TABLE OF CONTENTS

Chapter 1. TensorRT Overview................................................................................................................. 1
Chapter 2. Pulling A Container.................................................................................................................. 2
Chapter 3. Running TensorRT................................................................................................................... 3
Chapter 4. TensorRT Release 19.07......................................................................................................... 6
Chapter 5. TensorRT Release 19.06......................................................................................................... 9
Chapter 6. TensorRT Release 19.05......................................................................................................... 12
Chapter 7. TensorRT Release 19.04....................................................................................................... 14
Chapter 8. TensorRT Release 19.03....................................................................................................... 16
Chapter 9. TensorRT Release 19.02....................................................................................................... 19
Chapter 10. TensorRT Release 19.01.................................................................................................... 21
Chapter 11. TensorRT Release 18.12.................................................................................................... 23
Chapter 12. TensorRT Release 18.11.................................................................................................... 25
Chapter 13. TensorRT Release 18.10.................................................................................................... 27
Chapter 14. TensorRT Release 18.09.................................................................................................... 29
Chapter 15. TensorRT Release 18.08.................................................................................................... 31
Chapter 16. TensorRT Release 18.07.................................................................................................... 33
Chapter 17. TensorRT Release 18.06.................................................................................................... 35
Chapter 18. TensorRT Release 18.05.................................................................................................... 37
Chapter 19. TensorRT Release 18.04.................................................................................................... 39
Chapter 20. TensorRT Release 18.03.................................................................................................... 41
Chapter 21. TensorRT Release 18.02.................................................................................................... 43
Chapter 22. TensorRT Release 18.01.................................................................................................... 45
Chapter 23. TensorRT Release 17.12.................................................................................................... 47
Chapter 1.
TENSORRT OVERVIEW

The core of NVIDIA TensorRT is a C++ library that facilitates high performance inference on NVIDIA graphics processing units (GPUs). TensorRT takes a trained network, which consists of a network definition and a set of trained parameters, and produces a highly optimized runtime engine which performs inference for that network.

You can describe a TensorRT network using either a C++ or Python API, or you can import an existing Caffe, ONNX, or TensorFlow model using one of the provided parsers.

TensorRT provides APIs via C++ and Python that help to express deep learning models via the Network Definition API or load a pre-defined model via the parsers that allows TensorRT to optimize and run them on an NVIDIA GPU. TensorRT applies graph optimizations, layer fusion, among other optimizations, while also finding the fastest implementation of that model leveraging a diverse collection of highly optimized kernels. TensorRT also supplies a runtime that you can use to execute this network on all of NVIDIA's GPU's from the Kepler generation onwards.

TensorRT also includes optional high speed mixed precision capabilities introduced in the Tegra X1, and extended with the Pascal, Volta, and Turing architectures.

The TensorRT container allows for the TensorRT samples to be built, modified and executed. For more information about the TensorRT samples, see the TensorRT Sample Support Guide.
Chapter 2.
PULLING A CONTAINER

Before you can pull a container from the NGC container registry, you must have Docker and nvidia-docker installed. For DGX users, this is explained in Preparing to use NVIDIA Containers Getting Started Guide.

For users other than DGX, follow the NVIDIA® GPU Cloud™ (NGC) container registry nvidia-docker installation documentation based on your platform.

You must also have access and be logged into the NGC container registry as explained in the NGC Getting Started Guide.

There are four repositories where you can find the NGC docker containers.

nvcr.io/nvidia
   The deep learning framework containers are stored in the nvcr.io/nvidia repository.

nvcr.io/hpc
   The HPC containers are stored in the nvcr.io/hpc repository.

nvcr.io/nvidia-hpccvis
   The HPC visualization containers are stored in the nvcr.io/nvidia-hpccvis repository.

nvcr.io/partner
   The partner containers are stored in the nvcr.io/partner repository. Currently the partner containers are focused on Deep Learning or Machine Learning, but that doesn’t mean they are limited to those types of containers.
Chapter 3.
RUNNING TENSORRT

Before running the container, use the `docker pull` command to ensure an up-to-date image is installed. Once the pull is complete, you can run the container image. This is because nvidia-docker ensures that drivers that match the host are used and configured for the container. Without nvidia-docker, you are likely to get an error when trying to run the container.

1. Issue the command for the applicable release of the container that you want. The following command assumes you want to pull the latest container.

   ```bash
docker pull nvcr.io/nvidia/tensorrt:19.07-py<x>
   ```

2. Open a command prompt and paste the pull command. The pulling of the container image begins. Ensure the pull completes successfully before proceeding to the next step.

3. Run the container image. A typical command to launch the container is:

   ```bash
nvidia-docker run -it --rm nvcr.io/nvidia/tensorrt:<xx.xx>-py<x>
   ```

   Where:
   - `-it` means interactive
   - `--rm` means delete the image when finished
   - `<xx.xx>` is the container version. For example, `19.01`.
   - `py<x>` is the Python version. For example, `py3`.

4. Once inside the container refer to `/workspace/README.md` for additional information about the container contents.

5. The `/workspace/tensorrt` directory contains copies of the TensorRT C++ and Python samples that can be modified, compiled, or executed. Use the following commands to compile the TensorRT C++ samples.

   ```bash
   cd /workspace/tensorrt/samples
   make -j4
   ```

6. Once the compilation is complete the resulting binaries will be located in `/workspace/tensorrt/bin`. The contents of this directory should look like the following.
7. From within the `/workspace/tensorrt/bin` directory run a few of the samples to verify that TensorRT is working properly. Run the following commands to exercise the Caffe, UFF, and ONNX parsers that are provided with TensorRT.

```sh
./sample_mnist
./sample_uff_mnist
./sample_onnx_mnist
```

These samples do not require any arguments to run and these particular samples recognize handwritten digits from the MNIST dataset. Many of the samples can be run immediately with the data provided within the container, but a few of the samples require additional preparation, which is documented in the `README.md` file in each sample directory.

