进化深度学习

[加] 迈克尔·兰哈姆(Micheal Lanham) 著

殷海英 译

清華大學出版社

北 京

北京市版权局著作权合同登记号 图字：01-2024-0877

Micheal Lanham
Evolutionary Deep Learning: Genetic algorithms and neural networks
EISBN: 9781617299520

图书在版编目(CIP)数据

进化深度学习 / (加) 迈克尔 • 兰哈姆(Micheal Lanham) 著；殷海英译. —北京：清华大学出版社，2024.4

书名原文：Evolutionary Deep Learning: Genetic algorithms and neural networks
ISBN 978-7-302-65821-4

I. ①进… II. ①迈… ②殷… III. ①机器学习 IV. ①TP181

中国国家版本馆 CIP 数据核字(2024)第 054428 号

责任编辑：王 军
装帧设计：孔祥峰
责任校对：成凤进
责任印制：沈 露

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：定州启航印刷有限公司
经 销：全国新华书店
开 本：170mm×240mm 印 张：21.25 字 数：428 千字
版 次：2024 年 4 月第 1 版 印 次：2024 年 4 月第 1 次印刷
定 价：98.00 元

产品编号：106042-01

我想把这本书献给我的母亲莎朗·兰哈姆。她在我很小的时候就教会我如何超越常规思维，相信自己。

译 者 序

在 2013 年的秋季，我正式在目前我服务的大学开始讲授数据科学和人工智能的课程。回想过去的 10 年，在那些初级和中级的机器学习课程中，学生在完成课程的学习之后，经常问我如下这些问题，不知道阅读本书的你，是否也会有这样的困扰。首先就是关于特征的学习，在传统的机器学习过程中，特征需要人工提取，而在特征提取的过程中，如果不能达到最优，那么最后得到的模型性能也很难令人满意。其次，传统的机器学习技术可以很轻松地处理结构化数据，但像图片、视频或语音这种非结构化数据，传统的机器学习技术很难提取有效的特征。还有同学会说，在传统的机器学习中，需要大量的样本数据来进行特征提取，这对数据准备带来了不小的挑战。同时，传统机器学习技术很难处理高维稀疏数据、模型泛化能力差以及无法对时间序列进行建模，这些都影响机器学习技术在生产中的应用。

“是时候看一些高级的内容啦！”出于自己的私心，期待学生在下学期还选我的其他课程，我通常都会在初级人工智能课程结束时，为学生演示一些深度学习或深度进化学习的小例子，让他们看到如何通过简单的深度进化学习代码就可以轻松地处理语音、图像和视频，以及深度进化学习在机器视觉、自然语言处理等方面所取得的成果。

确实，如今深度进化学习在目标检测、语义分隔等方面有着广泛的应用，比如自动驾驶中的物体检测、医学图像分析等。还有我们常用的机器翻译，也使用了大量的深度进化学习技术，比如大家所熟知的谷歌翻译引擎，Siri 智能助手等。大家每天使用的在线购物网站、音视频网站中的推荐系统，都离不开深度进化学习技术。还有大家每天观看的短视频，那些听起来非常流畅又非常熟悉的解说配音，也是通过深度进化学习技术实现的更加自然的语音合成。大家之前就听说过的股市预测、疾病发生趋势等与预测相关的应用场景，深度进化学习都有非常出色的表现。

《进化深度学习》已经被我推荐给选修人工智能高级课程的本科生以及人工智能方向的硕士研究生。本书共 12 章，由三部分组成。第 I 部分：入门。如果你对模拟和进化或遗传计算不熟悉，请务必阅读整个部分。这部分也是一个很有帮助的复习，并演示了几个有趣的应用程序。第 II 部分：优化深度学习。如果你确实需要优化神经进化或对深度学习系统进行超参数调整，请阅读本部分或其中的特定章节。第III部分：高级应用。本部分分为三个子部分：进化生成建模(第 8 章和第 9 章)、NEAT(第 10 章

和第 11 章)及本能学习(第 12 章)。可以独立阅读每个子部分。相信通过阅读本书，你一定能在人工智能和机器学习领域迈上更高的台阶。

最后，我要由衷地感谢清华大学出版社的王军老师。感谢他为我出版了多种关于机器学习、人工智能、云计算和高性能计算的书籍和译作。他提供了一种新的方式，让我能够与大家分享我的知识和经验。他的支持和帮助使我能够将这些书籍带给读者，并希望这些书籍能对读者在相关领域的学习和工作有所帮助。

殷海英

埃尔赛贡多，加利福尼亚州

关 于 作 者

Micheal Lanham 是一位经验丰富的软件和技术创新者，拥有 25 年的工作经验。他作为研发工程师，在游戏、图形、网络、桌面、工程、人工智能、地理信息系统(GIS)和机器学习等领域开发了各种软件应用程序。在他的职业生涯中，Micheal 在游戏开发中使用神经网络和进化算法。他将自身的技能和经验运用于 GIS 和大数据/企业架构师工作中，为各种工程和业务方面的应用增强游戏化元素。自 2016 年底以来，Micheal 一直是一位热衷于分享知识的作者和演讲者。目前，他已完成多种关于增强现实、声音设计、机器学习和人工智能的书籍。他在人工智能和软件开发的许多领域都有所涉猎，但目前专注于生成式建模、强化学习和机器学习运维。Micheal 与家人居住在加拿大的卡尔加里，目前从事 AI、机器学习运维和工程软件开发的写作、教学和演讲工作。

致　谢

我要感谢开源社区，特别是以下项目：

- Python 中的分布进化算法(Distribution Evolutionary Algorithms in Python，DEAP)——https://github.com/DEAP/deap。
- Python 中的基因表达式编程框架(Gene Expression Programming Framework in Python，GEPPY)——https://github.com/ShuhuaGao/geppy。
- Python 中的增强拓扑的神经进化(NeuroEvolution of Argumenting Topologies in Python，NEAT Python)——https://github.com/CodeReclaimers/ neat-python。
- OpenAI Gym——https://github.com/openai/gym。
- Keras/TensorFlow——https://github.com/tensorflow/tensorflow。
- PyTorch——https://github.com/pytorch/pytorch。

