第 1 章

智能决策与复杂系统

内容提要

- 智能决策
- 人工智能
- 智能体
- 决策系统
- 金融复杂性
- 复杂系统
- 复杂科学
- 计算金融
- 大数据金融
- 社交金融
- 互联网金融
- 交易金融
- 科技金融
- 金融科技
- 新金融

1.1 智能决策

智能决策一直是人类关心的问题。运筹帷幄之中，决胜千里之外，智能决策是人类以及生物在演化过程中一直在学习和应用的技能。何为智能？为什么需要智能？在一个极其简单的环境下，个体能否展现出智能？因为环境的复杂性和不确定性，个体需要学习不同的智能策略来应对环境变化，从而获得生存机会。智能决策与复杂环境是分不开的，复杂环境会影响个体智能决策能力，而个体智能决策行为同样会影响环境复杂性，两者之间相互关联耦合并协同演化，共同构成了更加复杂和动态演化的复杂系统。如何在复杂环境中进行智能策略的学习，特别是在模拟的复杂环境下，如何训练智能体获得智能决策能力，是深度强化学习主要面对和需要解决的问题 [1-4]。

1.1.1 智能决策简介

2020 年 3 月新型冠状病毒感染在全球暴发，改变了世界运转模式以及人类日常生活，复杂社会经济系统受到了巨大冲击。在新型冠状病毒感染等外生冲击影响下，各行各业复工复产问题、供应链问题等直接影响了各经济体在全球产业链、价值链中的地位，应该如何防御和应对风险？这一问题已引起专家学者的较大关注，如汪寿阳教授团队研究了新型

冠状病毒感染对全球生产体系的冲击和中国产业链加速外移的风险 [5]。在疫情防控中，如何有效权衡经济效益和防疫效果，可以抽象为优化问题，即通过智能决策算法对经济运行和人类生活行为进行合理优化和决策。

智能决策的主体包括居民、社区、政府、经济体以及联合国，在不同的时间和空间尺度上都面对着大量的决策问题。对学生个体而言，选择食堂是一个决策问题；对旅行者而言，规划旅游路线是一个决策问题；对政府而言，疫情防控措施是一个决策问题；对经济体而言，选择贸易合作伙伴是一个决策问题；对国际组织而言，协调国际关系、化解国际冲突是决策问题。作为人类命运共同体的组成部分，不同尺度上、不同空间上、不同时间上的每一次决策，都会影响人类命运共同体的未来发展。

1.1.2 复杂金融系统中的智能决策

在复杂金融系统中，个人投资者、机构投资者和监管部门共同构成了一个动态演化的复杂巨系统。2008 年后，美国次贷危机引发的全球金融海啸促使科学家重新审视主流经济金融理论，提出了当前金融理论所面对的挑战。在极端金融事件的预警和预测方面，由于金融系统的非线性、动态性、随机性等复杂因素，如何有效地防范和预警风险，会直接影响全球金融经济系统的稳定和健康发展。基于智能算法进行风险预警和防控，具有重要意义和研究价值，也能给世界各个经济体的金融经济系统平稳运行提供一定保障。如何对动态演化市场环境进行动态监控，对市场环境状态进行建模分析，对系统性风险及其传染进行度量、识别、防控和预警，是深度强化学习能够有所作为的领域。

2008 年，Nature 杂志文章指出 [6]，传统理论无法预见当时的金融风险，需要在理论和方法上进行根本性的科学革命，新理论需要从实际数据出发来探寻市场规律，挖掘市场信息，从复杂市场结构中解构市场行为信息和个体行为规律。基于大数据的金融分析中，我们从海量高质量数据中挖掘市场的运行规律和多尺度特征，刻画和监控不同层次市场参与者的行为规律和演化特征。我们从微观到宏观、从个体到系统、从关联关系到因果关系、从理论到方法，进行多尺度、多层次、多角度的深度探索和挖掘，为金融经济系统的安全和稳定提供具有可操作性和实用性的研究方法和分析工具。深度强化学习方法融合了深度学习和强化学习，在智能识别和智能决策方面具有显著优势。深度学习模型适用于复杂经济金融系统中海量、多源、异构数据，强化学习模型适用于动态演化的复杂市场环境决策。

新的经济理论需要考虑异质经济人之间的相互作用 [7]，在此部分 ABM 模型和金融计算实验具有重要的应用价值。异质性个体之间的异质非线性相互作用构成了复杂性的来源，也使得复杂系统能够涌现更高层次的特征规律和功能表现，如市场对噪声的容错能力、对外在冲击的恢复能力等。如何构建异质性智能体之间的交互规则，使得系统能够更加鲁棒和稳定？我们可以将此问题建模成组合优化问题，融合深度强化学习进行智能决策和智能规划。在金融经济理论中存在着大量的序贯决策问题，深度强化学习是专门求解此类决策问题的智能学习方法。通过深入理解和学习深度强化学习，可以将一些看上去不是序贯决策问题且具有复杂实际应用背景的难题，建模成马尔可夫决策过程或者部分可观测的马尔可夫决策过程，随后运用深度强化学习算法进行训练和求解。

Schweitzer 等人指出经济学研究应该着眼于子系统之间的相互作用，以及由此而形成的复杂金融经济网络[8]。复杂金融经济网络是复杂金融经济系统的有效表示，能够比较高效地抽取和模型化复杂系统中个体之间的交互关系和结构特征[9,10]，其研究得到了大量科研人员关注[11,12]。深度强化学习系统需要对复杂巨系统和复杂系统中个体进行细致的表征，然后基于智能算法学习和度量特定问题的高层次特征，为智能决策提供更加有效的决策变量支持。复杂网络分析除了研究网络拓扑结构信息，也能够分析网络节点信息和网络连边信息以及全局网络特征信息。在深度图神经网络中，通过深度学习技术挖掘节点和连边信息以及网络拓扑结构特征，可为运用复杂网络分析相关问题提供额外的信息和研究思路。

金融市场是典型的复杂系统，复杂金融系统是一个由庞大数量、相互关联、互相影响的个体共同组成的系统，投资者行为能够决定宏观市场行为，从微观行为到宏观行为的跨越，是复杂系统研究人员希望理解和分析的关键问题。根据中国证券登记结算有限责任公司数据，截至 2020 年 1 月末，中国股市的投资者数量已经突破 1.6 亿人，其中包括了个人投资者和机构投资者。金融系统每天产生海量信息，包括投资者情绪、市场行情、交易行为和其他另类数据（Alternative Data）。复杂金融系统中海量、异构、多源的数据都是投资者的决策信息，但金融系统的复杂关联也导致了系统的脆弱性，在不可预知的风险和冲击面前，整个金融系统面临着巨大的崩塌风险。很多学者从微观层次上构建投资者交易网络[13]，通过对微观交易网络进行结构和动力学分析，为建立金融观察平台提供了丰富模型基础。

图1.1 是某只股票一年中交易者的股票买卖关系示意图，图中每个节点代表一个投资者，两个节点之间的连边对应两个投资者之间的股票买卖关系。通过 k-shell 算法进行分析和可视化[14]，可以得到图中的层次结构，为了显示清晰度，图中只显示了交易网络中最里层的投资者网络关系结构。层次结构表明投资者之间关系错综复杂，如何从如此复杂的拓扑结构中解构出市场交易行为以及解构出能够表征市场系统性风险的特征信息，是研究人员面对的较大挑战，经典理论和方法的局限性显而易见[15]，学者可以采取基于网络的建模方法[16-24]，基于系统论的视角来分析和研究复杂金融问题。网络模型能够对复杂系统进行较为真实的刻画和系统分析，将个体信息不仅是当作独立的特征变量分析，而是充分考虑个体之间的关联结构，从复杂网络结构和功能的视角讨论系统稳定性和脆弱性。近年来，复杂网络科学家们贡献了大量的复杂网络分析方法和理论思想，使得复杂网络方法成为了理解、描述、量化、预测并控制经济金融系统的强大工具[8]。复杂网络分析将更多因素引入了系统分析之中，使得模型的维度变得异常之高，一般方法很难能够同时考虑这么多的因素，结合复杂网络和机器学习以及深度学习、深度强化学习来处理超高维数据，能够使分析结果更加具有合理性、可行性和实用性。

复杂金融经济系统的稳定性和脆弱性问题，都对世界居民的日常生活和经济发展产生直接影响，如何在如此复杂多变的环境下应对突发事件是人们亟需解决的问题。如今，社会经济系统是一个高度耦合、深度关联、多尺度、多层次的复杂巨系统，传统方法已经很难处理具有庞大系统、动态环境以及海量数据的问题，人们需要结合最先进的智能算法和最优秀的计算平台来构建最有效的工具以生成应对策略，用复杂性对抗复杂性，用复杂智

能决策系统对抗复杂环境决策问题。自 AlphaGo 之后，深度强化学习一跃成为了热门的研究领域和极具前景的智能算法。在金融经济系统中，基于深度强化学习的智能投顾、智能资产管理、智能客服等都得到了大量的研究和应用。本书中大量的编程实践也采用了金融领域的智能交易和智能资产管理等应用实例，提供了入门深度强化学习理论和实践的基础案例，将理论和实践进行充分的融合和应用。

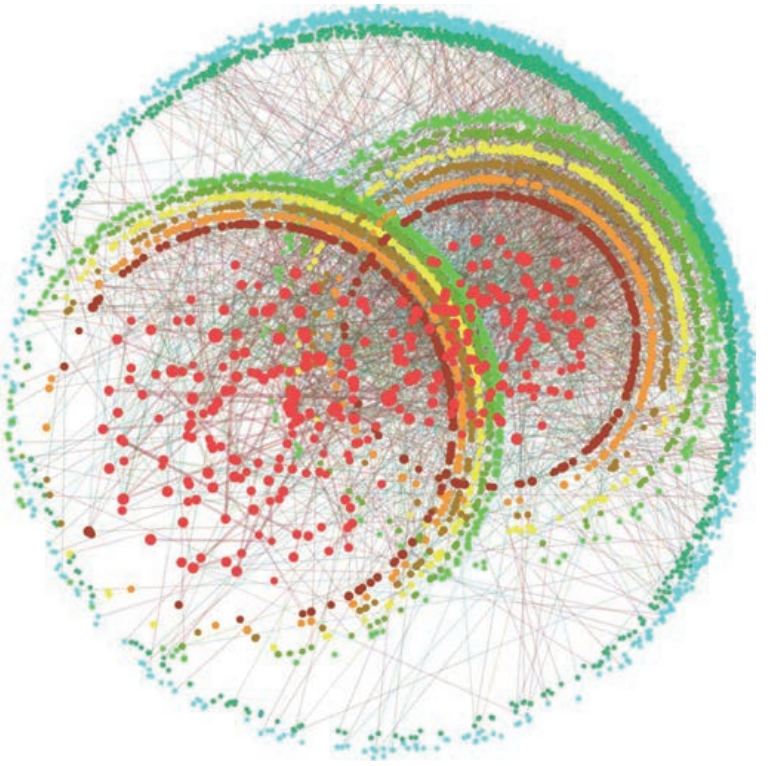


图 1.1　股票买卖关系图

1.2　复杂系统

复杂系统是由大量异质性个体组成的，且个体之间存在交互作用。复杂系统广泛存在于现实世界和虚拟世界之中。

1.2.1　复杂性科学

1984 年，在诺贝尔物理学奖得主、夸克之父盖尔曼（Murray Gell-mann），考温（George Cowan），安德逊（Philip Anderson），诺贝尔经济学奖获得者阿罗（Kenneth J. Arrow）等人倡导下，一批来自世界各地的政府机构、研究团体和私营企业的物理、生物、经济、计算机科学家，在美国新墨西哥州圣塔菲市西北郊外的一座小山丘上建立了享誉世界的圣塔菲研究所。圣塔菲研究所是非营利性研究机构，研究的大方向是跨学科的复杂性和复杂系统，并将研究复杂系统的交叉学科称为复杂性科学 [25-28]。

定义 1.1　复杂性科学

复杂性科学是指以复杂系统为研究对象，以超越还原论为方法论特征，以揭示和解释复杂系统运行规律为主要任务，以提高人们认识世界、探究世界和改造世界的能力为主要目的的一门“交叉学科”(interdiscipline)。复杂性科学主要包括早期研究阶段的一般系统论、控制论、信息论、人工智能，以及后期研究阶段的耗散结构理论、协同学、突变论、超循环理论、混沌、分形、自组织临界理论和元胞自动机等。♣

圣塔菲研究所的科学家们致力于构建“没有围墙的研究所”。圣塔菲研究所的研究范围广泛，融合了社会科学、经济金融学、计算机科学、生物学、生态学、信息学等学科，大力倡导不同学科之间的交叉融合。复杂性科学的关键不在于学科本身，而在于不同学科之间的交叉融合，在于学科之间的协同创新，共同解决科学问题和应用难题。经过了几十年的发展，复杂性科学经历了不同学科的兴起和衰落，如大家所熟知的老三论和新三论，从一般系统论、控制论、信息论，到耗散结构理论、协同学、突变论，以及超循环理论、混沌、分形、自组织临界理论和元胞自动机等，复杂性科学得到了不同领域学者的关注和研究。复杂性科学为不同学科提供了崭新的研究视角和创新性的研究成果，激发了新的研究思想，成就了新的研究方向，为人类揭示和解释复杂系统运行规律提供了强有力的工具，极大地提高了人们认识世界、探究世界和改造世界的能力。

21 世纪初，网络科学迎来了蓬勃的发展。针对蛋白质作用网络、细胞网络、神经网络、社会网络、经济金融网络等，科学家们在不同尺度、不同领域、不同层次进行了深入分析和研究，各领域相互借鉴、相互学习、共同发展、交叉融通，极大地促进了各个学科自身的发展。各个学科的发展都为复杂性科学做出了贡献，提供了思想的源泉和创新的火花，使得人类能够对身边的复杂系统、复杂模型进行深入的理解和探究，为人类认识世界、理解世界、预测世界提供了丰富的思想工具和技术方法，使得人类能够更好地可持续发展，为防控社会、经济、金融系统性风险提供了很多实用的工具和发展方向 [29]。

著名理论物理学家斯蒂芬·霍金称“21 世纪将是复杂性科学的世纪”。中国著名科学家钱学森在系统论和控制论领域做出了卓越贡献，丰富了复杂性科学方法，提出了复杂巨系统的概念。钱学森的《工程控制论》系统性给出了控制领域的经典方法和实例 [30]。在深入学习深度强化学习过程中，可以发现控制论中很多概念和思想都直接影响了强化学习理论的发展和算法演进。社会系统和物理系统的复杂性不在一个层次，物理学研究已经建立了许多精确的数学模型，可以进行论证、推演、理论分析、定量计算物理系统中个体动力学规律和系统演化规律 [31-33]，但社会经济系统的定量分析、预测、控制却异常困难和复杂。

成思危教授的《复杂性科学探索》指出了研究复杂系统的系统科学方法，包括定性判断与定量计算结合、微观分析与宏观分析结合、还原论与整体论结合、科学推理与哲学思辨相结合 [34]。复杂性科学是涉及多个学科的一门科学，汲取了不同学科的优秀方法和思想，是从不同学科和不同视角用不同方法和工具进行交叉研究的一门学科。除了部分物理系统，复杂系统中社会安全突发事件 [35-38]、神经网络的思维过程、动物种群的发展过程、胚胎的形成过程，都没有定量的数学模型，也没有大一统的理论，因为一些系统不易观测或者无

法模型化，不容易进行定量研究。但是通过分析系统作用机制，设计系统模型规则，可以在计算机上建立一定法则进行模拟，并进行定量分析。因此复杂性科学的研究需要定量与定性相结合 [25,39]，博观而约取，交叉而融通，以揭示和解释复杂系统运行规律。

1.2.2 复杂系统定义

2021 年 10 月，真锅淑郎（Syukuro Manabe）、哈塞尔曼（Klaus Hasselmann）和帕里西（Giorgio Parisi）三位科学家因为“发现了从原子到行星尺度的物理系统中无序和波动的相互作用”和“对地球气候进行物理建模、量化变化和可靠地预测全球变暖”而获得了诺贝尔物理学奖，为人们理解复杂物理系统做出了突破性贡献。帕里西在无序的复杂材料中发现了隐藏的模式，这是对复杂系统理论最重要的贡献之一，使理解和描述许多不同的、随机的材料和现象成为可能，在物理、数学、生物、神经科学和机器学习等领域都有着重要作用。

帕里西于 1999 年在论文《复杂系统：一个物理学家的观点》里写道 [40]：“复杂系统有许多可能的定义。我将复杂系统定义为：如果一个系统的行为在很大程度上取决于系统的细节，那么该系统就是复杂的，且这种依赖性往往是非常难以理解的。”复杂系统的定义非常之多，也各不相同。例如，复杂系统的另一个定义为“复杂系统是具有涌现和自组织行为的系统”。因此，对复杂系统进行定义，其本身也是很复杂的问题。下面将给出一个较为通俗的定义，以便理解和深入研究复杂系统。

定义 1.2　复杂系统的通俗定义

复杂系统由大量相互作用的成分组成，不存在中央控制，通过简单运作的规则产生出复杂的集体行为和复杂的信息处理，并通过学习和进化产生适应性。复杂系统存在的三个共性 [41]：

- 复杂的集体行为：个体简单，规则也简单，不存在中央控制或领导者，但集体却产生出了复杂的行为模式。
- 信号和信息处理：信息、信号的传递和利用。
- 适应性：所有的系统都通过学习和进化进行适应，即改变自身的行为以增加生存或成功的机会。

1.2.3 复杂系统类型

适应性造就复杂性 [42]。复杂系统中异质性个体具有自适应性，个体之间的相互作用在宏观层面涌现出复杂现象和有趣规律。异质个体行为影响周围环境，反过来环境变化也影响个体的行为。反馈是控制理论中的核心概念。个体与环境的交互过程，是个体基于环境反馈信息的学习、适应和应对的过程。基于经典还原论观点，通过研究复杂系统局部子系统性质而得到全局系统规律的思路和方法，已经很难准确地把握问题和解决问题，因为大部分子系统之间都不具有可加性，个体之间的关联使得子系统之间具有超线性或亚线性关

系，整体大于局部之和是复杂系统的重要特征 [37,39]。复杂性科学中，学者们通过非还原论方法来研究复杂系统，通过复杂模型和方法来发现和探索复杂系统的普遍规律，通过对复杂系统状态变量进行观察、量化、度量、建模、预测以及控制，深入探究复杂系统的动力学过程和演化规律，为理解和控制复杂系统提供有效工具 [36,43]。复杂系统包含了适应系统和非适应系统两大类。

1. 适应性复杂系统

个体具有适应性，复杂系统也具有适应性。个体之间通过物质、能量和信息的交换而产生相互作用，并通过与环境交互以适应环境变化，改变自身特征属性与交互行为，如此形成的系统称作适应系统，如生物网络 [44-48]、社会网络 [49-57]、金融经济网络 [58-60]、信息网络 [38,61] 中存在的大量适应性复杂系统。特别是有人参与的系统中，个体的行为决策过程受到环境因素影响，环境因素包括群体情绪、个体所处环境、地位等信息。个体感知环境信息，将其作为行为决策变量，进行信息处理和决策，表现出复杂行为模式。随着深入理解和分析深度强化学习系统，我们将发现深度强化学习系统本身也是一个适应性复杂系统。

信息社会中个体之间有着多重关联关系。社会系统中人与人之间息息相关，人类决策行为受到关联个体的影响，如同事、朋友、亲人等。社会系统受到了人类行为因素和环境因素影响，具有复杂的动力学特征和规律。除了动力学规律的复杂性，复杂系统同样包括了系统结构的复杂性，如社会网络系统的层次性、同配性和高聚类性，以及信息网络系统的无标度性和结构自相似性等。复杂系统结构的自相似性非常普遍，一些子系统和全局系统之间存在着关联，同时也具有相似的结构，如树状网络中每一棵子树也具有严格的树状结构。

一般而言，超大复杂系统由大量复杂子系统构成。大量个体构成复杂子系统，而大量复杂子系统进一步构成一个更大的复杂系统。复杂巨系统的组成部分也是系统，即为系统的子系统。在商业社会系统中，集团公司包括了大量子公司，子公司之间有着错综复杂的关联关系，子公司的员工之间也同样有着多重复杂关系。科研人员为了更好地理解和刻画如此复杂的超大系统，提出了很多有趣的理论和方法。波士顿大学 Stanley 教授等人提出的相依网络（Interdependent Network）是一套有效的理论分析框架，可研究社会和金融系统中“网络的网络”中“节点”相依关系对网络稳定性（Stability）和稳健性（Robustness）的影响 [62,63]，也为“系统的系统”和耦合系统研究提供了思路和方法。

2. 非适应性复杂系统

非适应性复杂系统是人类生活的重要组成部分。现实世界中大多数物理系统，如恒星、星系、行星、沙堆模型等，都属于非适应性复杂系统 [29,64]。在物理系统中，个体之间也会存在关联关系和相互作用，一些作用关系可以通过较为严格的函数或方程来表示和刻画，如行星之间的引力关系等。复杂物理系统的复杂性表现在维度高、空间大、非线性等特征，其中大多数问题没有解析解和高效求解方法，如行星轨迹预测问题、卫星发射问题等，都涉及大量的数值计算和数值优化，计算复杂度较高。不同于适应性复杂系统的个体，行星没

有自我意识，不能根据物理环境变化而改变自身的行为策略，只遵循已知或未知的物理规律。因此，在一般情况下，复杂物理系统和复杂社会系统的复杂性不在一个层次。

一般而言，非适应性复杂系统中的物理模型或化学模型与现实世界具有较高的相似性或一致性，如星系模拟系统和分子模拟系统等。适应性复杂系统中的虚拟模型与现实系统的相似性较低，如社会模拟系统中个体情感因素、属性等特征都很难准确量化和表征，个体之间的复杂交互模式也很难精确模拟，因而基于模拟仿真的社会系统或金融系统的动力学特征规律很难直接应用到现实世界之中。用深度强化学习对复杂系统和复杂环境进行准确建模，是智能体学习智能策略的基础。深度强化学习智能体的目标是期望在与复杂环境交互过程中获得较高的累积回报。因此，对复杂系统环境的建模和环境状态感知直接决定了智能体策略函数的优劣。

