The IBM PowerAI deep learning frameworks

(Release 3.1)

Highlights

- Includes popular deep learning frameworks available as easier to use distribution of pre-built binaries
- Optimized for the new IBM Power System S822LC for High Performance Computing
- Convenient installation and upgrades through standard Ubuntu system installer

Machine learning applications are among the most exciting innovations in IT in this decade. Deep learning is a subset of machine learning based on a programming model where neural networks make sense of data by classifying information based on exemplary training patterns.

This technology can be used for a broad set of purposes. For example, new driver assist technologies rely on machine and deep learning patterns to recognize objects in a rapidly changing environment, personal digital assistant technology is learning to categorize information contained in emails and text messages based on context, and in the enterprise, machine and deep learning applications can be used to identify high-value sales opportunities, provide assistance in call centers, detect instances of intrusion or fraud and suggest solutions to technical or business problems.

The PowerAI Deep Learning Frameworks were created to give developers and data scientists a platform on which to develop new machine learning-based applications and to analyze data with immediate productivity, ease of use, and high performance.

The third release of the PowerAI Deep Learning Frameworks and supporting libraries consists of some of the most advanced and popular

deep learning frameworks in the research community:

- BVLC Caffe
- NVIDIA Caffe
- IBM Caffe
- TensorFlow*
- Torch
- Theano
- OpenBLAS
- NCCL*
- NVIDIA DIGITS*

** = New in version 3.1*

The PowerAI Deep Learning Frameworks provide deep learning application developers and scientists an integrated environment that can be easily set up for immediate productivity. At the same time, the PowerAI Deep Learning Frameworks are updated to include advances in machine learning technology and leading-edge OpenPOWER hardware, helping to provide investment protection for machine and deep learning applications and a stable, compatible interface across future generations of the PowerAI Deep Learning Frameworks and co-optimized OpenPOWER hardware platforms.

The PowerAI Deep Learning Frameworks offer easy installation using standard Ubuntu installation processes.

High-performance Computing

Application developers can execute their deep learning algorithms either on an OpenPOWER central processing unit (CPU) or using a graphics processing unit (GPU) numeric accelerator. Numeric accelerators are accessed using device drivers and libraries provided by the accelerator manufacturers. The third release of the PowerAI Deep Learning Frameworks is based on the use of Ubuntu 16.04 on IBM® POWER® with NVIDIA CUDA 8 and cuDNN v5.1 packages.

Hardware platform description and ordering information

The PowerAI Deep Learning Frameworks include popular deep learning framework packages that are tuned for use with the following hardware configurations:

- IBM Power Systems™ S822LC for High Performance Computing (Model 8335-GTB) with up to four NVLINK attached NVIDIA Tesla P100 GPUs. (http://www-03.ibm.com/systems/power/hardware/s822lc-hpc/)
- IBM Power Systems™ S822LC (Model 8335-GCA and 8335-GTA) with up to two NVIDIA Tesla K80 accelerator cards, offering up to four numeric accelerator devices.

Please contact IBM for help configuring or placing an order.

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: After installing Ubuntu 16.04 update the *libc6* package to version *2.23-0ubuntu5* or higher. That version fixes problems affecting Torch and TensorFlow. You may need to enable the updates repository to install this update.

NVIDIA components

The Deep Learning packages require NVIDIA CUDA 8 and cuDNN 5.1, which can be installed as follows:

1. Download and install NVIDIA CUDA 8 from
https://developer.nvidia.com/cuda-downloads-power8

- This PowerAI release was tested with CUDA 8.0.44
- 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 5.1 for CUDA 8 POWER8 Deb packages from https://developer.nvidia.com/cudnn
- cuDNN v5.1 Runtime Library Ubuntu 16.04 Power8(Deb)
- cuDNN v5.1 Developer Library for Ubuntu 16.04 Power8 (Deb)
- cuDNN v5.1 Code Samples and User Guide Power8 (Deb)

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

3. Install the cuDNN v5.1 packages

```
$ sudo dpkg -i libcudnn5*deb
```

NVIDIA GPU driver update

The Deep Learning packages will work with the version 361 GPU driver that ships with CUDA 8 but IBM recommends using driver 361.93.03 or higher.

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

Software repository setup

The Deep Learning packages are published as an Ubuntu package that sets up an installation repository on the local machine. The repository can be enabled as follows:

1. Download the latest mldl-repo-local .deb file from:
https://download.boulder.ibm.com/ibmdl/pub/software/server/mldl/

2. Install the repository package:

```
$ sudo dpkg -i mldl-repo-local*.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 (BVLC) upstream Caffe (v1.0.0rc3)
- `caffe-ibm` - IBM Optimized version of the BVLC upstream Caffe v1.0-rc3
- `caffe-nv` - NVIDIA fork of Caffe v0.15.13
- `tensorflow` - Google TensorFlow, v0.9.0
- `theano` - Theano v0.8.2
- `torch` - Torch v7
- `digits` - NVIDIA DIGITS, v5.0.0-rc.1

Each can be installed with:

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

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.

Tuning Recommendations

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

- Enable Performance Governor

```
$ sudo apt-get install linux-tools-common
  cpufrequtils
$ sudo cpupower -c all frequency-set -g
  performance
```

- Enable GPU persistent mode:

Use *nvidia-persistenced* (http://docs.nvidia.com/deploy/driver-persistence/) or

```
$ sudo nvidia-smi -pm ENABLED
```

- Set GPU memory, graphics clocks:

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

- Set optimal SMT mode for TensorFlow:

```
$ sudo ppc64_64 --smt=2
```

Getting started with MLDL frameworks

General setup

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 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         /opt/DL/caffe-ibm    100       auto mode

  1         /opt/DL/caffe-bvlc    50       manual mode

  2         /opt/DL/caffe-ibm    100       manual mode

  3         /opt/DL/caffe-nv      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 info
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

Getting started with TensorFlow
The TensorFlow homepage (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/nlintz/TensorFlow-Tutorials
- https://github.com/aymericdamien/TensorFlow-Examples

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) has more information about DIGITS

The DIGITS Getting Started Guide (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: This DIGITS package will not work as-is with Torch. Torch support needs additional pre-requisite packages. Those packages are not supplied with the PowerAI distribution at this time.

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.

Extending Torch with additional Lua rocks
The Torch package includes several Lua rocks useful for creating Deep Learning applications. Additional Lua rocks can be installed locally to extend functionality. For example a rock providing NCCL bindings can be installed by:

```
$ source /opt/DL/torch/bin/torch-activate
$ source /opt/DL/nccl/bin/nccl-activate

$ luarocks install --local --deps-mode=all
"https://raw.githubusercontent.com/ngimel/
nccl.torch/master/nccl-scm-1.rockspec"
...
    nccl scm-1 is now built and installed
in /home/user/.luarocks/ (license: BSD)

$ luajit
LuaJIT 2.1.0-beta1 -- Copyright (C) 2005-
2015 Mike Pall. http://luajit.org/
    JIT: OFF
    > require 'torch'
    > require 'nccl'
    >
```

IBM Corporation
Systems Group
Route 100
Somers, NY 10589

Produced in the United States of America
December 2016

POD03115USEN-04