# 君正®

# Magik 开发平台

## 后量化使用指南

Date: Jul. 2022

北京君正集成电路股份有限公司
Ingenic Semiconductor Co., Ltd.

# 君正®

## 深度神经网络开放平台

### 后量化使用指南



**Release history**

| Date | Author | Revision | Change |
|---|---|---|---|
| Feb. 2022 | czhang | 1.0.1 | First release |
| Jul. 2022 | czhang | 2.0.0 | Update config,debug and support multi-input net |

## Disclaimer

This documentation is provided for use with Ingenic products. No license to Ingenic property rightsis granted. Ingenic assumes no liability, provides no warranty either expressed or implied relatingto the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining orsupporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes tothis document without notice. Anyone relying on this documentation should contact Ingenic forthe current documentation and errata.

**Ingenic Semiconductor Co., Ltd.**

**Ingenic Headquarters, East Bldg. 14, Courtyard #10**
**Xibeiwang East Road, Haidian District, Beijing, China,**
**Tel: 86-10-56345000**
**Fax:86-10-56345001**
**Http: //www.ingenic.com**

目录

# 1 MagikToolKit 简介

MagikTransformKit 工具支持对 onnx/tensorflow/mxnet/PyTorch/Caffe 格式模型进行转换后量化工作得到 magik 格式模型，该模型可在芯片端进行部署工作。

文档介绍转换量化过程 config 使用，单输入网络、多输入网络后量化 demo，上板流程。

# 2 MagikToolKit 使用

## 2.1 Config 介绍

MagikToolKit 工具通过解析 config 文件中信息，完成 tensorflow、onnx 格式源模型转换量化工作，生成板端部署的 bin 模型。

Config 文件可通过 MagikToolKit 工具传入参数”-gcfg”或”--gen_config”完成模板 config 文件生成。

```
Magik TransformKit$./magik-transform-tools -gcfg
INFO(magik): magik-transform-tools version:1.0.0(00010000_2923907)  cuda version:9.0.176  built:20220713-0924_GPU
2022-07-20 10:42:39.991508: I magik_config.cc:74-dump_config_json] Succeedly Generate Model Config File :./magik.cfg
```

工具传参生成模板 config 文件

```
{
    "SOC":"",
    "INPUT":[],
    "INPUT_SHAPE":[],
    "MEAN":[],
    "NORMAL":[],
    "COLOR":[],
    "OUTPUT":[],
    "QUANT_DATASET_PATH":"./path/to/"
}
```

后量化解析的配置文件

**SOC --** 部署芯片型号

**INPUT --** 指定模型的输入，可进行截取模型工作，不指定时使用模型中输入信息

**INPUT_SHAPE --** 指定网络输入形状信息，不指定输入形状信息情况：

1）输入形状可以从原生模型中获取；

2）一般情况输入形状信息和输入信息一一对应，如果形状信息长度小于输入信息长度，使用最后一个形状信息进行对输入信息进行广播；

**MEAN --** 指定模型输入的均值信息，默认值 0，支持通道级别广播、输入节点级别广播

**NORMAL --** 指定模型输入的方差信息，默认值 1，支持通道级别广播、输入节点级别广播

**COLOR --** 模型输入源是图片，指定图片处理方式”BRG”/”RGB”/”GRAY”，默认”RGB”，支持输入级别广播

**OUTPUT --** 指定模型的输出，可进行截取模型工作，不指定时使用模型中输出信息。

QUANT_DATASET_PATH -- 模型后量化校准数据集的路径，使用路径下所有图片（或 bin 文件）进行量化操作；选项默认为””，不设置有效路径时不进行后量化。

## 2.2 后量化 Demo

MagikToolKit 工具通过解析 config 文件可同时支持单输入网络、多输入网络后量化转换工作，生成上板的 bin 文件模型。单输入网络、多输入网络流程演示分为两个 demo 进行。

### 2.2.1 单输入网络后量化

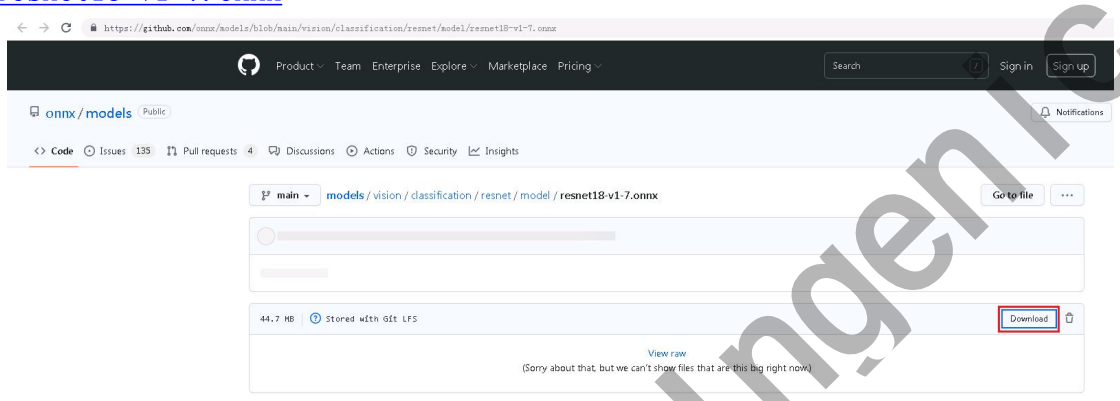
单输入网络后量化 demo 使用 onnx 官方提供的 resnet18 网络，demo 路径信息：

***your_root/magik-toolkit/Models/post/resnet18***

#### 1）准备模型和校准集合

当前使用的模型从 github 下载，链接：

https://github.com/onnx/models/blob/main/vision/classification/resnet/model/resnet18-v1-7.onnx




模型下载界面

后量化使用的校准数据集在 Val_2012_Images_part 目录下，这些量化图片从 ImageNet 图片测试集抽取

```
Magik resnet18$ls -R
.:
cfg         run_a1.sh   run_t41.sh    Val_2012_Images_part
resnet18-v1-7.onnx  run_t40.sh  run_x2500.sh  venus_sample_resnet18

./cfg:
magik_a1.cfg  magik_t40.cfg  magik_t41.cfg  magik_x2500.cfg

./Val_2012_Images_part:
ILSVRC2012_val_00000001.JPEG  ILSVRC2012_val_00000011.JPEG
ILSVRC2012_val_00000002.JPEG  ILSVRC2012_val_00000012.JPEG
```

量化的模型和量化图片

#### 2）配置文件

在 config 文件完成量化信息配置工作，可以通过 netron 工具查看原生模型结构确认模型输入输出、输入形状信息；根据训练代码对图片处理确认模型的前处理信息。

```
{
    "SOC":"T40",
    "INPUT":["data"],
    "INPUT_SHAPE":[1,3,224,224],
    "MEAN":[123.675,116.28,103.53],
    "NORMAL":[58.395,57.12,57.375],
    "COLOR":["RGB"],
    "OUTPUT":["resnetv15_dense0_fwd"],
    "QUANT_DATASET_PATH":"./Val_2012_Images_part"
}
```

单输入网络配置文件

**SOC --** 即将部署在 T40 芯片上

**INPUT --** 指定模型输入节点信息

**INPUT_SHAPE --** 指定模型输入形状
**MEAN --** 模型训练过程图片前处理过程的方差
**NORMAL --** 模型训练过程图片前处理过程的方差
**COLOR --** 模型输入图片处理成 RGB 格式
**OUTPUT --** 指定模型输出节点信息
**QUANT_DATASET_PATH --** 指定量化图片的路径

### 3）编写转换命令

```
../../../TransformKit/magik-transform-tools \
-inputpath resnet18-v1-7.onnx \
-outputpath ./venus_sample_resnet18/resnet18_t40_magik.mk.h \
-cfg ./cfg/magik_t40.cfg \
--save_quantize_model true
```

