

深度学习

Java语言实现

[日] 巢笼悠辅 (Yusuke Sugomori) 著
陈澎 王磊 陆明 译



机械工业出版社
China Machine Press

Java Deep Learning Essentials

深度学习

Java语言实现

[日] 巢笼悠辅 (Yusuke Sugomori) 著

陈澎 王磊 陆明 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

深度学习: Java 语言实现 / (日) 巢笼悠辅著; 陈澎, 王磊, 陆明译. —北京: 机械工业出版社, 2017.6

(智能系统与技术丛书)

书名原文: Java Deep Learning Essentials

ISBN 978-7-111-57298-5

I. 深… II. ①巢… ②陈… ③王… ④陆… III. JAVA 语言—程序设计 IV. TP312.8

中国版本图书馆 CIP 数据核字 (2017) 第 154253 号

本书版权登记号: 图字: 01-2016-8646

Yusuke Sugomori: Java Deep Learning Essentials (ISBN: 978-1-78528-219-5).

Copyright © 2016 Packt Publishing. First published in the English language under the title “Java Deep Learning Essentials”.

All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2017 by China Machine Press.

本书中文简体字版由 Packt Publishing 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

深度学习: Java 语言实现

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 张锡鹏

责任校对: 殷虹

印刷: 三河市宏图印务有限公司

版次: 2017 年 7 月第 1 版第 1 次印刷

开本: 186mm × 240mm 1/16

印张: 12.25

书号: ISBN 978-7-111-57298-5

定价: 49.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

FOREWORD

译者序

本书是一本实战型的深度学习和机器学习宝典，十分适合 Java 的深度学习入门者。本书涵盖了深度学习的主要成熟算法，一步步地剖析算法背后的数学原理，并提供大量通俗易懂的代码加以说明。同时，为了能更好地指导实践，作者生动地阐述了很多宝贵的工程经验和技術直觉。最后，本书介绍了该领域最新的研究和应用成果，还包括一些实用的网络资源及研究方法。总之，本书值得深度学习爱好者细细品味。

最令人吃惊的是，本书作者 Yusuke Sugomori 竟然是一位十分年轻的“老司机”，拥有丰富的工程经验。从本书内容中，我们能隐约领悟到作者探索深度学习的捷径，就是“敢于实践，善于实践，快速实践！”。因此，我们也建议读者从最基本的部分就边学边做，不断深入理解深度学习的内涵。

本书的译者分工如下，陆明负责第1、2、6章，王磊负责第3、4、5章，陈澎负责前言、附录及第7、8章，并负责全书的审校工作。感谢机械工业出版社的编辑给予的帮助！

特别感谢我即将出生的孩子，一直支持我的妻子和父母，感谢合作译者陆明和王磊的家人！

“轻鞭一挥芳径去，漫闻花儿断续长”，我们有理由对人工智能的未来怀有更无限的憧憬！

陈 澎

2017年3月于北京

前 言

目前,人工智能技术举世瞩目,深度学习也引起人们广泛关注。在实践上,深度学习推动了人工智能革命性进步,其相关算法已经应用到众多领域。然而,这种“革命性”的技术,常被认为非常复杂,让人敬而远之。而实际上,深度学习的理论和概念并不晦涩难懂。本书将一步步地介绍相关理论和公式,并引导读者从零开始完成编码实现。

本书内容

- 第1章:介绍深度学习的演化过程。
- 第2章:介绍与深度学习相关的机器学习算法。
- 第3章:介绍深度信念网络与栈式去噪自编码器。
- 第4章:集中介绍 dropout 和 CNN 的相关算法。
- 第5章:重点介绍深度学习库 DL4J 及实践经验。
- 第6章:面向实战,实践深度学习算法和相关 Java 库的工程开发。
- 第7章:广泛介绍 Teano、TensorFlow 和 Caffe 等深度学习框架。
- 第8章:介绍深度学习的最新动态及相关资源。

