

序言：机器学习和深度学习的前景

这本书的目标是什么？

这本书的目标是向高级初学者介绍基本的机器学习和深度学习的概念和算法，旨在成为机器学习和深度学习的各种“核心”功能的快速入门参考书，并提供大学课程中包含的代码示例。章节中的材料说明了如何使用Keras来解决一些问题，之后您可以通过进一步阅读以加深了解。

这本书还将为您节省搜索代码样例的时间，搜索样例——一个潜在的耗时很大的工程。在任何情况下，如果你不确定你是否能理解书中的介绍，那就浏览一下代码样例，感受问题的复杂程度。

可能这是一句废话了，但是请记住以下一点：通过阅读这本书，你不会成为机器学习或深度学习的专家。

从这本书中能学到什么？

第一章对人工智能进行了简要的介绍，第二章专门介绍用Pandas管理数据集的相关内容。第三章介绍机器学习的概念（监督学习和费监督学习）、任务类型（回归、分类和聚类）和线性回归（本章的后半部分），第四章介绍了分类算法，如知识网络、朴素贝叶斯、决策树、随机森林和SVM（支持向量机）。

第五章介绍了深度学习，并深入研究了卷积神经网络（CNN）。第六章涵盖了深度学习体系结构，如递归神经网络（RNN）和长短期记忆（LSTM）。

第六章也介绍了自然语言处理（NLP）的几个方面，包括一些基本概念和算法，以及强化学习（RL）和贝尔曼方程。附录一包含Keras，附录二包含TensorFlow 2.0。

还有一点，虽然Jupyter比较流行，但本书里的代码样例都是Python脚本。不过你可以通过各种在线教程快速了解Jupyter有用的特性。此外，可以看看完全在线的谷歌合作实验室，它基于Jupyter Notebook，以及有免费的GPU可供使用。

这本书需要多少Keras知识？

接触一些Keras的相关知识很有用，如果Keras对你来说很新鲜，可以阅读附录。如果你也想了解Keras和Logistic回归，在第三章中有三个例子，理解这些例子需要一些理论知识，包括激励函数、优化器和成本函数，所有涉及到的内容在第四章中都有讨论。

请记住，Keras已经很好地集成到TensorFlow 2中（在tf.keras命名空间中），它提供了一个超越“纯”TensorFlow的抽象层，这将使你能更快地开发原型。

我需要学习这本书的理论部分吗？

同样，答案取决于你计划在多大的程度上参与机器学习。除了创建模型之外，你还将使用各种算法来查看哪些算法提供了你的项目所需的精确度（或其他度量标准）。如果你感觉到在知识方面有所欠缺，机器学习的理论部分可以帮助你对于你的模型和数据进行“取证”分析，并帮助你确定如何改进模型以达到理想效果。

代码样例是如何创建的？

本书中的代码样例是使用Python 3和Keras创建和测试的，这两种语言内置在带有OS X 10.12.6（MacOS Sierra）的MacBook Pro上的TensorFlow 2中。代码样例主要来自作者的研究生课程《深度学习和Keras》。在某些情况下，一些代码样例包含了在线论坛中讨论的小段代码。需要记住的关键点是，鉴于本书的大小，代码样例必须尽可能地遵循“四C原则”：清晰（Clear）、简洁（Concise）、完整（Complete）和正确（Correct）。

这本书的技术前提是什么？

你需要对Python有所了解，并且知道如何从命令行启动Python代码（在类似于Unix的环境中，供Mac用户使用）。此外，基本线性代数（向量和矩阵）、概率统计（平均值、中值、标准差）和微积分中的基本概念（如导数）等知识将帮助你更好地掌握本书内容。如果你熟悉基本功能，一些Numpy和Matplotlib的知识也有帮助（比如Numpy数组）。

若要理解本书后半部分的代码样例，还需要另一个先决条件：基本熟悉神经网络，包括隐藏层和激励函数的概念（即使你并没有完全理解）。交叉熵的知识对理解一些代码样例也有帮助。

这本书的非技术前提是什么？

这个问题的答案更难量化，但拥有强烈的学习机器学习知识的渴望、阅读和理解代码样例的动机和自律是非常重要的。

即使是简单的机器语言API在初次遇到时也是一个挑战，因此要做好多次阅读代码样例的准备。

如何设置命令行解释器？

如果你是苹果用户，有三种方法。第一种方法是使用Finder导航到Applications>Utilities，然后双击Utilities应用。接下来，如果你已经有了可用的命令行解释器，你可以通过键入以下命令来启动新的解释器：

```
open/Applications/Utilities/Terminal.app
```

第二种方法是在MacBook上打开一个新的命令行解释器，只需点击解释器中的command+n，电脑就会启动另一个命令行解释器。

如果你是PC用户，你可以安装模拟bash命令的Cygwin（开源<https://cygwin.com/>），或者使用另外的工具包，比如MKS（一种商业产品）。请阅读介绍下载和安装过程的在线文档。请注意，如果自定义别名是在主启动文件以外的文件中定义的，则不会自动设置它们（例如.bash_login）。

