Ingenic®

Magik

Post-Training-Quantization User Guide

Date: Feb. 2022

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

Ingenic®

Magik Post-Training-Quantization User Guide

Copyright© Ingenic Semiconductor Co. Ltd 2022. All rights reserved.

Release history

Date	Author	Revision	Change
Feb. 2022	Owen	1.0.1	First release
Jul. 2022	Owen	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 rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to 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 or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the 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

Copyright of Ingenic

目录

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

Copyright of Ingenic

1 Introduction

The MagikToolKit supports the conversion of the onnx/tensorflow/mxnet/PyTorch/Caffe format model to obtain a magik format model, which can be deployed on the chip side.

The document introduces the use of config in the conversion and quantization process, the post-quantization demo of single-input network and multi-input network, and the board process.

2 Use of MagikToolKit

2.1 Introduction of Config

The MagikToolKit completes the conversion and quantification of source models in tensorflow and onnx formats by parsing the information in the config file, and generates the bin model deployed on the board.

MagikToolKit can generate template config files by passing in the parameter "-gcfg" or "--gen_config".

```
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
```

generate template config file

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

template config file

SOC -- Deployment chip model

INPUT -- Specify the input of the model, the model can be intercepted, and the input information in the model is used when not specified

INPUT_SHAPE -- Specify the network input shape information, the following cases do not specify the input shape information:

1) The input shape can be obtained from the native model;

2) Generally, there is a one-to-one correspondence between the input shape information and the input information. If the length of the shape information is less than the length of the input information, the last shape information is used to broadcast the input information.

MEAN -- Specifies the mean information of the model input, the default value is 0, supports channel-level broadcasting and input node-level broadcasting

NORMAL -- Specifies the variance information of the model input, the default value is 1, supports

Copyright of Ingenic

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

channel-level broadcasting and input node-level broadcasting

COLOR -- When the input source of the model is an image, specify the image processing method "BRG"/"RGB"/"GRAY", the default is "RGB", which supports input-level broadcasting

OUTPUT -- Specify the output of the model, you can intercept the model, and use the output information in the model when the output is not specified.

QUANT_DATASET_PATH -- The path of the model post-quantization calibration dataset, use all images (or bin files) under the path for quantization operations; the option defaults to "", and no post-quantization is performed if no valid path is set.

2.2 Post-quantization Demo

By parsing the config file, MagikToolKit can support both single-input network and multi-input network post-quantification conversion work, and generate a bin file model for chip deployment. The single-input network and multi-input network process demonstrations are divided into two demos.

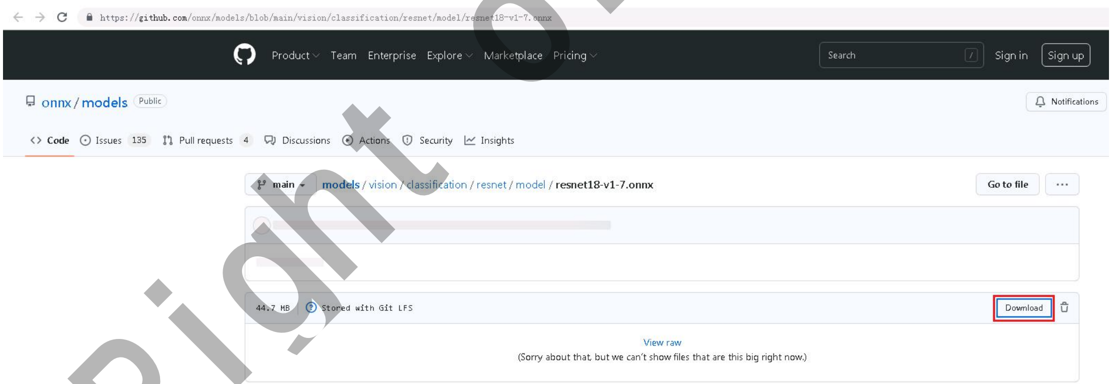
2.2.1 Single-input network post-quantization

The post-quantization demo of the single-input network uses the resnet18 network officially provided by onnx, and the demo path information:

your_root/magik-toolkit/Models/post/resnet18

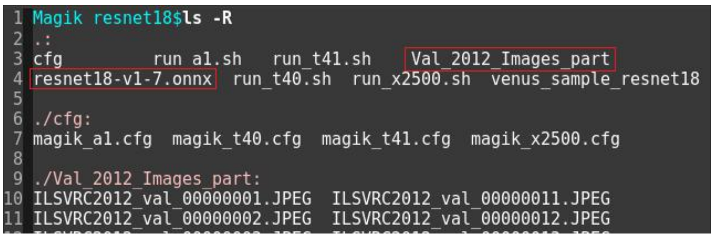
1）Prepare the model and calibration set

The currently used model is downloaded from github, link:
https://github.com/onnx/models/blob/main/vision/classification/resnet/model/resnet18-v1-7.onnx



Model download interface

The calibration dataset used for post-quantization is in the Val_2012_Images_part directory, and these quantized images are extracted from the ImageNet image test set.



The model to be quantified and the quantized picture

CopyRight of Ingenic

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

2）Config file

Complete the Json file quantitative information settings:

1)use the netron tool to view the native model structure to confirm the input and output and input shape information of the model;

2) confirm the pre-processing information of the model according to the pre-processing of the training code.

```
{
    "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"
}
```

Configuration files for single-input networks

SOC -- Coming soon to be deployed on the T40 chip

INPUT -- Specify model input node information

INPUT_SHAPE -- Specify Model Input Shape

MEAN -- Mean of image preprocessing during model training

NORMAL -- Variance of image preprocessing during model training

COLOR -- The model input image is processed into RGB format

OUTPUT -- Specify model output node information

QUANT_DATASET_PATH -- Specifies the path to the quantized image

3）Edit the conversion command

```
../../../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
```

conversion command

```
../../../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
```

simple conversion command

outputpath -- The generation path and name of the chip deployment file, the suffix requires ".mk.h"

inputpath -- The path address of the source model

config -- The path address of the config file

4）Conversion quantization

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

success tips

CopyRight of Ingenic

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

```
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
```

Generated inference model on chip

2.2.2 Multi-input network post-quantization

The post-quantization demo of the multi-input network uses the gru network, and the demo path: ***your_root/magik-toolkit/Models/post/gru***

1）Prepare the model and calibration set

The gru network is in the path of ***your_root/magik-toolkit/Models/post/gru***, no need to download.

