# 深度学习常用算法配置和使用方法

## YOLO配置方法

### 编译

1. 修改Makefile中的参数：

GPU=1

CUDNN=1

OPENCV=1

DEBUG=0

1. 执行make命令进行编译，可以加上j8等参数

### 测试和运行

1. 下载weights权重文件，放置在model目录下
2. 采用如下命令进行测试

./darknet detector test cfg/voc.data cfg/tiny-yolo-voc.cfg model/tiny-yolo-voc.weights data/horses.jpg [-thresh 0.1]

### 数据训练

1. 数据准备

将已有的数据PASCAL VOC数据格式进行保存,然后运行/scripts/voc\_label.py，会在labels文件夹下生成符合yolo格式的标签文件。

1. 训练命令

./darknet detector train ./train/voc.data ./train/tiny-yolo-voc.cfg

./darknet detector train 训练参数 网络配置 [预训练的权重]

其中训练参数参考如下

classes= 1 #类别数

train = /path/to/train\_list.txt #训练图片列表

valid = /path/to/valid\_list.txt #验证图片列表

names = /path/to/class\_name.list #类别名列表

backup = /path/to/backup #训练过程中输出的权重保存目录

图片列表文件列出所有的训练图片的绝对路径，比如

/home/hik/workspace/justin/darknet/darknet-master/train/VOCdevkit/VOCInriaPed/JPEGImages/crop001001.png

同时在该目录下，需要有同名的txt对图片的目标位置信息进行标记

0 0.146875 0.592708333333 0.059375 0.277083333333

0 0.19296875 0.555208333333 0.0671875 0.297916666667

0 0.29765625 0.560416666667 0.0609375 0.304166666667

格式为 **<obj\_type> < left> <top>< width>< height>** 并且转化为相对坐标

1. 网络配置修改方法，具体如下所示

[convolutional]

size=1

stride=1

pad=1

**filters=30 #输出的个数=num\*(classes+coords+1) =5\*(1+4+1)**

activation=linear

[region]

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

bias\_match=1

**classes=1 #类别数**

coords=4

num=5

softmax=1

jitter=.2

rescore=1

1. 预训练的权重

可以预训练的权重参数，对网络进行微调

具体命令

./darknet partial 网络配置 权重文件 输出文件 输出层数

其中输出层数用于指定需要导出多少层的数据到文件中。

### FAQ

1. 控制参数，比如训练过程中多久snap一次快照？

答：

1. Anchors 如何配置？

答：

1. 训练日志的具体含义？

答：

1. 如何配置batch\_size？

答：

## SSD配置方法

1. 假设待新增的数据集的名称为dataset\_name = ‘VOCInriaPed’。
2. 将数据按照VOC PASICAL的格式存放在指定目录下
3. 在data/目录下新增一个名字为dataset\_name的目录
4. 在该目录下新增一个labelmap\_voc.prototxt文件用来存在名字的映射表
5. 运行create\_list.py生成训练和验证数据清单，以及测试图像尺寸清单文件
6. 目录切换到caffe目录层，
7. 运行create\_data.sh创建数据库，其中调用了create\_annoset.py，具体参数设置如下

数据根目录 图片列表文件 数据库保存目录 软连接目录地址

据根目录和图列表文件中的文件路径相加构成最终的图片绝对路径

1. 修改ssd\_pascal.py对应的参数，运行程序会在model和jobs两个目录下生成所有的配置文件。但是训练的sh脚本在jobs对应的目录下面

### 训练过程可视化

1. 务必保存训练过程的日志文件
2. 在./tools/extra/下有解析的脚本，包括了：

parse\_log.sh 可用于解析日志文件用，会在运行目录下输出.train和.test结尾的文件，分别存储提出的loss和accuracy结果。

plot\_training\_log.py.example

extract\_seconds.py 训练过程可视化脚本，显示并保存图片

**运行**

**Python**

**./tools/extra/plot\_training\_log.py 7 ./path/to/save.png ./path/to/train.log**

调用格式和参数如下

Usage:

./plot\_training\_log.py chart\_type[0-7] /where/to/save.png /path/to/first.log ...

Notes:

1. Supporting multiple logs.

2. Log file name must end with the lower-cased ".log".

Supported chart types:

0: Test accuracy vs. Iters

1: Test accuracy vs. Seconds

2: Test loss vs. Iters

3: Test loss vs. Seconds

4: Train learning rate vs. Iters

5: Train learning rate vs. Seconds

6: Train loss vs. Iters

7: Train loss vs. Seconds

### FAQ

1. get\_image\_size程序，获取图像大小什么作用?

答：