复杂系统模型可以是方程、方程组、动态演化的网络模型等。复杂系统模型与现实系统相似度越高，复杂系统模型的数值模拟结果应用在现实系统中就越有效、越可靠。复杂系统环境的状态直接影响智能体行为和智能体回报，智能体在好模型（与现实系统相似度高）中学习的行为策略在现实世界具有较好的泛化能力以及能够进行有效应用。因此，在深度强化学习的应用和实践中，环境模型的构建非常重要，深入研究和建模实际复杂系统是解决实际复杂问题的基础和关键。

1.2.4 复杂系统研究

复杂系统建模是智能体学习和优化智能策略的基础和关键。如何高效、高质量地建模复杂系统呢？复杂网络方法和思想是研究复杂系统与复杂社会现象的重要工具。我们需要研究复杂网络结构的统计规律和属性特征，同时需要更多地关注和理解复杂网络动力学特征规律。复杂系统中存在着大量亟需解决的重要问题。复杂网络科学家 Barabási 等人在理解 [65]、量化 [66]、预测 [67,68] 和控制 [69,70] 复杂社会现象和复杂自然现象方面做了大量创新性工作，相关研究成果得到了广泛应用。人类移动行为动力学的相关研究 [65,66] 为交通网络拥堵、流行病传播等复杂现实问题提供了新思路和新方法。如何有效地避免交通堵塞，如何有效地预防传染病传播，如何阻断病毒传播路径等，都可以建模成复杂网络问题。深度强化学习方法训练智能体在基于复杂网络的环境模型中探索和学习，优化决策函数，探索智能化的解决方案。

复杂网络科学家 Barabási 等人综合分析了复杂网络可控性和可观察性 [69,70]，从复杂网络的视角理解和分析复杂系统的可控性，将控制的思想引入复杂网络研究中并进行了推广。复杂网络结构可控性问题的相关研究成果也被应用于一些复杂系统分析之中，如生物体神经网络和行为控制等。使用网络可控性研究方法分析网络是否可控，即如果人类想要控制网络中个体状态特征到达指定的状态特征，那么是否存在一定的控制操作可以满足人类的需求。但是，如何找到控制策略则需要更多的辅助工具和数值计算方法，这也限制了复杂网络控制的方法和思想在现实世界网络中的广泛应用。通过追溯深度强化学习的历史可以发现，强化学习发展初期融合了大量的控制论思想和方法，如最优化控制理论等。总而言之，复杂网络是研究和分析复杂系统的重要工具和方法。

1. 复杂社会系统研究

复杂社会系统是人们生活、工作和学习的环境。人们常说社会太复杂，因为人类社会是由大量异质且动态多变的个体组成的复杂自适应巨系统。近年来，社会系统和金融经济系统中的“黑天鹅”事件[67]（如突发社会安全事件、金融危机）等都给社会、经济、金融系统造成了一定程度的伤害，也给人们的生活、工作和学习造成了较大冲击。复杂系统之间存在一定程度的耦合，研究和分析耦合系统具有挑战性。金融经济系统的复杂性一部分来源于与社会系统的耦合关系，金融经济系统中个体之间存在着错综复杂的社会网络关系，个体的金融经济决策过程会受到社会关系的影响，因此，个体的决策环境具有复杂性，个体的决策行为具有复杂性，社会经济系统的整体行为也具有复杂性。

2011 年初，美国自然科学基金会基于哈佛大学学者对全球著名社会学家的调查分析，发布了社会科学研究的十大科学问题，其中第九大科学问题为“我们怎样才能坚强地应对罕见的、会造成极端后果的黑天鹅事件?”虽然自然灾害、事故灾难、公共卫生事件和社会安全事件等“黑天鹅”事件极其稀少，但是对社会、经济和金融系统的危害却十分严重。识别具有破坏性的稀有事件发生的时间和空间极具挑战。在社会科学研究领域中构建抵御突发事件冲击的观察和检测系统，是智能社会治理的重要研究方向。传统的社会科学研究方法包括定性分析、定量分析和实验研究。随着信息技术和计算机科学的发展，社会科学研究领域引入了大量的定量化分析和研究工具，如 SPSS、Maple、MATLAB、MathCAD 和 Mathematica 等。

Vespignani 等人分析了人类行为的可预测性[67,68]，挖掘了人类行为的特征规律，颠覆了人们多年来关于个体行为不可预测的观念。在传染病模型的构建和实际疫情防控中，人类行为可预测性的相关研究成果具有重要意义和作用。如何更好地预测人类行为？如何在大规模数据集上进行人类行为规律的探索？机器学习模型和深度强化学习算法能够有效地学习蕴含于人类行为日志数据中的信息，提高智能模型对人类行为的预测精度。

2019 年底的新型冠状病毒感染肆虐全球，流行病的突然暴发，人们的生活方式、学习方式、工作方式都发生了巨大变化。如何有效地预防和管控传染病的暴发和传播，一直都是人类潜心钻研的课题。人们用“黑天鹅”来表示复杂社会行为的不可预测性以及对人类认知和观念产生的巨大冲击。瑞士工程院院士和欧洲科学院院士 Didier Sornette 教授提出的“龙王”理论提出了不同于“黑天鹅”的方法和思想，用来理解和预测金融市场和人类社会中的群体行为[71-74]。Didier Sornette 教授是南方科技大学风险分析预测与管控研究院院长和讲席教授、瑞士苏黎世联邦理工学院（ETH Zurich）创业风险中心讲席教授、ETH 风险中心联合创始人、瑞士金融研究所金融学教授，在自然和社会极端事件预测中进行了大量的创新研究[73-81]。

复杂社会、经济、金融和自然系统中的很多重要问题可以用深度强化学习方法进行分析和研究，如地震预测、金融市场系统性风险预警等。科学研究中真正重要的问题很少，难度也很大，研究方法却很多。随着科学技术的进步，我们要勇于尝试新方法和新思路，为求解重要问题提出新见解和新思路。深度强化学习方法为复杂社会网络分析提供了强大的技术和算法支持，使得社会科学领域的智能决策更加具有可行性和可靠性。在国内外学者

的努力下，机器学习和复杂网络领域都得到了快速发展，已经有了大量非常成熟的算法和应用，许多研究成果发表在 Nature、Science 等国际顶级学术期刊上，引起了世界范围内学者的重视。深度强化学习在复杂系统决策领域已经取得了一些重大科学发现和理论突破，很多研究成果得到了广泛应用，且取得了较好的实际应用效果，如疫情防控、疫苗研发、蛋白质折叠等。

2. 虚拟社会系统研究

海内外学者对虚拟世界的潜在研究价值有很高的期许 [54,82]。特别是近年来虚拟现实（Virtual Reality，VR）、元宇宙（Metaverse）、数字孪生（Digital Twin，DT）等技术引发了社会各界的极大关注。大数据和高性能计算时代，人们对复杂社会系统演化行为的定量认知并不深刻，需要更加定量化的、高效的分析方法和研究思路。人们通过对海量数据的分析和高性能计算资源的调度，深化对虚拟社会个人行为和组织演化规律的定量认识，具有重要的现实意义和科学价值。

传统定量分析和实验研究的小样本数据集与虚拟社会系统中人类行为的海量日志数据集存在显著的规模差异，因此研究方法和工具也发生了变化。不同领域的科学家为了更科学和更高效地定量分析大规模高质量社会领域数据集，开拓了全新领域——计算社会科学 [83]。计算社会科学所用数据集具有大数据的基本特征，比如通信运营商记录的手机用户通话时间和位置数据集，科学家可以用此数据集研究人类的移动行为规律，并在传染病模型中引入社会个体移动规律，更加真实地仿真模拟传染病传播过程，为疫情防控提供更加可信和科学的政策建议 [65,66,68,84]。同样，深度强化学习方法也将大有可为。

在虚拟社会系统研究领域中，科学家通过虚拟世界日志数据研究现实世界人类行为规律，存在一个虚实映射的问题。在虚拟世界中所发现的角色或个体行为的特征规律不一定适用于现实世界人类行为，因为所发现的规律可能只是特定虚拟世界模型设定的微观规则在宏观层面的系统行为。在深度强化学习应用中，虚拟与现实映射问题同样需要重视，即复杂环境建模中环境模型与现实世界的差异问题。智能体在仿真模拟的环境模型中表现良好，却可能在现实世界中不能工作，这与机器学习领域中模型泛化和迁移问题相关。

3. 复杂经济系统研究

面对近年来的经济危机、欧债危机、粮食危机、金融危机、全球新型冠状病毒感染、俄乌冲突等极端事件，人类亟需反思，突发事件和复杂现象背后的形成机制和原因是什么？如何找到极端事件的起因？如何测度和监控极端事件的演化和发展？如何在下一次危机前进行有效的预警和防控？人类身处复杂系统之中，深知复杂系统的稳健性和脆弱性共存。例如，在网络科学中，复杂网络面对随机性的节点失效时呈现稳健性，面对蓄意的具有针对性的节点攻击时却表现出脆弱性。面对复杂社会经济系统中的极端事件，人类希望实现像对待自然现象一样，进行有效的观察、理解、描述、量化 [85-87] 、预测和管控。

人类如何才能做到有效地观察、理解、描述、量化、预测并管控复杂社会经济系统呢？南京大学盛昭瀚教授团队在社会经济系统的建模和计算实验研究领域取得了大量创新成果。在经济学领域，传统定量分析和实验研究的数据样本一般较少。随着信息技术的突飞猛进，

经济系统中个体对于信息技术的依赖，使得个体在信息系统中留下了详尽的数字痕迹，如个体上网、电话、位置信息、支付信息、购票信息、行程信息等，都反映了个体的行为特征和偏好。同样，很多平台或公司也收集了个体的性别、学历、偏好等属性信息，并通过整合个体和系统信息来进行智能商业决策，提高平台收益并扩张市场规模。商业数据的采集和分析会涉及到非常之多的隐私数据，信息社会中数据安全和隐私保护也是人们需要关注的重要问题。

除了现实世界个体的经济行为数据，大型多人在线角色扮演游戏（Massive Multiplayer Online Role-Playing Game，MMORPG）中也产生了大量完备和丰富的角色经济行为日志数据。虚拟世界中虚拟经济系统发挥了举足轻重的作用。我们通过研究虚拟世界中社会和经济系统，刻画虚拟社会经济系统的动力学演化规律，进而分析虚拟在线社会的相关经济问题，因此虚拟世界中的社会和经济系统也同样具有较大发展潜力和研究价值 [29]。

4. 复杂金融系统研究

金融经济系统中产品、供给、需求、价格等基本要素都有自身的特征属性，且加上异质金融个体之间错综复杂的关联关系和交互作用，造就了金融系统的复杂性。金融系统复杂性包括金融本质的特殊性与复杂性、金融产品与金融机构的特殊性与复杂性、金融市场与金融资产价格的特殊性与复杂性、金融风险的特殊性与复杂性、金融技术的特殊性与复杂性以及金融管理与调控的特殊性与复杂性 [88]。人们通过对金融系统复杂性的认识，深入了解现阶段金融市场形态演化规律的内在机制和驱动力，为理解复杂金融现象提供分析工具和分析思路。人们需要多角度、多尺度、多层次地理解和刻画复杂金融市场，深入探究金融市场系统性风险的度量、预警、传染和防控策略。

金融系统复杂性包括客观复杂性和主观复杂性两个方面 [88]。复杂金融系统的客观复杂性是指系统本身的状态、结构和演化动力学的复杂性，以及刻画金融系统的模型具有复杂性，与传统复杂性研究一致，包含如计算复杂性、算法复杂性和语法复杂性等。复杂金融系统的主观复杂性是指复杂金融系统中个体具有适应性和能动性。人类感知、意识、反应和行为的复杂性以及个体之间交互行动、信息反馈、信息级联和迭代等因素的复杂性，都将金融系统复杂性提升到了更高层次。有人参与的系统和无人参与的系统的复杂性是有本质区别的，与适应性复杂系统和非适应性复杂系统的区别类似。

面对复杂金融系统，人们如何能够在维持金融系统稳定性、稳健性同时，有效地配置资源，为社会经济系统高效运行提供动力，是一个亟需解决的问题。新的金融工具和金融方法关注和研究复杂金融系统风险的预警和防控问题。研究金融系统的主观复杂性需要结合心理学和金融学，包括行为金融领域研究预期判断、风险态度、决策方式、信息条件等，针对这些领域已经开展了大量的研究工作，取得了非常丰硕的研究成果。针对客观复杂性，我们需要结合数理知识与计算机科学，如计算机仿真、人工神经网络、随机过程、统计分析、混沌动力学、随机复杂性、数理金融与量化金融等 [88]。下面将简单介绍一些在大数据时代、人工智能时代发展和兴起的新金融。

5. 计算实验金融

天津大学张维教授课题组在金融系统的计算实验研究领域处于国际前沿。张维教授是国内极早从事金融工程与金融风险管理领域教育和研究的学者之一。张维教授等人在著作《计算实验金融研究》中阐明了计算实验金融的研究方法论，详细介绍了计算实验金融学的起源、发展历程和研究现状，利用计算实验金融方法对金融市场的各种异象做出合理解释，并对投资者生存、适应性市场假说、时间序列可预测性等金融学界广为关注的问题做出尝试性回答。计算实验金融融合了金融学、计算机科学、概率论、统计学等学科，是一门交叉学科。张维教授等人尝试在中国市场条件下，利用计算实验金融方法解决一些常规金融经济学方法难以解决的金融研究问题，倡导和推动了计算实验金融学在中国的发展。

6. 互联网金融

2015 年，中国人民银行等十部门发布《关于促进互联网金融健康发展的指导意见》，对互联网金融做了定义。互联网金融是传统金融机构与互联网企业利用互联网技术和信息通信技术实现资金融通、支付、投资和信息中介服务的新型金融业务模式。互联网与金融深度融合是大势所趋，已对并将继续对金融产品、业务、组织和服务等方面产生深远影响。

互联网金融作为“互联网 +”的重要产业之一，对社会经济系统产生了举足轻重的作用，也对人们日常生活和经济活动产生了深远影响。在大量金融创新的环境下，互联网金融改变了传统的金融行业和金融生态，深刻影响了商业银行的传统业务，也引发了新的金融系统风险源。金融从业人员和学者们需要深入研究互联网金融风险管理和互联网金融监管等问题，共同维护社会经济系统和金融系统的稳定和健康发展。

科学技术的创新和突破带来金融行业的变革和产业调整。互联网、人工智能、区块链、云计算、大数据、物联网等新技术不断涌现，加速了经济体和金融机构的发展和转型升级。通过新技术、新资源和新工具，复杂金融市场主体可以提升金融资源配置效率，为实体经济发展注入新活力。互联网金融的快速发展给社会带来巨大的经济利益，同时也要注意利弊的权衡，互联网金融中一些金融创新规避监管，盲目扩张，容易引发新的金融风险。面对国家防范化解重大系统性金融风险的首要任务，预警、防控和化解互联网金融带来的系统性金融风险显得尤为重要。

7. 科技金融

2019 年，科技部发布《国家“十二五”科学和技术发展规划》，指出科技金融是指通过创新财政科技投入方式，引导和促进银行业、证券业、保险业金融机构及创业投资等各类资本，创新金融产品，改进服务模式，搭建服务平台，实现科技创新链条与金融资本链条的有机结合，为初创期到成熟期各发展阶段的科技企业提供融资支持和金融服务的一系列政策和制度的系统安排。加强科技与金融的结合，不仅有利于发挥科技对经济社会发展的支撑作用，也有利于金融创新和金融的持续发展。

8. 金融科技

金融科技英译为 FinTech，是 Financial Technology 的缩写，可以简单理解为 Finance（金融）+ Technology（科技），指通过利用各类科技手段创新传统金融行业所提供的产品和服务，提升效率并有效降低运营成本。金融稳定理事会（FSB）给出了金融科技的定义，金融科技主要是指由人工智能、区块链、云计算、大数据分析等新兴前沿技术带动，对金融市场以及金融服务业务产生重大影响的新兴业务模式、新技术应用、新产品服务等。

金融科技涉及的技术具有更新迭代快、跨界、混合行业等特点，是人工智能、区块链、云计算、大数据分析等前沿颠覆性科技与传统金融业务、场景的叠加融合。金融科技主要包括大数据金融、人工智能金融、区块链金融和量化金融四个核心部分，包含如智能投顾和智能客服等产品。智能投顾是指投资人依靠专业智能机器人来进行投资决策，智能投资机器人结合投资者的财务状况、风险偏好等，运用已搭建的投资模型和计算平台为投资者提供投资和理财建议。证券行业的智能客服主要充当客服的身份，但随着金融科技的发展，也能够进行诊股、选股等智能操作。

1.3　复杂环境特征

复杂环境中智能体决策基于复杂环境特征，可以理解为智能体所处的复杂系统状态特征。面对不同环境特征，我们需要构建不同类型的智能体进行策略学习。拉塞尔（Stuart J. Russell）和诺维格（Peter Norvig）的经典人工智能教材《人工智能——一种现代方法》对智能体任务环境进行了非常深刻的分析 [89]。任务环境特征直接影响深度强化学习中智能体策略学习和算法分类，也是设计不同环境模型前必须确定的关键建模因素。

基于拉塞尔和诺维格对环境性质的分类，可以更好地理解和分析复杂环境特征，从而设计和训练对应的智能体模型。通过了解深度强化学习方法的分类情况，我们可以发现算法的特征和性质与复杂环境的特征和性质相互关联，可以在一个统一的框架下分析算法、分析问题、建模环境和智能体。我们将从不同角度刻画和分析复杂环境特征。

1.3.1　完全可观察的和部分可观察的环境

在智能体与环境的交互过程中，如果智能体能够感知到与智能体决策相关的全部环境状态信息，则认为复杂环境是完全可观察的；如果智能体无法完全感知与决策相关的环境状态，则环境是部分可观察的。复杂环境状态不能完全被观察的原因有很多，比如噪声干扰、感知器灵敏度差、数据丢失等情况，都使得智能体无法完全获得决策所需信息。

在部分可观察情况下，为了使智能体进行智能决策，可在智能体内部构建一个隐空间，将部分可观察的环境状态变量映射到隐空间，智能体基于隐空间的隐变量进行决策，如部分可观察马尔可夫决策过程（Partially Observable Markov Decision Process，POMDP）。

举几个例子。在围棋对弈中，对弈者基于棋盘落子信息进行决策，并完全由棋盘信息决定策略行为。对于对弈双方而言，棋盘落子信息是完全可观察的。自动驾驶中，自动驾

驶系统如果遇到大雾天气，路面情况会出现遮挡，不是完全可观察的。当智能体无地图时，智能体仅能感知所处位置周边小范围内的道路交通情况；当智能体拥有详细的地图信息时，智能体对所处环境是完全可观察的。在完全可观察环境中，智能体能够进行高效规划和学习；而在部分可观察环境中，智能体需要基于部分可观察信息进行假设和推理，如大雾天气中行车的驾驶员需要基于经验对部分可观察的路面进行判断，再进行决策。

1.3.2 单智能体和多智能体

复杂环境中智能体与环境交互，且环境中只有一个智能体，那么这种环境是单智能体任务环境；如果有多个智能体与环境交互，且智能体之间也互相作用、互相通信，那么这种环境是多智能体任务环境。毫无疑问，多智能体环境较单智能体环境更加复杂，因为多智能体环境中智能体不仅要与环境交互以获得环境信息，还需要考虑其他智能体的信息来进行决策。多智能体深度强化学习是一个非常有潜力和活力的研究方向。

智能体玩单机游戏时，复杂环境中只有一个智能体与环境进行交互，属于单智能体任务环境。围棋程序 AlphaGo 在训练过程中基于棋局进行落子决策，环境就是围棋棋盘落子情况，包括了对手方落子信息，属于多智能体任务环境。很多策略类游戏是需要多人合作完成的，在构建此类游戏智能体时需要同时构建多个智能体进行决策，训练智能体之间的合作、竞争和交互行为以及与环境交互，如 OpenAI Five 和 DeepMind 的 AlphaStar 都是多智能体强化学习的经典应用。

从单智能体到多智能体的延伸和拓展，也是一个非常有前景的方向。在很多实际应用场景中，多智能体决策更能贴合实际，如智能投顾和智能客服中的智能投资机器人。金融市场是一个多人博弈环境，个体行为和收益不仅与自身策略行为相关，也与其他参与者的策略行为相关，如果智能投资机器人能够考虑其他智能体的行为和策略信息，将可能更好地做出投资决策。

1.3.3 确定的和随机的环境

如果复杂环境的下一个状态完全由当前状态和智能体动作决定，那么环境是确定的；否则，环境是随机的。围棋游戏中的棋局是确定的环境，完全由当前棋局和下棋动作决定，不存在随机因素。军棋游戏中，暗棋和翻棋的棋局是不确定的，不确定性来源于部分可观察性。下暗棋时，棋子立起来不让对方看见，棋子的大小信息需要裁判给出，对弈者需要根据部分可观察的信息进行推理和判断，做出决策行为。

在金融市场中，投资者对市场状态信息的感知极其有限，能够获得的金融市场信息非常少，特别是散户投资者，只能获得部分公开信息，同时，投资者受限于信息处理能力等因素，获得有效决策信息较难。金融市场信息具有多源、异构、高频等特性，同时随机性因素较多，投资者通常需要在不确定的环境中做出投资决策。对于此类投资者而言，金融市场环境就是一个随机环境，而且市场随机性随着时间也会演化，增加了投资者做出正确决策的难度，也为智能投资机器人的建模和训练提出了极大挑战。

在自动驾驶场景中，自动驾驶汽车的感知系统所能收集到的信息是有限的，如摄像头观察距离是有限的，清晰度是有限的，路面的能见度也是有限的，这些不可观察的信息使得环境具有了不确定性，因此自动驾驶智能体面对的环境具有随机性。除此之外，一些突发的状况使得自动驾驶汽车所面对的随机性更大，如前方路面的车祸、车辆爆胎等。自动驾驶是当前人工智能领域最活跃的研究方向之一，也是资本投入最大的领域之一。

