

# 程序报告

学号： 上海交通大学 518143910019

姓名：周涵

## 一、问题重述

(简单描述对问题的理解, 从问题中抓住主干, **必填**)

本实验首先以 LeNet5+MINIST 手写数字识别的案例使我们了解了如何使用 Mindspore 进行简单的卷积神经网络的开发和图片分类的训练以及预测。在此基础上, 本实验需要以 MobileNetV2+垃圾分类数据集为例, 通过 Fine-Tune 以提高模型训练后的验证精度, 学会如添加检查点 (Checkpoint) 来保存模型的数据, 并最后通过加载模型 Checkpoint 进行推理。本实验目的是能使得机器准确识别 26 个标签的垃圾类别, 这在近日城市生活的垃圾分类热潮中十分实用, 并且也与我们的日常生活息息相关, 但由于生活垃圾种类繁多且缺乏统一分类标准, 人们在实际操作中常常遇到困难, 因此该实验的实际意义也十分重大, 能够利用技术手段建立准确的分类模型, 改善人们的居住生活。

=====

## 二、设计思想

(所采用的方法, 有无对方法加以改进, 该方法有哪些优化方向 (参数调整, 框架调整, 或者指出方法的局限性和常见问题), 伪代码, 理论结果验证等... **思考题, 非必填**)

### 参数调整:

一开始先将 `weight_decay` 设置为 0, 先确定一个较好的 `learning rate`, 然后再固定该 `learning rate`, 给 `weight_decay` 一个值 (如 0.0001, 0.001....), 然后根据 `validation accuracy`, 将 `weight_decay` 增大或者减小 10 倍 (增减 10 倍是粗调节, 当你确定了 `weight_decay` 合适的数量级后, 比如  $\lambda = 0.0001$ , 再进一步地细调节, 比如调节为 0.0002, 0.0003, 0.0009 之类。)

而一个 `epoch` 指代所有的数据送入网络中完成一次前向计算及反向传播的过程, 1 个 `epoch` 等于使用训练集中的全部样本训练一次可见, 随着 `epoch` 数量的增加, 神经网络中权重更新迭代的次数增多, 曲线从最开始的不拟合状态, 慢慢进入优化拟合状态, 最终进入过拟合。其大小与数据集的多样化程度有关, 多样化程度越强, `epoch` 应该越大。

=====

## 三、代码内容

(能体现解题思路的主要代码, 有多个文件或模块可用多个 "=====" 隔开, **必填**)

=====

### 1. 提高精度:

3. Notebook 模型调整参数部分, 你可以根据自己模型需求修改、增加、删除、完善部分超参数  
训练超参

```
onfig = 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": 24, # 鉴于CPU容器性能, 太大可能会导致训练卡住
    "eval_batch_size": 10,
    "epochs": 40, # 请尝试修改以提升精度
    "lr_max": 0.065, # 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": 0.0001, # 请尝试修改以提升精度
})
```

## 2. 加载模型:

```
# ----- 5. 请加载您最满意的模型 -----
# 首先加载网络模型
backbone = MobileNetV2Backbone()
head = MobileNetV2Head(input_channel=backbone.out_channels, num_classes=config.num_classes, reduction=config.reduction)
network = mobilenet_v2(backbone, head)

# 模型是在 results 文件夹下的模型

model_path = './results/ckpt_mobilenetv2/mobilenetv2-40.ckpt'
load_checkpoint(model_path, net=network)
```

## 3. 模型预测:

```
def predict(image):
    """
    加载模型和模型预测
    主要步骤:
        1. 图片处理, 此处尽量与训练模型数据处理一致
        2. 用加载的模型预测图片的类别
    :param image: OpenCV 读取的图片对象, 数据类型是 np.array, shape (H, W, C)
    :return: string, 模型识别图片的类别,
        包含 'Plastic Bottle', 'Hats', 'Newspaper', 'Cans' 等共 26 个类别
    """

    # ----- 实现图像处理部分的代码 -----
    # 该处是与 Notebook 训练数据预处理一致;
    # 如使用其它方式进行数据处理, 请修改完善该处, 否则影响成绩
    image = cv2.resize(image, (config.image_height, config.image_width))
    image = image_process(image)

    # ----- 实现模型预测部分的代码 -----
    logits = network(image)
    pred = np.argmax(logits.asnumpy(), axis=1)[0]
```