In addition to the C++ samples, the TensorRT container includes some Python samples that come ready to run. You will find the Python samples in the `/workspace/tensorrt/samples/python` directory. The TensorFlow and PyTorch deep learning frameworks are not included in the TensorRT container so you will not be able to run samples which depend on these frameworks without some additional preparation. The container as shipped does not have the `uff` and `graphsurgeon` Python modules and the corresponding `convert-to-uff` utility installed, which all depend on TensorFlow. A script has been added to the container to install these missing Python modules and their dependencies as well as the `convert-to-uff` script if desired, which is shown below.

```sh
/opt/tensorrt/python/python_setup.sh
```

Refer to the TensorRT Sample Support Guide for more information about the samples included in the TensorRT container.

8. You can verify that the TensorRT Python interface is working by running the following samples from within the `/workspace/tensorrt/samples/python/introductory_parser_samples` directory.

```sh
python caffe_resnet50.py -d /workspace/tensorrt/python/data
python uff_resnet50.py -d /workspace/tensorrt/python/data
python onnx_resnet50.py -d /workspace/tensorrt/python/data
```
The above Python samples are analogous to the C++ samples mentioned earlier, but instead recognize images using ResNet50 rather than handwritten digits from the MNIST dataset.

9. If you would like to extend the TensorRT container you have two options:

1. Add to or modify the source code in this container and run your own customized version.
2. Use `docker build` to add your customizations on top of this container if you want to add additional packages.

NVIDIA recommends option b for ease of migration to later versions of the TensorRT container.
Chapter 4.
TENSORRT RELEASE 19.07

The NVIDIA container image for TensorRT, release 19.07, is available on NGC.

Contents of the TensorRT container

This container includes the following:

- The TensorRT C++ samples and C++ API documentation. The samples can be built by running `make` in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory. The C++ API documentation can be found in the `/workspace/tensorrt/doc/html` directory.

- The TensorRT Python samples and Python API documentation. The Python samples can be found in the `/workspace/tensorrt/samples/python` directory. Many Python samples can be run using `python <script.py> -d /workspace/tensorrt/data`. For example:
  ```
  python caffe_resnet50.py -d /workspace/tensorrt/data
  ```
  The Python API documentation can be found in the `/workspace/tensorrt/doc/python` directory.

- TensorRT 5.1.5

The container also includes the following:

- **Ubuntu 18.04**

Container image 19.07–py2 contains Python 2.7; 19.07–py3 contains Python 3.5.

- NVIDIA CUDA 10.1.168 including cuBLAS 10.2.0.168

- NVIDIA cuDNN 7.6.1
TensorRT Release 19.07

- **NVIDIA NCCL 2.4.7** (optimized for NVLink™)

  Although NCCL is packaged in the container, it does not effect TensorRT nor inferencing in any way.

- **MLNX_OFED +3.4**
- **OpenMPI 3.1.3**

**Driver Requirements**

Release 19.07 is based on [NVIDIA CUDA 10.1.168](https://developer.nvidia.com/cuda-downloads), which requires [NVIDIA Driver](https://www.nvidia.com/en-us/download/index.html) release 418.67. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384.111+ or 410. The CUDA driver’s compatibility package only supports particular drivers. For a complete list of supported drivers, see the [CUDA Application Compatibility](https://developer.nvidia.com/cuda-appcompatibility) topic. For more information, see [CUDA Compatibility and Upgrades](https://developer.nvidia.com/cuda-downloads).

**GPU Requirements**

Release 19.07 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see [CUDA GPUs](https://www.nvidia.com/en-us/cuda/gpus/). For additional support details, see [Deep Learning Frameworks Support Matrix](https://developer.nvidia.com/cuda-support).

**Key Features and Enhancements**

This TensorRT release includes the following key features and enhancements.

- TensorRT container image version 19.07 is based on [TensorRT 5.1.5](https://developer.nvidia.com/tensorrt).
- Latest version of [NVIDIA cuDNN 7.6.1](https://developer.nvidia.com/cudnn)
- Latest version of [MLNX_OFED +3.4](https://www.mellanox.com/products/open-f Fabrics)
- Latest version of [Ubuntu 18.04](https://www.ubuntu.com/)

**Obtaining Missing Data Files**

Some samples require data files that are not included within the TensorRT container either due to licensing restrictions or because they are too large. Samples which do not include all the required data files include a `README.md` file in the corresponding source directory informing you how to obtain the necessary data files.

**Installing Required Python Modules**

You may need to first run the Python setup script in order to complete some of the samples. The following script has been added to the container to install the missing Python modules and their dependencies if desired: `~/opt/tensorrt/python/python_setup.sh`
Known Issues

There are no known issues in this release.
Chapter 5.
TENSORRT RELEASE 19.06

The NVIDIA container image for TensorRT, release 19.06, is available on NGC.

Contents of the TensorRT container

This container includes the following:

▶ The TensorRT C++ samples and C++ API documentation. The samples can be built by running `make` in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory. The C++ API documentation can be found in the `/workspace/tensorrt/doc/html` directory.

▶ The TensorRT Python samples and Python API documentation. The Python samples can be found in the `/workspace/tensorrt/samples/python` directory. Many Python samples can be run using `python <script.py> -d /workspace/tensorrt/python/data`. The Python API documentation can be found in the `/workspace/tensorrt/doc/python` directory.

▶ TensorRT 5.1.5

The container also includes the following:

▶ Ubuntu 16.04

Container image 19.06-py2 contains Python 2.7; 19.06-py3 contains Python 3.5.

▶ NVIDIA CUDA 10.1.168 including cuBLAS 10.2.0.168
▶ NVIDIA cuDNN 7.6.0
▶ NVIDIA NCCL 2.4.7 (optimized for NVLink™)

Although NCCL is packaged in the container, it does not effect TensorRT nor inferencing in any way.

▶ OpenMPI 3.1.3
Driver Requirements

Release 19.06 is based on NVIDIA CUDA 10.1.168, which requires NVIDIA Driver release 418.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384.111+ or 410. The CUDA driver’s compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.06 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

‣ TensorRT container image version 19.06 is based on TensorRT 5.1.5.
‣ Latest version of NVIDIA CUDA 10.1.168 including cuBLAS 10.2.0.168
‣ Latest version of NVIDIA NCCL 2.4.7
‣ Ubuntu 16.04 with May 2019 updates (see Announcements)

Obtaining Missing Data Files

Some samples require data files that are not included within the TensorRT container either due to licensing restrictions or because they are too large. Samples which do not include all the required data files include a README.md file in the corresponding source directory informing you how to obtain the necessary data files.

