

 Keras——一个极简的框架，帮您轻松入门深度学习

深度学习

Keras快速开发入门

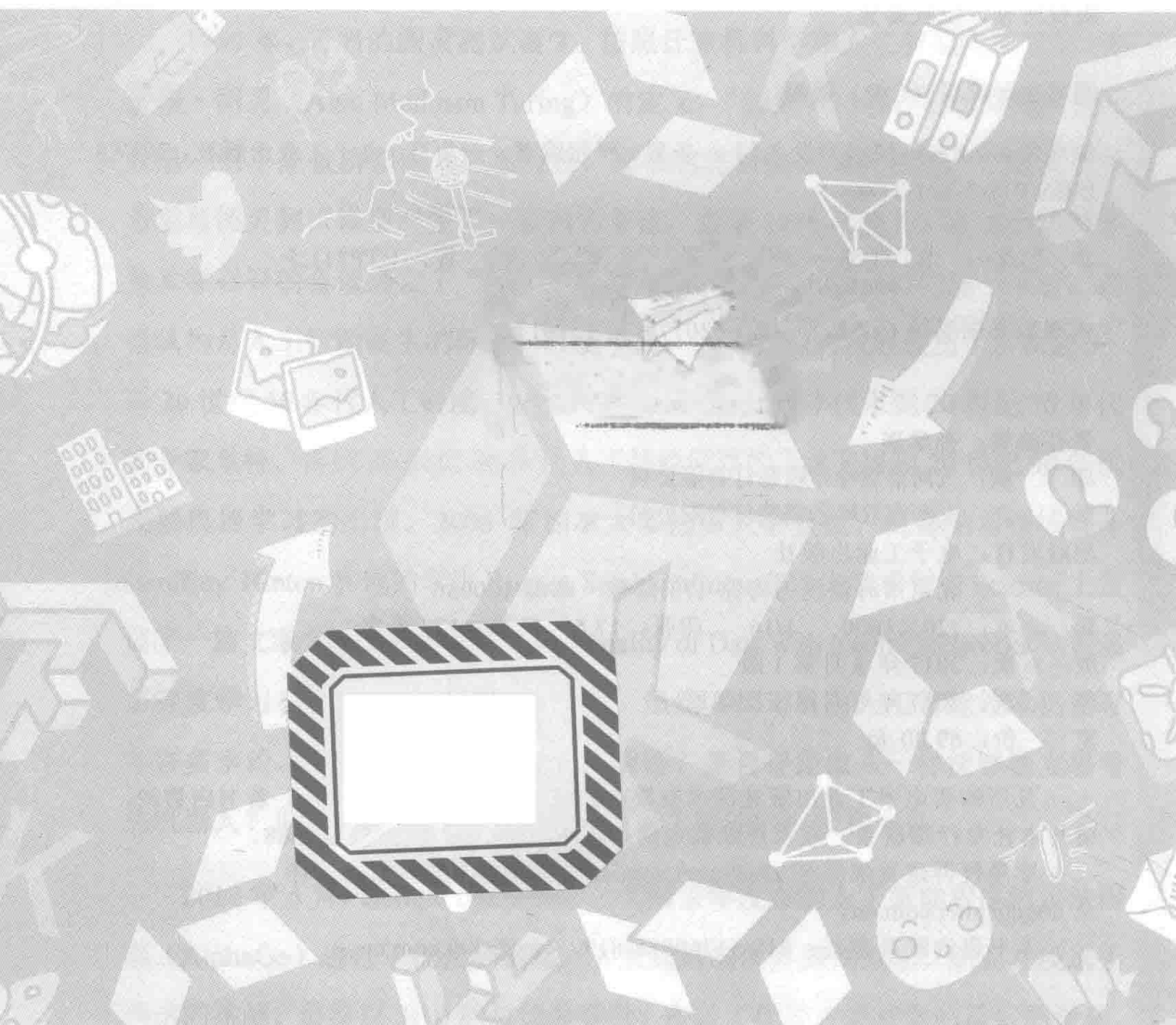
乐毅 严超 编著



深度学习

Keras快速开发入门

乐毅 严超 编著



電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书首先介绍了 Keras 深度学习框架的技术背景、特点以及基本模型的构成，并比较了不同深度学习框架的优缺点。从 Keras 的安装、配置和编译等基本环境入手，详细介绍了 Keras 的模型、网络结构、数据预处理方法、参数配置，以及调试技巧和可视化工具。帮助读者快速掌握 Keras 深度学习框架，从而解决工作和学习当中神经网络模型的应用问题。同时，本书还介绍了如何用 Keras 快速构建深度学习原型并着手实战。最后通过 Cifar-10、词向量和对抗网络（GANs）等实例向读者展示 Keras 作为深度学习开发工具的强大之处，从而帮助读者迅速获得深度学习开发经验。

本书是一本实践性很强的深度学习工具书，适合希望快速学习和使用 Keras 深度学习框架的工程师、学者和从业者。本书对于立志从事深度学习和 AI 相关的行业，希望用 Keras 开发实际项目的工程技术人员，是非常实用的参考手册和工具书。

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

深度学习：Keras 快速开发入门 / 乐毅，严超编著. —北京：电子工业出版社，2017.8
ISBN 978-7-121-31868-9

I. ①深… II. ①乐… ②严… III. ①软件设计 IV. ①TP311.5

中国版本图书馆 CIP 数据核字（2017）第 131017 号

责任编辑：孙学瑛

印 刷：三河市双峰印刷装订有限公司

装 订：三河市双峰印刷装订有限公司

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：17.5 字数：178 千字

版 次：2017 年 8 月第 1 版

印 次：2017 年 8 月第 1 次印刷

定 价：69.00 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 zlts@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

