

## 第二堂课

# 控制与信息理论

### 摘要：

- ◆ 信息理论：严格的部分。信息的概念，集合和区分所对应的数理逻辑概念。
- ◆ 同形式逻辑的紧密联系。利用自动机模型做不同的分析。这两种做法的共同特点：具备或不具备的特征。把两种分析联系在一起的工作。
- ◆ 两种描述自动机的方法：零件组合或者整体处理。
- ◆ 零件组合的方法：基本单元的性质，同神经元的类似之处。McCulloch & Pitts 理论：形式神经网络，他们主要的结论。
- ◆ 整体处理的方法：图灵机理论。图灵机和数学问题的联系，何种数学问题能够用图灵机解决。通用自动机的概念。图灵的主要结论。
- ◆ McCulloch & Pitts 的神经网络以及图灵自动机的局限。输入和输出元件。一般意义上的计算。利用传感器和运动元件的分析方法。

### 一、概述

【冯纽曼说信息理论包括两大块：严格的信息论和概率的信息论。以概率统计为基础的信息理论大概对于现代计算机设计更加重要。但是在此之前，我们必须先弄清楚严格的信息论那部分，它其实就是形式逻辑的另一种处理方式。】

【他接下来解释了一些形式逻辑的基本概念，简要地说明了真值函数的连接词，比如“与”“非”“如果...那么”“与非”以及它们之间的相互定义性。他还解释了变量和量词，包括“全称”量词和“存在”量词。他的结论是“如果你有这样的一台计算机，就可以表达一切数学计算，或者能够纯粹地用数学计算表达出的任何主题。”】

对于计算机，我不打算深谈。因为存在另外一种不一样的机器，同信息理论也很有关系，就是 McCulloch & Pitts 提出的神经网络理论，你可以说图灵机和 McCulloch & Pitts 的神经网络分别处于信息理论的两个极端。这两套理论都力图建立一个公理化的体系，用某种假设的理想机器来建立形式逻辑系统，但并不实际去制作这样的机器。他们都成功地说明了形式逻辑同他们设想的机器是完美兼容的，也就是说，机器能做到的一切工作，都能够被形式逻辑所刻画；反过来，任何能够用形式逻辑描述的事物，也都能够用这类机器来运行。【冯纽曼这里假设了 McCulloch & Pitts 的神经网络有着一条无限长的纸带，结果表明了它同图灵机的等价性。这个结果也就是图灵可计算性、函数的  $\lambda$  可定义性、以及一般递归的概念。请参见图灵的论文“可计算性和  $\lambda$  可定义性 (Computability and  $\lambda$ -Definability)”】

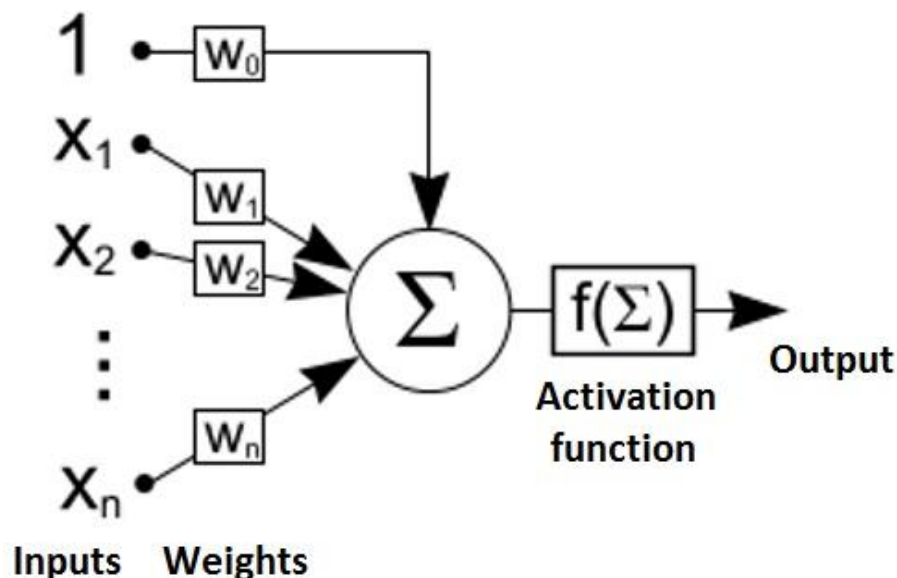
我会简单地介绍 McCulloch & Pitts 的神经网络以及图灵的工作。因为它们各自代表了一种重要的研究方式：组合方法和整体方法。McCulloch & Pitts 描述了一套方法，由非常简单的零件组成复杂结构。因此只需要对底层的零件作公理化定义就可以得到非常复杂的组合；图灵则是对于整个自动机进行了公理化的定义，他仅仅定义了自动机的功能，并没有涉及到具体的零件。

## 二、McCulloch & Pitts 的神经网络

McCulloch & Pitts 的工作具有很明确的目的，就是为了研究人类的神经系统而提出一套简单的数学逻辑模型。结果惊喜地发现这个模型跟形式逻辑等价，这部分实现了 McCulloch & Pitts 的研究目标，但仅仅是一部分。他们的模型研究的是神经本身，我之前也对这个问题很感兴趣。为了能够切入本质，而不被神经系统的生理和化学的具体细节所干扰，他们使用了一套公理化方法，用几个简单的假设来描述神经的行为。在此，我们暂时先不管这些假设具体是怎样实现的，同实际又是否一致。

他们接下来又对神经元作了进一步的理想处理。这一步引来了许多指责，但在我看来却是完全合理的。McCulloch & Pitts 并没有按照实际的情形去公理化神经元，而是设计了一种完全理想化的模型，它要比实际的简单许多。他们相信，这一极度简化的模型包括了神经元最核心的那些特性；而所有的其它性质只是无关紧要的细节，在初步研究中应当忽略。我认为科学界还需要很长的时间，才能完全认同他们的观点。我们并不知道在简化模型中被忽略的那些细节是否真的无关紧要，但至少，在理想化处理以后，神经网络变得相当容易理解了。