如果没有其他人不知疲倦地花费精力和时间开发和维护这些存储库，就不可能有本书的出版。这些也是所有对提高 EA 或 DL 技能感兴趣的人的优秀资源。

特别感谢我的家人一直以来对我写作、教学和演讲的支持。他们总是阅读我的文章或其他资料，并给我提出宝贵的意见。

非常感谢 Manning 出版社的编辑和制作团队，帮助我完成了本书。

感谢所有的审阅者：Al Krinker、Alexey Vyskubov、Bhagvan Kommadi、David Paccoud、Dinesh Ghanta、Domingo Salazar、Howard Bandy、Edmund Ronald、Erik Sapper、Guillaume Alleon、Jasmine Alkin、Jesús Antonino Juárez Guerrero、John Williams、Jose San Leandro、Juan J. Durillo、Kali Kaneko、Maria Ana、Maxim Volgin、Nick Decroos、Ninoslav Čerkez、Oliver Korten、Or Golan、Raj Kumar、Ricardo Di Pasquale、Riccardo Marotti、Sergio Govoni、Sadhana G、Simone Sguazza、Shivakumar Swaminathan、Szymon Harabasz 和 Thomas Heiman。他们提出了许多有益的建议，并完善了本书中的许多内容。

前　言

25 年前，当我开始从事机器学习和人工智能的工作时，有两项主导技术被认为是未来的重要发展方向。这两项技术在解决复杂问题方面都显示出了巨大的潜力，并且在计算上是等效的。这两项技术分别是进化算法和神经网络(深度学习)。

在接下来的几十年里，我目睹了进化算法的急剧衰落和深度学习的爆炸性增长。这场斗争的结果是由计算效率决定的，深度学习也展示了许多新颖的应用。另一方面，在大多数情况下，进化和遗传算法的知识与应用已经逐渐减少到成为附注或脚注的程度。

我写本书的目的是展示进化和遗传算法可以为深度学习系统提供收益的能力。这些收益在深度学习进入自动机器学习时代尤为重要，在这个时代，自动化大规模模型开发正逐渐成为主流。

我也相信，我们对通用人工智能和智能体的探索可以从进化的角度得到帮助。毕竟，进化是自然界用来形成我们智慧的工具。那么，为什么它不能改进人工智能呢？我猜想，可能是我们太急躁和傲慢了，认为人类可以独自解决这个问题。

通过本书，我希望将进化方法作为超越常规思维的一种方式，展示其在深度学习中的强大力量。我希望本书以有趣和创新的方式展示进化方法的基础知识，同时涉及进化深度学习网络(即 NEAT)和本能学习等先进领域。本能学习是我对我们应该如何更多地关注生物生命是如何进化的，并在寻找更智能的人工网络时反映出这些相同的特征的看法。

关 于 本 书

本书介绍了进化算法和遗传算法，从解决有趣的机器学习问题到将其中的概念与深度学习相结合。本书的开始部分介绍了 Python 中的模拟以及进化算法与遗传算法的概念。随着介绍的深入，重点转向展示它们在深度学习中的应用和价值。

本书读者对象

本书读者应该具备扎实的 Python 编程背景，并理解核心的机器学习和数据科学概念。在后面的内容中，深度学习的背景知识对理解概念至关重要。

本书组织结构：路线图

本书分为三个部分：入门、优化深度学习和高级应用。在第 I 部分，将介绍模拟、进化、遗传和其他算法的基础知识。以此为基础，继续展示进化和深度学习中遗传搜索的各种应用。最后，将介绍生成式建模、强化学习和广义人工智能的高级应用。下面是每章的概要。

第 I 部分：入门

- 第 1 章介绍了将进化算法与深度学习相结合的概念。
- 第 2 章提供了关于计算模拟的基本介绍，并介绍了如何利用进化进行计算。
- 第 3 章介绍了遗传算法的概念和 DEAP 框架的使用。
- 第 4 章介绍了遗传和进化算法的一些有趣应用，从推销员出差问题到生成 EvoLisa 的图像。

第 II 部分：优化深度学习

- 第 5 章演示了几种使用遗传算法或进化算法优化深度学习系统超参数的方法。
- 第 6 章介绍了使用神经进化研究深度学习系统的网络架构优化。
- 第 7 章着眼于使用进化优化卷积神经网络架构的高级应用。

第III部分：高级应用

- 第 8 章介绍或回顾了使用自编码器进行生成式模型的基础知识，然后展示了进化如何发展出进化自编码器。
- 第 9 章继续第 8 章的内容，介绍或回顾了生成式对抗网络，以及如何通过进化来优化它。
- 第 10 章介绍了 NEAT，并讨论了如何将其应用于各种基准应用。
- 第 11 章讨论了强化学习和深度强化学习的基础知识，然后展示了如何利用 NEAT 解决 OpenAI Gym 上的一些困难问题。
- 第 12 章展望了进化在机器学习中的未来，并探讨了它如何为广义人工智能提供见解。

关于代码

本书中的所有代码都是使用 Google Colab notebook 编写的，并可在作者的 GitHub 存储库中找到：https://github.com/cxbxmxcx/EvolutionaryDeepLearning。要运行代码，只需要在浏览器中导航到 GitHub 存储库，并找到相关的代码示例。所有代码示例的名称都以章节编号为前缀，然后是示例编号，如 EDL_2_2_Simulating_Life.ipynb。然后，只需要点击 Google Colab 标识即可在 Colab 中启动 notebook。任何依赖项都将在 Colab 上预先安装或作为 notebook 的一部分进行安装。

