

讲座，中国科学技术大学，2020年6月2日

逻辑、计算与智能

陈小平

一、纯逻辑：从公理化到形式化

二、应用逻辑：从数学到计算

三、计算：从模型到编码

四、智能：从信息空间到现实世界

五、结语：从继承到发展

从逻辑学到计算机科学的发展史

- Euclid(B.C 330-275): 《几何原本》，第一个公理系统，从5条公设证明了465条几何命题。
- Leibniz(1661-1716): 提出通用语言设想。
- Boole(1815-1864): 布尔代数，小范围的通用语言系统。
- Frege(1848-1925): 1879年初步完成谓词演算构建，第一个形式公理系统。
- Hilbert(1862-1943): Hilbert方案，试图用有穷主义方法证明数学相容性。
- Godel(1906-1978): 发明形式算术K_N和递归函数，证明了足够复杂的形式理论的相容性不能在系统内部形式证明。
- Turing(1912-1954): 1937年提出Turing机模型和Church-Turing论题，标志着计算机科学的诞生。1950年提出“图灵测试”，标志着人工智能的诞生。

一、纯逻辑：从公理化到形式化

一、纯逻辑：从公理化到形式化

❖ **逻辑** 研究推理的正确形式。常见的推理模式：

1. 演绎推理

例

鸟会飞
燕子是鸟　} 前提

燕子会飞　} 结论

◆ **保真性** 前提真则结论真。

2. 归纳推理 例

燕子会飞
乌鸦会飞
……

鸟会飞

◆ **保假性** 前提假则结论假。

3. 类比推理

例

燕子是鸟，燕子会飞
乌鸦是鸟

乌鸦会飞

◆ 不保真、不保假。

一、纯逻辑：从公理化到形式化

- **纯逻辑演算**K 演绎推理的形式公理化
  1. 语言K(Y)：逻辑符号，非逻辑符号；
  2. 公理模式：(K1) ~ (K5)；
  3. 推理规则：(MP)，(UG)；
  4. 公设：无；
  5. 形式推理$\Gamma \vdash p$，形式证明$\vdash p$，其中Γ是公式集、p是公式；
  6. 定义：$\wedge, \vee, \leftrightarrow$。
- **语义** 一阶结构、语义解释、Frege组合原则、语义后承$\Gamma \models p$；
- **元理论** 可靠性和完全性：$\Gamma \vdash p$ 当且仅当 $\Gamma \models p$。

一、纯逻辑：从公理化到形式化

2. 公理模式

(K1) $p\to(q\to p)$;

(K2) $(p\to(q\to r))\to((p\to q)\to(p\to r))$;

(K3) $(\neg p\to\neg q)\to(q\to p)$;

(K4) $\forall x\,p(x)\to p(t)$, 项t对$p(x)$中x自由;

(K5) $\forall x(p\to q)\to(p\to\forall x\,q)$, x不在p中自由出现.

3.推理规则

(MP) 从p和$p\to q$推出q. (分离规则)

(UG) 从p推出$\forall x\,p$. (概括规则)

一、纯逻辑：从公理化到形式化

❖ **例** 命题“并非雪是黑的等值于雪是白的∨雪是红的∨…∨雪是无色的”在一阶逻辑中表达为：

$$\neg C(snow, black) \leftrightarrow$$

$$(C(snow, white) \vee C(snow, red) \vee \ldots \vee \forall x \neg C(snow, x))$$

❖ **观察** 当对物体的性质和关系进行推理时，如对物体颜色进行推理时，需要这样的领域知识作为推理前提。但纯逻辑系统K并不包含这些知识，因此在应用中需要将具体领域的基础知识表达成一阶公式，这些基础知识称为该领域的公设。

❖ **公设** 非逻辑符号语义的规定；**公理** 逻辑符号语义的规定。

一、纯逻辑：从公理化到形式化

❖ **非逻辑符号的语义** 纯逻辑系统不要求给出非逻辑符号的语义定义；历史上某些定义曾导致悖论，引发第三次数学危机。

◆ **例(罗素悖论)** 令**D**是所有集合的集合，非逻辑符号**X**代表**D**中的一个集合$\{x\in \mathbf{D} \mid \neg(x\in x)\}$，$\in$代表**D**中集合之间的属于关系。则假设命题$\mathbf{X}\in\mathbf{X}$为真或者$\neg(\mathbf{X}\in\mathbf{X})$为真都产生矛盾。这违背了矛盾律、排中律和数学界的共识(任何命题有真假)。

❖ **纯逻辑的成果** $\Gamma \vdash p$具有有效性/保真性：只要Γ是真的，则p是真的，并与非逻辑符号的语义无关，故适用于不同应用领域。可靠性和完全性保证$\Gamma\vdash p$具有有效性。$\Gamma\vdash p$可以自动化。

二、应用逻辑：从数学到计算

二、应用逻辑：从数学到计算

- **悖论** 不明原因的逻辑矛盾；如罗素悖论在任何情况下都导致矛盾却找不到原因。逻辑矛盾的严格描述是不相溶性。
- **不相容性** 任何公式集Γ是不相容的，如果存在闭式p使得$\Gamma \vdash p$并且$\Gamma \vdash \neg p$都成立(即从Γ推出一对矛盾命题)；否则Γ是相容的。
- **平凡性** 如果Γ是不相容的，则对一切p，有$\Gamma \vdash p$。
- **观察** 19世纪末发现的一批数学悖论提示，数学可能隐含着被普遍忽略的深层缺陷，数学知识体系可能存在着未被发现的不相容性。史称第三次数学危机。