配套文件

可以通过发送电子邮件至info@merclearning.com，向出版商获取本书中的所有代码示例和图形以供下载。

读完这本书的“下一步”是什么？

这个问题的答案多种多样，很大程度上取决于你的目标。最好的答案是从书中尝试一种新的工具或技术来解决你关心的问题或任务，无论是专业方面还是个人方面。具体是什么工具或技术，取决于你是谁，因为数据科学家、经理、学生或开发人员的需求都是不同的。此外，在面对新的挑战时，别忘了运用你学会的东西。

O. 坎皮萨托

于加州旧金山

人工智能概论

本章对人工智能进行了简单的介绍，主要对这一多样化的主体进行了概述。与本书的其他章节不同，本章作为介绍性章节，涉及的技术内容难度较低，但其具有易读性，也值得浏览。在本章末尾简要介绍了机器学习和深度学习，这两者将在后面的章节中详细讨论。

许多以人工智能为重点的书籍倾向于从计算机科学的角度讨论人工智能，研究传统的算法和数据结构。但这本书将人工智能视为机器学习和深度学习的“保护伞”，因而在其他章节之前，本书将人工智能作为预备知识进行了粗略地讨论。

本章的第一部分首先讨论了“人工智能”这一术语的含义，探讨了决定人工智能的存在各种潜在方法，以及强人工智能和弱人工智能之间的区别。读者还将了解到一种众所周知的智能测试——图灵测试。

本章的第二部分讨论了人工智能的一些使用案例，以及神经计算、进化计算、自然语言处理和生物信息学的早期方法。

本章的第三部分介绍了人工智能的主要子领域，包括自然语言处理（使用NLU和NLG方法）、机器学习、深度学习、强化学习和深度强化学习。

本章没有涉及特定的代码示例，但本章的配套文件中包括一个用于解决红驴问题的基于Java的代码示例，也包括一个用于解决模仿问题的基于Python的代码示例（需要Python2.x）。

人工智能（AI）是什么？

“人工”这一词的字面意思是“合成的，人造的”，通常具有一定的贬义，指代一个做工低劣的替代品。但是人造物体（例如花）可以非常接近实物，并且当没有任何维护需求（阳光、水等）时，人造物体的优势就突显出来。

相比之下，“智能”的定义比“人工”一词更难以捉摸。斯特恩伯格在一篇关于人类意识的文章中给出了以下定义：“智能是个人从经验中学习、推理、习得重要信息以及应对日常生活需求的认知能力。”

你可能还记得标准化测试中的问题。问题要求给出序列中的下一个数字，如1、3、6、10、15、21。首先观察到相邻数数字之间的差每次增加1：从1到3，增加了2，而从3到6，增加了3，以此类推。基于这种规则，正确答案是28。这样的问题旨在衡量我们识别模式中显著特征的熟练程度。

顺便说一句，一个“求序列中的下一个元素”的数字问题可以有多个答案。例如，序列2、4、8如果以 2^n 作为生成公式，则下一个数字可能是16，而且这一答案是正确的；而如果以 $2^n + (n-1)*(n-2)*(n-3)$ 作为生成公式，则序列中的下一个数字是22（不是16）。有许多整数可以匹配以2、4、8作为初始数字的序列，下一个数字也可能不是16或22。

让我们回到斯特恩伯格对“智能”的定义，考虑以下问题：

- 你如何判断某人或某事物是智能的？
- 动物是智能的吗？
- 如果动物是智能的，该如何定义它们的智力

我们倾向于通过与人们的互动来评估它们的智力：我们向它们提问，并观测它们的回答。虽然这种方法并不直接，但我们经常采用这种方法来评估别人的智力。

我们也通过观察动物的行为来对动物的智力进行评估。聪明的汉斯是一匹有名的马，生活在约1900年的德国柏林，据说它精通算术，例如计算加法和平方根。

事实上，汉斯能够识别人类的情感，再加上它敏锐的听觉，当它的答案接近正确答案时，它能够察觉到观众的反应。有趣的是，汉斯在没有观众的情况下表现不佳。你可能不愿意把汉斯的行为归因于智能；然而，在得出结论之前，先回顾一下斯特恩伯格给出的定义。

又如，有些生物只在群体中表现出智能。蚂蚁是一种结构简单的昆虫，它们孤立的行为很难联系到人工智能，但蚁群面对复杂的问题，能给出极为优秀的解决方案。事实上，蚂蚁可以找出从巢穴到食物来源之间的最佳路线，可以解决如何搬运重物、如何形成桥梁的问题。因此，群体智慧来自昆虫个体间的有效交流。

评估智力的其中两个指标是大脑质量和大脑与身体质量的比值，海豚在这两个指标上都优于人类。海豚的呼吸是在自主控制下进行的，这可能是大脑质量过剩的原因，也是海豚大脑的两个半球轮流睡眠的原因。海豚在动物自我意识测试中得分很高，比如镜子测试，在这一测试中，它们认识到镜子中的像实际上是它们自己的像。海豚也可以表演复杂的把戏，这一点海洋世界的游客可以作证。这些事实都表明了海豚的记忆能力和执行复杂物理运动序列的能力。

使用工具的能力是评估智力的另一块试金石，经常被用来区分直立人和人类的早期祖先。海豚同样与人类共享这种特性：海豚在觅食时使用深海海绵来保护它们的喙口。因此，智力并不是人类独有的属性。许多生物都拥有一定的智力。

