卓越工程师培养系列

深度学习理论与实践

曹文明　王　浩　主　编

全　智　何志权　温　阳　副主编

清華大學出版社

北　京

内 容 简 介

深度学习是计算机科学的一个重要分支，是一种以人工神经网络为架构，对数据进行表征学习的算法的总称。深度学习是传统机器学习算法的发展和衍生，相关内容涉及代数、统计学、优化理论、矩阵计算等多个领域。本书是深度学习的基础入门级教材，在内容上尽可能覆盖深度学习算法相关基础知识。全书共 11 章，大致可分为三大部分：第一部分(第 1～3 章)主要介绍机器学习的基础知识和一些传统算法；第二部分(第 4～8 章)主要介绍人工神经网络等的相关理论、优化算法和各类经典神经网络模型；第三部分(第 9～11 章)为进阶知识，主要介绍非监督学习和强化学习的相关算法。

在学习本书的过程中，读者不仅要深入理解相关算法理论，更要多思多练。读者在阅读各章节内容后，可基于各章习题巩固知识，并将理论与实践结合，基于 torch、tensorflow 等深度学习平台在实际任务中演练所学理论知识和技能。本书可作为高等院校计算机或电子信息相关专业的本科生或研究生教材。

图书在版编目(CIP)数据

深度学习理论与实践 / 曹文明，王浩主编. —北京：清华大学出版社，2024.1
(卓越工程师培养系列)
ISBN 978-7-302-63466-9

Ⅰ. ①深… Ⅱ. ①曹… ②王… Ⅲ. ①机器学习－研究 Ⅳ. ①TP181

中国国家版本馆 CIP 数据核字(2023)第 083742 号

责任编辑：王 定
封面设计：周晓亮
版式设计：思创景点
责任校对：马遥遥
责任印制：宋 林

出版发行：清华大学出版社
网 址：https://www.tup.com.cn，https://www.wqxuetang.com
地 址：北京清华大学学研大厦 A 座 邮 编：100084
社 总 机：010-83470000 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
印 装 者：北京同文印刷有限责任公司
经 销：全国新华书店
开 本：185mm×260mm 印 张：16.25 字 数：375 千字
版 次：2024 年 1 月第 1 版 印 次：2024 年 1 月第 1 次印刷
定 价：79.80 元

产品编号：097051-01

前　言

自古以来，人类文明的发展始终伴随着对智能化的不懈追求。早期的人类通过总结经验、发现规律来改善工具，提升生产、生活的智能化水平。人类社会进入工业化时代以后，随着科学技术的发展和相关知识的演进，工具的智能化水平得到快速提升。近年来，随着计算机、互联网等的兴起，人类社会步入大数据时代。当前，智能化相关的研究成果呈井喷式爆发。与此同时，新的生活和生产环境又对智能化技术的演进提出了更高的要求。习近平总书记在党的二十大报告中强调，要加强科技基础能力建设，推动战略性新兴产业融合集群发展，构建新一代信息技术、人工智能、生物技术等一批新的增长引擎。

深度学习相关技术逐渐成为目前工业界和学术界探索和研究的重点，并且越来越多的相关算法在实际应用中取得了广泛的成功。这些算法的成功和发展离不开相关研究者对深度学习相关理论的探索。目前的深度学习算法依然存在黑盒属性、需要大量训练样本、可解释性差、调参困难、算法模型规模大、学习时效低等不足，并且和人类本身的学习方式及智能化水平相比，仍有着巨大的发展空间。这些也要求我们深入理解当前深度学习的相关算法理论，针对其特性和缺点不断加以改进提升。

本书的写作目的是将深度学习的相关算法理论和应用实践深入浅出地介绍给读者，力求使大学本科低年级学生能够理解和掌握相关内容。本书一共包含 11 章。第 1 章绪论，简要介绍人工智能、机器学习、深度学习的基本概念、发展历程和一些典型的应用实例。第 2 和第 3 章，基于回归和基础分类两个基本模型介绍机器学习的基础知识，为后续深度学习相关内容提供知识基础。第 4 和第 5 章是深度学习的理论基础章节，阐述人工神经网络基础，学习模型的训练和优化算法改进方法，以及模型的相关效果评估和具体实现方式。第 6 和第 7 章是深度学习理论在图像数据和序列数据处理上的应用和发展，分别介绍卷积神经网络及其经典模型、循环神经网络及其经典模型。第 8 章是深度学习理论在自然语言处理问题上的拓展和应用，重点阐述注意力机制及其经典模型，并探讨模型在自然语言处理中的算法演进。第 9 章是深度学习理论在网络图数据处理问题上的拓展和应用，重点阐述基于谱域和空间域的各类图神经网络模型。第 10 章探讨针对无监督特征学习问题的传统机器学习算法和相关深度学习模型。第 11 章介绍强化学习的相关理论和经典模型。

本书由主编曹文明负责整体编写和质量把控；王浩负责章节规划以及第 3 章到第 8 章的编写；全智负责第 1 章和第 2 章的编写；何志权负责第 9 章的编写；温阳负责第 10 章和第 11 章

的编写。此外，感谢广东省多媒体信息服务工程技术中心全体人员的帮助，感谢张婉莹、钟建奇、刘启凡、闫志越、宋晓宝、陈作胜、刘伊善、张汝君、蓝旭佳、熊敏等在材料收集、校稿等方面的建议和帮助。

因作者能力有限，书中难免有不当之处，还望读者海涵和指正。

本书提供教学课件、教学大纲、电子教案和教学视频，读者可扫下列二维码获取学习。


教学课件


教学大纲


电子教案


教学视频

曹文明　王浩　全智　何志权　温阳
于深圳大学

2023 年 8 月

目 录

第 1 章 绪 论

人类对工具智能化的追求自古就有。在我国漫长历史中，早在几千年前就有机关器械和占卜预测之术的描述。虽然很多的记载现在没有办法进行科学证实，但是这些充分体现了我国祖先对智能化的不懈追求。在 20 世纪 50 年代初，阿兰·图灵(Alan Turing)发表了著名的论文《计算机器与智能》(*Computing Machinery and Intelligence*)，并提出了图灵测试，这为随后几十年人工智能的发展指引了方向。之后，约翰·麦卡锡(John McCarthy)在 1956 年明确提出了人工智能(artificial intelligence，AI)的概念，并对其进行了初步定义：“人工智能指的是之前由人工执行的任务，现在由机器来代替执行。”此后，越来越多的研究者加入到人工智能的相关研究中，这个领域开始蓬勃发展起来。在我国，人工智能技术的研究和应用得到了高度重视，大多数的高等院校和大量的公司加入了这个赛道，对人工智能技术展开研究和探索，这大大加速了人工智能算法的应用、落地和推广。

人工智能作为一个跨学科的研究领域，其知识体系的涵盖范围非常广泛，如机器学习、人工神经网络等都是其中重要的内容。在日常生活中，人工智能的应用范围也非常广泛，如人脸识别、自动驾驶、智能家居、语音识别、文本识别等人们可以直观感受到的技术。当然也有一些人们在日常生活中使用，但很难直观察觉到的技术，如智能推荐、搜索排序等。人工智能可以给人类的生活带来便利。其可以自主完成各种任务，解放生产力，发展惠及大众。当然，任何科技本身都具有两面性，人工智能在给人们的生活带来便利的同时，也无疑会带来一些隐患。例如，随着人工智能的发展，人们的个人隐私与信息安全受到了严重的威胁；人工智能对人类生产力的释放可能导致大量现有的工作岗位被机器替代。因此，我们要成为掌握人工智能相关技术的人，这不仅仅有益于我们个人的发展，而且能让更多的人参与到这个领域的技术革新中来，同时也有助于规范技术的发展，从而减少由于相关技术的进步而给人类社会带来的负面影响。

在人工智能领域中，一个重要的研究方向就是机器学习(machine learning)。具体而言，机器学习主要探究如何令机器通过模拟人类的学习行为，来改善自身的性能。机器学习的具体概念是 1952 年由亚瑟·塞缪尔(Arthur Samuel)提出的。它作为人工智能的一部分，为人工智能在日常生活中的应用提供了理论基础，推动了人工智能的发展。本书也将对机器学习中的基础回归模型与基础分类模型进行简单介绍。

一般而言，机器学习算法在进行相关任务处理时主要包含两个步骤：①特征提取；②基于所提取的特征进行问题决策。但机器学习算法在进行特征提取时往往需要大量的人工辅助和干预，这会带来巨大的资源消耗。为了解决这个问题，大量研究者开始对特征提取的自动化方法展开研究。在 1957 年，弗兰克·罗森布拉特(Frank Rosenblatt)在亚瑟·塞缪尔的研究基础上，结合唐纳德·赫布(Donald Hebb)提出的类似脑细胞相互作用模型，创造了感知机算法。事实上，正是感知机的提出，开启了人们对深度学习(deep learning)的探索之路。深度学习是机器学习领域的重要研究方向。和传统机器学习算法相比，深度学习是一种端到端的学习方式，它极大地减少了特征提取阶段的人为干预，同时进一步提升了模型的学习效果。深度学习的出现不仅丰富了数据处理的方式，而且进一步扩大了人工智能算法的应用范围。

在感知机的基础上，1967 年专家提出了多层感知机概念，之后在 1986 年，适用于多层感知机(multilayer perception，MLP)训练的反向传播(backpropagation)算法被提出。受限于当时的硬件设备条件，一段时间内该方法并未受到广泛的关注。2006 年，杰弗里·辛顿(Geoffrey Hinton)提出使用无监督预训练权值进行初始化与有监督反向传播微调去解决多层网络梯度消失问题的技巧，从而使多层感知机和反向传播算法得到了进一步的传播和推广。2012 年，在 ImageNet(一个用于视觉对象识别软件研究的大型可视化数据库) 图片分类比赛中，基于深度学习的 AlexNet 一举夺魁。它的成功让更多的人投入到深度学习的相关研究中。

深度学习结构设计的最初灵感来源于人脑神经元。人脑的神奇之处在于，对从外部获取的图像、文本、音频等各类感知信息能高效快捷地进行提炼，并依据相关信息做出分析决策。深度学习就是模拟人脑的这种工作机制，直接将数据输入人工神经网络，然后由模型提炼信息并做出相应的分析或决策。深度学习算法在运行过程中会自动完成对特征信息的提取。和传统机器学习算法的特征提取方式相比，深度学习算法在特征的选择和特征的表征能力上更好地契合了任务本身的需求。这也是在项目实践中，深度学习方法优于传统方法的重要原因。

