

前 言

随着新一轮科技革命与产业变革的深入推进，人工智能技术已渗透至社会发展的各个领域，深度学习作为其核心支撑技术之一，对复合型创新人才的培养提出了更高要求。深度学习领域现有相关教材大多侧重理论学习，本书编写团队通过广泛调研产业一线需求发现，现阶段迫切需要培养相关专业学生掌握深度学习领域新技术和新方法的工程实践能力，使其在工作任务中能够扩展新思维、提升实战能力。

目前，深度学习的研究和应用方向十分广泛，尤其在视频、音频、语言等数据的处理方面更具有优势。为响应国家“新工科”建设战略，服务新一代信息技术产业需求，编者组织长期深耕相关领域的教学科研团队，结合多年一线教学经验与工程实践成果编写了本书。本书首先介绍深度学习基础知识，然后介绍深度神经网络的训练过程，并在此基础上介绍几种典型深度神经网络，最后讨论深度神经网络模型的压缩处理，相关章节理论介绍与实践训练相结合，期望能够帮助学习者实现从知识理解到知识运用的贯通。

本书注重知识体系的完整性与工程应用的针对性，以实际应用需求为导向，将理论介绍与编程实现紧密结合；聚焦产业技术痛点，系统解析典型深度神经网络原理、实现、优化策略、轻量化部署方法等，帮助读者掌握从实验室原型到工程落地的关键技能。本书注重可操作性与拓展性，配有实验案例，并提供实验案例的参考代码和数据集(读者可扫描相应二维码进行下载)，构建了“学—练—创”一体化的学习支持体系。编者期望通过本书为高校人工智能教学提供有力支撑，促进学生在掌握深度学习核心技术的同时，提升解决复杂工程问题的能力，成长为适应未来技术变革的创新型人才。

本书得到国防科技大学“十四五”教材建设规划项目的资助，国防科技大学电子科学学院在读研究生白楠、田纪龙、王文涛、刘泳呈等在出版过程中提供帮助，在此一并致谢。

由于编者水平有限，书中难免有不完善之处，恳请读者予以批评指正。

编 者

2025 年 11 月

目 录

第 1 章 深度学习概述	1
1.1 基础知识	1
1.1.1 神经网络	1
1.1.2 深度神经网络	6
1.2 深度学习的数学原理	7
1.2.1 激活函数	7
1.2.2 损失函数	11
1.2.3 梯度	12
1.3 深度神经网络的应用	13
1.3.1 计算机视觉	13
1.3.2 自然语言处理	16
1.3.3 语音处理	17
第 2 章 深度神经网络开发基础	18
2.1 深度神经网络框架	18
2.1.1 TensorFlow	18
2.1.2 PyTorch	19
2.1.3 PaddlePaddle	19
2.2 数据增强	20
2.2.1 图像数据增强	21
2.2.2 文本数据增强	21
2.2.3 语音数据增强	22
2.3 深度神经网络可视化	23
2.3.1 TensorFlow 可视化组件	23
2.3.2 PyTorch 可视化组件	25
2.3.3 PaddlePaddle 可视化组件	25
2.4 深度神经网络训练优化	26
2.4.1 指数加权移动平均数	26
2.4.2 梯度下降法	27
2.4.3 Adam 优化法	31
2.5 实验 1：深度学习环境安装与配置	32
2.5.1 安装 GPU 计算加速环境	32
2.5.2 安装 TensorFlow	40
2.5.3 安装 PyTorch	53

2.5.4	安装 PaddlePaddle	55
第 3 章	卷积神经网络	57
3.1	基础知识	57
3.1.1	基本概念	57
3.1.2	训练过程	58
3.2	CNN 结构详解	59
3.2.1	卷积层	60
3.2.2	池化层	61
3.2.3	全连接层	62
3.3	典型 CNN	62
3.4	实验 2: 基于 CNN 实现目标检测	66
3.4.1	实验准备	66
3.4.2	使用预训练模型进行推理	70
3.4.3	在预训练模型上继续训练	76
3.4.4	拓展: 使用 GPU 训练及训练可视化	79
第 4 章	循环神经网络	91
4.1	基础知识	91
4.1.1	基本概念	91
4.1.2	训练过程	92
4.2	RNN 结构详解	93
4.2.1	RNN 的基本结构	93
4.2.2	RNN 的结构类型	94
4.2.3	RNN 的具体结构	96
4.3	典型 RNN	98
4.3.1	LSTM	98
4.3.2	GRU	101
4.3.3	BRNN	103
4.3.4	DRNN	104
4.4	实验 3: 基于 RNN 的敏感词分类	105
4.4.1	实验准备	105
4.4.2	网络设计	110
4.4.3	网络训练	111
4.4.4	网络测试	112
第 5 章	生成对抗网络	114
5.1	基础知识	114
5.1.1	基本概念	114
5.1.2	训练过程	115
5.2	GAN 结构详解	115