1.3.4　片段式和延续式环境

在片段式环境中，智能体的交互过程被分成了一个一个独立的片段，相邻片段之间的决策行为互不影响。例如，现实中随处可见的车牌识别（Vehicle License Plate Recognition，VLPR）系统对相邻两辆车的识别行为互不影响。

在延续式环境中，智能体行为之间具有关联性。棋类游戏中前后落子具有关联性，胜负是由一盘棋所有的决策行为（多步的落子）共同决定的。在金融市场中，智能体最后的收益也是由投资期内所有行为所共同决定，智能体当前买入行为的价值也受到后续投资决策行为的影响。强化学习算法就是专门针对此类序贯决策问题而设计的学习框架，现实世界中的很多复杂问题是可以建模成序贯决策问题的。

在实际应用过程中，我们可以将片段式决策过程和延续式决策过程进行转换。在金融投资过程中，投资者投资过程可以看成延续式，前后投资行为互相关联，而在实际程序设计过程中，智能体在片段式环境中训练和学习投资策略函数更加容易，因此我们可以对智能体投资行为进行设定或限制，比如一定时期内只能有一次行为动作，最后强制平仓，计算投资收益，再重新开始新一轮投资周期。

在智能算法运用过程中，我们也要避免“手里拿着锤子，看什么都像钉子”的心理，需要从实际问题出发，找合适的解决方法，并非所有的问题都可以应用深度强化学习来解决。奥卡姆剃刀原理告诉我们“如无必要，勿增实体”，即“简单有效原理”，尤其在工程应用或实际场景中，简单模型能够解决的问题，无须使用复杂模型求解。

1.3.5　静态和动态环境

智能体在进行决策的过程中，如果环境发生了变化，那么环境是动态的；否则，环境就是静态的。相对而言静态环境比较简单，智能体不需要时刻关注环境变化。在围棋游戏中，棋盘局面在智能体的决策过程中不会发生变化，当然此时不考虑决策时间限制。在现实世界中，绝大部分智能体的决策环境都是动态演化的。

金融市场是一个极其复杂的动态环境，在投资者决策过程中，市场信息瞬息万变。投资者决策需要时间，决策信息的采集完成时间点和决策行为的执行时间点存在一定间隔，当策略执行时，智能体先前考虑的市场变量已经发生了改变，这会影响智能体决策行为的准确性，要做到精准决策就会更加困难。投资者选择执行限价订单，在交易系统输入股票价格数字的几秒钟之内股票价格也可能发生变化，导致限价订单不能完成交易。在金融市场中广泛使用的自动化交易等高科技交易算法，也不能保证信息能被完全并及时地获取、处

理和决策。在复杂金融市场环境中训练有效的自动投资智能体具有极大的挑战。

在自动驾驶系统的决策过程中，车辆自身在运动，周边的车辆也在运动，路面情况和物理环境都发生了变化。自动驾驶场景的环境时刻发生着变化，因此构建安全可靠的自动驾驶智能系统极具挑战，需要投入大量的时间和资源进行研究和开发，也是未来人工智能系统落地应用的突破之一。

1.3.6 离散和连续环境

环境状态信息和智能体决策信息都需要用变量来表示，而变量可以分为离散型变量和连续型变量。离散型变量可以表示类别、等级等，连续型变量能够表示时间、温度、体积、位置坐标等。特定环境状态信息需要选用合适的变量来表征，环境状态变量是智能体与环境进行有效交互的基础，也是智能体决策的基础。

一般来说，复杂环境变量融合了离散型变量和连续型变量。在围棋游戏中，棋盘位置可以用离散型整数表示，其他的价值变量可以用连续型实数表示。在自动驾驶智能系统中，红绿灯信息可以用分类离散型变量表示，车辆速度和位置坐标可以用连续型变量表示。在金融市场中，订单类型、股票类别可以用离散型变量表示，价格、交易量和换手率等可以用连续型变量表示。

1.3.7 已知和未知环境

已知的环境和未知的环境分类主要基于智能体对环境模型的了解程度。如果环境中不同状态之间的转移函数或动力学演化规律都是可获得的，那么对于智能体而言，环境模型是已知的。在围棋游戏中，智能体在清楚地预测下棋行为（落子）之后，环境的下一个状态信息就确定了。在物理系统中，物理环境模型蕴含了基本的物理规则，环境模型系统的演化严格按照物理规则进行，因此，在智能体决策过程中物理规则是智能体已有知识的一部分，能够为智能体决策所用。

强化学习算法可以分成基于模型（Model-based）的算法和无模型（Model-free）的算法，其智能体交互的环境模型分别对应已知环境和未知环境。基于模型的强化学习算法能够充分利用模型的动力学规律，智能体与环境交互更为高效，能以较小的代价获得更多高质量的经验数据或者模拟数据样本，因此，基于模型的强化学习算法能够充分利用复杂环境模型进行规划和学习，加速学习过程，提高学习效率，节约计算资源。

在无模型的强化学习方法中，智能体通过与环境的交互获得经验数据样本，感知环境的动力学过程，通过经验数据训练智能体。一般来说，智能体和环境的交互过程需要耗费很多计算资源和存储资源。在机器人训练中，机器人与真实环境交互非常缓慢，如训练机器人的行走，受限于真实环境和机械设备，机器人的动作和移动速度有限，影响了机器人训练效率。虚拟的物理环境模拟系统（环境模型）能够加快智能体训练过程，虚拟的物理环境和现实环境差异较小，智能体能够高效地获得较好的模拟数据完成训练。要使智能体能够高效获取环境信息，条件是要能够模型化复杂系统环境。如果模型化的虚拟环境与现

实环境差异较小，那么智能体在虚拟环境中的智能策略就能够较好地泛化和迁移到真实环境。

1.4　复杂环境建模

复杂环境建模是智能体学习和优化智能策略的基础，环境模型的好坏决定了智能体决策行为的优劣，因此需要深入分析复杂系统的特征规律和环境状态表示以建立环境模型。图1.2中给出了一个简单的智能体与复杂环境进行交互的框架，智能体在复杂环境中学习和优化智能策略。

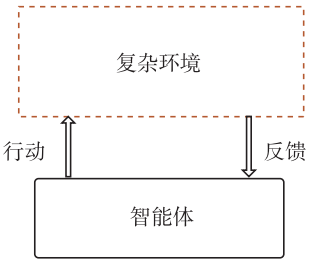


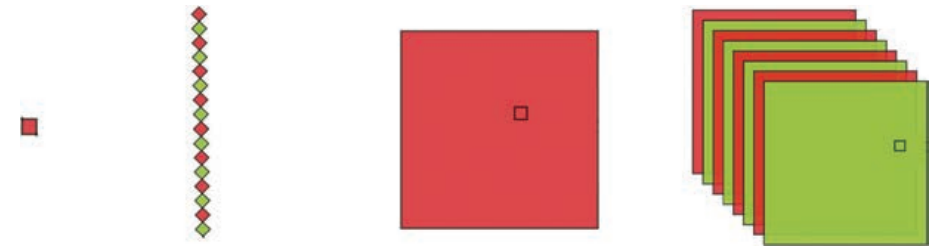
图 1.2　智能体与复杂环境的交互框架图

在图1.2 中，智能体和复杂环境是复杂系统的两个主要组成部分，智能体和复杂环境之间存在两个作用关系：行动和反馈。智能体输出动作，作用于复杂环境；复杂环境输出反馈信息，智能体接收反馈信息，反馈是复杂智能系统的关键，也是复杂系统控制的关键。反馈是经典控制论的一个核心概念，也是强化学习与自动化控制等经典学科之间联系的纽带。

智能体感知复杂环境状态信息，并基于自身策略输出行为动作，作用于复杂环境，复杂环境转移至新的状态并做出反馈，智能体感知到反馈信息后重新输出自己的行为动作，如此反复迭代。智能体收集反馈信息和环境信息并训练、优化自身策略。环境反馈信息包含了有关智能体“好”的行为的奖励信息以及“坏”的行为的惩罚信息，智能体通过反复试错，并学习优化，直到获得最优策略。在智能体和复杂环境进行交互的过程中，动作和反馈都需要通过变量进行表示，合适的智能体行为表示和环境反馈信息表示是计算机进行数值模拟和高效求解复杂问题的关键。

复杂系统环境建模需要收集大量的环境数据，对复杂系统环境进行刻画和表示，而大规模的数据集又给问题分析和求解带来了困难，因为在现实世界的复杂系统中，一些数据是不可获得的。在复杂系统中，智能体与环境交互，进行信息通信，并对复杂环境状态进行表征，感知环境特征，优化智能策略。因此，环境建模是智能体优化策略的数据来源，直接影响了智能体策略的智能水平。

复杂环境的状态特征、智能体行为动作以及反馈信息等都需要进行变量表示。图1.3给出了四种常用的数据类型结构示意图。



（a）　（b）　（c）　（d）

图 1.3　常用数据类型结构示意图

1. 标量型环境状态变量

图1.3（a）为标量（Scalar）x，是表征环境特征最常用的数据形式。环境的大小尺寸，可以用标量表示；智能体的速度大小可以用标量表示等。标量 x 按其测量尺度可以再进行细分，可简单分成 3 种：

（1）标量 x 为连续的实值变量，可表示间隔尺度，如身高、体重、时间等。间隔尺度是使用最广泛的数据形式，日常生活中随处可见。间隔尺度的标量能够进行加减乘除等运算，也能够比较大小关系。

（2）标量 x 为整数型分类变量，可表示有序尺度，如产品质量等级、学历等级等。有序尺度标量 x 只表示次序，例如小学、初中、高中和大学可分别用数字 1、2、3 和 4 表示，不等式 $2 > 1$ 表示初中学历高于小学学历，具有实际含义。有序尺度的指标不能够像间隔尺度指标一样进行加减法，如 $1 + 2 = 3$ 不能表示小学学历加初中学历等于高中学历，因此，有序尺度变量在模型构建过程中需要加以特别注意。

（3）标量 x 表示某些分类或属性时，称作名义尺度，如性别、季节等。与间隔尺度和有序尺度指标相比，名义尺度指标既不能进行加减运算也不能比较大小，例如，个体性别属性可以用 0 和 1 分别表示男性和女性，$0 < 1$ 并不能说明女性大于男性，名义尺度变量的数值大小关系没有实际含义。

在对环境和智能体属性进行表示的过程中，我们要细致分析特征的属性和可用变量的属性，运用合适的特征表示是对智能体进行有效训练和高效学习的基础。

2. 向量型环境状态变量

在图1.3（b）中，多个标量可以构成向量（Vector），向量可以表示环境和个体的多维特征属性，如三维空间中个体位置坐标可以用向量 (x_1, x_2, x_3) 表示。在数据分析中，时间序列数据也可以用向量表示，如股票价格时间序列、经济体历史数据等。

3. 矩阵型环境状态变量

图1.3（c）为矩阵（Matrix）示意图，将向量拼在一起可以构成矩阵。矩阵是线性代数的重要概念，也是机器学习常用的数据类型。很多数据分析软件将矩阵作为基本分析对象，如 MATLAB，就是矩阵实验室（Matrix Laboratory）的简称，MATLAB 将矩阵作为其主

要数据结构。一般矩阵可以表示如下：

$$
\begin{array}{c}
 \\
r_1 \\
r_2 \\
r_3 \\
\vdots \\
r_{n-1} \\
r_n
\end{array}
\begin{array}{c}
\begin{array}{cccccc}
c_1 & c_2 & c_3 & \cdots & c_{n-1} & c_n
\end{array} \\
\begin{pmatrix}
x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,n-1} & x_{1,n} \\
x_{2,1} & x_{2,2} & x_{2,3} & \cdots & x_{2,n-1} & x_{2,n} \\
x_{3,1} & x_{3,2} & x_{3,3} & \cdots & x_{3,n-1} & x_{3,n} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
x_{n-1,1} & x_{n-1,2} & x_{n-1,3} & \cdots & x_{n-1,n-1} & x_{n-1,n} \\
x_{n,1} & x_{n,2} & x_{n,3} & \cdots & x_{n,n-1} & x_{n,n}
\end{pmatrix}
\end{array}
$$

其中，矩阵元素 x_{ij} 的下标分别表示元素处于矩阵的第 i 行和第 j 列。

矩阵可以表示大部分数据类型。例如，最常见的图片数据可以用矩阵表示，图片大小就是矩阵大小，矩阵中每个元素对应图形中的像素点；研究国民经济各部门间关系时可采用投入产出方法，其主要研究对象是投入产出矩阵；博弈论中收益矩阵可以用于研究竞争或者合作之间的冲突问题；图数据（网络数据）的邻接矩阵、拉普拉斯矩阵都能够表示复杂网络或复杂图结构信息。现实世界中的很多研究对象都可以用矩阵表示。在强化学习中，智能体在不同状态之间的转化规律可以表示为状态之间的转移概率矩阵；在 AlphaGo 和 AlphaGo Zero 中，用 19×19 矩阵表示围棋棋盘信息。

4. 张量型环境状态变量

图1.3（d）为张量（Tensor）示意图，为张量最简单的表现形式，可以看成矩阵堆叠在一起构成的高维数据，严格定义可以参考数学或物理教材。深度学习领域流行的机器学习框架 TensorFlow 和 PyTorch 都将张量作为主要操作对象，类似于 MATLAB 将矩阵作为其主要操作对象。我们可以从另一个角度重新理解各个类型的数据，零阶张量为标量，一阶张量为向量，二阶张量为矩阵，随着数据维度增加，数据越来越复杂，分析方法和工具也越来越复杂。AlphaGo 不仅仅考虑了单步落子的信息，还考虑了历史落子信息，因此构造了大小为 $19 \times 19 \times 17$ 的张量表示棋局状态信息。

5. 网络型环境状态变量

矩阵型数据可以表示复杂网络的结构特征。在实际数据分析中，我们经常遇到关系型数据，可以采用复杂网络方法进行分析。一般来说，融合网络分析能有效提高智能决策系统的实用性和准确度。图1.4给出了 73 个村庄社会网络示意图，数据来源于一次大规模调查问卷 [90]。图中每一个网络节点对应村庄中一位村民，网络连边代表了“如果你必须做出一个艰难的个人决定，你会向谁寻求建议?”的关系。

在图1.4中，社会网络关系存在明显的社团结构，每一个社团对应一个村落。同村落中村民之间的关系较紧密，即村民在做艰难的个人决定时会向同村村民寻求建议，包括家庭成员等；不同村落间村民联系较少，或者根本就没有联系，只有少量异村村民之间互相寻求建议和帮助。我们能够挖掘图中丰富的社会结构信息，如识别村落中的意见领袖，判断

家族成员之间是否存在等级结构等。复杂网络方法提供了一个新的分析视角和强大的分析工具来处理关系型数据，能洞察更深刻的网络结构信息和语义信息。社会网络分析是社会科学领域常用的方法和工具，得到了大量社会学家认可和广泛使用，用于挖掘蕴含于网络拓扑结构中的社会信息。

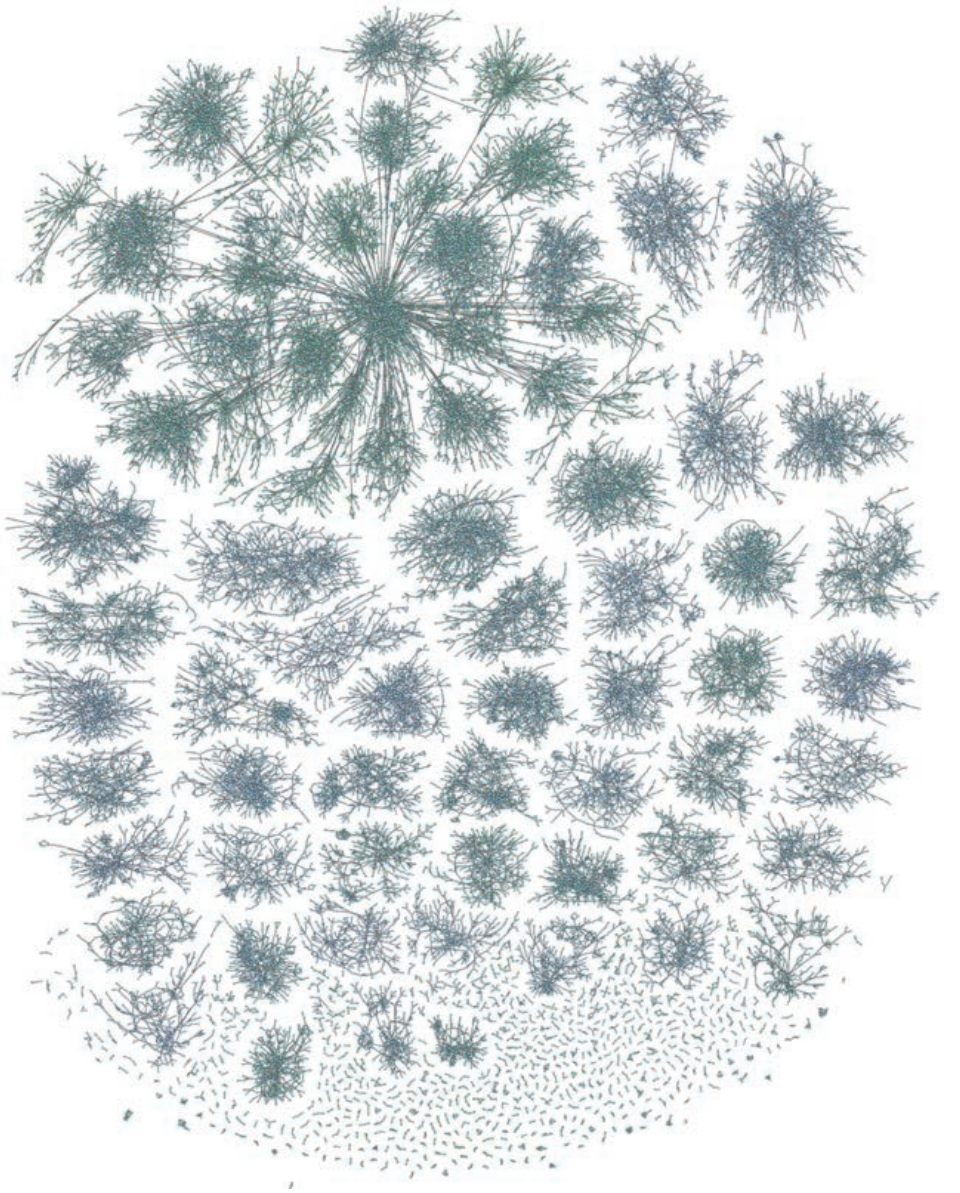


图 1.4　73 个村庄社会网络示意图

图1.4中的网络拓扑结构也很好地展示了“网络的网络”这一概念。社会网络中每一个小团体对应一个村庄中成员之间的联系网络，位于图1.4上部的不同村庄（社团）之间也存在着较为紧密的联系，中间位置处存在一个较为明显的中心节点，或称为意见领袖。社会网

络中个体之间相互作用关系也依赖于其他网络，可称之为相依网络或网络的网络。斯坦利（H. Eugene Stanley）教授等人为“网络的网络”提供了一个创新性的理论分析框架[62,63]。学者们研究网络中社会舆论、个体偏好等行为的传播过程，发现个体的相似性、互补性、同质性、流行性等因素都直接影响了个体的决策和社会系统的宏观演化规律。复杂社会网络的形成和演化同样受到环境因素的影响，如何捕捉到更加细致且更加有效的影响因素，如何更好地重现网络演化特征和规律，并融合网络的网络等方法模拟网络动力学特征，仍然是一个具有挑战的问题。

在现实世界中，网络形成和演化过程更加复杂，个体决策变量更加繁杂。复杂环境下个体决策行为受到诸多因素影响，如流行性和相似性等[91]。在金融交易系统中，如何表示投资者属性和环境因素仍然是一个重要的问题。在复杂决策环境中，影响主体决策的属性因素各不相同，很多因素因为隐私和采集难度问题而不可量化。近年来，一些基于机器学习的优化算法直接从复杂金融网络拓扑结构中解构市场行为信息和个体行为信息[92]。机器学习模型也能够学习复杂环境下个体之间网络关系的形成和演化模型[93]，将网络连边的建立过程表示成异质主体之间的博弈过程，网络主体考虑大量影响博弈均衡的因素，融合机器学习优化算法直接学习网络演化过程。很多学者运用智能算法研究网络演化过程中网络连边预测问题，取得了大量的研究成果[94-96]。

1.5　智能体建模

智能体作为智能决策的主体，如何处理复杂环境的不可观察性、随机性、连续性、不可知等特征性质，是智能系统建模的关键。

1.5.1　典型决策系统模型框架

智能体模型的主要功能是信息处理和智能决策，环境越复杂，对智能体要求就越高。一般来说，部分可观察的环境比完全可观察环境复杂，多智能体环境比单智能体环境复杂，随机性环境比确定性环境复杂，延续式环境比片段式环境复杂，动态环境比静态环境复杂，连续型环境比离散型环境复杂，未知环境比已知环境复杂。

在典型的决策支持系统中，我们从复杂环境中采集海量数据，进行建模和分析，模型输出指标和决策信息，决策者基于模型输出信息进行决策和行动，如图1.5所示。图1.5中模型的构建过程依赖于人类的领域知识（Domain Knowledge）和建模能力，模型优劣受人为因素影响较大。如果智能体能够自己调整模型架构或者模型参数，适应环境演化和经验数据，那么决策系统将更具有效性和稳定性，且模型的自动化程度将更高。