本书的使用要求

Java 8 或以上(支持 lambda 表达式), DeepLearning4J 0.4 或以上版本的 Java 库。

目标读者

本书是为那些想了解深度学习算法并期望应用到实践中的 Java 程序员而设计的。

内容涵盖机器学习和深度学习的核心概念和方法，但并不要求读者具有机器学习经验；同时，本书用极简的代码实现深度学习算法，这对一般 Java 程序员在语言技能和深度学习实现上有很大帮助。

下载示例代码

读者可使用在 <http://www.packtpub.com> 注册的账户下载本书的示例代码。如果你不是在官网购买的此书，可以访问 <http://www.packtpub.com/support> 注册，代码文件会直接通过电子邮件发送给你。

你可根据以下步骤下载代码文件：

- (1) 使用你的电子邮箱和密码登录或注册我们的网站。
- (2) 将鼠标悬停在上方的 SUPPORT（支持）标签处。
- (3) 点击 Code Downloads & Errata。
- (4) 在 Search（搜索）栏输入书籍名称。
- (5) 选择你要下载代码文件的书籍。
- (6) 从下拉菜单中选择你自何处购买此书。
- (7) 点击 Code Download。

也可以通过点击 Packt 官网该书页面上的 Code Files 按钮来下载代码文件。在 Search 栏输入书籍名称就可以访问书籍页面。但这需要先登录你的 Packt 账户。

下载文件后，请用以下软件的最新版本解压文件：

- WinRAR/7-Zip（对于 Windows）。
- Zipeg/iZip/UnRarX（对于 Mac）。
- 7-Zip/PeaZip（对于 Linux）。

本书的代码包也可从位于 <https://github.com/PacktPublishing/Java-Deep-Learning-Essentials> 的 GitHub 下载。<https://github.com/PacktPublishing/>有海量的其他书籍的代码包和视频，去看看吧！

