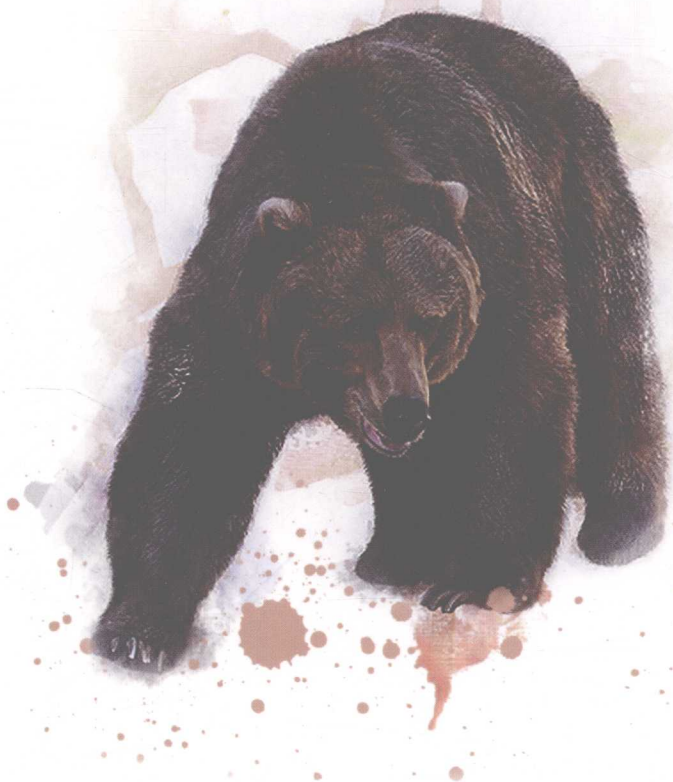




深度学习图像识别技术

基于TensorFlow Object Detection API
和OpenVINO™工具套件

庄建 张晶 许钰雯 编著



机械工业出版社
CHINA MACHINE PRESS

深度学习图像识别技术：基于 TensorFlow Object Detection API 和 OpenVINO™ 工具套件

庄 建 张 晶 许钰雯 编著

机械工业出版社

本书首先讲述了人工智能、深度学习、卷积神经网络、目标检测以及迁移学习的概念,接着详述了如何基于 TensorFlow Object Detection API 框架从零开始搭建深度学习目标检测应用的开发环境,如何训练自己的目标检测模型,然后详述了如何基于 OpenVINO™ 工具套件优化模型、部署模型以及用 C++ 和 Python 开发用户应用程序。最后通过工业光学字符自动识别、垃圾瓶自动分选、农作物病害自动识别和工业产品外观缺陷检测 4 个完整的深度学习目标检测工程案例来帮助读者加深深度学习图形检测的认识和理解。

本书适合从事 AI 行业相关技术的工程师阅读,也适合打算进入 AI 行业的大中专院校的学生学习参考。

图书在版编目(CIP)数据

深度学习图像识别技术:基于 TensorFlow Object Detection API 和 OpenVINO™ 工具套件 / 庄建, 张晶, 许钰雯编著. —北京:机械工业出版社, 2020.2
ISBN 978-7-111-64598-6

I. ①深… II. ①庄… ②张… ③许… III. ①人工智能—算法—应用—
图象识别 IV. ①TP391.413

中国版本图书馆 CIP 数据核字(2020)第 006144 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:张淑谦 责任编辑:张淑谦

责任校对:张艳霞 责任印制:孙 炜

保定市 中画美凯印刷有限公司印刷

2020 年 3 月·第 1 版第 1 次印刷

184mm×260mm·17.75 印张·437 千字

0001—2500 册

标准书号:ISBN 978-7-111-64598-6

定价:89.00 元

电话服务

客服电话:010-88361066

010-88379833

010-68326294

封底无防伪标均为盗版

网络服务

机 工 官 网: www.cmpbook.com

机 工 官 博: weibo.com/cmp1952

金 书 网: www.golden-book.com

机工教育服务网: www.cmpedu.com

序

最近十几年，随着以深度学习算法为推动力的人工智能产业化落地的日趋成熟，基于深度学习的应用开发也越来越流行。深度学习的开发本身涉及的知识面涵盖了算法基础理论、算法的代码实现和优化、数据工程（如数据的清洗、标注、预处理等）、主流算法框架和神经网络模型的训练及推理优化、硬件平台的选择及最终部署前的系统级优化等，其知识结构的广泛性和复杂性往往使初学者以及更加关注应用本身开发和部署的开发人员需要更长的学习时间或更陡的学习曲线才能掌握。

本书站在应用开发者（既包括专业人员，也包括大专院校相关专业的师生和业余开发爱好者）的角度，以应用深度学习目标检测技术为目标，侧重介绍了基于深度学习的人工智能的开发环境、模型训练及推理部署的全过程。特别是给出了一个在 Windows 平台上搭建深度学习开发平台、基于英特尔®OpenVINO™在 Visual Studio 中开发深度学习应用程序的方法，并在最后的 4 章中着重介绍了 4 个具体应用场景的工程实践，大大降低了深度学习推理部署的学习难度。

