

Software Download

Direct Download

PowerAI is distributed as a binary for Ubuntu 16.04 LTS from the following source:

<https://public.dhe.ibm.com/software/server/POWER/Linux/mldl/ubuntu/>

Release Guide

A complete Release Guide with package list, prerequisites, deployment guide, and developer information is available at: <https://developer.ibm.com/linuxonpower/deep-learning-powerai/releases/>

System setup

Operating system

The Deep Learning packages require Ubuntu 16.04 for IBM POWER8®. Ubuntu installation images can be downloaded from:

<http://www.ubuntu.com/download/server/power8>

NOTE: PowerAI Release 4 requires the version 4.4 linux kernel. Ubuntu 16.04 supports two different kernel versions: the base kernel (version 4.4), and the Hardware Enablement kernel (currently version 4.8; see <https://wiki.ubuntu.com/Kernel/RollingLTSEnablementStack>). Be sure to install the base 4.4 kernel for PowerAI.

NVIDIA components

The Deep Learning packages require NVIDIA CUDA 8 and cuDNN 6.0, which can be installed as follows:

1. Download and install NVIDIA CUDA 8 from <https://developer.nvidia.com/cuda-downloads-power8>
 - CUDA 8.0 or higher required; 8.0.61 is recommended
 - Select the *Installer Type* that best fits your needs
 - Follow the Linux installation instructions in the [CUDA Quick Start Guide](#) linked from the download page, including the steps describing how to set up the CUDA development environment by updating `PATH` and `LD_LIBRARY_PATH`.
2. Download NVIDIA cuDNN 6.0 for CUDA 8 POWER8 Deb packages from <https://developer.nvidia.com/cudnn>
 - cuDNN v6.0 Runtime Library Ubuntu 16.04 Power8(Deb)
 - cuDNN v6.0 Developer Library for Ubuntu 16.04 Power8 (Deb)
 - cuDNN v6.0 Code Samples and User Guide Power8 (Deb)

Registration in NVIDIA's Accelerated Computing Developer Program is required.

3. Install the cuDNN v6.0 packages
 - `$ sudo dpkg -i libcudnn6*deb`

NVIDIA GPU driver update

Using driver 384.59 or higher is required. 384.59 is recommended.

NVIDIA drivers for POWER8 are available from <https://www.nvidia.com> (select *DRIVERS, All NVIDIA Drivers*)

Software repository setup

The PowerAI Deep Learning packages are provided via two different installation methods:

- The local repository package (`mldl-repo-local`) creates an installation repository on the local machine. This method is best for systems with limited internet access or where strong control of upgrades is desired
- The network repository package (`mldl-repo-network`) creates a reference on the local machine to the PowerAI network repository. This method is best for internet connected systems that will be readily updated to new versions of PowerAI

These packages are mutually exclusive. Choose one or the other for your systems.

Software repository setup is similar for either method:

1. Download the desired repository package (.deb file) from: <https://public.dhe.ibm.com/software/server/POWER/Linux/mldl/ubuntu/>
2. Install the repository package:
 - `$ sudo dpkg -i mldl-repo-*.deb`
3. Update the package cache
 - `$ sudo apt-get update`

Installing the Deep Learning frameworks

Installing all frameworks at once

All the Deep Learning frameworks can be installed at once using the `power-mldl` meta-package:

```
$ sudo apt-get install power-mldl
```

Installing frameworks individually

The Deep Learning frameworks can be installed individually if preferred. The framework packages are:

- `caffe-bvlc` - Berkeley Vision and Learning Center (BVLIC) upstream Caffe (v1.0.0)
- `caffe-ibm` - IBM Optimized version of the BVLIC upstream Caffe v1.0.0
- `caffe-nv` - NVIDIA fork of Caffe v0.15.14
- `chainer` - Chainer, v1.23.0
- `tensorflow` - Google TensorFlow, v1.1.0
- `ddl-tensorflow` - Distributed Deep Learning custom operator for TensorFlow
- `theano` - Theano v0.9.0
- `torch` - Torch v7
- `digits` - NVIDIA DIGITS, v5.0.0

Each can be installed with:

```
$ sudo apt-get install <framework>
```

Installation note for IBM Caffe and DDL custom operator for TensorFlow

The `caffe-ibm` and `ddl-tensorflow` packages require the PowerAI OpenMPI package which is built with NVIDIA CUDA support. That OpenMPI package conflicts with Ubuntu's non-CUDA-enabled OpenMPI packages.

