ModelArts

SDK 参考

文档版本 01

发布日期 2022-06-10





版权所有 © 华为技术有限公司 2022。 保留一切权利。

非经本公司书面许可,任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部,并不得以任何形式传播。

商标声明



HUAWE和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标,由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束,本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定,华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因,本文档内容会不定期进行更新。除非另有约定,本文档仅作为使用指导,本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址: 深圳市龙岗区坂田华为总部办公楼 邮编: 518129

网址: https://www.huawei.com

客户服务邮箱: support@huawei.com

客户服务电话: 4008302118

目录

1 文档导读	1
2 SDK 简介	2
3 准备工作	3
3.1 (可选)安装 ModelArts SDK	3
3.2 获取访问密钥	4
3.3 查看项目 ID	5
4 Session 鉴权	7
4.1 Session 鉴权概述	7
4.2 用户名密码认证模式	7
4.3 用户 AK-SK 认证模式	8
5 OBS 管理	9
5.1 OBS 管理概述	9
5.2 上传文件至 OBS	9
5.3 上传文件夹至 OBS	10
5.4 从 OBS 下载文件	11
5.5 从 OBS 下载文件夹	12
6 数据管理	14
6.1 数据集管理	14
6.1.1 查询数据集列表	14
6.1.2 创建数据集	15
6.1.3 查询数据集详情	23
6.1.4 更新数据集	24
6.1.5 删除数据集	24
6.2 数据集版本管理	25
6.2.1 查询数据集版本列表	25
6.2.2 创建数据集版本	25
6.2.3 查询数据集版本详情	26
6.2.4 删除数据集版本	27
6.3 样本管理	27
6.3.1 查询样本列表	27
6.3.2 查询单个样本详情	28

6.3.3 批量删除样本	29
6.4 导入任务管理	29
6.4.1 查询导入任务列表	29
6.4.2 创建导入任务	29
6.4.3 查询导入任务状态	31
6.5 导出任务管理	32
6.5.1 查询导出任务列表	32
6.5.2 创建导出任务	32
6.5.3 查询导出任务状态	33
6.6 Manifest 管理	33
6.6.1 Manifest 管理概述	33
6.6.2 解析 Manifest 文件	34
6.6.3 创建和保存 Manifest 文件	36
6.6.4 解析 Pascal VOC 文件	37
6.6.5 创建和保存 Pascal VOC 文件	40
6.7 标注任务管理	40
6.7.1 创建标注任务	40
6.7.2 查询数据集的标注任务列表	41
6.7.3 查询标注任务详情	42
7 训练管理(推荐)	44
7.1 训练作业	44
7.1.1 创建训练作业	44
7.1.2 查询训练作业列表	49
7.1.3 查询训练作业详情	67
7.1.4 更新训练作业描述	83
7.1.5 删除训练作业	84
7.1.6 终止训练作业	85
7.1.7 查询训练日志	101
7.1.8 查询训练作业的运行指标	102
7.2 资源和引擎规格接口	104
7.2.1 查询资源规格列表	104
7.2.2 查询引擎规格列表	105
8 训练管理(旧版)	107
8.1 训练作业	
8.1.1 创建训练作业	
8.1.2 训练作业调试	
8.1.3 查询训练作业列表	
8.1.4 查询训练作业详情	
8.1.5 更新训练作业描述	
8.1.6 获取训练作业日志的文件名	
8.1.7 查询训练作业日志	
8.1.8 删除训练作业	

8.2 训练作业版本	125
8.2.1 创建训练作业版本	125
8.2.2 查询训练作业版本列表	129
8.2.3 查询训练作业版本详情	130
8.2.4 停止训练作业版本	134
8.2.5 删除训练作业版本	135
8.3 训练作业参数配置	136
8.3.1 创建训练作业参数	136
8.3.2 查询训练作业参数对象列表	138
8.3.3 查询训练作业参数列表	140
8.3.4 查询训练作业参数详情	141
8.3.5 更新训练作业参数	143
8.3.6 删除训练作业参数	146
8.4 可视化作业	147
8.4.1 创建可视化作业	147
8.4.2 查询可视化作业对象列表	148
8.4.3 查询可视化作业列表	150
8.4.4 查询可视化作业详情	152
8.4.5 更新可视化作业描述	153
8.4.6 停止可视化作业	154
8.4.7 重启可视化作业	
8.4.8 删除可视化作业	
8.5 资源和引擎规格接口	
8.5.1 查询预置算法	157
8.5.2 查询资源规格列表	
8.5.3 查询引擎规格列表	
8.6 作业状态参考	160
9 模型管理	162
9.1 模型调试	162
9.2 导入模型	166
9.3 查询模型列表	171
9.4 查询模型对象列表	173
9.5 查询模型详情	175
9.6 删除模型	178
10 服务管理	179
10.1 服务管理概述	179
10.2 服务调试	
10.3 部署在线服务	181
10.4 查询服务详情	189
10.5 推理服务测试	
10.6 查询服务列表	193
10.7 查询服务对象列表	195

ModelArts

SDK 参考	目录
10.8 更新服务配置	197
10.9 查询服务监控信息	200
10.10 查询服务日志	202
10.11 删除服务	204
11 修订记录	205

1 文档导读

本文档指导您如何安装和配置开发环境、如何通过调用ModelArts SDK提供的接口函数进行二次开发。

章节	内容
SDK简介	简要介绍ModelArts SDK的概念。
(可选)安装 ModelArts SDK 查看项目ID	介绍使用ModelArts SDK进行二次开发过程前的准备工作。
Session鉴权	Session模块的主要作用是实现与公有云资源的鉴权,并初始 化ModelArts SDK Client、OBS Client。
OBS管理概述	ModelArts SDK支持对OBS的SDK接口进行调用,包括创建 OBS桶,上传/下载文件和文件夹,删除OBS对象和桶。
ModelArts SDK具体 操作管理请参见如下 章节:	介绍使用ModelArts SDK进行的常用操作。
数据管理	
训练管理(旧版)	
模型管理	
服务管理	

2 SDK 简介

ModelArts服务软件开发工具包(ModelArts SDK)是对ModelArts服务提供的REST API进行的Python封装,以简化用户的开发工作。用户直接调用ModelArts SDK即可轻松管理数据集、启动Al训练以及生成模型并将其部署上线。

ModelArts SDK目前只提供Python语言的SDK,同时支持Python 2.7、Python 3.6和 Python 3.7。

SDK 版本说明

表 2-1 ModelArts SDK 版本说明

发布时间	版本号	说明
2019-08-08	1.1.3(推荐使用)	1.1.3版本在SDK旧版本基础上优化集成,主要包括 session、OBS管理、作业管理、模型管理、服务管理。本版本SDK的历史修改记录如下所示。 • 2020.04:新增OBS相关操作接口。新增支持 TensorFlow本地推理。

支持的区域

当前支持的"region_name"包括华北-北京一(cn-north-1)、华北-北京四(cn-north-4)、华东-上海一(cn-east-3)、华南-广州(cn-south-1)、乌兰一(cn-north-9)。

3 准备工作

3.1 (可选)安装 ModelArts SDK

ModelArts SDK支持在ModelArts开发环境Notebook和本地PC环境中使用。

ModelArts SDK已经集成在ModelArts开发环境Notebook中,无需安装,可以直接使用。

进入ModelArts控制台的开发环境模块,创建Notebook实例,在Terminal或Ipynb文件中直接调用ModelArts SDK的接口。

在Notebook中调用SDK,可直接参考接口说明执行OBS管理、作业管理、模型管理和服务管理等操作。

- ModelArts SDK支持在本地Windows/Linux环境安装配置使用。配置步骤如下:
 - a. 配置运行环境
 - 从Python官网下载并安装Python(建议使用2.7及以上版本的 Python)。若Python安装过程中没有安装通用包管理工具pip,则参见 pip官网完成pip安装。
 - 配置pip源。
 - o. 下载ModelArts SDK
 - c. 安装ModelArts SDK

□ 说明

Windows环境中,如果提示"不是内部或外部命令",请您在"环境变量"中设置"Path",增加Python和pip的安装路径。pip的安装路径一般为Python所在目录的Scripts文件夹。

下载 ModelArts SDK

下载ModelArts SDK软件包,获取最新版本的ModelArts SDK软件包。

安装 ModelArts SDK

 在本地环境执行命令python --version,确认环境已经安装完成Python。 C:\Users\xxx>python --version Python *.** 2. 执行命令**pip --version**,确认Python通用包管理工具pip已经存在。 C:\Users\xxx>pip --version

pip **.** from c:\users\xxx\appdata\local\programs\python\python**\lib\site-packages\pip (python *.*)

3. 执行如下命令,安装SDK。

pip install {SDK软件包路径}\modelarts-latest-py2.py3-none-any.whl

 $\hbox{C:} \verb|Users| xxxx> pip install C: \verb|Users| xxxx| Downloads| model arts-latest-py2.py3-none-any.whl \\$

Successfully installed Pillow-*.*.0 ... modelarts-*.*.* ...

在安装SDK时会默认同时安装所需的依赖包。当显示"Successfully installed"时,表示ModelArts SDK安装完成。

□ 说明

如果在安装过程中报错提示缺少相应的依赖包,请根据报错提示执行如下命令进行依赖包安装。

pip install xxxx

其中,xxxx为依赖包的名称。

3.2 获取访问密钥

用户进行Session鉴权时,若使用AK-SK认证模式,需要获取访问密钥。

1. 登录华为云,在页面右上方单击"控制台",进入华为云管理控制台。

图 3-1 控制台入口

2. 在控制台右上角的帐户名下方,单击"我的凭证",进入"我的凭证"页面。

图 3-2 我的凭证



3. 在"我的凭证"页面,选择"访问密钥>新增访问密钥",如<mark>图3-3</mark>所示。

图 3-3 单击新增访问密钥



4. 填写该密钥的描述说明,单击"确定"。根据提示单击"立即下载",下载密钥。

图 3-4 新增访问密钥



立即下载
取消

5. 密钥文件会直接保存到浏览器默认的下载文件夹中。打开名称为 "credentials.csv"的文件,即可查看访问密钥(Access Key Id和Secret Access Key)。

3.3 查看项目 ID

在调用接口的时候,部分URL中需要填入项目编号,所以需要获取到项目编号。项目编号获取步骤如下:

- 注册并登录管理ModelArts控制台。
- 2. 在页面右上角单击用户名,然后在下拉列表中单击"我的凭证",进入"我的凭证"页面。

□ 说明

如果您登录的是华为云官网,而非管理控制台。在单击用户名后,选择下拉列表中的"帐号中心",然后单击"管理我的凭证"进入"我的凭证"页面。

3. 在"API凭证"页面的项目列表中查看项目ID和名称(即"项目")。

图 3-5 查看项目 ID



多项目时,展开"所属区域",从"项目ID"列获取子项目ID。

4 Session 鉴权

4.1 Session 鉴权概述

Session模块的主要作用是实现与公有云资源的鉴权,并初始化ModelArts SDK Client、OBS Client。当成功建立Session后,您可以直接调用ModelArts的SDK接口。

当前ModelArts Notebook开发环境不需要session鉴权,可以直接使用。示例代码如下:

from modelarts.session import Session session = Session()

对于其它开发环境下使用ModelArts SDK时,session鉴权方式请参考以下三种认证方式,选择其中一种方式进行认证即可。

- 用户名密码认证模式: 支持OBS管理、训练管理、模型管理、服务管理的鉴权。
- 用户AK-SK认证模式: 支持OBS管理、训练管理、模型管理、服务管理的鉴权。

4.2 用户名密码认证模式

本模式支持OBS管理、训练管理、模型管理、服务管理的鉴权。

示例代码

帐号与用户的概念介绍,请参见IAM基本概念。获取您的帐号、用户名等信息,请参见获取用户名、用户ID、项目名称、项目ID。

● 使用帐号认证

"username"填写您的帐号名。 from modelarts.session import Session session = Session(username='***', password='***', region_name='***', project_id='***')

□ 说明

如果您的华为云帐号已经升级为华为帐号,则帐号认证方式将不可用,请创建一个IAM用户,使用IAM用户认证。

● 使用IAM用户认证

"account"填写您的帐号名,"username"填写您的IAM用户名。

from modelarts.session import Session session = Session(account="***", username="***", password="***", region_name="***", project_id="***")

4.3 用户 AK-SK 认证模式

本模式支持OBS管理、训练管理、模型管理、服务管理模块的鉴权。

示例代码

from modelarts.session import Session session = Session(access_key='***',secret_key='***', project_id='***', region_name='***')

其中,各参数说明如下:

- "access_key"和"secret_key"获取方式请参见获取访问密钥。
- "project_id"即项目ID。获取方式请参见查看项目ID。
- "region_name"即区域ID。获取方式请参见获取区域ID。

5 obs 管理

5.1 OBS 管理概述

ModelArts SDK 1.1.3支持对OBS进行管理,主要涵盖上传下载文件和文件夹。具体包括如下操作:

- 上传文件至OBS
- 上传文件夹至OBS
- 从OBS下载文件
- 从OBS下载文件夹

5.2 上传文件至 OBS

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

from modelarts.session import Session session = Session() session.obs.upload_file(src_local_file='/home/ma-user/file1.txt', dst_obs_dir='obs://bucket-name/dir1/')

示例代码执行后,本地源文件"file1.txt"被上传至"bucket-name"桶的"dir1"文件夹下,路径为"obs://bucket-name/dir1/file1.txt"。其中,桶名称和文件夹的名称均可以按照业务需求自定义。

参数说明

表 5-1 请求参数说明

参数	是否必 选	参数类型	描述
session	是	Object	会话对象。

参数	是否必 选	参数类型	描述
src_local_file	是	String	本地需要上传的文件路径。
dst_obs_dir	是	String	上传的目标OBS桶地址,必须以"obs://"作为前缀,上传的目标文件夹后缀必须以"/" 结尾。

表 5-2 失败响应参数说明

参数	参数类型	描述
error_code	String	调用失败时的错误码。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

5.3 上传文件夹至 OBS

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参考Session鉴权。

from modelarts.session import Session session = Session() session.obs.upload_dir(src_local_dir='/home/ma-user/', dst_obs_dir='obs://bucket-name/dir1/')

示例代码执行后,本地源文件夹"/ma-user/"被上传至"bucket-name"桶的 "dir1"文件夹下,路径为"obs://bucket-name/dir1/ma-user/"。其中,桶名称和 文件夹的名称均可以按照业务需求自定义。

参数说明

表 5-3 请求参数说明

参数	是否必 选	参数类型	描述
session	是	Object	会话对象。
src_local_dir	是	String	本地需要上传的文件夹路径。 当上传的文件夹下内容为空或者该文件夹下 包含多个文件夹且有文件夹下内容有空时, OBS对应路径下不产生该空文件夹。

参数	是否必 选	参数类型	描述
dst_obs_dir	是	String	上传的目标OBS桶地址,必须以"obs://"作为前缀,上传的目标文件夹后缀必须以"/"结尾。

表 5-4 失败响应参数说明

参数	参数类型	描述
error_code	String	调用失败时的错误码。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

5.4 从 OBS 下载文件

□ 说明

当文件夹下某个文件大小超过5GB时,无法使用此方式下载此文件。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参考Session鉴权。

from modelarts.session import Session session = Session() session.obs.download_file(src_obs_file="obs://bucket-name/dir1/file1.txt", dst_local_dir="/home/ma-user/")

示例代码执行后,OBS源文件"file1.txt"被下载至"/home/ma-user/file1.txt"。

参数说明

表 5-5 请求参数说明

参数	是否必 选	参数类型	描述
session	是	Object	会话对象。
src_obs_file	是	String	下载的源OBS文件,必须以"obs://"作为 前缀。
dst_local_dir	是	String	下载目标的本地文件夹,下载的本地目标文件夹后缀必须以"/"结尾。

表 5-6 失败响应参数说明

参数	参数类型	描述
error_code	String	调用失败时的错误码。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

5.5 从 OBS 下载文件夹

□ 说明

当单个文件大小超过5GB时,无法使用此方式下载此文件。但是此文件夹下其他小于5GB的文件可下载成功。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参考Session鉴权。

from modelarts.session import Session session = Session() session.obs.download_dir(src_obs_dir="obs://bucket-name/dir1/", dst_local_dir="/home/ma-user/")

示例代码执行后,OBS源文件夹"dir1"被下载至本地"/home/ma-user/dir1/"。

参数说明

表 5-7 请求参数说明

参数	是否必 选	参数类型	描述
session	是	Object	会话对象。
src_obs_dir	是	String	下载的源OBS文件夹,必须以"obs://"作为前缀,文件夹后缀必须以"/"结尾。当下载的文件夹下有文件夹且内容为空时,对应路径下不产生对应空文件夹。
dst_local_dir	是	String	下载的目标本地文件夹,下载的目标本地文件夹后缀必须以"/"结尾。

表 5-8 失败响应参数说明

参数	参数类型	描述
error_code	String	调用失败时的错误码。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

6数据管理

6.1 数据集管理

6.1.1 查询数据集列表

分页查询用户的数据集列表。

list_datasets(session, dataset_type=None, dataset_name=None, offset=None, limit=None)

示例代码

示例一:查询数据集列表

from modelarts.session import Session from modelarts.dataset import Dataset session = Session() # 查询数据集列表 dataset_list = Dataset.list_datasets(session) print(dataset_list) # 打印出查询结果

• 示例二:根据数据集类型查询数据集列表

查询图像分类数据集 dataset_list = Dataset.list_datasets(session, dataset_type=0) print(dataset_list)

● 示例三:根据数据集名称查询数据集列表

查询名称中包含dataset的数据集列表 dataset_list = Dataset.list_datasets(session, dataset_name="dataset") print(dataset_list)

• 示例四:分页查询数据集列表

默认一次返回10条数据集记录,可通过设置limit和offset进行分页查询 dataset_list = Dataset.list_datasets(session, offset=0, limit=50) # 查询第1-50条记录 print(dataset_list) dataset_list = Dataset.list_datasets(session, offset=1, limit=50) # 查询第51-100条记录 print(dataset_list)

参数说明

表 6-1 请求参数

参数	是否必 选	参数类型	描述
session	是	Object	会话对象,初始化方法请参见 <mark>Session鉴权</mark> 。
dataset_typ e	否	Integer	根据数据集类型查询数据集列表,默认为空。可选值如下:
			600: 视频标注 900: 自由格式
dataset_na me	否	String	模糊匹配数据集名称,默认为空。
offset	否	Integer	分页列表的起始页,默认为0。
limit	否	Integer	指定每一页返回的最大条目数,取值范围 [1,100],默认为10。

6.1.2 创建数据集

创建数据集,支持从OBS中导入数据。

 $create_dataset(session, \ dataset_name=None, \ data_type=None, \ data_sources=None, \ work_path=None, \ dataset_type=None, \ **kwargs)$

创建数据集支持两种用法:

- 根据**标注类型**创建数据集,一个数据集只能支持一种标注任务类型。create_dataset(session,dataset_name=None, dataset_type=None, data_sources=None, work_path=None, **kwargs)
- 根据**数据类型**创建数据集,用户可以在相同的数据集上创建不同类型的标注任务,如在图像数据集上创建图像分类、物体检测等标注任务。
 create_dataset(session,dataset_name=None, data_type=None, data_sources=None, work_path=None,
 **kwargs)

□ 说明

推荐使用根据数据类型创建数据集,根据标注类型创建数据集的功能将会下线。

示例代码

● 示例一:根据数据类型创建图像数据集

```
from modelarts.session import Session
from modelarts.dataset import Dataset
session = Session()
dataset_name = "dataset-image" # 数据集名称
                            # 数据集类型,图像类型数据集
data_type = "IMAGE"
data_sources = dict()
                            # 数据集数据来源
data_sources = dict()    # 数据集数据来源
data_sources["type"] = 0   # 数据来源类型,0表示OBS
data_sources["path"] = "/obs-gaia-test/data/image/image-classification/" # 数据在OBS中的路径
work_path = dict()
                         # 数据集的工作目录
work_path['type'] = 0
                             #数据集工作目录的类型,0表示OBS
work_path['path'] = "/obs-gaia-test/data/output/work_path/" # 数据集工作目录在OBS中的路径 create_dataset_resp = Dataset.create_dataset(session, dataset_name=dataset_name,
data_type=data_type,
                               data_sources=data_sources, work_path=work_path)
```

示例二:根据数据类型创建图像数据集(导入标注信息)

```
from modelarts.session import Session
from modelarts.dataset import Dataset
session = Session()
dataset_name = "dataset-image-with-annotations"
data_type = "IMAGE"
data_sources = dict()
data_sources["type"] = 0
data_sources["path"] = "/obs-gaia-test/data/image/image-classification/"
annotation_config = dict()
                            # 源数据的标注格式
annotation_config['scene'] = "image_classification" # 数据标注场景为图像分类标注
annotation_config['format_name'] = "ModelArts image classification 1.0" # 标注格式为ModelArts
image classification 1.0
data_sources['annotation_config'] = annotation_config
work_path = dict()
work_path['type'] = 0
work_path['path'] = "/obs-gaia-test/data/output/work_path/"
create_dataset_resp = Dataset.create_dataset(session, dataset_name=dataset_name,
data_type=data_type,
                            data_sources=data_sources, work_path=work_path)
```

• 示例三:根据数据类型创建表格数据集

```
from modelarts.session import Session
from modelarts.dataset import Dataset
session = Session()
dataset_name = "dataset-table"
data_type = "TABLE"
data_sources = dict()
data_sources["type"] = 0
data_sources["path"] = "/obs-gaia-test/data/table/table0/"
data_sources['with_column_header'] = True
work path = dict()
work_path['type'] = 0
work_path['path'] = "/obs-gaia-test/data/output/work_path/"
#表格类型的数据集需要指定表格数据的schema信息
schema0 = dict()
schema0['schema_id'] = 0
schema0['name'] = "name"
schema0['type'] = "STRING"
schema1 = dict()
schema1['schema_id'] = 1
```

```
schema1['name'] = "age"
schema1['type'] = "STRING"
schema2 = dict()
schema2['schema_id'] = 2
schema2['name'] = "label"
schema2['type'] = "STRING"
schemas = []
schemas.append(schema0)
schemas.append(schema1)
schemas.append(schema2)
create_dataset_resp = Dataset.create_dataset(session, dataset_name=dataset_name, data_type=data_type,

data_sources=data_sources, work_path=work_path, schema=schemas)
```

• 示例四:根据标注类型创建图像分类数据集

```
from modelarts.session import Session from modelarts.dataset import Dataset

session = Session()

dataset_name = "dataset-image-classification"
dataset_type = 0 # 数据集的标注类型,0表示图像分类标注类型
data_sources = dict()
data_sources["path"] = "/obs-gaia-test/data/image/image-classification/"
data_sources["type"] = "0"
work_path = dict()
work_path['type'] = 0
work_path['path'] = "/obs-gaia-test/data/output/work_path/"
create_dataset_resp = Dataset.create_dataset(session, dataset_name=dataset_name, dataset_type=dataset_type, data_sources=data_sources, work_path)
```

● 示例五:根据标注类型创建文本三元组数据集

```
dataset_name = "dataset-text-triplet"
dataset_type = 102 # 数据集标注类型,102表示文本三元组标注类型
data_sources = dict()
data sources['type'] = 0
data_sources['path'] = "/obs-gaia-test/data/text/text-classification/"
work_path = dict()
work_path['type'] = 0
work_path['path'] = "/obs-gaia-test/data/output/work_path/"
# 创建文本三元组标注类型的数据集,需要传入标签参数
label_entity1 = dict() # 标签对象
label_entity1['name'] = "疾病" # 标签名称
label_entity1['type'] = 101 # 标签类型, 101表示实体类型标签
label_entity2 = dict()
label_entity2['name'] = "疾病别称"
label_entity2['type'] = 101
label_relation1 = dict()
label_relation1['name'] = "又称为"
label_relation1['type'] = 102 # 标签类型,102表示关系类型标签
property = dict() # 关系类型标签需要在标签属性中指定起始实体标签和终止实体标签
property['@modelarts:from_type'] = "疾病" # 起始实体标签 property['@modelarts:to_type'] = "疾病别称" # 终止实体标签
label_relation1['property'] = property
labels = []
labels.append(label_entity1)
labels.append(label_entity2)
labels.append(label_relation1)
create dataset resp = Dataset.create dataset(session, dataset name=dataset name,
dataset_type=dataset_type, data_sources=data_sources, work_path=work_path, labels=labels)
```