在介绍优化并部署模型和进一步提升 AI 推理性能的内容时，本书不仅详述了如何通过使用英特尔®OpenVINO™开发工具套件的优化原理且辅以详尽的代码示例，同时还提示了开发者如何针对特定的应用场景，在开发过程中预先思考最终部署的硬件平台选型。对于适用于物联网或边缘计算的推理应用方案，方案的性价比、对于未来业务的扩展性和所选硬件平台的通用性都需要得到一个综合性的评估。开发者通过对 OpenVINO™的异步 API 和多设备 API 的调用，在英特尔公司提供的 X86 通用 CPU 芯片系列（如酷睿和至强），FPGA 芯片系列（如 Arria）和 VPU 芯片系列（如 Movidius）上，可以自动实现部署所需要的优化，很好地体现了软硬件协同应用开发的优势。对于快速商业化创新的应用方案起到了重要的赋能作用。

如果读者有意向长期致力于 AI 应用开发，建议在阅读本书及着手尝试之余，阅读更多的相关书籍及其他参考资料，使自己具备更完整的知识体系。

英特尔亚太研发有限公司物联网事业部视觉产品研发总监 顾典



前言

为什么要写这本书

2016 年 3 月 9~15 日，Google 子公司 DeepMind 开发的人工智能围棋程序 AlphaGo 4：1 战胜了拥有 18 个世界冠军头衔的棋王李世石，这个事件让以深度学习技术为代表的人工智能（AI）出现在公众的面前，成为茶余饭后最火爆的谈资之一，如图 1 所示。AI 一夜之间火遍大江南北，最直观的感受是：各种 AI 新闻刷屏各大科技媒体；AI 成为资本界的宠儿，一个接一个的 AI 独角兽不断出现，估值飞涨；各大高校纷纷开设人工智能学院；AI 算法工程师的薪资不断刷新人们的认知上限；以中美两国为代表的多个国家纷纷出台类似《新一代人工智能发展规划》的国家级行业发展推动政策。一时间，AI 风头无二，政府、资本界、工业界、学术界及媒体界纷纷投身到 AI 的洪流中。



图1 人工智能火爆

在 AI 项目落地的过程中，我们真真切切地感受到了 AI 技术（尤其是深度学习技术）的优势：基于深度学习算法的物体识别精度大大超越基于传统机器视觉算法的物体识别精度，让多年来因为识别精度达不到用户要求而无法落地的外观瑕疵检测应用在用户端落地。

随着 AI 项目落地越来越多，我们也越来越感到 AI 是一个绝对能赋能各行各业，提升行业运行效率，降低行业运行成本的好技术。从应用角度来说，AI 的主要作用是识别和预测；从技术角度来说，这是各行各业智能化和精准化的技术基础，例如，精准养

殖、精准种植、智能制造以及精准广告投放等。智能化精准化之后，必然带来成本节省、效率提升。随着 5G+大数据+云计算+物联网时代的来临，AI 必将爆发出超越人类想象的能力，成为一个让各行各业智能化精准化的基石。

参考计算机行业的发展历史，起初计算机只能在实验室里面被计算机科学家使用，后来计算机工具化后，人人都可以使用了。AI 行业发展也会这样，随着 AI 算法的工具化，会用 AI 技术的人也会从 AI 科学家扩展到每个人。

AI 头部公司（如谷歌、微软、百度和华为等）会提供 AI 技术平台和工具，全力做好开发者生态，支撑大量非 AI 行业的工程师都能快速学习并使用 AI 这个生产力工具，让大家在已经熟悉的“拖拉拽”和“所见即所得”的使用习惯中就可以把 AI 能力集成到自己的解决方案或产品中，解决本行业问题，如图 2 所示。这些使用 AI 工具、落地 AI 应用的工程师可以称之为 AI 应用工程师或 AI 应用开发者。

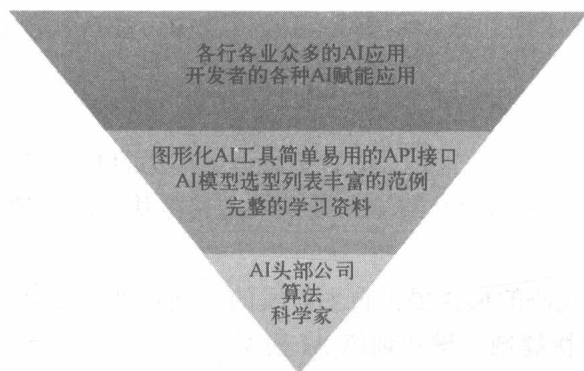


图2 AI 算法工具平台与生态

本书希望面向各行各业众多的 AI 应用开发者，以工具应用而非数学原理的视角，在边讲解边实践的过程中，让大家能够快速掌握并使用 AI 技术，解决本行业中的问题，推动行业的智能化和精准化。在应用 AI 技术不断解决行业问题的过程中反过来加深对 AI 技术及其背后数学原理的理解。