二、应用逻辑：从数学到计算

❖ **希尔伯特方案** 针对第三次数学危机，希尔伯特提出一个解决方案，要点如下：

1. 数学分支形式化，使得一个数学分支是不相容的，当且仅当它的形式化理论是不相容的，从而将数学分支的相容性问题归结为对应的形式化理论的相容性问题；

2. 用有穷主义方法，证明各数学分支形式化理论的相容性。有穷主义方法本身的无矛盾性必须是显而易见的、不证自明的。

❖ **科学理论的公理化** 牛顿力学是一个公理化系统，所有定理都从牛顿三定律(公设)推出；但牛顿力学没有形式化。

二、应用逻辑：从数学到计算

❖ **形式算术K_N** 初等数论片段的形式化理论/应用逻辑演算。

1. 一阶语言$K_N(Y)$：逻辑符号同K；非逻辑符号：个体常元**0**，一元后继函数符号′，二元函数符号**+**、**×**，二元常谓词符号**=**;
2. 公理模式：同K;
3. 推理规则：同K;
4. 公设：等词公设，算术公设;
5. 形式推理$\Gamma \vdash_{K_N} p$和形式证明$\vdash_{K_N} p$：公设与公理同样使用;
6. 定义：同K.

二、应用逻辑：从数学到计算

4. 公设（非逻辑符号基本属性的形式化定义）

(E1) $u = u$;

(E2) $u_k = u \to g(u_1, \ldots, u_k, \ldots, u_n) = g(u_1, \ldots, u, \ldots, u_n)$;

(E3) $u_k = u \to (P(u_1, \ldots, u_k, \ldots, u_n) \to P(u_1, \ldots, u, \ldots, u_n))$.

} 等词公设

(N1) $\neg(u' = 0)$; 自然数定义

(N2) $u' = v' \to u = v$; 自然数定义

(N3) $u + 0 = u$ 加法定义

(N4) $u + v' = (u + v)'$ 加法定义

(N5) $u \times 0 = 0$ 乘法定义

(N6) $u \times v' = (u \times v) + u$ 乘法定义

(N7) $p(0) \wedge \forall x(p(x) \to p(x')) \to \forall x p(x)$ 归纳法定义

} 算术公设

二、应用逻辑：从数学到计算

- **数学分支的形式化** 一个数学分支M的形式化在纯逻辑演算K上增加公设Γ(M基础性知识)，使得：1. M是Γ的模型，即M⊨Γ；2. 通过Γ⊢ p可推出M的其他知识p ($p\notin\Gamma$并且M⊨p)。
- **初等数论的一阶结构N** 初等数论的一阶结构N=(**N**, **F**, **P**)，其中**N**是自然数集，**F**包含**N**上的后继函数+1、加法函数+和乘法函数×，**P**包含**N**上的相等关系=，N还规定非逻辑符号语义解释：

  0解释为0；′解释为+1； **+**解释为+； **×**解释为×； **=**解释为=。
- **初等数论** 即N上所有真命题的集合Th(N)。令Th(K_N)={p | p是闭式且⊢$_{K_N}$ p}。K_N是N的完全形式化如果Th(K_N)=Th(N)。

二、应用逻辑：从数学到计算

❖ **完备性** 任何公式集Γ是完备的，如果Γ是相容的，并且对任何闭式p，$\Gamma \vdash p$或者$\Gamma \vdash \neg p$成立；否则Γ是不完备的。

❖ **定理** 对任意公式集Γ，若Γ完备且$M \models \Gamma$，则$Th(\Gamma) = Th(M)$。

◆ **注释** $Th(\Gamma)$是从Γ形式推出的所有闭式的集合，$Th(M)$是在M中为真(M有效)的所有命题的集合。

❖ **观察** 该定理应用于初等数论，只要证明了K_N的完备性，也就证明了初等数论中所有真命题集合$Th(N)$的相容性。这是哥德尔的初衷，最后他却证明了相反结果。

二、应用逻辑：从数学到计算

❖ **定理(哥德尔第一不完备性定理)** 若K_N是ω相容的，则K_N是不完备的。

◆ **证明** 通过引进哥德尔编码、递归函数和K_N可表示，构造自指命题$p(\underline{m})$，其含义为“$p(\underline{m})$在K_N中不可证”，并证明

$$\vdash_{K_N} p(\underline{m}) \text{ 和 } \vdash_{K_N} \neg p(\underline{m})$$

都不成立，因而K_N是不完备的。

❖ **定理(哥德尔第二不完备性定理)** 若K_N是相容的，则$\vdash_{K_N} con(K_N)$不成立，即K_N的相容性$con(K_N)$不能在K_N中证明。

二、应用逻辑：从数学到计算

❖ 第三次数学危机的成果

1. 哥德尔不完备性定理及相关结果如Godel-Rosser定理、K_N语法不可判定性和语义不可判定行定理表明，对于K_N这样复杂的形式化理论的相容性、判定性等问题不能通过有穷主义方法，在一个给定的形式化理论/应用逻辑演算内部得到证明。因此，希尔伯特方案无法实现。

