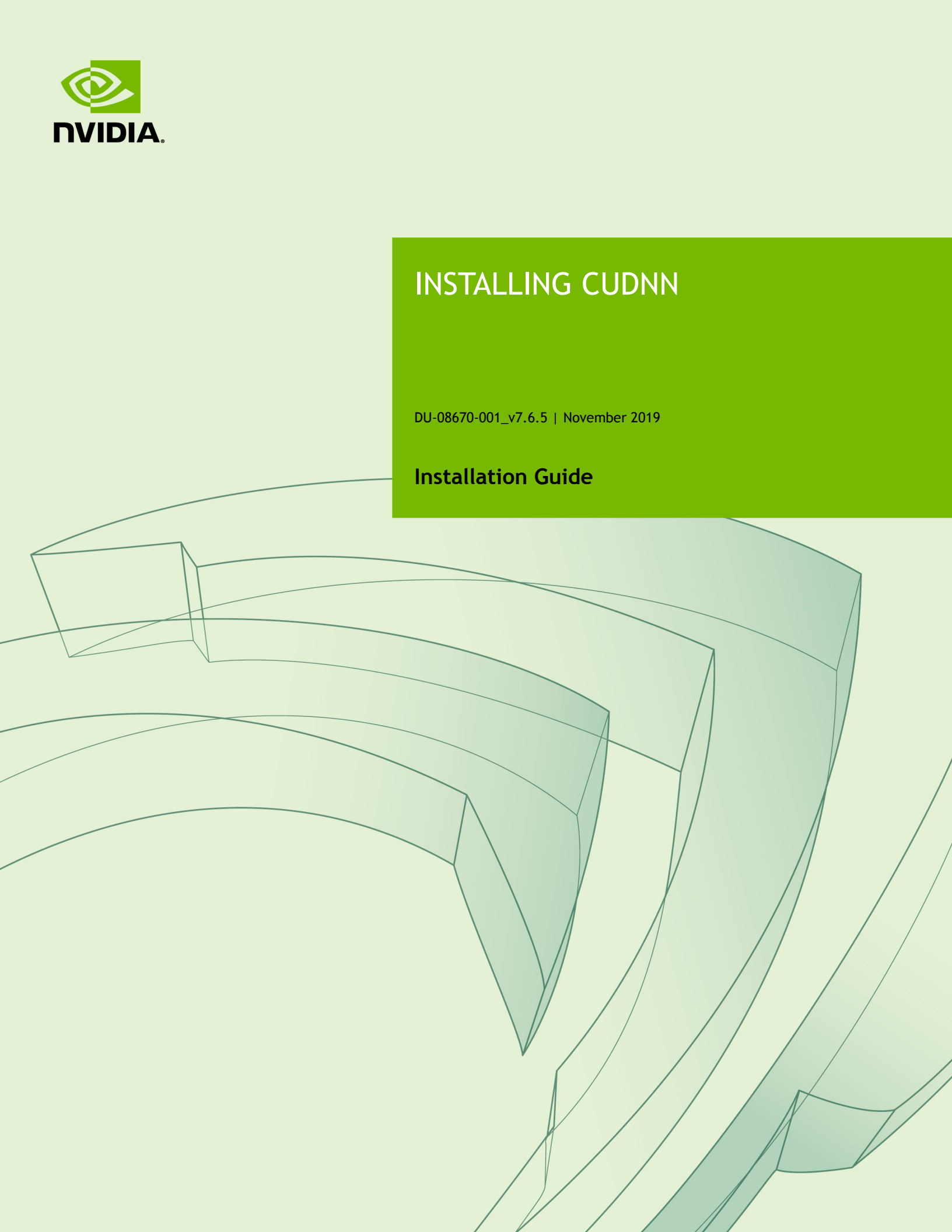

INSTALLING CUDNN

DU-08670-001_v7.6.5 | November 2019

Installation Guide

TABLE OF CONTENTS

Chapter 1. Overview 1
Chapter 2. Installing cuDNN On Linux 2
2.1. Prerequisites 2
2.1.1. Installing NVIDIA Graphics Drivers 2
2.1.2. Installing The CUDA Toolkit For Linux 2
2.2. Downloading cuDNN For Linux 2
2.3. Installing cuDNN On Linux 3
2.3.1. Installing From A Tar File 3
2.3.2. Installing From A Debian File 3
2.3.3. Installing From An RPM File 4
2.4. Verifying The cuDNN Install On Linux 4
2.5. Upgrading From v6 To v7 4
2.6. Troubleshooting 4
Chapter 3. Installing cuDNN On Mac OS X 5
3.1. Prerequisites 5
3.1.1. Installing NVIDIA Graphics Drivers 5
3.1.2. Installing The CUDA Toolkit For Mac OS X 5
3.2. Downloading cuDNN For Mac OS X 5
Chapter 4. Installing cuDNN On Windows 7
4.1. Prerequisites 7
4.1.1. Installing NVIDIA Graphic Drivers 7
4.1.2. Installing The CUDA Toolkit For Windows 7
4.2. Downloading cuDNN For Windows 7
4.3. Installing cuDNN On Windows 8
4.4. Upgrading From v6 To v7 9
4.5. Troubleshooting 9
Chapter 5. Cross-compiling cuDNN Samples 10
5.1. NVIDIA DRIVE OS Linux 10
5.1.1. Installing The CUDA Toolkit For DRIVE OS 10
5.1.2. Installing cuDNN For DRIVE OS 10
5.2. QNX 11
5.2.1. Installing The CUDA Toolkit For QNX 11
5.2.2. Installing cuDNN For QNX 11
5.2.3. Set The Environment Variables 11
5.2.4. Cross-compiling cuDNN Samples For QNX 11
Chapter 6. Package Manager Installation 12
6.1. Network Installation 12
6.1.1. Ubuntu 12
6.1.2. RHEL 12

Chapter 1. OVERVIEW

The NVIDIA CUDA Deep Neural Network library (cuDNN) is a GPU-accelerated library of primitives for deep neural networks. cuDNN provides highly tuned implementations for standard routines such as forward and backward convolution, pooling, normalization, and activation layers. cuDNN is part of the NVIDIA Deep Learning SDK.

Deep learning researchers and framework developers worldwide rely on cuDNN for high-performance GPU acceleration. It allows them to focus on training neural networks and developing software applications rather than spending time on low-level GPU performance tuning. cuDNN accelerates widely used deep learning frameworks and is freely available to members of the NVIDIA Developer Program.

Chapter 2. INSTALLING CUDNN ON LINUX

2.1. Prerequisites

