# 第 1 章　神奇的人工智能

制造智能机器自古以来就是人类一直追求的梦想，蕴藏在浩如烟海的民间故事、轶事和传说之中。《列子・汤问》中记载了一位名叫偃师的巧匠，他制作的机械人能歌善舞，与真人无异。一天，偃师带着它拜见周穆王，却因机器人朝向穆王的嫔妃们抛媚眼惹得穆王大怒，将它剖开后才确信是机器人，可见其惟妙惟肖。三国时期，传说蜀汉丞相诸葛亮发明了可以自动行走的木牛流马，走山路如履平地，北伐时常用以搬运军粮，立了大功。在中世纪的阿拉伯，博学者加扎利发明了众多玩偶机器人，包括送饮料的女侍者、可演奏音乐的机器人乐团等，受到时人的普遍欢迎。

随着机械制造工艺的进步和电子电气技术的发展，人类发明了很多自动化机器，如各种加工机器、电器设备等。这些设备虽然具有一定程度的自主性，但远未达到“智能”的程度。计算机的出现极大地推动了智能机器的发展，由此催生了一门全新的学科——“人工智能”。自 20 世纪 50 年代诞生以来，人工智能技术飞速发展，从最初的简单搜索和推理，到自动定理证明和专家系统，再到如今各种能听会说、能思会想的智能机器，人工智能技术已经悄无声息地走入我们的日常生活，极大地改变了社会生产方式和人们的生活方式，并为我们提供了无限的想象空间。可以预期的是，未来人工智能技术必然会在各个领域完成一个又一个的创举，成为

人类探索自然、改变自然的强大工具和亲密战友。

## 1.1 什么是人工智能

首先，让我们来看看“什么是人工智能”。要回答这一问题，需要先了解“什么是智能”。依维基百科的定义，**智能**(Intelligence)是指生物一般性的精神能力。这个能力包括：推理、理解、计划、解决问题、抽象思维、表达意念以及语言和学习的能力。可以将“智能”通俗地理解为“思考的能力”。而**人工智能**(Artificial Intelligence，AI)，就是让机器具有这种能力的科学，也就是说让机器像人一样能思会想。如果机器真的具备了这种能力，就可以称为**智能机器**(Intelligence Machine)。

从这种意义上讲，计算器是智能的，因为它可以推理出 1＋1＝2；汽车不是智能的，因为它只会在人的操纵下行走。然而，众所周知汽车不论在功能、复杂程度上都远高于计算器。因此，智能机器未必比非智能机器先进，非智能机器里也有很多智能的元素。例如，对于汽车来说，ABS 系统在汽车制动时，可自动调整制动力的大小，使车轮不被抱死，这是智能的，但却不容易被大家察觉。因此，智能更多的是功能上的概念，而非对某一设备的评价。事实上，几乎所有的机器都具有一定的智能元素，只不过有些智能元素是局部的、辅助的、不易察觉的，而有些智能元素却是全局的、主体的、表现出明显的智能性。此外，人们对智能的感知具有相对性。自动洗衣机在问世之初无疑是让人新奇的智能机器，但当它普及之后，人们就习以为常了，智能的标签就不存在了。这也说明了“智能”和“智能机器”本身的模糊性，人们总是把新颖的、具有超常能力的机器看作是智能的。值得注意的是，目前学术界和产业界并没有一个关于人工智能的确切定义。人工智能的先驱 John McCarthy(约翰·麦卡锡)在 1955 年曾给出这样的定义：“人工智能是制造智能机器的科学与工程”。维基百科的定义是：“人工智能又称机器智能，是指由人制造出来的机器所表现出来的智能。通常人工智能是指通过普通计算机程序的手段实现的类人智能技术。”虽然没有一个公认的定义，但研究者对人工智能的研究目标是明确的，即为机器赋予人的思想和行为能力。

## 1.2 人工智能简史

### 1.2.1 数理逻辑：人工智能的前期积累

人工智能的发展初期以如何刻画人类的智能行为作为研究目标，特别是对知识的表达和推理过程的形式化。换句话说，就是如何将人类的智能行为用计算机模拟出来。事实上，对人类知识结构和推理方法的研究最早可上溯到古希腊哲学

家亚里士多德(Aristotle)的三段论逻辑以及欧几里得(Euclid)的形式推理方法。13 世纪,加泰罗尼亚数学家和逻辑学家拉蒙·柳利(Raymundus Lullus)用机械手段模拟简单的逻辑操作,通过演绎运算从旧知识中推理出新知识。17 世纪,英国哲学家霍布斯(Hobbes)和数学家莱布尼茨(Leibniz)等进一步提出“推理就是计算”的思路,将逻辑变得可计算化。到了 20 世纪,在布尔(Boole)、费雷格(Frege)、希尔伯特(Hilbert)、罗素(Russell)等人的努力下,**数理逻辑**(Mathematical logic)成为一门独立的学科,标志着逻辑推理形式化的数学理论最终形成。

什么叫逻辑推理形式化呢?可以通过一个例子来简单理解。假设 $p$、$q$、$r$ 分别表示“今天下雨”“我们今天不野餐”“我们明天野餐”,那么“如果今天下雨,那么我们今天将不野餐。”可表示为 $p \rightarrow q$;“如果我们今天不野餐,那么我们明天将野餐。”可以表示为 $q \rightarrow r$。通过连续运用推理规则,即可由 $p \rightarrow q$ 和 $q \rightarrow r$ 推理出 $p \rightarrow r$。这意味着“如果今天下雨,那么则明天野餐。”在上述过程中,我们将事实表示为符号,将推理表示成符号间的蕴含关系(→),如果再加上一系列限制条件和演算规则,即可得到一套逻辑系统。在这套系统中,$p$、$q$、$r$ 是独立于事实本身的变量,因此该系统描述的不是某一个具体的推理任务,而是一类基于相同逻辑元素和统一推理规则的任务的抽象表示。基于此,推理过程被转化成符号演算,这是数理逻辑的基本思路。

数理逻辑的发展为未来的人工智能大厦奠定了第一块基石。数理逻辑的先驱们认为一切智能活动都可以转化为逻辑过程,因此逻辑过程的可计算意味着人类智能的可计算。希尔伯特(图 1-1)甚至曾经设想一个一致完备的逻辑体系,只要基本假设是合理的,就可以通过运算推导出领域内的一切知识。这个大一统的梦想

**图 1-1　大卫·希尔伯特(德国人,1862—1943 年,伟大的数学家)**

注:1900 年,希尔伯特在巴黎的国际数学家大会上提出了 23 个问题。这些问题为 20 世纪的数学研究指明了方向,被称为“希尔伯特问题”。希尔伯特的第 2 个问题为“算术公理之相容性”。在这一问题中,希尔伯特猜想一个公理系统可以一致、完备地生成所有真值命题。这一猜想在 1930 年被奥匈帝国数学家库尔特·哥德尔证明为伪。

最终被哥德尔(Godel)著名的不完备定理打破,但数理逻辑的强大描述能力已经深入人心,大大增加了人们制造智能机器的勇气。人们相信,只要逻辑系统设计得足够好,就有望将人类的智能过程通过计算完美地复现,尽管当时计算机还没有出现。

