Application Note
ZHCAG48 – Dec. 2025

AI 模型在 TI 处理器平台上的部署和调优

Kangjia Dong

摘要

在人工智能模型的边缘部署过程中，往往会因为算力限制、量化误差、算子差异等原因而导致模型部署困难，难以达到理想效果。Texas Instruments（TI）推出的 TIDL（TI Deep Learning）框架，作为一种针对嵌入式设备优化的深度学习推理引擎专为嵌入式 AI 应用优化，通过硬件加速与模型量化技术，在保证性能的同时尽量减少精度损失，实现快速的模型部署，为开发人员提供了强大的工具和技术支持。在深度学习应用的开发和部署过程中，精度问题是一个关键的挑战，本文将结合 TIDL 系统介绍在 TI 处理器平台上（TDA4X 和 AM6XA 系列）进行 AI 模型部署与精度调优的完整方法，结合实践案例和调试步骤，提供全面的指导。

目录

1 TI 处理器模型部署

TIDL（TI Deep Learning Library） 是 TI 平台基于深度学习算法的软件生态系统，可以运行在 TI 处理器平台上的 AI 推理加速框架，支持包括 AM62A、TDA4VM 等系列芯片，TIDL 运行在 C7x DSP 与专用的 MMA（Matrix Multiply Accelerator）上，能够显著提升卷积神经网络的推理速度，具有高效、灵活和可扩展的特点，可以将一些常见的深度学习算法模型快速的部署到 TI 处理器平台。TIDL 支持 ONNX、TensorFlow Lite、Caffe 等主流框架模型的转换与部署。Edgeai-tidl-tools 提供了模型量化、精度评估、离线仿真等功能，用于在 PC 端模拟 TIDL 运行效果，帮助开发者在部署前就能发现潜在的精度损失来源。通常情况下，将 AI 模型部署到 TI 处理器平台分为四个步骤：模型训练、编译模型、板端验证、应用集成。下文会对部署中的每一个步骤进行说明。

1.1 模型训练

随着社会的发展和科技的进步，AI 落地产品已经随处可见，从我们的日常生活到工业生产都可以看到他的身影。在日常开发中，算法工程师需要根据自己的业务的场景需求，如目标识别，语义分割等，选择已有的开源模型或者自行设计模型架构来满足应用需求。在设计完 AI 模型结构之后，还需要对数据集进行采集和标注，之后使用数据集在 TensorFLow, Pytorch 等框架训练出来对应的模型，然后进行精度验证，看是否满足需求。如图 1-1 所示，在模型训练阶段的支持情况和推荐操作，说明如下：

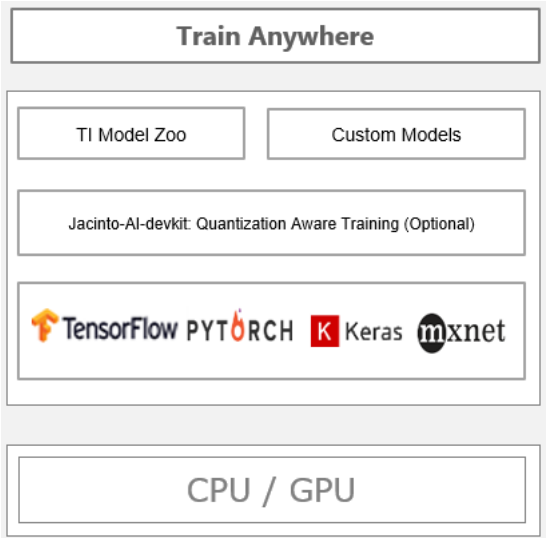


图 1-1. 模型训练

1. 在模型训练之前，可以参考 TI model Zoo 是否满足业务场景，可以基于 TI Model Zoo 中的模型更改或者重新训练来满足需求，可以参考以下链接： Edgeai-Model-Zoo
2. 如果 TI model Zoo 中的模型满足不了业务需求，可以自行设计模型结构。在使用自行设计模型结构时，需要考虑硬件加速器对应 AI 算子的支持，以达到后续部署推理速度。具体可以查看已经支持的 AI 算子说明。如现在遇到不支持的算子，可以通过以下几种方式解决：
    a. 咨询 E2E 是否有支持计划，如果有支持计划，后续可以升级 TIDL 软件组件来解决该问题。
    b. 在部署模型时，可以将不支持的算子放在 ARM 端计算或者自动生成到 C7 计算，通过编译模型时，更改模型编译参数(c7x_codegen)实现。

