## TABLE OF CONTENTS

Chapter 1. Overview Of DIGITS................................................................. 1  
Chapter 2. Installation Overview............................................................. 2  
Chapter 3. GPU....................................................................................... 3  
Chapter 4. Driver..................................................................................... 4  
Chapter 5. CUDA Toolkit....................................................................... 5  
  5.1. Deb packages.................................................................................. 5  
Chapter 6. Environment.......................................................................... 7  
Chapter 7. Troubleshooting.................................................................... 8  
  7.1. Support.......................................................................................... 8
Chapter 1.
OVERVIEW OF DIGITS

DIGITS (the Deep Learning GPU Training System) is a webapp for training deep learning models. The currently supported frameworks are: Caffe and Tensorflow. DIGITS puts the power of deep learning into the hands of engineers and data scientists.

DIGITS is not a framework. DIGITS is a wrapper for Caffe and TensorFlow; which provides a graphical web interface to those frameworks rather than dealing with them directly on the command-line.

DIGITS can be used to rapidly train highly accurate deep neural network (DNNs) for image classification, segmentation, object detection tasks, and more. DIGITS simplifies common deep learning tasks such as managing data, designing and training neural networks on multi-GPU systems, monitoring performance in real time with advanced visualizations, and selecting the best performing model from the results browser for deployment. DIGITS is completely interactive so that data scientists can focus on designing and training networks rather than programming and debugging.

DIGITS is available through multiple channels such as:

- GitHub download
- NVIDIA’s Docker repository, nvcr.io
Chapter 2.
INSTALLATION OVERVIEW

Getting CUDA and the NVIDIA driver installed correctly on your machine can be difficult. This guide provides you installation instructions for Ubuntu. Another good resource is the CUDA installation guide for Linux.
You will need an NVIDIA GPU to use CUDA. If you want to use cuDNN, you will need a GPU with compute capability >= 3.0. To find out what the compute capability of your card is, see one of these websites:

- https://en.wikipedia.org/wiki/CUDA#GPUs_supported

You can also use the DIGITS device_query tool to check for the compute major and minor versions:

```
$ digits/device_query.py
Device #0:
>>> CUDA attributes:
   name       Tesla K40c
   totalGlobalMem  12079136768
   clockRate   745000
   major       3
   minor       5
>>> NVML attributes:
   Total memory  11519 MB
   Used memory   23 MB
   Memory utilization  0%
   GPU utilization  0%
   Temperature   30 C
```
Chapter 4.
DRIVER

On Ubuntu, you can install a driver in two ways: with a run file or with a Deb package. It is recommended that you use a Deb package to install your driver, unless you have a new GPU that requires a newer driver version. Deb packages are simpler to install, uninstall and upgrade, while run file installers are useful if you need a newer driver version.

To install with a run file, download one from the NVIDIA Driver Downloads website and follow the instructions. If you run into any problems, look at the "Additional Information" section.

**Important** If you use a run file to install your driver, don't install the cuda Deb package. More information below.
Chapter 5.
CUDA TOOLKIT

On the CUDA Downloads website, you will see three options for installing the toolkit: runfile (local), deb (local), deb (network).

1. **deb (network)** - This is a Deb package, the preferred method. This gives you access to all of the packages in the CUDA repository, including multiple toolkit versions.
   
   Execute this command after reading warning below:
   ```
   dpkg -i cuda-repo-ubuntu1404_7.5-18_amd64.deb
   apt-get update
   ```

2. **deb (local)** - This is also a Deb package, a nice option if you have a bad network connection. The downside is that you can't get package updates and you have to install separate packages for CUDA 7.0, 7.5, etc.

   Execute this command after reading warning below:
   ```
   dpkg -i cuda-repo-ubuntu1404-7-5-local_7.5-18_amd64.deb
   apt-get update
   ```

3. **runfile (local)** - Shell script. Don't use this unless you have to for some reason. As with the driver (see above), it is more difficult to uninstall or upgrade your CUDA installation if you use a run file installer

   ```
   sh cuda_7.5.18_linux.run
   ```

### 5.1. Deb packages

If you chose to use a Deb package, here are some of the packages you can install:

1. **apt-get install cuda** - This will install the latest toolkit (currently 7.5) and the latest driver (currently nvidia-352).

   **Important** Don't install this package if you installed your driver with a run file. The Deb package may not be able to fully uninstall your run file driver installation.
2. `apt-get install cuda-toolkit-7-5` - Installs only the toolkit and not the driver.

3. `apt-get install cuda-drivers` - Installs only the driver and not the toolkit

For more information, see the "Meta Packages" section of the CUDA installation guide for Linux.
Chapter 6.
ENVIRONMENT

Set up your environment correctly so that the runtime linker can find your shared libraries. There are a few ways to do this:

Your environment will be set up automatically with the CUDA 8.0 installers.

1. Add an entry to `/etc/ld.so.conf.d/`.
   - Requires sudo privileges.
   - Enter this command:
     
     ```bash
     echo "/usr/local/cuda/lib64" | sudo tee /etc/ld.so.conf.d/cuda64.conf
     sudo ldconfig
     ```

2. Edit `LD_LIBRARY_PATH`.
   - Does not require sudo privileges.
   - The exact formula required depends on which shell you are using and how you login to your machine.
   - Use:
     
     ```bash
     echo "export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/usr/local/cuda/lib64" >> ~/.profile && source ~/.profile
     
     # Non-login interactive shell (bash)
     echo "export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/usr/local/cuda/lib64" >> ~/.bashrc && source ~/.bashrc
     ```

   For more information on setting persistent environment variables see:
   - [http://unix.stackexchange.com/q/117467/99570](http://unix.stackexchange.com/q/117467/99570)
   - [http://askubuntu.com/q/210884/336440](http://askubuntu.com/q/210884/336440)

3. Install the `cuda-ld-conf-7-0` package
   - This package is made available on NVIDIA’s [machine learning repo](https://github.com/NVIDIA/Machine-Learning-Repository).
   - When you install DIGITS with a Deb package, the package gets installed automatically.
   - Sets up option (1) for you automatically
Chapter 7.
TROUBLESHOOTING

For troubleshooting tips see the Nvidia DIGITS Troubleshooting and Support Guide.

7.1. Support


For more information about DIGITS, see:

- DIGITS website (https://developer.nvidia.com/digits)
- DIGITS project (https://github.com/NVIDIA/DIGITS/blob/digits-5.0/README.md)
- GitHub documentation (https://github.com/NVIDIA/nvidia-docker/wiki/DIGITS)

There may be slight variations between the NVIDIA-docker images and this image.
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 U.S. 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