Caffe V1 CPU Only on VirtualBox

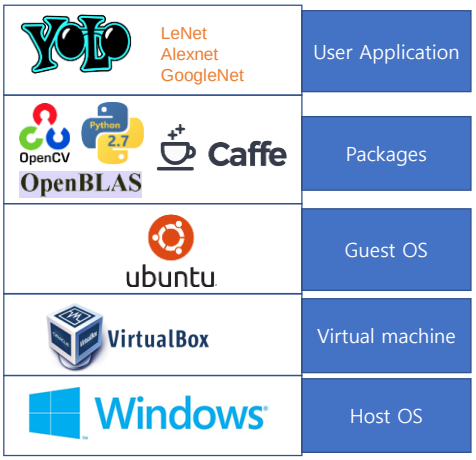
- Convolutional Architecture for Fast Feature Embedding -

2019 - 2020

Ando Ki, Ph.D.
adki@future-ds.com

Contents

Environment



- 1) Windows에 VirtualBox를 설치
  - Installing VirtualBox on Windows host machine.
- 2) VirtualBox에 Ubuntu를 설치
  - Installing Ubuntu guest Operating System on VirtualBox.
- 3) Ubuntu에 필요한 프로그램을 설치
  - Installing user programs and libraries on Ubuntu.
    - GNU GCC
    - OpenBLAS
    - OpenCV
    - Python
    - Caffe

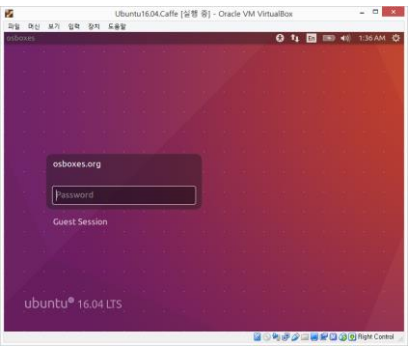
Ubuntu on VirtualBox

- Fresh Ubuntu 16.04 on VirtualBox
  - VirtualBox for Windows
    - 5.2 from https://www.virtualbox.org/wiki/Downloads

      VirtualBox-5.2.16-123759-Win.exe
    - extension pack as well

      Oracle_VM_VirtualBox_Extension_Pack-5.2.16.vbox-extpack
  - Ubuntu image (32-bit or 64-bit, depending on Windows)
    - Ubuntu 16.04.4 Xenial from https://www.osboxes.org/ubuntu/

      Ubuntu_16.04.4-VB-64bit.7z



User name: osboxes
Passwd: osboxes.org

Remove unused packages to make light and update

- Remove LibreOffice
  - *$ sudo apt-get remove --purge libreoffice**
  - *$ sudo apt-get clean*
  - *$ sudo apt-get autoremove*
- Update Ubuntu (it takes time)
  - *$ sudo apt-get update*
  - *$ sudo apt-get upgrade*

Contents

Install prerequisites

- OpenCV
  - $ sudo apt-get install libopencv-dev python-opencv
- OpenBLAS
  - $ sudo apt-get install libopenblas-dev
- Boost
  - $ sudo apt-get install libboost-all-dev
- Python PIP
  - $ sudo apt-get install python-pip python-dev build-essential
  - $ sudo -H pip install --upgrade pip
- Protobuf
  - $ sudo pip install protobuf
- General dependencies
  - $ sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev libopencv-dev libhdf5-serial-dev protobuf-compiler
  - $ sudo apt-get install the python-dev python-skimage cython
  - $ sudo apt-get install libgflags-dev libgoogle-glog-dev liblmdb-dev
- Git
  - $ sudo apt-get install git

Version 2 is recommended

Version 2 is recommended

Get Caffe and modify Makefile.config

- Make your own working directory
  - $ cd
  - $ mkdir caffe_v1 && cd caffe_v1
- Get Caffe
  - $ git clone https://github.com/BVLC/caffe.git
  - $ cd caffe # */home/yourlogin/caffe_v1/caffe*
  - $ cp Makefile.config.example Makefile.config
- Modify Makefile.config
  - Uncommon or set followings
    - CPU_ONLY := 1
    - OPENCV_VERSION := 3 ← *when OpenCV is version 3 (comment out when OpenCV is version 2)*
    - CUSTOM_CXX := g++
    - BLAS := open ← *when OpenBLAS is used*
    - INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include /usr/include/hdf5/serial
    - LIBRARY_DIRS := $(PYTHON_LIB) /usr/local/lib /usr/lib/x86_64-linux-gnu/hdf5/serial
- Install required packages for Python
  - $ cd ~/caffe_v1/caffe/python
  - $ sudo apt-get install python-pip
  - $ sudo pip install -r requirements.txt

```
$ python
>>> import cv2
>>> cv2.__version__
'3.0.0  '
>>> quit()
```

```
$ python
>>> import cv2
>>> cv2.__version__
  '2.4.9.1'
>>> quit()
```