**延伸阅读:哥德尔不完备定理**

哥德尔于1930年证明,任意一个足够强大的逻辑系统都是不完备的,总有一些定理在该逻辑系统中无法被证明为真,也无法被证明为伪。哥德尔的证明类似"说谎者悖论":如果有个人说"我说的是假话",我们是无法判断这句话的真假的。如果这个人说的是真话,那么由"我说的是假话"这句话的意义可推知他(她)实际在说假话,与前提"他在说真话"相互矛盾;反之,如果这个人说的是假话,则"我说的是假话"这句话就不是真的,因此这个人事实上说的是真话,又与前提"他在说假话"相互矛盾。哥德尔证明类似的悖论在任何一个足够强的逻辑系统中都存在,因此任何一个逻辑系统总有它无法理解的命题存在。这说明任何一个系统都有其固有局限性,不同层次的系统局限性各不相同。计算机无法突破其固有局限性,因此模拟人类智能的方法有可能永远无法超过人类。

### 1.2.2 图灵:人工智能的真正创始人

1936年,年仅24岁的英国科学家图灵(Turing)在他的论文《论可计算数及其在判定问题上的应用》中提出**图灵机**(Turing Machine)模型,证明基于简单的读写操作,图灵机有能力处理非常复杂的计算,包括逻辑演算。1945年6月,美国著名数学家和物理学家约翰·冯·诺伊曼(John von Neumann)等人联名发表了著名的"101页报告",阐述了计算机设计的基本原则,即著名的**冯·诺伊曼结构**。1946年2月14日,世界上第一台计算机ENIAC在美国宾夕法尼亚大学诞生。1951年,ENIAC的发明者电气工程师约翰·莫奇利(John William Mauchly)和普雷斯波·艾克特(J. Presper Eckert)依据冯·诺伊曼结构对ENIAC进行了升级,即著名的EDVAC计算机。计算机的出现为快速逻辑演算准备好了工具,奠定了人工智能大厦的第二块基石。

在美国人设计ENIAC的同时,图灵也在曼彻斯特大学负责曼彻斯特一号的软件开发工作,并开始关注让计算机执行更多智能性的工作。例如,他主张智能机器不该只复制成人的思维过程,还应该像孩子一样成长学习,这正是机器学习的早期思路;他认为可以通过模仿动物进化的方式获得智能;他还自己编写了一个下棋程序,这可能是最早的机器博弈程序了。为了对人工智能有个明确的评价标准,图灵于1950年提出了著名的**图灵测试**(Turing Test)。在这一测试中,图灵设想将一个人和一台计算机隔离开,通过打字进行交流。如果在测试结束后,机器有30%以上的可能性骗过测试者,让他(她)误以为自己是人,则说明计算机具有智能。这一测试标

准一直延续至今，可惜还没有一台计算机可以确定无疑地通过这一看似简单的测试。图灵的这些工作使他成为人工智能当之无愧的创始人(图 1-2)。

**图 1-2　图灵和他的图灵测试**

注：测试者通过键盘和机器及真人以自然语言对话，如果机器可以骗过测试者，让测试者以为它是真人，则认为该机器具有了智能。

### 1.2.3　达特茅斯会议：AI 的开端

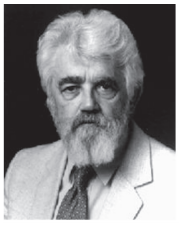
就在图灵开始他的人工智能研究不久，当时很多年轻人也开始关注这一崭新的领域，其中就包括美国达特茅斯学院数学助理教授约翰·麦卡锡(John McCarthy)、美国哈佛大学数学与神经学初级研究员马文·明斯基(Marvin Minsky)、贝尔电话实验室数学家克劳德·香农(Claude Shannon)、IBM 公司信息研究经理纳撒尼尔·罗切斯特(Nathaniel Rochester)。1956 年，这些年轻人聚会在达特茅斯学院，讨论如何让机器拥有智能，这次会议被称为“达特茅斯会议”(图 1-3 和图 1-4)。正是在这次会议上，研究者们正式提出“人工智能”这一概念，AI 从此走上历史舞台。当时讨论的研究方向包括以下几个方面：

- 可编程计算机；
- 编程语言；
- 神经网络；
- 计算复杂性；
- 自我学习；
- 抽象表示方法；
- 随机性和创见性。

可见，当时人工智能的研究非常宽泛，像编程语言、计算复杂性这些现在看来并不算 AI 的范畴也需要人工智能的学者们考虑。这是因为当时计算机刚刚诞生不久，很多事情还没有头绪，AI 研究者们不得不从基础做起。尽管如此，现代人工智能的主要研究内容在这次会议上已经基本确定了。

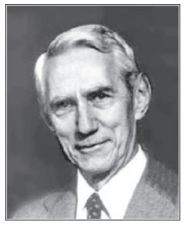
达特茅斯会议被公认为是人工智能研究的开始，会议的参加者们在接下来的

图 1-3　达特茅斯会议原址

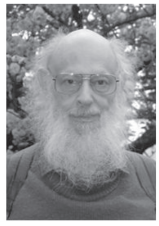


John McCarthy
约翰·麦卡锡

Marvin Minsky
马文·明斯基



Claude Shannon
克劳德·香农



Ray Solomonoff
雷·索洛莫洛夫

Allen Newell
艾伦·纽厄尔

Herbert Simon
希尔伯特·西蒙

Arthur Samuel
亚瑟·塞缪尔

Nathaniel Rochester
纳撒尼尔·罗切斯特

图 1-4　达特茅斯会议的几位参加者

数十年里都是这个方向的领军人物，完成了一次又一次的创举和突破。

### 1.2.4　人工智能的三起两落

历史总是曲折的，同时也是螺旋式前进的，人工智能的发展也是如此。我们可

以将人工智能的发展分为以下几个阶段。

**黄金十年(1956—1974 年)**　达特茅斯会议后的十年被称为黄金十年,这是人工智能的第一次高潮。当时很多人持有乐观情绪,认为经过一代人的努力,创造出与人类具有同等智能水平的机器并不是个难题。1965 年,希尔伯特·西蒙(Herbert Simon)就曾乐观预言:“二十年内,机器人将完成人能做到的一切工作。”在这近二十年里,包括 ARPA 在内的资助机构投入大笔资金支持 AI 研究,希望制造出具有通用智能的机器。这一时期的典型方法是**符号方法**(Symbolic Method),该方法基于人为定义的知识,利用符号的逻辑演算解决推理问题。**启发式搜索**(Heuristic Search)是这一时期的典型算法,这一算法通过引入问题相关的领域知识(称为启发信息)对搜索空间进行限制,从而极大地提高了符号演算的效率。这一时期的典型成果包括定理证明、基于模板的对话机器人(ELIZA、SHRDLU)等。

**AI 严冬(1974—1980 年)**　到了 20 世纪 70 年代,人们发现 AI 并不像预想的那么无所不能,只能解决比较简单的问题。这其中有计算资源和数据量的问题,也有方法论上的问题。当时的 AI 以逻辑演算为基础,试图将人的智能方式复制给机器。这种方法在处理确定性问题(如定理证明)时表现很好,但在处理包含大量不确定性的实际问题时则具有极大的局限性。一些研究者开始怀疑用逻辑演算模似智能过程的合理性。如休伯特·德莱弗斯(Hubert Dreyfus)就认为人类在解决问题时并不依赖逻辑运算[①],然而,不依赖逻辑运算的感知器模型被证明具有严重局限性[②],这使得研究者更加心灰意冷。AI 研究在整个 20 世纪 70 年代进入严冬。

**短暂回暖(1980—1987 年)**　到了 20 世纪 80 年代,人们渐渐意识到通用 AI 过于遥远,人工智能首先应该关注受限任务。这一时期发生了两件重要的事情,一是**专家系统**(Expert System)的兴起;二是**神经网络**(Neural Net)的复苏。前者通过积累大量领域知识,构造了一批可应用于特定场景下的专家系统,受到普遍欢迎;后者通过学习通用的非线性模型,可以得到更复杂的模型。这两件事事实上都脱离了传统 AI 的标准方法,从抽象的符号转向更具体的数据,从人为设计的推理规则转向基于数据的自我学习。

**二次低潮(1987—1993 年)**　20 世纪 80 年代后期到 20 世纪 90 年代初期,人们发现专家系统依然有很大的问题,知识的维护相当困难,新知识难以加入,老知识互相冲突。同时,日本雄心勃勃的“第五代计算机”也没能贡献有价值的成果。人们对 AI 的投资再次削减,AI 再次进入低谷。在这一时期,人们进一步反思传统人工智能中的符号逻辑方法,意识到推理、决策等任务也许并不是人工智能的当务之急,实现感知、移动、交互等基础能力也许是更现实、更迫切的事,而这些任务与

① Dreyfus H, Dreyfus SE, Athanasiou T. Mind over machine. Simon and Schuster. 2000.

