

实验 1 使用华为云 ModelArts 进行 AI 流程开发

1. 实验的目的， 意义

1.1 实验目的

熟悉 AI 的通用开发流程以及熟悉华为云平台使用。

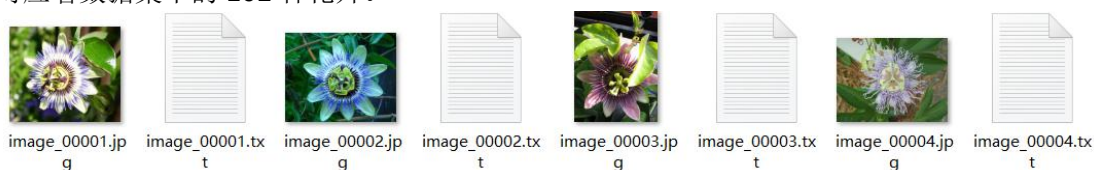
1.2 实验的意义

掌握准备模型所需数据集及运用相关工具处理数据集的方法，熟悉在云平台上训练和部署 AI 模型的一般流程，体会在华为云 ModelArts 平台开发和本地开发的不同之处，为之后的开发打下基础。

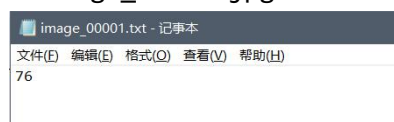
2. 记录自己的实验过程（要有必要的截图以及文字说明）

2.1 准备数据

先准备好实验所需的数据集，将所有花卉图片拷贝到 **dataset** 文件夹中，并为每个样本图片生成标签，标签是命名与对应图片名称相同的 **txt** 文件，标签的取值为 $[0,101]$ ，对应着数据集中的 102 种花卉。



如 **image_00001.jpg** 对应的花卉编号为 76，所以 **image_00001.txt** 的内容为：



2.2 创建桶并上传数据集

先进入华为云的控制台，建立自己的桶，并建立对应的文件夹，用来存放数据集：

桶名称	存储类别	区域	存储用量	Data+	对象数量	创建时间	操作
kdljly1	标准存储	华东-上海一	3.02 GB	该区域暂不支持	14	2020/11/05 14:49:05 G...	修改存储类别 删除

通过 **OBS Browser Plus** 软件将本地准备好的数据集上传到桶的对应文件夹中：



2.3 创建数据集并发布

进入华为云的 ModelArts 管理控制台，创建数据集，指定数据集的输入和输出文件夹：

创建数据集

名称: dataset-exp1

描述: 实验一的数据集

标注类型: 图像分类

数据输入位置: /kdjlyy1/exp1/dataset/

数据输出位置: /kdjlyy1/exp1/dataset_output/

创建成功后会自动进行标注：

名称	标注类型	标注进度 (已标注个数/总数)	待确认个数	创建时间	描述	操作
dataset-exp1	图像分类	6% (400/7169)	--	2020/11/09 21:24:48 ...	实验一的数据集	一键模型上线 发布 数据特征 更多

标注完成后划分训练集验证集的比例并发布：

数据集管理

V001

ID	Pv8wyEGLDpzcLlyMQP
保存时间	2020/11/09 21:45:04 GMT+08:00
版本格式	Default
状态	正常
是否校验	否
切分比例	0.8
描述	--
文件数量	7169
已标注	100% (7169/7169)
存储路径	/kdjlyy1/exp1/dataset_output/dataset-exp1-HkF218QzuuqTUmY54c3/annotation/V001/V001.manifest

设置为当前版本 删除

2.4 订阅模型，创建训练作业和可视化作业

在市场订阅图像分类-ResNet_v1_50 模型，点击创建训练作业，输入之前准备好的数据集和模型输出位置：

一键式参数配置

如您已保存过参数配置，可单击 [这里](#)

算法来源: 9.0.0 - 图像分类-ResNet_v1_50

训练输入: 数据来源: dataset-exp1 (图像分类)

训练输出: 模型输出: /kdjlyy1/exp1/model_output/

创建可视化作业，训练输出位置与模型输出位置相同：

创建可视化作业

[返回可视化作业列表](#)

