第3章

平行推演与应用

平行推演技术是通过构建虚拟模型，模拟实际场景下的各种可能情况并进行分析和预测，可为地下空间的规划、设计、施工和运行维护提供全面的技术支持。本章首先介绍平行推演的概念以及推演模型，包括基于数学和物理方程的机理模型、基于历史运行数据的数据模型，以及机理与数据相融合的模型；然后介绍数据驱动的决策流程与方法，以及基于大语言模型的决策优化策略；最后结合地下空间安全开发与智慧运维的需求，探讨平行推演技术在地质环境监测、地下结构体性能评估、地下设备状态诊断、资源分配与调度和突发事件应急管理五个场景下的具体应用，并分析应用过程中面临的主要挑战。

3.1 数字孪生平行推演模型

地下空间数字孪生的平行推演是指结合实时数据动态更新和虚拟仿真模拟的方法，对地下设施、结构或环境的性能变化、潜在风险以及未来发展趋势进行预测和推演。平行推演可以通过构建基于机理的模型、基于数据的模型或者机理与数据的融合模型来实现。

3.1.1 基于机理的模型

基于机理的模型是一种通过数学和物理方程来描述系统内部各部分相互作用及其与外部环境关系的模型。机理模型可以模拟地下结构和岩土体的行为，如结构受力后的变形、地下水渗流和地下管线与地层之间的热传导等。构建机理模型的方法包括解析方法和数值方法。解析方法是指使用数学公式和定理直接求解问题的方法。它依赖对系统内部机制的深入理解，通过建立系统的精确数学模型（如代数方程、微分方程、积分方程等），并利用数学工具进行推导和求解，从而得到问题的精确解或闭式解。数值方法是指使用数值近似和计算机算法来求解问题的方法。当解析方法难以直接求解复杂系统或无法得到精确解时，数值方法就成为一种重要的替代方案。它通过迭代和近似来逐步逼近问题的解，因此得到的是近似解而非精确解。常用的数值方法包括有限元分析法、有限差分法、有限体积法、边界元法和离散元法等。有限元分析法是一种求解偏微分方程数值解的通用方法，特别适用于复杂几何形状和边界条件的问题。有限差分法通过离散化连续空间或时间域，将微分方程转化为差分方程，从而得到数值解。有限体积法将计算区域划

分为一系列互不重叠的控制体积，将求解的微分方程对每一个控制体积进行积分从而求解。边界元法是一种基于边界积分方程的数值方法，只在定义域的边界上划分单元，用满足控制方程的函数去逼近边界条件。离散元法将研究对象离散成一系列独立的单元（如颗粒、块体等），通过单元之间的相互作用来模拟整体的行为。表 3.1 总结了现有机理模型相关数值方法的优缺点及应用范围。

表 3.1　机理模型相关数值方法

方法	优点	局限性	适用范围
有限元分析法	精度高，适用于复杂结构	计算量大，对计算资源要求高	分析地下应力及位移，分析材料特性和边界条件
有限差分法	实现简单，适用于规则网格	处理复杂几何和边界条件较困难	模拟地下水流动，地下热传导、热对流
有限体积法	适用于非结构化网格	复杂度较高	地下水渗流
边界元法	精度高、计算效率高	对硬件要求较高，推导过程复杂	地下空间与外部环境的交互作用分析
离散元法	能够模拟颗粒材料的力学行为和破裂过程	计算量大	模拟地下岩土体的力学行为和颗粒间的相互作用

3.1.2　基于数据的模型

基于数据的模型是指基于观测到的数据，通过统计建模、机器学习、深度学习等方法建立出来的描述现象的模型。基于数据的模型建立需要通过对数据进行处理和分析，发现数据背后的规律和趋势，以便更高效地分析和优化系统的性能，进行故障诊断、虚拟仿真和动态预测等[1]。建立基于数据的模型通常包括数据集建立、模型选择、模型训练、模型优化和模型部署等步骤，如图 3.1 所示。其中，数据

