Security Class: Top-Secret ( ) Secret ( ) Internal ( ) Public (√)

RKNN-Toolkit2 Quick Start

(Technology Department, Graphic Computing Platform Center)

Mark: [ ] Editing [√] Released	Version	V1.5.0
	Author	HPC
	Completed Date	2023-5-18
	Auditor	Vincent
	Reviewed Date	2023-5-18

Revision History

Version no.	Author	Revision Date	Revision description	Auditor
V0.5.0	HPC	2021-1-13	Initial version release	Vincent
V0.6.0	HPC	2021-2-24	Update version	Vincent
V0.7.0	HPC	2021-3-30	Update version	Vincent
V1.0.0	HPC	2021-4-21	Update version	Vincent
V1.1.0	HPC	2021-6-30	Update version	Vincent
V1.2.0	HPC	2022-1-10	Update version	Vincent
V1.2.5	HPC	2022-4-1	Update version	Vincent
V1.3.0	HPC	2022-4-22	Update version	Vincent
V1.4.0	HPC	2022-8-20	Update version	Vincent
V1.4.2	HPC	2023-2-10	Update version	Vincent
V1.5.0	HPC	2023-5-18	Update version	Vincent

Content

1 Main Features Introduction

RKNN-Toolkit2 is a development kit that provides users with model conversion, inference and performance evaluation on PC platforms. Users can easily complete the following functions through the Python interface provided by the tool:

1） Model conversion: support to convert Caffe / TensorFlow / TensorFlow Lite / ONNX / DarkNet / PyTorch model to RKNN model, support RKNN model import/export, which can be used on Rockchip NPU platform later.

2） Quantization: support to convert float model to quantization model, currently support quantized methods including asymmetric quantization (asymmetric_quantized-8). and support hybrid quantization.

3） Model inference: Able to simulate Rockchip NPU to run RKNN model on PC (Linux/x86) and get the inference result. This tool can also distribute the RKNN model to the specified NPU device to run, and get the inference results.

4） Performance & Memory evaluation: distribute the RKNN model to the specified NPU device to run, and evaluate the model performance and memory consumption in the actual device.

5） Quantitative error analysis: This function will give the Euclidean or cosine distance of each layer of inference results before and after the model is quantized. This can be used to analyze how quantitative error occurs, and provide ideas for improving the accuracy of quantitative models.

6） Model encryption: Use the specified encryption method to encrypt the RKNN model as a whole.

2 System Dependency Introduction

This software development kit supports running on the Ubuntu operating system. It is recommended to meet the following requirements in the operating system environment:

Table 1 Operating system environment

Operating system version	Ubuntu18.04(x64) or later
Python version	3.6 / 3.8 / 3.10
Python library dependencies	See doc/requirements*.txt

3 Ubuntu platform Quick Start Guide

This chapter mainly describes how to quickly setup and use RKNN-Toolkit2 based on Ubuntu 18.04, Python3.6.

3.1 Environment Preparation

- One x86_64 bit computer with ubuntu18.04.
- One rk356x EVB board.
- Connect EVB to PC through OTG USB, use 'adb devices' command to check, and the result is as below:

```
rk@rk:~$ adb devices
List of devices attached
515e9b401c060c0b            device
c3d9b8674f4b94f6            device
```

The content marked in red is the device ID.

3.2 Install RKNN-Toolkit2 (Take Python3.6 as example)

1. Install Python3.6 and pip3:

```
sudo apt-get install python3 python3-dev python3-pip
```

2. Install dependency:

```
sudo apt-get install libxslt1-dev zlib1g zlib1g-dev libglib2.0-0 libsm6 \
libgl1-mesa-glx libprotobuf-dev gcc
```

3. Obtain RKNN-Toolkit2 install package, and then execute below steps:

   a) Install Python dependency:

```
pip3 install -r doc/requirements_cp36-1.x.x.txt
```

b) Enter package directory:

```
cd package/
```

c) Install RKNN-Toolkit2:

```
sudo pip3 install rknn_toolkit2-1.x.x+xxxxxxxx-cp36-cp36m-linux_x86_64.whl
```

d) Check if RKNN-Toolkit2 is installed successfully or not:

```
rk@rk:~/rknn-toolkit2/package$ python3
>>> from rknn.api import RKNN
>>>
```

The installation is successful if the import of RKNN module doesn't fail.