② Minsky M, Papert S. Perceptrons: An essay in computational geometry. Cambridge, MA: MIT Press. 1969.

符号逻辑并没有必然联系。

**务实与复苏(1993—2010 年)** 经过 20 世纪 80 年代末和 20 世纪 90 年代初的反思,一大批脚踏实地的研究者脱去 AI 鲜亮的外衣,开始认真研究特定领域内特定问题的解决方法,如语音识别、图像识别、自然语言处理等。这些研究者并不在意自己是不是在做 AI,也不在意自己从事的研究与人工智能的关系。他们努力将自己的研究建立在牢固的数学模型基础上,从概率论、控制论、信息论、数值优化等各个领域汲取营养,一步步提高系统的性能。在这一过程中,研究者越来越意识到数据的重要性和统计模型的价值,**贝叶斯模型**(Bayes Model)和神经网络越来越受到重视,机器学习成为 AI 的主流方法。

**迅猛发展(2011 至今)** 人工智能再次进入大众的视野是在 2011 年。这一年苹果发布了 iPhone 4S,其中一款称为 Siri 的语音对话软件引起了公众的关注,重新燃起了人们对人工智能技术的热情。从技术上讲,这次人工智能浪潮既源于过去十年研究者在相关领域的踏实积累,同时也具有崭新的元素,特别是大数据的持续积累、以**深度神经网络**(Deep Neural Net, DNN)为代表的新一代机器学习方法的成熟,以及大规模计算集群的出现。这些新元素组合在一起,形成了聚合效应,使得一大批过去无法解决的问题得以解决,实现了真正的成熟落地。可以说,当前的人工智能技术比历史上任何一个时代都踏实和自信。

## 1.3 机器学习:现代人工智能的灵魂

从半个多世纪的发展历程可以看出,人工智能技术的进步走的是一条“反逻辑”的路。人类用一千多年的时间得到了可计算的逻辑,即数理逻辑。虽然绝大多数逻辑系统并不完备(可能存在不可证明真伪的命题),但在很多时候已经足以描述在数学和物理学上的很多知识(如概念、关系等)。这些知识是如此简洁美好,如果可以被计算机掌握,则有望实现理解、决策等智能行为,这也是最初的人工智能研究者所持有的基本思路。然而,人们在研究过程中一步步发现,人为设计的知识以及基于这些知识的推理过程在实际应用中非常困难。这不仅因为对知识进行形式化本身就很烦琐,即使完成了这一形式化,依然会有各种冲突和不确定性存在,使得推理很难完成。相反,从数据中学习得到的知识虽然可能是不精确、不全面的,但在很多时候更适合实际应用。因此,人工智能的研究者们不得不用数据学习逐渐取代人为设计。在这一过程中,我们失去了传统数理逻辑的简洁和清晰,越来越依赖从数据中得到统计规律,而这些规律天然具有模糊性和近似性。

这意味着当前人工智能技术与传统 AI 在方法论上已经有很大的不同了。当代人工智能的本质是让机器从数据中学习知识,而不再是对人类知识的复制,这一方法称为“机器学习”。基于这样的思路,人工智能已经不再是人的附庸,它将和人

类在平等的起跑线上汲取和总结知识，因而可能创造出比人类更巧妙的方法、生成比人类更高效的决策、探索人类从未发现过的知识空间。数据越丰富，计算能力越强，这种学习方法带来的效果越好，超越人的可能性越高。当前 AI 的很多成就很大程度是由庞大的数据资源和计算资源支撑的，典型的领域包括语音识别、图像识别、自然语言处理、生物信息处理等。21 世纪的 AI 是数据的 AI，是机器学习的 AI，“人工智能”里的“人工”更多的是设计学习原则，而非设计智能过程本身。基于此，本书将重点介绍基于机器学习的现代 AI 技术。关于传统 AI 方法，读者可参考朱福喜老师编著的《人工智能基础教程》一书[①]。

### 1.3.1　什么是机器学习

1959 年，亚瑟·塞缪尔（Arthur Samule）发表了一篇名为 *Some Studies in Machine Learning Using the Game of Checkers* 的文章。该文章描述了一种会学习的西洋棋计算机程序，只需告诉该程序游戏规则和一些常用知识，经过 8～10 小时的学习后，即可学到足以战胜程序作者的棋艺。这款西洋棋游戏是世界上第一个会自主学习的计算机程序，宣告了机器学习的诞生。

什么是机器学习？塞缪尔认为机器学习是“让计算机拥有自主学习的能力，而无须对其进行事无巨细的编程”的方法[②]。尼尔斯·约翰·尼尔森（Nils J. Nilsson）则认为机器学习是“机器在结构、程序、数据等方面发生了基于外部信息的某种改变，而这种改变可以提高该机器在未来工作中的预期性能”[③]。上述这些定义本质上是一致的，即认为机器学习是通过接收外界信息（包括观察样例、外来监督、交互反馈等），获得一系列知识、规则、方法和技能的过程。和传统算法相比，机器学习的一个巨大优势在于程序设计者不必定义具体的流程，只需告诉机器一些通用知识，定义一个足够灵活的学习结构，机器即可通过观察和体验积累实际经验，对所定义的学习结构进行调整、改进，从而获得面向特定任务的处理能力。

### 1.3.2　机器学习发展史

图 1-5 给出了机器学习发展历史上的一些重要人物和标志性事件。总体来说，20 世纪 50 年代以前是技术积累阶段，研究者在统计学习和优化方法上提出了一系列模型和准则。1950 年图灵提出图灵测试准则，开创了人工智能的广阔领域。机器学习伴随着人工智能的研究开始萌芽。1959 年亚瑟·塞缪尔的划时代

① 朱福喜，朱三元，伍春香. 人工智能基础教程[M]. 北京：清华大学出版社，2006.

② Samuel AL. Some studies in machine learning using the game of checkers. IBM Journal of Research and Development，1959. 3(3)：210-229.

③ Nilsson NJ. Introduction to machine learning. URL http://ai.stanford.edu/nilsson/mlbook.html，lecture notes，1998.