Modify Makefile and compiling and testing

- Compile and testing
  - $ make all
  - following takes time.
    - $ make test
    - $ make runtest
- Major directories
  - data: 데이터가 저장된 폴더
  - examples: 예제 프로그램이 저장된 폴더, i.e., network and solver
  - build: Caffe 실행 파일이 저장된 폴더
- Add following to your bash startup (.bashrc) at the home

Define and export CAFFE_ROOT and CAFFE_HOME

```
export CAFFE_HOME=${HOME}/caffe_v1/caffe
export CAFFE_ROOT=${HOME}/caffe_v1/caffe

if [ -n "${PATH}" ]; then
export PATH=${CAFFE_HOME}/build/tools:${PATH}
else
export PATH=${CAFFE_HOME}/build/tools
fi
```


Caffe command line options

usage: *caffe <command> <args>*

commands:

train	train or finetune a model
test	score a model
device_query	show GPU diagnostic information
time	benchmark model execution time

- '**.caffemodel**' file of shapshot: a output at a specific interval while training; a binary containing the current stat of the weights for each layer of the network.
- '**.solverstate**' file of snapshot: a binary contains the information required to continue training the model from where it last stopped.

Flags from tools/caffe.cpp:

```
  -gpu (Optional; run in GPU mode on given device IDs separated by ','.Use '-gpu all' to run on all available GPUs. The effective
       training batch size is multiplied by the number of devices.) type: string default: ""
  -iterations (The number of iterations to run.) type: int32 default: 50
  -level (Optional; network level.) type: int32 default: 0
  -model (The model definition protocol buffer text file.) type: string default: ""
  -phase (Optional; network phase (TRAIN or TEST). Only used for 'time'.) type: string default: ""
  -sighup_effect (Optional; action to take when a SIGHUP signal is received: snapshot, stop or none.) type: string default:
       "snapshot"
  -sigint_effect (Optional; action to take when a SIGINT signal is received: snapshot, stop or none.) type: string default: "stop"
  -snapshot (Optional; the snapshot solver state to resume training.) type: string default: ""
  -solver (The solver definition protocol buffer text file.) type: string default: ""
  -stage (Optional; network stages (not to be confused with phase), separated by ','.) type: string default: ""
  -weights (Optional; the pretrained weights to initialize finetuning, separated by ','. Cannot be set simultaneously with snapshot.)
       type: string default: ""
```

Python wrapper

- Python wrapper of Caffe
  - $ cd ~/caffe_v1/caffe; make pycaffe
  - Add following to your bash startup (.bashrc) at the home

```
export CAFFE_HOME=${HOME}/caffe_v1/caffe
export CAFFE_ROOT=${HOME}/caffe_v1/caffe

if [ -n "${PATH}" ]; then
export PATH=${CAFFE_HOME}/build/tools:${CAFFE_HOME}/python:${PATH}
else
export PATH=${CAFFE_HOME}/build/tools:${CAFFE_HOME}/python
fi

if [ -n "${PYTHONPATH}" ]; then
export PYTHONPATH=${CAFFE_HOME}/python:${PYTHONPATH}
else
export PYTHONPATH=${CAFFE_HOME}/python
fi
```

  - Testing

```
$ source ~/.bashrc
$ python
>>> import caffe
>>> print caffe.__version__
1.0.0
>>> quit()
```


Contents

- Installing Caffe V1
  - Environment
  - Ubuntu on VirtualBox
  - Remove unused packages to make light and update Ubuntu
- Installing CPU-only Caffe V1
  - Install prerequisites
  - Get Caffe and modify Makefile.config
  - Modify Makefile and compiling and testing
  - Building and testing Caffe
- Installing CPU-only Caffe V1 with Anaconda2
  - Install anaconda2
  - Install prerequisites
  - Get Caffe and modify Makefile.config
  - Modify Makefile and compiling and testing
  - Building and testing Caffe

This is not completed yet.
Many problems occurs.

Install Anaconda2

// To remove Anaconda2
$ conda install anaconda-clean
$ anaconda-clean --yes
// Then remove directories
•~/anaconda2
•~/.anaconda_backup

- Download Anaconda2
  - $ cd; $ mkdie tmp; $ cd tmp
  - $ wget https://repo.anaconda.com/archive/**Anaconda2-2.5.0-Linux-x86_64.sh**
- Run the installer (Installer requires bzip, please install it if you don't have it)
  - $ bash ./Anaconda2-5.1.0-Linux-x86_64.sh –p ${HOME}/anaconda2
- Ensure that your .bashrc is preparing Anaconda, by including these lines
  - ~/.bashrc: export PATH=${HOME}/anaconda2/bin:$PATH

added by Anaconda2 installer (at .bashrc file)
export PATH="/home/usernae/anaconda2/bin:$PATH"
. /home/username/anaconda2/etc/profile.d/conda.sh

Do not use "${HOME}" for "/home/username" where '/home/username' should be your home directory.