1.5.2　智能体建模框架

本节对智能体建模的几个模块分别进行介绍。

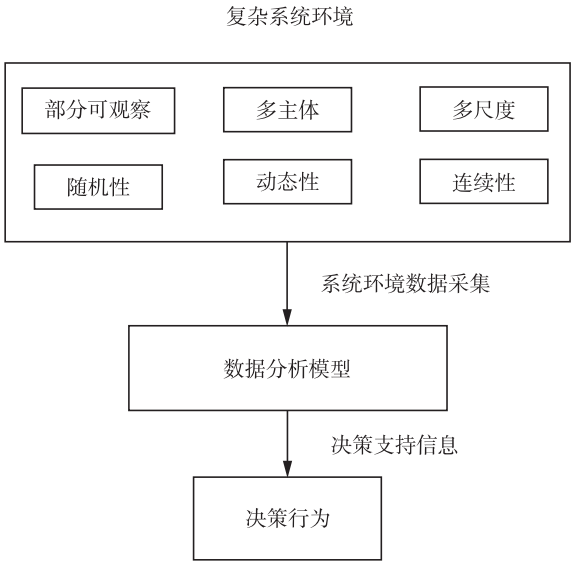


图 1.5　典型决策支持系统

1. 智能体建模框架示意图

图1.6给出了智能决策系统建模的框架示意图。其中的复杂系统环境包括了部分可观察、随机性、连续性、多主体等属性。决策智能体建模包含了几个关键组成部分，分别为感知模块、评价模块、学习模块和决策模块。在智能决策系统建模过程中，对各个模块进行了合理的抽象，模块之间能够进行信息通信和行为交互，智能体整合不同模块信息，优化各个模块的性能，这个过程将逐步提高智能策略性能。我们将对决策智能体的各模块进行简单介绍，为构建复杂智能决策系统提供基本的建模思路。

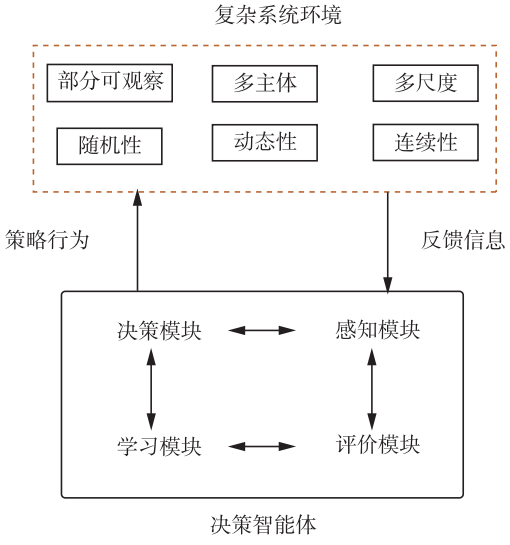


图 1.6　智能决策系统建模框架示意图

2. 感知模块

感知模块直接获得环境状态信息，感知环境的状态特征和环境的反馈信息。近年来，深度学习技术蓬勃发展，科学家们提出了众多深度学习模型，适应于不同的数据类型和复杂环境。感知模块将环境反馈和环境状态进行重新表示，映射到决策智能体的决策变量空间，决策智能体在决策空间进行智能决策。这一过程可以看作一个空间变换，主要由深度学习模型完成，比如深度神经网络（Deep Neural Networks，DNN）、深度卷积神经网络（Convolutional Neural Networks，CNN）、深度循环神经网络（Recurrent Neural Networks，RNN）、深度图神经网络（Graph Neural Networks，GNN）等。深度学习领域的飞速发展，使得越来越多的优秀深度神经网络模型为智能决策系统所用，成为智能决策系统的子模块。

深度学习模型的主要功能就是进行表示学习，将决策变量映射到隐空间，隐空间变量与决策问题之间强关联。深度学习模型去除了不必要的噪声信息和不相关信息，对信息进行了过滤和压缩，使得智能模型决策更加准确和高效。在实际运用过程中，复杂环境状态数据类型具有多样性，因此感知模块的深度学习模型也具有多样性，多模态的复杂环境数据也比较常见，因此感知模块可以融合多种深度学习模型，例如在自动驾驶智能系统中，决策智能体面对的数据包括图片数据、雷达数据、音频数据等。多模态深度学习技术融合了各式各样的深度学习模型，同时对视频、图形、音频和文本等数据进行处理，提高决策系统的智能化水平，这是深度学习研究前沿之一，也是未来发展方向。

3. 决策模块

决策模块是决策智能体模型的输出模块，相较于作为输入模块的感知模块，决策模块是决策智能体进行智能决策的关键，因为智能决策系统的目标就是训练和学习一个优秀的智能决策模块。一般智能决策模块用深度学习模型进行表示，以智能感知模块的表示数据作为输入，输出一个智能动作，或者动作的概率分布等。

类似于复杂环境，决策智能体也具有多属性特征，例如，决策智能体的动作可以分成离散型和连续型，或者同时输出两类动作。离散型动作比较常见，如电子游戏中游戏手柄的操作可以建模成整数型变量；机器人研究中连续型动作运用较多，如移动速度、角度、角速度等。在实际应用中，决策智能体并非只能有一种动作输出类型，而是可以同时输出多种类型动作。在金融市场中，智能交易机器人可以用离散型变量作为动作输出，用 -1、0 和 1 分别表示卖出、持有和买入操作，而交易量可以事先确定；同样，智能交易机器人也可以用 -1 到 1 之间的实数作为模型动作输出，表示投资者仓位变化比例，智能交易机器人的决策模块输出 0.5 表示买入 50% 最大持仓量的股票。决策智能体决策模块的动作类型可以根据具体问题进行调整。

决策智能体的决策模块也可以按照输出类型分成确定性策略和随机性策略，确定性策略直接输出动作，随机性策略输出动作的概率。确定性策略和随机性策略各有优缺点，各有其适用场景。在机器人研究中，确定性策略可以直接输出机器人的速度、角度等行为动作。

4. 评价模块

决策智能体基于感知模块将环境状态变量转化成决策模块的输入变量后，智能决策模块输出的动作如何能够体现出智能，如何评价，如何优化，都需要评价模块进行度量和更新。评价模块需要设定目标函数，决策智能体通过与环境的交互不断优化目标函数，同时优化策略模块。评价模块可以独立于策略函数，对行为进行价值评估，对有价值的动作给予较高的得分，从而引导策略函数输出最优动作。评价模块融合了感知模块和决策模块的信息，为高效训练决策智能体提供辅助信息。

5. 学习模块

学习模块结合感知模块、决策模块和评价模块，设定智能体训练规则，更新感知模块、决策模块和评价模块的模型参数，迭代训练并得到最优的感知模块、决策模块和评价模块。强化学习算法是智能决策系统的重要部分，经典的强化学习算法包括时序差分（Temporal Difference）算法、Q 学习（Q-learning）算法、SARSA 算法等，也包括了深度强化学习深度 Q 网络（Deep Q Network，DQN）算法、置信阈策略优化（Trust Region Policy Optimization，TRPO）算法、近端策略优化（Proximal Policy Optimization，PPO）算法、深度确定性策略梯度（Deep Deterministic Policy Gradient, DDPG）方法、Twin Delayed DDPG（TD3）、Actor-Critic 算法等。

1.6 智能决策系统建模

智能决策系统建模框架由环境和智能体组成，融合深度强化学习的智能决策系统将更加复杂。深度强化学习的训练过程就是智能体和环境的交互过程。为了对智能决策系统进行较为全面的认识，可以对智能决策系统的建模流程进行初步了解，我们在全局认识智能决策系统的基础上，从整体到局部，细化智能系统模块，完成智能系统模型构建、训练、验证、优化和应用部署。

图1.7给出了复杂系统环境下智能决策系统建模流程框架。框架包含了建模过程的 8 个环节，即问题提炼、数据采集、模型构建、算法实现、模型训练、模型验证、模型改进和模型运用。智能决策系统建模流程中各个环节之间都可以互相影响、互相关联，系统构建过程也是各个模块循环迭代优化的过程。

图1.7中各个模块之间也能够交替进行。智能决策系统模型在构建过程中会出现新的问题，新问题需要各个模块共同更新和修正来解决，因此智能决策系统建模流程并非是流水线，需要迭代更新，并全局优化。

1.6.1 问题提炼

智能决策系统建模的首要任务是明确所要解决的问题，界定问题所涉及的概念，并对问题进行抽象和提炼。现实社会极其复杂，我们所面对的问题通常也具有较高的复杂度，问题提炼要求对复杂环境进行简化建模，对问题进行抽象，对特征变量进行表示。一般来

说，单目标决策问题要易于多目标决策问题。问题界定之初，我们需要抓住主要矛盾，忽略次要矛盾，通过合理抽象，使得模型既能够完成既定目标，又能够便于数据收集和模型实现。

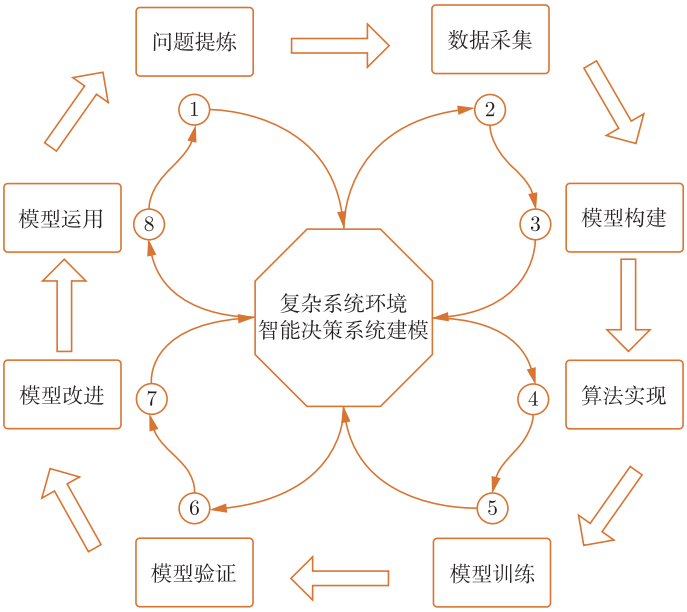


图 1.7　智能决策系统建模流程框架示意图

在智能系统建模和软件设计中，问题提炼部分可以看作需求分析，告诉开发者需要做什么，达到什么目标。对于问题的提炼切忌太理想化，设计一些不可能实现的目标，如解决准确预测金融危机这一问题，问题虽然很重要，但实现难度之大会让人觉得不切实际。

1.6.2　数据采集

数据采集是模型训练的基础，问题提炼过程中所涉及的对环境特征等变量进行表示都需要进行数据采集，采集数据的质量会影响决策智能体训练效率和模型最终绩效。在大数据时代，数据采集的途径越来越多，数据的规模越来越大，数据处理的难度也越来越高，例如数据的规模越大，所包含的噪声也越来越多，需要进行数据清洗，以提高数据质量。在数据采集过程中，我们要时刻对所提炼的问题进行审视，对数据与问题之间的关联进行分析，利用一些领域知识或者经验数据提高模型的可解释性。

高质量的数据是机器学习算法成功的关键之一。在计算机科学领域，开源代码平台中优质的深度学习模型和算法很丰富。一般而言，在实践应用过程中，开发者大部分时间和精力都在进行数据采集和数据预处理，在一些工程应用项目中，数据预处理时间可能占项目开发总时间的近 80%。

1.6.3 模型构建

模型构建包括环境模型构建和智能体模型构建。在当今开源时代，很多优秀的模型和代码库都共享在网络平台上，开发者可以互相学习，共同推进模型的发展和改进。在模型的设计过程中，开发者可以先从简单模型入手，然后通过模型的升级和迭代，逐步完善模型，提高模型性能和质量。切忌一开始就期望能够设计出完美模型，因为我们需要花大量的时间和精力在各子模块和模型细节之中。

研究人员从简单模型入手，在训练过程中发现问题、解决问题，设计新模块或者子模型，进行模型升级与迭代更新，不断循环改进。如面对时间序列预测问题，我们可以尝试一些经典的循环神经网络（RNN）模型，完成模型训练后，对模型性能进行测评，如果模型性能不理想，可以进一步考虑更高级更复杂的 LSTM 模型和 GRU 模型等。

1.6.4 算法实现

TensorFlow 和 PyTorch 等深度学习框架为算法实现提供了数量众多的可复用代码模块，面对复杂问题和复杂模型，都提供了很好的支持和实现方法，熟练地利用现有的成熟计算框架和预训练模型，能够提高模型开发效率，增强决策系统的稳定性和可靠性。由于智能决策系统建模需要编程开发能力，我们可以根据自己的编程水平选择不同的算法实现路径；初学者可以通过学习开源社区的优秀代码，完成算法实现；研究者或开发者为了精进自身算法实现能力，可以参考开源代码，手动设计代码框架，进行自主编程和算法改进，博采众长，逐步优化，完成智能决策模型的算法实现。

1.6.5 模型训练

高质量的样本数据和优秀的智能模型，是智能决策系统成功实现的基础。同时，深度学习模型的训练对硬件的要求较高，流行的计算平台 TensorFlow 和 PyTorch 框架都对分布式、GPU 等高性能计算模式进行了很好的集成，能够充分运用硬件资源，高效地完成模型训练。模型训练之初可以先设计一些简单数据集，如对完整数据集的随机抽样构建子数据集。尽量不要一开始就在完整的海量数据集上进行模型训练，不但耗费时间资源和硬件资源，出现问题难以进行问题定位、算法调试和模型改进，更重要的是，大数据集严重影响模型迭代更新效率。在模型训练和超参数调优过程中，深刻理解模型和算法原理是有效调参和高效训练模型的关键，我们不能因为开源代码的易获得性而忽略了对算法原理的理解和学习。

1.6.6 模型验证

模型训练完成后，我们需要对训练好的模型进行验证或测试，考察模型实际运用的效果，并重点关注模型的泛化性能。模型训练数据收集完成后，我们将采集到的数据分成训练集、验证集和测试集。智能模型在训练集上进行训练，在验证集上进行可信、可靠的模型验证。模型验证过程需要严格区分样本内验证和样本外验证，同时对测试集进行严格的

划分，避免因数据污染而导致验证无效。

交叉验证方法简单且易于理解，是机器学习中估计模型性能的常用验证方法。例如，K 折（K-Fold）交叉验证方法将数据集均匀拆分为 K 个子集，每次使用 $K-1$ 个数据子集作为训练集来模型训练，剩下的一个子集当作验证集进行模型验证，因此，K 折交叉验证方法需要进行 K 次模型训练。

1.6.7　模型改进

在实际建模过程中，人们很难一开始就能够完成模型的既定目标。模型优化过程如同深度强化学习中策略函数的迭代更新过程。模型训练完成后，开发人员对模型进行评估，定位问题，改进模块，此过程也是非常耗费时间、人力资源和计算资源的。

与深度强化学习类似，人类设计决策系统的过程，也是一个试错的过程。人们将模型的训练结果作为反馈信息，衡量模型优劣，针对性地改进模型，包括重新审视模型设计的每一个环节。比如问题提炼是否符合实际？数据采集是否准确？是否引入数据偏差？数据颗粒度是否合理？数据时效是否达到要求？模型设计是否可以改进？算法实现过程是否存在逻辑问题？模型超参数设定是否合理？训练过程是否充分？有无过拟合或欠拟合情况？诸多问题在模型改进的反复优化和迭代更新过程中需要进行深入全面的思考。

1.6.8　模型运用

经过验证和改进的模型可以应用和部署到现实的复杂环境之中，对模型的实际运用效果进行考察。随着环境变化与时间推移，复杂问题背景也会发生变化，我们需要时刻监控模型运行效果，及时发现模型不足，甚至可能需要重新进行问题提炼、数据采集等操作。智能系统的构建和迭代更新过程本身就是一个复杂系统演化过程，面对环境的变化而实时地学习和改进，迭代更新，可使得模型越来越智能，更好地解决现实世界的复杂决策问题。

智能决策系统的设计过程本身就是一个复杂工程，各个环节和各个子模型构成了一个复杂工程系统，环环相扣，耦合关联，相互影响。因此，任何一个小小的失误都有可能造成模型的崩塌，或导致其性能低下。为了进行高效的系统设计和模型训练，我们需要更多的策略和方法，同时也需要更多的算法理论基础知识和编程实践能力，以及深度思考和持续学习的能力。

1.7　应用实践

在复杂系统研究中，时间序列数据非常常见，时间序列在金融市场中更是十分普遍，如图1.8展示了 2017 年至 2021 年香港恒生指数的日度价格时间序列。

学者们提出多种将时间序列转化成复杂网络的方法 [97-103]，如时序网络 [104,105]、周期网络 [106,107]、最近邻网络 [108]、n-元组网络 [109]、循环网络 [110-115]、分段相关网络 [116]、可

视图网络 [117-120]、水平可视图网络 [120-128] 等。下面主要介绍一种基于可视图的时间序列转网络方法，通过复杂网络分析方法对时间序列进行分析，进而研究复杂系统的状态特征和演化规律。

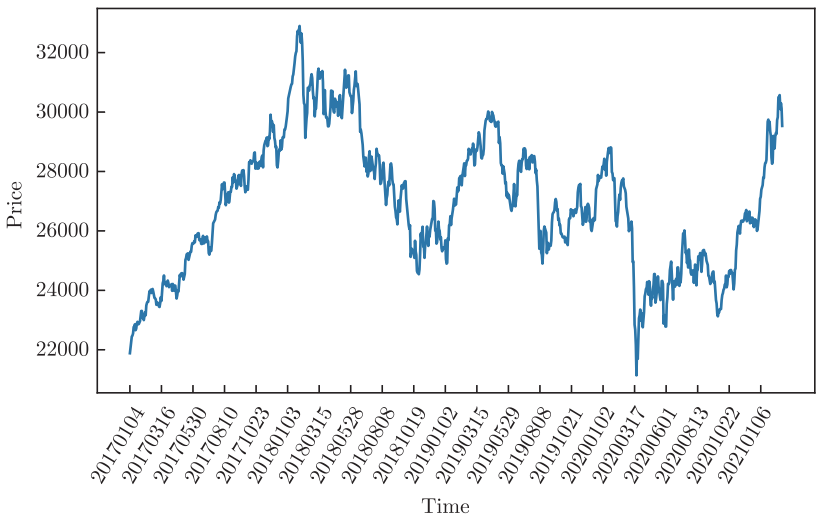


图 1.8　2017 年至 2021 年香港恒生指数的日度价格时间序列示意图

时间序列用 $\{x_i\}_{i=1,\ldots,L}$ 表示，在可视图算法中 [117]，每个时间点数据对应可视图网络节点，网络节点 i 和网络节点 j 之间的连边关系存在，必须满足以下条件：

$$\frac{x_j - x_k}{j-k} > \frac{x_j - x_i}{j-i} \tag{1.1}$$

其中，$i < k < j$。可视图网络表示成 $G = \langle V, E\rangle$，其中，集合 $V = \{v_i\}$ 表示可视图网络的节点集合，对应时间序列中数据点 x_i，连边集合表示为 $E = \{e_{ij}\}$，其元素 $e_{ij} = 1$ 表示节点 v_i 和节点 v_j 相连，说明原始时间序列中数据点 x_i 和数据点 x_j 满足公式 (1.1)。将图1.8中恒生指数时间序列转化成可视图网络，如图1.9所示，我们可以在图1.9的基础上通过网络分析方法挖掘蕴含于可视图结构之中的金融市场信息。

在复杂金融市场中，大量金融市场信息反映于金融时间序列的结构之中，时间序列反映了所有可获得的市场信息。可视图方法将时间序列转化成网络，将蕴含在时间序列中的金融市场信息转化成了网络结构信息，然后通过网络结构分析方法来挖掘出复杂金融市场信息。时间序列转化为可视图网络时，信息在一次一次的转化过程中会有所丢失，但信息存储方式的改变使得分析方法有更多的选择，可以选择一些信息挖掘能力更强的工具，如复杂网络分析方法，可以挖掘出一般算法不能获得的信息和知识，为智能决策提供更加有效的决策信息。

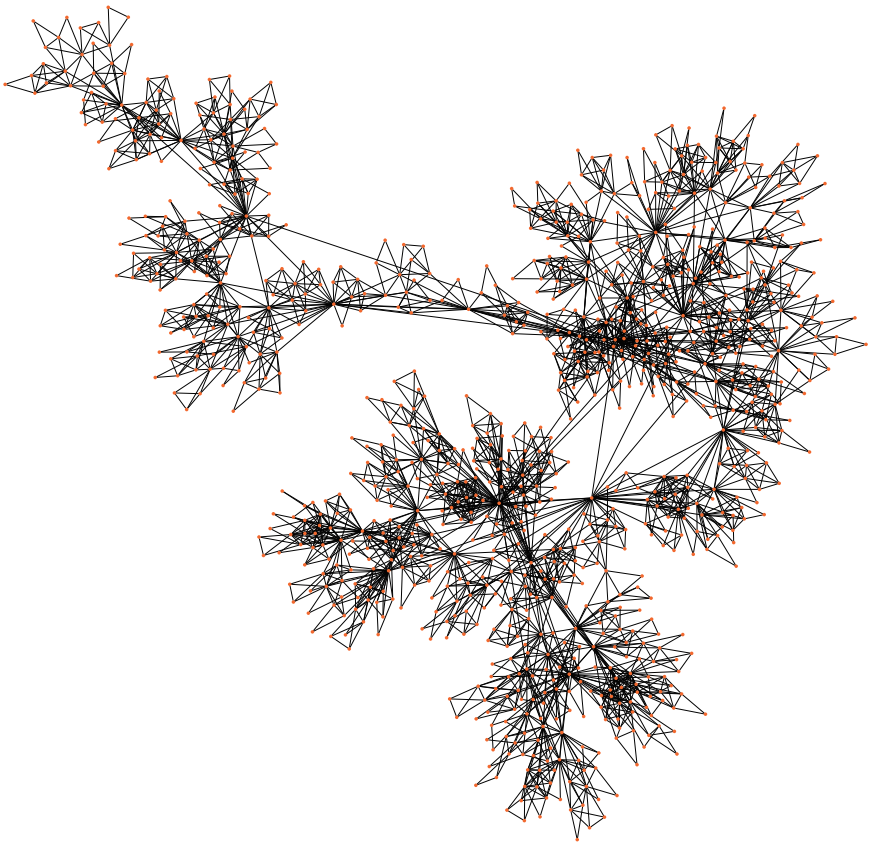


图 1.9　恒生指数时间序列转化成可视图网络示例

第 1 章习题

1. 什么是智能决策？请列举现实生活的例子。
2. 什么是复杂系统？请列举一例。
3. 复杂环境有哪些特征？
4. 什么是金融复杂性？如何刻画金融复杂性？
5. 金融复杂性的来源有哪些？
6. 什么是计算实验金融？
7. 复杂环境状态变量有哪些类型？
8. 智能决策系统建模包括哪些环节？

