

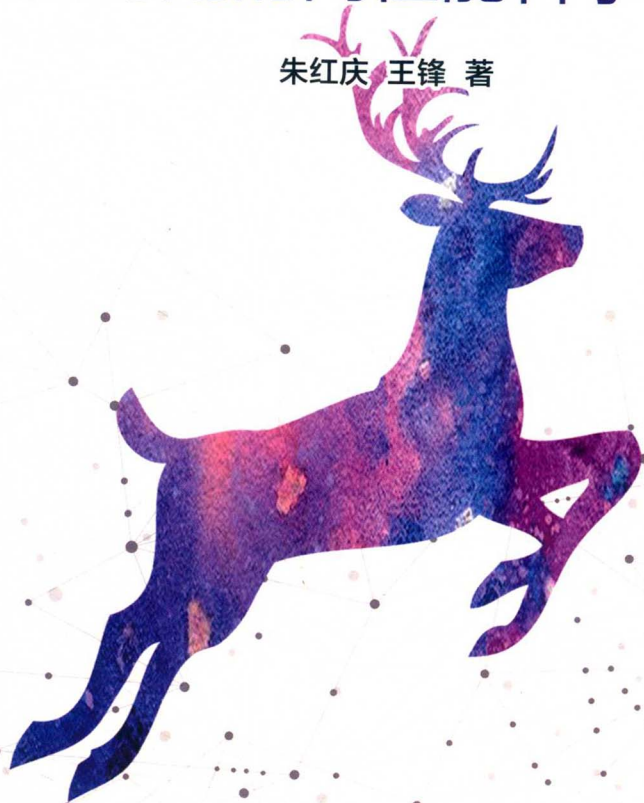
从零基础到具备核心编程技能，通过实践让编程变得更简单

Broadview[®]
www.broadview.com.cn

Julia机器学习核心编程

人人可用的高性能科学计算

朱红庆 王锋 著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Julia机器学习核心编程

人人可用的高性能科学计算

朱红庆 王锋 著



電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书以帮助读者学会、用好 Julia 语言为目标，深入浅出地讲解 Julia 语言的各项技术及实战技能。

全书共 12 章，首先讲解 Julia 语言的基本概念、编程模式、变量、数据类型、函数、程序控制结构等基础知识，然后介绍 Julia 语言的交互性与元编程、数值与科学计算、数据可视化和图像处理、数据库编程、Julia 编程的核心内幕等核心应用技能，再通过综合案例详细讲解 Julia 语言在商务网站开发方面的应用。书中融入了作者丰富的工作经验和多年的使用心得，还提供大量来自工作现场的实例，具有较强的实战性和可操作性。

本书适合没有 Julia 语言基础的初学者，也适合有一定 Julia 编程基础，但没有项目开发经验的人员，同时适合精通 Julia 编程的程序员阅读。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

Julia 机器学习核心编程：人人可用的高性能科学计算/朱红庆，王锋著. —北京：电子工业出版社，2019.9

ISBN 978-7-121-36883-7

I. ①J… II. ①朱… ②王… III. ①程序语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字（2019）第 120902 号

责任编辑：刘 伟

印 刷：北京季蜂印刷有限公司

装 订：北京季蜂印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：720×1000 1/16 印张：21 字数：381 千字

版 次：2019 年 9 月第 1 版

印 次：2019 年 9 月第 1 次印刷

定 价：89.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zltts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）51260888-819，faq@phei.com.cn。

前言

本书针对零基础和入门级读者，用实例引导读者深入学习，满足其在基础入门、扩展学习、职业技能、项目实战 4 个方面的需求。本书深入浅出地讲解 Julia 软件在编程中的核心要点及实战技能，系统学习后可以掌握 Julia 语言基础知识，提升在科学技术以及机器学习方面的编程能力，获取优良的团队协作技能和足够的项目实战经验。

本书内容

全书分为 12 章。从最基础的 Julia 软件的安装开始，系统地讲解 Julia 语言的基本概念、编程模式、变量、数据类型、函数、程序控制结构等基础知识；然后介绍 Julia 语言的交互性与元编程、数值与科学计算、数据可视化和图像处理、数据库编程、Julia 编程的核心内幕等核心运用技能；最后详细讲解 Julia 在商务网站创建方面的应用，借助 Julia 框架可以轻松地完成商务网站的创建。通过本书的学习，读者将对 Julia 在项目开发中的应用拥有切身的体会，为日后进行软件开发积累项目管理和实战开发经验。