人工智能、机器学习、深度学习 3 个基本概念的关系如图 1.1 所示，本书将由外至内、由浅入深地介绍人工智能特别是深度学习的相关算法。

早在 2015 年 7 月，国务院就发布《关于积极推进“互联网+”行动的指导意见》，第一次把人工智能上升为国家战略。2022 年 10 月，党的二十大报告明确提出推动战略性新兴产业融合集群发展，构建包括人工智能在内的一批新的增长引擎。深度学习在人们的现实生活中发挥着重要的作用。在医疗行业、交通运输业、制造业、建筑业、娱乐业、安防领域、教育领域甚至农业领域中都能看到深度学习的身影。下面将通过几个实例来展示深

度学习算法在相关领域的应用现状。

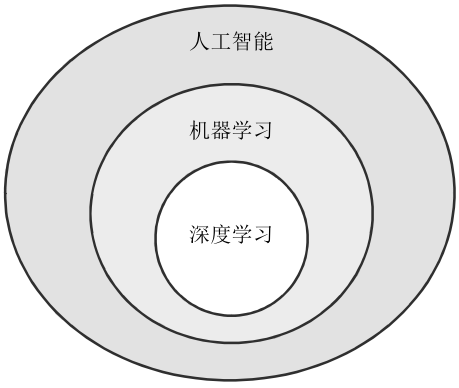


图 1.1 人工智能、机器学习、深度学习 3 个基本概念之间的关系

1.1 疫情防控中的应用

在新冠疫情暴发初期，医疗工作者需要通过肺部的 CT(computed tomography，电子计算机断层扫描)影像和 X 光射线图来确诊感染者，于是产生了大量的数据，需要大量放射学专家进行人工判别。这使疫情的初期筛查工作进展极为缓慢，进而影响了国家对疫情的有效管控。为了解决这个问题，相关研究者构建了基于深度学习的分类模型来辅助医疗工作者对患者进行初期筛查工作。随着数据量的累计和算法的调优，相关分类算法的准确率已接近人工检测标准，可以有效缓解医疗工作者在应对新冠方面的压力。

在筛查患者及防控疫情的同时，人们也在探寻着治愈新冠病毒感染的方法。通过技术手段可以检测出新冠病毒的蛋白质结构。此时需要创造出一种能够攻破其结构的新型蛋白质。这种新型蛋白质需要基于已有的安全蛋白结构进行组合，在设计目标上，不仅要其在空间结构上合理，还需要能够有效对抗新冠病毒。为达到这个目的，生物领域的科研人员往往需要先基于生物学知识进行结构设计，再进行结构合成，最后通过 X 光射线结晶学、核磁共振成像或低温电子显微镜对结构进行实验验证。上述研发过程既昂贵又耗时。为了提升效率、加快进度，研究人员设计了一种能够预测有效蛋白质结构的深度学习模型。具体来说，该模型将蛋白质模板数据集及蛋白质自由模型数据集作为输入数据，将结构预测问题转换为最邻近问题来训练网络，进而实现对潜在有效蛋白质结构的预测。这样的算法能够大大缩短研发时间，极大地降低了任务的试错成本。

新冠疫情的防控程度和防控措施难免会对公民的心理健康产生影响。因此，社会学家需要从经济、心理和社会学等多个角度分析衡量公众当前的心理状况。为了达到这个目的，需要依据微博等网络舆情数据对不同地区的公众情绪进行分析。对此，深度学习的自然语言处理(natural language processing，NLP)算法就有了用武之地。具体而言，针对与疫情相关的关键词可以构建一个情感语义数据集。然后，利用该数据集来调适出一个用于自然语言处理任务的深度学习模型，最后利用该模型对网络信息进行分析，就可以实现有效

的舆情管控和防谣治谣。

1.2 自动驾驶中的应用

自动驾驶研究大致可分为以下两种方案。

(1) 基于规则约束的决策方案：按照感知、规划、控制、执行的流程来进行自动驾驶系统的实现，每个模块内都有各自的深度学习模型来处理不同的数据输入。

(2) 端到端(end-to-end)的决策方案：系统从环境感知与定位模块获取实时数据后，经过一个深度学习的模型直接输出车辆的控制指令，即从感知直接映射到控制命令。

基于规则约束的决策方案更贴近人类在驾驶车辆时的操作逻辑，所以可解释性好，一旦出现系统问题，人们就可以快速诊断并且有针对性地解决问题。其缺点是系统复杂，成本高昂、计算量大，多个模型之间可能存在误差累积，使系统难以达到最佳性能。而端到端的决策方案虽然不具有可解释性，但是相对于基于规则约束的决策方案，其成本和复杂度都较低。但端到端的决策方案也存在缺陷，首先，此类方法的可解释性差，其次，对于不同的车辆或传感器，系统都需要重新校准。

以基于规则约束的决策方案为例，其感知模块相当于人类的眼睛，主要基于深度学习算法来处理各类传感器收集到的环境数据。感知模块大体又分为单目方案和双目(多目)方案。在单目方案中，系统往往不能够快速地感知图像的深度信息，所以在这种方案中，深度学习模型往往是先进行目标检测，再将检测出的目标物体与数据库的模型进行匹配，从而完成距离推算。这样的方案对数据库的数据量要求较大，并且若目标检测算法没能有效检测出障碍物，则很容易造成接下来的路径规划错误，影响比如行驶安全。因此，在单目方案中，常将激光雷达或毫米波雷达的数据与摄像头数据进行融合，从而更精准地定位目标。当存在两个以上的摄像头时，系统可以通过建立三角坐标系来测算距离。因此，在双目(多目)方案中，深度学习模型不需要从数据库中枚举匹配，而是通过实时的计算来检测目标。这样的方案对硬件算力有着极高的要求，且需要用其他传感器或更多的摄像头来弥补视觉盲区。但是在现实环境下，各种天气变化会对图像数据造成干扰，所以也须将激光雷达或毫米波雷达的数据与摄像头数据进行数据融合，从而更精准地定位目标。

规划模块又分为路线规划和轨迹规划。路线规划是一种宏观规划，就如同我们使用的各种电子地图一样，通过输入起点位置与终点位置来规划行驶路线。通常采用混合整数、动态规划、启发式算法或强化学习等方法来建立路线规划模型。轨迹规划则是短期的规划，是根据感知模块对当前环境的监测情况来调控短期路径的模型。这样的模型一般要求实时计算，对于车载计算平台的计算能力要求较高。

执行模块又称为决策模块，它主要处理的数据为时序性数据。也就是说，轨迹规划模型根据环境来输出备选路径，供执行模块选择，而执行模块需要决定何时停车，何时加速超车或何时减速等。人类在驾驶汽车时，决策往往需要根据经验来做出，所以执行模块需要结合不同时间的时序数据来推断各种决策产生的结果，然后进行决策选择并输出至执

行模块。一般采用强化学习模型来作为执行模块，这是因为强化学习模型能够在一定程度上代替人类的经验判断。

控制模块是用于替代人类的操作的模块，主要控制 3 个执行器：方向盘、加速踏板和制动踏板。根据执行模块输出的执行任务及规划模块输出的路径，控制模块需要合理地控制 3 个执行器并尽可能地按照任务规划的预期效果来控制车辆运动。在控制的过程中还需要根据采集到的车速、加速度和航向角等数据来实时修正控制量。

深度学习模型在上述 4 个模块的运行过程中都起到了至关重要的作用。

1.3 现代农业中的应用

在农业方面，目前深度学习还处于起步阶段。近期，一项特殊的比赛让我们看到了深度学习在农业方面的作用。

该赛事的参赛选手，一方为顶尖农人队伍，依靠经验种植农作物；另一方为深度学习算法队伍。双方在草莓的品质、产量、投入产出比等指标上展开比拼。在赛事进行的 4 个月时间内，深度学习算法队需要构建草莓种植管理系统，让算法模仿人类专家，实现草莓种植的自动化。在这套自动化系统中，深度学习模型需要处理各种传感器采集的温度、湿度、水肥等数据，并根据草莓的生长模型来向外部设备输出决策。整个过程的难点在于让深度学习模型形成实时且正确的决策。

最终大赛揭晓了深度学习算法队伍和顶尖农人队伍在产量、投入产出比和甜度 3 项指标上的对战结果：深度学习算法队伍的草莓产量平均值高出顶尖农人队伍平均值的 196.32%，投入产出比平均值高出顶尖农人队伍的平均值的 75.51%；而顶尖农人队伍的果实甜度整体平均值高出深度学习算法队伍的平均值的 5.24%以上。

可以看出，深度学习模型在有相关经验模型的支持下，在决策的及时性及正确性上能够与农业专家媲美。比赛举办方表示将探索出一批适用于小农生产模式的、低成本的、可复制的智能化农业应用模型，并且通过将人工经验数字化来为农产区提供植物智能化种植模型。本次比赛的获奖成果正在转换为数字农业解决方案，并已经在田间地头应用。

以上只是深度学习实践应用的冰山一角。事实上，深度学习已经融入人类社会的方方面面，并且在不同的研究领域都有出色的表现。在我国二十大战略的指引下，以深度学习为重要技术的人工智能必将得到蓬勃的发展，对推动我国数字化经济的高质量发展起到重要的作用。掌握深度学习的知识，能够更好地处理领域交叉或复杂数据问题。理解并掌握深度学习算法和相关知识，已经成为我国当代新型工科人才培养的重要目标。

第 2 章

基础回归模型

回归(regression)与分类(classification)是机器学习领域的两个基本问题。其中，回归模型在经济预测、数据挖掘、疾病自动诊断、销售预测和风险评估等方面有广泛的应用。目前，在我国回归模型已经被应用到多个领域。例如，运用回归模型定量估测新冠疫情对我国宏观经济基础性指标 GDP(gross domestic product，国内生产总值)的影响；通过构建回归模型分析全国人口增长规律，预测中国人口的未来增长趋势；在我国生物医药领域，使用多元回归模型预测各类疾病的生存期；在制造业领域，以中国各省域制造业工业生产总值作为研究对象，利用多元回归模型研究经济指标影响因素。本章围绕回归问题介绍传统机器学习算法中几种常见的回归模型，包括线性回归模型、参数估计模型，同时还对相关模型的优化求解方法和效果评估函数展开讨论。本章所涉及的概念和方法是开展深度学习的重要理论基础。

