**Makefile**

OPENMP=1 使用OPENMP利用多GPU加速

**如何构建自己的训练数据**

下载数据集标注工具，下载地址：(https://pan.baidu.com/s/1EE52cDStjIxsRgM\_a9pWQQ) (password: 4b2q) 或者 Spire Web.

**修改cfg文件**

o  修改batch为batch=64

o   修改subdivisions为subdivisions=16

o   修改max\_batches为（类别数量\*2000，但不要小于4000），如训练3个类别max\_batches=6000

o   修改steps为max\_batches的0.8与0.9，如steps=4800,5400

o   修改classes=80为自定义数据集的类别数量，主要需要修改3处（3个[yolo]层）

o   修改filters=255为filters=(classes+5)x3，在3个[yolo]层的前一个[convolutional]层

o   如果使用[Gaussian\_yolo]层，修改filters=57为filters=(classes+9)x3，在3个[Gaussian\_yolo]层的前一个[convolutional]层

**训练指令**

1. ./darknet detector train data/obj.data cfg/yolo-obj.cfg yolov4.conv.137

关闭Loss的显示窗口

1. ./darknet detector train data/obj.data cfg/yolo-obj.cfg yolov4.conv.137 -dont\_show
2. （通过浏览器查看训练过程./darknet detector train data/obj.data yolo-obj.cfg yolov4.conv.137 -dont\_show -mjpeg\_port 8090 -map，然后打开Chrome浏览器，输入http://ip-address:8090）
3. （如果需要在训练中计算mAP，每4期计算一次，需要在obj.data文件中设置valid=valid.txt，运行：./darknet detector train data/obj.data yolo-obj.cfg yolov4.conv.137 -map）
4. 如果训练中断，可以选择一个保存的权重继续训练，使用./darknet detector train data/obj.data yolo-obj.cfg backup\yolo-obj\_2000.weights
5. 输出每一类的AP

./darknet detector map data/obj.data cfg/yolo-obj.cfg backup\yolo-obj\_7000.weights

1. 训练中显示mAP的值

./darknet detector train data/obj.data cfg/yolo-obj.cfg yolov4.conv.137 -map

**训练的时候需要注意的是**

1. 在训练中，如果avg（loss）出现nan，则训练出了问题，如果是其他字段出现nan，这种情况是正常的
2. 如果需要改变cfg文件中的width=或height=，新的数字需要被32整除
3. 训练完成后，检测指令为

./darknet detector test data/obj.data yolo-obj.cfg yolo-obj\_8000.weights

1. 如果出现Out of memory，需要修改cfg文件中的subdivisions=16为32或64
2. Loss很大，mAP很低，是不是训练错了？在训练中使用-show\_imgs参数，能够可视化目标框真值，检查数据集是否出了问题。
3. 如果你训练的数据类别需要区分左右目标（如检测左右手，交通信号中的左右方向），则不能使用左右翻转图像增强.flip=0

### **多GPU训练**

首先在1块GPU上训练1000次

./darknet detector train cfg/coco.data cfg/yolov4.cfg yolov4.conv.137

停止训练，使用权重darknet/backup/yolov4\_1000.weights，在多块GPU上训练，运行

./darknet detector train cfg/coco.data cfg/yolov4.cfg /backup/yolov4\_1000.weights -gpus 0,1,2,3

注意：如果出现nan，应该降低学习率，如4块GPUlearning\_rate=0.00065（learning\_rate=0.00261/GPUs），还应该增加cfg文件中的burn\_in=为原先的4x，如burn\_in=4000