● 示例六:根据标注类型创建表格数据集

```
dataset_name = "dataset-table"
dataset_type = 400 # 数据集标注类型,400表示表格数据集
data_sources = dict()
data_sources['type'] = 0
data_sources['path'] = "/obs-gaia-test/data/table/table0/"
data_sources['with_column_header'] = True # 用来指明表格数据中是否包含表头
work_path = dict()
```

```
work_path['type'] = 0
work_path['path'] = "/obs-gaia-test/data/output/work_path/"
# 表格数据集需要传入表格数据的表头参数
schema0 = dict() # 表格的表头
schema0['schema_id'] = 0 # 第一列表头
schema0['name'] = "name" # 表头名称,该列表头为name
schema0['type'] = "STRING" # 表头数据类型,表示字符串
schema1 = dict()
schema1['schema_id'] = 1
schema1['name'] = "age"
schema1['type'] = "STRING"
schema2 = dict()
schema2['schema_id'] = 2
schema2['name'] = "label"
schema2['type'] = "STRING"
schemas = []
schemas.append(schema0)
schemas.append(schema1)
schemas.append(schema2)
create_dataset_resp = Dataset.create_dataset(session, dataset_name=dataset_name,
dataset_type=dataset_type, data_sources=data_sources, work_path=work_path, schema=schemas)
```

参数说明

表 6-2 请求参数

参数	是否 必选	参数类型	描述
session	是	Object	会话对象,初始化方法请参见 <mark>Session鉴权</mark> 。
dataset_name	是	String	数据集名称。
data_type	否	String	数据集的数据类型,与dataset_type参数二选一,推荐使用data_type。可选值如下: IMAGE: 图像 TEXT: 文本 AUDIO: 音频 TABLE: 表格 VIDEO: 视频 PLAIN: 自由格式

参数	是否 必选	参数类型	描述
dataset_type	否	Integer	根据数据集类型查询数据集列表,与data_type参数二选一。可选值如下:
data_sources	是	表6-3	数据集输入位置,用于将此目录及子目录下的源数据(如图片/文件/音频等)同步到数据集。对于表格数据集,该参数为导入目录。表格数据集的工作目录不支持为KMS加密桶下的OBS路径。
work_path	是	表6-7	数据集输出位置,用于存放输出的标注信息 等文件。
labels	否	List of 表 6-8	数据集标签列表,创建文本三元组标注类型 的数据集时需要传递该参数。
schema	否	List of 表 6-10	schema列表,用于表格数据集,指定表头的 名称和类型。
description	否	String	数据集描述,默认为空,描述不能包含^! <>=&"'等特殊字符,长度为0-256。

表 6-3 DataSource 参数

参数	是否 必选	参数类型	描述
type	是	Integer	数据类型。可选值如下:
			● 5: Al Gallery下载数据集

参数	是否 必选	参数类型	描述
path	是	String	数据源所在路径。 ● 字符限制: 不允许出现的特殊字符有换行符(\n)、回车符(\r)、制表符(\t)。
content_info	否	表6-4	从AI Gallery下载数据集时数据集资产的信息。
annotation_co nfig	否	表6-5	数据标注格式的说明。目前支持的标注格式 类型如下: 图像分类 物体检测 文本分类 声音分类
with_column_ header	否	Boolean	表格数据集必选参数,表格的第一行是否为表头。 • True: 第一行数据作为表头 • False: 第一行数据不作为表头,仅为样本数据

表 6-4 ContentInfo 参数

参数	是否 必选	参数类型	描述
content_id	是	String	AI Gallery中数据集资产的ID。
version_id	是	String	AI Gallery中数据集资产的版本ID。

表 6-5 AnnotationConfig 参数

参数	是否 必选	参数类型	描述
scene	是	String	支持的标注格式场景,可选值如下: image_classification: 图像分类 object_detection: 物体检测 text_classification: 文本分类 audio_classification: 声音分类

参数	是否 必选	参数类型	描述
fromat_name	是	String	不同标注场景下的标注格式。可选值如下: image_classification ModelArts imageNet 1.0 ModelArts image classification 1.0 object_detection ModelArts PASCAL VOC 1.0 YOLO text_classification ModelArts text classfication 1.0 ModelArts text classfication combine 1.0 audio_classification ModelArts audio classfication dir 1.0
parameters	否	表6-6	标注格式的高级参数,如样本分割符等。

表 6-6 AnnotationConfigParam 参数

参数	是否 必选	参数类型	描述
included_label s	否	List of 表 6-8	只导入包含指定标签的样本。
sample_label_s eparator	否	String	文本和标签之间的分割符。分隔符仅支持一个字符,必须为大小写字母,数字或@#Y%^&*_= ?/'::,,中的某一个字符,分割符需要转义。
label_separato r	否	String	标签和标签之间的分割符。分隔符仅支持一个字符,必须为大小写字母,数字或@#¥%^&*_= ?/'::,,中的某一个字符,分割符需要转义。
difficult_only	否	Boolean	是否只导入难例。

表 6-7 WorkPath 参数

参数	是否 必选	参数类型	描述
type	是	Integer	数据类型。可选值如下: ● 0: OBS桶(默认值)

参数	是否必选	参数类型	描述
path	是	String	数据集输出位置,用于存放输出的标注信息 等文件。
			● 格式为 "/桶名称/文件路径",例如"/ obs-bucket/flower/rose/"(使用目录作 为路径)。
			• 不能直接使用桶作为路径。
			输出位置不能与输入位置相同或者是输入 位置的子目录。
			● 长度限制:不少于3字符,不能超过700个字符。
			● 字符限制:不允许出现的特殊字符有换行符(\n)、回车符(\r)、制表符(\t)。

表 6-8 Label 参数

参数	是否 必选	参数类型	描述
name	是	String	标签名称。
type	是	Integer	标签类型,可选值如下:
property	否	表6-9	标签基本属性键值对,如颜色。

表 6-9 LabelProperty 参数

参数	是否 必选	参数类型	描述
@modelarts:co lor	否	String	内置属性:标签展示的颜色,为色彩的16进制代码,默认为空。例如:"#FFFFF0"。

参数	是否 必选	参数类型	描述
@modelarts:fr om_type	否	String	内置属性: 三元组关系标签的起始实体类型,创建关系标签时必须指定,该参数仅文本三元组数据集使用。
@modelarts:to _type	否	String	内置属性: 三元组关系标签的指向实体类型,创建关系标签时必须指定,该参数仅文本三元组数据集使用。

表 6-10 Schema 参数

参数	是否 必选	参数类型	描述
schema_id	否	Integer	Schema ID。
name	否	String	Schema名称。
type	否	String	Schema值类型,可选值如下: STRING SHORT INT LONG DOUBLE FLOAT BYTE DATE TIMESTAMP BOOLEAN
description	否	String	Schema描述。

6.1.3 查询数据集详情

查询数据集的详细信息,包括数据集的样本信息、版本信息等。

dataset.get_dataset_info()

示例代码

查询数据集详情

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

dataset = Dataset(session, dataset_id)

dataset_info = dataset.get_dataset_info()
print(dataset_info) # 输出数据集的详细信息

参数说明

无。

6.1.4 更新数据集

更新数据集的名称和描述信息。

dataset.update_dataset(dataset_name=None, description=None)

示例代码

更新数据集名称

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

dataset = Dataset(session, dataset_id)
dataset.update_dataset(dataset_name = "new-dataset-name")

参数说明

表 6-11 请求参数

参数	是否必选	参数类型	描述
dataset_name	否	String	数据集名称。
description	否	String	数据集描述信息。

6.1.5 删除数据集

根据数据集ID删除指定的数据集

delete_dataset(session, dataset_id)

示例代码

删除数据集

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

 $Dataset.delete_dataset(session, \ dataset_id="68ZXdK6CZwgvUICOOdC")$

参数说明

表 6-12 请求参数

参数	是否 必选	参数类型	描述
session	是	Object	会话对象,初始化方法请参见 <mark>Session鉴权</mark> 。
dataset_id	是	String	数据集的ID。

6.2 数据集版本管理

6.2.1 查询数据集版本列表

查询数据集的版本列表。

dataset.list_versions()

示例代码

查询数据集版本列表

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

dataset = Dataset(session, dataset_id)
version_list = dataset.list_versions()
print(version_list) # 打印数据集的版本列表

参数说明

无。

6.2.2 创建数据集版本

为数据集创建新的版本。

dataset.create_version(name=None, version_format=None, label_task_type=None, label_task_id=None, **kwargs)

示例代码

示例一: 为数据集创建新的版本

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

dataset = Dataset(session, dataset_id)
create_version_resp = dataset.create_version(name="V001", version_format="Default", label_task_type=0,
description="version 001")

示例二:基于标注任务创建数据集

from modelarts.session import Session from modelarts.dataset import Dataset

session = Session()

dataset = Dataset(session, dataset_id) create_version_resp = dataset.create_version(label_task_id="IbAhFai5KXWC3gthUfz", description="dataset version from label task")

参数说明

表 6-13 请求参数

参数	是否必选	参数类型	描述
name	否	String	版本名称,必须是中文、字母、数字、下划 线或中划线组成的合法字符串,长度为1-32 位。
version_forma t	否	String	数据集版本格式。可选值如下: ● Default: 默认格式
label_task_typ e	否	Integer	版本数据对应的标注类型。可选值如下:
label_task_id	否	String	基于标注任务创建数据集版本,标注任务 ID。
description	否	String	版本描述信息,默认为空,长度为0-256位, 不能包含!<>=&"'特殊字符。

6.2.3 查询数据集版本详情

根据版本ID查询数据集指定版本的详细信息。

dataset.get_version_info(version_id)

示例代码

查询数据集指定版本的详细信息

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

dataset = Dataset(session, dataset_id)
version_info = dataset.get_version_info(version_id="cSPuXPgnYp7ObRs6LaR")
print(version_info) # 打印数据集版本的详情

参数说明

表 6-14 请求参数

参数	是否 必选	参数类型	描述
version_id	是	String	数据集版本ID。

6.2.4 删除数据集版本

删除数据集的指定版本。

dataset.delete version(version id)

示例代码

删除数据集指定版本

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

dataset = Dataset(session, dataset_id)
dataset.delete_version(version_id="cSPuXPqnYp7ObRs6LaR")

参数说明

表 6-15 请求参数

参数	是否 必选	参数类型	描述
version_id	是	String	数据集版本ID。

6.3 样本管理

6.3.1 查询样本列表

查询数据集的样本列表,不支持表格类型数据集。

dataset.list_samples(version_id=None, offset=None, limit=None)

示例代码

示例一:查询数据集样本列表

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

dataset = Dataset(session, dataset_id) list_samples_resp = dataset.list_samples() print(list_samples_resp) # 打印样本列表的信息

● 示例二: 查询数据集指定版本的样本列表

list_samples_resp = dataset.list_samples(version_id = "cSPuXPgnYp7ObRs6LaR")
print(list_samples_resp)

参数说明

表 6-16 请求参数

参数	是否 必选	参数类型	描述
version_id	否	String	数据集版本ID。传入版本ID查询数据集相应版本的样本列表。
offset	否	Integer	分页列表的起始页,默认为0。
limit	否	Integer	指定每一页返回的最大条目数,取值范围 [1,100],默认为10。

6.3.2 查询单个样本详情

根据样本ID查询数据集中指定样本的详细信息。

dataset.get_sample_info(sample_id)

示例代码

根据ID查询数据集中样本的详细信息

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

dataset = Dataset(session, dataset_id)
sample_info = dataset.get_sample_info(sample_id="2551e78974aed9b60156d8376232f6bd")
print(sample_info) # 打印样本的详细信息

参数说明

表 6-17 请求参数

参数	是否 必选	参数类型	描述
sample_id	是	String	样本ID。

6.3.3 批量删除样本

根据样本的ID列表批量删除数据集中的样本。

dataset.delete_samples(samples)

示例代码

批量删除数据集中的样本

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

dataset = Dataset(session, dataset_id)
samples = []
samples.append("2551e78974aed9b60156d8376232f6bd")
samples.append("0d315fec1efc7568de5cccf522c10a1b")
dataset.delete_samples(samples)

参数说明

表 6-18 请求参数

参数	是否 必选	参数类型	描述
samples	是	List of String	待删除的样本ID列表。

6.4 导入任务管理

6.4.1 查询导入任务列表

查询数据集导入任务列表。

dataset.list_import_tasks()

示例代码

查询数据集导入任务列表

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

dataset = Dataset(session, dataset_id) list_tasks_resp = dataset.list_import_tasks() print(list_tasks_resp) # 打印导入任务列表

参数说明

无。

6.4.2 创建导入任务

支持从OBS中导入新的数据,导入方式包括目录导入和Manifest文件导入。

dataset.import_data(path=None, anntation_config=None, **kwargs)

不同类型的数据集支持的导入方式如表6-19所示。

表 6-19 不同数据集支持的导入方式

数据集类型	OBS目录 导入	Manifest文 件导入	备注
图像分类	支持	支持	-
物体检测	支持	支持	-
图像分割	支持	支持	-
文本分类	支持	支持	-
命名实体	不支持	支持	-
文本三元组	不支持	支持	-
声音分类	支持	支持	-
语音内容	不支持	支持	-
语音分割	不支持	支持	-
表格数据集	支持	不支持	新导入的表格数据的schema 和数据集一致。
视频标注	不支持	支持	-

示例代码

• 示例一: 物体检测数据集目录导入

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

dataset = Dataset(session, dataset_id)
annotation_config = dict()
annotation_config['scene'] = "object_detection"
annotation_config['format_name'] = "ModelArts PASCAL VOC 1.0"
import_resp = dataset.import_data(path="/obs-gaia-test/data/image/image-detection/",
annotation_config=annotation_config)

● 示例二:物体检测数据集Manifest文件导入

annotation_config = dict() # Manifest文件导入任务中,传入annotation_config参数可以导入标注信息 import_resp = dataset.import_data(

path="/obs-gaia-test/data/output/work_path/dataset-5932-Qdd1RUZ3wqBQrwrTr3v/annotation/V001/V001.manifest",annotation_config=annotation_config)

• 示例三:表格数据集目录导入

import_resp = dataset.import_data(

path="/obs-gaia-test/data/table/table1/", with_column_header=True)

表 6-20 请求参数

参数	是否 必选	参数类型	描述
path	是	String	导入的OBS路径或manifest路径。
			● 导入manifest时,path必须精确到具体 manifest文件。
			导入为目录时,目前仅支持数据集类型为 图片分类、物体检测、图像分割、文本分 类、声音分类和表格数据集。
			● 字符限制:不允许出现的特殊字符有换行符(\n)、回车符(\r)、制表符(\t)。
annotation_co nfig	否	表6-5	数据标注格式的说明。该字段为None,则不导入标注信息。如果根据Manifest文件导入,可以传入一个内容为空的dict对象实现导入标注信息。目前支持的标注格式类型如下:
			● 图像分类
			● 物体检测
			● 语音分类
			• 文本分类
with_column_ header	否	Boolean	表格数据集必选参数,表格的第一行是否为 表头。
			● True: 第一行数据作为表头
			• False:第一行数据不作为表头,仅为样本数据

6.4.3 查询导入任务状态

根据任务ID查询数据集导入任务的状态和详情。

dataset.get_import_task_info(task_id)

示例代码

查询数据集导入任务的详情

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

dataset = Dataset(session, dataset_id)
task_info = dataset.get_import_task_info(task_id="r4R52nJ4VJKcivuioCU")
print(task_info) # 打印导入任务的详细信息

表 6-21 请求参数

参数	是否 必选	参数类型	描述
task_id	是	String	导入任务的任务ID。

6.5 导出任务管理

6.5.1 查询导出任务列表

查询数据集导出任务列表。

dataset.list_export_tasks()

示例代码

查询数据集导出任务列表

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

dataset = Dataset(session, dataset_id) list_tasks_resp = dataset.list_export_tasks() print(list_tasks_resp) # 打印导出任务列表

参数说明

无。

6.5.2 创建导出任务

将当前数据集的样本导出到指定的OBS路径下。仅支持图像分类、物体检测、图像分割和自由格式数据集。

dataset.export_data(path)

示例代码

导出数据集到OBS目录

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

dataset = Dataset(session, dataset_id)
export_resp = dataset.export_data("/obs-gaia-test/data/output/export-test/")

表 6-22 请求参数

参数	是否 必选	参数类型	描述
path	是	String	数据导出的OBS路径。

6.5.3 查询导出任务状态

根据任务ID查询数据集导出任务的状态和详情。

dataset.get_export_task_info(task_id)

示例代码

查询数据集导出任务状态

from modelarts.session import Session from modelarts.dataset import Dataset session = Session()

dataset = Dataset(session, dataset_id)
task_info = dataset.get_export_task_info(task_id="iuHALF6xdkSAGKVN2jD")
print(task_info) # 查询导出任务的详细信息

参数说明

表 6-23 请求参数

参数	是否 必选	参数类型	描述
task_id	是	String	导出任务的任务ID。

6.6 Manifest 管理

6.6.1 Manifest 管理概述

在ModelArts使用过程中,需要做数据标注、模型训练、推理、数据集管理、市场发布等业务,这些业务都基于数据集进行的。为了规范对数据集的使用,适配各个使用场景,同时兼顾数据集管理的灵活性,本文档描述数据集管理的接口和描述规范——Manifest文件。

- Manifest文件中定义了标注对象和标注内容的对应关系。Manifest文件中也可以 只有原始文件信息,没有标注信息,如生成的未标注的数据集。
- Manifest文件使用UTF-8编码,Manifest处理程序需具备UTF-8处理能力。
- Manifest文件中文本分类的source数值可以包含中文,其他字段不建议用中文。

- Manifest文件可以由用户、第三方工具或ModelArts标注系统生成。
- Manifest文件名没有特殊要求,可以为任意合法文件名。

6.6.2 解析 Manifest 文件

解析manifest文件,支持本地和OBS。如果是OBS,需要Session信息。

manifest.parse_manifest(manifest_path, encoding='utf-8')

示例代码

通过Manifest路径来解析获取Manifest的信息。

from modelarts.session import Session from modelarts.dataset.format.manifest import Manifest

path = "obs://your-obs-bucket/manifest/V001.manifest"
session = Session()
manifest_info= Manifest.parse_manifest(path,session=session)

参数说明

表 6-24 请求参数

参数	是否必选	参数类型	描述
manifest_path	是	String	manifest文件路径,支持OBS和本地路径。 如果是OBS,需要Session信息。
encoding	否	String	文件编码格式,默认为utf-8。

表 6-25 manifest_info 返回参数说明

参数	参数类型	描述
size	Long	样本数量。
samples	JSON Array	样本列表。样本属性请见 <mark>表6-26</mark> 。

表 6-26 sample 样本属性

参数	参数类型	描述
source	String	被标注对象的URI,支持的scheme有OBS、 HTTPS、Content。其中Content为文本内 容,例如: "source": "s3://path-to- jpg","source": "content://I love machine learning"。

参数	参数类型	描述
annotations	JSON Array	样本标注信息。若不给出,则是未标注对 象。
		annotations值为一个对象列表。标注属性 请见 <mark>表6-27</mark> 。
usage	String	用途,可选值为TRAIN、EVAL、TEST、 INFERENCE。指明该对象用于训练、评 估、测试、推理,若没有给出该字段,则使 用者自行决定如何使用该对象。
inference_loc	String	当此manifest文件由推理服务生成时会有该 字段,表示推理输出的结果文件位置。
id	String	样本ID。
source_type	String	source的类型,比如csv。
source_property	String	source的属性。
hard	Boolean	是否是难例,true表示是难例,false为非难例。
hard_coefficient	Double	难度系数,范围为[0,1]。
hard_reasons	String	标签级别难例原因。通过中划线间隔单个难 例原因ID。
source_map	String	source的映射。

表 6-27 annotation 标注属性

参数	参数类型	描述
name	String	标注名称。
type	String	标注类型。
id	String	标注ID。
annotation_loc	String	标注文件的云存储路径,对于物体检测是必 选字段,对于其他类型是可选字段。
annotation_pro perty	String	标注属性。
confidence	Double	置信度,数值类型,范围 0<=confidence<=1,表示机器标注的置信 度。
creation_time	String	创建该标注的时间。是用户写入标注的时间,不是manifest生成时间。
annotated_by	String	标注人。

参数	参数类型	描述
annotation_for mat	String	描述标注文件的格式。默认为"PASCAL VOC"。
hard	Boolean	是否是难例。
hard_coefficient	Double	难度系数。
annotation_loc_ map	String	标注文件路径的映射。

6.6.3 创建和保存 Manifest 文件

需要先创建包含Manifest信息的对象,然后保存。Manifest信息请见表6-25。路径支持本地和OBS,如果是OBS,需要Session信息。

manifest_info.save(path, session=None, save_mode="w")

示例代码

在保存Manifest文件之前需要先创建包含Manifest信息的对象,包括Sample样本信息 及其标签信息Annotation,然后将诺干个样本组成Manifest。保存的时候调用save接 口,将session信息传入,即可保存到指定路径。

```
from modelarts.dataset.format.manifest.annotation import Annotation
from modelarts.dataset.format.manifest import Manifest
from modelarts.dataset.format.manifest.sample import Sample
from modelarts.session import Session
size = 0
sample_list = []
for i in range(19):
  size = size + 1
  source = "s3://obs-path/examples/image-classification/data/image_" + str(i) + ".jpg"
  usage = "TRAIN"
  inference_loc = "s3://obs-path/examples/image-classification/data/image_" + str(i) + ".txt"
  annotations_list = []
  for j in range(1):
     annotation_type = "modelarts/image_classification"
     if 0 == i % 2:
       annotation_name = "Cat"
       annotation_name = "Dog"
     annotation_creation_time = "2019-02-20 08:23:06"
     annotation format = "manifest"
     annotation_property = {"color": "black"}
     annotation_confidence = 0.8
     annotated_by = "human"
     annotations_list.append(
       Annotation(name=annotation_name, type=annotation_type,
               confidence=annotation_confidence,
               creation time=annotation creation time,
               annotated\_by = annotated\_by, \ annotation\_format = annotation\_format,
               annotation_property=annotation_property))
  sample list.append(
     Sample(source=source, usage=usage, annotations=annotations_list, inference_loc=inference_loc))
manifest_info = Manifest(samples=sample_list, size=size)
path = "obs://your-obs-bucket/manifest/V001.manifest"
session = Session()
manifest_info.save(path, session=session, save_mode="a")
```

表 6-28 请求参数

参数	是否必选	参数类型	描述
path	是	String	manifest文件保存路径。
session	否	Object	会话对象,初始化方法请参见 Session鉴权 。 当需要操作OBS时必填。
save_mode	否	String	保存模式。默认为w,即重写模式,另外还 支持a,为追加模式。

6.6.4 解析 Pascal VOC 文件

解析xml文件支持本地和OBS,如果是OBS,需要Session信息。

PascalVoc.parse_xml(xml_file_path, session=None)

示例代码

指定xml路径,通过调用parse_xml来解析获取xml文件的信息。

from modelarts.dataset.format.voc.pascal_voc import PascalVoc from modelarts.session import Session

path = "obs://your-obs-bucket/voc/test.xml" session = Session() pascal_voc = PascalVoc.parse_xml(path, session=session) print(pascal_voc) # 打印解析结果

参数说明

表 6-29 请求参数

参数	是否必选	参数类型	描述
xml_file_path	是	String	xml文件路径。
session	否	Object	会话对象,初始化方法请参见 Session鉴权 。 当需要操作OBS时必填。

表 6-30 pascal_voc 返回参数

参数	参数类型	描述
folder	String	文件夹名称。
file_name	String	文件名称。

参数	参数类型	描述
source	Object	数据源信息,详细请见 <mark>表6-31</mark> 。
width	Long	图片长度。
height	Long	图片高度。
depth	Long	图片深度。
segmented	String	分割。
mask_source	String	图像分割得到的mask文件的云存储路径, 目前只支持PNG格式。
voc_objects	JSON Array	标注对象列表,详细请见 <mark>表6-32</mark> 。

表 6-31 source 参数

参数	参数类型	描述
database	String	数据集名称,比如"The VOC2007 Database"。
annotation	String	标注,比如"PASCAL VOC2007"。
image	String	图片信息。

表 6-32 voc_object 参数

参数	参数类型	描述
name	String	文件夹名称。
properties	JSON Array	标注对象属性,为key-value列表格式,其 中key和value的值均为String类型。
pose	String	标注内容的拍摄角度。
truncated	String	标注内容是否被截断(0表示完整)。
occluded	String	标注内容是否被遮挡(0表示未遮挡)。
difficult	String	标注目标是否难以识别(0表示容易识 别)。
confidence	Double	置信度,数值类型,范围 0<=confidence<=1,表示机器标注的置信 度。
position	Object	标注对象的位置信息,详细请见表6-33。
parts	Object	子标注对象列表,即嵌套的voc_object列表,详细请见 <mark>表6-32</mark> 。