5.2.1	生成器	116
5.2.2	判别器	117
5.3	典型 GAN	117
5.3.1	DCGAN	117
5.3.2	StyleGAN	118
5.3.3	CycleGAN	120
5.4	实验 4: 基于 GAN 的图像季节风格转化	120
5.4.1	实验准备	121
5.4.2	网络训练	123
5.4.3	网络测试	126
第 6 章	Transformer	130
6.1	基础知识	130
6.1.1	基本概念	130
6.1.2	训练过程	132
6.2	Transformer 的结构详解	134
6.2.1	编码器	136
6.2.2	解码器	137
6.2.3	注意力机制	137
6.3	典型的 Transformer 网络	138
6.3.1	GPT	138
6.3.2	BERT	142
6.3.3	Swin-Transformer	146
6.4	实验 5: 基于 Transformer 的船舶轨迹预测	151
6.4.1	数据集及预处理	151
6.4.2	网络训练	152
6.4.3	网络测试	154
第 7 章	网络模型压缩	157
7.1	基础知识	157
7.1.1	网络模型压缩原理	157
7.1.2	网络模型压缩类型	157
7.2	剪枝与量化	158
7.2.1	模型剪枝	158
7.2.2	参数量化	159
7.3	低秩因子分解	162
7.3.1	奇异值分解	162
7.3.2	CP 分解	163
7.3.3	Tucker 分解	166
7.3.4	三种分解的区别与联系	167

7.4 知识蒸馏	168
7.5 实验 6: 利用模型压缩技术改进目标检测网络	170
7.5.1 YOLOv5 项目安装	170
7.5.2 准备数据集	172
7.5.3 修改配置文件	172
7.5.4 训练数据集	173
7.5.5 模型测试	174
7.5.6 稀疏化训练	175
7.5.7 网络模型剪枝及微调	176
参考文献	178

第 1 章 深度学习概述

深度学习(deep learning, DL)是一种基于人工神经网络(artificial neural network, ANN)的机器学习技术,其核心思想是通过模拟人脑中神经元的工作方式,使用多层的非线性信息处理单元来提取数据的特征和模式。深度学习模型通常包含大量的层级,即“深度”结构,每一层都可以对输入数据进行不同级别的抽象表示。深度学习研究在近年来取得了巨大的突破,广泛应用于计算机视觉(computer vision, CV)、自然语言处理(natural language processing, NLP)、语音识别等领域。

一般而言,深度学习的关键特点包括层次化的特征提取、端到端的学习方式及基于大数据的训练。通过多层次的神经网络结构,深度学习模型能够逐步提取数据中的抽象特征,从而实现对复杂模式的学习和表示。同时,深度学习模型可以直接从原始数据中学习端到端的映射关系,无须手工设计特征,大大简化了模型构建的过程。此外,深度学习模型通常需要大规模的数据集进行训练,以获得更好的泛化能力和预测准确度。

深度学习的应用领域非常广泛。例如,在计算机视觉领域,深度学习模型可以完成图像分类、目标检测、图像生成等任务;在语音处理领域,深度学习模型可以完成语音识别、语音合成等任务;在自然语言处理领域,深度学习模型可以完成文本分类、机器翻译、情感分析等任务。

1.1 基础知识

深度学习是一种利用深度神经网络(deep neural network, DNN)进行特征学习和模式识别的方法。学习的核心在于通过深度神经网络自动地从大量数据中学习复杂的特征,而不是依赖人工设计的特征。本节主要从神经网络和深度神经网络的角度来阐述深度学习的基础知识。

1.1.1 神经网络

神经网络的全称为人工神经网络,由研究者受到生物(主要是人类)神经网络的结构、运行模式、功能的启发而设计形成,目的是完成分类、识别、预测等任务。近年来,掀起新一轮人工智能热潮的深度学习也属于人工神经网络的范畴。

1. 生物学基础

人工神经网络是在研究和理解人脑结构及其对外界刺激的响应机制基础上,以生物神经系统的基本原理和结构为范本原型,以网络拓扑为理论基础,对复杂信息进行非线性关系表示和逻辑操作的一种数学模型。该模型可看成一个由大量神经元结构相互连接