全书不仅融入了作者丰富的工作经验和多年的使用心得，还提供了大量来自工作现场的实例，具有较强的实战性和可操作性。系统学习后，可以使初学者、应届毕业生快速成长为合格的初级程序员，通过实战积累项目开发经验和团队合作技能，在未来的职场中获取一个高的起点，并能迅速融入软件开发团队中。

本书特色

1. 结构科学、自学更易

本书在内容组织和范例设计上都充分考虑到初学者的特点，由浅入深，

循序渐进，无论你是否接触过 Julia 语言，都能从本书中找到最佳起点。

2. 视频讲解、细致透彻

为降低学习难度，提高学习效率，本书作者录制了同步微视频（模拟培训班模式），通过视频学习除能轻松学会专业知识外，还能获取到作者的软件开发经验，使学习变得更轻松、有趣。

3. 超多、实用、专业的范例和实战项目

本书结合实际工作中的范例逐一讲解 Julia 语言的各种知识和技术，并以两个项目的实战来总结本书内容，使读者在实战中掌握知识，轻松拥有项目开发经验。

超值助学资源

本书配备了极为丰富、超值的助学资源库（下载链接见文末），具体内容如下。

(1) 助学资源 1:【本书配套学习】资源库，提升读者学习 Julia 语言的效率。

- 免费赠送本书精选 130 余节教学微视频，总时长约 12 学时。
- 全书两个大型项目案例以及 260 余个范例源代码。
- 全书配套 PPT 电子教学课件。

(2) 助学资源 2:【职业成长】资源库，突破读者职业规划和职业发展弊端与瓶颈。

- 程序员职业规划手册、软件工程师技能手册。
- 面试（笔试）资源库：200 道求职常见面试（笔试）真题与解析。
- 常见错误及解决方案、开发经验及技巧大汇总。
- 200 套求职简历模板、200 套竞聘模板、200 套毕业答辩 PPT 模板。

(3) 助学资源 3:【Julia 语言学习】资源库，拓展学习本书的深度和广度。

- 电子书资源库：Julia 语言标准库函数查询手册、Julia 语言常用查询手册、Julia 经典范例源码解析、Julia 程序员职业规划、Julia 程序员面试技巧、Julia 常见错误及解决方案、Julia 开发经验及技巧大汇总等。

(4) 助学资源 4: 在线课程（VIP 会员），可免费学习 Julia、Java、JavaScript、C++、Oracle、iOS、Android 等 30 多种软件或系统开发在线课程，共计 500 余学时。

读者对象

- 没有任何 Julia 语言基础的初学者。
- 有一定的 Julia 编程基础，没有项目开发经验的人员。
- 正在进行毕业设计的学生。
- 大专院校及培训学校的老师和学生。

本书由朱红庆、王锋著，同时参与编写和资料收集的还有张雪年、石家欢、李晨、黄路路、常慧杰、张孝忠，在此一并表示感谢。

在本书编写过程中，虽然作者尽可能地将最好的讲解呈现给读者，但因水平所限，难免存在疏漏和不妥之处，敬请不吝指正。若读者在学习中遇到困难，或有任何建议，可发邮件至 elesite@163.com。

编者

说明：书中部分省略了大家熟知的“语言”二字，如 Julia、C、Python、Java 等，在本书中代表 Julia 语言、C 语言、Python 语言和 Java 语言等。

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- **下载资源：**本书提供资源文件，均可在 [下载资源](#) 处下载。
- **提交勘误：**您对书中内容的修改意见可在 [提交勘误](#) 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动：**在页面下方 [读者评论](#) 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/36883>