Please uninstall any `openmpi` or `libopenmpi` packages before installing IBM Caffe or DDL custom operator for TensorFlow. Purge any configuration files to avoid interference::

```
$ dpkg -l | grep openmpi  
$ sudo apt-get purge ...
```

Installation note for DIGITS

The `digits` and `python-socketio-server` packages conflict with Ubuntu's older `python-socketio` package. Please uninstall the `python-socketio` package before installing DIGITS.

Upgrading from a previous release

From R3.4 or earlier

PowerAI Release 4 has unique upgrade requirements due to

dependencies on new NVIDIA software:

NOTE: PowerAI Release 4 requires new versions of the NVIDIA GPU driver (384) and cuDNN (6.0). The recommended upgrade process is to uninstall the older version of PowerAI, update the NVIDIA components, then install the new version of PowerAI.

1. Remove the previous version of PowerAI, including the repo package

```
$ dpkg -l | egrep 'mldl|3ibm'  
$ sudo apt-get purge ...  
$ sudo apt-get update
```

2. Update the NVIDIA components

- Remove cuDNN v5

```
$ dpkg -l | grep cudnn  
$ sudo apt-get purge ...
```

- Remove the 361. series driver

```
$ dpkg -l | grep 361  
$ sudo apt-get purge ...  
$ sudo apt-get update
```

3. Install cuDNN v6.0 as described above

4. Install the 384-series NVIDIA Tesla GPU driver as described above

5. Reboot to activate the new drive

6. Install the new PowerAI packages as described above

Tuning Recommendations

Recommended settings for optimal Deep Learning performance on the S822LC for HPC:

- Enable Performance Governor

```
$ sudo apt-get install linux-tools-common  
linux-tools-generic cpufrequtils lsb-release  
$ sudo cpupower -c all frequency-set -g performance
```
- Enable GPU persistence mode:
Use `nvidia-persistenced` (<http://docs.nvidia.com/deploy/driver-persistence>) or

```
$ sudo nvidia-smi -pm ENABLED
```
- Set GPU memory, graphics clocks (Tesla P100-only):

```
$ sudo nvidia-smi -ac 715,1480
```

- Set optimal SMT mode for TensorFlow:
\$ sudo apt-get install powerpc-ibm-utils
\$ sudo ppc64_64 --smt=2

Getting started with MLDL frameworks

General setup

Most of the PowerAI packages install outside the normal system search paths (e.g. to `/opt/DL/...`), so each framework package provides a shell script to help simplify environmental setup.

We recommend that users update their shell rc file (e.g. `.bashrc`) to source the desired setup scripts. For example:

```
source /opt/DL/<framework>/bin/<framework>-activate
```

Each framework also provides a test script to verify basic function:

```
$ <framework>-test
```

Getting started with Caffe

Caffe alternatives

Packages are provided for upstream BVLC Caffe (`/opt/DL/caffe-bvlc`), IBM's optimized BVLC Caffe (`/opt/DL/caffe-ibm`), and NVIDIA's fork of Caffe (`/opt/DL/caffe-nv`). The system default Caffe (`/opt/DL/caffe`) can be selected using Ubuntu's alternatives system:

```
$ sudo update-alternatives --config caffe
```

There are 3 choices for the alternative `caffe` (providing `/opt/DL/caffe`).

Selection	Path	Priority	Status

* 0	<code>/opt/DL/caffe-ibm</code>	100	auto mode
1	<code>/opt/DL/caffe-bvlc</code>	50	manual mode
2	<code>/opt/DL/caffe-ibm</code>	100	manual mode
3	<code>/opt/DL/caffe-nv</code>	75	manual mode

Press enter to keep the current choice[*], or type selection number:

Users can activate the system default `caffe`:

```
source /opt/DL/caffe/bin/caffe-activate
```

Or they can activate a specific variant. For example:

```
source /opt/DL/caffe-bvlc/bin/caffe-activate
```

Attempting to activate multiple Caffe packages in a single login session will cause unpredictable behavior.

Caffe samples and examples

Each Caffe package includes example scripts and sample models, etc. A script is provided to copy the sample content into a specified directory:

```
$ caffe-install-samples <somedir>
```

More information on the available Caffe alternatives for each release of PowerAI can be accessed via the Release Readme files linked at:

<https://developer.ibm.com/linuxonpower/deep-learning-powerai/releases/>

More info on Caffe

Visit Caffe's website at <http://caffe.berkeleyvision.org/> for tutorials and example programs that you can run to get started.