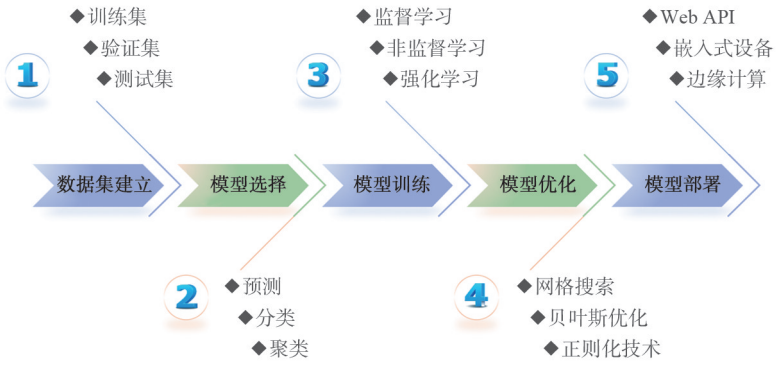

图 3.1　基于数据的模型建立过程

集分为训练集、验证集和测试集三部分，其占比通常分别为70%、15%、15%。训练集、验证集和测试集分别用于训练模型、调整模型参数和评估模型的最终性能。模型选择指根据预测对象选择合适的模型类型，如线性回归、决策树、支持向量机、神经网络等。模型训练过程使用训练集对模型进行训练，常用的方法包括监督学习、非监督学习和强化学习等。模型优化过程针对不同场景使用验证集评估并调整参数，使用的方法包括网格搜索、贝叶斯优化、正则化技术等。训练好的模型可部署到不同的环境中，包含Web API、嵌入式设备、边缘计算，并通过测试集数据检验模型在环境中的性能。

1. 数据集建立

数据集是指一组相关的数据样本，通常用于机器学习、数据挖掘、统计分析等领域。数据集主要分为结构化数据集和非结构化数据集两类。结构化数据集是指数据以明确定义的格式存储，每条数据都按照相同的数据结构进行组织，常见的形式包括表格、数据库等；而非结构化数据集则是指数据没有固定的格式，包括文本、图像、音频等形式。结构化数据集的优点在于可以方便地进行数据存储、查询、分析，适用于传统的数据分析方法；而非结构化数据集则更贴近真实世界的数据形式，挖掘其中的信息需要更多的技术手段和算法支持。

数据集的建立过程包括数据收集、数据处理、数据标注和数据存储。

（1）数据收集

数据收集是构建数据集的第一步，在平行推演中，收集的数据主要用于构建预测模型。数据收集的方法包括自动化采集、手动采集、批量采集和基于数据流的实时采集。由于地下空间中的数据量庞大，通常采用自动化采集方法，即通过脚本、API调用、网络爬虫等方式自动收集数据。此外，对于需要实时分析的场景，则需要通过实时数据流进行收集和处理数据，如物联网中的传感器数据。

（2）数据处理

数据处理是建立数据集的重要环节，其保证所建数据集的数据质量，如一致性、准确性等。数据处理包含数据清洗、数据转化和数据挖掘等。其中，数据清洗是指通过处理缺失值、重复值和异常值，确保数据的整洁性和一致性。数据转化是指对数据进行归一化、标准化或其他变换，如对类别数据进行编码、对时间序列数据进行平滑处理等。数据挖掘是一种较先进和自动化的数据处理方法，其旨在现有数据中找出有效、新颖、有潜在应用价值的模式与关系。

数据挖掘方法可分为操作型数据挖掘和模式检测挖掘两种。操作型数据挖掘是一种从数据中快速提取有用信息的方法，它通过建立各种模型（如聚类分析模型、预测回归模型）来帮助理解和分析数据。模式检测挖掘指的是通过算法或统计技术，在数据集中自动识别并提取特定结构或模式的方法，主要用于检测不寻常的

行为模式。一般来说，数据的不连续性、噪声、模糊性和不完整性会给提取带来各种问题，虽然大多数挖掘算法能分离这些与属性无关的数据的影响，但随着异常数据的增加，挖掘算法的预测准确性可能会下降。数据挖掘的典型流程如图 3.2 所示，其主要步骤包括业务理解、数据理解、数据预处理、建模、评估以及部署。