神经元的定义如下：我们也许应该把它叫做形式神经元，因为它并不是真正的神经细胞，而是仅仅具有神经细胞的一些关键性质。我们用一个圆圈代表一个神经元，从圆圈延伸出的直线则代表神经突触。箭头表示某神经元的突触作用于另一个神经元之上，也就是信号的传送方向。神经元有两个状态：激发和非激发。我们无需关心激发具体是什么意思，因为激发的主要性质通过实际运行来体现。这儿有一点点循环定义：所谓激发态，就是能够激发别的神经元的状态。大致在神经网络的末端，处于激发态的神经元会激发某些非神经元的东西来实现其目的。比如，肌肉会受刺激，从而产生身体的运动；或者刺激某种腺体产生化学变化，分泌荷尔蒙之类。所以，神经激发最终产生的结果是我们研究范围以外的事情，我们不考虑这类现象。



McCulloch-Pitts 的神经元模型（译者加）

（图片来源：<http://encefalus.com/neurology-biology/neural-networks-real-neurons/>）

【冯纽曼说明了控制神经相互作用的原理：根据 McCulloch & Pitts 模型，他也假设神经激发之间有一段均匀的延迟，虽然他们暂且不考虑“神经疲劳这一重要现象，也就是说，神经被激活之后，有一段时间因疲劳而对刺激失去反应”，这种神经疲劳对于生命功能有着重要的作用。但是，就算存在疲劳因素，我们还是可以用串联神经元的办法来实现连续的运动。冯纽曼还定义了神经阈值，提出了抑制性突触的概念，用小圈（而不是箭头）来表示】<sup>1</sup>

【接下来，冯纽曼讲解了“McCulloch & Pitts 的重要结论”，想象有一个具有若干输入和唯一输出的黑箱。再选择两个时间点： $t_1$  和  $t_2$ ，并确定从  $t_1$  到  $t_2$  何种输入组合会导致输出，何种输入得不到输出。】不论你如何选择这些条件，总可以在黑箱里设计一套神经网络来实现这些条件。这意味着神经网络在一般意义上是同符号逻辑等价的。这是一个事实，即任何被设计出来的系统都可以用有限长度的描述来严格表示，因此他们是等价的。这里我不打算证明它，因为任何形式系统的证明都是冗长且困难的。【这里简单说一下证明的方法，按照冯纽曼所说，所有的开关函数（真值函数、布尔函数）都可以用确定延迟的神经网络来实现。利用循环的方法，我们可以用神经元来实现任意容量的内存，并附加到以上的开关函数上。这就相当于给一套符号逻辑加上了一根无限长的纸带，也就是实现了图灵机。因此，对于每一台图灵机  $M$ ，总可以制造一套等价的神经网络，计算得到同样的数字】

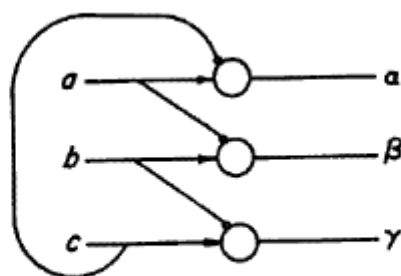


Fig. 1. Neural network in which dominance is not transitive

图：抑制不能传递的神经网络

【冯纽曼展示了如何建立一些样例网络。如上图，a 可以抑制 b，b 可以抑制 c，c 又可以抑制 a。如果输入端被激发（箭头处输入），且抑制端（箭头处输入）没有激发，则神经元进入激活状态。根据此规则，如果 a 和 b 都被激发，那么  $\alpha$  会激发，而  $\beta$  不激发；如果 b 和 c 都被激发，那么  $\beta$  会激发，而  $\gamma$  不激发；并且，如果 a 和 c 都被激发，那么  $\gamma$  会激发，而  $\alpha$  不激发。冯纽曼利用这个例子来说明人类行为的不确定特性，比如 a 比 b 强，b 比 c 强，照理说应该 a 比 c 强。但是在上面的神经网络中间，虽然 a 比 b 强，b 比 c 强，c 却又比 a 强。】

【下面，冯纽曼组合基本神经元得到了几个其他的神经网络：简单的存储器，计数器，基本的学习电路。也就是《冯纽曼著作选》5.342-345 中的电路。学习电路有两个输入端，a 和 b。如果 b 接着 a 之后受到一次刺激，则该电路会记录一次，当这个数字达到 256 以后，无论 a 是否受到刺激，一旦 b 受到刺激，电路都会发出一次脉冲。】

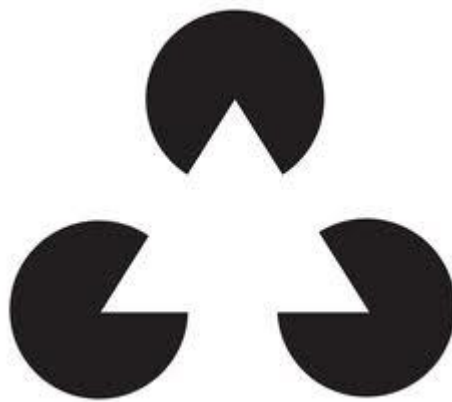
请大家注意，这些电路虽然看上去很复杂，但是从它们的组合方式去观察却很简单。而

<sup>1</sup> 译者注：神经在激发后会疲劳，在一段时间中间不能再次被激发，这一实际生理规律在理想模型中也是应该被考虑加入的。这样可以解决神经网络因为正反馈，而陷入简单同步的死循环等问题。在 Hebb 定律提出之后不久，Milner 就提出了可变阈值、抑制和疲劳的概念，补充了神经网络的特性。只有考虑了这些因素之后，神经网络才能产生比较高级的涌现现象，如同步、预测、和层级等等，才能较好地对于外界进行识别。可参见霍兰著《涌现》的相关章节。

且它们的复杂度与语法的复杂度相当。如果你能用一句话来描述它们的功能，那么你就能毫不费力地把电路图画出来。其实，McCulloch & Pitts 得到的结论也就是图形描述和语义描述没有太大的区别。中继元件的组合描述是和严格的语义描述完全兼容的。

我想阐述一下这个观点的哲学引申。根据上述观点，可以推论：只要能够用人类语言描述的东西，都能够用神经元方法表示出来。同时，神经元并不需要超常的聪明程度及复杂程度。事实上，这种理想的模型甚至不必有真实的神经元那么聪明或复杂，因为简化后的神经元，虽然以死板的、缺乏变化的方式运行，却已经能够做到一切你想得到的事情了。