c. 开发者自行实现自定义算子来高效率支持模型的运行。

3. 在训练模型时，训练框架可以自由选择，可以使用 TensorFlow, Pytorch 等框架。其他训练框架下，可以将训练的模型转为 ONNX 标准格式来进行后续的部署。此部分主要取决于算法工程师的喜好，并不依赖于处理器平台。

1.2 模型编译

模型编译的步骤类似于编译器的功能，主要是将开源的模型编译转化为 TI 平台的模型格式进行推理加速，从而进行加速运算。整个过程在 TI 提供的工具中，可以一条命令，或者脚本来实现，具体参考文档（Edgeai-tidl-tool 或者 Demo Verify）。如果想深入了解具体的原理和实现流程可以参考图 1-2 所示，主要流程分为以下几个步骤：

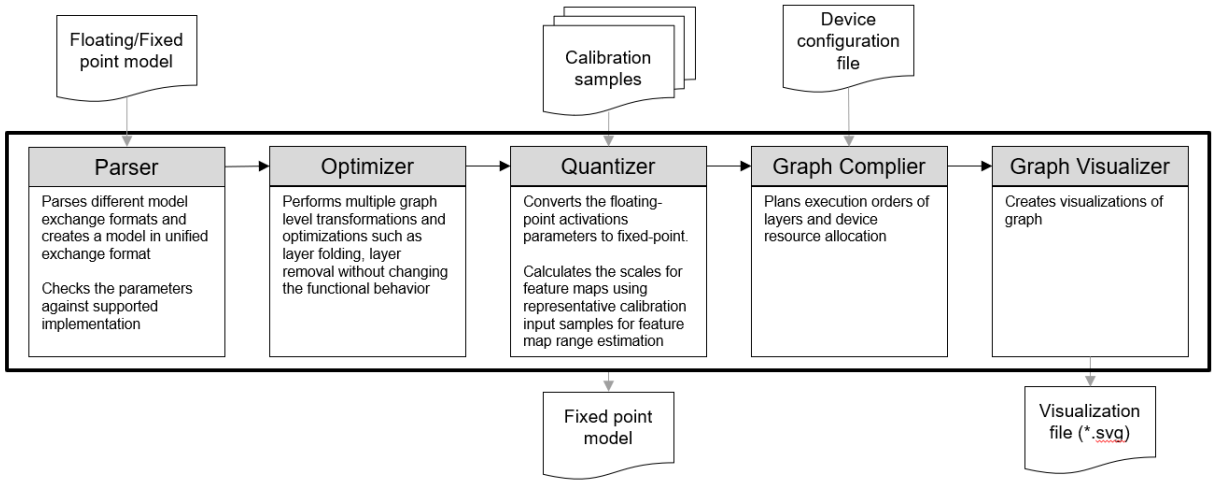


图 1-2. 模型编译

1. 首先获取训练好的模型权重和模型结构等信息。模型权重可以是 Float 或者 Int，模型格式是支持 ONNX、TFLITE 和 CAFFE 格式的模型，对于 Pytorch 格式的模型需要转化为 ONNX 格式的模型。具体参考大致参考以下代码：

```
import torch
input_vec = torch.rand(1,128,120,30) #参考模型的输入分辨率
path = "Conv2d.pth"  #Pytorch 生成的模型文件
model = torch.load(path)  #加载模型
torch.onnx.export(model, input_vec, "Conv2d.onnx", export_params=True, verbose=False,do_constant_folding=True, opset_version=11)  #格式转化 ONNX 格式
```