在许多情况下，原始源代码已经进行了重新格式化：添加了换行符并重新调整了缩进，以适应书中可用的版面空间。在极少数情况下，即使这样仍然不够，代码清单中也会包含行延续标记(➥)。此外，在文本中描述代码时，源代码中的注释通常会被删除。在许多代码清单中都提供代码注释，用来突出显示重要概念。

可以扫描封底二维码下载本书源代码。

目　录

第Ⅲ部分　高级应用

第I部分

入　门

进化算法和遗传算法已经存在了几十年。就计算能力而言，进化方法在机器学习领域远不如深度学习强大。然而，进化方法可以提供独特的工具，辅助解决各种优化问题，从超参数调整到网络架构。但在讨论这些模式之前，需要先介绍进化算法和遗传算法。

在第 1 章中，介绍了使用进化方法优化深度学习系统的概念。由于本书涵盖的深度学习优化方法属于自动化机器学习范畴，因此还介绍了结合进化的 AutoML(自动机器学习)方法。

在第 2 章中，介绍了康威的《生命游戏》(*Conway's Game of Life*)中的生命模拟，使用一个简单的场景，并通过遗传算法进行进化。接下来，在第 3 章中，介绍了不同形式的遗传算法，使用 Python 中的分布式遗传算法库 DEAP。最后，在第 4 章中，通过介绍其他多样化的进化方法，为本书的第 I 部分做了总结。

第1章

进化深度学习简介

本章主要内容

- 进化计算是什么，如何将其集成到深度学习系统中
- 进化深度学习的应用
- 建立优化深度学习网络的模式
- 自动化机器学习在优化网络中的作用
- 利用进化计算方法增强深度学习开发的应用

深度学习(Deep Learning，DL)已成为人工智能(Artificial Intelligence，AI)和机器学习(Machine Learning，ML)迅猛发展中普遍采用的技术。它从一开始被认为是伪科学(参见 Terrence J. Sejnowski 于 2018 年在 MIT Press 出版的 *The Deep Learning Revolution*)，发展到在从乳腺癌诊断到自动驾驶等各个主流应用领域都得到了广泛应用。虽然很多人认为它是一种未来的技术，但也有人更加务实和实际地看待其不断增长的复杂性和对数据的渴求。

随着深度学习变得越来越复杂，我们不断地往它里面输入越来越多的数据，希望在某个特定的领域能够获得一些重大的启示。遗憾的是，这种情况很少发生，而且经常会得到糟糕的模型、差的结果，以及面对愤怒的老板。这个问题将一直持续下去，直到我们为深度学习系统开发出高效的处理流程。

构建有效和健壮的深度学习系统的过程与构建任何其他机器学习或数据科学项目的过程相似。虽然某些阶段所需要的资源和复杂性可能有所不同，但所有步骤都是一样的。相对新的深度学习领域通常缺乏一个可以帮助你自动化完成一些过程的工具箱。

进化深度学习(Evolutionary Deep Learning，EDL)就是这样一个工具箱或一组模式和实践，可以帮助你自动开发深度学习系统。本书使用的 EDL 一词涵盖了广泛的进化计算方法和模式，应用于深度学习系统的各个方面，贯穿整个机器学习过程。

1.1 什么是进化深度学习

进化深度学习(EDL)是将进化算法与深度学习相结合的一组技术的总称。这些方法可用于优化深度学习系统，从数据收集到验证等各个方面。EDL 并不是新概念，将进化方法与深度学习相结合的工具已经有很多名字，包括 Deep Neural Evolution、Evolutionary Neural AutoML、Neuroevolution、Evolutionary AI 等。

EDL 是人工智能的两个独特子领域：进化计算(Evolutionary Computation，EC)和将深度学习应用于自动化和改进模型的应用。进化计算是一种方法，通过模拟生物或自然过程来解决复杂问题。这可以在深度学习之上应用，以实现自动化和优化解决方案，同时也有潜力发现新的策略和架构。

我们在 EDL 中涵盖的广泛方法并不新颖，它已经存在了 20 多年。虽然很多研究表明这些方法在自动调整深度学习模型方面非常成功，但在人工智能领域炒作的各种更前沿、更精心设计的例子的背后，它获得了相对较少的重视。

然而，对于现在接触深度学习的许多人来说，构建稳健、高性能的模型令人望而生畏，充满了挑战。其中许多挑战要求对所选择的深度学习框架的所有选项和特点有前沿而深入的了解，以便理解模型可能仅仅是错误拟合的情况。EDL 作为一种自动机器学习(AutoML)解决方案，旨在解决从业者(无论是经验丰富的人还是新手)将面临的大部分问题。

EDL 的目的是提供更好的机制和工具集，以便为构建深度学习解决方案提供优化和自动化机器学习能力。进化方法是一种优秀的、相对简单的机制，可以提供广泛的优化工具，并可应用于深度学习。虽然进化技术有可能自动化构建更先进的人工智能，但这并不是 EDL 或本书的写作目的。

相反，我们专注于使用进化技术构建更优化的网络。在此之前，将介绍操作并讨论使用进化计算和进化算法深入了解基本概念。下面开始简要介绍进化和进化过程。

进化计算简介

进化计算是人工智能的一个子领域，它使用生物和自然启发的过程解决复杂问题。用“进化”一词描述这类算法，是因为许多算法以自然选择理论为基础。

自然选择理论是由查尔斯 • 达尔文在他的著作《物种起源》(John Murray Press，

1859)中提出的，它定义了地球上生命的进化过程。该理论描述了强壮和适应环境的生命将继续生存和繁衍，而虚弱或适应能力不足的生命将死亡并灭绝。大约 1837 年，达尔文在环绕南美洲的“HMS Beagle”号船上作为一名博物学家进行自己的研究，发展了这一理论。作为一名虔诚的宗教信徒，达尔文在发表这部著名作品之前，与他的研究结果进行了长达 22 年的斗争。

