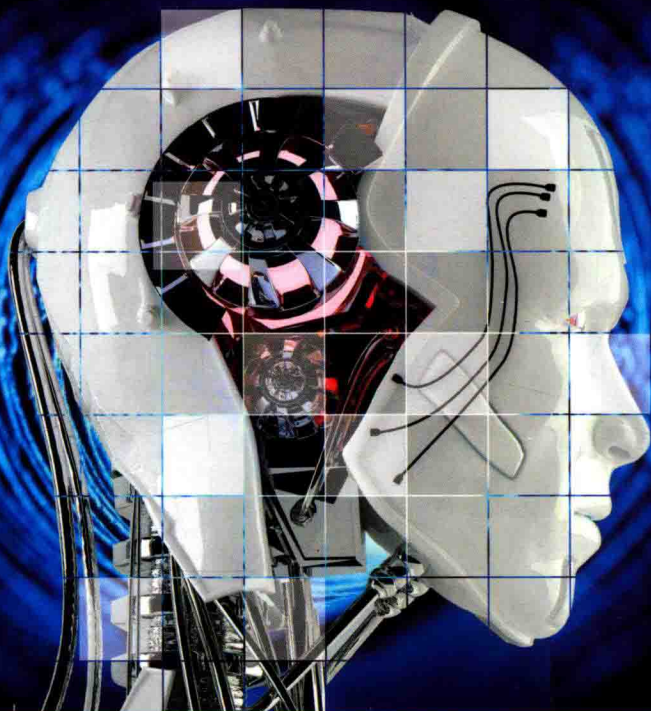


全彩印刷



缪 鹏 著

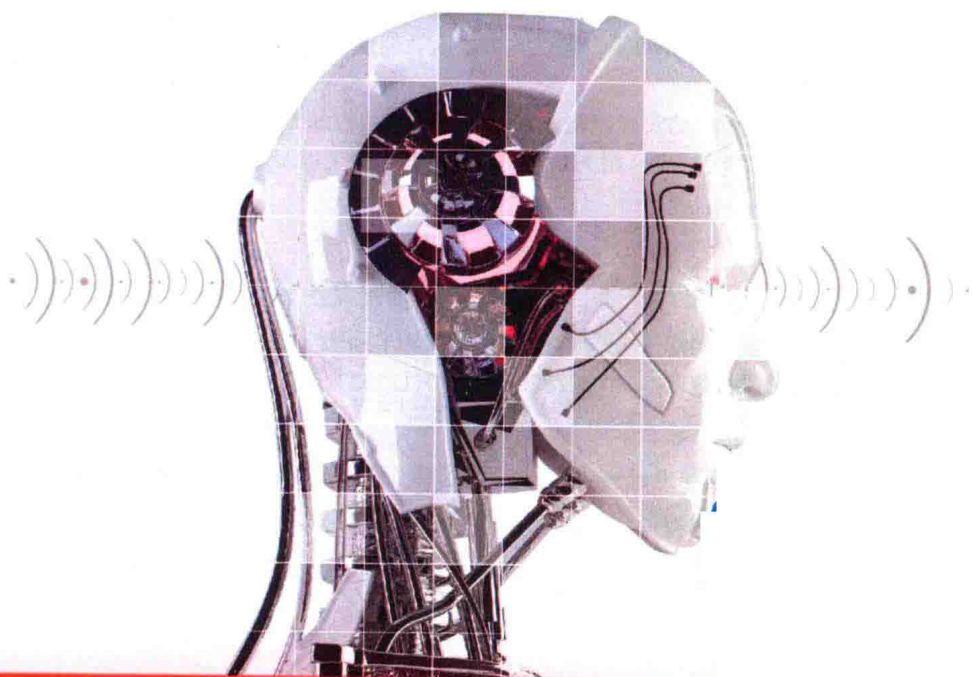
深度学习实践 计算机视觉

DEEP LEARNING PRACTICE
COMPUTER VISION

- 展现深度学习在计算机视觉方面的应用及工程实践
- 结合主流深度学习框架进行实例演示并给出实现代码



清华大学出版社



深度学习实践 计算机视觉

缪 鹏○著

清华大学出版社
北 京

内 容 简 介

本书主要介绍了深度学习在计算机视觉方面的应用及工程实践,以Python 3为开发语言,并结合当前主流的深度学习框架进行实例展示。主要内容包括:OpenCV入门、深度学习框架介绍、图像分类、目标检测与识别、图像分割、图像搜索以及图像生成等,涉及到的深度学习框架包括PyTorch、TensorFlow、Keras、Chainer、MXNet等。通过本书,读者能够了解深度学习在计算机视觉各个方向的应用以及最新进展。

本书的特点是依托工业环境的实践经验,具备较强的实用性和专业性。适合于广大计算机视觉工程领域的从业者、深度学习爱好者、相关专业的大学生和研究生以及对计算机视觉感兴趣的爱好者使用。

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

深度学习实践:计算机视觉/缪鹏著. —北京:清华大学出版社, 2019
ISBN 978-7-302-51790-0

I. ①深… II. ①缪… III. ①计算机视觉 IV. ①TP302.7

中国版本图书馆CIP数据核字(2018)第269496号

责任编辑:王金柱

封面设计:王翔

责任校对:闫秀华

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

社总机: 010-62770175

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

印装者: 北京天颖印刷有限公司

经 销: 全国新华书店

开 本: 180mm×230mm

印 张: 16.25

字 数: 364千字

版 次: 2019年2月第1版

印 次: 2019年2月第1次印刷

定 价: 79.00元

产品编号: 081322-01



前言

