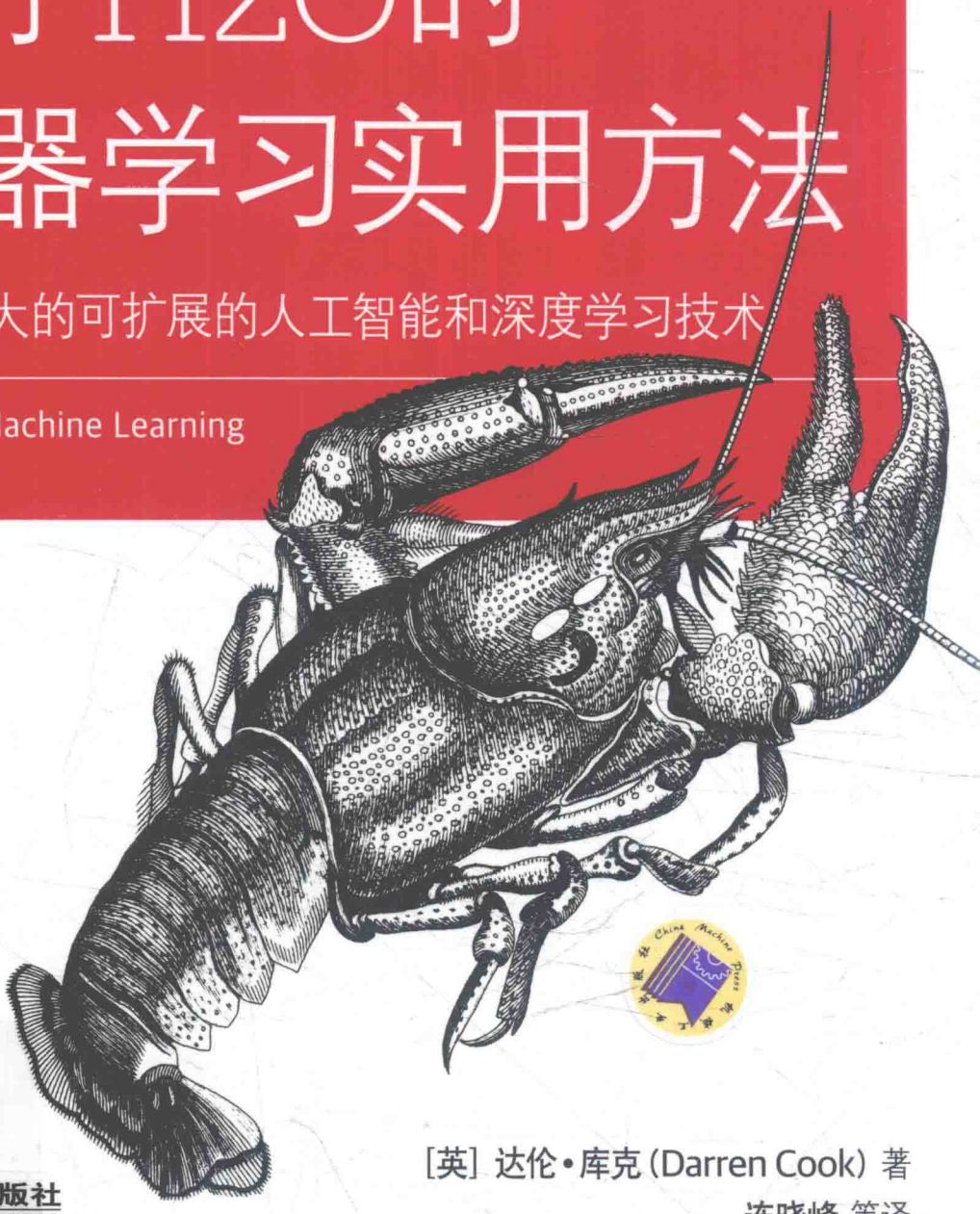
O'REILLY®

深度学习系列

基于H2O的机器学习实用方法

一种强大的可扩展的人工智能和深度学习技术

Practical Machine Learning with H2O



[英] 达伦·库克 (Darren Cook) 著

连晓峰 等译

机械工业出版社
CHINA MACHINE PRESS

深度学习系列

基于 H2O 的机器学习实用方法：一种强大的可扩展的人工智能和深度学习技术

[英] 达伦 · 库克（Darren Cook）著
连晓峰　等译

Beijing · Boston · Farnham · Sebastopol · Tokyo　O'REILLY®

O'Reilly Media, Inc. 授权机械工业出版社出版

机械工业出版社
CHINA MACHINE PRESS

本书主要介绍了H2O的基本概念和应用。全书共11章，首先介绍了H2O在R和Python下的安装和启动、数据导入/导出和操作以及本书所用的三种不同示例数据集和常用的模型参数。然后分别介绍了随机森林、梯度推进机、线性模型、深度学习和无监督式学习等算法在三种不同数据集中的应用，分析对比了默认算法和改进算法的性能。另外，还讨论了相关其他内容。

本书可作为从事机器学习、深度学习、人工智能等领域工作的工程技术人员的参考书，也可用于高等院校相关专业本科生、研究生以及教师的参考用书。

Practical Machine Learning with H2O:Powerful, Scalable Techniques for Deep Learning and AI/ by Darren Cook/ ISBN: 978-1-491-96460-6

北京市版权局著作权合同登记　图字：01-2018-0805 号。

图书在版编目（CIP）数据

基于 H2O 的机器学习实用方法：一种强大的可扩展的人工智能和深度学习技术 /（英）达伦·库克（Darren Cook）著；连晓峰等译 . —北京：机械工业出版社，2018.6

（深度学习系列）

书名原文：Practical Machine Learning with H2O:Powerful, Scalable Techniques for Deep Learning and AI

ISBN 978-7-111-60051-0

Ⅰ . ①基…　Ⅱ . ①达… ②连…　Ⅲ . ①机器学习　Ⅳ . ① TP181