转换命令使用

```
../../../TransformKit/magik-transform-tools \
-i resnet18-v1-7.onnx \
-o ./venus_sample_resnet18/resnet18_t40_magik.mk.h \
-cfg ./cfg/magik_t40.cfg \
--save_quantize_model true
```

简洁转换命令使用

**outputpath --** 上板文件生成路径和命名，后缀要求".mk.h"
**inputpath --** 源模型的路径地址
**config --** 配置文件路径地址

### 4）转换模型

```
*********************************************************************************************
*                        ^_^ Convert successfully, Enjoy it ^_^                             *
*********************************************************************************************
```

转换成功提示

```
Magik resnet18$ls -R
.:
cfg         run_a1.sh   run_t41.sh    Val_2012_Images_part
resnet18-v1-7.onnx  run_t40.sh  run_x2500.sh  venus_sample_resnet18

./venus_sample_resnet18:
ILSVRC2012_val_00000001.JPEG         Makefile        resnet18_t40_magik.bin
inference.cpp               makefile_files  save-magik
magik_model_resnet18_t40_magik.mk.h  readme.md       stb
```

生成上板文件

## 2.2.2 多输入网络后量化

多输入网络后量化 demo 使用 gru 网络，demo 路径：
***your_root/magik-toolkit/Models/post/gru***

### 1）准备模型和校准数据集

gru 网络在 ***your_root/magik-toolkit/Models/post/gru*** 路径下，无需下载。

多输入网络对量化数据集要求两级目录：

一级目录是量化数据集的包装，当前目录表示网络后量化可以使用数据集的数量

二级目录下是真实喂入模型的输入，要求文件命名前缀是网络输入名，后缀是.bin，网络输入和 bin 文件存在一一对应关系

```
Magik gru$ls -R
.:
cfg     quant_data  run_t40.sh  run_x2500.sh
model.onnx  run_a1.sh   run_t41.sh  venus_sample_gru

./cfg:
magik_a1.cfg  magik_t40.cfg  magik_t41.cfg  magik_x2500.cfg

./quant_data:
0  1  2

./quant_data/0:
740_input.bin  last_state_tgru.bin

./quant_data/1:
740_input.bin  last_state_tgru.bin
```

多输入网络校准目录二级结构

**2）配置文件**

```
{
    "SOC":"T40",
    "INPUT":["740"],
    "INPUT_SHAPE":[[1,3,1,256]],
    "MEAN":[],
    "NORMAL":[],
    "COLOR":[],
    "OUTPUT":["580"],
    "QUANT_DATASET_PATH":"./quant_data",
    "DEBUG_PATH":"./quant_data"
}
```

多输入网络配置文件

网络有两个输入，因为只需要截取一个输入，INPUT 只有一个输入同时指定输入的形状信息；另外一个输入和形状可从模型中获取，也可以指定

```
{
    "SOC":"T40",
    "INPUT":["740","last_state_tgru"],
    "INPUT_SHAPE":[[1,3,1,256],[1,16,128]],
    "MEAN":[],
    "NORMAL":[],
    "COLOR":[],
    "OUTPUT":["580"],
    "QUANT_DATASET_PATH":"./quant_data"
}
```

添加缺省的输入信息

网络单输入指定输入形状使用[shape1,...]，网络多输入时指定网络输入形状需要以[[shape1,...],[shape2,...]]嵌套中括号进行。

**3）编写转换命令**

```
../../../TransformKit/magik-transform-tools \
--outputpath ./venus_sample_gru/gru_t40_magik.mk.h \
--inputpath ./model.onnx \
--config ./cfg/magik_t40.cfg
```

转换命令用法

**4）转换模型**

```
Magik gru$ls -R
.:
cfg     quant_data  run_t40.sh  run_x2500.sh
model.onnx  run_a1.sh   run_t41.sh  venus_sample_gru

./venus_sample_gru:
0          inference.cpp          Makefile    readme.md
gru_t40_magik.bin  magik_model_gru_t40_magik.mk.h  makefile_files

./venus_sample_gru/0:
740_input.bin  last_state_tgru.bin
```

生成上板文件

# 3 精度分析

## 3.1 功能介绍

MagiToolKit 工具通过 config 文件配置选项进行调试后量化模型工作，定位问题，修复后量化模型精度差问题。

## 3.2 Config 配置

```
"QUANT_FEATURE_BIT": 8,
"QUANT_WEIGHT_BIT": 8,
"QUANT_FEATURE_METHOD":"KL",
"QUANT_WEIGHT_METHOD":"MAX_ABS",
"QUANT_DATASET_SIZE": 10,
"QUANT_NODE_SKIP":[],
"FEATURE_CORRECT_PATH":"",
"WEIGHT_CORRECT_PATH":"",
"WEIGHT_FAKE_QUANT": false,
"FEATURE_FAKE_QUANT": false,
"SAVE_OP_QUANT_BIT": false,
"LOAD_OP_QUANT_BIT": false,
"LOAD_OP_QUANT_MIN_MAX": false
```

Debug 选项

**DEUBG_PATH --** 是一个综合选项，默认”“，选项处于关闭状态。当指定有效路径（路径内指定 n 张图片或 bin 文件）时，打开转换工具的 debug 模式。

```
"DEBUG_PATH":"",
```

设置量化集路径

1）dump 模型量化前后仿真结果（量化前仿真结果和原生模型推理相同、量化后仿真结果和板端推理结果相同）,在运行路径下生成目录”magik_dump_data”存放仿真结果；

```
Magik resnet18$ls -R
.:
cfg             run_a1.sh    run_x2500.sh
magik_dump_path     run_t40.sh  Val_2012_Images_part
resnet18-v1-7.onnx  run_t41.sh  venus_sample_resnet18

./cfg:
magik_a1.cfg  magik_t40.cfg  magik_t41.cfg  magik_x2500.cfg

./magik_dump_path:
float_model  input_uint8.bin       ptq_model
input.bin    model_node_info.json  ptq_model_dequantize

./magik_dump_path/float_model:
```

DEBUG_PATH 生成仿真结果

**model_node_info.json --** 模型中算子类型信息

**input.bin --** dump 转换工具进行仿真推理的输入，可以作为原生模型推理输入或者上板推理输入

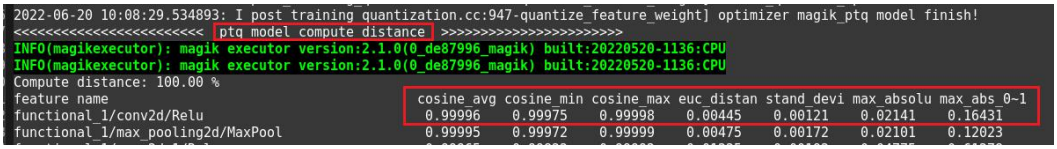
**float_model** -- 原生模型仿真推理结果

**ptq_model --** 后量化模型仿真推理结果

**ptq_model_dequantize --** 后量化模型仿真推理结果进行反量化

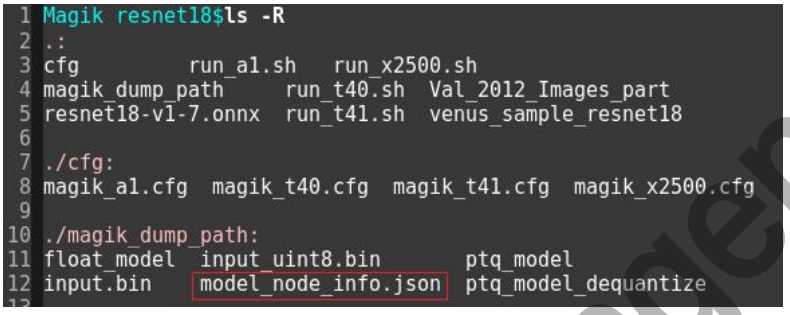
2）计算模型量化前后仿真结果余弦相似度。为了得到客观真实的量化效果，DEBUG_PATH 每一张图片（bin 文件）推理得到一组余弦相似度。在多组余弦相似度中，计算每个节点余弦相似度平均值、

