AI 普及给嵌入式设计人员带来新挑战

Microchip Technology Inc.
触摸和手势业务部
副总监
Yann LeFaou

探讨了人工智能（AI）的普及给嵌入式设计人员带来的新挑战。在创建“边缘机器学习（ML）”应用时，设计人员必须确保其能有效运行，同时最大限度地降低处理器和存储开销，以及物联网（IoT）设备的功耗。

从监控和访问控制到智能工厂和预测性维护，基于机器学习（ML）模型构建的人工智能（AI）在工业物联网边缘处理应用中已变得无处不在。随着这种普及，支持 AI 的解决方案的构建已经变得“大众化”——从数据科学家的专业领域转为嵌入式系统设计人员也需要了解的领域。这种大众化带来的挑战在于，设计人员并不一定具备定义要解决的问题以及以最恰当方式捕获和组织数据的能力。此外，与消费类解决方案不同，工业 AI 实现的现有数据集很少，通常需要用户从头开始创建自己的数据集。

融入主流

AI 已经融入主流，深度学习和机器学习（DL 和 ML）是我们现在习以为常的许多应用的背后力量，这些应用包括自然语言处理、计算机视觉、预测性维护和数据挖掘。早期的 AI 实现是基于云或服务器的，需要大量的处理能力和存储空间，以及 AI/ML 应用与边缘（终端）之间的高带宽连接。尽管生成式 AI 应用（如 ChatGPT、DALL-E 和 Bard）仍然需要此类设置，但近年来已经出现了边缘处理的 AI，即在数据捕获点实时处理数据。边缘处理极大减少了对云的依赖，使整体系统/应用更快、需要更少的功耗并且成本更低。许多人认为安全性得到了提高，但更准确地说，主要的安全重点从保护云与终端之间的通信转移到了使边缘设备更安全。

边缘的 AI/ML 可以在传统的嵌入式系统上实现，这些系统的设计人员可以使用强大的微处理器、图形处理单元和丰富的存储器器件，即类似于 PC 的资源。然而，越来越多的商业和工业物联网设备需要在边缘具备 AI/ML 功能，这些设备通常硬件资源有限，而且在许多情况下由电池供电。

在资源和功耗受限的硬件上运行的边缘 AI/ML 的潜力催生了“TinyML”这一术语。实际用例涵盖工业（如预测性维护）、楼宇自动化（环境监控）、建筑施工（监督人员安全）和安防等领域。

数据流

AI（及其子集 ML）需要从数据捕获/收集到模型部署的工作流程（见图 1）。对于 TinyML 而言，由于嵌入式系统资源有限，因此每个工作流程阶段的优化至关重要。

例如，TinyML 的资源需求被认为是 1 MHz 到 400 MHz 的处理速度、2 KB 到 512 KB 的 RAM 和 32 KB 到 2 MB 的存储空间（闪存）。此外，150 µW 至 23.5 mW 的小功耗预算也常常带来挑战。

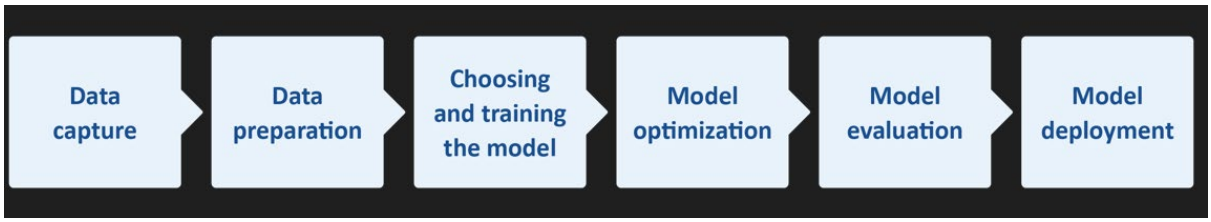

图 1——上图为简化的 AI 工作流程。虽然图中未显示，但模型部署本身必须将数据反馈回流程中，甚至可能影响数据的收集。

此外，在将 AI 嵌入资源有限的嵌入式系统时，还有更重要的考虑因素或权衡。模型是系统行为的关键，但设计人员经常发现自己在模型质量/精度（影响系统可靠性/依赖性和性能，主要是运行速度和功耗）之间做出妥协。