2.1 线性回归模型

回归分析(regression analysis)通常用来探究一个或多个预测变量与响应变量之间的关联关系。其中，预测变量也称为自变量，通常用 $x_1, x_2, \cdots, x_p$ 表示；响应变量也称为因变量，通常用 y 表示。按照自变量和因变量之间的关系类型，回归模型可分为线性回归模型和非线性回归模型。本节主要介绍回归分析中的线性回归模型。

线性回归(linear regression)是利用回归分析，对自变量和因变量之间的线性关系进行建模的统计分析方法。其表达形式是一个或多个被称为回归系数的模型参数的线性组合。根据自变量的数量，线性回归可分为两种类型，即一元线性回归和多元线性回归。

2.1.1 一元线性回归

一元线性回归的目标是探究单个自变量与因变量之间的线性关系。举个简单的例子：假设房屋面积 x 与房屋价格 y 之间存在简单的线性关系 $y=wx$，其中 $w>0$ 为比例系数，即房屋价格与房屋面积成正比关系。然而在许多实际场景中，自变量与因变量之间并不是简单的正比关系。为了让模型更具有普适性，我们在比例关系的基础上引入常数项 w_0。因此针对单一样本，一元线性回归模型通常写作

$$y=w_0+w_1x+\varepsilon\,, \tag{2.1}$$

其中，y 是实际观测值也称因变量，x 是自变量，w_0 和 w_1 表示模型的回归系数，ε 表示模型预测值 $\hat{y}$ 与实际观测值 y 之间的拟合误差。在线性回归中，ε 一般与自变量相互独立，其分布服从期望 $E(\varepsilon)=0$ 且方差 $\mathrm{var}(\varepsilon)=\sigma^2$ 的正态分布(也称为高斯分布)。

图 2.1 展示了利用一元线性回归模型拟合数据的实例。从图中可以看出，数据的真实值并未全部落在回归线上，而是分布在回归线的周围。这说明真实值 y 与预测值 $\hat{y}$ 之间存在拟合误差 ε，该误差通常来自除自变量 x 外其他潜在因素的影响。

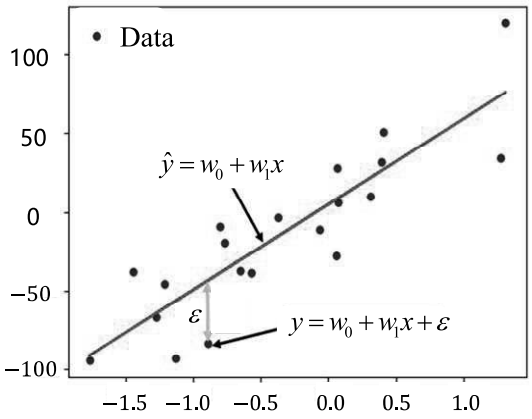


图 2.1 一元线性回归模型

2.1.2 多元线性回归

多元线性回归的目标是探究多个自变量与因变量之间的线性关系。例如，影响房屋价格的因素不止房屋面积这一个因素，还有地域、房间数、布局等诸多因素。假设有 n 个影响因素，且这些因素对房价的影响程度是不同的。于是，我们可以通过对这些因素赋予不同的权重 $w_1,w_2,\cdots,w_n$ 来表示它们对因变量的影响程度。因此多元线性回归模型可写作

$$y=\boldsymbol{w}^{\top}\boldsymbol{x}+\varepsilon\,, \tag{2.2}$$

其中，y 是因变量，向量 $\boldsymbol{x}=\left[x_0,x_1,x_2,\cdots,x_n\right]^{\top}$ 表示自变量(其中 $x_0=1$)，$\boldsymbol{w}=\left[w_0,w_1,w_2,\cdots,w_n\right]^{\top}$ 表示模型的回归系数，ε 表示模型的拟合误差。

对于某个数据集而言，我们通常需要对数据集中的每一个样本都进行回归分析。所以对于一个包含 m 个样本的数据集，我们有如下方程组：

$$\begin{cases} y^{(1)} = \boldsymbol{w}^{\top}\boldsymbol{x}^{(1)} + \varepsilon_1, \\ y^{(2)} = \boldsymbol{w}^{\top}\boldsymbol{x}^{(2)} + \varepsilon_2, \\ \vdots \\ y^{(m)} = \boldsymbol{w}^{\top}\boldsymbol{x}^{(m)} + \varepsilon_m, \end{cases} \tag{2.3}$$

其中，$\boldsymbol{x}^{(i)} = \left[x_0^{(i)}, x_1^{(i)}, \cdots, x_n^{(i)}\right]^{\top} \left(\forall i \in [1,2,\cdots,m]\right)$ 表示第 i 个样本的特征值向量，$x_j^{(i)} \left(\forall j \in [0,1,\cdots,n]\right)$ 表示第 i 个样本的第 j 个特征值，$\varepsilon_i \left(\forall i \in [1,2,\cdots,m]\right)$ 对应不同样本的拟合误差，$y^{(i)} \left(\forall i \in [1,2,\cdots,m]\right)$ 表示第 i 个样本的实际观测值。

定义 2.1　(线性回归模型的矩阵表示)

基于上述分析，当我们对一个完整的数据集进行多元线性回归分析时，数据集包含 m 个样本，且每个样本有 n 个特征值，则相应的线性回归模型可以表示为

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\varepsilon}, \tag{2.4}$$

即

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^{(1)^{\top}} \\ \boldsymbol{x}^{(2)^{\top}} \\ \vdots \\ \boldsymbol{x}^{(m)^{\top}} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix} + \begin{bmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_m \end{bmatrix} \tag{2.5}$$

图 2.2 中给出了一个使用二元线性回归模型拟合数据的实例。从图中可以看出，二元线性回归模型的拟合结果是一个平面，数据的真实值分布在平面附近。由此可以推出，多元线性回归模型的拟合结果为多维空间中的某个超平面。它是平面中的直线和空间中的平面在高维空间中的推广。

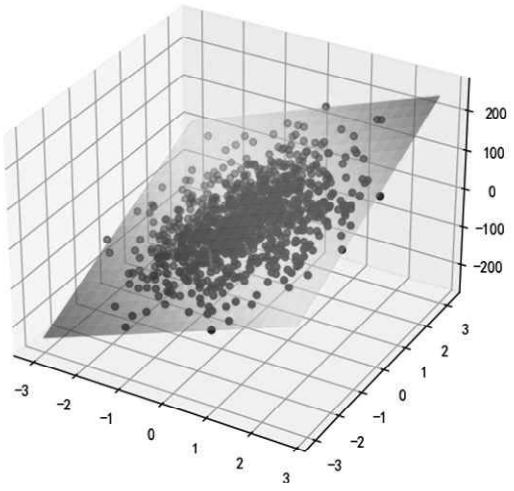


图 2.2　二元线性回归模型的拟合结果

2.1.3 多项式回归

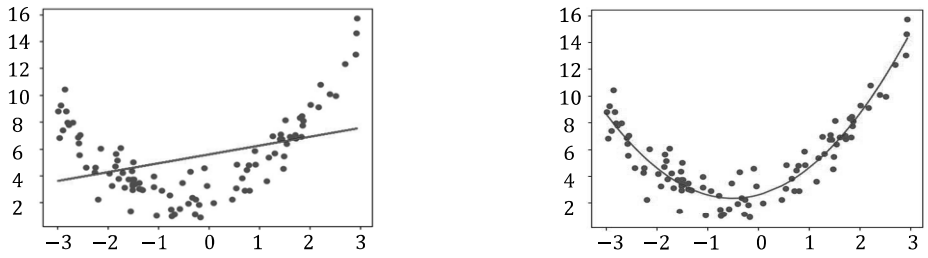
对于许多复杂的实际问题，仅仅使用简单的线性回归模型无法很好地对真实数据进行拟合，因此在本节我们介绍一种多项式回归模型，以处理这类复杂问题。

多项式回归(polynomial regression)的目标是探究一个因变量与相关自变量的多项式之间的线性关系。当多项式次数(多项式中次数最高项的次数)为 1 时，多项式回归退化为普通多元线性回归。当多项式次数为 2 时，针对单一样本，多项式回归模型可以表示为

$$y = \boldsymbol{w}^{\top}\boldsymbol{x} + \varepsilon, \tag{2.6}$$

其中，y 是因变量，$\boldsymbol{x} = \left[1, x_1, x_2, x_1x_2, x_1^2, x_2^2\right]^{\top}$ 是自变量，$\boldsymbol{w} = \left[w_0, w_1, w_2, w_3, w_4, w_5\right]^{\top}$ 表示模型的回归系数。从式可以看出，多项式回归的本质是多元线性回归。

图 2.3 给出了在同一训练集中，普通线性回归模型和多项式回归模型的拟合结果。由图 2.3 可以看出，相较于普通线性回归模型，多项式回归模型往往能够更准确地拟合因变量与自变量之间的非线性关系。因此相较于普通线性回归模型，多项式回归模型的拟合能力更强。



(a) 普通线性回归模型　　(b) 多项式回归模型

图 2.3　普通线性回归模型和多项式回归模型对相同训练集拟合结果的对比

在多项式回归模型中，多项式的次数是非常重要的参数。如图 2.4 所示，当次数较低时，多项式回归模型往往不能很好地拟合数据；反之，当次数较高时，多项式回归模型会对训练数据过度拟合，这样就使模型缺少泛化能力。