1950年，著名的图灵测试诞生，按照计算机科学和人工智能之父艾伦·麦席森·图灵（Alan Mathison Turing）的定义：“如果一台机器能够与人类展开对话，而不能被辨别出其机器身份，那么称这台机器具有智能。”随后几年围绕着图灵测试模型产生了一系列的争论，直至1956年夏天，在美国达特茅斯大学召开的会议确立了“人工智能（Artificial Intelligence）”这一术语，并被认为是人工智能诞生的标志。时至今日，人工智能的发展经历了多次起伏，从20世纪50年代人工智能概念的兴起到20世纪60年代末到20世纪70年代的专家系统，再到20世纪80年代人工神经网络的飞速发展和20世纪90年代浅层机器学习的出现。2006年加拿大多伦多大学教授、机器学习领域泰斗Geoffrey Hinton和他的学生Ruslan Salakhutdinov在顶尖学术刊物Science上发表了一篇文章“Reducing the Dimensionality of Data with Neural Networks”，开启了深度学习在学术界和工业界的浪潮。虽然人们对目前人工智能技术仍然存在许多争论，到底什么是真正的人工智能？是否存在像人一样能够独立思考的机器人？但这些疑问阻挡不了人类对人工智能的探索和向往。

2017年5月，Google Deepmind公司对去年发布的人工智能程序阿尔法围棋（AlphaGo）进行了优化和训练，并发起对战世界围棋冠军、职业九段选手柯洁的挑战，最终以3:0的总比分获胜，击破了代表人类智慧的最后的堡垒。

Deepmind 公司的围棋程序 (AlphaGo) 不同于传统的人工智能程序, 它能够
将大量棋谱数据输入计算机进行自我学习, 而不像专家系统需要人类的智能
去干预和调整算法。AlphaGo 的这一特点有力地证明了人工智能已经迎来了一个
更高的发展阶段, 即深度学习技术。

与此同时, 国内外众多学者、从业者或业余爱好者都纷纷加入到深度学习
(Deep Learning) 的研究和工作中。然而, 许多读者朋友对深度学习技术中
复杂的数学公式和理论感到困惑, 以至于常常无从下手。目前主流的深度学习
框架, 从 TensorFlow 难以理解的设计和表述方法到 Caffe 冗长的神经网络模
型定义, 往往使读者很难清晰、完整和快速地掌握深度学习技术。本书选择
Keras 这一深度学习框架向读者介绍深度学习技术和应用, 力求使用简洁、高
效和丰富的实例帮助读者快速掌握这门技术, 这得益于 Keras 良好的模块化、
极简的设计、快速原型迭代和易扩展性等特点。通过学习本书的内容, 读者
能够快速搭建满足产品需求的神经网络模型, 从而加快产品研发周期。