另一个关键因素是决定使用哪种类型的 AI/ML。通常有三种算法可供使用：监督学习、无监督学习和强化学习。

解决方案

即使是对 AI 和 ML 有良好理解的设计人员，可能也会在优化 AI/ML 工作流程的每个阶段并在模型精度与系统性能之间找到完美平衡方面遇到困难——那么缺乏以往经验的嵌入式设计人员如何应对这些挑战呢？

首先，重要的是不要忽视一个事实：如果模型小且 AI 任务仅限于解决简单问题，那么部署在资源有限的物联网设备上的模型将会更有效。

幸运的是，ML（特别是 TinyML）进入嵌入式系统领域，带来了新的（或增强的）集成开发环境（IDE）、软件工具、架构和模型——其中许多都是开源的。例如，TensorFlow™ Lite for Microcontrollers（TF Lite Micro）是一个面向 ML 和 AI 的免费开源软件库，它专为在只有几 KB 存储器的器件上实现 ML 而设计。此外，程序可以用开源和免费的 Python 语言编写。

关于 IDE，Microchip 的 MPLAB® X 就是此类环境的一个示例。该 IDE 可与公司的 MPLAB ML 一起使用，MPLAB ML 是专门开发的 MPLAB X 插件，用于构建优化的 AI 物联网传感器识别代码。MPLAB ML 由 AutoML 提供支持，可将 AI ML 工作流程的每一步完全自动化，无需重复、繁琐和耗时的模型构建。特征提取、训练、验证和测试确保满足单片机和微处理器存储器限制的优化模型，使开发人员能够快速在基于 Microchip Arm® Cortex®的 32 位 MCU 或 MPU 上创建和部署 ML 解决方案。

流程优化

工作流程优化任务可以通过使用现成的数据集和模型来简化。例如，如果一个支持 ML 的物联网设备需要图像识别，从现有的标记静态图像和视频片段数据集开始进行模型训练（测试和评估）是合理的；需要注意的是，监督学习算法需要标记数据。

许多图像数据集已经存在于计算机视觉应用中。然而，由于它们是为基于 PC、服务器或云的应用设计的，通常都很大。例如，ImageNet 包含超过 1400 万张标注图像。

根据 ML 应用的不同，可能只需要少量子集；例如，有很多人但只有少量静物的图像。例如，如果在建筑工地使用支持 ML 的摄像头，当有不戴安全帽的人进入其视野时，它们可以立即发出报警。ML 模型需要训练，但可能只需要少量戴或不戴安全帽的人的图像。然而，对于帽子类型，可能需要更大的数据集和足够的数据集范围，以考虑不同的光照条件等各种因素。

图 1 中第 1 步到第 3 步的内容分别是获得正确的实时（数据）输入和数据集、准备数据和训练模型。模型优化（第 4 步）通常是压缩，这有助于减少存储器需求（处理期间的 RAM 和用于存储的 NVM）和处理延迟。

在处理方面，许多 AI 算法（如卷积神经网络（CNN））在处理复杂模型时会遇到困难。一种流行的压缩技术是剪枝（见图 2），剪枝有四种类型：权重剪枝、单元/神经元剪枝和迭代剪枝。

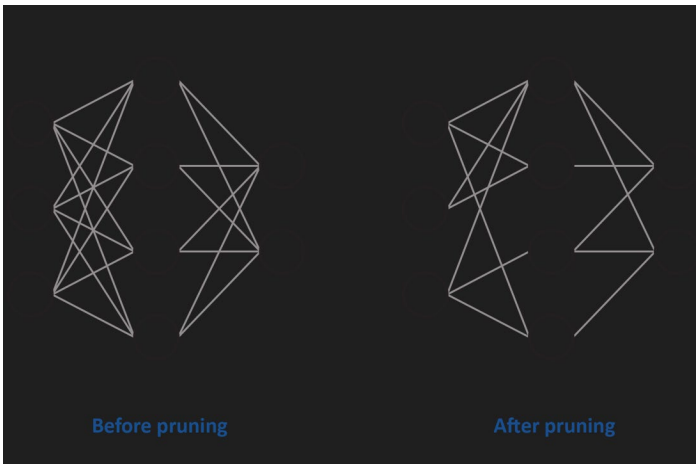

图 2——剪枝减少了神经网络的密度。上图中，某些神经元之间的连接权重被设为零。但有时神经元也可以被剪掉（图中未显示）。

量化是另一种流行的压缩技术。量化是将高精度格式（如 32 位浮点（FP32））的数据转换为低精度格式（如 8 位整数（INT8））的过程。量化模型（见图 3）的使用可以通过以下两种方式之一纳入机器训练。