基于达尔文的理论，进化计算的基石是模拟一个个体或一群个体在系统中寻找最优解。其目的是通过允许个体的变化，演化出能在这样的人工环境中存活和繁衍的个体。个体的变化机制会因进化计算方法而异，但在所有情况下，都需要一个机制来量化个体的生存状况。

用来量化个体生存或繁衍能力的术语称为适应度(fitness)。适应度是进化计算中通用的术语，用于定义个体在环境中的生存或表现能力。适应度可以用多种方式衡量，但在所有情况下，它是决定个体或个体群体解决问题的效率的重要因素。

自然选择和适应度的概念被用作多种计算方法的基石，这些方法旨在或浅或深地模拟生物繁殖的过程。其中一些方法甚至模拟了细胞中发生的染色体分裂和 DNA 共享的遗传有丝分裂过程。以下是当前一些值得注意的进化计算算法。

- 人工生命(Artificial Life)：从康威的《生命游戏》和冯 • 诺依曼细胞自动机开始，这些过程使用智能体(agents)模拟了生命本身的人工过程。在该算法中，智能体经常根据其与其他智能体或环境的接近程度来移动、流动、生存或死亡。虽然智能体模拟通常用于模仿真实世界，但也可以用于优化过程。
- 差分进化(Differential Evolution)：一种将差分计算与进化算法相结合来优化搜索的过程。这种技术通常会与另一种进化计算方法分层，如人工生命。在该算法中，智能体通过获取向量差异并将其重新应用于种群来进化或改变。
- 进化算法(Evolutionary Algorithms)：一种更广泛的进化计算方法，以自然选择的形式将进化应用于问题。这些方法通常专注于模拟个体群体。
- 进化编程(Evolutionary Programming)：进化算法的一种特殊形式，它使用代码创建算法。在该算法中，个体用一段代码表示，通过运行代码生成的最优值来度量其适应度。有多种实现进化编程的方法。而在很多情况下会使用更具体的方法，如基因表达。
- 遗传算法(Genetic Algorithm)：这种算法使用了我们在生物体中看到的低水平细胞有丝分裂，允许遗传特征传递给后代。遗传算法是通过将个体的特征编码到基因序列中来模拟这一过程，其中任意基因序列可以像 0 或 1 序列一样简单，评估某种适应度指标。该适应度用于模拟生物选择过程和双亲个体的交配，以产生新的组合后代。

- 遗传编程(Genetic Programming)：该算法使用遗传算法构建编程代码。在遗传算法中，个体的特征较为通用，但在遗传编程中，一个特征或基因可以代表任意数量的函数或其他代码逻辑。遗传编程是一种专门的技术，可以开发新的算法代码。例如，可以使用遗传编程编写智能体模拟代码，用于解决迷宫问题或创建图像。
- 基因表达编程(Gene Expression Programming，GEP)：该算法是对遗传编程的进一步扩展，用于开发代码或数学函数。在遗传编程中，代码被抽象为高级函数，而在基因表达编程中，目的是开发具体的数学方程。基因表达编程与遗传编程的一个关键区别是使用表达式树来表示函数。在遗传编程中，表达式树表示代码，而在基因表达编程中，表达式树表示数学表达式树。这样做的好处是代码将根据放置位置遵循明确定义的运算顺序。
- 粒子群优化(Particle Swarm Optimization，PSO)：它属于人工生命的一个子集，模拟了人工制造且具有一定智能的粒子。在该算法中，评估每个粒子的适应度，并选择最优粒子作为其他粒子聚集的焦点。
- 群体智能(Swarm Intelligence)：这是一种模拟昆虫或鸟类群体行为的搜索方法，用于寻找优化问题的峰值。它与粒子群优化非常相似，但实现方式因适应度评估而异。

图 1-1 展示了本书中用于应用 EDL 的一系列进化计算方法的层次结构。还有其他几种进化计算方法可以用来改进深度学习模型，但作为介绍，将重点介绍图中的基本方法，着重于生命模拟和遗传模拟领域。

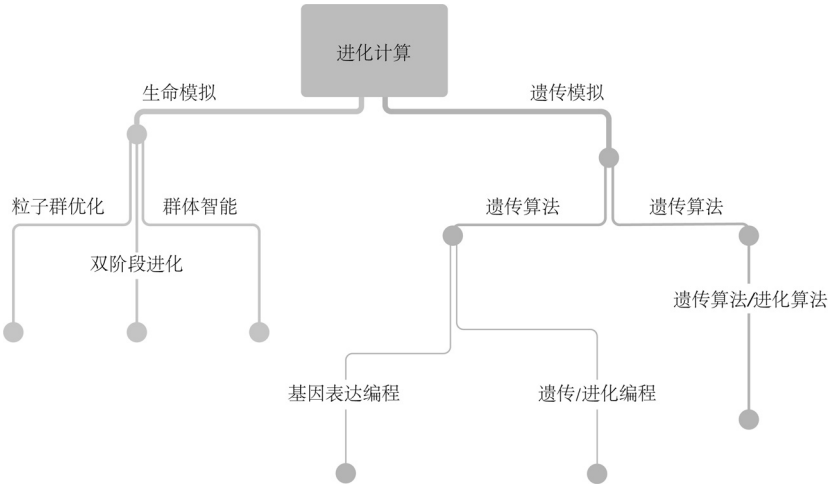

图 1-1 用于应用 EDL 的进化计算子集

生命模拟是进化计算的一个具体子集，采用模拟在自然界中观察到的自然过程的

方法，如粒子或鸟群的聚集方式。另一方面，遗传模拟则模拟了我们在生物生命中观察到的细胞有丝分裂过程。更具体地说，它模拟了有机体在进化过程中基因和染色体的遗传传递过程。

1.2　EDL 的缘由和应用领域

EDL 既是一个概念，也是一套用于深度学习优化的工具和技术。在概念上，EDL 是利用进化计算对深度学习网络进行优化的模式和实践。然而，它也提供了一套可以叠加在深度学习之上甚至作为深度学习的替代方法的工具。