2. 模型优化会将一些算子合并以减少多层算子跳转运算的开销。如会把卷积层和 BatchNorm 算子合并，也会根据模型的整理结构将一些变换层，如 Slice 层或者 Transpose 合并成为其他层，此时并不会影响精度，但是会在推理时减少计算量 。

3. 量化是整个过程中最为关键的一步，主要是将在相关 Float 的权重/参数转化为 Int8/Int16，该过程会进一步缩减推理的计算量和减少内存带宽从而加速推理。TI 默认支持对称和非对称量化算法，在编译参数中进行配置，同时 TI 新一代架构中也支持预量化模型，即已经在训练时及已经将参数存储为 Int8/Int16，具体可以参考 Prequantized Models。

4. 结构编译会根据模型每层的输入和输出分配内存资源和 DMA 通道相关资源，用于板端计算时的吞吐。此过程会尽可能的将相关权重放在芯片内部的 RAM 进行计算，或者通过 Ping-Pong Buffer 的方式将每一层的权重不断替换到 RAM 中。

5. 结构可视化会将编译模型结构生成 SVG 格式的图片用于检查编译之后的模型结构，具体如图 1-3 所示。可以使用任意浏览器将该文件打开，鼠标停留到具体每一层可以进一步看到相关层的参数，如 Scale/Pad 等信息。

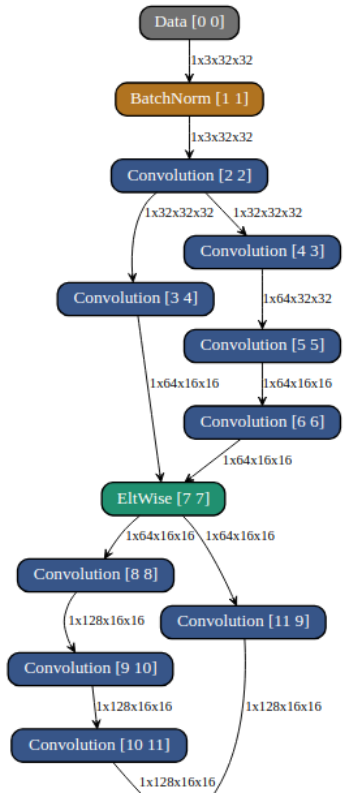


图 1-3 SVG 模型结构

1.3 板端验证

板端验证需要准备好板端环境以外，需要额外将模型编译之后生成两个文件分别，对应模型结构信息和模型权重信息拷贝到板端，原始输入可以是 BMP 格式图片或者 BIN 文件和模型对应分辨率相匹配。根据文件系统的 SDK 情况，采用不同工具验证，以下为不同 SDK 对应情况：

1. Processor SDK 验证可以参考 Demo Verify

2. Edge AI SDK 验证可以参考 Benchmark on TI SOC

板端验证时，可以设定相关参数将结果保存下来，从而看到推理的结果。

> ***注：****板端推理保存的结果为 Int8/Int16, 需要将最后输出的结果转为 Float 来查看推理精度，或者进一步进行后处理进行可视化模型的检测结果。*

1.4 应用集成

在确认模型精度和推理速度满足产品应用之后，需要对整个系统应用进行系统集成，具体可以根据使用的 SDK 情况，选择不同的示例进行更改集成，以下对两个不同的 SDK 分别进行说明：

1. Processor SDK 应用集成可以参考对应 SDK 的 Vision App 目录下的示例，根据自己的应用情况，基于现有的示例进行更改集成自己应用。图 1-4 Processor SDK 示例数据流，是将摄像头经过 CSI 传输进来的图像数据经过 ISP 处理，交给 C66 进行预处理将 YUV 转为 RGBP，然后进入到 C7X/MMA 进行推理，将推理的结构送给 C66 进行后处理，后处理的结果缩放到显示接口去显示。集成自己的应用时，根据自己模型的分辨率和数据输入格式更改对应的处理节点即可。