另一方面，McCulloch & Pitts 没有说的事情当然也同样重要。首先，不是说我们这样设计出来的电路一定存在于自然界中。其次，不是说在这个简化处理中，神经元被忽略的那些功能是无紧要的。最后，虽然理论上我们可以用神经元电路描述任何事情，但这并不是说实际上得到这样的描述就会很简单。换个角度讲，就像人类神经活动一样，也许你能够理解所有单个神经元的运作原理，但是你仍然解释不了这些神经元的整体涌现结果。



没有轮廓的三角形，但是你的眼睛却可以帮你勾勒出它的轮廓（译者加）

图片来源：[http://vision.xitek.com/Classroom/200910/21-29313\\_6.html](http://vision.xitek.com/Classroom/200910/21-29313_6.html)

当你看到一个三角形的时候，你认识到“这是一个三角形”，知道它的大小。怎么解释这个事实呢？三角形的几何定义是很简单的，就是三根直线的组合。但是实际上，三角形的辨认是非常任意的。比如三条边是弯曲的、只能看到三角形的顶点（插入一张三角形的视觉补偿图片）、三角形用阴影表达等等，在这些情况下你都可以认出其中的三角形来。但是，如果你要把所有的这些情况都放到三角形的程序描述中，那么这个描述就会变得无限的冗长。

不仅如此，辨认三角形的能力仅仅是你辨认几何图形的能力中间的极小一部分，几何图形的辨认能力又只不过是视觉认知能力的极小一部分。理论上，你都可以给这些能力写出具体的描述出来。但是这并不是解读一张图片真正的视觉机制，因为要“进入”一张图片之后，你才能把意义赋予它。这是超越图形范围以外的事情，因此不能够用以上的几何图形方式来描述。比如给你看一张墨水染成的抽象图案，每个人都会把自己的理解放进图片中，但是这些放进去的东西取决于这个人过去的经历和他的人格，而不能在图片本身中找到。所以心理医生可以利用此类方法来对人进行心理测试。

这些看上去好像是一些零碎的事情，但基本的事实是：人的大脑超乎想象的复杂，其中有五分之一都是用于视觉处理的。所以，按照我们之前的估算，大约有 20 亿个神经元是专门用于建立视觉图形的。除了直接描述视觉大脑以外，现在我们还没有别的更简单的办法去描述什么成分构成了视觉类比（visual analogy）。

### 三、不同层次的复杂性

一般地说，自动机的文字描述应该比给出自动机的全部设计图纸要简单。但是这个结论也不是在所有情况下都成立。在形式逻辑中，只要自动机的结构不是非常复杂，那么它的文字描述的确要比自动机本身简单。但在复杂度很高的情形下，实际的自动机结构反而要比它的描述更简单一些。

我实际上正在用自己的方式解释一条非常漂亮的逻辑定理，这是由哥德尔曾经提出的一条定理：从逻辑上说，对于一个对象的描述要比这个对象本身高一个级别。因此，渐进地说，前者总是比后者要长。我觉得为了理解这种性质，必须运用上述定理，因为在你去深入研究这件事情的时候，你已经触及到复杂度这个概念的本质。有理由怀疑，在一切模糊的、难以定义的事物背后，都存在着复杂度渐近的性质（比如“看上去相似是什么意思？”），给人的感觉是，不管你用多少文字，都没有办法把问题彻底描述清楚。这类事物就属于高复杂度的情况，也就是实际去做，让机器实际去运行，反而比直接描述它要更快。因此，在这种情况下电路本身的执行要比描述出电路所有的功能和可能的条件更快。

形式神经网络能够做任何你能用文字描述出来的事情，这个理念非常重要，在复杂度不太高的时候，这种处理方法能够非常大程度地简化问题。但是在复杂度很高的情形下，定理所描述的情况则完全有可能相反，也就是逻辑电路反而比文字形容更加简单。【冯纽曼在讨论完图灵机之后再次提到了这个问题】

【冯纽曼接着举了两个例子，说明理想化的神经元并不能解释实际的神经细胞是如何实现某一个具体功能的。在第一个例子中，一束神经元被用于传送某一个连续的数字，用来代表血压之类的身体指标。神经元并不是用二进制的方法来表示这个数字，而是以发射的频率来代表血压的高低，血压越高则发射越频繁。可以用神经疲劳来解释这个现象，一旦神经发射之后，会有一段时间的疲劳期，在其中，神经细胞不应激，而之后的刺激越强，则神经细胞离开疲劳期应激的速度就越快。接下来的问题是“为什么自然界从未使用二进制，而是使用脉冲频率的方法来表示大小呢？”，冯纽曼说他对这类问题很感兴趣。他提出了一个猜测：频率调制的表示方法比二进制要更加可靠，请参见下面的第 1.1.2.3 节，以及《计算机与人脑》的 77-79 页，还有冯纽曼的著作选集 5.306-308 以及 5.375-376。】

### 四、有关记忆

【第二个理想神经元同实际不符的例子是关于记忆的实现。冯纽曼之前用理想神经元建立过存储器电路。他说这样的内存电路可以做到任意大。但他觉得，实际上自然并不是用这样的机制来实现记忆的。】这不是自然界实现记忆的机制，原因很简单，像神经元这样的开关元件很容易疲劳，因此要实现 1bit 的信息储存，一个神经元是远远不够的，可能要十几个以上，这是一种严重的浪费，因为开关元件并不适合用作储存信息。对于人工制造的计算机，如世界第一台计算机 ENIAC 中，内存不足一直是一个制约效率的严重问题。虽然 ENIAC 由庞大的 2 万个电子真空管组成，它的内存容量却只有可怜的 20 位数字。这个尴尬局面就是因为把真空管，或者说开关元件用来干它不适合干的工作所导致的。只要把内存实现的机制加以修改，计算机器的效率就可以大大提升<sup>2</sup>。