目前人工智能领域越来越受到公众的关注，因此人工智能算法工程师也渐渐浮出水面，成为招聘网站上一个非常耀眼的岗位，各类创业投资也紧紧围绕着 AI 主题旋转。

我认为目前人工智能算法工程师主要分为两类。

- 科学家型：主要研究前沿算法，在各大高校和企业的研究院居多。
- 工程师型：主要将最新的算法应用到具体的业务场景，在企业开发部门居多，为本书主要针对对象。

人工智能算法按特征学习的深浅分为机器学习、深度学习，另外也有强化学习方向。按应用场景则可分为：计算机视觉、自然语言和语音处理等。

编写本书主要基于以下事实，笔者在学习机器学习和深度学习的过程中，发现理论方面的书籍十分丰富，包括周志华老师的《机器学习》与 Ian Goodfellow 的《深度学习》；教学视频也十分丰富，包括斯坦福大学吴恩达教授的 CS229 与李飞飞教授的 CS231，以及台湾大学（National Taiwan University）林轩田老师和李宏毅老师的课程。但是很少有关于一个方向（比如计算机视觉）比较丰富的工程应用书籍，包括当前主流框架的综合介绍，笔者当时从理论到实践走了不少弯路，也踩过不少坑，故希望本书能在这个方面做出一点小小的贡献，成为理论与实践的桥梁，让读者相对容易地迈出由 0 到 1 的那一步。

本书主要关注计算机视觉领域，基于开源项目介绍最新的算法，在此也感谢各位开源人士，借助他们的成果，我们学习到了很多知识，本书各章主要内容如下：

第 1 章对深度学习与计算机视觉进行简要介绍，也会简单介绍开发环境的搭建。

第 2 章主要介绍 OpenCV 的基本操作及部分高级操作，包括人脸和人眼的检测与识别。

第3章着重介绍目前常用的几类深度学习框架，包括 PyTorch、Chainer、TensorFlow-Keras 和 MXNet-Gluon，另外本书中偶尔还会用到 ChainerCV 和 GluonCV。

第4章对图像分类进行了介绍，包括经典的网络类型（VGG、ResNet、Inception、Xception、DenseNet），并展示了部分实践操作。

第5章对目标检测与识别进行了介绍，包括三种主流的网络结构：YOLO、SSD、Faster R-CNN，并展示了实践操作。

第6章介绍图像分割技术，主要从前背景分割（Grab Cut）、语义分割（DeepLab 与 PSPNet）和实例分割（FCIS、Mask R-CNN、MaskLab、PANet）三个粒度阐述。