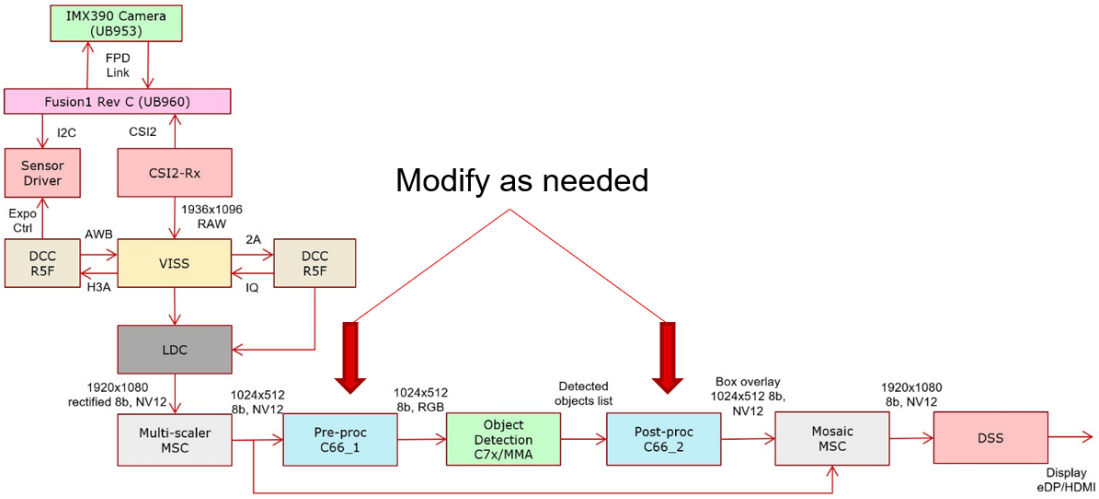


图 1-4 Processor SDK 示例

2. Edge AI SDK 应用集成可以参考对应 SDK 的 Edge AI 目录下的示例，同样根据自己的需求更改对应的处理节点即可。

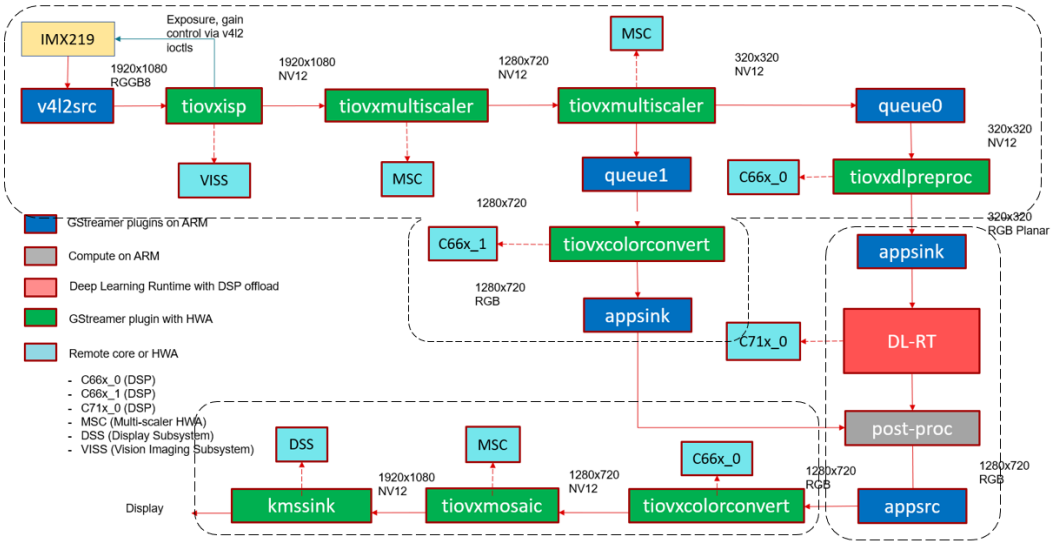
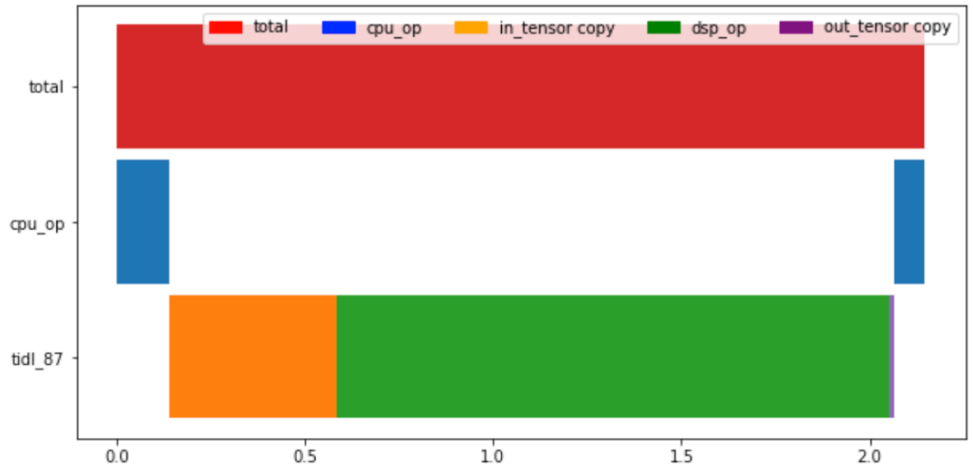