Installing Required Python Modules

You may need to first run the Python setup script in order to complete some of the samples. The following script has been added to the container to install the missing Python modules and their dependencies if desired: /opt/tensorrt/python/python_setup.sh

Announcements

In the next release, we will no longer support Ubuntu 16.04. Release 19.07 will instead support Ubuntu 18.04.

Known Issues

There are no known issues in this release.
Chapter 6.
TENSORRT RELEASE 19.05

The NVIDIA container image for TensorRT, release 19.05, is available on NGC.

Contents of the TensorRT container

This container includes the following:

- The TensorRT C++ samples and C++ API documentation. The samples can be built by running `make` in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory. The C++ API documentation can be found in the `/workspace/tensorrt/doc/html` directory.
- The TensorRT Python samples and Python API documentation. The Python samples can be found in the `/workspace/tensorrt/samples/python` directory. Many Python samples can be run using `python <script.py> -d /workspace/tensorrt/python/data`. The Python API documentation can be found in the `/workspace/tensorrt/doc/python` directory.
- **TensorRT 5.1.5**

The container also includes the following:

- **Ubuntu 16.04**

  Container image 19.05–py2 contains Python 2.7; 19.05–py3 contains Python 3.5.

- **NVIDIA CUDA 10.1 Update 1** including cuBLAS 10.1 Update 1
- **NVIDIA cuDNN 7.6.0**
- **NVIDIA NCCL 2.4.6** (optimized for NVLink™)

Although NCCL is packaged in the container, it does not effect TensorRT nor inferencing in any way.

- **OpenMPI 3.1.3**
Driver Requirements

Release 19.05 is based on CUDA 10.1 Update 1, which requires NVIDIA Driver release 418.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384.111+ or 410. The CUDA driver’s compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.05 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

- TensorRT container image version 19.05 is based on TensorRT 5.1.5.
- Latest version of NVIDIA CUDA 10.1 Update 1 including cuBLAS 10.1 Update 1
- Latest version of NVIDIA cuDNN 7.6.0
- Latest version of TensorRT 5.1.5
- Ubuntu 16.04 with April 2019 updates

Obtaining Missing Data Files

Some samples require data files that are not included within the TensorRT container either due to licensing restrictions or because they are too large. Samples which do not include all the required data files include a README.md file in the corresponding source directory informing you how to obtain the necessary data files.

Installing Required Python Modules

You may need to first run the Python setup script in order to complete some of the samples. The following script has been added to the container to install the missing Python modules and their dependencies if desired: /opt/tensorrt/python/python_setup.sh

Known Issues

There are no known issues in this release.
The NVIDIA container image for TensorRT, release 19.04, is available on NGC.

Contents of the TensorRT container

This container includes the following:

- The TensorRT C++ samples and C++ API documentation. The samples can be built by running `make` in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory. The C++ API documentation can be found in the `/workspace/tensorrt/doc/html` directory.

- The TensorRT Python samples and Python API documentation. The Python samples can be found in the `/workspace/tensorrt/samples/python` directory. Many Python samples can be run using `python <script.py> -d /workspace/tensorrt/python/data`. The Python API documentation can be found in the `/workspace/tensorrt/doc/python` directory.

- TensorRT 5.1.2

The container also includes the following:

- Ubuntu 16.04

  Container image `19.04-py2` contains Python 2.7; `19.04-py3` contains Python 3.5.

- NVIDIA CUDA 10.1.105 including cuBLAS 10.1.0.105
- NVIDIA cuDNN 7.5.0
- NVIDIA NCCL 2.4.6 (optimized for NVLink™)

  Although NCCL is packaged in the container, it does not effect TensorRT nor inferencing in any way.

- OpenMPI 3.1.3
Driver Requirements

Release 19.04 is based on CUDA 10.1, which requires NVIDIA Driver release 418.xx.x. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384.111+ or 410. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.04 supports CUDA compute capability 3.0 and higher. This corresponds to GPUs in the Kepler, Maxwell, Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

‣ TensorRT container image version 19.04 is based on TensorRT 5.1.2 RC.
‣ Latest version of NVIDIA NCCL 2.4.6
‣ Latest version of cuBLAS 10.1.0.105
‣ Ubuntu 16.04 with March 2019 updates

Obtaining Missing Data Files

Some samples require data files that are not included within the TensorRT container either due to licensing restrictions or because they are too large. Samples which do not include all the required data files include a README.md file in the corresponding source directory informing you how to obtain the necessary data files.

Installing Required Python Modules

You may need to first run the Python setup script in order to complete some of the samples. The following script has been added to the container to install the missing Python modules and their dependencies if desired: /opt/tensorrt/python/python_setup.sh

Known Issues

There are no known issues in this release.
Chapter 8.
TENSORRT RELEASE 19.03

The NVIDIA container image for TensorRT, release 19.03, is available on NGC.

Contents of the TensorRT container

This container includes the following:

- The TensorRT C++ samples and C++ API documentation. The samples can be built by running `make` in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory. The C++ API documentation can be found in the `/workspace/tensorrt/doc/html` directory.

- The TensorRT Python samples and Python API documentation. The Python samples can be found in the `/workspace/tensorrt/samples/python` directory. Many Python samples can be run using `python <script.py> -d /workspace/tensorrt/python/data`. The Python API documentation can be found in the `/workspace/tensorrt/doc/python` directory.

- **TensorRT 5.1.2**

The container also includes the following:

- **Ubuntu 16.04**

  - **Container image 19.03-py2** contains Python 2.7; **19.03-py3** contains Python 3.5.

  - **NVIDIA CUDA 10.1.105** including **cuBLAS 10.1.105**

  - **NVIDIA cuDNN 7.5.0**

  - **NVIDIA NCCL 2.4.3** (optimized for NVLink™)

  Although NCCL is packaged in the container, it does not effect TensorRT nor inferencing in any way.

