

# 人工智能实验 2-深度学习实现花卉识别

郑海刚

# 1、实验目的

实验的主要目的是加深对神经网络相关知识的理解，如全连接神经网络、卷积神经网络等，熟悉其工作原理和实现。熟悉使用 Tensorflow、PyTorch、MindSpore 三种常用的深度学习框架，掌握深度学习开发环境和方法，了解不同框架的优缺点，能独立使用深度学习框架实现自定义的神经网络。了解产业趋势，掌握国产一站式 AI 开发平台 ModelArts 实现深度学习的一般流程，加深对国产深度学习框架的了解。

# 2、实验内容

与传统图像分类方法不同，使用神经网络无需人工提取特征，可以根据输入图像，自动学习包含丰富语义信息的特征，得到更为全面的图像特征描述，可以很好地表达图像不同类别的信息。

本实验以小组形式完成，组队同实验一。要求使用课程介绍的神经网络的知识，基于给定的数据集实现花卉识别，具体内容如下：

(1) 在本地分别使用 TensorFlow、MindSpore、Pytorch 三种框架，训练神经网络模型实现花卉识别，包括但不限于课程介绍的各种模型。CPU 或 GPU 平台不做限制，操作系统不做限制，框架版本不做限制，开发语言不做限制（主流是 python）。

(2) 在完成(1)的基础上，将本地的实现移植到云端 ModelArts 平台，即在 ModelArts 平台分别使用 TensorFlow、MindSpore、Pytorch 实现模型的训练，训练后的模型可下载到本地进行推理测试或者部署为在线服务实现在线预测图片。

注意：不能使用 model zoo 等定义好的模型或者其他预训练好的模型，要求必须自己一层层实现模型的定义。可以自行探索预训练模型 fine-tune 的效果，在实验报告或者答辩中可以将自定义的模型与其进行对比。模型在测试集上的精度非唯一评分标准，更注重模型设计、理解、实现。

解压实验包在“花卉图像识别”目录可以看到如下子目录：

- 📁 flower\_photos
- 📁 mindspore
- 📁 pytorch
- 📁 tensorflow
- 📁 TestImages
- 📄 readMe.md







flower\_photos 是训练集数据，有 6 种图片，每种图片的数量不等；

TestImages 有 6 张测试图片，每个类别一张；

tensorflow 有模型训练和 modelarts 部署参考代码，lab2\_flower\_classify.ipynb 以记事本的形式，详细说明了数据处理、模型训练、推理预测等过程，需要补充完成模型定义等核心代码，lab2\_flower\_classify.py 是对应 python 代码，需要补充的代码以 TODO 的形式做了标注；

pytorch 和 mindspore 文件夹分别提供了在 modelarts 进行在线部署的参考代码。

训练数据集包含 6 个类别的花卉，同一类别的花卉图片存放在同一个文件夹下，所以总共存放在 6 个文件夹下，每个类别的数量和图片大小都不等，允许对训练集进行各种预处理，但不可将测试集的 6 张图片用做训练，需要保证测试评估用的图片是没有参与过训练的。对于有兴趣探索的同学，可以了解下开源数据集 `oxford_flowers102` (<https://www.robots.ox.ac.uk/~vgg/data/flowers/102/>)，

-  bee\_balm
-  blackberry\_lily
-  blanket\_flower
-  bougainvillea
-  bromelia
-  foxglove

类别	数量
Bee balm (蜂香薄荷)	66
Blackberry lily (黑莓百合花)	48
Blanket flower (天人菊)	49
Bougainvillea (叶子花)	128
Bromelia (凤梨花)	63
Foxglove (毛地黄)	162

## 3、实验原理

该实验可以划分为数据处理、模型构建、模型评估与测试三个主要步骤。其中数据处理包括图像预处理、数据集划分两个部分；模型构建主要包括模型定义、模型训练以及模型保存三个部分；模型评估与测试主要包括读取花卉图像、运行模型推理进行图像特征提取，输出模型识别结果，评估模型质量。

因实验所用的开源环境官方会不定期更新，小版本众多，以最新的官方资料为准。同时因篇幅限制，指导书给出了参考资料，不再对步骤及配置做详细说明。

## 4、本地基于不同框架实现花卉识别

### 4.1 基于 TensorFlow 框架的花卉识别

TensorFlow 是谷歌于 2015 年开源的端到端深度学习框架，源自 Google Brain 内部孵化 DistBelief 项目，在工业界应用广泛。有 1.x 和 2.x 两个大版本，两个版本接口区别比较大，1.x 版本接口较混乱，建议使用 2.x 版本，具体安装、api 介绍、使用请参考 TensorFlow 官方教程：<https://tensorflow.google.cn/>

