1.手册简介

NVIDIA® TensorRT™ 是一个促进高性能机器学习推理的 SDK。 它旨在与 TensorFlow、PyTorch 和 MXNet 等训练框架以互补的方式工作。 它特别专注于在 NVIDIA 硬件上快速高效地运行已经训练好的网络。
有关如何安装 TensorRT 的说明，请参阅 NVIDIA TensorRT 安装指南。

NVIDIA TensorRT 快速入门指南适用于想要试用 TensorRT SDK 的用户； 具体来说，您将学习如何快速构建应用程序以在 TensorRT 引擎上运行推理。

1.1. Structure of this Guide

第 1 章提供了有关如何打包和支持 TensorRT 以及它如何融入开发者生态系统的信息。

第 2 章提供了对 TensorRT 功能的广泛概述。

第 3 章和第 4 章分别介绍了 C++ 和 Python API。

后续章节提供有关高级功能的更多详细信息。

附录包含网络层参考和常见问题解答。

1.2. Samples

NVIDIA TensorRT 示例支持指南说明了本手册中讨论的许多主题。可在此处找到其他侧重于嵌入式应用程序的示例。

1.3. Complementary GPU Features

多实例 GPU或 MIG 是具有 NVIDIA Ampere 架构或更高架构的 NVIDIA GPU 的一项功能，可实现用户控制的将单个 GPU 划分为多个较小 GPU 的功能。物理分区提供具有 QoS 的专用计算和内存切片，并在 GPU 的一部分上独立执行并行工作负载。对于 GPU 利用率低的 TensorRT 应用程序，MIG 可以在对延迟影响很小或没有影响的情况下产生更高的吞吐量。最佳分区方案是特定于应用程序的。

1.4. Complementary Software

NVIDIA Triton™推理服务器是一个更高级别的库，可提供跨 CPU 和 GPU 的优化推理。它提供了启动和管理多个模型的功能，以及用于服务推理的 REST 和 gRPC 端点。

NVIDIA DALI ®为预处理图像、音频和视频数据提供高性能原语。 TensorRT 推理可以作为自定义算子集成到 DALI 管道中。可以在此处找到作为 DALI 的一部分集成的 TensorRT 推理的工作示例。

TensorFlow-TensorRT (TF-TRT)是将 TensorRT 直接集成到 TensorFlow 中。它选择 TensorFlow 图的子图由 TensorRT 加速，同时让图的其余部分由 TensorFlow 本地执行。结果仍然是您可以照常执行的 TensorFlow 图。有关 TF-TRT 示例，请参阅TensorFlow 中的 TensorRT 示例。

PyTorch 量化工具包提供了以降低精度训练模型的工具，然后可以将其导出以在 TensorRT 中进行优化。

此外， PyTorch Automatic SParsity (ASP)工具提供了用于训练具有结构化稀疏性的模型的工具，然后可以将其导出并允许 TensorRT 在 NVIDIA Ampere GPU 上利用更快的稀疏策略。

TensorRT 与 NVIDIA 的分析工具、 NVIDIA Nsight™ Systems和NVIDIA® Deep Learning Profiler (DLProf)集成。

TensorRT 的一个受限子集经过认证可用于NVIDIA DRIVE ®产品。某些 API 被标记为仅在 NVIDIA DRIVE 中使用，不支持一般用途。

1.5. ONNX

TensorRT 从框架中导入训练模型的主要方式是通过ONNX交换格式。 TensorRT 附带一个 ONNX 解析器库来帮助导入模型。在可能的情况下，解析器向后兼容 opset 7； ONNX模型 Opset 版本转换器可以帮助解决不兼容问题。

GitHub 版本可能支持比 TensorRT 附带的版本更高的 opset，请参阅 ONNX-TensorRT运算符支持矩阵以获取有关受支持的 opset 和运算符的最新信息。

TensorRT 的 ONNX 算子支持列表可在此处找到。

PyTorch 原生支持ONNX 导出。对于 TensorFlow，推荐的方法是tf2onnx 。

将模型导出到 ONNX 后的第一步是使用Polygraphy运行常量折叠。这通常可以解决 ONNX 解析器中的 TensorRT 转换问题，并且通常可以简化工作流程。有关详细信息，请参阅此示例。在某些情况下，可能需要进一步修改 ONNX 模型，例如，用插件替换子图或根据其他操作重新实现不受支持的操作。为了简化此过程，您可以使用ONNX-GraphSurgeon 。

1.6. Code Analysis Tools

有关在 TensorRT 中使用 valgrind 和 clang sanitizer 工具的指导，请参阅故障排除章节。

1.7. API Versioning

TensorRT 版本号 (MAJOR.MINOR.PATCH) 遵循Semantic Versioning 2.0.0的公共 API 和库 ABI。版本号变化如下：

1. 进行不兼容的 API 或 ABI 更改时的主要版本
2. 以向后兼容的方式添加功能时的次要版本
3. 进行向后兼容的错误修复时的 PATCH 版本

请注意，语义版本控制不会扩展到序列化对象。要重用计划文件和时序缓存，版本号必须在主要、次要、补丁和构建版本之间匹配。校准缓存通常可以在主要版本中重复使用，但不保证兼容性。

1.8. Deprecation Policy

弃用用于通知开发人员不再推荐使用某些 API 和工具。从 8.0 版本开始，TensorRT 具有以下弃用政策：

- 弃用通知在发行说明中传达。已弃用的 API 元素在可能的情况下使用 `TRT_DEPRECATED` 宏进行标记。
- TensorRT 在弃用后提供 12 个月的迁移期。
- API 和工具在迁移期间继续工作。
- 迁移期结束后，API 和工具会以符合语义版本控制的方式移除。

对于在 TensorRT 7.x 中明确弃用的任何 API 和工具，12 个月的迁移期从 TensorRT 8.0 GA 发布日期开始。

1.9. Support

可以在 https://developer.nvidia.com/tensorrt在线找到有关 TensorRT 的支持、资源和信息。这包括博客、示例等。
此外，您可以在https://devtalk.nvidia.com/default/board/304/tensorrt/ 访问 NVIDIA DevTalk TensorRT 论坛，了解与 TensorRT 相关的所有内容。该论坛提供了寻找答案、建立联系以及参与与客

户、开发人员和 TensorRT 工程师讨论的可能性。

1.10. How Do I Report A Bug?

我们感谢所有类型的反馈。如果您遇到任何问题，请按照以下步骤进行报告。

1. 注册NVIDIA 开发者网站。
2. 登录开发者网站。
3. 点击右上角你的名字。
4. 单击我的帐户>我的错误并选择提交新错误。
5. 填写错误报告页面。具有描述性，如果可能，请提供您正在遵循的步骤以帮助重现问题。
6. 单击提交错误。