2. 采用其他手段排除了引起第三次数学危机的隐患。

3. 哥德尔不完备性定理证明的技术发明如递归函数等，引发了可计算性理论的诞生，从而奠定了计算机科学的基础。

三、计算：从模型到编码

三、计算：从模型到编码

❖ **计算模型：递归函数** 三个基本函数及由它们经有限次应用三个规则生成的函数称为(一般)递归函数。

❶ **基本函数** 零函数z，$z(n) = 0$；后继函数s，$s(n) = n+1$；投影函数p^k_i，$p^k_i(n_1, \ldots, n_k) = n_i$, $i=1, \ldots, k$。

❷ **规则** 复合规则、递归规则、μ算子。

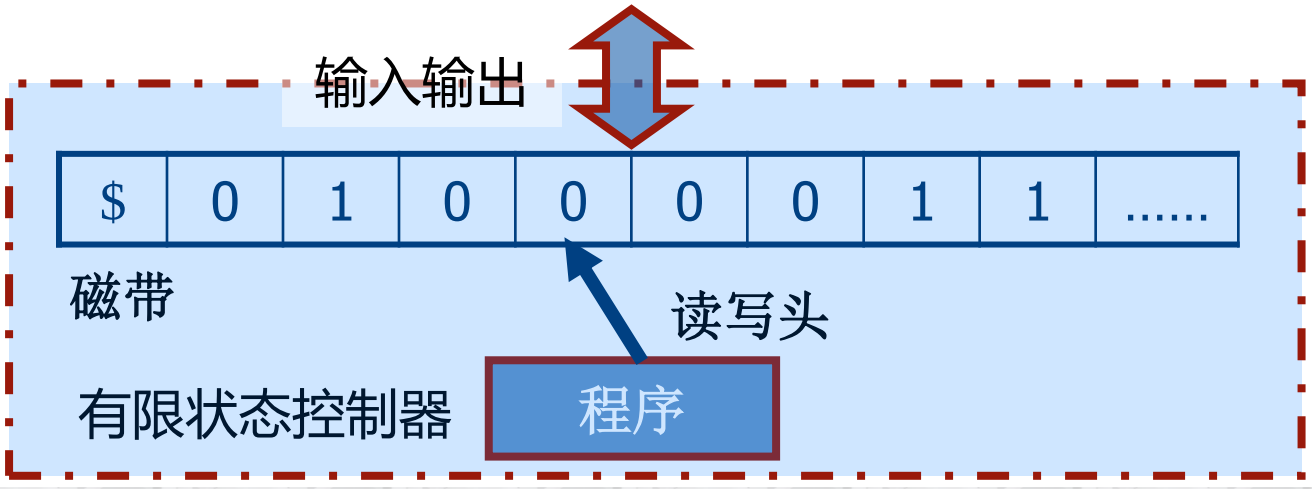
❖ **观察** 三个基本函数是能行可计算的，三个规则的应用保持能行可计算性，所以递归函数是能行可计算的，即对递归函数的任何输入，保证在有限时间内得到计算结果。

❖ 递归函数是计算机科学诞生之前出现的一种计算模型。

三、计算：从模型到编码

❖ 计算模型：标准图灵机模型-构成

1. 一条左端有限、含无限多个存贮单元的磁带，每个单元存贮一个符号；
2. 一个有限状态控制器，存贮一个程序，每时刻处于有限个状态之一；
3. 一个读写头，任一时刻注视一个存贮单元，可写一个符号并移动。



