米文动力 SDK 使用说明

V1.2

北京米文动力科技有限公司

2019 年 4 月

目录

文档状态：

[ ]草稿 [ ]正式修订 [√]正式发布	文件标识	米文动力加速 SDK 使用说明
	系统版本	V1.2
	作者	Juns
	完成 日期	2019/04/17

文档修订记录：

文档版本号	修订日期	修订原因	修订人
V1.0	2019/01/04	创建	Juns
V1.1	2019/03/01	添加客户端开启全性能模式的说明	haoran
V1.2	2019/04/17	功能描述性文字修正	haoran

一、内容介绍

本 SDK 提供了如下网络模型的模型加速功能。

网络名称	框架	输入大小
yolov3	darknet	416x416
yolov3 tiny	darknet	416x416
yolov2 tiny	darknet	416x416

Darknet 版本信息

框架地址	git clone https://github.com/AlexeyAB/darknet.git
框架版本	git checkout 2c5e383c04655fe45f3f533eb3a69a80acbf3561

二、使用方法

建议 SDK 使用方法：

在使用 SDK 前，建议执行如下命令开启平台的全性能模式，以获得最佳的 SDK 加速体验。

```
sudo nvpmodel -m 0
sudo ~/jetson_clocks.sh
```

1. 运行 demo

```
cd /opt/miivii/features/miivii-accelerator/
bash bin/demo
```

2. 编译程序

```
cp -r /opt/miivii/features/miivii-accelerator /home/nvidia/
cd /home/nvidia/miivii-accelerator
bash build.sh
```

3. 查看代码

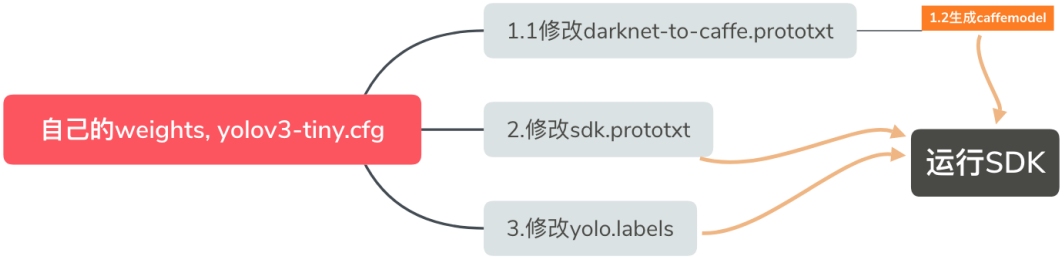
代码在/opt/miivii/features/miivii-accelerator/src 内部

建议查看名字中带有 *min* 的 cpp 文件，如 *yolov3-tiny-min.cpp*，min 代表程序的最小例子。

三、示例：如何使用 SDK 加速自训练的模型

下面以 yolov3-tiny 为例，示例如何使用 SDK 加速自己训练的模型。

基本流程如下图：



1. 准备模型

1) 修改 yolov3-tiny_darknet_to_caffemodel.prototxt

/home/nvidia/miivii-accelerator/networks/yolov3-tiny/yolov3-

tiny_darknet_to_caffe.prototxt

修改这个文件，目的是用于把 weights 转化成对应的 caffemodel。将第 442 行，

和第 567 行的数值，改成与自己 yolov3-tiny.cfg 中最后一个 convolutional 层的

filters 数值相同。

```
436 layer {
437   name: "conv10"
438   type: "Convolution"
439   bottom: "relu9"
440   top: "conv10"
441   convolution_param {
442     num_output: 255
443     kernel_size: 1
444     stride: 1
445     pad: 0
446   }
  447 }
  ...
561 layer {
562   name: "conv13"
563   type: "Convolution"
564   bottom: "relu12"
565   top: "conv13"
566   convolution_param {
567     num_output: 255
568     kernel_size: 1
569     stride: 1
570     pad: 0
571   }
  572 }
```

2) 生成 caffemodel

这里利用上面修改的 yolov3-tiny_darknet_to_caffemodel.prototxt 和自己的

weights 文件，来生成 caffemodel。

假设自己的 weights 为*/home/nvidia/own.weights*，执行以下步骤

cd /home/nvidia/miivii-accelerator/scripts/

替换 model_transfer.sh 的-w 参数为/home/nvidia/own.weights

bash model_transfer.sh

即可生成转换后的模型 yolov3-tiny.caffemodel

2. 修改 yolov3-tiny_SDK.prototxt

修改这个文件，是为了和上面生成的 caffemodel 配合使用，来通过 sdk 达到加速目的。

我们准备了通过 yolov3-tiny 对应的

yolov3-tiny 对应的 prototxt，在

/home/nvidia/miivii-accelerator/networks/yolov3-tiny/yolov3-tiny_SDK.prototxt

将 prototxt 的第 416 行和 527 行，改成与自己 yolov3-tiny.cfg 中最后一个 convolutional 层的 filters 数值相同。

```
410 layer {
411   name: "conv10"
412   type: "Convolution"
413   bottom: "leaky9"
414   top: "conv10"
415   convolution_param {
416     num_output: 255
417     kernel_size: 1
418     stride: 1
419     pad: 0
420   }
421 }
...
526   convolution_param {
```

```
527     num_output: 255
528     kernel_size: 1
529     stride: 1
530     pad: 0
531   }
```

3. 修改 yolo.labels

```
cp /opt/miivii/models/yolo/yolov3-tiny/yolo.labels \
  /home/nvidia/miivii-accelerator/
```

假设我们要识别的类型只有为 dog, cat 两类，则修改后的 yolo.labels 内容为：

```
cat
dog
```

4. 运行 SDK

确保我们拥有了

- 修改过的 *yolov3-tiny_SDK.prototxt*
- 生成的 *yolov3-tiny.caffemodel*
- 修改过的 yolo.labels

运行如下命令

```
/home/nvidia/miivii-accelerator/bin/yolov3-tiny-video\
  /opt/miivii/data/yolov3_1016.mp4 <path to>yolo.labels \
  <path to>yolov3-tiny.caffemodel \
  <path to>yolov3-tiny_SDK.prototxt
```

您的模型已经生效了，享受 Jetson 平台的速度吧。

5. 其他

第一次运行后，会在 caffemodel 路径下面，生成对应的 tensorcache 文件。第二次开始可以参照示例代码，直接使用 tensorcache 文件。可以显著减少初始化时间。

四、 其他示例

连接 gmsl 或者 usb 摄像头的演示：

```
cd /opt/miivii/features/miivii-accelerator/
bin/yolov3-video 0
```

其中 0 为摄像头在系统中的设备名称，通常 gmsl 摄像头的名称为 0 和 1，usb 摄像头为 2.