参数	参数类型	描述	
mask_color	String	图像分割mask图像的颜色。	

表 6-33 Position 说明

type	形状	标注信息
point	点	点的坐标
		<x>100<x></x></x>
		<y>100<y></y></y>
line	线	各点坐标
		<x1>100<x1></x1></x1>
		<y1>100<y1></y1></y1>
		<x2>200<x2></x2></x2>
		<y2>200<y2></y2></y2>
bndbox	矩形框	左下和右上两个点坐标
		<x_min>100<x_min></x_min></x_min>
		<y_min>100<y_min></y_min></y_min>
		<x_max>200<x_max></x_max></x_max>
		<y_max>200<y_max></y_max></y_max>
polygon	多边形	各点坐标
		<x1>100<x1></x1></x1>
		<y1>100<y1></y1></y1>
		<x2>200<x2></x2></x2>
		<y2>100<y2></y2></y2>
		<x3>250<x3></x3></x3>
		<y3>150<y3></y3></y3>
		<x4>200<x4></x4></x4>
		<y4>200<y4></y4></y4>
		<x5>100<x5></x5></x5>
		<y5>200<y5></y5></y5>
		<x6>50<x6></x6></x6>
		<y6>150<y6></y6></y6>
circle	圆形	圆心坐标和半径
		<cx>100<cx></cx></cx>
		<cy>100<cy></cy></cy>
		<r>50<r></r></r>

6.6.5 创建和保存 Pascal VOC 文件

需要先创建包含Pascal VOC信息的对象,然后保存。Pascal VOC信息请见<mark>表6-30</mark>。路径支持本地和OBS,如果是OBS,需要Session信息。

pascal_voc.save_xml(xml_file_path, save_mode='w', session=None)

示例代码

在保存Pascal VOC的XML文件之前需要先创建包含Pascal VOC信息的对象,包括voc object信息等。保存的时候调用save_xml接口,将session信息传入,即可保存到指定路径。

```
from modelarts.dataset.format.voc.pascal_voc import PascalVoc
from modelarts.dataset.format.voc.voc_object import VocObject
from modelarts.session import Session
path = "obs://your-obs-bucket/voc/test2.xml"
size list = [640, 321, 3]
file_name = "000000089955.jpg"
voc_object_tags = ["trafficlight", "trafficlight"]
"@modelarts:shape": "bndbox", "@modelarts:feature": [[347, 186], [382, 249]]}, {"@modelarts:color": "#FFFE0", "@modelarts:shortcut": "D",
                "@modelarts:shape": "bndbox", "@modelarts:feature": [[544, 50], [591, 149]]}]
voc_objects = []
for i in range(len(voc_object_tags)):
  object_tag = voc_object_tags[i]
  object_properties = voc_object_properties[i]
  voc_objects.append(VocObject(name=object_tag, properties=object_properties))
pascal_voc = PascalVoc(file_name=file_name, width=size_list[0], height=size_list[1], depth=size_list[2],
              voc_objects=voc_objects)
session = Session()
pascal_voc.save_xml(path, session=session)
```

参数说明

表 6-34 请求参数

参数	是否 必选	参数类型	描述
xml_file_path	是	String	Pascal VOC格式的XML文件保存路径。
session	否	Object	会话对象,初始化方法请参见 Session鉴权 。 当需要操作OBS时必填。
save_mode	否	String	保存模式。默认为w,即重写模式,另外还 支持a,为追加模式。

6.7 标注任务管理

6.7.1 创建标注任务

基于数据集创建标注任务。

dataset.create_label_task(self, task_name=None, task_type=None, **kwargs)

示例代码

示例一:基于图像类型的数据集创建物体检测标注任务。

from modelarts.session import Session from modelarts.dataset import Dataset

session = Session()

dataset = Dataset (session, dataset_id="VukxA2FlaTUm7tkDtq0") # 初始化数据集
create_task_resp = dataset.create_label_task(task_name="obj_detection_task", task_type=1,
description="label task")

参数说明

表 6-35 请求参数

参数	是否 必选	参数类型	描述
task_name	是	String	标注任务的名称。
task_type	是	Integer	标注任务的类型。可选值如下:
description	否	String	标注任务的描述信息。

6.7.2 查询数据集的标注任务列表

查询当前数据集的所有标注任务列表。

dataset.get_label_tasks(is_workforce_task=False, **kwargs)

示例代码

● 示例一:查询数据集下所有的标注任务,根据标注任务创建时间降序排序。 from modelarts.session import Session from modelarts.dataset import Dataset session = Session()
dataset = Dataset(session,dataset_id="VukxA2FlaTUm7tkDtq0")
list_label_task_resp = dataset.get_label_tasks(sort_key="create_time", sort_dir="desc")
print(list_label_task_resp)

示例二: 查询数据集下所有的团队标注任务。
 list_label_task_resp = dataset.get_label_tasks(is_workforce_task=True)
 print(list_label_task_resp)

参数说明

表 6-36 请求参数

参数	是否 必选	参数类型	描述
is_workforce_t ask	否	Boolean	过滤条件,是否只获取团队标注任务。 • True: 只查询团队标注任务 • False: 默认值,查询所有标注任务
sort_key	否	String	排序字段。可选值如下: • create_time:根据创建时间排序 • task_name:根据任务名称进行排序
sort_dir	否	String	排序方式。可选值如下: • asc: 按照升序排序 • desc: 默认值,按照降序排序

6.7.3 查询标注任务详情

查询标注任务的详细信息。

datset.get_label_task_info(task_id=None)

示例代码

查询标注任务的详情。

 $task_info = dataset.get_label_task_info(task_id="xs9ZKzLluKzccQfsyi2") \\ print(task_info)$

参数说明

表 6-37 请求参数

参数	是否 必选	参数类型	描述
task_id	是	String	标注任务的ID。

了 训练管理(推荐)

7.1 训练作业

7.1.1 创建训练作业

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 示例一: 提交常用框架训练作业

Estimator中同时指定framework_type和framework_version,会提交一个常用框架训练作业。

```
from modelarts.session import Session
from modelarts.train_params import TrainingFiles
from modelarts.train_params import OutputData
from modelarts.train_params import InputData
from modelarts.estimatorV2 import Estimator
session = Session()
#训练脚本里接收的参数,请根据实际情况填写
parameters = [{"name": "mod", "value":"gpu"}]
parameters = [{"name": "epoc_num", "value":2}]
estimator = Estimator(session=session,
               training_files=TrainingFiles(code_dir= "obs://bucket_name/code_dir/",
boot file="boot file.py"),
               outputs=[OutputData(obs_path="obs://bucket_name/output/", name="output_dir")],
               parameters=parameters,
               framework_type='PyTorch',
                                                          # 常用框架类型
               framework_version='PyTorch-1.4.0-python3.6', # 常用框架版本
               train_instance_type="modelarts.p3.large.public",
               train_instance_count=1,
               log url="obs://bucket name/log/",
               env_variables={"USER_ENV_VAR": "customize environment variable"},
               job_description='This is a image net train job')
job_instance = estimator.fit(inputs=[InputData(obs_path="obs://bucket_name/input/",
name="data_url")],
                    job_name="job_name_1")
```

● 示例二: 提交自定义镜像训练作业

Estimator中同时指定user_image_url和user_command,会提交自定义镜像训练作业,并使用自定义启动命令来启动训练任务。

```
from modelarts.session import Session
from modelarts.train_params import TrainingFiles
from modelarts.train_params import OutputData
from modelarts.train_params import InputData
from modelarts.estimatorV2 import Estimator
session = Session()
#训练脚本里接收的参数,请根据实际情况填写
parameters = [{"name": "mod", "value":"gpu"}]
parameters = [{"name": "epoc_num", "value": 2}]
estimator = Estimator(session=session,
              training_files=TrainingFiles(code_dir= "obs://bucket_name/code_dir/",
boot_file="boot_file.py")
              outputs=[OutputData(obs_path="obs://bucket_name/output/", name="output_dir")],
              parameters=parameters,
              user_image_url="sdk-test/pytorch1_4:1.0.1",
                                                          # 自定义镜像地址
              user_command="/home/ma-user/anaconda3/envs/PyTorch-1.4/bin/python /home/ma-
user/modelarts/user-job-dir/train/test-pytorch.py", # 自定义启动命令
              train_instance_type="modelarts.p3.large.public",
              train_instance_count=1,
              log_url="obs://bucket_name/log/",
              env_variables={"USER_ENV_VAR": "customize environment variable"},
              job description='This is a image net train job')
job_instance = estimator.fit(inputs=[InputData(obs_path="obs://bucket_name/input/",
name="data_url")],
                   job_name="job_name_2")
```

● 示例三:向专属资源池提交训练作业

```
from modelarts.session import Session
from modelarts.train_params import TrainingFiles
from modelarts.train_params import OutputData
from modelarts.train_params import InputData
from modelarts.estimatorV2 import Estimator
session = Session()
#训练脚本里接收的参数,请根据实际情况填写
parameters = [{"name": "mod", "value":"gpu"}]
parameters = [{"name": "epoc_num", "value":2}]
estimator = Estimator(session=session,
              training_files=TrainingFiles(code_dir= "obs://bucket_name/code_dir/",
boot_file="boot_file.py")
              outputs=[OutputData(obs_path="obs://bucket_name/output/", name="output_dir")],
               parameters=parameters,
               framework_type='PyTorch',
              framework_version='PyTorch-1.4.0-python3.6',
                                                           # 专属资源池id
              pool_id="your pool id",
               train_instance_type="modelarts.pool.visual.xlarge", # 专属池的虚拟子规格
              train_instance_count=1,
              log_url="obs://bucket_name/log/"
              env_variables={"USER_ENV_VAR": "customize environment variable"},
              job_description='This is a image net train job')
job_instance = estimator.fit(inputs=[InputData(obs_path="obs://bucket_name/input/",
name="data_url")],
                   job_name="job_name_3")
```

● 示例四:使用数据集创建训练作业

framework_type='PyTorch', #常用框架类型framework_version='PyTorch-1.4.0-python3.6', #常用框架版本train_instance_type="modelarts.p3.large.public", train_instance_count=1, log_url="obs://bucket_name/log/", job_description='This is a image net train job') job_instance = estimator.fit(dataset_id="your dataset id", dataset_version_id="your dataset version id", job_name="job_name_5")

参数说明

表 7-1 Estimator 请求参数说明

参数	是否必 选	类型	描述
session	是	Object	会话对象,初始化方法请参考 <mark>Session</mark> <mark>鉴权</mark> 。
training_files	否	TrainingFiles Object	训练脚本在OBS中的路径,具体请参 考 TrainingFiles初始化 。
outputs	否	Array of OutputData objects	训练的输出位置信息,具体请参考 OutputData初始化。
parameters	否	JSON Array	训练任务的运行参数。格式为
			[{"name":"your name", "value": "your value"}],value的值可以是 string,也可以是int。
train_instance_ type	是	String	训练作业选择的资源规格,请参考 <mark>查</mark> 询资源规格列表
train_instance_ count	是	int	训练作业计算节点个数。
framework_typ e	否	String	训练作业选择的引擎规格,请参考 <mark>查</mark> 询引擎规格列表 。
framework_ver sion	否	String	训练作业选择的引擎版本,请参考 <mark>查</mark> 询引擎规格列表 。
user_image_url	否	String	自定义镜像训练作业的自定义镜像的 SWR-URL。
user_command	否	String	自定义镜像训练作业的启动命令。
log_url	否	String	训练作业日志的保存位置,是一个 OBS路径,如"obs://xx/yy/zz/"。
job_description	否	String	训练作业的描述。

参数	是否必选	类型	描述
volumes	否	JSON Array	训练作业挂载卷信息,格式参考: [{ "nfs": { "local_path": "/xx/yy/zz", "read_only": False, "nfs_server_path": "xxx.xxx.xxx.xxx:/" } }]
env_variables	否	Dict	训练作业的环境变量。
pool_id	否	String	训练作业选择的资源池ID。可在 ModelArts管理控制台,单击左侧"专 属资源池",在专属资源池列表中查 看资源池ID。

表 7-2 TrainingFiles 初始化参数说明

参数	是否必 选	类型	描述
code_dir	是	String	训练作业的代码目录,是一个OBS路 径,需要以"obs:/"开头,如"obs://xx/ yy/"。
boot_file	是	String	训练作业的代码启动文件,需要在代码目录下,可填写相对路径,如 "boot_file.py",也可填写绝对路径,如"obs://xx/yy/boot_file.py"。

表 7-3 OutputData 初始化参数说明

参数	是否必选	类型	描述
obs_path	是	String	数据实际输出到OBS的路径。
name	是	String	数据输出通道名称。

表 7-4 fit 请求参数说明

参数	是否必 选	类型	描述
inputs	否	Array of InputData Object	保存在OBS中的训练作业输入数据。 inputs和(dataset_id, dataset_version_id)两者不可同时出 现。
wait	否	Boolean	是否等待训练作业结束,默认为 False。
job_name	否	String	训练作业名称。
show_log	否	Boolean	作业提交成功后,是否输出训练作业 的日志,默认为False。
dataset_id	否	String	训练作业的数据集ID,可参考 <mark>数据管</mark> <mark>理</mark> 获取。需要与dataset_version_id同 时出现,但是不可与inputs同时出现。
dataset_versio n_id	否	String	训练作业的数据集版本ID,可参考 <mark>数</mark> 据管理获取。需要与dataset_id同时出现,但是不可与inputs同时出现。

表 7-5 InputData 初始化参数说明

参数	是否必 选	类型	描述
obs_path	是	String	训练作业需要的数据集OBS路径,如 "obs://xx/yy/"。
name	是	String	数据输入通道名称。

表 7-6 训练作业创建成功响应说明

参数	类型	描述
TrainingJob	Object	训练对象,该对象包含job_id等属性,对训练作业的查询、更新、删除等操作时,可通过job_instance.job_id获取训练作业ID。

表 7-7 调用训练接口失败响应参数

参数	类型	描述
error_msg	String	调用失败时的错误信息,调用成功时无此字段。

参数	类型	描述
error_code	String	调用失败时的错误码,具体请参见 <mark>错误码</mark> ,调用成功时 无此字段。
error_solution	String	调用失败时的提示解决信息,调用成功时无此字段。

7.1.2 查询训练作业列表

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

参数说明

表 7-8 get_job_list 请求参数说明

参数	是否必选	参数类型	描述
session	是	Object	会话对象,初始化方法请参考 Session <mark>鉴权</mark> 。
offset	否	Integer	查询作业的偏移量,最小为0。 例如设置为1,则表示从第二条 开始查。
limit	否	Integer	查询作业的限制量。最小为1, 最大为50。
sort_by	否	String	查询作业排列顺序的指标。默认 使用create_time排序。
order	否	String	查询作业排列顺序,默认为 "desc",降序排序。也可以选 择对应的"asc",升序排序。 缺省值: desc 枚举值: • asc: 升序 • desc: 降序
group_by	否	String	查询作业要搜索的分组条件。

参数	是否必选	参数类型	描述
filters	否	Array of objects	查询作业要过滤的一系列条件。

表 7-9 filters

参数	是否必选	参数类型	描述
key	否	String	分组条件键值。
operator	否	String	分组条件键值键关系,支持between、like、in、not、rely_to。 缺省值: in 枚举值: like: 类似 in: 包含 not: 非 between: 范围
value	否	Array of strings	分组条件键对应值。

表 7-10 get_job_list 返回参数说明

参数	参数类型	描述
total	Integer	查询到当前用户名下的所 有作业总数。
count	Integer	查询到当前用户名下的所 有符合查询条件的作业总 数。
limit	Integer	查询作业的限制量。最小 为1,最大为50。
offset	Integer	查询作业的偏移量,最小 为0。例如设置为1,则表 示从第二条开始查。
sort_by	String	查询作业排列顺序的指 标。默认使用create_time 排序。

参数	参数类型	描述
order	String	查询作业排列顺序,默认 为"desc",降序排序。 也可以选择对应的 "asc",升序排序。
group_by	String	查询作业要搜索的分组条 件。
workspace_id	String	作业所处的工作空间,默 认值为"0"。
ai_project	String	作业所属的AI项目,默认 值为"default-ai- project"。
items	Array of JobResponse objects	查询到当前用户名下的所 有符合查询条件的作业详 情。

表 **7-11** JobResponse

参数	参数类型	描述
kind	String	训练作业类型。默认使用 job。 枚举值: • job: 训练作业 • hetero_job: 异构作业 • autosearch_job: 自动搜索作业 • mrs_job: mrs作业 • edge_job: 边缘作业
metadata	JobMetadata object	训练作业元信息。
status	Status object	训练作业状态信息。创建 作业无需填写。
algorithm	JobAlgorithmResponse object	训练作业算法。目前支持 三种形式: id只取算法的id; subscription_id +item_version_id取算 法的订阅id和版本id; code_dir+boot_file取 训练作业的代码目录和 启动文件。

参数	参数类型	描述
tasks	Array of TaskResponse objects	异构训练作业的任务列 表。
spec	spec object	训练作业规格参数。

表 7-12 JobMetadata

参数	参数类型	描述
id	String	训练作业ID,创建成功后 由ModelArts生成返回, 无需填写。
name	String	训练作业名称。限制为 1-64位只含数字、字母、 下划线和中划线的名称。
workspace_id	String	指定作业所处的工作空 间,默认值为"0"。
description	String	对训练作业的描述,默认为"NULL",字符串的长度限制为[0, 256]。
create_time	Long	训练作业创建时间戳,单 位为毫秒,创建成功后由 ModelArts生成返回,无 需填写。
user_name	String	训练作业创建用户的用户 名,创建成功后由 ModelArts生成返回,无 需填写。
annotations	Map <string,string></string,string>	训练作业申明模板,异构 作业job_template字段默 认为Template RL,其余 默认为Template DL。

表 7-13 Status

参数	参数类型	描述
phase	String	训练作业一级状态,状态 值稳定不变,可选值如 下: "Creating"、 "Pending"、 "Running"、 "Failed"、 "Completed"、 "Terminating"、 "Terminated"、
secondary_phase	String	训练作业二级状态,状态 值不稳定,可选值如下: "Creating"、 "Queuing"、 "Running"、 "Failed"、 "Completed"、 "Terminating"、 "Terminated"、 "CreateFailed"、 "TerminatedFailed"、 "Unknown"、
duration	Long	训练作业运行时长,单位 为毫秒。
node_count_metrics	Array <array<integer>></array<integer>	训练作业运行时节点数变 化指标。
tasks	Array of strings	训练作业子任务名称。
start_time	String	训练作业开始时间,格式 为时间戳。
task_statuses	Array of objects	训练在子任务状态信息。

表 7-14 task_statuses

参数	参数类型	描述
task	String	训练作业子任务名称。
exit_code	Integer	训练作业子任务退出码。
message	String	训练作业子任务错误消 息。

表 7-15 JobAlgorithmResponse

参数	参数类型	描述
id	String	算法管理的算法id。 枚举值: id只取算法的id; subscription_id +item_version_id取算 法的订阅id和版本id; code_dir+boot_file取 训练作业的代码目录和 启动文件。
name	String	算法名称。
subscription_id	String	订阅算法的订阅ID。应与 item_version_id一同出 现。
item_version_id	String	订阅算法的版本。应与 subscription_id一同出 现。
code_dir	String	训练作业的代码目录。 如: "/usr/app/"。应与 boot_file一同出现,若填 入id或subscription_id +item_version_id则无需 填写。
boot_file	String	训练作业的代码启动文件,需要在代码目录下。如: "/usr/app/boot.py"。应与code_dir一同出现,若填入id或subscription_id+item_version_id则无需填写。
autosearch_config_path	String	自动化搜索作业的yaml配 置路径,需要提供一个 OBS路径。
autosearch_framework_p ath	String	自动化搜索作业的框架代 码目录,需要提供一个 OBS路径。
command	String	自定义镜像训练作业的自 定义镜像的容器的启动命 令。可填code_dir。
parameters	Array of Parameter objects	训练作业的运行参数。

参数	参数类型	描述
policies	policies object	作业支持的策略。
inputs	Array of Input objects	训练作业的数据输入。
outputs	Array of Output objects	训练作业的结果输出。
engine	engine object	训练作业的引擎。使用算 法管理的算法id或订阅算 法subscription_id +item_version_id创建作 业时,无需填写。
environments	Map <string,string></string,string>	训练作业的环境变量。格 式: "key":"value",无需 填写。

表 7-16 Parameter

参数	参数类型	描述
name	String	参数名称。
value	String	参数值。
description	String	参数描述信息。
constraint	constraint object	参数属性。
i18n_descripti on	i18n_descript ion object	国际化描述。

表 7-17 constraint

参数	参数类型	描述
type	String	参数种类。
editable	Boolean	是否可编辑。
required	Boolean	是否必须。
sensitive	Boolean	是否敏感。
valid_type	String	有效种类。
valid_range	Array of strings	有效范围。

表 7-18 i18n_description

参数	参数类型	描述
language	String	国际语种。
description	String	对应描述。

表 7-19 policies

参数	参数类型	描述
auto_search	auto_search object	超参搜索配置。

表 7-20 auto_search

参数	参数类型	描述
skip_search_p arams	String	需要排除的超参组合。
reward_attrs	Array of objects	搜索指标列表。
search_param s	Array of objects	搜索参数。
algo_configs	Array of objects	搜索算法配置。

表 7-21 reward_attrs

参数	参数类型	描述
name	String	指标名称。
mode	String	搜索方向。 • max: 指定时,表示指标值越大越好;
		● min: 指定时,表示指标值越小越好。
regex	String	指标正则表达式。

表 7-22 search_params

参数	参数类型	描述
name	String	超参名称。

参数	参数类型	描述
param_type	String	参数类型。
		● continuous:指定时,表示参数类型为连续 值;
		● discreate:指定时,表示参数类型为离散值。
lower_bound	String	超参下界。
upper_bound	String	超参上界。
discrete_point s_num	String	连续型超参离散化取值个数。
discrete_value s	Array of strings	离散型超参的取值列表。

表 7-23 algo_configs

参数	参数类型	描述
name	String	搜索算法名称。
params	Array of AutoSearchA lgoConfigPar ameter objects	搜索算法参数。

表 7-24 AutoSearchAlgoConfigParameter

参数	参数类型	描述
key	String	参数键。
value	String	参数值。
type	String	参数种类。

表 7-25 Input

参数	参数类型	描述
name	String	数据输入通道名称。
description	String	数据输入通道描述信息。
local_dir	String	数据输入通道映射的容器本地路径。

参数	参数类型	描述
remote	InputDataInf o object	数据实际输入信息。
remote_constr aint	Array of objects	数据输入约束。

表 7-26 InputDataInfo

参数	参数类型	描述
dataset	dataset object	数据输入信息为数据集。
obs	obs object	数据输入输出信息为OBS方式。

表 7-27 dataset

参数	参数类型	描述
id	String	训练作业的数据集ID。
version_id	String	训练作业的数据集版本ID。
obs_url	String	训练作业需要的数据集OBS路径URL,ModelArts 会通过数据集ID和数据集版本ID自动解析生成。 如:"/usr/data/"。

表 7-28 obs

参数	参数类型	描述
obs_url	String	训练作业需要的数据集OBS路径URL。如: "/usr/data/"。

表 7-29 remote_constraint

参数	参数类型	描述
data_type	String	数据输入类型,包括数据存储位置、数据集两种 方式。

参数	参数类型	描述
attributes	String	数据输入为数据集时的相关属性。 枚举值:
		• data_format: 数据格式。
		• data_segmentation:数据切分方式。
		● dataset_type: 标注类型。

表 7-30 Output

参数	参数类型	描述
name	String	数据输出通道名称。
description	String	数据输出通道描述信息。
local_dir	String	数据输出通道映射的容器本地路径。
remote	remote object	数据实际输出信息。

表 7-31 remote

参数	参数类型	描述
obs	obs object	数据实际输出到OBS。

表 7-32 obs

参数	参数类型	描述
obs_url	String	数据实际输出到OBS的路径。

表 7-33 engine

参数	参数类型	描述
engine_id	String	训练作业选择的引擎规格ID。engine_id、 engine_name + engine_version和image_url方式 三选一。
engine_name	String	训练作业选择的引擎名称。若填入engine_id,则 无需填写。
engine_versio n	String	训练作业选择的引擎版本名称。若填入 engine_id,则无需填写。

参数	参数类型	描述
image_url	String	训练作业选择的自定义镜像地址。

表 7-34 TaskResponse

参数	参数类型	描述
role	String	异构训练作业的任务角色。 枚举值: • learner(支持GPU\CPU规格); • worker(支持CPU规格)。
algorithm	algorithm object	算法管理算法配置。
task_resource	FlavorRespon se object	训练作业、算法的规格信息。