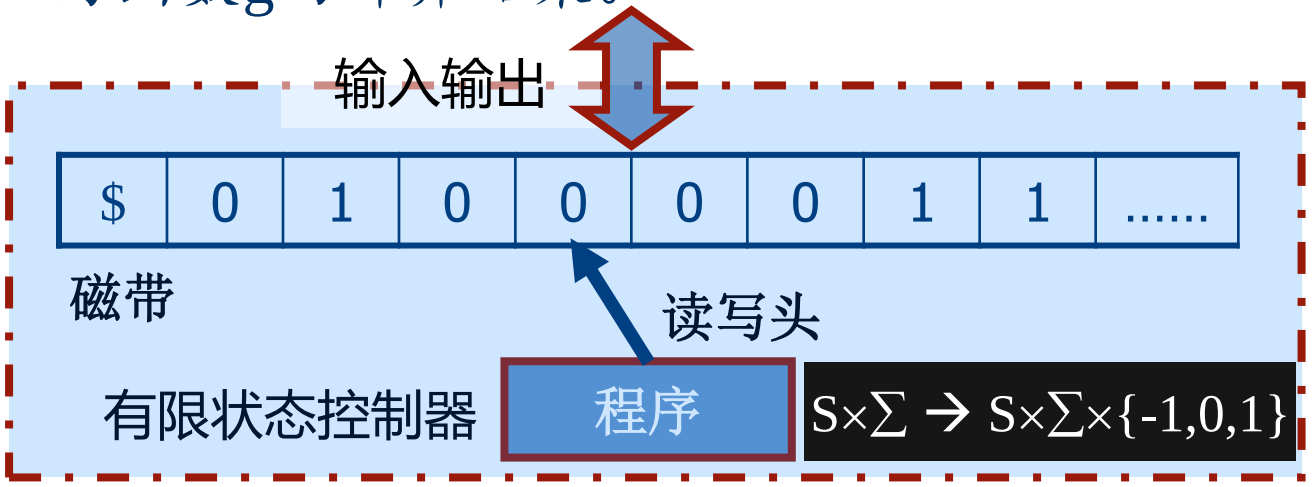


艾伦-图灵
计算机之父
人工智能之父

三、计算：从模型到编码

❖ 计算模型：标准图灵机模型-运行

1. 待计算函数g自变量的值输入到磁带上；
2. 从初始状态运行事先存贮的计算g的程序；
3. 运行到终止状态时停机，磁带上的内容即为函数g的计算结果。



◆ 有限状态控制器/程序

$S \times \Sigma \rightarrow S \times \Sigma \times \{-1,0,1\}$

S：状态集，从当前状态转移到下一状态；

∑：符号集，改写当前单元的符号(可相同)；

{-1,0,1}：读写头左移、不动或右移到下一单元。

三、计算：从模型到编码

- **车赤-图灵论题**(Church-Turing Thesis) 一个函数是可计算的，当且仅当该函数是图灵机可计算的。
- **与图灵机等价的计算模型** 图灵机是通用电子数字计算机的理论模型，因为合理的计算模型都验证与图灵机等价：
  - 递归函数；
  - K_N可表示函数；
  - λ-演算；
  - Post机器；
  - ……

三、计算：从模型到编码

❖ **信息编码：哥德尔数** 符号/符号串/符号串序列编码为自然数。

1. 符号u的哥德尔数g(u)规定为奇数：

u	′	+	×	¬	→	∀	=	0	x_i (i=1,2, …)
g(u)	1	3	5	7	9	11	13	15	15+2i

2. 符号串$u_0u_1\ldots u_k$的哥德尔数为偶数，幂指数为奇数：

$g(u_0u_1\ldots u_k) = 2^{g(u_0)}3^{g(u_1)}\ldots p_{k+1}^{g(u_k)}$，$p_{k+1}$是第k+1个素数；

3. 符号串序列$S_0, S_1, \ldots, S_k$的哥德尔数为偶数，幂指数为偶数：

$g(S_0, S_1, \ldots, S_k) = 2^{g(S_0)}3^{g(S_1)}\ldots p_{k+1}^{g(S_k)}$，$p_{k+1}$是第k+1个素数。

三、计算：从模型到编码

- 计算过程编码：递归函数

  1. 由三个基本函数经有限次应用三个规则编码而成的函数；

  2. 任意定义的可计算函数可编码为递归函数。

- 计算过程编码：图灵机有限状态控制器/程序

  $S\times\sum \rightarrow S\times\sum\times\{-1,0,1\}$

  任意定义的可计算函数可编码为图灵机计算的函数。

- 计算过程的其他编码方案 K_N可表示(知识/推理的编码方案)、λ-演算、Post机器、程序设计语言……

三、计算：从模型到编码

❖ 什么是计算？

计算 = 编码

编码 = 信息编码 + 计算过程编码

信息编码 = 数据编码 + 知识编码

计算过程编码 = 过程型编码 | 函数型编码 | 逻辑型编码 |…

❖ 对比

1. 计算模型观点：选择一个计算模型，完成指定的计算任务；
2. 编码观点：根据问题求解需要，组合使用所需的编码方案。

三、计算：从模型到编码

❖ 例 组合哥德尔编码和递归函数编码，下列9个集合成为可计算的：

1. $\{g(u) \mid u是K_N项\}$；
2. $\{g(u) \mid u是K_N公式\}$；
3. $\{g(S) \mid S是K_N公式序列\}$；
4. $\{g(p) \mid p是Ki公理\}$，i=1,2,3,4,5；
5. $\{g(p) \mid p是Ei公设\}$，i=1,2,3；
6. $\{g(p) \mid p是Ni公设\}$，i=1,2,3,4,5,6,7；
7. $\{(n_1, n_2, n_3) \mid n_1=g(p), n_2=g(p\rightarrow q), n_3=g(q)\}$；
8. $\{(n_1, n_2) \mid n_1=g(p), n_2=g(\forall x\, p)\}$；
9. $\{(n_1, n_2) \mid n_1=g(p), n_2=g(S), S是p的一个K_N证明\}$。

计算通常同时涉及信息编码和计算过程编码

四、智能：从信息空间到现实世界

第一种AI经典思维：基于模型的强力法

强力法原理：

1. 基于问题的精确模型，建立知识表示/搜索空间；

2. 压缩搜索空间；

3. 在压缩空间中通过推理法或搜索法，穷举问题的可能解，找出其中的最优解。

推理的主要工具：

逻辑推理、概率推理、决策论规划(MDP/POMDP…)