而成的复杂网络，它将信息的加工和存储结合在一起，具有独特的知识表示方式和智能化的自适应学习能力。

1889 年, Cajal 创立了生物神经元学说, 他指出神经系统由相对独立的神经细胞构成。神经细胞也称为神经元, 具有信息处理功能, 是人体神经系统的基本单元, 人脑有 $10^{11} \sim 10^{12}$ 个神经元。每个神经元与其他神经元相连接, 按不同形式的结合方式构成极其庞大而又复杂的网络, 即生物神经网络。人脑神经网络中各神经元之间连接的强弱, 可以根据外部的激励信号做适应性变化; 每个神经元将所接收的多个激励信号综合后呈现出兴奋或抑制两种状态。通过神经元及其连接的可塑性, 人类大脑具有学习、记忆、认识和决策等功能。

1) 生物神经元结构

一个典型的生物神经元如图 1.1 所示。

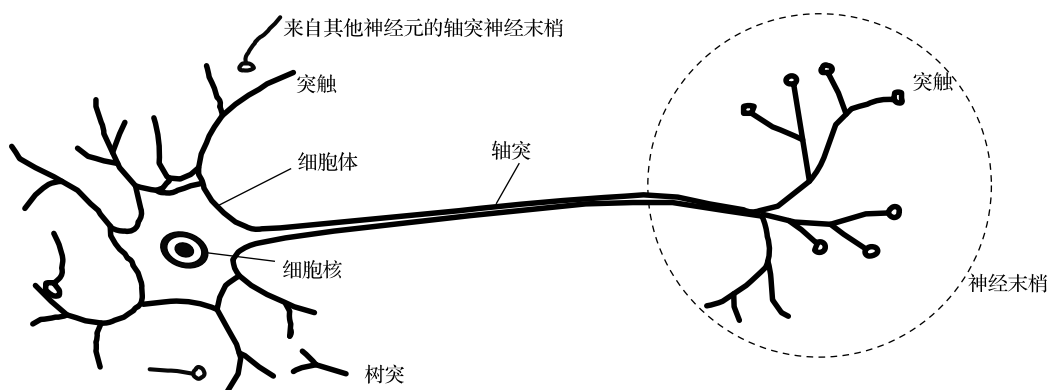


图 1.1 生物神经元

典型的生物神经元由细胞体、树突、轴突和突触组成。

细胞体：由细胞核、细胞质和细胞膜三部分构成。细胞体是神经元的主体，它是接收与处理信息的部件。

树突：在细胞体周围向外伸出的许多突起的神经纤维。它是细胞体的信息输入端。

轴突：由细胞体向外延伸最长的突起纤维体，它是细胞体的信息输出端。轴突的末端部分是神经元的末梢，简称神经末梢。

突触：两个神经元之间进行信息传递的特化连接结构，相当于神经元之间信息传递的输入与输出接口。最常见的类型是一个神经元的神经末梢与另一个神经元的树突或细胞体形成的连接。

2) 生物神经元的信息传输过程

生物神经元中的细胞体相当于一个处理器，树突为输入端，轴突为输出端，突触为输入与输出接口。生物神经元的工作方式可以概括为“刺激叠加，瞬间冲动”。神经元有两种状态：兴奋和抑制。如果神经细胞受到外界一定强度叠加信号的刺激，则神经元处于兴奋状态；反之，神经元处于抑制状态。通过轴突，当前神经元会将自身状态(兴奋或抑制)传导至其相邻的神经元，这样，信息通过一个个神经元在网络中传导下去，从而帮

助生物进行记忆、学习等活动。

3) 人脑神经信息处理的特点

(1) 具有并行分布式处理能力。人脑中单个神经元的信息处理速度实际上并不快，每次大约 1ms，比通常的电子门电路速度(0.001ms)慢几个数量级。但是，人脑对复杂过程的处理和反应都很快，一般只需要几百毫秒，而且不与复杂程度呈线性关系。

例如，人脑要判定眼睛看到的两个图形是否一样，实际上约需 400ms，按神经元处理的速度，如果采用串行工作模式，不可能在数百毫秒内完成。实际上，人脑可以看成由众多神经元组成的超高密度并行处理系统，具有更快的处理速度。

