Darknet yolov2/3 model 如何转bmodel 并在darknet框架内作推理

目前我们提供了yolov2/3 bmodel 在darknet 框架下作推理的demo 实例（<https://github.com/BM1880-BIRD/bm1880-ai-demo-program/tree/master/darknet-yolov2-object-classification-v2>），下面说明一下如何将Darknet 训练出来的yolov2/3 model转为bmodel以方便开发者做yolo 网络的客制化。

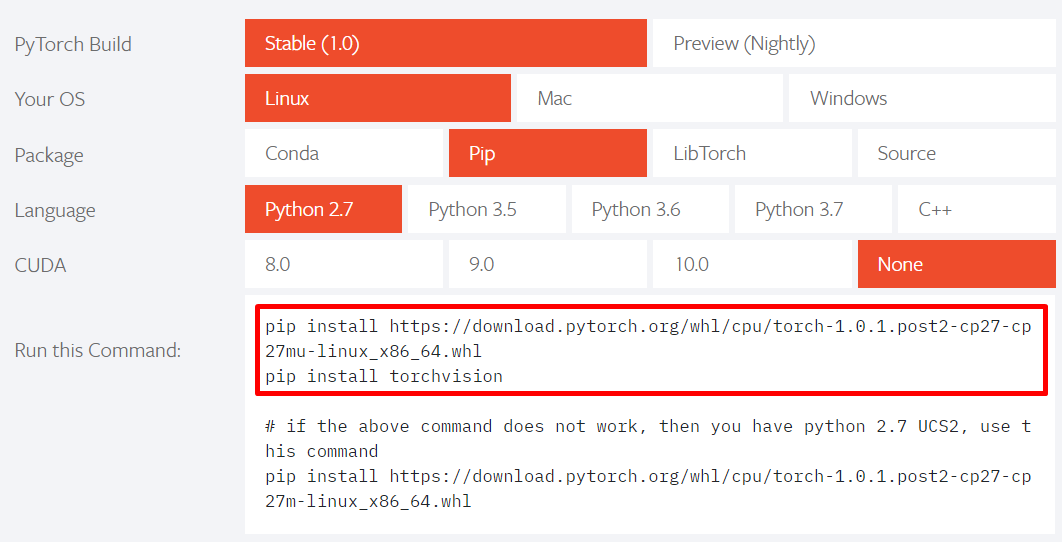
1. Darknet model（cfg 和weights）如何转bmodel.
2. 笔者转换主要用到了下面两个git hub 上的开源Tool .

<https://github.com/ChenYingpeng/caffe-yolov3>

<https://github.com/marvis/pytorch-caffe-darknet-convert>

主要用到了前者中caffe-yolov3/yolov3\_darknet2caffe.py 做yolov3 到 caffe model的转换；用后者内的darknet2caffe.py做yolov2 到 caffe model的转换.

这两个py script 的成功运行需要先安装pytorch 和caffe . pytorch笔者用的是pip 安装方式：<https://pytorch.org/>　按需求选择安装即可。Caffe 的安装这里不详细描述（caffe 需加入对upsample以及reorg两个layer 支持，网上可以自行查找）。



由于darknet2caffe.py目前只支持tiny yolo model 转caffe，少了对reorg和多参数route layer的转换支持。这里笔都加了如下的patch 以支持这两个layer.



笔记将caffe-yolov3/yolov3\_darknet2caffe.py 拷贝到pytorch-caffe-darknet-convert目录下以减化环境变量的设置。

yolov3和yolov2的model 转caffe 分别执行如下命令，会生成caffe 的prototxt和caffemodel.

python yolov3\_darknet2caffe.py yolov3.cfg yolov3.weights　yolov3.prototxt yolov3.caffemodel

python darknet2caffe.py yolov2.cfg yolov2.weights yolov2.prototxt yolov2.caffemodel

**这种转换方式的好处是，不用再另外写caffe 的prototxt ，只需要有darknet 下的cfg 和weights 文件即可。**

换到生成caffemodel ，只需要保留到卷积layer, 如下所示。



这之后是用calibration tool 作量化和转换生成bmodel , 请参考如下的resnet实例。

<https://sophon-edge.gitbook.io/project/toolkit/bmnet-compiler>

**需要注意的是prototxt input layer 的写法，笔者转的是原生model ，这里要做data 归一化 (0.00392156862 = 1/255)(如下示例) ，客制化model 需要与原model train的时候的data 处理保持一样(减均值和归一化等).**