使用 EDL 的原因和应用范围不仅取决于你在深度学习方面的专业水平，还取决于需求所达到的范围。这并不意味着深度学习的初学者不能从使用 EDL 中受益。事实上，本书探讨了许多神经网络的微妙之处，这些微妙之处通过 EDL 被暴露出来，并且对所有从业者都有帮助。

关于 EDL 的应用范围，答案很简单：无处不在。它可以用于基本的超参数优化，寻找非连续解的神经权重搜索，平衡生成对抗网络中的对抗网络，甚至替代深度强化学习。你确实可以将本书介绍的技术应用于任何深度学习系统。

回答为什么使用 EDL 的问题归结为必要性。进化方法为所有深度学习系统提供了进一步优化或增强解决方案的选择。然而，EDL 在计算上要求较高，可能不适用于那些简单的系统。然而，对于复杂或新颖的问题，进化方法为所有深度学习从业者提供了新的解决方案。

1.3　深度学习优化的需求

深度学习是一项强大而又相对较新且常常被误解的技术，它有许多优点，也有一些缺点。其中一个缺点是需要理解和优化模型。这个过程可能需要数小时的数据注释或模型超参数调整。

在几乎所有情况下，都不能直接使用未经优化的模型，通常需要优化深度学习系统的各个方面，从调整学习率到选择激活函数。优化网络模型通常成为主要任务，如果手动进行，可能需要花费大量的时间和精力。

优化深度学习网络可以涵盖多种因素。除了通常的超参数调整外，还需要关注网络架构本身。

优化网络架构

随着网络层数的增加或节点类型的增多，网络变得更加复杂，这直接影响损失/误差如何通过网络进行反向传播。图 1-2 展示了当深度学习系统变得更加复杂和庞大时最常遇到的问题。

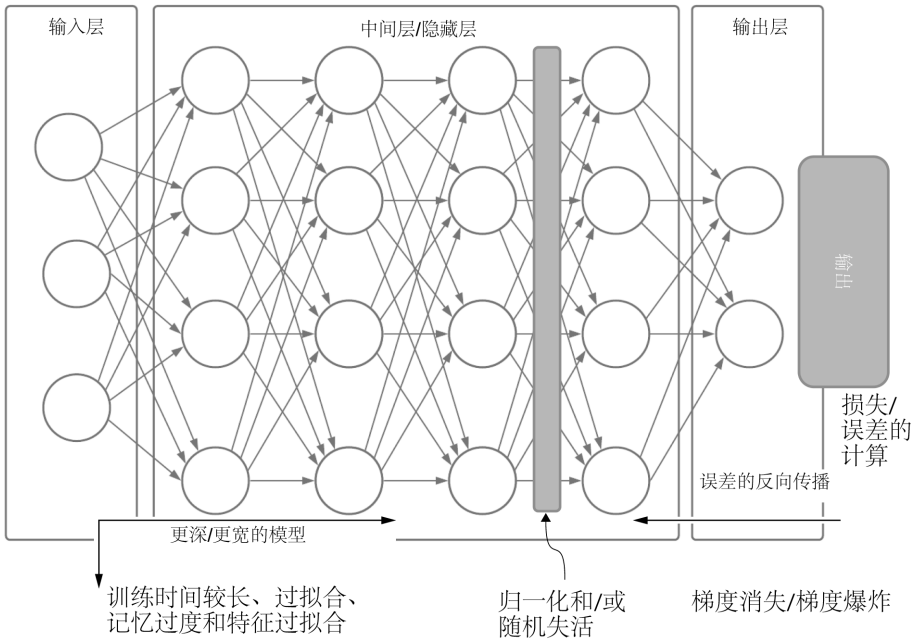

图 1-2 在扩展深度学习系统时常见的问题

在更大的网络中，损失量需要被分解成越来越小的组件，最终趋近于零。当这些损失组件或梯度趋近于零时，称为梯度消失(vanishing gradient)问题，这通常与深度网络相关。相反，这些组件也可能通过连续的层传递，使输入信号被放大。这导致梯度组件变得很大，称为梯度爆炸(exploding gradient)。

可以使用各种技术解决这两个梯度问题，比如对输入数据进行归一化处理，以及在网络的各个层中使用特殊类型的层函数，如图 1-2 所示的归一化和随机失活。这些技术增加了计算复杂性和对网络的要求，可能会过度平滑数据中重要和独特的特征。因此，为了获得良好的网络性能，需要更大、更多样化的训练数据集。

归一化可以解决深度网络的梯度消失和梯度爆炸问题，但随着模型的增长，会出现其他问题。例如，随着模型的增长，模型处理更大的输入集和图像的能力增强。然而，这可能会引起一个被称为网络记忆化的副作用，尤其是当输入的训练集太小时。发生这种情况是因为网络非常庞大，它可能开始记住输入块的集合，甚至整个图像或文本集合。

你可能听说过一些尖端的深度学习模型，比如来自 OpenAI 的自然语言处理器 GPT-3，它们在一定程度上受到记忆过度的影响。即使将代表多种形式文本的数十亿个文档输入这样的模型中，这个问题仍然存在。即使在如此多样化和庞大的训练数据集下，像 GPT-3 这样的模型也被发现会重播整个段落的记忆文本。这个“问题”可能是一个数据库的有效特征，但对于深度学习模型却不适合。

针对记忆过度问题已经开发出了一种称为随机失活(dropout)的解决方法，通过该方法，在每次训练过程中，网络层中一定比例的节点可能会被停用。通过在每次训练过程中打开和关闭节点，可以创建一个更加通用的网络。然而，这样做的代价是需要将网络的规模增加 100%～200%。

除了这些问题之外，将更多层添加到深度网络中会增加更多的权重，这些权重需要在数十亿甚至数万亿次迭代中进行逐个训练。训练这样的模型需要指数级增长的计算能力，而现在只有那些能够承担高昂成本的组织或公司才能开发出顶尖的模型。