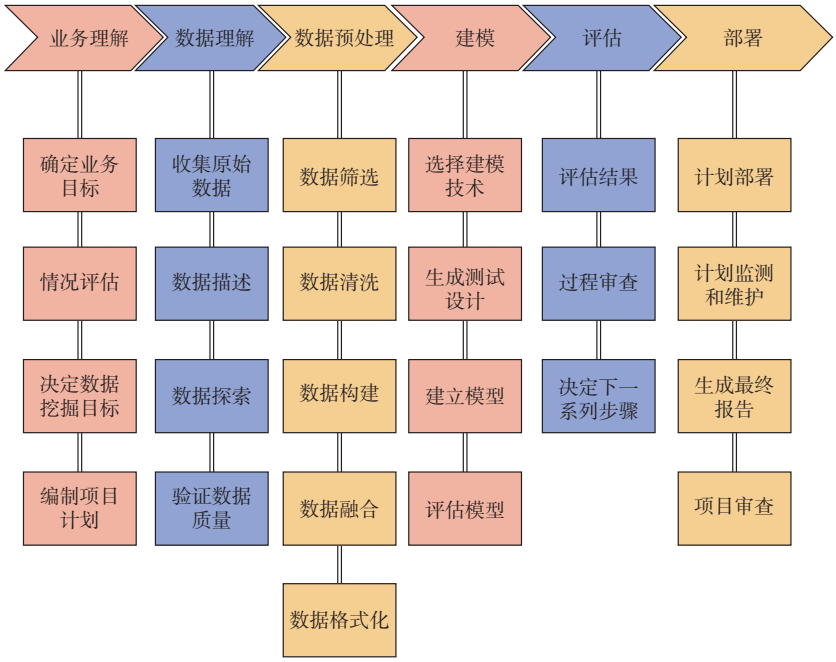

图 3.2　数据挖掘的典型流程[2]

机器学习在数据挖掘方面具有很大的潜力。与针对特定问题的数据挖掘算法不同，机器学习算法可用于不同问题的挖掘和分析。由于大多数机器学习算法可用于寻找优化问题的近似解，因此如果数据分析问题可表述为优化问题，就可采用机器学习算法。遗传算法作为一种机器学习算法，是基于自然选择和遗传学原理的优化技术，通常用于解决复杂的优化问题。

传统的遗传算法是一种基于自然选择和遗传机制的搜索算法，通常只使用一个全局种群，所有个体都在这单一种群中进行进化。与传统遗传算法不同，并行遗传算法之一的岛屿模型遗传算法通常将种群划分为多个子种群，且每个子种群在一定程度上可独立进化，如图 3.3 所示。在分析时，可以将子种群分配到不同的线程或计算机节点上进行并行计算，从而大大提升计算效率。

（3）数据标注

数据标注是指对数据集中的每个数据点进行标记，以帮助模型理解和学习数据中的特征。数据标注主要包括标注分类和数据注释两个步骤。标注分类是指将

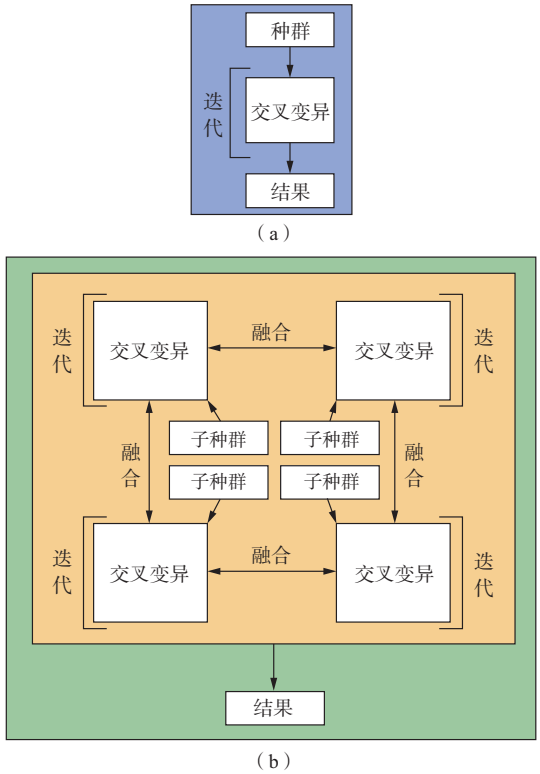

图 3.3　传统遗传算法与并行遗传算法[3]

(a) 传统遗传算法；(b) 并行遗传算法

数据集中的数据点归类到预定义的类别或标签中。每个数据点根据其特征被分配到一个或多个类别中，而这些类别也可以是离散的标签。标注分类使得模型能够从数据中识别类别，处理更加复杂的任务。数据注释是指在数据集中的数据点上添加更为详细的信息或标签，以增强数据的含义和使用价值。注释可以包括边界框、关键点、语义分割标签、属性描述等。数据注释可以增强数据集的信息量，从而促进模型更全面的学习和分析。