(2) 具有可塑性和自组织性。大脑皮层神经元之间的大部分突触连接是后天由环境激励逐渐形成的，并能够动态改变。这种随环境刺激不同而形成或改变神经元之间突触连接的现象称为可塑性。在环境的刺激下，形成并调整神经元之间的突触连接，从而逐渐构成神经网络的现象称为自组织性。

(3) 具有信息存算一体能力。常规的计算机系统通常将数据存储与运算分开，处理过程中数据的读取将耗费较多时间和能源；而人脑皮层中记忆和处理是有机结合的，信息处理与信息存储合在一起，即存算一体。其优点是便于大量相关知识参与信息处理过程，这对提高网络的信息处理速度和智能性至关重要。

(4) 具有冗余性、容错性和鲁棒性。信息在大脑的神经网络中分散存储在很多神经元里，而且每个神经元实际上存储了多种不同信息的部分内容。在分布存储的内容中，有许多信息用于支撑完成同一功能，这就使神经网络具有冗余性，当大脑的某个神经元损坏或死亡时，不至于丢失记忆的信息。容错性指的是大脑将信息分散存储在许多神经元及它们的突触连接之中，如果部分神经元有损伤，通过自组织功能可使神经系统的总体功能继续有效。由于网络具有高连接度，一定范围内的误差及干扰不会使网络的性能恶化，即网络具有鲁棒性。

2. 神经元数学模型

在生物学研究的基础上，科学家模拟大脑生物过程的基本特性，提出人工神经网络模型。

人工神经网络模型最早由美国心理学家麦卡洛克(McMulloch)和数理逻辑学家皮茨(Pitts)于 1943 年提出，是一种根据生物神经元的功能和结构抽象而来的神经元描述模型，称为 M-P 模型，其结构如图 1.2 所示。

图 1.2 中的神经元 j ，其输入为 $x_1, x_2, \dots, x_n$ ，可用向量 $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ 表示； $w_{j1}, w_{j2}, \dots, w_{jn}$ 为输入的权值，用向量 $\mathbf{w} = [w_{j1}, w_{j2}, \dots, w_{jn}]^T$ 表示； θ_j 为神经元 j 的阈值，其作用相当于设定一个门限，当全部输入信号加权求和超过阈值时，该神经元输出为 1(兴奋)，否则输出为 0(抑制)。

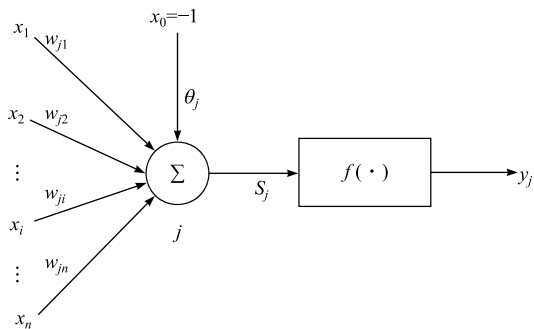


图 1.2 M-P 模型

在图 1.2 的神经元模型中, 神经元又称为节点, 节点 j 的总输入向量 S_j 定义为

$$S_j = \sum_{i=1}^n w_{ji} x_i - \theta_j \quad (1.1)$$

如果令 $x_0 = -1$ 、 $w_{j0} = \theta_j$, 即令 $\mathbf{x}$ 及 $\mathbf{w}_j$ 包括 x_0 和 w_{j0} , 则扩展后的总输入向量及连接权向量为

$$\mathbf{S}_j = \sum_{i=0}^n w_{ji} x_i = \mathbf{w}_j \mathbf{x} \quad (1.2)$$

生物神经元整合突触输入信号, 当总输入超过某个阈值时, 神经元状态改变, 并将状态传递到下一个神经元。在人工神经网络中, 激活函数模拟了神经元的激活阈值和信号传递机制。总输入通过激活函数 $f(\cdot)$ 后, 得到当前人工神经元 j 的输出 y_j :

$$y_j = f(\mathbf{S}_j) = f\left(\sum_{i=0}^n w_{ji} x_i\right) = f(\mathbf{w}_j \mathbf{x}) \quad (1.3)$$

因为神经元传递的信号随输入的增加而加强, 但不可能无限制地增加, 必有一个最大值, 所以式(1.3)中 $f(\cdot)$ 是单调上升函数, 而且必须是有界函数。

M-P 模型基于以下几点假设:

- (1) 每一个神经元是一个多输入单输出的信息单元;
- (2) 神经元输出有阈值特性, 当输入总和超过阈值时, 神经元才被激活, 未超过阈值时, 神经元不会被激活;
- (3) 神经元本身是非时变的, 即其突触延时和突触强度均为常数。