中国版本图书馆 CIP 数据核字（2018）第 110106 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码 100037）
策划编辑：顾　谦　　责任编辑：顾　谦
责任校对：郑　婕　佟瑞鑫　封面设计：Karen Montgomery，张健
责任印制：常天培
北京圣夫亚美印刷有限公司印刷
2018 年 7 月第 1 版第 1 次印刷
184mm × 240mm · 13.75 印张 · 304 千字
0001—4000 册
标准书号：ISBN 978-7-111-60051-0
定价：69.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务		网络服务	
服务咨询热线：	010-88361066	机 工 官 网：	www.cmpbook.com
读者购书热线：	010-68326294	机 工 官 博：	weibo.com/cmp1952
	010-88379203	金　书　网：	www.golden-book.com
封面无防伪标均为盗版		教育服务网：	www.cmpedu.com

译者序

H2O 是一种可通过一个简单易用的开源框架来进行机器学习和数据分析的软件，并可支持广泛的操作系统和语言以及大数据应用。本书共分为 11 章，第 1 章介绍了如何在 R 和 Python 中安装 H2O 以及一些基本的概念和术语；第 2 章主要介绍了数据的导入 / 导出、如何在 R 和 Python 中加载数据和操作数据，其中包括数据汇总、行 / 列操作、数据索引和数据拆分、数据帧和模型文件；第 3 章主要介绍了三种示例数据集：建筑节能、手写体和足球比分，及其各自特点；第 4 章分析归纳了常用的模型参数，包括支持测度、回归指数、分类指数等，着重介绍了提前终止和交叉验证的作用；第 5 章介绍了随机森林算法在上述三种示例数据集中的应用，其中分别从默认算法和改进参数两个方面进行了比较分析；第 6 章介绍了梯度推进机（GBM）算法的特点，并在得到的三种不同的示例数据集中进行了实验比较；第 7 章则是在上述三种数据集中采用了线性模型（GLM）算法；第 8 章介绍了深度学习的基本概念、激活函数和各种参数，同样在三种数据集中进行了实验对比；第 9 章介绍了无监督学习算法，其中包括 k 均值聚类、深度学习自动编码器和层叠自动编码器；第 10 章讨论了相关的其他内容；第 11 章是对所有实验结果进行了详细分析和比较，分析了可能出现的问题及其解决办法。

