Caffe

Caffe V1 on Raspberry Pi

- Convolutional Architecture for Fast Feature Embedding -

2019 - 2020

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

Contents

Install dependencies

- $ sudo apt-get -y update && apt-get -y upgrade
- $ sudo apt-get install -y libprotobuf-dev libleveldb-dev\
-     libsnappy-dev libhdf5-serial-dev protobuf-compiler\
-     libgflags-dev libgoogle-glog-dev liblmdb-dev

- It is assumed that OpenCV V3 has been installed.

If something is missing while 'apt-get install', run 'sudo apt-get update' and then run again.

Install Caffe V1 (1/2): download

- $ git clone https://github.com/BVLC/caffe
- $ cd caffe
- $ cp Makefile.config.example Makefile.config
- $ sudo vi Makefile.config

```
# CPU_ONLY := 1
# OPENCV_VERSION := 3
PYTHON_INCLUDE := /usr/include/python2.7 ₩
                /usr/lib/python2.7/dist-packages/numpy/core/include
INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include
LIBRARY_DIRS := $(PYTHON_LIB) /usr/local/lib /usr/lib
```

Depending on your OpenCV

It may not be needed to change.

```
CPU_ONLY := 1
OPENCV_VERSION := 3
PYTHON_INCLUDE := /usr/include/python2.7 \
             /usr/local/lib/python2.7/dist-packages/numpy/core/include
INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include /usr/include/hdf5/serial/
LIBRARY_DIRS := $(PYTHON_LIB) /usr/local/lib /usr/lib /usr/lib/arm-linux-gnueabihf/hdf5/serial/
```

Install Caffe V1 (2/2): compilation

- $ make all
  - It takes about 30 minute
- $ make test
- $ make runtest
- $ sudo vi ~/.bashrc
  - make sure that system search path include following
    - "....../caffe/build/tools"

Major directories
- data: 데이터가 저장된 폴더
- examples: 예제 프로그램이 저장된 폴더, i.e., network and solver
- build: Caffe 실행 파일이 저장된 폴더

```
export CAFFE_HOME=${HOME}/work/...../caffe
export CAFFE_ROOT=${HOME}/work/..../caffe

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

Define and export CAFFE_ROOT and CAFFE_HOME

5

Caffe command line options

- $ /home/pi/work/caffe/build/tools/caffe

```
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

 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: ""
```

- '**.caffemodel**' file of snapshot: 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.

(6)

Install Caffe V1: Python wrapper

- $ cd $HOME/work/caffe
- $ make pycaffe
- $ ./scripts/download_model_binary.py models/bvlc_googlenet
- $ sudo vi ~/.bashrc
  - add following to the bash startup (.bashrc) at the home

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

7

Testing Python wrapper

- Install required packages for Python
  - $ cd ~/caffe-v1-projects/caffe/python
  - $ sudoapt-get install python-pip
  - $ sudopip install -r requirements.txt

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

8

㈜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