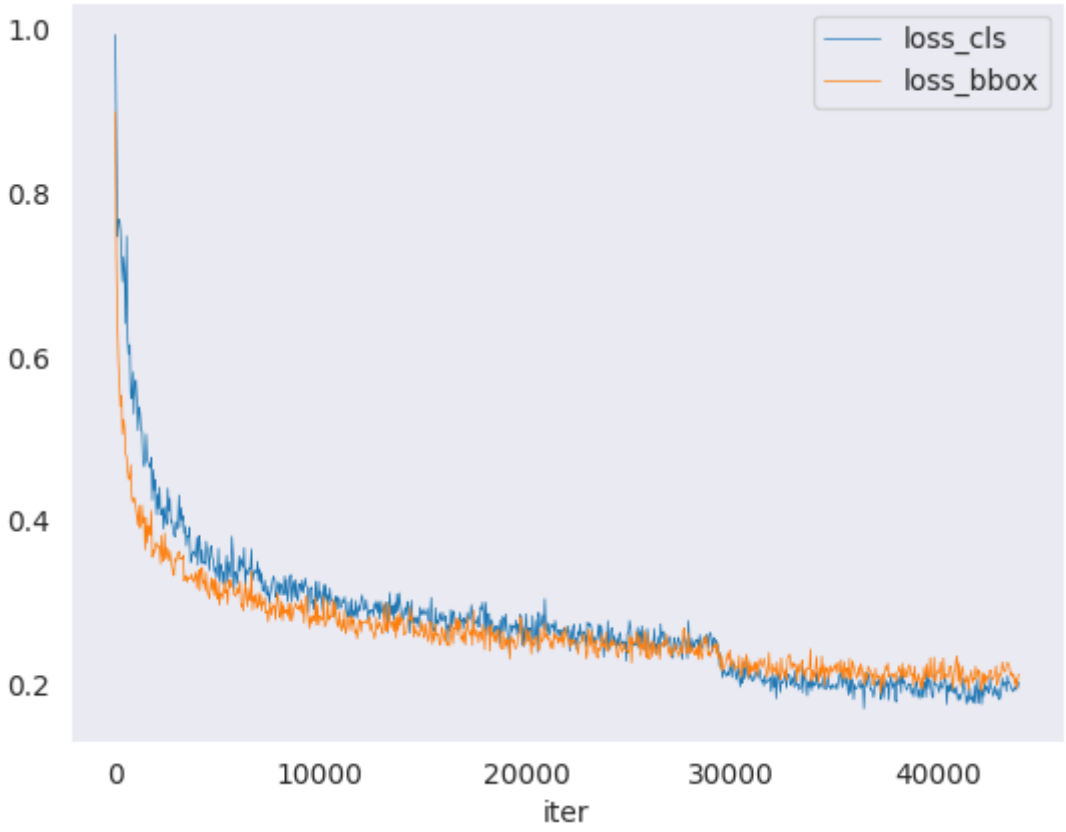
mmdetection2.6 有用的工具

在tools文件夹下， 除了训练与测试的script，也提供了很多的其他的有用工具。

1. 日志分析

`tools/analyze_logs.py` 利用给定的训练日志文件，可以打印出loss函数与map的曲线，运行`pip install serborn`来安装对应的依赖。

```
python tools/analyze_logs.py plot_curve [--keys ${KEYS}] [--title ${TITLE}] [--legend ${LEGEND}] [--backend ${BACKEND}] [--style ${STYLE}] [--out ${OUT_FILE}]
```



Examples:

- 打印分类损失

  ```
  python tools/analyze_logs.py plot_curve log.json --keys loss_cls --legend loss_cls
  ```

- 打印分类与回归的损失，并保存为一个pdf文件

  ```
  python tools/analyze_logs.py plot_curve log.json --keys loss_cls loss_bbox --out losses.pdf
  ```

- 在同一个图中对比两个runs的map

  ```
  python tools/analyze_logs.py plot_curve log1.json log2.json --keys bbox_mAP --legend run1 run2
  ```

- 计算平均训练速度

```
python tools/analyze_logs.py cal_train_time log.json [--include-outliers]
```

训练速度的计算结果输出如下所示：.

```
-----Analyze train time of work_dirs/some_exp/20190611_192040.log.json-----
slowest epoch 11, average time is 1.2024
fastest epoch 1, average time is 1.1909
time std over epochs is 0.0028
average iter time: 1.1959 s/iter
```

2. 可视化

2.1 可视化数据集

`tools/browse_dataset.py` 这个脚本程序，用来帮助使用者来浏览目标检测的数据集，包括图片与bbox，或者保存图片到指定的目录。

```
python tools/browse_dataset.py ${CONFIG} [-h] [--skip-type ${SKIP_TYPE[SKIP_TYPE...]}] [--output-dir ${OUTPUT_DIR}] [--not-show] [--show-interval ${SHOW_INTERVAL}]
```

