

OpenVINO™ 工具套件

[入门指南](#)

[操作步骤](#)

[指南](#)

[资源](#)

[性能信息](#)

[API 参考](#)

[概述](#)

[指南](#)

[模型优化器开发人员指南](#)

[准备和优化您的训练模型](#)

[配置模型优化器](#)

[将模型转换为中间表示 \(IR\) 文件](#)

[使用常规转换参数转换模型](#)

[转换 Caffe* 模型](#)

[转换 TensorFlow* 模型](#)

[将 YOLO* 模型从 DarkNet 转换为中间表示文件](#)

[从 TensorFlow 转换 FaceNet 模型](#)

[从 TensorFlow 转换神经协作过滤模型](#)

[从 TensorFlow 转换 DeepSpeech 模型](#)

[利用 TensorFlow 的十亿单词基准转换语言模型](#)

[转换 TensorFlow* 对象检测 API 模型](#)

[转换 TensorFlow*-Slim 图像分类模型库模型](#)

[从 TensorFlow 转换 CRNN 模型](#)

[从 TensorFlow 转换 GNMT](#)

[从 TensorFlow 转换 BERT](#)

[转换 MXNet* 模型](#)

[从 MXNet 转换样式转移模型](#)

[转换您的 Kaldi* 模型](#)

[将 Kaldi* ASpiRE 链时延神经网络 \(TDNN\) 模型转换为中间表示文件](#)

[转换您的 ONNX* 模型](#)

[将 ONNX* Mask R-CNN 模型转换为中间表示文件](#)

[模型优化技术](#)

[切除模型的一部分](#)

[模型优化器中的子图替换](#)

[案例研究 \(已弃用\) : 转换使用 TensorFlow* 对象检测 API 创建的 SSD 模型](#)

[案例研究 \(已弃用\) : 转换使用 TensorFlow* 对象检测 API 创建的 Faster R-CNN 模型](#)

[支持的框架层](#)

[IR 符号参考](#)

[操作规范](#)

[模型优化器中的自定义层](#)

[使用新基元扩展模型优化器](#)

[使用新基元扩展 MXNet 模型优化器](#)

[使用自动生成功能创建模型优化器和推理引擎扩展](#)

[Caffe* 自定义层的传统模式](#)

[卸载子图推理](#)

[模型优化器常见问题解答](#)

[已知问题](#)

[中间表示和操作集](#)

[推理引擎开发人员指南](#)

[推理引擎简介](#)

[推理引擎 API 变更历史记录](#)

[推理引擎内存基元](#)

[推理引擎设备查询 API](#)

[nGraph 在推理引擎中的位置](#)

[推理引擎内核可扩展性](#)

[添加自定义 nGraph 操作](#)

[将推理引擎与您的应用集成](#)

[从推理引擎插件 API 迁移到核心 API](#)

[性能主题简介](#)

[推理引擎 Python* API 概述](#)

[使用 nGraph 构建模型](#)

[使用推理引擎神经网络 Builder 构建模型 \(已弃用 \)](#)

[图形调试功能](#)

[使用动态批处理功能](#)

[使用静态形状推理功能](#)

[使用 GPU 内核调整](#)

[使用低精度 8 位整数推理](#)

[用于验证转换后模型的实用程序](#)

[使用交叉检查工具在插件之间进行逐层比较](#)

[支持的设备](#)

[GPU 插件](#)

[GPU 插件的 RemoteBlob API](#)

[CPU 插件](#)

[FPGA 插件](#)

[VPU 插件](#)

[MYRIAD 插件](#)

[HDDL 插件](#)

[异构插件](#)

[多设备插件](#)

[GNA 插件](#)

[已知问题](#)

[优化声明](#)

[法律信息](#)

[词汇表](#)

[部署管理器指南](#)

[深度学习工作台开发人员指南](#)

[简介](#)

[安装深度学习工作台](#)

[从 Docker Hub* 安装](#)

[从英特尔® OpenVINO™ 工具套件分发版软件包安装](#)

[使用模型和示例数据集](#)

[选择模型](#)

[导入模型](#)

[导入 Frozen TensorFlow* SSD MobileNet v2 COC 教程](#)

