

AI 冲击！嵌入式开发如何转型？嵌入式开发者必看

AI 浪潮铺天盖地，嵌入式领域也被卷入其中。这一变化带来了前所未有的挑战，但同时也藏着许多机会。

对于嵌入式开发者来说，如何在这波浪潮里，从被冲击到与 AI 融合，找到自己的发展方向，是当下最需要解决的问题。



01

AI给嵌入式带来的改变：从挑战到新机遇

传统嵌入式开发遇到的难题：

算力和资源不匹配：AI 技术不断进步，像 CNN、RNN 这些 AI 模型，对计算资源的要求越来越高。但嵌入式设备因为使用场景的关系，一般都得做到低功耗、小内存。

比如说，一个普通的 CNN 模型运行起来，可能需要好几 GB 的内存和很厉害的计算核心支持，可大多数嵌入式设备的内存也就几十 MB 甚至更少，计算能力也不强。这么大的差距，让传统嵌入式开发在运行 AI 模型时困难重重。

开发思路大转变：以前的嵌入式开发，主要是按照确定的规则来编程，用 if - else 这样的条件语句实现各种功能。

但 AI 开发不一样，它靠数据驱动，得收集、整理、标注大量数据，然后训练和优化模型。这种开发思路的改变，要求开发者不能只盯着程序逻辑设计，还要更加关注数据质量和模型性能。

工具跟不上：以前嵌入式开发者用惯的工具，在面对 AI 开发时就不好使了。他们得去适应新的 AI 框架，像 TensorFlow Lite、PyTorch Mobile，还有边缘计算工具。

02

AI 给嵌入式带来的新机会

边缘智能兴起：AI 技术往边缘端发展，在本地设备上进行 AI 推理变得越来越重要。就像人脸识别门禁系统、语音唤醒设备，在本地就能完成 AI 推理，不用老是依赖云端，反应更快，还更能保护隐私。

据 ABI Research 预测，到 2026 年，边缘 AI 芯片市场规模会达到 [X] 亿美元，每年增长超过 [X]%，这足以看出边缘智能市场潜力巨大。

硬件升级：AI 专用芯片（NPU、TPU）越来越普及，让嵌入式设备的 AI 处理能力更强了。

比如 STM32 AI 套件，把 STM32 微控制器低功耗、高性能的特点，和 AI 加速功能结合起来，开发者用它就能很容易在嵌入式设备上实现 AI 应用。

还有树莓派搭配 Edge TPU，也给开发者提供了方便的 AI 开发平台。

应用场景变多：智能家居、工业预测性维护、自动驾驶感知层等领域，对嵌入式 AI 的需求一下子就爆发了。

在智能家居里，有了嵌入式 AI 技术，智能音箱识别语音、理解语义更准；在工业领域，实时分析设备运行数据，能提前预测设备故障，减少停机时间；

在自动驾驶感知层，嵌入式 AI 能快速识别路况、行人和车辆，保障行车安全。

03

嵌入式开发者怎么转型？4个关键方法

方法1：补充 AI 基础知识

(1) 入门必学：机器学习基础，包括监督学习和无监督学习，这是理解 AI 算法的基础。开发者得掌握不同机器学习算法的原理和适用场景，

比如决策树、支持向量机、聚类算法这些。神经网络原理是深入了解 AI 的关键，得弄明白神经元怎么工作、神经网络的结构和训练方法。

另外，从数据预处理到模型评估、优化的整个模型训练流程，也得了解清楚。

(2) 轻量化技术：因为嵌入式设备资源有限，掌握模型压缩技术就特别重要。量化就是降低数据的表示精度，在不太影响模型性能的情况下，减少计算量和内存占用；剪枝是去掉模型里不重要的连接或神经元，简化模型结构；

知识蒸馏是把大模型的知识转移到小模型里，提升小模型的性能。TinyML 技术专注在资源少的微控制器上实现 AI 应用，开发者要学会把复杂的 AI 模型适配到这些小设备上。

(3) 推荐工具：Edge Impulse 是个不用写代码的嵌入式 AI 平台，从收集数据、训练模型到部署，一站式服务，特别适合新手快速上手。OpenMV 是专门用来做机器视觉开发的开发板，集成了摄像头和很强的图像处理能力，配合 Python 语言，开发者能轻松实现各种机器视觉应用。

方法2：提升软硬协同优化能力

(1) 选对硬件：选支持 AI 加速的 MCU 是实现嵌入式 AI 应用的重要一步。比如 ESP32 - S3，它里面有神经网络处理器（NPU），能高效处理 AI 任务，还能保持低功耗运行。NXP i.MX RT 系列计算能力强，外设接口也丰富，是嵌入式 AI 开发的热门选择。

(2) 适配模型：把 AI 模型转换成嵌入式设备能兼容的格式，像 ONNX、TFLite，这样模型就能在嵌入式设备上正常运行。转换的时候，要优化模型，减少内存占用，提高推理速度。比如说，通过模型量化技术，可以把 32 位浮点数模型变成 8 位整数模型，既能降低内存占用，还能提高计算效率。