- **OpenMPI 3.1.3**
Driver Requirements

Release 19.03 is based on CUDA 10.1, which requires NVIDIA Driver release 418.xx+. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384.111+ or 410. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.03 supports CUDA compute capability 3.0 and higher. This corresponds to GPUs in the Kepler, Maxwell, Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

- TensorRT container image version 19.03 is based on TensorRT 5.1.2 RC.
- Latest version of NVIDIA CUDA 10.1.105 including cuBLAS 10.1.105
- Latest version of NVIDIA cuDNN 7.5.0
- Latest version of NVIDIA NCCL 2.4.3
- Ubuntu 16.04 with February 2019 updates

Obtaining Missing Data Files

Some samples require data files that are not included within the TensorRT container either due to licensing restrictions or because they are too large. Samples which do not include all the required data files include a README.md file in the corresponding source directory informing you how to obtain the necessary data files.

Installing Required Python Modules

You may need to first run the Python setup script in order to complete some of the samples. The following script has been added to the container to install the missing Python modules and their dependencies if desired: /opt/tensorrt/python/python_setup.sh

Known Issues

- If using or upgrading to a 3-part-version driver, for example, a driver that takes the format of xxx.yy.zz, you will receive a Failed to detect NVIDIA driver version message. This is due to a known bug in the entry point script's parsing of 3-part driver versions. This message is non-fatal and can be ignored. This will be fixed in the 19.04 release.
Chapter 9.
TENSORRT RELEASE 19.02

The NVIDIA container image for TensorRT, release 19.02, is available.

Contents of TensorRT

This container includes the following:

- The TensorRT C++ samples and C++ API documentation. The samples can be built by running `make` in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory. The C++ API documentation can be found in the `/workspace/tensorrt/doc/html` directory.
- The TensorRT Python samples and Python API documentation. The Python samples can be found in the `/workspace/tensorrt/samples/python` directory. Many Python samples can be run using `python <script.py> -d /workspace/tensorrt/python/data`. The Python API documentation can be found in the `/workspace/tensorrt/doc/python` directory.

The container also includes the following:

- **Ubuntu 16.04**

---

**Container image 19.02-py2 contains Python 2.7; 19.02-py3 contains Python 3.5.**

- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.4.2
- NVIDIA Collective Communications Library (NCCL) 2.3.7 (optimized for NVLink™)

---

Although NCCL is packaged in the container, it does not effect TensorRT nor inferencing in any way.
Driver Requirements

Release 19.02 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.02 supports CUDA compute capability 3.0 and higher. This corresponds to GPUs in the Kepler, Maxwell, Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

- TensorRT container image version 19.02 is based on TensorRT 5.0.2.
- Ubuntu 16.04 with January 2019 updates

Obtaining Missing Data Files

Some samples require data files that are not included within the TensorRT container either due to licensing restrictions or because they are too large. Samples which do not include all the required data files include a README.txt or README.md file in the corresponding source directory informing you how to obtain the necessary data files.

Installing Required Python Modules

You may need to first run the Python setup script in order to complete some of the samples. The following script has been added to the container to install the missing Python modules and their dependencies if desired: /opt/tensorrt/python/python_setup.sh

Known Issues

- If using or upgrading to a 3-part-version driver, for example, a driver that takes the format of xxx.yy.zz, you will receive a Failed to detect NVIDIA driver version. message. This is due to a known bug in the entry point script’s parsing of 3-part driver versions. This message is non-fatal and can be ignored. This will be fixed in the 19.04 release.
Chapter 10.
TENSORRT RELEASE 19.01

The NVIDIA container image for TensorRT, release 19.01, is available.

Contents of TensorRT

This container includes the following:

- The TensorRT C++ samples and C++ API documentation. The samples can be built by running `make` in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory. The C++ API documentation can be found in the `/workspace/tensorrt/doc/html` directory.
- The TensorRT Python samples and Python API documentation. The Python samples can be found in the `/workspace/tensorrt/samples/python` directory. Many Python samples can be run using `python <script.py> -d /workspace/tensorrt/python/data`. The Python API documentation can be found in the `/workspace/tensorrt/doc/python` directory.

The container also includes the following:

- **Ubuntu 16.04**

  Container image 19.01-py2 contains Python 2.7; 19.01-py3 contains Python 3.5.

  - NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
  - NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.4.2
  - NCCL 2.3.7 (optimized for NVLink™)

Although NCCL is packaged in the container, it does not effect TensorRT nor inferencing in any way.

- **OpenMPI 3.1.3**
Driver Requirements

Release 19.01 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.01 supports CUDA compute capability 3.0 and higher. This corresponds to GPUs in the Kepler, Maxwell, Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

‣ TensorRT container image version 19.01 is based on TensorRT 5.0.2.
‣ Latest version of OpenMPI 3.1.3
‣ Ubuntu 16.04 with December 2018 updates

Obtaining Missing Data Files

Some samples require data files that are not included within the TensorRT container either due to licensing restrictions or because they are too large. Samples which do not include all the required data files include a README.txt or README.md file in the corresponding source directory informing you how to obtain the necessary data files.

Installing Required Python Modules

You may need to first run the Python setup script in order to complete some of the samples. The following script has been added to the container to install the missing Python modules and their dependencies if desired: /opt/tensorrt/python/python_setup.sh

Known Issues

‣ If using or upgrading to a 3-part-version driver, for example, a driver that takes the format of xxx.yy.zz, you will receive a Failed to detect NVIDIA driver version. message. This is due to a known bug in the entry point script's parsing of 3-part driver versions. This message is non-fatal and can be ignored. This will be fixed in the 19.04 release.
Chapter 11.
TENSORRT RELEASE 18.12

The NVIDIA container image for TensorRT, release 18.12, is available.

Contents of TensorRT

This container includes the following:

- The TensorRT C++ samples and C++ API documentation. The samples can be built by running `make` in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory. The C++ API documentation can be found in the `/workspace/tensorrt/doc/html` directory.
- The TensorRT Python samples and Python API documentation. The Python samples can be found in the `/workspace/tensorrt/samples/python` directory. Many Python samples can be run using `python <script.py> -d /workspace/tensorrt/python/data`. The Python API documentation can be found in the `/workspace/tensorrt/doc/python` directory.