本书共包括 9 章, 每章的主要内容如下:

第 1 章 “Keras 概述” 介绍了 Keras 框架的技术背景、特点、优势以及与其他
框架的对比。

第 2 章 “Keras 的安装与配置” 介绍了如何安装与配置 Keras, 为进一步
的 Keras 开发做准备。

第 3 章 “Keras 快速上手” 体现了 Keras 开发极易入手的特点, 一定
Python 基础的开发人员都能非常简单地完成一个 Keras 模型训练。

第 4 章 “Keras 模型的定义” 详细介绍了 Keras 的两种类型模型——
Sequential 模型和函数式模型, 以及它们的参数定义和接口。

第 5 章 “Keras 网络结构” 深入 Keras 的内部结构, 详细介绍了 Keras 的

网络结构及其层的定义，并对每层的参数进行说明和分析。

第 6 章“Keras 数据预处理”讲解了 Keras 提供的常用数据预处理工具方法，以及对于特定数据预处理使用场景的解析。

第 7 章“Keras 内置网络配置”介绍了 Keras 提供的内置网络配置，帮助读者完善模型优化体系的同时，熟悉 Keras 内置网络配置方法。

第 8 章“Keras 实用技巧和可视化”帮助读者完善了 Keras 实用技巧与可视化方法，打通 Keras 与下层框架（Theano、TensorFlow）的使用阻隔。

第 9 章“Keras 实战”完成了三个经典的 Keras 实战练习，增强读者对完整 Keras 训练过程的理解。

本书的内容由我与朋友严超共同完成，在写作过程中我们经常交流、探讨，互相指正对方的不足和错误，从而达到共同进步，在此对严超表示感谢。同时还要感谢电子工业出版社的孙学瑛老师对本书提出了很多非常好的建议，帮助我们顺利出版此书。

最后，由于深度学习技术发展迅速，各种知识和应用工具变化很快，Github 上 Keras 项目还处在活跃开发阶段，并即将发布支持 CNTK 和 MXNet 后端的实现，这使得 Keras 在不断更新和修正。笔者才疏学浅，理解有限，加之编写时间也较仓促，书中难免有错谬之处，敬请广大读者朋友批评指正，不胜感激。

乐毅

2017 年 6 月

本书快览

第 1 章	Keras 概述	1
第 2 章	Keras 的安装与配置	16
第 3 章	Keras 快速上手	25
第 4 章	Keras 模型的定义	37
第 5 章	Keras 网络结构	72
第 6 章	Keras 数据预处理	145
第 7 章	Keras 内置网络配置	168
第 8 章	Keras 实用技巧和可视化	203
第 9 章	Keras 实战	228

目 录

第 1 章 Keras 概述	1
1.1 Keras 简介	1
1.1.1 Keras 2	1
1.1.2 Keras 功能构成	4
1.2 Keras 特点	6
1.3 主流深度学习框架	8
1.3.1 Caffe	8
1.3.2 Torch	10
1.3.3 Keras	12
1.3.4 MXNet	12
1.3.5 TensorFlow	13
1.3.6 CNTK	14
1.3.7 Theano	14
第 2 章 Keras 的安装与配置	16
2.1 Windows 环境下安装 Keras	16

2.1.1	硬件配置	16
2.1.2	Windows 版本	18
2.1.3	Microsoft Visual Studio 版本	18
2.1.4	Python 环境	18
2.1.5	CUDA	18
2.1.6	加速库 CuDNN	19
2.1.7	Keras 框架的安装	19
2.2	Linux 环境下的安装	20
2.2.1	硬件配置	20
2.2.2	Linux 版本	21
2.2.3	Ubuntu 环境的设置	22
2.2.4	CUDA 开发环境	22
2.2.5	加速库 cuDNN	23
2.2.6	Keras 框架安装	24
第 3 章	Keras 快速上手	25
3.1	基本概念	25
3.2	初识 Sequential 模型	29
3.3	一个 MNIST 手写数字实例	30
3.3.1	MNIST 数据准备	30
3.3.2	建立模型	31
3.3.3	训练模型	32