[导入 MXNet* MobileNet v2 教程](#)

[导入 ONNX* MobileNet v2 教程](#)

[选择数据集](#)

[导入数据集](#)

[生成数据集](#)

[数据集类型](#)

[下载和剪切数据集](#)

[选择环境](#)

[运行基准推理](#)

[评估和解释模型性能](#)

[运行单一推理](#)

[运行推理范围](#)

[查看推理结果](#)

[可视化模型](#)

[比较两个模型版本的性能](#)

[调整模型，提升性能](#)

[INT8 校准](#)

[Winograd 算法调整](#)

[精度测量](#)

[测量精度](#)

[配置精度设置](#)

[部署性能标准并将其集成到应用中](#)

[深度学习工作台中的高级主题](#)

[重启深度学习工作台](#)

[进入运行深度学习工作台的 Docker* 容器](#)

[配置传输层安全 \(TLS\)](#)

[故障排查](#)

[安全性](#)

[简介](#)

[安全地使用深度学习工作台](#)

[使用加密模型](#)

[优化指南](#)

[OpenVX* 开发人员指南](#)

[OpenCV* 开发人员指南](#)

[OpenCL™ 开发人员指南](#)

英特尔® 深度学习部署工具套件简介

本文档

[部署挑战](#)

[部署工作流程](#)

[模型优化器](#)

[模型优化器工作流程](#)

[支持的框架和格式](#)

[支持的模型](#)

[中间表示](#)

[nGraph 集成](#)

[推理引擎](#)

[另请参阅](#)

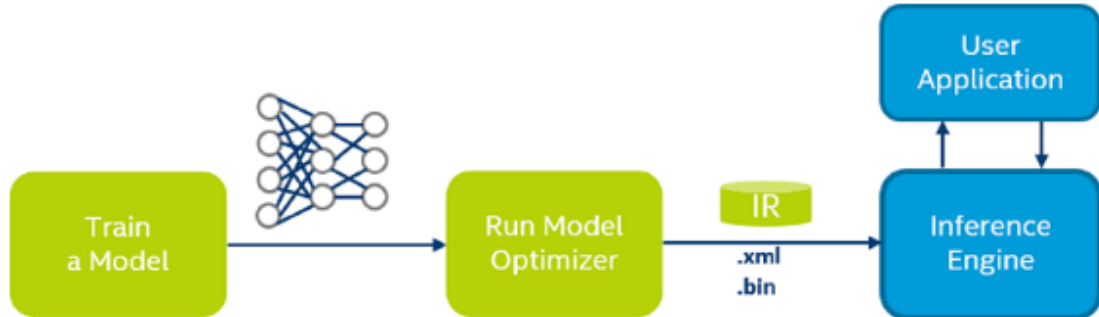
部署挑战

将深度学习网络从训练环境部署到嵌入式平台进行推理是一项复杂的任务，带来了许多技术挑战：

- 业界广泛使用了许多深度学习框架，例如 Caffe*、TensorFlow*、MXNet* 和 Kaldi* 等。
- 深度学习网络的训练通常在数据中心或服务器群内执行，推理可能在面向性能和功耗优化的嵌入式平台内执行。从软件角度（编程语言、第三方依赖关系、内存消耗、支持的操作系统）和硬件角度（不同的数据类型、有限的功耗）来看，此类平台一般都存在限制，因此通常不建议（有时甚至不可能）使用原始训练框架进行推理。一种替代解决方案是使用面向特定硬件平台优化的专用推理 API。
- 部署工作包括支持各种越来越复杂的层和网络，这进一步增加了难度。显然，确保转换网络的准确性并非易事。

部署工作流程

该流程假定您拥有使用其中一种[支持框架](#)进行训练的网络模型。以下方案展示了部署经过训练的深度学习模型的典型工作流程：



步骤包括：

1. 针对特定框架（用于训练模型）[配置模型优化器](#)。
2. 运行[模型优化器](#)，根据经过训练的网络拓扑、权重和偏差值以及其他可选参数，生成模型的[中间表示 \(IR\)](#)文件。
3. 使用目标环境中的[推理引擎](#)和提供的[推理引擎示例应用](#)，测试 IR 格式的模型。
4. [将推理引擎集成](#)到您的应用中，在目标环境中部署模型。