最小值、最大值、欧式距离、标准差、最大差、归一化后最大差。




余弦相似度计算结果

3）生成模型中待量化算子信息 config 文件，包含工具对模型中支持量化的算子类型（all_type 选项），和每种量化类型算子包含的具体算子名（Conv2D 选项、MatMul 选项等）




转换工具运行目录下


Resnet18 网络量化算子信息

```
"QUANT_FEATURE_BIT": 8,
"QUANT_WEIGHT_BIT": 8,
```

量化位宽设置

**QUANT_FEATURE_BIT --** 指定激活量化位宽，默认 8 比特量化，当默认的激活量化位宽无法保证模型精度，可提升激活量化位宽，当前后量化算法最高支持激活进行 12 比特量化。

**QUANT_WEIGHT_BIT --** 指定权重量化位宽，默认 8 比特量化，当前的后量化算法最高支持权重进行 8 比特量化。

```
"QUANT_FEATURE_METHOD":"KL",
"QUANT_WEIGHT_METHOD":"MAX_ABS",
```

量化方法设置

**QUANT_FEATURE_METHOD --** 激活的量化方法，默认使用”MAX_ABS”方法进行量化，当前激活支持{KL, ADMM, MSE}方法量化。

**QUANT_WEIGHT_METHOD --** 权重的量化方法，默认使用”KL”方法进行量化，当前权重支持{MAX_ABS, ADMM}方法量化。

```
"QUANT_DATASET_SIZE": 40,
```

量化集数量设置

**QUANT_DATASET_SIZE --** 指定量化过程使用校准数据集（图片或 bin 文件）的数量，默认括号内为空，表示不指定任何算子跳过量化。

```
"QUANT_NODE_SKIP":[
    "NODE_NAME0",
    "NODE_NAME0"
],
"QUANT_OP_SKIP":[
    "OP_TYPE0",
    "OP_TYPE1"
],
```

指定算子跳过量化

**QUANT_NODE_SKIP --** 跳过指定算子名量化，调试该层算子是否导致模型精度损失，默认括号内为空不指定跳过量化。

**QUANT_OP_SKIP --** 跳过指定算子类型量化，调试该类型算子是否导致模型精度损失，默认括号内为空不指定跳过量化。

```
"WEIGHT_FAKE_QUANT": false,
"FEATURE_FAKE_QUANT": false,
```

模型伪量化设置

**FEATURE_FAKE_QUANT --** 仅对模型激活进行伪量化操作，确认是否激活量化导致模型精度丢失，选项默认为关。

**WEIGHT_FAKE_QUANT --** 仅对模型权重进行伪量化操作，确认是否权重量化导致模型精度丢失，选项默认为关。

```
"WEIGHT_CORRECT_PATH":"",
```

模型权重量化校准设置

**WEIGHT_CORRECT_PATH --** 默认为空，不进行校正工作；输入校准的有效路径时，使用路径下所有图片（或 bin 文件）对权重量化损失进行校正工作，能够有限改善权重量化带来损失。

```
"LOAD_OP_QUANT_BIT":false,
"SAVE_OP_QUANT_BIT":false
```

生成和加载算子量化信息设置

**SAVE_OP_QUANT_BIT --** 保存激活量化位宽信息为 config 文件，编辑文件指定激活的量化位宽。

**LOAD_OP_QUANT_BIT --** 加载 SAVE_OP_QUANT_BIT 选项生成的 config 文件，对量化算子的激活进行指定位宽的量化，指定量化位宽不超过 12 比特，优先级高于 QUANT_FEATURE_BIT 选项。

```
Magik resnet18$ls
cfg                          run_a1.sh      Val_2012_Images_part
magik_dump_path              run_t40.sh     venus_sample_resnet18
magik_node_quant_bit_info.json  run_t41.sh
resnet18-v1-7.onnx           run_x2500.sh
```

运行路径下生成 magik_node_quant_bit_info.json

## 3.3 使用流程

网络进行上板部署效果不理想时，或者用户对量化前后的余弦相似度有要求时，在 config 文件添加 DEBUG_PATH 选项，指定有效的 Debug 路径，定位具体问题。

基于上述介绍完成 resnet18 网络进行调试工作

```
"DEBUG_PATH":"./Val_2012_Images_part"
```

设置目录调试

```
"DEBUG_PATH":"./Val_2012_Images_part/ILSVRC2012_val_00000001.JPEG"
```

设置图片调试

```
magik_ptq model finish!
<<<<<<<<<<<<<<<<<<<<<<<< ptq model compute distance >>>>>>>>>>>>>>>>>>>>>>>>
INFO(magikexecutor): magik executor version:2.1.0(0_e011a9b_magik) built:20220629-1945:CPU
INFO(magikexecutor): magik executor version:2.1.0(0_e011a9b_magik) built:20220629-1945:CPU
Compute distance: 100.00 %
feature name                                    cosine_avg cosine_min cosine_max euc_distan stand
 devi max_absolu max_abs_0~1
data/BatchNormScale                              0.99813    0.99813    0.99813    0.09554    0.08
244    0.14382    0.40347
resnetv15_relu0_fwd                              0.99078    0.99078    0.99078    0.09715    0.03
879    0.36473    1.83665
resnetv15_pool0_fwd                              0.99310    0.99310    0.99310    0.09006    0.04
120    0.31800    1.34751
resnetv15_stage1_relu0_fwd                       0.96739    0.96739    0.96739    0.18047    0.04
435    0.36860    3.04702
resnetv15_stage1_batchnorm1_fwd                  0.89750    0.89750    0.89750    0.22711    0.16
008    1.40185    3.95415
resnetv15_stage1_activation0                     0.94775    0.94775    0.94775    0.24293    0.15
166    1.15034    3.58933
resnetv15_stage1_relu1_fwd                       0.87179    0.87179    0.87179    0.30926    0.06
702    0.95211    8.48043
resnetv15_stage1_batchnorm3_fwd                  0.78378    0.78378    0.78378    0.32996    0.25
589    2.41494    5.94339
resnetv15_stage1_activation1                     0.88374    0.88374    0.88374    0.36573    0.28
741    2.69360    6.55498
```

模型余弦相似度差的情况

1）平均相似余弦相似低于 98%

原因一：激活量化位宽不够

```
"FEATURE_FAKE_QUANT": true,
```

使用 FEATURE_FAKE_QUANT 选项转换，确认是否激活量化余弦相似度是否较差

```
CollectFeatureDistribution: 100.00 %
<<<<<<<<<<<<<<<<<<<<<<<< feature fake quant compute distance >>>>>>>>>>>>>>>>>>>>>>>>
INFO(magikexecutor): magik executor version:2.1.0(0_e011a9b_magik) built:20220629-1945:CPU
INFO(magikexecutor): magik executor version:2.1.0(0_e011a9b_magik) built:20220629-1945:CPU
Compute distance: 100.00 %
feature name                                    cosine_avg cosine_min cosine_max euc_distan
 devi max_absolu max_abs_0~1
data/BatchNormScale                              1.00000    1.00000    1.00000    0.00000
000    0.00000    0.00000
resnetv15_relu0_fwd                              0.99603    0.98908    0.99768    0.06089
724    0.32499    1.45408
resnetv15_pool0_fwd                              0.99411    0.98747    0.99635    0.08119
358    0.49372    1.82746
resnetv15_stage1_relu0_fwd                       0.97882    0.97038    0.98819    0.13945
676    0.30361    2.33458
resnetv15_stage1_batchnorm1_fwd                  0.95833    0.92273    0.98343    0.14171
054    1.15185    2.93956
resnetv15_stage1_activation0                     0.97891    0.95662    0.99170    0.15207
583    0.92095    2.63010
resnetv15_stage1_relu1_fwd                       0.92205    0.86979    0.96418    0.23283
277    0.64575    5.61058
resnetv15_stage1_batchnorm3_fwd                  0.87873    0.78425    0.94458    0.24272
502    2.04129    5.03272
```

