Installing and Running Caffe2

Caffe2 is a deep learning framework that supports both convolutional and recurrent networks

Frameworks Dependencies

Caffe2 requires protobuf, atlas, glog, gtest, limdb, leveldb, snappy, OpenMP, OpenCV, pthread-stubs, cmake, python-protobuf, and numpy.

For GPU acceleration CUDA and cuDNN are required. The current version of cuDNN is supported (5.1.10).

Example process for installing the dependencies on a Ubuntu 16.04 system are listed below.

sudo apt-get install libprotobuf-dev protobuf-compiler libatlas-base-dev libgoogle-glog-dev libgtest-dev liblmdb-dev libleveldb-dev libsnappy-dev python-dev python-pip libiomp-dev libopencv-dev libpthread-stubs0-dev cmake python-protobuf git
sudo pip install numpy [matplotlib ipython (if one wants to also use ipython notebooks)]

Install GPU libraries

wget http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/cuda-repo-ubuntu1604_8.0.61-1_amd64.deb
sudo dpkg -i cuda-repo-ubuntu1604_8.0.61-1_amd64.deb
wget http://developer.download.nvidia.com/compute/redist/cudnn/v5.1/libcudnn5_5.1.10-1+cuda8.0_amd64.deb
wget http://developer.download.nvidia.com/compute/redist/cudnn/v5.1/libcudnn5-dev_5.1.10-1+cuda8.0_amd64.deb
sudo dpkg -i libcudnn5*

Building Caffe2

git clone --recursive https://github.com/caffe2/caffe2.git
cd caffe2
mkdir build && cd build
cmake ..
make

Test Caffe2

There are different ways to verify that the Caffe2 build is functioning properly, one is to use example binaries located $CAFFE2_ROOT/build/caffe2/binaries and the other is with python scripts located in $CAFFE2_ROOT/build/caffe2/python. The binaries ending in "_test" can all be used to verify that the Caffe2 build is working properly. Below is an example output from the elementwise_op_gpu_test.

```
user@gpu-platform:~/framework/caffe2/build/caffe2/binaries$ ./elementwise_op_gpu_test
Running main() from gtest_main.cc
[==========] Running 4 tests from 1 test case.
[----------] Global test environment set-up.
[----------] 4 tests from ElementwiseGPUTest
[ RUN ] ElementwiseGPUTest.And
[ OK ] ElementwiseGPUTest.And (118 ms)
[ RUN ] ElementwiseGPUTest.Or
[ OK ] ElementwiseGPUTest.Or (2 ms)
[ RUN ] ElementwiseGPUTest.Xor
[ OK ] ElementwiseGPUTest.Xor (1 ms)
[ RUN ] ElementwiseGPUTest.Not
[ OK ] ElementwiseGPUTest.Not (1 ms)
[----------] 4 tests from ElementwiseGPUTest (122 ms total)

[----------] Global test environment tear-down
[==========] 4 tests from 1 test case ran. (122 ms total)
[ PASSED ] 4 tests.
user@gpu-platform:~/framework/caffe2/build/caffe2/binaries$
```

Multiple python scripts are provided to permit easy testing of Caffe2 with different neural network architectures. Example results from the convnet_benchmarks.py is shown below.

```
user@gpu-platform:~/frameworks/caffe2/build/caffe2/python$ python convnet_benchmarks.py --batch_size 1 --model AlexNet --cudnn_ws 500 --iterations 50
AlexNet: running forward-backward.
I0313 09:52:50.316722 9201 net_dag.cc:112] Operator graph pruning prior to chain compute took: 3.1274e-05 secs
I0313 09:52:50.316797 9201 net_dag.cc:363] Number of parallel execution chains 34 Number of operators = 61
I0313 09:52:50.316933 9201 net_dag.cc:525] Starting benchmark.
I0313 09:52:50.316938 9201 net_dag.cc:526] Running warmup runs.
I0313 09:52:51.110822 9201 net_dag.cc:536] Main runs.
I0313 09:52:52.975484 9201 net_dag.cc:547] Main run finished. Milliseconds per iter: 37.2928. Iters per second: 26.8149
```

If a import error is received when attempting to run any of the python scripts, add caffe2 to the PYTHONPATH or manually inside the script. Example lines for the terminal are

export PYTHONPATH=$PYTHONPATH:$CAFFE2_ROOT/build

Or from within the Python script one could use the Python sys module to ad the path.

```
CAFFE_ROOT=path/to/caffe2
import sys
sys.path.insert(0, CAFFE_ROOT+'/build')
```

This script can be used to test different batch sizes and networks, other than AlexNet. This command will provide results for two different batch sizes, 16 and 32, with the inception network configuration. Exemplary output from using a single GPU on a K80 is included below.

for BS in 16 32; do python convnet_benchmarks.py --batch_size $BS --model Inception --cudnn_ws 1000 ; done

```
user@sas03:~/frameworks/caffe2/build/caffe2/python$ for BS in 16 32; do python convnet_benchmarks.py --batch_size $BS --model Inception --cudnn_ws 500 ; done
Inception: running forward-backward.
I0317 00:54:50.650635 13008 net_dag.cc:123] Operator graph pruning prior to chain compute took: 0.000371156 secs
I0317 00:54:50.651175 13008 net_dag.cc:374] Number of parallel execution chains 381 Number of operators = 410
I0317 00:54:50.652022 13008 net_dag.cc:536] Starting benchmark.
I0317 00:54:50.652041 13008 net_dag.cc:537] Running warmup runs.
I0317 00:54:53.824313 13008 net_dag.cc:547] Main runs.
I0317 00:54:55.567314 13008 net_dag.cc:558] Main run finished. Milliseconds per iter: 174.29. Iters per second: 5.73757
Inception: running forward-backward.
I0317 00:54:59.600677 13051 net_dag.cc:123] Operator graph pruning prior to chain compute took: 0.000389294 secs
I0317 00:54:59.601260 13051 net_dag.cc:374] Number of parallel execution chains 381 Number of operators = 410
I0317 00:54:59.602026 13051 net_dag.cc:536] Starting benchmark.
I0317 00:54:59.602046 13051 net_dag.cc:537] Running warmup runs.
I0317 00:55:05.079577 13051 net_dag.cc:547] Main runs.
I0317 00:55:08.394855 13051 net_dag.cc:558] Main run finished. Milliseconds per iter: 331.52. Iters per second: 3.01641
- See more at: http://www.nvidia.com/object/caffe2-installation.html#sthash.RbTBKeK5.dpuf
```