为了将主要精力集中在模型设计、超参数调整等核心内容，本实验提供基于 TensorFlow2.x 版本的框架代码，只需在此基础上自行完成模型设计、模型实现等步骤，然后进行模型训练和测试，鼓励有兴趣的同学不使用参考代码从零实现。指导书只给出关键说

明,详细代码说明请查看 lab2\_flower\_classify.ipynb 文件,最好使用 jupyter 打开,jupyter 安装方式参考 <https://jupyter.org/install> 或者使用“高性能云计算服务中心”平台的“Jupyterlab 1.1”打开或者使用 ModelArts->开发环境->Notebook 打开。

打开后如下图,可以直接运行其中的代码。lab2\_flower\_classify.py 是对应 python 代码文件。



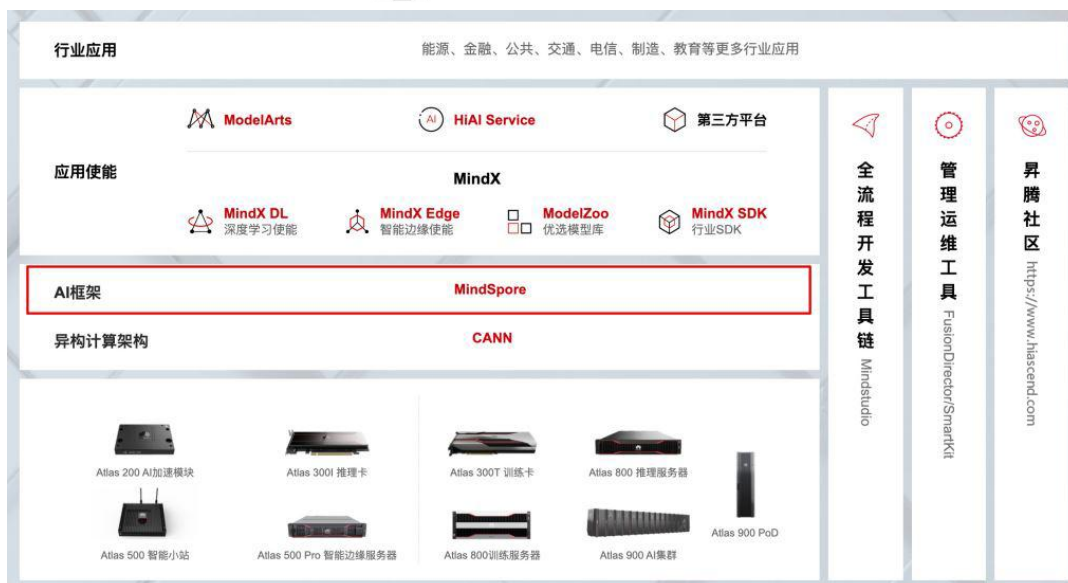
定义模型,训练完成后,本地预测效果如下图所示,可以根据需要显示置信度。



## 4.2 基于 MindSpore 框架的花卉识别

MindSpore (昇思) 是国内华为于 2019 年推出的最佳匹配昇腾 AI 处理器算力的全场景深度学习框架,2020 年正式开源,支持端、边缘、云全场景的计算框架,为数据科学家和算法工程师提供设计友好、运行高效的开发体验,推动人工智能软硬件应用生态繁荣发展。目前 MindSpore 支持在 EulerOS、Ubuntu、Windows 系统上安装,但是在 Windows 上只支持 CPU,Ubuntu 上同时支持 CPU 和 GPU、Ascend910, EulerOS 上只支持 Ascend910。

虽然市面上的 AI 框架众多,但 AI 计算框架发展还远未收敛,例如面向自动微分与张量计算的 AI 即时编译加速技术、面向超大规模神经网络的自动并行技术。华为昇腾 AI 全栈及 MindSpore 所处的位置如下图,MindSpore 框架的优势之一在于更好地适配底层自研 AI 芯片。



在 TensorFlow 实现花卉识别的基础上，按照 MindSpore 的使用方式，替换对应的 api。或者基于开源的 MindSpore 图像分类代码，进行修改完成实验，具体使用和实现不做说明，可参考以下资料。

