Darknet-NNPACK on Raspberry Pi

Aug. 2019

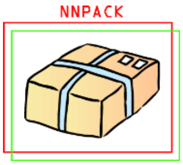
Ando Ki, Ph.D.
adki@future-ds.com

Contents

- What is NNPACK
- Building NNPACK
- Build and Run Darknet with NNPACK
- Gprof profiling
- Run Darknet with OpenCV
- Run Darknet with OpenCV and USB-Cam

What is NNPACK (1/2)

- https://github.com/Maratyszcza/NNPACK
- https://github.com/shizukachan/NNPACK
- https://github.com/digitalbrain79/NNPACK-darknet



- NNPACK is an acceleration package for neural networks on multi-core CPUs
  - Built-in expert-tuned kernels with very high performance:
    - Fast Fourier transform; Winograd transform
    - Matrix-matrix multiplication (GEMM); Matrix-vector multiplication (GEMV)
    - Max-pooling.
  - Multi-threaded SIMD-aware implementations of neural network layers.
  - Implemented in C99 and Python without external dependencies.

What is NNPACK (2/2)

- NNPACK layers
  - Convolutional layer
    - Training-optimized forward propagation (nnp_convolution_output)
    - Training-optimized backward input gradient update (nnp_convolution_input_gradient)
    - Training-optimized backward kernel gradient update (nnp_convolution_kernel_gradient)
    - Inference-optimized forward propagation (nnp_convolution_inference)
  - Fully-connected layer
    - Training-optimized forward propagation (nnp_fully_connected_output)
    - Inference-optimized forward propagation (nnp_fully_connected_inference)
  - Max pooling layer
    - Forward propagation, both for training and inference, (nnp_max_pooling_output)
  - ReLU layer (with parametrized negative slope)
    - Forward propagation, both for training and inference, optionally in-place, (nnp_relu_output)
    - Backward input gradient update (nnp_relu_input_gradient)
  - Softmax layer
    - Forward propagation, both for training and inference, optionally in-place (nnp_softmax_output)

Building NNPACK: prerequisites

- PIP
  - a tool for installing Python packages from the Python Package Index.
- PEACH-Py
  - Portable Efficient Assembly Code-generator in Higher-level Python
  - an assembler embedded in Python
- Confu
  - Ninja-based configuration system
- Ninja
  - a small build system with a focus on speed; e.g., Make
- NNPACK
  - https://github.com/Maratyszcza/NNPACK
  - https://github.com/shizukachan/NNPACK
  - Acceleration package for neural networks on multi-core CPUs

Building NNPACK: prepare for build

- Step 1: install pip if not available
  - $ sudo apt-get install python-pip
- Step 2: Install PeachyPy and Confu
  - $ sudo pip install --upgrade git+https://github.com/Maratyszcza/PeachPy
  - $ sudo pip install --upgrade git+https://github.com/Maratyszcza/confu
- Step 3: Install Ninja
  - $ cd ~/work
  - $ git clone https://github.com/ninja-build/ninja.git
  - $ cd ninja
  - $ git checkout release
  - $ ./configure.py --bootstrap
  - $ export NINJA_PATH=$PWD
  - $ cd

Building NNPAC: NNPACK

- Step 4: Build NNPACK for Darknet (i.e., modified NNPACK for Darknet)

```
$ cd ~/work
$ git clone https://github.com/digitalbrain79/NNPACK-darknet.git
$ cd NNPACK-darknet
$ confu setup
$ python ./configure.py --backend auto
$ $NINJA_PATH/ninja
$ sudo cp -a lib/* /usr/lib/
$ sudo cp include/nnpack.h /usr/include/
$ sudo cp deps/pthreadpool/include/pthreadpool.h /usr/include/
$ cd
```

Build Darknet with NNPACK

- Step 5: Build Darknet

```
$ cd work
$ git clone https://github.com/digitalbrain79/darknet-nnpack.git
$ cd darknet-nnpack
```

  ➲ Have a look at ‘Makefile’

```
$ make
```

Makefile

```
GPU=0
CUDNN=0
OPENCV=0
NNPACK=1
ARM_NEON=1
OPENMP=0
DEBUG=0
....
```

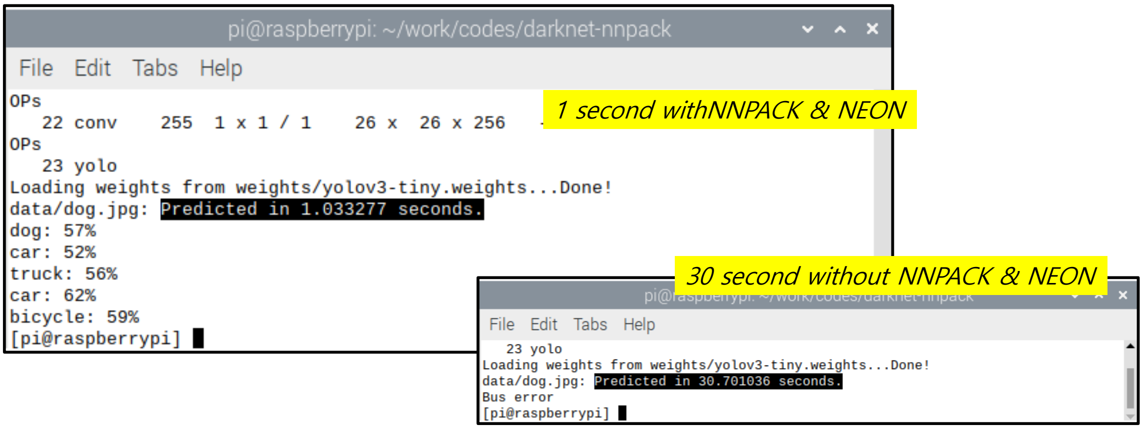
- Step 6: Download weights

```
$ mkdir weights && cd weights
$ wget https://pjreddie.com/media/files/yolov3-tiny.weights
$ cd ..
```