如果是，使用 QUANT_FEATURE_BITS 选项，提升激活量化位宽至 10,12；

```
"QUANT_FEATURE_BIT": 10,
```

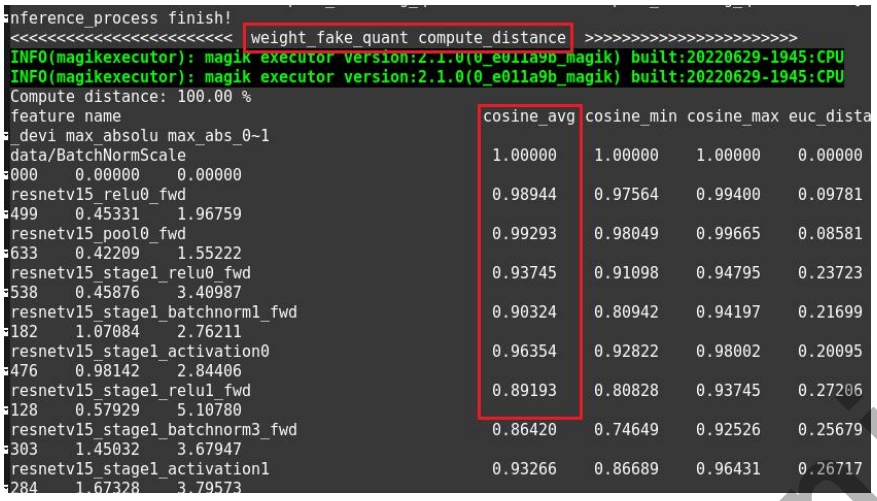
提升激活量化位宽

再次转换，检查激活量化余弦相似是否改善，检查整体量化余弦相似度是否改善

原因二：weight 量化位宽不够

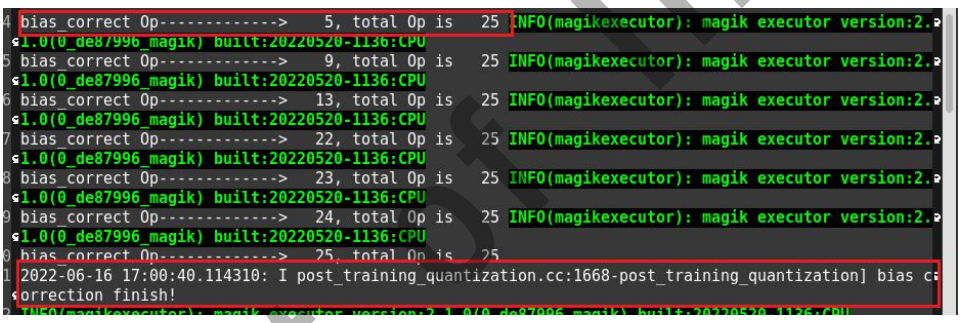
```
"WEIGHT_FAKE_QUANT": true
```

使用 WEIGHT_FAKE_QUANT 选项转换，确认是否权重量化余弦相似度是否较差；




如果是，使用 WEIGHT_COREECT_PATH 选项进行校正 WEIGHT 量化带来的精度损失，权重矫正路径尽量保证和量化校准、debug 集合不重复


设置矫正集路径信息




权重矫正过程

再次转换，检查权重量化余弦相似是否改善，检查整体量化余弦相似度是否改善；

原因三：算子量化问题


DEBUG_PATH 选项生成模型待量化信息

1）QUANT_OP_SKIP 选项结合 all_type 的内容，跳过指定类型算子量化，使用 2 分法定位哪一个类算子量化存在问题；

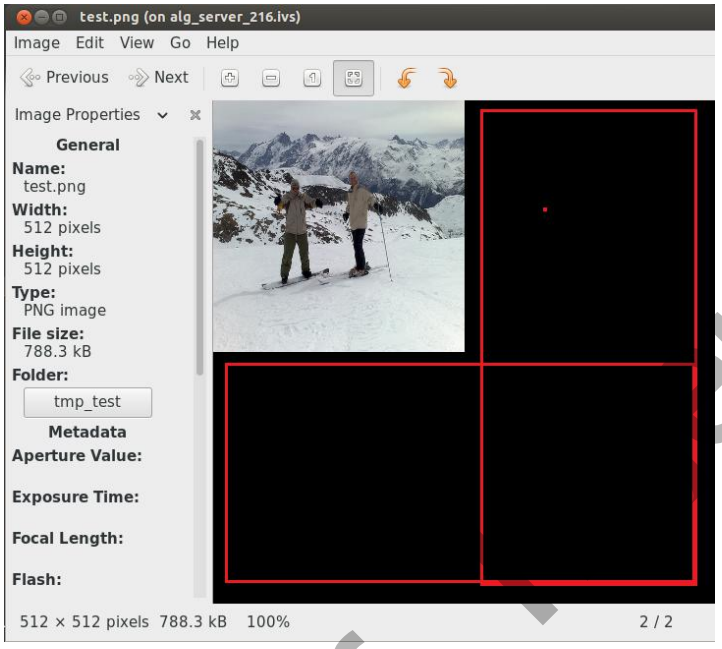
2）基于 QUANT_OP_SKIP 选项定位结果，可使用 QUANT_NODE_SKIP 选项结合跳过指定算子名量化，使用 2 分法定位该类算子存在问题的具体节点

定位具体算子反馈相关对接人员，定位问题

原因四：挑选合适图片和适量图片

检查量化图片数量是否较少，建议 10~200 图片进行量化

在真实量化过程存在量化图片存在大量无效部分




图片存在大量填充值

在实现过程中，量化图片的选择对量化精度损失有一定，避免选择图片中存在大量黑背景、灰背景等。

2）平均余弦度都是 99%，上板部署精度丢失

原因一：板端推理问题

板端推理结果和后量化模型仿真结果无法对齐，使用 DEBUG_PATH 选项生成 input.bin 结果在板端推理，和 ptq_model 结果进行比对工作：

1）input.bin 以 NHWC 格式、float32 格式存放；

2）ptq_model 目录存放后量化模型推理结果，结果以 NHWC 格式、uint8 格式存放（激活进行 10、12 比特量化时结果以 uint16 格式存放）

```
./magik_dump_path:
float_model  input_uint8.bin       ptq_model
input.bin    model_node_info.json  ptq_model_dequantize
```

后量化模型仿真结果

原因二：原生模型仿真问题

转换工具原生模型仿真结果和原生模型推理无法对齐，使用 DEBUG_PATH 选项生成 input.bin 结果作为原生模型输入进行推理，和 float_model 结果进行比对工作：

1）input.bin 以 NHWC 格式、float32 格式存放，在原生模型推理需要对输入进行均值方差处理，仿真推理时已将均值方差合并在模型中；

2）float_model 目录下存放的原生模型仿真推理结果，结果以 NHWC 格式、float 格式存放

```
./magik_dump_path:
float_model  input_uint8.bin       ptq_model
input.bin    model_node_info.json  ptq_model_dequantize

./magik_dump_path/float_model:
data_BatchNormScale.bin          resnetv15_stage2_relu0_fwd.bin
flatten_170.bin                  resnetv15_stage2_relu1_fwd.bin
resnetv15_dense0_fwd.bin         resnetv15_stage3_activation0.bin
resnetv15_pool0_fwd.bin          resnetv15_stage3_activation1.bin
resnetv15_pool1_fwd.bin          resnetv15_stage3_batchnorm1_fwd.bin
resnetv15_relu0_fwd.bin          resnetv15_stage3_batchnorm2_fwd.bin
resnetv15_stage1_activation0.bin     resnetv15_stage3_batchnorm4_fwd.bin
```

原生模型推理仿真

模型仿真推理出错需反馈相关对接人员，修复问题。

# 4 混合位宽

在调试后量化模型效果过程中，激活量化效果不理想，添加 FEATURE_QUANT_BITS 选项提升量化位宽改善量化效果，但是提升量化位宽导致量化模型推理效率变差