表 7-35 algorithm

参数	参数类型	描述
code_dir	String	算法启动文件所在目录绝对路径。
boot_file	String	算法启动文件绝对路径。
inputs	inputs object	算法输入通道信息。
outputs	outputs object	算法输出通道信息。
engine	engine object	异构作业所依赖的引擎。

表 7-36 inputs

参数	参数类型	描述
name	String	数据输入通道名称。
local_dir	String	数据输入输出通道映射的容器本地路径。
remote	remote object	数据实际输入信息,异构作业只支持OBS。

表 7-37 remote

参数	参数类型	描述
obs	obs object	数据输入输出信息为OBS方式。

表 7-38 obs

参数	参数类型	描述
obs_url	String	训练作业需要的数据集OBS路径URL。如: "/usr/data/"。

表 7-39 outputs

参数	参数类型	描述
name	String	数据输出通道名称。
local_dir	String	数据输出通道映射的容器本地路径。
remote	remote object	数据实际输出信息。
mode	String	数据传输模式,默认为 "upload_periodically"。
period	String	数据传输周期,默认为30s。

表 7-40 remote

参数	参数类型	描述
obs	obs object	数据实际输出到OBS。

表 7-41 obs

参数	参数类型	描述
obs_url	String	数据实际输出到OBS的路径。

表 7-42 engine

参数	参数类型	描述
engine_id	String	异构作业引擎规格的ID。如"caffe-1.0.0- python2.7"。

参数	参数类型	描述
engine_name	String	异构作业引擎规格的名称。如"Caffe"。
engine_versio n	String	异构作业引擎规格的版本。
v1_compatibl e	Boolean	是否为v1兼容模式。
run_user	String	引擎默认启动用户uid。

表 7-43 FlavorResponse

参数	参数类型	描述
flavor_id	String	资源规格的ID。
flavor_name	String	资源规格的名称。
max_num	Integer	资源规格的最大节点数。
flavor_type	String	资源规格的类型。可选值如下:
		• CPU;
		• GPU;
		• Ascend。
billing	billing object	资源规格计费信息。
flavor_info	flavor_info object	资源规格详细信息。
attributes	Map <string,st ring></string,st 	其他规格属性。

表 7-44 billing

参数	参数类型	描述
code	String	计费码。
unit_num	Integer	计费卡数。

表 7-45 flavor_info

参数	参数类型	描述
max_num	Integer	可以选择的最大节点数量(max_num,为1代表 不支持分布式)。

参数	参数类型	描述
сри	cpu object	cpu规格信息。
gpu	gpu object	gpu规格信息。
npu	npu object	Ascend规格信息。
memory	memory object	内存信息。

表 7-46 cpu

参数	参数类型	描述
arch	String	cpu架构。
core_num	Integer	核数。

表 7-47 gpu

参数	参数类型	描述
unit_num	Integer	gpu卡数。
product_num e	String	产品名。
memory	String	内存。

表 7-48 npu

参数	参数类型	描述
unit_num	String	npu卡数。
product_name	String	产品名。
memory	String	内存。

表 7-49 memory

参数	参数类型	描述
size	Integer	内存大小。
unit	String	内存单元数。

表 7-50 spec

参数	参数类型	描述
resource	Resource object	训练作业资源规格信息。flavor_id和pool_id+ [flavor_id]方式二选一。
volumes	Array of objects	训练作业挂载卷信息。
log_export_pa th	log_export_p ath object	训练作业日志输出信息。

表 7-51 Resource

参数	参数类型	描述
policy	String	训练作业资源规格模式,可选值如下: "regular"、"economic"、"turbo"。
flavor_id	String	训练作业选择的资源规格ID。
flavor_name	String	使用flavor_id时,由ModelArts返回的只读规格名称。
node_count	Integer	训练作业选择的资源副本数。 最小值: 1
pool_id	String	训练作业选择的资源池ID。
flavor_detail	flavor_detail object	训练作业、算法的规格信息。

表 7-52 flavor_detail

参数	参数类型	描述
flavor_type	String	资源规格的类型。可选值如下:
		• CPU;
		• GPU;
		Ascend。
billing	billing object	资源规格计费信息。
flavor_info	flavor_info object	资源规格详细信息。

表 7-53 billing

参数	参数类型	描述
code	String	计费码。
unit_num	Integer	计费卡数。

表 7-54 flavor_info

参数	参数类型	描述
max_num	Integer	可以选择的最大节点数量(max_num,为1代表 不支持分布式)。
сри	cpu object	cpu规格信息。
gpu	gpu object	gpu规格信息。
npu	npu object	Ascend规格信息。
memory	memory object	内存信息。
disk	disk object	磁盘信息。

表 7-55 cpu

参数	参数类型	描述
arch	String	cpu架构。
core_num	Integer	核数。

表 7-56 gpu

参数	参数类型	描述
unit_num	Integer	gpu卡数。
product_num e	String	产品名。
memory	String	内存。

表 **7-57** npu

参数	参数类型	描述
unit_num	String	npu卡数。
product_name	String	产品名。
memory	String	内存。

表 **7-58** memory

参数	参数类型	描述
size	Integer	内存大小。
unit	String	内存单元数。

表 7-59 disk

参数	参数类型	描述
size	String	磁盘大小。
unit	String	磁盘大小单位,一般为GB。

表 7-60 volumes

参数	参数类型	描述
nfs	nfs object	nfs方式的挂载卷。

表 7-61 nfs

参数	参数类型	描述
nfs_server_pat h	String	nfs服务端路径。
local_path	String	挂载到训练容器中的路径。
read_only	Boolean	nfs挂载卷在容器中是否只读。

表 7-62 log_export_path

参数	参数类型	描述
obs_url	String	训练作业日志保存的OBS地址。
host_path	String	训练作业日志保存的宿主机的路径。

表 7-63 调用训练接口失败响应参数

参数	类型	描述
error_msg	String	调用失败时的错误信息,调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参见 <mark>错误码</mark> ,调用成功时 无此字段。
error_solution	String	调用失败时的提示解决信息,调用成功时无此字段。

表 7-64 调用训练接口失败响应参数

参数	类型	描述
error_msg	String	调用失败时的错误信息,调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参见 错误码 ,调用成功时 无此字段。
error_solution	String	调用失败时的提示解决信息,调用成功时无此字段。

7.1.3 查询训练作业详情

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 方式一:根据指定的job_id查询。

from modelarts.session import Session from modelarts.estimatorV2 import Estimator session = Session() estimator = Estimator(session=session, job_id="618222c4-dc2f-4cfe-bc49-72b075b7552f") job_info = estimator.get_job_info() print(job_info)

● 方式二: 根据创建训练作业生成的训练作业对象查询。

job_info = job_instance.get_job_info()
print(job_info)

参数说明

表 7-65 Estimator 请求参数说明

参数	是否必 选	参数类型	描述
session	是	Object	会话对象,初始化方法请参考 Session <mark>鉴权</mark> 。
job_id	是	String	训练作业的id,可通过 <mark>创建训练作业</mark> 生成的训练作业对象查询,如 "job_instance.job_id",或从 <mark>查询训</mark> 练作业列表 的响应中获得。

表 7-66 get_job_info 返回参数说明

参数	参数类型	描述
kind	String	训练作业类型。默认使用job。
		枚举值:
		● job: 训练作业
		● hetero_job: 异构作业
		● autosearch_job: 自动搜索作业
		● mrs_job: mrs作业
		● edge_job: 边缘作业
metadata	JobMetadata object	训练作业元信息。
status	Status object	训练作业状态信息。创建作业无需填写。
algorithm	JobAlgorithm	训练作业算法。目前支持三种形式:
	Response	● id只取算法的id;
	object	● subscription_id+item_version_id取算法的订 阅id和版本id;
		• code_dir+boot_file取训练作业的代码目录和 启动文件。
tasks	Array of TaskRespons e objects	异构训练作业的任务列表。
spec	spec object	训练作业规格参数。

表 7-67 JobMetadata

参数	参数类型	描述
id	String	训练作业ID,创建成功后由ModelArts生成返 回,无需填写。
name	String	训练作业名称。限制为1-64位只含数字、字母、 下划线和中划线的名称。
workspace_id	String	指定作业所处的工作空间,默认值为"0"。
description	String	对训练作业的描述,默认为"NULL",字符串的 长度限制为[0, 256]。
create_time	Long	训练作业创建时间戳,单位为毫秒,创建成功后 由ModelArts生成返回,无需填写。
user_name	String	训练作业创建用户的用户名,创建成功后由 ModelArts生成返回,无需填写。
annotations	Map <string,st ring></string,st 	训练作业申明模板,异构作业job_template字段 默认为Template RL,其余默认为Template DL。

表 7-68 Status

参数	参数类型	描述
phase	String	训练作业一级状态,状态值稳定不变,可选值如 下: "Creating"、"Pending"、 "Running"、"Failed"、"Completed"、 "Terminating"、"Terminated"、 "Abnormal"。
secondary_ph ase	String	训练作业二级状态,状态值不稳定,可选值如 下: "Creating"、"Queuing"、 "Running"、"Failed"、"Completed"、 "Terminating"、"Terminated"、 "CreateFailed"、"TerminatedFailed"、 "Unknown"、"Lost"。
duration	Long	训练作业运行时长,单位为毫秒。
node_count_ metrics	Array <array<l nteger="">></array<l>	训练作业运行时节点数变化指标。
tasks	Array of strings	训练作业子任务名称。
start_time	String	训练作业开始时间,格式为时间戳。
task_statuses	Array of objects	训练在子任务状态信息。

表 7-69 task_statuses

参数	参数类型	描述
task	String	训练作业子任务名称。
exit_code	Integer	训练作业子任务退出码。
message	String	训练作业子任务错误消息。

表 7-70 JobAlgorithmResponse

参数	参数类型	描述
id	String	算法管理的算法id。
		枚举值:
		● id: 只取算法的id;
		● subscription_id+item_version_id: 取算法的 订阅id和版本id;
		● code_dir+boot_file:取训练作业的代码目录 和启动文件。
name	String	算法名称。
subscription_i d	String	订阅算法的订阅ID。应与item_version_id一同出现。
item_version_i d	String	订阅算法的版本。应与subscription_id一同出现。
code_dir	String	训练作业的代码目录。如:"/usr/app/"。应与 boot_file一同出现,若填入id或subscription_id +item_version_id则无需填写。
boot_file	String	训练作业的代码启动文件,需要在代码目录下。 如: "/usr/app/boot.py"。应与code_dir一同出 现,若填入入id或subscription_id +item_version_id则无需填写。
autosearch_co nfig_path	String	自动化搜索作业的yaml配置路径,需要提供一个 OBS路径。
autosearch_fr amework_pat h	String	自动化搜索作业的框架代码目录,需要提供一个 OBS路径。
command	String	自定义镜像训练作业的自定义镜像的容器的启动命令。可填code_dir。
parameters	Array of Parameter objects	训练作业的运行参数。

参数	参数类型	描述
policies	policies object	作业支持的策略。
inputs	Array of Input objects	训练作业的数据输入。
outputs	Array of Output objects	训练作业的结果输出。
engine	engine object	训练作业的引擎。使用算法管理的算法id或订阅 算法subscription_id+item_version_id创建作业 时,无需填写。
environments	Map <string,st ring></string,st 	训练作业的环境变量。格式:"key":"value",无 需填写。

表 7-71 Parameter

参数	参数类型	描述
name	String	参数名称。
value	String	参数值。
description	String	参数描述信息。
constraint	constraint object	参数属性。
i18n_descripti on	i18n_descript ion object	国际化描述。

表 7-72 constraint

参数	参数类型	描述
type	String	参数种类。
editable	Boolean	是否可编辑。
required	Boolean	是否必须。
sensitive	Boolean	是否敏感。
valid_type	String	有效种类。
valid_range	Array of strings	有效范围。

表 7-73 i18n_description

参数	参数类型	描述
language	String	国际语种。
description	String	对应描述。

表 7-74 policies

参数	参数类型	描述
auto_search	auto_search object	超参搜索配置。

表 7-75 auto_search

参数	参数类型	描述
skip_search_p arams	String	需要排除的超参组合。
reward_attrs	Array of objects	搜索指标列表。
search_param s	Array of objects	搜索参数。
algo_configs	Array of objects	搜索算法配置。

表 7-76 reward_attrs

参数	参数类型	描述
name	String	指标名称。
mode	String	搜索方向。
		● max: 指定时,表示指标值越大越好;
		● min: 指定时,表示指标值越小越好。
regex	String	指标正则表达式。

表 7-77 search_params

参数	参数类型	描述
name	String	超参名称。

参数	参数类型	描述
param_type	String	参数类型。
		● continuous:指定时,表示参数类型为连续 值;
		● discreate:指定时,表示参数类型为离散值。
lower_bound	String	超参下界。
upper_bound	String	超参上界。
discrete_point s_num	String	连续型超参离散化取值个数。
discrete_value s	Array of strings	离散型超参的取值列表。

表 7-78 algo_configs

参数	参数类型	描述
name	String	搜索算法名称。
params	Array of AutoSearchA IgoConfigPar ameter objects	搜索算法参数。

表 7-79 AutoSearchAlgoConfigParameter

参数	参数类型	描述
key	String	参数键。
value	String	参数值。
type	String	参数种类。

表 7-80 Input

参数	参数类型	描述
name	String	数据输入通道名称。
description	String	数据输入通道描述信息。
local_dir	String	数据输入通道映射的容器本地路径。

参数	参数类型	描述
remote	InputDataInf o object	数据实际输入信息。
remote_constr aint	Array of objects	数据输入约束。

表 7-81 InputDataInfo

参数	参数类型	描述
dataset	dataset object	数据输入信息为数据集。
obs	obs object	数据输入输出信息为OBS方式。

表 7-82 dataset

参数	参数类型	描述
id	String	训练作业的数据集ID。
version_id	String	训练作业的数据集版本ID。
obs_url	String	训练作业需要的数据集OBS路径URL,modelarts 会通过数据集ID和数据集版本ID自动解析生成。 如:"/usr/data/"。

表 7-83 obs

参数	参数类型	描述
obs_url	String	训练作业需要的数据集OBS路径URL。如: "/usr/data/"。

表 **7-84** remote_constraint

参数	参数类型	描述
data_type	String	数据输入类型,包括数据存储位置、数据集两种 方式。

参数	参数类型	描述
attributes	String	数据输入为数据集时的相关属性。 枚举值:
		• data_format: 数据格式。
		• data_segmentation:数据切分方式。
		● dataset_type: 标注类型。

表 7-85 Output

参数	参数类型	描述
name	String	数据输出通道名称。
description	String	数据输出通道描述信息。
local_dir	String	数据输出通道映射的容器本地路径。
remote	remote object	数据实际输出信息。

表 7-86 remote

参数	参数类型	描述
obs	obs object	数据实际输出到OBS。

表 7-87 obs

参数	参数类型	描述
obs_url	String	数据实际输出到OBS的路径。

表 7-88 engine

参数	参数类型	描述
engine_id	String	训练作业选择的引擎规格ID。engine_id、 engine_name + engine_version和image_url方式 三选一。
engine_name	String	训练作业选择的引擎名称。若填入engine_id,则 无需填写。
engine_versio n	String	训练作业选择的引擎版本名称。若填入 engine_id,则无需填写。

参数	参数类型	描述
image_url	String	训练作业选择的自定义镜像地址。

表 7-89 TaskResponse

参数	参数类型	描述
role	String	异构训练作业的任务角色。 枚举值: ● learner(支持GPU\CPU规格) ● worker(支持CPU规格)
algorithm	algorithm object	算法管理算法配置。
task_resource	FlavorRespon se object	训练作业、算法的规格信息。

表 7-90 algorithm

参数	参数类型	描述
code_dir	String	算法启动文件所在目录绝对路径。
boot_file	String	算法启动文件绝对路径。
inputs	inputs object	算法输入通道信息。
outputs	outputs object	算法输出通道信息。
engine	engine object	异构作业所依赖的引擎。

表 7-91 inputs

参数	参数类型	描述
name	String	数据输入通道名称。
local_dir	String	数据输入输出通道映射的容器本地路径。
remote	remote object	数据实际输入信息,异构作业只支持OBS。

表 7-92 remote

参数	参数类型	描述
obs	obs object	数据输入输出信息为OBS方式。

表 7-93 obs

参数	参数类型	描述
obs_url	String	训练作业需要的数据集OBS路径URL。如: "/usr/data/"。

表 7-94 outputs

参数	参数类型	描述
name	String	数据输出通道名称。
local_dir	String	数据输出通道映射的容器本地路径。
remote	remote object	数据实际输出信息。
mode	String	数据传输模式,默认为 "upload_periodically"。
period	String	数据传输周期,默认为30s。

表 **7-95** remote

参数	参数类型	描述
obs	obs object	数据实际输出到OBS。

表 7-96 obs

参数	参数类型	描述
obs_url	String	数据实际输出到OBS的路径。

表 **7-97** engine

参数	参数类型	描述
engine_id	String	异构作业引擎规格的ID。如"caffe-1.0.0- python2.7"。

参数	参数类型	描述
engine_name	String	异构作业引擎规格的名称。如"Caffe"。
engine_versio n	String	异构作业引擎规格的版本。
v1_compatibl e	Boolean	是否为v1兼容模式。
run_user	String	引擎默认启动用户uid。

表 7-98 FlavorResponse

参数	参数类型	描述
flavor_id	String	资源规格的ID。
flavor_name	String	资源规格的名称。
max_num	Integer	资源规格的最大节点数。
flavor_type	String	资源规格的类型。可选值如下:
		• CPU;
		• GPU;
		• Ascend。
billing	billing object	资源规格计费信息。
flavor_info	flavor_info object	资源规格详细信息。
attributes	Map <string,st ring></string,st 	其他规格属性。

表 7-99 billing

参数	参数类型	描述
code	String	计费码。
unit_num	Integer	计费卡数。

表 7-100 flavor_info

参数	参数类型	描述
max_num	Integer	可以选择的最大节点数量(max_num,为1代表 不支持分布式)。

参数	参数类型	描述
сри	cpu object	cpu规格信息。
gpu	gpu object	gpu规格信息。
npu	npu object	Ascend规格信息。
memory	memory object	内存信息。

表 7-101 cpu

参数	参数类型	描述
arch	String	cpu架构。
core_num	Integer	核数。

表 7-102 gpu

参数	参数类型	描述
unit_num	Integer	gpu卡数。
product_num e	String	产品名。
memory	String	内存。

表 7-103 npu

参数	参数类型	描述
unit_num	String	npu卡数。
product_name	String	产品名。
memory	String	内存。

表 **7-104** memory

参数	参数类型	描述
size	Integer	内存大小。
unit	String	内存单元数。

表 **7-105** spec

参数	参数类型	描述
resource	Resource object	训练作业资源规格信息。flavor_id和pool_id+ [flavor_id]方式二选一。
volumes	Array of objects	训练作业挂载卷信息。
log_export_pa th	log_export_p ath object	训练作业日志输出信息。

表 7-106 Resource

参数	参数类型	描述
policy	String	训练作业资源规格模式,可选值如下: "regular"、"economic"、"turbo"。
flavor_id	String	训练作业选择的资源规格ID。
flavor_name	String	使用flavor_id时,由ModelArts返回的只读规格名称。
node_count	Integer	训练作业选择的资源副本数。 最小值: 1
pool_id	String	训练作业选择的资源池ID。
flavor_detail	flavor_detail object	训练作业、算法的规格信息。

表 7-107 flavor_detail

参数	参数类型	描述
flavor_type	String	资源规格的类型。可选值如下:
		• CPU;
		• GPU;
		Ascend。
billing	billing object	资源规格计费信息。
flavor_info	flavor_info object	资源规格详细信息。

表 **7-108** billing

参数	参数类型	描述
code	String	计费码。
unit_num	Integer	计费卡数。

表 7-109 flavor_info

参数	参数类型	描述
max_num	Integer	可以选择的最大节点数量(max_num,为1代表 不支持分布式)。
сри	cpu object	cpu规格信息。
gpu	gpu object	gpu规格信息。
npu	<mark>npu</mark> object	Ascend规格信息。
memory	memory object	内存信息。
disk	disk object	磁盘信息。

表 7-110 cpu

参数	参数类型	描述
arch	String	cpu架构。
core_num	Integer	核数。

表 **7-111** gpu

参数	参数类型	描述
unit_num	Integer	gpu卡数。
product_num e	String	产品名。
memory	String	内存。

表 **7-112** npu

参数	参数类型	描述
unit_num	String	npu卡数。
product_name	String	产品名。
memory	String	内存。

表 **7-113** memory

参数	参数类型	描述
size	Integer	内存大小。
unit	String	内存单元数。

表 7-114 disk

参数	参数类型	描述
size	String	磁盘大小。
unit	String	磁盘大小单位,一般为GB。

表 **7-115** volumes

参数	参数类型	描述
nfs	nfs object	nfs方式的挂载卷。

表 7-116 nfs

参数	参数类型	描述
nfs_server_pat h	String	nfs服务端路径。
local_path	String	挂载到训练容器中的路径。
read_only	Boolean	nfs挂载卷在容器中是否只读。

表 7-117 log_export_path

参数	参数类型	描述
obs_url	String	训练作业日志保存的OBS地址。
host_path	String	训练作业日志保存的宿主机的路径。

表 7-118 调用训练接口失败响应参数

参数	类型	描述
error_msg	String	调用失败时的错误信息,调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参见 错误码 ,调用成功时 无此字段。
error_solution	String	调用失败时的提示解决信息,调用成功时无此字段。

7.1.4 更新训练作业描述

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 方式一:根据指定的job_id更新。

from modelarts.session import Session from modelarts.estimatorV2 import Estimator session = Session() estimator = Estimator(session=session, job_id="your job id") estimator.update_job_configs(description="update job description")

● 方式二: 根据创建训练作业生成的训练作业对象更新。 job_instance.update_job_configs(description="update job description fourth")

参数说明

表 7-119 Estimator 请求参数说明

参数	是否必 选	参数类型	描述
session	是	Object	会话对象,初始化方法请参考 Session <mark>鉴权</mark> 。
job_id	是	String	训练作业的id,可通过 <mark>创建训练作业</mark> 生成的训练作业对象查询,如 "job_instance.job_id",或从 <mark>查询训</mark> 练作业列表 的响应中获得。

表 7-120 update_job_configs 请求参数说明

参数	是否必 选	参数类型	描述
description	是	String	需要更改的训练作业的描述信息。

无成功响应参数

表 7-121 调用训练接口失败响应参数

参数	类型	描述
error_msg	String	调用失败时的错误信息,调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参见 错误码 ,调用成功时 无此字段。
error_solution	String	调用失败时的提示解决信息,调用成功时无此字段。

7.1.5 删除训练作业

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 方式一:根据指定的job_id删除。

from modelarts.session import Session from modelarts.estimatorV2 import Estimator session = Session() Estimator.delete_job_by_id(session=session, job_id="your job id")

方式二:根据创建训练作业生成的训练作业对象删除。

job_instance.delete_job()

参数说明

表 7-122 delete_job_by_id 请求参数说明

参数	是否必 选	参数类型	描述
session	是	Object	会话对象,初始化方法请参考 Session <mark>鉴权</mark> 。
job_id	是	String	训练作业的id,可通过 创建训练作业 生成的训练作业对象查询,如 "job_instance.job_id",或从 查询训 练作业列表 的响应中获得。

无成功响应参数。

表 7-123 调用训练接口失败响应参数

参数	类型	描述
error_msg	String	调用失败时的错误信息,调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参见 错误码 ,调用成功时 无此字段。
error_solution	String	调用失败时的提示解决信息,调用成功时无此字段。

7.1.6 终止训练作业

终止训练作业,只可终止创建中、等待中、运行中的作业。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 方式一:根据指定的job_id终止。

from modelarts.session import Session from modelarts.estimatorV2 import Estimator session = Session() info = Estimator.control_job_by_id(session=session, job_id="your job id") print(info)

● 方式二:根据创建训练作业生成的训练作业对象终止。 job_instance.control_job()

参数说明

表 7-124 control_job_by_id 请求参数说明

参数	是否必 选	参数类型	描述
session	是	Object	会话对象,初始化方法请参考 Session <mark>鉴权</mark> 。
job_id	是	String	训练作业的id,可通过 创建训练作业 生成的训练作业对象查询,如 "job_instance.job_id",或从 <mark>查询训</mark> 练作业列表 的响应中获得。