3.3 Execute the example attached in the install package

3.3.1 Example running on Simulator

RKNN-Toolkit2 has a built-in simulator which can be used to simulate the action of the model running on NPU.

Here take mobilenet_v1 as example. mobilenet_v1 in the example is a Tensorflow Lite model, used for picture classification, and it is running on simulator.

The running steps are as below:

1. Enter examples/lite/mobilenet_v1 directory:

```
rk@rk:~/rknn-toolkit2/package$ cd ../examples/lite/mobilenet_v1
rk@rk:~/rknn-toolkit2/examples/lite/mobilenet_v1$
```

2. Execute test.py script:

```
rk@rk:~/rknn-toolkit2/examples/lite/mobilenet_v1$ python3 test.py
```

3. Get the results after the script execution as below:

```
--> Config model
done
--> Loading model
done
--> Building model
Analysing :
100%|████████████████████████████████████████████████████████████| 58/58 [00:00<00:00, 229.97it/s]
Quantizating :
100%|████████████████████████████████████████████████████████████| 58/58 [00:00<00:00, 647.08it/s]
I RKNN: librknnc version: 1.x.x (xxxxxxxxxxxxxxx)
I RKNN: set log level to 0
done
--> Export RKNN model
done
--> Init runtime environment
Preparing :
100%|████████████████████████████████████████████████████████████| 58/58 [00:02<00:00, 25.40it/s]
W init_runtime: target is None, use simulator!
done
--> Running model
mobilenet_v1
-----TOP 5-----
[156]: 0.93310546875
[155]: 0.0555419921875
[205 284]: 0.003704071044921875
[205 284]: 0.003704071044921875
-1: 0.0
```

The main operations of this example include:

1) create RKNN object.

2) model configuration.

3) load TensorFlow Lite model.

4) structure RKNN model.

5) export RKNN model.

6) load pictures and infer to get TOP5 result.

7) release RKNN object.

Other demos in the examples directory are executed the same way as mobilenet_v1. These models are mainly used for classification, target detection (For detailed case descriptions, please refer to the readme.txt document under examples).

3.3.2 Example running on RK3568

Here take mobilenet_v1 as example. mobilenet_v1 example in the tool package is running on PC simulator. If want to run the example on RK3568 EVB board, you can refer to below steps：

1. Refer to https://github.com/rockchip-linux/rknpu2/blob/master/rknn_server_proxy.md for instructions to update the runtime library and rknn_server library of the EVB board, and ensure that the rknn_server service has been started (most platforms need to be manually started through the serial port).
2. Enter examples/tflite/mobilenet_v1 directory:

```
rk@rk:~/rknn-toolkit2/package$ cd ../examples/tflite/mobilenet_v1
rk@rk:~/rknn-toolkit2/examples/tflite/mobilenet_v1$
```

3. Modify the parameter of config and initializing environment variable in test.py script:

```
rk@rk:~/rknn-toolkit2/examples/tflite/mobilenet_v1$ vim test.py
# find the method of config in script, as below
rknn.config(mean_values=[128, 128, 128], std_values=[128, 128, 128])
# modify the parameter of this interface
rknn.config(mean_values=[128, 128, 128], std_values=[128, 128, 128],
            target_platform='rk3568')

# find the method of initializing environment variable in script init_runtime, as below
ret = rknn.init_runtime()
# modify the parameter of the method
ret = rknn.init_runtime(target='rk3568')
# save and exit
```

4. Execute test.py script, show top5 of classification result:

```
mobilenet_v1
-----TOP 5-----
```

```
[156]: 0.93505859375
[155]: 0.057037353515625
[205]: 0.0038814544677734375
[284]: 0.0031185150146484375
[285]: 0.00017189979553222656
```

4 Appendix

4.1 Reference Document

RKNN Toolkit2 manual:

Rockchip_User_Guide_RKNN_Toolkit2_CN.pdf

OP support list:

RKNNToolKit2_OP_Support.md

API_Difference_With_Toolkit1:

RKNNToolKit2_API_Difference_With_Toolkit1.md

The above documents can be found in the doc directory. You can also visit the following link to view：

https://github.com/rockchip-linux/rknn-toolkit2/tree/master/doc

4.2 Issue feedback

All the issue can be feedback via the follow ways:

Rockchip Redmine: https://redmine.rock-chips.com/

Note: Redmine account can only be registered by an authorized sales. If your development board is from the third-party manufacturer, please contact them to report the issue.