人类神经系统的内存需求可能是非常巨大的，估计其数量级为 10 的 16 次方。关于这个

---

<sup>2</sup>译者按：现代计算机内存有两种，高速内存还是由集成电路中的微型晶体管实现，称为 SRAM；低速内存则由电容等机制实现，称为 DRAM。

估计本身是否合理，我不想多讨论，因为说不清楚。但是我相信，因为人脑神经元的数量仅为 10 的 10 次方，远不足以储存这样巨大的记忆容量，我们最好老实地承认，对于记忆究竟是什么，我们还一无所知。当然我们可以做各种各样的猜测，比如认为记忆是神经突触的某种细微变化，因此不受上述数量级的限制等等。我不知道有没有什么证据证明这种猜测，但是我想应该没有。你可能怀疑神经细胞远不仅仅是开关电路那么简单，多余的功能就包括记忆的实现，但是事实上我们什么也不知道。很可能记忆的实现是同神经元是完全不同性质的机制<sup>3</sup>。

寻找记忆元件的主要困难，是因为记忆似乎无处不在。很难在大脑中间简单地定位任何东西，因为大脑具有强大的自组织能力。即便是你找到某一个特定部位所对应的功能，在手术切除大脑的相关部位之后，很可能大脑会进行重新组织，重新分配各部分的职责，从而使这种功能因此又得到了恢复。大脑具有相当强大的灵活适应能力，因此难以给不同部位确定具体的功能，我怀疑在所有的大脑功能中间，记忆是分布得最广泛的。[参见：《计算机与人脑》63-68 页]<sup>4</sup>

## 五、图灵机

我这里想要提两件事[疲劳和内存]，在 McCulloch & Pitts 对神经系统的分析中间，明显没有提到它们，之后再谈图灵机。McCulloch & Pitts 最后得出结论，认为公理化的神经元体系同形式逻辑是等价的。图灵则得到了反向的结果，图灵感兴趣的是形式逻辑而不是自动机。他想要证明形式逻辑中的一个重要猜想，也就是所谓的可计算判定问题。这个问题是问，对于一类逻辑表达，是否存在着一套固定方法来判定这个表达式的真假。图灵之所以讨论自动机，其实是为了给他的可计算性证明提供一个铺垫，让这个形式系统中的理论显得更加清楚和前后一致。

【接下来，冯纽曼简述了图灵自动机的定义。McCulloch & Pitts 自动机的基本结构是神经元，图灵机则是由不同状态组成的，任何时候，图灵机都处在有限的状态组合中的某一状态中；“外部世界”则是由纸带所代表。每个运行周期中，自动机读入一格纸带，可能修改这一格的内容，然后把纸带向左或者向右移动一格。图灵机按照预定的规则字典运行，其中规定了对于不同的状态，读入不同的符号时候下一个状态为何，并且在这一格写入何种符号。纸带有特殊的一格，表明了开始位置，有限长度的程序可以预先写在纸带上，计算所得的二进制结果则写在开始位置之后。】

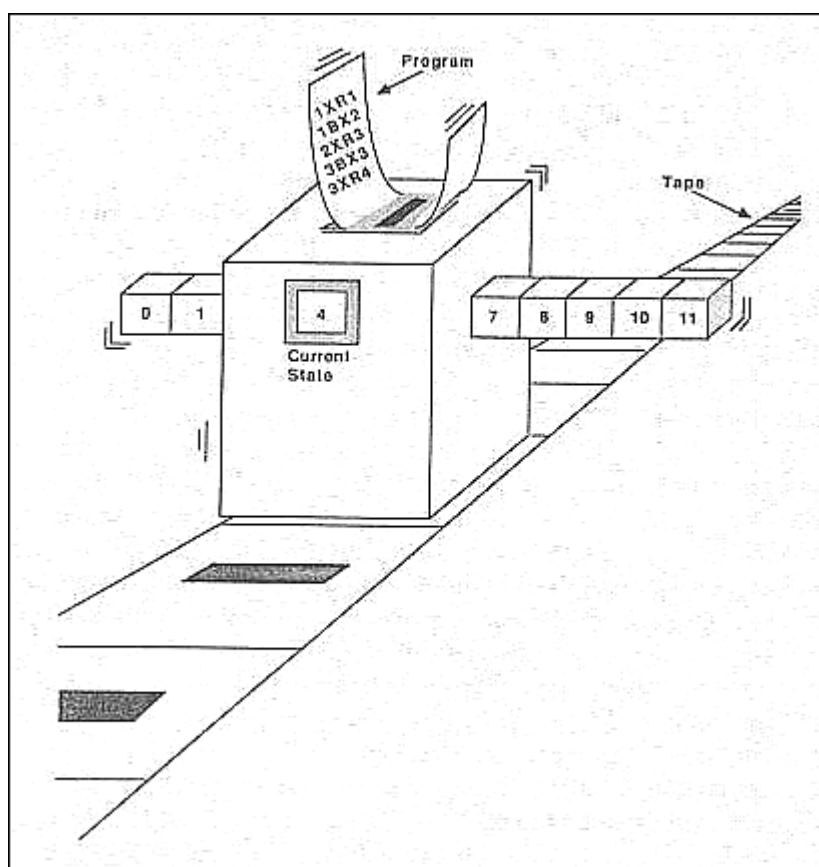
【接着冯纽曼说到了图灵关于有限自动机的结论：存在通用图灵机  $A_U$ ，具有以下性质：对于任何图灵机  $A$  和程序  $I$ ，存在模拟程序  $I_A$ ，只要提供  $I_A$  和  $I$  给  $A_U$ ， $A_U$  就可以模拟  $A$  的运算，输出和  $A$  执行  $I$  得到的相同结果。】 $A_U$  具有模拟任何图灵机去执行任何指令的能力，哪怕这个图灵机比  $A_U$  本身要复杂得多也没关系。因为通用图灵机本身的复杂性不足，可以由模拟器程序包含的内容来填补。这项研究的深刻之处在于，只要把通用图灵机恰当地制作出来，它能够完成任意复杂的工作，因为其他的各种需求可以通过编写程序来满足。但是，只有当任意的图灵机  $A$  具有最起码的复杂度，从而能够实现通用图灵机的时候，它才能有

---

<sup>3</sup>译者按：现代神经科学研究证明记忆并不局限于大脑某一部位，大脑的微观工作机制，即细胞突触形成、神经环路的建立，本质上就是一种学习和记忆过程；此外，由于一个神经元可以具有成千上万个突触，而记忆则包含于大量神经元组合之中，所以可以轻易突破正文中提到的 10 的 16 次方的内存容量限制。当然，要把某一记忆和具体神经元建立确切联系，尚有大量工作要做。