在 AI 项目实践过程中，我们发现了一个易学易用的开源深度学习目标检测训练框架 TensorFlow Object Detection API 和开源商用免费的深度学习推理计算加速工具包 OpenVINO™ 工具套件，基于这两个 AI 工具，我们可以在无需学习 AI 数学原理的情况下，落地 AI 目标检测（图像识别）应用。

由此，落地 AI 目标检测应用，就可以从学习 AI 数学原理、研发 AI 模型、训练 AI 模型以及部署 AI 模型这种非常艰难的学习研发过程，变成搭建 AI 工具 TensorFlow Object Detection API 运行环境、下载 AI 预训练模型、基于 AI 预训练模型训练自己的 AI 模型和部署 AI 模型这样相对容易的学习和使用过程，大大降低了应用 AI 技术的门槛，如图 3 所示。



图3 AI 算法工程师与 AI 应用工程师的对比

本书将按照搭建 AI 工具开发环境、训练 AI 模型和部署 AI 模型的顺序，介绍 TensorFlow Object Decction API 和 OpenVINO™ 工具套件的使用，希望众多对 AI 技术感兴趣的人，先把 AI 技术用起来，用到自己的行业中，在应用的过程中不断加深对 AI 技术的理解。

本书的特色

本书以快速学习 AI 工具，落地 AI 目标检测应用为目标，结合工程实践经验，以应用 AI 工具为视角，深入浅出地介绍 AI 技术，尤其是基于深度学习的目标检测（Object Detection）技术。

本书以 Python 源代码的形式提供自主研发的一键式训练工具，让读者在掌握每一个训练步骤后，可以方便快捷地一键式训练 AI 模型。

本书提供了 4 个完整的项目范例（含图片数据集），方便读者在边干边学的过程中快速掌握 AI 工具，形成自己对 AI 技术的认知和 AI 技术应用能力。

本书以初学者的视角，尽量把所有概念都以非数学的方式解释清楚，每一个操作步骤都会解释为什么要这样做，以方便初学者理解，不仅做到知其然，还要做到知其所以然。

本书的适用对象

第一、打算学习并入门 AI 技术的高中、大中专、职高以及本科学生。

第二、在各传统行业中，如农业、可再生资源、制造业及工业控制等，希望应用 AI 目标检测算法解决本行业问题的工程技术人员。

第三、已经在 AI 行业，但非 AI 算法工程师，希望自己动手亲自体验并学习 AI 技术的从业人员，如标注工程师、销售工程师、售后工程师、市场经理或产品经理等。

如何阅读本书

本书倡导“理解要做，不理解也先做，在做的过程中加深理解”的原则。对于初学者，当发现有一两个名词或概念不熟悉不理解的时候，先照着书中的指导做，在做的过程中会渐渐理解。

“学以致用、边学边用、急用先学、立竿见影”，希望读者带着自己的项目需求来阅读本书，一边学习 AI 技术，一边思考如何应用 AI 技术解决自己的问题。

本书一共 9 章，第 1~5 章属于基础篇，分别介绍 AI、深度学习和目标检测技术的基础知识，搭建 AI 工具运行环境，训练 AI 模型，部署 AI 模型和在部署过程中如何进一步提升 AI 推理计算性能；第 6~9 章属于实战篇，通过 4 个真实项目案例和数据集，供读者在实践中加深对深度学习目标检测技术的理解。

勘误和反馈

由于笔者的水平有限，书中难免会出现错误或者不准确的地方，恳请读者批评指正。读者可以将书中的错误以及遇到的任何问题反馈给我们，我们将尽量在线为读者提供最满意的解答。反馈意见接收邮箱地址：dlod_openvino@163.com。

本书勘误主页为：<https://github.com/dlod-openvino/book/blob/master/README.md>。

本书配套的项目源代码以及图片数据读者可通过两种途径获取：

- 360 网盘下载：https://yunpan.360.cn/surl_yrkBV8cRhDT（提取码：d52e）。
- 见图书封底说明文字。

致谢

本书在编写过程中得到了广大企事业单位及科研人员的大力支持：英特尔（中国）有限公司的李勇、周兆靖、许鑫和张智勇提供了 OpenVINO™ 工具套件技术支持并对本书进行了校阅；神州数码（中国）有限公司提供了幻影峡谷样机；广东荣旭智能技术有限公司提供了工程案例图片；上海诺达佳自动化技术有限公司提供了 AIPC 样机 eBox3000；本书的代码在深圳市集和诚科技发展有限公司提供的 GPU 箱体电脑 BRAV-7302 上完成了测试。

