

“好程序员成长”丛书



Python
快乐编程
TensorFlow

深度学习

项目实战

©千锋教育高教产品研发部 / 编著

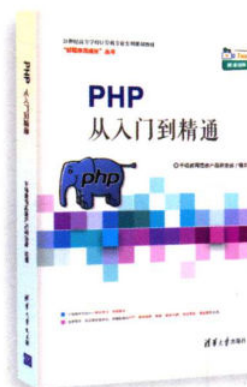
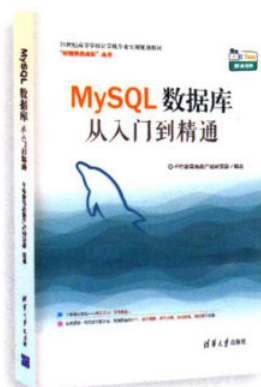
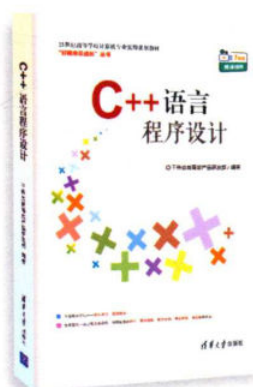
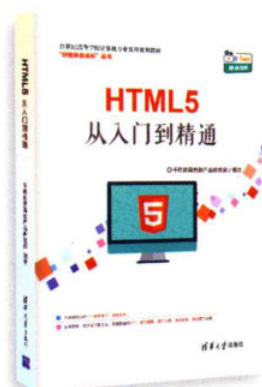
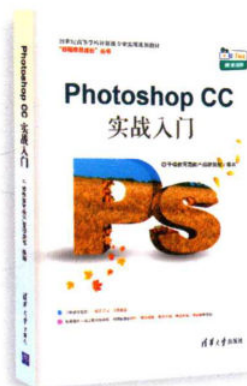
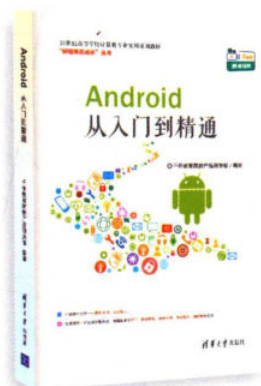
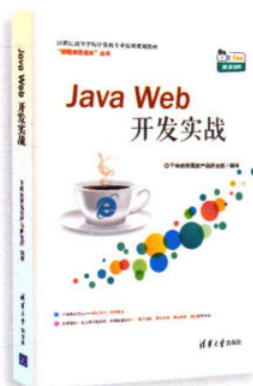
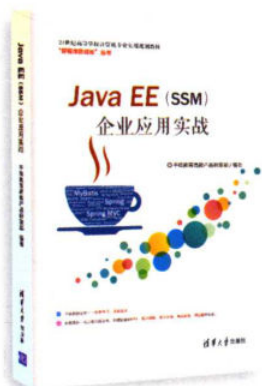
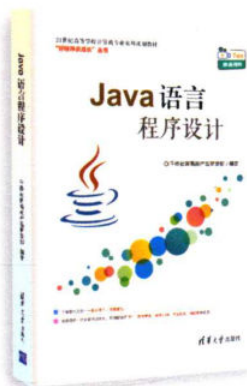
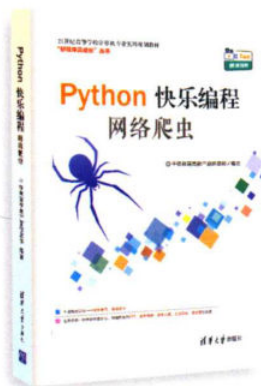
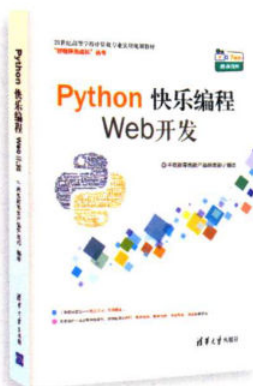
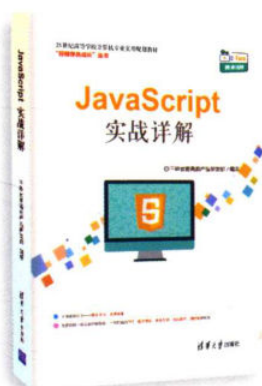
 千锋图书定位——快乐学习，实战就业