全书语言精练、深入浅出，给出了丰富的具体实例，并进行了详细的对比分析。作者在数据处理与分析方面具有丰富的开发经验。

本书主要由连晓峰翻译，韩忠明进行了审校和整理。同时王伟、李东红、申震云、刘鹏华、于健、苏鹏飞、叶璐、王炎、贾涵、潘兵、毋冬、张云、郭朝晖、张晓伟等人也参与了部分内容的翻译。

由于译者的水平有限，书中不当或错误之处恳请各位业内专家学者和广大读者不吝赐教。

译　者

原书前言

经过漫长的发展，机器学习逐渐变得越来越成熟。它最早可以追溯到 20 世纪 50 年代，当时开发出第一个根据经验进行学习的程序（下跳棋），这也是第一个神经网络。人工智能研究人员多次提到这是一个当今早就不再关注的“指日可待”的重大突破。但也许研究人员是一直沿着正确的道路进行研究，或许只是需要一个更大数量级的处理能力，或一个微小的算法调整，就可以将毫无意义的研究变成富有成效并具有远大前途的工作。

在 20 世纪 90 年代早期，人们普遍将神经网络看作在人工智能方面取得的一项新的重大突破。当时本人曾多次尝试在计算机中实际应用神经网络，但在与利用特定领域的知识工程以及重新改进的树搜索方法所得到的结果相比时，发现其处理能力非常糟糕（结果仍是相当普通的）。而且其处理大规模数据的能力也很差。20 年后，当听说深度学习这一新概念在计算机领域取得了令人瞩目的成果时，本人感到非常困惑的是这与多年前所放弃研究的神经网络究竟有何不同。答案其实是“并没有太大不同”，或许只需要一个具有更强大处理能力（在这种情况下需提高 5~6 个数量级）的算法，就能取得丰硕成果。

H2O 是一种用于机器学习和数据分析的软件。怀着想要了解深度学习所具有的强大功能的这一想法，促使本人深入学习 H2O（尽管并不仅限于此，还包括搜索树、线性模型、无监督学习等），结果立刻就留下了深刻印象。H2O 具有下列特性：

- 开源（免费的 Apache 许可证）；
- 易于使用；
- 可扩展到大数据；
- 文档完备且具有商业支持；
- 第三版（即成熟架构）；
- 广泛的操作系统 / 语言支持。

H2O.ai（开发 H2O 的公司）聚集了高质量的优秀团队，将会发展得越来越好。在整个开发过程中，公司不仅始终坚持“如何才能使得 H2O 更好地发挥作用”，而且始终坚持“如何使其在大数据范畴下有效工作”的发展理念。

如果机器学习已完全成熟，那么 H2O 就好比既是一辆经济实用的家用车，同时也是一辆大载重量的货车。如果进一步深化比喻，本书将不仅介绍仪表盘控件所做的工作，而且还将讨论实现车辆从 A 到 B 的最佳方式。为了尽可能实用，本书只对学习算法相关的数学原理或理论进行最低程度的解释。

当然 H2O 也并非完美，以下是一些需要注意的问题。首先，H2O 不支持 GPU（它可实现深度学习，尤其是更快的学习）[⊖]。其次，集群都是针对低层（大数据），而不支持高层（复杂但相对较少的数据），因此对于后者，可能只需要一个单一的、快速的、有多个内核

⊖ Deep Water 是一个正在开发的新的 H2O 项目，可允许与其他深度学习库进行交互，因此 H2O 将会很快支持 GPU。

的机器。另外，集群也不具备高可用性（HA）。第三，H2O 是 Java 语言编译的，可以实现很好的优化，且 H2O 算法以执行速度快而著称，当然从理论上来说，更加优化的 C++ 语言可能会更快。第四，在 H2O 中不具有 SVM（状态向量机）算法。最后，H2O 还试图支持众多开发平台，因此每个平台都有一些简单的接口，有时会由于试图保持所有平台同步而导致开发过程较慢。