<sup>4</sup>译者按：现代神经医学和诊断科学的发展，已经可以把大脑的单个神经元活动和认知过程进行实时的比对，证明大脑的不同部位的确有特定的功能，这种功能常常可以定位到具体的神经元上。同时，大脑恢复能力并没有冯纽曼所说的那么强大，功能丧失常常是永久性的。另一方面，记忆很可能确实是一种分布式计算。

这样的能力。换句话说，没有达到通用图灵机复杂度的系统，不管你给它写什么程序，都是不可能完成某些工作的。但是一旦复杂性超越了某一个确定的阈值，只要给予适当的程序，所有的自动机都能够相互模拟了<sup>5</sup>。



图灵机模型（译者加）

图片来源：[http://blogs.warwick.ac.uk/jmiles/imagetag/turing\\_machine](http://blogs.warwick.ac.uk/jmiles/imagetag/turing_machine)

【冯纽曼跟着解释，通用图灵机  $A_U$  是怎么模拟任意一个图灵机  $A$  的。模拟程序  $I_A$  就像一本对照词典一样，包括了所有  $A$  的状态和每一个符号的组合，说明在这个情形下， $A$  的下一个状态是什么，在纸带上应该写入什么符号。 $A_U$  按照这本词典运行，就可以得到跟  $A$  一模一样的输出了。】到此，我想指出，人类第一次具有了某种通用的工具，理论上说，只要任何人能够做到的事，这种工具也能复现此过程。这里并没有无中生有，凭空制造复杂度，因为程序还是得具体的写出来。而且，实现通用计算的这一过程也是基于严格的计算理论的，亦即关于怎样描述对象以及如何在字典中间查阅条目并执行的过程。

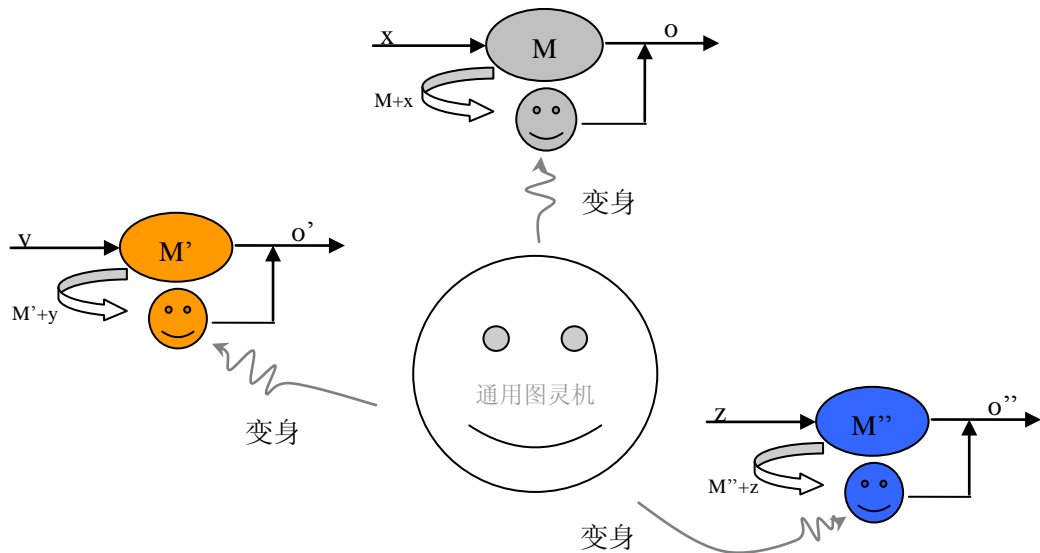
不仅如此，图灵还发现了自动机的极限。也就是说，无法制造这样一种自动机，这台自动机能够预测其他自动机在有限时间内是否能解决某个问题，完成停机。换句话说，虽然你可以制造能够模拟任何图灵机的通用图灵机，你却无法制造可以预测任何图灵机运行结果的“预测机”。做过的事情可以重复，但是没有做过的事情，却没有办法预测<sup>6</sup>。

以上结论同形式逻辑的结构本身具有深刻的联系。虽然在这课堂上不能深谈，我想说一下大概的意思，以便熟悉形式逻辑的人去研究。这一点跟类型论（theory of types）和哥德尔

<sup>5</sup>译者按：以上通用图灵机的概念非常深刻，是现代计算机的理论依据，因为只有明白了这点，才能制造可编程的计算机，而不是仅仅具有特定功能的计算器，从计算器到计算机，是对自然认识的一次飞跃。

<sup>6</sup>译者注：这里提到的“预测机”无法实现的问题实际上牵扯到了可计算理论的不可判定问题类，对图灵机行为的预测就是一种不可判定问题。关于更多的不可判定问题的讨论，请参考[这里](#)。

定理有关，你可以在一个逻辑类型中间做任何可行的事情。但是一件事情是否可行这个问题本身，却属于高一个层次的逻辑类型。我之前提过，低复杂度的东西，很容易预先判断它的行为。但是对于高复杂度的形式逻辑对象，却很难提前预测它的行为，最好的办法就是把它实际制造出来运行。在此情形下，一个问题的有效范围要比问题本身处在高一个层次的类型。



通用图灵机工作原理示意图（译者加）

通用图灵机可以被看作是一个可以变身为其它任意一台图灵机（甚至包括通用图灵机它自己）的机器。只要给它输入适当的指令，它就可以原封不动地模拟其它图灵机（ $M, M', M''$ ）的工作。我们通常用的通用可编程计算机实际上就是通用图灵机，程序员给机器输入的程序就相当于任意一台特定的图灵机（例如求解  $1+1$  的值得图灵机）的描述。