论文将“机器学习”这一重要概念引入人工智能，并开始独立解决实际问题。整个20世纪60年代，以符号逻辑为研究对象的**符号学派**(Symbolism)是人工智能研究的主流，人工神经网络、概率模型、遗传算法等更侧重“学习”的方法开始萌芽。进入20世纪70年代，人工智能的冬天来临，机器学习研究也走入困境，特别是在马文·明斯基发表《感知器》一书后，被寄予厚望的人工神经网络的研究几乎停滞。

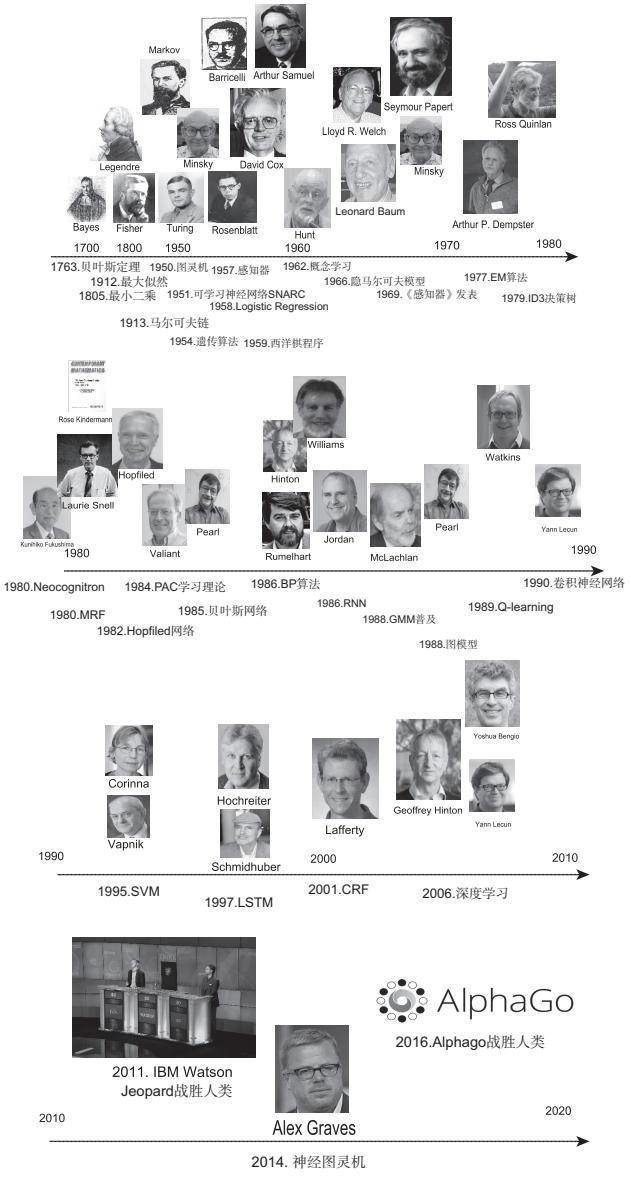



**图 1-5　机器学习发展史上的若干重要人物和重要事件**

注：坐标轴上方为人物，下方为对应人物的对应事件。

这一状态一直持续到 20 世纪 80 年代。因为基于符号逻辑的人工智能方法无法提供足够的学习空间，一些学者开始转向统计学习方法，形成了两个主要研究方向：一是基于概率模型的**贝叶斯学派**（Bayesianism）；二是基于神经网络模型的**连接学派**（Connectionism）。贝叶斯学派的代表人物包括 Judea Pearl，S. L. Lauritzen，D. J. Spiegelhalter 等，连接学派的代表人物包括 John Hopfield，David Rumelhart，Geoffrey Hinton 等。

这两个学派在基本思路上有很大的差异，但都认为机器学习（包括人工智能）应该有更灵活的学习框架，而非在人为定义的符号系统中小修小改。整个 20 世纪 80 年代，机器学习的研究者们在人工智能领域的边缘默默积累，贝叶斯学派提出了图模型，连接学派发展出卷积神经网络、递归神经网络等新型网络结构和高效的反向传播（Back Propagation，BP）训练算法。

进入 20 世纪 90 年代以后，以符号演算为基础的传统人工智能方法越来越表现出其局限性。第一，随着任务越来越复杂，对知识的定义越来越困难，不仅知识数量越来越多，不同知识之间还经常出现矛盾；第二，知识系统越复杂，新知识的加入越困难，产生的结果越难以估计；第三，对一些没有先验知识的领域，推理系统无法工作；第四，人为创建的知识在面对实际问题时经常会产生偏差，甚至会带来严重错误。相比之下，以统计方法为基础的机器学习方法可以通过灵活的结构从数据中学习知识，可以方便处理数据中的噪声和矛盾。基于此，以统计学习为特征的机器学习方法成为人工智能领域的主流方法。

进入 21 世纪以后，计算机的性能比以前有了大幅提高，这为以统计学习为特征的机器学习方法提供了更加广阔的发展空间。今天，机器学习在信号处理、自然语言理解、图像处理、生物与医学等各方面取得了前所未有的成功。如今，当我们谈论人工智能的时候，很多时候谈论的是机器学习。另一方面，互联网积累了大量人为编辑的数据（如维基百科），这些数据的出现一定程度上解决了传统符号方法在知识积累上的瓶颈，使得以**知识图谱**（Knowledge Graph）为代表的新一代符号方法取得了长足的进步。有意思的是，新生代的符号主义研究者们开始主动拥抱机器学习，利用机器学习方法对知识进行抽象与推理。新符号主义是机器学习领域中的重要力量。关于机器学习和人工智能的发展历史，有兴趣的读者可参考最近出版的一些科普著作[①]。

### 1.3.3　机器学习的基本框架

研究者对机器学习有各种各样的表述。本书中，我们将从“知识”和“经验”两个概念来理解机器学习。所谓知识，是人类已经获得的可形式化的某种理性表达

① 张江. 科学的极致：漫谈人工智能[M]. 北京：人民邮电出版社，2015.

(如英语语法和数学公式等),这些知识也被称为**先验知识**(Prior Knowledge,即已经掌握的知识)。所谓经验,是指机器在运行环境中得到的反馈(比如,我们知道沸水是不能喝的,因为有过一次被烫伤的经历,由此总结出了一条“不能喝沸水”的经验)。经验中包含大量有用的信息,只是掩盖在复杂的表象之下,很难被直接利用。

“知识”和“经验”是机器学习系统的两个基本信息来源,基于其中任何一种信息源都可以构造一个有效的智能系统。但是,基于单一信息源的系统存在明显缺陷:纯粹基于知识的系统封闭而不思进取;纯粹基于经验的系统博闻而不求甚解。一种很自然的想法是将两者结合起来。这类似一个新生儿,从诞生的那一刻起父母通过遗传给他一个合理的神经结构(可以认为是一个基于知识的“设计”),可以进行呼吸、哭闹等基本动作,但更高级的能力(如语言、推理等)则需要通过后天学习,从经验中进行总结。因此,人类本身就是一个既有先验知识,也有后天学习的综合系统。我们认为这种先验知识和后天经验学习相结合的能力获取方式是现代机器学习乃至人工智能的基本特征之一,而如何平衡这两者的关系产生了风格迥异的学习方法。图 1-6 给出基于知识—经验的机器学习框架。下面从学习目标、学习结构、训练数据、学习方法四个方面展开讨论。

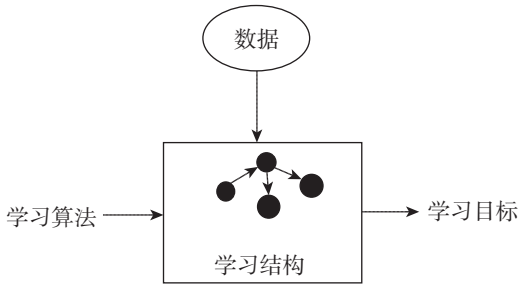



**图 1-6　基于知识—经验的机器学习基础框架**

注:首先确定学习目标,之后基于先验知识设计学习结构。参考该学习目标和学习结构,选择合适的学习方法,利用数据对学习结构进行修正,使之能更好地完成目标任务。

**学习目标**:机器学习任务的目标是多种多样的。从应用角度看,学习目标可分为**感知**(Perception)、**推理**(Induction)、**生成**(Generation)等。其中,感知包括听声、看画等;推理包括寻找原因,作出决策等;生成包括生成语音、图片、文字等。从任务性质看,学习目标可分为**预测**(Prediction)和**描述**(Description)两类,前者是指给定一部分数据(如昨天的股市指数)对另一部分数据进行预测(如今天的股市指数),后者是指对数据的内在规律进行发现和刻画(如股市指数在一段时间内的变化规律)。