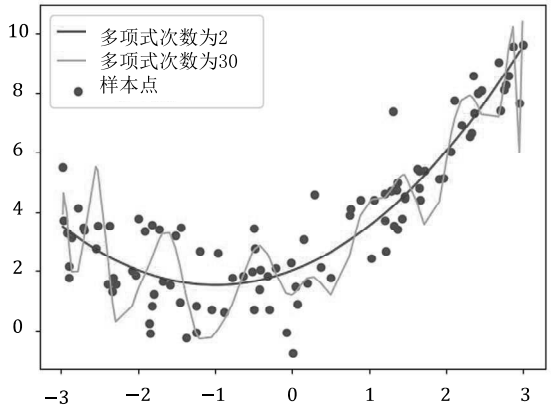


图 2.4　多项式次数过高导致模型过拟合

2.2 参数估计模型

在建立模型时，模型参数的设定十分重要，参数设定的优劣必然影响模型的拟合效果。本节介绍几种常用的参数估计方法，包括最小二乘估计、岭回归、套索回归和弹性回归。

2.2.1 最小二乘估计

最小二乘估计是一种常用的参数估计方法。该方法通过最小化模型输出结果与真实值的误差平方和来寻求模型的最佳参数。一般情况下，对于 m 个样本，每个样本有 n 个特征值，为常数项 w_0 增加自变量系数 $x_0^{(i)}=1\left(\forall i\in\left[1,2,\cdots,m\right]\right)$，给定模型的输入 $\boldsymbol{X}\in\mathbb{R}^{m\times(n+1)}$ 和实际值 $\boldsymbol{y}=\left[y^{(1)},y^{(2)},\cdots,y^{(m)}\right]^{\top}$，则模型的输出 $\hat{\boldsymbol{y}}$ 与输入 $\boldsymbol{X}$ 的关系如下

$$\hat{\boldsymbol{y}}=\boldsymbol{X}\boldsymbol{w}, \tag{2.7}$$

其中，$\hat{\boldsymbol{y}}=[\hat{y}^{(1)},\hat{y}^{(2)},\cdots,\hat{y}^{(m)}]^{\top}$ 表示模型的输出值(也称预测值)，$\boldsymbol{w}=[w_0,w_1,\cdots,w_n]^{\top}$ 为模型的回归参数。

最小二乘估计的目的是找到最优的参数 $\boldsymbol{w}$ 使模型的拟合效果最好。最小二乘估计通常采用实际值和预测值的误差的平方和来评估拟合效果，误差平方和越小，说明二者差距越小，模型拟合效果越好。误差平方和 $Q(\boldsymbol{w})$的计算公式如下

$$Q\left(\boldsymbol{w}\right)=\sum_{i=1}^{m}\left(y^{(i)}-\hat{y}^{(i)}\right)^2. \tag{2.8}$$

$Q(\boldsymbol{w})$最小时，模型的拟合效果达到最好，此时对应的回归参数 $\boldsymbol{w}$ 即为所求的最优参数。平方又叫二乘方，所以这种方法被称为最小二乘估计。

下面我们以一元线性回归模型为例来说明最小二乘估计的过程。设数据集的样本数为 m，对于其中的第 i 个样本 $(\forall i\in[1,2,\cdots,m])$，模型的输出 $\hat{y}^{(i)}$ 与特征值 $x^{(i)}$ 的线性关系为

$$\hat{y}^{(i)}=w_0+w_1x^{(i)}. \tag{2.9}$$

为了通过最小二乘法求出模型参数的最优解，我们先将一元线性回归方程(2.9)代入式(2.8)中，得到

$$Q\left(\boldsymbol{w}\right)=\sum_{i=1}^{m}\left(y^{(i)}-\hat{y}^{(i)}\right)^2=\sum_{i=1}^{m}\left(y^{(i)}-w_0-w_1x^{(i)}\right)^2. \tag{2.10}$$

$Q(\boldsymbol{w})$取最小值时的参数 $\boldsymbol{w}$ 对应最优参数，所以我们可以将求解参数最优值的问题转换为 $Q(\boldsymbol{w})$的最小化问题。依据凸优化相关理论，我们可以通过计算 $Q(\boldsymbol{w})$对参数 w_0、w_1 的偏导数，并令其偏导数为 0 来解决该问题，具体操作如下

$$\frac{\partial Q}{\partial w_0}=2(-1)\sum_{i=1}^{m}\left(y^{(i)}-w_0-w_1x^{(i)}\right)=0. \tag{2.11}$$

$$\frac{\partial Q}{\partial w_1}=2(-1)\sum_{i=1}^{m}\left(y^{(i)}-w_0-w_1x^{(i)}\right)x^{(i)}=0. \tag{2.12}$$

令 $\overline{x}=\frac{1}{m}\sum_{i=1}^{m}x^{(i)},\overline{y}=\frac{1}{m}\sum_{i=1}^{m}y^{(i)}$，可解得

$$w_1=\frac{\sum_{i=1}^{m}\left(x^{(i)}-\overline{x}\right)\left(y^{(i)}-\overline{y}\right)}{\sum_{i=1}^{m}\left(x^{(i)}-\overline{x}\right)^2}, \tag{2.13}$$

$$w_0=\overline{y}-w_1\overline{x}=\overline{y}-\frac{\sum_{i=1}^{m}\left(x^{(i)}-\overline{x}\right)\left(y^{(i)}-\overline{y}\right)}{\sum_{i=1}^{m}\left(x^{(i)}-\overline{x}\right)^2}\overline{x}. \tag{2.14}$$

对于一般的多元线性回归模型，我们可以通过矩阵运算来表示上述求解过程。由式可知，对于包含 m 个 n 维样本的数据集，其多元线性回归方程的矩阵表示为

$$\hat{\boldsymbol{y}}=\boldsymbol{X}\boldsymbol{w}.$$

则相应的 $Q(\boldsymbol{w})$可以表示为

$$Q(\boldsymbol{w})=\left\|\boldsymbol{y}-\hat{\boldsymbol{y}}\right\|_2^2=\left\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\right\|_2^2=(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\top}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}). \tag{2.15}$$

和一元线性回归模型的求解一样，我们把问题转换为 $Q(\boldsymbol{w})$的最小化问题。若输入 $\boldsymbol{X}$ 为列满秩矩阵，此时 $\boldsymbol{w}$ 的求解过程如下。

(1) 展开 $Q(\boldsymbol{w})$。

$$Q(\boldsymbol{w})=(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\top}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})=\boldsymbol{y}^{\top}\boldsymbol{y}-\boldsymbol{w}^{\top}\boldsymbol{X}^{\top}\boldsymbol{y}-\boldsymbol{y}^{\top}\boldsymbol{X}\boldsymbol{w}+\boldsymbol{w}^{\top}\boldsymbol{X}^{\top}\boldsymbol{X}\boldsymbol{w}. \tag{2.16}$$

(2) 化简 $Q(\boldsymbol{w})$。式中，$\boldsymbol{w}^{\top}\boldsymbol{X}^{\top}\boldsymbol{y}$ 和 $\boldsymbol{y}^{\top}\boldsymbol{X}\boldsymbol{w}$ 互为转置，且两者均为标量，因此，$\boldsymbol{w}^{\top}\boldsymbol{X}^{\top}\boldsymbol{y}=\boldsymbol{y}^{\top}\boldsymbol{X}\boldsymbol{w}$。所以

$$Q(\boldsymbol{w})=\boldsymbol{y}^{\top}\boldsymbol{y}-2\boldsymbol{w}^{\top}\boldsymbol{X}^{\top}\boldsymbol{y}+\boldsymbol{w}^{\top}\boldsymbol{X}^{\top}\boldsymbol{X}\boldsymbol{w}. \tag{2.17}$$

(3) 计算 $Q(\boldsymbol{w})$关于向量 $\boldsymbol{w}$ 的梯度，并令其为零，这样有

$$\nabla Q(\boldsymbol{w})=\begin{bmatrix}\frac{\partial Q}{\partial w_0}\\ \frac{\partial Q}{\partial w_1}\\ \vdots\\ \frac{\partial Q}{\partial w_n}\end{bmatrix}=2\boldsymbol{X}^{\top}\boldsymbol{X}\boldsymbol{w}-2\boldsymbol{X}^{\top}\boldsymbol{y}=\boldsymbol{0}, \tag{2.18}$$

其中，∇ 称为向量微分算子或 Nabla 算子，$\nabla Q(\boldsymbol{w})$ 表示计算 $Q(\boldsymbol{w})$关于模型参数 $\boldsymbol{w}$ 的梯度。

由于 $\boldsymbol{X}$ 为满秩矩阵，所以可以推导出

$$\boldsymbol{w}=\left(\boldsymbol{X}^{\top}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\top}\boldsymbol{y}. \tag{2.19}$$

2.2.2 岭回归

上文所介绍的最小二乘估计是一种常用的模型参数估计方法，它并不适用于所有情况，它要求模型输入的 $\boldsymbol{X}$ 为列满秩矩阵。遗憾的是，在现实生活中，模型的输入之间经常存在近似线性的关系，此时的矩阵 $\boldsymbol{X}$ 不满足列满秩条件，导致 $\boldsymbol{X}^{\top}\boldsymbol{X}$ 逆矩阵的求解十分困难，进而影响了模型参数的估计。为了解决这种缺陷，研究人员提出了岭回归模型，即在最小二乘估计优化函数的基础上添加 L_2 范数正则项，以确保逆矩阵的存在。岭回归的优化函数如下

$$Q(\boldsymbol{w})=\left\|\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y}\right\|_2^2+\alpha\left\|\boldsymbol{w}\right\|_2^2, \tag{2.20}$$

其中，$\left\|\boldsymbol{w}\right\|_2^2=\sum_{i=1}^{n+1}w_i^2$，$\alpha\left(\alpha>0\right)$ 为平衡参数，用来调节目标方程中正则项与均方误差之间的比重。岭回归在最小二乘估计的基础上加入了一个平方偏差因子来调节回归系数 $w_j\left(\forall j\in\left[0,1,\cdots,n\right]\right)$。若系数 w_j 较大，模型就会做出惩罚。

下面我们来学习岭回归的参数求解过程。同求解最小二乘估计模型一样，我们将求解参数 $\boldsymbol{w}$ 的问题转换为 $Q(\boldsymbol{w})$的最小化问题。具体的操作流程如下。

(1) 展开并化简 $Q(\boldsymbol{w})$。有

$$Q(\boldsymbol{w})=(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\top}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})+\alpha\left\|\boldsymbol{w}\right\|_2^2=\boldsymbol{y}^{\top}\boldsymbol{y}-2\boldsymbol{w}^{\top}\boldsymbol{X}^{\top}\boldsymbol{y}+\boldsymbol{w}^{\top}\boldsymbol{X}^{\top}\boldsymbol{X}\boldsymbol{w}+\alpha\boldsymbol{w}^{\top}\boldsymbol{w}. \tag{2.21}$$