Ensure you meet the following requirements before you install cuDNN.

- A GPU of compute capability 3.0 or higher. To understand the compute capability of the GPU on your system, see: CUDA GPUs.
- For the latest compatibility software versions of the OS, CUDA, the CUDA driver, and the NVIDIA hardware, see the cuDNN Support Matrix.

2.1.1. Installing NVIDIA Graphics Drivers

Install up-to-date NVIDIA graphics drivers on your Linux system.

1. Go to: NVIDIA download drivers
2. Select the GPU and OS version from the drop down menus.
3. Download and install NVIDIA graphics driver as indicated in that webpage. For more information, select the **ADDITIONAL INFORMATION** tab for step-by-step instructions for installing a driver.
4. Restart your system to ensure the graphics driver takes effect.

2.1.2. Installing The CUDA Toolkit For Linux

Refer to the following instructions for installing CUDA on Linux, including the CUDA driver and toolkit: NVIDIA CUDA Installation Guide for Linux.

2.2. Downloading cuDNN For Linux

In order to download cuDNN, ensure you are registered for the NVIDIA Developer Program.

1. Go to: NVIDIA cuDNN home page.
2. Click **Download**.
3. Complete the short survey and click **Submit**.
4. Accept the Terms and Conditions. A list of available download versions of cuDNN displays.
5. Select the cuDNN version you want to install. A list of available resources displays.

2.3. Installing cuDNN On Linux

The following steps describe how to build a cuDNN dependent program. Choose the installation method that meets your environment needs. For example, the tar file installation applies to all Linux platforms, and the debian installation package applies to Ubuntu 14.04,16.04, and 18.04.

In the following sections:

- your CUDA directory path is referred to as **/usr/local/cuda/**
- your cuDNN download path is referred to as **<cudnnpath>**

2.3.1. Installing From A Tar File

1. Navigate to your **<cudnnpath>** directory containing the cuDNN Tar file.
2. Unzip the cuDNN package.

```
$ tar -xzvf cudnn-9.0-linux-x64-v7.tgz
```

3. Copy the following files into the CUDA Toolkit directory, and change the file permissions.

```
$ sudo cp cuda/include/cudnn.h /usr/local/cuda/include
$ sudo cp cuda/lib64/libcudnn* /usr/local/cuda/lib64
$ sudo chmod a+r /usr/local/cuda/include/cudnn.h /usr/local/cuda/lib64/
libcudnn*
```

