

程序报告

学号： 3190100923

姓名： 陈志博

一、问题重述

本次实验采用神经网络的方式来进行垃圾分类。实验调用的是 MindSpore 库，这是最佳匹配 Ascend（昇腾）芯片的开源 AI 计算框架，同时支持 CPU 和 GPU 平台。实验以 MobileNetV2+ 垃圾分类数据集为例，旨在给我们介绍如何利用 MindSpore 在 CPU/Gpu 平台上进行对于超参数的微调

二、设计思想

此次试验基本只是要求我们学习调整参数的过程，因此调整参数的策略基本上是先针对每一个超参数做较大范围的尝试，取出表现相对较好的部分在进行较小范围内的微调。

三、代码内容

```
参数：
config = EasyDict({
    "num_classes": 26, # 分类数，即输出层的维度
    "reduction": 'mean', # mean, max, Head 部分池化采用的方式
    "image_height": 224,
    "image_width": 224,
    "batch_size": 5, # 鉴于 CPU 容器性能，太大可能会导致训练卡住
    "eval_batch_size": 10,
    "epochs": 100, # 请尝试修改以提升精度
    "lr_max": 3e-2, # 请尝试修改以提升精度
    "decay_type": 'cosine', # 请尝试修改以提升精度
    "momentum": 0.9, # 请尝试修改以提升精度
    "weight_decay": 7e-7, # 请尝试修改以提升精度
    "dataset_path": "./datasets/5fbdf571c06d3433df85ac65-momodel/garbage_26x100",
    "features_path": "./results/garbage_26x100_features", # 临时目录，保存冻结层 Feature
    Map, 可随时删除
    "class_index": index,
    "save_ckpt_epochs": 1,
    "save_ckpt_path": './results/ckpt_mobilenetv2',
    "pretrained_ckpt": './src_mindspore/mobilenetv2-200_1067_cpu_gpu.ckpt',
    "export_path": './results/mobilenetv2.mindir'
})
```

四、实验结果

交叉验证：准确率 93.4%

```
178806
2020-12-29T10:51:04.917825125Z validating the model...
2020-12-29T10:51:05.232519054Z [WARNING] ME(48:140002331334464,MainProcess):2020-12-29-18:51:05.
32.083 [mindspore/train/model.py:684] CPU cannot support dataset sink mode currently.So the eval
ating process will be performed with dataset non-sink mode.
2020-12-29T10:54:00.90899462Z {'loss': 0.39710878499199553, 'acc': 0.934375}
2020-12-29T10:54:00.965936937Z Chosen checkpoint is mobilenetv2-50.ckpt
2020-12-29T10:54:00.965965774Z training is ok!!!
2020-12-29T10:54:01.214732983Z SYSTEM: Finishing...
```

测试：80.77%

在 130 张 照片上测 试模型	✓	12s	得分: 80.77
------------------------	---	-----	-----------

五、总结

需要设置的参数包括 `lr_max`, `weight_decay`, `batchsize`, `momentum` 以及对于训练所得不同 `epoch` 生成的模型的选择。在尝试的过程中我发现 `batchsize` 在特别小的时候效果特别好，因此我的参数选择是 5；`lr_max` 则是学习律的调整，较大的学习律可以增强随机性，使得训练不会永远陷入局部最优，但过大的学习律则会使得每一次 `epoch` 都过于随机，导致多重 `epoch` 的训练没有什么意义，因此我选择的是 ；`weight_decay` 是正则化项，适当的数值可以防止过拟合。我一开始选择的数值过小，导致虽然每次交叉验证的时候准确率能达到 94% 的准确率，但测试的准确率一直只有 75 左右，相差过大。`Momentum` 经过测试我发现在 0.9 左右时表现最好。

经过此次试验我发现超参数的选择十分的复杂且耗时，而且对于不同模型超参数的选择也都是完全不同的。