 免费提供教学资源，包括PPT、教学视频、教学大纲、考试系统、测试题等



清华大学出版社

◆ 丛书推荐 ◆



资源下载 · 扩展阅读



清华社官方微信号



ISBN 978-7-302-54126-4



9 787302 541264 >

定价：59.80元

“好程序员成长”丛书

Python 快乐编程 TensorFlow 深度学习

项目实战

©千锋教育高教产品研发部 / 编著

清华大学出版社
北京

此为试读, 需要完整PDF请访问: www.ertongbook.com

内 容 简 介

TensorFlow 是目前应用最为广泛的主流深度学习框架,由谷歌公司于 2015 年开源。本书是 TensorFlow 的入门书籍,旨在帮助读者快速理解该框架的基本概念和操作方法,初步掌握通过 TensorFlow 解决实际问题的能力。

对于刚接触深度学习的初学者而言,一本简单易懂、易上手的教材至关重要。本书摆脱纯理论的讲解方式,将理论与实际操作相结合,通过丰富的代码实例和详尽的步骤介绍,与读者一起“跳出枯燥,快乐学习”。全书包括 14 章。前 3 章从 TensorFlow 简介与安装入手,帮助读者快速了解该框架的基础概念和操作方法;第 4 章介绍线性回归算法;第 5 章讲述神经网络算法基础;第 6~9 章分别通过实例详细讲解手写数字识别问题、卷积神经网络算法、图像数据处理和循环神经网络算法;第 10~13 章介绍 TensorFlow 产品化、高层封装、可视化等实用技巧,以及遗传算法、K-means 算法等进阶操作;第 14 章通过汽车车牌识别项目的实现来进一步学习 TensorFlow 在图像识别方面的应用。

本书适合于想系统学习深度学习和应用 TensorFlow 的相关从业人员以及在校学生等。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Python 快乐编程:TensorFlow 深度学习项目实战/千锋教育高教产品研发部编著. —北京:清华大学出版社,2020.2

(“好程序员成长”丛书)

ISBN 978-7-302-54126-4

I. ①P… II. ①千… III. ①软件工具—程序设计 ②机器学习 IV. ①TP311.561 ②TP181

中国版本图书馆 CIP 数据核字(2019)第 246896 号

责任编辑:黄 芝

封面设计:胡耀文

责任校对:焦丽丽

责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-83470236

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:17

字 数:411 千字

版 次:2020 年 2 月第 1 版

印 次:2020 年 2 月第 1 次印刷

印 数:1~1500

定 价:59.80 元

产品编号:083972-01



为什么要写这样一本书

当今世界是知识爆炸的世界,科学技术与信息技术急速发展,新技术层出不穷。但教科书却不能将这些知识内容及时编入,致使教科书的知识内容出现陈旧不实用的问题,教材的陈旧性与滞后性尤为突出。在初学者还不会编写一行代码的情况下,就开始讲解算法,这样只会吓跑初学者,让初学者难以入门。

IT 行业不仅需要理论知识,更需要实用型、技术过硬、综合能力强的人才,高校毕业生求职面临的第一道门槛就是技能与经验的考验。学校通常注重学生的素质教育和理论知识,忽略了对学生的实践能力培养,在高校毕业生求职时,学生常常因技术经验不足而被拒之门外。

如何解决这一问题

为了杜绝这一现象的发生,本书倡导的理念是快乐学习,实战就业。在语言描述上力求准确、通俗易懂,在章节编排上力求循序渐进,在语法阐述时尽量避免术语和公式,从项目开发的实际需求入手,将理论知识与实际应用相结合。旨在让初学者能够快速成长为初级程序员,并拥有一定的项目开发经验,从而在职场中拥有一个较高的起点。



千锋教育

前言

在瞬息万变的 IT 时代,一群怀揣梦想的人创办了千锋教育,投身到 IT 培训行业。自 2011 年以来,一批批有志青年加入千锋教育,为了梦想笃定前行。千锋教育秉承用良心做教育的理念,为培养“顶级 IT 精英”而付出一切努力,为什么会有这样的梦想,我们先来听一听用人企业和求职者的心声:

“现在符合企业需求的 IT 技术人才非常紧缺,这方面的优秀人才我们会像珍宝一样对待,可为什么至今没有合格的人才出现?”