推理法工作原理

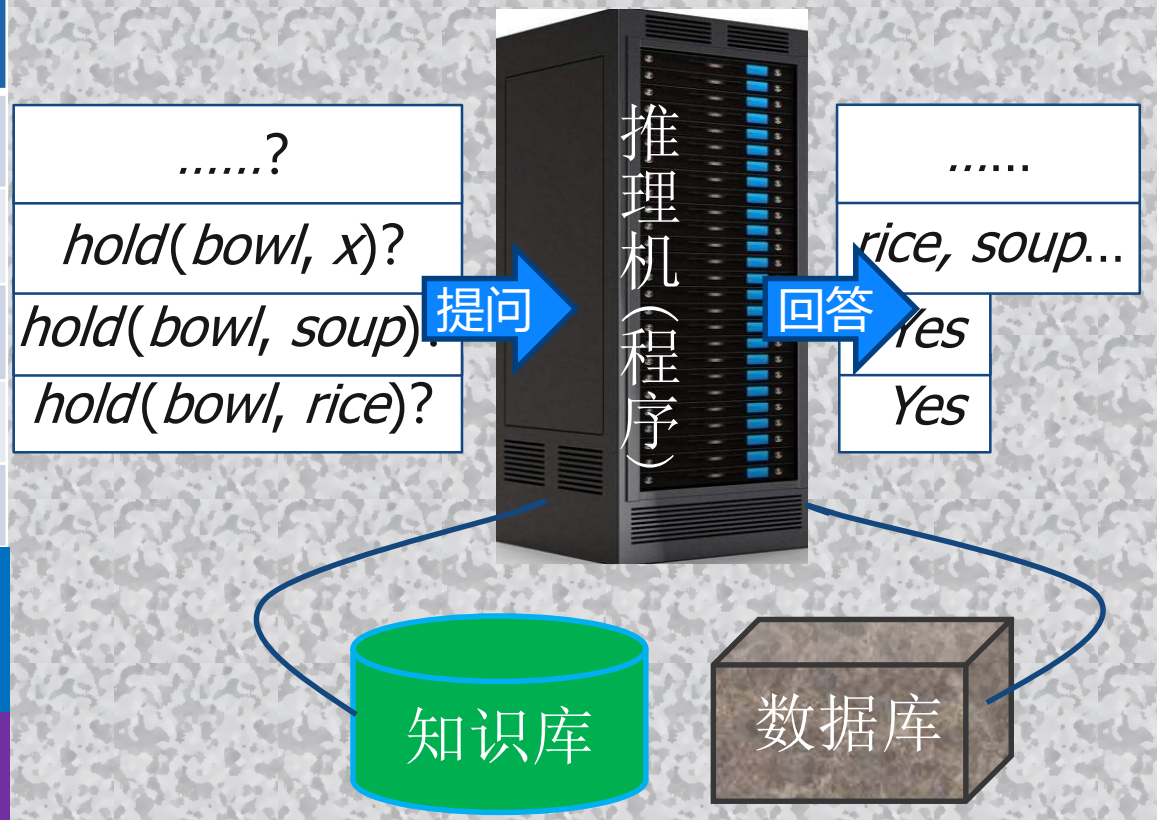
❖ **自动推理系统**：利用已有知识库和数据库，通过推理对提问自动产生答案。

1. 事实推理 用已知知识/数据推理，例：

逻辑公式	语义解释
$\forall x,y(dish(x)\wedge food(y)\rightarrow hold(x, y))$	餐具可以盛食物
food(*rice*)	米饭是食物
food(*soup*)	汤是食物
dish(*bowl*)	碗是餐具
……	……

2. 假设推理 用假设知识推理并检验。例：设计相对论日食观测；搜索本-拉登藏身之处。

3. 反事实推理 用违反事实的知识进行推理。例：“追光实验”。



第二种AI经典思维：基于元模型的训练法

- **训练法原理：**

1. 用人工神经元网络表示问题的元模型(输入-输出对应)；

2. 用标注数据，依数据拟合原理，训练元模型参数值(人工神经元网络连接权重)，得到问题的一个具体模型。

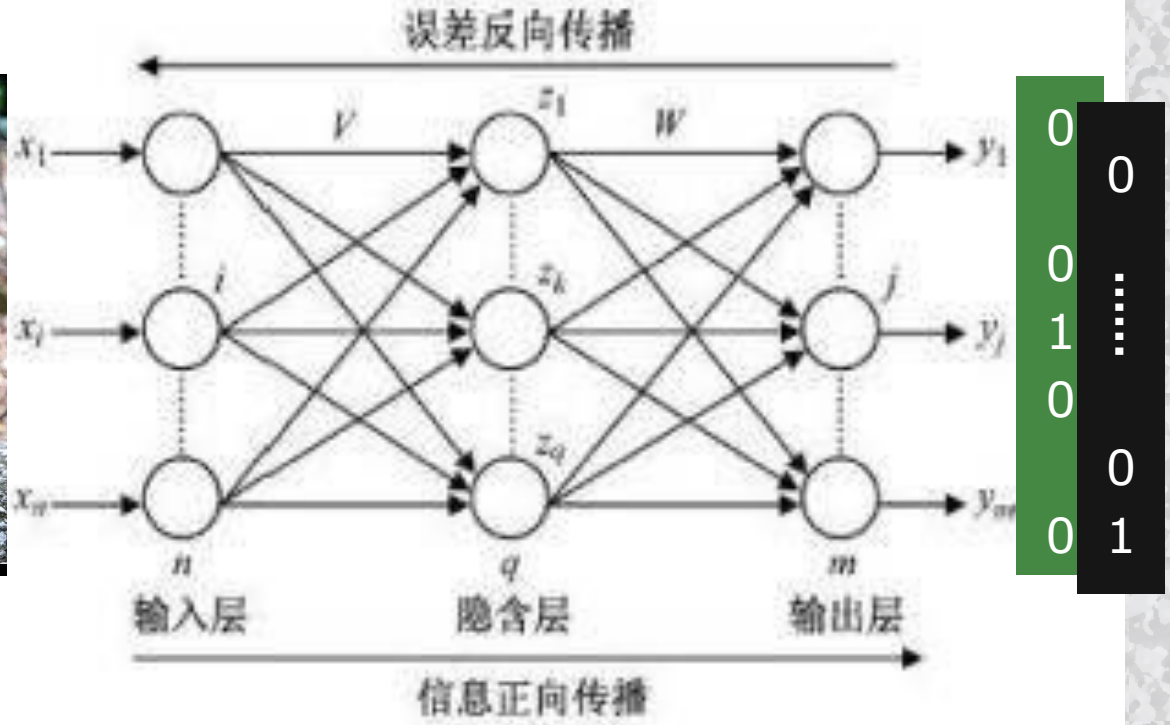
- **机器学习方法分类**：监督学习（训练法）、无监督学习
- **训练者职责**：设计和决定训练科目(训练任务)、训练材料(带标注的训练数据的采集)、训练过程(学习算法)、评价准则（训练成功的标准和指标）、测试手段(测试方法与工具).....