**学习结构**:学习结构又称**模型**,定义了用以表达系统知识的具体形式。**函数**(Function)是一种常见的模型,该模型将知识表达为由某一输入到某一输出的映射,学习时通过改变函数参数来吸收从数据中得到的新知识;**图**和**网络**(Network)

是另一种常见的模型，该模型将知识表达为图或网络中节点的属性以及节点之间的联系，学习时通过改变这些属性和联系来吸收从数据中得到的新知识。

**训练数据**：数据是经验的累积，利用数据对系统进行学习可以更新先验知识、提高系统实用性。数据的质量、数量和对实际场景的覆盖程度都会直接影响学习的结果，因此数据积累是机器学习研究的基础，“数据是最宝贵的财富”已经成为机器学习从业者的共识。

在收集和整理数据时，通常会关注数据是否准确、是否完整，不同数据间的相关性如何。另外，我们一般不会直接使用原始数据，而是通过一系列预处理过程对数据进行清洗过滤，并将数据中最显著的部分提取出来（称为**特征提取**）进行学习。

**学习方法**：学习方法是学习过程的具体实现，即通常所说的**算法**。机器学习算法可分为**有监督学习**（Supervised Learning）、**无监督学习**（Unsupervised Learning）、**半监督学习**（Semi-Supervised Learning）和**强化学习**（Reinforcement Learning）四种。其中，监督学习需要人为对数据进行标注（如给猫的图片标上“猫”，给影评标上正面或负面评价等）；无监督学习不需要标注；半监督学习需要部分标注；而强化学习只需要间接标注（见第 5 章）。需要特别注意的是，算法的选择是由学习结构、学习目标及数据特性等几方面因素共同决定的，不存在一种普适算法在所有任务中全面胜出。

总之，我们认为机器学习是一种将人类先验知识和后天经验相结合，以提高计算机处理某种特定任务能力的计算框架。这一框架包括学习目标、学习结构、训练数据和学习算法四个部分。基于这一框架，我们依赖先验知识设计合理的学习结构，设计相应的学习算法，从经验数据中得到知识并对现有学习结构进行更新，使得既定的学习目标得到优化。

## 1.4　让人惊讶的“学习”

2011 年以来，以深度学习为代表的机器学习技术突飞猛进，发展速度超出了很多人的想象。下面来看几个有趣的例子。

### 1.4.1　从猴子摘香蕉到星际大战

人工智能的一个经典问题是：如图 1-7 所示，在一个房间内有一只猴子、一个箱子和一束香蕉。香蕉挂在天花板下方，但猴子的高度不足以碰到它。那么这只猴子怎样才能摘到香蕉呢？传统符号方法会定义若干命题及推理规则，这些命题和规则代表猴子能进行的所有操作（如朝前后左右移动、搬动箱子、爬上箱子等），以及每个操作在特定状态下产生的结果（如搬动箱子后可以爬上箱子）。通过启发式搜索算法，如果可以找到一条从“猴子刚进屋”到“猴子吃到香蕉”的路径，即可找

到让猴子吃到香蕉所需要的动作序列。

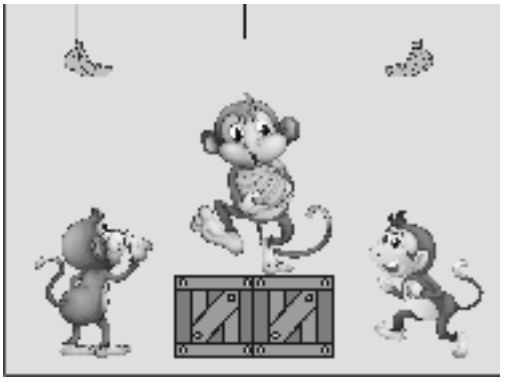


**图 1-7　猴子摘香蕉：人工智能经典问题**

上述符号方法的缺点很明显：当场景稍微改变一些，原有系统就需要作非常复杂的重新设计。如香蕉晃动、多挂几串、地板上有个坑、猴子左手使不上力、多了几只猴子等，这些复杂性使得传统符号方法很难实现。为此，研究者考虑利用现代机器学习方法来解决这一问题：不是试图建立所有规则，而是让猴子不断尝试各种方法去获得香蕉，每向正确的方向前进一步都给猴子一定鼓励，这样猴子就可以摆脱人为规则的束缚，在尝试中学会在各种场合下摘到香蕉的技能。

一个典型的例子是 DeepMind 基于深度神经网络和强化学习教会机器打电子游戏[①]。这一任务和摘香蕉类似，游戏中每作出一个正确动作就给机器一定奖励。经过大量尝试以后，机器从对游戏一无所知成长为游戏高手，甚至超过了绝大多数人类玩家。图 1-8 是机器人操作游戏杆玩外星入侵者游戏的视频截图。

### 1.4.2　集体学习的机器人

如果把摘香蕉的猴子看作机器人，处理摘香蕉这个任务的过程就是观察—计划—执行，这显然和人类处理问题的方式有所不同。我们一般会在执行过程中依据当前的行为结果不断进行重新规划，直到任务完成。比如将猴子摘香蕉变成让猴子穿针，针在风的吹动下不断摆动，这个复杂的任务别说猴子，连人都无法在任务初期就形成一个完整的行动计划。因此，我们首先会确定一个近期目标，如走近悬针的位置，再调整目标，将线接近针孔，最后将目标调整为将线送入针孔中。在这一过程中，所有近期目标的达成都会面临很多的不确定性（如风把针吹走），当不确定事件发生时，我们会即时调整策略，确保最终目标能够

① Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, et al. Human-level control through deep reinforcement learning. Nature. 2015,518(7540): 529-533.

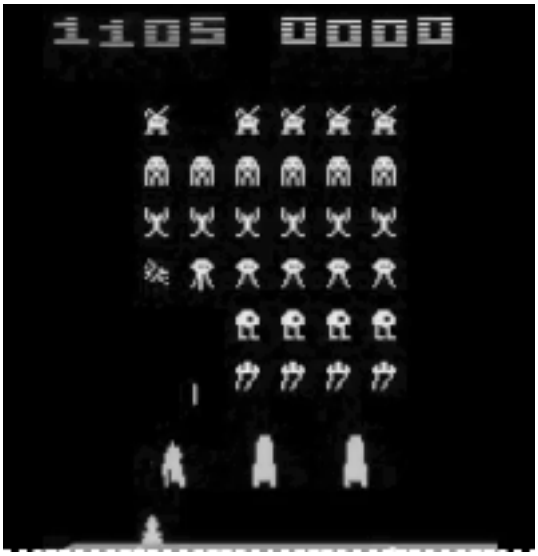


**图 1-8　机器人操作游戏杆玩太空入侵者游戏**

注：图片来自 DeepMind 视频。

实现。

近年来，研究者试图让机器具有类似的能力，并取得了突破性进展。首先，研究者发现利用复杂的神经网络可以有效提取环境信息，包括视觉、听觉、触觉、红外、超声等[①②③]；其次，利用强化学习方法（见第 5 章），机器可以在不断尝试中学会完成复杂任务的技巧，极大地提高了应对复杂场景的能力[④⑤⑥⑦]；最后，群体学习

① Koehn P, Och FJ, Marcu D. Statistical phrase-based translation. In: Proceedings of Association for Computational Linguistics (NAACL), Association for Computational Linguistics, 2003:48-54.