表 7-125 返回参数说明

参数	参数类型	描述
kind	String	训练作业类型。默认使用job。
		枚举值:
		● job: 训练作业
		● hetero_job: 异构作业
		● autosearch_job: 自动搜索作业
		● mrs_job: mrs作业
		● edge_job: 边缘作业
metadata	JobMetadata object	训练作业元信息。
status	Status object	训练作业状态信息。创建作业无需填写。
algorithm	Response	训练作业算法。目前支持三种形式:
		• id只取算法的id;
	object	● subscription_id+item_version_id取算法的订 阅id和版本id;
		● code_dir+boot_file取训练作业的代码目录和 启动文件。
tasks	Array of TaskRespons e objects	异构训练作业的任务列表。
spec	spec object	训练作业规格参数。

表 7-126 JobMetadata

参数	参数类型	描述
id	String	训练作业ID,创建成功后由ModelArts生成返 回,无需填写。
name	String	训练作业名称。限制为1-64位只含数字、字母、 下划线和中划线的名称。
workspace_id	String	指定作业所处的工作空间,默认值为"0"。
description	String	对训练作业的描述,默认为"NULL",字符串的 长度限制为[0, 256]。
create_time	Long	训练作业创建时间戳,单位为毫秒,创建成功后 由ModelArts生成返回,无需填写。
user_name	String	训练作业创建用户的用户名,创建成功后由 ModelArts生成返回,无需填写。

参数	参数类型	描述
annotations	Map <string,st ring></string,st 	训练作业申明模板,异构作业job_template字段 默认为Template RL,其余默认为Template DL。

表 7-127 Status

参数	参数类型	描述
phase	String	训练作业一级状态,状态值稳定不变,可选值如 下: "Creating"、"Pending"、 "Running"、"Failed"、"Completed"、 "Terminating"、"Terminated"、 "Abnormal"。
secondary_ph ase	String	训练作业二级状态,状态值不稳定,可选值如 下: "Creating"、"Queuing"、 "Running"、"Failed"、"Completed"、 "Terminating"、"Terminated"、 "CreateFailed"、"TerminatedFailed"、 "Unknown"、"Lost"。
duration	Long	训练作业运行时长,单位为毫秒。
node_count_ metrics	Array <array<l nteger="">></array<l>	训练作业运行时节点数变化指标。
tasks	Array of strings	训练作业子任务名称。
start_time	String	训练作业开始时间,格式为时间戳。
task_statuses	Array of objects	训练在子任务状态信息。

表 7-128 task_statuses

参数	参数类型	描述
task	String	训练作业子任务名称。
exit_code	Integer	训练作业子任务退出码。
message	String	训练作业子任务错误消息。

表 **7-129** JobAlgorithmResponse

参数	参数类型	描述
id	String	算法管理的算法id。 枚举值: id: 只取算法的id; subscription_id+item_version_id: 取算法的 订阅id和版本id; code_dir+boot_file: 取训练作业的代码目录 和启动文件。
name	String	算法名称。
subscription_i d	String	订阅算法的订阅ID。应与item_version_id一同出现。
item_version_i d	String	订阅算法的版本。应与subscription_id一同出现。
code_dir	String	训练作业的代码目录。如:"/usr/app/"。应与 boot_file一同出现,若填入id或subscription_id +item_version_id则无需填写。
boot_file	String	训练作业的代码启动文件,需要在代码目录下。 如: "/usr/app/boot.py"。应与code_dir一同出 现,若填入入id或subscription_id +item_version_id则无需填写。
autosearch_co nfig_path	String	自动化搜索作业的yaml配置路径,需要提供一个 OBS路径。
autosearch_fr amework_pat h	String	自动化搜索作业的框架代码目录,需要提供一个 OBS路径。
command	String	自定义镜像训练作业的自定义镜像的容器的启动 命令。可填code_dir。
parameters	Array of Parameter objects	训练作业的运行参数。
policies	policies object	作业支持的策略。
inputs	Array of Input objects	训练作业的数据输入。
outputs	Array of Output objects	训练作业的结果输出。
engine	engine object	训练作业的引擎。使用算法管理的算法id或订阅 算法subscription_id+item_version_id创建作业 时,无需填写。

参数	参数类型	描述
environments	Map <string,st ring></string,st 	训练作业的环境变量。格式:"key":"value",无 需填写。

表 7-130 Parameter

参数	参数类型	描述
name	String	参数名称。
value	String	参数值。
description	String	参数描述信息。
constraint	constraint object	参数属性。
i18n_descripti on	i18n_descript ion object	国际化描述。

表 7-131 constraint

参数	参数类型	描述
type	String	参数种类。
editable	Boolean	是否可编辑。
required	Boolean	是否必须。
sensitive	Boolean	是否敏感。
valid_type	String	有效种类。
valid_range	Array of strings	有效范围。

表 7-132 i18n_description

参数	参数类型	描述
language	String	国际语种。
description	String	对应描述。

表 **7-133** policies

参数	参数类型	描述
auto_search	auto_search object	超参搜索配置。

表 7-134 auto_search

参数	参数类型	描述
skip_search_p arams	String	需要排除的超参组合。
reward_attrs	Array of objects	搜索指标列表。
search_param s	Array of objects	搜索参数。
algo_configs	Array of objects	搜索算法配置。

表 7-135 reward_attrs

参数	参数类型	描述
name	String	指标名称。
mode	String	搜索方向。 • max: 指定时,表示指标值越大越好; • min: 指定时,表示指标值越小越好。
regex	String	指标正则表达式。

表 7-136 search_params

参数	参数类型	描述
name	String	超参名称。
param_type	String	参数类型。
		• continuous: 指定时,表示参数类型为连续 值;
		● discreate: 指定时,表示参数类型为离散值。
lower_bound	String	超参下界。
upper_bound	String	超参上界。

参数	参数类型	描述
discrete_point s_num	String	连续型超参离散化取值个数。
discrete_value s	Array of strings	离散型超参的取值列表。

表 7-137 algo_configs

参数	参数类型	描述
name	String	搜索算法名称。
params	Array of AutoSearchA lgoConfigPar ameter objects	搜索算法参数。

表 7-138 AutoSearchAlgoConfigParameter

参数	参数类型	描述
key	String	参数键。
value	String	参数值。
type	String	参数种类。

表 7-139 Input

参数	参数类型	描述
name	String	数据输入通道名称。
description	String	数据输入通道描述信息。
local_dir	String	数据输入通道映射的容器本地路径。
remote	InputDataInf o object	数据实际输入信息。
remote_constr aint	Array of objects	数据输入约束。

表 7-140 InputDataInfo

参数	参数类型	描述
dataset	dataset object	数据输入信息为数据集。
obs	obs object	数据输入输出信息为OBS方式。

表 7-141 dataset

参数	参数类型	描述
id	String	训练作业的数据集ID。
version_id	String	训练作业的数据集版本ID。
obs_url	String	训练作业需要的数据集OBS路径URL,modelarts 会通过数据集ID和数据集版本ID自动解析生成。 如:"/usr/data/"。

表 7-142 obs

参数	参数类型	描述
obs_url	String	训练作业需要的数据集OBS路径URL。如: "/usr/data/"。

表 7-143 remote_constraint

参数	参数类型	描述
data_type	String	数据输入类型,包括数据存储位置、数据集两种方式。
attributes	String	数据输入为数据集时的相关属性。 枚举值:
		• data_format: 数据格式。
		• data_segmentation:数据切分方式。
		● dataset_type: 标注类型。

表 7-144 Output

参数	参数类型	描述
name	String	数据输出通道名称。

参数	参数类型	描述
description	String	数据输出通道描述信息。
local_dir	String	数据输出通道映射的容器本地路径。
remote	remote object	数据实际输出信息。

表 7-145 remote

参数	参数类型	描述
obs	obs object	数据实际输出到OBS。

表 7-146 obs

参数	参数类型	描述
obs_url	String	数据实际输出到OBS的路径。

表 **7-147** engine

参数	参数类型	描述
engine_id	String	训练作业选择的引擎规格ID。engine_id、 engine_name + engine_version和image_url方式 三选一。
engine_name	String	训练作业选择的引擎名称。若填入engine_id,则 无需填写。
engine_versio n	String	训练作业选择的引擎版本名称。若填入 engine_id,则无需填写。
image_url	String	训练作业选择的自定义镜像地址。

表 7-148 TaskResponse

参数	参数类型	描述
role	String	异构训练作业的任务角色。 枚举值:
		● learner(支持GPU\CPU规格) ● worker(支持CPU规格)
algorithm	algorithm object	算法管理算法配置。

参数	参数类型	描述
task_resource	FlavorRespon se object	训练作业、算法的规格信息。

表 7-149 algorithm

参数	参数类型	描述
code_dir	String	算法启动文件所在目录绝对路径。
boot_file	String	算法启动文件绝对路径。
inputs	inputs object	算法输入通道信息。
outputs	outputs object	算法输出通道信息。
engine	engine object	异构作业所依赖的引擎。

表 7-150 inputs

参数	参数类型	描述
name	String	数据输入通道名称。
local_dir	String	数据输入输出通道映射的容器本地路径。
remote	remote object	数据实际输入信息,异构作业只支持OBS。

表 7-151 remote

参数	参数类型	描述
obs	obs object	数据输入输出信息为OBS方式。

表 7-152 obs

参数	参数类型	描述
obs_url	String	训练作业需要的数据集OBS路径URL。如: "/usr/data/"。

表 7-153 outputs

参数	参数类型	描述
name	String	数据输出通道名称。
local_dir	String	数据输出通道映射的容器本地路径。
remote	remote object	数据实际输出信息。
mode	String	数据传输模式,默认为 "upload_periodically"。
period	String	数据传输周期,默认为30s。

表 **7-154** remote

参数	参数类型	描述
obs	obs object	数据实际输出到OBS。

表 7-155 obs

参数	参数类型	描述
obs_url	String	数据实际输出到OBS的路径

表 **7-156** engine

参数	参数类型	描述
engine_id	String	异构作业引擎规格的ID。如"caffe-1.0.0-python2.7"。
engine_name	String	异构作业引擎规格的名称。如"Caffe"。
engine_versio n	String	异构作业引擎规格的版本。
v1_compatibl e	Boolean	是否为v1兼容模式。
run_user	String	引擎默认启动用户uid。

表 7-157 FlavorResponse

参数	参数类型	描述
flavor_id	String	资源规格的ID。

参数	参数类型	描述
flavor_name	String	资源规格的名称。
max_num	Integer	资源规格的最大节点数。
flavor_type	String	资源规格的类型。可选值如下:
		• CPU;
		• GPU;
		• Ascend。
billing	billing object	资源规格计费信息。
flavor_info	flavor_info object	资源规格详细信息。
attributes	Map <string,st ring></string,st 	其他规格属性。

表 **7-158** billing

参数	参数类型	描述
code	String	计费码。
unit_num	Integer	计费卡数。

表 7-159 flavor_info

参数	参数类型	描述
max_num	Integer	可以选择的最大节点数量(max_num,为1代表 不支持分布式)。
сри	cpu object	cpu规格信息。
gpu	gpu object	gpu规格信息。
npu	npu object	Ascend规格信息。
memory	memory object	内存信息。

表 7-160 cpu

参数	参数类型	描述
arch	String	cpu架构。
core_num	Integer	核数。

表 7-161 gpu

参数	参数类型	描述
unit_num	Integer	gpu卡数。
product_num e	String	产品名。
memory	String	内存。

表 7-162 npu

参数	参数类型	描述
unit_num	String	npu卡数。
product_name	String	产品名。
memory	String	内存。

表 **7-163** memory

参数	参数类型	描述
size	Integer	内存大小。
unit	String	内存单元数。

表 7-164 spec

参数	参数类型	描述
resource	Resource object	训练作业资源规格信息。flavor_id和pool_id+ [flavor_id]方式二选一。
volumes	Array of objects	训练作业挂载卷信息。
log_export_pa th	log_export_p ath object	训练作业日志输出信息。

表 7-165 Resource

参数	参数类型	描述
policy	String	训练作业资源规格模式,可选值如下: "regular"、"economic"、"turbo"。

参数	参数类型	描述
flavor_id	String	训练作业选择的资源规格ID。
flavor_name	String	使用flavor_id时,由ModelArts返回的只读规格名称。
node_count	Integer	训练作业选择的资源副本数。 最小值: 1
pool_id	String	训练作业选择的资源池ID。
flavor_detail	flavor_detail object	训练作业、算法的规格信息。

表 7-166 flavor_detail

参数	参数类型	描述
flavor_type	String	资源规格的类型。可选值如下:
		• CPU;
		• GPU;
		Ascend。
billing	billing object	资源规格计费信息。
flavor_info	flavor_info object	资源规格详细信息。

表 **7-167** billing

参数	参数类型	描述
code	String	计费码。
unit_num	Integer	计费卡数。

表 7-168 flavor_info

参数	参数类型	描述	
max_num	Integer	可以选择的最大节点数量(max_num,为1代表 不支持分布式)。	
cpu	cpu object	cpu规格信息。	
gpu	gpu object	gpu规格信息。	
npu	npu object	Ascend规格信息。	

参数	参数类型	描述
memory	memory object	内存信息。
disk	disk object	磁盘信息。

表 7-169 cpu

参数	参数类型	描述
arch	String	cpu架构。
core_num	Integer	核数。

表 7-170 gpu

参数	参数类型	描述
unit_num	Integer	gpu卡数。
product_num e	String	产品名。
memory	String	内存。

表 7-171 npu

参数	参数类型	描述
unit_num	String	npu卡数。
product_name	String	产品名。
memory	String	内存。

表 7-172 memory

参数	参数类型	描述
size	Integer	内存大小。
unit	String	内存单元数。

表 7-173 disk

参数	参数类型	描述	
size	String	磁盘大小。	
unit	String	磁盘大小单位,一般为GB。	

表 7-174 volumes

参数	参数类型	描述	
nfs	nfs object	nfs方式的挂载卷。	

表 7-175 nfs

参数	参数类型	描述
nfs_server_pat h	String	nfs服务端路径。
local_path	String	挂载到训练容器中的路径。
read_only	Boolean	nfs挂载卷在容器中是否只读。

表 7-176 log_export_path

参数	参数类型	描述	
obs_url	String	训练作业日志保存的OBS地址。	
host_path	String	训练作业日志保存的宿主机的路径。	

表 7-177 调用训练接口失败响应参数

参数	类型	描述	
error_msg	String	调用失败时的错误信息,调用成功时无此字段。	
error_code	String	调用失败时的错误码,具体请参见 错误码 ,调用成功时 无此字段。	
error_solution	String	调用失败时的提示解决信息,调用成功时无此字段。	

7.1.7 查询训练日志

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

方式一: 根据指定的job_id查询。
from modelarts.session import Session
from modelarts.estimatorV2 import Estimator
session = Session()
estimator = Estimator(session=session, job_id="your job id")
info = estimator.get_job_log()
print(info)

● 方式二:根据<mark>创建训练作业</mark>生成的训练作业对象查询。 log = job_instance.get_job_log(task_id="worker-0") print(log)

参数说明

表 7-178 Estimator 初始化参数说明

参数	是否必 选	参数类型	描述
session	是	Object	会话对象,初始化方法请参考 Session <mark>鉴权</mark> 。
job_id	是	String	训练作业的id,可通过 创建训练作业 生成的训练作业对象查询,如 "job_instance.job_id",或从 <mark>查询训</mark> 练作业列表 的响应中获得。

表 7-179 get_job_log 请求参数说明

参数	是否必 选	参数类型	描述
task_id	否	String	要查看哪个工作节点的日志,默认值 为"worker-0";如果在 <mark>创建训练作业</mark> 时参数train_instance_count选择了 2,则可选值为 "worker-0","worker-1",依次类推。

表 7-180 返回参数

参数	参数类型	描述
content	String	日志内容。
		● 如果日志大小没有超过上限(n兆),则返回 全部内容;
		● 如果日志超过了上限(n兆),则返回最新的n 兆的日志。
current_size	Integer	当前返回的日志大小(单位:字节)。最大为5 兆。
full_size	Integer	完整的日志大小(单位:字节)。

表 7-181 调用训练接口失败响应参数

参数	类型	描述
error_msg	String	调用失败时的错误信息,调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参见 错误码 ,调用成功时 无此字段。
error_solution	String	调用失败时的提示解决信息,调用成功时无此字段。

7.1.8 查询训练作业的运行指标

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 方式一:根据指定的job_id查询。

from modelarts.session import Session from modelarts.estimatorV2 import Estimator session = Session() estimator = Estimator(session=session, job_id="your job id") info = estimator.get_job_metrics() print(info)

• 方式二:根据创建训练作业生成的训练作业对象查询。

info = job_instance.get_job_metrics(task_id="worker-0")
print(info)

参数说明

表 7-182 Estimator 初始化参数说明

参数	是否必选	参数类型	描述
session	是	Object	会话对象,初始化方法请参考 Session <mark>鉴权</mark> 。
job_id	是	String	训练作业的id,可通过 <mark>创建训练作业</mark> 生成的训练作业对象查询,如 "job_instance.job_id",或从 <mark>查询训</mark> <mark>练作业列表</mark> 的响应中获得。

表 7-183 get_job_log 请求参数说明

参数	是否必 选	参数类型	描述
task_id	否	String	要查看哪个工作节点的日志,默认值 为"worker-0";如果在 <mark>创建训练作业</mark> 时参数train_instance_count选择了 2,则可选值为 "worker-0","worker-1",依次类推。

表 7-184 返回参数说明

参数	参数类型	描述
metrics	Array of objects	运行指标。

表 7-185 metrics

参数	参数类型	描述
metric	String	运行指标,可选值如下: cpuUsage(CPU使用率)、memUsage(物理内存使用率)、gpuUtil(GPU使用率)、gpuMemUsage(显存使用率)、npuUtil(NPU使用率)、npuMemUsage(NPU显存使用率)。
value	Array of numbers	运行指标对应数值,1min统计一个平均值。

表 7-186 调用训练接口失败响应参数

参数	类型	描述
error_msg	String	调用失败时的错误信息,调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参见 <mark>错误码</mark> ,调用成功时 无此字段。
error_solution	String	调用失败时的提示解决信息,调用成功时无此字段。

7.2 资源和引擎规格接口

7.2.1 查询资源规格列表

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

from modelarts.session import Session from modelarts.estimatorV2 import Estimator session = Session() info = Estimator.get_train_instance_types(session=session) print(info)

参数说明

表 7-187 get_train_instance_types 参数说明

参数	是否必选	参数类型	描述
session	是	Object	会话对象,初始化方法请参考 Session <mark>鉴权</mark> 。

表 7-188 成功响应参数说明

参数类型	描述
List	资源规格参数列表。

表 7-189 调用训练接口失败响应参数

参数	类型	描述	
error_msg	String	调用失败时的错误信息,调用成功时无此字段。	

参数	类型	描述
error_code	String	调用失败时的错误码,具体请参见 <mark>错误码</mark> ,调用成功时 无此字段。
error_solution	String	调用失败时的提示解决信息,调用成功时无此字段。

7.2.2 查询引擎规格列表

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

from modelarts.session import Session from modelarts.estimatorV2 import Estimator session = Session() info = Estimator.get_framework_list(session=session) print(info)

参数说明

表 7-190 get_train_instance_types 参数说明

参数	是否必 选	参数类型	描述
session	是	Object	会话对象,初始化方法请参考 Session <mark>鉴权</mark> 。

表 7-191 get_framework_list 成功响应参数说明

参数类型	描述
List	引擎规格参数列表,请参见 <mark>表3</mark> 。

表 7-192 framework_list 参数说明

参数	参数类型	描述
framework_type	String	引擎类型。
framework_versio	String	引擎版本。

表 7-193 调用训练接口失败响应参数

参数	类型	描述
error_msg	String	调用失败时的错误信息,调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参见 <mark>错误码</mark> ,调用成功时 无此字段。
error_solution	String	调用失败时的提示解决信息,调用成功时无此字段。

る 训练管理(旧版)

8.1 训练作业

8.1.1 创建训练作业

在训练平台训练方式中,如果训练失败的话,可以到训练平台或调用查询训练作业日 志查看详细的日志信息。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session 鉴权请参见Session鉴权。

示例一:使用OBS存储位置创建训练作业

```
from modelarts.session import Session
from modelarts.estimator import Estimator
session = Session()
estimator = Estimator(
              modelarts_session=session,
              framework_type='PyTorch',
                                                                # AI引擎名称
              framework_version='PyTorch-1.0.0-python3.6',
                                                                     # AI引擎版本
              code_dir='/bucket/src/',
                                                             # 训练脚本目录
              boot_file='/bucket/src/pytorch_sentiment.py',
                                                                    # 训练启动脚本目录
              log_url='/bucket/log/',
                                                             # 训练日志目录
              hyperparameters=[
                         {"label":"classes",
"value": "10"},
{"label":"lr",
                          "value": "0.001"}
              output_path='/bucket/output/',
                                                                # 训练输出目录
              train_instance_type='modelarts.vm.cpu.2u',
                                                                   # 训练环境规格
              train_instance_count=1,
                                                              # 训练节点个数
              job_description='pytorch-sentiment with ModelArts SDK')
                                                                         # 训练作业描述
job_instance = estimator.fit(inputs='/bucket/data/train/', wait=False, job_name='my_training_job')
```

```
示例二: 使用数据集创建训练作业
from modelarts.session import Session
from modelarts.estimator import Estimator
session = Session()
estimator = Estimator(
            modelarts_session=session,
            framework_type='PyTorch',
                                                         # AI引擎名称
```

```
framework_version='PyTorch-1.0.0-python3.6',
                                                                     # AI引擎版本
              code_dir='/bucket/src/',
                                                             # 训练脚本目录
              boot_file='/bucket/src/pytorch_sentiment.py',
                                                                   # 训练启动脚本目录
              log_url='/bucket/log/',
                                                            # 训练日志目录
              hyperparameters=[
                         {"label":"classes",
                         "value": "10"},
{"label":"lr",
"value": "0.001"}
              output path='/bucket/output/',
                                                                # 训练输出目录
              train_instance_type='modelarts.vm.cpu.2u',
                                                                  # 训练环境规格
              train_instance_count=1,
                                                              # 训练节点个数
             job_description='pytorch-sentiment with ModelArts SDK')
                                                                         # 训练作业描述
job_instance = estimator.fit(dataset_id='4AZNvFkN7KYr5EdhFkH',
dataset_version_id='UOF9BIeSGArwVt0oI6T', wait=False, job_name='my_training_job')
```

● 示例三: 创建自定义镜像的训练作业

```
from modelarts.session import Session
from modelarts.estimator import Estimator
session = Session()
estimator = Estimator(
              modelarts_session=session,
              log_url='/bucket/log/',
                                                            # 训练日志目录
              hyperparameters=[
                         {"label":"classes",
                         "value": "10"},
{"label":"lr",
"value": "0.001"}
              output_path='/bucket/output/',
                                                                # 训练输出目录
              train_instance_type='modelarts.vm.cpu.2u',
                                                                  # 训练环境规格
              train_instance_count=1,
                                                              # 训练节点个数
              user_command='bash -x /home/work/run_train.sh python /home/work/user-job-
dir/app/mnist/mnist_softmax.py --data_url /home/work/user-job-dir/app/
mnist data',
                                                  # 自定义镜像启动命令
              user_image_url='100.125.5.235:20202/jobmng/cpu-base:1.0',
                                                                         # 自定义镜像下载地址
              job_description='pytorch-sentiment with ModelArts SDK')
                                                                         # 训练作业描述
job_instance = estimator.fit(inputs='/bucket/data/train/', wait=False, job_name='my_training_job')
```

• 示例四: 向专属资源池提交训练作业

```
from modelarts.session import Session
from modelarts.estimator import Estimator
session = Session()
estimator = Estimator(
              modelarts_session=session,
              framework_type='PyTorch',
                                                               # AI引擎名称
              framework_version='PyTorch-1.0.0-python3.6',
                                                                     # AI引擎版本
              code_dir='/bucket/src/',
boot_file='/bucket/src/pytorch_sentiment.py',
                                                            # 训练脚本目录
                                                                   # 训练启动脚本目录
              log_url='/bucket/log/',
                                                            # 训练日志目录
              hyperparameters=[
                         {"label":"classes",
"value": "10"},
                         {"label":"lr"
                          "value": "0.001"}
              output_path='/bucket/output/',
                                                                # 训练输出目录
        pool_id="your pool id",
                                                        # 专属资源池id
              train_instance_type='your instance type',
                                                                  #训练环境规格,如果为None,
默认使用专属池默认规格
                                                             # 训练节点个数
              train_instance_count=1,
              job description='pytorch-sentiment with ModelArts SDK')
                                                                        # 训练作业描述
job_instance = estimator.fit(inputs='/bucket/data/train/', wait=False, job_name='my_training_job')
```

