

兼容TensorFlow 1.x与2.x版本

共75个实例，覆盖了TensorFlow的大量接口

提供了大量可重用代码，可应用于真实场景

涉及图像识别、文本分类、数值分析、机器翻译、语音合成等

李大学 | 力荐

京东终身荣誉技术顾问/磁云科技创始人

深度学习之 TensorFlow

工程化项目实战

李金洪◎编著



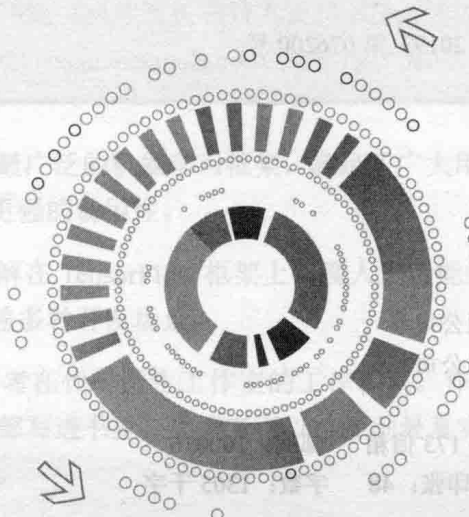
 中国工信出版集团

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

深度学习之 TensorFlow

工程化项目实战

李金洪◎编著



电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

这是一本非常全面的、专注于实战的 AI 图书，兼容 TensorFlow 1.x 和 2.x 版本，共 75 个实例。

全书共分为 5 篇：第 1 篇，介绍了学习准备、搭建开发环境、使用 AI 模型来识别图像；第 2 篇，介绍了用 TensorFlow 开发实际工程的一些基础操作，包括使用 TensorFlow 制作自己的数据集、快速训练自己的图片分类模型、编写训练模型的程序；第 3 篇，介绍了机器学习算法相关内容，包括特征工程、卷积神经网络（CNN）、循环神经网络（RNN）；第 4 篇，介绍了多模型的组合训练技术，包括生成式模型、模型的攻与防；第 5 篇，介绍了深度学习在工程上的应用，侧重于提升读者的工程能力，包括 TensorFlow 模型制作、部署 TensorFlow 模型、商业实例。

本书结构清晰、案例丰富、通俗易懂、实用性强。适合对人工智能、TensorFlow 感兴趣的读者作为自学教程。另外，本书也适合社会培训学校作为培训教材，还适合大中专院校的相关专业作为教学参考书。

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

深度学习之 TensorFlow 工程化项目实战 / 李金洪编著. —北京：电子工业出版社，2019.5
ISBN 978-7-121-36392-4

I. ①深… II. ①李… III. ①人工智能—算法 IV. ①TP18

中国版本图书馆 CIP 数据核字（2019）第 076200 号

策划编辑：吴宏伟

责任编辑：牛 勇

印 刷：三河市良远印务有限公司

装 订：三河市良远印务有限公司

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：787×1092 1/16 印张：48 字数：1305 千字

版 次：2019 年 5 月第 1 版

印 次：2019 年 5 月第 1 次印刷

定 价：159.00 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。

作者介绍



李金洪

“大蛇智能”网站创始人、“代码医生”工作室主程序员。

精通Python、C、C++、汇编、Java和Go等多种编程语言。擅长神经网络、算法、协议分析、逆向工程和移动互联网安全架构等技术。在深度学习领域，参与过某移动互联网后台的OCR项目、某娱乐节目机器人的语音识别和声纹识别项目，以及人脸识别、活体检测等多个项目。在“代码医生”工作室工作期间，完成过金融、安全、市政和医疗等多个领域的AI算法外包项目。

出版过《Python带我起——入门、进阶、商业实战》《深度学习之TensorFlow——入门、原理与进阶实战》两本书。

本书编辑

吴宏伟

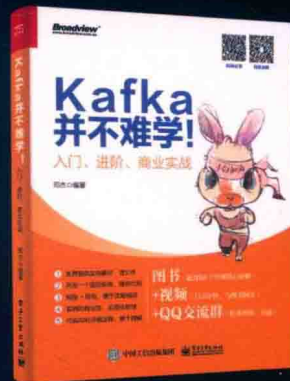
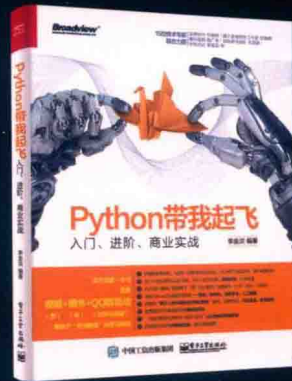
邮箱: wuhongwei@phei.com.cn