换言之，如果还是用车辆来比喻：一级方程式赛车会在直道上击败 H2O，但实际情况中尚未出现这种情况。

选择 H2O 的用户及原因

一些知名公司（http://www.h2o.ai/customers/）已正在利用 H2O 进行大数据处理，H2O 网站已声称目前有超过 5000 家公司在使用 H2O。同时，开发 H2O 的 H2O.ai 公司拥有 80 多名员工，其中一半以上是开发人员。

但这些都是给投资者留下深刻印象的统计数据，而对于开发人员毫无意义。对于那些感觉已掌握所有所需机器学习库的 R 和 Python 开发人员而言，H2O 所具有的主要优点是易用性和防止数据集过大而导致存储空间不足的有效扩展性。对于觉得已掌握 SparkML 的用户来说，H2O 算法数量较少，但运算速度明显更快。另外 H2O 还有一个优点，就是智能默认值意味着代码更加紧凑且宜于阅读：直接利用一行语句就可以得到一个性能良好的最先进的深度学习模型。本书的目的之一是介绍如何调整模型，但正如在书中所见，有时会由于无法比默认模型更好而不得不放弃调节。

关于读者

在基于一些假设条件的基础上，将本书控制在 1000 页以内。首先假设读者已熟悉 R 或 Python 语言。由于不会用到高级语言特性，因此具备任何编程语言的能力都足以理解书中内容，但本书中的示例仅提供了 R 和 Python 这两种语言。由于熟悉 pandas，Python 用户可能更具优势，尤其是因为 pandas 会使得所有的数据科学更加容易。

另外，还假设有一些心理暗示：为重复保存每个示例两次：希望 R 用户能够理解 Python 示例中所发生的，而 Python 用户可以理解 R 示例。对于 R 用户来说，这些 Python 示例是一个好的开始（对于 R 用户也是如此）。

即使使用电子表格或 SQL 表，也需要具有一些数据处理的经验。在此，假设读者对机器学习和人工智能，及其在公共基础设施中越来越多的应用具有一些正确认识。也许读者阅读本书的目的是想实现部分功能，并确保这些转变是出于伦理道德，同时让所有人受益，不论种族、性别、国籍或信仰。如果的确如此，向您致敬。

另外，还假设读者了解一些统计知识。这并没什么可担心的，本书的书名中强调了实用性，尽量将机器学习算法的相关理论降低到只需了解如何进行调节（而不是能够从头开

始实现）。若想了解更多知识，可以利用维基百科或搜索引擎。但至少应该了解模式中中位数的含义，知道什么是标准差和正态分布。

但更重要的是，希望读者能够了解统计存在误导，且机器学习可能会过拟合（https://en.wikipedia.org/wiki/Overfitting）。当提到 p=0.05 对一个实验具有显著意义时，这意味着每进行 20 次该实验，可能会有一次是错误的。体会显著的最明显时刻是在 xkcd（https://xkcd.com/882/）上。

在谈到计时的某些时候，这是介绍“我的机器”的一个好机会。书中所用的是一个中端便携式计算机，有一些年份了，具有 8GB 内存、四核、8 个超线程。但已足以运行书中的所有示例，实际上 4GB 的系统内存就足够了。然而，对于某些网格搜索（在第 5 章中介绍），“假设”不能在本机运行而在云平台启动集群（第 10 章简单介绍了“聚类”概念）。这样做只是出于实用性考虑：并不想等待 24h 后完成一个实验，然后再写出来。

本书的排版约定

本书通过不同排版风格表达内容，具体说明如下：

斜体字（Italic）

表示新术语、文件名和文件扩展名。

等宽字体（Constant width）

用于程序清单，以及在段落中引用的程序元素。如变量或函数名、数据库、数据类型、环境变量、语句和关键字。

等宽粗体字（**Constant width**）

显示应由用户逐字输入的命令或其他文本。

等宽斜体字（*Constant width*）

显示由用户提供的值或由上下文确定的值替代的文本。

表示提示或建议

表示一般注释

表示警告或注意事项

代码示例使用

https://github.com/DarrenCook/h2o/ 提供了可供下载的补充材料（代码示例、练习等）。

本书用于帮助实现所需工作。一般来说，本书提供了示例代码，可以在程序和文档中使用。不需要与作者联系以获得许可，除非复制了代码的主要部分。例如，在编写一个使用了本书中几个代码段的程序时，不需要任何权限。但销售或分销 O'Reilly 图书中的光盘实例需要授权。引用本书或引用示例代码来回答问题不需要权限。但将本书中的大量示例代码导入到产品文档需要授权。