表 8-1 Estimator 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法请参见 <mark>Session鉴</mark> 权。
train_instance_c ount	是	Int	训练作业计算节点个数。
code_dir	否	String	训练作业的代码目录,如"/bucket/ src/"。当填入model_name时不需要填 写。
boot_file	否	String	训练作业的代码启动文件,需要在代码目录下,如"/bucket/src/boot.py"。当填入model_name时不需要填写。
model_name	否	String	训练作业的内置算法模型名称。填入 model_name后app_url与boot_file_url不需 填写,framework_type和 framework_version也不需要填写。 "model_name"请从 <mark>查询预置算法</mark> 接口中 获取。
output_path	是	String 训练作业的输出位置。	
hyperparameter s	否	JSON Array	训练作业的运行参数,为label-value格式, 其中lable和value的值均为String类型;当为 自定义镜像训练作业的时候,此参数为容器 环境变量。
log_url	否	String	训练作业的日志OBS输出路径URL,默认为空。如:"/usr/log/"。
train_instance_t ype	是	String	训练作业选择的资源规格。若选择在训练平 台训练,请从 <mark>查询资源规格列表</mark> 接口获取。
framework_type	否	String	训练作业选择的引擎规格。请从 <mark>查询引擎规</mark> 格列表接口获取引擎规格。当填入 model_name时不需要填写。
framework_versi on	否	String	训练作业选择的引擎版本。请从 <mark>查询引擎规</mark> 格列表接口获取引擎版本。当填入 model_name时不需要填写。
job_description	否	String	训练作业的描述。
user_image_url	否	String	自定义镜像训练作业的自定义镜像的SWR- URL。如: "100.125.5.235:20202/jobmng/ custom-cpu-base:1.0"。

参数	是否 必选	参数类型	描述
user_command	否	String	自定义镜像训练作业的自定义镜像的容器的 启动命令。形式为: "bash /home/work/ run_train.sh python /home/work/user-job- dir/app/train.py {python_file_parameter}"。
pool_id	否	String	训练作业选择的资源池ID,可在ModelArts 管理控制台,单击左侧"专属资源池",在 专属资源池列表中查看资源池ID。

表 8-2 fit 请求参数说明

参数	是否必选	参数类型	描述
inputs	是	String	训练作业的数据存储位置。 inputs和(dataset_id、 dataset_version_id)、data_source三者不可 同时出现,但必须有其一。 本地训练只支持该参数。
dataset_id	否	String	训练作业的数据集ID。获取数据集ID,请参见 <mark>查看数据集的基本信息</mark> 。 应与dataset_version_id同时出现,但不可与 inputs同时出现。
dataset_versio n_id	否	String	训练作业的数据集版本ID。获取数据集版本ID,请参见 查看数据集的基本信息 。 应与dataset_id同时出现,但不可与inputs同时出现。
wait	否	Boolea n	是否等待训练作业结束,默认为False。
job_name	否	String	训练作业的名称,支持[a-zA-Z0-9]{1,64}。 若不填,则会动态生成一个job_name。

表 8-3 训练平台训练的成功响应说明

参数	类型	描述
TrainingJob	Object	训练对象。该对象包含job_id、 version_id等属性,对训练作业的查询、 更新、删除等操作,如可通过 job_instance.job_id获取训练作业ID。

8.1.2 训练作业调试

创建在线训练作业前,可先通过创建本地训练作业来进行调试,可以参考案例 "tensorflow_mlp_mnist_local_mode" (单击进入ModelArts控制台)。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 步骤1:创建本地训练作业。当"train_instance_type"为"local"时,表示创建的是本地训练作业,可以用来调试代码和参数。

```
from modelarts.session import Session
from modelarts.estimator import Estimator
from modelarts.environment import Environment
from modelarts.environment.conda_env import CondaDependencies
session = Session()
env = Environment("tensorflow_mlp_mnist")
cd = CondaDependencies.create(pip_packages=["tensorflow==1.13.1", "requests"],
conda_packages=["python=3.6.2"])
env.conda = cd
src_local_path = "/home/ma-user/work/tensorflow_mlp_mnist_local_mode/train/"
train_file = "tensorflow_mlp_mnist.py"
estimator = Estimator(modelarts_session=session,
             code_dir=src_local_path,
                                          # 本地训练脚本目录
             boot_file=train_file,
                                        # 本地训练启动脚本目录
             train_instance_type='local',
                                         # 指定为本地训练
             train_instance_count=1,
                                          # 训练节点个数
             environment=env)
                                          # 训练脚本运行的环境
job_instance = estimator.fit(wait=False, job_name='my_training_job')
```

步骤2:本地训练作业完成后,可以创建为在线训练作业。当 "train_instance_type"为训练环境规格时,表示创建的是在线训练。

```
from modelarts.session import Session
from modelarts.estimator import Estimator
from modelarts.environment import Environment
from modelarts.environment.conda_env import CondaDependencies
session = Session()
env = Environment("tensorflow_mlp_mnist")
cd = CondaDependencies.create(pip_packages=["tensorflow==1.13.1", "requests"],
conda_packages=["python=3.6.2"])
env.conda = cd
src_local_path = "/home/ma-user/work/tensorflow_mlp_mnist_local_mode/train/"
train_file = "tensorflow_mlp_mnist.py"
estimator = Estimator(modelarts_session=session,
             code_dir=src_local_path,
                                                    # 训练脚本目录
                                                 # 训练启动脚本目录
             boot_file=train_file,
             train_instance_type='modelarts.vm.cpu.2u',
                                                          # 在线训练
                                                    # 训练节点个数
             train_instance_count=1,
                                                    # 训练脚本运行的环境
             environment=env)
job_instance = estimator.fit(wait=False, job_name='my_training_job')
```

表 8-4 Environment 参数说明

参数	是否必 选	类型	说明
name	是	String	环境名称。

参数	是否必 选	类型	说明
conda	否	CondaDepend encies	conda环境,具体请参见 表8-5 。

表 8-5 CondaDependencies 参数说明

参数	是否必 选	类型	说明
channels	否	List	python包的下载源。
pip_packages	否	List	conda虚拟环境需要使用的 python包,如tensorflow, pillow等。
conda_packages	否	List	conda虚拟环境需要使用的conda 包,如指定python版本。

表 8-6 Estimator 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法请参见 <mark>Session鉴</mark> <mark>权</mark> 。
train_instance_c ount	是	Int	训练作业计算节点个数。
code_dir	否	String	训练作业的代码目录,如"/bucket/ src/"。当填入model_name时不需要填 写。
boot_file	否	String	训练作业的代码启动文件,需要在代码目录 下,如"/bucket/src/boot.py"。当填入 model_name时不需要填写。
model_name	否	String	训练作业的内置算法模型名称。填入 model_name后app_url与boot_file_url不需 填写,framework_type和 framework_version也不需要填写。 "model_name"请从 <mark>查询预置算法</mark> 接口中 获取。
output_path	是	String	训练作业的输出位置。
hyperparameter s	否	JSON Array	训练作业的运行参数,为label-value格式, 其中lable和value的值均为String类型;当为 自定义镜像训练作业的时候,此参数为容器 环境变量。

参数	是否 必选	参数类型	描述
log_url	否	String	训练作业的日志OBS输出路径URL,默认为 空。如:"/usr/log/"。
train_instance_t ype	是	String	训练作业选择的资源规格。若选择在训练平 台训练,请从 <mark>查询资源规格列表</mark> 接口获取。
framework_type	否	String	训练作业选择的引擎规格。请从 <mark>查询引擎规</mark> 格列表接口获取引擎规格。当填入 model_name时不需要填写。
framework_versi on	否	String	训练作业选择的引擎版本。请从 <mark>查询引擎规格列表</mark> 接口获取引擎版本。当填入 model_name时不需要填写。
job_description	否	String	训练作业的描述。
user_image_url	否	String	自定义镜像训练作业的自定义镜像的SWR- URL。如: "100.125.5.235:20202/jobmng/ custom-cpu-base:1.0"。
user_command	否	String	自定义镜像训练作业的自定义镜像的容器的启动命令。形式为: "bash /home/work/run_train.sh python /home/work/user-job-dir/app/train.py {python_file_parameter}"。
pool_id	否	String	训练作业选择的资源池ID,可在ModelArts 管理控制台,单击左侧"专属资源池",在 专属资源池列表中查看资源池ID。

表 8-7 fit 请求参数说明

参数	是否必 选	参数类型	描述
inputs	是	String	训练作业的数据存储位置。 inputs和(dataset_id、 dataset_version_id)、data_source三者不可 同时出现,但必须有其一。 本地训练只支持该参数。
dataset_id	否	String	训练作业的数据集ID。获取数据集ID,请参见 <mark>查看数据集的基本信息</mark> 。 应与dataset_version_id同时出现,但不可与 inputs同时出现。

参数	是否必 选	参数类型	描述
dataset_versio n_id	否	String	训练作业的数据集版本ID。获取数据集版本ID,请参见 <mark>查看数据集的基本信息</mark> 。 应与dataset_id同时出现,但不可与inputs同时出现。
wait	否	Boolea n	是否等待训练作业结束,默认为False。
job_name	否	String	训练作业的名称,支持[a-zA-Z0-9]{1,64}。 若不填,则会动态生成一个job_name。

8.1.3 查询训练作业列表

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

from modelarts.session import Session from modelarts.estimator import Estimator session = Session() job_list_info = Estimator.get_job_list(modelarts_session=session, status=8, per_page=10, page=1, order="asc", search_content="job") print(job_list_info)

表 8-8 get_job_list 请求参数说明

参数	是否必选	参数类 型	描述
modelarts_se ssion	是	Object	会话对象,初始化方法请参见 <mark>Session鉴</mark> 权。
status	否	Integer	作业状态的查询,默认为所有状态。例如查看创建失败的作业,可选值为3、5、6、13。详细作业状态列表请参见作业状态参考。
per_page	否	Integer	指定每一页展示作业的总量,默认为 "10",可选的范围为[1,1000]。
page	否	Integer	指定要查询页的索引,默认为"1"。

参数	是否必选	参数类 型	描述
sortBy/ sort_by	否	String	当使用AK/SK认证方式时,参数名为 "sortBy";当使用帐户认证方式时,参数 名为"sort_by"。指定查询的排序方式, 默认是作业名称"job_name",目前支持 的排序还有作业描述"job_desc",作业状 态"status",运行时长"duration",引 擎类型"engine_type"以及创建时间 "create_time"。
order	否	String	可选值有: • "asc"为递增排序。 • "desc"为递减排序,默认为"desc"。
search_conte nt	否	String	指定要查询的文字信息,例如训练作业名 字,默认为空,字符串的长度为[0,100]。

表 8-9 get_job_list 返回参数说明

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参见 <mark>错</mark> <mark>误码</mark> 。 调用成功时无此字段。
job_total_count	Integer	查询到的用户创建作业总数。
job_count_limit	Integer	用户还可以创建训练作业的数量。
is_success	Boolean	接口调用是否成功。
quotas	Integer	训练作业的最大运行数量。
jobs	JSON Array	训练作业的属性列表,具体请参见 <mark>表</mark> 8-10 。

表 8-10 jobs 属性列表

参数	参数类型	描述
job_id	Long	训练作业的ID。
job_name	String	训练作业的名称。
version_id	Long	训练作业的版本ID。

参数	参数类型	描述
status	Byte	训练作业的运行状态,详细作业状态 列表请参见 <mark>作业状态参考</mark> 。
create_time	Long	训练作业的创建时间,时间戳格式。
duration	Long	训练作业的运行时长,单位为毫秒。
job_desc	String	训练作业的具体描述。
version_count	Long	训练作业的版本数。

8.1.4 查询训练作业详情

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• 方式一:根据指定的job_id和version_id查询

from modelarts.session import Session from modelarts.estimator import Estimator session = Session() estimator = Estimator(modelarts_session=session, job_id="182626", version_id="278813") job_info = estimator.get_job_info() print(job_info)

- 方式二:根据创建训练作业生成的训练作业对象查询 job_info = job_instance.get_job_info() print(job_info)
- 方式三:根据<mark>查询训练作业版本列表</mark>返回的指定训练作业版本对象查询 job_info = job_version_instance_list[0].get_job_info() print(job_info)

表 8-11 Estimator 请求参数说明

参数	是否必 选	参数类 型	描述
modelarts_sess ion	是	Object	会话对象,初始化方法请参见 Session<mark>鉴权</mark>。
job_id	是	String	训练作业的ID。job_id可通过 创建训练作业 生成的训练作业对象查询,如 "job_instance.job_id"。或可通过 查询训练 作业列表 的响应中获取。
version_id	是	String	训练作业的版本ID。version_id可通过 <mark>创建训练作业</mark> 生成的训练作业对象查询,如 "job_instance.version_id"。或可通过 <mark>查询</mark> 训练作业列表 的响应获取。

表 8-12 get_job_info 返回参数说明

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。
		调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参见 <mark>错误码</mark> 。
		调用成功时无此字段。
is_success	Boolean	接口调用是否成功。
job_id	Long	训练作业的ID。
job_name	String	训练作业的名称。
job_desc	String	训练作业的描述信息。
version_id	Long	训练作业的版本ID。
version_name	String	训练作业的版本名称。
pre_version_id	Long	训练作业前一版本的名称。
engine_type	Short	训练作业的引擎类型。 "engine_type"和 "engine_name"对应关系如下:
		engine_type: 1, engine_name: "TensorFlow"
		engine_type: 2, engine_name: "MXNet"
		• engine_type: 3, engine_name: "Ray"
		• engine_type: 4, engine_name:"Caffe"
		engine_type: 5, engine_name: "Spark_MLlib"
		engine_type: 9, engine_name: "XGBoost-Sklearn"
		• engine_type: 10, engine_name: "PyTorch"
		• engine_type: 12, engine_name: "Horovod"

参数	参数类型	描述
engine_name	String	训练作业的引擎名称。目前支持的引擎名称 如下:
		Caffe
		Horovod
		MXNet
		PyTorch
		• Ray
		Spark_MLlibTensorFlow
		XGBoost-Sklearn
::-	1	
engine_id	Long	训练作业的引擎ID。
engine_version	String	训练作业使用的引擎版本。
status	Integer	训练作业的状态,详细作业状态列表请参见 作业状态参考。
app_url	String	训练作业的代码目录。
boot_file_url	String	训练作业的代码启动文件。
create_time	Long	训练作业的创建时间。
parameter	JSON Array	训练作业的运行参数,为label-value格式; 当为自定义镜像训练作业的时候,此参数为 容器环境变量。
duration	Long	训练作业的运行时间,单位为毫秒。
spec_id	Long	训练作业资源规格ID。
core	String	资源规格的核数。
сри	String	资源规格CPU内存。
gpu_num	Integer	资源规格GPU的个数。
gpu_type	String	资源规格GPU的类型。
worker_server_n um	Integer	训练作业worker的个数。
data_url	String	训练作业的数据集。
train_url	String	训练作业输出文件OBS路径。
dataset_version_ id	String	训练作业的数据集版本ID。
dataset_id	String	训练作业的数据集ID。
data_source	JSON Array	训练作业使用的多数据集。

参数	参数类型	描述
model_id	Long	训练作业的模型ID。
model_metric_li st	JSON Array	训练作业的模型评测参数。
system_metric_li st	JSON Array	训练作业的系统监控指标。
user_image_url	String	自定义镜像训练作业的自定义镜像的SWR-URL。
user_command	String	自定义镜像训练作业的自定义镜像的容器的启动命令。

表 8-13 data_source 属性列表

参数	参数类型	描述
dataset_id	String	训练作业的数据集ID。
dataset_version	String	训练作业的数据集版本ID。
type	String	数据集类型。 "obs":表示使用OBS的数据。 "dataset":表示使用数据集的数据。
data_url	String	OBS的桶路径。

表 8-14 model_metric_list 属性列表

参数	参数类型	描述
metric	JSON Array	训练作业的模型单个分类测评参数。
total_metric	JSON Array	训练作业的模型总测评参数。

表 8-15 system_metric_list 属性列表

参数	参数类型	描述
cpuUsage	JSON Array	训练作业CPU资源占用率。
memUsage	JSON Array	训练作业内存资源占用率。
gpuUtil	JSON Array	训练作业GPU资源占用率。

表 8-16 metric 属性列表

参数	参数类型	描述
metric_values	JSON Array	训练作业模型单个分类测评参数指标。
reserved_data	JSON Array	预留字段。
metric_meta	JSON Array	训练作业模型单个分类,包含类ID和类 名。

表 8-17 metric_values 属性列表

参数	参数类型	描述
recall	JSON Array	训练作业模型单个分类召回率。
precision	JSON Array	训练作业模型单个分类精确率。
accuracy	JSON Array	训练作业模型单个分类准确率。

表 8-18 total_metric 属性列表

参数	参数类型	描述
total_metric_met a	JSON Array	预留字段。
total_reserved_d ata	JSON Array	预留字段。
total_metric_valu es	JSON Array	训练作业模型总测评参数指标。

表 8-19 total_metric_values 属性列表

参数	参数类型	描述
f1_score	Float	训练作业模型总召回。
recall	Float	训练作业模型总召回率。
precision	Float	训练作业模型总精确率。
accuracy	Float	训练作业模型总准确率。

8.1.5 更新训练作业描述

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

方式一:根据指定的job_id更新
from modelarts.session import Session
from modelarts.estimator import Estimator
session = Session()
estimator = Estimator(modelarts_session=session, job_id="182626")
job_description = estimator.update_job_description(description='update description')

- 方式二:根据**创建训练作业**生成的训练作业对象更新 job_description = job_instance.update_job_description(description='update description')
- 方式三:根据**查询训练作业版本列表**返回的指定训练作业版本对象更新 job_description = job_version_instance_list[0].update_job_description(description='update description')

参数说明

表 8-20 Estimator 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法请参见 <mark>Session鉴</mark> <mark>权</mark> 。
job_id	是	String	训练作业的ID。job_id可通过 <mark>创建训练作业</mark> 生成的训练作业对象查询,如 "job_instance.job_id"。或可通过 <mark>查询训</mark> <mark>练作业列表</mark> 的响应获取。

表 8-21 update_job_description 请求参数说明

参数	是否必选	参数类型	描述
description	是	String	需要更改的训练作业的描述 信息。

表 8-22 update_job_description 返回参数说明

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参 见 <mark>错误码</mark> 。 调用成功时无此字段。

参数	参数类型	描述
is_success	Boolean	接口调用是否成功。

8.1.6 获取训练作业日志的文件名

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• 方式一:根据指定的job_id和version_id获取

from modelarts.session import Session from modelarts.estimator import Estimator session = Session() estimator = Estimator(modelarts_session=session, job_id="182626", version_id="278813") job_log_list = estimator.get_job_log_file_list()

- 方式二:根据**创建训练作业**生成的训练作业对象获取 job_log_list = job_instance.get_job_log_file_list()
- 方式三:根据**查询训练作业版本列表**返回的指定训练作业版本对象获取 job_log_list = job_version_instance_list[0].get_job_log_file_list()

表 8-23 Estimator 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法请参见 <mark>Session鉴权</mark> 。
job_id	是	String	训练作业的ID。job_id可通过 创建训练作业 生成的训练作业对象查询,如 "job_instance.job_id"。或可通过 <mark>查询训练</mark> 作业列表的响应获取。
version_id	是	String	训练作业的版本ID。version_id可通过 <mark>创建训练作业</mark> 生成的训练作业对象查询,如 "job_instance.version_id"。或可通过 <mark>查询</mark> 训练作业列表 的响应获取。

表 8-24 get_job_log_file_list 返回参数说明

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

参数	参数类型	描述
error_code	String	调用失败时的错误码,具体请参见 <mark>错误</mark> 码。 调用成功时无此字段。
log_file_list	List	训练作业的日志文件名。单机作业日志仅 有一个文件,分布式作业日志有多个文 件。
is_success	Boolean	接口调用是否成功。

8.1.7 查询训练作业日志

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• 方式一:根据指定的job_id和version_id查询

from modelarts.session import Session from modelarts.estimator import Estimator session = Session() estimator = Estimator(modelarts_session=session, job_id="182626", version_id="278813") job_log = estimator.get_job_log(log_file='job-job-0713-191758.0') print(job_log)

- 方式二:根据**创建训练作业**生成的训练作业对象查询 job_log = job_instance.get_job_log(log_file='job-job-0713-191758.0') print(job_log)
- 方式三:根据<mark>查询训练作业版本列表</mark>返回的指定训练作业版本对象查询 job_log = job_version_instance_list[0].get_job_log(log_file='job-job-0713-191758.0') print(job_log)

表 8-25 Estimator 请求参数说明

参数	是否 必选	参数类 型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法请参见 <mark>Session鉴</mark> <mark>权</mark> 。
job_id	是	String	训练作业的ID。job_id可通过 <mark>创建训练作业</mark> 生成的训练作业对象查询,如 "job_instance.job_id"。或可通过 <mark>查询训</mark> <mark>练作业列表</mark> 的响应获取。
version_id	是	String	训练作业的版本ID。version_id可通过 <mark>创建</mark> <mark>训练作业</mark> 生成的训练作业对象查询,如 "job_instance.version_id"。或可通过 <mark>查询</mark> <mark>训练作业列表</mark> 的响应获取。

表 8-26 get_job_log 请求参数说明

参数	是否 必选	参数类型	描述
log_file	是	String	训练作业日志文件的文件名。
start_byte	否	Long	获取日志的起始位置,默认为0。 "start_byte"限制范围为[-1, +∞],如果 设置为-1,则表示获得最新长度为 "offset"的日志。
offset	否	Long	获取日志的长度,默认为2048。"offset" 限制长度为[-2048, 2048]。

表 8-27 get_job_log 返回参数说明

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参 见 <mark>错误码</mark> 。 调用成功时无此字段。
content	String	请求获得日志的内容。
lines	Integer	获得日志的行数。
start_line	String	该段日志内容的开始位置。
end_line	String	该段日志内容的结束位置。

8.1.8 删除训练作业

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

方式一: 根据指定的job_id删除 from modelarts.session import Session

from modelarts.estimator import Estimator

session = Session()

Estimator.delete_job_by_id(modelarts_session=session, job_id="155500")

方式二:根据创建训练作业生成的训练作业对象删除

status = job_instance.delete_job()

参数说明

表 8-28 delete_job_by_id 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
job_id	是	String	训练作业的id。job_id可通过 <mark>创建训练作业</mark> 生成的训练作业对象查询,如 "job_instance.job_id"。或可通过 <mark>查询训</mark> <mark>练作业列表</mark> 的响应获取。

表 8-29 delete_job_by_id 返回参数说明

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。
error_code	String	调用失败时的错误码。 调用成功时无此字段。
is_success	Boolean	接口调用是否成功。

8.2 训练作业版本

8.2.1 创建训练作业版本

创建训练作业版本的前提条件是训练作业已存在,可基于<mark>创建训练作业</mark>或者<mark>查询训练作业版本列表</mark>返回对象的job_id和version_id来创建。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 示例一:使用OBS存储位置创建训练作业版本

```
from modelarts.session import Session
from modelarts.estimator import Estimator
session = Session()
estimator = Estimator(
             modelarts_session=session,
             framework_type='PyTorch',
                                                             # AI引擎名称
             framework_version='PyTorch-1.0.0-python3.6',
                                                                   # AI引擎版本
             code_dir='/bucket/src/',
                                                           # 训练脚本目录
             boot_file='/bucket/src/pytorch_sentiment.py',
                                                                 # 训练启动脚本目录
             log_url='/bucket/log/',
                                                          # 训练日志目录
             hyperparameters=[
                        {"label":"classes",
```

● 示例二:使用数据集创建训练作业版本

```
from modelarts.session import Session
from modelarts.estimator import Estimator
session = Session()
estimator = Estimator(
              modelarts_session=session,
              framework_type='PyTorch',
                                                               # AI引擎名称
              framework_version='PyTorch-1.0.0-python3.6',
                                                                     # AI引擎版本
              code dir='/bucket/src/',
                                                            # 训练脚本目录
              boot_file='/bucket/src/pytorch_sentiment.py',
                                                                   # 训练启动脚本目录
              log url='/bucket/log/',
                                                            # 训练日志目录
              hyperparameters=[
                         {"label":"classes",
                         "value": "10"},
{"label":"lr",
                          "value": "0.001"}
              output_path='/bucket/output/',
                                                                # 训练输出目录
              train_instance_type='modelarts.vm.gpu.p100',
                                                                    # 训练环境规格
              train_instance_count=1,
                                                             # 训练节点个数
              job_description='pytorch-sentiment with ModelArts SDK')
                                                                        # 训练作业描述
job_version_instance = estimator.create_job_version(job_id='182626', pre_version_id=278813, inputs='/
bucket/data/train/', wait=False, job_desc='create a job version')
```