QQ: 83744810

欢迎投稿



好书分享



前言

关注并访问公众号“xiangyuejiqiren”，在公众号中回复“深2”得到相关资源的下载链接。



3-1 使用AI模型识别图像.py	5-2 使用nasnet-mobile模型来识别图像.py	4-1 将模型数据制作成内存对象数据集.py
3-2 通过代码生成数据集.py	4-3 将图片制作成内存对象数据集.py	4-4 将exec文件制作成内存对象数据集.py
4-5 将图片文件制作成tfRecord数据集.py	4-6 interleave例子.py	4-7 Dataset对象的操作方法.py
4-8 将内存数据转成Dataset数据集.py	4-9 from_tensor_slices的注意事项.py	4-10 将图片文件制作成Dataset数据集.py
4-11 将tfRecord文件制作成Dataset数据集...	4-12 在动态图读取Dataset数据集.py	4-13 在动态图读取Dataset数据集_tflite.py
4-14 在不同场景中使用数据集.py	5-1 mydataset.py	5-2 model.py
5-3 train.py	5-4 test.py	5-5 测试TF-Hub库中的mobilenet_v2模型.py
5-6 使用模型评估人物的年龄.py	6-1 使用静态图训练一个具有保存检查点功能...	6-2 使用动态图训练一个具有保存检查点功能...
6-3 动态图另一种便捷方法.py	6-4 从动态图种获取变量.py	6-5 静态图中使用动态图.py
6-6 使用估算器框架训练一个回归模型.py	6-7 为估算器添加钩子.py	6-8 自定义hook.py
6-9 将估算器模型转为静态图模型.py	6-10 tf.layers模型.py	6-11 keras回归模型.py
6-12 使用tf.keras预测模型.py	6-13 在静态图中使用tf.keras.py	6-14 tfjs回归例子.html
6-15 使用估算器框架进行分布式训练.py	6-16 使用估算器框架进行分布式训练ops.py	6-17 使用估算器框架进行分布式训练chief.py
6-18 使用估算器框架进行分布式训练work.py	6-19 用ResNet识别桔子和苹果.py	6-20 在TF1框架中训练mnist.py
6-21 查看TF1模型及解参.py	6-22_tflitecode.py	7-1 用wide and deep模型预测人口收入.py
7-2 用boosted_trees模型预测人口收入.py	7-3 使用feature_column处理连续值特征列.py	7-4 将连续值特征转成离散值特征.py
7-5 将离散文本特征列转化为one-hot编码与...	7-6 根据特征列生成交叉列.py	7-7 序列特征工程.py
7-8 聚类COCO数据集中标注.py	7-9 mnistkmeans.py	7-10 电影推荐系统.py
7-11 用lattice预测收入.py	7-12 lattice结合dnn.py	7-13 preprocess.py
7-14 MKR.py	7-15 train.py	7-16 data_loader.py
8-1 读取fashion-mnist数据集.py	8-2 Capsulemodel.py	8-3 使用数量词识别白黑图中的服装图案.py
8-4 capsnet_em.py	8-5 train_EM.py	8-6 NLP文本预处理.py
8-7 TextCnn模型.py	8-8 使用keras注意力机制模型进行文本分类.py	8-9 使用keras注意力机制模型分析评论者情...
8-10 keras注意力机制模型.py	8-11 yolo_v3.py	8-12 使用YOLOV3模型进行实例检测.py
8-13 annotation.py	8-14 generator.py	8-15 box.py

本书由大蛇智能官网提供内容有关的技术支持。在阅读过程中，如有不理解的技术点，可以到论坛 <https://bbs.aianaconda.com> 发帖进行提问。

TensorFlow 是目前使用最广泛的机器学习框架，满足了广大用户的需求。如今 TensorFlow 已经更新到 2.x 版本，具有更强的易用性。

本书通过大量的实例讲解在 TensorFlow 框架上实现人工智能的技术，兼容 TensorFlow 1.x 与 TensorFlow 2.x 版本，覆盖多种开发场景。