本书在出版的过程中还得到了机械工业出版社张淑谦等编辑的无私帮助与支持，在此一并表示感谢！

编 者

目 录

序

前言

第 1 章 人工智能、深度学习与

目标检测 1

1.1 人工智能简介 1

1.1.1 什么是人工智能 1

1.1.2 人工智能发展简史 2

1.1.3 人工智能与深度学习的关系 4

1.2 深度学习简介 4

1.2.1 神经网络 4

1.2.2 神经元 5

1.2.3 深度神经网络 7

1.2.4 深度卷积神经网络 7

1.3 目标检测 9

1.3.1 目标检测算法发展简史 10

1.3.2 深度学习目标检测算法 11

1.4 迁移学习简介 13

1.4.1 训练深度学习模型依赖大数据 13

1.4.2 大数据造成的问题 14

1.4.3 迁移学习 14

1.4.4 TensorFlow 预训练模型库 16

1.5 本章小结 17

第 2 章 搭建深度学习开发环境 18

2.1 深度学习训练所需的硬件 18

2.1.1 英伟达显卡选型 18

2.1.2 英伟达显卡驱动安装 20

2.1.3 测试驱动程序安装 22

2.1.4 设置英特尔®集成显卡为系统主显示 输出 22

2.1.5 幻影峡谷：便携式 AI 训练 “服务器” 23

2.2 深度学习开发环境所需的软件 24

2.3 安装 Python 和 Anaconda 25

2.3.1 Python 和 Anaconda 简介 25

2.3.2 下载并安装 Anaconda 25

2.3.3 测试 Anaconda 安装 27

2.3.4 配置 Anaconda 软件包下载服务器 28

2.3.5 配置虚拟环境 tf_gpu 30

2.3.6 Anaconda 的进阶学习 31

2.4 安装 Visual Studio Code 31

2.4.1 Visual Studio Code 简介 31

2.4.2 安装 31

2.4.3 在 Visual Studio Code 中编辑 Python 代码 33

2.4.4 在 Visual Studio Code 中运行 Python 代码 34

2.4.5 在 Visual Studio Code 中调试 Python 代码 34

2.4.6 在 Visual Studio Code 安装 Pylint 35

2.4.7 在 Visual Studio Code 中一键 美化 Python 代码 35

2.5 安装 TensorFlow 37

2.5.1 TensorFlow 简介 37

2.5.2 下载并安装 38

2.5.3 测试安装 40

2.5.4 pip install 与 conda install 41

2.6 安装 Git 工具 41

2.6.1 Git 简介 41

2.6.2 下载并安装 41

2.6.3 测试安装 46

2.7 安装 TensorFlow Object Detection API 框架 46

2.7.1 TensorFlow Object Detection API 简介 46

2.7.2 下载并安装	47	3.8 训练模型	84
2.7.3 安装依赖的 python 软件包	48	3.9 使用 TensorBoard 观察训练 过程	85
2.7.4 配置环境变量	49	3.9.1 什么是 TensorBoard	85
2.7.5 安装 COCO API	50	3.9.2 TensorBoard 的使用方法	85
2.7.6 编译 proto 文件	53	3.10 评估训练好的模型	87
2.7.7 测试安装	54	3.11 导出训练好模型的冻结图	89
2.8 安装 LabelImg	56	3.11.1 检查点文件	89
2.8.1 LabelImg 简介	56	3.11.2 冻结 TensorFlow 模型	91
2.8.2 下载并安装	57	3.12 用训练好的冻结图模型做目标 检测	93
2.8.3 测试安装	57	3.13 用 Python 程序一键训练模型	95
2.9 本章小结	58	3.13.1 为新项目一键创建文件夹结构	95
第 3 章 训练模型	59	3.13.2 一键训练模型	97
3.1 TensorFlow Object Detection API 软件框架简介	59	3.14 本章小结	99
3.2 使用 TensorFlow 预训练模型	60	第 4 章 优化并部署模型	100
3.2.1 如何选择预训练模型	60	4.1 OpenVINO™ 工具套件简介	101
3.2.2 预训练模型的文件构成	61	4.2 OpenVINO™ 典型开发流程	102
3.2.3 一个典型的深度学习训练流程	62	4.3 安装 OpenVINO™ 工具套件	103
3.3 准备图片：下载猫狗数据集	62	4.3.1 版本选择	103
3.3.1 Kaggle 数据集下载流程	63	4.3.2 系统要求	104
3.3.2 训练图片的数量问题	66	4.3.3 下载并安装 OpenVINO™ 工具套件	105
3.3.3 训练图片的样本不平衡问题	67	4.4 安装 Cmake	106
3.4 使用 LabelImg 标注图片	67	4.5 安装 Microsoft Visual Studio 2017	107
3.4.1 LabelImg 简介	67	4.6 安装硬件驱动	110
3.4.2 建立猫狗项目文件夹结构	68	4.6.1 英特尔®显卡驱动	110
3.4.3 标注图片	69	4.6.2 英特尔®神经计算棒二代驱动	111
3.4.4 标注文件 (*.xml) 简介	72	4.6.3 英特尔®视觉计算加速卡驱动	113
3.4.5 复制 10% 的数据到 eval 文件夹	73	4.7 设置环境变量	116
3.4.6 复制数据到 test 文件夹	74	4.8 运行演示程序	118
3.5 依据标注类型创建标签映射文件	75	4.8.1 demo_benchmark_app.bat	119
3.6 创建 TensorFlow TFRecord 文件	75	4.8.2 demo_security_barrier_camera.bat	120
3.6.1 将 *.xml 文件转换为 *.csv 文件	76	4.8.3 demo_squeezenet_download_ convert_run.bat	120
3.6.2 将 *.csv 文件转换为 *.tfrecord 文件	77		
3.7 修改预训练模型的配置文件	79		
3.7.1 预训练模型的配置文件	80		
3.7.2 配置文件的结构	81		
3.7.3 修改 ssd_inception_v2_coco.config 配置文件	81		

