

# 智能电源管理系统中的机器学习

Paul Gorday, DSP 及机器学习业务总监

Vivek Chandrasekharan, 首席应用工程师

John Carpenter, 产品开发经理

## 摘要

用于 MCU 实施的机器学习 (tinyML) 技术是一个不断发展的领域; 其可为电池管理和电机控制带来新的增强功能。ML 算法能发现复杂传感器数据中的信息和模式, 用于优化性能并提高对整个系统健康状况的了解。除 tinyML 技术及其所取得的进步外, AutoML 工具的热度也在上升。这一工具可自动收集数据、训练 ML 算法, 以及生成和部署 MCU 固件; 其与片上系统 (SoC) 的传感器数据访问相结合, 可在目前电源管理系统中开发基于 ML 的解决方案。本文讨论了如何利用 Qorvo 的智能电源管理系统 IC 开发机器学习 (ML) 应用。Qorvo 高度集成的电源管理 SoC 将 Arm® Cortex® M0 和 M4F 内核 MCU 与带有一系列传感器的模拟前端相融合, 实现了智能控制与监控功能。

## 简介: 机器学习

### 从数据中学习

机器学习模型的性能直接源于数据; 这在复杂、高维数据中或人类难以确定最佳算法时成为一种优势。然而, 对数据的依赖意味着良好的结果取决于能否获得良好的训练数据——这些数据代表了设备在不同环境条件和制造公差下的实际行为。数据收集通常是 ML 项目中最昂贵、最耗时的阶段。在成本过高的情况下, 来自物理模型的合成数据可以满足要求。

Qorvo 的评估工具包包含一个图形用户界面 (GUI), 其中集成了数据记录功能, 以方便数据收集过程。内部 MCU 可用的相同传感器数据可保存到计算机上的文件中, 用于离线模型训练及测试。一些 AutoML (自动 ML) 工具还通过将硬件抽象层和数据流功能集成到评估固件中来支持实时数据收集和模型测试。

### 模型选择

机器学习的另一个优势是可扩展性; 仅需更换训练数据就能针对不同的用例训练单个模型。亦可使用参数化代码结构来实现模型, 并为每个用例存储一组不同的模型系数。此过程可以实现自动化, 从而节省与手动固件修改相关的工程时间和资源。

机器学习模型在复杂性和性能方面差异很大。决策树或树集合是大多数 AutoML 工具均包含的一种低复杂度方法；这些方法非常适合 MCU 实施，实施时可能仅需几百字节到几 K 字节的 RAM/闪存。其中，决策树通常面向分类应用，而称为回归树的相关结构可用于估算连续值。

分类决策树示例如图 1 所示。其中，放电过程中多电芯电池组的状态可以用两个特征来描述：最强电芯的充电状态，以及最强和最弱电芯间的电压差。在树的每个决策节点上，二元决策将特征空间一分为二。节点的连续层将特征空间细分为更小的区域，最终形成一个粗略的非线性决策边界；该边界试图将数据分为两类（如正常和异常）。该图中，训练数据显示为蓝色或白色点，而决策树使用的分类区域则以蓝色或绿色区域表示。

神经网络也包含在 AutoML 工具中。神经网络可提供平滑的决策边界和更高的性能，但代价在于实施复杂性的增加。图 2 显示了一个简单的 2 层神经网络模型图及分类区域；该网络采用与决策树示例相同的数据训练而成。网络中的每个神经元都对其输入进行加权求和，然后执行非线性激活函数。比较这两个示例中的分类区域可以看到，神经网络在处理训练过程中未见的数据时预期表现更好。不过，随着数据特征数量的增加，在 MCU 中实现复杂的神经网络将变得不切实际。

其它常见的 ML 模型包括支持向量机 (SVM)、核近似分类器、最近邻分类器、朴素贝叶斯分类器、逻辑回归和孤立森林。为获得最佳性能，很多 AutoML 工具都具备一个十分有用的功能，即自动模型选择和自动模型超参数调整（节点数、激活类型等）。还有一些在训练和比较模型时所需的工具，可用于目标处理器类型以及准确性、延迟和内存的资源预算[1]。