现在考虑以下问题：无生命的物体，如计算机，能拥有智能吗？人工智能宣称的目标是创造表现出与人类相似的思维的计算机软件和硬件系统，换句话说，这些系统表现出通常情况下与人类智能相关的特征。

思考能力呢？机器会思考吗？记住思维和智能是有区别的，思考是推理、分析、评估和阐述想法和概念的工具，所以不是每一个具有思维能力的事物都是智能的。智能更类似于高效且有效的思维。

许多人对这个问题带有偏见，他们认为计算机是由硅和电源制成的，因此不能思考。而在另一个极端，计算机的运算速度比人类快得多，因此必然比人类更加智能。但真相很可能介于这两个极端之间。正如我们之前所讨论的，不同种类的动物拥有不同程度的智力。然而我们更感兴趣的是确定机器智能存在的测试。也许拉斐尔说得最好：人工智能是一门科学，它让机器来做一些原本需要人类来完成的事情。

强人工智能和弱人工智能

目前有两个人工智能的主要阵营——强人工智能和弱人工智能。弱人工智能方法出自麻省理工大学，它将任何表现出智能行为的系统都视为人工智能。这一阵营关注的是一个程序是否能正确执行，而忽略系统是否以与人类相同的方式来执行任务。电气工程、机器人和相关领域的人工智能项目的结果主要与如何得到令人满意的性能有关。

另外一种人工智能的方法叫做生物合理性，出自卡耐基梅隆大学。据此方法，当一个系统表现出的智能行为应该基于人类使用的方法。例如，考虑一个有听觉能力的系统：强人工智能的支持者可能通过模拟人类听觉系统来实现；而弱人工智能的支持者可能只关心系统的性能，这种模拟会包括耳蜗、耳道、耳膜和耳朵的其他部分，每个部分在系统中执行相应的任务。

因此，弱人工智能的支持者仅根据他们构建的系统的性能来衡量其是否成功。他们认为研究人工智能的目的是解决难以解决的问题，而不考虑这些问题究竟如何解决。

另一方面，强人工智能的支持者关心他们构建的系统的结构。他们认为，仅仅凭借掌握启发式程序、算法和人工智能程序的知识，计算机就可以拥有意识和智能。众所周知，好莱坞出品过各种属于强人工智能阵营的电影（如《我》《机器人》《银翼杀手》）。

图灵测试

前一节提出了三个问题，前两个问题已经解决了：如何判断某事物是否智能，以及动物是否是智能的。第二个问题的答案可能是否定的，也可能不是。有的人比别人聪明，有的动物比别的动物聪明，在机器智能中也有同样的问题。

艾伦·图灵尝试用操作术语来回答关于智能的问题。他希望将功能（做什么）和实现（如何构建）分开。他设计了一种名为“图灵测试”的工具，这将在下一节中讨论。

图灵测试的定义

艾伦·图灵提出了两种模仿游戏，其中一个人或实体的行为与另一个保持一致。在第一个游戏中，一个人（称为提问者）在一个房间里，房间中央有一个窗帘，窗帘的另一边是另一个人，由提问者来确定这个人的性别。提问者（无关性别）通过一系列问题来完成这一任务。

这个游戏假设男性在回答中会撒谎，而女性不会撒谎。为了避免提问者通过声音判断性别，交流是通过计算机而不是口头语言来进行。如果窗帘另一边的人成功欺骗了提问者，他就赢得了模仿游戏。

在图灵最初设计的测试中，一男一女都坐在窗帘后面，提问者必须正确识别两个人。

图灵可能是基于当时流行的游戏来设计测试的，这甚至可能是他的机器智能测试背后的推动力。

关于图灵测试的其他有趣的问题可参考以下两个链接：

<https://futurism.com/the-byte/scientists-invented-new-turing-test>

<https://theconversation.com/our-turing-test-for-androids-will-judge-how-lifelike-humanoid-robots-can-be-120696>

你可能不知道的是，埃里希·弗洛姆是二十世纪著名的社会学家和精神分析专家，他认为男女平等但并不相同。例如，性别使人们对颜色、花朵，或者购物时间的认知不同。区分男女和智能问题有什么关系？图灵可能明白思维存在不同的类型，理解并容忍差异的存在是很重要的。

一个查询测试

图灵的第二个游戏更适合拿来研究人工智能。同样地，提问者坐在一个有窗帘的房间里，这一次由一台电脑或一个人躲在窗帘后面，电脑扮演男性角色，在这种情况下，撒谎变得很容易。

另一方面，这个人不会撒谎。提问者提问，评估得到的回答，以确定他是在与人交流还是与机器交流。如果电脑成功欺骗了提问者，它就通过了图灵测试，进而被认定为是智能的。

启发法

启发法非常有用，人工智能的应用经常依赖于启发法。启发法本质上是解决问题的“经验法则”，换句话说，启发法是一系列能多次使问题解决的指导方法。启发法与算法的差别在于，算法是解决问题的一组确定的规则，其输出是完全可预测的。

启发法是一种寻找近似解的方法，可以在其他方法太耗时或太复杂（或两者都有）时使用。使用启发法可能会得到不错的解，但不能保证一定能得到解。启发法在人工智能发展早期尤为流行。