(4) 数据存储

数据存储是指将数据按选择的格式和标准存储在数据集中。数据存储的格式可以是结构化的(如关系数据库)、半结构化的(如 JavaScript Object Notation 文件)或非结构化的(如音频、视频文件)。数据仓库是为存储大量结构化、半结构化和非结构化数据而设计的，它能够高效地处理和管理大规模的历史数据。数据仓库从多个异构数据源中抽取、清洗、转换和加载数据，然后将这些数据集中存储，形成面向主题的、集成的数据集合。数据仓库采用优化的存储结构，如星形或雪花形

模型，使得数据存储不仅高效，而且能够支持复杂的查询和分析操作。

2. 模型选择

选择合适的模型是建立基于数据的模型的关键步骤，它直接影响到模型的性能和结果的准确性。模型通常基于问题性质、数据特性以及目标要求进行选择。对于平行推演而言，模型的功能集中在演化预测上，涉及对连续数值进行预测，如耗能预测、运动预测、劣化预测等，因此常用的模型包括线性回归、多项式回归、决策树、随机森林、支持向量机和神经网络模型等，各模型的优缺点及适用场景如表 3.2 所示。

表 3.2　常用的演化预测模型

模型	优势	局限性	适应场景
线性回归	解释性强、计算效率高	无法处理复杂的非线性关系	变量之间存在线性关系
多项式回归	灵活性高	易过拟合	存在非线性关系，但可通过较低次多项式拟合
决策树	解释性强	易过拟合	数据具有非线性和交互作用
随机森林	鲁棒性高	解释性较差、计算复杂	数据维度较高，且存在噪声和异常值
支持向量机	对于高维数据处理能力强	计算复杂度高、参数敏感性高	数据维度较高，且存在非线性关系
神经网络模型	对于复杂数据处理能力强	数据需求量大、调参难度大	存在较为复杂的非线性关系

3. 模型训练

模型训练是指基于给定的数据集，使用特定的算法来调整和优化模型的参数，以便模型能够学习数据中的特征和模式，从而能够对未知数据进行准确的预测或分类。在训练过程中，模型会根据数据的输入和输出不断调整其参数，以最小化损失函数，提高模型的预测精度。训练过程通常需要多次迭代，直到模型达到收敛状态或满足预设的停止条件。训练过程分为两个阶段，第一个阶段是数据由低层次向高层次传播的阶段，用于生成模型的预测值，即前向传播阶段；第二个阶段是当前向传播得出的结果与预期不相符时，将误差从高层次向低层次进行传播训练的阶段，即反向传播阶段，如图 3.4 所示。

4. 模型优化

在训练完成后，需要使用验证集或测试集对模型的性能进行评估，评估指标通

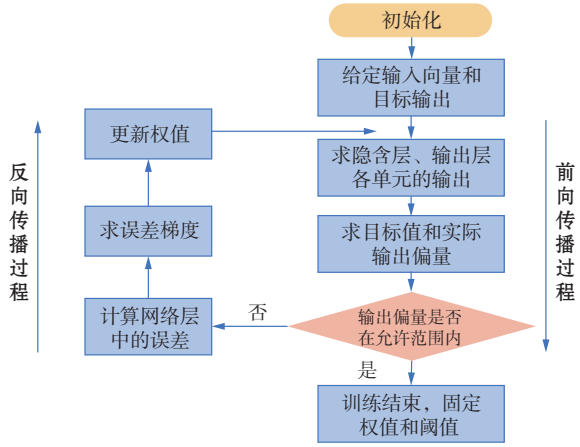

图 3.4　模型训练过程

常包括准确率、精确率、召回率等，这些指标反映模型在不同方面的性能表现。根据评估结果，可采用包括网格搜索、贝叶斯优化、引入正则化技术等对模型进行调优，以提高模型的泛化能力和鲁棒性。

网格搜索是一种穷举搜索方法，它系统地遍历多种参数组合，通过交叉验证确定最佳效果参数。首先，需要为模型中的每个重要参数指定一个搜索范围或候选值列表，并使用交叉验证（如 k-Fold 交叉验证）来评估每种参数组合的性能。然后，根据交叉验证的结果（如平均准确率、调和平均数等），选择表现最好的参数组合。

贝叶斯优化则利用贝叶斯定理来指导搜索过程，通过构建目标函数的概率模型（如高斯过程）来预测最佳参数组合。贝叶斯优化能够通过概率模型的构建来有效处理高维搜索空间的不确定性，在不确定性大的区域进行探索，从而避免陷入局部最优。