第7章介绍图像搜索技术，主要指以图搜图方面（CBIR），以及对应的实践展示。

第8章主要介绍图像生成技术，包括三个大方向：Auto-Encoder、GAN 和 Neural Style Transfer。

计算机视觉是一个非常大的方向，涉及的内容非常多，本书只涉及了其中部分领域，未涉及 OCR、目标追踪、三维重建和光场等方面的内容。

本书面向的主要是已经拥有机器学习和深度学习基础，但在计算机视觉领域实践较少，对各个方向了解较少的读者，其他感兴趣的读者也可作为科普读物。希望本书能为计算机视觉感兴趣的读者打开一扇窗户，引领大家迈出从理论到实践的关键一步。另外由于笔者学识、经验和能力水平所限，书中难免有错误或误解的地方，欢迎广大读者批评指正。

阅读本书需要的知识储备包括以下几种：

- 线性代数
- 概率论
- 统计学
- 高等数学，主要指函数方面
- 机器学习
- 深度学习
- Python 编程技术（特别需要熟悉 Numpy 库）
- Linux 基础知识（可选项）

如果在学习过程中遇到任何问题或不太理解的概念，那么最好的方式是通过网络寻找答案，请相信我们所遇到的问题，有很大一部分是大家都会遇到的问题，网上说不定已经有了详细地讨论，这时只需要去发现即可；如果没有找到对应的解决方法，那么在对应的社区提问也是很好的一种方式。

希望读者在阅读本书时，谨记计算机是负责资源调度的，永远会有时间资源和空间资源的平衡问题。GPU 的使用就是并行利用空间换取时间，而 IO 密集型与计算密集型则是另外两个常常遇到的问题。在做深度学习方面的实践时，这些问题都应该考虑到位，特别是面临海量数据的时候，比如上亿级别的图像搜索业务。这些知识在计算机操作系统的书籍当中有非常详细的论述，如果读者希望在计算机领域有长足的发展，那么这是一本最基本最重要的书籍，建议好好学习。

对于本书的完成，要特别感谢王金柱编辑给予的帮助和指导，感谢体贴的妻子体谅笔者分出部分时间来撰写此书。

读者联系邮箱：booksaga@126.com。

缪 鹏

2018 年 7 月 1 日

目 录

第 1 章 深度学习与计算机视觉	1
1.1 图像基础	3
1.2 深度学习与神经网络基础	4
1.2.1 函数的简单表达	5
1.2.2 函数的矩阵表达	5
1.2.3 神经网络的线性变换	6
1.2.4 神经网络的非线性变换	6
1.2.5 深层神经网络	6
1.2.6 神经网络的学习过程	8
1.3 卷积神经网络 CNN	9
1.4 基础开发环境搭建	14
1.5 本章总结	15
第 2 章 OpenCV 入门	16
2.1 读图、展示和保存新图	17
2.2 像素点及局部图像	18
2.3 基本线条操作	19
2.4 平移	20
2.5 旋转	20
2.6 缩放	21
2.6.1 邻近插值	22
2.6.2 双线性插值	22
2.7 翻转	23
2.8 裁剪	23
2.9 算术操作	23
2.10 位操作	24
2.11 Masking 操作	25
2.12 色彩通道分离与融合	26
2.13 颜色空间转换	27