日常生活中有很多启发法的应用。例如，很多人喜欢用启发法而不是问路来辨别驾驶方向；当晚上离开高速公路时，有时很难找到返回主干道的路线。一个被证明有效性的启

发法是，每当司机来到一个岔路口时，就用更多的路灯来指向正确的方向。你可能有一个找回掉落的隐形眼镜的最喜欢的方法，或者在拥挤的购物中心找到停车位，这两者都是启发法的例子。

人工智能问题往往很大且计算复杂，而且通常无法通过简单的算法来解决。人工智能问题及其领域倾向于体现大量的人类专业知识，在用强人工智能的方法来解决问题的情况下更是如此。有些类型的问题使用人工智能可以很好地解决，而其他类型的问题更适合传统的计算机科学方法，包括通过简单的决策或精确的计算来产生解决方案。我们考虑几个例子：

- 医疗诊断
- 使用带条形码扫描功能的收银机购物
- 自动取款机
- 双人游戏，如国际象棋和跳棋

医学诊断属于科学领域，多年来一直受益于人工智能的发展，尤其是专家系统的发展，给医学诊断做出了巨大的贡献。专家系统通常建立在涉及大量人类专业知识和许多规则的领域中，这些规则通常是这样的：如果遇到某种情况，那么应该做出何种行动。举个小例子：如果你头疼，那就吃两片阿司匹林，或者是“早上给我打电话”。

在众多人工智能技术中，专家系统变得尤为流行(而且非常有用)，因为它能够存储的规则比人类大脑所能存储的要多得多。专家系统是最成功的人工智能技术之一，通过专家系统给出的问题的解决方案是全面和有效的。事实上，专家系统可以帮助人类做出更准确的决定(甚至“挑战”不正确的选择)。

遗传算法

一个优秀的范例是达尔文的进化论，它涉及到自然选择——一种在自然界中以数千年或数百万年为单位的速率发生着的现象。相比之下，计算机内部的进化比自然选择快得多。

遗传算法是一种“模仿”自然选择过程的启发法，大致过程为选择最合适的个体进行繁殖，以产生下一代。

让我们来比较一下人工智能的使用与动植物世界的进化过程。在进化过程中，物种通过自然选择、繁殖、突变和重组的遗传算子来适应环境。

遗传算法(GA)是来自普遍域的一种特定的方法论，又被称为进化计算，是人工智能的一个分支。通过遗传算法提出的解决问题的方法在很大程度上与生物适应现实世界的环境的原理相似。

如果你感兴趣的话，下面的链接包含一些关于遗传算法的有趣细节：

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

知识表示

当我们考虑人工智能相关的问题时，表示(representation)问题变得很重要。人工智能系统要实现获取和存储知识，以处理知识并产生智能结果的功能，还需要识别和表示知识的能力。选择表示的形式是解决问题和理解问题本质的内在本质。

正如乔治·波利亚（一位著名的数学家）所说，一个好的表示形式几乎和为一个特定问题设计的算法或解决方案一样重要。恰当且自然的表示有助于产生高效易懂的解决方案。

举一个表示方法选择的例子。考虑著名的传教士和食人族问题，问题的目标是用一艘船将三名传教士和三名食人族从一条河的西岸转移到东岸。在从西到东的转移过程中，你可以通过选择适当的表示来查看解决方案。这个问题有两个限制：船在任何时候都不能容纳两个人，两岸的食人族数量永远不能超过传教士

这个问题（以及相似的“嫉妒的丈夫”问题）的解决方案如下：

https://en.wikipedia.org/wiki/Missionaries_and_cannibals_problem#targetText=The%20missionaries%20and%20cannibals%20problem.an%20example%20of%20problem%20representation

基于逻辑的方法

人工智能研究人员使用基于逻辑的方法来实现表示知识和解决问题。特里·维诺格拉德的《积木世界》（1972）是一个将逻辑用于解决此类问题的开创性例子。在该书中，一个机器手臂摆放桌面上的积木。这个项目包括语言理解和场景分析，以及人工智能的其他方面。

此外，生产式规则和生产式系统被用来构造许多成功的专家系统。基于清晰简洁地表示启发法的可行性，生产式规则和专家系统颇具吸引力。已有数以千计的专家系统采用这种方式构建起来。

语义网络

语义网络是另一种复杂的图形化知识表示方法，语义网络先于使用继承的面向对象语言（其中来自特定类的对象继承了超类的很多属性）产生。

构建语义网络的大部分工作量都集中在表示语言的结构和知识上。例子包括Stuart Shapiro SNePS（语义网处理系统）和罗杰·尚克在自然语言处理方面的工作。

知识表示也存在其他选择：图形化的方法对感官有更大的吸引力，如视觉、空间和运动。最早的图形化方法可能是状态空间表示法，它表示了系统所有可能的状态。

人工智能和游戏

自二十世纪中叶以来计算机出现后，通过训练计算机掌握复杂的棋盘游戏，计算机科学取得了显著的进步，人类的编程技术也更加熟练。通过使用人工智能的观点和方法，一些计算机游戏得以提升，包括国际象棋、跳棋、围棋和奥赛罗。

