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

RKNN-Toolkit2 Quick Start

(Technology Department, Graphic Computing Platform Center)

Mark: [ ] Editing [√] Released	Version	V1.1.0
	Author	HPC
	Completed Date	2021-6-30
	Auditor	Vincent
	Reviewed Date	2021-6-30

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

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....9

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 not supported 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: distribute the RKNN model to the specified NPU device to run, and evaluate the model performance in the actual device.

5） Memory evaluation: Evaluate 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.

6） 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.

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	numpy==1.16.6 onnx==1.7.0 onnxoptimizer==0.1.0 onnxruntime==1.6.0 tensorflow==1.14.0 tensorboard==1.14.0 protobuf==3.12.0 torch==1.6.0 torchvision==0.7.0 psutil==5.6.2 ruamel.yaml==0.15.81 scipy==1.2.1 tqdm==4.27.0 requests==2.21.0 tflite==2.3.0 opencv-python==4.4.0.46 PuLP==2.4 scikit_image==0.17.2

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/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/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: x.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]: 0.84521484375
    [155]: 0.08837890625
    [205]: 0.014312744140625
    [284]: 0.0083465576171875
    [194]: 0.00255584716796875
```

The main operations of this example include: create RKNN object, model configuration, load TensorFlow Lite model, structure RKNN model, export RKNN model, load pictures and infer to get TOP5 result, 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.

3.3.2 Example running on rk356x

Not supported yet.

4 Reference Document

For more detailed usage and interface descriptions of RKNN-Toolkit2, please refer to <Rockchip_User_Guide_RKNN_Toolkit2_EN.pdf >.