模型优化器

模型优化器是一种跨平台命令行工具，可促进训练和部署环境之间的过渡，执行静态模型分析并自动调整深度学习模型，以在端点目标设备上实现最佳执行。

模型优化器旨在支持多种深度学习[支持框架和格式](#)。

运行模型优化器时，您无需考虑要使用的目标设备，可以在所有目标中使用相同的 MO 输出。

模型优化器工作流程

该流程假定您拥有使用其中一种[支持框架](#)进行训练的网络模型。模型优化器的工作流程可描述如下：

- 为用于训练模型的其中一个支持的深度学习框架[配置模型优化器](#)。

- 以经过训练的网络作为输入，该网络包括特定的网络拓扑以及调整后的权重和偏差（带有一些可选参数）。
- [运行模型优化器](#)，以执行特定的模型优化（例如，某些网络层的水平融合）。优化是针对特定框架的，请参考相应的文档页面：[转换 Caffe 模型](#)，[转换 TensorFlow 模型](#)，[转换 MXNet 模型](#)，[转换 Kaldi 模型](#)，[转换 ONNX 模型](#)。
- 模型优化器生成网络的[中间表示 \(IR\)](#)文件作为输出，用作所有目标上的推理引擎的输入。

支持的框架和格式

- Caffe* (大多数公共分支)
- TensorFlow*
- MXNet*
- Kaldi*
- ONNX*

支持的模型

如欲获取支持模型的列表，请参见面向特定框架或格式的页面：

- [支持的 Caffe* 模型](#)
- [支持的 TensorFlow* 模型](#)
- [支持的 MXNet* 模型](#)
- [支持的 ONNX* 模型](#)
- [支持的 Kaldi* 模型](#)

中间表示

描述深度学习模型的中间表示在连接 OpenVINO™ 工具套件组件方面起着重要作用。IR 是一对文件：

- `.xml`：拓扑文件 - 一种描述网络拓扑的 XML 文件
- `.bin`：训练数据文件 - 一种包含权重和偏差二进制数据的 `.bin` 文件

可以使用[推理引擎](#)读取、加载和推理中间表示 (IR) 文件。推理引擎 API 在许多[支持的英特尔® 平台](#)上提供了一个统一的 API。训练后优化工具还可以使用、修改和写入 IR，该工具可提供量化功能。

更多详细信息请参见有关[中间表示和操作集](#)的专用描述。

nGraph 集成

OpenVINO 工具套件基于 nGraph 功能，用于图形构建 API、图形转换引擎和 Reshape，可替代以前的 NN Builder API 产品。nGraph 函数用作 CNNNetwork API 下运行时模型的中间表示。如果使用某些常规 API 方法时要求向后兼容，则 CNN 网络的常规表示形式仍然可用。请参见[nGraph Flow 概述](#)，详细了解 nGraph 与推理引擎的集成以及与传统表示形式的共存。

推理引擎

推理引擎是一种提供统一 API，将推理和应用逻辑集成起来的运行时。

- 以模型作为输入。该模型以模型优化器生成的特定形式的[中间表示 \(IR\)](#)来表示。
- 优化目标硬件的推理执行。
- 提供推理解决方案，节省嵌入式推理平台的空间。

推理引擎支持多种图像分类网络的推理，包括 AlexNet、GoogLeNet、VGG 和 ResNet 网络系列，用于图像分割的完全卷积网络（如 FCN8）以及对象检测网络（如 Faster R-CNN）。

如欲获取受支持硬件的完整列表，请参见[支持的设备](#)部分。

推理引擎包包含[头文件](#)、运行时库和[示例控制台应用](#)，这些示例演示了如何在应用中使用推理引擎。

另请参阅

- [推理引擎示例](#)
- [英特尔® 深度学习部署工具套件网页](#)

优化声明

如欲了解有关编译器优化的更多完整信息，请参见[优化声明](#)。

有关编译器优化的更多完整信息，请参阅我们的[优化声明](#)

支持

[英特尔® OpenVINO™ 工具套件分发版的英特尔® 开发人员专区论坛](#)

Cookie

[英特尔 Cookie 和类似技术声明](#)