训练法工作原理

- **计算过程**：对每一个“输入”用人工神经网络计算对应的“输出”。例如，在著名的ImageNet图像分类任务中：

数 据	人工标注
7种鱼的图片	0 - 6
公鸡、母鸡图片	7 - 8
26种鸟的图片	9 - 34
……	……
卫生纸图片	999



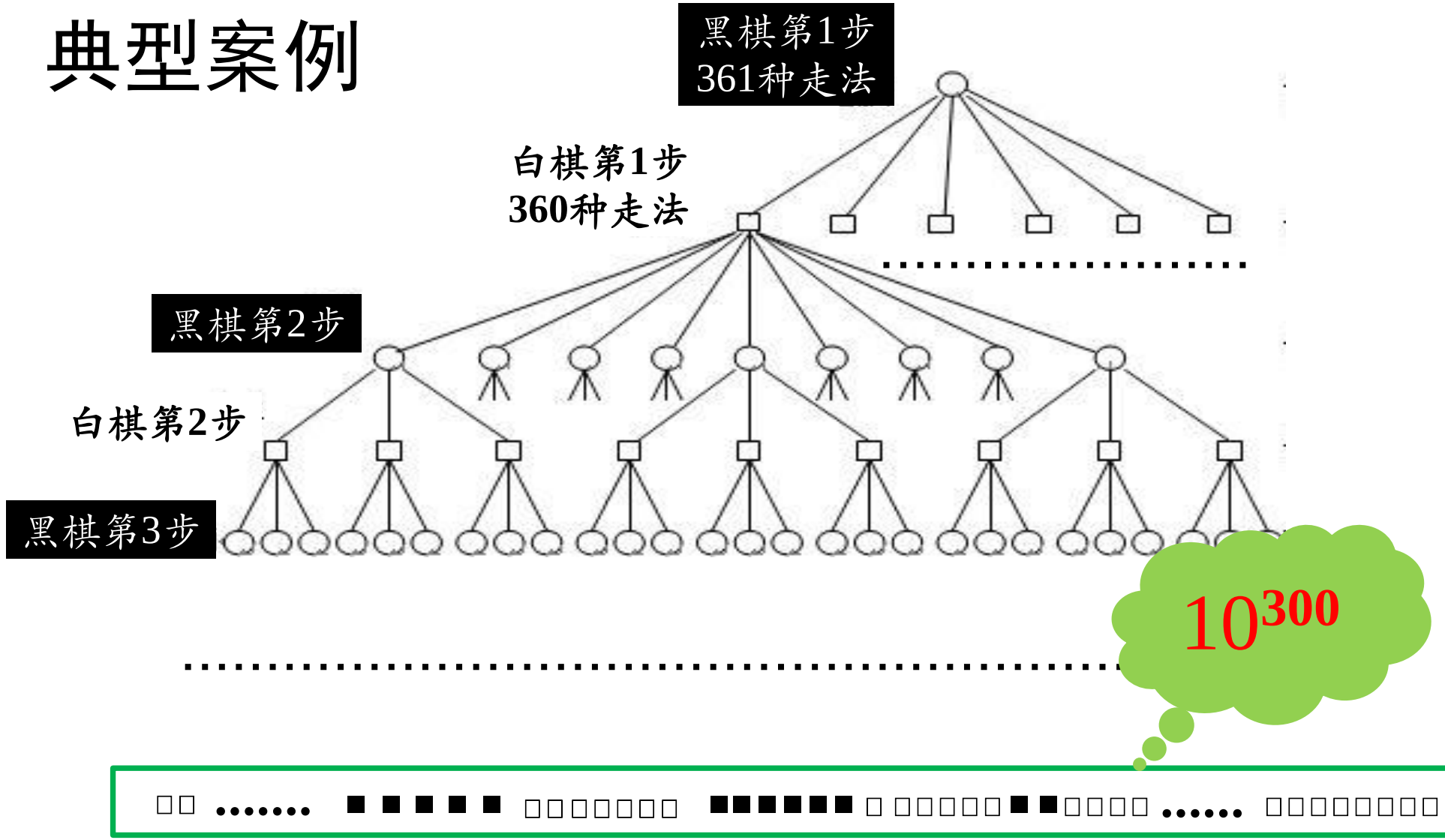


- **训练过程**：预先采集训练样本并进行人工标注。用**监督学习算法**训练神经网络，尽可能缩小网络输出的整体误差。
- **隐含大量人工工作，包括数据标注等等。**