“面试的时候,用人企业问能做什么,这个项目如何来实现,需要多长的时间。我们当时都蒙了,回答不上来。”

“这已经是面试过的第十家公司了,如果再不行的话,是不是要考虑转行了,难道大学里的四年都白学了?”

“这已经是参加面试的第 N 个求职者了,都是计算机专业的,当问到项目如何实现,怎么连思路都没有呢?”

这些心声并不是个别现象,而是中国社会目前的一种普遍现象。高校的 IT 教育与企业的真实需求脱节,如果高校的相关教材不能与时俱进,毕业生将面临难以就业的困境。很多用人单位表示,高校毕业生表象上知识丰富,但绝大多数在实际工作中用之甚少,甚至完全用不上高校学习阶段所学知识。针对上述存在的问题,国务院也作出了关于加快发展现代职业教育的决定。很庆幸,千锋所做的事情就是配合高校达成产学合作。

千锋教育致力于打造 IT 职业教育全产业链人才服务平台,在全国拥有数十家分校,数百名讲师。我们坚持以教学为本的方针,采用面对面教学的方式,传授企业实用技能。教学大纲实时紧跟企业需求,拥有全国一体化就业体系。千锋的价值观是“做真实的自己,用良心做教育”。

针对高校教师的服务:

(1) 千锋教育自 2011 年以来不断优化 IT 教育培训课程,精心设计了包含“教材+授课资源+考试系统+测试题+辅助案例”的教学资源包,节约教师的备课时间,缓解教师的教学压力,显著提高教学质量。

(2) 本书配套课件、源代码、习题答案、教学大纲等电子资源,可从清华大学出版社网站下载。

(3) 本书配备了千锋教育优秀讲师录制的教学视频,按本书知识结构体系部署到了教学辅助平台(扣丁学堂)上,也可直接扫描封底刮刮卡二维码,登录文泉课堂观看视频,可以作为教学资源使用,也可以作为备课参考。

高校教师如需索要配套教学资源,请关注(扣丁学堂)师资服务平台,扫描下方二维码关注微信公众平台索取。



扣丁学堂

针对高校学生的服务:

(1) 学 IT 有疑问,就找“千问千知”。它是一个有问必答的 IT 社区,平台上的专业答疑辅导老师承诺工作时间 3 小时内答复您学习 IT 时遇到的专业问题。读者也可以通过扫描下方二维码,关注千问千知微信公众平台,浏览其他学习者在学习分享的问题和收获。



千问千知

(2) 学习太枯燥,想了解其他学校的伙伴都是怎样学习的? 你可以加入扣丁俱乐部。“扣丁俱乐部”是千锋教育联合各大校园发起的公益计划,专门面向对 IT 感兴趣的大学生提供免费的学习资源和问答服务,已有超过 30 多万名学习者获益。

就业难,难就业,千锋教育让就业不再难!

关于本教材

本教材可作为高等院校本、专科计算机相关专业的 Python 人工智能入门与进阶教材,包含了千锋教育 Python 人工智能课程的精华内容,是一本适合广大计算机编程爱好者的优秀读物。

抢 红 包

本书配套源代码、习题答案的获取方法:添加小千 QQ 号或微信号 2133320438。

注意! 小千会随时发放“助学金红包”。

致 谢

千锋教育 Python 教学团队将多年积累的教学实战案例进行整合,通过反复精雕细琢最终完成了这本著作。另外,多名院校老师也参与了教材的部分编写与指导工作。除此之外,千锋教育 500 多名学员也参与了教材的试读工作,他们站在初学者的角度对教材提供了许多宝贵的修改意见,在此一并表示衷心的感谢。

千锋学科

HTML5 前端开发、JavaEE 分布式开发、Python 全栈+人工智能、全链路 UI/UE 设计、智能物联网+嵌入式、360 网络安全学院、大数据+人工智能培训、全栈软件测试、PHP 全栈+服务器集群、云计算+信息安全、Unity 游戏开发、区块链。

