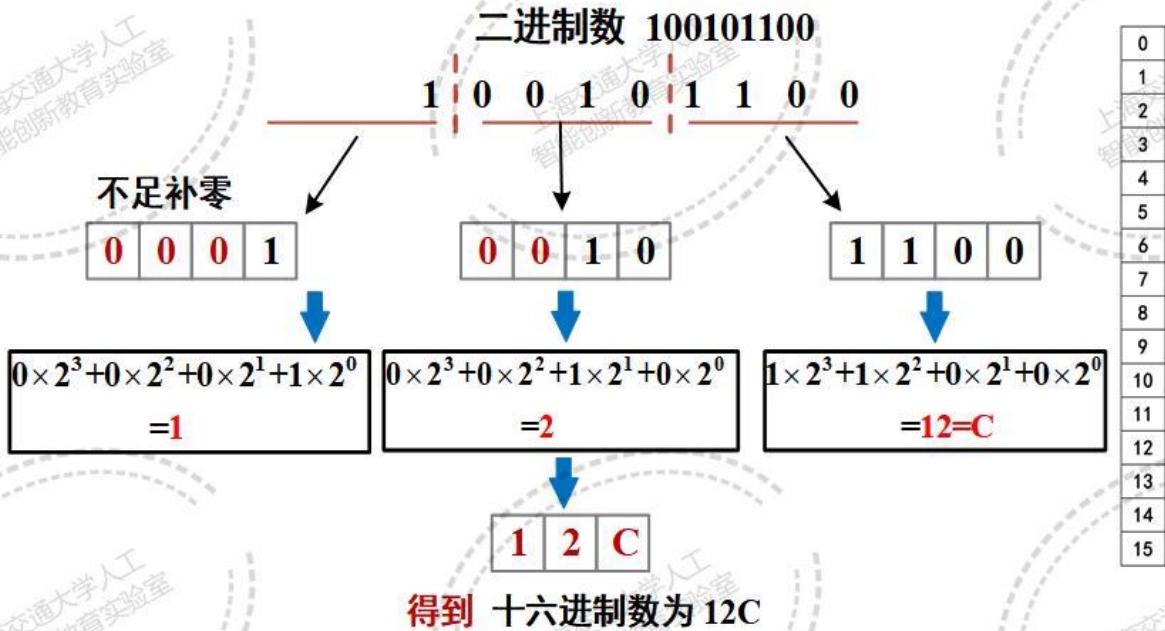
150		余数	从最后一个余数读到第一个余数
2	150	0	
2	75	1	150/2商为75, 余0
2	37	1	75/2商为37, 余1
2	18	0	37/2商为18, 余1
2	9	1	18/2商为9, 余0
2	4	0	9/2商为4, 余1
2	2	0	4/2商为2, 余0
2	1	1	2/2商为1, 余0
	0		1/2商为0, 余1

150的二进制数就是10010110

十进制整数转换成二进制整数采用“除而取余法”

将十进制数除以2，余数是二进制权位上的数，所得商继续除以2，运算直到商为0为止

二进制位运算—不同进制数相互转换



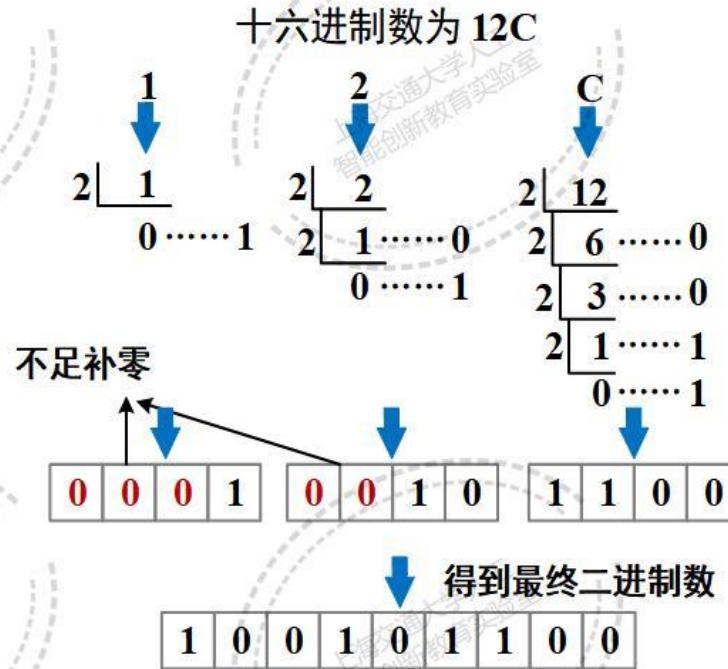
二进制数转换成十六进制数的方法是“取四合一”