需要指出的是, ANN 是对生物神经网络的某种抽象、简化和模拟, 人类尚未完全实现对人脑神经系统功能的真实描述。

3. 网络结构

人工神经网络的互连结构(或称拓扑结构)是指神经元之间的连接模式, 它是构造神经网络的基础, 也是形成不同神经网络的主要来源。从结构上进行区分, 人工神经网络主要可分为前馈神经网络和反馈神经网络。

1) 前馈神经网络

前馈神经网络, 简称前馈网络, 各神经元从输入层开始接收前一级输入, 并输出到下一级直至输出层。整个网络中无反馈, 可用一个有向无环图表示。前馈网络又可分为单层前馈网络和多层前馈网络两种形式。

(1) 单层前馈网络结构如图 1.3 所示。

(2) 多层前馈网络结构如图 1.4 所示。

多层前馈网络指除拥有输入、输出层外, 还含有一个或多个隐藏层的前馈网络。其中, 隐藏层是由既不属于输入层又不属于输出层的神经元所构成的处理层, 也称为中间层。多层前馈网络的输入层的输出是第一隐藏层的输入信号, 而第一隐藏层的输出则是第二隐藏层的输入信号, 以此类推, 直到输出层。

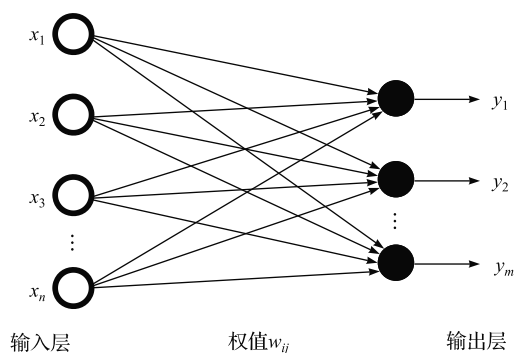


图 1.3 单层前馈网络结构

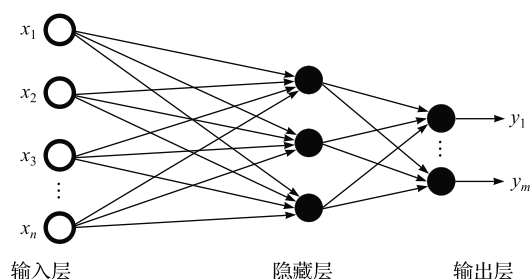


图 1.4 多层前馈网络结构

2) 反馈神经网络

反馈神经网络，又称自联想记忆网络，是指允许采用反馈连接方式的神经网络。反馈连接方式是指一个神经元的输出可以被反馈至同层或前层的神经元，其典型结构如图 1.5 所示。

前馈网络属于非循环连接模式，它的每个神经元的输入都没有包含该神经元先前的输出；反馈神经网络每个神经元的输入都有可能包含该神经元先前输出的反馈信息，类似于人类的短期记忆。

反馈神经网络的目的是设计一组稳定的平衡点(吸引子)，使得网络在接收到初始输入后，通过内部动态演化自动收敛到这些预设的平衡状态。通过反馈连接和非线性动力学特性，反馈神经网络的状态在演化过程中趋于稳定。