千锋校区

北京|大连|广州|成都|杭州|长沙|哈尔滨|南京|上海|深圳|武汉|郑州|西安|青岛|重庆|太原|合肥

意见反馈

在本书的编写过程中,编者虽然力求完美,但难免有一些不足之处,欢迎各界专家和读者朋友们给予宝贵意见。

编 者

2019 年 9 月于北京

目 录

第 1 章 初识 TensorFlow	1
1.1 深度学习介绍	1
1.2 TensorFlow 简介	4
1.3 TensorFlow 环境搭建	7
1.4 TensorFlow 测试	11
1.5 本章小结	12
1.6 习题	12
第 2 章 TensorFlow 基础	14
2.1 张量	14
2.2 会话	17
2.3 变量与占位符	18
2.4 矩阵	20
2.4.1 创建矩阵	20
2.4.2 矩阵基本运算	21
2.5 本章小结	22
2.6 习题	23
第 3 章 TensorFlow 进阶	25
3.1 TensorFlow 的计算模型	25
3.1.1 计算图的工作原理	25
3.1.2 计算图的使用	26
3.2 TensorFlow 的嵌入层	28
3.3 TensorFlow 的多层	29
3.4 TensorFlow 实现损失函数	31
3.4.1 损失函数	31
3.4.2 损失函数工作原理及实现	34
3.5 TensorFlow 实现反向传播	37
3.5.1 反向传播算法	37
3.5.2 反向传播算法的工作原理及实现	38

3.6	TensorFlow 实现随机训练和批量训练	42
3.7	TensorFlow 创建分类器	47
3.8	TensorFlow 实现模型评估	51
3.8.1	模型评估方法	51
3.8.2	模型评估工作原理及实现	51
3.9	本章小结	56
3.10	习题	56
第 4 章 基于 TensorFlow 的线性回归		58
4.1	线性回归简介	58
4.2	TensorFlow 求逆矩阵	63
4.3	TensorFlow 求矩阵的分解	65
4.4	TensorFlow 实现线性回归算法	68
4.5	线性回归中的损失函数	71
4.6	TensorFlow 实现戴明回归	74
4.7	TensorFlow 实现 Ridge 回归与 Lasso 回归	78
4.8	TensorFlow 实现逻辑回归	83
4.9	本章小结	86
4.10	习题	86
第 5 章 神经网络算法基础		88
5.1	神经网络算法简介	88
5.2	TensorFlow 实现激活函数	89
5.2.1	Sigmoid 函数	89
5.2.2	Tanh 函数	90
5.2.3	ReLU 数	91
5.3	TensorFlow 实现单层神经网络	92
5.4	TensorFlow 实现神经网络常见层	94
5.5	本章小结	98
5.6	习题	98
第 6 章 数字识别问题		99
6.1	MNIST 数据处理	99
6.2	神经网络模型训练进阶	102
6.2.1	程序与数据的拆分	102
6.2.2	变量管理	108
6.3	TensorFlow 模型持久化	111
6.3.1	TensorFlow 实现保存或加载模型	111
6.3.2	TensorFlow 模型持久化的原理及数据格式	113

6.4	本章小结	119
6.5	习题	119
第 7 章	TensorFlow 实现卷积神经网络	120
7.1	卷积神经网络简介	120
7.2	TensorFlow 实现简单的 CNN	126
7.3	TensorFlow 实现进阶 CNN	131
7.4	TensorFlow 实现图片风格渲染	138
7.5	本章小结	143
7.6	习题	143
第 8 章	图像数据处理	145
8.1	TFRecords	145
8.2	图像数据的预处理	148
8.2.1	图像预处理方法简介	148
8.2.2	图像预处理实例	158
8.3	多线程输入数据处理框架	160
8.3.1	队列与多线程	160
8.3.2	输入文件队列	163
8.3.3	组合训练数据	165
8.4	数据集的使用方法	167
8.5	本章小结	172
8.6	习题	172
第 9 章	TensorFlow 实现循环神经网络	174
9.1	循环神经网络简介	174
9.2	通过 TensorFlow 实现垃圾短信预测	175
9.3	通过 TensorFlow 实现 LSTM 模型	180
9.4	通过 TensorFlow 实现多层 LSTM 模型	186
9.5	本章小结	193
9.6	习题	193
第 10 章	TensorFlow 产品化	194
10.1	TensorFlow 的单元测试	194
10.2	TensorFlow 并发执行	198
10.3	TensorFlow 分布式实践	199
10.4	TensorFlow 产品化开发	200
10.5	本章小结	202
10.6	习题	202

