









Yolov2 根据自己的数据 训练网络

1. 准备数据

- a) 准备标记好的数据：  
重点是准备 label。

|   |     |                  |         |       |
|---|-----|------------------|---------|-------|
|                                      | 001 | 2015/12/10 6:32  | JPEG 文件 | 10 KB |
| <input checked="" type="checkbox"/>  | 001 | 2015/12/23 10:16 | 文本文档    | 1 KB  |
|                                      | 002 | 2015/12/10 6:32  | JPEG 文件 | 12 KB |
|                                      | 002 | 2015/12/23 10:16 | 文本文档    | 0 KB  |
|                                      | 003 | 2015/12/10 6:32  | JPEG 文件 | 17 KB |
|                                      | 003 | 2015/12/23 10:16 | 文本文档    | 1 KB  |
|                                      | 004 | 2015/12/10 6:32  | JPEG 文件 | 11 KB |
|                                      | 004 | 2015/12/23 10:16 | 文本文档    | 1 KB  |

一张图片对应一个文本文档。

**注意：需要名字相同！**

Label 文本文档格式：

class\_number box1\_x1\_ratio box1\_y1\_ratio box1\_width\_ratio box1\_height\_ratio

class\_number box2\_x1\_ratio box2\_y1\_ratio box2\_width\_ratio box2\_height\_ratio









例如：stop sign



0 0.119691119691 0.54381443299 0.200772200772 0.407216494845

**注意：需要将训练图片以及训练 label 的 txt 文件放到同一个文件夹，如上图。**

函数会找到路径中的图片后缀名.jpg，自动替换成.txt，所以只需要把这些 txt 格式的标记文件复制到图片所在的目录下即可。系统根据替换后的路径地址来读取对应标记文件。

|   |     |                  |         |       |
|---|-----|------------------|---------|-------|
|                                      | 001 | 2015/12/10 6:32  | JPEG 文件 | 10 KB |
| <input checked="" type="checkbox"/>  | 001 | 2015/12/23 10:16 | 文本文档    | 1 KB  |
|                                      | 002 | 2015/12/10 6:32  | JPEG 文件 | 12 KB |
|                                      | 002 | 2015/12/23 10:16 | 文本文档    | 0 KB  |
|                                      | 003 | 2015/12/10 6:32  | JPEG 文件 | 17 KB |
|                                      | 003 | 2015/12/23 10:16 | 文本文档    | 1 KB  |
|                                      | 004 | 2015/12/10 6:32  | JPEG 文件 | 11 KB |
|                                      | 004 | 2015/12/23 10:16 | 文本文档    | 1 KB  |

b) 准备 cfg/xxx.data

例如：

```
classes=1
train = data/train.list
names = /home/zxu /darknet/data/stop.names
backup = /home/zxu /darknet/backup/stop/
```

需要准备的文件：

- train.list 包含所有训练图片的绝对路径

例如：

```
1 /home/zxu/new_darknet/darknet/data/stopsign/001.JPEG
2 /home/zxu/new_darknet/darknet/data/stopsign/002.JPEG
3 /home/zxu/new_darknet/darknet/data/stopsign/003.JPEG
4 /home/zxu/new_darknet/darknet/data/stopsign/004.JPEG
5 /home/zxu/new_darknet/darknet/data/stopsign/005.JPEG
6 /home/zxu/new_darknet/darknet/data/stopsign/006.JPEG
7 /home/zxu/new_darknet/darknet/data/stopsign/007.JPEG
8 /home/zxu/new_darknet/darknet/data/stopsign/008.JPEG
9 /home/zxu/new_darknet/darknet/data/stopsign/009.JPEG
10 /home/zxu/new_darknet/darknet/data/stopsign/010.JPEG
11 /home/zxu/new_darknet/darknet/data/stopsign/011.JPEG
12 /home/zxu/new_darknet/darknet/data/stopsign/012.JPEG
13 /home/zxu/new_darknet/darknet/data/stopsign/013.JPEG
14 /home/zxu/new_darknet/darknet/data/stopsign/014.JPEG
15 /home/zxu/new_darknet/darknet/data/stopsign/015.JPEG
```