目 录

译者序

前言

第1章 深度学习概述..... 1

1.1 人工智能的变迁..... 2

1.1.1 人工智能的定义..... 2

1.1.2 人工智能曾经的辉煌..... 3

1.1.3 机器学习的演化..... 8

1.1.4 机器学习的局限性..... 10

1.2 人与机器的区分因素..... 12

1.3 人工智能与深度学习..... 13

1.4 小结..... 20

第2章 机器学习算法——

为深度学习做准备..... 22

2.1 入门..... 22

2.2 机器学习中的训练需求..... 23

2.3 监督学习和无监督学习..... 25

2.3.1 支持向量机..... 26

2.3.2 隐马尔可夫模型..... 28

2.3.3 神经网络..... 28

2.3.4 逻辑回归..... 29

2.3.5 增强学习..... 30

2.4 机器学习应用流程..... 30

2.5 神经网络的理论和算法..... 35

2.5.1 单层感知器..... 36

2.5.2 逻辑回归..... 42

2.5.3 多类逻辑回归..... 45

2.5.4 多层感知器..... 50

2.6 小结..... 57

第3章 深度信念网络与栈式去

噪自编码器..... 58

3.1 神经网络的没落..... 58

3.2 神经网络的复兴..... 59

3.2.1 深度学习的进化——突破

是什么..... 59

3.2.2 预训练的深度学习..... 60

3.3 深度学习算法..... 64

3.3.1 限制玻尔兹曼机..... 65

3.3.2 深度信念网络..... 74

3.3.3 去噪自编码器..... 80

3.3.4 栈式去噪自编码器..... 84

3.4 小结..... 86

第4章 dropout 和卷积神经

网络 87

4.1 没有预训练的深度学习

算法 87

4.2 dropout 88

4.3 卷积神经网络 97

4.3.1 卷积 98

4.3.2 池化 101

4.3.3 公式和实现 101

4.4 小结 114

第5章 探索 Java 深度学习

库——DL4J、ND4J

以及其他 115

5.1 从零实现与使用库/框架 115

5.2 DL4J 和 ND4J 的介绍 117

5.3 使用 ND4J 实现 118

5.4 使用 DL4J 实现 124

5.4.1 设置 124

5.4.2 构建 127

5.4.3 CNNMnistExample.java/
LenetMnistExample.java ... 135

5.4.4 学习速率的优化 140

5.5 小结 142

第6章 实践应用——递归神经

网络等 143

6.1 深度学习热点 143

6.1.1 图像识别 144

6.1.2 自然语言处理 145

6.2 深度学习的挑战 157

6.3 最大化深度学习概率和能力
的方法 159

6.3.1 面向领域的方法 159

6.3.2 面向分解的方法 162

6.3.3 面向输出的方法 164

6.4 小结 165

第7章 其他重要的深度

学习库 166

7.1 Theano 166

7.2 TensorFlow 171

7.3 Caffe 176

7.4 小结 177

第8章 未来展望 179

8.1 深度学习的爆炸新闻 179

8.2 下一步的展望 181

8.3 对深度学习有用的新闻资源 ... 185

8.4 小结 187

深度学习概述

人工智能（Artificial Intelligence, AI）可能是你最近屡屡听闻的一个术语。人工智能正在成为热议的话题，无论是在学术社区，还是在商业领域，人们都接踵而来。大型的高科技公司，譬如谷歌（Google）和脸谱（Facebook）都在积极寻求收购人工智能领域的初创企业。随着海量资金流入人工智能领域，这个领域的并购最近也变得异常活跃。日本信息技术及移动运营商软银在 2014 年 6 月发布了一款名为“胡椒”（Pepper）的机器人——它能够理解人类的情感，一年之后他们已经着手准备向普通消费者销售“胡椒”这款机器人了。毫无疑问，这是人工智能领域的重要进展。

人工智能的概念已经存在几十年。可是，为什么它最近突然变得如此火热？原因之一就是近期推动相关人工智能领域发展的深度学习——常被称为“人工智能”。深度学习这个词在几乎所有的场合都与人工智能同时被提及。深度学习华丽登场之后，它的技术能力也开始爆炸性地急速成长，人们开始相信人工智能在将来的某一天会变成现实。听起来深度学习是我们必须要了解的一种技术。那么，到底什么是深度学习呢？

为了回答前面的问题，我们将在本章沿着人工智能的历史轨迹及研究领域，探讨人工智能流行背后的秘密。本章涉及的内容包括下面几个方面：

- 传统人工智能的方法和技术。
- 机器学习及其演变为深度学习的介绍。
- 深度学习及最近的使用案例的介绍。

如果你已经了解了什么是深度学习，或者你想查看深度学习的某个具体算法、实现技术，你可以跳过本章的内容，直接进入第 2 章。

虽然深度学习是一种创新的技术，但它并非想象中那么复杂。实际上，它非常简

单。读完这本书，你会发现它有多么强大。我衷心地希望本书能帮助你理解深度学习，让深度学习为你的研究和业务添砖加瓦。

1.1 人工智能的变迁

那么，为什么深度学习突然站到了聚光灯下？你可能会问这个问题，特别是如果你熟悉机器学习的话，因为深度学习与机器学习算法并没有那么大的差异（别担心，如果你对此了解不多，我们在本书后面会逐一介绍相关的内容）。实际上，我们可以说深度学习是神经网络（这是一种机器学习算法）的改进版，神经网络试图模拟人类大脑的结构。不过，深度学习能取得的成绩要深远得多，并且它也不同于任何一种机器学习算法（包括神经网络）。如果你了解深度学习经历过什么样的发展过程，就能对深度学习本身有更深入的理解。既然如此，就让我们开始浏览人工智能的变迁史。这部分的内容很轻松，你可以品着咖啡轻松地快速翻过。

1.1.1 人工智能的定义

突然之间，人工智能在全世界变得如此热门。但是，真正的“人工智能”事实上还不存在。尽管相关的研究取得了进展，但要达到真正的人工智能尚需时日。无论你高兴与否，人类的大脑，即被我们称之为“智能”的东西，其结构异常复杂，想要复制它并没有那么容易。