第 2 章

人工智能与机器学习

内容提要

- 人工智能
- 机器学习
- 机器视觉
- 自然语言处理
- 人机对话
- 智能投顾
- 智能策略
- 智能客服
- 金融科技
- 智慧金融
- 监督学习
- 无监督学习
- 半监督学习
- 自监督学习
- 强化学习
- 模仿学习
- 对抗学习

2.1 人工智能简介

人工智能（Artificial Intelligence，AI）定义很多。人工智能先驱马文 • 明斯基（Marvin Lee Minsky）认为：“人工智能就是研究‘让机器来完成那些如果由人来做需要智能的事情’的科学。”

围棋对弈被认为是人类智能的堡垒，一直是人工智能技术期望攻克的难题。AlphaGo 作为第一个战胜围棋世界冠军（李世石，2016 年）的人工智能程序，开启了又一轮人工智能浪潮 [1]。人工智能在教育、医疗、机械、商业等领域都有着广泛运用，人类从“互联网 +”和“物联网 +”走向了“人工智能 +”，出现了“人工智能 + 教育”“人工智能 + 工业”“人工智能 + 银行”“人工智能 + 科学”“人工智能 + 农业”等新兴领域。在 2021 年世界人工智能大会上，中国科学院鄂维南院士表示：“传统科学领域，如化学、材料、电子工程、化学工程、机械工程、生物工程等，才是人工智能更大的主战场。人工智能带来的不仅仅是科学研究范式的改变，也是传统行业的转型升级。”

通过百余年的发展，科学家们已经对自然科学研究领域模型有了非常深入的探索，物理模型、化学模型、生物模型已经能够与真实世界非常接近。传统科学领域引入人工智能算法（如深度强化学习）后，智能体在模拟环境中的智能策略在现实世界也有较好的迁移效果，具有较好的泛化性能，是人工智能算法在实际运用过程中非常重要的方面。智能体

交互的环境模型与现实世界的差异，直接影响了智能体的策略函数在现实世界的应用效果。物理模型、化学模型、生物模型与现实世界高度吻合，因此深度强化学习模型能有效地应用在物理、化学、生物等自然科学领域。

2.1.1　人工智能 + 农业

农业是社会的基石，是人类赖以生存的根基。随着自然环境的恶化，土壤、气候、水资源等问题对农业生产造成了不利影响。在人口增加、环境恶化的复杂环境下，农业发展是人类面临的重大难题。人工智能在农业方面有着大量应用，如人工智能育种、农业土壤管理、农业灌溉和水资源管理、肥料使用、天气预测、计算机视觉识别农作物、检测杂草、识别害虫、智能农业机械制造、自动化收割等。农业的方方面面都需要智能化工业、机械、生物学等技术的支持和发展。农业无人机技术在播种、肥料使用、农药使用方面有着广泛应用，极大提高了农业种植效率。

2.1.2　人工智能 + 教育

人工智能在教育行业也有着较多应用，如智能化的教育评价、智能化的学习资源推荐、智能化的教学过程管理等。对学生而言，人工智能技术可实现个性化学习辅助和自适应学习，对学生在学习中出现的难点和薄弱点能够自动识别和智能发现，并做到个性化地推荐学习资源，为学生制定个性化的、极具针对性的学习方案。智能系统自动给出智能化的学习路径和规划，使得学生获得更高效的学习效果。对于教师而言，人工智能系统可构建智能题库，优化整合教学资源并自动管理教学过程，使得教师可以将更多的精力放在学生的身心健康和思想品质上，更好地实现价值塑造和能力培养。

人工智能在教育行业的应用并不是替代教师，而是更好地辅助教师完成教学和人才培养的工作。“教书育人”是教师的本职工作，人工智能能够很好地辅助教师完成“教书”部分，“育人”部分需要教师投入更多的心思和精力，教师的工作不仅仅是知识的传授，现有的人工智能还不能很好地辅助教师完成学生情感、认知以及情绪相关的工作。

2.1.3　人工智能 + 工业

人工智能在现代工业中的应用可谓是由来已久。从工业革命到信息革命，工业自动化是不变的主题，自动化领域一直是人工智能发展的另一通道，也是一个持续发展的研究领域。自动化控制、最优化控制等都是人工智能领域的重要技术和方法。

在工业领域中，人工智能技术包含分析技术（Analytics Technology）、大数据技术（Big Data Technology）、云或网络技术（Cloud or Cyber Technology）、专业领域知识（Domain Knowledge）以及证据（Evidence），这些技术近些年迅速发展。证据是指在工业应用中，收集工业数据与关联证据，改进人工智能模型，迭代更新，与时俱进，更好地完成工业任务。人工智能工业应用与智能复杂系统建模类似，需要收集数据、建立模型、实施验证、迭代更新，包括数据更新、模型更新等。

2.1.4 人工智能 + 金融

近年来，人工智能技术在银行业得到了蓬勃发展和广泛应用。随着人工智能相关技术的发展，自然语言处理、图像识别、深度强化学习算法和智能推荐算法等人工智能技术为银行提供了大量的智能机器人，应用于产品推广、客户呼叫、客户服务、智能选股、智能投顾等领域。在信用卡领域中，大量人工智能系统应用于业务推荐、客户信誉评级、智能催收等场景。互联网金融、大数据金融、金融科技、科技金融等都是人工智能技术应用的重要领域。

2.2 人工智能前沿

近年来，AlphaGo、AlphaStar、AlphaFold、GPT 系列等人工智能系统一次一次惊艳了世界，加速了人工智能技术和方法的蓬勃发展 [1,2]。世界各地顶级研究机构和知名大学科研人员设计了大量人工智能和机器学习算法，在医疗、教育、工业、农业、金融等领域得到了广泛应用。社会学领域科学家也运用人工智能算法对复杂社会经济系统进行了大量研究，一些经典机器学习算法在各个领域的普及程度和接受度都比较好。

人工智能技术蓬勃发展，也伴随着其他的质疑。模型和方法的可靠性、可解释性、安全性、是否符合伦理等问题越来越受到各个领域的专家学者的重视 [135]。

随着深度学习技术的兴起，深度神经网络模型的可解释性受到了各领域专家的质疑，特别是人文社科领域的专家学者。线性回归、决策树、随机森林等具有较好的可解释性，但深度神经网络模型的黑盒性质，在很多应用领域都受到了限制，如在经济领域中，一些可解释性较弱的机器学习模型的应用存在一定局限性，但是学者们运用机器学习方法仍取得了很多研究成果 [129-134]，也催生了大量针对深度学习模型可解释性的研究和工具 [93]。

各个领域专家学者在应用人工智能技术解决复杂问题的过程中也遇到了不小的挑战 [136]，特别是人文社会科学领域中，模型可解释性至关重要。在一些商业应用中，对可解释性的要求可以稍微宽松一些，如推荐系统中只要模型结果能够提高销售额度和流量，对算法可解释性可不做过多要求。深度学习模型得益于深度神经网络的出色表示学习能力，能提高预测模型的精度，受到了各个领域的专家学者和行业人员青睐。但深度神经网络模型的黑箱问题使得经济学研究中模型预测结果和模型的经济学含义得不到合理解释 [137]，模型的可信度和应用受到限制，因此任何技术和方法的两面性都值得研究人员的关注和探究。

随着大数据和高性能计算的普及，人们已经不仅仅满足于分析大数据的相关关系 [138]，而是更加期望对复杂系统的动力学过程以及演化机理进行探究，以更好地理解和管控复杂系统的极端行为和异常现象，挖掘复杂系统和复杂模型更多的因果关系和演化机制 [93,139]。社会治理、经济治理、危机防控等问题，都需要人工智能算法提供更加可靠和可信的解决方案，也需要解决方案更加公平、公正和透明。中美贸易战 [140,141]、经济体破产、全球新型冠状病毒感染等社会经济危机事件，亟需各个学科的交叉融合和科学家们深度的跨学科合作 [83]，专家学者们需要更加细致地分析和建模复杂环境，博采众长，为智能决策提供更

加可靠的分析方法 [142, 143]。

2.3　人工智能简史

1950 年，马文 • 明斯基和邓恩 • 埃德蒙一起建造了世界上第一台神经网络计算机。同年，图灵提出了“图灵测试”。图灵测试是指测试者在与被测试者（一个人和一台机器）隔开的情况下，通过一些装置（如键盘）向被测试者随意提问，测试进行多次后，如果机器让平均每个参与者做出超过 30% 的误判，那么这台机器就通过了测试，并被认为具有人类智能 [144]。

1956 年，达特茅斯学院举行了长达一个月的会议，创造了一个时髦的新词——人工智能（Artificial Intelligence, AI），人工智能迎来了第一个春天，达特茅斯会议的主题是用机器来模仿人类学习以及其他方面的智能。参会人员是人工智能领域的顶级专家和元老，包括约翰 • 麦卡锡（John McCarthy）、马文 • 明斯基（Marvin Minsky）、克劳德 • 香农（Claude Shannon）、艾伦 • 纽厄尔（Allen Newell）、赫伯特 • 西蒙（Herbert Simon）等科学家。1956 年也成为了人工智能元年。参会人员中的克劳德 • 香农（Claude Shannon）是信息论创始人，赫伯特 • 西蒙（Herbert Simon）是诺贝尔经济学奖得主 [145]。

20 世纪 70 年代，著名数学家拉特希尔向英国政府提交了一份关于人工智能的研究报告。报告尖锐地指出人工智能看上去宏伟的目标根本无法实现，各国政府和机构也停止或减少了资金投入，人工智能在 20 世纪 70 年代陷入了第一个“寒冬”。

20 世纪 80 年代，Hopfield 网络、神经网络反向传播算法和专家系统让人工智能再次兴起，其中值得一提的是反向传播（Back Propagation，BP）算法。1986 年，深度学习之父、2018 年图灵奖得主 J. Hinton 和他的合作者运用反向传播算法训练神经网络。BP 算法是目前训练人工神经网络最常用的算法，在基于深度神经网络模型的监督学习中，BP 算法占据核心地位。反向传播算法描述了如何利用误差信息，从最后一层（输出层）开始到第一个隐藏层逐步调整深度神经网络模型权值参数，达到训练深度神经网络的目的。受限于当时计算机的算力和数据，这一轮人工智能浪潮持续时间有限。20 世纪 90 年代，人工智能进入第二个“寒冬”。

进入 21 世纪后，深度神经网络模型以深度学习的形式再次回归科研界和工业界，人工智能进入第三个春天。2016 年，DeepMind 的 AlphaGo 横空出世，AlphaGo 是第一个击败人类职业围棋选手，第一个战胜围棋世界冠军的人工智能系统。AlphaGo 由 DeepMind 公司戴密斯 • 哈萨比斯领衔的团队开发，主要作者是 David Silver，其工作原理主要是“深度强化学习”和蒙特卡洛树搜索，AlphaGo 之后深度强化学习闪耀登场。2017 年，谷歌下属公司 DeepMind 在国际学术期刊 Nature 上发表的一篇研究论文报告了新版围棋程序 AlphaGo Zero，在无人类先验知识（围棋棋谱数据）的训练下，运用自我博弈，能够迅速自学围棋，并以 100:0 的战绩击败“前辈”AlphaGo。2018 年，DeepMind 提出了 Alpha Zero 和 AlphaFold。2020 年，DeepMind 的第二代 AlphaFold 在国际蛋白质结构预测竞赛（CASP）中获得冠军。第二代 AlphaFold 能够基于氨基酸序列精确地预测蛋白质 3D 结构，

其准确性能与使用冷冻电子显微镜（CryoElectron Microscopy）、核磁共振或 X 射线晶体学等实验技术解析 3D 结构相媲美。

人工智能近 70 年的发展取得了辉煌成就。人工智能浪潮起起落落，春寒交错，只有经历过才知道。图2.1展示了人工智能发展历程的简要历史。伴随着人工智能的飞速发展，人们不禁要问：人工智能发展的终极目标是什么？虽然人工智能讨论的主体是用机器来模仿人类学习以及其他方面的智能，但是如何确定人工智能目标的完成情况？一般来说，现在大家都比较接受的观点是通用人工智能（Artificial General Intelligence，AGI）的实现。AGI 具有一般人类智慧，包括了推理、学习、记忆等基本的人类能力，可以执行人类执行的智力任务。通用人工智能是人工智能研究的主要目标，也将通用人工智能称为强 AI（Strong AI）或者完全 AI（Full AI），与弱 AI（Weak AI）相比，强 AI 可以尝试执行全方位的人类认知任务。从现如今的人工智能发展来看，离通用人工智能的目标还相距甚远，路漫漫其修远兮。

图 2.1　人工智能发展历程简史

2.4　人工智能流派

人工智能近 70 年的发展，融合了诸多学科的知识和思想。人工智能流派分类有多种，一般来说人们习惯将人工智能分为三个主要学派：符号主义（Symbolicism）、联结主义

（Connectionism）和行为主义（Actionism）[145]，如图 2.2 所示。

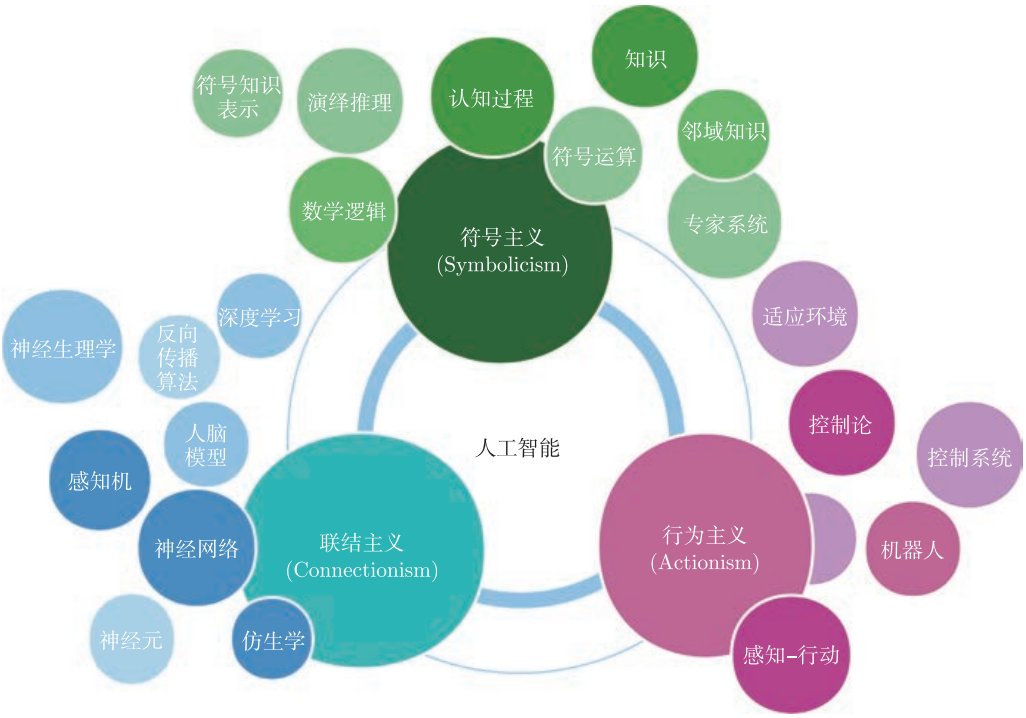


图 2.2　人工智能流派

2.4.1　符号主义学派

人工智能发展早期阶段（20 世纪 50 年代至 20 世纪 70 年代），人们基于符号知识表示和演绎推理技术取得了很大成就。符号主义学派认为人工智能源于数理逻辑，将人类认知和思维过程抽象成符号运算系统，认知过程就是在符号表示上的数学运算。数理逻辑是数学的一个分支，其研究对象是将证明和计算这两个直观概念进行符号化的形式系统。符号主义学派的专家学者受数理逻辑影响较大。

符号主义学派的代表人物有赫伯特 • 西蒙（Herbert Simon）、纽厄尔（Newell）和尼尔逊（Nilsson）等，其中，赫伯特 • 西蒙是世界上唯一一位同时获得过图灵奖和诺贝尔经济学奖的科学家。20 世纪 50 年代，赫伯特 • 西蒙、纽厄尔和约翰 • 肖（John Shaw）一起，成功设计了世界上最早的启发式程序“逻辑理论家”，证明了数学名著《数学原理》第 2 章中的 38 个定理。1963 年，经过改进后的“逻辑理论家”可证明全部 52 个定理。赫伯特 • 西蒙也被认为是符号主义学派的先驱，“逻辑理论家”也开创了机器定理证明这一新的科学领域。

国内外各领域专家学者在机器定理证明领域做出了卓越贡献，如计算机科学家提出的命题逻辑判定算法，中国智能科学研究的开拓者和领军人、首届国家最高科学技术奖获得者吴文俊院士提出的初等几何和微分几何定理机器证明的理论和方法。吴文俊开创的数学机械化在国际上被誉为“吴方法”，由中国人工智能学会（Chinese Association for Artificial

Intelligence，CAAI）发起、经科学技术部核准设立的“吴文俊人工智能科学技术奖”是中国历史上第一次以“人工智能”命名的奖项。

2.4.2 联结主义学派

联结主义学派的代表性人物是生理学家麦卡洛克（McCulloch）和数理逻辑学家皮茨（Pitts）。皮茨等人提出的以感知机（Perceptron）为基础的脑模型是现代人工智能的基础，基于仿生学的思想模拟大脑神经元以及神经元之间的联结，研究神经网络（Neural Network）模型和脑模型。1957 年，罗森布拉特（Rosenblatt）提出的感知机分类算法，是支持向量机（Support Vector Machines，SVM）和神经网络的基础。通过将多个感知机组成一层网络，将多层感知机的神经网络互相连接，叠加成最终的多层神经网络，一般称作多层感知机（Multi-Layer Perceptron，MLP）。多层感知机是一种经典的前馈人工神经网络模型。随着层数的增加，神经网络模型的表示学习能力越来越强，训练的难度也越来越大。

在深度学习发展中，大部分系统基于深度神经网络模型。深度神经网络模型中的深度是指模型叠加的神经元层数非常之多，一些神经网络的层次已经可以达到上千层，主要得益于鲁梅尔哈特（Rumelhart）等人提出的反向传播（BP）算法和近年来提出的残差网络（ResNet）模型等。在实际应用中，深度神经网络模型并不需要太多层，能够满足问题要求即可，合适的才是最好的。

图2.3展示了一个 5 层的神经网络，每一层都由互不关联的神经元组成，只有相邻层之间的神经元才有连接。图2.3左边第一层为神经网络模型输入层，中间 3 层为隐藏层，最右一层为输出层。为使深度学习模型适应不同的数据类型，科学家们发展了各式各样的深度神经网络模型，如深度前馈神经网络（Feedforward Neural Networks，FNN）、深度卷积神

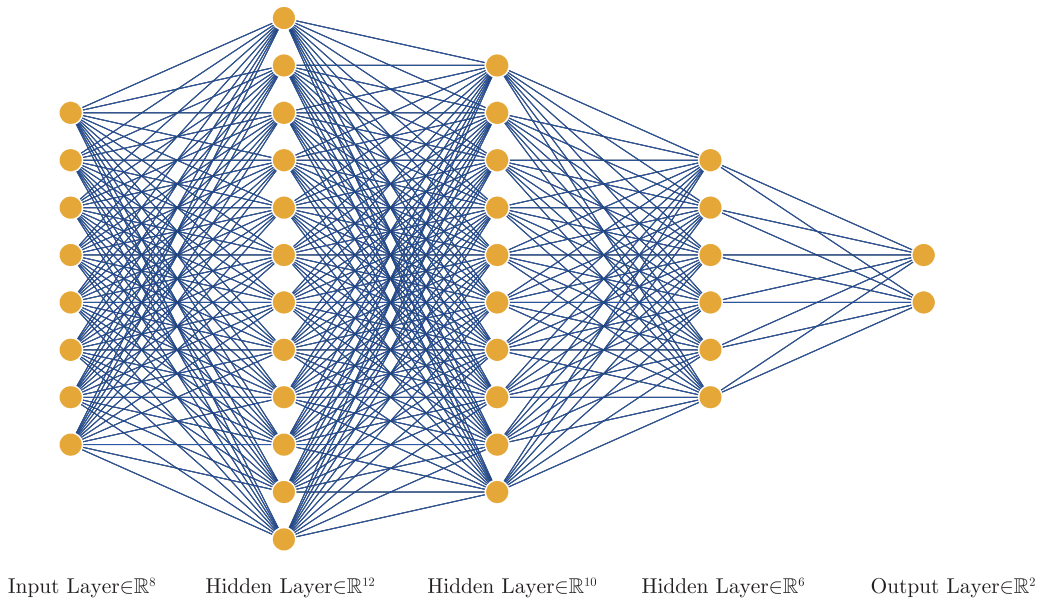


图 2.3　神经网络模型

经网络（Convolutional Neural Networks，CNN）、深度循环神经网络（Recurrent Neural Networks，RNN）、深度图神经网络（Graph Neural Networks，GNN）。众多优秀的深度学习模型是这一波人工智能浪潮的强力助推器。

2.4.3　行为主义学派

20 世纪末，行为主义引起了许多人的兴趣，行为主义思想在 20 世纪 50 年代已经成熟，主要得益于控制论的发展，控制论代表人物维纳（Norbert Wiener）在该领域做出了巨大贡献。控制论和自组织系统以及钱学森等人提出的工程控制论和生物控制论、复杂巨系统等，都为人们研究复杂系统提供了深刻的思想洞见，影响了许多领域，在不同的学科得到了广泛应用。控制论本身也是一个交叉学科，融合了不同领域的思想和方法，控制论把神经系统的工作原理与信息理论、控制理论、逻辑以及计算机联系起来，模拟人类在控制过程中的智能行为和作用。基于“感知—行动”的智能行为模拟方法，智能体能够适应环境变化，并基于环境状态做出智能决策。深度强化学习方法与行为主义学派有着极强的联系。

