Installing Deep Learning Packages

EE488C Special Topics in EE <Deep Learning & AlphaGo>
2016 Fall, School of EE, KAIST

- In this document, we explain how to install some deep learning packages. You don't need to read this document if you plan to use one of 24 servers by remote login or if you plan to use VirtualBox on your computer with virtual hard disk downloaded from KLMS because in such cases all packages are already installed. If you want to install VirtualBox to run Ubuntu linux on your computer, read "EE488C VirtualBox Howto.pdf".
- We assume Ubuntu is already installed.
- To use GPU for accelerated deep learning processing, you need to install CUDA Toolkit and cuDNN library. Follow the steps below. You don't need this step if you plan to use VirtualBox since VirtualBox does not support GPU virtualization anyway.
  - Download CUDA Toolkit 8.0 from https://developer.nvidia.com/cuda-downloads (choose Linux -> x86_64 -> Ubuntu -> 16.04 or 14.04 (depending on the version of your Ubuntu))
  - We recommend to download *.deb local.
  - Installation can be done as follows. In the first line below, use the file name of the file you just downloaded.

```
$ sudo dpkg -i cuda-repo-ubuntu1404-8-0-local_8.0.44-1_amd64.deb
$ sudo apt-get update
$ sudo apt-get install cuda
$ echo 'export PATH=/usr/local/cuda/bin:$PATH' >> ~/.bashrc
$ echo ' export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH' >> ~/.bashrc
$ source ~/.bashrc
```

  - Now, download cuDNN from https://developer.nvidia.com/cudnn
    - You need to register first.
    - Unzip the file and move ./cuda/include/cudnn.h to /usr/local/cuda/include and move ./cuda/lib64/* to /usr/local/cuda/lib64 as follows. The names of files and directories below may be different for your case. For example, the file name in the first line below may be different depending on the version of cuDNN you downloaded.

```
$ tar xvzf cudnn-8.0-linux-x64-v5.0-ga.tgz
cuda/include/cudnn.h
cuda/lib64/libcudnn.so
cuda/lib64/libcudnn.so.5
cuda/lib64/libcudnn.so.5.0.5
cuda/lib64/libcudnn_static.a
$ sudo mv ./cuda/include/cudnn.h /usr/local/cuda/include
$ sudo mv ./cuda/lib64/* /usr/local/cuda/lib64
```

- Installing TensorFlow
  - To install TensorFlow without GPU support (do this if you want to install TensorFlow in VirtualBox since VirtualBox does not support GPU virtualization):

```
$ sudo apt-get install python-pip python-dev
$ export TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-
0.11.0rc1-cp27-none-linux_x86_64.whl
$ sudo pip install --upgrade $TF_BINARY_URL
```

  - To install TensorFlow with GPU support:

```
$ sudo apt-get install python-pip python-dev
$ export TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow-
0.11.0rc1-cp27-none-linux_x86_64.whl
$ sudo pip install --upgrade $TF_BINARY_URL
```

  - The above instructions assume you use python 2.7 and 64-bit Ubuntu linux. Refer to https://www.tensorflow.org/versions/r0.11/get_started/os_setup.html#pip-installation for other installation options.

- Installing Theano (not needed for EE488C)

```
$ sudo apt-get install python-numpy python-scipy python-dev python-pip python-nose g++
libopenblas-dev git python-sphinx
$ sudo apt-get install python-matplotlib ipython ipython-notebook python-pandas python-sympy
$ sudo pip install Theano
```

- Installing MXNET (not needed for EE488C)

```
$ cd ~
$ sudo apt-get update
$ sudo apt-get install -y build-essential git libatlas-base-dev libopencv-dev
$ git clone --recursive https://github.com/dmlc/mxnet
$ cd mxnet/make
# edit config.mx file if you want to enable CUDA
$ make -j$(nproc)

# Test using cpu
$ python example/image-classification/train_mnist.py

# Test using gpu
$ python example/image-classification/train_mnist.py --network lenet --gpus 0
```

- Installing Caffe (not needed for EE488C)
  - Before installing Caffe, install some requires packages first:

```
$ sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev libopencv-dev libhdf5-
serial-dev protobuf-compiler
$ sudo apt-get install --no-install-recommends libboost-all-dev
$ sudo apt-get install libatlas-base-dev libgflags-dev libgoogle-glog-dev liblmdb-dev
```

  - Download caffe-master.zip from https://github.com/BVLC/caffe and do the following:

```
$ cd ~
$ unzip caffe-master.zip
$ cd caffe-master
$ cp Makefile.config.example Makefile.config
```

  - Edit ‘Makefile.config’ file as follows:
    - If you want to use GPU & cuDNN, then uncomment the line “# USE_CUDNN := 1”.
    - If you want to use CPU only, then uncomment the line “# CPU _ONLY:= 1”.
  - Then, compile and test Caffe as follows:

```
$ make all
$ make test
$ make runtest
```

  - To compile the Python and MATLAB wrappers do “make pycaffe” and “make matcaffe”, respectively. Be sure to set your MATLAB and Python paths in Makefile.config first.
  - To install models:

```
$ cd ~/caffe-master
$ sudo pip install pyyaml
$ python scripts/download_model_binary.py models/bvlc_reference_caffenet
```