许多人认为，对于大多数深度学习从业者来说，更宽更深的网络发展趋势很快会达到一个瓶颈，将任何未来的尖端发展留给像谷歌 DeepMind 这样的人工智能巨头。因此，简单的解决方案是寻找可以简化这些大型网络开发的替代方法。这就是我们重新将进化计算应用于深度学习，以优化网络架构和/或权重的原因。

幸运的是，EDL 提供了多种潜在的方法，可以自动优化网络的大小和形式，以解决本书将讨论的各种问题。自动优化是 EDL 的核心，本书中的许多练习将重点展示这些技术。

由于进化算法提供了多种优化模式，可以解决多种问题，因此 EDL 可以在机器学习开发过程的各个方面发挥作用。这些方面包括调整模型超参数以适应数据或特征工程、模型验证、模型选择和架构选择等。

1.4　用自动化机器学习实现自动优化

EDL 提供了一套工具，可以帮助自动优化深度学习系统，以获得更强健的模型。因此，它应被视为一种 AutoML(自动机器学习)工具。许多商业化的 AutoML 平台，如 Google AutoML，使用各种进化方法开发模型。

在继续之前，还需要讨论自动化机器学习(Automated Machine Learning，AML)和 AutoML 这些术语的命名或误用。本书会交替使用 AML 和 AutoML，它们通常被认为是相同的，对于我们的目的来说也是如此。然而，在某种程度上，AML 和 AutoML 可能被认为是不同的，前者通常用来描述产生优化模型的黑盒系统。

自动化优化和开发任何人工智能/机器学习模型被认为是所有研发项目开发过程

中的下一步。这是超越研究和开发的进化阶段，将模型构建过程形式化，使从业者能够将模型推向全面的商业化和产品化。

什么是自动化机器学习

自动化机器学习(AML)或AutoML是用于自动化和增强人工智能/机器学习构建的工具或一组工具。它不是一种具体的技术，而是一系列方法和策略的集合，进化算法或进化优化方法被视为其中的一个子集。它是一个可以在整个人工智能/机器学习工作流程中使用的工具，如图 1-3 所示。

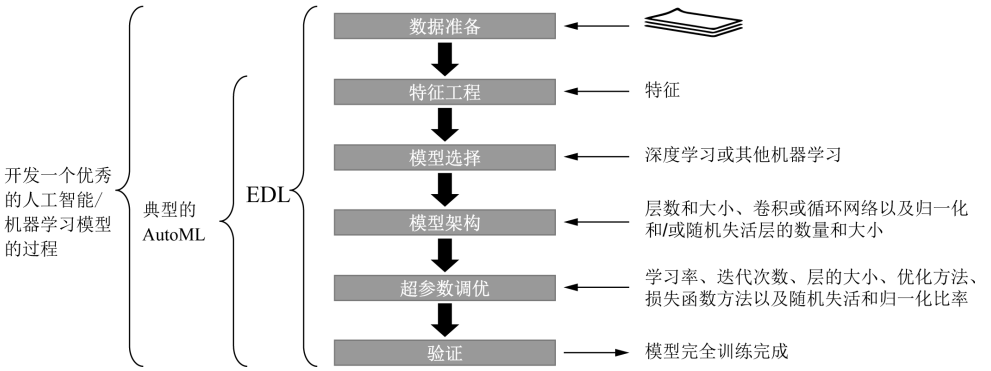

图 1-3 使用 AutoML 和/或 EDL 开发良好的人工智能/机器学习模型的步骤

AutoML 工具

以下是提供 AutoML 的工具和平台列表。

- DataRobot: 作为 AutoML 的第一个平台和起点，DataRobot 提供了一组不同的工具来自动构建模型。
- Google Cloud AutoML: 这个流行且强大的平台来自当前人工智能领域的主要参与者。这个平台处理各种各样的数据，从图像到结构化数据。
- Amazon SageMaker AutoPilot: 这个强大的平台对于依赖结构化数据的开发模型自动化很有用。
- H2O AutoML: 该工具提供了各种用于自动化机器学习工作流程的功能。
- Azure Machine Learning: 该平台提供了对各种形式数据进行模型调整的自动化过程。
- AutoKeras: 这个优秀的工具提供了网络架构的自动化开发。
- AutoTorch: 该工具提供了自动化的架构搜索。

在本列表范围之外，还有许多其他工具和平台可供使用。

图 1-3 描述了用于构建一个能够自信地推断新数据的良好模型的典型人工智能/机

器学习工作流程。这个工作流程通常由各种人工智能/机器学习的从业者手动完成，但也有自动化所有步骤的各种尝试。以下是对每个步骤的更详细总结，包括它们如何通过 AML 进行自动化。

- 数据准备：为人工智能/机器学习训练准备数据耗时且昂贵。通常，准备数据并自动化这个任务可以显著提高用于优化复杂模型的数据工作流程的性能。AutoML 在线服务通常假设用户已经准备并清洗了数据，这是大多数机器学习模型所需要的。通过进化方法，有几种方法可以自动化数据的准备工作，虽然这个任务不是 EDL 特有的，但将在后面的章节中介绍。
- 特征工程：这是利用先前的领域知识从数据中提取相关特征的过程，专家根据他们的直觉和经验选择相关特征。由于领域专家的成本高且有主观意见，因此自动化该任务可以降低成本并提高标准化程度。根据 AutoML 工具的不同，特征工程可能会包含在这个过程中。
- 模型选择：随着人工智能/机器学习的发展，已经创建了数百种能够解决相似问题的模型。通常情况下，数据科学家会花费几天甚至几周的时间来选择一组模型进行进一步评估。自动化该过程可以加快模型的开发，并帮助数据科学家确认他们正在使用正确的模型。一个好的 AutoML 工具可以从数十种甚至数百种模型中进行选择，包括深度学习变体或模型集成。
- 模型架构：根据人工智能/机器学习和深度学习的领域不同，定义正确的模型架构通常非常关键。以自动化的方式正确地完成这一步骤可以节省大量调整架构和重新运行模型所需要的时间。根据具体实现，一些 AutoML 系统的模型架构可能会有所不同，但通常是一些众所周知的变体。
- 超参数优化：微调模型超参数的过程可能耗时且容易出错。为了克服这些问题，许多从业者依赖于直觉和以往的经验。尽管这在过去是可行的，但随着模型越来越复杂，这个任务变得难以承受。通过自动化超参数调优，不仅可以减轻建模者的工作负担，还可以发现模型选择或架构中的潜在缺陷。
- 验证集选择：有许多选项可以评估模型的性能，从决定用多少数据进行训练和测试，到可视化模型的输出性能。对模型的验证进行自动化提供了一种强大的方式，可以在数据发生变化时重新评估模型的性能，并使模型在长期更具可解释性。对于在线的 AutoML 服务而言，这是一个关键的优势，也是使用这类工具的一个有力理由。