书中的内容主要源于作者在代码医生工作室的工作积累。作者将自己在真实项目中使用 TensorFlow 的经验与技巧全部写进书里，让读者可以接触到最真实的案例、最实战的场景，尽快搭上人工智能的“列车”。

作者将自身的项目实战经验浓缩到三本书里，形成了“深度学习三部曲”。三本书形成一套完善的知识体系，构成了完备的技术栈闭环。

本书是“深度学习三部曲”的最后一本。

- 《Python 带我起飞——入门、进阶、商业实战》，主要讲解了 Python 基础语法。与深度学习关系不大，但包含了开发神经网络模型所必备的基础知识。
- 《深度学习之 TensorFlow——入门、原理与进阶实战》，主要讲解了深度学习的基础

网络模型及 TensorFlow 框架的基础编程方法。

- 《深度学习之 TensorFlow 工程化项目实战》，主要讲解在实战项目中用到的真实模型，以及将 TensorFlow 框架用于各种生产环境的编程方法。

这三本书可以将一个零基础的读者顺利带入深度学习行业，并让其能够成为一名合格的深度学习工程师。

本书特色

1. 兼容 TensorFlow 1.x 与 2.x 版本，提供了大量的编程经验

本书兼顾 TensorFlow 1.x 与 2.x 两个版本，给出了如何将 TensorFlow 1.x 代码升级为 TensorFlow 2.x 可用的代码。

2. 覆盖了 TensorFlow 的大量接口

TensorFlow 是一个非常庞大的框架，内部有很多接口可以满足不同用户的需求。合理使用现有接口可以在开发过程中起到事半功倍的效果。然而，由于 TensorFlow 的代码迭代速度太快，有些接口的配套文档并不是很全。作者花了大量的时间与精力，对一些实用接口的使用方法进行摸索与整理，并将这些方法写到书中。

3. 提供了高度可重用代码，公开了大量的商用代码片段

本书实例中的代码大多都来自代码医生工作室的商业项目，这些代码的易用性、稳定性、可重用性都很强。读者可以将这些代码提取出来直接用在自己的项目中，加快开发进度。

4. 书中的实战案例可应用于真实场景

本书中大部分实例都是当前应用非常广泛的通用任务，包括图片分类、目标识别、像素分割、文本分类、语音合成等多个方向。读者可以在书中介绍的模型的基础上，利用自己的业务数据集快速实现 AI 功能。

5. 从工程角度出发，覆盖工程开发全场景

本书以工程实现为目标，全面覆盖开发实际 AI 项目中所涉及的知识，并全部配有实例，包括开发数据集、训练模型、特征工程、开发模型、保护模型文件、模型防御、服务端和终端的模型部署。其中，特征工程部分全面讲解了 TensorFlow 中的特征列接口。该接口可以使数据在特征处理阶段就以图的方式进行加工，从而保证了在训练场景下和使用场景下模型的输入统一。

6. 提供了大量前沿论文链接地址，便于读者进一步深入学习

本书使用的 AI 模型，大多来源于前沿的技术论文，并在原有论文基础上做了一些结构改进。这些实例具有很高的科研价值。读者可以根据书中提供的论文链接地址，进一步深入学习更多的前沿知识，再配合本书的实例进行充分理解，达到融会贯通。本书也可以帮助 AI 研究者进行学术研究。

7. 注重方法与经验的传授

本书在讲解知识时，更注重传授方法与经验。全书共有几十个“提示”标签，其中的内容都是含金量很高的成功经验分享与易错事项总结，有关于经验技巧的，也有关于风险规避的，可以帮助读者在学习的路途上披荆斩棘，快速进步。

本书读者对象

- 人工智能爱好者
- 人工智能专业的高校学生
- 人工智能专业的教师
- 人工智能初学者
- 人工智能开发工程师
- 使用 TensorFlow 框架的工程师
- 集成人工智能的开发人员