行为主义学派关注智能体在复杂动态环境中的最优化策略。智能体基于环境状态能够在环境中自主进行决策，决策行为作用于环境，影响着环境，同时也影响了智能体自身的状态，智能体与环境共同演化和发展。智能体与环境交互的过程是一个动态的迭代过程，智能体的决策行为不由人类事先设定好，智能体能够学习和进化自身行为策略，在学习中演化，在演化中学习，最终达到最优化目标。深度强化学习训练智能体过程就是一个与复杂环境交互的过程，有效提升和高效训练智能体是深度强化学习算法的关键。

2.5　人工智能基础

人工智能学科是众多理论和技术的结合体，横跨了多个学科领域，交叉融合，整合优化，互相促进，共同发展。人工智能技术一直以来是计算机学院的学科，是融合了计算机、数学、物理学、心理学、认知科学、哲学等多学科的交叉学科。图2.4简单列举了 10 个相关的基础理论和技术领域，但是人工智能学科不仅仅局限于这些领域。周志华教授领导的南京大学人工智能学院推出了《南京大学人工智能本科专业教育培养体系》，此书是南京大学探索人工智能本科专业人才培养方面的初步成果，也是国内第一本人工智能本科专业教育培养体系著作。

要深入理解和完全掌握人工智能技术，深厚的理论基础知识是非常必要的，扎实的理论基础知识决定了未来发展的高度。我们将简单介绍几个常用的相关学科的基本理论、技术和概念。图2.4列举了自然语言处理、机器人技术、机器学习、计算机视觉、自动化、最优化控制、运筹学、模式识别、深度学习、生物识别等技术，其中，机器学习是人工智能最关键的技术，也融合了众多基础学科的理论和技术。很多人工智能专业包含了众多的先修课程，包括计算机公共必修课、数学与自然科学基础课、数据结构与算法、计算机组成原理、计算机操作系统、程序设计基础、最优化算法、计算机视觉与模式识别、自然语言处理、计算机网络、数据库原理及应用、机器学习、分布式并行计算、数字逻辑、脑与认知科学等课程。数学是入门人工智能

的必经之路，也是一条快捷的通道，所以我们从数学角度理解和学习人工智能算法和机器学习技术，能够触及问题的本质和核心，是深刻理解模型的关键核心。

图 2.4　人工智能部分基础理论和技术

2.5.1　运筹学

运筹学是管理学中的核心基础课程。在管理学中，管理即决策，由此可见运筹学也是智能决策研究的基础性理论课程。几十年来运筹学的发展已经涵盖线性规划、非线性规划、整数规划、组合规划、图论、决策分析、排队论、可靠性数学理论、博弈论等分支，其中，图论和博弈论等可以算作独立的学科。

运筹学发展了非常之多的算法和方法来进行智能决策，比如动态规划方法和强化学习方法都能够解决序贯决策问题，但是一些高维空间或者复杂动态环境问题，容易出现“维数灾难”问题，同时动态规划方法在没有环境模型的情况下也极具挑战。大规模组合优化问题也是深度强化学习方法的应用领域。

2.5.2　最优化控制

最优化控制理论是和强化学习理论融合度非常之高的基础理论，很多强化学习论著中的模型和符号都是从最优化控制理论中借鉴而来的。在给定的约束条件下，最优化控制是要寻求一个控制信号 $u(t)$，使得给定的系统性能指标达到极大值或极小值。比如，考虑线性动力学方程：

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \tag{2.1}$$

其中，向量 $\boldsymbol{x} = [x_1, ..., x_N]^{\mathrm{T}}$ 表示系统中 N 个组成部分的状态值，$A \in \mathbb{R}^{N\times N}$ 表示了系统中 N 个组成部分之间的相互作用关系，$\boldsymbol{u}(t)$ 是控制信号。线性动力学方程描述了系统

的演化规律，控制信号 $\boldsymbol{u}(t)$ 将系统状态变量 $\boldsymbol{x}(t)$ 控制到给定的状态，并同时满足约束条件，或使得性能指标函数最大化或最小化。在强化学习中，$\boldsymbol{u}(t)$ 就是智能体输出的行为动作，也可表示成 $\boldsymbol{a}(t)$。状态在强化学习中一般用 $\boldsymbol{s}$ 表示，而非 $\boldsymbol{x}$。在后续深度强化学习算法介绍过程中，我们将采用常用的强化学习术语，而非控制论相关术语。

2.5.3　交叉学科

2020 年全国研究生教育会议决定，新增交叉学科作为新的学科门类，成为我国第 14 个学科门类。面对突飞猛进的信息技术水平和社会环境变化，如何应对变化莫测的自然、社会和人文环境，需要融合不同学科的优势，研究应对策略。高校需要面向未来复杂环境培养创新人才，通过学科交叉、校企合作、流程优化、体制改革等管理举措，整合校内外各种资源，推动前沿性、引领性甚至颠覆性的高水平创新，完成高校在科研攻关与人才培养方面的重要使命。

北京师范大学教育学部高等教育研究院杜瑞军指出：“知识的组织、探索、发现过程越来越昂贵，必须需要国家的投入，仅仅依靠个人，或者某一个组织很难实现。通过设立学科门类，有利于国家根据学科门类组建队伍、建立平台、投入资源。”复杂性科学和人工智能科学等跨学科领域为交叉学科发展提供了参考经验。

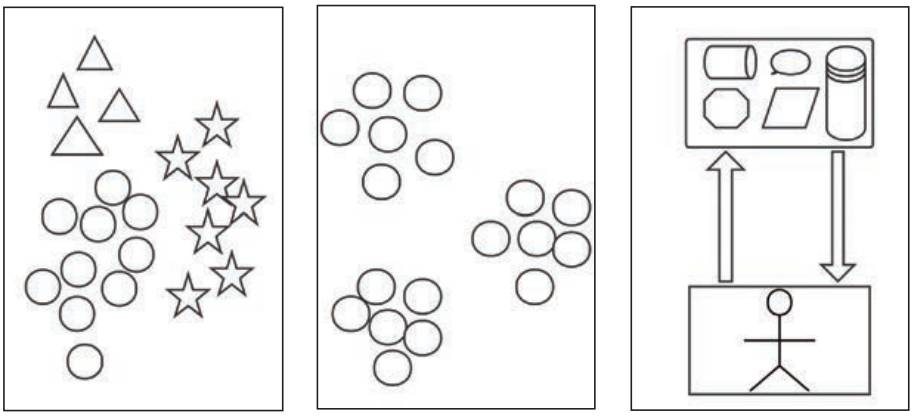
2.5.4　人工智能和机器学习相关会议

国际人工智能联合会议（International Joint Conference on Artificial Intelligence, IJCAI）是人工智能领域的顶级综合会议。国际人工智能协会（The Association for the Advancement of Artificial Intelligence，AAAI）每年主办人工智能领域最有影响的学术会议之一“AAAI Conference on Artificial Intelligence”，前身为非盈利学术研究组织——美国人工智能协会，研究人员和科学家展示各自专业领域中的新成果和新思想，也是人工智能领域的顶级综合会议。

神经信息处理系统大会（Neural Information Processing Systems，NeurIPS）、国际机器学习会议（International Conference on Machine Learning，ICML）、国际表征学习大会（International Conference on Learning Representations，ICLR）都是国际领先的机器学习大会，是公认的深度学习领域国际顶级会议，这些大会关注机器学习、深度学习等各个方面的前沿研究，并且在人工智能、机器视觉、语音识别、文本理解等重要应用领域发布了众多有影响力的论文。人工智能和机器学习的大多数会议都有官方网站，会有接收论文列表，如 NeurIPS （neurips.cc），可以快速查阅相关论文。

2.6　机器学习分类

机器学习分类方法有很多种，按照学习任务类型可以将机器学习分成三类：监督学习、无监督学习和强化学习，如图2.5所示。



（a）监督学习　　（b）无监督学习　　（c）强化学习

图 2.5　机器学习分类示意图

如图2.5（a）所示，监督学习的样本具有明显的分类信息（形状），在算法构建之前，数据中已标记了样本所属类别，如三角形、圆形和五角星。监督学习的模型在训练过程中需要学会如何将新的无标记的样本分类到正确的分类之中，因此监督学习模型也需要能够准确分类已标注的训练样本。

如图2.5（b）所示，无监督学习的样本数据没有明显的分类信息（形状）。无监督学习算法需要分析样本之间的相关性或者距离关系并进行聚类，聚类的目标是使相同类别的样本尽可能相似，而不同类别样本间的差异尽可能大。因此，无监督学习算法能发现样本数据内在的关联结构和分类属性。

如图2.5（c）所示，强化学习的样本需要智能体与环境进行交互才能获得。强化学习算法的目标函数是最大化智能体与环境交互所获得的累积收益。

总地来说，机器学习三类算法都是从数据中学习，只是指导学习的信号不一样：监督学习有明确的监督信息，即样本标签信息；无监督学习则需要机器学习算法探索数据内在结构，找到合适的分类或样本属性特征；强化学习中智能体在与环境的交互过程中收集样本数据（包括反馈信号）进行学习。

学习机器学习之前需要理解一些问题。机器学习从哪里学？机器学习学什么？机器学习怎么学？一般来说，机器学习算法都是基于大数据，因此机器学习模型是从样本数据中学习，学习数据中隐含的规律和结构，我们再将样本数据的内在结构或隐含规律建模成数学公式或函数形式。具体而言，机器学习算法将数据信息编码至设定的函数参数之中。

机器学习算法学到的是模型参数或者函数参数，所以机器学习模型可以看作函数模型。函数就是一个映射，从输入数据映射到输出数据，比如分类任务的监督学习模型就是将样本属性变量映射到标签变量的函数。机器学习模型通过学习到的函数模型可以表征数据规律和重现数据内在结构。至于如何学习，机器学习算法涉及很多技术细节，比如神经网络的反向传播（BP）算法等。在这一部分，我们将简单介绍机器学习算法。

我们用数学语言描述机器学习的过程，就是建模或学习一个函数映射的过程：

$$\boldsymbol{y} = f_{\boldsymbol{w}}(\boldsymbol{x}) \tag{2.2}$$

其中，f 是模型，也是函数，$\boldsymbol{w}$ 是模型参数，机器学习的目标就是从给定的样本数据 $(\boldsymbol{x}, \boldsymbol{y})$ 中学习到模型参数 $\boldsymbol{w}$。当然，如此简单的描述并不严格，但这是为了更容易理解机器学习过程而做的必要的简化。

2.6.1　监督学习

监督学习是人工智能和机器学习技术落地运用最广泛的学习算法。在一些需求比较明确的现实场景中，我们能够较好地模型化和参数化映射关系，且目标函数比较好确定。一般来说，监督学习的数据形式比较规整，如 $\{(x_k, y_k)\}_{k=1,2,3,\ldots,N}$ 所示，其中，N 表示样本数量，监督学习是构建一个函数映射，将样本特征数据 $\boldsymbol{x}$ 映射到标签数据 $\boldsymbol{y}$。常见的监督学习任务可分为回归和分类。我们为了衡量监督学习算法的效果，构建基于均方误差（Mean Square Error）的目标函数：

$$\mathcal{L}(\boldsymbol{w}) = \frac{1}{N}\sum_{k=1}^{N}\left(f_{\boldsymbol{w}}(x_k) - y_k\right)^2 \tag{2.3}$$

该公式度量了标记数据 y_k 与模型预测 $f_{\boldsymbol{w}}(x_k)$ 之间的差异。均方误差目标函数并非监督学习中唯一的目标函数形式。均方误差一般适用于回归问题，对于分类任务，监督学习可采用交叉熵损失（Cross Entropy Loss）函数作为目标函数 $\mathcal{L}(\boldsymbol{w})$。

一般来说，机器学习算法需要函数模型 $\boldsymbol{y} = f_{\boldsymbol{w}}(\boldsymbol{x})$ 能够在训练集上很好地拟合样本数据 $(\boldsymbol{x}, \boldsymbol{y})$，即函数模型的预测值 $f_{\boldsymbol{w}}(x_k)$ 与真实值 y_k 差距越小越好，目标函数 $\mathcal{L}(\boldsymbol{w})$ 越小越好。我们确定好目标函数之后，通过优化算法最小化目标函数 $\mathcal{L}(\boldsymbol{w})$，可以得到模型参数：

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}|(x_k, y_k)_{k=1,2,3,\cdots,N}) \tag{2.4}$$

机器学习模型完成监督学习后，我们可以在验证集和测试集上对模型 $f_{\boldsymbol{w}}$ 进行验证和测试，验证和测试后的模型可以进行模型预测和模型生成等应用。在经典的多元统计分析中，监督学习的例子包括线性回归、判别分析等。

2.6.2　无监督学习

一般来说，无监督学习的样本数据如 $\{x_k\}_{k=1,2,3,\cdots,N}$ 所示，其中，N 表示样本数量。经典多元统计学中有很多无监督学习的例子，如 K 均值聚类、系统聚类（层次聚类）、主成分分析（Principal Component Analysis，PCA）等。无监督学习模型可以基于下游任务，挖掘原始数据的内在结构和规律，并对原始数据进行表征，有利于下游任务的分类、回归等应用。我们将以自编码器（Auto-Encoder）为例进行简单说明。为了衡量无监督学习算法的效果，同样构建一个基于均方误差的目标函数：

$$\mathcal{L}(\boldsymbol{w}_1, \boldsymbol{w}_2) = \frac{1}{N}\sum_{k=1}^{N}\left[f_{\boldsymbol{w}_2}(f_{\boldsymbol{w}_1}(x_k)) - x_k\right]^2 \tag{2.5}$$

我们为了模型简化，可以设定：

$$f_{\boldsymbol{w}}(\boldsymbol{x}) = f_{\boldsymbol{w}_2}(f_{\boldsymbol{w}_1}(\boldsymbol{x})) \tag{2.6}$$

其中，$\boldsymbol{w} = (\boldsymbol{w}_1, \boldsymbol{w}_2)$。无监督学习虽然没有监督信号，即标记数据 y_k，但在自编码器模型中样本 x_k 将自身作为监督学习信号。自编码器模型学习一个恒等映射 $f_{\boldsymbol{w}}(\boldsymbol{x})$，即模型 $f_{\boldsymbol{w}}$ 将 $\boldsymbol{x}$ 映射到自身，满足：

$$\boldsymbol{x} = f_{\boldsymbol{w}}(\boldsymbol{x}) \tag{2.7}$$

因此，$f_{\boldsymbol{w}}$ 叫作自编码器。自编码器模型结构的简单示例如图2.6所示。

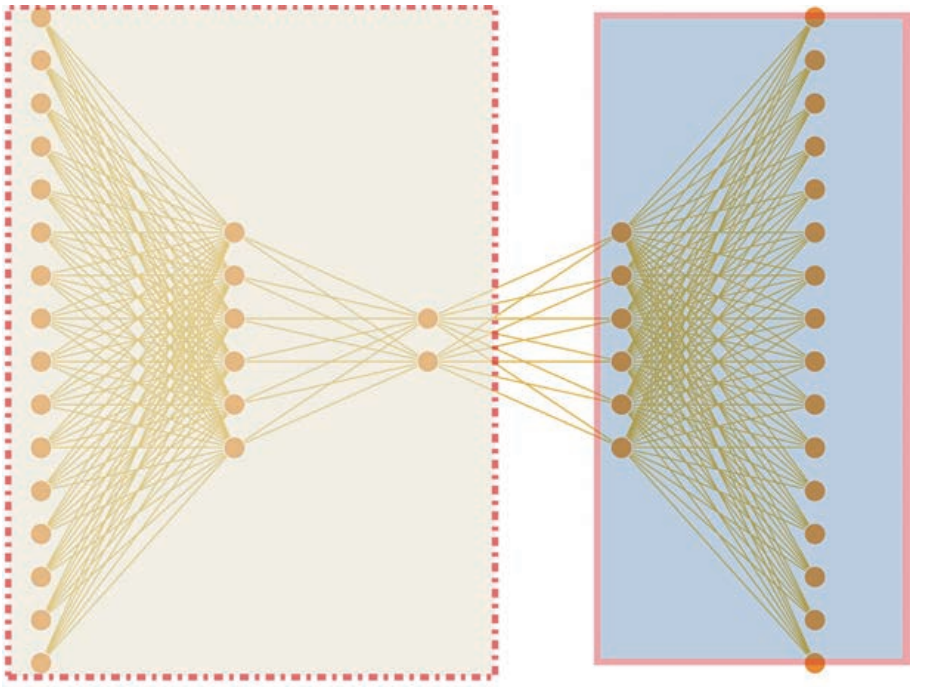


图 2.6　自编码器模型结构示意图

图2.6中自编码器输入参数向量大小为 16，输出同样是大小为 16 的向量。自编码器模型一共分成了 5 层，第一层为输入层，最后一层为输出层，中间三层为隐藏层。自编码器模型的性能由各层神经元数量和连接权重决定。

同样，目标函数可以改写成：

$$\mathcal{L}(\boldsymbol{w}) = \frac{1}{N}\sum_{k=1}^{N}\left(f_{\boldsymbol{w}}(x_k) - x_k\right)^2 \tag{2.8}$$

确定好目标函数之后，我们通过优化方法最小化目标函数，可以得到模型参数：

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}|(x_k)_{k=1,2,3,\cdots,N}) \tag{2.9}$$

自编码器模型完成了无监督学习后，获得了两个子模型 $f_{\boldsymbol{w}_1}$ 和 $f_{\boldsymbol{w}_2}$，其中，模型 $f_{\boldsymbol{w}_1}$ 可以看作学习模型，对应的函数值 $y_k = f_{\boldsymbol{w}_1}(x_k)$ 可以是原始数据 x_k 的特征表示。模型 $f_{\boldsymbol{w}_2}$ 可以看作生成模型，对应的函数值 $x_k = f_{\boldsymbol{w}_2}(y_k)$ 可以作为基于给定数据 y_k 生成的新样本数据。

图2.6虚线框中的自编码器模型部分可以表示函数 $f_{\boldsymbol{w}_1}$，实线框中的模型部分可以表示函数 $f_{\boldsymbol{w}_2}$。函数 $f_{\boldsymbol{w}_1}$ 的输出为一个二维向量，模型函数 $f_{\boldsymbol{w}_1}$ 将样本数据从十六维空间映射到二维空间，即将一个十六维向量重新表示为二维向量，完成了数据降维和数据压缩的操作。模型函数 f_{w_2} 重新将二维的特征表示还原成了十六维向量。自编码器模型训练好后，我们可以将模型进行拆分，获取函数 f_{w_1} 的输出作为样本特征变量输入下游任务，如分类和回归。

自编码器还有很多其他应用，比如降噪。从全局视角来看，自编码器模型只是学习到了一个恒等映射，即

$$\boldsymbol{x} = f_{\boldsymbol{w}}(\boldsymbol{x}) = f_{\boldsymbol{w}_2}(f_{\boldsymbol{w}_1}(\boldsymbol{x})) \tag{2.10}$$

如果我们给原始数据加上噪声 ϵ，得到 $x_k + \epsilon$，那么将其重新代入模型中后构建目标函数：

$$\mathcal{L}(\boldsymbol{w}) = \frac{1}{N}\sum_{k=1}^{N}[f_{\boldsymbol{w}}(x_k + \epsilon) - x_k]^2 \tag{2.11}$$

运用优化算法最小化目标函数，即可得到具有降噪功能的模型 $f_{\boldsymbol{w}}$ 或 $f_{\boldsymbol{w}_2}(f_{\boldsymbol{w}_1}(x))$。自编码器模型 $f_{\boldsymbol{w}}$ 将具有噪声的数据 $\boldsymbol{x} + \epsilon$ 还原成了 $\boldsymbol{x}$，去掉了噪声 ϵ。监督学习和无监督学习的思想和方法将一直贯穿机器学习的始末，特别是在理解和应用强化学习算法中，监督学习思想同样具有重要价值。

2.6.3　强化学习

强化学习模型不同于监督学习和无监督学习，强化学习模型主要解决序贯决策问题。序贯决策问题是指目标函数值需要通过一系列关联的动作来确定，不是简单地通过一次或者互不关联的行为确定。与监督学习和无监督学习类似，强化学习算法也是学习一个映射函数，即策略函数，策略函数将环境状态空间映射到动作空间。强化学习过程更加贴合人类学习过程，智能体通过与环境的交互来迭代优化策略函数。

图2.7给出了深度强化学习与经典机器学习、深度学习的异同。对于模型输入图片，经典机器学习能够识别出“老虎!”。深度学习得益于深度神经网络模型强大的表征能力，能够对图片进行更加细致的识别，模型输出“强壮的老虎!”。深度强化学习除了融合深度学习强大的感知能力，更加侧重于决策能力，能够输出决策行为，如图2.7所示，模型输出“强壮的老虎，快跑!”。

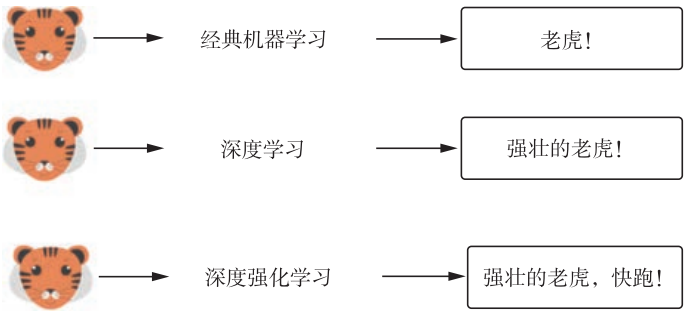


图 2.7　深度强化学习与经典机器学习、深度学习的异同

2.7　机器学习基础

监督学习、无监督学习和强化学习都用到了机器学习中一些基础概念和分析方法。我们先了解一些机器学习的基本概念和操作，然后深入理解监督学习、无监督学习及强化学习的算法原理，最后熟练运用和改进机器学习算法。本节将简单介绍机器学习中常用的激活函数、损失函数、优化算法等，为深刻理解深度强化学习夯实基础[146,147]。