第 11 章 TensorFlow 的进阶用法	204
11.1 TensorFlow 实现遗传算法	204
11.2 TensorFlow 实现 K-means 算法	208
11.3 TensorFlow 求解常微分方程	212
11.4 本章小结	213
11.5 习题	214
第 12 章 TensorFlow 高层封装	215
12.1 TensorFlow 的常见封装方法简介	215
12.2 Keras	217
12.2.1 序贯模型	218
12.2.2 函数式模型	220
12.3 Estimator	222
12.3.1 Estimator 的基本用法	223
12.3.2 Estimator 自定义模型	225
12.4 本章小结	227
12.5 习题	228
第 13 章 TensorFlow 可视化	229
13.1 TensorBoard 简介	229
13.2 TensorBoard 可视化	231
13.2.1 TensorFlow 命名空间与 TensorBoard 图上节点	231
13.2.2 TensorBoard 节点信息	236
13.3 本章小结	238
13.4 习题	238
第 14 章 TensorFlow 实现车牌识别	239
14.1 项目简介	239
14.2 生成训练数据集	240
14.3 数据读取	244
14.4 构建神经网络模型	246
14.5 开始模型训练	251
14.6 测试模型准确度	254
14.7 本章小结	257

本章学习目标

- 了解深度学习的概念和发展现状;
- 了解不同的深度学习工具及它们之间的区别;
- 掌握 TensorFlow 的环境搭建方法;
- 了解 TensorFlow 的测试方法。

目前,随着计算机和人工智能技术取得了突飞猛进的进步,人工智能有望成为大多数行业的助力。在即将到来的人工智能时代,深度学习技术已成为人工智能领域的重要技术支撑。而 TensorFlow 作为目前最主流的开源人工智能学习系统,在深度学习领域有着举足轻重的作用,它将复杂的数据结构传输至人工智能神经网络中进行处理和分析。其开源性也大大降低了深度学习在各个行业中的应用难度。本书将与大家一起循序渐进地学习 TensorFlow 的有关知识,希望通过本书的学习大家可以掌握常见的 TensorFlow 用法,以及训练神经网络的方法。本章将对 TensorFlow 的基础知识和安装方法进行讲解。

1.1 深度学习介绍

深度学习是机器学习的一个重要分支,通过构建具有多个隐藏层的机器学习模型和海量的训练数据来学习更有用的特征,从而最终提升分类或预测的准确性。用一句话对人工智能、机器学习和深度学习进行概括:机器学习是实现人工智能的一种方法,而深度学习则是实现机器学习的一种技术。

很多人往往混淆机器学习与深度学习的概念,不妨看一下台湾大学李宏毅教授对机器学习的定义:机器学习在形式上,近似于在数据对象中通过统计或推理的方法寻找一个适用特定输入和预期输出功能的函数,如图 1.1 所示。

所谓机器学习,在形式上可以看作一个函数,通过对特定的输入进行处理,得到一个预期的结果。例如: $f(\text{一段音频}) = \text{"你好"}$, $f(\text{含有猫的图片}) = \text{"猫"}$ 。但是如何才能让计算机在接收一串语音后知道这句话是“你好”而不是其他内容呢?这就需要构建一个评估体系来判断计算机通过学习是否能够输出理想的结果,如此便可以通过训练数据(training data)来“培养”机器学习算法的能力,如图 1.2 所示。



图 1.1 机器学习可以看作一个函数

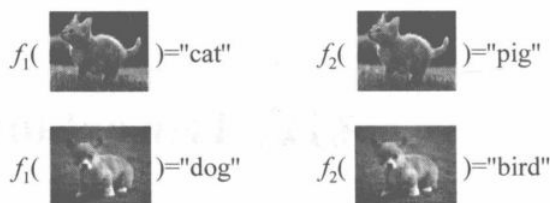


图 1.2 机器学习的过程

从图 1.2 可以看出, f_2 对图像的识别是错误的, 学习效果并不理想, 经过训练数据的“培养”, 将输出结果不理想的 f_2 改善为输出结果较为理想的 f_1 , 判定的准确度提高了, 这种改善的过程便可以称为学习。这个学习过程是由机器完成的, 这便是“机器学习”。