\*\*\*\*\*

## 补充材料：Arthur Burks 与 Kurt Godel 的通信

【冯纽曼的第二堂课到此结束，关于最后两段，编者想做一点补充】<sup>7</sup>

【在课程最后一段中，冯纽曼提到了哥德尔的一条定理“你可以在一个逻辑类型中间做任何可行的事情。但是一件事情是否可行这个问题本身，却属于高一个层次的类型。”因为编者不知道哥德尔曾得到过这样的定理，这段话显得十分费解。另外他在之前引用的有关哥德尔的讨论（47 页），还包括冯纽曼的希克森研讨会上的文章，“自动机的一般和逻辑的理论”（参见冯纽曼文集的 5.310-311），。所以我给哥德尔教授写信询问此事，不过我认为他给出的答案是关于参考文献最模糊的解释。因此，编者把与哥德尔的通信稍作编辑，附在下面：

我的信件如下：

<sup>7</sup>译者按：当时计算机还是“尖端科技”，其原理还显得十分神秘，故须作此介绍。而今天图灵机早已家喻户晓，我们集智也进行过多次科普（请读者参见：《[图灵机遇计算理论](#)》以及《[系统中的观察者 7：自指——连接图形和衬底的金带](#)》），图灵机介绍部分兹省略不译。下面有一段编者 Burks 教授同哥德尔本人的通讯来往，很有史料及科学价值，全文照译，请读者留意。



鄙人正在编注已故冯纽曼教授关于自动机原理的未完成手稿，在 1949 年伊利诺伊大学的以及此前的希克森会议上，他都引用了您的著作，但我却查不到出处，因为有可能冯纽曼生前曾同您谈及过相关问题，我冒昧给您致函，请求指点。

手稿的背景如下：冯纽曼当时在讨论视觉识别，他说可以把人的眼睛和神经系统看成一部确定的有限自动机。他提出如下的可能，即描述这类自动机的行为的最简方式，是把自动机的结构依样画葫芦地描述出来。这是令人信服的。但是我不能明白他接下来说的话：

“在这个范围内，有可能实际对象本身就是它自己的最简单描述。这是说，任何用文字或者形式逻辑来描述一个对象的尝试，都可能导致比对象本身更加复杂，牵涉到更多关系的结果，也就是越说越复杂。现代逻辑研究中的某些结论暗示了，在遇到高度复杂的对象的时候，这样的奇特性质反而是正常的。”这里似乎提到了您的工作，鄙人随信附上全文，敬请参阅。

在伊利诺伊大学的课上，冯纽曼也提出了类似的观点：视觉产生过程最简单的精确描述方法，就是把大脑的视觉部分神经元连接罗列出来。他接下来说，对于形式逻辑的研究表明，当自动机不太复杂的时候，对于自动机功能的描述要比自动机本身简单。但对于复杂自动机，情况却相反。他这里指名道姓提到了您的定理“我实际上正在曲解一条逻辑定理，但是这个定理本身是非常漂亮的。这是由哥德尔曾经提出的一条定理：从逻辑上说，对于一个对象的描述要比这个对象本身高一个级别。因此，渐进地说，前者总是比后者要长。我觉得为了理解这种性质，必须运用上述定理，因为在你去深入研究这件事情的时候，你已经触及到复杂度这个概念的本质。”

此后，在叙述了图灵机以及图灵对于停机问题不可判定性的论述之后，他再次回到这个问题上面。他说这些问题跟类型论和您的研究结果有着深刻联系。此处的录音记录有些乱，我尽量保留了原文“这一点跟类型论和哥德尔定理有关，你可以在一个逻辑类型中间做任何可行的事情。但是一件事情是否可行这个问题本身，却属于高一个层次的类型。我之前提过，低复杂度的东西，很容易预先判断它的行为。但是高复杂度的形式逻辑对象，却很难提前预测它的行为，最好的办法就是把它实际制造出来运行。在此情形下，一个问题的有效范围要比问题本身处在高一个层次的类型。”

鄙人能够理解一个对象的描述要比对象本身高一个类型层次。但是，除此以外，我不明白冯纽曼的意图何在。我有两个想法，但是似乎都不能够自圆其说。一是把哥德尔配数当作一个公式的描述来对待。但是，至少在某些情况下，哥德尔配数的长度是可以小于公式的长度的，否则自指的不可判定的公式就是不可能的了。另外一种可能的理解是同您 1936 年的论文《关于证明的长度》中间提出的证明长度定理有关。有两个系统： $S$  和较大的系统  $S_1$ ，您的定理说：对于每一个递归函数  $F$ ，在两个系统下都存在一个可证明的陈述使得在这两个系统下的最短的证明满足一个不等式，也就是在小系统  $S$  下的证明的哥德尔配数要比  $F$  作用在大系统  $S_1$  中间证明的哥德尔配数大。这符合冯纽曼说的结论，问题是结论正好相反：类型越高，证明越短而不是越长。

冯纽曼教授的这段叙述令我十分迷惑不解，如您能不吝赐教，鄙人不胜感激。

哥德尔的复信如下：

“我对于冯纽曼当时的意图，大致有了一些概念，但是因为我从未和他谈及这些问题，这仅仅是我个人的猜测。”

“我认为冯纽曼提到在下的定理的时候，并非是指存在不可判定的断言或者证明长度的问题。而是一个认识论上的事实，即语言  $A$  的完全描述是无法用语言  $A$  本身来给出的，这是因为在  $A$  里面，没有办法定义句子的真假。这条定理才是包含算术的形式系统不可判定性的

真正原因。但在我 1931 年的论文<sup>8</sup>中间没有提出这条定理，直到 1934 年我在普林斯顿的讲课上才正式提出。同样性质的定理也由塔斯基于 1933 年提出<sup>9</sup>”

“上述定理清楚地表明，在某些情况下，相比对于机制本身，对机制会做什么的描述要牵涉更多的东西<sup>10</sup>。在这个意义上，它需要用更加抽象的术语来表达，也就是更高的层次。但这仅仅是抽象的程度，同需要用到的符号数量没有关系。正如您所说的，符号数量和层次高低可能是成反比的。”

“但是，如果从通用图灵机角度来看，冯纽曼的想法会变得更加清楚。对于图灵机，我们可以说其行为的完全描述是无穷大的。因为既然不存在通用的停机判定方法，那么要描述一台具体图灵机的行为，只有把其所有的状态枚举出来才行。当然，为了不陷入无限大带来的困难，我们这里做了一个假定，即只有可判定的描述才是完全的描述。通用图灵机的两种复杂度之比为无穷大，同时它又可以被当作其他各种有限机制的一种极限，这立即就导出了冯纽曼的结论<sup>11</sup>。”】