非常感谢，但请不必，这归于属性。属性通常包括标题、作者、出版社和 ISBN。例如本书（O'Reilly）。版权所有 ©2017 Darren Cook，978-1-491-96469-6"。

如果感觉所使用的示例未合理使用或达到上述许可，请随时与 O'Reilly 出版社联系：permissions@oreilly.com。

如何联系我们

敬请对本书提出宝贵的建议和意见：

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway Noth
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室（100035）
奥莱利技术咨询（北京）有限公司

针对本书，提供了一个网页，其中列出了勘误表、示例和任何附加信息。可访问 http://bit.ly/practical-machine-learning-with-h2o。

关于本书的任何评论或技术咨询，请发邮件到 bookquestions@oreilly.com。

有关图书、课程、会议和新闻的更多信息，请参见网站 http://www.oreilly.com。

Facebook 地址：http://facebook.com/oreilly。

Twitter 地址：http://twitter.com/oreillymedia。

YouTube 地址：http://www.youtube.com/oreillymedia。

原书致谢

首先要感谢技术顾问：谢谢对本书的肯定，真心感谢。同时也对本书的错误抱歉。感谢 Katharine Jarmul、Yulin Zhuang、Hugo Mathien、Erin LeDell、Tom Kraljevic。如果没有采纳他们所提出的建议或包括了一些调侃，敬请谅解。除此之外，还感谢 Erin 和 Tom，他

们为本人提供了非常有用的 H2O.ai 问题的解答。另外还要非常感谢 Arno Candel、Tomas Nykodym、Michal Kurka、Navdeep Gill、SriSatish Ambati、Lauren DiPerna 以及所有在此未列出的人员（抱歉!）

感谢 Nicole Tache 作为本书前半部分的编辑，以及 Debbie Hardin 在 Nicole 决定生孩子而离开本项目后接手本书其余部分的编辑。感谢你们保持冷静，全神贯注于本书建模而忘记截止日期。

感谢家人默默忍受本人写书的漫长时间。

最后，感谢所有人：在 StackOverflow、博客文章、视频教程、写书、维护维基百科准确地回答问题的人。他们大量的工作，弥补了作者的知识漏洞。这为本人提供了一个完整的循环：不要犹豫，让我知道书中的所有错误。或者简单地说，如何能做得更好。

目 录

第 1 章
安装和快速启动

当得知 H2O 非常易于安装时或许是非常开心的。在此，将首先介绍如何利用 CRAN 在 R 中安装 H2O，然后介绍如何利用 pip 在 Python 中安装[⊖]。

之后，将深入讨论第一个机器学习项目：加载数据、建模、预测并评估是否成功实现。那时就可以向家人、朋友甚至是公交车上坐在旁边的陌生人吹嘘：在深度学习以及其他相关领域您是一位专家。

在经过深入分析随机元素是如何导致误入歧途之后，本章将以一个与 H2O 相关的 Web 接口（Flow）示例结束。

1.1 安装准备

本书中的示例都是在 R 和 Python 中运行的，所以需要先安装其中一种。另外，还需要安装 Java。如果已选定某种编程语言，建议都采用 64 位版本，包括操作系统（在下载页面，64 位的版本通常标记为“x64”，而 32 位的版本标记为“x86”）。

这时或许想知道选择 R 或 Python 有何区别？没有任何区别，稍后将会解释为什么。与利用人机交互友好的工具（如 Jupyter 或 RStudio）相比，采用脚本技术在性能上也并没有什么优势。