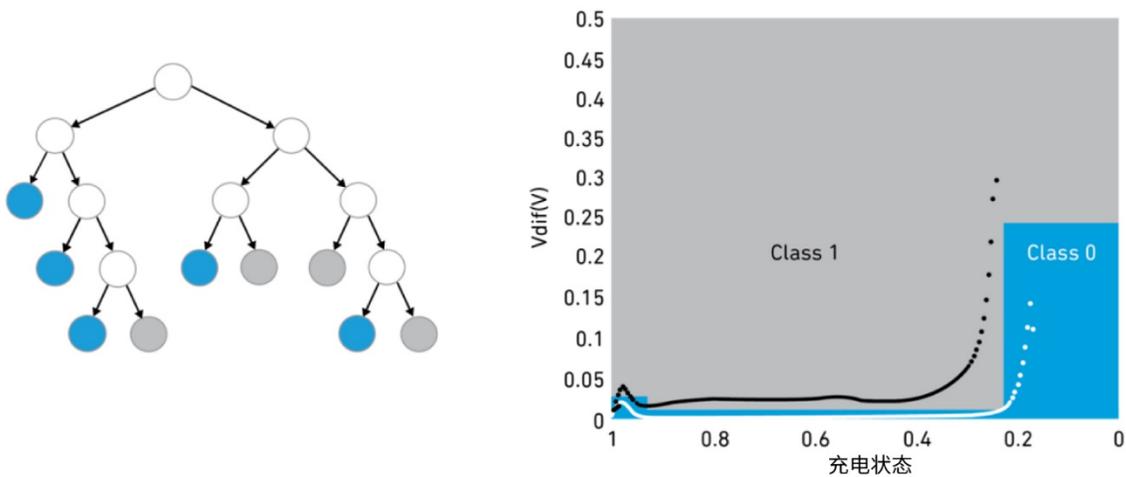
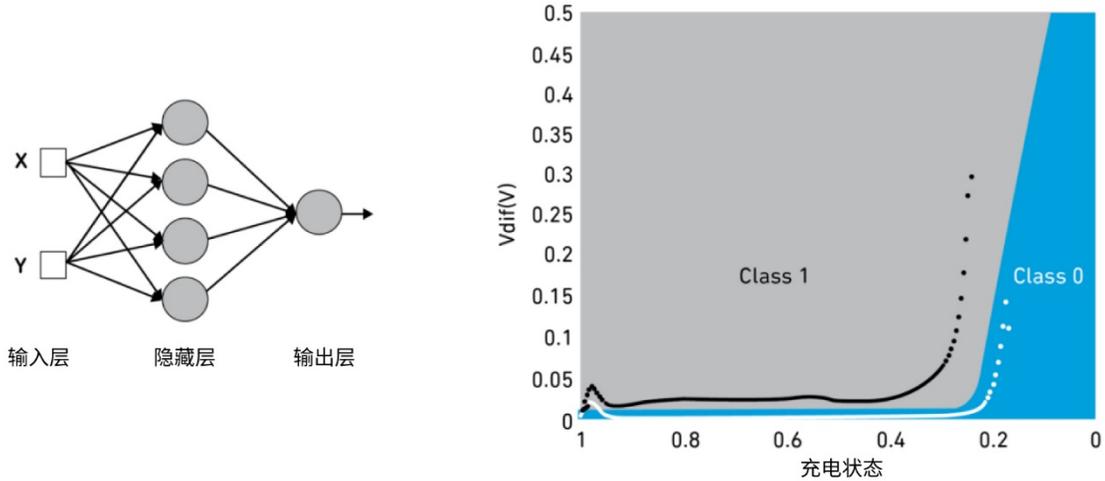


图 1，简单的决策树分类器



QORVO

© 2023 Qorvo US, Inc.

图 2，简单的神经网络分类器

### 模型部署

AutoML 工具的另一个重要特性是为经过训练的 ML 模型自动生成代码。在大多数情况下，代码生成可以针对特定的处理器，例如 Qorvo 智能电源管理 SoC 中使用的 Cortex M0 和 M4 处理器。这通常被称为工具的“无代码”或“低代码”功能，因为它消除或最大限度地减少了手工编码的需要。根据工具的不同，代码可以提供为可编辑的源代码，或以编译后的二进制文件提供，以便集成到电源管理固件中。

### ML 开发流程

成功的 ML 开发通常是一个迭代过程，涉及数据的收集和整理、模型的训练与分析，以及模型部署和监控（图 3）；术语“MLOps (ML Operations)”用于描述此过程[2]。性能监控是 MLOps 的关键要素，可确保实际性能与预期相符。差异往往可追溯到训练数据中未体现的现场条件，因此可能需要进行新一轮的数据收集与训练。

图 3 描述了智能电池管理系统 (BMS) 应用的 ML 开发流程。它利用了 Qorvo 的评估套件 (如 PAC25140EVK1 [3])，其中包括用于状态监控、硬件/软件设置和传感器数据记录的图形用户界面 (GUI)。.csv 格式的数据日志文件可导入 AutoML 工具，进行特征选择、模型训练、模型分析和代码生成。将生成的模型代码集成至 BMS 固件，并部署到 EVK 中进行物理测试和监控。