4.9 编译并运行 Inference Engine	
范例和演示程序	120
4.9.1 编译 samples 文件夹中的范例	121
4.9.2 编译 demos 文件夹中的范例	124
4.9.3 从 Open Model Zoo 中下载预训练模型	126
4.9.4 下载英特尔®范例视频	129
4.9.5 运行预训练模型	129
4.10 使用 Model Optimizer 优化模型	131
4.10.1 转换 TensorFlow* Object Detection API 模型	131
4.10.2 用 OpenVINO™ 工具套件范例程序测试 IR 模型	134
4.10.3 用 OpenVINO™ 工具套件演示程序测试 IR 模型	136
4.11 编写 OpenVINO™ 应用程序	139
4.11.1 Inference Engine 简介	139
4.11.2 Inference Engine Plugin 构架	140
4.11.3 Inference Engine 应用程序典型开发流程	141
4.11.4 查看模型的输入和输出张量	141
4.12 OpenVINO™ AI 推理计算 C++ 范例	142
4.12.1 设置环境变量和 Visual Studio 项目属性	142
4.12.2 开发 AI 推理计算 C++ 应用程序	149
4.12.3 切换 AI 推理计算硬件	157
4.13 OpenVINO™ AI 推理计算 Python 范例	158
4.13.1 设置环境变量 PYTHONPATH	158
4.13.2 开发 AI 推理计算 Python 应用程序 (OpenCV 版)	160
4.13.3 开发 AI 推理计算 Python 应用程序 (OpenVINO™ 版)	163
4.13.4 AI 推理计算用 Python 还是 C++?	167
4.14 本章小结	167
第 5 章 进一步提升 AI 推理计算性能	168
5.1 性能评价指标	168
5.2 同步和异步模式	169
5.2.1 同步模式范例	170
5.2.2 异步模式范例	172
5.3 多设备和异构插件	175
5.3.1 异构插件	176
5.3.2 多设备插件	177
5.4 本章小结	181
第 6 章 工业领域光学字符识别范例	182
6.1 项目背景	182
6.2 新建 OCR 项目工程文件夹	182
6.3 收集并标注图片	184
6.4 训练模型	187
6.5 导出 TensorFlow 冻结图模型	189
6.6 测试模型	190
6.7 基于 OpenVINO™ 工具套件优化并加速模型	193
6.8 基于 OpenVINO™ 工具套件部署模型	194
6.9 本章小结	203
第 7 章 垃圾瓶自动分选项目范例	204
7.1 项目背景	204
7.2 新建垃圾瓶分类项目工程文件夹	205
7.3 收集并标注图片	206
7.4 训练模型	210
7.5 导出 TensorFlow 冻结图模型	211
7.6 测试模型	213
7.7 基于 OpenVINO™ 工具套件优化并加速模型	214
7.8 基于 OpenVINO™ 工具套件部署模型	216
7.9 本章小结	225
第 8 章 农作物病虫害自动识别项目范例	226
8.1 项目背景	226

参考文献 271

第 1 章 人工智能、深度学习与目标检测

1.1 人工智能简介

1.1.1 什么是人工智能

人工智能（Artificial Intelligence，AI）是研究用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。人工智能是计算机科学的一个分支，它企图了解智能的实质，并生产出一种新的与人类智能相似的方式做出反应的智能机器，该领域的研究包括机器人、语音识别、图像识别、自然语言处理和专家系统等。

当前，AI 已经进入了人们生活的方方面面，例如在语音识别方面有语音助手、翻译机及智能音箱等应用；在图像识别方面有智能驾驶、人脸识别和医学影像识别等应用，如图 1-1 所示。