图 1-5 Edge AI SDK 示例

2 模型快速评估

在一些场景时，开发者想尽快评估验证自己的模型在 TI 平台推理时间和精度是否能满足应用开发的需求，但是身边又没有开发板的情况时，可以使用 Edge AI Studio 在云端来快速验证结果。该工具包含有以下几个子工具，具体如下：

1. Model Composer: 该工具提供了 TI 已经支持的模型，开发者只需上传自己数据集即可进行自己模型的训练，从而快速满足业务场景。
2. Model Analyzer: 该工具可以在云端选择需要评估的板子型号，进行远端控制板子进行性能评估，开发者可以选择 TI 已有的模型或者上传自己的模型进行评估。如图 2-1 显示了模型在板端的推理耗时以及占用资源。




```
TFL-CL-000-mobileNetV1 :
 Inferences Per Second    :  592.40 fps
 Inferece Time Per Image  :    1.69 ms
 DDR BW Per Image         :    3.31 MB
```

图 2-1 板端推理

3. Model Selection Tool: 该工具提供了 TI 模型池已经验证的相关模型的推理精度和速度，可以帮助开发者迅速知晓相关性能。
4. Model Development Tools for Programmers：提供开源的相关代码，可以对于满足高级开发者对自定义的需求。

3 模型部署问题与解决方法

在模型部署中，开发者会遇到一些问题，不知道如何解决，无从下手。不知道是本身 SDK 存在的 Bug 还是哪里误操作导致的问题。下文将详细说明，对一些问题如何快速定位并且解决。

3.1 常见问题

以下问题是基于基础环境能使用的情况下，**不包含环境搭建所产生的问题**。可以通过确认开发者 PC 是否能正常运行基础的编译示例，以及板端是否能运行基础的推理，来判断环境是否正常。具体可以参考 Demo Verify 来验证环境的问题。以下是一些常见的与 TI 平台模型部署的相关问题（在保证同样环境下，只有一个模型复现该问题在编译，TI 提供的多示例可以正常运行）：

1. 模型训练相关问题，如训练不收敛，精度达不到需求等等。

答：*模型训练与 TI 处理器平台无关，训练一般可以参考开源模型的训练过程，一般开源论坛会有一些答案。如果想避免模型的麻烦也可以找一些商业的公司来提供，行业有相关的公司可以提供商业的解决方案。*

2. 模型编译过程/板端验证出错，具体表现为编译过程/推理过程卡住或者有报错 log。

答： *在板端推理时，将 Debug Log 的等级打开，输出所有的 Log。根据使用的工具不同，具体参数为 debug_level=3 或者 debugTraceLevel = 3。在保证其他模型能正常编译/板端推理的情况下且使用最新版本的 SDK 进行编译/推理，只有一个模型卡住时，可以将该问题提交 E2E 进行问答。问题可能是 SDK 本身存在的一些 Bug，或者相关编译/推理参数配置不对。*