- After updating the bashrc, source it to load the new anaconda path or open a new terminal ($ source ~/.bashrc)
- Check conda
  - $ conda –V
    - 3.19.1

Note that latest Python or Anaconda may not work with Caffe due to version mis-match, such as protobuf version.

Prepare Caffe virtual environment

- Make a virtual environment for Caffe v1
  - $ conda create --name ***caffe_v1*** python=2.7
- Ensure that your .bashrc is preparing Anaconda, by including these lines
  - ~/.bashrc: . /${HOME}/anaconda2/etc/profile.d/conda.sh
- As a precaution unset PYTHONPATH to avoid conflicts with packages on your root file system (e.g., /usr/bin/python)
  - $ unset PYTHONPATH
- Activate conda and add packages
  - $ **conda activate caffe_v1**
  - **(caffe_v1) $**
  - **...**
  - **(caffe_v1) $ conda deactivate**

$ conda activate ml-suite
$ conda deactivate

// To check packages in the conda
$ conda list

// To check conda environment
$ conda env list

// To remove conda environment
$ conda-env remove -n ml-suite
or
$ conda remove --name ml-suite --all

Install dependencies

- Activate conda and add packages
  - $ conda activate caffe_v1
  - (caffe_v1) $ sudo apt-get install -y build-essential cmake git pkg-config
  - (caffe_v1) $ sudo apt-get install -y libprotobuf-dev libleveldb-dev libsnappy-dev protobuf-compiler
  - (caffe_v1) $ sudo apt-get install -y libatlas-base-dev
  - (caffe_v1) $ sudo apt-get install -y --no-install-recommends libboost-all-dev
  - (caffe_v1) $ sudo apt-get install -y libgflags-dev libgoogle-glog-dev liblmdb-dev
  - (caffe_v1) $ conda insall -menpo opencv3

Compile Caffe V1

- Make a directory for Caffe V1
  - $ cd; $ mkdir caffe_v1; cd caffe_v1
- Get Caffe V1
  - $ git clone https://github.com/BVLC/caffe.git
    - If you are asked to enter username for something, then check path.
- Install required packages for Python
  - $ cd ~/caffe_v1/caffe/python
  - (caffe_v1) $ sudo apt-get install python-pip
  - (caffe_v1) $ sudo pip install -r requirements.txt
- Modify 'Makefile.config'
  - $ cp Makefile.config.example Makefile.config
  - modify 'Makefile.config' as shown on the right-hand side.

'Makefile.config'
1. Uncomment (No space in the beginning):
CPU_ONLY := 1

2. Uncomment for OpenCV 3
OPENCV_VERSION := 3

3. Uncomment
CUSTOM_CXX := g++

4. Change
BLAS := open

5. Comment out:
PYTHON_INCLUDE := /usr/include/python2.7 ₩
/usr/lib/python2.7/dist-packages/numpy/core/include

6. Uncomment:
ANACONDA_HOME := $(HOME)/anaconda2

PYTHON_INCLUDE := $(ANACONDA_HOME)/include ₩
$(ANACONDA_HOME)/include/python2.7 ₩
$(ANACONDA_HOME)/lib/python2.7/site-packages/numpy/core/include

7. Comment:
PYTHON_LIB := /usr/lib

8. Uncomment:
PYTHON_LIB := $(ANACONDA_HOME)/lib

9. Add path for hdf5
INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include /usr/include/hdf5/serial
LIBRARY_DIRS := $(PYTHON_LIB) /usr/local/lib /usr/lib/x86_64-linux-gnu/hdf5/serial

10. Uncomment:
USE_PKG_CONFIG := 1

Compile Caffe V1

- Add following in CMakeLists.txt
  - set(CMAKE_CXX_STANDARD 11)
  - set(CMAKE_CXX_STANDARD_REQUIRED ON)
  - message(STATUS "C++11 support has been enabled by default.")

- Compile Caffe
  - (caffe_v1) $ mkdir build
  - (caffe_v1) $ cd build
  - (caffe_v1) $ cmake ..
  - (caffe_v1) $ make all

This is not completed yet.
Many problems occurs.

㈜퓨쳐디자인시스템
34051 대전광역시 유성구 문지로 193, KAIST 문지캠퍼스, F723호
(042) 864-0211~0212 / contact@future-ds.com / www.future-ds.com

Future Design Systems, Inc.
Faculty Wing F723, KAIST Munji Campus, 193 Munji-ro, Yuseong-gu, Daejeon 34051, Korea
+82-042-864-0211~0212 / contact@future-ds.com / www.future-ds.com

FUTURE
Design Systems