为了改善量化效果的同时保证推理效率，我们提出混合位宽量化方法，部分算子进行 8 比特量化，部分算子进行高比特量化（10、12 比特）。

```
"LOAD_OP_QUANT_BIT":false,
"SAVE_OP_QUANT_BIT":false
```

混合位宽选项

基于 resnet18 网络后量化使用激活的混合量化

1）设置 SAVE_OP_QUANT_BIT 选项为 true，使用工具转换网络，保存当前模型节点的量化信息在 magik_node_quant_bit_info.json 文件

```
{
    "SOC":"T40",
    "INPUT":[],
    "INPUT_SHAPE":[],
    "MEAN":[123.675,116.28,103.53],
    "NORMAL":[58.395,57.12,57.375],
    "COLOR":["RGB"],
    "OUTPUT":[],
    "QUANT_DATASET_PATH":"./Val_2012_Images_part",
    "DEBUG_PATH":"./Val_2012_Images_part",

    "QUANT_FEATURE_BIT":8,
    "SAVE_OP_QUANT_BIT": true,
    "LOAD_OP_QUANT_BIT": false
}
```

设置 LOAD_OP_QUANT_BIT

```
2022-07-03 16:41:48.224876: I post_training_quantization.cc:1716-post_training_quantization] optimizer magik_float model finish!
2022-07-03 16:41:51.129777: I post_training_quantization.cc:1720-post_training_quantization] shape inference process finish!
2022-07-03 16:41:51.573926: I post_training_quantization.cc:437-dump_quant_bit_info] model quant info save in ./magik_node_quant_bit_info.json
```

提示信息

```
Magik resnet18$ls
cfg                        run_a1.sh      Val_2012_Images_part
magik_dump_path            run_t40.sh     venus_sample_resnet18
magik_node_quant_bit_info.json  run_t41.sh
resnet18-v1-7.onnx         run_x2500.sh
```

模型量化信息文件

2）编辑混合量化 magik_node_quant_bit_info.json 文件

```
{
    "data/BatchNormScale":  {
        "QUANT_FEATURE_BIT":    8
    },
    "resnetv15_relu0_fwd":  {
        "QUANT_FEATURE_BIT":    8
    },
    "resnetv15_pool0_fwd":  {
        "QUANT_FEATURE_BIT":    8
    },
    "resnetv15_stage1_relu0_fwd":   {
        "QUANT_FEATURE_BIT":    8
    },
    "resnetv15_stage1_batchnorm1_fwd":  {
        "QUANT_FEATURE_BIT":    8
    },
    "resnetv15_stage1_activation0": {
        "QUANT_FEATURE_BIT":    8
    },
    "resnetv15_stage1_relu1_fwd":   {
        "QUANT_FEATURE_BIT":    8
    },
    "resnetv15_stage1_batchnorm3_fwd":  {
        "QUANT_FEATURE_BIT":    8
    },
```

编辑前文件

```
{
    "data/BatchNormScale":  {
        "QUANT_FEATURE_BIT":    8
    },
    "resnetv15_relu0_fwd":  {
        "QUANT_FEATURE_BIT":    8
    },
    "resnetv15_pool0_fwd":  {
        "QUANT_FEATURE_BIT":    8
    },
    "resnetv15_stage1_relu0_fwd":   {
        "QUANT_FEATURE_BIT":    4
    },
    "resnetv15_stage1_batchnorm1_fwd":  {
        "QUANT_FEATURE_BIT":    8
    },
    "resnetv15_stage1_activation0": {
        "QUANT_FEATURE_BIT":    8
    },
    "resnetv15_stage1_relu1_fwd":   {
        "QUANT_FEATURE_BIT":    4
    },
    "resnetv15_stage1_batchnorm3_fwd":  {
        "QUANT_FEATURE_BIT":    8
    },
    "resnetv15_stage1_activation1": {
        "QUANT_FEATURE_BIT":    8
    },
```

编辑后文件

3）设置 LOAD_OP_QUANT_BIT 选项为 true，加载 magik_node_quant_bit_info.json 文件，再次转换网络

```
{
    "SOC":"T40",
    "INPUT":[],
    "INPUT_SHAPE":[],
    "MEAN":[123.675,116.28,103.53],
    "NORMAL":[58.395,57.12,57.375],
    "COLOR":["RGB"],
    "OUTPUT":[],
    "QUANT_DATASET_PATH":"./Val_2012_Images_part",
    "DEBUG_PATH":"./Val_2012_Images_part",

    "QUANT_FEATURE_BIT":8,
    "SAVE_OP_QUANT_BIT": false,
    "LOAD_OP_QUANT_BIT": true
}
```

加载混合量化配置文件选项

```
INFO(magikexecutor): magik executor version:2.1.0(0_e011a9b_magik) built:20220629-1945:CPU
INFO(magikexecutor): magik executor version:2.1.0(0_e011a9b_magik) built:20220629-1945:CPU
Compute distance: 100.00 %
feature name                                cosine_avg cosine_min cosine_max euc_distan stand_devi max_absolu max_abs 0~1
resnetv15_relu0_fwd                          0.99991    0.99984    0.99994    0.00926    0.00414    0.22999    1.00333
resnetv15_pool0_fwd                          0.99993    0.99984    0.99997    0.00854    0.00452    0.22688    0.81948
resnetv15_stage1_relu0_fwd                   0.99680    0.99528    0.99746    0.05570    0.01440    0.60417    4.62447
resnetv15_stage1_batchnorm1_fwd              0.99717    0.99416    0.99832    0.03718    0.02879    0.81609    2.05455
resnetv15_stage1_activation0                 0.99889    0.99764    0.99941    0.03506    0.02417    0.49921    1.42254
resnetv15_stage1_relu1_fwd                   0.99272    0.98858    0.99526    0.07186    0.01592    0.68211    6.06604
resnetv15_stage1_batchnorm3_fwd              0.99188    0.98670    0.99522    0.06324    0.04965    0.79096    2.00974
resnetv15_stage1_activation1                 0.99654    0.99271    0.99835    0.06081    0.05183    0.73262    1.71296
resnetv15_stage2_relu0_fwd                   0.98926    0.97628    0.99389    0.08399    0.02478    0.61709    4.11824
resnetv15_stage2_batchnorm1_fwd              0.99098    0.98028    0.99440    0.06635    0.03722    0.47118    1.61330
resnetv15_stage2_batchnorm2_fwd              0.99400    0.98543    0.99737    0.05369    0.01969    0.40841    2.19772
resnetv15_stage2_activation0                 0.99397    0.98562    0.99651    0.06631    0.02945    0.47134    2.06470
resnetv15_stage2_relu1_fwd                   0.98573    0.96970    0.99075    0.09227    0.02318    1.06106    8.27366
resnetv15_stage2_batchnorm4_fwd              0.99097    0.97909    0.99445    0.06903    0.04814    0.64882    1.77714
resnetv15_stage2_activation1                 0.99224    0.98154    0.99503    0.07536    0.04058    0.66376    2.36548
resnetv15_stage3_relu0_fwd                   0.98687    0.97452    0.99082    0.08960    0.02514    0.88949    6.05156
resnetv15_stage3_batchnorm1_fwd              0.99148    0.98272    0.99411    0.06476    0.03940    0.50072    1.51799
resnetv15_stage3_batchnorm2_fwd              0.99444    0.98848    0.99642    0.05430    0.01242    0.14932    1.30015
resnetv15_stage3_activation0                 0.99223    0.98472    0.99471    0.07186    0.02620    0.53050    2.66910
resnetv15_stage3_relu1_fwd                   0.98349    0.97216    0.98850    0.09627    0.01764    0.65745    6.76264
resnetv15_stage3_batchnorm4_fwd              0.99204    0.98496    0.99451    0.07078    0.04639    0.59675    1.66385
resnetv15_stage3_activation1                 0.98911    0.97794    0.99296    0.08374    0.03349    0.62221    2.88259
resnetv15_stage4_relu0_fwd                   0.98030    0.95969    0.98651    0.10363    0.01755    0.55185    6.13422
resnetv15_stage4_batchnorm1_fwd              0.98894    0.97018    0.99395    0.07906    0.07374    0.57841    1.16288
resnetv15_stage4_batchnorm2_fwd              0.99397    0.99001    0.99620    0.07322    0.03047    0.27928    1.28544
resnetv15_stage4_activation0                 0.98463    0.96332    0.99011    0.09190    0.03749    0.65910    2.90522
resnetv15_stage4_relu1_fwd                   0.97508    0.93132    0.98232    0.11535    0.01769    0.50291    6.46273
resnetv15_stage4_batchnorm4_fwd              0.98187    0.93704    0.98913    0.09402    0.34814    2.75256    1.40305
resnetv15_stage4_activation1                 0.98609    0.95201    0.99161    0.09545    0.27433    2.64793    1.73313
resnetv15_pool1_fwd                          0.99440    0.97893    0.99738    0.07191    0.13059    0.61561    0.64375
flatten_170                                  0.99440    0.97893    0.99738    0.07191    0.13059    0.61561    0.64375
resnetv15_dense0_fwd                         0.99341    0.97172    0.99734    0.05460    0.29603    1.16724    0.39274
```