The container also includes the following:

- Ubuntu 16.04

Container image 18.12-py2 contains Python 2.7; 18.12-py3 contains Python 3.5.

- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.4.1
- NCCL 2.3.7 (optimized for NVLink™)

Although NCCL is packaged in the container, it does not effect TensorRT nor inferencing in any way.

- OpenMPI 3.1.2
Driver Requirements

Release 18.12 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 18.12 supports CUDA compute capability 3.0 and higher. This corresponds to GPUs in the Kepler, Maxwell, Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

- TensorRT container image version 18.12 is based on TensorRT 5.0.2.
- Ubuntu 16.04 with November 2018 updates

Obtaining Missing Data Files

Some samples require data files that are not included within the TensorRT container either due to licensing restrictions or because they are too large. Samples which do not include all the required data files include a README.txt or README.md file in the corresponding source directory informing you how to obtain the necessary data files.

Installing Required Python Modules

You may need to first run the Python setup script in order to complete some of the samples. The following script has been added to the container to install the missing Python modules and their dependencies if desired: /opt/tensorrt/python/python_setup.sh

Known Issues

There are no known issues in this release.
Chapter 12.
TENSORRT RELEASE 18.11

The NVIDIA container image for TensorRT, release 18.11, is available.

Contents of TensorRT

This container includes the following:

- The TensorRT C++ samples and C++ API documentation. The samples can be built by running `make` in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory. The C++ API documentation can be found in the `/workspace/tensorrt/doc/html` directory.
- The TensorRT Python samples and Python API documentation. The Python samples can be found in the `/workspace/tensorrt/samples/python` directory. Many Python samples can be run using `python <script.py> -d /workspace/tensorrt/python/data`. The Python API documentation can be found in the `/workspace/tensorrt/doc/python` directory.

The container also includes the following:

- **Ubuntu** 16.04

  Container image 18.11-py2 contains Python 2.7; 18.11-py3 contains Python 3.5.

- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.4.1
- NCCL 2.3.7 (optimized for NVLink™)
- OpenMPI 3.1.2
Driver Requirements

Release 18.11 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

- TensorRT container image version 18.11 is based on TensorRT 5.0.2.
- Latest version of NCCL 2.3.7.
- Latest version of NVIDIA cuDNN 7.4.1.
- Ubuntu 16.04 with October 2018 updates

Obtaining Missing Data Files

Some samples require data files that are not included within the TensorRT container either due to licensing restrictions or because they are too large. Samples which do not include all the required data files include a README.txt or README.md file in the corresponding source directory informing you how to obtain the necessary data files.

Installing Required Python Modules

You may need to first run the Python setup script in order to complete some of the samples. The following script has been added to the container to install the missing Python modules and their dependencies if desired: /opt/tensorrt/python/python_setup.sh

Known Issues

There are no known issues in this release.
Chapter 13.
TENSORRT RELEASE 18.10

The NVIDIA container image of TensorRT, release 18.10, is available.

Contents of TensorRT

This container includes the following:

- The TensorRT C++ samples and C++ API documentation. The samples can be built by running `make` in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory. The C++ API documentation can be found in the `/workspace/tensorrt/doc/html` directory.
- The TensorRT Python samples and Python API documentation. The Python samples can be found in the `/workspace/tensorrt/samples/python` directory. Many Python samples can be run using `python <script.py> -d /workspace/tensorrt/python/data`. The Python API documentation can be found in the `/workspace/tensorrt/doc/python` directory.

The container also includes the following:

- Ubuntu 16.04

Container image 18.10-py2 contains Python 2.7; 18.10-py3 contains Python 3.5.

- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.4.0
- NCCL 2.3.6 (optimized for NVLink™)
- OpenMPI 3.1.2
Driver Requirements

Release 18.10 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

‣ TensorRT container image version 18.10 is based on TensorRT 5.0.0 RC.
‣ Latest version of NCCL 2.3.6.
‣ Added support for OpenMPI 3.1.2.
‣ Ubuntu 16.04 with September 2018 updates

Obtaining Missing Data Files

Some samples require data files that are not included within the TensorRT container either due to licensing restrictions or because they are too large. Samples which do not include all the required data files include a README.txt or README.md file in the corresponding source directory informing you how to obtain the necessary data files.

Installing Required Python Modules

You may need to first run the Python setup script in order to complete some of the samples. The following script has been added to the container to install missing Python modules and their dependencies if desired: /opt/tensorrt/python/python_setup.sh

Known Issues

There are no known issues in this release.
Chapter 14.
TENSORRT RELEASE 18.09

The NVIDIA container image of TensorRT, release 18.09, is available.

Contents of TensorRT

This container includes the following:

- The TensorRT C++ samples and C++ API documentation. The samples can be built by running `make` in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory. The C++ API documentation can be found in the `/workspace/tensorrt/doc/html` directory.
- The TensorRT Python samples and Python API documentation. The Python samples can be found in the `/workspace/tensorrt/samples/python` directory. Many Python samples can be run using `python <script.py> -d /workspace/tensorrt/python/data`. The Python API documentation can be found in the `/workspace/tensorrt/doc/python` directory.

The container also includes the following:

- **Ubuntu 16.04**

  Container image 18.09-py2 contains Python 2.7; 18.09-py3 contains Python 3.5.

- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.3.0
- NCCL 2.3.4 (optimized for NVLink™)

Driver Requirements

Release 18.09 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you
may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

**Key Features and Enhancements**

This TensorRT release includes the following key features and enhancements.

- TensorRT container image version 18.09 is based on TensorRT 5.0.0 RC.
- Latest version of cuDNN 7.3.0.
- Latest version of CUDA 10.0.130 which includes support for DGX-2, Turing, and Jetson Xavier.
- Latest version of cuBLAS 10.0.130.
- Latest version of NCCL 2.3.4.
- Ubuntu 16.04 with August 2018 updates

**Installing Required Python Modules**

Some samples require data files that are not included within the TensorRT container either due to licensing restrictions or because they are too large. The following script has been added to the container to install these missing Python modules and their dependencies if desired: `/opt/tensorrt/python/python_setup.sh`

Samples which do not include all the required data files include a `README.txt` file in the corresponding source directory informing you how to obtain the necessary data files. You may need to first run the Python setup script in order to complete some of the samples.