(2) 计算 $Q(\boldsymbol{w})$关于向量 $\boldsymbol{w}$ 的梯度，并令其为零，则有

$$\nabla Q(\boldsymbol{w})=\begin{bmatrix}\dfrac{\partial Q(\boldsymbol{w})}{\partial w_0}\\ \dfrac{\partial Q(\boldsymbol{w})}{\partial w_1}\\ \vdots\\ \dfrac{\partial Q(\boldsymbol{w})}{\partial w_n}\end{bmatrix}=2\boldsymbol{X}^{\top}\boldsymbol{X}\boldsymbol{w}-2\boldsymbol{X}^{\top}\boldsymbol{y}+2\alpha\boldsymbol{w}=\boldsymbol{0}. \tag{2.22}$$

解得

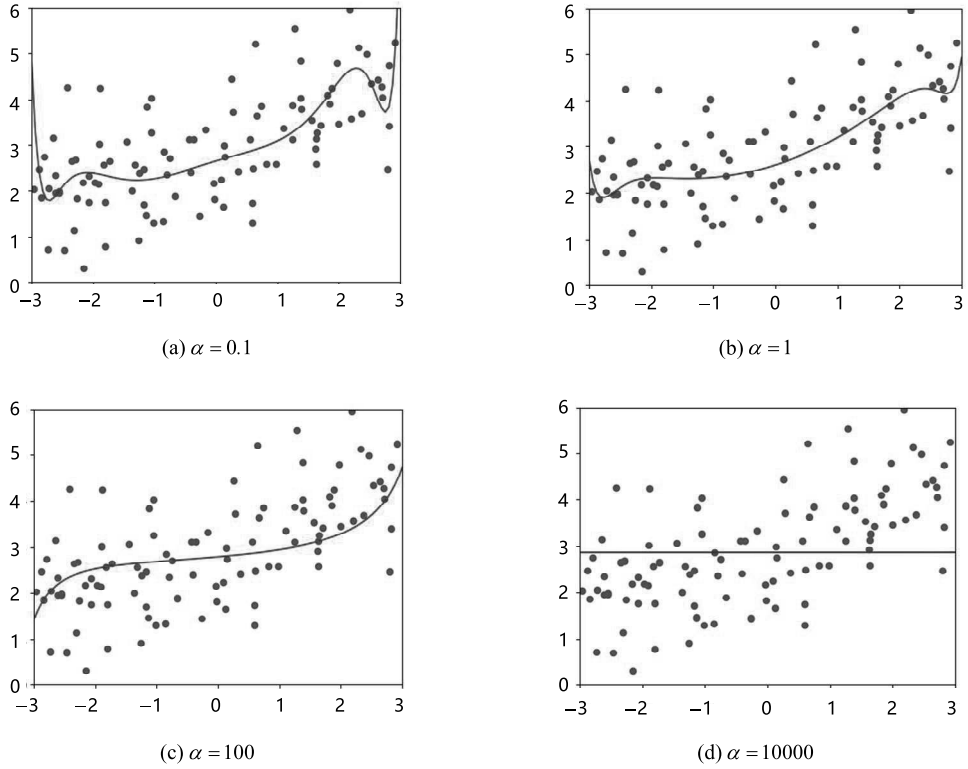
$$\boldsymbol{w}=\left(\boldsymbol{X}^{\top}\boldsymbol{X}+\alpha\boldsymbol{I}\right)^{-1}\boldsymbol{X}^{\top}\boldsymbol{y}, \tag{2.23}$$

其中，$\boldsymbol{I}\in\mathbb{R}^{(n+1)\times(n+1)}$ 是单位矩阵。

当 $\boldsymbol{X}$ 不满足列满秩条件时，意味着 $\boldsymbol{X}$ 的列向量之间存在线性相关组，此时 $\boldsymbol{X}^{\top}\boldsymbol{X}$ 无法求逆，这种现象被称为多重共线性(multi-collinearity)。多重共线性使模型的参数 $\boldsymbol{w}$ 无法通过最小二乘估计法求解。而岭回归在 $\boldsymbol{X}^{\top}\boldsymbol{X}$ 的基础上增加了 $\alpha\boldsymbol{I}$，这在一定程度上破坏了

多重共线性。我们可以对 $\boldsymbol{X}^{\top}\boldsymbol{X}+\alpha\boldsymbol{I}$ 进行求逆操作，进而完成相应的参数估计。

岭回归实际上是在最小二乘估计的基础上增加了正则项，正则项的引入能够防止模型过拟合，为了观察平衡参数 α 对拟合效果和系数 w_j 的影响，我们在图 2.4 的过拟合模型中增加了 L_2 正则项，从而得到了岭回归模型，并设置了不同的平衡参数 α，在不同 α 值的情况下模型的拟合效果如图 2.5 所示。



(a) $\alpha=0.1$　　(b) $\alpha=1$

(c) $\alpha=100$　　(d) $\alpha=10000$

图 2.5　对相同训练集不同 α 值的岭回归的拟合效果对比

该模型的回归系数 $w_j\left(\forall j\in[0,1,\cdots,10]\right)$ 在不同平衡参数 α 下的结果变化如图 2.6 所示。结合图 2.5 和图 2.6 可以看出，当 α 较小时，正则项的作用几乎为零，此时模型仍然过拟合；当 α 的值非常大时，均方误差函数失效，$\boldsymbol{w}$ 的元素都趋近于 0。显然，α 的设置影响正则项的作用力度与模型拟合效果。因此，平衡参数 α 的设置对于模型的拟合效果的提升至关重要。

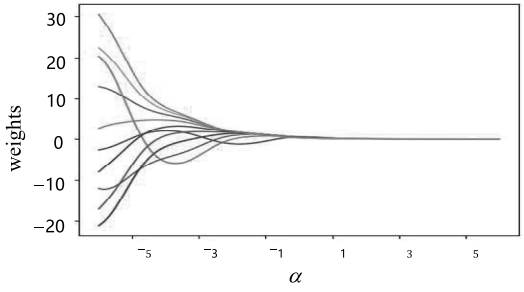


图 2.6　不同 α 下岭回归回归系数的变化

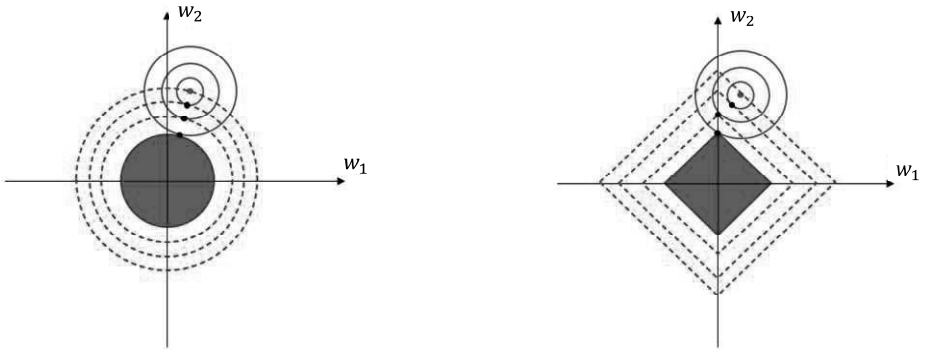
2.2.3 套索回归

岭回归使用 L_2 范数的平方作为正则项来压缩回归系数。实践表明这种方法在一定程度上可以提高模型的鲁棒性和稳定性。若使用 L_1 范数替代 L_2 范数的平方作为正则项，我们就得到了套索(lasso)回归模型，其优化函数形式如下

$$\min_{\boldsymbol{w}} \|\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y}\|_2^2+\alpha\|\boldsymbol{w}\|_1, \tag{2.24}$$

其中，$\alpha(\alpha>0)$ 为平衡参数，用来调节目标方程中正则项与均方误差之间的比例。和岭回归相比，套索回归的 L_1 范数正则项能够对 $\boldsymbol{w}$ 进行稀疏选择，在 $\boldsymbol{w}$ 中绝对值较小的元素可能会被置为零，即训练好的套索回归模型会完全忽略某些特征对输出的影响。

我们可以借助图 2.7 来了解岭回归和套索回归的区别。

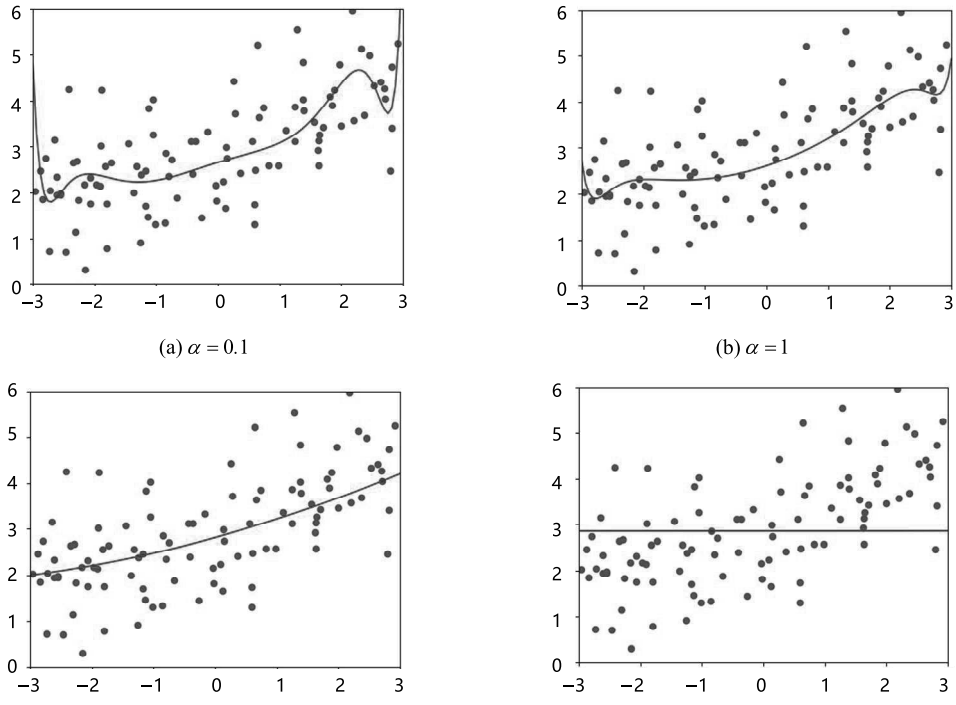


(a) 岭回归　　(b) 套索回归

图 2.7　岭回归和套索回归正则项对标准损失函数的惩罚过程

在二维特征空间中，我们使用圆形和方形区域分别表示岭回归和套索回归的正则项，同心圆轮廓表示回归模型的标准损失函数。岭回归和套索回归都是通过寻找同心圆轮廓与正则项区域相交的第一个点来确定回归系数的。当平衡参数 α 减小时，正则项对标准损失函数的惩罚强度逐渐减弱，正则项区域逐渐变小。与岭回归的圆形正则项区域不同，套索回归的菱形正则项区域与同心圆轮廓相交的第一个点在轴上(图 2.7 中为 y 轴)的概率更大，当同心圆区域与套索回归的第一个交点在轴上时，模型就会忽略另一个特征维度(模型将该特征所对应的回归参数设置为零)。所以套索回归不仅有助于减少过拟合，还可以帮助我们进行特征选择，忽略对当前任务不重要的特征。