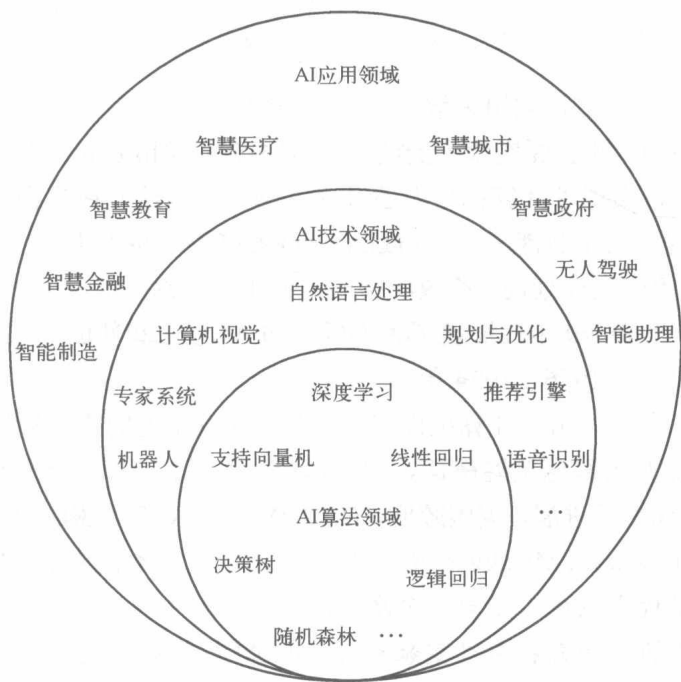


图 1-1 AI 技术应用范围

1.1.2 人工智能发展简史

1946 年，全球第一台通用计算机 ENIAC 诞生。它最初为美军作战研制，每秒能完成 5000 次加法，400 次乘法等运算。ENIAC 为人工智能的研究提供了物质基础。

1950 年，被誉为计算机科学之父和人工智能之父的英国数学家和逻辑学家艾伦·麦席森·图灵（Alan Mathison Turing）发表了一篇论文 *Computing Machinery and Intelligence*，提出若机器通过“图灵测试（The Turing Test）”，则被认为具有人类智能。

1956 年，约翰·麦卡锡（John McCarthy）、马文·闵斯基（Marvin Minsky）、克劳德·香农（Claude Shannon）等科学家们在美国达特茅斯学院（Dartmouth College）的讨论会上，讨论了一个主题：用机器来模仿人类学习以及其他方面的智能。这次讨论会足足开了两个月，最终没有达成共识，但是他们却为这次讨论的内容取了一个名字：人工智能。因此，1956 年被称为人工智能元年，如图 1-2 所示。

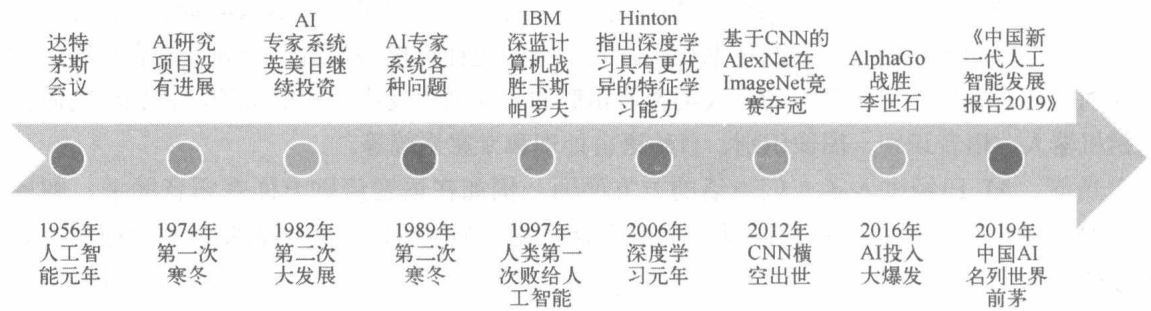


图 1-2 人工智能（AI）发展简史

达特茅斯会议后，计算机被用来解决代数应用题、证明几何定理、学习和使用英语。1965 年，美国麻省理工学院 AI 实验室的约瑟夫·魏岑鲍姆（Joseph Weizenbaum）教授开发了 Eliza 聊天机器人，实现了计算机与人通过文本来交流，这些成果在得到广泛赞赏的同时很多学者开始惊呼“人工智能来了，再过十年机器就要超越人类了！”

ARPA（美国高等研究计划局）等政府机构向人工智能这一新兴领域投入了大笔资金。从 20 世纪 50 年代后期到 60 年代涌现了大批成功的 AI 程序和新的研究方向。人工智能进入了第一个黄金发展期（1956~1974 年）。

到了 20 世纪 70 年代，由于计算机性能的不足，计算难度的指数级增长以及数据量缺失等问题，人工智能开始遭遇发展瓶颈。AI 开始遭遇批评，随之而来的还有研究资金上的困难，由于 AI 研究缺乏进展，英国政府，DARPA 和 NRC 等机构逐渐停止了对 AI 研究的资助。同时，由于 Marvin Minsky 对感知器的激烈批评，联结主义（即神经网络）销声匿迹了十年。人工智能发展进入了第一个寒冬。