关于作者

本书由李金洪主笔编写，参与本书编写的还有以下作者。

石昌帅

代码医生工作室成员，具有丰富的嵌入式及算法开发经验，参与多款机器人、图像识别等项目开发，擅长机器人定位、导航技术、计算机视觉技术，熟悉 NVIDIA Jetson 系列、Raspberry Pi 系列等平台软硬件开发、算法优化。从事的技术方向包括机器人导航、图像处理、自动驾驶等。

甘月

代码医生工作室成员，资深 iOS 高级工程师，有丰富的 iOS 研发经验，先后担任 iOS 主管、项目经理、iOS 技术总监等职务，精通 Objective-C、Swift、C 等编程语言，参与过银行金融、娱乐机器人、婚庆、医疗等领域的多个项目。擅长 Mac 系统下的 AI 技术开发。

江泉宇

代码医生工作室成员，是大蛇智能社区成长最快的 AI 学者。半年时间，由普通读者升级为社区的资深辅导员。在校期间曾参加过电子设计大赛（获省级一等奖）、Google 校企合作的 AI 创新项目、省级创新训练 AI 项目。熟悉 Python、C 和 Java 等编程语言。擅长图像处理方向、特征工程方向及语义压缩方向的 AI 任务。

目 录

第 1 篇 准备	
第 1 章 学习准备	2
1.1 TensorFlow 能做什么	2
1.2 学习 TensorFlow 的必备知识	3
1.3 学习技巧：跟读代码	4
1.4 如何学习本书	4
第 2 章 搭建开发环境	5
2.1 准备硬件环境	5
2.2 下载及安装 Anaconda	6
2.3 安装 TensorFlow	9
2.4 GPU 版本的安装方法	10
2.4.1 在 Windows 中安装 CUDA	10
2.4.2 在 Linux 中安装 CUDA	13
2.4.3 在 Windows 中安装 cuDNN	13
2.4.4 在 Linux 中安装 cuDNN	14
2.4.5 常见错误及解决方案	16
2.5 测试显卡的常用命令	16
2.6 TensorFlow 1.x 版本与 2.x 版本共存的解决方案	18
第 3 章 实例 1：用 AI 模型识别图像是桌子、猫、狗，还是其他	21
3.1 准备代码环境并预训练模型	21
3.2 代码实现：初始化环境变量，并载入 ImgNet 标签	24
3.3 代码实现：定义网络结构	25
3.4 代码实现：载入模型进行识别	26
3.5 扩展：用更多预训练模型完成图片分类任务	28

这些实例具有很高的科研价值。读者可以参考本书中提供的论文链接地址，进一步深入学习更多的前沿知识，再配合本书的实例进行充分训练，达到融会贯通。本书也可以帮助 AI 研究人员进行学术攻关。

第 2 篇 基础

第 4 章 用 TensorFlow 制作自己的数据集.....	30
4.1 快速导读	30
4.1.1 什么是数据集	30
4.1.2 TensorFlow 的框架	31
4.1.3 什么是 TFDS	31
4.2 实例 2: 将模拟数据制作成内存对象数据集.....	32
4.2.1 代码实现: 生成模拟数据	32
4.2.2 代码实现: 定义占位符	33
4.2.3 代码实现: 建立会话, 并获取数据.....	34
4.2.4 代码实现: 将模拟数据可视化.....	34
4.2.5 运行程序	34
4.2.6 代码实现: 创建带有迭代值并支持乱序功能的模拟数据集	35
4.3 实例 3: 将图片制作成内存对象数据集.....	37
4.3.1 样本介绍	38
4.3.2 代码实现: 载入文件名称与标签.....	39
4.3.3 代码实现: 生成队列中的批次样本数据.....	40
4.3.4 代码实现: 在会话中使用数据集.....	41
4.3.5 运行程序	42
4.4 实例 4: 将 Excel 文件制作成内存对象数据集	42
4.4.1 样本介绍	43
4.4.2 代码实现: 逐行读取数据并分离标签.....	43
4.4.3 代码实现: 生成队列中的批次样本数据.....	44
4.4.4 代码实现: 在会话中使用数据集.....	45
4.4.5 运行程序	46
4.5 实例 5: 将图片文件制作成 TFRecord 数据集	46
4.5.1 样本介绍	47
4.5.2 代码实现: 读取样本文件的目录及标签.....	47
4.5.3 代码实现: 定义函数生成 TFRecord 数据集.....	48
4.5.4 代码实现: 读取 TFRecord 数据集, 并将其转化为队列.....	49
4.5.5 代码实现: 建立会话, 将数据保存到文件.....	50
4.5.6 运行程序	51
4.6 实例 6: 将内存对象制作成 Dataset 数据集.....	52
4.6.1 如何生成 Dataset 数据集	52
4.6.2 如何使用 Dataset 接口	53
4.6.3 tf.data.Dataset 接口所支持的数据集变换操作.....	54