混合量化余弦相似度

4）重复步骤 2）、步骤 3）对模型进行逐层调试，设置节点量化位宽，观察余弦相似度是否合理可用。

# 5 加载最小值最大值

加载最小值最大值是 magik 工具开放量化接口的功能，magik 工具使用客户量化参数进行转换。该功能基于混合量化保存模型算子量化信息文件进行。

1）设置 SAVE_OP_QUANT_BIT 选项为 true，使用工具转换网络，保存当前模型节点的量化信息在 magik_node_quant_bit_info.json 文件，同 2.4 步骤 1）相同

2）编辑混合量化 magik_node_quant_bit_info.json 文件，更新 feature 的最小值、最大值信息

```
{
    "data/BatchNormScale":  {
        "QUANT_FEATURE_BIT":    8
    },
    "resnetv15_relu0_fwd":  {
        "QUANT_FEATURE_BIT":    8,
        "MAX": 1.7,
        "MIN": 0
    },
    "resnetv15_pool0_fwd":  {
        "QUANT_FEATURE_BIT":    8
    },
```

更新 feature 最值

3）设置 LOAD_OP_QUANT_MIN_MAX 为 ture，再次转换网络

```
"LOAD_OP_QUANT_MIN_MAX": true
```

打开加载选项

```
INFO(magikexecutor): magik executor version:2.1.0(0_e011a9b_magik) built:20220629-1945:CPU
run_model: 100.00 %
compute_feature_scale: 5.88 %2022-07-03 17:36:51.536283: I quantize_feature.cc:526-mse_compute_feature_scale] Updata Node: resnetv15_relu0_fwd, MIN: 0, MAX: 1.7
compute_feature_scale: 100.00 %
ComputeScale: 100.00 %
```

更新最值提示

# 6 模型上板

上板 demo 基于 yolov5s 网络进行，路径：***your_root/magik-toolkit/Models/post/yolov5s***

## 6.1 生成上板模型

在 ***your_root/magik-toolkit/Models/post/yolov5s*** 路径，我们提供 yolov5s 网络原生模型、校准集、配置文件、转换脚本，使用脚本可直接生成上板模型。

```
yolov5s]$ sh run_t40.sh
```

生成模型

## 6.2 代码编译

在 ***your_root/magik-toolkit/Models/post/yolov5s/venus_sample_yolov5s*** 路径下我们还提供了上板运行的 inference.cpp、Makfile、测试图片，以及生成的 yolov5s_t40_magik.bin 文件。

```
10_w1024_h714.nv12   magik_input_nhwc_1_384_640_3.bin   readme.md
fall_1054_sys.jpg    magik_model_yolov5s_t40_magik.mk.h  save-magik
inference.cpp        Makefile                            stb
inference_nv12.cpp   makefile_files                      yolov5s_t40_magik.bin
```

上板文件

另外我们还需要用到 venus 库及 mips 编译工具：

venus 库在 ***your_root/magik-toolkit/InferenceKit/nna1/mips720-glibc229/lib/uclibc*** 下，在 Makefile 文件中已添加路径，无需额外添加。

mips 编译工具由方案同事提供，需要添加编译工具路径到环境变量中

### 6.2.1 模型的加载

提供的实例 inference.cpp 中模型是通过参数传入的，运行前注意同步拷贝到板端对应的目录并在运行时传入。

### 6.2.2 超参数的设置

```
void generateBBox(std::vector<venus::Tensor> out_res, std::vector<magik::venus::ObjBbox_t>& candidate_boxes,
int img_w, int img_h)
{
  float person_threshold = 0.3;
  int classes = 80;
  float nms_threshold = 0.6;
  std::vector<float> strides = {8.0, 16.0, 32.0};
  int box_num = 3;
  std::vector<float> anchor = {10,13,  16,30,  33,23, 30,61,  62,45,  59,119, 116,90,  156,198,  373,326};
```

strides 和 anchor 按 yolov5 的实际使用设置，person_threshold 和 nms_threshold 分别对应原始代码中的 conf-thres（置信度阈值）和 iou-thres（iou 阈值），classes 是类别数。

### 6.2.3 编译

TOPDIR – venus 库的相对目录

libtype – 根据实际板子的需求确定是否 muclibc（这里以 muclibc 为例）

build_type – release 模式，运行得到结果，默认设置

– profile 模式，运行时网络结构可视化及网络每层运行时间及 GOPs 统计

– debug 模式，运行得到结果的同时保存每层量化 feature 的结果

- nmem 模式，统计模型运行时 nmem 内存占用情况，运行程序，内存使用情况保存在/tmp/nmem_memory.txt

直接 make 编译 inference.cpp 即可生成 venus_yolov5s_bin_uclibc_*，即我们上板需要的可执行文件。

```
Magik venus_sample_yolov5s$make
mips-linux-gnu-g++ -I../../../../InferenceKit/nna1/mips720-glibc229//include -std=c++11 -mfp64 -mnan=2008 -mabs=2008 -Wall -EL -O3 -march=mips32r2 -flax-vector-conversions -lpthread -lrt -ldl -lm -muclibc -o inference.o -c inference.cpp
mips-linux-gnu-g++ -std=c++11 -mfp64 -mnan=2008 -mabs=2008 -Wall -EL -O3 -march=mips32r2 -flax-vector-conversions -lpthread -lrt -ldl -lm -muclibc inference.o -o venus_yolov5s_bin_uclibc_release -I../../../InferenceKit/nna1/mips720-glibc229//include -L../../../../InferenceKit/nna1/mips720-glibc229//lib/uclibc -lvenus
```

使用 make 命令编译

注意：我们同时提供了用于实际板端运行的输入为 nv12 的代码用例 inference_nv12.cpp 及输入数据 10_w1024_h714.nv12，如有需要，可修改 Makefile 进行编译及使用测试。

## 6.3 上板运行

venus 库路径：***magik-toolkit/InferenceKit/nna1/mips720-glibc229/lib/uclibc/***下

### 6.3.1 release(发布库)

编译: make build_type=release

在当前文件夹下生成 venus_yolov5s_bin_uclibc_release 可执行文件，拷贝 venus 库(libvenus.so)、可执行文件(venus_yolov5s_bin_uclibc_release)、模型文件(yolov5s_t40_magik.bin)、测试图片(fall_1054_sys.jpg)至开发板运行即可：

**./venus_yolov5s_bin_uclibc_release yolov5s_t40_magik.bin fall_1054_sys.jpg**

注：运行前添加库路径至 LD_LIBRARY_PATH:

**export LD_LIBRARY_PATH=$lib_path:$LD_LIBRARY_PATH**

清除 **make build_type=release clean**

```
[root@Ingenic-uc1_1:venus_sample_ptq_yolov5s]# ./venus_yolov5s_bin_uclibc_release yolov5s_t40_magik.bin fall_1054_sys.jpg
The soc-nna version is 20220525
INFO(magik): venus memory map size: 0
INFO(magik): venus version:0.9.6.2.ALPHA(00000906_aa6e9e7)  built:20220721-2110(7.2.0 r5.1.3 glibc2.29 mips@NNA1)
INFO(magik): model version:0.9.6.NNA1_aa6e9e7
ori_image w,h: 388 ,574
model-->640 ,640 4
input shape:
-->384 640
scale---> 0.668990
resize padding over:
resize valid_dst, w:260 h 384
padding info top :0 bottom 0 left:190 right:190
test_net run time_ms:68.389000ms
pad_x:190 pad_y:0 scale:0.668990
post net time_ms:5.054000ms
box:    5 40 357 409 0.88
box:    95 324 379 512 0.73
```

### 6.3.2 debug（用于核对数据的库）

详见 Magik 量化使用指南步骤 7.2，输入的处理有些不同。

### 6.3.3 profile（网络可视化及每层运行时间统计）

编译: make build_type=profile

在当前文件夹下生成 venus_yolov5s_bin_uclibc_profile 可执行文件，拷贝 venus 库(libvenus.p.so)、可执行文件(venus_yolov5s_bin_uclibc_profile)、模型文件(yolov5s_t40_magik.bin)、测试图片(fall_1054_sys.jpg)至开发板运行即可：

**./venus_yolov5s_bin_uclibc_profile yolov5s_t40_magik.bin fall_1054_sys.jpg**

注：运行前添加库路径至 LD_LIBRARY_PATH:

**export LD_LIBRARY_PATH=$lib_path:$LD_LIBRARY_PATH**

清除 **make build_type=profile clean**

```
root@Ingenic-uc1_1:venus_sample_ptq_yolov5s# ./venus_yolov5s_bin_ptq_uclibc_profile ../yolov5s_magik.bin fall_1054_sys.jpg
./venus_yolov5s_bin_ptq_uclibc_pr
ofile ../yolov5s_magik.bin fall_1054_sys.jpg
Warning : The version number is not obtained. Please upgrade the soc-nna!
INFO(magik): venus version:0.9.0(00000900_8450a71)  built:20220211-1107(7.2.0 glibc2.26 mips@aie)
w:388 h:574
ori_image w.h: 388 ,574
model-->640 ,640 4
input shape:
-->384 640
scale---> 0.668990
resize padding over:
resize valid_dst, w:260 h 384
padding info top :0 bottom 0 left:190 right:190
pad_x:190 pad_y:0 scale:0.668990
box:  7 41 352 406 0.88
box:  97 329 380 510 0.71
[I/magik::venus]:
Timing cycle = 10
===== Detailed DispatchProfiler Summary: N/A, Exclude 5warm-ups =====
LayerName           OperatorType        InputShape       OutputShape      FilterShape      Stride    Pad          Dila      Avg(ms) Max(ms) Min(ms) Last(ms) Avg(%)  GOPs    BandWidth(MB)
AppendInputOutputNOpOptimizer_input_nop_0_BatchNormScale Normalize          {1,384,640,4}    {1,384,640,4}    N/A              N/A       N/A       N/A       1.808   1.809   1.807   2.82%   0.000   1.875
414_Quantize        Convolution         {1,384,640,4}    {1,192,320,32}   {3,3,4,32}       {1,1}     {1,1,1,1}    {1,1}     6.839   6.842   6.835   6.8400  10.66%  0.142   2.817
417_Quantize        Convolution         {1,192,320,32}   {1,96,160,64}    {3,3,32,64}      {2,2}     {1,1,1,1}    {1,1}     2.181   2.183   2.179   2.1830  3.40%   0.566   2.831
420_Quantize        Convolution         {1,96,160,64}    {1,96,160,32}    {1,1,64,32}      {1,1}     {0,0,0,0}    {1,1}     0.755   0.756   0.753   0.7540  1.18%   0.063   1.409
423_Quantize        Convolution         {1,96,160,32}    {1,96,160,32}    {1,1,32,32}      {1,1}     {0,0,0,0}    {1,1}     1.056   1.057   1.055   1.0550  1.65%   0.031   0.939
426_Quantize        Convolution         {1,96,160,32}    {1,96,160,32}    {3,3,32,32}      {1,1}     {1,1,1,1}    {1,1}     1.142   1.162   1.128   1.1290  1.78%   0.283   0.947
427                 Add                 {1,96,160,32}    {1,96,160,32}    N/A              N/A       N/A          N/A       0.609   0.611   0.608   0.6110  0.95%   0.000   1.406
432_0_Quantize      Convolution         {1,96,160,32}    {1,96,160,32}    {1,1,32,32}      {1,1}     {0,0,0,0}    {1,1}     1.093   1.096   1.090   1.0910  1.70%   0.031   0.939
432_1_Quantize      Convolution         {1,96,160,64}    {1,96,160,32}    {1,1,64,32}      {1,1}     {0,0,0,0}    {1,1}     0.755   0.757   0.754   0.7540  1.18%   0.063   1.409
432                 Concat              {1,96,160,64}    {1,96,160,64}    N/A              N/A       N/A          N/A       0.999   1.004   0.994   1.0000  1.56%   0.000   1.875
435_Quantize        Convolution         {1,96,160,64}    {1,96,160,64}    {1,1,64,64}      {1,1}     {0,0,0,0}    {1,1}     1.340   1.365   1.304   1.3610  2.09%   0.126   1.880
438_Quantize        Convolution         {1,96,160,64}    {1,48,80,128}    {3,3,64,128}     {2,2}     {1,1,1,1}    {1,1}     1.766   1.767   1.765   1.7660  2.75%   0.566   1.478
441_Quantize        Convolution         {1,48,80,128}    {1,48,80,64}     {1,1,128,64}     {1,1}     {0,0,0,0}    {1,1}     0.372   0.372   0.371   0.3720  0.58%   0.063   0.712
444_Quantize        Convolution         {1,48,80,64}     {1,48,80,64}     {1,1,64,64}      {1,1}     {0,0,0,0}    {1,1}     0.342   0.343   0.341   0.3430  0.53%   0.031   0.473
447_Quantize        Convolution         {1,48,80,64}     {1,48,80,64}     {3,3,64,64}      {1,1}     {1,1,1,1}    {1,1}     0.568   0.569   0.567   0.5680  0.89%   0.283   0.505
448                 Add                 {1,48,80,64}     {1,48,80,64}     N/A              N/A       N/A          N/A       0.337   0.340   0.335   0.3350  0.53%   0.000   0.703
```

### 6.3.4 nmem 模式(用于统计网络运行时 nmem 占用情况，保存在/tmp/nmem_memory.txt）

编译: make build_type=nmem

在当前文件夹下生成 venus_yolov5s_bin_uclibc_nmem 可执行文件，拷贝 venus 库(libvenus.m.so)、可执行文件(venus_yolov5s_bin_uclibc_nmem)、模型文件(yolov5s_t40_magik.bin)、测试图片(fall_1054_sys.jpg)至开发板运行即可：

**./venus_yolov5s_bin_uclibc_nmem yolov5s_t40_magik.bin fall_1054_sys.jpg**

注：运行前添加库路径至 LD_LIBRARY_PATH:

**export LD_LIBRARY_PATH=$lib_path:$LD_LIBRARY_PATH**

清除 **make build_type=nmem clean**

```
[root@Ingenic-uc1_1:venus_sample_ptq_yolov5s]# ./venus_yolov5s_bin_ptq_uclibc_nmem ../yolov5s_magik.bin fall_1054_sys.jpg
./venus_yolov5s_bin_ptq_uclibc_nm
em ../yolov5s_magik.bin fall_1054_sys.jpg
Warning : The version number is not obtained. Please upgrade the soc-nna!
INFO(magik): venus version:0.9.0(00000900_8450a71)  built:20220211-1007(7.2.0 glibc2.26 mips@aie)
w:388 h:574
ori_image w,h: 388 ,574
model-->640 ,640 4
input shape:
-->384 640
scale---> 0.668990
resize padding over:
resize valid_dst, w:260 h 384
padding info top :0 bottom 0 left:190 right:190
pad_x:190 pad_y:0 scale:0.668990
box:   7 41 352 406 0.88
box:   97 329 380 510 0.71
```

