主讲人 ：正点原子团队
硬件平台：正点原子ATK-DLRV1126开发板

资料下载：www.openedv.com/docs/index.html
教学平台：www.yuanzige.com
天猫店铺：zhengdianyuanzi.tmall.com
技术论坛：www.openedv.com/forum.php
公众平台：正点原子

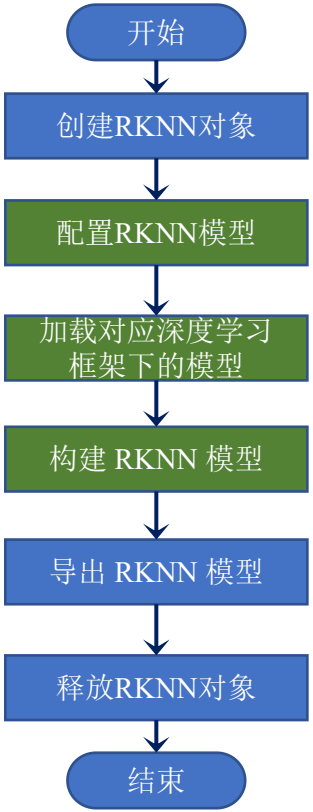
RKNN Toolkit转换模型

1，模型转换流程

2，打印日志

3，转换模型实操

1、模型转换流程



from rknn.api import RKNN
rknn = RKNN()

新数据 =（原数据- 均值）/ 标准差
归一化的值 = (X - mean_values)/std_values

0表示R，1表示G，2表示B
‘0 1 2‘ 表示RGB，'2 1 0‘表示BGR

调用rknn.config()配置模型，主要进行：输入的预处理（如归一化操作、通道顺序的调整）、量化类型选择、量化参数优化算法选择、模型优化等级配置、指定 RKNN 模型目标运行平台等

调用rknn.load_tensorflow()、rknn.load_tflite()、rknn.load_pytorch()、rknn.load_onnx()……等等接口，加载对应深度学习框架下的模型

调用rknn.build ()，根据加载的模型结构及权重参数，构建对应的 RKNN 模型，在这一部分会进行：是否对模型进行量化、确定量化的数据集、是否对模型进行预编译、要推理的批量大小

rknn.export_rknn()

rknn-toolkit V1.7.5支持3种量化类型：非对称量化asymmetric_quantized-u8 → uint8
动态定点量化dynamic_fixed_point-i8 → int8 和 dynamic_fixed_point-i16 → int16
rknn-toolkit2 V1.5.2目前支持asymmetric_quantized-8 → int8

若只是进行模型转换，不需要将RKNN模型发送到板端，则Ubuntu不需要通过ADB来链接开发板

rknn.release()

2、打印日志

在创建RKNN对象时，可以通过设置verbose 和 verbose_file参数来打印日志信息，其中：

① verbose：是否在屏幕上打印日志信息，当verbose为True时，详细的日志信息会输出到屏幕。

② verbose_file：用于指定将日志信息保存到对应的文件中。

- 举例子：

假设verbose=True，且verbose_file=’./mobilenet_build.log’，日志信息被输出到屏幕的同时，也会被记录在当前目录的mobilenet_build.log文件中。

③ 若出现了Error（错误）级别的日志，而 verbose_file被设置为None，那么Error（错误）级别的日志将被自动地写入到 log_feedback_to_the_rknn_toolkit_dev_team.log 文件中。

打印日志参数设置方法如下：

```
# 只在屏幕输出详细的日志信息
rknn = RKNN(verbose=True)

# 在屏幕输出详细的日志信息的同时，也将日志信息写到mobilenet_build.log 文件中
rknn = RKNN(verbose=True, verbose_file='./mobilenet_build.log')
```

3、转换模型实操

① 可参考和运行 rknn-toolkit-1.7.5/examples 或者 rknn-toolkit2-1.5.2 /examples 下的例子

其它：

② TFLite下的模型转换RKNN模型，可参考视频资料里的convert-tflite-to-rknn.py

③ TensorFlow下的模型转换RKNN模型，可参考视频资料里的convert-tf-to-rknn.py

注意 rknn.load_tensorflow()接口要确定模型的输入和输出节点，且输入和输出节点的确定

④ PyTorch下的模型转换RKNN模型，可参考视频资料里的convert-pytorch-to-rknn.py

⑤ ONNX 模型转换为RKNN模型，可参考convert-onnx-to-rknn.py

可将PyTorch 模型转换为TorchScript 或者ONNX 格式以后，再调用对应的API来转换模型为RKNN模型：

rknn.load_pytorch()接口使用TorchScript格式的模型文件；

rknn.load_onnx()接口使用ONNX 格式的模型文件。

PyTorch转换TorchScript 或者ONNX 格式的文件，请参考：https://github.com/ultralytics/yolov5/issues/251