Multiple input networks require a two-level directory for quantified datasets:

1)The first-level directory is a wrapper for the quantization dataset, and the current directory represents the number of datasets that can be used by post-quantization of the network

2)The second-level directory is the real input of the feeding model. The file name prefix is the network input name, and the suffix is .bin. There is a one-to-one correspondence between the network input and the bin file.

```
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
```

two-level directory

2）Config file

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

Configuration files for multi-input networks

The network has two inputs, because only one input needs to be intercepted, and INPUT has only one input and specifies the shape information of the input; the other input and shape can be obtained from the model, or can be specified.

CopyRight of Ingenic

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

```
{
    "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"
}
```

Complete input information

Specify the input shape using [shape1,...] for single network input, and use [[shape1,...],[shape2,...]] when specifying the network input shape when the network is multi-input.

3）Edit the conversion command

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

conversion command

4）Conversion quantization

```
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
```

Generated inference model on chip

3 Debug Quantized Model Accuracy

3.1 Introduction of functions

MagikToolKit debugs the post-quantization model through the Json file configuration options, locates problems, and fixes the problem of poor accuracy of the post-quantization model.

3.2 debug configuration options

```
"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 Options

DEUBG_PATH -- Is a comprehensive option, the default is "", the option is off. When a valid path is specified (n images or bin files are specified in the path), the debug mode of MagikTools is turned on.

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


Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

CopyRight of Ingenic

Set quantization dataset path

1) Download the inference results of the simulation before and after model quantization (the pre-quantization simulation results are the same as the original model inference results, and the post-quantization simulation results are the same as the board-side inference results), and generate a directory "magik_dump_data" under the running path to store the simulation results;

```
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:
```

Generated Simulation Results

model_node_info.json -- Operator type information in the model.

input.bin -- The input used by the conversion tool for simulation inference can be used as the inference input of the native model or the inference input of the upper board.

float_model -- Native model simulation inference results

ptq_model -- Post-quantization model simulation inference results

ptq_model_dequantize -- Post-quantization model simulation inference results for restoration work

2) Calculate the cosine similarity of simulation results before and after model quantization. In order to obtain an objective and real quantization effect, each image (bin file) of DEBUG_PATH is inferred to obtain a set of cosine similarity. In multiple sets of cosine similarity, calculate the average, minimum, maximum, Euclidean distance, standard deviation, maximum difference, and normalized maximum difference of each node’s cosine similarity.

```
2022-06-20 10:08:29.534893: I post_training_quantization.cc:947-quantize_feature_weight] optimizer magik_ptq model finish!
<<<<<<<<<<<<<<<<<<<<<<<<<  ptq model compute distance  >>>>>>>>>>>>>>>>>>>>>>>>>
INFO(magikexecutor): magik executor version:2.1.0(0_de87996_magik) built:20220520-1136:CPU
INFO(magikexecutor): magik executor version:2.1.0(0_de87996_magik) built:20220520-1136:CPU
Compute distance: 100.00 %
feature name                          cosine_avg cosine_min cosine_max euc_distan stand_devi max_absolu max_abs_0~1
functional_1/conv2d/Relu               0.99996    0.99975    0.99998    0.00445    0.00121    0.02141    0.16431
functional_1/max_pooling2d/MaxPool     0.99995    0.99972    0.99999    0.00475    0.00172    0.02101    0.12023
```

Cosine similarity calculation result

3) Generate the Json file of the operator information to be quantized in the model, including the operator types (all_type option) that the tool supports quantization in the model, and the specific operator name (Conv2D option, MatMul option, etc.) included in each quantization type operator.

```
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
```

Generate model information Json file

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