游戏推动了人工智能的发展，也激发了人们对人工智能的兴趣。1959年，亚瑟·塞缪尔对跳棋的研究，为早期人工智能的发展做出了贡献。他的程序基于50个启发式的表格，并

且在不同版本的程序之间进行对抗。在一系列的比赛中，失败的程序将采用获胜程序的启发法。这个程序有高超的跳棋技术，但从未掌握这个游戏。

几个世纪以来，人们一直试图训练机器来下好象棋。人们对国际象棋的迷恋可能是源于普遍观点，即下好象棋需要智力。

1959年，纽厄尔、西蒙和肖开发了第一个真正的象棋程序，该程序遵循香农-图灵范式。理查德·格林布拉特的程序是第一个参与俱乐部级象棋对抗的程序。计算机象棋程序在20世纪70年代稳步发展，直到70年代末，象棋程序已经达到了专家水平（相当于国际象棋锦标赛玩家的前1%）。

1983年，肯·汤普森的《美女》是第一个正式达到大师级别的程序。随后，来自卡耐基梅隆大学的海泰克取得了成功，作为第一个高级大师级别（超过2400级）的程序，它成为了一个重要的里程碑。此后不久，深度思考项目（也来自卡耐基梅隆大学）被开发出来，并成为第一个能够定期击败大师的项目。

当IBM在20世纪90年代接管该项目时，“深度思考”（Deep Thought）演变成了“深蓝”（Deep Blue），深蓝与世界冠军加里·卡斯帕罗夫进行了六场比赛，后者1996年在费城赢得了一场拯救人类的比赛。然而在1997年，卡斯帕罗夫在对阵深蓝的继承者（Deeper Blue）时，输掉了比赛，棋坛震动。

在随后与卡斯帕罗夫、克拉姆尼克和其他世界锦标赛级别的球员的六场比赛中，对抗程序表现良好，但这并不是真正的世界锦标赛。尽管人们普遍认为这些程序可能仍然比不上最优秀的人类棋手，但大多数人愿意承认，顶级的对抗程序下棋与成就最高的人类有所不同（如果有人想到图灵测试的话）。

1989年，埃德蒙顿阿尔伯塔大学的乔纳森·谢弗用他的奇努克程序开始了他征服跳棋游戏的长远目标。在1992年的一次共40场比赛的跳棋赛中，对战多年的跳棋世界冠军马里昂·廷斯利，奇努克输了4场，平局34场。1994年，这对对手在六场比赛后达成平手，当时廷斯利因为健康原因不得不弃权。从那之后，谢弗和他的团队始终在坚持研究跳棋，包括游戏结束时的棋局（所有的8个棋子和或更少的结局），和游戏开始时的棋局。

其他使用人工智能技术的游戏有双陆棋、扑克、桥牌、奥赛罗和围棋（通过被称为the new drosophila，“新果蝇”）。

AlphaZero的成功

谷歌创建了AlphaZero，这是一个基于人工智能的软件程序，通过自学来学习如何玩游戏。AlphaZero是2016年击败了世界最佳人类棋手的AlphaGo的继承者。AlphaZero在围棋游戏中轻松击败了AlphaGo。

此外，在学习了国际象棋的规则之后，AlphaZero进行了自我训练（再次通过自学方式），并且在一天之内成为了世界上最顶尖的棋手。AlphaZero可以打败任何人类棋手以及任何下棋的计算机程序。

一个很有趣的点是，AlphaZero开发了自己的下棋策略，这种策略不仅不同于人类，而且包括了被认为违反直觉的象棋动作。

不幸的是，AlphaZero无法告诉我们它是如何开发出一种优于之前任何一种下棋方法的策略的。既然AlphaZero 100%自学，是世界排名第一的棋手，那么AlphaZero算不算智能的？

专家系统

专家系统是自人工智能诞生以来一直被研究的领域之一，是一门可以称之为巨大成功的人工智能学科。专家系统有许多特点，使其成为人工智能研究和开发的理想选择。这些特点包括知识库与推理机的分离、超过任何甚至所有专家的总和、知识与搜索技术的关系、推理和不确定性等。

诞生最早、引用最多的专家系统之一是启发式DENDRAL。其目的是根据质谱图鉴定未知化合物。DENDRAL是在斯坦福大学开发的，目的是对火星土壤进行化学分析。它是第一个表明在特定学科中编码领域专家知识的可行性的系统之一。

可能最著名的专家系统还是同样来自斯坦福大学（1984）的MYCIN。开发MYCIN是为了方便对传染性血液疾病的调查。然而，比它的应用领域更能体现其重要性的是，MYCIN为随后的所有基于知识的系统的建立提供了范例。MYCIN有400多条规则，最终被用来为斯坦福医院的住院医生提供培训对话。

20世纪70年代，PROSPECTOR（也在斯坦福大学）被开发用于矿物勘探。PROSPECTOR也是早期使用推理网络的一个很有价值的例子。

随后在20世纪70年代出现的其他著名的成功系统有XCON（大约有10,000条规则），它是为了帮助在VAX计算机上配置电路板而开发的；GUIDON，一个辅导系统，是MYCIN的一个分支；TEIRESIAS，一个针对MYCIN的知识获取工具；HEARSAY I和HEARSAY II，使用黑板结构进行语音理解的主要例子。