然而，我们却经常看到“某某产品用人工智能”的广告，难道他们是在忽悠吗？实际上说，是的！奇怪吗？你可能看到类似“用人工智能的推荐系统”或者“用人工智能驱动的产品”，可是这些句子中的“人工智能”并不符合“人工智能”的定义。严格地说，只是“人工智能”这个词的概念过分外延了。其实人工智能的研究及技术成果只达到其真正内涵的一部分，而人们使用“人工智能”这个词也恰恰意指这个部分。

让我们看几个例子。粗略划分起来，我们通常可以将人工智能划分为三大类，分别是：

- 简单重复性的机器运动，这些运动是由我们提前通过程序设定的。譬如可以进行高速作业的工业机器人，它们只能处理相同的工作。
- 依据人为设定的规则，针对特定的任务执行搜索或猜测结果工作。譬如，iRobot

Roomba 扫地机器人可以依据不断地碰撞障碍物判断出房间的形状，完成房间的清扫工作。

- 依据从现有数据中找出的度量规则（Measurable Regularity）为未知数据提供答案。譬如，依据用户购买历史的产品推荐系统，或者依据某个分类对广告网络中的横幅广告（Banner Ad）进行分发。

人们用人工智能这个词描述这些类别，更不用说使用深度学习的新技术，它也被称之为人工智能。虽然它们之间在形式和内容上都是截然不同的。那么，到底哪一种我们应该专属地称之为“人工智能”呢？非常不幸，针对这个问题，人们持有不同的观点，因此我们也无法给出一个客观的答案。从学术的角度出发，依据机器能达到的层次，我们可以用“强人工智能（Strong AI）”或者“弱人工智能（Weak AI）”加以描述。为了避免概念不清的问题，我们在本书中所特指的人工智能是类似于人类的智能，而这种智能却很难判定是否由人类产生。人工智能的领域正在迅猛发展着，随着深度学习的驱动，人工智能成为现实的可能性也以指数方式增长。这个领域比之前任何一个历史时刻都生机勃勃。这一波潮流会持续多久取决于未来的研究成果。

1.1.2 人工智能曾经的辉煌

人工智能这一话题最近突然变成热点，不过，这并非它第一次如此引人注目。回顾人工智能的历史，我们发现这一领域的研究已经开展了好几十年，这其中的兴衰往复也历经好几轮。最近的这一波浪潮已经是它的第三次高峰了。由此，有些人可能会想，这次的繁荣是否又仅仅是昙花一现。

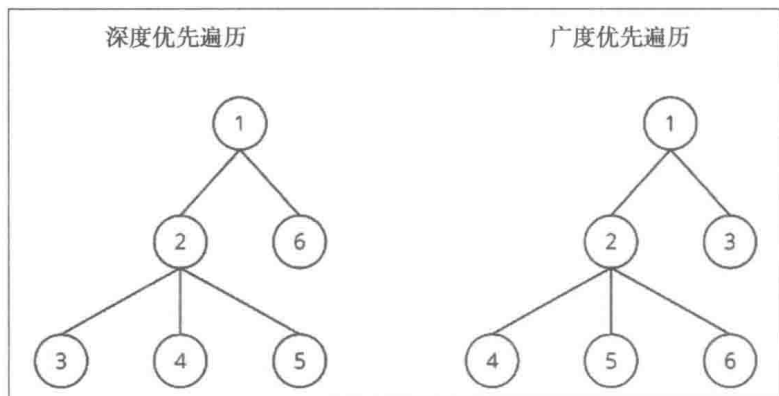
不过，最近的这波浪潮跟以往的历次有着巨大的不同，那正是深度学习。深度学习已经实现了过去技术实现不了的东西。那到底什么是深度学习呢？简而言之，就是机器可以从提供的数据找出特征量，并进行学习。我们可以预见在不远的将来人工智能极可能变为现实，因为迄今机器还无法凭借自身理解任何新的概念，我们需要按照人工智能以往的技术替它预先设定某些定量的特性它才能够理解，而这一新的成就将打破这一屏障。