②④ Zhang M, Geng X, Bruce J, Caluwaerts K, Vespignani M, SunSpiral V, Abbeel P, Levine S. Deep reinforcement learning for tensegrity robot locomotion. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). 2017.

③ Yahya A, Li A, Kalakrishnan M, Chebotar Y, Levine S. Collective robot reinforcement learning with distributed asynchronous guided policy search. arXiv preprint arXiv: 161000673. 2017: 79-86.

⑤ Chu P, Vu H, Yeo D, Lee B, Um K, Cho K. Robot reinforcement learning for automatically avoiding a dynamic obstacle in a virtual environment. In: Advanced Multimedia and Ubiquitous Engineering, Springer, 2015: 157-164.

⑥ Kober J, Peters J. Reinforcement learning in robotics: A survey. In: Reinforcement Learning, Springer, 2012: 579-610.

⑦ Gu S, Holly E, Lillicrap T, Levine S. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In: Proceedings 2017 IEEE International Conference on Robotics and Au-tomation (ICRA), IEEE, Piscataway, NJ, USA. 2017.

方法使得多台机器可以共享学习成果，一台机器学会了，其他机器马上得到同样的知识[①②]。这些技术为人工智能领域带来了深刻的变革。首先，复杂神经网络相当于给机器装上了灵敏的感觉器官，可从原始感知信号中抽取有价值的信息；第二，不必为机器设计复杂的推理系统，只需给它提供足够的经验数据，机器就可以自己学习如何完成任务的技能；第三，当机器可以群体学习的时候，学习速度将大幅提高，远远超过人类的进化速度。这意味着在不远的将来，很多复杂的任务在机器面前可能变得不再困难。

图 1-9 是谷歌公司发布的一个机器人群体学习系统，其中一群机器人正在努力学习从盘中抓取物体的本领。每个机器人的手臂类似一个钳子，可以放下和收紧。这群机器人开始对如何完成抓取任务一无所知，有的只是一个摄像头和抓住物体后的奖励信号。谷歌的研究者们在两个月的时间里用 14 台机器收集了 80 万次随机抓举尝试，并用这些数据训练深度神经网络。经过训练后，这些机器人学会了如何在盘子中找到物体并将它抓起来的技巧，而且一旦某一个机器人学会了一种抓取方法，它立即通过网络通知其他机器人，使得学习速度成倍提高。

**图 1-9　机器人群体学习**

注：若干机器人协同学习，从随机状态开始，经过多次尝试后可通过学习得到抓取物体的能力[③]。

### 1.4.3　图片和文字理解

机器学习另一个有趣的例子是如何理解一幅图片的内容，并用自然语言描述

①③ Levine S, Pastor P, Krizhevsky A, Ibarz J, Quillen D. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. The International Journal of Robotics Research, 2018, 37(4-5): 421-436.

② Yahya A, Li A, Kalakrishnan M, Chebotar Y, Levine S. Collective robot reinforcement learning with distributed asynchronous guided policy search. arXiv preprint arXiv: 161000673. 2017: 79-86.

出来。传统图像处理方法需要对图片中的物体进行检测和识别，提取出图片中包含的主要对象。进而，考虑各个对象的属性、不同对象之间的位置关系、对象组合之后形成的整体效果，通过一些确定好的模板即可生成对这些对象的描述。但是，图片内容检测和对象识别本身就是很困难的事，即便完成了这一检测和识别，将这些对象所描述的事实用自然语言表达出来也是件非常困难的事。

2015 年，科学家们提出了一种基于神经网络的端对端学习方法[①]。这一方法的思路是以互联网上大量带标签或评语的图片为训练数据，学习图片和这些标签、评语之间的对应关系。系统一开始对这些对应关系一无所知，但经过大量学习，机器即可找出图片内容和单词之间的内在联系，并将这些单词连贯成自然语句，表现出来的效果就如同“理解”了这幅图片。图 1-10 给出了两个例子，其中，上面一幅图被机器理解成“一个女人正在公园里扔飞盘”，下面一幅图被机器理解成“一个小女孩抱着只泰迪熊坐在床上”。

A woman is throwing a frisbee in a park.

A little girl is sitting on a bed with a teddy bear.

**图 1-10　基于神经网络模型的图片理解**

注：左边两幅图是原图；右边两幅图表示带下划线的单词（frisbee 和 girl）所对应的图片内容[②]。

### 1.4.4　Alpha Go

2016 年人工智能界发生了一件令人瞩目的大事：DeepMind 的 AlphaGo 围棋机器人战胜了韩国棋手李世石九段。这距离它战胜欧洲围棋冠军华裔法籍棋士

①② Cho K, Courville A, Bengio Y. Describing multimedia content using attention-based encoder-decoder networks. IEEE Transactions on Multimedia, 2015, 17(11): 1875-1886.

樊麾二段仅过去半年时间。[①]

机器在人机对弈中战胜人类已经不是新闻，最典型的莫过于 IBM 的深蓝于 1997 年战胜当时的世界国际象棋冠军卡斯帕罗夫，成为首个在标准比赛时限内击败国际象棋世界冠军的计算机系统。那次胜利被认为是人工智能领域的重要成就。依靠强大的计算能力和内存资源，深蓝可以搜索估计 12 步之后的棋局，而一名人类象棋高手最多可估计约 10 步。深蓝的基本算法是启发式搜索。[②]

比起国际象棋，围棋的搜索空间要大得多，启发式搜索很难奏效，因此对围棋的处理要复杂得多。对于人类棋手，人们往往将处理这种复杂性的能力归结为一种灵性。围棋经典著作《棋经十三篇》中称之为“势”。围棋高手们往往把对“势”的把握看作棋力的象征。通过“势”与“利”的高超运用，顶尖高手们谋划、布局、引诱、潜伏、外攻、内陷、假弃、长取，纵横捭阖、经天纬地。正因为如此，围棋经常被神秘化，与攻伐、理政、怡情、处世等高级智慧联系起来。因此，在 AlphaGo 之前，很多围棋界人士都断言机器永远不可能战胜人类顶尖棋手。

然而事实却有趣得多：当 DeepMind 的研究者利用神经网络将棋局映射到一个连续的状态空间后，他们发现在这个空间里判断盘面的价值会非常容易。这一连续空间就如同人类棋手脑海里的感觉空间——看到一个盘面，在这个空间里自然形成了优劣强弱的判断，只不过人类是通过长期训练得到的一种直觉，而机器是纯粹计算出来的。基于这一感觉空间，机器学会了人类的灵性，并借此击败了人类。在后续的改进版本 AlphaGo-Zero 中，机器甚至抛开了对人类的学习，仅通过自我对弈即学习到了强大的棋力。这说明机器不但可以学习人类的灵性，而且可以创造自己的灵性，从而摆脱人类固有经验的束缚，实现独立的机器智能。

### 1.4.5 机器智能会超过人类智能吗

2016 年以来，人工智能成为公众关注的热点，特别是 AlphaGo 以压倒性优势战胜人类以后，很多人（包括一些业界领袖）都在思考一个问题：机器未来会超过人吗？这是个见仁见智的问题。我们认为，机器学习技术虽然在近年取得了长足进步，但距离成熟还有相当长的路要走。另一方面，机器在众多任务上一项一项超过人类并不奇怪，人类的历史就是一部被机器超越的历史，从汽车到飞机，从计算器到 GPU。当前以机器学习为代表的人工智能技术飞速发展，人们用大量真实数据、更强大的计算资源去训练更复杂的模型，完成以前无法想象的任务；人们研究迁移学习、协同学习和群体学习等各种知识继承方法，让机器具有类人的适应能

① https://zh.wikipedia.org/wiki/AlphaGo.