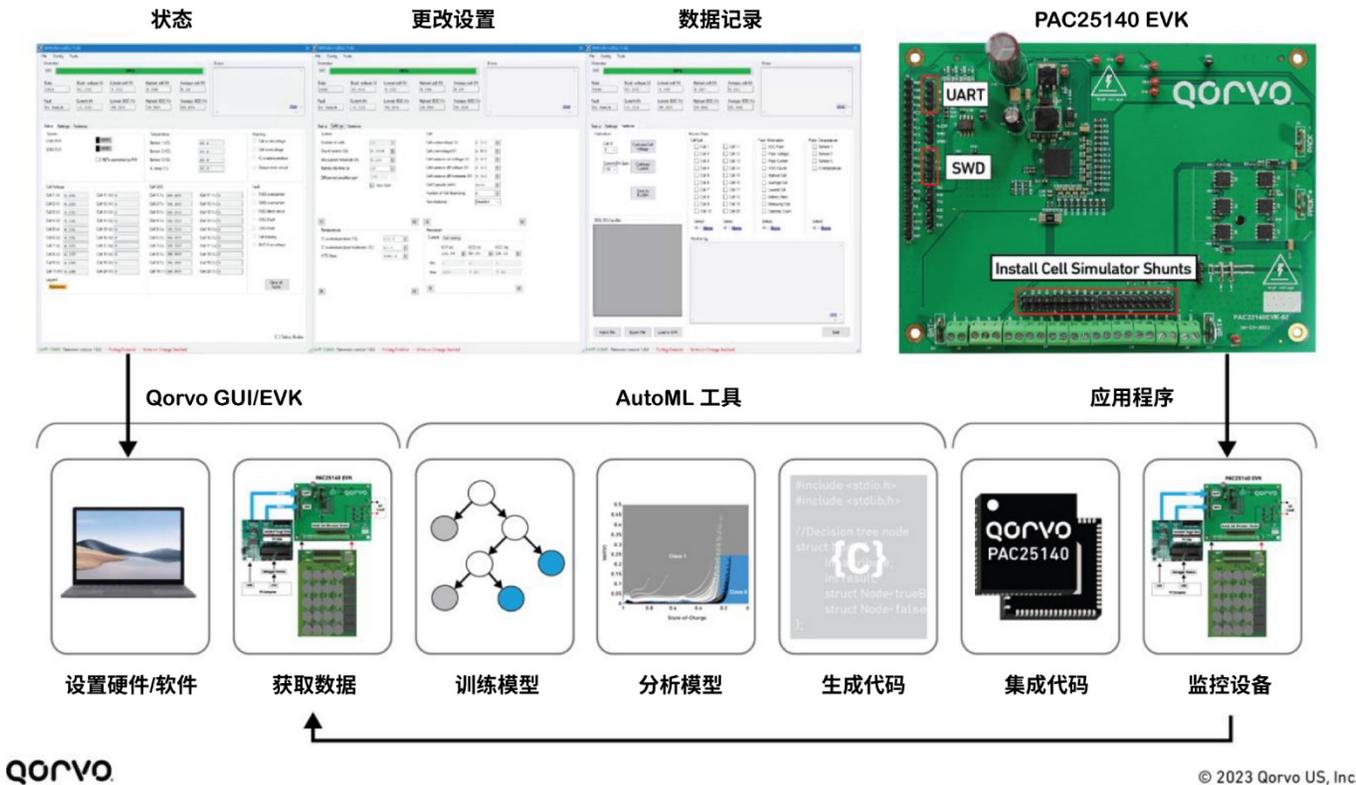
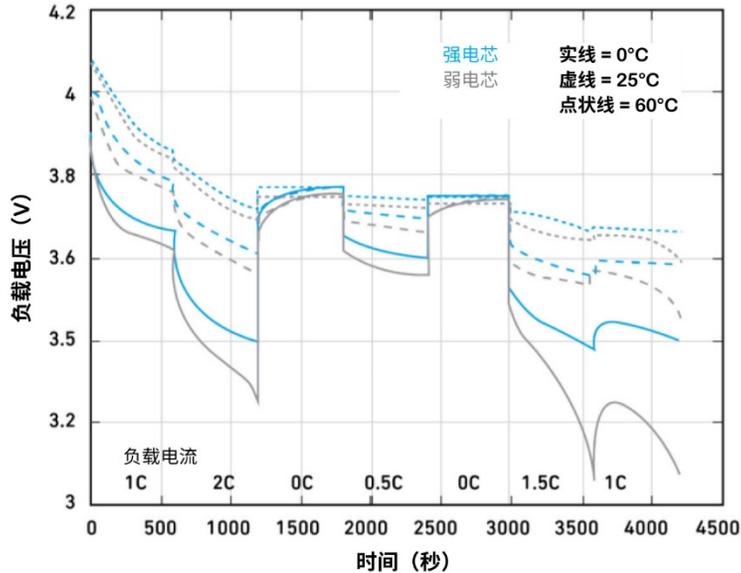