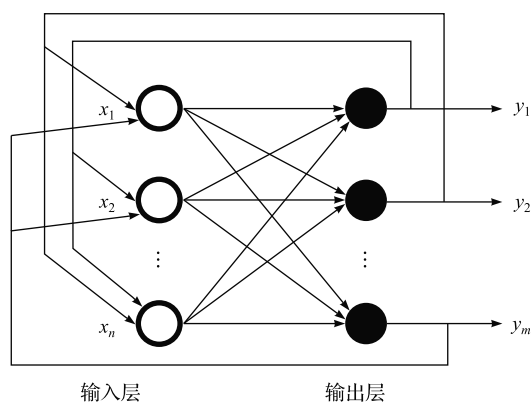


图 1.5 反馈神经网络结构

3) 反馈神经网络与前馈神经网络的区别

(1) 结构不同。

前馈神经网络：没有反馈环节。

反馈神经网络：一个动态系统，存在稳定性问题，这也是反馈神经网络的关键问题。

(2) 模型不同。

前馈神经网络：模型是从输入到输出的映射关系，不考虑延时。

反馈神经网络：考虑延时，是一个动态系统，模型是动态方程(微分方程)。

(3) 网络工作过程不同。

前馈神经网络：通过学习得到连接权值，然后完成指定任务。

反馈神经网络：设定网络的初始状态，然后系统运行，若系统是稳定的，则最后将达到一个稳定状态。

(4) 学习方法不同。

前馈神经网络：使用误差反向传播算法，主要是反向传播(back propagation, BP)算法。

反馈神经网络：通常使用赫布(Hebb)算法与 BP 算法。Hebb 算法基于 Donald Hebb 在 1949 年提出的一种学习规则，该规则用来描述神经元的行为如何影响神经元之间的连接。该规则及其扩展内容可以概括为如果相连接的两个神经元同时被激活，则可以认为这两个神经元之间的关系比较近，因此将这两个神经元之间连接的权值增加；若一个被激活而另一个被抑制，则两者间的权值应该减小。

(5) 应用范围不同。

前馈神经网络：主要用于静态模式识别、分类、回归任务。

反馈神经网络：主要用于联想记忆、时序预测、动态系统建模等。

1.1.2 深度神经网络

为了使神经网络模型具有更强的数据表示能力和非线性分类能力，可以考虑增加神经网络模型的复杂度。提升神经网络模型的复杂度通常有两种方式：第一，增加神经网络的层数(尤其是隐藏层的数量)，模型可以逐步学习更抽象、更复杂的特征表示，每一层对输入数据进行非线性变换，形成分层的特征提取过程；第二，增加每一层的神经元数量(或通道数)，模型可以并行学习更多特征，提升对输入数据的表征能力。

上述两种方式分别增加了网络的深度与宽度，深度更关注特征的层级抽象(从低级到高级)，而宽度更关注特征的多样性(同一层级的多个特征)。在实际任务中，需根据数据特点和硬件资源灵活选择侧重于哪种方式。此外，还可以结合复杂的网络结构模块，如注意力机制、残差连接(residual connection)、多尺度特征融合等，以及设计并采用更有效的激活函数，进一步增强神经网络模型。

深度神经网络的实质是通过构建具有很多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。但随着神经网络中隐藏层的增多，训练难度也会成倍增加。虽然 BP 算法理论上能训练任意深度的神经网络，但误差在多个隐藏层内逆传播时，往往会“发散”而不能收敛到稳定状态。误差信号逐渐衰减会导致梯度消失，而误差信号持续增大会导致梯度爆炸，从而使得深度神经网络训练失败。

2006 年，Hinton 等提出了基于深度信念网络(deep belief network, DBN)的非监督贪心逐层训练算法，为解决深层网络结构相关的优化训练难题带来希望，即“预训练(pre-training)+微调(fine-tuning)”的策略。预训练是指每次训练一层隐藏层节点，训练时将上一层隐藏层节点的输出作为输入，而本层隐藏层节点的输出作为下一层隐藏层节点的输入；微调的含义是在预训练全部完成后，对整个网络进行监督学习微调训练，微调训练一般使用 BP 算法。“预训练+微调”的方法可以视为将大量参数分组，为每组先找到局部看起来比较好的设置，然后将这些局部较优的结果联合起来进行全局寻优。

此外，LeCun 等提出的卷积神经网络(convolutional neural network, CNN)采用了权值共享的策略，即一组神经元使用相同的连接权值来降低深度神经网络的训练难度。

1.2 深度学习的数学原理

深度学习并不是黑箱式的操作，背后有其理论依据和数学描述。下面将介绍深度学习的一些重要数学原理。

1.2.1 激活函数

人工神经元考虑了生物神经元三个最重要的功能：

加权——可对每个输入信号进行不同程度的加权；

求和——确定全部输入信号的组合效果；

激活——通过激活函数 $f(\cdot)$ ，确定其输出。

通过对这三个功能的模拟可基本反映生物神经元的主要特性。在深度学习中，激活函数起着非常重要的作用。它将神经元接收到的信号进行非线性映射，提高神经网络的表达能力，使其能够更好地学习数据的特征。

在神经网络领域，常用的激活函数主要有以下几个。

1. 阈值型激活函数

阈值型激活函数如图 1.6 所示。

从图 1.6 中可以发现，这种模型的神经元没有内部状态，激活函数 $f(x)$ 是一个阶跃函数。

2. 分段线性强饱和型激活函数

分段线性强饱和型激活函数又称为伪线性激活函数，其输入和输出之间在一定范围内满足线性关系，一直延续到输出最大值 1 为止，但当达到最大值后，输出就不再增加，如图 1.7 所示。