```
{
    "all_type": ["BatchNormScale", "Conv2D", "EltwiseAdd", "Flatten", "GlobalAvgPool", "MatMul", "MaxPool"],
    "BatchNormScale":   ["data/BatchNormScale"],
    "Conv2D":   ["resnetv15_relu0_fwd", "resnetv15_stage1_relu0_fwd", "resnetv15_stage1_batchnorm1_fwd", "resnetv15_stage1_relu1_fwd", "resnetv15_stage1_batchnorm3_fwd", "resnetv15_stage2_relu0_fwd", "resnetv15_stage2_batchnorm1_fwd", "resnetv15_stage2_batchnorm2_fwd", "resnetv15_stage2_relu1_fwd", "resnetv15_stage2_batchnorm4_fwd", "resnetv15_stage3_relu0_fwd", "resnetv15_stage3_batchnorm1_fwd", "resnetv15_stage3_batchnorm2_fwd", "resnetv15_stage3_relu1_fwd", "resnetv15_stage3_batchnorm4_fwd", "resnetv15_stage4_relu0_fwd", "resnetv15_stage4_batchnorm1_fwd", "resnetv15_stage4_batchnorm2_fwd", "resnetv15_stage4_relu1_fwd", "resnetv15_stage4_batchnorm4_fwd"],
    "EltwiseAdd":   ["resnetv15_stage1_activation0", "resnetv15_stage1_activation1", "resnetv15_stage2_activation0", "resnetv15_stage2_activation1", "resnetv15_stage3_activation0", "resnetv15_stage3_activation1", "resnetv15_stage4_activation0", "resnetv15_stage4_activation1"],
    "Flatten":  ["flatten_170"],
    "GlobalAvgPool":    ["resnetv15_pool1_fwd"],
    "MatMul":   ["resnetv15_dense0_fwd"],
    "MaxPool":  ["resnetv15_pool0_fwd"]
}
```

Resnet18 model_node_info.json

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

Quantization bit width setting

QUANT_FEATURE_BIT -- Specify the activation quantization bit width, the default is 8-bit quantization. When the default activation quantization bit width cannot guarantee the model accuracy, the activation quantization bit width can be increased. The current post-quantization algorithm supports up to 12-bit quantization for activation.

QUANT_WEIGHT_BIT -- Specify the weight quantization bit width, the default is 8-bit quantization, and the current post-quantization algorithm supports up to 8-bit quantization for weights.

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

Quantization method settings

QUANT_FEATURE_METHOD -- Activation quantization method, the default use "MAX_ABS" method for quantization, the current activation supports {KL, ADMM, MSE} method quantization.

QUANT_WEIGHT_METHOD -- The quantization method of the weight, the "KL" method is used for quantization by default, and the current weight supports {MAX_ABS, ADMM} method quantization

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

Quantization Set Number Setting

QUANT_DATASET_SIZE -- Specifies the number of calibration datasets (images or bin files) to be used for the quantization process. If not specified, use all images in the directory.

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

Specifies the operator to skip quantization

QUANT_NODE_SKIP -- Skip the quantization of the specified operator name, and debug whether the operator of this layer causes the loss of model accuracy. By default, the parentheses are empty, indicating that no operator is specified to skip the quantization.

QUANT_OP_SKIP -- Skip the quantization of the specified operator type, and debug whether the operator of this type causes the loss of model accuracy. By default, the parentheses are empty, indicating that no operator is specified to skip the quantization.

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


Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

CopyRight of Ingenic

Model with fake quantization settings

FEATURE_FAKE_QUANT -- Only perform fake quantization operation on model activation to confirm whether activation of quantization leads to loss of model accuracy. The option defaults to OFF.

WEIGHT_FAKE_QUANT -- Only perform fake quantization operation on the model weights to confirm whether the weight quantization leads to the loss of model accuracy. The option is off by default.

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

Model Weight Quantization Correction Settings

WEIGHT_CORRECT_PATH -- The default is empty, and no correction work is performed; when an effective path for correction is input, all images (or bin files) under the path are used to correct the loss of weight quantization, which can improve the loss of weight quantization to a limited extent.

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

Generate and load operator quantization information settings

SAVE_OP_QUANT_BIT -- Save the activated quantization bit width information as a Json file, and edit the file to specify the activated quantization bit width.

LOAD_OP_QUANT_BIT -- Load the Json file generated by the SAVE_OP_QUANT_BIT option, and quantize the specified bit width for the activation of the quantization operator. The specified quantization bit width does not exceed 12 bits, and the priority is higher than the QUANT_FEATURE_BIT option.

```
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
```

Generate magik_node_quant_bit_info.json under the running path

3.3 Debug instructions

When the network deployment effect on the board is not satisfactory, or when the user has high requirements for the cosine similarity before and after quantization, add the DEBUG_PATH option to the Json file, specify a valid Debug path, and locate specific problems.

Debugging based on the above introduction of resnet18 network quantization

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

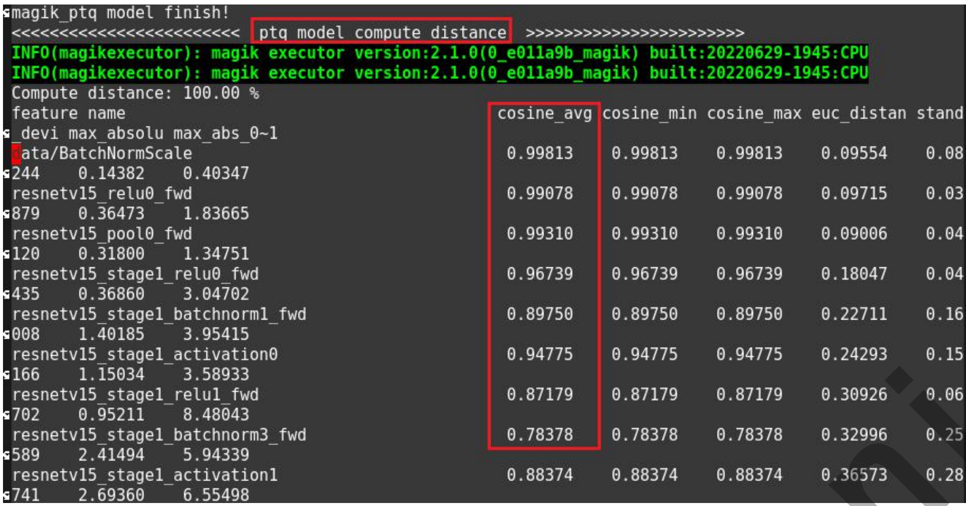
set valid directory

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

Set a valid image

CopyRight of Ingenic

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.


```
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
```

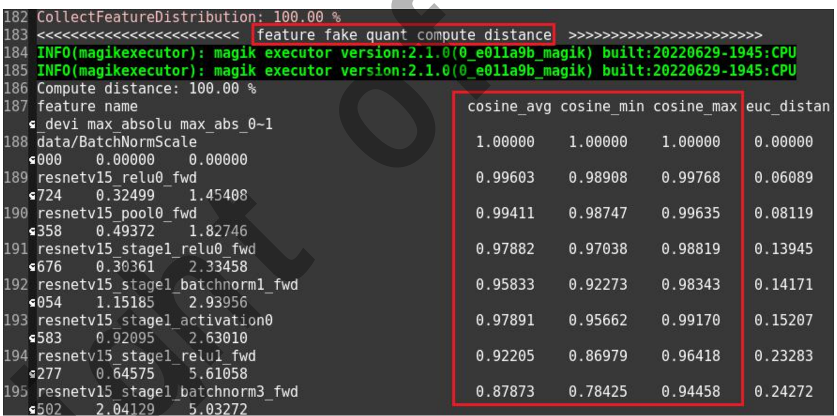
When the model cosine similarity is poor

1）Average similarity cosine similarity below 98%

Reason 1: The activated quantization bit width is not sufficient to represent floating-point activation.

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

Use the FEATURE_FAKE_QUANT option to convert to confirm whether the quantized cosine similarity is poor or not


```
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
```

If as described above, use the QUANT_FEATURE_BITS option to increase the active quantization bit width to 10, 12;

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

Increase activation quantization bit width

Convert again, check if activation quantized cosine similarity improves, check if overall quantized cosine similarity improves.

Reason 2: The weight quantization bit width cannot fully represent the floating-point weight.