采用套索回归对上文的数据进行拟合，结果如图 2.8 所示。



(a) $\alpha = 0.1$　(b) $\alpha = 1$

(c) $\alpha = 100$　(d) $\alpha = 10\,000$

图 2.8　对相同训练集不同 α 值的套索回归的拟合效果对比

2.2.4　弹性回归

弹性回归[12]是对套索回归的改进。当数据中有一组高度相关的变量时，套索回归倾向于从这组变量中选择一个变量而忽略其他变量，因此对于特征向量互相关的数据，套索回归的表现很不稳定。鉴于岭回归在面对特征向量互相关的数据时所表现出的稳定性，研究者发现可以通过结合岭回归和套索回归的正则项来解决上述问题，这样的模型称为弹性回归模型。弹性回归模型通过平衡参数 α 和 ρ 在 L_1 正则项和 L_2 正则项中进行权衡，通过控制二者的惩罚力度来间接控制模型的稳定性，其优化函数如下

$$\min_{\boldsymbol{w}} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \alpha\rho\|\boldsymbol{w}\|_1 + \frac{\alpha(1-\rho)}{2}\|\boldsymbol{w}\|_2^2, \tag{2.25}$$

其中，$\alpha(\alpha > 0)$ 和 $\rho(\rho > 0)$ 为正则化参数，用于调节 L_1 正则化和 L_2 正则化的惩罚强度。我们可以使用弹性回归对上文中的数据进行拟合，从观察 α 和 ρ 对模型拟合效果的影响。

当固定 $\rho = 0.2$ 时，设置不同的 α 以观察模型拟合效果的变化，结果如图 2.9 所示。

当固定 $\alpha = 10$ 时，设置不同的 ρ 以观察模型拟合效果的变化，结果如图 2.10 所示。

当存在多个相互关联的特征时，套索回归可能会随机选择其中之一，而弹性回归往往会将多个相互关联的重要特征同时保留在模型中。弹性回归在岭回归和套索回归之间权衡，这使得模型在不失套索回归优点的同时能够继承岭回归的稳定性。通过图 2.11 可以对比套索回归和弹性回归的回归系数 $\boldsymbol{w}$ 随 α 的变化情况。其中实线部分代表索套回归的回

归系数 $\boldsymbol{w}$，虚线部分代表弹性回归的回归系数 $\boldsymbol{w}$，不同线条代表不同的回归系数 $w_j\left(\forall j \in [0,1,\cdots,10]\right)$。

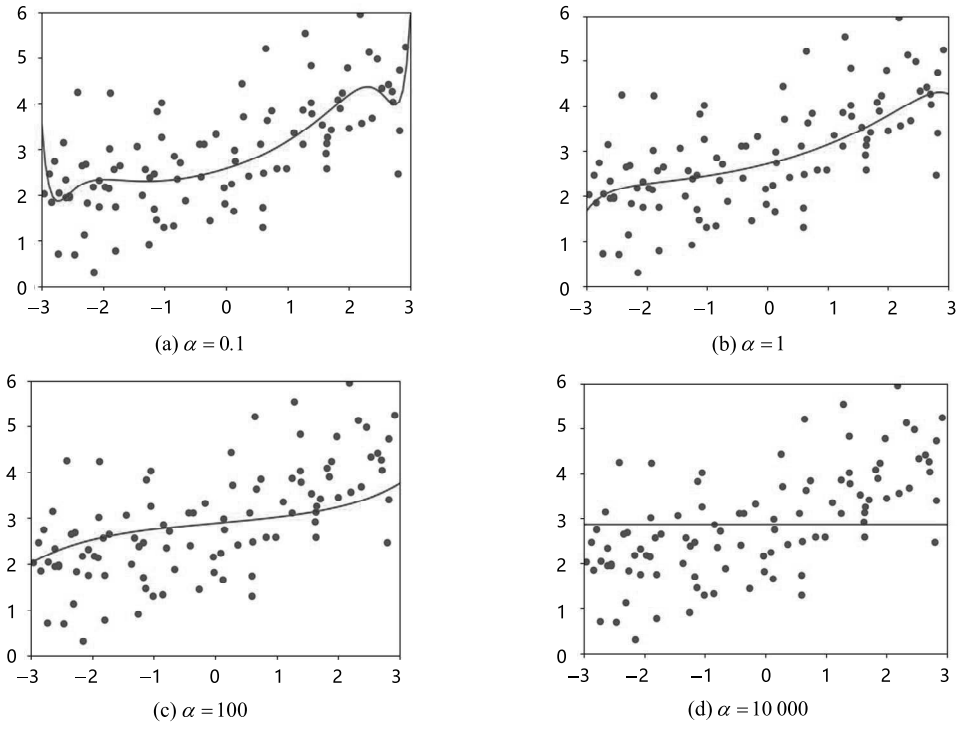


图 2.9　对相同训练集不同 α 值的弹性回归的拟合效果对比

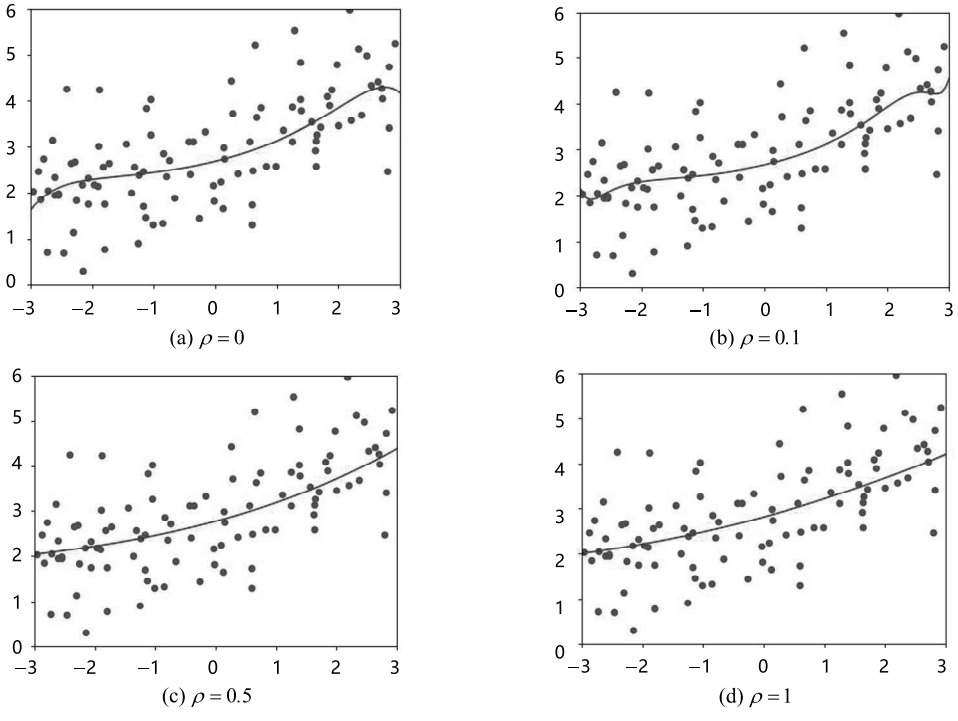


图 2.10　对相同训练集不同 ρ 值的弹性回归的拟合效果对比

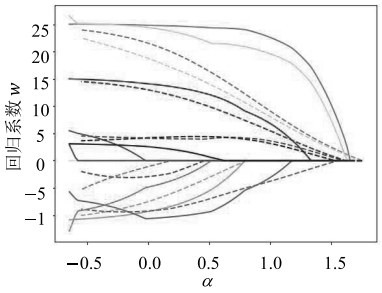


图 2.11　套索回归和弹性回归的回归系数 $\boldsymbol{w}$ 随 α 的变化情况

2.3 梯度下降算法

数值优化算法对于模型的求解非常重要，而梯度下降算法是一种非常重要的数值优化方法。本节逐一介绍梯度的概念、梯度下降算法的具体求解过程和梯度下降算法的一般分类方式。

2.3.1 梯度的概念

梯度的概念建立在导数、偏导数和方向导数的概念的基础之上。下面我们将依次介绍导数、偏导数和方向导数，最后引出梯度的概念。

定义 2.2　(导数)

若函数 $y=f\left(x\right)$ 在点 x^0 的邻域内有定义，则当自变量 x 在 x^0 处取得增量 Δx (点 $x^0+\Delta x$ 仍然在该邻域内)时，相应地，y 取得增量 $\Delta y=f\left(x^0+\Delta x\right)-f\left(x^0\right)$，如果 Δy 与 Δx 在 $\Delta x\to 0$ 时的极限存在，即有

$$\lim_{\Delta x\to 0}\frac{\Delta y}{\Delta x}=\lim_{\Delta x\to 0}\frac{f\left(x^0+\Delta x\right)-f\left(x^0\right)}{\Delta x}, \tag{2.26}$$

则称 $y=f\left(x\right)$ 在 x^0 处可导，上述极限就是 $y=f\left(x\right)$ 在 x^0 处的导数，记为 $f'\left(x^0\right)$ 或 $\left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{x=x^0}$。

对于一元函数而言，$y=f\left(x\right)$ 在 $x=x^0$ 处的导数的几何意义，就是曲线 $y=f\left(x\right)$ 在点 $P\left(x^0,f\left(x^0\right)\right)$ 处的切线的斜率值 k，即 $k=f'\left(x^0\right)$。从物理角度来看，路程对于时间的导数为速度，速度对于时间的导数为加速度。