- stop.names 是测试时 BOX 上显示的名字，例如：stop

- backup 路径：隔一定迭代数保存或者更新一次权重。

2. 准备网络配置文件 cfg/yolo\_xxx.cfg。

方法：复制 cfg/yolo.cfg 然后微调。

理解配置文件：

|  |                           |
|--|---------------------------|
| [net]  |                           |
| batch=64   | 每 batch 个样本更新一次参数。        |
| subdivisions=8   |                           |
| 如果内存不够大，将 batch 分割为 subdivisions 个子 batch，每个子 batch 的大小为 batch/subdivisions。 |                           |
| 在 darknet 代码中，会将 batch/subdivisions 命名为 batch。                               |                           |
| height=416   | input 图像的高                |
| width=416  | Input 图像的宽                |
| channels=3   | Input 图像的通道数              |
| momentum=0.9   | 动量                        |
| decay=0.0005   | 权重衰减正则项，防止过拟合             |
| angle=0  | 通过旋转角度来生成更多训练样本           |
| saturation = 1.5   | 通过调整饱和度来生成更多训练样本          |
| exposure = 1.5   | 通过调整曝光量来生成更多训练样本          |
| hue=.1   | 通过调整色调来生成更多训练样本           |
| learning rate=0.0001   | 初始学习率                     |
| max batches = 45000  | 训练达到 max batches 后停止学习    |
| policy=steps   | 调整学习率的 policy，有如下 policy: |
| CONSTANT, STEP, EXP, POLY, STEPS, SIG, RANDOM                                |                           |
| steps=100,25000,35000  | 根据 batch num 调整学习率        |
| scales=10,.1,.1  | 学习率变化的比例，累计相乘             |
| [convolutional]  |                           |

```

batch normalize=1          是否做 BN
filters=32                输出多少个特征图
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
activation=leaky           激活函数:
                           logistic, loggy, relu, elu, relie,
plse, hardtan, lhtan, linear, ramp, leaky, tanh, stair

[maxpool]
size=2                    池化层尺寸
stride=2                  池化步进

[convolutional]
batch normalize=1
filters=64
size=3
stride=1
pad=1
activation=leaky

[maxpool]
size=2
stride=2

.....

#####

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

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

[route]                   the route layer is to bring finer
grained features in from earlier in the network
layers=-9

[reorg]                   the reorg layer is to make these
features match the feature map size at the later layer.
                           The end feature map is 13x13, the
feature map from earlier is 26x26x512.
                           The reorg layer maps the 26x26x512
feature map onto a 13x13x2048 feature map
                           so that it can be concatenated
with the feature maps at 13x13 resolution.
stride=2

[route]
layers=-1,-3

[convolutional]
batch normalize=1
size=3
stride=1

```

```

pad=1
filters=1024
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=125
tx,ty,tw,th,to
activation=linear

[region]
anchors = 1.08,1.19, 3.42,4.41, 6.63,11.38, 9.42,5.11,
16.62,10.52

也可以通过 k means 从训练样本中学出
bias_match=1
classes=20
coords=4
num=5
softmax=1
jitter=.2
rescore=1
整 1.delta (预测值与真实值的差)

object scale=5
loss 对总 loss 计算贡献的权重
noobject scale=1
loss 对总 loss 计算贡献的权重
class scale=1
coord scale=1
权重

absolute=1
thresh = .6
random=0
Training, 随机使用不同尺寸的图片进行训练。

```

region 前最后一个卷积层的 filters 数是特定的, 计算公式为  $filter=num*(classes+5)$  5 的意义是 5 个坐标, 论文中的

网络需要识别的物体种类数  
每个 box 的 4 个坐标 tx, ty, tw, th  
每个 grid cell 预测几个 box, 和 anchors  
的数量一致。当想要使用更多 anchors 时需要调大 num, 且如果调大 num 后训练时 Obj 趋近 0 的话可以尝试调大 object scale  
使用 softmax 做激活函数  
通过抖动增加噪声来抑制过拟合  
暂理解为一个开关, 非 0 时通过重打分来调

栅格中有物体时, bbox 的 confidence  
栅格中没有物体时, bbox 的 confidence  
类别 loss 对总 loss 计算贡献的权重  
bbox 坐标预测 loss 对总 loss 计算贡献的权重

random 为 1 时会启用 Multi-Scale Training, 随机使用不同尺寸的图片进行训练。

一般需要修改的部分：

- [Net]的相关参数, e.g. batch, subdivision, 根据内存做调整 learning rate 等.
- 最后一层[convolutional]的 filters= num\*(classes+5)
- [region]中 classes 的值

### 3. 准备初始权重：

- a) 从官网下载网络 cfg 文件对应的权重

curl -O [http://pjreddie.com/media/files/darknet19\\_448.conv.23](http://pjreddie.com/media/files/darknet19_448.conv.23)

