【國產 IC 開發套件】

型號：HUB 8735

(RTL8735)

離線 AI 模型轉換

(20230317)

指導單位：經濟部工業局

主辦單位：財團法人資訊工業策進會

執行單位：物聯網智造基地

合作單位：振邦科技股份有限公司

目　　錄

圖目錄

一、 需求環境

目前 AI 模型轉換環境需求如下：

Ubuntu 16.04.6

Acuity-Toolkit 6.0.12

Acuity toolkit 或搭配使用 Verisilicon_Tool_VivanteIDE (需要 license)都可以在芯原官方申請 https://verisilicon.com/en/Home

二、 支援成功類型

目前轉換成功且能順利更換 HUB8735 內 AI 模型檔案的類型為：Darknet 訓練出來的 yolov3-tiny, yolov4-tiny, yolov7-tiny。其他類型包含 Tensorflow, TF lite, CAFFE, Onnx 等訓練出來的做轉換都會有一些錯誤需要 acuity toolkit 版本修正或自行 debug 修正。

三、 轉換 AI 模型

先下載 acuity toolkit 並安裝在 Ubuntu 底下。接下來轉換 AI 模型的步驟都是在 Ubuntu 底下 acuity toolkit 目錄下進行。

(一) 導入模型

編輯 0_import_model.sh 如圖 1，使用 pegasus import darknet 的部分

並修改 Name，其餘訓練模式請註解掉。

將 darknet 底下的--model 跟--weights 指定到訓練好的 cfg 檔以及 weights 檔。並將 generate inputmeta 中的--channel-mean-value 修改成" 0 0 0 0.00392156" 。

將訓練好的照片選幾張放到 acuity toolkit/data 目錄中，並將照片路徑寫在 dataset.txt 中。

```
#!/bin/bash


NAME=yolov4
ACUITY_PATH=../bin/

pegasus=${ACUITY_PATH}pegasus
if [ ! -e "$pegasus" ]; then
    pegasus=${ACUITY_PATH}pegasus.py
fi

$pegasus import darknet\
    --model ./model/yolov4-tiny-custom.cfg \
    --weights ./model/yolov4-tiny-custom_last.weights \
    --output-model ${NAME}.json \
    --output-data ${NAME}.data

#generate inpumeta  --source-file dataset.txt
$pegasus generate inputmeta \
        --model ${NAME}.json \
        --input-meta-output ${NAME}_inputmeta.yml \
        --channel-mean-value "0 0 0 0.00392156"  \
        --source-file dataset.txt
```

圖 1、0_import_model 文件內容

資料來源：本計畫整理

開啟 Terminal 並執行 *bash* 0_import_model.sh。

```
I Load model in yolov4.json
I Generate input meta yolov4_inputmeta.yml
I ----------------Error(0),Warning(0)----------------
```

圖 2、導入模型完成

資料來源：本計畫整理

(二) 量化模型

編輯 1_quantize_model.sh，修改 NAME 等參數設定如圖 3。

```
#!/bin/bash

#NAME=mobilenet_tf
NAME=yolov4
ACUITY_PATH=../bin/

pegasus=${ACUITY_PATH}pegasus
if [ ! -e "$pegasus" ]; then
    pegasus=${ACUITY_PATH}pegasus.py
fi

#--quantizer asymmetric_affine --qtype  uint8
#--quantizer dynamic_fixed_point  --qtype int8(int16,note s905d3 not support int16 quantize)
# --quantizer perchannel_symmetric_affine --qtype int8(int16, note only T3(0xBE) can support perchannel quantize)
$pegasus  quantize \
        --quantizer asymmetric_affine \
        --qtype uint8 \
        --rebuild \
        --with-input-meta  ${NAME}_inputmeta.yml \
        --model  ${NAME}.json \
        --model-data  ${NAME}.data
```

圖 3、1_quantize_model.sh 文件內容

資料來源：本計畫整理

開啟 Terminal 並執行 *bash* 1_quantize_model.sh。

```
I End quantization...
I Dump net quantize tensor table to yolov4.quantize
I Save net to yolov4.data
I ----------------Error(0),Warning(0)----------------
```

圖 4、量化模型成功

資料來源：本計畫整理

(三) 生成 Binary

編輯 2_export_case_code.sh，修改 NAME 等參數設定如圖 5。

```
#!/bin/bash

#NAME=mobilenet_tf
NAME=yolov4
ACUITY_PATH=../bin/

pegasus=$ACUITY_PATH/pegasus
if [ ! -e "$pegasus" ]; then
    pegasus=$ACUITY_PATH/pegasus.py
fi

$pegasus export ovxlib\
    --model ${NAME}.json \
    --model-data ${NAME}.data \
    --model-quantize ${NAME}.quantize \
    --with-input-meta ${NAME}_inputmeta.yml \
    --dtype quantized \
    --optimize VIPNANOQI_PID0XAD  \
    --viv-sdk ${ACUITY_PATH}vcmdtools \
    --pack-nbg-viplite

rm -rf ${NAME}_nbg_viplite

mv ../*_nbg_viplite ${NAME}_nbg_viplite

cd ${NAME}_nbg_viplite

mv network_binary.nb ${NAME}.nb

cd ..

#save normal case demo export.data
mkdir -p ${NAME}_normal_case_demo
mv  *.h *.c .project .cproject *.vcxproj BUILD *.linux *.export.data ${NAME}_normal_case_demo

# delete normal_case demo source
#rm  *.h *.c .project .cproject *.vcxproj  BUILD *.linux *.export.data

rm *.data *.quantize *.json *_inputmeta.yml
```

圖 5、2_export_case_code.sh 文件

資料來源：本計畫整理

其中--optimize 請改成”VIPNANOQI_PID0XAD”，檔案路徑在 acuity toolkit/bin 底下，若該目錄內沒有此檔案，請自行新增並存成 VIPNANOQI_PID0XAD，檔案內容如圖 6。

```
[device]
target = VIP8000NANONI_PID0XAD
pid = 0XAD
optimization = True
core = 8
mad_per_core = 64
input_buffer_depth = 12
float16_support = False
dynamic_fixed_point-8_support = True
dynamic_fixed_point-16_support = True
asymmetric_quantized-u8_support = True
evis = 2
```

圖 6、VIPNANOQI_PID0XAD 文件

開啟 Terminal 並執行 *bash* 2_export_case_code.sh。

如果步驟(一)、(二)、(三)執行都沒錯誤，之後可以直接執行

bash 0_import_model.sh && *bash* 1_quantize_model.sh && **bash** 2_export_case_code.sh

最後可以在 xxx_nbg_viplite 底下找到轉換完成的 xxx.nb 檔。

四、 模型套用

如何將轉換好的檔案套用到 HUB8735 Arduino 上，可參考官方文件

https://www.amebaiot.com/zh/amebapro2-apply-ai-model-docs/