Mindspore 官网：<https://www.mindspore.cn/>

MindSpore 安装指南：<https://www.mindspore.cn/install>

MindSpore 教程：<https://www.mindspore.cn/tutorials/zh-CN/master/index.html>

MindSpore 实现花卉分类教程：

[https://gitee.com/mindspore/course/blob/master/flowers\\_classification/flowers\\_classification.ipynb](https://gitee.com/mindspore/course/blob/master/flowers_classification/flowers_classification.ipynb)

## 4.3 基于 Pytorch 框架的花卉识别

Pytorch 由 Facebook 人工智能研究院于 2017 年开源的深度学习框架，对于有 Python 编程基础，对 Numpy 熟悉的人，上手非常快，社区资源多，在学术界广泛使用。安装使用请参考：<https://pytorch.org/get-started/locally/>。具体实现请阅读官方手册或开源代码完成<https://pytorch.apachecn.org/#/docs/1.7/06>。

对于想深入学习的，推荐《动手学深度学习-pytorch 版本》<https://zh.d2l.ai/index.html> 及配套视频。

## 4.4 本地计算资源：高性能云服务中心

为了更好的满足实验教学对计算资源的需要，实验与创新实践中心建设了“高性能计算云服务中心”。本地训练模型推荐使用“哈尔滨工业大学（深圳）高性能计算云服务中心”提供的 CPU 或 GPU 资源进行训练，平台地址为 <http://hpc.hitsz.edu.cn/>，使用统一认证登录，具体使用说明请参考 <http://hpc.hitsz.edu.cn/help/>。

因平台资源有限，无法像华为云等公有云平台提供无限资源，为了高效利用平台资源，避免部分同学占着资源不释放导致其他同学无法分配到资源，平台会对使用到的 CPU、GPU、存储进行计费处理，每个账号有默认金额，足以满足实验要求，实验过程请合理使用集群资

源，尽量少的使用虚拟机的方式，更多的使用 slurm 集群共享模式（pytorch on slurm 教程 <https://wiki.hitsz.org/hpc-doc/pytorch/>），会使得计算资源的利用更高效。



## 5、云端 ModelArts 实现花卉识别

### 5.1 ModelArts 介绍

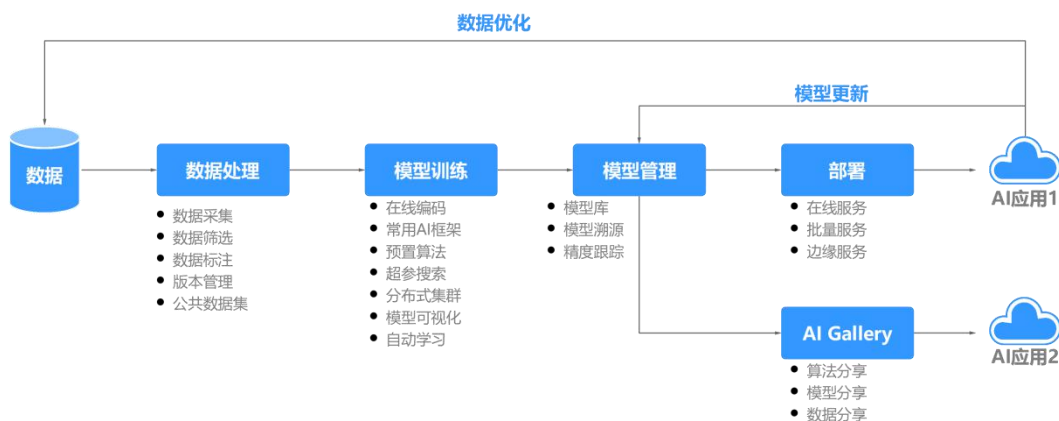
ModelArts 是云端面向 AI 开发者的一站式开发平台，提供海量数据预处理及半自动化标注、大规模分布式训练、自动化模型生成及端-边-云模型按需部署能力，帮助用户快速创建和部署模型，管理全周期 AI 工作流。



“一站式”是指 AI 开发的各个环节，包括数据处理、算法开发、模型训练、模型部署都可以在 ModelArts 上完成。从技术上看，ModelArts 底层支持各种异构计算资源，开发者可以根据需要灵活选择使用，而不需要关心底层的技术。同时，ModelArts 支持 Tensorflow、PyTorch、MindSpore、MXNet 等主流开源的 AI 开发框架，也支持开发者使用自研的算法框