正则化技术是一种在模型训练中添加约束或惩罚项的方法，旨在防止模型过拟合数据。正则化技术主要通过约束模型的参数，使模型在训练数据上表现良好的同时，也能在未知数据上保持较好的泛化性能。常用的正则化技术包括 L1 正则化（在损失函数中添加模型参数的绝对值之和作为正则化项）、L2 正则化（在损失函数中添加模型参数的平方和作为正则化项）和 Dropout（在训练过程中，以一定概率随机丢弃部分输出，从而减少节点之间的共适应性）。

5. 模型部署

模型部署是指将训练好的模型转换为可以在生产环境中运行的形式，并通过适当的接口或框架使其能够接收输入数据、进行预测或决策，并返回结果的过程。

常用的部署方式包括云部署、边缘部署、容器化部署和本地服务器部署。其中，云部署是指将模型托管在云服务提供方的平台上，通过云平台的计算资源和服务来运行和管理模型。云平台可以根据需求动态调整计算资源，实现水平扩展或缩减，适应流量的波动。边缘部署是指将模型部署在靠近数据源或用户终端的边缘设备上，如工业自动化设备、智能摄像头等。边缘部署方式减少了数据传输的延迟，适用于对响应时间敏感的应用场景。容器化部署是指将模型及其依赖环境打包到一个容器中，并通过容器化技术来部署模型。这种方式便于模型的快速部署和扩展，同时可以实现资源的动态分配和负载均衡。本地服务器部署是指将模型部署在自有的物理服务器或数据中心上，利用自有的计算资源进行模型的推理和管理。本地部署确保了对数据的完全控制，适合需要高度数据隐私和安全性的应用场景。

3.1.3　机理与数据的融合模型

机理与数据的融合模型是指将物理机理嵌入数据驱动模型中，以充分发挥机理模型的可解释性和泛化能力强、数据驱动模型灵活和可学习的优势。融合模型的最大优势是可以将虚拟模型本身与数字孪生系统中的设计、控制、运维等任务需求灵活对接，并充分发掘海量传感器获得的数据，这是机理模型方法所不具备的。

近年来，物理信息神经网络(physics-informed neural networks，PINN)作为机理与数据融合建模方法中的一个重要方向，在学术界引起了广泛的关注。PINN是一种在损失函数中引入物理系统方程的正则项约束的建模方法，它将物理系统的先验知识融入神经网络中，以提高模型的物理解释性和泛化能力。此外，PINN通过引入物理知识作为先验信息，可以在数据较少或噪声较大的情况下仍然进行有效的训练和预测，为数据量较少的复杂系统分析和预测提供了新的思路和方法。

3.2　数据驱动决策与优化

3.2.1　数据驱动的决策流程与方法

目前，各行业在日常运行中产生了大量的静态和流式数据。随着数据的爆炸式增长和信息技术的飞速发展，决策者收集、存储、访问和分析数据的能力不断提高。因此，数据驱动决策(data-driven decision-making，D^3M)已成为一个重要的决策方法。数据驱动决策方法的具体应用有智慧城市中利用交通历史数据进行交通流量优化，电网运行中分析历史用电数据以预测电力需求，在紧急事态响应中通过数据分析确定最优路线和资源分配并提高应急响应效率等。支持数据驱动决策的算法包括数学编程/优化、基于规则的系统和启发式方法、马尔可夫和概率模型等，这些方法在成本估计、维护计划、联合调度、多状态多组件系统优化等维护决策中

广泛应用，如图 3.5 所示。数据驱动决策分为可编程数据驱动决策（P-D³M）和非可编程数据驱动决策（NP-D³M）两大类。

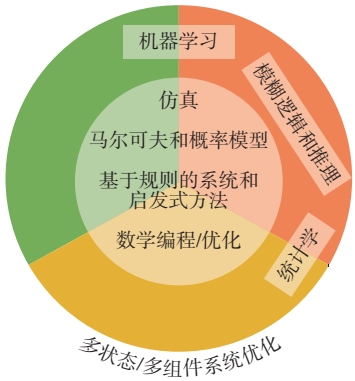

图 3.5　数据驱动决策方法分类[2]