典型的 AML/AutoML 工作流程通常只试图解决特征工程步骤及其后续步骤，该过程通常以迭代方式完成，可以是单个步骤或多个步骤的组合。有些步骤，如超参数调优，是针对具体模型类型的，对于深度学习而言，可能需要大量的时间来优化模型。

尽管这种新型商业化 AutoML 服务能够成功地处理各种类型和形式的数据，但生

成的模型缺乏创新性且成本较高。为了完成 AutoML 构建调优模型所需要的所有任务，需要大量的计算能力。而所开发的模型实质上是前代基准模型的重构，往往缺乏新颖的优化见解。

那些想要在预算有限的情况下获得更具创新性的自动化模型的人工智能/机器学习从业者常常会转向开发自己的 AutoML 解决方案，而 EDL 则是一个主要选择。正如本书后面所述，进化方法可以提供各种解决方案，用于自动构建并优化深度学习模型、超参数、特征工程和网络架构。

1.5 进化深度学习的应用

既然理解了为什么需要将进化计算和深度学习结合到 AutoML 解决方案中，那么可以继续讨论如何实现。也就是说，我们如何将遗传算法等方法应用于深度学习，以改进人工智能解决方案的工作方式？有无数种可能可以将进化计算与深度学习合并，但在本书，我们将坚持一些基本的实用策略。

了解这些策略将使你能够修改现有的深度学习网络或创建自己的进化计算/深度学习组合模型。这将让你能够更快地创建先进的优化网络并使用更少的资源，使你能够选择策略，甚至随着经验的积累，开发新的策略。

为了实现这些崇高的目标，我们将从基础开始探索深度学习和进化计算的基本原理。我们将构建基本模型来解决这两个子领域的问题，然后在后面的章节中，将探讨如何将它们结合起来以提高性能和自动化水平。

进化计算可以以多种形式应用于深度学习，涵盖了多种自动化策略，这些策略可以包含在 AutoML 中。图 1-4 展示了可以应用于深度学习的各种进化计算或 EDL 子集，以及它们在人工智能/机器学习模型开发工作流程中的应用位置。

1.5.1 模型选择：权重搜索

如前所述，选择的基础模型和层类型通常由要解决的问题的类型决定。在大多数情况下，可以快速手动完成模型选择的优化。然而，模型选择不仅涉及选择层类型，还包括选择优化形式、初始权重和用于训练模型的损失函数。

通过优化模型的层类型、优化机制甚至损失函数形式，可以使网络更加稳健，更有效地学习。我们将看到一些示例，对初始模型权重、优化类型和损失指标进行调整，以适应各种问题。

