前言:  与其他框架不同，Darknet构建网络架构不是通过代码直接堆叠，而是通过解析cfg文件进行生成的。cfg文件格式是有一定规则，虽然比较简单，但是有些地方需要对yolov3有一定程度的熟悉，才能正确设置。本文是【从零开始学习YOLOv3】的第一部分，主要讲最基础的cfg文件内容理解、设置以及总结。

下边以**yolov3.cfg**为例进行讲解。

**1. Net层**

[net]  
*#Testing*  
*#batch=1*  
*#subdivisions=1*  
*#在测试的时候，设置batch=1,subdivisions=1*  
*#Training*  
batch=16  
subdivisions=4  
*#这里的batch与普遍意义上的batch不是一致的。*  
*#训练的过程中将一次性加载16张图片进内存，然后分4次完成前向传播，每次4张。*  
*#经过16张图片的前向传播以后，进行一次反向传播。*  
width=416  
height=416  
channels=3  
*#设置图片进入网络的宽、高和通道个数。*  
*#由于YOLOv3的下采样一般是32倍，所以宽高必须能被32整除。*  
*#多尺度训练选择为32的倍数最小320\*320，最大608\*608。*  
*#长和宽越大，对小目标越好，但是占用显存也会高，需要权衡。*  
momentum=0.9  
*#动量参数影响着梯度下降到最优值的速度。*  
decay=0.0005  
*#权重衰减正则项，防止过拟合。*  
angle=0  
*#数据增强，设置旋转角度。*  
saturation = 1.5  
*#饱和度*  
exposure = 1.5  
*#曝光量*  
hue=.1  
*#色调*  
  
learning\_rate=0.001  
*#学习率:刚开始训练时可以将学习率设置的高一点，而一定轮数之后，将其减小。*  
*#在训练过程中，一般根据训练轮数设置动态变化的学习率。*  
burn\_in=1000  
max\_batches = 500200  
*#最大batch*  
policy=steps  
*#学习率调整的策略，有以下policy：*  
*#constant, steps, exp, poly, step, sig, RANDOM，constant等方式*  
*#调整学习率的policy，*  
*#有如下policy：constant, steps, exp, poly, step, sig, RANDOM。*  
*#steps#比较好理解，按照steps来改变学习率。*  
  
steps=400000,450000  
scales=.1,.1  
*#在达到40000、45000的时候将学习率乘以对应的scale*

**2. 卷积层**

[convolutional]  
batch\_normalize=1   
*#是否做BN操作*  
filters=32   
*#输出特征图的数量*  
size=3   
*#卷积核的尺寸*  
stride=1   
*#做卷积运算的步长*  
pad=1   
*#如果pad为0,padding由padding参数指定。*  
*#如果pad为1，padding大小为size/2，padding应该是对输入图像左边缘拓展的像素数量*  
activation=leaky   
*#激活函数的类型：logistic，loggy，relu，*  
*#elu，relie，plse，hardtan，lhtan，*  
*#linear，ramp，leaky，tanh，stair*  
*# alexeyAB版添加了mish, swish, nrom\_chan等新的激活函数*

feature map计算公式：

**3. 下采样**

可以通过调整卷积层参数进行下采样：

[convolutional]  
batch\_normalize=1  
filters=128  
size=3  
stride=2  
pad=1  
activation=leaky

可以通过带入以上公式，可以得到OutFeature是InFeature的一半。

也可以使用maxpooling进行下采样：

[maxpool]  
size=2  
stride=2

**4. 上采样**

[upsample]  
stride=2

上采样是通过线性插值实现的。

**5. Shortcut和Route层**

[shortcut]  
**from**=-3  
activation=linear  
*#shortcut操作是类似ResNet的跨层连接，参数from是−3，*  
*#意思是shortcut的输出是当前层与先前的倒数第三层相加而得到。*  
*# 通俗来讲就是add操作*  
  
[route]  
layers = -1, 36  
*# 当属性有两个值，就是将上一层和第36层进行concate*  
*#即沿深度的维度连接，这也要求feature map大小是一致的。*  
[route]  
layers = -4  
*#当属性只有一个值时，它会输出由该值索引的网络层的特征图。*  
*#本例子中就是提取从当前倒数第四个层输出*