每4位二进制数转换成1位十六进制数

从右到左开始转换

不足4位时用0补位

二进制位运算一不同进制数相互转换

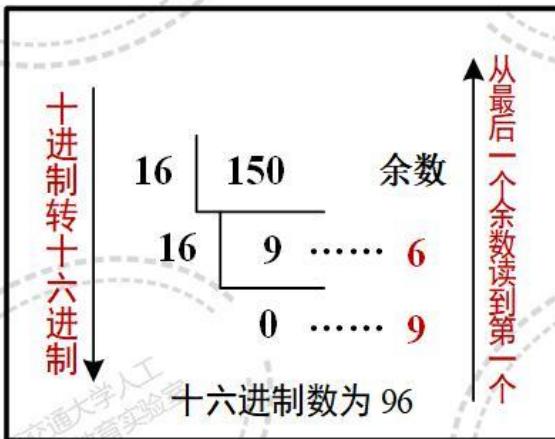


十六进制数通过除以2取余数法转换成二进制数

每位16进制数转换成4位二进制数

不足时左边补0

二进制位运算一不同进制数相互转换



间接法：先将十进制数转换成二进制数，再将二进制数转换成十六进制数

直接法：十进制数按照除以16取余数法，知道商为0为止

二进制位运算一不同进制数相互转换

十六进制转十进制

十六进制数为 96

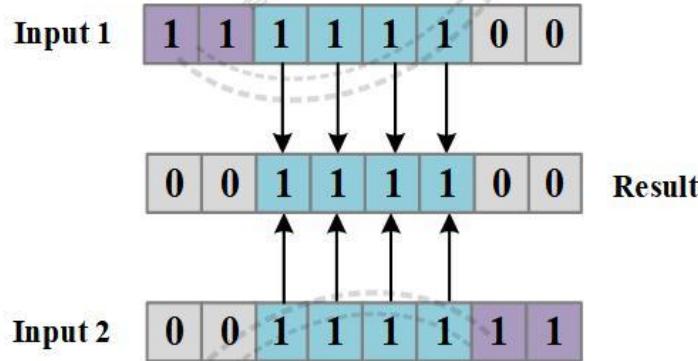
$$9 \times 16^1 + 6 \times 16^0 = 150$$

十六进制转换成十进制，
直接按权展开再求和即可

二进制位运算一补码

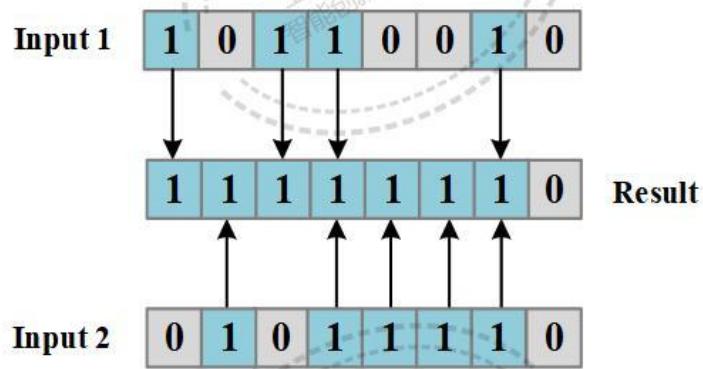
- 计算机中对于有符号数，用最高位作为符号位，“0”代表“+”，“1”代表“-”
- 其余数位用作数值位，代表数值
- 计算机中的符号数有三种表示方法，即原码、反码和补码
- 正整数的补码是其二进制表示，与原码相同
- 求负整数的补码，将其原码除符号位外的所有位取反（0变1，1变0，符号位为1不变）后加1

二进制位运算—计算机位运算



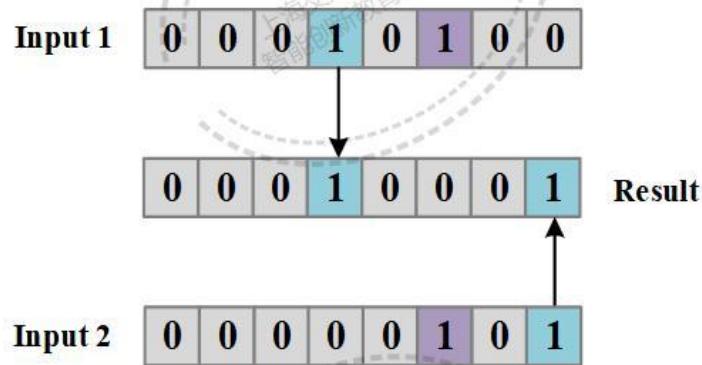
按位与 (&)

二进制位运算—计算机位运算



按位或 ($|$)