2.7.1　激活函数

机器学习模型可理解成函数映射，一般不是简单函数或常见的线性函数，如多元线性回归模型，而是非线性函数模型，或者是包含了非线性函数的嵌套函数等，其中，激活函数是机器学习模型的重要组成部分。激活函数一般为非线性函数，是机器学习模型非线性特征的主要来源，增强了模型的特征表示能力，常见的激活函数包括 sigmoid 函数、tanh 函数、整流线性单元（Rectified Linear Unit, ReLU）函数等。

图2.6中神经网络模型的输入参数为 $\boldsymbol{x}$，$\boldsymbol{x}$ 是一个列向量，大小为 $n_{\boldsymbol{x}}=16$。神经网络模型下一层隐藏层有 $n_{\boldsymbol{h}}=6$ 个神经元，隐藏层中每个神经元都基于 $n_{\boldsymbol{x}}=16$ 个输入参数计算数值：

$$\boldsymbol{h}=\sigma(\boldsymbol{W}\cdot\boldsymbol{x}+\boldsymbol{b}) \tag{2.12}$$

其中，$\boldsymbol{W}$ 为输入层和隐藏层之间的参数矩阵，大小为 $n_{\boldsymbol{h}}\times n_{\boldsymbol{x}}$, 偏置项 $\boldsymbol{b}$ 的大小为 $n_{\boldsymbol{h}}$, σ 为非线性激活函数。

1. sigmoid 函数

激活函数是机器学习中常见操作算子，常见的激活函数是 sigmoid 函数，数学函数表示如下：

$$\text{sigmoid}(x)=\frac{1}{1+\mathrm{e}^{-x}} \tag{2.13}$$

$\text{sigmoid}(x)$ 函数输出值介于 0 到 1 之间，既可以表示概率，也可以做分类标识，具有非常多的优良性质。神经网络模型计算梯度时，$\text{sigmoid}(x)$ 函数的导数为

$$\text{sigmoid}'(x)=\left(1-\frac{1}{1+\mathrm{e}^{-x}}\right)\frac{1}{1+\mathrm{e}^{-x}}=(1-\text{sigmoid}(x))\,\text{sigmoid}(x) \tag{2.14}$$

因此，sigmoid(x) 函数的导数仍然是 sigmoid(x) 函数的函数，在最优化过程中梯度计算较为简便。

2. tanh **函数**

激活函数 tanh(x) 为非线性函数，具体形式如下：

$$\tanh(x) = \frac{\mathrm{e}^{x} - \mathrm{e}^{-x}}{\mathrm{e}^{x} + \mathrm{e}^{-x}} \tag{2.15}$$

tanh(x) 函数在机器学习模型中也经常使用，tanh(x) 函数输出值介于 -1 到 1 之间。同样，tanh(x) 函数具有较多优良性质，适合很多机器学习模型。

3. 整流线性单元函数

在深度学习模型中，整流线性单元（ReLU）函数备受青睐，定义如下：

$$f(z) = \begin{cases} 0, & z \leqslant 0 \\ z, & z > 0 \end{cases} \tag{2.16}$$

ReLU 函数的优势显而易见，极其简单，却有着深刻含义。相较于 sigmoid(x) 函数和 tanh(x) 函数，ReLU 函数的非线性特征并不明显，只是一个分段函数，但是在一些机器学习任务中 ReLU 函数表现出了较好的性能，如图像识别任务等。ReLU 函数的导数同样极其简单：

$$f'(z) = \begin{cases} 0, & z < 0 \\ 1, & z > 0 \end{cases} \tag{2.17}$$

但 ReLU 函数在 0 点处不可导。

4. Leaky ReLU 函数

在 ReLU 函数的定义中，当变量小于 0 时，函数的导数为 0，在深度学习中容易产生梯度消失的困境，因此，人们用 Leaky ReLU 函数解决此问题，具体形式如下：

$$f(z) = \begin{cases} \alpha z, & z \leqslant 0 \\ z, & z > 0 \end{cases} \tag{2.18}$$

Leaky ReLU 函数的参数 α 为一个很小的正数。Leaky ReLU 函数的导数同样极其简单，满足：

$$f'(z) = \begin{cases} \alpha, & z < 0 \\ 1, & z > 0 \end{cases} \tag{2.19}$$

Leaky ReLU 函数在 0 点处也不可导。在实际使用过程中，我们很难确定不同激活函数的优劣，可以通过替换不同激活函数进行尝试，具体问题具体分析，选择合适的激活函数。

5. softmax 函数

深度神经网络模型最后一层（输出层）一般不使用上述激活函数，模型输出层经常使用 softmax 函数进行归一化，使得神经网络模型输出一个概率分布向量，具体形式如下：

$$f(z_i) = \frac{\mathrm{e}^{z_i}}{\sum_{k=1}^{N} \mathrm{e}^{z_k}} \tag{2.20}$$

其中，N 为输出层神经元数量，i 为输出层神经元编号。神经网络输出层为一个概率分布向量，适合在分类任务中使用。

在一般机器学习模型训练过程中，激活函数可看作一个超参数。我们可以设定不同的激活函数来分析模型效果，确定最优的激活函数。机器学习算法中激活函数不仅包括上述几种类型，还有很多变种，适用于不同的机器学习问题，各有优缺点。

2.7.2 损失函数

在机器学习中，监督学习算法学习样本数据来拟合模型参数，而参数的优化效果需要目标函数度量。在大部分情况下，监督学习模型中的目标函数用损失函数表示，强化学习中的目标函数用累积收益函数表示。

1. 均方误差损失

一般而言，我们用神经网络模型的输出值和目标值之间的差异来构造损失函数，损失越小，模型的输出值和目标值之间的差异越小，说明模型参数越好。监督学习模型的损失函数可以表示如下：

$$\mathcal{L}(\boldsymbol{w}) = \frac{1}{N} \sum_{k=1}^{N} \left(f_{\boldsymbol{w}}(x_k) - y_k\right)^2 \tag{2.21}$$

其中，y_k 为目标值，样本 x_k 为神经网络模型输入，$\boldsymbol{w}$ 为神经网络模型参数，模型预测值为 $f_{\boldsymbol{w}}(x_k)$，而损失函数则为目标值 y_k 和预测值 $f_{\boldsymbol{w}}(x_k)$ 之间的误差平方和，最后进行平均。损失函数式 (2.21) 称作均方误差。

我们从距离定义的角度可以认为，均方误差是预测值 $f_{\boldsymbol{w}}(x_k)$ 与目标值 y_k 之间欧氏距离的平方和的均值。我们基于不同的距离定义能够得到不同的损失函数，其中闵可夫斯基（Minkowski）距离（闵氏距离）可定义如下：

$$\mathcal{D}_q(x_i, x_j) = \left[\sum_{k=1}^{p} (x_{ik} - x_{jk})^q\right]^{\frac{1}{q}} \tag{2.22}$$

式中，x_i 和 x_j 为 p 维空间中的数据点。当 $q = \infty$ 时，即为切比雪夫（Chebychev）距离：

$$\mathcal{D}_\infty(x_i, x_j) = \max_{1 \leqslant k \leqslant p} |x_{ik} - x_{jk}| \tag{2.23}$$

当 $q=2$ 时，即为欧氏距离：

$$\mathcal{D}_2(x_i,x_j)=\left[\sum_{k=1}^{p}(x_{ik}-x_{jk})^2\right]^{\frac{1}{2}} \tag{2.24}$$

当 $q=1$ 时，即为布洛克（Block）距离：

$$\mathcal{D}_1(x_i,x_j)=\sum_{k=1}^{p}|x_{ik}-x_{jk}| \tag{2.25}$$

距离公式在模型正则化时也经常用到。

2. 平均绝对误差损失

布洛克距离公式可以用来定义平均绝对误差（Mean Absolute Error, MAE）损失函数：

$$\mathcal{L}(\boldsymbol{w})=\frac{1}{N}\sum_{k=1}^{N}|f_{\boldsymbol{w}}(x_k)-y_k| \tag{2.26}$$

均方误差和平均绝对误差损失函数都可以用来作为目标函数，优化模型参数更新。均方误差具有较好的性质，如可导，所以均方误差损失函数方便计算梯度下降所需要的偏导数，而平均绝对误差损失函数在一些数据点不可导。

3. 交叉熵损失

在机器学习分类问题中，研究人员偏好交叉熵（Cross Entropy Loss，CEL）损失函数作为目标函数，使用非常广泛。为了深入了解交叉熵损失函数的意义，我们先介绍一个相关的概念，叫作 Kullback-Leibler 散度（KL-divergence）。KL 散度在强化学习中有着较多应用，如在生成对抗网络（Generative Adversarial Networks，GAN）中 KL 散度也具有重要作用。KL 散度衡量两个概率分布 $P(x)$ 和 $Q(x)$ 之间的距离（不相似度）：

$$D_{\mathrm{KL}}(P||Q)=\mathrm{E}_{x\sim P}\left[\log\frac{P(x)}{Q(x)}\right]=\mathrm{E}_{x\sim P}\left[\log P(x)-\log Q(x)\right] \tag{2.27}$$

两个概率分布之间的相似性越大，其距离 $D_{\mathrm{KL}}(P||Q)$ 就越小。当 $P(x)=Q(x)$ 时，相似性最大，KL 散度为 0。

在机器学习分类问题中，模型输出为样本属于不同类别的概率分布。我们通过将模型输出的概率分布和真实的概率分布进行对比，计算两个分布之间的距离，并最小化模型输出的概率分布和真实的概率分布之间的距离，使得模型预测概率越来越准确。

KL 散度作为两个概率分布的距离需要满足非负性，也就是说 KL 散度必须大于或等于 0。学习机器学习相关理论和方法时，我们理解一个公式最好的方式就是证明它或者证明公式的一些重要性质，因此，我们简单证明 KL 散度大于或等于 0，证明过程中需要用到一个简单的不等式：

$$\log x\leqslant x-1\quad(x>0) \tag{2.28}$$

我们将不等式运用于 KL 散度定义中：

$$\begin{aligned} D_{\mathrm{KL}}(P||Q) &= \mathrm{E}_{x\sim P}\left[\log\frac{P(x)}{Q(x)}\right] \\ &= \mathrm{E}_{x\sim P}\left[-\log\frac{Q(x)}{P(x)}\right] \\ &= -\int P(x)\left[\log\frac{Q(x)}{P(x)}\right]\mathrm{d}x \\ &\geqslant -\int P(x)\left[\frac{Q(x)}{P(x)}-1\right]\mathrm{d}x \\ &= -\int Q(x)\mathrm{d}x+\int P(x)\mathrm{d}x \\ &= -1+1=0 \end{aligned} \tag{2.29}$$

当 $P(x)=Q(x)$ 时，对应于 $x=1$，即式 (2.28) 中的等号成立，此时 KL 散度为 0，即两个概率分布 $P(x)$ 和 $Q(x)$ 的距离为 0。

深入理解 KL 散度定义，我们可以发现公式：

$$\begin{aligned} D_{\mathrm{KL}}(P||Q) &= \mathrm{E}_{x\sim P}\left[\log P(x)-\log Q(x)\right] \\ &= \mathrm{E}_{x\sim P}\log P(x)-\mathrm{E}_{x\sim P}\log Q(x) \end{aligned} \tag{2.30}$$

对于给定的 $P(x)$，第一项 $\mathrm{E}_{x\sim P}\log P(x)$ 是一个常数项，即为 -1 乘上概率分布 $P(x)$ 的熵：

$$H(P)=-\mathrm{E}_{x\sim P}\log P(x)=-\int P(x)\log P(x)\mathrm{d}x \tag{2.31}$$

因此，KL 散度所定义的概率分布 $P(x)$ 和 $Q(x)$ 之间的距离大小，关键在于第二项 $\mathrm{E}_{x\sim P}\log Q(x)$，即为 $P(x)$ 和 $Q(x)$ 之间的交叉熵 $H(P,Q)$。

在连续情况下，交叉熵 $H(P,Q)$ 定义如下：

$$H(P,Q)=-\mathrm{E}_{x\sim P}\log Q(x)=-\int P(x)\log Q(x)\mathrm{d}x \tag{2.32}$$

因此，最小化交叉熵损失函数就是最小化 KL 散度距离，也就是最小化概率分布 $Q(x)$ 与 $P(x)$ 之间的差异。我们通过调整 $Q(x)$ 的概率分布使得 $Q(x)$ 与 $P(x)$ 之间距离最小，$Q(x)$ 与 $P(x)$ 也相应地最相似。

在离散情况下，K 分类问题的交叉熵公式定义如下：

$$H(P,Q)=-\sum_{k=1}^{K}P(x_k)\log Q(x_k) \tag{2.33}$$

其中，$Q(x)$ 可以看作机器学习模型输出的概率分布，$P(x)$ 为真实的概率分布。最小化交叉熵损失函数就是最小化 KL 散度距离，机器学习模型优化算法通过调整估计的概率分布

$Q(x)$（调整和优化模型参数），使得估计分布 $Q(x)$ 与真实分布 $P(x)$ 距离最小，则 $Q(x)$ 与 $P(x)$ 分布越相似。估计分布 $Q(x)$ 与真实分布 $P(x)$ 越一致，则机器学习模型分类越准确。

在机器学习模型训练集中有 N 个样本的情况下，K 分类问题的交叉熵公式定义如下：

$$H(P,Q)=-\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}P(x_{ik})\log Q(x_{ik}) \tag{2.34}$$

在机器学习模型中，一般分类问题将交叉熵公式作为模型损失函数，即目标函数。机器学习模型对估计分布函数 $Q(x)$ 进行模型化和参数化，如建模成深度神经网络，参数为 $\boldsymbol{w}$，则 K 分类问题的交叉熵损失函数可写作：

$$\mathcal{L}(\boldsymbol{w})=-\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}P(x_{ik})\log Q_{\boldsymbol{w}}(x_{ik}) \tag{2.35}$$

我们通过最优化方法，找到最优参数 $\boldsymbol{w}$ 使得损失函数最小：

$$\hat{\boldsymbol{w}}=\arg\min_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}) \tag{2.36}$$

机器学习模型完成模型训练后，分类模型 $Q_{\boldsymbol{w}}(\boldsymbol{x})$ 可以估计新样本 $\boldsymbol{x}$ 属于 K 个分类的概率，并完成新样本分类任务。

4. 二分类交叉熵损失函数

深入理解交叉熵损失函数的性质和意义十分重要，我们以二分类任务（$K=2$）为例来分析交叉熵损失函数。数据样本中的每个样本 x_i 对应一个标签 y_i。如果标签 $y_i=0$，说明样本 x_i 属于第一类；如果 $y_i=1$，说明样本 x_i 属于第二类。机器学习算法预测样本 x_i 属于第一类的概率为 $\hat{y}_{i,0}$，属于第二类的概率为 $\hat{y}_{i,1}$，显然 $\hat{y}_{i,0}+\hat{y}_{i,1}=1$。因此，机器学习算法预测样本 x_i 属于第一类和第二类的概率也可以分别表示为 $\hat{y}_{i,0}$ 和 $1-\hat{y}_{i,0}$，将其代入交叉熵损失函数公式可以得到二分类问题的损失函数：

$$\mathcal{L}=-\frac{1}{N}\sum_{i=1}^{N}[y_i\log\hat{y}_{i,0}+(1-y_i)\log(1-\hat{y}_{i,0})] \tag{2.37}$$

其中，N 为训练样本数量。在计算时，我们规定了 $0\log 0=0$。由于 y_i 取 0 或者 1，对于一个样本 x_i，不论 $\hat{y}_{i,0}$ 为何值，式 (2.37) 中，$y_i\log\hat{y}_{i,0}$ 和 $(1-y_i)\log(1-\hat{y}_{i,0})$ 总有一项为 0。当 $\hat{y}_{i,0}=y_i$ 时，不论 $y_i=0$ 或 $y_i=1$，交叉熵均为 0，此时正好对应式 (2.35) 中 $P(x)=Q(x)$ 的情况，表明模型的预测与现实情况完全一致。

我们还可以从另一个角度对式 (2.37) 进行分析，深入理解二分类问题交叉熵定义。若以随机变量 Y 表示样本所属类别，样本属于第一类则 $Y=0$，样本属于第二类则 $Y=1$。因此，随机变量 Y 服从 0-1 分布（两点分布），也称伯努利分布，具体公式如下：

$$P(Y=y)=p^y(1-p)^{1-y} \tag{2.38}$$

其中，$y=0$ 或者 $y=1$。$y=0$ 表示样本属于第一类，$y=1$ 表示样本属于第二类。p 表示样本属于第一类的估计概率，因此属于第二类的估计概率为 $1-p$。我们为了估计参数 p，基于最大似然估计方法可知，样本最大似然概率为

$$\mathcal{L}_{\mathrm{MLE}}=\prod_{i=1}^{N}\hat{y}_i^{y_i}(1-\hat{y}_i)^{(1-y_i)} \tag{2.39}$$

式中，$\hat{y}_i$ 即为 p 的待估计值。我们对最大似然函数取对数后可得

$$\mathcal{L}=\log\mathcal{L}_{\mathrm{MLE}}=\sum_{i=1}^{N}y_i\log\hat{y}_i+(1-y_i)\log(1-\hat{y}_i) \tag{2.40}$$

由此可见，交叉熵与最大似然估计也存在着对应关系。从概率论角度来说，两者是等价的。我们为了理解一个理论，不仅仅需要证明它或者证明理论公式的一些重要性质，还可以从不同的角度进行理解和分析，能够获得更加透彻的理解和深刻的启发。

2.7.3 优化算法

机器学习模型融合了激活函数等非线性函数，而目标函数则度量了模型预测或回归结果的优劣。机器学习模型确定损失函数后，损失函数包含机器学习模型的参数 θ，参数估计过程就是一个典型的优化问题，最小化目标函数或损失函数 $\mathcal{L}(\theta)$，估计最优参数 θ^*，使得 $\mathcal{L}(\theta^*)$ 最小，图2.8给出了简化版的模型参数优化过程示意图。

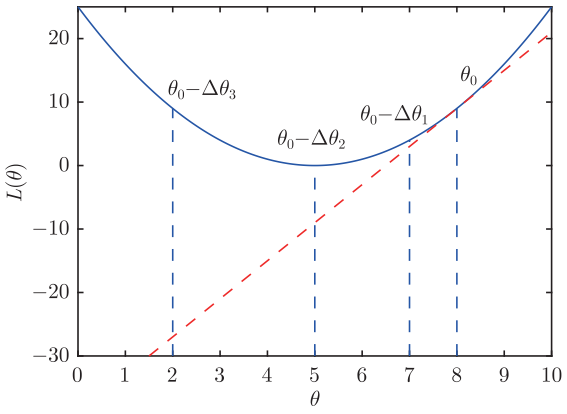


图 2.8　机器学习模型参数优化过程示意图

图2.8中的 $\mathcal{L}(\theta)$ 为损失函数，模型参数为 θ。图2.8示例做了简化，模型参数 θ 是标量，一维空间的最优化问题简单易懂，一般实际应用中，模型参数空间是超高维空间，大多属于非凸函数的优化问题。图2.8中 $\mathcal{L}(\theta)=(\theta-5)^2$，是凸函数，函数只有一个最小值，只需要对损失函数进行求导，导数为零的参数即最优参数 $\theta^*=5$。在实际应用中，高维参数空间的极值点不止一个或者目标函数为非凸函数，目标函数在高维空间中存在大量的高峰和低

谷结构，存在着大量的极大值点和极小值点，因此模型训练和参数优化过程异常复杂，绝大部分情况下机器学习模型只能获得一个局部最优点。

在机器学习模型优化过程中，梯度是一个非常重要的概念。若函数 $f(x,y,z)$ 为三元函数，则函数在点 (x,y,z) 处的梯度为如下向量：

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right) \tag{2.41}$$

其中，偏导数在点 (x,y,z) 处都可计算，梯度 ∇f 为目标函数 f 在此处变化最快的方向。图2.8示例为一维空间，梯度方向即为导数方向。在求解过程中，将模型参数 θ 初始化为一个随机值，设为 θ_0，目标函数在随机初始值 θ_0 附近，按照导数方向增加参数 θ 可以使得目标函数越来越大。损失函数优化过程中，我们需要损失函数越小越好，因此参数 θ 的变化方向应该为导数或梯度的反方向，即目标函数或损失函数可以运用梯度下降算法更新参数。

在图2.8中，$\theta_0 - \Delta\theta_1$ 位置处的目标函数值 $\mathcal{L}(\theta_0 - \Delta\theta_1)$ 小于 $\mathcal{L}(\theta_0)$，因此参数向左移动 $\Delta\theta_1$ 是一个不错的移动距离。移动距离 $\Delta\theta_1$ 由机器学习中一个重要超参数学习率 α 所决定，从图中可以看出，超参数学习率 α 直接影响了目标函数 $\mathcal{L}(\theta)$ 的优化效果，如果移动距离更大一些，$\mathcal{L}(\theta_0 - \Delta\theta_2)$ 直接达到了极小值，即为最优值，如果继续增加移动距离，$\mathcal{L}(\theta_0 - \Delta\theta_3)$ 值又增加了，因此 $\theta_0 - \Delta\theta_3$ 不是一个好的参数更新操作。在机器学习模型优化过程中，合适的移动步长和精确的梯度直接影响目标函数的优化效果，而参数移动步长则直接由超参数学习率 α 所决定。超参数学习率 α 在实际优化过程中至关重要，是机器学习模型训练过程中第一个需要调整的超参数。

1. 随机梯度下降算法

机器学习模型训练过程中的关键问题为目标函数优化。我们为了训练模型，需要更新参数，优化损失函数，并基于损失函数梯度下降的方向更新参数，找到损失函数的极小值。一般而言，机器学习模型优化结果并不一定是损失函数最小值对应的全局最优解，常常只是一个局部最优解。在机器学习或深度学习模型中，随机梯度下降（Stochastic Gradient Descent，SGD）算法是最常用的参数更新算法，伪代码如 Algorithm 1所示 [148]。