4.6.4	代码实现：以元组和字典的方式生成 Dataset 对象	58
4.6.5	代码实现：对 Dataset 对象中的样本进行变换操作	59
4.6.6	代码实现：创建 Dataset 迭代器	60
4.6.7	代码实现：在会话中取出数据	60
4.6.8	运行程序	61
4.6.9	使用 <code>tf.data.Dataset.from_tensor_slices</code> 接口的注意事项	62
4.7	实例 7：将图片文件制作成 Dataset 数据集	63
4.7.1	代码实现：读取样本文件的目录及标签	64
4.7.2	代码实现：定义函数，实现图片转换操作	64
4.7.3	代码实现：用自定义函数实现图片归一化	65
4.7.4	代码实现：用第三方函数将图片旋转 30°	65
4.7.5	代码实现：定义函数，生成 Dataset 对象	66
4.7.6	代码实现：建立会话，输出数据	67
4.7.7	运行程序	68
4.8	实例 8：将 TFRecord 文件制作成 Dataset 数据集	69
4.8.1	样本介绍	69
4.8.2	代码实现：定义函数，生成 Dataset 对象	70
4.8.3	代码实现：建立会话输出数据	71
4.8.4	运行程序	72
4.9	实例 9：在动态图中读取 Dataset 数据集	72
4.9.1	代码实现：添加动态图调用	72
4.9.2	制作数据集	73
4.9.3	代码实现：在动态图中显示数据	73
4.9.4	实例 10：在 TensorFlow 2.x 中操作数据集	74
4.10	实例 11：在不同场景中使用数据集	77
4.10.1	代码实现：在训练场景中使用数据集	78
4.10.2	代码实现：在应用模型场景中使用数据集	79
4.10.3	代码实现：在训练与测试混合场景中使用数据集	80
4.11	<code>tf.data.Dataset</code> 接口的更多应用	81
第 5 章	10 分钟快速训练自己的图片分类模型	82
5.1	快速导读	82
5.1.1	认识模型和模型检查点文件	82
5.1.2	了解“预训练模型”与微调 (Fine-Tune)	82
5.1.3	学习 TensorFlow 中的预训练模型库——TF-Hub 库	83
5.2	实例 12：通过微调模型分辨男女	83
5.2.1	准备工作	84

5.2.2	代码实现：处理样本数据并生成 Dataset 对象	85
5.2.3	代码实现：定义微调模型的类 MyNASNetModel	88
5.2.4	代码实现：构建 MyNASNetModel 类中的基本模型	88
5.2.5	代码实现：实现 MyNASNetModel 类中的微调操作	89
5.2.6	代码实现：实现与训练相关的其他方法	90
5.2.7	代码实现：构建模型，用于训练、测试、使用	92
5.2.8	代码实现：通过二次迭代来训练微调模型	94
5.2.9	代码实现：测试模型	96
5.3	扩展：通过摄像头实时分辨男女	100
5.4	TF-slim 接口中的更多成熟模型	100
5.5	实例 13：用 TF-Hub 库微调模型以评估人物的年龄	100
5.5.1	准备样本	101
5.5.2	下载 TF-Hub 库中的模型	102
5.5.3	代码实现：测试 TF-Hub 库中的 MobileNet_V2 模型	104
5.5.4	用 TF-Hub 库微调 MobileNet_V2 模型	107
5.5.5	代码实现：用模型评估人物的年龄	109
5.5.6	扩展：用 TF-Hub 库中的其他模型处理不同领域的分类任务	113
5.6	总结	113
5.7	练习题	114
5.7.1	基于 TF-slim 接口的练习	115
5.7.2	基于 TF-Hub 库的练习	115
第 6 章	用 TensorFlow 编写训练模型的程序	117
6.1	快速导读	117
6.1.1	训练模型是怎么回事	117
6.1.2	用“静态图”方式训练模型	117
6.1.3	用“动态图”方式训练模型	118
6.1.4	什么是估算器框架接口（Estimators API）	119
6.1.5	什么是 tf.layers 接口	120
6.1.6	什么是 tf.keras 接口	121
6.1.7	什么是 tf.js 接口	122
6.1.8	什么是 TFLearn 框架	123
6.1.9	该选择哪种框架	123
6.1.10	分配运算资源与使用分布策略	124
6.1.11	用 tfdbg 调试 TensorFlow 模型	127
6.1.12	用钩子函数（Training Hooks）跟踪训练状态	127
6.1.13	用分布式运行方式训练模型	128