2.14 颜色直方图	28
2.15 平滑与模糊	29
2.16 边缘检测	31
2.17 人脸和眼睛检测示例	32
2.18 本章总结	35
第3章 常见深度学习框架	36
3.1 PyTorch	38
3.1.1 Tensor	39
3.1.2 Autograd	42
3.1.3 Torch.nn	43
3.2 Chainer	45
3.2.1 Variable	46
3.2.2 Link 与 Function	47
3.2.3 Chain	50
3.2.4 optimizers	51
3.2.5 损失函数	51
3.2.6 GPU 的使用	52
3.2.7 模型的保存与加载	54
3.2.8 FashionMnist 图像分类示例	54
3.2.9 Trainer	59
3.3 TensorFlow 与 Keras	66
3.3.1 TensorFlow	66
3.3.2 Keras	67
3.4 MXNet 与 Gluon	73
3.4.1 MXNet	73
3.4.2 Gluon	74
3.4.3 Gluon Sequential	74
3.4.4 Gluon Block	75
3.4.5 使用 GPU	76
3.4.6 Gluon Hybrid	77
3.4.7 Lazy Evaluation	79
3.4.8 Module	80
3.5 其他框架	81
3.6 本章总结	81

第 4 章 图像分类	82
4.1 VGG	84
4.1.1 VGG 介绍	84
4.1.2 MXNet 版 VGG 使用示例	85
4.2 ResNet	89
4.2.1 ResNet 介绍	89
4.2.2 Chainer 版 ResNet 示例	90
4.3 Inception	95
4.3.1 Inception 介绍	95
4.3.2 Keras 版 Inception V3 川菜分类	97
4.4 Xception	116
4.4.1 Xception 简述	116
4.4.2 Keras 版本 Xception 使用示例	116
4.5 DenseNet	122
4.5.1 DenseNet 介绍	122
4.5.2 PyTorch 版 DenseNet 使用示例	122
4.6 本章总结	126
第 5 章 目标检测与识别	128
5.1 Faster RCNN	129
5.1.1 Faster RCNN 介绍	129
5.1.2 ChainerCV 版 Faster RCNN 示例	131
5.2 SSD	139
5.2.1 SSD 介绍	139
5.2.2 SSD 示例	140
5.3 YOLO	148
5.3.1 YOLO V1、V2 和 V3 介绍	148
5.3.2 Keras 版本 YOLO V3 示例	150
5.4 本章总结	157
第 6 章 图像分割	158
6.1 物体分割	159
6.2 语义分割	164
6.2.1 FCN 与 SegNet	166
6.2.2 PSPNet	171

6.2.3 DeepLab	172
6.3 实例分割	176
6.3.1 FCIS	177
6.3.2 Mask R-CNN	178
6.3.3 MaskLab	180
6.3.4 PANet	181
6.4 本章总结	181
第 7 章 图像搜索	183
7.1 Siamese Network	185
7.2 Triplet Network	186
7.3 Margin Based Network	188
7.4 Keras 版 Triplet Network 示例	190
7.4.1 准备数据	190
7.4.2 训练文件	191
7.4.3 采样文件	195
7.4.4 模型训练	202
7.4.5 模型测试	206
7.4.5 结果可视化	210
7.5 本章小结	216
第 8 章 图像生成	218
8.1 VAE	219
8.1.1 VAE 介绍	219
8.1.2 Chainer 版本 VAE 示例	220
8.2 生成对抗网络 GAN	221
8.2.1 GAN 介绍	221
8.2.2 Chainer DCGAN RPG 游戏角色生成示例	229
8.3 Neural Style Transfer	238
8.3.1 Neural Style Transfer 介绍	238
8.3.2 MXNet 多风格转换 MSG-Net 示例	241
8.4 本章总结	246
后记	247

第 1 章

深度学习与计算机视觉

深度学习与计算机视觉近几年非常火，而它们又和人工智能联系紧密，但它们到底是什么，能解决什么问题呢？本章便试着通俗简要地回答这个问题。

首先是对世界的认识，对于人类来说，可以靠各种感官来感受周围的世界，包括眼、口、鼻、耳、舌、身，这样我们就认识了这个世界是由颜色、形状、美丑、味道、温度甚至感情的憎恶等构成的。那么有没有方法让计算机也有这些感受和认知，再进行推理、判断和决策呢？笔者认为这就是人工智能所要解决的终极问题。