读到这里，你可能会觉得这看起来并没有什么不同，但二者的差异就如天上人间，完全是两个世界。从一开始到机器有能力依据自身的判断度量特征量（Feature Quantity），经历了一个漫长的历程。凭借深度学习，人们在机器智能的道路上终于有了很大的进步。那么，以往技术和深度学习之间的巨大区别到底在哪里呢？为了

解释这个问题，首先让我们简略地回顾下历史上人工智能领域都发生了什么，了解这些有助于我们看清它们之间的差异到底是什么。

机器学习的第一波浪潮发生在20世纪50年代晚期。那个时候，主流搜索以及搜索应用程序的开发都基于固定的规则——不言自明，这些规则都是经由人工定义的。简而言之，搜索就是做分类。这种方式下，如果我们想要机器做任何形式的搜索，必须预先定义好每一种在处理过程中可能出现的模式。跟人比起来，机器的计算速度要快得多。无论模式数量有多庞大，机器总是能轻松地搞定它们。机器可以持续不断地进行数以百万计的搜索，最终找出最佳答案。然而，虽然机器可以高速地进行计算，但如果它仅仅是毫无目的随机搜索，也会空耗大量的时间。显然，我们不应该忽略时间这个重要的约束条件。因此，更进一步的研究都围绕着如何更高效地进行搜索展开。在所有的这些研究中，最广为人知的是“深度优先遍历（DFS）”和“广度优先遍历（BFS）”。

考虑每种可能的模式，搜索最有效的路径，从而在有限的时间内做出最好的选择。通过这种方式，你可以每次都获得最优的答案。基于这样的假设，深度优先遍历与广度优先遍历这两种图数据结构的搜索或遍历算法应运而生。这两种遍历的起始点都是树或者图的根节点，深度优先遍历在回溯之前总是沿着某一支尽可能地访问更多的节点，而广度优先遍历则会首先访问它自己的所有邻节点，之后才切换到下一层的邻节点。下面是一些实例图，通过它们我们能比较形象地了解深度优先遍历和广度优先遍历之间的差异：



在特定的领域，这些搜索算法能取得不错的效果，特别是像国际象棋或者日本将棋这样的领域。棋牌类游戏是机器学习尤其擅长的领域。如果提前给定大量关于输赢模式的输入，或者以往的比赛历史数据，以及所有允许的下棋操作，机器就能对场上

的局势进行评估，从非常海量的模式中选出下一子最优的移动。

如果你对这一领域感兴趣，我们可以进一步探究下机器到底是如何下棋的。譬如，机器的第一步移动是“白子”，那么下一步关于“白子”和“黑子”的移动就各有 20 种可能性。还记得我们前面图示介绍的树形模型吗？按照这个模型，从树的顶部作为游戏开始，它有 20 个分支分别表示“白子”下一步可能的移动。这 20 个分支中的每一个分支又带有 20 个分支，表示“黑子”下一步可能的移动，以此类推。在这个例子里，依据“白子”的移动情况，“黑子”在这棵树上会有 $20 \times 20 = 400$ 个分支，“白子”有 $400 \times 20 = 8000$ 个分支，接着“黑子”又有 $8000 \times 20 = 160\,000$ 个分支，依次这样计算下去……感兴趣的话，你可以随意推算下这个数值。

机器生成这棵树，并依据分支对棋盘上的各种可能情况做出预测，立刻就能给出最好的后续落子安排。这个过程有多深入（即分析过程会生成多少层的树结构，并以此进行评估）是由机器的计算速度决定的。当然，每一步的落子情况也应该加以考虑，并且应该内置于程序之中，这样一来下棋程序就不像我们想象中的那么简单了，不过本书不会就此展开深入详细的讨论。正如你所看到的那样，在下棋这件事上，机器打败人并不是让人吃惊的一件事。机器可以同时评估计算海量的模式，处理时间也比人短得多。机器打败国际象棋冠军并非新闻，事实上机器已经在比赛中打败过人类。由于种种这样的故事，人们对人工智能期望很高，希望它能变为现实。