- b) 只提取前 23 层的 conv 的参数

```
./darknet partial cfg/darknet19_448.cfg darknet19_448.weights
darknet19_448.conv.23 23
```

### 4. 开始训练

```
./darknet detector train cfg/xxx.data cfg/yolo-xxx.cfg
darknet19_448.conv.23
```

```
Loading weights from backup/yolo_stop.backup...Done!
Learning Rate: 1e-05, Momentum: 0.9, Decay: 0.0005
Resizing
352
Loaded: 0.001235 seconds
Region Avg IOU: 0.799871, Class: 1.000000, Obj: 0.698639, No Obj: 0.005964, Avg Recall: 1.000000, count: 8
Region Avg IOU: 0.779900, Class: 1.000000, Obj: 0.736524, No Obj: 0.005702, Avg Recall: 1.000000, count: 9
Region Avg IOU: 0.739508, Class: 1.000000, Obj: 0.735668, No Obj: 0.007315, Avg Recall: 0.875000, count: 8
Region Avg IOU: 0.702540, Class: 1.000000, Obj: 0.702615, No Obj: 0.005459, Avg Recall: 0.875000, count: 8
28001: 1.115829, 1.115829 avg, 0.000010 rate, 2.736555 seconds, 896032 images
Loaded: 0.000049 seconds
Region Avg IOU: 0.845248, Class: 1.000000, Obj: 0.835385, No Obj: 0.005535, Avg Recall: 1.000000, count: 7
Region Avg IOU: 0.736168, Class: 1.000000, Obj: 0.791480, No Obj: 0.006046, Avg Recall: 1.000000, count: 8
Region Avg IOU: 0.785120, Class: 1.000000, Obj: 0.781473, No Obj: 0.006262, Avg Recall: 0.875000, count: 8
Region Avg IOU: 0.749950, Class: 1.000000, Obj: 0.785686, No Obj: 0.006771, Avg Recall: 1.000000, count: 8
28002: 0.495414, 1.053787 avg, 0.000010 rate, 2.804729 seconds, 896064 images
Loaded: 0.000045 seconds
Region Avg IOU: 0.758670, Class: 1.000000, Obj: 0.665752, No Obj: 0.006581, Avg Recall: 0.833333, count: 6
Region Avg IOU: 0.780089, Class: 1.000000, Obj: 0.684668, No Obj: 0.005054, Avg Recall: 1.000000, count: 9
Region Avg IOU: 0.816498, Class: 1.000000, Obj: 0.844629, No Obj: 0.006122, Avg Recall: 1.000000, count: 7
Region Avg IOU: 0.728449, Class: 1.000000, Obj: 0.610729, No Obj: 0.006037, Avg Recall: 0.833333, count: 6
28003: 0.988208, 1.047230 avg, 0.000010 rate, 2.922716 seconds, 896096 images
Loaded: 0.000046 seconds
Region Avg IOU: 0.706433, Class: 1.000000, Obj: 0.644910, No Obj: 0.006141, Avg Recall: 0.888889, count: 9
Region Avg IOU: 0.750583, Class: 1.000000, Obj: 0.762214, No Obj: 0.007000, Avg Recall: 1.000000, count: 8
Region Avg IOU: 0.706065, Class: 1.000000, Obj: 0.668558, No Obj: 0.005632, Avg Recall: 0.900000, count: 10
Region Avg IOU: 0.834719, Class: 1.000000, Obj: 0.766813, No Obj: 0.006434, Avg Recall: 1.000000, count: 8
28004: 1.013693, 1.043876 avg, 0.000010 rate, 3.001028 seconds, 896128 images
Loaded: 0.000046 seconds
Region Avg IOU: 0.833198, Class: 1.000000, Obj: 0.815037, No Obj: 0.005670, Avg Recall: 1.000000, count: 7
Region Avg IOU: 0.663999, Class: 1.000000, Obj: 0.666338, No Obj: 0.006125, Avg Recall: 0.875000, count: 8
Region Avg IOU: 0.737731, Class: 1.000000, Obj: 0.700281, No Obj: 0.006468, Avg Recall: 0.857143, count: 7
Region Avg IOU: 0.790323, Class: 1.000000, Obj: 0.730600, No Obj: 0.007298, Avg Recall: 1.000000, count: 9
28005: 0.724686, 1.011957 avg, 0.000010 rate, 2.622374 seconds, 896160 images
```