# 7 数据核对

验证 PC 端和板端的数据是否能对齐，可按如下步骤操作：

## 7.1 PC 端

使用由 Ingenic 提供的 PC 端推理库，通过设置相关环境变量可以实现算子逐层数据保存成二进制文件与板端产生的二进制文件进行比对，PC 端的具体操作步骤如下：

首先需要切换到 ***magik-toolkit/Models/post/pc_inference*** 文件夹下，在 PC 端通过设置环境变量

**export MAGIK_CPP_DUMPDATA=true**

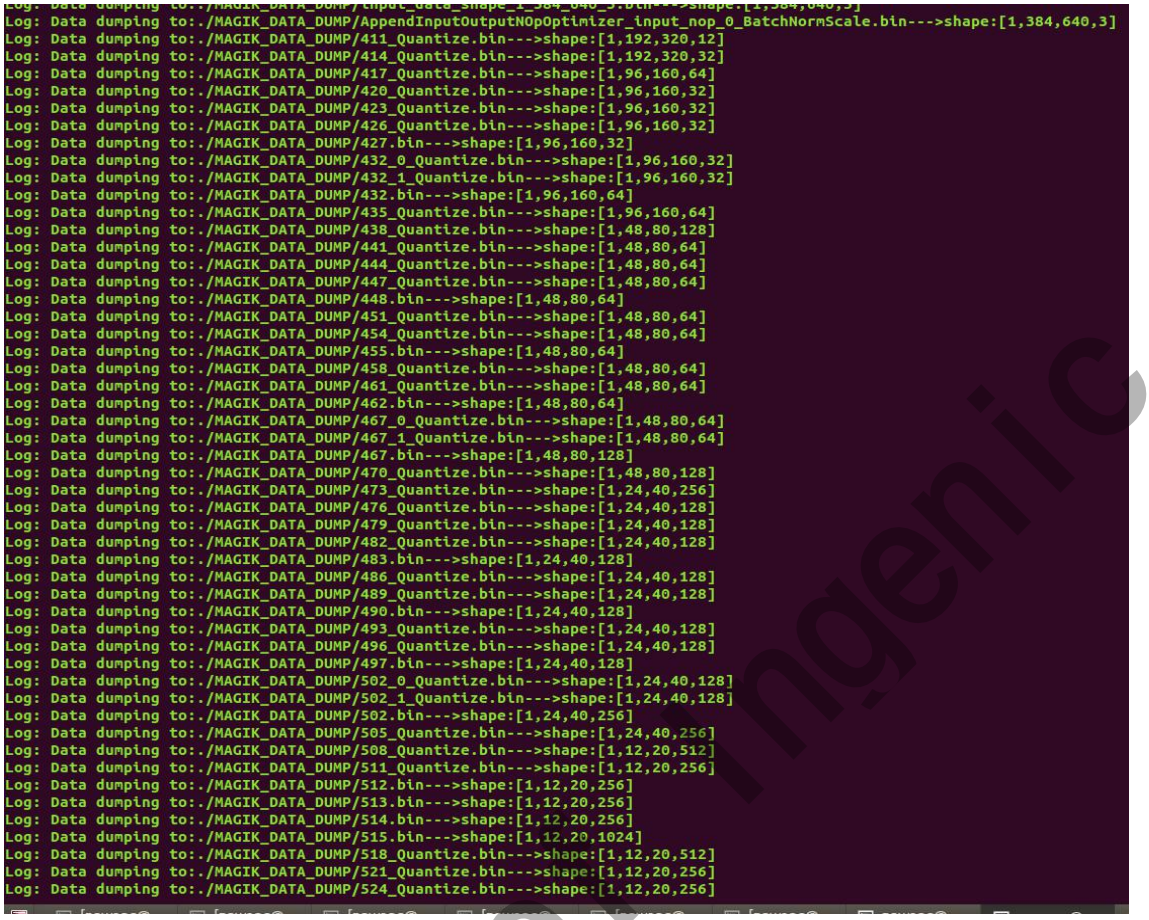
可保存每层量化结果到./MAGIK_DATA_DUMP/文件夹下，具体分步运行命令为:

**make clean**

**make -j12**

**./pc_inference_bin ../yolov5s/venus_sample_yolov5s/save-magik/model_quant.mgk fall_1054_sys.jpg ../yolov5s/venus_sample_yolov5s/ 640 384 3**

上述命令中 model_quant.mgk 是转化工具转化模型时生成的.mgk 文件，fall_1054_sys.jpg 是输入的图片，640 ,384 分别对应输入的宽和高。最终保存./MAGIK_DATA_DUMP/下（具体见下图），可见输入层有保存为 input_data_shape_1_384_640_3.bin，后面的 shape 命名的规则是 n,h,w,c, 即高为 384，宽为 640，最后面是输出层；如果您不希望分步执行的话，可以运行我们提供的 **exector.sh** 脚本来一键运行。




## 7.2 板端

(1) magik-toolkit/Models/post/yolov5s/venus_sample_yolov5s/下运行：

**make build_type=debug**

在当前文件夹下生成上板可执行文件(venus_yolov5s_bin_uclibc_debug)，拷贝该文件至开发板。

(2) 将上面pc端运行后产生MAGIK_DATA_DUMP文件夹里的input_data_shape_1_384_640_3.bin 拷贝至开发板。

(3) 将 magik-toolkit//Models/post/yolov5s/venus_sample_yolov5s/中 yolov5s_t40_magik.bin 文件拷贝至开发板，见步骤 6.2。

(4) 拷贝 venus 库(libvenus.d.so)、可执行文件(venus_yolov5s_bin_uclibc_debug)、模型文件(yolov5s_magik.bin)至开发板运行：

**./venus_yolov5s_bin_uclibc_debug yolov5s_t40_magik.bin**
**input_data_shape_1_384_640_3.bin**

运行的时候就会自动保存每层的输出及其他信息（如下图）：

```
[root@Ingenic-uc1_1:venus_sample_ptq_yolov5s]# ./venus_yolov5s_bin_uclibc_debug
yolov5s_t40_magik.bin input_data_shape_1_384_640_3.bin
image_bin shape:1 384 640 3
The soc-nna version is 20220525
INFO(magik): venus memory map size: 0
INFO(magik): venus version:0.9.6.2.ALPHA(00000906_aa6e9e7)  built:20220722-0937(
7.2.0 r5.1.3 glibc2.29 mips@NNA1)
INFO(magik): model version:0.9.6.NNA1_aa6e9e7
model-->640 ,640 4
input shape:
-->384 640
1,384,640,4,
[root@Ingenic-uc1_1:venus_sample_ptq_yolov5s]# ls
414_Quantize_bt.bin
414_Quantize_out.bin
414_Quantize_weight.bin
417_Quantize_bt.bin
417_Quantize_out.bin
417_Quantize_weight.bin
420_Quantize_bt.bin
420_Quantize_out.bin
420_Quantize_weight.bin
423_Quantize_bt.bin
423_Quantize_out.bin
423_Quantize_weight.bin
426_Quantize_bt.bin
426_Quantize_out.bin
426_Quantize_weight.bin
427_out.bin
432_0_Quantize_bt.bin
432_0_Quantize_out.bin
432_0_Quantize_weight.bin
432_1_Quantize_bt.bin
432_1_Quantize_out.bin
432_1_Quantize_weight.bin
435_Quantize_bt.bin
435_Quantize_out.bin
435_Quantize_weight.bin
```

其中 _*_QuantizeFeature_out.bin 和 PC 端运行时保存在 ./MAGIK_DATA_DUMP/ 下的 _*_QuantizeFeature.bin 是一一对应的，直接核对 md5 值是否一致即可，在保证输入完全一致的情况下中间层有不对应情况及时反馈。