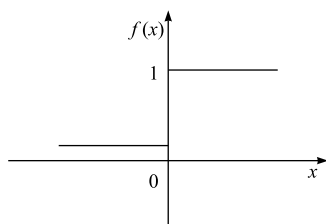


图 1.6 阈值型激活函数

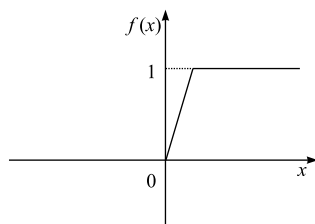


图 1.7 分段线性强饱和型激活函数

3. S 型激活函数

S 型(sigmoid)激活函数是重要的激活函数之一，其表达式为

$$f(x) = \text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (1.4)$$

其函数图像如图 1.8 所示。

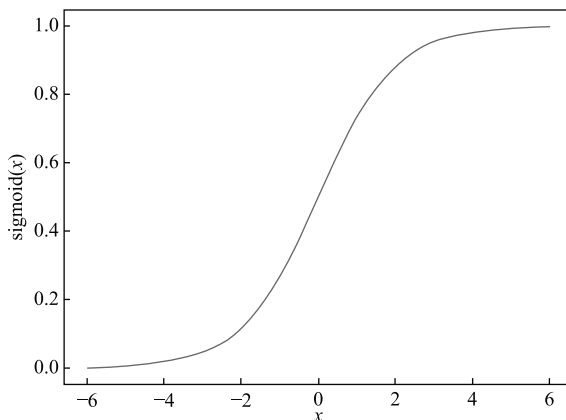


图 1.8 sigmoid 激活函数

对该函数求导可得

$$f'(x) = f(x) \cdot [1 - f(x)]$$

可见该函数的特性一是连续可微，二是导数形式便于计算，因此成为神经网络中常用的激活函数。

4. 子阈累积型激活函数

子阈累积型激活函数如图 1.9 所示，它也是一个非线性函数。当产生的激活值超过 T 时，该神经元被激活产生反响。在线性范围内，系统的反响是线性的，超出线性范围后响应达到最大值。

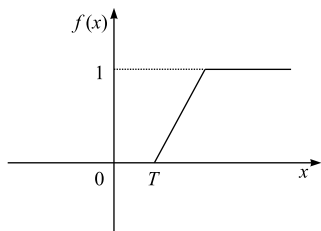


图 1.9 子阈累积型激活函数

进入深度学习阶段，对激活函数的研究也得到了拓展，目前比较主流的激活函数还有以下形式。

5. tanh 激活函数

tanh 激活函数也称为双曲正切激活函数，其表

达式为

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (1.5)$$

其函数图像如图 1.10 所示。

tanh 激活函数形状和 sigmoid 激活函数很相似，根据 tanh 和 sigmoid 激活函数的定义可知 $\tanh(x) = 2\text{sigmoid}(2x) - 1$ ，因此 tanh 激活函数可由 sigmoid 激活函数伸缩平移得到，所以 tanh 激活函数具有和 sigmoid 激活函数类似的特点。

tanh 激活函数的特点如下：

(1) 连续光滑、严格单调；

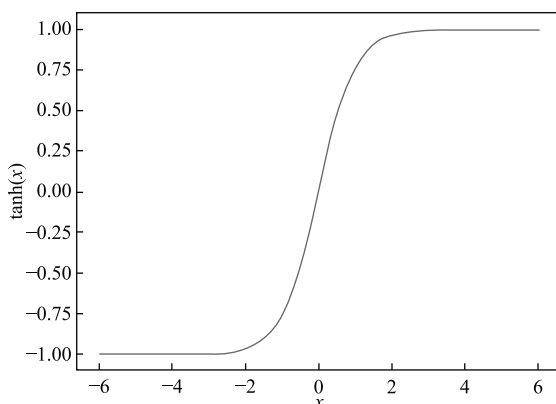


图 1.10 tanh 激活函数

(2) 输出范围为 $(-1,1)$ ，以 $(0,0)$ 为对称中心，均值为 0；

(3) 输入在 0 附近时，输出变化明显，输入离 0 越远，输出变化越小，最后输出趋近于 1 不变。

根据 tanh 激活函数的特点可以分析其优缺点。优点：输出关于原点对称，均值为 0，因此输出有正有负，可以规避 zigzag 现象；另外，关于原点对称本身是一个很好的优点，有益于网络的学习。缺点：存在梯度消失问题，当 $f(x)=\tanh(x)$ 时，其导数计算为

$$f'(x) = \frac{4e^{2x}}{(e^{2x} + 1)^2}, \text{ 取值范围为 } (0,1], \text{ 其取值范围相对 sigmoid 激活函数的导数而言更广,}$$