3. 板端推理结果相对于在 PC 端推理原始模型精度上有较大差距，此种场景下可以理解为推理结果完全对应不上，而不是百分比的精度损失。

答：*该问题需要逐步定位，并且找到问题点所在，具体可以分以下几个步骤进行逐级定位：*

a. *首先确认原始输入是一致，原始输入最好使用二进制文件或者 BMP 格式问题。（JPEG 文件在不同版本的 OpenvCV 处理之后得到的输入值不一样）*

b. *进一步确认将编译模型参数改成 32bit, 查看 PC 推理模型和 PC_TIDL Tool 推理模型的精度对比，如果此处发现精度基本一致，说明问题可能是由量化误差造成的。如果此时发现，精度对比差距较大，则按照之前所述提交 E2E。*

c. *进一步定位是由量化误差所导致还是有其他问题导致，可以将编译模型参数设置为 16bit，对比 Pytorch/Tensorflow，PC_TIDL Tool 以及板端推理结果，如果此时板端和 PC_TIDL Tool 结果一致但是和 Pytorch/Tensorflow 有差距，这里是由量化所造成的精度损失，如果板端和 PC_TIDL Tool 推理结果不一致，则按照之前所述提交 E2E。*

d. *进一步可能定位可能会发现，模型编译参数为 16bit 下精度满足需求，设置为 8bit，精度不满足需求。这里 8bit 不满情况也是两种情况，第一种是板端和 PC_TIDL Tool 结果不一致，此时按照 Bug 提交 E2E，另一种是量化算法造成的正常精度损失问题，该问题是一个大类的精度调优问题，将在下一节做进一步的说明。*

> ***注：****提交 E2E 论坛时，请说明芯片的型号，对应 SDK 的版本，可以复现问题的模型文件，模型文件对应的输入，编译/推理参数设置，以及编译/推理所产生的 Log，这些信息可以更快地让相关工程师复现问题并且深入的调试。*

3.2 精度调优

在确认该模型经过量化存在精度损失时，可以按照如下图 3-1 模型精度调优的步骤，进一步提升模型的精度，具体主要有以下几种方式：

1. 在训练模型时，使用 QAT 算法进行量化训练，这样可以在模型训练时达到一个 int8 的模型权重，从而避免 PTQ（后量化引入的精度损失），具体可以参考. ONNX QDQ Models 和 TFLite Pre-Quantized Models。
2. 在使用 PTQ 量化时，可以使用校准参数提升量化精度，可以参考编译参数。
3. 整个模型使用 16bit 量化，进而提升精度。
4. 使用混合量化，可以指定那一层用 16bit 或者使用自动混合量化来满足精度指标。此种场景下，需要考虑推理精度和推理性能的平衡。