目 录

第 1 章 深入理解 Julia 语言的生态系统	1
1.1 Julia 为什么与众不同	1
1.2 Julia 的应用领域	3
1.2.1 数据可视化和绘图	3
1.2.2 构建、部署或嵌入代码	4
1.2.3 与数据互动	5
1.2.4 可扩展的机器学习	5
1.2.5 丰富的科学计算生态系统	6
1.2.6 并行和异构计算	6
1.3 安装 Julia 软件	7
1.3.1 在 Ubuntu 上安装 Julia	7
1.3.2 在 Fedora/CentOS/RHEL 等 Linux 发行版上安装 Julia	8
1.3.3 在 Windows 上安装 Julia	9
1.3.4 在 Mac 上安装 Julia	11
1.3.5 从源代码构建 Julia	11
1.4 Julia 在数据科学中的重要性	12
1.5 使用 REPL	14
1.5.1 在 Julia 中使用帮助	15
1.5.2 REPL 中的图表	17
1.6 使用 Jupyter Notebook	18
1.7 使用 Juno 编写 Julia	23
1.8 包管理	25
1.8.1 使用 <code>Pkg.status()</code> 显示所有已安装的包	25
1.8.2 使用 <code>Pkg.add()</code> 添加包	26
1.8.3 安装未注册包	27

1.8.4	使用 <code>Pkg.update()</code> 更新包	28
1.8.5	METADATA	28
1.8.6	开发包	28
1.8.7	创建一个新的包	29
1.9	多重分派	29
1.10	LLVM 和 JIT	31
1.11	本章小结	32
第 2 章	使用 Julia 进行快速编程	33
2.1	重温编程范式	33
2.1.1	命令式编程范式	34
2.1.2	逻辑式编程范式	34
2.1.3	函数式编程范式	35
2.1.4	面向对象的编程范式	36
2.1.5	开始 Julia REPL 编程	37
2.2	Julia 中的变量	38
2.2.1	命名约定	40
2.2.2	整数、位、字节和布尔数据类型	41
2.3	Julia 中的浮点数	43
2.4	浮点数的特殊符号	45
2.4.1	浮点数操作	46
2.4.2	任意精度的计算	46
2.4.3	使用系数编写表达式	47
2.5	Julia 中的算术和逻辑运算	48
2.5.1	执行算术运算	48
2.5.2	执行按位运算	48
2.5.3	自更新运算符	49
2.5.4	运算符的优先级	50
2.5.5	类型转换（数字）	51
2.6	数组和矩阵	51
2.6.1	Julia 中的列表解析式	53
2.6.2	矩阵运算	55
2.6.3	多维数组操作	58

2.6.4 稀疏矩阵	59
2.7 DataFrames	59
2.7.1 DataFrames 中的 NA 数据类型	59
2.7.2 DataFrame 表格	62
2.8 本章小结	63
第 3 章 Julia 函数	64
3.1 创建函数	64
3.2 函数参数	67
3.2.1 值传递和引用传递	68
3.2.2 分享传递	68
3.2.3 return 关键字	68
3.3 参数类型	69
3.3.1 无参函数	70
3.3.2 可变参数	70
3.3.3 可选参数	72
3.4 理解函数中变量的作用域	73
3.5 嵌套函数	76
3.6 匿名函数	78
3.7 多重分派	79
3.8 递归	83
3.9 内建函数	84
3.10 本章小结	92
第 4 章 Julia 数据类型	93
4.1 Julia 类型系统	93
4.1.1 静态类型语言与动态类型语言	94
4.1.2 整数类型	96
4.1.3 浮点类型	97
4.1.4 Char 类型	97
4.1.5 字符串类型	97
4.1.6 布尔类型	98
4.2 类型转换	98

4.3	用户自定义数据类型和复合类型	103
4.4	内部构造	106
4.5	模块和接口	108
4.5.1	在模块中包含文件	110
4.5.2	模块文件路径	112
4.6	模块预编译	113
4.7	多重分派与解释	114
4.8	本章小结	116
第 5 章	Julia 流程控制及异常处理	117
5.1	分支结构	117
5.1.1	简单条件语句	118
5.1.2	if 条件语句	119
5.1.3	循环结构	124
5.1.4	循环范围定义	128
5.1.5	break 和 continue	130
5.2	异常处理	131
5.2.1	throw()	133
5.2.2	error()	136
5.2.3	try-catch/try-finally 结构	137
5.3	Julia 中的 tasks	138
5.4	本章小结	140
第 6 章	Julia 互操作性和元编程	141
6.1	与操作系统交互	141
6.1.1	文件系统操作	142
6.1.2	I/O 操作	146
6.1.3	例证	147
6.2	调用 C 和 Python 语言	149
6.2.1	从 Julia 调用 C 语言	149
6.2.2	从 Julia 调用 Python	150
6.3	表达式和宏	152
6.4	元编程	157