第 4 章 Keras 模型的定义	36
4.1 Keras 模型	36
4.2 Sequential 模型	38
4.2.1 Sequential 模型接口	38
4.2.2 Sequential 模型的数据输入	48
4.2.3 模型编译	49
4.2.4 模型训练	50
4.3 函数式模型	51
4.3.1 全连接网络	52
4.3.2 函数模型接口	53
4.3.3 多输入和多输出模型	63
4.3.4 共享层模型	67
第 5 章 Keras 网络结构	71
5.1 Keras 层对象方法	71
5.2 常用层	72
5.2.1 Dense 层	72
5.2.2 Activation 层	74
5.2.3 Dropout 层	75
5.2.4 Flatten 层	75
5.2.5 Reshape 层	76
5.2.6 Permute 层	77
5.2.7 RepeatVector 层	78
5.2.8 Lambda 层	79

5.2.9	ActivityRegularizer 层.....	80
5.2.10	Masking 层.....	81
5.3	卷积层.....	82
5.3.1	Conv1D 层.....	82
5.3.2	Conv2D 层.....	84
5.3.3	SeparableConv2D 层.....	87
5.3.4	Conv2DTranspose 层.....	91
5.3.5	Conv3D 层.....	94
5.3.6	Cropping1D 层.....	97
5.3.7	Cropping2D 层.....	97
5.3.8	Cropping3D 层.....	98
5.3.9	UpSampling1D 层.....	99
5.3.10	UpSampling2D 层.....	100
5.3.11	UpSampling3D 层.....	101
5.3.12	ZeroPadding1D 层.....	102
5.3.13	ZeroPadding2D 层.....	103
5.3.14	ZeroPadding3D 层.....	104
5.4	池化层.....	105
5.4.1	MaxPooling1D 层.....	105
5.4.2	MaxPooling2D 层.....	106
5.4.3	MaxPooling3D 层.....	108
5.4.4	AveragePooling1D 层.....	109
5.4.5	AveragePooling2D 层.....	110
5.4.6	AveragePooling3D 层.....	111

5.4.7	GlobalMaxPooling1D 层.....	112
5.4.8	GlobalAveragePooling1D 层.....	113
5.4.9	GlobalMaxPooling2D 层.....	113
5.4.10	GlobalAveragePooling2D 层.....	114
5.5	局部连接层.....	115
5.5.1	LocallyConnected1D 层.....	115
5.5.2	LocallyConnected2D 层.....	117
5.6	循环层.....	120
5.6.1	Recurrent 层.....	120
5.6.2	SimpleRNN 层.....	124
5.6.3	GRU 层.....	126
5.6.4	LSTM 层.....	127
5.7	嵌入层.....	129
5.8	融合层.....	131
5.9	激活层.....	134
5.9.1	LeakyReLU 层.....	134
5.9.2	PReLU 层.....	134
5.9.3	ELU 层.....	135
5.9.4	ThresholdedReLU 层.....	136
5.10	规范层.....	137
5.11	噪声层.....	139
5.11.1	GaussianNoise 层.....	139
5.11.2	GaussianDropout 层.....	139
5.12	包装器 Wrapper.....	140

5.12.1	TimeDistributed 层.....	140
5.12.2	Bidirectional 层.....	141
5.13	自定义层.....	142
第 6 章	Keras 数据预处理	144
6.1	序列数据预处理.....	145
6.1.1	序列数据填充.....	145
6.1.2	提取序列跳字样本.....	148
6.1.3	生成序列抽样概率表.....	151
6.2	文本预处理.....	153
6.2.1	分割句子获得单词序列.....	153
6.2.2	OneHot 序列编码器.....	154
6.2.3	单词向量化.....	155
6.3	图像预处理.....	159
第 7 章	Keras 内置网络配置	167
7.1	模型性能评估模块.....	168
7.1.1	Keras 内置性能评估方法.....	168
7.1.2	使用 Keras 内置性能评估.....	170
7.1.3	自定义性能评估函数.....	171
7.2	损失函数.....	171
7.3	优化器函数.....	174
7.3.1	Keras 优化器使用.....	174
7.3.2	Keras 内置优化器.....	176