ImageNet a large visual database for use in visual object recognition, More than 14 million images have been hand-annotated.

https://en.wikipedia.org/wiki/ImageNet

典型案例



□□ ..... ■ ■ ■ ■ ■ □□□□□□□ ■■■■■■ □ □□□□□■■□□□□ ..... □□□□□□□□□□

围棋可能下出什么棋，由围棋规则决定，棋手只是选择规则允许的走法。

围棋博弈树/搜索树

10^{300}有多恐怖？

- 地球上的海水共有13亿8600万立方千米，大约7×10^{34}滴水。
- 一瓶500毫升矿泉水包含约2.5×10^{4}滴水。10^{4}与10^{34}之比！
- AlphaGo Zero最终版用40天下了二千九百万盘棋。2.9×10^{7}与10^{300}相比如何？——几乎没下过几盘棋！
- 如果单纯使用强力法搜索2.9×10^{7}盘棋，无法足够好地自动凝练自博获得的下棋策略，所以不可能达到AlphaGo Zero的效果。
- 假设全球所有现役九段棋手人工标注2.9×10^{7}盘棋，每人每分钟标注一盘棋，需要至少100年！因此，单纯使用训练法，仅标注数据都不可行。

AlphaGo Zero四大核心技术

1. 简化的决策论模型；
2. 蒙特卡洛树搜索；

} 强力法(推理法+搜索法)