6.5	内置宏	158
6.6	类型自省和反射能力	164
6.6.1	类型自省	164
6.6.2	反射能力	165
6.7	本章小结	167
第 7 章	用 Julia 进行数值科学计算	168
7.1	数据处理	168
7.1.1	读取文本文件	171
7.1.2	读取 CSV 格式文件	174
7.1.3	使用 DataFrames	177
7.1.4	NA	178
7.1.5	DataArray	180
7.1.6	DataFrame	181
7.2	线性代数与微分学	182
7.2.1	线性代数	182
7.2.2	微积分	184
7.3	统计数据	186
7.3.1	简单统计	186
7.3.2	在元编程中使用函数	188
7.3.3	使用 DataFrames 进行基本信息统计	189
7.3.4	使用 Pandas	190
7.3.5	高级统计	192
7.4	优化	198
7.4.1	JuMP	198
7.4.2	Convex	200
7.5	本章小结	202
第 8 章	Julia 数据可视化编程	203
8.1	基本图表	203
8.1.1	条形图	206
8.1.2	饼图	207
8.1.3	散点图	209

8.1.4	直方图	210
8.1.5	3D 图表	211
8.2	Vega 库	213
8.2.1	瀑布图	216
8.2.2	Aster 图	217
8.2.3	等值线图	218
8.2.4	面积图	219
8.2.5	箱线图	219
8.2.6	带状图	220
8.2.7	散点图	221
8.2.8	文字云	222
8.3	Gadfly 库	223
8.3.1	使用绘图函数与 Gadfly 库进行交互	224
8.3.2	使用 Gadfly 库绘制数据块	226
8.4	本章小结	230
第 9 章	Julia 数据库编程	231
9.1	如何连接数据库	231
9.2	关系数据库	232
9.2.1	SQLite	233
9.2.2	MySQL	233
9.3	NoSQL 数据库	235
9.4	REST 简介	238
9.4.1	JSON 简介	239
9.4.2	Web 框架	243
9.5	本章小结	247
第 10 章	Julia 的核心编程结构	248
10.1	Julia 的内部代码	248
10.1.1	FemtoLisp	248
10.1.2	Julia 核心 API	249
10.2	提升性能	249
10.2.1	全局变量	249

10.2.2	输入声明	250
10.2.3	其他技巧	252
10.3	标准库	253
10.4	理解 LLVM 和 JIT	256
10.5	并行计算	258
10.5.1	注意全局变量	260
10.5.2	并行循环	262
10.6	TCP 套接字和服务器	264
10.6.1	建立 TCP/IP 连接	264
10.6.2	socket 和 streams 模块	266
10.7	创建包	267
10.7.1	包命名指南	267
10.7.2	生成包	268
10.8	本章小结	269
第 11 章	创建 Web 图书商务网站	270
11.1	安装 Genie	270
11.2	使用 Genie 快速搭建服务器	271
11.3	创建 Genie 应用程序项目	275
11.4	Genie 的 MVC 结构	279
11.4.1	使用控制器	279
11.4.2	HTML 视图	282
11.4.3	使用布局	283
11.4.4	JSON 转换	285
11.5	使用 SearchLight 模组访问数据库	289
11.5.1	设置数据库连接	289
11.5.2	使用 SearchLight 迁移来管理数据库架构	290
11.5.3	创建图书的数据库表	291
11.5.4	编写表迁移文件	291
11.5.5	运行迁移文件	292
11.6	定义模型	293
11.6.1	使用模型	294
11.6.2	自动加载数据库配置	295

11.7 本章小结	297
第 12 章 Julia 机器学习框架	298
12.1 安装 Flux 框架	298
12.2 模型构建基础	299
12.2.1 简单模型	301
12.2.2 构建图层	302
12.2.3 堆叠	304
12.2.4 层助手	305
12.3 循环模型	305
12.4 状态模型	308
12.5 序列	309
12.6 梯度截断	310
12.7 正则化	311
12.7.1 基本层	312
12.7.2 循环层	313
12.7.3 激活功能	315
12.7.4 规范化和正规化	315
12.8 优化器	316
12.9 优化参考	318
12.10 机器学习训练	318
12.10.1 损失函数	318
12.10.2 数据集	319
12.10.3 回调	320
12.11 本章小结	320