道格·莱纳特的人工数学系统（AM）是20世纪70年代研究和开发工作的另一个重要成果，此外还有用于不确定性推理的登普斯特-斯查费理论，以及扎德在模糊逻辑方面的工作。

自20世纪80年代以来，数以千计的专家系统被应用在诸如配置、诊断、指令、监控、规划、预后、治疗和控制方面。今天，除了独立的专家系统，许多专家系统已经被嵌入到其他用于控制的软件系统中，包括医疗设备和汽车中的专家系统（例如应该何时对汽车进行牵引控制）。

此外，许多专家系统外壳，如Emycin、OPS、EXSYS和CLIPS，已经成为行业标准。许多知识表示语言也已经发展起来。如今，许多专家系统在幕后发挥作用以改善人们的日常体验，例如在线购物车。

神经计算

麦卡洛克和皮茨试图理解动物神经系统的行为，并在神经计算方面进行了早期研究。但他们的人工神经网络模型（ANN）有一个严重的缺点：它不包括学习机制。

弗兰克·罗森布拉特开发了一种名为概念学习规则的迭代算法，用于在单层网络（所有神经元都直接链接到输入端的网络）中找到合适的权重。这一新兴领域的研究可能受到了明斯基和帕皮特的公告的严重阻碍，他们认为某些问题不能用单层感知器来解决，例如异或函数。这一公告发布后，联邦政府对神经网络研究的资助被大幅削减。

随着霍普菲尔德的工作，该领域在20世纪80年代经历了第二波崛起。霍普菲尔德的异步网络模型（霍普菲尔德网络）使用能量函数来近似求解NP完全问题。

20世纪80年代中期，研究人员还开发了反向传播算法，这是一种适用于多层网络的学习算法。基于反向传播算法的网络通常用于预测道琼斯平均指数，并在光字符识别系统中读取印刷材料。

神经网络也用于控制系统。ALVINN是卡耐基梅隆大学的一个项目，它使用反向传播网络感知高速公路路况并协助控制实验室车辆的转向。

这项研究的一个直接应用是，对因睡眠不足、酗酒或因其他情况受到干扰而造成车辆偏离高速公路车道的驾驶员发出警告。展望未来，希望有一天类似的系统可以用来驾驶车辆，这样我们就可以自由地阅读报纸和用手机聊天，以利用额外的空闲时间。

进化计算

遗传算法通常被归类为进化计算。遗传算法是霍兰德开发的一种算法，使用概率和并行性来解决组合问题（也称为优化问题）。

然而，进化计算不只关注最优化模型。罗德尼·布鲁克斯曾是麻省理工学院计算机科学和人工智能实验室的主任，他成功创造了人类水平的人工智能，并且因为摆脱了对基于符号的方法的依赖而将其作为人工智能研究的圣杯。基于符号的方法依赖于启发法和表征范例的使用。

在他看来，智能系统可以设计成多个层次，其中更高的层次依赖于它们下面的层。例如，如果你想建造一个能够避开障碍物的机器人，实现避障的程序会建立在一个较低的层，只负责机器人的运动。

布鲁克斯认为，智能是通过主体与环境相互作用而产生的。他最有名的成就可能是他实验室里制造的昆虫机器人，这些机器人体现了这种智能哲学。在智能哲学的观点下，一群自主机器人与其周围的环境以及彼此之间进行互动。

自然语言处理

如果我们希望建立智能系统，自然会要求我们的系统拥有语言理解能力。这是一条被很多早期的研究者理解的公理。Eliza是一个著名的早期应用程序，由麻省理工学院的约瑟夫·韦森鲍姆开发。约瑟夫是与肯尼斯·科尔比（斯坦福大学精神病学家）一起工作的计算机科学家。

Eliza被设定去模仿卡尔·罗杰斯学校的精神病学家扮演的角色。例如，如果用户输入“我觉得累了”，Eliza是一个使用反向传播算法的应用程序，它可以学习英语文本的正确发音。据说其英语发音的准确率高达95%。显然，程序的问题在于英语单词发音中固有的不一致性，如粗读和连读，以及一些来自其他语言的单词，如“pizza”和“fizzy”。

特里·维诺格拉德写了另一个著名的程序，该程序以ETAOIN SHRDLU的第二组字母命名，这是排字机上英语中最常用的字母。维诺格拉德的程序可能会回应道“你说你感觉累了。多说点。”“对话”将以这种方式继续，机器的回答几乎没有原创性。一个真正的精神病学家可能会采取这种方式，希望病人能发现他们自己真实的（也许是隐藏的）感受和沮丧。与此同时，Eliza只是用模式匹配来模仿人类的互动。

尽管韦森鲍姆的学生（和公众）完全知道Eliza只是一个程序，但他们仍然对与Eliza互动有浓厚的兴趣，韦森鲍姆为此感到不解。同时，科尔比仍然致力于这个项目，并成功开发了一个名为“医生”的程序。

尽管Eliza对自然语言处理（NLP）的贡献甚微，但它是一个尝试拥有我们人类最后的特殊能力——感受情绪能力的软件。当人和机器（如机器人）之间的界限变得不那么清晰时，会发生什么？也许在五十多年后，这些机器人将不再像凡人，而是更像神仙？