- 训练后量化涉及使用 FP32 格式的模型，当训练完成后，再进行量化以便部署。例如，可以使用标准 TensorFlow 在 PC 上进行初始模型训练和优化。然后模型可以进行量化，并通过 TensorFlow Lite 嵌入到物联网设备中。
- 量化感知训练可仿真推断时量化，创建一个模型供下游工具用于生成量化模型。

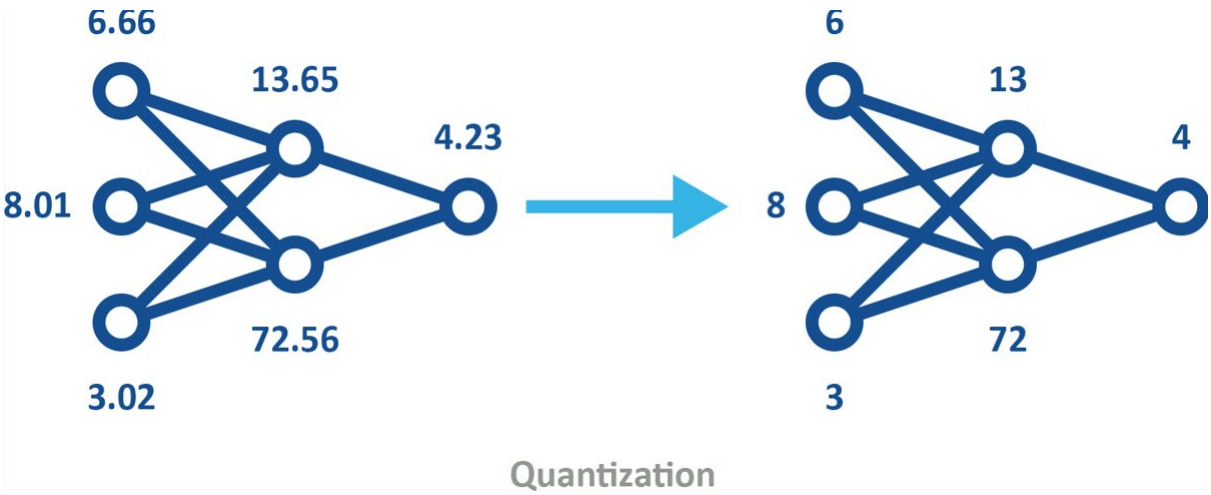

图 3——量化模型使用低精度，从而减少存储器和存储需求并提高能源效率，同时仍保留相同的形状。

虽然量化很有用，但不应过度使用，因为它类似于通过使用较少的位表示颜色和/或使用较少的像素来压缩数字图像——即，会存在一个图像变得难以解释的点。

总结

正如我们在开头所提到的，AI 现在已经深深融入嵌入式系统领域。然而，这种大众化意味着以前不需要了解 AI 和 ML 的设计工程师正面临将 AI 解决方案实现到其设计中的挑战。

尽管创建 ML 应用并充分利用有限硬件资源的挑战可能令人望而却步，但这对经验丰富的嵌入式系统设计人员来说并不是一个新挑战。好消息是，工程社区内有丰富的信息（和培训），以及像 MPLAB X 这样的 IDE、MPLAB ML 这样的模型构建工具以及各种开源数据集和模型。这种生态系统可帮助不同理解水平的工程师快速完成现在可以在 16 位甚至 8 位单片机上实现的 AL 和 ML 解决方案。

关于作者：

Yann LeFaou 是 Microchip 触摸和手势业务部的副总监。在这个职位中，LeFaou 领导一个团队开发电容式触摸技术，并推动公司在单片机和微处理器上的机器学习（ML）计划。他在 Microchip 担任过一系列连续的技术和市场职位，包括领导公司在电容式触摸、人机界面和家用电器技术方面的全球市场活动。LeFaou 拥有法国电力机械专业学院（ESME Sudria）的学位。