```
"WEIGHT_FAKE_QUANT": true
```

Set the WEIGHT_FAKE_QUANT option to true to confirm whether the weighted quantized cosine similarity is poor;

Copyright of Ingenic

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

```
nference_process finish!
<<<<<<<<<<<<<<<<<<<<<<<<  weight_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_dista
_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                                0.98944    0.97564    0.99400    0.09781
499    0.45331    1.96759
resnetv15_pool0_fwd                                0.99293    0.98049    0.99665    0.08581
633    0.42209    1.55222
resnetv15_stage1_relu0_fwd                         0.93745    0.91098    0.94795    0.23723
538    0.45876    3.40987
resnetv15_stage1_batchnorm1_fwd                    0.90324    0.80942    0.94197    0.21699
182    1.07084    2.76211
resnetv15_stage1_activation0                       0.96354    0.92822    0.98002    0.20095
476    0.98142    2.84406
resnetv15_stage1_relu1_fwd                         0.89193    0.80828    0.93745    0.27206
128    0.57929    5.10780
resnetv15_stage1_batchnorm3_fwd                    0.86420    0.74649    0.92526    0.25679
303    1.45032    3.67947
resnetv15_stage1_activation1                       0.93266    0.86689    0.96431    0.26717
284    1.67328    3.79573
```

If described above, use the WEIGHT_COREECT_PATH option to correct for the loss of precision caused by WEIGHT quantization. The weight correction path should try to ensure that the quantization calibration and debug sets are not repeated.

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

Set path information for calibration set

```
bias_correct Op------------->    5, total Op is   25 INFO(magikexecutor): magik executor version:2.
1.0(0_de87996_magik) built:20220520-1136:CPU
bias_correct Op------------->    9, total Op is   25 INFO(magikexecutor): magik executor version:2.
1.0(0_de87996_magik) built:20220520-1136:CPU
bias_correct Op------------->   13, total Op is   25 INFO(magikexecutor): magik executor version:2.
1.0(0_de87996_magik) built:20220520-1136:CPU
bias_correct Op------------->   22, total Op is   25 INFO(magikexecutor): magik executor version:2.
1.0(0_de87996_magik) built:20220520-1136:CPU
bias_correct Op------------->   23, total Op is   25 INFO(magikexecutor): magik executor version:2.
1.0(0_de87996_magik) built:20220520-1136:CPU
bias_correct Op------------->   24, total Op is   25 INFO(magikexecutor): magik executor version:2.
1.0(0_de87996_magik) built:20220520-1136:CPU
bias_correct Op------------->   25, total Op is   25
2022-06-16 17:00:40.114310: I post_training_quantization.cc:1668-post_training_quantization] bias c
orrection finish!
```

Weight Correction Process

Convert again, check whether the weighted quantized cosine similarity is improved, and check whether the overall quantized cosine similarity is improved;

Reason 3: There is a problem in the operator quanmodel_node_info.jsontization process