最近，几个麻省理工学院的机器人，包括Cog、Kismet和Paro，已经被开发出不可思议的能力来伪装人类的情绪，并引起与他们互动的人的情绪反应。特克尔研究了疗养院的儿童与老人与这些机器热形成的关系——涉及真实情感和关怀的关系。特克尔提到了重新定义关系一词的潜在需要，以将人们与这些所谓的关系人工制品之间的关系包括在内。然而她仍然坚信，这种关系永远不会取代只能产生在必须每天面对自己的死亡的人类之间的纽带。

维诺格拉德的积木世界包括一个能够实现各种目标的机器人手臂。例如，如果SHRDLU被要求举起一个上面有一个绿色小块的红色木块，它知道它必须在举起红色块之前移除绿色块。与Eliza不同，SHRDLU能够理解英语命令，并对这些命令做出适当的反应。

HEARSAY是语音识别领域的一个很有野心的项目，它采用了黑板结构。在这种结构中，各种语言成分（如语音和短语）和独立知识源（代理）可以自由交流。句法和语义都被用来删减不合适的单词组合。

HWIM（hear what I mean的缩写）项目使用增强的过渡网络来理解口语。它有1000个关于旅行预算管理的词汇。也许这个项目在范围上显得太有野心，因为它的表现不如HEARSAY II。

解析在这些自然语言处理程序的成功中扮演了不可或缺的角色。SHRDLU使用上下文无关语法来帮助解析英语命令。上下文无关语法为处理符号串提供了一种句法结构。然而，为了有效处理自然语言，还必须考虑语义。

解析树展示了组成句子的单词之间的关系。例如，许多橘子可以分为主语和谓语，主语可以分为名词短语和介词短语等等。本质上说，解析树给出了句子的语义。

这些早期的语言处理系统都一定程度上应用了世界知识。然而，在20世纪80年代后期，自然语言处理进步的最大绊脚石是常识知识的问题。例如，尽管许多成功的程序在自然语言处理和人工智能的特定领域构建的，但这些程序经常被批评过于微观，这意味着这些程序不具有一般的、真实世界的知识或共性认知。例如，一个程序可能非常了解某个特定的场景，比如在餐馆点餐，但是它不知道服务员是否还活着，也不知道他们通常是否穿衣服。在过去的25年里，得克萨斯州奥斯汀市中冶公司的道格拉斯·莱纳特一直在建立最大的常识知识库来解决这个问题。

自然语言处理经历了一些有趣的发展。在其初始阶段（如本节前面所述）之后，自然语言处理依赖统计数据来管理句子的解析树。夏尼亚克描述了如何扩充上下文无关语法（CFGs），使每条规则都有一个相应的概率。这些概率可以从佩恩树银行获得，该银行包含超过100万字的人工解析的英文文本，大部分来自《华尔街日报》。夏尼亚克展示了这种统计方法如何成功地从《纽约时报》的头版获得一个句子的解析（即使对大多数人来说，这也不是很容易）。

自然语言处理发展的下一步涉及到深度学习体系结构RNN（神经网络）、LSTM（学习结构和双向学习结构），这些将在第五章中讨论。最新的架构名为transformer，是谷歌在2017年开发的。基于transformer（以及attention）的BERT，是目前可用于解决NLP任务的最强大的开源系统之一。自然语言处理的另一种方法涉及深度强化学习（在第六章中简要讨论）。

生物信息学

生物信息学是一门新兴学科，涉及计算机科学的算法和技术在分子生物学中的应用。它主要涉及生物数据的管理和分析。在结构基因组学中，人们试图为每种观察到的蛋白质指定一种代表性结构。自动发现和数据挖掘可以帮助实现这一目标。

尤里查和格拉斯哥证明了基于案例的推理如何有助于发现每种蛋白质的代表性结构。在《人工智能与生物信息学》特刊2004年的一篇调查文章中，格拉斯哥、尤里查和罗斯特指出：“最近生物信息学的相关领域中发展最快的可能是微阵列数据的分析。”

微生物学家被他们获得的数据的种类和数量淹没。他们被要求仅基于巨大的数据库来理解分子序列、结构和数据。许多研究人员认为，来自知识表示和机器学习的人工智能技术也将被证明是有帮助的。

本章的下一部分简要介绍了人工智能的主要部分，包括机器学习和深度学习。

AI的主要部分

本书的后续章节深入探讨了人工智能的各个重要部分，包括：

- ML (机器学习, Machine Learning)
- DL (深度学习, Deep Learning)

- NLP (自然语言处理, Natural Language Processing)
- RL (强化学习, Reinforcement Learning)
- DRL (深度强化学习, Deep Reinforcement Learning)

传统的人工智能（20世纪）是基于规则的集合，由此产生了20世纪80年代的专家系统。传统的人工智能还包括由约翰·麦卡锡（1956年第一次正式人工智能会议的成员之一）创建的LISP。

传统的人工智能主要是一套与条件逻辑相结合的规则，对于20世纪80年代开发的强大的专家系统来说也是如此。然而，基于规则的决策系统可能涉及数千条规则。即使是简单的对象也需要许多规则：试想一套规则来定义一把椅子、一张桌子，甚至只是一个苹果。传统的人工智能有一些显著的局限性，主要是因为其需要大量的规则。