● 示例三: 创建自定义镜像的训练作业版本

```
from modelarts.session import Session
from modelarts.estimator import Estimator
session = Session()
estimator = Estimator(
              modelarts_session=session,
              log_url='/bucket/log/',
                                                             # 训练日志目录
              hyperparameters=[
                         {"label":"classes",
                         "value": "10"},

{"label":"lr",

"value": "0.001"}
              output_path='/bucket/output/',
                                                                # 训练输出目录
              train_instance_type='modelarts.vm.gpu.p100',
                                                                     # 训练环境规格
              train_instance_count=1,
                                                              # 训练节点个数
              user_command='bash -x /home/work/run_train.sh python /home/work/user-job-
dir/app/mnist/mnist_softmax.py --data_url /home/work/user-job-dir/app/
                                                   # 自定义镜像启动命令
mnist data'.
              user_image_url='100.125.5.235:20202/jobmng/cpu-base:1.0',
                                                                          # 自定义镜像下载地址
              job_description='pytorch-sentiment with ModelArts SDK')
                                                                         # 训练作业描述
job_version_instance = estimator.create_job_version(job_id='182626', pre_version_id=278813, inputs='/
bucket/data/train/', wait=False, job_desc='create a job version')
```

示例四:使用内置模型创建训练作业版本

```
"value": "0.001"}
],
output_path='/bucket/output/', # 训练输出目录
train_instance_type='modelarts.vm.gpu.p100', # 训练环境规格
train_instance_count=1, # 训练节点个数
job_description='pytorch-sentiment with ModelArts SDK') # 训练作业描述
job_version_instance = estimator.create_job_version(job_id='182626', pre_version_id=278813, inputs='/bucket/data/train/', wait=False, job_desc='create a job version')
```

参数说明

表 8-30 Estimator 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
train_instance_c ount	是	Long	训练作业worker的个数。
code_dir	否	String	训练作业的代码目录,如"/bucket/ src/"。当填入model_name时不需要填 写。
boot_file	否	String	训练作业的代码启动文件,需要在代码目录下,如"/bucket/src/boot.py"。当填入model_name时不需要填写。
model_name	否	Long	训练作业的内置算法模型名称。填入 model_name后app_url与boot_file_url不需 填写,framework_type和 framework_version也不需要填写。 "model_name"请从 <mark>查询预置算法</mark> 接口中 获取。
output_path	是	String	训练作业的输出位置。
hyperparameter s	否	JSON Array	训练作业的运行参数,为label-value格式, 其中lable和value的值均为String类型;当为 自定义镜像训练作业的时候,此参数为容器 环境变量。
log_url	否	String	训练作业的日志OBS输出路径URL,默认为空。如:"/usr/log/"。
train_instance_t ype	是	Long	训练作业选择的资源规格。若选择在训练平 台训练,请从 <mark>查询资源规格列表</mark> 接口获取。
framework_type	否	String	训练作业选择的引擎规格。请从 <mark>查询引擎规格列表</mark> 接口获取引擎规格。当填入model_name时不需要填写。
framework_versi on	否	String	训练作业选择的引擎版本。请从 <mark>查询引擎规</mark> <mark>格列表</mark> 接口获取引擎版本。当填入 model_name时不需要填写。

参数	是否 必选	参数类型	描述
user_image_url	否	String	自定义镜像训练作业的自定义镜像的SWR- URL。如: "100.125.5.235:20202/jobmng/ custom-cpu-base:1.0"。
user_command	否	String	自定义镜像训练作业的自定义镜像的容器的 启动命令。形式为: "bash /home/work/ run_train.sh python /home/work/user-job- dir/app/train.py {python_file_parameter}"。

表 8-31 create_job_version 请求参数说明

参数	是否 必选	参数类型	描述
job_id	是	String	训练作业的ID。job_id可通过 <mark>创建训练作业</mark> 生成的训练作业对象查询,如 "job_instance.job_id"。或可通过 <mark>查询训</mark> <mark>练作业列表</mark> 的响应获取。
pre_version_id	是	Long	训练作业前一版本的ID。pre_version_id可 通过 <mark>创建训练作业</mark> 生成的训练作业对象查 询,如"job_instance.version_id"。或通过 <mark>查询训练作业列表</mark> 的响应获取。
inputs	是	String	训练作业的数据存储位置。inputs和 dataset_id、dataset_version_id、 data_source不可同时出现,但必须有其 一。本地训练只支持该参数。
dataset_id	否	String	训练作业的数据集ID。应与 dataset_version_id同时出现,但不可与 inputs同时出现。获取数据集ID,请参见 <mark>查</mark> 看数据集的基本信息 。
dataset_version_ id	否	String	训练作业的数据集版本ID。应与dataset_id 同时出现,但不可与inputs同时出现。数据 集版本ID,请参见 <mark>查看数据集的基本信息</mark> 。
wait	否	Boolean	是否等待创建训练作业版本结束,默认为 "False"。
job_desc	否	String	训练作业版本的描述。

表	8-32	create_	job	_version	成功响应说明
---	------	---------	-----	----------	--------

参数	类型	描述
TrainingJob	Object	训练对象。该对象包含job_id、 version_id等属性,及对训练作业的查 询、更新、删除等操作,如可通过 job_version_instance.job_id获取训练作 业ID。

8.2.2 查询训练作业版本列表

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

from modelarts.session import Session from modelarts.estimator import Estimator session = Session() estimator = Estimator(session, job_id="182626") job_version_instance_list = estimator.get_job_version_object_list() print(job_version_instance_list)

参数说明

表 8-33 Estimator 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
job_id	是	String	训练作业的ID。job_id可通过 <mark>创建训练作业</mark> 生成的训练作业对象查询,如 "job_instance.job_id"。或可通过 <mark>查询训</mark> <mark>练作业列表</mark> 的响应获取。

表 8-34 get_job_version_object_list 请求参数说明

参数	是否 必选	参数类型	描述
is_show	否	Boolean	是否打印训练作业版本详情列表,默认为 "True"。

[&]quot;get_job_version_object_list"成功响应后返回训练对象列表,其中列表中的每个元素参数请参见表8-35。

表 8-35 TrainingJob 对象描述

参数	参数类型	描述
TrainingJob	Object	训练对象。该对象包含job_id、version_id等属性,及对训练作业的查询、更新、删除等操作,如可通过job_version_instance.job_id获取训练作业ID。

8.2.3 查询训练作业版本详情

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 方式一:根据指定的job_id查询

from modelarts.session import Session from modelarts.estimator import Estimator session = Session() estimator = Estimator(session, job_id="182626") job_version_info = estimator.get_job_version_info() print(job_version_info)

 方式二:根据创建训练作业版本生成的训练作业版本对象查询 job_version_info = job_version_instance.get_job_version_info() print(job_version_info)

参数说明

表 8-36 Estimator 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
job_id	是	String	训练作业的ID。job_id可通过 <mark>创建训练作业</mark> 生成的训练作业对象查询,如 "job_instance.job_id"。或可通过 <mark>查询训</mark> <mark>练作业列表</mark> 的响应获取。

表 8-37 get_job_version_info 返回参数说明

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

参数	参数类型	描述
error_code	String	调用失败时的错误码,具体请参见 <mark>错误码</mark> 。 调用成功时无此字段。
is_success	Boolean	接口调用是否成功。
job_id	Long	训练作业的ID。
job_name	String	训练作业的名称。
job_desc	String	训练作业的描述信息。
version_count	Long	训练作业的版本数。
versions	JSON Array	训练作业的运行版本参数。

表 8-38 versions 属性列表

参数	参数类型	描述
version_id	Long	训练作业的版本ID。
version_name	String	训练作业的版本名称。
pre_version_id	Long	训练作业前一版本的ID。
engine_type	Long	训练作业的引擎类型。
engine_id	Long	训练作业的引擎ID。
engine_version	String	训练作业的引擎版本。
status	Integer	训练作业的状态。
app_url	String	训练作业的代码目录。
boot_file_url	String	训练作业的代码启动文件。
create_time	Long	训练作业的创建时间。
parameter	JSON Array	训练作业的运行参数,为label-value格式; 当为自定义镜像训练作业的时候,此参数为 容器环境变量。
duration	Long	训练作业的运行时间,单位为毫秒。
spec_id	Long	训练作业资源规格ID。
core	String	资源规格的核数。
cpu	String	资源规格CPU内存。
gpu_num	Integer	资源规格GPU的个数。
gpu_type	String	资源规格GPU的类型。
worker_server_num	Integer	训练作业worker的个数。

参数	参数类型	描述
data_url	String	训练作业的数据集。
train_url	String	训练作业输出文件OBS路径。
log_url	String	训练作业的日志OBS输出路径URL,默认为 空。如:"/usr/log/"。
dataset_version_id	String	训练作业的数据集版本ID。
dataset_id	String	训练作业的数据集ID。
data_source	JSON Array	训练作业使用的多数据集。
model_id	String	训练作业的模型ID。
model_metric_list	JSON Array	训练作业的模型评测参数。
system_metric_list	JSON Array	训练作业的系统监控指标。
user_image_url	String	自定义镜像训练作业的自定义镜像的SWR-URL。
user_command	String	自定义镜像训练作业的自定义镜像的容器的启动命令。

表 8-39 data_source 属性列表

参数	参数类型	描述
dataset_id	String	训练作业的数据集ID。
dataset_version	String	训练作业的数据集版本ID。
type	String	数据集类型。 "obs":表示使用OBS的数据。 "dataset":表示使用数据集的数据。
data_url	String	OBS的桶路径。

表 8-40 model_metric_list 属性列表

参数	参数类型	描述
metric	JSON Array	训练作业的模型单个分类测评参数。
total_metric	JSON Array	训练作业的模型总测评参数。

表 8-41 system_metric_list 属性列表

参数	参数类型	描述
cpuUsage	JSON Array	训练作业CPU资源占用率。
memUsage	JSON Array	训练作业内存资源占用率。
gpuUtil	JSON Array	训练作业GPU资源占用率。

表 8-42 metric 属性列表

参数	参数类型	描述
metric_values	JSON Array	训练作业模型单个分类测评参数指标。
reserved_data	JSON Array	预留字段。
metric_meta	JSON Array	训练作业模型单个分类,包含类ID和类 名。

表 8-43 metric_values 属性列表

参数	参数类型	描述
recall	JSON Array	训练作业模型单个分类召回率。
precision	JSON Array	训练作业模型单个分类精确率。
accuracy	JSON Array	训练作业模型单个分类准确率。

表 8-44 total_metric 属性列表

参数	参数类型	描述
total_metric_meta	JSON Array	预留字段。
total_reserved_data	JSON Array	预留字段。
total_metric_values	JSON Array	训练作业模型总测评参数指标。

表 8-45 total_metric_values 属性列表

参数	参数类型	描述
f1_score	Float	训练作业模型总召回。
recall	Float	训练作业模型总召回率。
precision	Float	训练作业模型总精确率。

参数	参数类型	描述
accuracy	Float	训练作业模型总准确率。

8.2.4 停止训练作业版本

只有当作业处于运行中时,可以停止正在创建的训练作业版本。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 方式一:根据指定的job_id和version_id停止

from modelarts.session import Session from modelarts.estimator import Estimator session = Session() estimator = Estimator(session, job_id="182626", version_id="278813") status = estimator.stop_job_version()

- 方式二:根据**创建训练作业版本**生成的训练作业版本对象停止 status = job_version_instance.stop_job_version()
- 方式三:根据**查询训练作业版本列表**返回的指定训练作业版本对象停止 status = job_version_instance_list[0].stop_job_version()

参数说明

表 8-46 Estimator 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
job_id	是	String	训练作业的ID。job_id可通过 <mark>创建训练作业</mark> 生成的训练作业对象查询,如 "job_instance.job_id"。或可通过 <mark>查询训</mark> <mark>练作业列表</mark> 的响应获取。
version_id	是	String	训练作业的版本ID。version_id可通过 <mark>查询</mark> 训练作业版本列表 的响应获取。

表 8-47 stop_job_version 返回参数说明

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

参数	参数类型	描述
error_code	String	调用失败时的错误码,具体请参见 <mark>错误码</mark> 。 调用成功时无此字段。
is_success	Boolean	接口调用是否成功。

8.2.5 删除训练作业版本

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• 方式一:根据指定的job_id和version_id删除

from modelarts.session import Session from modelarts.estimator import Estimator session = Session() estimator = Estimator(session, job_id="182626", version_id="278813") status = estimator.delete_job_version()

- 方式二:根据**创建训练作业版本**生成的训练作业版本对象删除 status = job_version_instance.delete_job_version()
- 方式三:根据**查询训练作业版本列表**返回的指定训练作业版本对象删除 status = job_version_instance_list[0].delete_job_version()

表 8-48 Estimator 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
job_id	是	String	训练作业的ID。job_id可通过 <mark>创建训练作业</mark> 生成的训练作业对象查询,如 "job_instance.job_id"。或可通过 <mark>查询训</mark> <mark>练作业列表</mark> 的响应获取。
version_id	是	String	训练作业的版本ID。version_id可通过 <mark>查询</mark> 训练作业版本列表 的响应获取。

表 8-49 delete_job_version 返回参数说明

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

参数	参数类型	描述
error_code	String	调用失败时的错误码,具体请参见 错误 码 。 调用成功时无此字段。
is_success	Boolean	接口调用是否成功。

8.3 训练作业参数配置

8.3.1 创建训练作业参数

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 示例一:使用OBS存储位置创建训练作业参数

```
from modelarts.session import Session
from modelarts.estimator import Estimator
session = Session()
estimator = Estimator(
               modelarts_session=session,
                                                                  # AI引擎名称
               framework_type='PyTorch',
               framework_version='PyTorch-1.0.0-python3.6',
                                                                        # AI引擎版本
              code_dir='/bucket/src/',
boot_file='/bucket/src/pytorch_sentiment.py',
                                                                # 训练脚本目录
                                                                       # 训练启动脚本目录
               log_url='/bucket/log/',
                                                               # 训练日志目录
               hyperparameters=[
                          {"label":"classes",
"value": "10"},
{"label":"lr",
                            'value": "0.001"}
               output_path='/bucket/output/',
                                                                    # 训练输出目录
              train_instance_type='modelarts.vm.gpu.p100',
                                                                         # 训练环境规格
               train_instance_count=1)
                                                                  # 训练节点个数
job_config_instance = estimator.create_job_configs(config_name='my_job_config', inputs='/bucket/
data/train/', config_desc='my job config')
```

示例二:使用数据集创建训练作业参数

```
from modelarts.session import Session
from modelarts.estimator import Estimator
session = Session()
estimator = Estimator(
              modelarts_session=session,
              framework_type='PyTorch',
                                                              # AI引擎名称
              framework_version='PyTorch-1.0.0-python3.6',
                                                                    # AI引擎版本
              code_dir='/bucket/src/',
                                                            # 训练脚本目录
              boot_file='/bucket/src/pytorch_sentiment.py',
                                                                   # 训练启动脚本目录
              log_url='/bucket/log/',
                                                            # 训练日志目录
              hyperparameters=[
                        {"label":"classes",
"value": "10"},
                         {"label":"lr",
                          "value": "0.001"}
              output_path='/bucket/output/',
                                                                # 训练输出目录
              train_instance_type='modelarts.vm.gpu.p100',
                                                                    # 训练环境规格
              train_instance_count=1)
                                                              # 训练节点个数
```

job_config_instance = estimator.create_job_configs(config_name='my_job_config', dataset_id='4AZNvFkN7KYr5EdhFkH', dataset_version_id='UOF9BleSGArwVt0oI6T', config_desc='my job config')

表 8-50 Estimator 请求参数说明

参数	是否	参数类型	描述
	必选		
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
train_instance_c ount	是	Long	训练作业worker的个数。
code_dir	否	String	训练作业的代码目录,如"/bucket/ src/"。当填入model_name时不需要填 写。
boot_file	否	String	训练作业的代码启动文件,需要在代码目录 下,如"/bucket/src/boot.py"。当填入 model_name时不需要填写。
model_name	否	Long	训练作业的内置算法模型名称。填入 model_name后app_url与boot_file_url不需 填写,framework_type和 framework_version也不需要填写。
output_path	是	String	训练作业的输出位置。
hyperparameter s	否	JSON Array	训练作业的运行参数,为label-value格式; 当为自定义镜像训练作业的时候,此参数为 容器环境变量。
log_url	否	String	训练作业的日志OBS输出路径URL,默认为 空。如:"/usr/log/"。
train_instance_t ype	是	Long	训练作业选择的资源规格。若选择在训练平 台训练,请从 <mark>查询资源规格列表</mark> 接口获取。
framework_type	否	String	训练作业选择的引擎规格。请从 <mark>查询引擎规</mark> 格列表接口获取引擎规格。当填入 model_name时不需要填写。
framework_versi on	否	String	训练作业选择的引擎版本。请从 <mark>查询引擎规格列表</mark> 接口获取引擎版本。当填入 model_name时不需要填写。
job_description	否	String	训练作业的描述。
user_image_url	否	String	自定义镜像训练作业的自定义镜像的SWR- URL。如: "100.125.5.235:20202/jobmng/ custom-cpu-base:1.0"。

参数	是否 必选	参数类型	描述
user_command	否	String	自定义镜像训练作业的自定义镜像的容器的启动命令。形式为: "bash /home/work/run_train.sh python /home/work/user-job-dir/app/train.py {python_file_parameter}"。

表 8-51 create_job_configs 请求参数说明

参数	是否 必选	参数类型	描述
config_name	否	String	训练作业参数名称。限制为1-20位只含数字、字母、下划线或者中划线的名称。当不填写时,默认会按日期动态生成。
config_desc	否	String	对训练作业的描述,默认为空,字符串的长 度限制为[0,256]。
inputs	否	String	训练作业的OBS数据存储位置。
dataset_id	否	String	训练作业的数据集ID。应与 dataset_version_id同时出现,但不可与 inputs同时出现。
dataset_version_ id	否	String	训练作业的数据集版本ID。应与dataset_id 同时出现,但不可与inputs同时出现。

表 8-52 create_job_configs 成功响应说明

参数	类型	描述
TrainingJob	Object	训练对象。该对象包含config_name等属性,及对训练作业参数的查询、删除等操作,如可通过 job_config_instance.config_name获取训练作业参数名称。

8.3.2 查询训练作业参数对象列表

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

from modelarts.session import Session from modelarts.estimator import Estimator session = Session() job_config_instance_list = Estimator.get_job_configs_object_list(modelarts_session=session, is_show=True, per_page=10, page=1, sort_by="create_time", order="asc", search_content="configname") print(job_config_instance_list)

表 8-53 get_job_configs_object_list 请求参数说明

参数	是否必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
per_page	否	Integer	指定每一页展示作业参数的总量,默认 为"10","per_page"可选的范围为 [1,1000]。
page	否	Integer	指定要查询页的索引,默认为"1"。
sortBy/sort_by	否	String	当使用AK/SK认证方式时,参数名为sortBy;当使用用户名密码认证方式时,参数名为sort_by。指定查询的排序方式,默认是作业名称"job_name",目前支持的排序还有作业描述"job_desc",作业状态"status",运行时长"duration",引擎类型"engine_type"以及创建时间"create_time"。
order	否	String	可选值有: • "asc"为递增排序,默认为 "asc"。 • "desc"为递减排序。
search_content	否	String	指定要查询的文字信息,例如参数名 称。默认为空。
is_show	否	Boolean	是否打印训练作业参数列表,默认为 "True"。

[&]quot;get_job_configs_object_list"成功响应后返回训练对象列表,其中列表中的每个元素参数请参见表8-54。

表 8-54 TrainingJob 对象描述

参数	类型	描述
TrainingJob	Object	训练对象。该对象包含config_name等属性,及对训练作业参数的查询、删除等操作,如可通过 job_config_instance.config_name获取训练作业参数名称。

8.3.3 查询训练作业参数列表

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

from modelarts.session import Session from modelarts.estimator import Estimator session = Session() job_paras_list = Estimator.get_job_configs_list(modelarts_session=session, per_page=10, page=1, sort_by="create_time", order="asc", search_content="configname") print(job_paras_list)

表 8-55 get_job_configs_list 请求参数说明

参数	是否必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴</mark> 权。
per_page	否	Integer	指定每一页展示作业参数的总量,默认 为10,"per_page"可选的范围为 [1,1000]。
page	否	Integer	指定要查询页的索引,默认为1。
sortBy/sort_by	否	String	当使用AK/SK认证方式时,参数名为sortBy;当使用帐户认证方式时,参数名为sort_by。指定查询的排序方式,默认是作业名称"job_name",目前支持的排序还有作业描述"job_desc",作业状态"status",运行时长"duration",引擎类型"engine_type"以及创建时间"create_time"。
order	否	String	可选值有: • "asc"为递增排序,默认为 "asc"。 • "desc"为递减排序。
search_content	否	String	指定要查询的文字信息,例如参数名 称。默认为空。

表 8-56 get_job_configs_list 返回参数说明

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参见 <mark>错误</mark> 码。 调用成功时无此字段。
config_total_count	Integer	查询到的训练作业参数的总数。
configs	JSON Array	configs参数属性列表。
is_success	Boolean	调用是否成功。

表 8-57 configs 属性列表说明

参数	参数类型	描述
config_name	String	训练作业参数的名称。
config_desc	String	训练作业参数的描述信息。
create_time	Long	训练作业的创建时间。
engine_type	Short	训练作业的引擎类型。
engine_name	String	训练作业的引擎名称。
engine_id	Long	训练作业的引擎ID。
engine_version	String	训练作业使用的引擎版本。

8.3.4 查询训练作业参数详情

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 方式一: 根据指定的config_name查询

from modelarts.session import Session
from modelarts.sestimator import Estimator
session = Session()
estimator = Estimator(modelarts_session=session, config_name="my_job_config")
job_paras_info = estimator.get_job_configs_info()
print(job_paras_info)

 方式二: 根据创建训练作业参数返回的对象查询 job_paras_info = job_config_instance.get_job_configs_info() print(job_paras_info)

● 方式三: 根据**查询训练作业参数对象列表**返回的对象查询

job_paras_info = job_config_instance_list[0].get_job_configs_info()
print(job_paras_info)

表 8-58 Estimator 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
config_name	是	String	训练作业参数名称。

表 8-59 get_job_configs_info 返回参数说明

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参见 错误码。 调用成功时无此字段。
config_name	String	训练作业参数的名称。
config_desc	String	训练作业参数的描述信息。
worker_server_num	Integer	训练作业worker的个数。
app_url	String	训练作业的代码目录。
boot_file_url	String	训练作业的代码启动文件。
model_id	Long	训练作业的模型ID。
parameter	JSON Array	训练作业的运行参数,为label- value格式;当为自定义镜像训练作 业的时候,此参数为容器环境变 量。
spec_id	Long	训练作业资源规格ID。
data_url	String	训练作业的数据集。
dataset_id	String	训练作业的数据集ID。
dataset_version_id	String	训练作业的数据集版本ID。
engine_type	Short	训练作业的引擎类型。
engine_name	String	训练作业的引擎名称。
engine_id	Long	训练作业的引擎ID。

参数	参数类型	描述
engine_version	String	训练作业使用的引擎版本。
train_url	String	训练作业的输出文件OBS路径URL, 默认为空,如"/usr/train/"。
log_url	String	训练作业的日志OBS输出路径URL, 默认为空。如:"/usr/train/"。
user_image_url	String	自定义镜像训练作业的自定义镜像 的SWR-URL。
user_command	String	自定义镜像训练作业的自定义镜像 的容器的启动命令。
is_success	Boolean	接口调用是否成功。

8.3.5 更新训练作业参数

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 示例一:使用OBS存储位置更新训练作业参数

```
from modelarts.session import Session
from modelarts.estimator import Estimator
session = Session()
estimator = Estimator(
              modelarts_session=session,
                                                                 # AI引擎名称
              framework_type='PyTorch',
              framework_version='PyTorch-1.0.0-python3.6',
                                                                       # AI引擎版本
              code_dir='/bucket/src/',
boot_file='/bucket/src/pytorch_sentiment.py',
                                                               # 训练脚本目录
                                                                      # 训练启动脚本目录
              log_url='/bucket/log/',
                                                              # 训练日志目录
              hyperparameters=[
{"label":"classes",
                          "value": "10"},
{"label":"lr",
                           "value": "0.001"}
              output_path='/bucket/output/',
                                                                   # 训练输出目录
              train_instance_type='modelarts.vm.gpu.p100',
                                                                        # 训练环境规格
              train_instance_count=1)
                                                                 # 训练节点个数
update_info = estimator.update_job_configs(config_name='my_job_config', inputs='/bucket/dataset/',
config_desc='update')
```