② https://www.research.ibm.com/deepblue/meet/html/d.3.2.html.

力；人们甚至开始研究如何让机器具有自主创造力、目标驱动力、情感和艺术[①]。如果考虑到快速增长的数据量、强大的分布式计算资源、开放的知识共享模式，可以预期机器将获得越来越强大的能力。基于此，有理由相信未来总会有一天机器会在绝大多数任务上超过人类，至少是绝大部分人类。然而，像所有工具一样，再强大的机器也是为人类服务的，只要保证机器的控制权握在理性人手中，并提前预知风险，机器就不会成为人类的敌人。

## 1.5　开始你的机器学习之旅

机器学习在很大程度上是一种权衡（trade-off）的艺术，没有一种机器学习的方法一定优于另一种，一种算法在获得某种优势的同时也将受限于某种劣势。设计一个好的机器学习系统需要对各种因素通盘考虑，结合任务需求和数据特性，选择合适的机器学习方法。

### 1.5.1　训练、验证与测试

我们从一个最简单的机器学习任务开始。要完成这一学习任务，我们将实验过程分为**训练（Training）**和**测试（Testing）**两个阶段。训练相当于我们平时在课堂上学知识（机器学习是学习模型）；测试相当于我们的期末考试，用来测试训练过程是否取得了足够好的学习效果。

- 训练：给定一个包含若干样本的**训练集**（Training Set），对模型进行参数调整，使得该模型在训练集上的性能越来越好。
- 测试：将训练完成的模型在一个独立的**测试集**（Test Set）上进行测试，通过在该测试集上获得的性能来判断模型的好坏。

这里有一个问题：为什么模型性能要在一个独立的测试集上验证，而不是在训练集上？这是因为在很多情况下，经过反复训练可以让模型对训练数据有充分的代表性，因此在该数据集上表现出良好性能，但对不包含在训练集中的数据性能反而会越来越差。这有些像我们平时读书备考，如果只是将课本背得滚瓜烂熟，却不会举一反三，那么考试时肯定得不到好成绩，因为考试肯定会出书本上没有的问题，过于沉溺书本就失去了对新问题的解决能力。机器学习也是如此，训练过度后，模型对训练数据描述得过细，以至于失去了代表新数据的能力，这种现象称为**过拟合**（Over-Fitting）。相反，如果对训练数据的学习达不到要求，得到的模型在其他数据集上的性能也不会好，这种现象称为**欠拟合**

① Ushveridze A. Can turing machine be curious about its turing test results? three informal lectures on physics of intelligence. arXiv preprint arXiv: 160608109. 2016.

(Under-Fitting)。这类似于一个学生连课本上的知识点都没有掌握,在考试中肯定也会一败涂地。

如何防止过拟合呢?一个简单的方法是在训练时用测试集检验模型的性能,当模型性能在测试集上开始下降的时候,即认为出现了过拟合,此时停止训练会得到一个在测试集上性能最好的模型。但这一方法在训练时用到了测试数据信息,得到的模型对测试集产生了依赖。为防止这一问题,通常会单独设计一个**验证集**(Validation Set),基于验证集进行模型选择,选择出的模型在测试集上进行测试,将该测试结果作为模型性能的评价。

### 1.5.2 Occam 剃刀准则

一般来说,越复杂的模型含有的参数越多,越容易对训练数据描述过细,产生过拟合。然而,过于简单的模型又不具有较好的描述能力,无法学到足够的知识。因此,选择合适的模型复杂度对解决实际问题特别重要。一般遵循的准则是:"在保证足够描述能力的前提下尽量选择最简单的模型",这一准则称为 **Occam 剃刀准则**(Occam's Razor)。

### 1.5.3 没有免费的午餐

机器学习中有那么多模型,有没有一种模型完胜其他模型呢?答案是没有。所谓模型好坏都是相对特定任务、特定场景、特定数据而言的。如果一个模型在某一场景、某一数据下具有某种优势,则在其他场景、其他数据下必然具有相应的劣势,这一原则称为 **No Free Lunch** 原则,即常说的"天下没有免费的午餐"。这一原则是机器学习实践中的基本准则[①②],它告诉我们对具体任务要具体分析,选择与任务相匹配的模型,才能得到较好的效果。这也提示我们要学习每种模型背后的基础假设和适用条件,唯其如此,才能对不同任务设计出合理的模型结构和合理的学习方法。

### 1.5.4 对初学者的几点建议

常有初学者问这样的问题:机器学习难吗?答案应该是"Yes or NO"。一方面,机器学习确实很难,有那么多的算法、理论、公式,发展又如此迅速,新方法层出不穷,让人无所适从。另一方面,如果理解了各种算法的发展脉络和内在联系,就会发现绝大部分算法都是顺延着某一主线一脉相承下来的,不同算法之间都有或

① Wolpert DH. The lack of a priori distinctions between learning algorithms. Neural Computation. 1996, 8(7): 1341-1390.

② Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation. 1997,1(1): 67-82.

多或少的关联，如果将这些脉络和关联理清楚，掌握机器学习并不是很难的事情。

特别要注意的是，机器学习是一门科学。既然是科学，就有自己的理论体系和思维方式。对于初学者，应该尽可能理解每种方法的基本思想和基本原理。这一点对年轻人尤为重要：现在有很多开源工具可用，很容易养成拿来主义、不求甚解的坏习惯，这对从事这方面的研究是非常有害的。另一方面，机器学习又是一门实践性很强的科学，理论联系实践非常重要。初学者应多动手实践，在实践中提高自己分析问题和解决问题的能力。最后，要认识到机器学习本身是有局限性的，还有很多问题需要解决。谦虚谨慎地学习，在实践中积累经验，是初学者应有的态度。

## 1.6　AIDemo 示例系统

本书配套的所有资源都可以从网址 http://aibook.cslt.org 下载，这些资源既包括一些基础阅读材料，也包括一些可动手操作的示例系统，以下称为 AIDemo 系统。AIDemo 中的所有示例都基于 Linux 操作系统，因此需要一些 Linux 的基础知识；另外，这些示例绝大部分是用 Python 语言编写的，如果希望对这些程序进行较细致的学习，需要对 Python 有初步的了解。访问本书的官方网址，可以学习关于 Linux 和 Python 的基础知识。

为了方便读者搭建 AIDemo 示例系统，我们将所有示例程序及其运行环境打包成一个虚拟机，只要安装这一虚拟机，即可体验这些示例，免去复杂的环境配置过程。我们选择 VirtualBox 虚拟机软件，该软件可免费下载安装，可移植性较好。安装 AIDemo 虚拟机需要主机至少有 4GB 内存，最好有 8GB。AIDemo 中的示例程序需要主机有网络环境，有声音输入输出设备。本节首先介绍 VirtualBox 和 AIDemo 的安装过程，之后以一个简单的人脸检测系统为例介绍 AIDemo 中示例程序的运行方法。

### 1.6.1　AIDemo 环境搭建

(1) 在浏览器中输入网址 https://www.virtualbox.org/wiki/Downloads.

(2) 选择 5.2 版本，并选择合适的宿主机类型。对 Windows 用户，应选择 Windows Hosts。下载安装包，进行默认安装即可。

(3) 安装完成后，通过以下网址下载虚拟机映像(.ova 文件)：

http://aibook.cslt.org/aidemo/ova.html.