架。



通过 ModelArts 技术结合花卉识别场景的实践，掌握 ModelArts 开发流程，了解 OBS 和 IAM 的基本应用；了解人工智能的应用，通过实践提升 ModelArts 开发的能力。使用 ModelArts 需要在华为云注册，参考《华为账号注册和华为云登录指南》进行账号注册与实名认证。

## 5.2 对象存储 obs

不同于本地训练，数据和模型可以直接通过文件系统保存在磁盘，在云上 ModelArts 上实现模型训练，存储一般使用对象存储。

对象存储是一类重要的存储方式，不同的云厂家有不同的产品，比如亚马逊的 S3、阿里云的 OSS。无论是哪家的产品，它们提供的服务是一样的，区别在于收费、性能、使用方式等。OBS 是一项面向 Internet 访问的服务，提供了基于 HTTP/HTTPS 协议的 Web 服务接口，没有提供 POSIX 接口。这也限制了其访问方式不能像访问本地文件一样便捷。**OBS 不支持编辑，如果要修改文件只能重新上传替换之前的文件。**

华为的**对象存储服务**（Object Storage Service，OBS）是一个基于对象的海量存储服务，为客户提供海量、安全、高可靠、低成本的数据存储能力，没有总数据容量和对象/文件数量的限制，为用户提供了超大存储容量的能力，适合存放任意类型的文件。

华为在 obs 之上开发了并行文件系统，是对象存储服务（OBS）提供的一种经过优化的高性能文件系统，作为对象存储服务的子产品，并行文件系统支持用户按照标准的 OBS 接口读取数据，也支持通过部署在弹性云服务器中的 PFS 客户端（obsfs 工具），按照 POSIX 文件语义读写数据。通过 obsfs 用户可以将创建的并行文件系统挂载到云端 Linux 服务器上，并能像操作本地文件系统一样对并行文件系统内的文件和目录进行在线处理。感兴趣的同学可自行了解。

ModelArts 需要用到 obs 存储数据集、保存模型，Obs 官方教程：

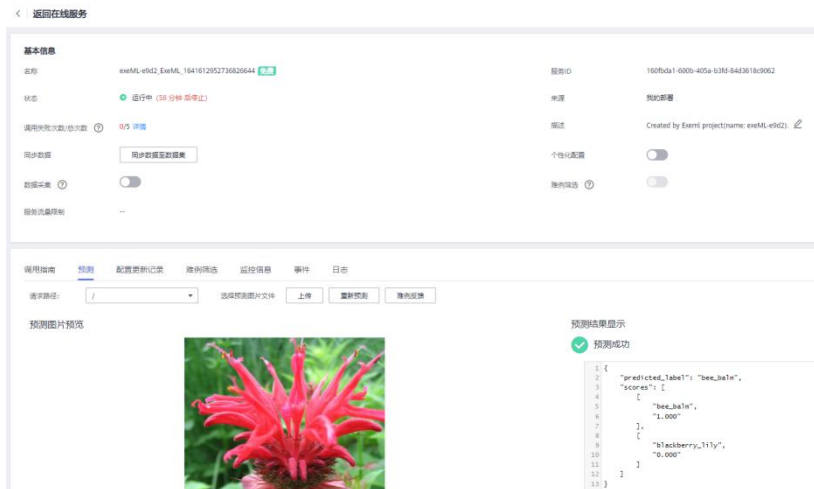
<https://support.huaweicloud.com/obs/index.html>

## 5.3 ModelArts 实现花卉识别

ModelArts 官方教程：<https://support.huaweicloud.com/modelarts/index.html>。

ModelArts 支持使用预置算法、自定义算法和常用框架训练深度学习模型，不同模式适用于不同开发对象和场景。我们需要在本地实现花卉识别的基础上，对代码进行适当改造，

移植到 ModelArts 平台利用云端资源进行训练，并且可以直接部署成在线服务。在线服务效果如下图所示，能够上传图片并成功输出结果。



### 5.3.1 使用预置算法实现

ModelArts 内置了很多现成算法，只需上传数据集，给数据打上标签，就能自动训练、一键部署，无需写任何代码，大大降低了人工智能的门槛。鼓励大家体验，实验不做要求，具体过程参考以下教程或自行在官网查找资料。

使用 AI Gallery 的订阅算法实现花卉识别

[https://support.huaweicloud.com/bestpractice-modelarts/modelarts\\_10\\_0025.html](https://support.huaweicloud.com/bestpractice-modelarts/modelarts_10_0025.html)