**Known Issues**

The TensorRT Release Notes ([TensorRT-Release-Notes.pdf](TensorRT-Release-Notes.pdf)) is missing from the container. Refer to the online TensorRT Release Notes instead.
Chapter 15.
TENSORRT RELEASE 18.08

The NVIDIA container image of TensorRT, release 18.08, is available.

Contents of TensorRT

This container includes the following:

- The TensorRT documentation and C++ samples. The samples can be built by running `make` in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory.
- The TensorRT Python examples. The Python examples can be found in the `/workspace/tensorrt/python/examples` directory. Most Python examples can be run using `python <script.py> /workspace/tensorrt/python/data`. The Python API documentation can be found in the `/usr/lib/python<x.y>/dist-packages/docs` directory.

The container also includes the following:

- Ubuntu 16.04

  Container image 18.08–py2 contains Python 2.7; 18.08–py3 contains Python 3.5.

- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.425
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.2.1
- NCCL 2.2.13 (optimized for NVLink™)

Driver Requirements

Release 18.08 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.
Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

- TensorRT container image version 18.08 is based on TensorRT 4.0.1.
- Latest version of cuDNN 7.2.1.
- A new script has been added to the container that will install uff, graphsurgeon, as well as other Python modules that are required to execute all of the Python examples.
- Ubuntu 16.04 with July 2018 updates

Installing Required Python Modules

Some samples require data files that are not included within the TensorRT container either due to licensing restrictions or because they are too large. The following script has been added to the container to install these missing Python modules and their dependencies if desired: /opt/tensorrt/python/python_setup.sh

Samples which do not include all the required data files include a README.txt file in the corresponding source directory informing you how to obtain the necessary data files. You may need to first run the Python setup script in order to complete some of the samples.

Known Issues

There are no known issues in this release.
Chapter 16.
TENSORRT RELEASE 18.07

The NVIDIA container image of TensorRT, release 18.07, is available.

Contents of TensorRT

This container includes the following:

- The TensorRT documentation and C++ samples. The samples can be built by running `make` in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory.
- The TensorRT Python examples. The Python examples can be found in the `/workspace/tensorrt/python/examples` directory. Most Python examples can be run using `python <script.py> /workspace/tensorrt/python/data`. The Python API documentation can be found in the `/usr/lib/python2.7/dist-packages/docs` directory.

The container also includes the following:

- Ubuntu 16.04

  Container image 18.07-py2 contains Python 2.7; 18.07-py3 contains Python 3.5.

- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.425
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.4
- NCCL 2.2.13 (optimized for NVLink™)

Driver Requirements

Release 18.07 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.
Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

- TensorRT container image version 18.07 is based on TensorRT 4.0.1.
- Latest version of CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.425.
- Ubuntu 16.04 with June 2018 updates

Known Issues

Some samples require data files that are not included within the TensorRT container either due to licensing concerns or because they are too large. Samples which do not include all the required data files instead include a README.txt file in the corresponding source directory informing you how to obtain the necessary data files. The data files required for the samples sampleNMT and sampleUffSSD cannot be easily created within the TensorRT container using the default packages. You should instead prepare the data files for these samples outside the container and then use docker cp to copy the necessary files into the TensorRT container or use a mount point when running the TensorRT container.
Chapter 17.
TENSORRT RELEASE 18.06

The NVIDIA container image of TensorRT, release 18.06, is available.

Contents of TensorRT

This container includes the following:

- The TensorRT documentation and C++ samples. The samples can be built by running `make` in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory.

- The TensorRT Python examples. The Python examples can be found in the `/workspace/tensorrt/python/examples` directory. Most Python examples can be run using `python <script.py> /workspace/tensorrt/python/data`. The Python API documentation can be found in the `/usr/lib/python2.7/dist-packages/docs` directory.

The container also includes the following:

- Ubuntu 16.04

Container image 18.06–py2 contains Python 2.7; 18.06–py3 contains Python 3.5.

- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.333 (see section 2.3.1)

- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.4

- NCCL 2.2.13 (optimized for NVLink™)

Driver Requirements

Release 18.06 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.
Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

‣ TensorRT container image version 18.06 is based on TensorRT 4.0.1.
‣ Ubuntu 16.04 with May 2018 updates

Known Issues

Some samples require data files that are not included within the TensorRT container either due to licensing concerns or because they are too large. Samples which do not include all the required data files instead include a README.txt file in the corresponding source directory informing you how to obtain the necessary data files. The data files required for the samples sampleNMT and sampleUffSSD cannot be easily created within the TensorRT container using the default packages. You should instead prepare the data files for these samples outside the container and then use docker cp to copy the necessary files into the TensorRT container or use a mount point when running the TensorRT container.
Chapter 18.
TENSORRT RELEASE 18.05

The NVIDIA container image of TensorRT, release 18.05, is available.

Contents of TensorRT

This container image contains an example deployment strategy using TensorRT inference exposed via a REST server. Three trained models, NVcaffe, ONNX and TensorFlow, are included to demonstrate the inference REST server. You can also perform inference using your own NVcaffe, ONNX and TensorFlow models via the REST server.

This container also include the following:

▶ The TensorRT documentation and samples. The samples can be built by running make in the /workspace/tensorrt/samples directory. The resulting executables are in the /workspace/tensorrt/bin directory.
▶ The example NVcaffe MNIST model and the caffe_mnist script are located in the /workspace/tensorrt_server directory. The script runs the REST server to provide inference for that model via an HTTP endpoint.
▶ The example Inception-v1 ONNX model and the onnx_inception_v1 script are also located in the /workspace/tensorrt_server directory. This example and script runs the REST server to provide inference for that model via an HTTP endpoint.
▶ The example ResNet-152 TensorFlow model and the tensorflow_resnet script are also located in the /workspace/tensorrt_server directory. This example and script runs the REST server to provide inference for that model via an HTTP endpoint.