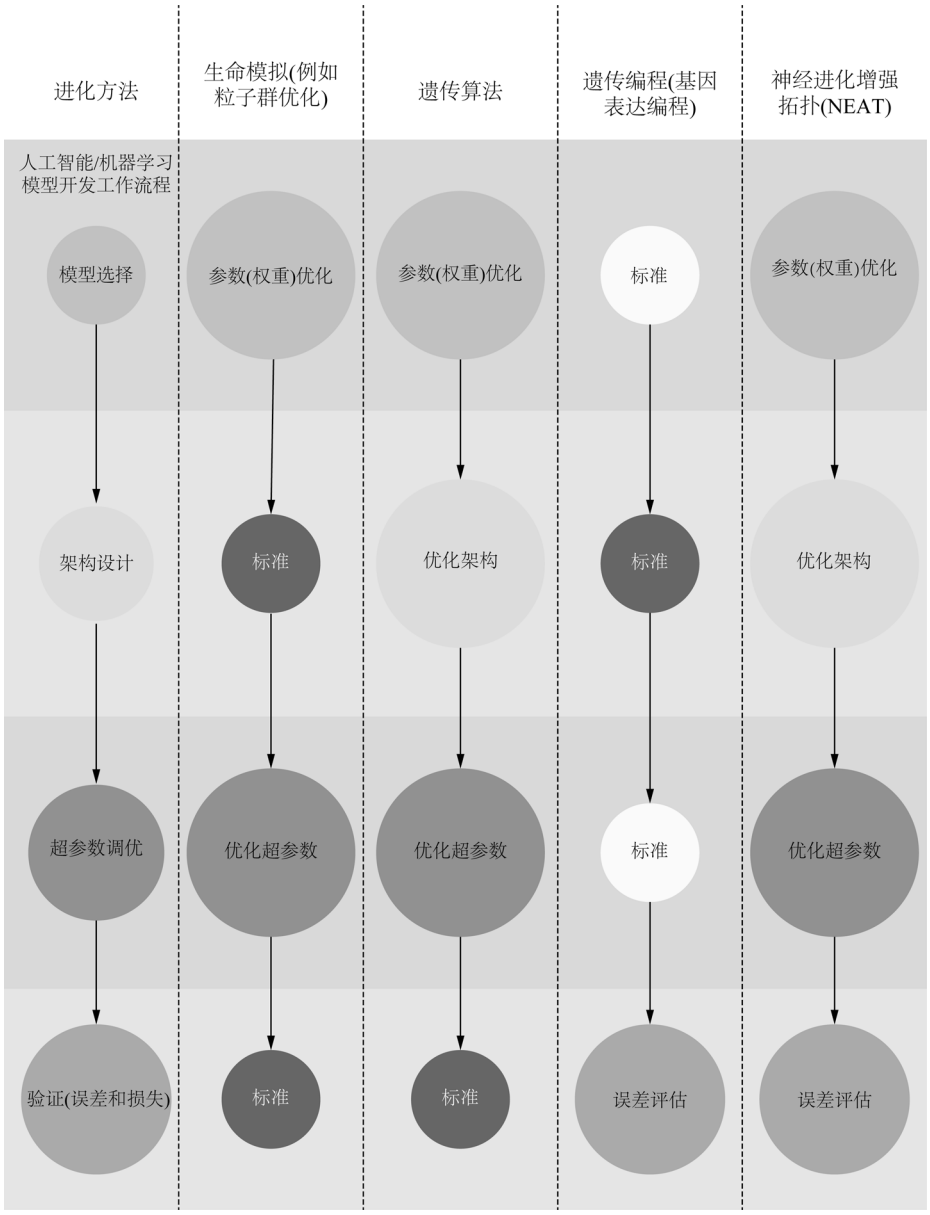

图 1-4　将进化计算应用于深度学习的人工智能/机器学习模型开发流程

1.5.2　模型架构：架构优化

在构建深度学习网络时，往往会过度设计模型或者增加模型中的节点和层数。然后随着时间的推移，会对网络进行缩减，使其在解决问题时更加优化。在许多情况下，网络过大可能导致输入数据的记忆过度，从而造成过拟合。相反，对于学习数据的种

类和数量过小的网络，通常会导致欠拟合。

为了解决过拟合和欠拟合问题，可以应用遗传算法来自动精简网络至其最低形态。这不仅提高了模型性能并限制了过拟合或欠拟合问题，还通过减小网络大小来缩短训练时间。这种技术在优化较大、较深的网络时效果很好。

1.5.3 超参数调优

超参数调优是我们在人工智能和机器学习中进行的一个过程，通过调整定义模型的各种控制变量来优化模型。在深度学习中，参数用于表示模型的权重。我们通过将这些控制变量称为超参数来区分它们。

进化算法为添加自动的超参数优化提供了多种替代措施，适用于各种模型，包括深度学习。粒子群优化、差分进化和遗传算法都被成功地应用过。将在多种框架下探索这些方法，以评估其性能。

1.5.4 验证和损失函数的优化

在开发强大的深度学习模型时，经常依赖于几种已建立的模式来生成高质量的网络。这可能包括通过反复检查训练和测试损失来验证模型的训练效果和性能。我们希望确保这两个损失指标之间没有出现过大的差异。

在典型的监督学习训练场景中，经常会使用与标签比较相符的已建立的指标标准。随着更先进的生成式深度学习场景的出现，可以优化损失函数的形式甚至验证指标标准。

像自动编码器、嵌入层和生成对抗网络这样的网络架构提供了使用组合决策进行损失和模型验证的机会。使用进化计算，可以通过 AutoML 方式优化这些网络形式。

1.5.5 神经进化增强拓扑结构

神经进化增强拓扑(Neuroevolution of augmenting topology，NEAT)是一种将超参数和架构优化与权重搜索相结合的技术，用于自动构建新的深度学习模型，这些模型可能还会开发出自己的损失和验证方法。虽然 NEAT 几乎 20 年前就被开发出来，但直到最近才将其应用于深度学习和深度强化学习的各种应用。

1.5.6 目标

在本书中，将探讨前面提到的一系列技术及其在深度学习中的应用。我们专注于实用的技术，这些技术可以通过实际解决方案应用于各种问题，特别关注各种形式的 AML/AutoML 如何应用于优化深度学习系统和评估各种技术的性能。我们的重点还包

括进化方法之外的更广泛的技术。

在接下来的章节中，逐步介绍 AutoML 过程的各个部分，为熟悉深度学习的人引入关键概念。在介绍进化计算的基础知识后，继续展示超参数优化，然后是数据和特征工程、模型选项选择及模型架构。最后，进一步探讨更复杂的示例，旨在改进生成式深度学习和深度强化学习问题。

通过本书的学习，你将能够自如地描述和使用深度学习以及进化计算的某些子集，单独运用或结合运用来优化网络。你将能够构建模型来解决问题，同时了解哪种方法对特定类别的问题更有效，包括在不同优化和 AutoML 的应用中，能够将进化计算应用于深度学习模型。

1.6　本章小结

- 深度学习是一种强大的技术，能够解决许多人工智能和机器学习问题，但它很复杂，需要大量的数据，并且在开发、训练和优化过程中成本高昂。
- 进化计算(EC)是人工智能和机器学习的一个子领域，它的定义基于自然选择理论。尽管进化计算的发展速度不如深度学习，但它仍然提供了解决各种复杂问题的技术。
- EDL 是一个广泛的术语，涵盖了将进化方法与深度学习结合在一起的概念。神经进化、进化超参数优化和神经进化增强拓扑是 EDL 的示例。EDL 定义了进化计算方法的一个子集，可用于在机器学习工作流程的多个阶段自动化和改进深度学习模型的开发。

AML 和 AutoML 定义了一套旨在自动化整个人工智能和机器学习模型开发工作流程的工具和技术。许多形式的进化计算已经被用于自动化模型开发工作流程。谷歌和其他公司已经大量投资于开发 AutoML，以帮助消费者构建适合自己需求的强大模型。虽然这些服务功能强大，但通常它们的工作方式像一个黑盒子，限制了对新的尖端模型更灵活的定制。