1 服务选型

2 规格确认

3 完成

[使用指南](#)

[建议反馈](#)

* 作业类型

TensorBoard

* 名称

tensor-5333

* 训练输出位置 ①

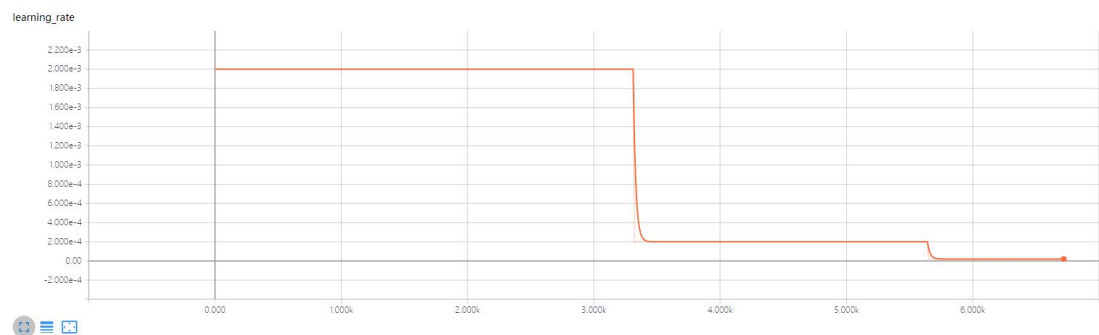
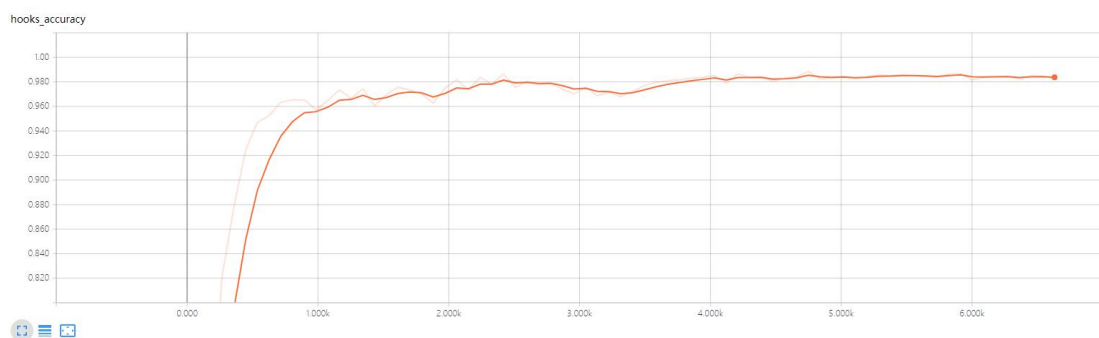
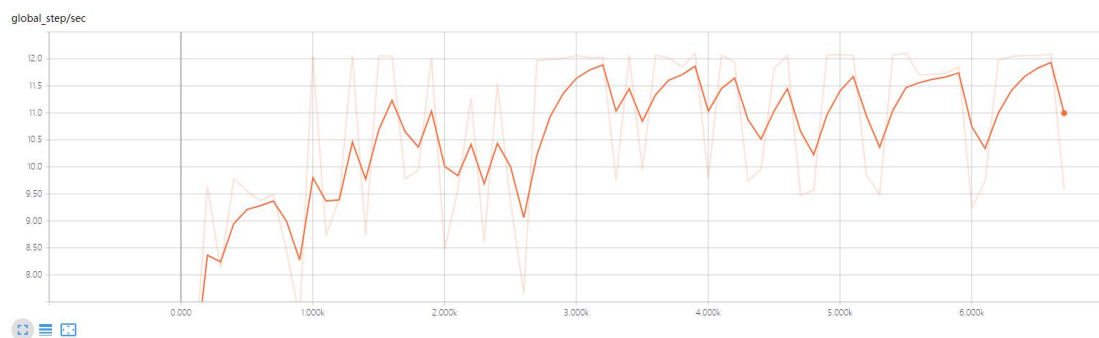
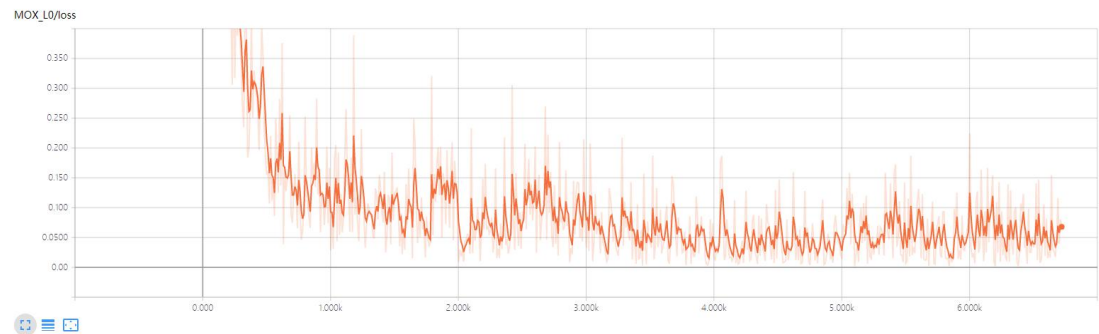
/kdjljy1/exp1/model_output/