---

<sup>8</sup>译者按：即哥德尔最著名的论文：《〈数学原理〉及有关系统中的形式不可判定命题》，其中提出了哥德尔定理。

<sup>9</sup>译者按：即塔斯基定理，说在任何形式系统中间，真值的概念不能够在这个系统内部定义。关于这段讨论，读者也可参看《[系统中的观察者 7：自指——连接图形和衬底的金带](#)》一文。

<sup>10</sup>译者按：比如可以十分简洁地写出某个算术表达式，但是这个断言是否为真，可能会牵涉到比这个算术表达本身多得多的内容，例如哥德巴赫猜想就远不仅仅是两个数加起来那么简单）

<sup>11</sup>译者按：通用图灵机可以模拟一切图灵机，它的行为描述是无限复杂的，但是它本身的描述却是具体的，是有限长度的，故对于图灵机，行为复杂度/机制复杂度= $\infty$ ，所有具体产生复杂度的机制，随着其复杂度的提高，最后都无限接近于图灵机。这个问题非常值得深思！

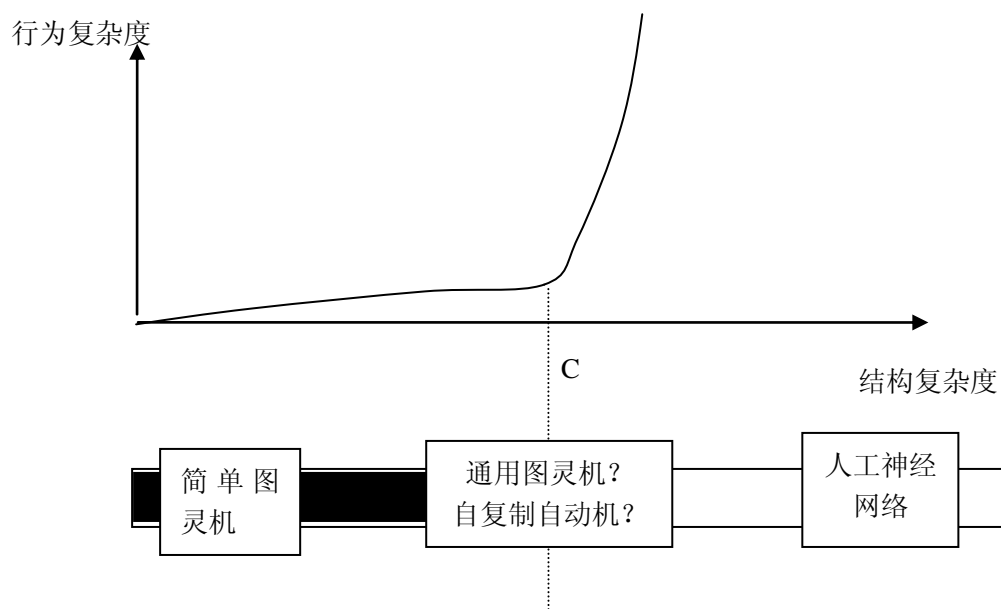
## Jake 点评

冯纽曼在这一章花了大量的篇幅介绍了 McCulloch & Pitts 的人工神经元模型以及图灵的图灵机模型。在今天看来，这两套模型都已经成为了教科书中的经典，甚至已经略显过时。但是，jake 认为，这一章最精彩的部分并不在这里，而在于冯纽曼提出来的描述层次，以及不同层次的描述复杂性的问题。包括编者 Arthur Burks 就此问题请教著名的数理逻辑学家库特·哥德尔，和哥德尔给出的回信与评论。

实际上，仅仅从字面意思上，冯纽曼在一堂课上同时介绍了 McCulloch & Pitts 的人工神经元模型以及图灵机模型就已经为此问题埋下了伏笔。我们不妨思考这样一个问题：对于一个复杂的人工神经网络，我们如何造出一台图灵机来精确地模拟这个神经网络的行为？首先，根据正文的叙述，McCulloch & Pitts 已经证明了人工神经网络等价于通用计算机，所以必然可以用图灵机来模拟。但是，这里的关键问题是：如果我们造出了这台模拟神经网络图灵机，它的复杂度（简单地可以理解为描述机器行为的代码长度）与原神经网络相比谁大谁小呢？

我们知道图灵机的运作是一种串行的方式，每一个时刻都由那个笨重的机器头——这个中心集权者进行精细的控制操作；而人工神经网络则是一种并行运作的机器，每一个时刻，决策的权力都下放到整个系统中所有的神经元，网络整体所完成的运算是所有神经元的集体行为决定的。因此，如果要把这种复杂的集体行为“压缩到”一台串行运作的图灵机的行为上面，势必会大大增加图灵机运算的复杂度。因而，冯纽曼倾向于认为，在这种情况下，对人工神经网络行为最好的模拟无非是运行这个网络本身。换句话说，神经网络行为的最短描述长度就是它自身的代码长度。

这样，我们实际上可以画出一张有关自动机复杂度的示意图：



如图所示，我们可以用一台自动机的结构复杂度（用多少代码来实现这个自动机）和它运行之后所表现出的行为复杂度两个数轴来表示自动机。对于一般的自动机来说，它的行为复杂度会随着它的结构复杂度增加而增加（不过这个结论似乎存在着反例，例如一维的元胞自动机，结构复杂度很小，但是其行为复杂度很大）。但是这种变化曲线并不是简单线性的，

而是存在着某个阈值点（C 点），这就是冯纽曼在书中反复提到的复杂度阈值。当自动机的结构复杂度超过一定的阈值之后，自动机的行为会变得异常复杂而难以预测。

下面的条状区域就是对所有自动机根据复杂度而完成的分类。在复杂度阈值 C 左侧的自动机为普通的简单图灵机（例如计算  $2+100$  的图灵机）；C 右侧的自动机为诸如 McCulloch & Pitts 神经网络的复杂自动机。而这个阈值 C 对应什么样的自动机呢？