```
{
    "all_type": ["BatchNormScale", "Conv2D", "EltwiseAdd", "Flatten", "GlobalAvgPool", "MatMul", "MaxPool"],
    "BatchNormScale":   ["data/BatchNormScale"],
    "Conv2D":   ["resnetv15_relu0_fwd", "resnetv15_stage1_relu0_fwd", "resnetv15_stage1_batchnorm1_fwd", "resnetv15_stage1_relu1_fwd", "resnetv15_stage1_batchnorm3_fwd", "resnetv15_stage2_relu0_fwd", "resnetv15_stage2_batchnorm1_fwd", "resnetv15_stage2_batchnorm2_fwd", "resnetv15_stage2_relu1_fwd", "resnetv15_stage2_batchnorm4_fwd", "resnetv15_stage3_relu0_fwd", "resnetv15_stage3_batchnorm1_fwd", "resnetv15_stage3_batchnorm2_fwd", "resnetv15_stage3_relu1_fwd", "resnetv15_stage3_batchnorm4_fwd", "resnetv15_stage4_relu0_fwd", "resnetv15_stage4_batchnorm1_fwd", "resnetv15_stage4_batchnorm2_fwd", "resnetv15_stage4_relu1_fwd", "resnetv15_stage4_batchnorm4_fwd"],
    "EltwiseAdd":   ["resnetv15_stage1_activation0", "resnetv15_stage1_activation1", "resnetv15_stage2_activation0", "resnetv15_stage2_activation1", "resnetv15_stage3_activation0", "resnetv15_stage3_activation1", "resnetv15_stage4_activation0", "resnetv15_stage4_activation1"],
    "Flatten":  ["flatten_170"],
    "GlobalAvgPool":    ["resnetv15_pool1_fwd"],
    "MatMul":   ["resnetv15_dense0_fwd"],
    "MaxPool":  ["resnetv15_pool0_fwd"]
}
```

model_node_info.json

1）The QUANT_OP_SKIP option is combined with the content of all_type, skips the quantization of the specified type of operator, and uses the 2-point method to locate which type of operator quantization is problematic;

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

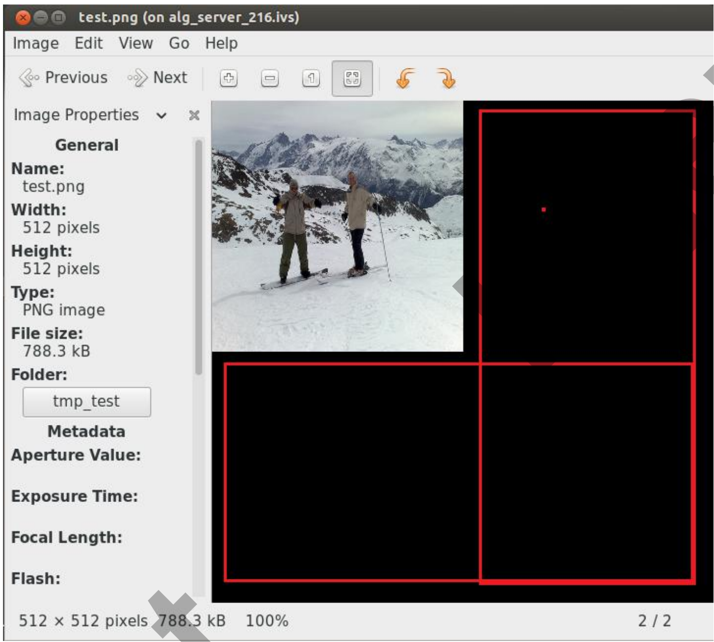
2 ） Based on the QUANT_OP_SKIP option to locate the result, you can use the QUANT_NODE_SKIP option in combination with skipping the quantization of the specified operator name, and use the 2-point method to locate the specific node that has problems with this type of operator

Locate the specific operator and feed back to the relevant docking personnel to fix the problem

Reason 4: Not selecting the right picture or the right amount of pictures

Check whether the number of quantized pictures is small, it is recommended that 10~200 pictures be quantized;

In the real quantization process, there are a large number of invalid parts in the quantized picture;



The image has a lot of padding values

In the implementation process, the selection of quantized pictures has a certain loss of quantization accuracy, so as to avoid a large number of black backgrounds, gray backgrounds, etc. in the selected pictures.

2) The average cosine degree is 99%, and the accuracy of the model deployed on the chip is lost

Reason 1: There is a problem with inference on the chip side

The board-side inference results and the post-quantization model simulation results cannot be aligned. Use the DEBUG_PATH option to generate the input.bin results for inference on the chip side, and compare them with the ptq_model results:

1) input.bin is stored in NHWC format and float32 format;

2) The inference results of the quantized model are stored in the ptq_model directory, and the results are stored in NHWC format and uint8 format (the results are stored in uint16 format when 10- and 12-bit quantization is performed on activation)

Copyright Ingenic

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

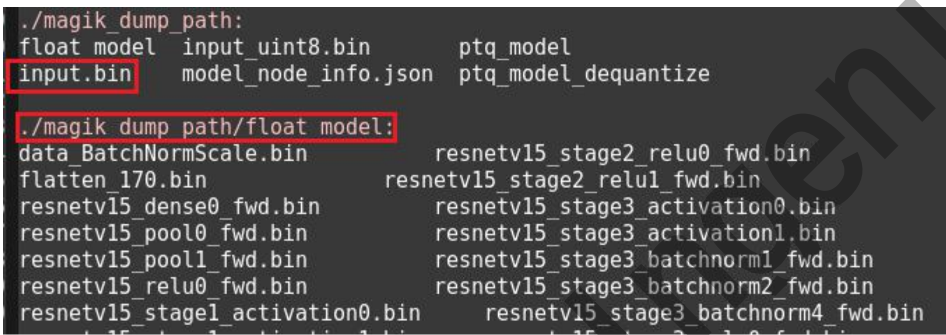
Simulation results of the post-quantization model

Reason 2: There is a problem with the MagikTools native model simulation inference

The native model simulation result of the conversion tool cannot be aligned with the native model inference. Use the DEBUG_PATH option to generate the input.bin result as the native model input for inference, and compare it with the float_model result:

1) input.bin is stored in NHWC format and float32 format. In the native model inference, the input needs to be processed with mean variance, and the mean variance has been merged into the model during simulation inference;

2) The results of model simulation inference stored in the float_model directory are stored in NHWC format and float format



Simulation results of the native model

If there is an error in the simulation inference, it needs to be reported to the relevant docking personnel to fix the problem.

4 Activate mixed bitwidth quantization

In the process of quantizing the model effect after debugging, the activation quantization effect is not ideal, adding the FEATURE_QUANT_BITS option to increase the quantization bit width improves the quantization effect, but increasing the quantization bit width leads to the deterioration of the inference efficiency of the quantization model

In order to improve the quantization effect and ensure the inference efficiency, we propose a hybrid bit-width quantization method. Some operators perform 8-bit quantization, and some operators perform high-bit quantization (10 and 12 bits).

Mixed bitwidth options

Demonstration of activated mixed bitwidth quantization based on resnet18 network

1) Set the SAVE_OP_QUANT_BIT option to true, use the tool to convert the network, and save the quantization information of the current model node in the ***magik_node_quant_bit_info.json*** file

CopyRight of Ingenic

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

```
{
    "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
}
```

Set LOAD_OP_QUANT_BIT to true

```
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
```

Tips

2）Edit the mixed bitwidth quantization magik_node_quant_bit_info.json file

```
{
    "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
    },
```

file before editing

```
{
    "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
    },
```

file after editing

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

CopyRight of Ingenic

3）Set the SAVE_OP_QUANT_BIT option to false, Set the LOAD_OP_QUANT_BIT option to true, load the magik_node_quant_bit_info.json file, and convert the network again

```
{
    "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
}
```

Load mixed bitwidth quantization profile options

```
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

Hybrid bitwidth quantized cosine similarity

4）Repeat steps 2) and 3) to debug the model layer by layer, set the node quantization bit width, and observe whether the cosine similarity is improved and available.

5 load min max

Load Min Max is a function of MagikTools' open quantization interface, and MagikTools uses customer quantization parameters for conversion. This function is based on the mixed bit-width quantization saving model operator quantization information file.