Algorithm 1: 随机梯度下降算法伪代码

Input: 损失函数 $\mathcal{L}$，机器学习模型参数 $\boldsymbol{\theta}$，学习率 α，最大训练次数 S
Output: 最优参数 $\boldsymbol{\theta}^*$

```
初始化模型参数 θ
for k = 0, 1, 2, 3, ··· , S do
    随机抽样小批量样本计算损失函数 L
    通过反向传播算法计算目标函数梯度 ∇_θ L
    梯度下降方法更新参数 θ：θ_k = θ_{k−1} − α∇_θ L
返回最优参数 θ* = θ_k
```

在随机梯度下降算法的伪代码 Algorithm 1中，输入参数为损失函数 $\mathcal{L}$、模型参数 $\boldsymbol{\theta}$、学习率 α 以及最大训练次数 S 等。在随机梯度下降算法中，梯度估计是关键，很多深度学

习计算平台的核心功能就是自动求梯度。在计算梯度时，如果将全部样本代入梯度函数中计算梯度容易耗费有限的资源，影响模型更新效率；如果只代入一个样本则容易造成梯度不稳定，影响模型参数更新的有效性。所以，在实际计算中，一般采用随机抽样小批量样本计算梯度 $\nabla_{\boldsymbol{\theta}}\mathcal{L}$，并按照梯度下降来更新参数：

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} - \alpha\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L} \tag{2.42}$$

因此该算法叫作随机梯度下降算法。

在实际应用中，除了设定最大训练次数 S，还可以设定提前终止训练的规则，以节省资源，提高训练效率，并防止出现过拟合（Overfitting）。所谓提前终止（Early stopping），是指在每一轮（Epoch）训练结束时，判断验证数据集上模型精确度是否不再提高，若模型精确度不再提高就停止训练。机器学习模型的一轮训练是指遍历了所有训练数据，随机采样完成模型参数更新。随机梯度下降算法是众多梯度更新算法的基础，很多算法都是基于随机梯度下降算法的改进。

2. 动量随机梯度下降算法

机器学习模型的损失函数可以看作能量函数，最优参数对应着能量的最低点。最小化损失函数的过程可以想象成小球在高维空间的光滑曲面上滚动，寻找最低点的过程。借鉴物理系统中物体运动的动量概念，科研人员改进随机梯度下降算法，设计了包含动量的随机梯度下降算法，即动量随机梯度下降（Momentum SGD）算法，其伪代码如 Algorithm 2所示 [149]。

Algorithm 2: 动量随机梯度下降算法伪代码

Input: 损失函数 $\mathcal{L}$，模型参数 $\boldsymbol{\theta}$，学习率 α，超参数 β，最大训练次数 S
Output: 最优模型参数 $\boldsymbol{\theta}^*$

1 初始化 $g_0 = 0$
2 **for** $k = 1, 2, 3, \cdots, S$ **do**
3 　随机抽样小批量样本计算损失函数 $\mathcal{L}$
4 　通过反向传播算法计算梯度 $\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}$
5 　结合动量思想，更新梯度时叠加上一次的梯度方向：$g_k = \beta g_{k-1} - \alpha\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}$
6 　更新参数 $\boldsymbol{\theta}$：$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + g_k$
7 返回最优参数 $\boldsymbol{\theta}^*$

动量随机梯度下降算法伪代码 Algorithm 2与随机梯度下降算法的部分代码类似。为了算法描述的完整性和可读性，我们将相似部分代码也保留在伪代码 Algorithm 2之中。动量随机梯度下降算法特别之处在于第 5 行，计算梯度方向时结合动量思想，参数更新的梯度 g_k 不仅仅只受当前梯度的影响，还需叠加上一次更新的梯度方向。动量随机梯度下降算法的参数更新过程为

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} - \alpha\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L} + \beta g_{k-1} \tag{2.43}$$

动量随机梯度下降算法在更新参数 $\boldsymbol{\theta}_k$ 时，不仅受当前梯度 $\alpha\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}$ 的影响，还叠加上一次

梯度方向 g_{k-1}，类似物理学中物体运动的惯性。超参数 β 调节叠加上一次梯度方向的强弱程度，学习率超参数 α 调节学习速率。

3. Nesterov 动量随机梯度下降算法

Nesterov 动量随机梯度下降（Nesterov Momentum SGD）算法对动量随机梯度下降算法进行了改进，在计算梯度时可以使用提前位置的梯度进行更新，Nesterov 动量随机梯度下降算法伪代码如 Algorithm 3所示 [150]。

Algorithm 3: Nesterov 动量随机梯度下降算法伪代码

Input: 损失函数 $\mathcal{L}$，模型参数 $\boldsymbol{\theta}$，学习率 α，超参数 β，最大训练次数 S
Output: 最优模型参数 $\boldsymbol{\theta}^*$

1 初始化 $g_0 = 0$
2 **for** $k = 1, 2, 3, \cdots, S$ **do**
3　　随机抽样小批量样本计算损失函数 $\mathcal{L}$
4　　根据上一次迭代梯度计算临时模型参数 $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta}_{k-1} + \beta g_{k-1}$
5　　通过反向传播算法和临时参数计算梯度 $\boldsymbol{\nabla}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_{k-1} + \beta g_{k-1})$
6　　结合动量思想更新梯度，考虑上一次更新的方向：$g_k = \beta g_{k-1} - \alpha \boldsymbol{\nabla}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_{k-1} + \beta g_{k-1})$
7　　更新参数 $\boldsymbol{\theta}$：$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + g_k$
8 返回最优参数 $\boldsymbol{\theta}^*$

在实际应用中，一般使用 Nesterov 动量随机梯度下降算法较多，其收敛速度比动量随机梯度下降算法要更快一些。因为 Nesterov 动量随机梯度下降算法提前运用了更新的参数计算梯度，可以认为提前运用了更加准确的梯度信息。Nesterov 动量随机梯度下降算法关键改进之处是伪代码中第 4 行，计算梯度时根据上一次迭代梯度计算一个临时参数：

$$\bar{\boldsymbol{\theta}} = \boldsymbol{\theta}_{k-1} + \beta g_{k-1} \tag{2.44}$$

一般而言，$\boldsymbol{\theta}_{k-1} + \beta g_{k-1}$ 比 $\boldsymbol{\theta}_{k-1}$ 更接近最优参数，因此通过 $\boldsymbol{\theta}_{k-1} + \beta g_{k-1}$ 计算的梯度更加准确。在 Nesterov 动量随机梯度下降算法中，采用反向传播算法和临时参数计算的梯度 $\boldsymbol{\nabla}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_{k-1} + \beta g_{k-1})$ 加速了参数更新过程，能以更少的迭代步数达到最优参数 $\boldsymbol{\theta}^*$。

在机器学习和深度学习中，基于梯度下降的高性能优化算法的关键是尽可能准确地估计目标函数的梯度，因此很多算法改进专注于对梯度的准确估计。在借鉴很多经典的数值分析方法，即数值优化方法后，科研人员发展了大量的梯度优化算法，也改进了大量现有的梯度计算方法。

4. 自适应梯度下降算法

在随机梯度下降算法、动量随机梯度下降算法和 Nesterov 动量随机梯度下降算法中，梯度决定了参数更新的方向，超参数学习率 α 决定了参数更新的步长大小，但是如何选择合适的梯度更新步长却较为困难，即超参数学习率 α 的调优尤为关键。在实际运用过程中，数值优化方法基于梯度差异来自动调整学习率，将极大减少梯度优化算法对于学习率的依赖程度。比如我们将损失函数看作能量函数或者地貌函数，在地势平坦的地方，梯度较小，可以加大

移动步长；在地势比较陡峭的地方，梯度较大，可以减小移动步长。自适应梯度下降（Adagrad）算法基于此思想进行参数更新，算法具体细节伪代码如 Algorithm 4所示 [151]。

Algorithm 4: 自适应梯度下降算法伪代码

Input: 损失函数 $\mathcal{L}$，模型参数 $\boldsymbol{\theta}$，学习率 α，最大训练次数 S，非常小的常数 ϵ
Output: 最优模型参数 $\boldsymbol{\theta}^*$
1 初始化 $t_0 = 0$
2 **for** $k = 1, 2, 3, \cdots, S$ **do**
3 　抽样小批量样本计算损失函数 $\mathcal{L}$
4 　通过反向传播算法计算梯度 $\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_{k-1})$
5 　更新参数 t：$t_k = t_{k-1} + \left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_{k-1})\right)^2$
6 　更新参数 $\boldsymbol{\theta}$：$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} - \dfrac{\alpha}{\sqrt{t_k + \epsilon}}\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_{k-1})$
7 返回最优参数 $\boldsymbol{\theta}^*$

自适应梯度下降算法中的自适应是指学习率的自适应调整，算法的关键在于伪代码 Algorithm 4中第 6 行，更新参数公式中损失函数梯度 $\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}(\theta_{k-1})$ 前面的系数：

$$\frac{\alpha}{\sqrt{t_k + \epsilon}} \tag{2.45}$$

当梯度很大时，累积的梯度平方很大，实际的学习率 $\dfrac{\alpha}{\sqrt{t_k + \epsilon}}$ 较小，会降低参数的更新速度。公式中的 ϵ 为一个非常小的数，是为了避免梯度更新过程中出现分母为 0 的情况，其取值一般介于 10^{-4} 到 10^{-8} 之间。众多梯度下降优化算法各具优缺点，方法改进过程中可能引入新问题，如自适应梯度下降算法伪代码第 5 行中的 t 累积了梯度的平方，t_k 随着迭代步数的增加越来越大，因此参数 $\boldsymbol{\theta}$ 的更新速度 $\dfrac{\alpha}{\sqrt{t_k + \epsilon}}$ 将越来越小，直至趋近于 0 而无法更新参数。

5. RMSprop 梯度下降算法

Hinton 等人为了避免自适应梯度下降算法中梯度越来越小的问题，提出了 RMSprop 梯度下降算法，改进了 Adagrad 算法。RMSprop 梯度下降算法伪代码如 Algorithm 5所示。

Algorithm 5: RMSprop 梯度下降算法伪代码

Input: 损失函数 $\mathcal{L}$，模型参数 $\boldsymbol{\theta}$，学习率 α，超参数 γ，最大训练次数 S
Output: 最优模型参数 $\boldsymbol{\theta}^*$
1 初始化 $t_0 = 0$
2 **for** $k = 1, 2, 3, \cdots, S$ **do**
3 　抽样小批量样本计算损失函数 $\mathcal{L}$
4 　通过反向传播算法计算梯度 $\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_{k-1})$
5 　更新参数 t：$t_k = \gamma t_{k-1} + (1-\gamma)\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_{k-1})\right)^2$
6 　更新参数 $\boldsymbol{\theta}$：$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} - \dfrac{\alpha}{\sqrt{t_k + \epsilon}}\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_{k-1})$
7 返回最优参数 $\boldsymbol{\theta}^*$

RMSprop 算法不同于 Adagrad 算法之处是参数 t 的更新方式，Adagrad 算法使用梯度平方累积求和公式：

$$t_k = t_{k-1} + \left(\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_{k-1})\right)^2 \tag{2.46}$$

RMSprop 算法伪代码 Algorithm 5中第 5 行是算法关键：

$$t_k = \gamma t_{k-1} + (1-\gamma)\left(\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_{k-1})\right)^2 \tag{2.47}$$

比较二者可以发现，RMSprop 算法引入了衰减因子 γ。在梯度平方累积求和过程中，衰减因子使得 t_k 并不是一直增大，距离当前时刻越远的梯度信息衰减越明显，不容易出现 t_k 无限增大而导致梯度消失的情况，因此解决了 Adagrad 实际学习率逐渐减小的问题。衰减因子 γ 的取值通常为 0.9、0.99 或 0.999。

6. Adadelta 梯度下降算法

在梯度下降算法、动量梯度下降算法、Nesterov 动量梯度下降算法以及 RMSprop 算法的伪代码中，一些优化算法效率与学习率初始化值相关性较大且比较敏感，因此学习率 α 是关键的超参数，一些学习率会随着迭代而衰减，机器学习模型中学习率可以作为超参数进行调优。Adadelta 算法为了进一步解决超参数学习率动态调整的问题，继续进行了改进，不需要设定初始化学习率 α，具体算法伪代码如 Algorithm 6所示 [152]。

Algorithm 6: Adadelta 梯度下降算法伪代码

Input: 损失函数 $\mathcal{L}$，模型参数 $\boldsymbol{\theta}$，超参数 γ，最大训练次数 S

Output: 最优模型参数 $\boldsymbol{\theta}^*$

初始化 $g_0 = 0$

初始化 $t_0 = 0$

初始化 $\Delta_0 = 0$

for $k = 1, 2, 3, \cdots, S$ **do**

　抽样小批量样本计算损失函数 $\mathcal{L}$

　通过反向传播算法计算梯度 $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_{k-1})$

　更新参数 t：$t_k = \gamma t_{k-1} + (1-\gamma)\left(\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_{k-1})\right)^2$

　更新参数 g：$g_k = -\dfrac{\sqrt{\Delta_{k-1}+\epsilon}}{\sqrt{t_k+\epsilon}} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_{k-1})$

　更新参数 $\boldsymbol{\theta}$：$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + g_k$

　更新参数 Δ：$\Delta_k = \gamma \Delta_{k-1} + (1-\gamma) g_k^2$

返回最优参数 $\boldsymbol{\theta}^*$

Adadelta 算法的主要改进之处在于。Adadelta 算法改进了 RMSprop 算法，不需要初始化学习率 α 且参数更新公式中已经消除了学习率 α。Adadelta 算法较前面算法改进较大，其中参数更新核心公式为

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} - \frac{\sqrt{\Delta_{k-1}+\epsilon}}{\sqrt{t_k+\epsilon}} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_{k-1}) \tag{2.48}$$

公式的关键之处为梯度 $\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_{k-1})$ 之前的权重 $\frac{\sqrt{\Delta_{k-1}+\epsilon}}{\sqrt{t_k+\epsilon}}$，其分母 $\sqrt{t_k+\epsilon}$ 比较好理解，与 RMSprop 算法一致，自适应地调节学习率。当累积梯度较大时实际学习率变小，减小参数更新速度；当累积梯度较小时实际学习率变大，增加参数更新速度。权重的分子 $\sqrt{\Delta_{k-1}+\epsilon}$ 替代了 RMSprop 算法中的学习率 α，因此 Adadelta 算法中的权重 $\frac{\sqrt{\Delta_{k-1}+\epsilon}}{\sqrt{t_k+\epsilon}}$ 可以看作实际学习率。

7. Adam 梯度下降算法

众多流行的梯度下降算法以 SGD 算法为基础，Momentum SGD 算法、Nesterov Momentum SGD 算法、RMSprop 算法和 Adadelta 算法在此基础上一步一步地改进，提升了优化算法的效率和稳定性。Adam 梯度下降算法融合了诸多算法的精髓，成为了机器学习领域常用的优化算法。当然，机器学习和深度学习领域还有很多其他类型的优化算法，实际问题的复杂性使得并不存在一种适合所有优化问题的最好的优化算法。在计算条件允许的情况下，我们可以用不同的优化算法进行训练模型并进行对比分析，类似于超参数调优。

Adam 梯度更新算法的具体细节伪代码如 Algorithm 7所示 [153]。在 Adam 算法的伪代码 Algorithm 7中，计算变量增多，如 g_k^2、β_1^t 和 β_2^t 等。Adam 与 RMSProp 类似，借鉴动量随机梯度下降算法的思想，启用了两个衰减因子 β_1 和 β_2，利用了上一次的参数更新方向。在 Adam 算法中，β_1 通常取值为 0.9，β_2 取值为 0.999，ϵ 取值为 10^{-8}。

Algorithm 7: Adam 梯度下降算法伪代码

Input: 损失函数 $\mathcal{L}$，模型参数 $\boldsymbol{\theta}$，学习率 α，超参数 β_1，β_2，最大训练次数 S

Output: 最优模型参数 $\boldsymbol{\theta}^*$

初始化 $g_0 = 0$

初始化 $t_0 = 0$

初始化 $\Delta_0 = 0$

令 $t = 0$

for $k = 1, 2, 3, \cdots, S$ **do**

- $t = t + 1$
- 抽样小批量样本计算损失函数 $\mathcal{L}$
- 通过反向传播算法计算梯度 $\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_{k-1})$
- 更新 g：$g_k = \boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_{k-1})$
- 更新 m：$m_k = \beta_1 m_{k-1} + (1-\beta_1) g_k$
- 更新 v：$v_k = \beta_2 v_{k-1} + (1-\beta_2) g_k^2$
- 更新 $\hat{m}$：$\hat{m}_k = \frac{m_k}{1-\beta_1^t}$
- 更新 $\hat{v}$：$\hat{v}_k = \frac{v_k}{1-\beta_2^t}$
- 更新参数 $\boldsymbol{\theta}$：$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} - \frac{\alpha}{\sqrt{\hat{v}_k+\epsilon}}\hat{m}_k$

返回最优参数 $\boldsymbol{\theta}^*$

在梯度下降算法家族中存在大量优秀算法，且随着不断改进和升级，优化算法越来越复杂。流行的机器学习计算平台都提供了很多优化器，可以实现众多经典的随机梯度下降算法，如基础的随机梯度下降算法。SGD 算法加入动量因子改进后发展出了 Momentum SGD 以及 Nesterov Momentum SGD。自适应梯度下降算法中的学习率能够随着梯度变化而自适应调节，随后 RMSprop 梯度下降算法和 Adadelta 梯度下降算法也汲取了自适应调节学习率的思想。Adam 梯度下降算法融合了诸多算法的精髓，成为了机器学习领域中常用的算法。在实际工程应用中，具体选用何种梯度优化算法，与选择激活函数一样，可以在计算条件允许的前提下作为超参数调优，尝试不同的优化算法。

2.8　应用实践

近年来，基于机器学习的复杂网络分析方法迅速发展，网络嵌入（Network Embedding，NE）或图嵌入（Graph Embedding，GE）方法得到了大量研究者关注 [154]。网络嵌入或图嵌入方法用低维、稠密、实值的向量表示节点属性、连边属性和全局网络属性，将网络信息映射到低维稠密空间。网络嵌入或图嵌入方法也被称作网络表示学习（Network Representation Learning, NRL），相关算法很多 [154]，且各具特色，大部分方法基于矩阵分解和随机游走 [155]。在网络表示学习中，邻近节点学习到的特征具有相似的表示向量。这里我们介绍一种能够保留有向图的不对称传递性的网络嵌入算法，称为高阶邻近保留嵌入算法（High-Order Proximity preserved Embedding，HOPE）[156]。

在可视图网络中，我们可以设定连边方向，如网络连边 $i \to j$ 中包含了 $i < j$，显然 i 和 j 对应原时间序列的时间标签。我们专注于有向图非对称性嵌入问题，保留网络节点非对称传递性。高阶邻近性源自不对称传递性，采用机器学习优化算法最小化损失函数 [156]：

$$\min \ \|\mathcal{S} - \mathcal{U}^s \mathcal{U}^{t\top}\|^2 \tag{2.49}$$

$\mathcal{S}$ 是可视图网络高阶相似性测度指标值，$\mathcal{U}^s$ 和 $\mathcal{U}^t$ 是每个网络节点嵌入向量组成的嵌入向量矩阵，高阶邻近保留嵌入算法与传统的奇异值分解（Singular Value Decomposition，SVD）方法类似。

图2.9给出了可视图网络嵌入后特征向量的相似性矩阵热度图，热度图的颜色代表了相关系数值。在可视图算法转化过程中，可视图网络中的每一个节点对应恒生指数某一天，节点信息可以对应这一天的市场状态信息。我们运用 HOPE 算法进行网络嵌入后，嵌入空间中每一个点对应网络中一个节点，嵌入空间的特征向量对应某一天的市场状态特征，因此，嵌入空间中节点的相似性矩阵可以表示金融市场状态之间的相似性矩阵。

在图2.9中，相似性矩阵热度图的对角线存在明显的分块结构，说明市场存在明显的状态切换行为。随着市场演化，市场状态存在显著的差别。各类网络嵌入算法能挖掘不同网络结构特征，面对具体问题时，我们需要细致分析问题的背景和结构特征，选择合适的分析方法。

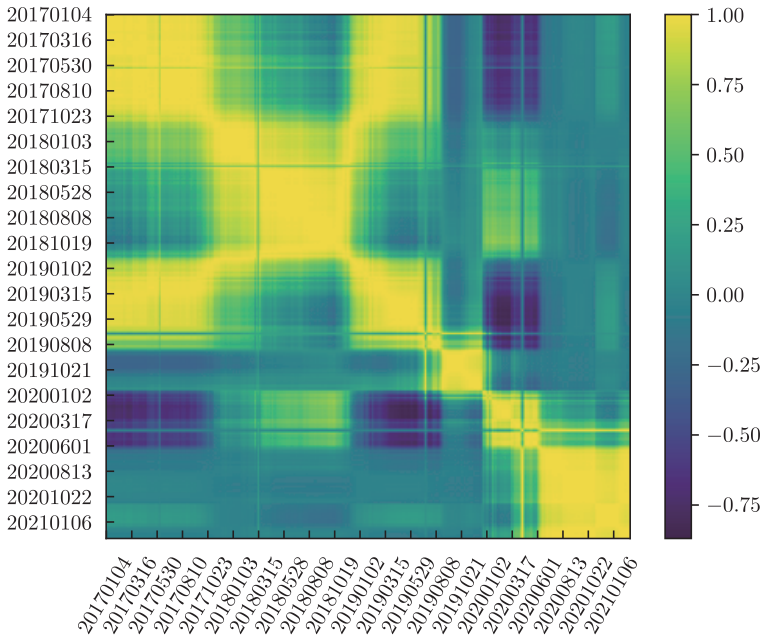


图 2.9　特征向量的相似性矩阵热度图

第 2 章习题

1. 什么是人工智能？
2. 举例说明人工智能技术在复杂金融系统中的应用。
3. 简要阐述人工智能的历史。
4. 简要阐述人工智能的三个主要学派。
5. 人工智能的基础理论和技术有哪些？
6. 一般机器学习可分成哪三类学习范式？
7. 强化学习、监督学习和无监督学习三者之间的联系和区别是什么？
8. 机器学习中有哪些常用的激活函数？
9. 机器学习中有哪些常用的损失函数？
10. 机器学习中有哪些常用的优化算法？