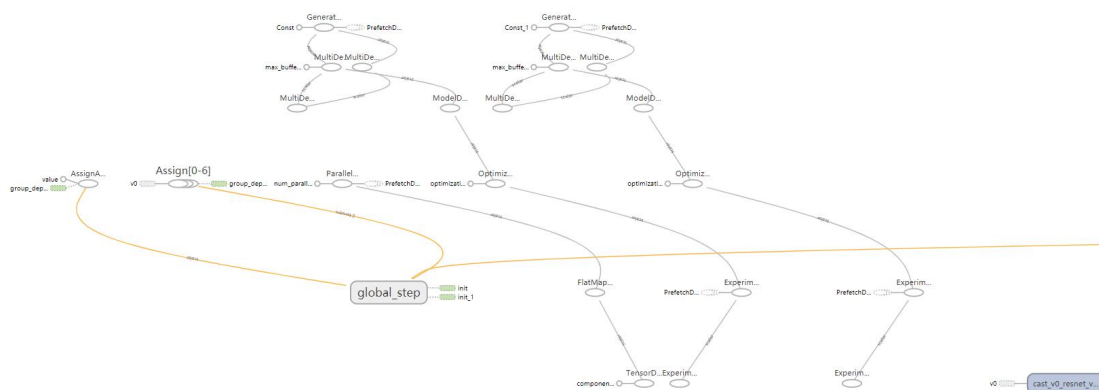
选择

* 计费模式

按需计费

等待模型完成训练，可以在 Tensorboard 上看到模型的各项监控指标和 Tensorflow 可视化的计算图：





2.5 导入模型并部署

模型训练完成后，导入刚才训练好的模型：

★ 元模型来源

从训练中选择 从模板中选择 从容器镜像中选择 从对象存储服务（OBS）中选择

导入ModelArts训练作业中训练完成的模型。请在下方选择需要导入的训练作业及其版本。

★ 选择训练作业 trainjob-d53b ★ 版本 V0001 ★ 模型类型 saved_model

该模型硬件依赖cpu.gpu

★ 部署类型

☒ 在线服务 ☒ 批量服务 ☒ 边缘服务

推理代码 https://kdjlyy1.obs.myhwclouds.com/exp1/model_output/model/customize_service.py

并成功部署：

在线服务 / service-afdb

服务ID: 5006a818-53ce-43db-9dda-6ac573ea58fc

名称: service-afdb

状态: 运行中 (55分钟 后停止)