**6. YOLO层**

[convolutional]  
size=1  
stride=1  
pad=1  
filters=18  
*#每一个[region/yolo]层前的最后一个卷积层中的*  
*#filters=num(yolo层个数)\*(classes+5) ,5的意义是5个坐标，*  
*#代表论文中的tx,ty,tw,th,po*  
*#这里类别个数为1，（1+5）\*3=18*  
activation=linear  
  
[yolo]   
mask = 6,7,8   
*#训练框mask的值是0,1,2，*   
*#这意味着使用第一，第二和第三个anchor*  
anchors = 10,13, 16,30, 33,23, 30,61, 62,45,\  
 59,119, 116,90, 156,198, 373,326  
*# 总共有三个检测层，共计9个anchor*  
*# 这里的anchor是由kmeans聚类算法得到的。*  
classes=1   
*#类别个数*  
num=9   
*#每个grid预测的BoundingBox num/yolo层个数*  
jitter=.3   
*#利用数据抖动产生更多数据，*  
*#属于TTA（Test Time Augmentation）*  
ignore\_thresh = .5  
*# ignore\_thresh 指的是参与计算的IOU阈值大小。*  
*#当预测的检测框与ground true的IOU大于ignore\_thresh的时候，*  
*#不会参与loss的计算，否则，检测框将会参与损失计算。*  
*#目的是控制参与loss计算的检测框的规模，当ignore\_thresh过于大，*  
*#接近于1的时候，那么参与检测框回归loss的个数就会比较少，同时也容易造成过拟合；*  
*#而如果ignore\_thresh设置的过于小，那么参与计算的会数量规模就会很大。*  
*#同时也容易在进行检测框回归的时候造成欠拟合。*  
*#ignore\_thresh 一般选取0.5-0.7之间的一个值*  
*# 小尺度（13\*13）用的是0.7，*  
*# 大尺度（26\*26）用的是0.5。*

**7. 模块总结**

Darket-53结构如下图所示：

它是由重复的类似于ResNet的模块组成的，其下采样是通过卷积来完成的。通过对cfg文件的观察，提出了以下总结：

**不改变feature大小的模块：**

1. 残差模块：

[convolutional]  
batch\_normalize=1  
filters=128  
size=1  
stride=1  
pad=1  
activation=leaky  
  
[convolutional]  
batch\_normalize=1  
filters=256  
size=3  
stride=1  
pad=1  
activation=leaky  
  
[shortcut]  
**from**=-3  
activation=linear

1. 1×1卷积：可以降低计算量

[convolutional]  
batch\_normalize=1  
filters=512  
size=1  
stride=1  
pad=1  
activation=leaky

1. 普通3×3卷积：可以对filter个数进行调整

[convolutional]  
batch\_normalize=1  
filters=1024  
size=3  
stride=1  
pad=1  
activation=leaky

**改变feature map大小**

1. feature map减半：

[maxpool]  
size=2  
stride=2

或者

[convolutional]  
batch\_normalize=1  
filters=128  
size=3  
stride=2  
pad=1  
activation=leaky

1. feature map加倍:

[maxpool]  
size=2  
stride=1

**特征融合操作**

1. 使用Route层获取指定的层（13×13）。
2. 添加卷积层进行学习但不改变feature map大小。
3. 进行上采样（26×26）。
4. 从backbone中找到对应feature map大小的层进行Route或者Shortcut（26×26）。
5. 融合完成。

后记：以上就是笔者之前使用darknet过程中收集和总结的一些经验，掌握以上内容并读懂yolov3论文后，就可以着手运行代码了。目前使用与darknet一致的cfg文件解析的有一些，比如原版Darknet，AlexeyAB版本的Darknet，还有一个pytorch版本的yolov3。AlexeyAB版本的添加了很多新特性，比如 [conv\_lstm], [scale\_channels] SE/ASFF/BiFPN, [local\_avgpool], [sam], [Gaussian\_yolo], [reorg3d] (fixed [reorg]), [batchnorm]等等。而pytorch版本的yolov3可以很方便的添加我们需要的功能。之后我们将会对这个版本进行改进，添加孔洞卷积、SE、CBAM、SK等模块。