## 四、实验结果

(实验结果, 必填)

1. 训练得出的模型精度为: 92.8%, 用于预测的模型损失率为 0.0028  
加载的 Checkpoint 模型为: **mobilenetv2-40.ckpt**

job-0714115240

X

日志 可视化

```
2021-07-14T03:53:28.101262326Z epoch: 21, time cost: 0.47628307342529297, avg loss: 0.003931762184947729
2021-07-14T03:53:28.710100809Z epoch: 22, time cost: 0.4756746292114258, avg loss: 0.003657605731859803
2021-07-14T03:53:29.327300981Z epoch: 23, time cost: 0.488250732421875, avg loss: 0.0038627709727734327
2021-07-14T03:53:29.971931136Z epoch: 24, time cost: 0.5110726356506348, avg loss: 0.003616147441789508
2021-07-14T03:53:30.584246104Z epoch: 25, time cost: 0.4832775592803955, avg loss: 0.0033488075714558363
2021-07-14T03:53:31.191461111Z epoch: 26, time cost: 0.47878026962280273, avg loss: 0.0034246714785695076
2021-07-14T03:53:31.825153291Z epoch: 27, time cost: 0.5049035549163818, avg loss: 0.0033219854813069105
2021-07-14T03:53:32.421900279Z epoch: 28, time cost: 0.468433141708374, avg loss: 0.0031509315595030785
2021-07-14T03:53:33.022614348Z epoch: 29, time cost: 0.4722716808319092, avg loss: 0.003066999837756157
2021-07-14T03:53:33.621485413Z epoch: 30, time cost: 0.47051262855529785, avg loss: 0.003089873120188713
2021-07-14T03:53:34.21881347Z epoch: 31, time cost: 0.4680633544921875, avg loss: 0.0030782707035541534
2021-07-14T03:53:34.886324797Z epoch: 32, time cost: 0.5384819507598877, avg loss: 0.00300297979726545143
2021-07-14T03:53:35.501620597Z epoch: 33, time cost: 0.4864635467529297, avg loss: 0.0030121600721031427
2021-07-14T03:53:36.112730252Z epoch: 34, time cost: 0.47716736793518066, avg loss: 0.0029078712686896324
2021-07-14T03:53:36.715957163Z epoch: 35, time cost: 0.4740123748779297, avg loss: 0.002906344598159194
2021-07-14T03:53:37.340844333Z epoch: 36, time cost: 0.494856595993042, avg loss: 0.0028622678946703672
2021-07-14T03:53:37.947601648Z epoch: 37, time cost: 0.47791147232055664, avg loss: 0.0028504831716418266
2021-07-14T03:53:38.553012498Z epoch: 38, time cost: 0.4767649173736572, avg loss: 0.0028376709669828415
2021-07-14T03:53:39.153213021Z epoch: 39, time cost: 0.47159600257073535, avg loss: 0.0028241530526129415
2021-07-14T03:53:39.757881074Z epoch: 40, time cost: 0.47594213485717773, avg loss: 0.002815604442730546
2021-07-14T03:53:39.88684267Z validating the model...
2021-07-14T03:53:50.307024342Z {'loss': 0.30268804814105715, 'acc': 0.9278673835125448}
2021-07-14T03:53:50.323324074Z Chosen checkpoint is mobilenetv2-40.ckpt
2021-07-14T03:53:50.323339784Z training is OK!!!
```

## 2. 模型预测返回结果:

```
# 输入图片路径和名称
image_path = './datasets/5fbd571c06d3433df85ac65-momodel/garbage_26x100/val/00_00/00037.jpg'

# 使用 opencv 读取图片
image = cv2.imread(image_path)

# 打印返回结果
print(predict(image))

Plastic Bottle
```

## 五、总结

(自评分析(是否达到目标预期,可能改进的方向,实现过程中遇到的困难,从哪些方面可以提升性能,模型的超参数和框架搜索是否合理等), **思考题, 非必填**)

本次实验达到了预期的目标,尽管模型精度依旧有待提高,但是通过调参的过程以及实验过程让我对深度学习以及卷积神经网络的搭建和训练预测有了深入的了解,也让我对于如何利用 MindSpore 框架有了初步掌握和理解,体会到了其便捷之处。