### 5.3.2 自定义算法实现

跟本地用 TensorFlow 等框架实现模型训练过程一致，需要自己写训练代码、推理代码，关联 obs 内存储的数据集进行训练。具体使用请先阅读官方指导书或以下参考链接，指导书只对部分配置做了说明。

使用自定义算法构建模型（手写数字识别）

[https://support.huaweicloud.com/bestpractice-modelarts/modelarts\\_10\\_0080.html](https://support.huaweicloud.com/bestpractice-modelarts/modelarts_10_0080.html)

自定义训练算法的训练脚本开发注意事项：

[https://support.huaweicloud.com/engineers-modelarts/modelarts\\_23\\_0240.html](https://support.huaweicloud.com/engineers-modelarts/modelarts_23_0240.html)

以下以 TensorFlow 为例做简单说明，训练代码在本地 TensorFlow 训练的代码上进行修改，具体改动 1 是训练集和模型保存路径按以下方式获取，改动 2 使用 model.save 接口保存模型，可以参考 lab2\_flower\_classify\_tf\_modelarts.py 文件。



```
import argparse
# 创建解析
parser = argparse.ArgumentParser(description="train flower classify",
                                formatter_class=argparse.ArgumentDefaultsHelpFormatter)
# 添加参数
parser.add_argument('--train_url', type=str,
                    help='the path model saved')
parser.add_argument('--data_url', type=str, help='the training data')
# 解析参数
args, unknown = parser.parse_known_args()

path = args.data_url
model_path = args.train_url
print(path)
print(model_path)
# /home/ma-user/modelarts/inputs/data_url_0/
# /home/ma-user/modelarts/outputs/train_url_0/
```

修改完代码后上传训练代码和数据集到 obs 存储，训练代码和数据集需要在同一个桶，可以在同一个目录下也可以不同目录。

创建算法参考配置如下，添加 data\_url 和 train\_url 这两个配置，这两个变量对应的具体路径在创建训练任务时根据数据和模型在 obs 存储的目录指定。

创建方式

自定义脚本 自定义镜像

选择常用引擎创建训练作业。每个算法的代码目录最多支持1000个文件，文件深度不超过32，总大小不超过5GB。

\* AI引擎 TensorFlow tensorflow\_2.1.0-cuda\_10.1-py\_... 显示旧版引擎 引擎升级说明

\* 代码目录 /flower-img/self\_algorithm/ 选择

\* 启动文件 /flower-img/self\_algorithm/lab2\_flower\_classify\_train.py 选择

输入数据配置

为您的算法定义处理“输入数据”的参数。在您的算法代码中需要解析该参数获取到训练的数据集。

映射名称 数据来源1 代码路径参数 data\_url

添加约束 ☐ 是 ☒ 否

添加输入数据配置

输出数据配置

为您的算法定义处理“输出数据”的参数。在您的算法代码中需要解析该参数获取到训练的输出路径。

映射名称 输出数据1 代码路径参数 train\_url

训练完之后会在配置的输出路径（train\_url）上生成对应文件。生成的模型文件 saved\_model.pb 和数据文件 variables（可以看出数据集并没有跟训练代码存放在同一个目录）。

上传对象	新建文件夹	删除	更多
<input type="checkbox"/>	名称	存储类别	大小
<a href="#">返回上一级</a>			
<input type="checkbox"/>	variables	--	--
<input type="checkbox"/>	saved_model.pb	标准存储	241.36 KB
<input type="checkbox"/>	lab2_flower_classify_train.py	标准存储	7.13 KB

variables 对象下的具体文件

对象 / self\_algorithm / variables

对象 已删除对象 碎片

对象是数据存储的基本单位，在OBS中文件和文件夹都是对象。您可以上传任何类型（文本、图片、视频等）的文件，并在桶若需将对象移动到桶内其他位置，推荐下载使用OBS Browser+图形化管理工具。

上传对象 新建文件夹 删除 更多

名称	存储类别	大小
variables.index	标准存储	3.42 KB
variables.data-00001-of-00002	标准存储	129.78 MB
variables.data-00000-of-00002	标准存储	10.43 KB