2.3.2. Installing From A Debian File

1. Navigate to your **<cudnnpath>** directory containing cuDNN Debian file.
2. Install the runtime library, for example:

```
sudo dpkg -i libcudnn7_7.0.3.11-1+cuda9.0_amd64.deb
```

3. Install the developer library, for example:

```
sudo dpkg -i libcudnn7-devel_7.0.3.11-1+cuda9.0_amd64.deb
```

4. Install the code samples and the cuDNN Library User Guide, for example:

```
sudo dpkg -i libcudnn7-doc_7.0.3.11-1+cuda9.0_amd64.deb
```

2.3.3. Installing From An RPM File

1. Download the rpm package **libcudnn*.rpm** to the local path.
2. Install the rpm package from the local path. This will install the cuDNN libraries.

```
rpm -ivh libcudnn7-*.x86_64.rpm
rpm -ivh libcudnn7-devel-*.x86_64.rpm
rpm -ivh libcudnn7-doc-*.x86_64.rpm
```

2.4. Verifying The cuDNN Install On Linux

To verify that cuDNN is installed and is running properly, compile the mnistCUDNN sample located in the **/usr/src/cudnn_samples_v7** directory in the debian file.

1. Copy the cuDNN sample to a writable path.

```
$cp -r /usr/src/cudnn_samples_v7/ $HOME
```

2. Go to the writable path.

```
$ cd  $HOME/cudnn_samples_v7/mnistCUDNN
```

3. Compile the mnistCUDNN sample.

```
$make clean && make
```

4. Run the mnistCUDNN sample.

```
$ ./mnistCUDNN
```

If cuDNN is properly installed and running on your Linux system, you will see a message similar to the following:

```
Test passed!
```

2.5. Upgrading From v6 To v7

cuDNN v7 can coexist with previous versions of cuDNN, such as v5 or v6.

2.6. Troubleshooting

Join the NVIDIA Developer Forum to post questions and follow discussions.

Chapter 3. INSTALLING CUDNN ON MAC OS X

3.1. Prerequisites

Ensure you meet the following requirements before you install cuDNN.

- A GPU of compute capability 3.0 or higher. To understand the compute capability of the GPU on your system, see: CUDA GPUs.
- For the latest compatibility software versions of the OS, CUDA, the CUDA driver, and the NVIDIA hardware, see the cuDNN Support Matrix.

3.1.1. Installing NVIDIA Graphics Drivers

Install up-to-date NVIDIA graphics drivers on your Mac OS X system.

1. Go to: NVIDIA download drivers
2. Select the GPU and OS version from the drop down menus.
3. Download and install NVIDIA graphics driver 396 or newer. For more information, select the **ADDITIONAL INFORMATION** tab for step-by-step instructions for installing a driver.
4. Restart your system to ensure the graphics driver takes effect.

3.1.2. Installing The CUDA Toolkit For Mac OS X

Refer to the following instructions for installing CUDA on Mac OS X, including the CUDA driver and toolkit: NVIDIA CUDA Installation Guide for Mac OS X.

3.2. Downloading cuDNN For Mac OS X

In order to download cuDNN, ensure you are registered for the NVIDIA Developer Program.

1. Go to: NVIDIA cuDNN home page.
2. Click **Download**.
3. Complete the short survey and click **Submit**.
4. Accept the Terms and Conditions. A list of available download versions of cuDNN displays.
5. Select the cuDNN version to want to install. A list of available resources displays.
6. Extract the cuDNN archive to a directory of your choice.

Chapter 4.
INSTALLING CUDNN ON WINDOWS

4.1. Prerequisites

Ensure you meet the following requirements before you install cuDNN.

- A GPU of compute capability 3.0 or higher. To understand the compute capability of the GPU on your system, see: CUDA GPUs.
- For the latest compatibility software versions of the OS, CUDA, the CUDA driver, and the NVIDIA hardware, see the cuDNN Support Matrix.

4.1.1. Installing NVIDIA Graphic Drivers

Install up-to-date NVIDIA graphics drivers on your Windows system.

1. Go to: NVIDIA download drivers
2. Select the GPU and OS version from the drop down menus.
3. Download and install NVIDIA driver as indicated in that webpage. For more information, select the **ADDITIONAL INFORMATION** tab for step-by-step instructions for installing a driver.
4. Restart your system to ensure the graphics driver takes effect.

4.1.2. Installing The CUDA Toolkit For Windows

Refer to the following instructions for installing CUDA on Windows, including the CUDA driver and toolkit: NVIDIA CUDA Installation Guide for Windows.