机器学习

大约在20世纪中叶，机器学习（人工智能的一个子集）主要依靠数据来优化和“学习”如何执行任务，这一过程通常伴随着新的或改进后的算法，如线性回归、kNN、决策树。随机森林和SVM，除了线性回归，所有其他算法都是分类器。

正如您将看到的，机器学习是一个包含其他子领域的多样化且充满活力的领域。

数据（而不是规则）在机器学习中非常重要，通常是以下类型之一：

- 监督学习（大量标记数据）
- 半监督学习（大量部分标记的数据）
- 无监督学习：大量数据，聚类
- 强化学习：试验、反馈和改进

根据吴恩达（Coursera的联合创始人）所说，“99%的机器学习都是受监督的。”

除了对数据进行分类之外，机器学习算法还可以分为以下主要类型：

- 分类器（针对图像、垃圾邮件、欺诈等）
- 回归（股价、房价等）
- 聚类（无监督分类器）
- 深度学习

机器学习的一个重要分支是深度学习，它也起源于20世纪中叶。深度学习体系结构以感知机作为神经网络的基础，通常涉及大型或海量数据集。这样的体系结构还涉及启发法和经验结果。现在深度学习在一些图像分类上的准确率可以超越人类。

虽然机器学习设计多层感知机（MLP），但深度学习引入了深度神经网络，具有新算法和新架构（例如，卷积神经网络、RNN和LSTM）。

强化学习

强化学习（也是机器学习的一个子集）使用试错法以最大化回报。深度强化学习结合了深度学习和强化学习的优点。特别地，强化学习中使用了神经网络。

深度强化学习在许多不同领域都有应用，其中三个最受欢迎的领域是：

- 游戏（围棋、象棋等）
- 机器人学
- 自然语言处理

一些在游戏中使用强化学习的著名成功实例：

- Alpha Go （混合RL）
- Alpha Zero （完整RL）
- 贪婪算法的频繁涉及
- Deep RL: 结合深度学习和RL

机器人技术

机器人以各种方式渗透到我们个人和职业生活的方方面面，包括：

- 外科手术（辅助外科医生）
- 放射学（检测癌症）
- 药品管理不力
- 宗教比较理论
- 法律/房地产/军事/科学
- 喜剧（包括单口相声）
- 音乐（指挥管弦乐队）
- 餐馆（美食）
- 协调舞蹈队
- 许多其他领域

机器人卡车司机正在取代人类的工作，但它们也有优势：它们唯一的成本是机器的维护。此外，机器人不会像人类一样分心，它们不会卷入事故中，也不需要工资或任何形式的休假。然而，尽管机器人取得了令人欣喜的成就，但《星际迷航》中的角色“Data”仍然只是一个梦。

自然语言处理是计算机科学和人工智能的一个领域，涉及计算机和人类语言之间的交互。在早期，自然语言处理涉及基于规则的技术或统计技术。自然语言处理和机器学习可以处理或分析大量的自然语言数据，其中计算机程序执行处理任务。

有许多自然语言处理任务可以用机器学习技术来解决。涉及自然语言处理的一些有趣的领域包括：

- 语种间翻译
- 从文本中寻找有意义的信息
- 汇总文档
- 检测宣扬仇恨的言论

尽管机器学习有许多优点和先进之处，但仍有一些问题需要解决。其中一个问题是职业偏见：人工智能系统推断白人男性是医生，白人女性是家庭主妇。另一个问题涉及到检测性别偏见，例如，在约2018年，18%的维基百科撰写者是女性，而84%到90%的维基百科编辑是男性。

在下面的文章中分析的另一个问题涉及数据偏差和算法偏差：

<https://www.forbes.com/sites/charlestowersclark/2018/09/19/can-we-make-artificial-intelligence-accountable/>

最后是人工智能造成的伦理问题，其中包括一些发人深省的问题（比如失业和机器人权利）。下面这篇文章提到了大量的伦理问题：

<https://www.weforum.org/agenda/2016/10/top-10-ethical-issues-in-artificial-intelligence/>

代码示例

配套光盘包含以下文件：

- RubiksCube.py
- Board.java
- Search.java

Python文件是魔方问题的解决方案，两个Java文件是红驴问题的解决方案。

要运行一个Java程序，请下载java运行时环境（JRE）：

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

为编译和运行一个Java程序，请下载Java SDK：

<https://www.java.com/en/>

如果你没有安装Python，以下有与Python相关的下载：

<http://www.python.org/getit/>

如果你没有安装Java，你可以在网上查找相关的安装指导，以及编译和启动Java代码的指导。

小结

在这一章中，我们学习了人工智能、强弱人工智能和智能图灵测试。然后我们了解了启发式算法及其在算法中的有效性，接着是遗传算法和知识表示。接下来，我们看到了人工智能最初是如何应用于不同的领域的，如游戏和专家系统。

我们还了解了神经计算、进化计算、自然语言处理和生物信息学的早期方法。此外，我们了解了人工智能的主要子领域，包括自然语言处理、机器学习、深度学习、强化学习和深度强化学习。