## 训练过程的日志输出

< trainjob-8014

作业ID: 5219f2aa-e16e-4218-9e21-bd12de31dfd0

作业状态: 已完成

创建时间: 2022/01/08 14:58:12 GMT+08:00

运行时间: 00:00:39

算法名称: algorithm-af52

AI引擎: TensorFlow | tensorflow,2.10-cuda,10.1-py,3.7-ubuntu,18.04-x86\_64

代码目录: /flower-img/self\_algorithm/

启动文件: /flower-img/self\_algorithm/lab2\_flower\_classify\_train.py

计算节点个数: 1

规格: [限时免费]GPU: 1\*NVIDIA-V100(32GB) | CPU: 8核 64GB

```
397 Epoch 25/30
398 91/91 - 0s - loss: 0.3586 - accuracy: 0.8352 - val_loss: 0.5639 - val_accuracy: 0.7391
399 Epoch 26/30
400 91/91 - 0s - loss: 0.3402 - accuracy: 0.8571 - val_loss: 0.5499 - val_accuracy: 0.6522
401 Epoch 27/30
402 91/91 - 0s - loss: 0.3354 - accuracy: 0.8571 - val_loss: 0.5449 - val_accuracy: 0.6522
403 Epoch 28/30
404 91/91 - 0s - loss: 0.3322 - accuracy: 0.8791 - val_loss: 0.5365 - val_accuracy: 0.6957
405 Epoch 29/30
406 91/91 - 0s - loss: 0.3243 - accuracy: 0.8681 - val_loss: 0.5302 - val_accuracy: 0.6957
407 Epoch 30/30
408 91/91 - 0s - loss: 0.2933 - accuracy: 0.8791 - val_loss: 0.5205 - val_accuracy: 0.6957
409 Model: "sequential"
410
411 Layer (type) Output Shape Param #
412 -----
413 conv2d (Conv2D) multiple 2432
414 _
415 max_pooling2d (MaxPooling2D) multiple 0
416 _
417 dropout (Dropout) multiple 0
418 _
419 conv2d_1 (Conv2D) multiple 18496
420 _
421 conv2d_2 (Conv2D) multiple 36928
422 _
423 max_pooling2d_1 (MaxPooling2D) multiple 0
424 _
425 dropout_1 (Dropout) multiple 0
426 _
427 conv2d_3 (Conv2D) multiple 73856
```

应用部署较复杂，部署过程比较耗时，调试也非常不方便，如果部署不成功，只能修改代码添加 **print** 语句，打印关键信息，再重新上传修改后的代码到 **obs**，再重新建 AI 应用（虽然 **obs** 存储的文件已经更新了，原有的 AI 应用的镜像用的旧的数据，要重新构建镜像），部署在线服务，然后看日志判断可能出错的地方。考虑到以上困难，实验对在线部署不做要求，仍然鼓励大家尝试部署体验。如不做在线部署，需要将训练好的模型下载到本地进行评估测试。

更新推理代码之后，要重新建 AI 应用，从训练作业导入的推理代码部署需要遵从模型包规范，[https://support.huaweicloud.com/engineers-modelarts/modelarts\\_23\\_0091.html](https://support.huaweicloud.com/engineers-modelarts/modelarts_23_0091.html)。即推理代码必须跟训练代码在同一个目录，如下图，且推理代码文件必须命名为 **customize\_service.py**，再根据请求和返回参数修改配置文件 **config.json**。

<input type="checkbox"/>	variables	--	--	--
<input type="checkbox"/>	customize_service.py	标准存储	4.39 KB	未加密
<input type="checkbox"/>	config.json	标准存储	4.17 KB	未加密
<input type="checkbox"/>	saved_model.pb	标准存储	241.36 KB	未加密
<input type="checkbox"/>	lab2_flower_classify_train.py	标准存储	7.13 KB	未加密

下图推理代码这一项，是自动加载显示的，不能被配置。

\* 名称

flower-classify-v1

\* 版本

0.0.1

标签 ?

输入标签, 按Enter键创建

描述

使用minist的推理代码测试

15/100

\* 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

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。

\* 选择训练作业 ?

trainjob-8014\_copy-8829

☐ 动态加载 ?