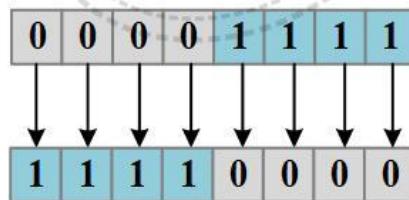
二进制位运算—计算机位运算



- 按位异或运算符(\wedge)
- 这个数的每个位设为1的条件是两个输入数的同一位不同，如果相同就设为0

二进制位运算一计算机位运算

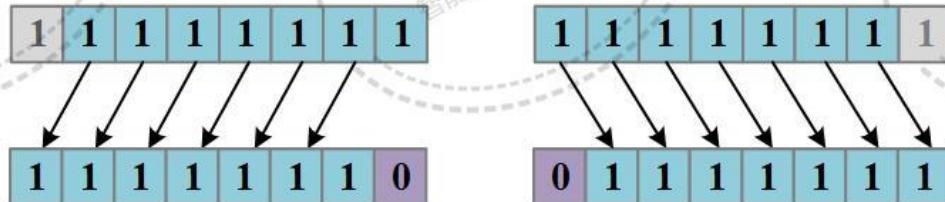
Input 1



Result

按位取反 (\sim)

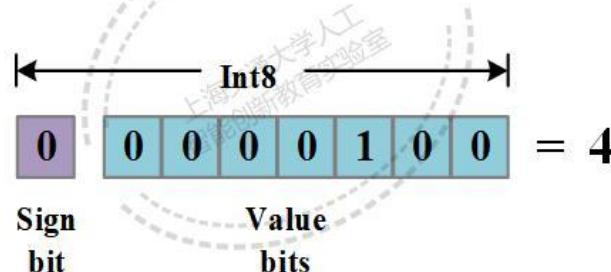
二进制位运算—计算机位运算



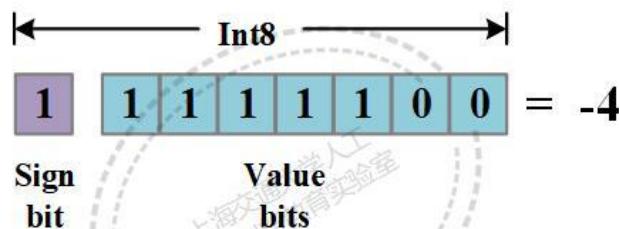
- 按位左移 ($<<$) / 右移运算符 ($>>$)
- 按位左移和按位右移的效果相当把一个整数乘于或除于一个因子为2的整数

上图是无符整形的移位操作，移除的位数被直接抛弃，空位补0，称为逻辑移位

二进制位运算—计算机位运算

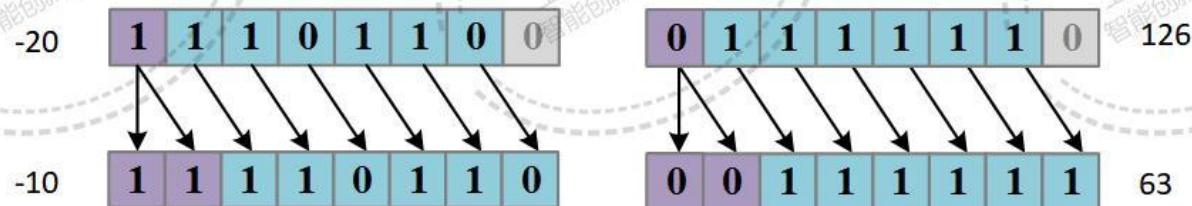


二进制原码表示正数的编码



负数的编码方式称为二进制补码表示

二进制位运算—计算机位运算



- 用二进制补码表示负数，就可以和正数一样对负数进行按位左移右移
- 对有符号整型数的右移如上图所示，使用符号位填充空白位
- 对有符号整型数的左移与无符号整型数的左移相同
- 计算机计算加减乘除，都是转换为加法和位移运算完成的

二进制位运算——二进制系统的优势

技术实现简单，计算机是由逻辑电路组成，逻辑电路通常只有两个状态，开关的接通与断开，这两种状态正好可以用“1”和“0”表示

简化运算规则：两个二进制数和、积运算组合各有三种（与、非、异或），运算规则简单，有利于简化计算机内部结构，提高运算速度

适合逻辑运算：逻辑代数是逻辑运算的理论依据，二进制只有两个数码，正好与逻辑代数中的“真”和“假”相吻合

易于进行转换，二进制与十进制数易于互相转换

用二进制表示数据具有抗干扰能力强，可靠性高等优点。因为每位数据只有高低两个状态，当受到一定程度的干扰时，仍能可靠地分辨出它是高还是低