对于可编程数据驱动决策问题，基于历史数据的分析可以准确预测可编程决策模型的未知参数，这些参数可以反馈、更新和驱动模型。对于非可编程数据驱动决策问题，即当可编程模型无法描述决策问题时，基于机器学习的数据分析可以从数据中发现有用的规则或知识/选项，从而为决策提供支持。两类数据驱动决策的技术框架如图 3.6 所示。

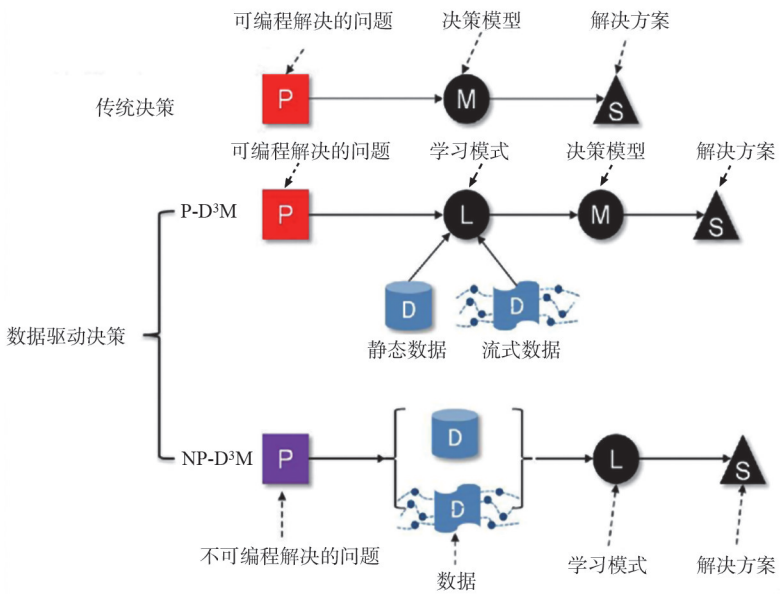

图 3.6　数据驱动决策框架[3]

1. 可编程数据驱动决策

可编程数据驱动决策(programmable data-driven decision-making,P-D^{3}M)是基于数据挖掘或统计学习的决策模型的推导,以及相应的可编程模型(也称为编程或优化模型)对决策提供支持。常用的模型包括多目标决策模型和多层次决策模型。多目标决策模型允许在多个相互冲突的目标之间找到最优解,而多层次决策模型模拟了决策过程中不同层级之间的信息流动和决策制定,以更全面地考虑复杂决策场景中的多种因素。

2. 非可编程数据驱动决策

与P-D^{3}M相反,非可编程数据驱动决策(non-programmable data-driven decision-making,NP-D^{3}M)指的是在决策过程中使用数据分析,但不依赖传统的编程方法来处理数据。这种决策方式通常涉及使用图形界面、预配置的软件工具、拖放式界面或人工智能(AI)辅助工具,这些工具使得非技术用户也能够进行数据分析和决策。NP-D^{3}M适用于决策模型的推导在计算上不可行或成本过高的情况。大型且高度复杂的决策问题中的动态性和不确定性可能会导致严重的模型不匹配,或使可编程模型变得难以实现。为了解决这个问题,NP-D^{3}M探索了一种学习机制,它能从数据中发现规则和模式,从而直接做出决策,使决策者能在基于数据证据的基础上制定战略。基于规则的方法和强化学习是非可编程数据驱动决策方法的常用技术。

(1) 基于规则的方法

NP-D^{3}M中基于规则的方法是一种决策支持系统,它依赖预定义的规则集来处理数据和做出决策。这些规则通常由领域专家根据业务逻辑、法律法规、行业标准或最佳实践来制定。决策规则通常从决策树中提取。作为一种合成的、易懂的和通用的知识表示法,基于规则的决策制定已被广泛应用于分类、排序和选择等许多领域。

(2) 强化学习

强化学习(reinforcement learning, RL)是机器学习的一个重要分支,主要研究如何让智能体(agent)在与环境的交互中学会做出最优决策。强化学习、监督学习和无监督学习是机器学习中的三大类别,其主要特点是不依赖大量的标注数据,而是通过智能体与环境之间的试错(trial-and-error)过程来学习。强化学习提供了一个基于智能体与环境之间互动的学习框架来解决决策问题。强化学习的基础是奖励驱动行为,即智能体通过最大化未来奖励来做出决策。智能体与环境互动,通过观察先前决策的后果,学会根据所获奖励修改自己的决策。

RL的本质是找到一个函数来解决决策问题。以下是强化学习的一般步骤。