2.2 可视化模型

首先按照这里的描述转换模型为ONNX，现在仅仅RetainNet支持这种操作，其他的模型将在后来的版本中逐步支持，然后使用Netron工具来实现可视化。

2.3 可视化预测结果

如果需要轻量化的GUI来可视化检测结果，可以参考DetVisGUI project。

3. 误差分析

`tools/coco_error_analysis.py` 利用不同的标准分析了每一类的结果，也可以绘图来提供有用的信息。

```
python tools/coco_error_analysis.py ${RESULT} ${OUT_DIR} [-h] [--ann ${ANN}] [--types ${TYPES[TYPES...]}]
```

4. 模型复杂度

`tools/get_flops.py` 是一个利用 flops-counter.pytorch 来计算FLOPS与参数的方法。

```
python tools/get_flops.py ${CONFIG_FILE} [--shape ${INPUT_SHAPE}]
```

得到的最终结果如下图所示：

```
==============================
Input shape: (3, 1280, 800)
Flops: 239.32 GFLOPs
Params: 37.74 M
==============================
```

注意：这个工具依然在实验阶段，并且不能够保证结果完全正确。可以使用这个结果进行简单的比较，但是在论文或者技术报告中，使用这个结果时必须要好好检查一下。

- FLOPS与输入图像的代销相关，然而模型参数没有关系，默认的输入尺寸是（1， 3， 1280, 800）
- 一些自定义的操作例如GN等没有算在FLOPS中。更多信息参考 mmcv.cnn.get_model_complexity_info()
- 两步法的FLOPS依赖于proposal的数目。

5. 模型转换

5.1 mmdetection模型转换为ONNX（实验阶段）

mmdetection提供了一个脚本将模型转换为ONNX。

```
python tools/pytorch2onnx.py ${CONFIG_FILE} ${CHECKPOINT_FILE} --output_file ${ONNX_FILE} [--shape ${INPUT_SHAPE} --verify]
```

现在这个工具依然在实验阶段，好多自定义的操作不支持。

5.2 mmdetection1.x版本的模型转换为mmdetection2.x

`tools/upgrade_model_version.py`更新先前版本的checkpoint权重为新的版本。不保证全部实现。

```
python tools/upgrade_model_version.py ${IN_FILE} ${OUT_FILE} [-h] [--num-classes NUM_CLASSES]
```

5.3 RegNet模型转换为mmdetection

`tools/regnet2mmdet.py`将pycls预训练的regnet模型转换为mmdetection风格的模型。

```
python tools/regnet2mmdet.py ${SRC} ${DST} [-h]
```

5.4 Detectron resnet转换为Pytorch

`tools/detectron2pytorch.py`将原始detectron框架的resnet预训练模型转换为pytorch风格的模型。

```
python tools/detectron2pytorch.py ${SRC} ${DST} ${DEPTH} [-h]
```

5.5 准备一个待发布的模型

`tools/publish_model.py`帮助使用者准备自己的待发布的模型。

在你讲模型传到AWS之前，你可能想要：

- 将模型的权重转换为cpu tensor
- 删除优化器的状态
- 计算checkpoint文件的的哈希值，并且将hash id加到文件名后边

```
python tools/publish_model.py ${INPUT_FILENAME} ${OUTPUT_FILENAME}
```

例子：

```
python tools/publish_model.py work_dirs/faster_rcnn/latest.pth faster_rcnn_r50_fpn_1x_20190801.pth
```

最终的输出名字为`faster_rcnn_r50_fpn_1x_20190801-{hash id}.pth`

6. 数据集转换

`tools/convert_datasets/`提供了工具来转换Cityscapes数据集与Pascal VOC数据集为COCO格式的数据集。

```
python tools/convert_datasets/cityscapes.py ${CITYSCAPES_PATH} [-h] [--img-dir ${IMG_DIR}] [--gt-dir ${GT_DIR}] [-o ${OUT_DIR}] [--nproc ${NPROC}]
python tools/convert_datasets/pascal_voc.py ${DEVKIT_PATH} [-h] [-o ${OUT_DIR}]
```

7. 其他

7.1 评估标准

`tools/eval_metric.py`按照config文件评估某个结果pickle文件。

```
python tools/eval_metric.py ${CONFIG} ${PKL_RESULTS} [-h] [--format-only] [--eval ${EVAL[EVAL ...]}]
                            [--cfg-options ${CFG_OPTIONS [CFG_OPTIONS ...]}]
                            [--eval-options ${EVAL_OPTIONS [EVAL_OPTIONS ...]}]
```

7.2 打印整个config文件

`tools/print_config.py` 打印整个config文件，包括import的内容

```
python tools/print_config.py ${CONFIG} [-h] [--options ${OPTIONS [OPTIONS...]}]
```

7.3 测试模型的鲁棒性

参考robustness_benchmarking.md.

专栏所有文章请点击下列文章列表查看：

知乎专栏：小哲AI专栏文章分类索引跳转查看

AI研习社专栏：小哲AI专栏文章分类索引