6.1.14	用 T2T 框架系统更方便地训练模型	128
6.1.15	将 TensorFlow 1.x 中的代码移植到 2.x 版本	129
6.1.16	TensorFlow 2.x 中的新特性——自动图	130
6.2	实例 14: 用静态图训练一个具有保存检查点功能的回归模型	131
6.2.1	准备开发步骤	131
6.2.2	生成检查点文件	131
6.2.3	载入检查点文件	132
6.2.4	代码实现: 在线性回归模型中加入保存检查点功能	132
6.2.5	修改迭代次数, 二次训练	135
6.3	实例 15: 用动态图 (eager) 训练一个具有保存检查点功能的回归模型	136
6.3.1	代码实现: 启动动态图, 生成模拟数据	136
6.3.2	代码实现: 定义动态图的网络结构	137
6.3.3	代码实现: 在动态图中加入保存检查点功能	138
6.3.4	代码实现: 按指定迭代次数进行训练, 并可视化结果	139
6.3.5	运行程序, 显示结果	140
6.3.6	代码实现: 用另一种方法计算动态图梯度	141
6.3.7	实例 16: 在动态图中获取参数变量	142
6.3.8	小心动态图中的参数陷阱	144
6.3.9	实例 17: 在静态图中使用动态图	145
6.4	实例 18: 用估算器框架训练一个回归模型	147
6.4.1	代码实现: 生成样本数据集	147
6.4.2	代码实现: 设置日志级别	148
6.4.3	代码实现: 实现估算器的输入函数	148
6.4.4	代码实现: 定义估算器的模型函数	149
6.4.5	代码实现: 通过创建 config 文件指定硬件的运算资源	151
6.4.6	代码实现: 定义估算器	152
6.4.7	用 tf.estimator.RunConfig 控制更多的训练细节	153
6.4.8	代码实现: 用估算器训练模型	153
6.4.9	代码实现: 通过热启动实现模型微调	155
6.4.10	代码实现: 测试估算器模型	158
6.4.11	代码实现: 使用估算器模型	158
6.4.12	实例 19: 为估算器添加日志钩子函数	159
6.5	实例 20: 将估算器代码改写成静态图代码	161
6.5.1	代码实现: 复制网络结构	161
6.5.2	代码实现: 重用输入函数	163
6.5.3	代码实现: 创建会话恢复模型	163
6.5.4	代码实现: 继续训练	163

