

YOLOv5训练PASCAL VOC数据集

课程演示环境: Ubuntu

1 安装pytorch

1) 安装Anaconda

Anaconda 是一个用于科学计算的 Python 发行版, 支持 Linux, Mac, Windows, 包含了众多流行的科学计算、数据分析的 Python 包。

1. 先去官方地址下载好对应的安装包 下载地址:<https://www.anaconda.com/download/#linux>
2. 然后安装anaconda

```
bash ~/Downloads/Anaconda3-2020.07-Linux-x86_64.sh
```

anaconda会自动将环境变量添加到PATH里面, 如果后面你发现输出conda提示没有该命令, 那么你需要执行命令 `source ~/.bashrc` 更新环境变量, 就可以正常使用了。如果发现这样还是没用, 那么需要添加环境变量。编辑`~/.bashrc` 文件, 在最后面加上

```
export PATH=/home/bai/anaconda3/bin:$PATH
```

注意: 路径应改为自己机器上的路径

保存退出后执行: `source ~/.bashrc` 再次输入 `conda list` 测试看看, 应该没有问题。

添加Anaconda国内镜像配置

清华TUNA提供了 Anaconda 仓库的镜像, 运行以下三个命令:

```
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/
conda config --set show_channel_urls yes
```

2) 安装pytorch

更新: 如果使用yolov5版本v4.0以上的代码, pytorch1.6改为pytorch1.7

首先为pytorch创建一个anaconda虚拟环境, 环境名字可自己确定, 这里本人使用pytorch1.7作为环境名:

```
conda create -n pytorch1.7 python=3.8
```

安装成功后激活pytorch1.7环境:

```
conda activate pytorch1.7
```

在所创建的pytorch环境下安装pytorch的1.7版本, 执行命令:

```
conda install pytorch torchvision cudatoolkit=10.2 -c pytorch
```

注意：10.2处应为cuda的安装版本号

编辑~/.bashrc 文件，设置使用pytorch1.7环境下的python3.8

```
alias python='/home/bai/anaconda3/envs/pytorch1.7/bin/python3.8'
```

注意：python路径应改为自己机器上的路径

保存退出后执行：`source ~/.bashrc`

该命令将自动回到base环境，再执行 `conda activate pytorch1.7` 到pytorch环境。

2 yolov5项目克隆和安装

1) 克隆yolov5项目

克隆项目到本地

```
git clone https://github.com/ultralytics/yolov5.git
```

或下载yolov5版本v4.0 <https://github.com/ultralytics/yolov5/releases/tag/v4.0>

2) 安装所需库

使用清华镜像源：

在yolov5路径下执行：

```
pip install -r requirements.txt -i https://pypi.tuna.tsinghua.edu.cn/simple
```

注意：simple 不能少，是 https 而不是 http

3) 下载预训练权重文件

下载yolov5s.pt, yolov5m.pt, yolov5l.pt, yolov5x.pt权重文件，并放置在weights文件夹下

百度网盘下载链接：

链接：<https://pan.baidu.com/s/1RztRLwk3Dye9gkmXxG40JA> 提取码：e2q0

更新：如果使用yolov5版本v4.0以上的代码，下载相应的权重

4) 安装测试

测试图片：

在yolov5路径下执行

```
python detect.py --source ./inference/images/ --weights weights/yolov5s.pt --conf 0.4
```

更新：

```
python detect.py --source ./data/images/ --weights weights/yolov5s.pt --conf 0.4
```

3 准备数据集

1) 下载项目文件:

从百度网盘下载到yolov5目录下

- VOCtrainval_06-Nov-2007.tar
- VOCtrainval_11-May-2012.tar
- VOCtest_06-Nov-2007.tar
- get_voc_ubuntu.py

2) 生成训练集和验证集文件

解压数据集

```
tar xf VOCtrainval_11-May-2012.tar
tar xf VOCtrainval_06-Nov-2007.tar
tar xf VOCtest_06-Nov-2007.tar
```

执行python脚本:

```
python get_voc_ubuntu.py
```

在VOCdevkit / VOC2007和VOCdevkit / VOC2012目录下可以看到生成了文件夹labels ;

在yolov5目录下生成了文件2007_train.txt, 2007_val.txt, 2007_test.txt, 2012_train.txt, 2012_val.txt, train.txt, train.all.txt。

在VOC目录下生成了images和labels文件夹;

- labels下的文件是JPEGImages文件夹下每一个图像的yolo格式的标注文件, 这是由annotations的xml标注文件转换来的
- train.txt和2007_test.txt分别给出了yolov5训练集图片和yolov5验证集图片的列表, 含有每个图片的路径和文件名。
- VOC/images文件夹下有train和val文件夹, 分别放置yolov5训练集和验证集图片; VOC/labels文件夹有train和val文件夹, 分别放置yolov5训练集和验证集标签 (yolo格式) 。

4 修改配置文件

1) 新建文件data/voc-new.yaml

可以复制data/voc.yaml再根据自己情况的修改; 可以重新命名如: data/voc-new.yaml

然后修改配置参数

```
# download command/URL (optional)
#download: bash data/scripts/get_voc.sh

# train and val data as 1) directory: path/images/, 2) file: path/images.txt, or
# 3) list: [path1/images/, path2/images/]
train: VOC/images/train/ # 16551 images
val: VOC/images/val/ # 4952 images
```

2) 新建文件models/yolov5s-voc.yaml

可以复制models/yolov5s.yaml再根据自己情况的修改；可以重新命名如：models/yolov5s-voc.yaml

然后修改配置参数

```
# parameters
nc: 20 # number of classes
```

5 训练PASCAL VOC数据集

1) 训练命令

在yolov5路径下执行

```
python train.py --data data/voc-new.yaml --cfg models/yolov5s-voc.yaml --weights
weights/yolov5s.pt --batch-size 16 --epochs 200
```

2) 训练过程可视化：

在yolov5路径下执行

```
tensorboard --logdir=./runs
```

6 测试训练出的网络模型

1) 测试图片

在yolov5路径下执行

```
python detect.py --source ./VOC/images/val/000001.jpg --weights
runs/exp0/weights/best.pt --conf 0.4
```

更新：

```
python detect.py --source ./VOC/images/val/000001.jpg --weights
runs/train/exp/weights/best.pt --conf 0.4
```

2) 性能统计

在yolov5路径下执行

```
python test.py --data data/voc-new.yaml --weights runs/exp0/weights/best.pt --
batch-size 16
```

更新：

```
python test.py --data data/voc-new.yaml --weights
runs/train/exp/weights/best.pt --batch-size 16
```

7 导出ONNX文件

```
pip install onnx>=1.7.0 -i https://pypi.tuna.tsinghua.edu.cn/simple # for ONNX  
export  
pip install coremltools==4.0 -i https://pypi.tuna.tsinghua.edu.cn/simple # for  
CoreML export
```

```
python models/export.py --weights weights/yolov5s.pt --img 640 --batch 1 #  
export at 640x640 with batch size 1
```