

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

RKNN-Toolkit2 Quick Start

(Technology Department, Graphic Computing Platform Center)

Mark:	Version	V0.7.0
<input type="checkbox"/> Editing	Author	HPC
<input checked="" type="checkbox"/> Released	Completed Date	2021-3-30
	Auditor	Vincent
	Reviewed Date	2021-3-30

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd

(All rights reserved)

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

Content

1 MAIN FEATURES INTRODUCTION.....	4
2 SYSTEM DEPENDENCY INTRODUCTION.....	5
3 UBUNTU PLATFORM QUICK START GUIDE.....	6
3.1 ENVIRONMENT PREPARATION.....	6
3.2 INSTALL RKNN-TOOLKIT2(TAKE PYTHON3.6 AS EXAMPLE).....	6
3.3 EXECUTE THE EXAMPLE ATTACHED IN THE INSTALL PACKAGE.....	7
3.3.1 Simulate the running example on PC.....	7
3.3.2 Example running on rk356x.....	8
4 REFERENCE DOCUMENT.....	8

1 Main Features Introduction

RKNN-Toolkit2 is a development kit that provides users with model conversion, inference and performance evaluation on PC and Rockchip NPU 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, asymmetric_quantized-16) and support hybrid quantization. **Asymmetric_quantized-16 and hybrid quantization not support yet.**
- 3) Model inference: Able to simulate Rockchip NPU to run RKNN model on PC 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 evaluation: Able to simulate Rockchip NPU to run RKNN model on PC, and evaluate model performance (including total time and time-consuming information of each layer). This tool can also distribute the RKNN model to the specified NPU device to run, and evaluate the model performance in the actual device. **Not supported yet.**
- 5) Memory evaluation: Evaluate system and NPU memory consumption at runtime of the model. When using this function, the RKNN model must be distributed to the NPU device to run, and then call the relevant interface to obtain memory information. **Not supported yet.**
- 6) Quantitative error analysis: This function will give the 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.

2 System Dependency Introduction

This software development kit supports running on the Ubuntu (Windows, Mac OS X or Debian not supported yet) 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
Python library dependencies	<div>numpy==1.16.6</div> <div>onnx==1.7.0</div> <div>onnxoptimizer==0.1.0</div> <div>onnxruntime==1.7.0</div> <div>tensorflow==1.14.0</div> <div>tensorboard==1.14.0</div> <div>protobuf==3.12.0</div> <div>torch==1.6.0</div> <div>torchvision==0.7.0</div> <div>mxnet==1.7.0</div> <div>psutil==5.6.2</div> <div>ruamel.yaml==0.15.81</div> <div>scipy==1.2.1</div> <div>tqdm==4.27.0</div> <div>requests==2.21.0</div> <div>tflite==2.3.0</div> <div>opencv-python==4.4.0.46</div> <div>PuLP==2.4</div>

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) Enter package directory:

```
cd package/
```

- b) Install Python dependency

```
pip3 install -r doc/requirements.txt
```

c) Install RKNN-Toolkit2

```
sudo pip3 install rknn_toolkit2*.whl
```

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

```
rk@rk:~/rknn-toolkit2-v0.7.0/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 Simulate the running example on PC

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-v0.7.0/package$ cd ../examples/lite/mobilenet_v1
rk@rk:~/rknn-toolkit2-v0.7.0/examples/lite/mobilenet_v1$
```

2. Execute test.py script

```
rk@rk:~/rknn-toolkit2-v0.7.0/examples/lite/mobilenet_v1$ python3 test.py
```

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

```
--> config model
done
--> Loading model
```

<Rockchip_User_Guide_RKNN_Toolkit2_EN.pdf>.

Rockchip