Master ST Computer Science

Deep Learning in Computer Vision

University Bordeaux Faculty S&T academic year 2019 – 2020

Development Environment

During these labs, we will program mainly in Python and use the libraries: Numpy, OpenCV and Keras. More precisely we will use the following versions of the libraries and their dependencies:

- Python 2.7.x
- Numpy 1.12.x or above
- OpenCV 3.2.x or above
- virtualenv
- Matplotlib 2.0.x or above
- tensorflow
- keras 2.2.x

tensorflow will be used as backend of keras and not directly. The version to install depends on whether you want GPU acceleration. Indeed, GPU acceleration of tensorflow may be available if you have a Nvidia graphics card/GPU. It hen requires specific version of CUDA and cuDNN.

1. CREMI

On CREMI machines all software is correctly installed in a **virtualenv** isolated python environment. To activate this virtual environment, you have to do :

source /net/ens/DeepLearning/tensorflow/bin/activate

Your prompt should be changed and indicate that the virtual environment is active: starting by *"(tensorflow)*".

To exit the virtual environment, you can just type: deactivate

Most CREMI machines have a Nvidia GPU. For example, machines in room 203 have a Nvidia RTX2070 GPU. You can see the available machines and their GPUs on this page: https://services.emi.u-bordeaux.fr/exam/?page=wol

On these machines, tensorflow with GPU acceleration is installed. As the installed version of CUDA is 8.0, cuDNN 6.0 and tensorflow 1.4.1 have been installed.

2. Personal Computer

If you want to install the development environment on your personal computer, I would recommend to have a linux distribution, Ubuntu for example. On Mac OS or Windows, it is quite easy to install a linux distribution in a virtual machine (with VirtualBox for example).

If you install a development environment on your personal computer, you should still check that your code is correctly working on CREMI computers. Indeed, your work will be evaluated on CREMI computers.

2.1 GPU acceleration

Both OpenCV and tensorflow can benefit from GPU acceleration for some operations. To have GPU acceleration in OpenCV, you need to install CUDA. To have GPU acceleration in tensorflow, you need to install CUDA and cuDNN

In these labs, we do not use any gpu-accelerated operation from OpenCV. However, for tensorflow, if you have gpu-acceleration, deep networks training will be much faster.

Details instructions how to install CUDA and cuDNN are beyond the scope of this document. Just know that pre-compiled binaries of tensorflow are very picky on required CUDA and cuDNN versions. For example, if you have CUDA 8.0, you should install cuDNN 6.0 and tensorflow-gpu 1.4.x. If you have CUDA 9.0, you should install cuDNN 7.0.x and tensorflow-gpu 1.6.x This page https://www.tensorflow.org/install/source#linux gives tensorflow/cuDNN/CUDA tested build configurations.

2.2 Ubuntu

All Ubuntu version before 18.04 come with a packaged OpenCV version below 3.2. Thus on older Ubuntu distributions, you will have to install OpenCV from sources (see section 2.3).

The procedure to install all software for the lab would be the following, where commands are in bold :

#1) install python2.7 sudo apt install python-dev

#2) install numpy
You can install with the distribution package manager
sudo apt install python-numpy
or you could install with one python package manager (pip, anaconda, ...)
sudo pip install numpy

#3) install opencv 3.2 (from packages) on Ubuntu 18.04
On older Ubuntu versions or for a newer OpenCV version on 18.04, we would have to install it from a PPA or from sources, see 2.3)
sudo apt install libopencv-dev
#and python opencv bindings
sudo apt install python-opencv

#numpy should be installed before OpenCV python bindings.#For all other python libraries, we install them in a virtual environment.#4) install virtualenvsudo apt install virtualenv

#4.1) create a virtualenv named (for example) tensorflow virtualenv --system-site-packages tensorflow

#4.2) active the virtualenv source tensorflow/bin/activate

#5) install matplotlib **pip install matplotlib**

#6) install tensorflow (without GPU acceleration)
pip install tensorflow
#If you have an Nvidia GPU
- if you have installed CUDA 8 and cuDNN 6.0, you could install tensorflow-gpu 1.4.1
pip install tensorflow-gpu=1.4.1
- if you have installed CUDA 9 and cuDNN 7.0, you could install tensorflow-gpu 1.6.0
pip install tensorflow-gpu=1.6.0

#7) install keras [2.2.2] **pip install keras**

Warning : with tensorflow 1.4.x and keras 2.2.x, there is small error in the code (see https://github.com/kerasteam/keras/issues/9621). You should edit the file tensorflow/lib/python2.7/site-packages/keras/backend/tensorflow_backend.py and in the softmax function, line 3146, replace the line : return tf.nn.softmax(x, axis=axis) by :
return tf.nn.softmax(x, axis)

2.3 OpenCV

To have an up-to-date OpenCV version, I would recommend to install OpenCV from sources.

The following commands show how to install OpenCV version 3.4.3. The last version (as of 06/09/2019) is 4.1.1.

#1) Get the latest version of opencv and opencv_contrib source code from OpenCV github repository
https://github.com/opencv/opencv/releases
https://github.com/opencv/opencv_contrib/releases
For example opencv-3.4.3.tar.gz and opencv_contrib-3.4.3.tar.gz

#2) Decompress these source code.tar.bz2
tar xvf opencv-3.4.3.tar.gz
tar xvf opencv_contrib-3.4.3.tar.gz

#3) Compile OpenCV cd opencv-3.4.3 mkdir build

#(the following command is all on the same line)
cmake -DENABLE_FAST_MATH=ON -DBUILD_TESTS=OFF DOPENCV_EXTRA_MODULES_PATH=../../opencv_contrib-3.4.3/modules ..
#If you have a Nvidia GPU and CUDA installed, you can enable GPU acceleration by adding DWITH_CUDA=ON to the previous cmake command
#However, if you have a recent gcc version (gcc 6 or above), it is possible that you will get CUDA compilation
errors.
Here is the command we had to sue on Debian 9 (with gcc 6.3) :
cmake -DENABLE_FAST_MATH=ON -DCUDA_NVCC_FLAGS:STRING="-ccbin clang-3.8 -Wnodeprecated-gpu-targets" -DBUILD_TESTS=OFF
-DOPENCV_EXTRA_MODULES_PATH=../../opencv_contrib-3.4.3/modules ..

make
it may take a while...

sudo make install