(3) 控制能耗：在嵌入式设备里，能耗控制一直是个重要问题。通过动态电压频率调整（DVFS）技术，根据设备的工作负载，实时调整电压和频率，既能保证算力需求，又能最大程度降低功耗。设备空闲的时候，就降低电压和频率，减少能耗；有 AI 任务要处理时，再提高电压和频率，满足计算需求。

方法3：专注垂直领域场景

工业领域：设备故障预测是工业里的一个重要应用场景。在设备上装上振动传感器，实时采集设备的振动数据，再用时序模型分析，就能提前预测设备可能出现的故障。视觉质检在工业生产里也必不可少，用 YOLO Tiny 等轻量级目标检测模型，搭配摄像头模组，能快速准确检测产品缺陷。

消费电子：语音交互在消费电子领域已经很常见了，像智能音箱、智能手表。通过 Wake Word Detection 技术，设备能准确识别用户的唤醒词，快速做出响应。手势控制也是个热门方向，用 CNN 结合 IMU 传感器，能识别和解析用户的手势，让交互更方便。

农业与环保：在农业领域，边缘端图像分类技术可以用来识别病虫害。在农田里安装摄像头，实时采集农作物图像，用 AI 模型分析，就能及时发现病虫害，采取防治措施。在环保领域，把温湿度等传感器数据融合起来，结合 AI 预测模型，能实时监测和预测环境变化。

方法4：参与开源社区和实战

学习资源：Kaggle 上的微型竞赛给开发者提供了实践 AI 技术的平台，比如 MCU 图像分类竞赛，开发者能在竞赛里学到最新算法和技术，还能和其他开发者交流经验。

GitHub 上也有很多嵌入式 AI 项目，把这些项目复现一遍，开发者就能深入了解嵌入式 AI 的开发流程和技术要点。

开发板实战：用 Arduino Nano 33 BLE Sense 可以实现关键词识别，这个开发板集成了多种传感器和蓝牙低功耗功能，很适合开发语音识别应用。

Jetson Nano 是功能强大的 AI 开发板，在上面部署目标检测模型，开发者能深入了解 AI 模型在嵌入式设备上的部署和优化过程。

学习路径规划：从新手到高手

(1) 基础层：掌握嵌入式和 AI 交叉技能

必学内容：Python 是一种简单又好用的编程语言，在 AI 开发里用得很多。掌握 Python 基础语法、数据结构，还有常用库，像 NumPy、Pandas 这些，是进行 AI 开发的基础。

PyTorch 和 TensorFlow 是现在最流行的两个 AI 框架，开发者要学会用它们构建、训练和部署 AI 模型。同时，C/C++ 是嵌入式开发的主要编程语言，在嵌入式系统里应用广泛，开发者得熟练掌握 C/C++ 的嵌入式开发技巧。

推荐课程：Coursera 上的《AI for Everyone》课程，由 AI 领域的知名专家授课，把 AI 的基本概念和应用讲得很明白，特别适合初学者入门。

Fast.ai 的《Practical Deep Learning》课程更注重实践，通过大量案例和项目，能帮助开发者快速掌握深度学习的实际应用。

(2) 进阶层：学会嵌入式 AI 全栈开发

核心技能：模型量化工具，比如 TensorFlow Lite Converter，能把训练好的模型转换成适合嵌入式设备运行的量化模型，提高模型运行效率。

嵌入式 RTOS（实时操作系统），像 FreeRTOS、Zephyr，能为嵌入式系统提供多任务管理、内存管理等功能，让开发者更高效地开发复杂的嵌入式 AI 应用。

项目案例：基于 STM32Cube.AI 的口罩检测系统，利用 STM32 微控制器的硬件资源和 STM32Cube.AI 工具的模型转换、优化功能，能实时检测人员是否佩戴口罩。

ESP32 – CAM 智能门铃开发，结合 ESP32 的 WiFi 功能和摄像头模块，还有 AI 图像识别技术，能识别访客并通知用户。

(3) 高层：实现自主优化与创新

深度优化：定制 AI 算子可以根据具体应用需求，优化 AI 算法，提高计算效率。利用 CUDA/OpenCL 等并行计算框架，能在 GPU 等硬件平台上实现高效的 AI 计算。

设计低功耗 AI 芯片，比如 RISC – V+NPU 架构，把 RISC – V 架构的灵活性和 NPU 的 AI 加速能力结合起来，给嵌入式 AI 应用提供更强大的硬件支持。

行业前沿：联邦学习在边缘端的应用，可以在保护数据隐私的前提下，让多方数据联合起来学习，提高 AI 模型的性能。

神经架构搜索（NAS）是通过自动化的方式，寻找最优的神经网络架构，为嵌入式 AI 应用提供更高效的模型结构。

最后

嵌入式开发者要以“嵌入式为根本，AI 为工具”，积极接受工具链的变化，专注实际场景应用。