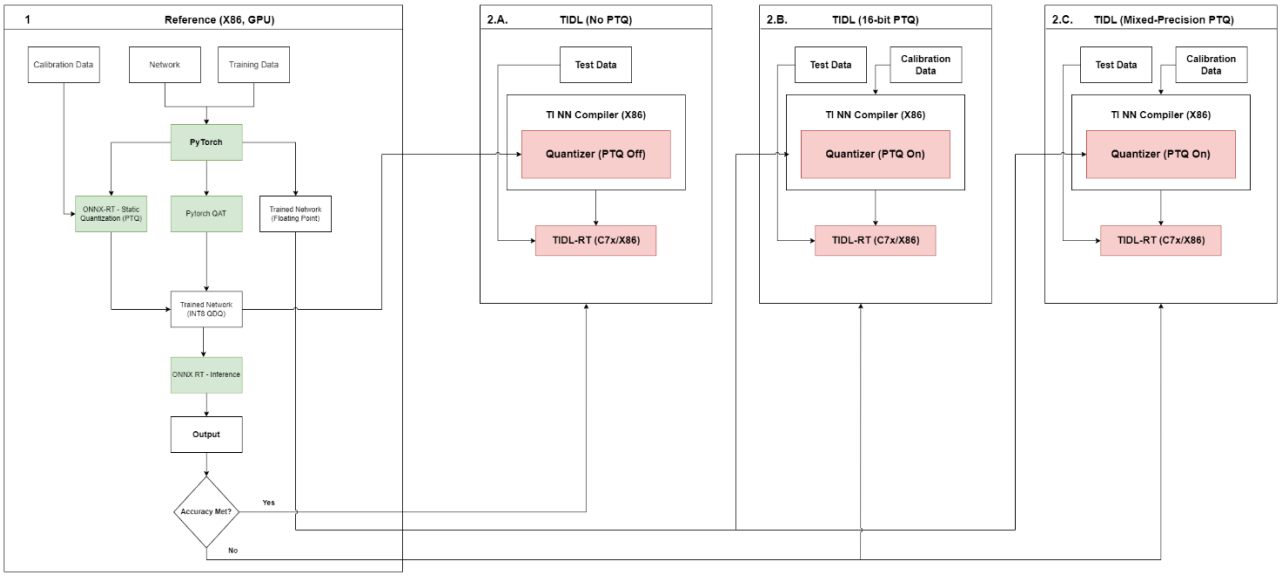


图 3-1 模型精度调优

3.3 模型性能

模型性能问题是一个非常常见的问题，该问题可以通过以下几种方式优化，具体如下：

1. 在模型设计阶段，TI 提供模型设计手册，尤其是卷积层的设计可以帮助实现更高的推理利用率，该部分文档可以联系对应的 FAE 获取。
2. TI 默认提供一些工具，可以协助在模型编译前做优化，可以参考优化示例。
3. 模型内存做进一步优化，从而提升性能，具体参考该文档。
4. 除此以外，TI 在板端提供模型推理的分析，可以打开该参数，从而显示每一层的推理时间，进而定位处是模型的哪一部分导致的推理时间变慢，可以将该部分模型结果替换为其他更为简易的结构，从而缩短推理时间，具体 Log 如下：

```
Network Cycles 6294273
Layer,  Layer Cycles,kernelOnlyCycles, coreLoopCycles,LayerSetupCycles,dmaPipeupCycles, dmaPipeDownCycles, PrefetchCycles ,copyKerCoeffCycles
  1,     81811,     48850,     49277,     7779,     14969,     18,     1007,     16,
  2,     71051,     52722,     53246,     1473,     3290,     16,     0,     0,
  3,     34063,     16700,     17307,     7379,     3952,     18,     17,     16,
  4,     60926,     45133,     45431,     6625,     4176,     18,     777,     9,
  5,     29990,     5996,     6040,     871,     3432,     9,     0,     0,
  6,     30806,     14975,     15275,     6575,     4114,     61,     10,     9,
  7,     20355,     5508,     5810,     6360,     3480,     11,     10,     9,
  8,     34670,     20921,     21031,     6222,     2291,     18,     727,     9,
```

在系统应用集成中，可能也会发现模型推理效果不及预期，这里可能是预处理时间或者后处理时间过长导致的整个链路的帧率收到影响，或者由于系统资源受限而导致的整体性能下降，这部分需要跟进开发者的具体应用强相关，这里不做进一步讨论。

4 总结

本文以在 TI 平台如何使用 AI 工具链并且部署模型为示例，对该过程遇到的问题进行分析和定位，可以在开发者使用带有 AI 加速器的 TI 处理器更快上手，同时在遇到一些问题时，具备单独调试和解决该问题的能力。

5 参考文献

1. Edgeai-tidl-tools
2. TI Deep Learning Library User Guide
3. Optimizing TI Deep Learning Performance on C7xMMA Processors via Memory and DDR Bandwidth Reduction
4. TI Deep Learning Library Upgrade Solution

重要通知和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做出任何明示或暗示的担保，包括但不限于对适销性、与某特定用途的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他安全、安保法规或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 TI 销售条款)、TI 通用质量指南 或 ti.com 上其他适用条款或 TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。 除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

最后更新日期：2025 年 10 月