对于多元函数来说，其几何图形为一个曲面，曲面上一点的切线有无数条，那么取哪一条切线的斜率值作为该点的导数呢？这种情况下，导数便不能再作为切线斜率来解释，由此引入偏导数的概念。

定义 2.3 （偏导数）

若多元函数 $y=f\left(x_1,x_2,\cdots,x_n\right)$ 在点 $P\left(x_1^0,x_2^0,\cdots,x_n^0\right)$ 的邻域内有定义，固定 $x_i=x_i^0\left(\forall i\in[2,3,\cdots,n]\right),y=f\left(x_1,x_2^0,\cdots,x_n^0\right)$ 可以看作关于 x_1 的一元函数，若该一元函数在 $x_1=x_1^0$ 处可导，即有

$$\lim_{\Delta x_1\to 0}\frac{f\left(x_1^0+\Delta x_1,x_2^0,\cdots x_n^0\right)-f\left(x_1^0,x_2^0,\cdots,x_n^0\right)}{\Delta x_1}=A. \tag{2.27}$$

若函数的极限 A 存在，则称 A 为函数 $y=f\left(x_1,x_2,\cdots,x_n\right)$ 在点 $P\left(x_1^0,x_2^0,\cdots,x_n^0\right)$ 处关于自变量 x_1 的偏导数，记作 $f_{x_1}\left(x_1^0,x_2^0,\cdots,x_n^0\right)$ 或 $\left.\frac{\partial f}{\partial x_1}\right|_{\left(x_1^0,x_2^0,\cdots,x_n^0\right)}$。

对于二元函数 $z=f(x,y)$ 来说，它在点 $P(x^0,y^0)$ 关于自变量 x 的偏导数 $f_x(x^0,y^0)$ 的几何意义为曲面 $z=f(x,y)$ 与平面 $y=y^0$ 的交线在点 x^0 处的导数，也就是交线在点 x^0 处的切线关于 x 轴的斜率。同理，$z=f(x,y)$ 在点 $P(x^0,y^0)$ 关于自变量 y 的偏导数 $f_y(x^0,y^0)$ 的几何意义为曲面 $z=f(x,y)$ 与平面 $x=x^0$ 的交线在点 y^0 处的切线关于 y 轴的斜率。

偏导数的物理意义表示函数在某一点处沿着某个坐标轴正方向上的变化率。例如，$f_x(x^0,y^0)$ 表示函数 $z=f(x,y)$ 在点 $P(x^0,y^0)$ 处沿着 x 轴正方向的变化率，$f_y(x^0,y^0)$ 表示函数 $z=f(x,y)$ 在点 $P(x^0,y^0)$ 处沿着 y 轴正方向的变化率。在解决实际问题时，我们不仅需要知道函数在固定点处沿着坐标轴正方向的变化率，还需要知道函数在该点沿着其他特定方向的变化率。例如，在气象学中，热空气通常向温度较低的地方流动，这就需要确定大气温度、气压沿着某些方向的变化率，因此我们需要讨论函数在固定点处沿任意指定方向的变化率的问题。由此引入了方向导数的概念，它可以用来衡量曲面上在某一点处沿着任意方向的切线斜率。

定义 2.4 （方向导数）

设函数 $y=f\left(x_1,x_2,\cdots,x_n\right)$ 在点 $P\left(x_1^0,x_2^0,\cdots,x_n^0\right)$ 的邻域 $U(P)$ 内有定义，自该点引射线 l，并设 $P'\left(x_1^0+\Delta x_1,x_2^0+\Delta x_2,\cdots,x_n^0+\Delta x_n\right)$ 为 l 上的另一点且 $P'\in U(P)$。考虑函数的增量 $f\left(x_1^0+\Delta x_1,x_2^0+\Delta x_2,\cdots,x_n^0+\Delta x_n\right)-f\left(x_1^0,x_2^0,\cdots,x_n^0\right)$ 与 P、P' 两点之间的距离为 $\rho=\sqrt{\left(\Delta x_1\right)^2+\left(\Delta x_2\right)^2+\cdots+\left(\Delta x_n\right)^2}$。当点 P' 沿着 l 趋于点 P 时，即有

$$\lim_{\rho\to 0}\frac{f\left(x_1^0+\Delta x_1,x_2^0+\Delta x_2,\cdots,x_n^0+\Delta x_n\right)-f\left(x_1^0,x_2^0,\cdots,x_n^0\right)}{\rho}=B. \tag{2.28}$$

若极限 B 存在，则称 B 为函数 $y=f\left(x_1,x_2,\cdots,x_n\right)$ 在点 $P\left(x_1^0,x_2^0,\cdots,x_n^0\right)$ 处沿方向 l 的方向导数，记作 $\left.\frac{\partial f}{\partial l}\right|_{\left(x_1^0,x_2^0,\cdots,x_n^0\right)}$。

某二元函数 $z=f(x,y)$ 在点 $P(x^0,y^0)$ 处沿 l 方向的方向导数如图 2.12 所示。

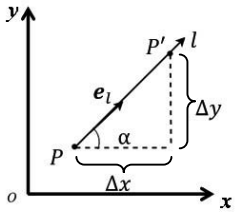


图 2.12　方向导数

假设 x 轴正方向到射线 l 的夹角为 α，即有与射线 l 同方向的单位向量 $\boldsymbol{e}_l=[\cos\alpha,\sin\alpha]^{\top}$，可以看到 $\Delta x=\rho\cos\alpha,\Delta y=\rho\sin\alpha$。简单来说，方向导数就是从点 P 沿着 l 方向移动 ρ 单位长度到点 P'，当自变量 x、y 从点 P 到点 P' 时，因变量 z 也会发生 Δz 的变化，即

$$\Delta z=f\left(x^0+\rho\cos\alpha,y^0+\rho\sin\alpha\right)-f\left(x^0,y^0\right). \tag{2.29}$$

当移动长度 ρ 趋于 0 时，Δz 与 ρ 的比值即为函数 $z=f\left(x,y\right)$ 在点 $P\left(x^0,y^0\right)$ 处沿方向 l 的方向导数。关于二元函数方向导数的存在及计算，我们有以下定理。

定理 2.1　(定理)

若函数 $z=f(x,y)$ 在点 $P(x^0,y^0)$ 处可微，那么函数在该点沿任一方向的方向导数都存在，且有

$$\left.\frac{\partial f}{\partial l}\right|_{\left(x^0,y^0\right)}=\frac{\partial f}{\partial x}\cos\alpha+\frac{\partial f}{\partial y}\sin\alpha, \tag{2.30}$$

其中，α 为 x 轴正方向到射线 l 的转角。

证明：

根据函数 $z=f(x,y)$ 在点 $P(x^0,y^0)$ 处可微的假定，我们可以得到

$$\begin{aligned}\Delta z&=f\left(x^0+\Delta x,y^0+\Delta y\right)-f\left(x^0,y^0\right)\\&=\frac{\partial f}{\partial x}\Delta x+\frac{\partial f}{\partial y}\Delta y+o\left(\rho\right).\end{aligned} \tag{2.31}$$

等式两边都除以 ρ，有

$$\begin{aligned}\frac{f\left(x^0+\Delta x,y^0+\Delta y\right)-f\left(x^0,y^0\right)}{\rho}&=\frac{\partial f}{\partial x}\frac{\Delta x}{\rho}+\frac{\partial f}{\partial y}\frac{\Delta y}{\rho}+\frac{o\left(\rho\right)}{\rho}\\&=\frac{\partial f}{\partial x}\cos\alpha+\frac{\partial f}{\partial y}\sin\alpha+\frac{o\left(\rho\right)}{\rho}.\end{aligned} \tag{2.32}$$

等式两边令 ρ 趋于 0，可以得到

$$\lim_{\rho\to 0}\frac{f\left(x^0+\Delta x,y^0+\Delta y\right)-f\left(x^0,y^0\right)}{\rho}=\frac{\partial f}{\partial x}\cos\alpha+\frac{\partial f}{\partial y}\sin\alpha. \tag{2.33}$$

由此证明了方向导数存在且值为

$$\left.\frac{\partial f}{\partial l}\right|_{\left(x^0,y^0\right)}=\frac{\partial f}{\partial x}\cos\alpha+\frac{\partial f}{\partial y}\sin\alpha.$$

从定理 2.1 的证明过程中我们还可以推导得到

$$f_x\left(x^0,y^0\right)\cos\alpha+f_y\left(x^0,y^0\right)\sin\alpha=\left[f_x\left(x^0,y^0\right),f_y\left(x^0,y^0\right)\right]\left[\cos\alpha,\sin\alpha\right]^{\top}. \tag{2.34}$$

该结果说明，对于一个可微函数，在某点的任意方向的方向导数为函数在该点处偏导数的线性组合，其中系数为所选定方向的单位向量。当所求方向导数的方向与坐标轴正方向一致时，方向导数即为偏导数。

上述推断对一般多元函数依然成立。例如，若函数 $y=f\left(x_1,x_2,\cdots,x_n\right)$ 在点 $P\left(x_1^0,x_2^0,\cdots,x_n^0\right)$ 处的偏导数存在，则该函数在点 P 处沿着单位向量 $\boldsymbol{e}_1=\left[1,0,0,\cdots,0\right]^{\top}$ 的方向导数为

$$\left.\frac{\partial f}{\partial l}\right|_{\left(x_1^0,x_2^0,\cdots,x_n^0\right)}=\left[\frac{\partial f}{\partial x_1},\frac{\partial f}{\partial x_2},\cdots,\frac{\partial f}{\partial x_n}\right]\left[1,0,0,\cdots,0\right]^{\top}=\frac{\partial f}{\partial x_1}. \tag{2.35}$$

方向导数的物理意义是函数在某一点处沿某特定方向上的变化率，是一个具体的值。很显然，一个点不止有一个方向，每个方向都有其对应的方向导数，那么在求解优化问题时所需的模型参数的更新方向是什么特定方向呢?由此就引出梯度[14]的概念。

定义 2.5 （梯度）

设函数 $y=f\left(x_1,x_2,\cdots,x_n\right)$ 在平面区域 D 内具有一阶连续偏导数，定义单位矩阵 $\boldsymbol{E}\in\mathbb{R}^{n\times n},\boldsymbol{E}=\left[\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_n\right]^{\top}$，则对于每一点 $P\left(x_1^0,x_2^0,\cdots,x_n^0\right)\in D$，都可定出一个向量

$$\frac{\partial f}{\partial x_1}\boldsymbol{e}_1+\frac{\partial f}{\partial x_2}\boldsymbol{e}_2+\cdots+\frac{\partial f}{\partial x_n}\boldsymbol{e}_n,$$