3、转换模型实操

PyTorch转换TorchScript 或者ONNX 格式的文件的操作步骤：

① 按照链接https://github.com/ultralytics/yolov5/issues/251 的指导，在训练的虚拟环境下（需要有训练的GPU，不要在VMware下安装的Ubuntu中操作，否则会报错找不到libcufft.so.11）安装依赖的环境：

```
// 原来的文件默认安装torchvision>=0.8.1，装完后可实现将.pt文件转换为.torchvision格式文件
pip install -r requirements.txt -i https://pypi.tuna.tsinghua.edu.cn/simple
// 若需要将.pt文件转换为.onnx格式文件，仍需要再安装onxx，推荐onnx>=1.9.0
pip install onnx==1.12.0 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

3、转换模型实操

② 在将.pt文件导出为.onnx或者. torchscript格式文件之前，先修改models/yolo.py文件的后处理部分的代码，原来代码：

```
def forward(self, x):
        z = []  # inference output
        for i in range(self.nl):
            x[i] = self.m[i](x[i])  # conv
        #return x[0],x[1],x[2]

            bs, _, ny, nx = x[i].shape  # x(bs,255,20,20) to x(bs,3,20,20,85)
            x[i] = x[i].view(bs, self.na, self.no, ny, nx).permute(0, 1, 3, 4, 2).contiguous()

            if not self.training:  # inference
                if self.dynamic or self.grid[i].shape[2:4] != x[i].shape[2:4]:
                    self.grid[i], self.anchor_grid[i] = self._make_grid(nx, ny, i)

                if isinstance(self, Segment):  # (boxes + masks)
                    xy, wh, conf, mask = x[i].split((2, 2, self.nc + 1, self.no - self.nc - 5), 4)
                    xy = (xy.sigmoid() * 2 + self.grid[i]) * self.stride[i]  # xy
                    wh = (wh.sigmoid() * 2) ** 2 * self.anchor_grid[i]  # wh
                    y = torch.cat((xy, wh, conf.sigmoid(), mask), 4)
                else:  # Detect (boxes only)
                    xy, wh, conf = x[i].sigmoid().split((2, 2, self.nc + 1), 4)
                    xy = (xy * 2 + self.grid[i]) * self.stride[i]  # xy
                    wh = (wh * 2) ** 2 * self.anchor_grid[i]  # wh
                    y = torch.cat((xy, wh, conf), 4)
                z.append(y.view(bs, self.na * nx * ny, self.no))

        return x if self.training else (torch.cat(z, 1),) if self.export else (torch.cat(z, 1), x)
```

3、转换模型实操

② 将def forward(self, x)修改为如下（此处修改只是用于转换模型，若训练模型的话，请把修改的地方还原回原来的代码形式）。此处修改，针对的是YOLOV5的V7.0版本！其它版本不一定是这样子改！此处修改，就是让模型有3个输出（原来的代码是只有一个输出）：

```
def forward(self, x):
        z = []  # inference output
        for i in range(self.nl):
            x[i] = self.m[i](x[i])  # conv
        return x[0],x[1],x[2]
        """
            bs, _, ny, nx = x[i].shape  # x(bs,255,20,20) to x(bs,3,20,20,85)
            x[i] = x[i].view(bs, self.na, self.no, ny, nx).permute(0, 1, 3, 4, 2).contiguous()

            if not self.training:  # inference
                if self.dynamic or self.grid[i].shape[2:4] != x[i].shape[2:4]:
                    self.grid[i], self.anchor_grid[i] = self._make_grid(nx, ny, i)

                if isinstance(self, Segment):  # (boxes + masks)
                    xy, wh, conf, mask = x[i].split((2, 2, self.nc + 1, self.no - self.nc - 5), 4)
                    xy = (xy.sigmoid() * 2 + self.grid[i]) * self.stride[i]  # xy
                    wh = (wh.sigmoid() * 2) ** 2 * self.anchor_grid[i]  # wh
                    y = torch.cat((xy, wh, conf.sigmoid(), mask), 4)
                else:  # Detect (boxes only)
                    xy, wh, conf = x[i].sigmoid().split((2, 2, self.nc + 1), 4)
                    xy = (xy * 2 + self.grid[i]) * self.stride[i]  # xy
                    wh = (wh * 2) ** 2 * self.anchor_grid[i]  # wh
                    y = torch.cat((xy, wh, conf), 4)
                z.append(y.view(bs, self.na * nx * ny, self.no))

        return x if self.training else (torch.cat(z, 1),) if self.export else (torch.cat(z, 1), x)
        """
```

3、转换模型实操

③ 修改好models/yolo.py文件中的def forward(self, x)函数以后，可执行如下命令，将.pt文件导出.onnx

或者. torchscript格式文件：

```
// 转换为TorchScript格式，得到best.torchscript文件
python export.py --weights ./runs/train/exp/weights/best.pt --img 640 --batch 1 --include torchscript
// 转换为ONNX格式，得到best.onnx文件
python export.py --weights ./runs/train/exp/weights/best.pt --img 640 --batch 1 --include onnx
// 若需要指定opset，可根据安装的onnx库的版本来调整，如安装的onnx库的版本是1.12.0，后面加上--opset 12
python export.py --weights ./runs/train/exp/weights/best.pt --img 640 --batch 1 --include onnx --opset 12
```

天猫店铺：https://zhengdianyuanzi.tmall.com