3. 深层神经网络（残差网络）；
4. 强化学习。

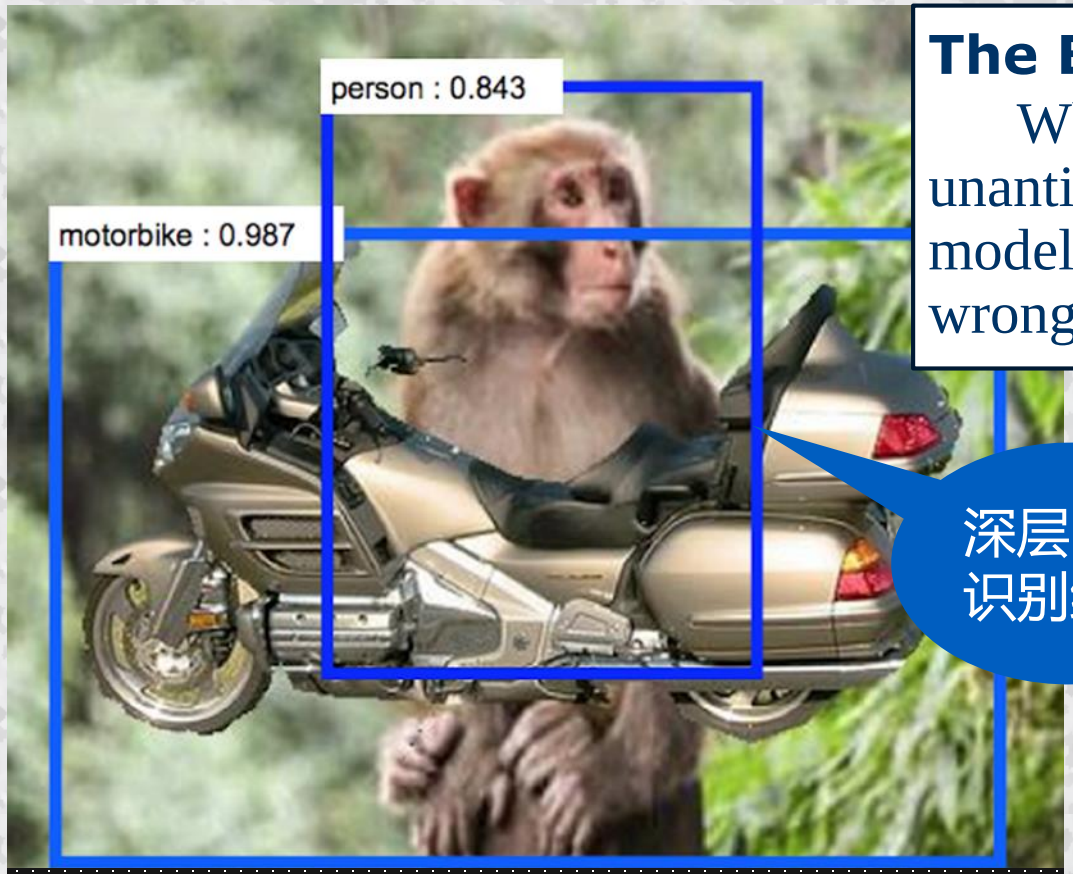
} 训练法

Mastering the game of Go without human knowledge, *Nature*, Oct. 18, 2017

AlphaGo仅仅是深度学习的胜利？——巨大而有害的误解！

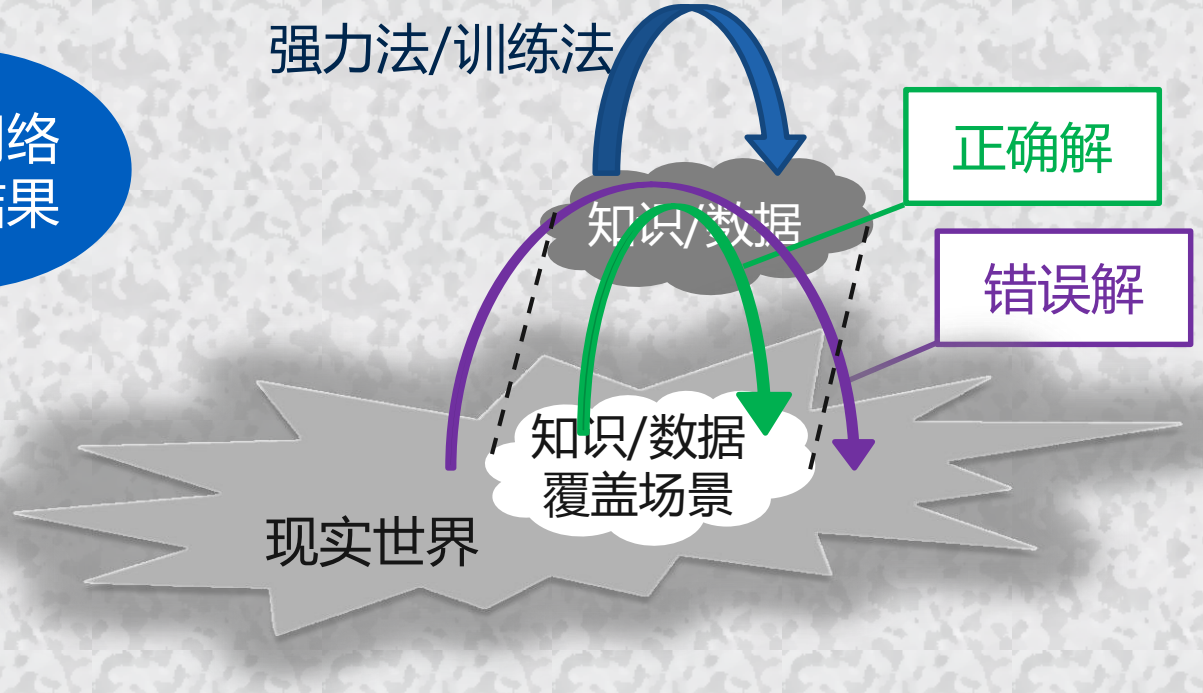
AlphaGo Zero没有重大原始创新？——巨大而有害的误解！

强力法和训练法的主要应用瓶颈——脆弱性



The Brittleness Problem

When an intelligent system is confronted by unanticipated situations which do not fit the internal model of the system, the system is likely to reach wrong conclusions or even breaks.



现实世界：隐变量

❖ **信息空间** 例如初等数论的一阶结构N=(**N**, **F**, **P**)，其中 **N** 是自然数集，**F**={+1, +, ×}, **P**={=}. 根据函数x + y = z的定义，对任何自然数x和y，存在唯一确定的自然数z，使得x + y = z。因此，函数定义和计算过程不受任何隐变量的干扰。

❖ **现实世界** 必然存在不可预知和不可控因素（隐变量）的干扰。例如，即使对于固定的物体（如一只松鼠）和固定的遮挡物（如一片树叶），可以存在无穷多种干扰方式，如无限多种不同的遮挡方式，并且无法获得关于隐变量干扰的充分知识和数据。

围棋是最难的智能问题吗？

人工智能中有没有本质上比围棋更难的问题？

有！而且更多！

——非封闭性问题

现有AI技术产业化应用的成功条件

在下列三个条件下
现有人工智能技术有望大面积应用

1. **领域封闭性**：应用领域中不存在隐变量；
2. **失误非致命性**：应用中不会出现致命性失误；
3. **基础条件成熟性**：具备应用所需的基本条件。

工业机器人：非封闭性问题的封闭化(结构化)

自然生产过程是非封闭性/非结构化的



改造为封闭性/结构化的自动化生产过程

参考案例：汽车生产线

服务机器人：非封闭性问题的准封闭化

高铁全线封闭，消除行车过程中的环境干扰。



乘客须遵守乘车规定，并未完全封闭化。

参考案例：高铁

五、结语：从继承到发展

精确性思维传统：面向封闭性问题

Ευκλειδης
公理化逻辑







Isaac Newton & Gottfried Leibniz
数学分析



N. Bernoulli
概率论



Herbert Simon
现代决策论

?

五、结语：从继承到发展

现有人工智能技术可解决封闭性问题

人工智能基础研究需突破精确性思维

“融差性”思维正在提供全新机遇

精确性+融差性+？

将重新定义人工智能，开辟人工智能新天地。

陈小平，人工智能中的不确定性问题与哲学研究，《智能与智慧——人工智能遇见中国哲学家》，中信出版社，2020.


谢 谢 ！