(4) 在 VirtualBox 的“管理”菜单中选择“导入虚拟电脑”，在弹出的窗口中选择前一步所下载的 .ova 文件，完成 AIDemo 虚拟机安装。AIDemo 安装完成后的 VirtualBox 界面如图 1-11 所示。

图 1-11　安装完成 AIDemo 后的 VirtualBox 界面

(5) AIDemo 虚拟机安装完成后，可双击该虚拟机图标启动系统，根据 AIDemo 主页(http://aibook.cslt.org/aidemo/demo.html)的提示索取密码，登录后即可开始体验 AIDemo 中的示例程序。

(6) 我们会随时更新 AIDemo 系统，以方便读者体验最新的人工智能技术。请参考 AIDemo 主页上的提示信息进行示例程序的远程更新。

### 1.6.2　AIDemo 示例基础

如果 AIDemo 虚拟机已经安装完成，且已经掌握了 Linux 和 Python 的基础知识，就可以开始体验一个简单的人工智能系统了。

首先以用户 *tutorial* 的身份登录 AIDemo 虚拟机，可以看到桌面上有一个主文件夹，鼠标双击该文件夹进入 aibook→demo，即可看到若干文件夹，如图 1-12 所示。这些文件夹的内容如下。

- data：存储 AIDemo 系统所需的数据资源；
- env：存储 AIDemo 系统所使用的 Python 运行环境；
- image：图像处理示例程序；
- speech：语音处理示例程序；
- lang：自然语言处理示例程序；
- robot：机器人示例程序；

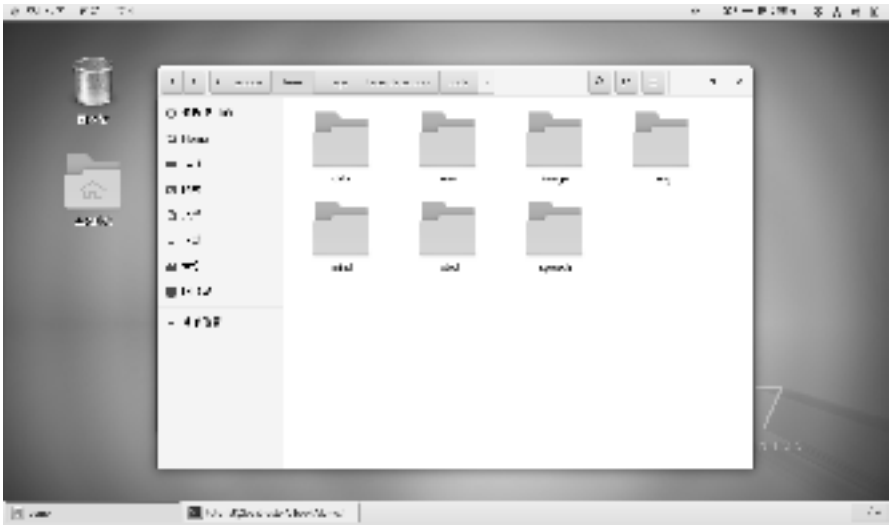


图 1-12　AIDemo 所包含的文件夹

- mind：思维学习示例程序。

打开 image 或 speech 等目录，可以看到每个目录下有若干文件夹，每个文件夹对应一个示例程序。打开某一个示例程序，可以看到该文件夹下包括一个 code 目录和一个 doc 目录，前者保存了示例系统的源代码，后者保存了该示例的说明文档。AIDemo 中的很多示例是从免费代码库 github 上下载后重新整理而成的，通常将从 github 上直接下载的代码放到 org 目录下。

认真阅读 doc 下的说明文档，可以了解运行相应示例程序的具体步骤。对大多数示例系统，code 目标下的 run. sh 是主程序入口，运行该程序即可启动该示例的默认过程。这一运行过程需要在命令行窗口中执行。在 AIDemo 虚拟机桌面上右击，选择“打开终端”，进入相应示例程序的文件夹，再进入 code 目录，通过运行下述命令启动主程序：

```
sh run. sh
```

绝大多数示例程序都设计了实验环节，这些实验通过修改 run. sh 或其他配置文件，改变默认程序的运行特性，从而让读者加深对该示例的理解。修改 run. sh 或配置文件可以通过双击这些文件，启动图形界面编辑器来完成，也可以通过更复杂的编辑工具（如 *vim*）完成。

### 1.6.3　人脸检测（Face-detection）：第一个示例程序

本小节选择人脸检测作为例子来说明如何运行 AIDemo 中的示例程序，该示例程序保存在 image/face-detection 下。所谓人脸检测，是指从一张照片中将人脸

找出来，并用方框进行标注。人脸检测是第 2 章要介绍的人脸识别技术的基础，只有把人脸找到，才有可能对其进行识别。这一任务看似简单，但当图片中包含的场景比较复杂时，检测过程很容易出错。这里将忽略技术细节，仅介绍如何启动示例系统，并通过修改代码来改变检测系统的行为方式。

Face-detection 示例程序事实上是机器视觉处理工具包 OpenCV 的演示样例。首先认真阅读 image/face-detection/doc/README，了解运行步骤。然后打开一个终端，进入目录 aibook/demo/image/face-detection/code，执行 run. sh：

```
cd aibook/demo/image/face-detection/code
sh run. sh
```

运行上述命令得到图 1-13(a)所示的输入照片，回车后得到图 1-13(b)所示的检测结果，再次回车则退出程序。

(a)　　(b)

**图 1-13　Face-detection 的默认主程序运行结果**

(a)原图；(b)检测结果

Face-detection 示例程序设计了三个实验来加深读者对人脸检测系统的理解。在第一个实验中，读者将通过修改 detect. py 中的参数来观察不同参数对检测结果的影响。例如，face_minSize 用来设定脸的最小尺寸，当调小这一数值时，会有更多脸检测出来；相反，当调大这一数值时，只有足够大的脸才能被检测到。例如，当我们设 face_minSize＝(130，130)时，就只能检测出一张脸了，如图 1-14(a)所示。如果进一步将眼睛的最小尺寸 eye_minSize 增大，如设 eye_minSize＝(60，60)，则连眼睛都检测不出来了，如图 1-14(b)所示。

在第二个实验中，我们用一张包含更多人脸的照片来考查该人脸检测系统的性能。修改 detect. py，将输入图片变量 photo_fn 定义为一张包含多个人脸的照片，如 img 目录下的 crowd.jpg，即设置 photo_fn＝img/crowd. jpg。用默认参数配置运行 run. sh，可得如图 1-15 所示的检测结果。

在第三个实验中，我们鼓励读者用手机自拍一张照片，作为 face-detection 的

(a)　　(b)

**图 1-14　Face-detection 修改参数后的运行结果**

(a)设 face_minSize=(130,130)之后的运行结果；(b)设 face_minSize=(60,60)之后的运行结果

**图 1-15　Face-detection 对多人脸照片的运行结果**

输入图来进行人脸检测，观察光照、角度、姿态等对检测结果的影响。实验时，需要将照片上传到 AIDemo 虚拟机，并将 detect.py 中的 photo_fn 参数设成该文件的路径。将照片上传到虚拟机的方法有多种，一种简单的方法是将文件上传到第三方共享网站(如百度云)，再从 AIDemo 虚拟机上通过浏览器下载；另一种方法是利用 VirtualBox 的文件夹共享功能，在主机和 AIDemo 虚拟机间交换文件。

通过上述实验过程，读者即可对人脸检测技术有个直观了解，对该实验的检测系统在性能、适用性上有更深入的认识。更重要的是，读者可以通过对模型参数进行调优，积累设计人工智能系统的基本技能，为深入学习和熟练应用人工智能技术打下基础。