深度学习则是机器学习的一个分支, 它在涉及语音、图像等复杂对象的应用中具有优越的性能。与人工规则构造特征的方法相比, 利用大数据来学习特征, 更能够刻画数据丰富的内在信息。在大数据时代, 深度学习的神经网络模型可以更加有效地捕捉海量数据中所含的巨量复杂信息, 并对未来或未知事件做更精准的预测。与以往的机器学习相比, 深度学习对使用者的要求有所降低, 使用者只需调节相关参数, 学习的效果一般都较为理想, 这促进了机器学习从实验技术走向工程实践。以上只是对深度学习的一个简单概括, 并不能全面地解释什么是深度学习, 因为神经网络中的深层构架差异巨大, 对不同任务或目标的优化会有不同的操作。通过机器学习的发展历程来理解深度学习可能是一种更好的方法。

在人工智能的发展初期, 计算机主要发挥了善于处理形式化的数学规则的特性, 比人类更加快速、高效地完成形式化的任务。这让人工智能在初期相对朴素和形式化的环境中取得了成功, 这种环境对计算机所需具备的关于世界的知识要求很低。例如, 在形式和规则十分固定的国际象棋领域, 人工智能取得了巨大成就。在 1997 年, IBM 公司研制的“深蓝”(Deep Blue)击败了当时的国际象棋世界冠军 Garry Kasparov。事实上, 一台计算机理解国际象棋中固定的 64 格棋盘、严格按照规则进行移动的 32 个棋子以及胜利条件并不难, 相关概念完全可以由一个非常简短、完全形式化的规则列表进行描述并输入计算机中。

然而, 在处理抽象的、非形式化的任务时, 人工智能却显得比人类“笨拙”得多, 人工智能的处理水平往往难以达到人类平均水平。例如, 人类可以很轻松地通过直觉识别出静物油画中的一串香蕉, 但是机器却难以识别被油画抽象出的“香蕉”。如今随着人工智能相关领域的飞速发展, 计算机对于非形式化任务的处理能力取得了巨大进步, 计算机完成识别对象和语音任务的能力已经达到人类的一般水平。人类的大脑中存储了巨量的有关世界的知识来维持日常生活的需要, 让计算机实现强人工智能就需要让其理解这些关于世界的巨量知识。然而, 许多相关知识具有主观性, 难以通过形式化的方法进行描述, 让计算机理解这些非形式化的知识无疑是人工智能的一项巨大挑战。

在了解深度学习的神经网络工作原理之前, 有必要先了解一下人类大脑的工作机理。在 1981 年, Hubel、Wiesel 和 Sperry 等人发现了一种可以有效地降低反馈神经网络复杂性的独特的神经网络结构, 进而提出了卷积神经网络。卷积神经网络的发现揭示了人类视觉的分级系统, 在收到视觉刺激后, 信息从视网膜出发, 经过低级区时提取目标的边缘特征, 在高一级的区域对目标的基本形状或目标的局部进行识别, 再到下一层更高级的区域对整个目标进行识别, 以及到更高层的前额叶皮层进行分类判断等, 即高层的特征是低层特征的组

合,信息的表达由低层到高层越来越抽象和概念化。这个发现激发了人们对于神经系统的进一步思考,大脑的工作过程是一个对接收信号不断迭代、不断抽象和概念化的过程。以识别油画中的香蕉为例,首先摄入原始信号(瞳孔摄入像素),然后进行初步处理(大脑皮层某些神经细胞发现香蕉的边缘和方向),对处理后的信息进行抽象(大脑判定香蕉的形状,比如是长型略微弯曲的),进一步抽象(大脑进一步判定该物体是香蕉),最后识别出图中画的是一串香蕉。由此例可以看出,大脑是一个深度架构,认知的过程是通过大脑逐层分级处理表示的信息实现的。

无论在计算机科学领域还是人类的日常生活中,信息的表示起到了至关重要的作用。例如,大多数学生可能已经习惯了阅读国内英语考试中全部由小写字母组成的文章,可以很快地阅读并完成后面的题目,但是在有些国际性英语能力测试中会出现全部由大写英文字母组成的文章,这时考生可能就需要花更多的时间去适应大写字母组成的单词。同样的单词以不同的表示方式出现,会对考生的阅读产生巨大的影响。相应地,不同的表示方式同样会对机器学习的算法性能产生影响。接下来通过图示的方法展示不同表示方式对算法性能的影响,如图 1.3 所示。