对于计算机来说，一切皆为数字。比如性别为男性可以用 1 表示，女性则用 0 表示，这些都是公认的，即一种个体的属性可以使用数字来表示。既然如此，那么用向量来表示也不会有问题，如 $[1,0,0]$ 代表“男”， $[0,0,1]$ 代表“女”。一般地，一个个体会包含很多的属性，那么把这些属性全部组合起来是不是就可以代表这个个体呢？当然可以，这对计算机来说就是有智慧的第一步——能认识并识别出不同的个体。

用眼睛观察世界对人类来说轻而易举，但对只认识数字的计算机来说就是一项非常难的任务。那么计算机视觉主要想解决什么问题呢？简单说就是让计算机能像人一样看事物，并能理解看到的事物，粒度从非常小的苍蝇到非常大的宇宙，从静态的物体到动态的行为过程，等等。此时便会涉及到一个根本性的问题：怎么样在计算机中表示这么多不同的物体呢？

以前人们经常使用的就是规则，即人类自己定义如何表示某个（或某类）物体，如从颜色、形状、纹理等等方面描述，但要知道，这个世界是非常大的，物体种类可以说是不计其数，万一规则冲突了怎么办？所以说基于规则的方法局限性非常大。于是就产生了这样的想法：计算机的计算能力这么厉害，有没有可能让它自己学习这些规则呢，比如给计算机看一些正确的例子？这样机器学习就产生了，深度学习是机器学习的一个子领域，而机器学习属于人工智能的研究范围。

机器学习主要是让计算机从历史经验（即数据）中学习知识，可将其理解为发现历史规律，总结经验教训，所以也可称为模式识别。机器学习常常可分为三种类型：监督学习、非监督学习和半监督学习。如果将机器学习简单理解为学生读书学习的过程，那么监督学习可理解为学生跟着老师学习，老师学识丰富；而非监督学习则是学生完全自学，自力更生；半监督学习则是两者综合，老师学识有限或学识丰富但指导时间有限，学生自己也需要自学。

最近几年机器学习领域发展起来的原因主要有以下几点。

- (1) 互联网快速发展，积累了大量的原始数据，包括图像、文本、影音等。
- (2) 计算机硬件飞速发展，计算能力大大提高。
- (3) 学术研究的突破，如以 Hinton 为代表的团队。

深度学习在很大程度上可理解为表示学习，即如何在计算机中用数字表示一个或一类物体。这种数字组成的东西也常常被称为特征，顾名思义：独特的表征，即在计算机中只有某种物体才会用那样一组数字来表示，因此深度学习也称作特征学习。如图 1-1 所示的鸟在计算机中可用独特的数字或数字组合来表示，比如：单个数字 99、向量 [123, 999, 888] 或者二维向量，甚至是更高维的向量。

那么这些数字表示什么意义呢？人类制定的规则，这些数字表示的意义一般比较明显，比如表示颜色、形状、有没有羽毛等。而在深度学习中，物体的特征向量常常很难与人类的直观意义匹配，即人们不懂这些数字代表什么意义，但计算机懂——计算机能在大量的特征向量中区分出个体。

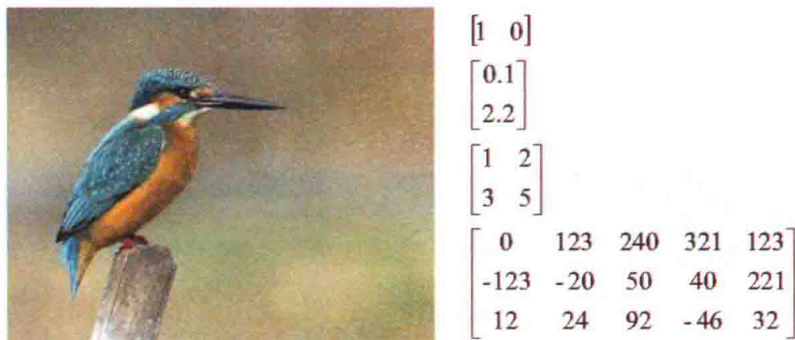


图 1-1 视觉图片与数字特征表示

本章主要介绍机器学习、深度学习与计算机视觉相关概念之间的关系，并介绍开发环境的安装。

1.1 图像基础

图像是人通过眼睛对外界的一种视觉感受，它可以存在于人们的脑海里，也可以通过某种介质（如照片或数码照片）保存下来，本书主要讨论的是计算机对图像的处理，所以明白计算机怎么看待图像是非常重要的。如前所述，计算机中所有文件都用数字表示，那么图像也不例外。