1) Set the SAVE_OP_QUANT_BIT option to true, use the tool to convert the network, and save the quantization information of the current model node in the magik_node_quant_bit_info.json file, which is the same as step 1) in 2.4

2) Edit the mixed quantization magik_node_quant_bit_info.json file to update the minimum and maximum values of activation

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

```
{
    "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
    },
```

Update activation min max

3)Set the SAVE_OP_QUANT_BIT option to false, Set LOAD_OP_QUANT_MIN_MAX to true and convert the network again

```
"LOAD_OP_QUANT_MIN_MAX": true
```

Open loading options

```
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 %
```

Update prompt

6 Model Board Process

Demo based on yolov5s network, route: ***your_root/magik-toolkit/Models/post/yolov5s***

6.1 Generate Board Model

In ***your_root/magik-toolkit/Models/post/yolov5s*** path, We provide yolov5s network native model, calibration set, configuration file and conversion script, which can be used to directly generate the on board model.

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

Generate the on board model

6.2 Code Compilation

In ***your_root/magik-toolkit/Models/post/yolov5s/venus_sample_yolov5s*** path, We also provide inference.cpp, Makefile, test image and generated yolov5s_t40_magik.bin file for running on the board

```
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
```

Board file

In addition, we also need to use the Venus library and MIPS compilation tool:

Venus library is in ***your_root/magik-toolkit/InferenceKit/nna1/mips720-glibc229/lib/uclibc,*** the path has been added in the makefile file, so there is no need to add additional paths.

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

CopyRight of Ingenic

The mips compilation tool is provided by the solution colleagues. You need to add the compilation tool path to the environment variable.

6.2.1 Loading Of Model

The model in the provided instance inference.cpp is passed in through parameters. Before running, pay attention to synchronously copying it to the corresponding directory on the board end and passing it in at run time.

6.2.2 Setting Of Super Parameters

```
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};
```

Setting Of Super Parameters

Stripes and anchor are set according to the actual use of yolov5, person_ threshold and nms_threshold corresponds to conf thresholds and IOU thresholds in the original code respectively, and classes is the number of categories.

6.2.3 Compile

TOPDIR – relative directory of Venus Library

libtype – determine whether to use muclibc according to the requirements of the actual board (here take muclibc as an example)

build_type – release mode, run to get results, default setting

– profile mode， visualization of network structure at runtime and statistics of running time and GOPs at each layer of the network

– debug mode， save the results of each layer of quantitative features while running the results

– nmem mode, count the memory usage of nmem when the model is running, run the program, and save the memory usage in /tmp/nmem_ memory.txt

Directly make and compile information.cpp to generate Venus_ yolov5s_ bin_ uclibc_*， That is, the executable file we need on the board.

```
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 -mucli
bc -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
```

Compile using the make command

Note: We also provide the code case inference_nv12.cpp and 10_w1024_h714.nv12 with input of nv12 for the actual board end operation. If necessary, you can modify the makefile for compilation and use testing.

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

Copyright of Ingenic

6.3 Operation On Board

Venus library path: ***magik-toolkit/InferenceKit/nna1/mips720-glibc229/lib/uclibc/***

6.3.1 Release

Compile：make build_type=release

Generate the venus_yolov5s_bin_uclibc_release executable file in the current folder, copy the Venus Library (libvenus.so), executable file (venus_yolov5s_bin_uclibc_release), model file (yolov5s_t40_magik.bin), test image (fall_1054_sys.jpg) to the development board for operation:

./venus_yolov5s_bin_uclibc_release yolov5s_t40_magik.bin fall_1054_sys.jpg

Note: add the library path to LD_LIBRARY_PATH before running:

export LD_LIBRARY_PATH=$lib_path:$LD_LIBRARY_PATH

make build_type=release clean

```
[root@Ingenic-uc1_1:venus_sample_ptq_yolov5s]# ./venus_yolov5s_bin_uclibc_releas
e 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

For details, please see step 7.2 of magik quantification guide. The input processing is somewhat different.

6.3.3 Profile

Compile：make build_type=profile

Generate the venus_yolov5s_bin_uclibc_profile executable file in the current folder, copy the Venus Library (libvenus.p.so), executable file (venus_yolov5s_bin_uclibc_profile), model file (yolov5s_t40_magik.bin), test image (fall_1054_sys.jpg) to the development board and run it:

./venus_yolov5s_bin_uclibc_profile yolov5s_t40_magik.bin fall_1054_sys.jpg

Note: add the library path to LD_LIBRARY_PATH before running:

export LD_LIBRARY_PATH=$lib_path:$LD_LIBRARY_PATH

make build_type=profile clean

Copyright of Ingenic

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

```
[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    BandWid
th(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   1.80
7   1.8070   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

Compile：make build_type=nmem

Generate the venus_yolov5s_bin_uclibc_nmem executable file in the current folder, copy the Venus Library (libvenus.m.so), executable file (venus_yolov5s_bin_uclibc_nmem), model file (yolov5s_t40_magik.bin), test image (fall_1054_sys.jpg) to the development board and run it:

./venus_yolov5s_bin_uclibc_nmem yolov5s_t40_magik.bin fall_1054_sys.jpg

Note: add the library path to LD_LIBRARY_PATH before running:

export LD_LIBRARY_PATH=$lib_path:$LD_LIBRARY_PATH

make build_type=nmem clean

```
[root@Ingenic-uc1_1:venus_sample_ptq_yolov5s]# ./venus_yolov5s_bin_ptq_uclibc_nmem ../y
olov5s_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:20220211-1107(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
```

7 Data Check

To verify whether the data of PC end and board end can be aligned, follow the following steps:

7.1.1 PC Port

Using the PC side reasoning library provided by ingenic, by setting the relevant environment variables,

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

the operator can save the data layer by layer into binary files and compare them with the binary files generated at the board side. The specific operation steps at the PC side are as follows:
First, you need to switch to the ***magik-toolkit/Models/post/pc_inference*** folder and set environment variables on the PC port.

export MAGIK_CPP_DUMPDATA=true

The quantization results of each layer can be saved to the ./MAGIK_DATA_DUMP/ folder. The specific step-by-step operation commands are:

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

The model_quant.mgk in the above command is generated when the conversion tool converts the model .mgk file, and fall_1054_sys.jpg are the input pictures, 640 and 384 correspond to the input width and height respectively. Finally result save in the./MAGIK_DATA_DUMP/ (see the following figure for details), and it can be seen that the input layer is saved as input_data_shape_1_384_640_3.bin, and the naming rules of the following shape are n, h, w, c, that is, the height is 384, the width is 640, and the output layer is at the end; If you don't want to execute step by step, you can run the **exector.sh** script provided by us to run with one click.

```
Log: Data dumping to:./MAGIK_DATA_DUMP/AppendInputOutputNOpOptimizer_input_nop_0_BatchNormScale.bin--->shape:[1,384,640,3]
Log: Data dumping to:./MAGIK_DATA_DUMP/411_Quantize.bin--->shape:[1,192,320,12]
Log: Data dumping to:./MAGIK_DATA_DUMP/414_Quantize.bin--->shape:[1,192,320,32]
Log: Data dumping to:./MAGIK_DATA_DUMP/417_Quantize.bin--->shape:[1,96,160,64]
Log: Data dumping to:./MAGIK_DATA_DUMP/420_Quantize.bin--->shape:[1,96,160,32]
Log: Data dumping to:./MAGIK_DATA_DUMP/423_Quantize.bin--->shape:[1,96,160,32]
Log: Data dumping to:./MAGIK_DATA_DUMP/426_Quantize.bin--->shape:[1,96,160,32]
Log: Data dumping to:./MAGIK_DATA_DUMP/427.bin--->shape:[1,96,160,32]
Log: Data dumping to:./MAGIK_DATA_DUMP/432_0_Quantize.bin--->shape:[1,96,160,32]
Log: Data dumping to:./MAGIK_DATA_DUMP/432_1_Quantize.bin--->shape:[1,96,160,32]
Log: Data dumping to:./MAGIK_DATA_DUMP/432.bin--->shape:[1,96,160,64]
Log: Data dumping to:./MAGIK_DATA_DUMP/435_Quantize.bin--->shape:[1,96,160,64]
Log: Data dumping to:./MAGIK_DATA_DUMP/438_Quantize.bin--->shape:[1,48,80,128]
Log: Data dumping to:./MAGIK_DATA_DUMP/441_Quantize.bin--->shape:[1,48,80,64]
Log: Data dumping to:./MAGIK_DATA_DUMP/444_Quantize.bin--->shape:[1,48,80,64]
Log: Data dumping to:./MAGIK_DATA_DUMP/447_Quantize.bin--->shape:[1,48,80,64]
Log: Data dumping to:./MAGIK_DATA_DUMP/448.bin--->shape:[1,48,80,64]
Log: Data dumping to:./MAGIK_DATA_DUMP/451_Quantize.bin--->shape:[1,48,80,64]
Log: Data dumping to:./MAGIK_DATA_DUMP/454_Quantize.bin--->shape:[1,48,80,64]
Log: Data dumping to:./MAGIK_DATA_DUMP/455.bin--->shape:[1,48,80,64]
Log: Data dumping to:./MAGIK_DATA_DUMP/458_Quantize.bin--->shape:[1,48,80,64]
Log: Data dumping to:./MAGIK_DATA_DUMP/461_Quantize.bin--->shape:[1,48,80,64]
Log: Data dumping to:./MAGIK_DATA_DUMP/462.bin--->shape:[1,48,80,64]
Log: Data dumping to:./MAGIK_DATA_DUMP/467_0_Quantize.bin--->shape:[1,48,80,64]
Log: Data dumping to:./MAGIK_DATA_DUMP/467_1_Quantize.bin--->shape:[1,48,80,64]
Log: Data dumping to:./MAGIK_DATA_DUMP/467.bin--->shape:[1,48,80,128]
Log: Data dumping to:./MAGIK_DATA_DUMP/470_Quantize.bin--->shape:[1,48,80,128]
Log: Data dumping to:./MAGIK_DATA_DUMP/473_Quantize.bin--->shape:[1,24,40,256]
Log: Data dumping to:./MAGIK_DATA_DUMP/476_Quantize.bin--->shape:[1,24,40,128]
Log: Data dumping to:./MAGIK_DATA_DUMP/479_Quantize.bin--->shape:[1,24,40,128]
Log: Data dumping to:./MAGIK_DATA_DUMP/482_Quantize.bin--->shape:[1,24,40,128]
Log: Data dumping to:./MAGIK_DATA_DUMP/483.bin--->shape:[1,24,40,128]
Log: Data dumping to:./MAGIK_DATA_DUMP/486_Quantize.bin--->shape:[1,24,40,128]
Log: Data dumping to:./MAGIK_DATA_DUMP/489_Quantize.bin--->shape:[1,24,40,128]
Log: Data dumping to:./MAGIK_DATA_DUMP/490.bin--->shape:[1,24,40,128]
Log: Data dumping to:./MAGIK_DATA_DUMP/493_Quantize.bin--->shape:[1,24,40,128]
Log: Data dumping to:./MAGIK_DATA_DUMP/496_Quantize.bin--->shape:[1,24,40,128]
Log: Data dumping to:./MAGIK_DATA_DUMP/497.bin--->shape:[1,24,40,128]
Log: Data dumping to:./MAGIK_DATA_DUMP/502_0_Quantize.bin--->shape:[1,24,40,128]
Log: Data dumping to:./MAGIK_DATA_DUMP/502_1_Quantize.bin--->shape:[1,24,40,128]
Log: Data dumping to:./MAGIK_DATA_DUMP/502.bin--->shape:[1,24,40,256]
Log: Data dumping to:./MAGIK_DATA_DUMP/505_Quantize.bin--->shape:[1,24,40,256]
Log: Data dumping to:./MAGIK_DATA_DUMP/508_Quantize.bin--->shape:[1,12,20,512]
Log: Data dumping to:./MAGIK_DATA_DUMP/511_Quantize.bin--->shape:[1,12,20,256]
Log: Data dumping to:./MAGIK_DATA_DUMP/512.bin--->shape:[1,12,20,256]
Log: Data dumping to:./MAGIK_DATA_DUMP/513.bin--->shape:[1,12,20,256]
Log: Data dumping to:./MAGIK_DATA_DUMP/514.bin--->shape:[1,12,20,256]
Log: Data dumping to:./MAGIK_DATA_DUMP/515.bin--->shape:[1,12,20,1024]
Log: Data dumping to:./MAGIK_DATA_DUMP/518_Quantize.bin--->shape:[1,12,20,512]
Log: Data dumping to:./MAGIK_DATA_DUMP/521_Quantize.bin--->shape:[1,12,20,256]
Log: Data dumping to:./MAGIK_DATA_DUMP/524_Quantize.bin--->shape:[1,12,20,256]
```