7.4	激活函数.....	180
7.4.1	添加激活函数方法.....	180
7.4.2	Keras 内置激活函数.....	181
7.4.3	Keras 高级激活函数.....	185
7.5	初始化参数.....	189
7.5.1	使用初始化方法.....	189
7.5.2	Keras 内置初始化方法.....	190
7.5.3	自定义 Keras 初始化方法.....	196
7.6	正则项.....	196
7.6.1	使用正则项.....	197
7.6.2	Keras 内置正则项.....	198
7.6.3	自定义 Keras 正则项.....	198
7.7	参数约束项.....	199
7.7.1	使用参数约束项.....	199
7.7.2	Keras 内置参数约束项.....	200
第 8 章	Keras 实用技巧和可视化.....	202
8.1	Keras 调试与排错.....	202
8.1.1	Keras Callback 回调函数与调试技巧.....	202
8.1.2	备份和还原 Keras 模型.....	215
8.2	Keras 内置 Scikit-Learn 接口包装器.....	217
8.3	Keras 内置可视化工具.....	224

第 9 章 Keras 实战	227
9.1 训练一个准确率高于 90%的 Cifar-10 预测模型	227
9.1.1 数据预处理	232
9.1.2 训练	233
9.2 在 Keras 模型中使用预训练词向量判定文本类别	239
9.2.1 数据下载和实验方法	240
9.2.2 数据预处理	241
9.2.3 训练	245
9.3 用 Keras 实现 DCGAN 生成对抗网络还原 MNIST 样本	247
9.3.1 DCGAN 网络拓扑结构	250
9.3.2 训练	254

第 1 章

Keras 概述

深度学习 (Deep Learning) 越来越成为人工智能领域研究和应用的热门方向, 并不断涌现出很多优秀的深度学习框架。其中 Keras 框架自诞生以来, 由于其良好的模块化, 极简的设计和快速原型迭代等特性, 越来越受到业内人士的青睐。也正由于这些特点, Keras 成为机器学习领域值得推荐的深度学习框架之一。

本章首先介绍了 Keras 的由来及技术背景, 然后比较了当前主流的深度学习框架, 最后总结了 Keras 框架的特点。

1.1 Keras 简介

1.1.1 Keras 2

Keras 深度学习框架的作者是 Francois Chollet, 现就职于谷歌, 从事机器学习与人工智能技术的研究。Keras 在 2015 年 3 月首次推出, 现在用户数量已经突破了 10 万。其中有数百人为 Keras 代码库做出了贡献, 更

有数千人为 Keras 社区做出了贡献。截止本书截稿之日，Keras 在 GitHub 上的 star 数已达 16333 个。Keras 不仅催生了众多创业公司，提高了研究者的成果率，而且简化了大公司的工程流程图，并为数以千计没有机器学习经验的工程师打开了一扇通向深度学习的大门。

今年 3 月，Keras 通过官方博客发布了全新的 Keras 2 版本，它带有一个更易使用的新 API，实现了与 TensorFlow 的直接整合。这是在 TensorFlow 核心整合 Keras API 所准备的重要一步。

Keras 2 相比以前的版本有很多新变化，主要体现在以下几个方面。

1. 与 TensorFlow 等后端框架整合

尽管 Keras 自 2015 年 12 月已经作为运行时后端（runtime backend）开始支持 TensorFlow，但 Keras API 却一直与 TensorFlow 代码库相分离。Keras 2 改变了这一情况，从 TensorFlow 1.2 版本开始，Keras API 可作为 TensorFlow workflow 的一部分直接使用，这是 TensorFlow 在向数百万新用户开源的道路上迈出的一大步。

Keras 作为一个 API 技术规范被理解和使用，而不是一个特殊的代码库。事实上，Keras 技术规范的发展会出现的两个不同分支：(a) TensorFlow 的内部实现（如 tf.keras），其完全由 TensorFlow 设计和实现，与 TensorFlow 的所有功能深度兼容；(b) 外部的多后端实现，同时支持 Theano 和 TensorFlow（以后可能会支持更多的后台，如微软 CNTK 和 MXNET 后端已经处于密集开发阶段和发布前期）。

类似的，SkyMind（商业智能和企业软件公司）正在用 Scala 语言实