AI引擎

TensorFlow

python3.6

推理代码 ?

https://flower-img.obs.myhwclouds.com/self\_algorithm/customize\_service.py

运行时依赖

安装方式

名称

版本

模型配置文件 config.json 编写说明

[https://support.huaweicloud.com/engineers-modelarts/modelarts\\_23\\_0092.html](https://support.huaweicloud.com/engineers-modelarts/modelarts_23_0092.html)

TensorFlow 2.1 自定义保存模型加载与推理

[https://support.huaweicloud.com/engineers-modelarts/modelarts\\_23\\_0301.html](https://support.huaweicloud.com/engineers-modelarts/modelarts_23_0301.html)

Pytorch 框架的自定义训练和在线部署参考  
[https://support.huaweicloud.com/engineers-modelarts/modelarts\\_23\\_0175.html](https://support.huaweicloud.com/engineers-modelarts/modelarts_23_0175.html)。

对于 pytorch 框架, 如要在线部署, 训练时必须采用 state\_dict 的保存方式, 如下, 只保存模型权重参数, 不保存模型结构, 格式为 pt,.pth 格式没试过, 可能也行。

`torch.save(net.state_dict(), model_path + "/flower_mlp.pt")`

训练的时候如果用的 pytorch1.8 的环境, 在线部署时 AI 应用的推荐配置如下, 使用 python3.7 的环境, 通过 pip 的方式安装 pytorch1.8, 不然可能会因为版本差距过大在加载模型的时候出错。

\* 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

此选项将导入您存储在对象存储服务 (OBS) 中的模型。目前仅支持从对象存储服务 (OBS) 导入 TensorFlow、MXNet、Caffe、PyTorch、Spark MLlib、Scikit Learn、XGBoost 模型。对于 Image 类型的模型建议您使用从容器镜像中选择的导入方式导入。您的模型文件需要存放于 model 目录下, 选择模型时请您选择 model 目录的上一级。如果您的模型推理代码, 请放置 model 目录下。文件名需为 customize\_service.py。模型包规范详情参见 模型包规范。

\* 选择元模型

/flower-img/pytorch/

\* AI引擎

PyTorch

python3.7

☐ 动态加载 ?

推理代码 ?

https://flower-img.obs.myhwclouds.com/pytorch/model/customize\_service.py

运行时依赖

安装方式	名称	版本	约束	操作
pip	numpy	1.15.0	当前版本以上	删除
pip	Pillow			删除
pip	torch	1.8.0	当前版本以上	删除
pip	torchvision	0.8.1	当前版本以上	删除

增加

Mindspore 在 modelarts 上训练可以使用 CPU、GPU 或昇腾 910，如果用 CPU 或 GPU 则选择 AI 引擎：MPI | mindspore\_1.3.0-cuda\_10.1-py\_3.7-ubuntu\_1804-x86\_64。

algorithm-1c9c TensorFlow | tensorflow\_2.1.0-cuda\_10.1... tf-mnist  
algorithm-af52 TensorFlow | tensorflow\_2.1.0-cuda\_10.1... tf-flower

data\_url /flower-img/train/ 数据集 数据存储位置

训练数据

增加训练输入

train\_url /flower-img/mindspore/model/ 数据存储位置

模型

增加训练输出

增加超参

增加环境变量

选择CPU或GPU都报错

公共资源池 专属资源池

CPU GPU Ascend

CPU: 2 核 8GB 50GB 获取输入数据大小

启动命令

创建训练作业提交如果出现以上错误，选择 GPU，3200GB 存储的这个节点，如下图。

公共资源池 专属资源池

CPU GPU Ascend

GPU: 1\*NVIDIA-V100(32GB) | CPU: 8 核 64GB 3200GB 获取输入数据大小

GPU: 8\*NVIDIA-V100(32GB) | CPU: 72 核 512GB 3200GB

GPU: 1\*NVIDIA-V100(32GB) | CPU: 8 核 64GB 3200GB

[限时免费] GPU: 1\*NVIDIA-V100(32GB) | CPU: 8 核 64GB 780GB

Mindspore 在 modelarts 部署只支持昇腾 310 芯片，而昇腾 310 支持的模型格式比较特殊，所以部署较为复杂。需要在训练的时候先生成 onnx 格式的模型，再通过模型转换功能转成 om 格式，再进行部署，官方文档也没做详细说明。可以参考以下

ARM-Ascend 模板：

<https://support.huaweicloud.com/inference-modelarts/inference-modelarts-0072.html#inference-modelarts-0072>

图像分类 mindspore 订阅算法：

<https://developer.huaweicloud.com/develop/aigallery/algorithm/detail?id=6b454013-dab9-4028-8c7a-47375067202c>

### 5.3.3 代金券的获取与使用

华为公司对本课程使用 modelArts 等云上资源提供了支持，将为每位同学发放华为云代金券。在实验开始前确保注册并实名认证华为云账号，助教将统一收集账号信息并申请代金券的发放，在实验开始前及时查看账户是否有代金券，如无代金券及时上报给助教。