7.1.2 Board Port

(1) In magik-toolkit/Models/post/yolov5s/venus_sample_yolov5s/ , and run：

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.

make build_type=debug

Generate the upper board executable file (venus_yolov5s_bin_uclibc_debug) in the current folder, and copy the file to the board.

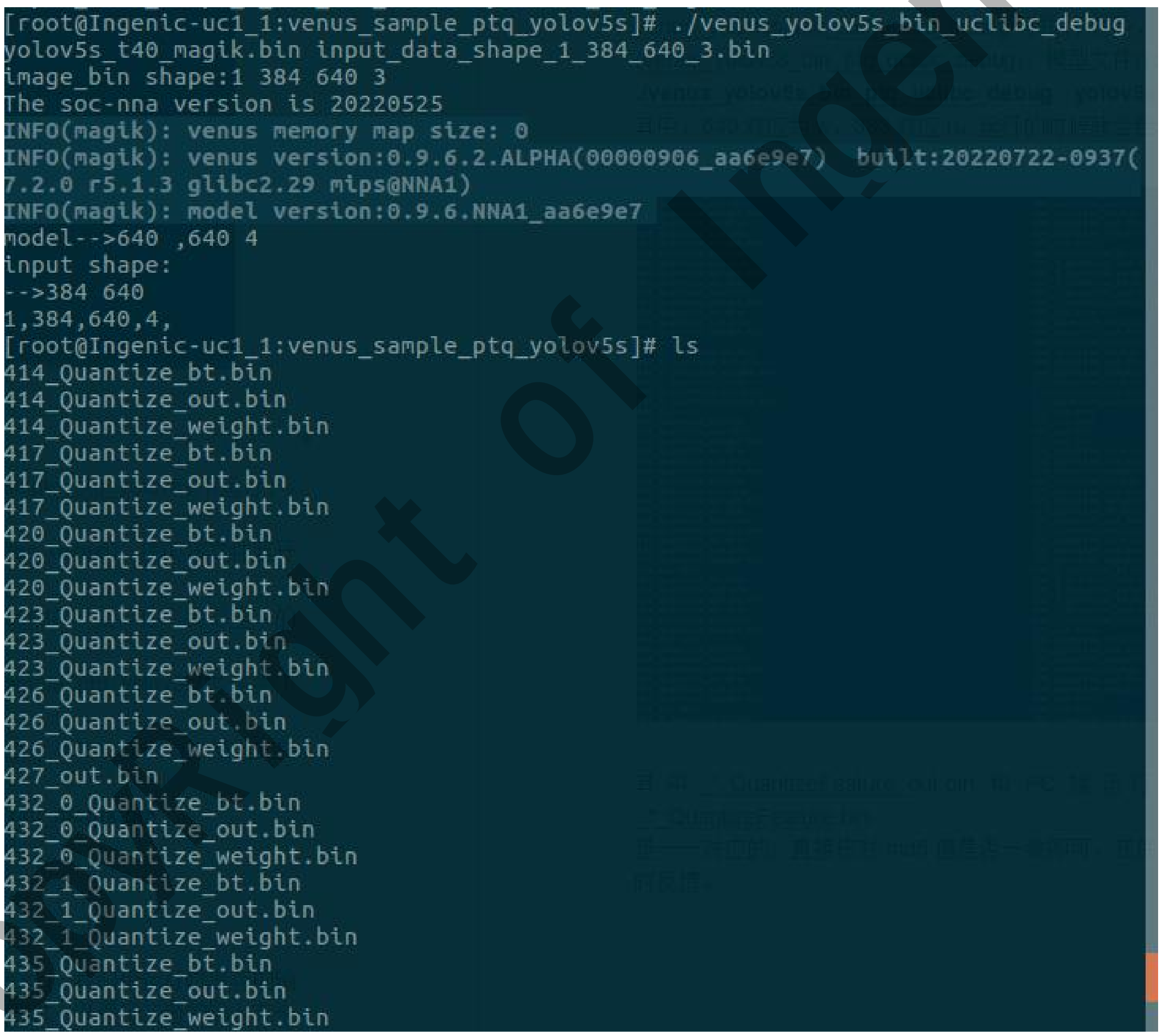
(2) Copy the input_data_shape_1_384_640_3.bin in the MAGIK_DATA_DUMP folder generated after the above PC side runs to the board.

(3) Copy the yolov5s_t40_magik.bin file in magik-toolkit//Models/post/yolov5s/venus_sample_yolov5s/ to the development board, please see step 6.2.

(4) Copy the Venus Library (libvenus.d.so), executable file (venus_yolov5s_bin_uclibc_debug), model file (yolov5s_magik.bin) to the board for operation:

./venus_yolov5s_bin_uclibc_debug yolov5s_t40_magik.bin input_data_shape_1_384_640_3.bin

When running, the output and other information of each layer will be automatically saved (as shown below):



```
[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
```

Among them, _*_QuantizeFeature_out.bin and _*_QuantizeFeature.bin stored in ./MAGIK_DATA_DUMP/ during PC operation are one-to-one corresponding. It is enough to directly check whether the md5sum of value is consistent. Under the condition of ensuring that the input is completely consistent, the middle layer will timely feed back if there is any inconsistency.

Copyright of Ingenic

Copyright® 2005-2022Ingenic Semiconductor Co., Ltd. All rights reserved.