20 世纪 80 年代初，一类名为“专家系统”的 AI 程序开始为全世界的企业所采纳，人工智能研究迎来了新一轮高潮。这个黄金发展期中，AI 专家系统用实践证明了可以让 AI 变得实用起来。AI 专家系统具体指一种程序，能够依据一组从专门知识中推演出的

逻辑规则在某一特定领域回答或解决问题。由于它只专注于某一个领域，因而设计简单，易于实现。

在研究资金上，日本于1982年4月制订了为期10年的“第五代计算机技术开发计划”，总投资1000亿日元，并称其将会成为人类计算机历史上的伟大变革。随后，英国、美国也纷纷响应，开始向AI和信息技术领域的研究提供大量资金。人工智能发展进入了第二个黄金发展期（1982~1989年）。

第二次人工智能黄金发展期仅仅持续了7年左右。由于AI专家系统维护费用居高不下，加上其操作复杂、太局限于经验知识和规则，难以构筑有效的系统，资本和政府支持再次撤出，日本雄心勃勃的“第五代计算机技术开发计划”也以失败告终，人工智能发展进入了第二个寒冬。

20世纪90年代中期，互联网和芯片计算能力的快速发展，加快了人工智能应用落地的速度。典型的事件：1997年5月，IBM公司的国际象棋计算机深蓝（DeepBlue）战胜了国际象棋世界冠军卡斯帕罗夫，这是人类第一次败给人工智能。

2006年，被称为“神经网络之父”“深度学习鼻祖”的Geoffrey Hinton在*Science*发文^[1]指出“多隐层神经网络具有更为优异的特征学习能力，并且其在训练上的复杂度可以通过逐层初始化来有效缓解”，让人们看到了人工智能通过深度学习（Deep Learning）技术超越人类的希望，2006年也被称为深度学习元年。

在GPU加速和大数据的加持下，深度学习发展如火如荼。2012年，Geoffrey Hinton及其学生Alex Krizhevsky等创造了一个“大型的深度卷积神经网络（CNN）”^[2]，即现在众所周知的AlexNet，赢得了当年的ImageNet图像分类大赛冠军，首次实现了Top 5误差率从25.8%降到了15.4%，这个表现震惊了整个计算机视觉界。从那时起，卷积神经网络（CNN）成为家喻户晓的名字，2012年后的ImageNet图像分类大赛全部是基于CNN的神经网络模型夺冠。

2016年，Google子公司DeepMind开发的基于深度神经网络的人工智能围棋程序AlphaGo以4:1战胜了拥有18个世界冠军头衔的棋王李世石，以深度学习技术为代表的人工智能再次火爆全球。随后，以中美两国为代表的多个国家纷纷出台类似《新一代人工智能发展规划》的国家级行业发展推动政策；AI再次成为资本界的宠儿，一个接一个的AI独角兽不断出现，估值飞涨。人工智能发展进入第三个黄金发展期（2006~现在）。

2017年7月8日，我国发布《新一代人工智能发展规划》，拉开了我国从国家层面支持和发展人工智能产业的序幕。

2019年5月24日，我国发布《中国新一代人工智能发展报告2019》（以下简称《报告》）。《报告》显示，我国人工智能发展已在部分方面具有优势。具体体现在：我国人工智能论文发文量全球领先，企业数量等多项指标居全球第二。2013~2018年，全球人工智能领域的论文文献产出共30.5万篇，其中我国发表论文7.4万篇；在全球居前1%的人工智能高被引论文中，我国居全球第二；在全球高被引前100篇论文中，我国有16篇入选。此外，据相关机构统计，截至2018年底，全球共成立人工智能企业15916家，我国人工智能企业数量为3341家，也位居世界第二位。

目前，AI 已在金融、医疗及安防等多个领域实现了技术落地，而且应用场景也越来越丰富，引发了各个行业的深刻变革。未来 AI 的发展将是技术与产业的结合，实现 AI 技术赋能各行各业，解决痛点、创造价值、降本增效。在这个趋势下，作为 AI 工程师，应该更多地从应用价值切入，推动 AI 项目落地，而不是仅仅停留在 AI 模型调优上。

1.1.3 人工智能与深度学习的关系

2012 年之前的人工智能算法基本都是由人去构造并提取特征(包括很多稀奇古怪的特征)的。2012 年，Geoffrey Hinton 及其学生 Alex Krizhevsky 使用深度卷积神经网络(CNN)让计算机自动去提取特征，在大算力和大数据的加持下，计算机自动提取特征比用人力去构造并提取特征更有优势，准确率更高、成本更低。

由于当前人工智能的大爆发是由深度学习引起的，所以各种媒体文章上，人工智能和深度学习两个概念经常被混用。从技术层面上来说，深度学习是机器学习诸多算法中的一种，而机器学习又是人工智能的一个子集，如图 1-3 所示。