不幸的是，现实总是不尽如人意。我们发现将搜索算法应用于实际还有一面难于逾越的高墙。如你所知，现实是非常复杂的。机器擅长的是按照固定的规则集，高速地处理事务；但如果仅仅交给它一个任务，它无法凭借自身的能力找到该应用什么规则，采取怎样的行动。无论何时，人类在行动的时候，下意识地会做各种评估，丢弃大量无关的因素或选项，从现实世界数百万计的事件（模式）中进行选择，做出决定。而机器无法像人类那样进行下意识的决策。如果我们试图创造出能正确考虑现实世界情况的机器，我们可以假设两种可能：

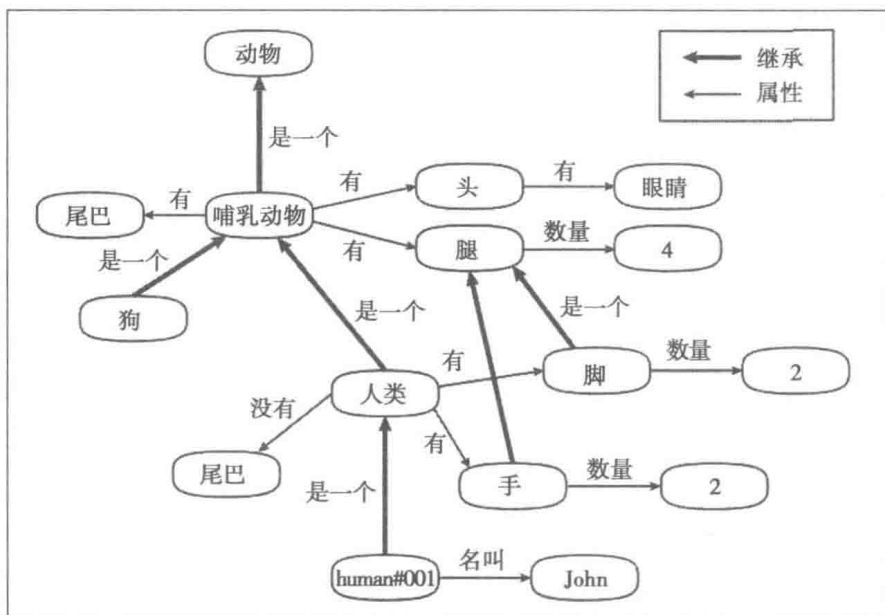
- 机器试图完成一个任务或目标，但不考虑接下来发生的事件或者可能性。
- 机器试图完成一个任务或目标，但不考虑不相关的事件或者可能性。

无论这两种机器中的哪一种，在完成人们给它们的任务之前都会僵死，或者在处理中迷失；尤其是后者，还没有开始它第一步的动作可能就已经僵住了。原因是这类元素的数量几乎是无限的，如果它试图以这种无限模式的方式思考或者搜索的话，机器无法在一个现实的时间里将它们区分开来。这个问题就是人工智能领域非常著名的

挑战之一，也被称为“框架问题（Frame Problem）”。

机器在国际象棋或者日本将棋上能取得巨大的成功原因在于搜索空间（Searching Space），即机器要在什么空间内进行处理，预先定义就是有限的（控制在一定的范围内）。你无法写出海量的模式，因此无法定义什么是最好的解决方案。即使你强制限制了模式的数量，或者定义了一个优化的解决方案，由于方案所需要的巨量计算，你也无法在一个经济的时间范围内得到结果。总之，那个时候的研究中，机器仅仅只能遵循人们定义的详细规则进行计算。因此，虽然这种搜索能在特定领域取得成功，它离真实的人工智能的距离依然比较遥远。因此，第一次人工智能的浪潮伴随着人们的失望迅速地平静下来。