4.2. Downloading cuDNN For Windows

In order to download cuDNN, ensure you are registered for the NVIDIA Developer Program.

1. Go to: NVIDIA cuDNN home page.
2. Click **Download**.
3. Complete the short survey and click **Submit**.
4. Accept the Terms and Conditions. A list of available download versions of cuDNN displays.
5. Select the cuDNN version to want to install. A list of available resources displays.
6. Extract the cuDNN archive to a directory of your choice.

4.3. Installing cuDNN On Windows

The following steps describe how to build a cuDNN dependent program. In the following sections the CUDA v9.0 is used as example:

- Your CUDA directory path is referred to as **`C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0`**
- Your cuDNN directory path is referred to as **`<installpath>`**

1. Navigate to your **`<installpath>`** directory containing cuDNN.
2. Unzip the cuDNN package.

   ```
   cudnn-9.0-windows7-x64-v7.zip
   ```

   or

   ```
   cudnn-9.0-windows10-x64-v7.zip
   ```
3. Copy the following files into the CUDA Toolkit directory.
   a) Copy **`<installpath>\cuda\bin\cudnn64_7.dll`** to **`C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\bin`**.
   b) Copy **`<installpath>\cuda\ include\cudnn.h`** to **`C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\include`**.
   c) Copy **`<installpath>\cuda\lib\x64\cudnn.lib`** to **`C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\lib\x64`**.
4. Set the following environment variables to point to where cuDNN is located. To access the value of the **`$(CUDA_PATH)`** environment variable, perform the following steps:
   a) Open a command prompt from the **Start** menu.
   b) Type **`Run`** and hit **Enter**.
   c) Issue the **`control sysdm.cpl`** command.
   d) Select the **Advanced** tab at the top of the window.
   e) Click **Environment Variables** at the bottom of the window.
   f) Ensure the following values are set:

   ```
   Variable Name: CUDA_PATH
   Variable Value: C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0
   ```
5. Include **`cudnn.lib`** in your Visual Studio project.
   a) Open the Visual Studio project and right-click on the project name.

b) Click **Linker > Input > Additional Dependencies**.
c) Add **`cudnn.lib`** and click **OK**.

4.4. Upgrading From v6 To v7

cuDNN v7 can coexist with previous versions of cuDNN, such as v5 or v6.

4.5. Troubleshooting

Join the NVIDIA Developer Forum to post questions and follow discussions.

Chapter 5. CROSS-COMPILING CUDNN SAMPLES

This section describes how to cross-compile cuDNN samples.

5.1. NVIDIA DRIVE OS Linux

Follow the below steps to cross-compile cuDNN samples on NVIDIA DRIVE OS Linux.

5.1.1. Installing The CUDA Toolkit For DRIVE OS

1. Download the CUDA for Ubuntu package:**cuda*ubuntu*_amd64.deb**
2. Download the cross compile package: **cuda*-cross-aarch64*_all.deb**
3. Execute the following commands:
   a. `$ sudo dpkg -i cuda*ubuntu*_amd64.deb`
   b. `$ sudo dpkg -i cuda*-cross-aarch64*_all.deb`
   c. `$ sudo apt-get update`
   d. `$ sudo apt-get install cuda-toolkit-10-1 -y`
   e. `$ sudo apt-get install cuda-cross-aarch64* -y`

5.1.2. Installing cuDNN For DRIVE OS

1. Download the cuDNN Ubuntu package for your preferred CUDA Toolkit version: ***libcudnn7-cross-aarch64_*.deb**
2. Download the cross compile package: **libcudnn7-devel-cross-aarch64_*.deb**
3. Execute the following commands:
   a. `$ sudo dpkg -i *libcudnn7-cross-aarch64_*.deb`
   b. `$ sudo dpkg -i libcudnn7-devel-cross-aarch64_*.deb`

5.1.3. Cross-compiling cuDNN Samples For DRIVE OS

Copy the **cudnn_samples_v7** directory to your home directory:

```
$ cp -r /usr/src/cudnn_samples_v7 $HOME
```

For each sample, execute the following commands:

1. `$ cd $HOME/cudnn_samples_v7/(each sample)`
2. `$ make TARGET_ARCH=aarch64`

5.2. QNX

Follow the below steps to cross-compile cuDNN samples on QNX:

5.2.1. Installing The CUDA Toolkit For QNX

1. Download the CUDA for Ubuntu package:**cuda*ubuntu*_amd64.deb**
2. Download the cross compile package: **cuda*-cross-aarch64*_all.deb**
3. Execute the following commands:
   a. `$ sudo dpkg -i cuda*ubuntu*_amd64.deb`
   b. `$ sudo dpkg -i cuda*-cross-aarch64*_all.deb`
   c. `$ sudo apt-get update`
   d. `$ sudo apt-get install cuda-toolkit-10-1 -y`
   e. `$ sudo apt-get install cuda-cross-aarch64* -y`

5.2.2. Installing cuDNN For QNX

1. Download the cuDNN Ubuntu package for your preferred CUDA Toolkit version: ***libcudnn7-cross-aarch64_*.deb**
2. Download the cross compile package: **libcudnn7-devel-cross-aarch64_*.deb**
3. Execute the following commands:
   a. `$ sudo dpkg -i *libcudnn7-cross-aarch64_*.deb`
   b. `$ sudo dpkg -i libcudnn7-devel-cross-aarch64_*.deb`

5.2.3. Set The Environment Variables

To set the environment variables, issue the following commands:

1. **export CUDA_PATH={PATH}/install/cuda/**
2. **export QNX_HOST={PATH}/host/linux/x86_64**
3. **export QNX_TARGET={PATH}/target/qnx7**

5.2.4. Cross-compiling cuDNN Samples For QNX

Copy the **cudnn_samples_v7** directory to your home directory:

```
$ cp -r /usr/src/cudnn_samples_v7 $HOME
```

For each sample, execute the following commands:

1. `$ cd $HOME/cudnn_samples_v7/(each sample)`
2. `$ make TARGET_OS=QNX TARGET_ARCH=aarch64 HOST_COMPILER={SET FULL PATH to YOUR CROSS COMPILER}`

   For example: **make TARGET_OS=QNX TARGET_ARCH=aarch64 HOST_COMPILER=$QNX_HOST/usr/bin/aarch64-unknown-nto-qnx7.0.0-g++**

Chapter 6.
PACKAGE MANAGER INSTALLATION

The Package Manager installation interfaces with your system's package manager. When using RPM or Deb, the downloaded package is a repository package, not the actual installation package. This repository package informs the package manager only where to find the actual installation packages, but will not install them.

If the actual installation packages are available in an online repository, they will be automatically downloaded in a later step. Otherwise, the repository package also installs a local repository containing the installation packages on the system.

Whether the repository is available online or installed locally, the installation procedure is identical and made of several steps. See below.

6.1. Network Installation

6.1.1. Ubuntu

1. Download and install the repository: **`sudo dpkg -i`** http://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1604/x86_64/nvidia-machine-learning-repo-ubuntu1604_1.0.0-1_amd64.deb
2. Execute the steps below to install cuDNN library:
   a. `$ sudo apt-get update && sudo apt-get install libcudnn7-devel`
   b. `$ sudo apt-get install libcudnn7=[cudnn_version+cuda_version]`
   c. `$ sudo apt-get install libcudnn7-devel=[cudnn_version+cuda_version]`

   For example, for CUDA 9.0 and cuDNN 7.4.1:

   ```
   $ sudo apt-get install libcudnn7=7.4.1.5-1+cuda9.0 sudo apt-get install
    libcudnn7-devel=7.4.1.5-1+cuda9.0
   ```

6.1.2. RHEL

1. Download and install the repository: **`rpm -ivh`** http://developer.download.nvidia.com/compute/machine-learning/repos/rhel7/x86_64/nvidia-machine-learning-repo-rhel7-1.0.0-1.x86_64.rpm

2. Install the cuDNN library:

   a. For the latest version:

   ```
   $ sudo yum install libcudnn7
   ```

   ```
   $ sudo yum install libcudnn7-devel
   ```

   b. For other versions:

   ```
   $ sudo yum install libcudnn7=[cudnn_version+cuda_version]
   ```

   ```
   $ sudo yum install libcudnn7-devel=[cudnn_version+cuda_version]
   ```

   For example, for CUDA 9.0 and cuDNN 7.4.2:

   ```
   $ sudo yum install libcudnn7=7.4.2.24-1+cuda9.0
   ```

   ```
   $ sudo yum install libcudnn7-devel=7.4.2.24-1+cuda9.0
   ```

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, DALI, DIGITS, DGX, DGX-1, DGX-2, DGX Station, DLProf, Jetson, Kepler, Maxwell, NCCL, Nsight Compute, Nsight Systems, NvCaffe, PerfWorks, Pascal, SDK Manager, Tegra, TensorRT, TensorRT Inference Server, Tesla, TF-TRT, and Volta 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