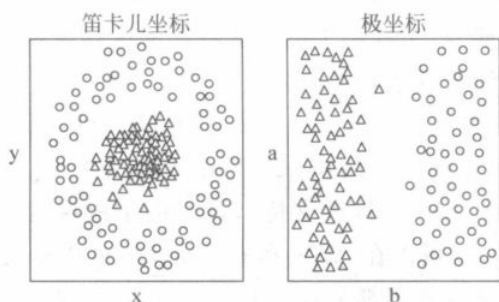


图 1.3 两种不同的表示方式

在图 1.3 中,左图使用了笛卡尔坐标表示两种类型的数据。显然,在这种表示方式下,无法用一条直线来分隔三角形和圆两种类型的数据;而右图使用极坐标表示,可以很容易用一条垂直的线将两种类型的数据分隔开。

一般情况下,处理人工智能的方法可以概括为:提取一个恰当的特征集,然后将这些特征提供给简单的机器学习算法。例如,在语音识别中,对声道大小这一特征的识别可以作为判断说话者的性别以及大致年龄的重要线索。

然而在大多数情况下,人类很难确定应该提取的信息特征。例如,希望一个程序能够检测出油画中的水果——香蕉。香蕉的特征有黄色的果皮,长型略微弯曲的外形,但是仅以油画中的某一个像素值很难准确地描述香蕉看上去像什么,因为不同的场景下香蕉的摆放角度和光影效果都会不同,如图 1.4 所示。

为了解决上述问题,此时就需要让计算机自身去发掘表示的特征。通过学习让程序去理解一个表示的特征往往比直接输入人为总结的特征更加准确。这就要求计算机学会从原始数据中提取高层次的、抽象的特征。

深度学习让计算机可以通过组合低层特征形成更加抽象的高层特征(或属性类别)。深度学习算法可以从原始图像去学习一个低层次表达,例如边缘检测器、小波滤波器等,然后在这些低层次表达的基础上,通过线性或者非线性组合,来获得一个高层次的表达。

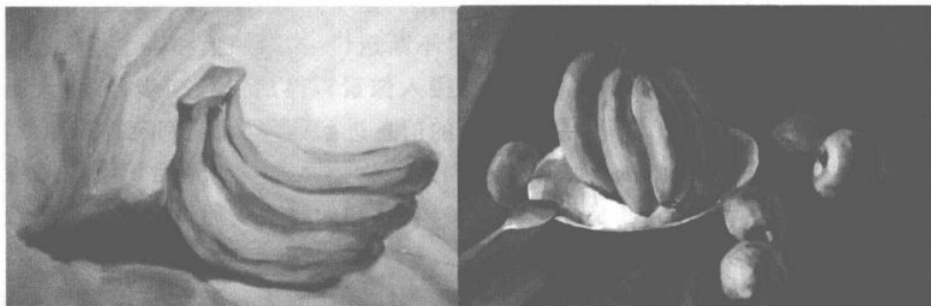


图 1.4 两种不同情景下的香蕉

1.2 TensorFlow 简介

1.1 节主要对深度学习的概况进行了介绍,深度学习的发展离不开相关算法和学习工具的出现与改进。目前,TensorFlow 是众多深度学习工具中最受欢迎的一个。它最初由 Google 公司的 Brain 小组研发,主要用于机器学习和深度神经网络方面的研究,事实上,该系统的通用性使其被广泛用于其他算法中。

TensorFlow 是一个通过数据流图(data flow graphs)对张量(tensor)进行计算的开源软件,这也是 TensorFlow 软件名称的由来。TensorFlow 的编程接口支持 C++、Python、Java、Go 和 R 语言(Haskell API 也将被支持),是目前主流深度学习框架中支持的开发语言最全面的框架。TensorFlow 可以在 AWS 和 Google Cloud 中运行,支持 Windows、Linux 和 Mac 等操作系统。本书所有的代码都是基于 Python 语言在 Windows 上实现和运行的。

TensorFlow 使用 C++ Eigen 库,可以在 ARM 架构上编译和优化,使得它可以在各种服务器和移动设备上部署自己的训练模型,这让它成为了支持运行平台最多的深度学习框架。