1.1.1 安装 R

通过各种 Linux 操作系统发行版的软件包管理器，可以很容易地执行 R 的安装：在 Debian/Ubuntu/Mint 等版本中执行 sudo apt-get install r-base，在 Red-Hat/Fedora/Centos 等版本中执行 sudo yum install R。

Mac 操作系统用户可以访问 https://cran.r-project.org/bin/macosx/，并按照指示进行安装。

对于 Windows 操作系统用户，可以访问 http://cran.rstudio.com/bin/windows/，下载并运行 .exe 文件，然后按照指示执行。在选择组件（Select Components）页面，可以选择安装 32 位和 64 位两种版本。本书选择只安装 64 位版本，当然完全可以安装两种版本。

安装 R 的另外一种可选方法是安装 RStudio，可以从命令行执行运行 H2O 所需的所有

⊖ 第 10 章介绍了安装 H2O 的一些其他方法（见“安装最新版本”），其中包括如何从数据源进行编译。如果在实际过程中遇到了仅在最新开发版本中才修复的 bug，或希望对 H2O 进行黑客攻击，那么可能希望采用这些安装方法。

命令，但通过 RStudio 可以更容易（特别是在 Windows 操作系统上，其中的命令行还停留在 1995 年）。如果选用 RStudio，可以访问 https://www.rstudio.com/products/rstudio/download/，下载并安装。

1.1.2 安装 Python

在 Python 2.7 或 Python 3.5 下，H2O 都能正常运行，对于本书的所有示例也都如此。如果使用的是 Python 的早期版本，那么可能需要升级。另外，还需要 Python 的软件包管理器——pip。

在 Linux 操作系统中，对于 Debian/Ubuntu/Mint 等操作系统，需要执行 sudo apt-get python-pip；如果是选择 Python 3，应执行 sudo apt-get python3-pip（Python 是 pip 的依赖项，因此通过安装 pip，也会得到 Python）。对于 RedHat/Fedora/Centos 等操作系统，最佳的命令形式会根据所用的版本不同而不同，因此需要查看最新的 Linux Python 指令（https://packaging.python.org/en/latest/install_requirements_linux/）。

Mac 操作系统用户，请参阅在 Macintosh 上使用 Python（http://bit.ly/2gn4HFs）。

Windows 操作系统用户，请参阅在 Windows 操作系统上使用 Python（http://bit.ly/2gn4HFs）。注意，应选择安装 64 位版本（除非仍继续使用 32 位版本的 Windows 操作系统）。

或许可能还想尝试 Anaconda（https://www.continuum.io/downloads）。这是一个 Python 发行版，其中包含了几乎所有需要的数据科学包。好处是可以以普通用户身份进行安装，这在没有 root 访问权限时非常有用。同时还提供了 Linux、Mac 和 Windows 版本。

1.1.3 隐私保护

H2O 中有一些在每次启动时均调用 Google 分析的代码[⊖]。这些代码是匿名执行的，只是用于检查所使用的版本，但如果不喜欢，或违反了公司的政策，可以在 home 目录（Windows 中的“C:\Users\YourName\”）中创建一个 .h2o_no_collect 的空文件来停止执行这些代码。如果在信息日志中查看“选择放弃发送使用测度”，即可了解上述代码的工作过程。另一种退出方式是按照第 10 章的“从命令行运行”中所给出的方法。

1.1.4 安装 Java

在此，还需要安装 Java，可以从 Java 下载页面（http://bit.ly/16mhImY）获得。选择 JDK[⊖]。如果认为已安装过 Java JDK，但不是很确定，可以先继续安装 H2O，如果提示出错，然后再返回来（重新）安装 Java。

例如，在已安装了 64 位 R 的 64 位 Windows 上安装 Java 时，首先执行 library(h2o)，这时提示已安装了 32 位的 JDK。因此需要下载最新版本的 JDK（http://bit.ly/16mhImY）。