ARM NEON is an advanced SIMD architecture for ARM processors which implement the ARMv7-A or ARMv7-R profile.

Run Darknet with NNPACK

- Step 7: Run Darknet
  - $ ./darknet detect cfg/yolov3-tiny.cfg weights/yolov3-tiny.weights data/dog.jpg


pi@raspberrypi: ~/work/codes/darknet-nnpack
File Edit Tabs Help

```
OPs
   22 conv    255  1 x 1 / 1    26 x  26 x 256  -
OPs
   23 yolo
Loading weights from weights/yolov3-tiny.weights...Done!
data/dog.jpg: Predicted in 1.033277 seconds.
dog: 57%
car: 52%
truck: 56%
car: 62%
bicycle: 59%
[pi@raspberrypi] 
```

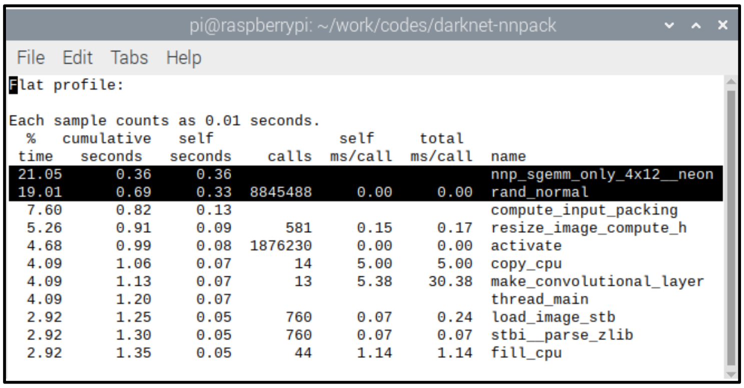
1 second withNNPACK & NEON

30 second without NNPACK & NEON

pi@raspberrypi: ~/work/codes/darknet-nnpack
File Edit Tabs Help

```
   23 yolo
Loading weights from weights/yolov3-tiny.weights...Done!
data/dog.jpg: Predicted in 30.701036 seconds.
Bus error
[pi@raspberrypi] 
```

Gprof profiling



```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  ms/call  ms/call  name
 21.05      0.36     0.36                             nnp_sgemm_only_4x12__neon
 19.01      0.69     0.33  8845488     0.00     0.00  rand_normal
  7.60      0.82     0.13                             compute_input_packing
  5.26      0.91     0.09      581     0.15     0.17  resize_image_compute_h
  4.68      0.99     0.08  1876230     0.00     0.00  activate
  4.09      1.06     0.07       14     5.00     5.00  copy_cpu
  4.09      1.13     0.07       13     5.38    30.38  make_convolutional_layer
  4.09      1.20     0.07                             thread_main
  2.92      1.25     0.05      760     0.07     0.24  load_image_stb
  2.92      1.30     0.05      760     0.07     0.07  stbi__parse_zlib
  2.92      1.35     0.05       44     1.14     1.14  fill_cpu
```

- Compile with '-pg' option

```
ifeq ($(GPROF), 1)
CFLAGS+= -pg
endif
```

- Run as normal
- Run gprof with gmon.out
  - $ gprof darknet gmon.out

Run Darknet with OpenCV

- Step 5: Build Darknet with OpenCV
  - ...
  - $ cd darknet-nnpack
  - Have a look at ‘Makefile’
  - $ make

Makefile

```
GPU=0
CUDNN=0
OPENCV=1
NNPACK=1
ARM_NEON=1
OPENMP=0
DEBUG=0
....
```

- Step 6: Run Darknet
  - $ ./darknet detect cfg/yolov3-tiny.cfg weights/yolov3-tiny.weights data/dog.jpg

Run Darknet with OpenCV and USB-Cam

- Step 5: Build Darknet with OpenCV
  - ...
  - $ cd darknet-nnpack
  - Have a look at ‘Makefile’
  - $ make

Makefile

```
GPU=0
CUDNN=0
OPENCV=1
NNPACK=1
ARM_NEON=1
OPENMP=0
DEBUG=0
....
```

- Step 6: Run Darknet
  - $ ./darknet detector demo cfg/coco.data cfg/yolov3-tiny.cfg weights/yolov3-tiny.weights

㈜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