第一波人工智能的浪潮随风逝去，不过，人工智能领域的研究还在继续。接下来的20世纪80年代迎来了人工智能的第二波高潮。这一次，名为“知识表示（Knowledge Representation, KR）”的运动引领着潮流。知识表示试图用便于机器理解的方式描述知识。如果世界上所有的知识都整合到计算机中，并且计算机能够理解这些知识，那么即使给它一个复杂的任务，它也应该能够提供正确的答案。基于这一假设，人们研发了各种方法，对知识进行组织，便于机器更好地理解。譬如，以结构化的方式构建网页——语义网，就是这些方式中的一个例子，试图让机器更容易地理解信息。下面是一个例子，展示了语义网是如何使用知识表示的方式描述信息的：



让机器获取知识并不像人们给机器发布指令告诉它如何干活那么片面，更像是让机器有能力响应人们提出的问题，并给出回答。如何将这一技术应用到现实生活的一个简单的例子是“积极 - 消极分析 (Positive-Negative Analysis)”，它是情感分析的主题之一。如果你提前输入机器的数据为语句中的每一个词定义了“积极”或“消极”的语气（也被称之为“词典”），机器可以将语句和词典进行对照，判断整个语句是积极的还是消极的。

这一技术被用于社交网络或者博客发帖、回复的积极 - 消极分析。如果你询问机器“对这篇博客发帖的回复是积极的还是消极的？”，它就会依据现有的知识（词典）对文章的评论进行分析，以此为依据回答你的问题。自第一波人工智能的浪潮以来，机器仅能遵循人们设置的规则工作，第二波人工智能的浪潮的确展示出了新的进展。

通过将知识与机器整合，机器变得无比的强大。对实现人工智能而言，这个想法本身是个不错的主意；不过，要真正实现它前面还隔着两道高墙。首先，正如你可能已经意识到的那样，将所有真实世界中的知识都输入到计算机中需要耗费大量的工作，而现在由于互联网的普遍使用，我们可以直接从网络上获取大量的开放数据。回到当时那个时代，要收集数以百万计的数据，再进行分析，将这些知识导入到计算机是一个几乎不现实的任务。实际上，将所有世界上的数据容纳到数据库中的工作一直在进行着，这就是著名的 Cyc 项目 (<http://www.cyc.com/>)。Cyc 的最终目标是依据知识数据库构建一个名为“知识库 (Knowledge Base)”的推理引擎。下面是一个使用 Cyc 项目知识表示的例子：

```
(#$isa #BarackObama #UnitedStatesPresident)
    "Barack Obama belongs to the collection of U.S. presidents."

($genls #Tree-ThePlant #Plant)
    "All trees are plants."

($capitalCity #Japan #Tokyo)
    "Tokyo is the capital of Japan."
```

其次，这种方法中，机器并没有真正理解知识的实际含义。即便知识已经进行了结构化、系统化，机器也仅能将它作为一个符号加以标识，却并未理解其背后的概念。毕竟，知识是由人输入的，机器所执行的仅仅是依据字典对数据进行比较，并进而猜

测它的含义。举个例子，你知道“苹果”和“绿色”的概念，我们从小被教育“绿苹果 = 苹果 + 绿色”，然后你自然一眼就能理解“一个绿苹果就是一个绿色的苹果”，然而机器并没有这种能力。这被称为“符号关联问题（Symbol Grounding Problem）”，它跟框架问题一样，也是人工智能领域的一大难题。

这是个好点子，它也的确改进了人工智能。不过，这种方法无法在现实中实现人工智能，因为它无法凭空创造人工智能。所以，人工智能的第二波浪潮也不知不觉渐渐冷却下来，由于无法达到人们对人工智能的预期，谈论人工智能的人也越来越少。当被问到“我们能否真正实现人工智能”这个问题时，回答“不能”的人逐渐增多。