该向量称为函数 $y=f\left(x_1,x_2,\cdots,x_n\right)$ 在点 $P\left(x_1^0,x_2^0,\cdots,x_n^0\right)$ 处的梯度，记作 $\nabla f\left(x_1^0,x_2^0,\cdots,x_n^0\right)$。

以二元函数 $z=f\left(x,y\right)$ 为例，其在点 $\left(x^0,y^0\right)$ 处的梯度记作

$$\nabla f\left(x^0,y^0\right)=\frac{\partial f}{\partial x}\boldsymbol{i}+\frac{\partial f}{\partial y}\boldsymbol{j}, \tag{2.36}$$

其中，$\boldsymbol{i}=\left[1,0\right]^{\top},\boldsymbol{j}=\left[0,1\right]^{\top}$。

若 ϕ 是 $\nabla f\left(x^0,y^0\right)$ 与自该点引出的任意射线 l 方向上的单位向量 $\boldsymbol{e}_l=\left(\cos\alpha,\sin\alpha\right)$ 之间的夹角，则由方向导数的计算公式，可得

$$\begin{aligned}\left.\frac{\partial f}{\partial l}\right|_{\left(x^0,y^0\right)}&=\left(\frac{\partial f}{\partial x},\frac{\partial f}{\partial y}\right)(\cos\alpha,\sin\alpha)\\&=\left|\left(\frac{\partial f}{\partial x},\frac{\partial f}{\partial y}\right)\right|\left|\boldsymbol{e}_l\right|\cos\phi\\&=\left|\nabla f\left(x^0,y^0\right)\right|\cos\phi.\end{aligned}\tag{2.37}$$

依据式(2.12)可知，函数在点 $P\left(x^0,y^0\right)$ 处的方向导数可以看作梯度在射线 l 上的投影。当射线 l 的方向与梯度的方向一致时，$\cos\phi=1,\left.\frac{\partial f}{\partial l}\right|_{\left(x^0,y^0\right)}$ 取得最大值，即函数在定点处沿梯度方向的方向导数最大。简而言之，梯度的方向是函数 $z=f\left(x,y\right)$ 增长最快的方向。反之，当射线 l 的方向与梯度的方向相反时，$\cos\phi=-1,\left.\frac{\partial f}{\partial l}\right|_{\left(x^0,y^0\right)}$ 取得最小值，即函数在定点处沿梯度反方向的方向导数最小，也就是说，梯度反方向是函数 $z=f\left(x,y\right)$ 下降最快的方向。由此我们可以得到以下结论。

(1) 梯度是一个向量，既有大小又有方向。

(2) 梯度的方向是函数在该定点的方向导数取得最大值的方向，即函数值上升最快的方向。

(3) 梯度的大小是函数在该定点的方向导数的最大值。

上述结论是基于二元函数推导得出的，但对于一般多元函数同样有效。由梯度的定义可知，梯度的大小即梯度的模为

$$\left|\nabla f\left(x_1^0,x_2^0,\cdots,x_n^0\right)\right|=\sqrt{\left(\frac{\partial f}{\partial x_1}\right)^2+\left(\frac{\partial f}{\partial x_2}\right)^2+\cdots+\left(\frac{\partial f}{\partial x_n}\right)^2}.\tag{2.38}$$

2.3.2 梯度下降法算法

从梯度的概念可以得知，函数在一点处的梯度方向是该点函数值上升最快的方向，梯度反方向是函数值下降最快的方向。在机器学习中，模型的优化目标常常是最小化某一个目标函数，即找到目标函数的最小值。因此，对于这类问题一个比较直观的求解思路就是求目标函数在当前位置的梯度，并沿着梯度的反方向去探寻模型的解。

假设有一个可微分的二元凸函数 $z=f\left(x,y\right)$，优化目标是找到该函数的最小值点。我们可以将这个函数想象为一个盆地，我们的目标是走到该盆地的最低点。显然我们可以找到当前位置下降最快的方向，然后沿着该方向行走一段距离。到达新的位置后继续寻找对于新的位置下降最快的方向并沿该方向再走一段距离，重复上述过程最终便能走到盆地的最低点。

对应到函数中，我们需要求出给定点的梯度，然后沿着梯度的反方向更新函数参数，重复此过程直至模型稳定。若函数为凸函数，当目标函数的梯度为 0 时，参数停止更新，模型到达了函数的最小值点。而对于非凸函数，重复上述过程虽然无法保证能找到全局最优解，但一般情况下我们至少也可以找到某个局部最优解。

具体的过程如图 2.13 所示。假设从点 P_0 出发，计算出该点的梯度，沿着梯度的反方向移动一定距离到下一点 P_1。到达点 P_1 后，计算函数在该点的梯度，得到梯度的反方向为下一步的移动方向，并沿此方向再移动一定距离到下一点 P_2。重复此过程，直到满足停止条件。简单来说，梯度下降算法就是通过沿着目标函数梯度的反方向来不断更新模型参数，以探求目标函数最小值点的过程。

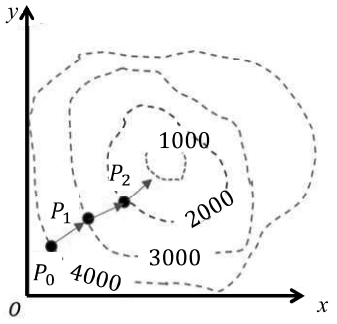


图 2.13　梯度下降原理模拟图

对于一个多元目标函数 $J\left(w_0,w_1,\cdots,w_n\right)$，其梯度为

$$\nabla J\left(w_0,w_1,\cdots,w_n\right)=\left(\frac{\partial J}{\partial w_0},\frac{\partial J}{\partial w_1},\cdots,\frac{\partial J}{\partial w_n}\right). \tag{2.39}$$

对于该函数，梯度下降算法的求解过程可以用如下公式表示：

$$\begin{aligned}\text{Repeat}\{\quad & w_0=w_0-\eta\frac{\partial J}{\partial w_0},\\ & w_1=w_1-\eta\frac{\partial J}{\partial w_1},\\ & \vdots\\ & w_n=w_n-\eta\frac{\partial J}{\partial w_n},\ \}\\ \text{Until for } \forall i\in[0,1,\cdots,n],\ & \frac{\partial J}{\partial w_i}\leqslant\epsilon,\end{aligned} \tag{2.40}$$

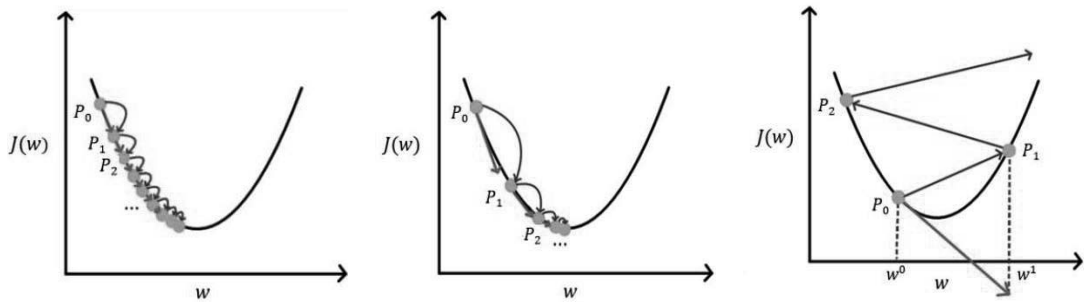
其中，$\eta>0$ 表示算法的迭代步长或学习率，ϵ 为算法停止更新的阈值参数。重复上述过程，不断地迭代计算当前位置目标函数的梯度，并沿着梯度的反方向进行参数更新，直至模型收敛、参数 $w_0,w_1,\cdots,w_n$ 不变则停止迭代。此时目标函数关于模型参数的梯度为零，算法达到局部最优解[15]。

在使用梯度下降算法时，为确保算法正常运行，我们需要重点考虑以下几个问题。

1. 学习率 η 的取值

学习率 η 可以理解为寻找盆地最低点场景中每一步行走的距离，其取值在梯度下降算法中至关重要。如果 η 太小，每次移动的距离就小，整个算法得到最优解的速度就会很慢；反之 η 过大，移动的距离太大，可能导致模型振荡不收敛，从而无法给出最优解。

假设目标函数 $J(w)$，对其进行梯度下降操作求最小值。图 2.14 展示了学习率 η 的取值对算法结果的影响。η 取值过小时，参数更新的程度较小，收敛过程缓慢，如图 2.14(a)所示，初始值点 P_0 经过第一次更新后到达点 P_1，再继续更新，需要经过多次更新才能到达 $J(w)$ 的最小值实现收敛。η 取值过大时，参数更新的程度过大，可能跳出可控制区域，造成损失值爆炸不收敛的情况，如图 2.14(c)所示，初始值点 $P_0\left(w^0, J\left(w^0\right)\right)$，经过一次更新到达点 $P_1\left(w^1, J\left(w^1\right)\right)$，相似地在后续更新过程中，由于步长过大导致数值不断增加，从而无法收敛。因此，选择合适的学习率[16]，如图 2.14(b)所示，不仅可以减少参数迭代的次数，而且能提高运行速度，加快模型的收敛。



(a) η 取值过小　(b) η 取值适中　(c) η 取值过大

图 2.14　学习率 η 取值情况

2. 参数初始值的选择

当目标函数是凸函数时，该函数梯度为零的点一定是最优解[17]。对于凸函数来说，不论初始值取到哪里，使用梯度下降算法并选定一个合适的步长，一定能找到最优解。但对于非凸函数来说，可能存在局部最小值和鞍点，设置的初始值不同，模型给出的解可能不同。因此，一般情况下，我们需要基于不同的初始值进行梯度下降操作，最后选择多次运算的结果中的最小值对应的解作为算法的输出值。

3. 梯度下降算法步骤

通过上文的介绍，我们已经对利用梯度下降算法求解优化目标的思想有了大概的了解。接下来我们将分别依据代数运算和矩阵运算的方式来描述梯度下降算法的具体步骤。

(1) 基于代数运算的梯度下降算法。

基于代数运算的梯度下降算法的具体步骤如下。