在计算机中，图像的最基本组成单元为像素，图片是包含很多个像素的集合。像素一般就是图片中某个位置的颜色，很多个像素点排列起来，就可以组成一个二维平面点阵，这就是图像。比如电脑桌面背景，如果是 $1920\text{px} \times 1080\text{px}$ 的大小，那就意味着有 1920×1080 （2073600）个像素：1920 列，1080 行。通常图像表达会用色彩空间的概念，常见的有 RGB、LAB、HSL 和灰度等，本书主要关注 RGB 和灰度这两种，其他色彩空间可查阅相关资料¹。RGB 图像又称为三通道彩色图，灰度图相对应就可以叫作单通道图。通道数可简单理解为表示单个像素所需要的数字的个数。

图像分两类：模拟图像和数字图像。两者之间最大的区别是像素的值域，模拟图像像素的值域是连续的，是人类所认识感受到的；而数字图像的值域则是离散的、有限的，是计算机等电子设备所认知的事物。本书所讨论的就是计算机所认知的图像，即数字图像，后面不再说明，这也是计算机视觉的主要任务。

1 <http://poynton.ca/ColorFAQ.html>（注意区分大小写）

在计算机中，灰度图中的像素通常用 0~255 之间的一个整数数字表示，0 表示黑色，255 表示白色，数字从 0 变到 255 表示颜色由黑变白的一个过程。颜色越黑则越接近 0，越白则越接近 255。



RGB 彩色空间则使用三个整数数字来代表一个像素，如 (0,100,200)，分别代表红色部分的颜色值为 0，绿色部分为 100，蓝色部分为 200。RGB 分别代表英文单词 Red、Green 和 Blue，其对应的取值范围都是 0 ~ 255，数值越大表示颜色越浅，越小则越饱和。所以 RGB 像素不同的组合总数为： $256 \times 256 \times 256 = 16777216$ ，其中 (0,0,0) 表示黑色，(255,255,255) 表示白色。

基于以上认识，像素点阵就可以使用矩阵来表示，差异就是不同空间表示像素的方法不同。灰度图可简单理解为一个二维矩阵，里面填满了 0 ~ 255 间的整数；而彩色图则是三维矩阵，维度分别代表高、宽和通道数，如图 1-2 所示可以更形象直观地理解，一个 4×4 的灰度图像矩阵和一个 4×4 的 RGB 彩色图像（除非特殊说明，后期本书中的彩色图像一般指 RGB 空间格式）矩阵。

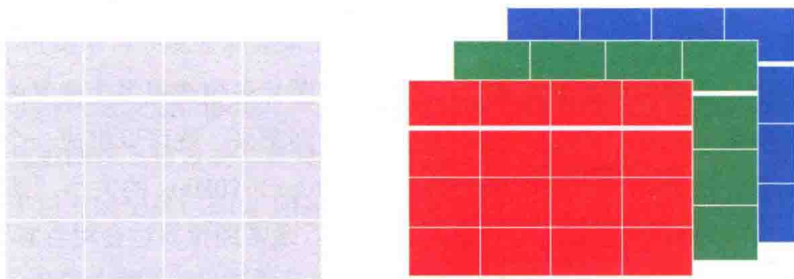


图 1-2 灰度图与 RGB 彩图

1.2 深度学习与神经网络基础

深度学习这个词语很时髦，这里通俗地解释一下它的概念。

深度学习就是使用神经网络来进行学习，将一种表示（Representation）转换为另外一种表示，那么神经网络是什么呢？简单来说神经网络就是一个函数，但它可以非常简