来源: 我的部署

调用失败次数/总次数: 0/0

网络配置: 未设置

描述: --

个性化配置: 未设置

数据采样: 未设置

同步数据: 同步数据至数据集

推荐筛选: 未设置

使用指南 | 建议反馈 | 修改 | 停止 | 删除

2.6 预测结果

上传测试集的图片，可以在右侧看到用该模型预测的结果：

调用指南 | 预测 | 配置更新记录 | 推荐筛选 | 监控信息 | 事件 | 日志 | 共享

请求路径: / 选择预测图片文件 上传 重新预测 清除预测

预测图片预览

预测结果显示

预测成功

```

1 {
2   "predicted_label": "5",
3   "scores": [
4     [
5       "5",
6       "0.995"
7     ],
8     [
9       "101",
10      "0.001"
11     ],
12     [
13       "90",
14       "0.001"
15     ],
16     [
17       "0",
18       "0.000"
19     ],
20     [
21       "1",
22       "0.000"
23     ]
24   ]
25 }

```



3. 实验过程中碰到的问题，以及你是如何解决的？

3.1 处理数据集的问题。

实验提供的数据集不能直接输入模型，要提前对数据集进行处理：首先将样本集的训练集和验证集的花卉图片拷贝到同一个文件夹下（在 ModelArts 平台训练模型时再划分训练集、验证集），然后利用代码，为每张花卉图片生成对应的标签，写入正确的标签值，完成数据集的准备工作。

3.2 生成可视化作业的问题。

第一次生成可视化作业时，等模型训练完成后，打开可视化作业发现 Tensorboard 上没有模型的各项监控指标和可视化的计算图。通过查看官方的教程发现，可视化作业的训练输出位置应该与模型的训练输出位置保持一致，重新生成可视化作业即可正常显示。

4. 实验验收时所涉及的问题

4.1 在训练模型之前有没有认真查看数据集，是否发现什么特点？

花卉的种类共有 102 种，但是各个种类之间的样本数量差距比较大，部分种类花卉的样本总数在 30 张左右，有些种类的花卉样本数量达到 200 多。数据集的花卉图片特征比较清晰，噪声比较小，无太多干扰项。但是少数种类的花卉在颜色、形状上的个体差异比较大。

4.2 根据本次实验，简述在云平台进行 AI 开发的一般流程。

首先，要准备训练模型所需要的训练集并生成对应的标签，然后将模型上传到云平台的对应文件夹中，再在云平台创建数据集，对数据进行标注。数据集创建完成后需要订阅模型创建作业，输入准备好的数据集和各项超参来训练模型，并可以在 Tensorboard 上查看模型的训练结果和各项参数，最后可以用训练好的模型在测试集上进行测试。

4.3 如果训练出的模型太大，怎样进行模型压缩？

进行模型压缩的方法主要分为两大类：采用新的卷积计算方法和在已训练好的模型上做裁剪。

① 采用新的卷积计算方法：这种方法直接修改网络结构或者使用新的卷积计算方式，从而减少参数，达到压缩模型的效果，例如 SqueezedNet，MobileNet。

② 在已训练好的模型上做裁剪：可以通过剪枝、权值共享、量化、神经网络二值化等方法实现，具体如下：

剪枝：神经网络是由一层一层的节点通过边连接，每个边上会有权重，所谓剪枝，就是当我们发现某些边上的权重很小，可以认为这样的边不重要，进而可以去掉这些边。在训练的过程中，在训练完大模型之后，看看哪些边的权值比较小，把这些边去掉，然后继续训练模型；

权值共享：就是让一些边共用一个权值，达到缩减参数个数的目的。假设相邻两层之间是全连接，每层有 **1000** 个节点，那么这两层之间就有 $1000*1000=100$ 万个权重参数。可以将这一百万个权值做聚类，利用每一类的均值代替这一类中的每个权值大小，这样同属于一类的很多边共享相同的权值，假设把一百万个权值聚成一千类，则可以把参数个数从一百万降到一千个；

量化：一般而言，神经网络模型的参数都是用的 **32bit** 长度的浮点型数表示，实际上不需要保留那么高的精度，可以通过量化，比如用 **0~255** 表示原来 **32** 个 **bit** 所表示的精度，通过牺牲精度来降低每一个权值所需要占用的空间；

神经网络二值化：比量化更为极致的做法就是神经网络二值化，也即将所有的权值不用浮点数表示了，用二进制的数表示，要么是**+1**，要么是**-1**，用二进制的方式表示，原来一个 **32bit** 权值现在只需要一个 **bit** 就可以表示，可以大大减小模型尺寸。