===========================================================

===========================================================

另外，如果有自己实现caffe model (prototxt), 可以用如下的python script 完成caffe model的生成。

执行方式：将yolov2.prototxt 和yolov2.weights放到相同的目录。

执行：　python convert\_weights\_to\_caffemodel\_yolo2.py



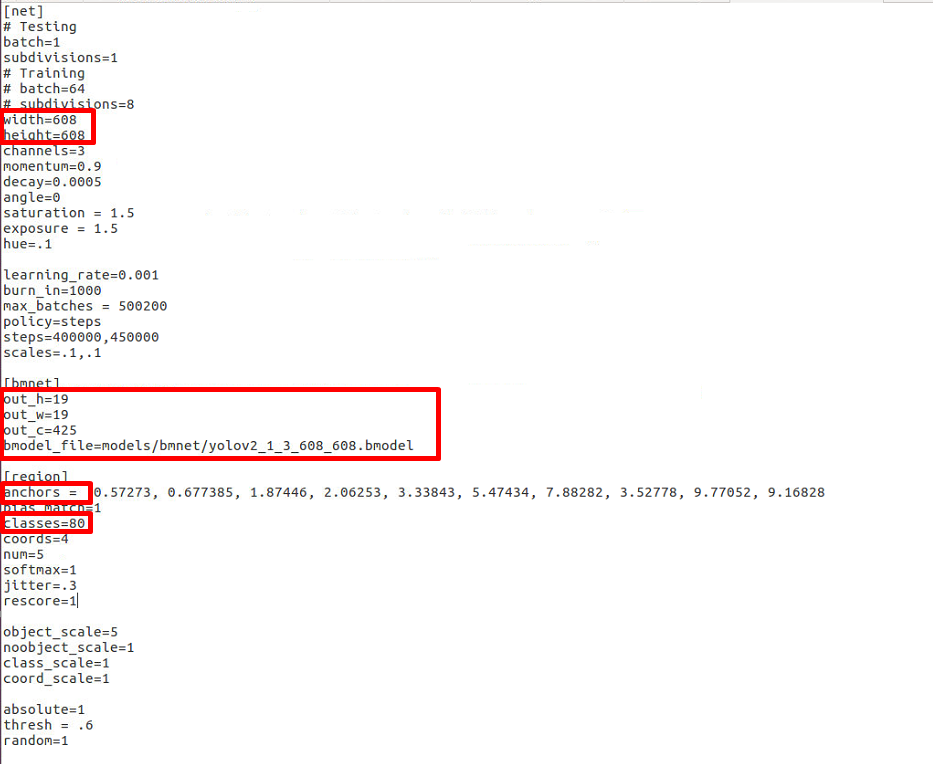
1. 将bmodel 导入到darknet 中做推理
2. 将自己的bmodel 换到下的目录

<https://github.com/BM1880-BIRD/bm1880-ai-demo-program/tree/master/darknet-yolov2-object-classification-v2/darknet/models/bmnet>

1. 更改darknet 运行的cfg 文件
2. Yolov2

需要改动的是下面红框里的内容，需要根据自己的客制化模型参数做对应的修改。

darknet/cfg/bmnet\_yolov2.cfg



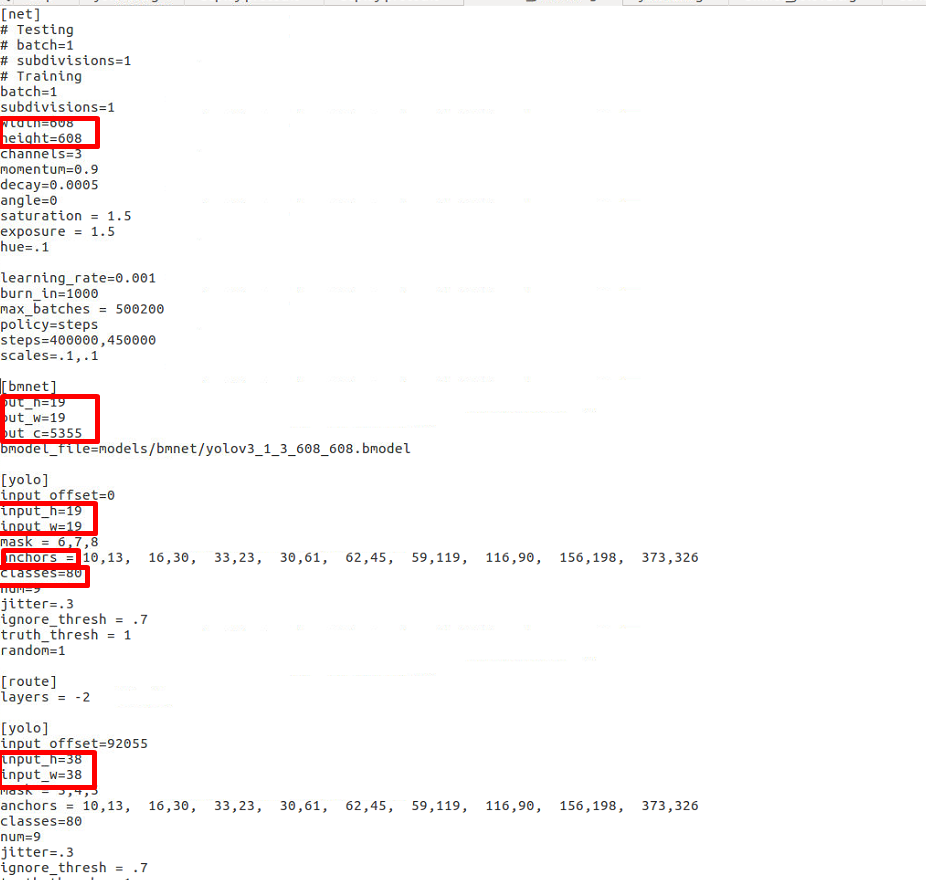
其中, out\_c = out\_c=5\*(4+1+classes), classes 是物体检测的种类数量。

Anchors 值要与darknet training 时一致

1. Yolov3

darknet/cfg/bmnet\_yolov3.cfg

下面红框内的部分需要注意一下修改：



在原生608模型中，out\_c = 19\*19\*255+38\*38\*255+76\*76\*255 /19\*19，其中255=３\*(4+1+80)，80是类别数量。Bmnet model的输出 = 19(out\_h)X19(out\_w)X5355(out\_c)

红框内的值都要依据自己的模型重新填写。