单，如 $y=x+3$ ；也可以非常复杂，复杂到难以用数学公式进行解析表达，一般来讲，神经网络会包含两个主要部分：线性变换函数和非线性变换函数。

1.2.1 函数的简单表达

简单说函数就是将输入通过一些操作或变换变为输出，简记为 $y=f(x)$ ，就可以说 x 经过函数 f 的作用变成了 y 。很多人就学过以下函数，这些函数通常做的就是将一个数字（标量，scalar）映射（变成）为另外一个数字（标量），当然可能存在一一对应，一多对应和多一对应这些情况，此处只讨论一一对应，即一个 x 只能映射为一个 y ，如下所示：

$$\begin{aligned}y &= ax + b \\y &= ax^2 + bx + c \\y &= ae^{bx} + cx + d\end{aligned}$$

1.2.2 函数的矩阵表达

如果输入的是多个数字组成的向量（vector）呢，比如一个点在二维平面空间的坐标 (x, y) ，然后输出是一个标量，比如高度 z 。假设可以用一个简单的线性函数来表示，即： $z = a \times x + b \times y + c$ ，这样便表示了整个操作过程。但输入通常会被当作一个变量，此时应该怎么表示这个式子呢？此时便引出了以下矩阵和向量的操作：将 $[a, b]$ 视为矩阵 A ，将 $[x, y]$ 视为向量 X ，然后进行矩阵与向量的乘法操作，其实就是行的元素与列的元素对应相乘然后相加。如果输入的维度更高，那么只需要增加输入 X 向量中的元素个数即可，同时对应增加线性变换矩阵 A 中每行的元素个数。如果输出多个值怎么办呢？其实只需要将线性变换矩阵 A 的行数增加即可，有多少个值 A 中就有多少行，此时输出也可以使用矩阵 Z 表示，如下所示：

$$\begin{aligned}z &= \begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + c = AX + c \\Z &= \begin{bmatrix} z1 \\ z2 \end{bmatrix} = \begin{bmatrix} a & b \\ e & d \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} + c = \begin{bmatrix} ax + by + c \\ ex + dy + c \end{bmatrix}\end{aligned}$$

1.2.3 神经网络的线性变换

函数的矩阵操作也可称为线性变换，它是神经网络中最基础的操作之一。神经网络中的线性变换只是将变换矩阵 A 和输入 X 的维度变得更大而已。对于图像来说， X 已经是成百上千级别的矩阵变量了，但原理还是一样：对应相乘然后做连加操作。

比如对于一个 2×2 灰度图片，可以想象将所有的元素拉伸为一维数组（当然这样做会失去图像的空间特性），然后进行线性变换，可用以下式子表示，此处省略常数项：

$$\begin{aligned} Z &= \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \end{bmatrix} \begin{bmatrix} x_1 & x_2 & y_1 & y_2 \end{bmatrix}^T \\ &= \begin{bmatrix} w_{11}x_1 + w_{12}x_2 + w_{13}y_1 + w_{14}y_2 \\ w_{21}x_1 + w_{22}x_2 + w_{23}y_1 + w_{24}y_2 \end{bmatrix} \end{aligned}$$

这样操作之后就得到一个输出，输出包括两个数字，即可理解为 2 维输出。现实情况中输出会有更多的维度。另外值得一提的是，此处每一行的参数个数与图片的高和宽的乘积一样，其本质就是卷积操作。

1.2.4 神经网络的非线性变换

神经网络使用线性变换可以做非常多的线性操作，但这个世界还有非常多的非线性映射，比如二次函数 x^2 ，此时就需要通过非线性变换来解决此类问题。

过去非常多的学者为之努力过，并提出了使用激活函数来进行非线性变换。目前常用的激活函数及对应的导数如图 1-3 所示，可以看到其是否有梯度消失或爆炸、饱和等性质，如想直观了解更多的激活函数可参见相关网站²。