非技术类媒体文章使用 AI 的频率远远大于深度学习，因为公众对深度学习这个名字没有 AI 这么熟悉。可以看到幼儿园、早教培训班、中小学的人工智能教材都用的是 AI 这个名词。产品广告和市场宣传资料中，也多用 AI，其目的也是让公众容易看懂。



图 1-3 人工智能与深度学习

从技术的角度来讲，总是用 AI 代替深度学习是不恰当的。本书在 1.2 节深度学习(Deep Learning)简介之前，基本用 AI 这个名词。在 1.2 节之后，涉及具体技术时，就会用深度学习。

总之，读者在阅读本书的时候，为了避免认知混淆，可以简单地认为 AI=深度学习。AI 多面向公众，深度学习多面向技术人员。

1.2 深度学习简介

深度学习(Deep Learning)就是用深度神经网络来自动学习对象特征，然后让深度神经网络具备识别对象的能力。那什么是深度神经网络？本书先从神经网络说起。

1.2.1 神经网络

提到神经网络，人们的第一印象就是类似于人脑那样充满智慧的东西，如图 1-4 所示。

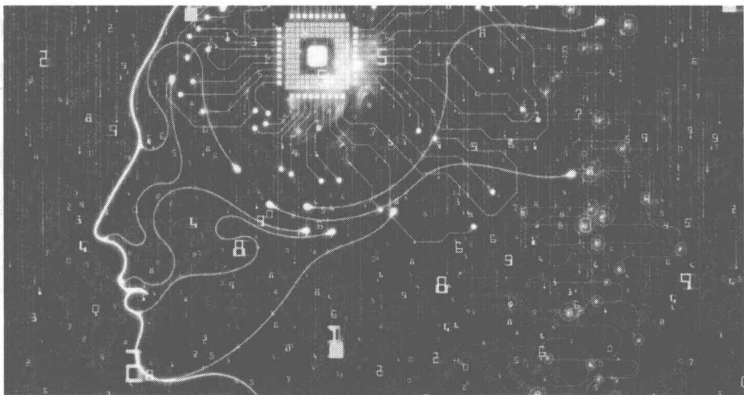


图 1-4 “神经网络”（引用自 <https://newsroom.intel.com>）

实际上，神经网络非常简单，就是多个神经元（Neuron）的堆叠，如图 1-5 所示。

这种网状的拓扑结构具备一定的智慧能力（分类能力），所以给它起了一个通俗易懂的名字——神经网络（Neural Network）。一听这个名字就知道包含着人们对其智慧能力的肯定和对其智慧潜力的期望。

1.2.2 神经元

如 1.2.1 节所述，神经网络由神经元（Neuron）堆叠而成，神经网络的基本组件就是神经元。以小学数学的视角看，神经元就是四则混合运算：神经元输入“ a_k ”与权重“ w_k ”相乘，再与偏置“ b ”相加，所得的和“ z ”经过激活函数“ $\sigma(z)$ ”处理，得到本级神经元的输出“ a ”，这也是下一级神经元的输入，如图 1-6 所示。

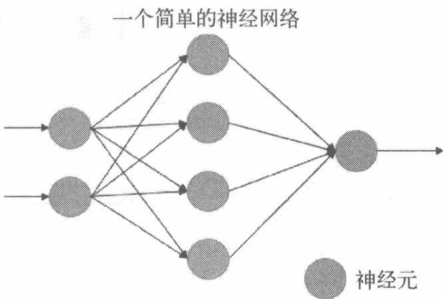


图 1-5 一个简单的神经网络

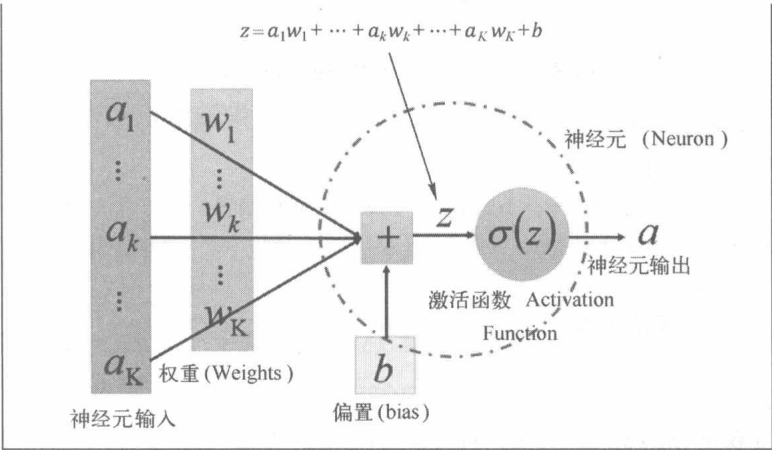


图 1-6 神经元（Neuron）