深入理解 Julia 语言的生态系统

与其他现有的流行编程语言相比，Julia 是一种全新的编程语言。Julia 语言于 2012 年 2 月第一次向全世界公开并宣布开源。Julia 语言的起源最早可以追溯到 2009 年，当时三位开发人员，麻省理工学院 (MIT) 的 Viral Shah、Stefan Karpinski 和 Jeff Bezanson 在阿兰·埃德尔曼教授的监督下开发了这种语言。Julia 经过不断的发展，变得越来越有名，在 TIOBE 指数上排名不断提高，并在 Stack Overflow 上获得了支持。不少研究人员对 Julia 语言很感兴趣，特别是来自科学计算领域的研究人员。

本章重点知识：

- 为计算机配置 Julia 编程环境
- Julia 的特殊之处
- Julia 对数据科学的重要性
- 使用 REPL 和 Jupyter Notebook 进行编程
- 理解 LLVM 和 JIT

1.1 Julia 为什么与众不同

科学计算程序是所有程序中对计算性能要求最高的。多年来，科学界一直使用动态语言来构建应用程序，但是动态语言有一个致命的缺点，就是相

对于静态语言要慢得多。但是为什么在对速度要求如此高的科学计算领域，人们还是首选动态语言来进行编程呢？这样做的一个主要原因是，这些应用程序通常是由物理学家、生物学家、金融专家和其他领域专家开发的，尽管他们具有一定的编程经验，但毕竟不是经验丰富的开发人员，静态语言复杂的结构和语法对他们来说难以快速上手。

所以，相较于难以上手的静态语言，他们更偏爱动态语言；相较于更好的性能，他们更看重功能的实现难度和代码的可读性。而且，现在有一些特殊的软件包可以改善代码的性能，比如 Numpy for Python。随着编译器技术和动态语言设计的进步，如今动态语言已经可以兼顾性能和可读性。比如 Python，它是一种动态语言，并且可以提供 C 语言的性能（C 是一种静态类型语言）。2012 年，Julia 问世。根据 JuliaLang 主页上提供的不同基准测试数据，Julia 的性能表现可以与 C/C++ 媲美。同时 Julia 与 C/C++ 相比，它可以提供一个可以有效用于原型设计的环境，其交互友好性可以与 Python 媲美。Julia 能做到如此水平，归功于 Julia 的设计和基于 JIT 编译器的 LLVM。

下面的一段话来自 Julia 开发团队对 Julia 为什么被创造出来的一些看法。

“我们很贪心：我们想要更多，我们想要一种开源的语言，拥有自由许可。我们希望它兼具 C 的速度与 Ruby 的活力；有像 Lisp 这样的真正的宏，同时有一些明显的、熟悉的数学符号，就像 MATLAB 一样。我们还想要它像 Python 一样有可用于通用编程的特性，并且像 R 一样容易进行统计分析，像 Perl 一样自然地用于字符串处理，像 MATLAB 一样强大的线性代数计算能力，像 Shell 一样擅长将程序黏合在一起。它简单易学，但又让最专业的黑客感到高兴。我们希望它是交互式的，可以被编译。”

Julia 受到了很多其他语言的影响。比如受 Python 的影响，具有高可读性和快速原型设计能力；受 MATLAB 的影响，可以很容易地操作矢量化数值函数，特别是矩阵；受 R 的影响，对数学和统计操作有良好的支持。Julia 从这些语言中学到了很多，并试图将所有这些世界中最好的东西融合在一起，成为一个最让人喜爱的语言。

Julia 非常擅长科学计算，但又并不局限于此。因此，它也可以用于网络 and 通用编程。Julia 的开发团队想用一种语言在兼顾功能和效率的同时，不影响其易用性。Julia 的核心大部分都在 Julia 中实现，其解析器用 Scheme 编写，Julia 的高效和跨平台 I/O 由 Node.js 的 libuv 提供。

下面列举 Julia 的一些特色功能。