⊖ 截至 2016 年 6 月，可在 http://bit.ly/2f96Hyu 中查看。

⊖ 尽管运行 H2O，也可以使用“Server JRE”或“JRE”，但建议还是安装 JDK。

安装完成后，再测试一下，发现一切正常。

1.2 利用 R（CRAN）安装 H2O

（如果不使用 R，则跳转到“利用 Python（pip）安装 H2O”）。

启动 R，输入 install.packages（“h2o”）。天哪，我的意思是这太容易安装了！该命令还负责处理任何依赖项。

如果是第一次使用CRAN[⊖]，那么会提示选择哪一个镜像站点。在此选择一个最接近的，或者也可以选择任何一个想要访问的站点，然后就一切 OK 了。

如果要在整个站点域上安装 H2O（即该机器上的所有用户均可使用 H2O），需要在根目录下运行 R、sudo R，然后输入 install.package（“h2o”）。

接下来，通过输入 library（h2o）来检查是否正常工作。如果一切正常，则继续下一步：h2o.init()。如果运气好，可以看到有关 H2O 启动的许多输出信息，以及关于集群的所有信息，如图 1-1 所示。如果运气不好，则错误信息会提示缺少哪些依赖项，或者存在哪些问题。

```
> h2o.init()

H2O is not running yet, starting it now...

Note:  In case of errors look at the following log files:
    /tmp/Rtmp6btQEF/h2o_darren_started_from_r.out
    /tmp/Rtmp6btQEF/h2o_darren_started_from_r.err

java version "1.7.0_101"
OpenJDK Runtime Environment (IcedTea 2.6.6) (7u101-2.6.6-0ubuntu0.14.04.1)
OpenJDK 64-Bit Server VM (build 24.95-b01, mixed mode)

Starting H2O JVM and connecting: ..... Connection successful!

R is connected to the H2O cluster:
    H2O cluster uptime:         3 seconds 535 milliseconds
    H2O cluster version:        3.8.2.2
    H2O cluster name:           H2O_started_from_R_darren_rge683
    H2O cluster total nodes:    1
    H2O cluster total memory:   1.71 GB
    H2O cluster total cores:    8
    H2O cluster allowed cores:  2
    H2O cluster healthy:        TRUE
    H2O Connection ip:          localhost
    H2O Connection port:        54321
    H2O Connection proxy:       NA
    R Version:                  R version 3.2.5 (2016-04-14)

Note:  As started, H2O is limited to the CRAN default of 2 CPUs.
       Shut down and restart H2O as shown below to use all your CPUs.
           > h2o.shutdown()
           > h2o.init(nthreads = -1)

>
```

图 1-1 运行 h2o.init()（在 R 中）

⊖ CRAN 是 R 的软件包管理器。更多详情，请参见 https://cran.r-project.org/。

接下来，回顾一下刚才的运行过程。工作一切正常，说明运气很好[⊖]。这值得好好庆贺一番。事实上，可以录一段视频，然后发布在社交媒体上，并特别注明正在阅读本书以及本书非常好。

CRAN 上的 H2O 版本可能滞后最新、最好的版本长达 1~2 个月。除非受到现已能够修复的 bug 的影响，否则不必担心。

默认情况下，h2o.init() 只会使用到机器的两个内核，以及 1/4 的系统存储空间[⊖]。另外，还可以通过 h2o.shutdown() 来判断 H2O 的作用。然后重新启动 H2O，而这次是使用所有内核：h2o.init(nthreads = -1)。还可以声明使用 4GB 存储空间以及所有内核：h2o.init(nthreads = -1, max_mem_size= “4g”)。

1.3 利用 Python（pip）安装 H2O

（如果对 Python 不感兴趣，可直接跳转到 “第一个学习示例”）。

在命令行中，输入 pip install –U h2o。好，就是这么简单。

-U 表明要升级任何依赖项。在 Linux 操作系统下，可能需要在根目录下执行，因此应输入 sudo pip install -U h2o。或者以一个本地用户身份来安装，即 pip install -U --user h2o。

