YOLOX项目实战训练自己的数据集-Ubuntu

课程演示环境:ubuntu16.04, cuda 10.2, cudnn8.2.0

1 安装软件

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/Anaconda3-2021.05-Linux-x86_64.sh
```

anaconda会自动将环境变量添加到PATH里面,如果后面你发现输入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Aanaconda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
```

安装pytorch
首先创建一个anaconda虚拟环境,环境名字可自己确定,这里本人使用mypytorch作为环境名:

```
conda create -n mypytorch python=3.8

```

安装成功后激活mypytorch环境:

```
conda activate mypytorch
```

在所创建的pytorch环境下安装pytorch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mypytorch环境下的python3.6

```
alias python='/home/bai/anaconda3/envs/mypytorch/bin/python3.6
```

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 mypytorch` 到pytorch环境。

2 YOLOX项目克隆和安装

1) 克隆YOLOX并安装

网址: https://github.com/Megvii-BaseDetection/YOLOX

```
git clone https://github.com/Megvii-BaseDetection/YOLOX.git
```

或者直接下载YOLOX的代码并解压。
在YOLOX目录下执行:

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

2) 安装apex

```
git clone https://github.com/NVIDIA/apex
cd apex
pip install -v --disable-pip-version-check --no-cache-dir --global-option="--cpp_ext" --global-option="--cuda_ext" ./
```

注意:cuda的版本应和cudatoolkit一致

如使用2021/08/19以后的YOLOX仓库代码,安装apex不再需要安装

3) 安装pycocotools

```
pip install cython
git clone https://github.com/cocodataset/cocoapi.git
```

3) 下载预训练权重文件

`下载yolox_s.pth,yolox_m.pth,yolox_l.pth,yolox_x.pth, yolox_darknet53.47.3.pth, yolox_nano.pth, yolox_tiny.pth权重文件,并放置在YOLOX/weights文件夹下`
百度网盘下载链接:
链接: https://pan.baidu.com/s/1mAzybimhxmws5RTivs-pkQ
提取码:vnbp
更新:如使用2021/08/19以后的YOLOX仓库代码,权重文件和legacy权重文件不兼容

也可以在github上面下载预训练文件;

4) 安装测试
测试图片:

```
python tools/demo.py image -n yolox-s -c weights/yolox_s.pth.tar --path
assets/dog.jpg --conf 0.3 --nms 0.65 --tsize 640 --save_result
或
python tools/demo.py image -f exps/default/yolox_s.py -c weights/yolox_s.pth.tar
--path assets/dog.jpg --conf 0.3 --nms 0.65 --tsize 640 --save_result
测试视频
python tools/demo.py video -n yolox-s -c weights/yolox_s.pth.tar
--path
driving.mp4 --conf 0.3 --nms 0.65 --tsize 640 --save_result --device gpu

```

注意:--device cpu和--device gpu可指定所用的设备

3. 标注自己的数据集

1) 安装图像标注工具labelImg

克隆labelImg

```
git clone https://github.com/tzutalin/labelImg.git
```

使用Anaconda安装
到labelImg路径下执行命令

```
conda install pyqt=5
pip install lxml
pyrcc5 -o libs/resources.py resources.qrc
python labelImg.py
```

2) 添加自定义类别

修改文件labelImg/data/predefined_classes.txt

```
ball
messi
trophy
```

3)使用labelImg进行图像标注

4) 使用"精灵标注助手"进行图像标注

网址: http://www.jinglingbiaozhu.com/

4 准备自己的数据集

1) 下载项目文件

从百度网盘下载

- VOCdevkit_bm.zip (下载到YOLOX/datasets目录下并解压)
- testfiles.zip (下载到YOLOX目录下并解压)
- split_voc.py (下载到YOLOX目录下)

2)解压建立或自行建立数据集

使用PASCAL VOC数据集的目录结构:

- 建立文件夹层次为 VOCdevkit / VOC2007
- VOC2007下面建立三个文件夹:Annotations,JPEGImages和ImageSets/Main
- JPEGImages放所有的训练和测试图片;Annotations放所有的xml标记文件;ImageSets/Main下存放
  训练集、验证集、测试集划分文件(目前为空)

3)划分训练集和测试集

执行python脚本:

```
python split_voc.pyl

```

mageSets/Main目录下可以看到生成四个文件

- train.txt给出了训练集图片文件的列表(不含文件名后缀)
- val.txt给出了验证集图片文件的列表
- test.txt给出了测试集图片文件的列表
- trainval.txt给出了训练集和验证集图片文件的列表

5 修改配置文件

1) 修改文件exps/example/yolox_voc/yolox_voc_s.py

可以复制原文件再根据自己情况的修改;可以重新命名如:yolox_voc_s_bm.py
然后,修改类别数

```
self.num_classes = 20
```

和

```
image_sets=[('2007', 'trainval'), ('2012', 'trainval')],
```

注意:
`yolox-s`

```
self.depth = 0.33
self.width = 0.50
```

`yolox-m`

```
self.depth = 0.67
self.width = 0.75
```

`yolox-l`

```
self.depth = 1.0
self.width = 1.0
```

`yolox-x`

```
self.depth = 1.33
self.width = 1.25
```

注:yolox-s, yolox-m, yolox-l, yolox-x是从小到大的网络模型
其它模型的文件可在YOLOX/exps/default下找到

2) 修改文件voc_classes.py和coco_classes.py

在YOLOX/yolox/data/datasets文件夹下的两个文件,修改类别名称列表

6 训练自己的数据集

1) 训练命令

在YOLOX路径下执行:

```
python tools/train.py -f exps/example/yolox_voc/yolox_voc_s_bm.py -d 1 -b 16 --
fp16 -o -c weights/yolox_s.pth
```

注意:如果出现显存溢出,可减小batch-size

2) 训练过程可视化

在YOLOX路径下执行:

```
tensorboard --logdir=./YOLOX_outputs/yolox_voc_s_bm
```

默认训练的epoches为300, 在文件yolox/exp/yolox_base.py中定义

3) 训练结果的查看

查看YOLOX_outputs/yolox_voc_s_bm目录下的文件

1) 测试图片

```
python tools/demo.py image -f exps/example/yolox_voc/yolox_voc_s_bm.py -c
YOLOX_outputs/yolox_voc_s_bm/best_ckpt.pth --path testfiles/img1.jpg --conf 0.3 --nms 0.65 --tsize
640 --save_result --device gpu
```

批量测试图片:

```
python tools/demo.py image -f exps/example/yolox_voc/yolox_voc_s_bm.py -c
YOLOX_outputs/yolox_voc_s_bm/best_ckpt.pth --path testfiles --conf 0.3 --nms 0.65
--tsize 640 --save_result --device gpu
```

2)测试视频

```
python tools/demo.py video -f exps/example/yolox_voc/yolox_voc_s_bm.py -c
YOLOX_outputs/yolox_voc_s_bm/best_ckpt.pth --path testfiles/messi.mp4 --conf 0.3
--nms 0.65 --tsize 640 --save_result --device gpu
```

3)性能统计

batch testing for fast evaluation:

```
python tools/eval.py -f exps/example/yolox_voc/yolox_voc_s_bm.py -c
YOLOX_outputs/yolox_voc_s_bm/best_ckpt.pth -b 16 -d 1 --conf 0.001 --fp16 --fuse
```

For speed test:

```
python tools/eval.py -f exps/example/yolox_voc/yolox_voc_s_bm.py -c
YOLOX_outputs/yolox_voc_s_bm/best_ckpt.pth -b 1 -d 1 --conf 0.001 --fp16 --fuse
```