cfg/yolov3-voc.cfg文件解读

```
[net]
# Testing
# batch=1
# subdivisions=1
# Training
batch=64
subdivisions=16
width=416
height=416
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches = 50200
policy=steps
steps=40000,45000
scales=.1,.1
```

[net]为特殊的层，配置整个网络

batch: 一批训练样本的样本数量
在测试的时候batch和subdivisions都设置为1

net->batch /= subdivs;
l.batch = net->batch

只有((*net.seen)/net.batch)%net.subdivisions == 0时
才会更新网络参数

动量参数

权重衰减正则项

数据增强参数，通过旋转角度来生成更多训练样本

数据增强参数，通过调整饱和度来生成更多训练样本

数据增强参数，通过调整曝光量来生成更多训练样本

数据增强参数，通过调整色调来生成更多训练样本

```
[net]
# Testing
# batch=1
# subdivisions=1
# Training
batch=64
subdivisions=16
width=416
height=416
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches = 50200
policy=steps
steps=40000,45000
scales=.1,.1
```

学习率决定着权值更新的速度

在迭代次数小于burn_in时，其学习率的更新有一种方式，大于burn_in时，才采用policy的更新方式

训练次数达到max_batches后停止学习

学习率调整的策略：constant, steps, exp, poly, step, sig, RANDOM，constant等方式

steps和scale是设置学习率的变化，迭代到40000次时，学习率衰减10倍，45000次迭代时，学习率又会在前一个学习率的基础上衰减10倍

```
[convolutional]
batch_normalize=1
filters=32
size=3
stride=1
pad=1
activation=leaky

# Downsample

[convolutional]
batch_normalize=1
filters=64
size=3
stride=2
pad=1
activation=leaky
```

[convolutional]卷积层的配置说明

是否进行BN处理，1为是，0为不是

卷积核个数，也是输出通道数

卷积核尺寸

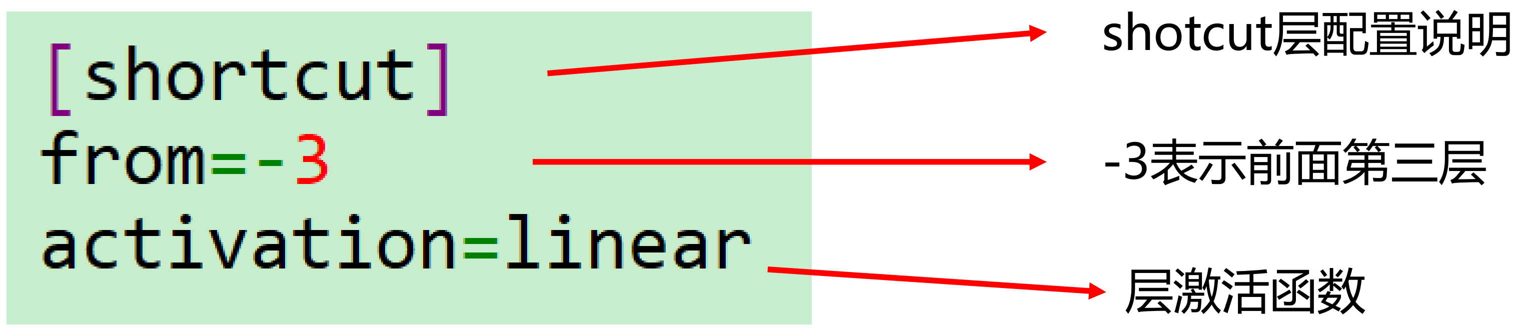
卷积步幅

卷积时是否进行补零padding；padding的个数与卷积核尺寸有关，为size/2向下取整，如3/2=1

★ 卷积核尺寸3×3配合padding且步长为1时，不改变特征图的大小

★ 卷积核尺寸为3×3，配合padding且步长为2时，特征图变为原来的一半大小

网络层激活函数



```
[shortcut]
from=-3
activation=linear
```

A shortcut layer is a skip connection, akin to the one used in ResNet. The from parameter is -3, which means the output of the shortcut layer is obtained by adding feature maps from the previous and the 3rd layer backwards from the shortcut layer.

```
[convolutional]
size=1
stride=1
pad=1
filters=75
activation=linear

[yolo]
mask = 6,7,8
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,  156,198,  373,326
classes=20
num=9
jitter=.3
ignore_thresh = .5
truth_thresh = 1
random=1
```

filters=num×(classes+5)，5的意义是4个坐标加一个置信率

num表示YOLO中每个cell预测的框的个数，YOLOV3中为3

此处的值要根据自己的数据集进行更改，例如识别4个类别，则：

filters=3 ×(4+5)=27。Yolo层前的卷积层的三个fileters都需要修改。

```
[convolutional]
size=1
stride=1
pad=1
filters=75
activation=linear

[yolo]
mask = 6,7,8
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,  156,198,  373,326
classes=20
num=9
jitter=.3
ignore_thresh = .5
truth_thresh = 1
random=1
```

不同尺度上对应的anchor box索引

anchors的大小

目标类别数目

每个grid cell总共预测几个box,和anchors的数量一致。

数据增强手段，此处jitter为随机调整宽高比的范围

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，参与loss的计算，否则，检测框的不参与损失计算

```
[route]
layers = -4

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[upsample]
stride=2

[route]
layers = -1, 61
```

router层可以包含一个或两个值的属性。

- 当属性只有一个值时，它输出由该值索引的图层的特征图。 在我们的示例中，它是-4，因此路由层将route层输出倒数的第4层的特征图。
- 当属性有两个值时，它会返回由其值所索引的层的拼接特征图。 在我们的示例中，它是-1, 61，并且路由层将输出前一层（-1）和第61层的特征图，沿深度维度拼接。

```
[convolutional]
size=1
stride=1
pad=1
filters=75
activation=linear

[yolo]
mask = 3,4,5
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,  156,198,  373,326
classes=20
num=9
jitter=.3
ignore_thresh = .5
truth_thresh = 1
random=1

[route]
layers = -4
```

```
[convolutional]
size=1
stride=1
pad=1
filters=75
activation=linear

[yolo]
mask = 0,1,2
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,  156,198,  373,326
classes=20
num=9
jitter=.3
ignore_thresh = .5
truth_thresh = 1
random=1
```

The anchors describes 9 anchors, but only the anchors which are indexed by attributes of the mask tag are used. Here, the value of mask is 0,1,2, which means the first, second and third anchors are used. This make sense since each cell of the detection layer predicts 3 boxes. In total, we have detection layers at 3 scales, making up for a total of 9 anchors.