为了进行测试，启动 Python，输入 import h2o，如果一切正常，接下来输入 h2o.init()。这时会滚动显示一些信息，最后是显示一个表格，其中包括可用的节点个数、存储空间总量和总的内核个数，如图 1-2 所示[⊖]（如果需要报告 bug，请确保包含表中所有信息）。

```
In [3]: import h2o

In [4]: h2o.init()
Checking whether there is an H2O instance running at http://localhost:54321..... not found.
Attempting to start a local H2O server...
  Java Version: java version "1.7.0_111"; OpenJDK Runtime Environment (IcedTea 2.6.7) (7u111-
ild 24.111-b01, mixed mode)
  Starting server from /usr/local/h2o_jar/h2o.jar
  Ice root: /tmp/tmpsNCJ_v
  JVM stdout: /tmp/tmpsNCJ_v/h2o_darren_started_from_python.out
  JVM stderr: /tmp/tmpsNCJ_v/h2o_darren_started_from_python.err
  Server is running at http://127.0.0.1:54321
Connecting to H2O server at http://127.0.0.1:54321... successful.
--------------------------  ------------------------------
H2O cluster uptime:         02 secs
H2O cluster version:        3.10.0.7
H2O cluster version age:    5 days
H2O cluster name:           H2O_from_python_darren_a9x6hb
H2O cluster total nodes:    1
H2O cluster free memory:    1.710 Gb
H2O cluster total cores:    8
H2O cluster allowed cores:  8
H2O cluster status:         accepting new members, healthy
H2O connection url:         http://127.0.0.1:54321
H2O connection proxy:
Python version:             2.7.6 final
--------------------------  ------------------------------
```

图 1-2 运行 h2o.init()（在 Python 中）

⊖ 正是这种推理能力才能区分普通人和数据科学分析人员。

⊖ 关于如何查阅 Java 的安装步骤以及获得系统默认值，请参见 http://bit.ly/2gn5h6e。

⊖ 或许会看到一些警告？通过 import warnings;warning.filterwarning(“ignore”,category=DeprecationWarning) 可在屏幕显示中忽略这些警告信息，但通常对这些警告视而不见。

如果的确显示了这样的一张表，就表明一切正常。

默认情况下，H2O 实例允许使用所有内核，且（通常）需要 25% 的系统存储空间。这通常已可以保证正常运行，但出于讨论的目的，分析一下如果提供了所有的 4GB 存储空间，且只使用 8 个内核中的两个会是什么情况？首先通过 h2o.shutdown() 关闭 H2O，然后输入 h2o.init(nthreads=2,max_mem_size=4)。那么根据信息表中的相关部分就可确定 H2O 的实际工作情况：

```
...
H2O cluster total free memory:  3.56 GB
H2O cluster total cores:        8
H2O cluster allowed cores:      2
...
```

在 H2O 中不能使用 virtualenv[⊖]。确切地说，可以安装，但不能启动 H2O。如果确实想这样安装，请按照第 10 章中启动 H2O 的命令行操作。包括 h2o.init()，以及所有一切操作，这样才能正常工作。

1.4 第一个学习示例

至此，一切都已经安装完毕，那么接下来就开始具体应用。Python 和 R 的 API 非常类似，可在本例中进行对比。如果使用 Python，请参考例 1-1；若使用 R，则参考例 1-2。重复执行之前运行的 import/library 和 h2o.init 代码，不必担心，这没有任何影响。

在此，将详细介绍运行过程，但需要强调的是这是一个完整的脚本。下载数据、准备数据、针对该数据集创建一个最先进的多层神经网络模型（即深度学习模型）进行竞争学习，然后利用该模型进行预测。

Iris 数据集

如果之前没有听说过 Iris 数据集，说明这一定是所阅读的第一本机器学习相关书籍！该数据集包括一组关于 iris 植物的 150 组观测值，其中包括 4 次测量（每个萼片和花瓣的长度和宽度）及其所属的物种。在此，共有 3 种物种，且每种物种具有 50 个观测值。

这是一种常用的机器学习实验数据集，特点是数据量很小，可以快速学习，同时可在一张图表中有效查看，但数据量又足以提供足够的感兴趣样本，且并不普通：四组测量值都无法完全有效区分数据。

⊖ 至少针对 3.10.x.x 版本及其更早的版本而言，不能使用 virtualenv。