TensorFlow 基于计算图实现自动微分系统,使用数据流图进行数值计算,图中的节点代表数学运算,图中的线条则代表在这些节点之间传递的张量(多维数组)。TensorFlow 追求对运行平台和开发语言最广泛的支持,力求统一深度学习领域,但是这也使得 TensorFlow 的系统设计趋于复杂化,TensorFlow 在 GitHub 上的总代码已超过 100 万行。它在接口设计中创造了很多新的抽象概念,如计算图、会话、命名空间和 Place-Holder 等,同一个功能又提供了多种实现方法,使用上可能有细微的区别,频繁接口变更也导致了向后兼容性上的问题。由于直接使用 TensorFlow 过于复杂,Google 官方在内的很多开发者尝试构建一个高级 API 作为 TensorFlow 更易用的接口,目前已知的高级 API 包括 Keras、Sonnet、TFLearn、TensorLayer、Slim、Fold 和 PrettyLayer 等。其中 Keras 在 2017 年成为第一个被 Google 公司添加到 TensorFlow 核心中的高级框架,这让 Keras 变成 TensorFlow 的默认 API,使 Keras + TensorFlow 的组合成为 Google 官方认可并大力支持的平台。在本书后续章节中,将介绍 Keras 的相关用法。

前面提到了数据流图,大家可能对这个词有些陌生。数据流图通过“节点”和“线”的有向图来描述数学计算。它从数据传递和加工的角度,以图形方式来表达系统的逻辑功能、数据在系统内部的逻辑流向和逻辑变换过程,是结构化系统分析方法的主要表达工具及用

于表示软件模型的一种图示方法。“节点”可以用来表示施加的数学操作,也可以表示数据输入的起点或数据输出的终点,或者读取/写入持久变量的终点。“线”用来表示节点之间的输入或输出关系,通过“线”来输送张量。当输入端的所有张量准备就绪后,“节点”将被分配到各种计算设备,异步、并行地执行运算。

TensorFlow 使用了向量运算的符号图方法,简化了指定新的网络的方法,可以自定义神经网络结构,支持并行设计,充分利用硬件资源。由于其支持自动求微分,这省去了通过反向传播求解梯度的步骤。符号图方法支持自由的算法表达,可实现深度学习以外的机器学习算法,因此其在深度学习以外的许多领域也受到青睐。TensorFlow 所具备的优势具体如下。

1. 高灵活性

TensorFlow 并不仅仅是一个服务于人工神经网络的库,其采用的符号图方法支持自由的算法表达,可实现深度学习以外的机器学习算法。TensorFlow 提供了相关工具来组合“子图”,用户可以在 TensorFlow 的基础上编写自己的“上层库”。在 TensorFlow 中可以用类似 Python 中自定义新函数的方法定义一个新的复合操作。TensorFlow 的底层代码由 C++ 编写,也可以在需要时用 C++ 代码来为底层添加功能。

2. 良好的可移植性

TensorFlow 既可以在 CPU 上也可以在 GPU 上运行,在不改变代码的情况下,TensorFlow 可以实现训练模型在多个 CPU 上进行规模化的运算,甚至可以将训练好的模型植入手机 APP 中加以应用,或者作为云端服务运行在云服务器上。

3. 将科研和产品联系在一起

过去如果要将科研中的机器学习想法用到产品中,需要大量的代码重写工作。在 Google 公司,科学家用 TensorFlow 尝试新的算法,产品团队则用 TensorFlow 来训练和使用计算模型,并直接提供给在线用户。使用 TensorFlow 可以让应用型研究者将想法迅速应用到产品中,也可以让学术性研究者更直接地彼此分享代码,从而提高科研产出率。

4. 自动求微分

基于梯度的机器学习算法会受益于 TensorFlow 自动求微分的能力。作为 TensorFlow 用户,你只需要定义预测模型的结构,将这个结构和目标函数(objective function)结合在一起并添加数据,TensorFlow 将自动为你计算相关的微分导数。计算某个变量相对于其他变量的导数仅通过扩展图结构来完成,这种方式更加直观。

5. 多语言支持

TensorFlow 有一个合理的 C++ 使用界面,也有一个易用的 Python 使用界面来构建和执行 graphs。在 TensorFlow 中可以直接编写 Python/C++ 程序,也可以通过例如 IPython 这样的交互界面来使用 TensorFlow。它同时鼓励用户创造自己最喜欢的语言界面,比如 Go、Java、Lua、JavaScript 或者 R 语言。

6. 性能最优化

TensorFlow 可以通过线程、队列和异步操作等功能使得工作站的计算潜能更充分地发挥出来。可以通过自定义的方式将 TensorFlow 图中的计算元素分配到不同设备上,TensorFlow 可以帮你管理好这些不同副本。

以上便是 TensorFlow 所具备的主要优点,接下来介绍另外三个常见的深度学习工具。