6.6	实例 21: 用 tf.layers API 在动态图上识别手写数字	165
6.6.1	代码实现: 启动动态图并加载手写图片数据集	165
6.6.2	代码实现: 定义模型的类	166
6.6.3	代码实现: 定义网络的反向传播	167
6.6.4	代码实现: 训练模型	167
6.7	实例 22: 用 tf.keras API 训练一个回归模型	168
6.7.1	代码实现: 用 model 类搭建模型	168
6.7.2	代码实现: 用 sequential 类搭建模型	169
6.7.3	代码实现: 搭建反向传播的模型	171
6.7.4	代码实现: 用两种方法训练模型	172
6.7.5	代码实现: 获取模型参数	172
6.7.6	代码实现: 测试模型与用模型进行预测	173
6.7.7	代码实现: 保存模型与加载模型	173
6.7.8	代码实现: 将模型导出成 JSON 文件, 再将 JSON 文件导入模型	175
6.7.9	实例 23: 在 tf.keras 接口中使用预训练模型 ResNet	176
6.7.10	扩展: 在动态图中使用 tf.keras 接口	178
6.7.11	实例 24: 在静态图中使用 tf.keras 接口	178
6.8	实例 25: 用 tf.js 接口后方训练一个回归模型	180
6.8.1	代码实现: 在 HTTP 的头标签中添加 tfjs 模块	180
6.8.2	代码实现: 用 JavaScript 脚本实现回归模型	181
6.8.3	运行程序: 在浏览器中查看效果	181
6.8.4	扩展: tf.js 接口的应用场景	182
6.9	实例 26: 用估算器框架实现分布式部署训练	182
6.9.1	运行程序: 修改估算器模型, 使其支持分布式	182
6.9.2	通过 TF_CONFIG 进行分布式配置	183
6.9.3	运行程序	185
6.9.4	扩展: 用分布策略或 KubeFlow 框架进行分布式部署	186
6.10	实例 27: 在分布式估算器框架中用 tf.keras 接口训练 ResNet 模型, 识别图片中是橘子还是苹果	186
6.10.1	样本准备	186
6.10.2	代码实现: 准备训练与测试数据集	187
6.10.3	代码实现: 制作模型输入函数	187
6.10.4	代码实现: 搭建 ResNet 模型	188
6.10.5	代码实现: 训练分类器模型	189
6.10.6	运行程序: 评估模型	190
6.10.7	扩展: 全连接网络的优化	190

6.11	实例 28: 在 T2T 框架中用 <code>tf.layers</code> 接口实现 MNIST 数据集分类	191
6.11.1	代码实现: 查看 T2T 框架中的数据集 (problems)	191
6.11.2	代码实现: 构建 T2T 框架的工作路径及下载数据集	192
6.11.3	代码实现: 在 T2T 框架中搭建自定义卷积网络模型	193
6.11.4	代码实现: 用动态图方式训练自定义模型	194
6.11.5	代码实现: 在动态图中用 <code>metrics</code> 模块评估模型	195
6.12	实例 29: 在 T2T 框架中, 用自定义数据集训练中英文翻译模型	196
6.12.1	代码实现: 声明自己的 <code>problems</code> 数据集	196
6.12.2	代码实现: 定义自己的 <code>problems</code> 数据集	197
6.12.3	在命令行下生成 <code>TFrecoder</code> 格式的数据	198
6.12.4	查找 T2T 框架中的模型及超参, 并用指定的模型及超参进行训练	199
6.12.5	用训练好的 T2T 框架模型进行预测	201
6.12.6	扩展: 在 T2T 框架中, 如何选取合适的模型及超参	202
6.13	实例 30: 将 TensorFlow 1.x 中的代码升级为可用于 2.x 版本的代码	203
6.13.1	准备工作: 创建 Python 虚环境	203
6.13.2	使用工具转换源码	204
6.13.3	修改转换后的代码文件	204
6.13.4	将代码升级到 TensorFlow 2.x 版本的经验总结	205
第 3 篇 进阶		
第 7 章 特征工程——会说话的数据		208
7.1	快速导读	208
7.1.1	特征工程的基础知识	208
7.1.2	离散数据特征与连续数据特征	209
7.1.3	了解特征列接口	210
7.1.4	了解序列特征列接口	210
7.1.5	了解弱学习器接口——梯度提升树 (TFBT 接口)	210
7.1.6	了解特征预处理模块 (<code>tf.Transform</code>)	211
7.1.7	了解因子分解模块	212
7.1.8	了解加权矩阵分解算法	212
7.1.9	了解 <code>Lattice</code> 模块——点阵模型	213
7.1.10	联合训练与集成学习	214
7.2	实例 31: 用 <code>wide_deep</code> 模型预测人口收入	214
7.2.1	了解人口收入数据集	214
7.2.2	代码实现: 探索性数据分析	217
7.2.3	认识 <code>wide_deep</code> 模型	218