1.1.3 机器学习的演化

为了找到实现人工智能的方法人们经历了一段异常艰难的岁月，终于找到一种完全不同的方式，稳健地构建出一种通用技术。这种方式的名称是“机器学习”。如果你曾经涉足数据挖掘领域，应该对这个名字耳熟能详。相对于人工智能以往的种种方法，机器学习要强大得多，是一种潜力无穷的工具，正如本章前文所介绍过的，以往的方法仅能依据人预先提供的知识进行搜索和判断，机器学习要高级得多。机器学习出现之前，机器仅能在已输入的数据中搜寻答案。大家的关注点都在机器如何能更快地从已有的知识中抽离出相关问题的知识。这样，机器就能更快地回答一个它已知的问题，但是，一旦碰到它未知的问题，它就不行了。

另一方面，机器学习领域中，机器的学习是照本宣科的。机器可以依据它学习的知识回答未知的问题。那么，机器是如何学习的呢？这里的“学习”到底是什么意思？简单地说，学习就是机器获得能够将问题划分成“是”或“不是”能力的过程。本章接下来的内容中我们会提供更进一步的细节，我们现在所能说的是机器学习就是一种模式识别的方法。

我们认为，从根本上说，这个世界上的每个问题都可以用答案为“是”或者“否”的问题所替换。举个例子，“你喜欢什么颜色？”这个问题，可以通过“你喜欢红色吗？喜欢绿色吗？喜欢蓝色吗？喜欢黄色吗？……”这样的问题所替代。机器学习中，使用高速计算和处理能力作为武器，机器用大量的训练数据，将复杂的问题替换为答案为“是/否”类型的问题，找出哪些数据答案为“是”，哪些数据答案为“否”的规律（换句话说，它在学习）。之后，使用学习的结果，机器可以

对新提供的数据进行分析，判断它们的结果为“是”还是“否”，并返回答案结果。概括来说，机器学习可以通过辨识和归类给定数据的模式，来回答未知数据的问题。

实际上，这种方法并没有想象中那么难。人们也经常无意识地对数据进行模式分类。譬如，如果你在一个聚会上碰到了你感兴趣的男人/女人，你可能会非常急切地想要了解你面前的他/她是否对你有同样的感觉。在你的脑海里，你会对他/她的说话方式、样貌、面部表情或者姿态进行分析，与你之前的经历（也就是数据）进行比较，通过这些进一步决定是否要去约会。这和基于模式识别的推断是同一个道理。

机器学习是一种以机械的方式，由机器主导而非人主导，进行模式识别的方法。那么，机器是如何识别模式，并对它们进行分类的呢？机器学习的分类标准是一种基于数学公式的推算，名叫“概率统计模型（Probabilistic Statistical Model）”。这种方式基于多种数学模型，已经被研究得非常透彻了。

学习，换句话说，就是模型参数的调整，一旦学习完成，就意味着模型构建完成。接下来，机器就可以将未知数据划分到最可能的模式中（即最适合的模式中）。按照数学对数据进行分类是一个重大优点。对人类而言，我们几乎无法对多维的数据，或者多模式的数据进行处理，而机器学习却可以使用几乎同样的数值公式完成分类。机器需要的仅仅是增加一个向量，或者矩阵中的维度数（本质上说，进行多维分类时，它并不是由分类直线或者分类曲线完成，而是由超平面完成的）。

发明这一方法之前，机器在没有人为帮助时几乎没有任何能力处理未知数据；通过机器学习，机器甚至能处理即使人也无法处理的数据。研究人员为机器学习带来的可能性欢欣雀跃，积极地投身到改善机器学习的工作中。机器学习概念的历史悠久，不过由于科学家们缺乏足够的数据库，长期以来无法进行大量的研究，证明它的有效性。不过，最近很多开源数据出现在互联网上，研究人员能比较容易地利用这些数据，对他们的算法进行实验。由此，人工智能的第三波浪潮随之而来。机器学习周边的环境也给了它极大助力。机器学习在能正确地识别模式之前需要学习大量的数据。除此之外，它还需要有能力处理这些数据。它处理的数据和模式类型越多，数据的数量以及计算的次数也越大。因此，很明显，之前的技术无法支持机器学习的发展。

不过，时代在进步，机器的处理能力得到了大幅增强。除此之外，网络也日益成