The container also includes the following:

▶ Ubuntu 16.04 including Python 2.7 environment
▶ NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.333 (see section 2.3.1)
TensorRT Release 18.05

- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.2
- NCCL 2.1.15 (optimized for NVLink™)

Driver Requirements

Release 18.05 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

- TensorRT container image version 18.05 is based on TensorRT 3.0.4.
- Fixed an issue with INT8 deconvolution bias. If you have seen an issue with deconvolution INT8 accuracy especially regarding TensorRT 2.1, then this fix should solve the issue.
- Fixed an accuracy issue in FP16 mode for NVCaffe models.
- Ubuntu 16.04 with April 2018 updates

Known Issues

There are no known issues in this release.
Chapter 19.
TENSORRT RELEASE 18.04

The NVIDIA container image of TensorRT, release 18.04, is available.

Contents of TensorRT

This container image contains an example deployment strategy using TensorRT inference exposed via a REST server. Three trained models, NVCAffe, ONNX and TensorFlow, are included to demonstrate the inference REST server. You can also perform inference using your own NVCAffe, ONNX and TensorFlow models via the REST server.

This container also include the following:

- The TensorRT documentation and samples. The samples can be built by running make in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory.
- The example NVCAffe MNIST model and the `caffe_mnist` script are located in the `/workspace/tensorrt_server` directory. The script runs the REST server to provide inference for that model via an HTTP endpoint.
- The example Inception-v1 ONNX model and the `onnx_inception_v1` script are also located in the `/workspace/tensorrt_server` directory. This example and script runs the REST server to provide inference for that model via an HTTP endpoint.
- The example ResNet-152 TensorFlow model and the `tensorflow_resnet` script are also located in the `/workspace/tensorrt_server` directory. This example and script runs the REST server to provide inference for that model via an HTTP endpoint.

The container also includes the following:

- Ubuntu 16.04 including Python 2.7 environment
- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.333 (see section 2.3.1)
TensorRT Release 18.04

- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.1
- NCCL 2.1.15 (optimized for NVLink™)

Driver Requirements

Release 18.04 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

- TensorRT container image version 18.04 is based on TensorRT 3.0.4.
- Fixed an issue with INT8 deconvolution bias. If you have seen an issue with deconvolution INT8 accuracy especially regarding TensorRT. 2.1, then this fix should solve the issue.
- Fixed an accuracy issue in FP16 mode for NVCaffe models.
- Latest version of NCCL 2.1.15
- Ubuntu 16.04 with March 2018 updates

Known Issues

There are no known issues in this release.
Chapter 20.  
TENSORRT RELEASE 18.03

The NVIDIA container image of TensorRT, release 18.03, is available.

Contents of TensorRT

This container image contains an example deployment strategy using TensorRT inference exposed via a REST server. Three trained models, NVCAffe, ONNX and TensorFlow, are included to demonstrate the inference REST server. You can also perform inference using your own NVCAffe, ONNX and TensorFlow models via the REST server.

This container also include the following:

- The TensorRT documentation and samples. The samples can be built by running make in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory.
- The example NVCAffe MNIST model and the `caffe_mnist` script are located in the `/workspace/tensorrt_server` directory. The script runs the REST server to provide inference for that model via an HTTP endpoint.
- The example Inception-v1 ONNX model and the `onnx_inception_v1` script are also located in the `/workspace/tensorrt_server` directory. This example and script runs the REST server to provide inference for that model via an HTTP endpoint.
- The example ResNet-152 TensorFlow model and the `tensorflow_resnet` script are also located in the `/workspace/tensorrt_server` directory. This example and script runs the REST server to provide inference for that model via an HTTP endpoint.

The container also includes the following:

- Ubuntu 16.04 including Python 2.7 environment
- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.333 (see section 2.3.1)
TensorRT Release 18.03

- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.1
- NCCL 2.1.2 (optimized for NVLink™)

Driver Requirements

Release 18.03 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

- TensorRT container image version 18.03 is based on TensorRT 3.0.4.
- Fixed an issue with INT8 deconvolution bias. If you have seen an issue with deconvolution INT8 accuracy especially regarding TensorRT 2.1, then this fix should solve the issue.
- Fixed an accuracy issue in FP16 mode for NVcaffe models.
- Latest version of cuBLAS 9.0.333
- Latest version of cuDNN 7.1.1
- Ubuntu 16.04 with February 2018 updates

Known Issues

There are no known issues in this release.
Chapter 21.
TENSORRT RELEASE 18.02

The NVIDIA container image of TensorRT, release 18.02, is available.
TensorRT container image version 18.02 is based on TensorRT 3.0.4.

Contents of TensorRT

This container image contains an example deployment strategy using TensorRT inference exposed via a REST server. Three trained models, NVCaffe, ONNX and TensorFlow, are included to demonstrate the inference REST server. You can also perform inference using your own NVCaffe, ONNX and TensorFlow models via the REST server.

This container also include the following:

- The TensorRT documentation and samples. The samples can be built by running make in the /workspace/tensorrt/samples directory. The resulting executables are in the /workspace/tensorrt/bin directory.
- The example NVCaffe MNIST model and the caffe_mnist script are located in the /workspace/tensorrt_server directory. The script runs the REST server to provide inference for that model via an HTTP endpoint.
- The example Inception-v1 ONNX model and the onnx_inception_v1 script are also located in the /workspace/tensorrt_server directory. This example and script runs the REST server to provide inference for that model via an HTTP endpoint.
- The example ResNet-152 TensorFlow model and the tensorflow_resnet script are also located in the /workspace/tensorrt_server directory. This example and script runs the REST server to provide inference for that model via an HTTP endpoint.

The container also includes the following:

- Ubuntu 16.04 including Python 2.7 environment
- NVIDIA CUDA 9.0.176 including:
CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.282 Patch 2 which is installed by default

- cuBLAS 9.0.234 Patch 1 as a debian file. Installing Patch 1 by issuing the `dpkg -i /opt/cuda-cublas-9-0_9.0.234-1_amd64.deb` command is the workaround for the known issue described below.

- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.0.5
- NCCL 2.1.2 (optimized for NVLink™)

Driver Requirements

Release 18.02 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

- Latest version of cuBLAS
- Ubuntu 16.04 with January 2018 updates

Known Issues

cuBLAS 9.0.282 regresses RNN seq2seq FP16 performance for a small subset of input sizes. This issue should be fixed in the next update. As a workaround, install cuBLAS 9.0.234 Patch 1 by issuing the `dpkg -i /opt/cuda-cublas-9-0_9.0.234-1_amd64.deb` command.
Chapter 22.
TENSORRT RELEASE 18.01

The NVIDIA container image of TensorRT, release 18.01, is available.

TensorRT container image version 18.01 is based on TensorRT 3.0.1.

Contents of TensorRT
This container image contains an example deployment strategy using TensorRT inference exposed via a REST server. Three trained models, NVCaffe, ONNX and TensorFlow, are included to demonstrate the inference REST server. You can also perform inference using your own NVCaffe, ONNX and TensorFlow models via the REST server.

This container also includes the following:

- The TensorRT documentation and samples. The samples can be built by running make in the /workspace/tensorrt/samples directory. The resulting executables are in the /workspace/tensorrt/bin directory.
- The example NVCaffe MNIST model and the caffe_mnist script are located in the /workspace/tensorrt_server directory. The script runs the REST server to provide inference for that model via an HTTP endpoint.
- The example Inception-v1 ONNX model and the onnx_inception_v1 script are also located in the /workspace/tensorrt_server directory. This example and script runs the REST server to provide inference for that model via an HTTP endpoint.
- The example ResNet-152 TensorFlow model and the tensorflow_resnet script are also located in the /workspace/tensorrt_server directory. This example and script runs the REST server to provide inference for that model via an HTTP endpoint.

The container also includes the following:
- Ubuntu 16.04 including Python 2.7 environment
TensorRT Release 18.01

- NVIDIA CUDA 9.0.176 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.282
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.0.5
- NCCL 2.1.2 (optimized for NVLink™)

Driver Requirements

Release 18.01 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This TensorRT release includes the following key features and enhancements.

- Latest version of cuBLAS
- Latest version of cuDNN
- Latest version of NCCL
- Ubuntu 16.04 with December 2017 updates

Known Issues

cuBLAS 9.0.282 regresses RNN seq2seq FP16 performance for a small subset of input sizes. As a workaround, revert back to the 11.12 container.
Chapter 23.
TENSORRT RELEASE 17.12

The NVIDIA container image of TensorRT, release 17.12, is available.

Contents of TensorRT

This container image contains an example deployment strategy using TensorRT inference exposed via a REST server. Three trained models, NVcaffe, ONNX and TensorFlow, are included to demonstrate the inference REST server. You can also perform inference using your own NVcaffe, ONNX and TensorFlow models via the REST server.

This container also include the following:

- The TensorRT documentation and samples. The samples can be built by running make in the `/workspace/tensorrt/samples` directory. The resulting executables are in the `/workspace/tensorrt/bin` directory.
- The example NVcaffe MNIST model and the `caffe_mnist` script are located in the `/workspace/tensorrt_server` directory. The script runs the REST server to provide inference for that model via an HTTP endpoint.
- The example Inception-v1 ONNX model and the `onnx_inception_v1` script are also located in the `/workspace/tensorrt_server` directory. This example and script runs the REST server to provide inference for that model via an HTTP endpoint.
- The example ResNet-152 TensorFlow model and the `tensorflow_resnet` script are also located in the `/workspace/tensorrt_server` directory. This example and script runs the REST server to provide inference for that model via an HTTP endpoint.

The container also includes the following:

- **Ubuntu** 16.04
- **NVIDIA CUDA** 9.0.176 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.234
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.0.5
- NCCL 2.1.2 (optimized for NVLink™)

**Driver Requirements**

Release 17.12 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

**Key Features and Enhancements**

This is the first TensorRT container release.

**Known Issues**

There are no known issues in this release.
Notice

THE INFORMATION IN THIS GUIDE AND ALL OTHER INFORMATION CONTAINED IN NVIDIA DOCUMENTATION REFERENCED IN THIS GUIDE IS PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE INFORMATION FOR THE PRODUCT, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the product described in this guide shall be limited in accordance with the NVIDIA terms and conditions of sale for the product.

THE NVIDIA PRODUCT DESCRIBED IN THIS GUIDE IS NOT FAULT TOLERANT AND IS NOT DESIGNED, MANUFACTURED OR INTENDED FOR USE IN CONNECTION WITH THE DESIGN, CONSTRUCTION, MAINTENANCE, AND/OR OPERATION OF ANY SYSTEM WHERE THE USE OR A FAILURE OF SUCH SYSTEM COULD RESULT IN A SITUATION THAT THREATENS THE SAFETY OF HUMAN LIFE OR SEVERE PHYSICAL HARM OR PROPERTY DAMAGE (INCLUDING, FOR EXAMPLE, USE IN CONNECTION WITH ANY NUCLEAR, AVIONICS, LIFE SUPPORT OR OTHER LIFE CRITICAL APPLICATION). NVIDIA EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR SUCH HIGH RISK USES. NVIDIA SHALL NOT BE LIABLE TO CUSTOMER OR ANY THIRD PARTY, IN WHOLE OR IN PART, FOR ANY CLAIMS OR DAMAGES ARISING FROM SUCH HIGH RISK USES.

NVIDIA makes no representation or warranty that the product described in this guide will be suitable for any specified use without further testing or modification. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to ensure the product is suitable and fit for the application planned by customer and to do the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this guide. NVIDIA does not accept any liability related to any default, damage, costs or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this guide, or (ii) customer product designs.

Other than the right for customer to use the information in this guide with the product, no other license, either expressed or implied, is hereby granted by NVIDIA under this guide. Reproduction of information in this guide is permissible only if reproduction is approved by NVIDIA in writing, is reproduced without alteration, and is accompanied by all associated conditions, limitations, and notices.

Trademarks

NVIDIA, the NVIDIA logo, and cuBLAS, CUDA, cuDNN, cuFFT, cuSPARSE, DALI, DIGITS, DGX, DGX-1, Jetson, Kepler, NVIDIA Maxwell, NCCL, NVLink, Pascal, Tegra, TensorRT, and Tesla are trademarks and/or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2019 NVIDIA Corporation. All rights reserved.

www.nvidia.com