1.2.5 深层神经网络

前文讲解了神经网络的两个最重要的基本组成单元，即线性变换和非线性变换，使用它们的组合既可以模拟线性变换又可以模拟非线性变换。但世界上有无数函数（线性 + 非线性），那么怎么去模拟更多的函数呢？答案就是 Deep。

所谓 Deep，其实质就是不断地叠加这种线性和非线性操作，每次操作如果被称为一个网络层，那么叠加很多次这些操作，就形成了所谓的深层网络结构，如图 1-4 所示。

² <https://dashee87.github.io/data%20science/deep%20learning/visualising-activation-functions-in-neural-networks/>

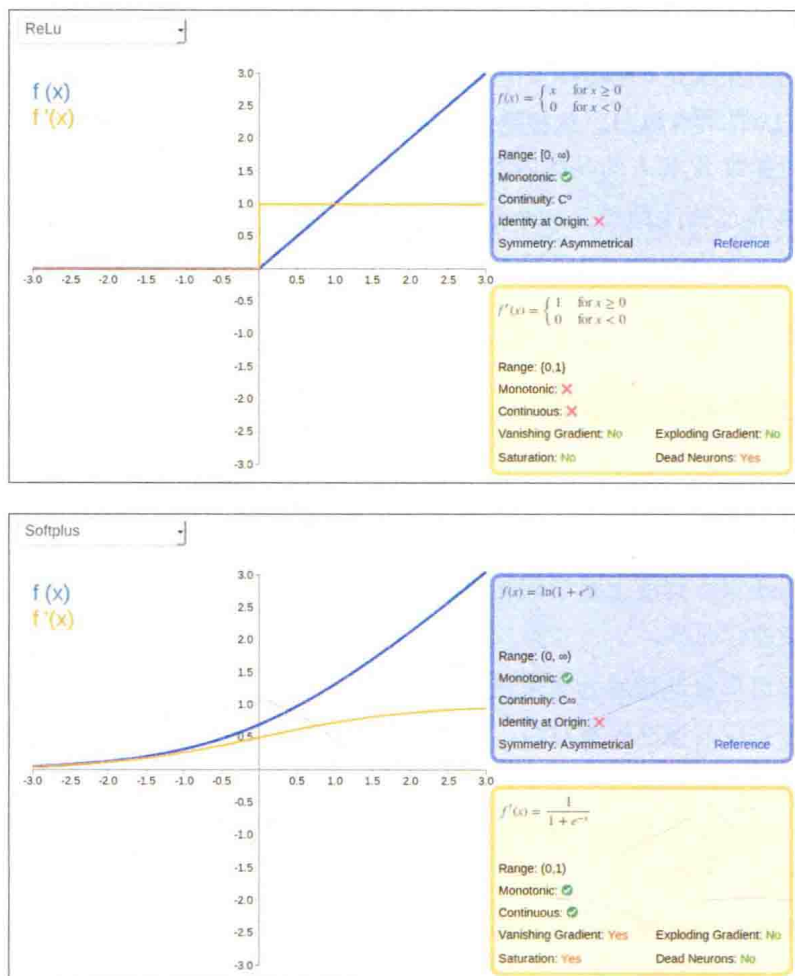


图 1-3 常见激活函数

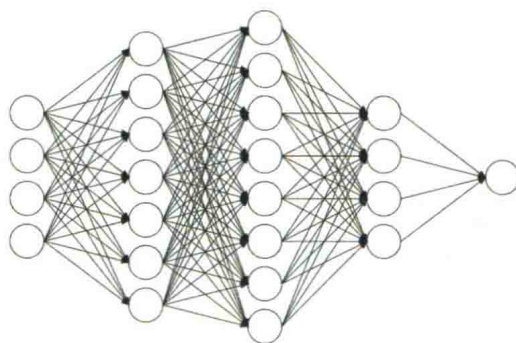


图 1-4 神经网络示意图