图 3, ML 开发流程

### 实施示例：BMS 弱电芯检测

检测多电芯电池组中的弱电芯是智能电池管理系统（BMS）的功能之一。一种直接的方法是比较电池组中所有电芯的负载电压，并确定其中某个电芯是否与其它电芯存在明显差异。图 4 显示了强电芯和弱电芯在不同温度下的放电周期。从图中可以看出，电流、环境温度和充电状态等特征都会影响电芯间的电压差；因此在电芯几乎耗尽之前很难通过单一的阈值测试来区分弱电芯。在放电周期的早期检测出弱电芯有助于最大限度地减少过热和相关安全问题。机器学习能够在复杂的特征集中发现某种模式，为这一问题提供了潜在的解决方案。Qorvo 的智能 BMS 解决方案可监控上述特征，并由嵌入式 MCU 中的 tinyML 算法对这些特征进行处理，从而提升性能。



QORVO

© 2023 Qorvo US, Inc.

图 4，强电芯和弱电芯的放电周期示例

为了进一步探索这种方法，我们借助两个 10s 电池组的 20 个放电周期创建了一组训练数据。第一个电池组（0 类）包括 10 个强电芯，第二个电池组（1 类）包括 9 个强电芯和 1 个弱电芯。在 20 次放电周期中，每次放电时，两个电池组的负载电流恒定在 0.2 至 4C（C 为电池容量）之间；环境温度恒定在 0 至 60 摄氏度之间。

选择以下四个特征来进行弱电芯检测：

- $i\_series$  = 通过电池组中所有电芯的串联电流
- $t\_amb$  = 环境温度
- $v\_diff$  = 电池组中最强和最弱电芯间的电压差
- $soc$  = 电池组中最强电芯的充电状态

对这四个特征进行同步采样，以创建训练实例  $[i\_series, t\_amb, soc, v\_diff]$ 。两类电芯的 20 个放电周期每 10 秒采样一次，共产生 11,000 多个训练实例。使用 Matlab 的“分类学习器”应用程序[4]将训练数据创建为多个 ML 模型。训练结束后，使用一组独立的测试数据（每个类别增加 20 个放电周期）重新评估分类准确性；结果汇总如下表 1 所示。请注意，表中的结果是对单个样本实例的分类。针对电池组的健康状况，预计将合并多个样本结果以形成更稳健的决策。

模型	训练数据			测试数据			复杂度
	准确率 (%)	误检率 (%)	漏检率 (%)	准确率 (%)	误检率 (%)	漏检率 (%)	
决策树 (深度=4, 分支=8)	96.1	2.9	5.0	94.3	6.1	5.2	低
决策树 (深度=6, 分支=18)	97.3	2.1	3.3	96.7	3.0	3.6	低
树集合 (深度=4, 30个决策树)	97.3	2.4	3.0	96.4	3.8	3.4	中
神经网络 (1个隐藏层, 4个节点)	97.1	3.7	1.9	97.5	3.6	1.3	低



表 1，弱电芯检测示例的模型训练结果

## 总结

基于 MCU 的机器学习（称为“tinyML”）作为一种新兴技术，可为电源管理应用提供前景广阔的增强功能；例如，用于提高安全性和可靠性的故障与异常检测、可降低系统成本的系统参数虚拟感测，以及可提升系统效率与可靠性的新型预测性维护等应用。Qorvo 的智能电源管理 SoC 包括 ARM Cortex MCU 和一系列模拟传感器，tinyML 算法可从中学习并改善性能。Qorvo 提供支持数据收集和固件测试的评估套件；这些套件与第三方 AutoML 工具相结合，为当今 SoC 的 ML 算法开发带来了必要的元素。

## 参考文献

1. State of the tinyAutoML Market, June 10, 2022. Available: <https://www.tinyml.org/event/auto-ml-forum/>
2. tinyML Deployment Working Group White Paper #1, February 20, 2023. Available: <https://www.tinyml.org/news/tinyml-deployment-working-group-white-paper>
3. PAC25140EVK1. Available: <https://www.qorvo.com/products/p/PAC25140#evaluation-tools>
4. Matlab Classification Learner. Available: <https://www.mathworks.com/help/stats/train-classification-models-in-classification-learner-app.html>