使用过程可在费用中心查看哪些服务进行了收费以及收费明细，如有代金券费用会之恩

进行抵扣。如对扣费有疑问可联系华为云客服。

账期

2022/01

导出月账单

账单设置

汇总

账单详情

开发者联盟账单

应付金额

2022/01

¥0.53

=

现金支付  
¥0.00

+

代金券抵扣  
¥0.53

+

欠费金额  
¥0.00

汇总类别	总费用 (¥)	=	现金支付 (¥)	+	代金券抵扣 (¥)	+	欠费金额 (¥)
消费汇总	0.53		0.00		0.53		0.00
总计	0.53		0.00		0.53		0.00

展开趋势和汇总图表

更多维度按天按月分析,可以到成本中心查看。

按产品汇总

2022/01

默认按照产品类型搜索

🔍📄🔗

官网价 ¥0.56208212 = 优惠金额 ¥0.00000000 + 抹零金额 ¥0.03208212 + 应付金额 ¥0.53

展开更多

账期	账号	产品类型	产品	计费模式	账单类型	官网价(¥)	优惠金额(¥)	抹零金额(¥)	应付金额(¥)	现金支付(¥)	代金券抵扣(¥)	现金券抵扣(¥)	储值卡抵扣(¥)	欠费金额(¥)
2022/01	hackenzheng_hit...	对象存储服务...	云存储	按需	消费	0.01874881	0.00000000	0.00874881	0.01	0.00	0.01	0.00	0.00	0.00
2022/01	hackenzheng_hit...	ModelArts Mo...	modelarts虚拟...	按需	消费	0.54333331	0.00000000	0.02333331	0.52	0.00	0.52	0.00	0.00	0.00

注意：华为云代金券请用于实验用到的 ModelArts 和 obs 等产品，代金券金额足以保证完成实验，每人限发一张，个人使用其他产品产生的扣费自行承担！！使用过程中务必注意系统算出来的计费金额，尽可能选择低计费的模式，比如 obs 桶选择单 az，比如训练时能用单 GPU 卡完成的不要选择多 GPU 卡的节点。

计算资源一般会在任务结束后自动释放停止计费，但 obs 里面的数据如果一直在会一直计费，在实验结束后请检查下存储，将模型等大文件进行删除避免一直计费。

## 6、实验报告

每人需单独完成实验报告，只需写自己完成的部分，代码提交小组最终完整版，每种框架的实现分目录保存，不需要提交模型结构和参数文件，模型精度不是唯一评分标准，提交截止时间为答辩前一天，具体时间见作业提交系统。每个人都要提交一份报告和代码，最后一次课小组答辩所有成员都要上台。提交平台 <http://grader.tery.top:8000/#/courses>。

实验报告首页至少包括以下信息：

实验名称、姓名、学号、班级、上课学期、实验地点、授课教师、同组成员  
实验报告内容包含但不限于以下内容。

### 一、实验内容

概述你是用何种方式实现的

### 二、算法与模型介绍

使用的何种模型，需要详细说明模型结构及选型，包含但不限于卷积层、池化层、全连接层、损失函数等主要参数说明及选型理由，需要有完整的模型结构图。

### 三、算法实现

#### 3.1 实验环境

#### 3.2 数据集处理

如数据预处理、数据集划分、数据扩充、数据重采样、数据增强等方式及关键代码，说

明处理的理由与对实验结果的提升。如果没有用到可以不写。

### 3.2 模型构建与训练

模型定义代码；

模型训练：如超参数设置与调整、优化算法选择等详细说明，如训练集上的训练速度、收敛速度、收敛精度，需要有训练日志或 loss 曲线等截图。如有多种实现请分别说明。

### 3.3 实验结果：模型测试与评估

如本地预测结果截图与说明；如 ModelArts 在线部署的预测结果截图与说明；

是否评估过模型效果，用的什么指标，结果如何；

### 3.4 深度学习框架总结

总结对比三种深度学习框架在本地训练模型的优缺点，如相同模型结构、超参数等情况下，训练速度、精度、开发效率等方面的差别，推理速度、精度等区别。

总结对比同一框架在本地和 modelarts 使用和训练上的区别。

## 四. 总结

### 4.1 碰到的问题及解决方法

### 4.2 实验的启发、总结、建议

如对 mindspore、modelarts 的认识及建议。

## 参考文献