可在一定程度上改进梯度消失问题，但仍然不能从根本上解决随着网络层数增多，梯度连乘导致梯度消失这一问题。

6. ReLU 系列激活函数

ReLU(rectified linear unit)系列激活函数包括 ReLU、Leaky ReLU、PReLU、ELU 等，以及在 PReLU 基础上发展出的 Dice 函数。这个系列的激活函数采用分段的思想，其中一段为线性，从而解决部分梯度消失的问题。

ReLU 激活函数的数学表示为

$$f(x) = \text{ReLU}(x) = \max(0, x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (1.6)$$

其函数图像如图 1.11 所示。

ReLU 激活函数的分段性使其具有如下优点：

- (1) 输入 ≥ 0 时可保持梯度为恒定值不衰减，从而缓解梯度消失问题；
- (2) 输入 < 0 时导数为 0，当神经元激活值为负值时，梯度不再更新，增加了网络的稀疏性，从而使模型的鲁棒性更好；
- (3) 计算速度快，ReLU 激活函数的导数是 if-else 的实现逻辑，计算方便快捷。

ReLU 激活函数的缺点主要有如下几点：

- (1) 输入 ≥ 0 时梯度为 1，可能导致梯度爆炸问题；

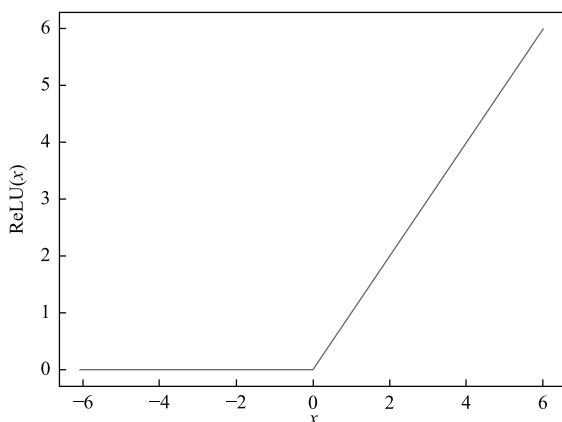


图 1.11 ReLU 激活函数

(2) 输入 < 0 时导数为 0，一旦神经元激活值为负，神经元就进入永久性 Dead 状态，梯度不再更新，导致梯度消失，学习率过大容易导致所有神经元都进入 Dead 状态，所以需设置较小的学习率；

(3) ReLU 激活函数的输出均值大于 0，容易改变输出的分布，但可以通过批归一化 (batch normalization) 改善这个问题。

总体而言，相比 sigmoid 和 tanh 激活函数，ReLU 激活函数可以改善梯度消失问题，且计算高效，因此使用比较广泛。

针对神经元激活值为负时神经元进入永久性 Dead 状态的问题，可以对 ReLU 激活函数负半轴的输出进行调整，使其不为 0，从而解决该问题。Leaky ReLU、ELU、SELU 等激活函数便是采用这种思路解决神经元永久性 Dead 问题。

以 Leaky ReLU 为例，其数学表达为

$$f(x) = \text{Leaky ReLU}(x) = \begin{cases} \alpha x, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (1.7)$$

参数 α 通常取非常小的正数，即负半轴的输入、输出线性斜率很小，其函数图像如图 1.12 所示。

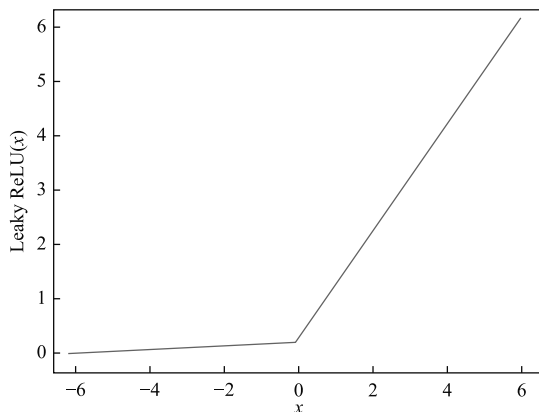


图 1.12 Leaky ReLU 激活函数