在本章中，冯纽曼怀疑通用图灵机有可能位于这个 C 附近，因为当一个自动机的结构复杂度超过了实现通用图灵机的复杂度的时候，它就可以通过吸收不同的输入代码而“变身”成任意特定的图灵机。而在后续的章节中，冯纽曼怀疑这个 C 对应的恰恰是自复制自动机。在编者与哥德尔的通信中，以及关于冯纽曼所说的不同层次的复杂度的讨论中，我们可以明显看出，冯纽曼暗含了如下的一种猜想：尽管我们还不知道这个复杂度阈值 C 是什么，但是似乎我们可以比较有把握地断定，这个 C 值点恰恰与自指——这个构成了图灵停机问题、哥德尔定理以及程序自复制问题的核心概念有关。

总之，我认为这个不同层次的复杂性的问题是冯纽曼在本章给我们留下的一个非常宝贵的遗产，也是理解复杂性问题的核心的地带。但是由于我能力有限，还不足以回答冯纽曼这里的问题，不过我这几年来一直围绕着这个问题展开探索，在此，我愿意把了解到的相关的概念（更重要的是参考文献）列举出来，希望能够起到抛砖引玉的作用。

### 1、Kolmogorov 复杂性

冯纽曼的这本书的出版是 1966 年，就在几乎同一年，Kolmogorov 在 1965 年和 Chaitin 在 1966 年提出了“Kolmogorov Complexity”或者也叫“算法复杂性(Algorithmic Complexity)”的概念。我不知道 Kolmogorov 等人是不是了解冯纽曼关于自动机复杂度，特别是复杂度阈值的讨论，但是，它们要解决的问题似乎是同样的。

Kolmogorov 复杂度定义在所有的 01 序列上。一个序列的复杂度定义为能够生成这个序列的所有图灵机（递归函数）程序中最短的图灵机的代码长度。

这样，序列的复杂度就对应为它的可压缩性。对于一个随机序列来说，我们找不到一段程序来生成它，于是我们只能用最笨的方法：把这个序列逐字的打印出来。因此，它的 Kolmogorov 复杂性就是序列自己的长度。而对于一个有规则的序列，例如 010101……重复 200 遍，我们可以写出一个程序为：“Print(‘01’ 200 times)”，这个程序的代码长度是 21，于是该序列的 Kolmogorov 复杂度就是在  $\log(200)$  数量级上。

关于 Kolmogorov 复杂性，我推荐这本书：Li, Ming and Vitányi, Paul, *An Introduction to Kolmogorov Complexity and Its Applications*, Springer, 1997。

我们看到，Kolmogorov 复杂性已经非常接近冯纽曼要探索的自动机的复杂度的概念了。

### 2、自指与跨层次

在正文中，冯纽曼以及哥德尔曾多次提到了有关“逻辑类型”、“命题的真伪在形式系统内部无法判定”等问题。这其实都已经牵扯到了自指以及跨层次的缠结等问题（参见侯世达：《哥德尔、埃舍尔、巴赫——一集异璧之大成》，1994 商务印书馆）。而对这些问题，jake 已经做过一定的探索，并总结在了[《系统中的观察者 7：自指——连接图形和衬底的金带》](#)这篇文章中，在此不再赘述。

在这里，我想进一步指出的是，传统的有关自指与跨层次缠结的问题多针对的是逻辑结论，例如一个命题是否可以被证明，一个图灵机是否能停机，等等，这些问题都仅仅有“是或否”两种答案。就我所知，我们还没有看到有关这种自指与缠结层次的定量研究，例如产生悖论的概率、停机的概率有多大？

也没有将复杂度与自指联系起来的研究：例如如果要真正实现一个自指句子（或者自指

的图灵机)它在不同层次的复杂度(程序长度)上有何关系?我想,冯纽曼在此呼唤的有关自动机的新理论恰恰就是这类自指或者跨层次缠绕的定量理论。例如冯纽曼怀疑复杂度阈值恰恰就在自指的自动机中产生,而对于这类自动机来说,从高层次来描述它的复杂性(预测它的行为)比从低层次来描述它的复杂性(对自动机直接描述)更大。

就我所知,数学家、计算机科学家 Chaitin 的确考虑过类似的问题(参见:Chaitin: Information, Randomness & Incompleteness, World Scientific 1987; Chaitin: Information-Theoretic Incompleteness, World Scientific 1992)。读者不妨参考。

### 3、不确定性原理

此处纯粹是我对冯纽曼问题的一种猜测,没有太多的根据,请读者斟酌其中的结论。我们都知道,在量子力学中有一个著名的“不确定性原理”(Uncertainty Principle),它指出动量的波动与位置的波动满足不等式:

$$\Delta p \Delta x \geq \frac{\hbar}{2}$$

虽然这个不确定性原理是量子物理学家海森堡提出来的,但是它实际上深深植根于数学之中。从一般意义上来说,任何一个时间域上的函数  $f(t)$  和它的傅立叶变换即频率域上的函数  $g(w)$  都存在着不确定性关系,而位置和动量上的概率波恰恰构成了傅立叶变换对,所以它们才存在着不确定性原理(请读者参考 R. N. Bracewell 殷勤业(译):《傅立叶变换及其应用》,西安交通大学出版社)。

实际上,从时间域和频率域上对同一个信号的描述无非是从两种不同的视角来对信号进行描述,而不确定性原理则告诉我们,我们从原则上不可能对未知的信号同时在时间域和频率域上都给出精确的测量。

在这里,如果我们描述的对象是一个未知的序列,我们可以从序列展开本身的数值来描述,也可以从能够生成该序列的图灵机程序这个角度来描述它。

我们猜想,在这种对未知序列的不同角度的描述中也存在着不确定性原理,也就是我们不能同时提高对序列的直接描述和程序生成描述的精度。联系到描述的长度,我们知道,对序列的描述精度一般与程序的长度正相关,程序越长说明描述的越详细,精度越高。这样,我们也许可以得到类似不确定性原理的不等式:

$$L_p + L_t \geq c$$

即近似描述未知序列的图灵机程序的代码长度  $L_p$  加上从序列本身的角度进行描述的代码长度  $L_t$  的和有一个下限值,我们不能同时提高这两种代码长度值。(之所以中间变成了加号而不是乘号是因为熵与对应概率之间的对数关系)。也许,这种不确定性原理就是冯纽曼在这里讨论的不同层次复杂度之间的关系的定量刻画。也是自指性的一种定量的反应?