7.2.4	部署代码文件	219
7.2.5	代码实现: 初始化样本常量	220
7.2.6	代码实现: 生成特征列	220
7.2.7	代码实现: 生成估算器模型	222
7.2.8	代码实现: 定义输入函数	223
7.2.9	代码实现: 定义用于导出冻结图文件的函数	224
7.2.10	代码实现: 定义类, 解析启动参数	225
7.2.11	代码实现: 训练和测试模型	226
7.2.12	代码实现: 使用模型	227
7.2.13	运行程序	228
7.3	实例 32: 用弱学习器中的梯度提升树算法预测人口收入	229
7.3.1	代码实现: 为梯度提升树模型准备特征列	230
7.3.2	代码实现: 构建梯度提升树模型	230
7.3.3	代码实现: 训练并导出梯度提升树模型	231
7.3.4	代码实现: 设置启动参数, 运行程序	232
7.3.5	扩展: 更灵活的 TFBT 接口	233
7.4	实例 33: 用 feature_column 模块转换特征列	233
7.4.1	代码实现: 用 feature_column 模块处理连续值特征列	234
7.4.2	代码实现: 将连续值特征列转化成离散值特征列	237
7.4.3	代码实现: 将离散文本特征列转化为 one-hot 与词向量	239
7.4.4	代码实现: 根据特征列生成交叉列	246
7.5	实例 34: 用 sequence_feature_column 接口完成自然语言处理任务的数据预处理工作	248
7.5.1	代码实现: 构建模拟数据	248
7.5.2	代码实现: 构建词嵌入初始值	249
7.5.3	代码实现: 构建词嵌入特征列与共享特征列	249
7.5.4	代码实现: 构建序列特征列的输入层	250
7.5.5	代码实现: 建立会话输出结果	251
7.6	实例 35: 用 factorization 模块的 kmeans 接口聚类 COCO 数据集中的标注框	253
7.6.1	代码实现: 设置要使用的数据集	253
7.6.2	代码实现: 准备带聚类的数据样本	253
7.6.3	代码实现: 定义聚类模型	255
7.6.4	代码实现: 训练模型	256
7.6.5	代码实现: 输出图示化结果	256
7.6.6	代码实现: 提取并排序聚类结果	258
7.6.7	扩展: 聚类与神经网络混合训练	258

7.7	实例 36: 用加权矩阵分解模型实现基于电影评分的推荐系统	259
7.7.1	下载并加载数据集	259
7.7.2	代码实现: 根据用户和电影特征列生成稀疏矩阵	260
7.7.3	代码实现: 建立 WALS 模型, 并对其进行训练	261
7.7.4	代码实现: 评估 WALS 模型	263
7.7.5	代码实现: 用 WALS 模型为用户推荐电影	264
7.7.6	扩展: 使用 WALS 的估算器接口	265
7.8	实例 37: 用 Lattice 模块预测人口收入	265
7.8.1	代码实现: 读取样本, 并创建输入函数	266
7.8.2	代码实现: 创建特征列, 并保存校准关键点	267
7.8.3	代码实现: 创建校准线性模型	270
7.8.4	代码实现: 创建校准点阵模型	270
7.8.5	代码实现: 创建随机微点阵模型	271
7.8.6	代码实现: 创建集合的微点阵模型	271
7.8.7	代码实现: 定义评估与训练函数	272
7.8.8	代码实现: 训练并评估模型	273
7.8.9	扩展: 将点阵模型嵌入神经网络中	274
7.9	实例 38: 结合知识图谱实现基于电影的推荐系统	278
7.9.1	准备数据集	278
7.9.2	预处理数据	279
7.9.3	搭建 MKR 模型	279
7.9.4	训练模型并输出结果	286
7.10	可解释性算法的意义	286
第 8 章	卷积神经网络 (CNN) —— 在图像处理中应用最广泛的模型	287
8.1	快速导读	287
8.1.1	认识卷积神经网络	287
8.1.2	什么是空洞卷积	288
8.1.3	什么是深度卷积	290
8.1.4	什么是深度可分离卷积	290
8.1.5	了解卷积网络的缺陷及补救方法	291
8.1.6	了解胶囊神经网络与动态路由	292
8.1.7	了解矩阵胶囊网络与 EM 路由算法	297
8.1.8	什么是 NLP 任务	298
8.1.9	了解多头注意力机制与内部注意力机制	298
8.1.10	什么是带有位置向量的词嵌入	300
8.1.11	什么是目标检测任务	300