Here are links to a couple of the example programs from <http://caffe.berkeleyvision.org/>:

- **LeNet MNIST tutorial** — <http://caffe.berkeleyvision.org/gathered/examples/mnist.html> — Train a neural network to understand handwritten digits
- **CIFAR-10 tutorial** — <http://caffe.berkeleyvision.org/gathered/examples/cifar10.html> — Train a convolutional neural network to classify small images

Optimizations in IBM Caffe

The IBM Caffe packages in PowerAI is based on BVLC Caffe and includes optimizations and enhancements from IBM

- CPU/GPU layer-wise reduction
- Large Model Support (LMS)
- IBM PowerAI Distributed Deep Learning

For more information including tuning instructions, please visit the [PowerAI Readme \(https://public.dhe.ibm.com/software/server/POWER/Linux/midl/ubuntu/README.html\)](https://public.dhe.ibm.com/software/server/POWER/Linux/midl/ubuntu/README.html)

Getting started with Chainer

The [Chainer homepage \(http://www.chainer.org/\)](http://www.chainer.org/) includes documentation for the Chainer project, including a Quick Start example.

Getting started with TensorFlow

The [TensorFlow homepage \(https://www.tensorflow.org/\)](https://www.tensorflow.org/) has a variety of information including Tutorials, How Tos, and a Getting Started guide.

Additional tutorials and examples are available from the community:

- <https://github.com/nlitz/TensorFlow-Tutorials>
- <https://github.com/aymericdamien/TensorFlow-Examples>

Distributed Deep Learning (DDL) custom operator for TensorFlow

This release of PowerAI includes a Technology Preview of the IBM PowerAI Distributed Deep Learning (DDL) custom operator for TensorFlow. The DDL custom operator uses CUDA-aware OpenMPI and NCCL to provide high-speed communications for distributed TensorFlow.

The DDL custom operator can be found in the `ddl-tensorflow` package. For more information about DDL and about the TensorFlow operator, see:

- `/opt/DL/ddl/doc/README.md`
- `/opt/DL/ddl-tensorflow/doc/README.md`
- `/opt/DL/ddl-tensorflow/doc/README-API.md`

For more information including tuning instructions, please visit the [PowerAI Readme \(https://public.dhe.ibm.com/software/server/POWER/Linux/midl/ubuntu/README.html\)](https://public.dhe.ibm.com/software/server/POWER/Linux/midl/ubuntu/README.html)

Additional TensorFlow features

PowerAI TensorFlow packages include TensorBoard. https://www.tensorflow.org/get_started/summaries_and_tensorboard:

- HDF5
- NCCL
- Experimental XLA JIT Compilation (see <https://www.tensorflow.org/versions/master/experimental/xla/>)

Getting started with Theano

Here are some links to help you get started with Theano:

- Theano tutorial — <http://deeplearning.net/software/theano/tutorial/index.html#tutorial>
- MNIST digit classification — <http://deeplearning.net/tutorial/logreg.html>

Visit <http://deeplearning.net/software/theano/> for the latest from Theano.

Getting started with DIGITS

The first time it's run `digits-activate` will create a `.digits` subdirectory containing the DIGITS jobs directory, as well as the `digits.log` file

Multiple instances of the DIGITS server can be run at once, including by different users, but users may need to set the network port number to avoid conflicts.

To start DIGITS server with default port (5000):

```
$ digits-devserver
```

To start DIGITS server with specific port:

```
$ digits-devserver -p <port_num>
```

NVIDIA's [DIGITS website \(https://developer.nvidia.com/digits\)](https://developer.nvidia.com/digits) has more information about DIGITS

The [DIGITS Getting Started Guide \(https://github.com/NVIDIA/DIGITS/blob/master/docs/GettingStarted.md\)](https://github.com/NVIDIA/DIGITS/blob/master/docs/GettingStarted.md) describes how to train a network model to classify the MNIST hand-written digits dataset.

Additional DIGITS examples are available at <https://github.com/NVIDIA/DIGITS/tree/master/examples>

NOTE: The PowerAI Torch package is updated to work with DIGITS. Manual installation of individual lua rocks is no longer required.

Getting started with Torch

<https://github.com/torch/torch7/wiki/Cheatsheet> contains info for people new to Torch, including tutorials and examples.

- The Torch project has a demos repository at <https://github.com/torch/demos>
- Tutorials can be found at—
<https://github.com/torch/tutorials>

Visit <http://torch.ch/> for the latest from Torch.



© Copyright IBM Corporation 2017

IBM Corporation
Systems Group
Route 100
Somers, NY 10589

Produced in the United States of America
August 2017

IBM, the IBM logo, ibm.com, POWER, Power, POWER8, and Power systems are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

TensorFlow includes software (BoringSSL) developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)

TensorFlow includes cryptographic software written by Eric Young (ey@cryptsoft.com)

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

THE INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.



Please Recycle