示例二:使用数据集更新训练作业参数

```
from modelarts.session import Session
from modelarts.estimator import Estimator
session = Session()
estimator = Estimator(
             modelarts_session=session,
             framework_type='PyTorch',
                                                             # AI引擎名称
             framework_version='PyTorch-1.0.0-python3.6',
                                                                   # AI引擎版本
             code_dir='/bucket/src/',
                                                           # 训练脚本目录
             boot_file='/bucket/src/pytorch_sentiment.py',
                                                                 # 训练启动脚本目录
             log_url='/bucket/log/',
                                                          # 训练日志目录
             hyperparameters=[
                        {"label":"classes",
```

参数说明

表 8-60 Estimator 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
train_instance_c ount	是	Long	训练作业worker的个数。
code_dir	否	String	训练作业的代码目录,如"/bucket/ src/"。当填入model_name时不需要填 写。
boot_file	否	String	训练作业的代码启动文件,需要在代码目录下,如"/bucket/src/boot.py"。当填入model_name时不需要填写。
model_name	否	Long	训练作业的内置算法模型名称。填入 model_name后app_url与boot_file_url不需 填写,framework_type和 framework_version也不需要填写。
output_path	是	String	训练作业的输出位置。
hyperparameter s	否	JSON Array	训练作业的运行参数,为label-value格式; 当为自定义镜像训练作业的时候,此参数为 容器环境变量。
log_url	否	String	训练作业的日志OBS输出路径URL,默认为 空。如:"/usr/log/"。
train_instance_t ype	是	Long	训练作业选择的资源规格。若选择在训练平 台训练,请从 <mark>查询资源规格列表</mark> 接口获取。
framework_type	否	String	训练作业选择的引擎规格。请从 <mark>查询引擎规</mark> 格列表接口获取引擎规格。当填入 model_name时不需要填写。
framework_versi on	否	String	训练作业选择的引擎版本。请从 <mark>查询引擎规格列表</mark> 接口获取引擎版本。当填入 model_name时不需要填写。
job_description	否	String	训练作业的描述。

参数	是否 必选	参数类型	描述
user_image_url	否	String	自定义镜像训练作业的自定义镜像的SWR- URL。如: "100.125.5.235:20202/jobmng/ custom-cpu-base:1.0"。
user_command	否	String	自定义镜像训练作业的自定义镜像的容器的 启动命令。形式为: "bash /home/work/ run_train.sh python /home/work/user-job- dir/app/train.py {python_file_parameter}"。

表 8-61 update_job_configs 请求参数说明

参数	是否必选	参数类 型	描述
config_name	是	String	训练作业参数名称。限制为1-20位只含数字、字母、下划线或者中划线的名称。当不填写时,默认会按日期动态生成。
config_desc	否	String	对训练作业的描述,默认为空,字符串的长 度限制为[0,256]。
inputs	否	String	训练作业的OBS数据存储位置。
dataset_id	否	String	训练作业的数据集ID。应与 dataset_version_id同时出现,但不可与 inputs同时出现。
dataset_version _id	否	String	训练作业的数据集版本ID。应与dataset_id 同时出现,但不可与inputs同时出现。
data_source	否	JSON Array	训练作业使用的数据集。不可与inputs、 dataset_id、dataset_version_id同时使用。

表 8-62 data_source 属性列表

参数	是否必 选	参数类 型	描述
dataset_id	否	String	训练作业的数据集ID。
dataset_version	否	String	训练作业的数据集版本ID。
type	是	String	数据集类型。可选值: "obs"、 "dataset"。
data_url	否	String	obs的桶路径,不可与dataset_id/ dataset_version同时出现。

表 8-63 update_job_configs 返回参数

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参见 <mark>错误码。</mark> 码。 调用成功时无此字段。
is_success	Boolean	接口调用是否成功。

8.3.6 删除训练作业参数

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• 方式一: 根据指定的config_name删除

from modelarts.session import Session from modelarts.estimator import Estimator session = Session() estimator = Estimator(modelarts_session=session, config_name="my_job_config") status = estimator.delete_job_configs()

- 方式二: 根据**创建训练作业参数**生成的训练作业版本对象删除 status = job_config_instance.delete_job_configs()
- 方式三: 根据**查询训练作业参数对象列表**返回的指定训练作业版本对象删除 status = job_config_instance_list[0].delete_job_configs()

参数说明

表 8-64 Estimator 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
config_name	是	String	训练作业参数名称。

表 8-65 delete_job_configs 返回参数说明

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。
		调用成功时无此字段。

参数	参数类型	描述
error_code	String	调用失败时的错误码,具体请参见 <mark>错误码</mark> 。 调用成功时无此字段。
is_success	Boolean	接口调用是否成功。

8.4 可视化作业

8.4.1 创建可视化作业

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

from modelarts.session import Session from modelarts.estimator import VisualizationJob session = Session() job = VisualizationJob(modelarts_session=session)

job_visualization_instance = job.create_visualization_job(train_url='/bucket/train/',

job_name='visualization_job', job_desc='my visualization job')

表 8-66 create_visualization_job 请求参数

参数	是否必选	参数类型	描述
job_name	否	String	可视化作业名称。限制为1-20位只含数字,字母,下划线,中划线的名称。
job_desc	否	String	对可视化作业的描述,默认为空,字符 串的长度限制为[0, 256]。
train_url	是	String	OBS路径地址。可视化文件路径,提供给可视化作业读取显示的可视化文件,通常位于训练作业的训练输出位置,在训练代码中使用"tf.summary"或"tensorboardx.SummaryWriter"等模块生成,文件名通常以"events.out.tfevents"开头。

表 8-67 create_visualization_job 成功响应说明

参数	类型	描述
VisualizationJob	Object	可视化作业对象。该对象包含 visualization_id、create_time、 job_name、status属性,及对可视化作业 的查询、更新、停止、重启和删除等操 作。

表 8-68 VisualizationJob 属性说明

参数	参数类型	描述
create_time	Long	可视化作业的创建时间。
job_name	String	可视化作业的名称。
status	Byte	可视化训练作业的运行状态,详细 作业状态列表请参见 作业状态参 考。
job_id	String	可视化作业ID。
is_success	Boolean	接口调用是否成功。

8.4.2 查询可视化作业对象列表

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

from modelarts.session import Session from modelarts.estimator import VisualizationJob session = Session()

job_visualization_instance_list = VisualizationJob.get_visualization_job_object_list(modelarts_session=session, is_show=True, status=8, per_page=10, page=1, sort_by="create_time", order="asc", search_content="job") print(job_visualization_instance_list)

表 8-69 get_visualization_job_object_list 请求参数说明

参数	是否必 选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 Session<mark>鉴权</mark>。
status	否	String	可视化作业的运行状态,详细作业状态列 表请查看 作业状态参考 。

参数	是否必选	参数类型	描述
per_page	否	Integer	指定每一页展示作业的总量,默认为 "10","per_page"可选的范围为 [1,100]。
page	否	Integer	指定要查询页的索引,默认为"1"。
sortBy/sort_by	否	String	当使用AK/SK认证方式时,参数名为sortBy;当使用用户名密码认证方式时,参数名为sort_by。指定查询的排序方式,默认是作业名称"job_name",目前支持的排序还有作业描述"job_desc",作业状态"status",运行时长"duration"以及创建时间"create_time",日志存储目录"log_dir"。
order	否	String	可选值。 • "asc"为递增排序,默认为 "asc"。 • "desc"为递减排序。
search_content	否	String	指定要查询的文字信息,例如可视化作业 名字,默认为空,字符串的长度为[0, 100]。
is_show	否	Boolean	是否打印可视化作业列表,默认为 "True"。

表 8-70 get_visualization_job_object_list 成功响应后返回可视化作业对象列表,其中列表中的每个元素参数说明如下:

参数	类型	描述
VisualizationJob	Object	可视化作业对象。该对象包含 visualization_id、create_time、 job_name、status属性,及对可视化作业 的查询、更新、停止、重启和删除等操 作。

表 8-71 VisualizationJob 属性说明

参数	参数类型	描述
create_time	Long	可视化作业的创建时间。
job_name	String	可视化作业的名称。

参数	参数类型	描述
status	Byte	可视化训练作业的运行状态,详细作业状 态列表请参见 <mark>作业状态参考</mark> 。
job_id	String	可视化作业ID。
is_success	Boolean	接口调用是否成功。

8.4.3 查询可视化作业列表

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

from modelarts.session import Session from modelarts.estimator import VisualizationJob session = Session() job_list = VisualizationJob.get_visualization_job_list(modelarts_session=session, status=8, per_page=10, page=1, sort_by="create_time", order="asc", search_content="job") print(job_list)

表 8-72 get_visualization_job_list 请求参数说明

参数	是否必 选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 Session<mark>鉴权</mark>。
status	否	String	可视化作业的运行状态,详细作业状态列 表请查看 作业状态参考 。
per_page	否	Integer	指定每一页展示作业的总量,默认为 "10","per_page"可选的范围为[1, 100]。
page	否	Integer	指定要查询页的索引,默认为"1"。
sortBy/sort_by	否	String	当使用AK/SK认证方式时,参数名为sortBy;当使用用户名密码认证方式时,参数名为sort_by。指定查询的排序方式,默认是作业名称"job_name",目前支持的排序还有作业描述"job_desc",作业状态"status",运行时长"duration"以及创建时间"create_time",日志存储目录"log_dir"。

参数	是否必 选	参数类型	描述
order	否	String	可选值。
search_content	否	String	指定要查询的文字信息,例如可视化作业 名字,默认为空,字符串的长度为[0, 100]。

表 8-73 get_visualization_job_list 返回参数说明

参数	参数类型	描述
error_code	String	调用失败时的错误码,具体请参见 <mark>错误</mark> <mark>码</mark> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。
job_total_count	Integer	查询的可视化作业总数。
job_count_limit	Integer	用户还可以创建可视化作业的数量。
jobs	JSON Array	可视化作业的属性列表,具体请参见 <mark>表</mark> 8-74。

表 8-74 jobs 属性列表

参数	参数类型	描述
job_id	Integer	可视化作业ID。
job_name	String	可视化作业的名称。
status	Integer	可视化作业的运行状态,详细作业状态 列表请参见 <mark>作业状态参考</mark> 。
create_time	Long	可视化作业的创建时间。
duration	Long	可视化作业的运行时长,单位为毫秒。
job_desc	String	可视化作业的具体描述。
service_url	String	可视化作业的endpoint。
train_url	String	可视化作业的日志存储路径。

8.4.4 查询可视化作业详情

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 方式一: 根据指定的visualization_id查询

from modelarts.session import Session from modelarts.estimator import VisualizationJob session = Session() job = VisualizationJob(modelarts_session=session, visualization_id='8992') job_info = job.get_visualization_job_info() print(job_info)

- 方式二: 根据**创建可视化作业**生成的可视化作业对象查询 job_info = job_visualization_instance.get_visualization_job_info() print(job_info)
- 方式三: 根据查询可视化作业对象列表返回的指定可视化作业对象查询 job_info = job_visualization_instance_list[0].get_visualization_job_info() print(job_info)

参数说明

表 8-75 VisualizationJob 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
visualization_id	是	String	可视化作业ID。

表 8-76 get_visualization_job_info 返回参数说明

参数	参数类型	描述
error_code	String	调用失败时的错误码,具体请参见 <mark>错误码</mark> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。
job_name	String	可视化作业的名称。
service_url	String	可视化作业的endpoint。
is_success	Boolean	接口调用是否成功。
duration	Long	可视化训练作业的运行时间。
create_time	Long	可视化训练作业的创建时间。
train_url	String	可视化训练作业输出文件OBS路径。

参数	参数类型	描述
job_id	Long	可视化训练作业的ID。
job_desc	String	可视化训练作业的描述信息。
resource_id	String	可视化训练作业的资源ID。
status	Integer	可视化作业的运行状态,详细作业状态列表 请参见 作业状态参考 。

8.4.5 更新可视化作业描述

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 方式一: 根据指定的visualization_id更新

from modelarts.session import Session from modelarts.estimator import VisualizationJob session = Session()

job = VisualizationJob(modelarts_session=session, visualization_id='8992') job_description = job.update_visualization_job(job_desc='update visualization job')

- 方式二: 根据**创建可视化作业**生成的可视化作业对象更新 job_description = job_visualization_instance.update_visualization_job(job_desc='update visualization_instance.update_visualization_job(job_desc='update visualization_job(job_desc='update visual
- 方式三: 根据**查询可视化作业对象列表**返回的指定可视化作业对象更新 job_description = job_visualization_instance_list[0].update_visualization_job(job_desc='update visualization job')

参数说明

表 8-77 VisualizationJob 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
visualization_id	是	String	可视化作业ID。

表 8-78 update_visualization_job 请求参数说明

参数	是否 必选	参数类 型	描述
job_desc	是	String	对可视化作业的描述,字符串的长度限制为 [0,256]。

表 8-79 update_visualization_job 返回参数	饭明
--------------------------------------	----

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参见 <mark>错误码</mark> 。 调用成功时无此字段。
is_success	Boolean	接口调用是否成功。

8.4.6 停止可视化作业

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• 方式一: 根据指定的visualization_id停止

from modelarts.session import Session from modelarts.estimator import VisualizationJob session = Session() job = VisualizationJob(modelarts_session=session, visualization_id='8992') status = job.stop_visualization_job()

- 方式二: 根据**创建可视化作业**生成的可视化作业对象停止 status = job_visualization_instance.stop_visualization_job()
- 方式三: 根据**查询可视化作业对象列表**返回的指定可视化作业对象停止 status = job_visualization_instance_list[0].stop_visualization_job()

参数说明

表 8-80 VisualizationJob 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
visualization_id	是	String	可视化作业ID。

表 8-81 stop_visualization_job 返回参数说明

参数	参数类型 描述	
error_code	String	调用失败时的错误码,具体请参见 错误码 。 调用成功时无此字段。

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。
is_success	Boolean	接口调用是否成功。

8.4.7 重启可视化作业

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• 方式一: 根据指定的visualization_id重启

from modelarts.session import Session from modelarts.estimator import VisualizationJob session = Session() ioh = VisualizationJoh(modelarts session=session visualizationJoh(modelarts session=session)

job = VisualizationJob(modelarts_session=session, visualization_id='8992')
resp = job.restart_visualization_job()

- 方式二: 根据**创建可视化作业**生成的可视化作业对象重启 status = job_visualization_instance.restart_visualization_job()
- 方式三: 根据**查询可视化作业对象列表**返回的指定可视化作业对象重启 status = job_visualization_instance_list[0].restart_visualization_job()

参数说明

表 8-82 VisualizationJob 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
visualization_id	是	String	可视化作业ID。

表 8-83 restart_visualization_job 返回参数说明

参数	参数类型	描述
error_code	String	调用失败时的错误码,具体请参见 <mark>错误码</mark> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。
is_success	Boolean	接口调用是否成功。

8.4.8 删除可视化作业

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session 鉴权请参见Session鉴权。

方式一: 根据指定的visualization_id删除

from modelarts.session import Session from modelarts.estimator import VisualizationJob session = Session() job = VisualizationJob(modelarts_session=session, visualization_id='8992')

status = job.delete_visualization_job()

- 方式二: 根据创建可视化作业生成的可视化作业对象删除 status = job_visualization_instance.delete_visualization_job()
- 方式三: 根据查询可视化作业对象列表返回的指定可视化作业对象删除 status = job_visualization_instance_list[0].delete_visualization_job()

参数说明

表 8-84 VisualizationJob 请求参数说明

参数	是否 必选	参数类型	描述
modelarts_sessi on	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
visualization_id	是	String	可视化作业ID。

表 8-85 delete_visualization_job 返回参数说明

参数	参数类型	描述
error_code	String	调用失败时的错误码,具体请参见 <mark>错误码</mark> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。
is_success	Boolean	接口调用是否成功。

8.5 资源和引擎规格接口

8.5.1 查询预置算法

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

from modelarts.session import Session from modelarts.estimator import Estimator session = Session() algo_info = Estimator.get_built_in_algorithms(modelarts_session=session) print(algo_info)

参数说明

表 8-86 get_built_in_algorithms 请求参数说明

参数	是否必选	参数类型	描述
modelarts_session	是	Object	会话对象,初始化方法见 <mark>Session</mark> <mark>鉴权</mark> 。

表 8-87 get_built_in_algorithms 返回参数说明

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。
error_code	String	调用失败时的错误码,具体请参见 <mark>错误码。</mark> 码。 调用成功时无此字段。
model_total_count	Integer	模型的数量。
models	JSON Array	模型的参数列表。
is_success	Boolean	接口调用是否成功。

表 8-88 models 说明

参数	参数类型	描述
model_id	Integer	模型ID。
model_name	String	模型名称。

参数	参数类型	描述
model_usage	Integer	模型用途。
		● 1代表图像分类
		● 2代表检测物体的类别和位置
		● 3代表图像语义分割
		● 4代表自然语言处理
model_precision	String	模型精度描述。
model_size	Long	模型大小,单位为字节(Byte)。
model_train_dataset	String	模型训练数据集。
model_dataset_forma t	String	使用模型需要的数据集格式。
model_description_url	String	模型描述链接。
parameter	JSON Array	模型的运行参数,为label-value格式; 当为自定义镜像训练作业的时候,此参 数为容器环境变量。该样例请参考请求 样例。
create_time	Long	模型的创建时间。
engine_id	Long	模型的引擎ID。
engine_name	String	模型的引擎名称。
engine_version	String	模型的引擎版本。

8.5.2 查询资源规格列表

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

from modelarts.session import Session from modelarts.estimator import Estimator session = Session() algo_info = Estimator.get_train_instance_types(modelarts_session=session) print(algo_info)

参数说明

表 8-89 请求参数说明

参数	是否必 选	参数类型	描述
modelarts_session	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。

表 8-90 成功响应参数说明

参数类型	描述	
List	资源规格参数列表。	

表 8-91 失败响应参数说明

参数	参数类型	描述
error_code	String	调用失败时的错误码。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

8.5.3 查询引擎规格列表

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

from modelarts.session import Session
from modelarts.estimator import Estimator
session = Session()
engine_list = Estimator.get_framework_list(modelarts_session=session)
print(engine_list)

参数说明

表 8-92 get_framework_list 请求参数说明

参数	是否必选	参数类型	描述
modelarts_session	是	Object	会话对象,初始化方法见 <mark>Session</mark> <mark>鉴权</mark> 。

表 8-93 get_framework_list 成功响应参数说明

参数类型	描述	
List	引擎规格参数列表,请参见 <mark>表8-94</mark> 。	

表 8-94 framework_list 参数说明

参数	参数类型	描述
framework_type	String	引擎类型。
framework_versio	String	引擎版本。

表 8-95 失败响应参数说明

参数	参数类型	描述
error_code	String	调用失败时的错误码。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

8.6 作业状态参考

作业状态如表8-96所示。

表 8-96 作业状态

状态值	说明
0	JOBSTAT_UNKNOWN,作业状态未知。
1	JOBSTAT_INIT,作业初始化状态。
2	JOBSTAT_IMAGE_CREATING,作业镜像正在创建。
3	JOBSTAT_IMAGE_FAILED,作业镜像创建失败。
4	JOBSTAT_SUBMIT_TRYING,作业正在提交。
5	JOBSTAT_SUBMIT_FAILED,作业提交失败。
6	JOBSTAT_DELETE_FAILED,作业删除失败。
7	JOBSTAT_WAITING,作业正在排队中。
8	JOBSTAT_RUNNING,作业正在运行中。
9	JOBSTAT_KILLING,作业正在取消。
10	JOBSTAT_COMPLETED,作业已经完成。
11	JOBSTAT_FAILED,作业运行失败。
12	JOBSTAT_KILLED,作业取消成功。

状态值	说明
13	JOBSTAT_CANCELED,作业取消。
14	JOBSTAT_LOST,作业丢失。
15	JOBSTAT_SCALING,作业正在扩容。
16	JOBSTAT_SUBMIT_MODEL_FAILED,提交模型失败。
17	JOBSTAT_DEPLOY_SERVICE_FAILED,部署服务失败。
18	JOBSTAT_CHECK_INIT,审核作业初始化。
19	JOBSTAT_CHECK_RUNNING,审核作业正在运行中。
20	JOBSTAT_CHECK_RUNNING_COMPLETED,审核作业已经完成。
21	JOBSTAT_CHECK_FAILED,审核作业失败。
22	MOUNT_FAILED,挂载失败。

9 模型管理

9.1 模型调试

训练完成后,可先创建本地模型,在本地调试完成后再部署到推理服务上。推荐使用tensorflow_mlp_mnist_local_mode案例学习,单击如下链接直接进入ModelArts使用。

Run in ModelArts

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

步骤1 将自定义的推理文件和模型配置文件保存在训练生成的模型文件目录下。如训练生成的模型保存在"/home/ma-user/work/tensorflow_mlp_mnist_local_mode/train/model/"中,则推理文件"customize_service.py"和模型配置文件"config.json"也保存在该目录中。

步骤2 创建模型运行的conda虚拟环境。

步骤3 创建本地模型。

本地模型创建好后,可以参见服务调试部署为本地服务。

步骤4 本地模型创建完,可以调用接口发布模型。

model.publish_model(obs_location=obs_location)

指定参数"obs_location"后,会将本地的模型文件上传到该目录下。参数可省略,示例如下:

model.publish_model()

此时模型文件会上传到默认OBS桶以当前时间戳结尾的目录中。该目录会在命令执行后打印出来,示例如下:

Successfully upload file /home/ma-user/work/tensorflow_mlp_mnist_local_mode/train/model to OBS modelarts-cn-north-4-08aae033/model-0107-224502

----结束

表 9-1 创建模型场景参数说明

参数	是否	参数类型	描述	
	必选			
session	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。	
model_nam e	否	String	模型名称,名称只能字母,中文开头,为字母、数字、下划线、中文或者中划线组成的合法字符,支持1-64个字符。若未输入该参数,系统会自动生成模型name。	
model_vers ion	是	String	模型版本,格式需为"数值.数值.数值",其中数值为1-2位正整数。版本不可以出现以0开头的版本号形式,如"01.01.01"等。	
publish	否	Bool	是否发布模型。可选值:	
source_loca tion_type	否	String	模型位置类型。可选值: OBS_SOURCE: source_location为OBS路径。(默认值) LOCAL_SOURCE: source_location为本地路径。	
source_loca tion	是	String	模型文件所在路径,指定到模型文件的父目录。 • 当source_location_type为OBS_SOURCE时,模型文件所在路径为OBS路径,格式为"/obs_bucketname//model_file_parent_dir/"。 • 当source_location_type为LOCAL_SOURCE时,模型文件所在路径为本地路径,格式为"/local_path//model_file_parent_dir/"。	

参数	是否 必选	参数类型	描述	
environme nt	否	Environme nt实例	描述模型正常运行需要的环境,如使用的 python版本、tensorflow版本等。请参见 <mark>表9-2</mark>	
source_job_ id	否	String	来源训练作业的ID,模型是从训练作业产生的可填写,用于溯源;如模型是从第三方元模型导入,则为空,默认值为空。	
source_job_ version	否	String	来源训练作业的版本,模型是从训练作业产生的可填写,用于溯源;如模型是从第三方元模型导入,则为空,默认值为空。	
source_type	否	String	模型来源的类型,当前仅可取值auto,用于区分通过自动学习部署过来的模型(不提供模型下载功能);用户通过训练作业部署的模型不设置此值。默认值为空。	
model_type	是	String	模型类型,取值为:TensorFlow/MXNet/ Spark_MLlib/Scikit_Learn/XGBoost/ MindSpore/Image/PyTorch。	
model_algo rithm	否	String	模型算法,表示模型的算法实现类型,如果已在模型配置文件中配置,则可不填。如: predict_analysis、object_detection 、 image_classification。	
description	否	String	模型描述信息,不超过100个字符,且不能包含特殊字符!<>=&'"。	
execution_c ode	否	String	存放执行脚本的OBS路径。推理脚本必须放于模型所在路径(请参见"source_location"参数)的model目录下,名称固定为: "customize_service.py"。	
input_para ms	否	params结 构数组	模型推理输入参数列表,默认为空。如果已在模型配置文件中配置apis信息时,则可不填,后台自动从配置文件的apis字段中读取输入参数信息。	
output_par ams	否	params结 构数组	模型推理输出参数列表,默认为空。如果已在模型配置文件中配置apis信息时,则可不填,后台自动从配置文件的apis字段中读取输出参数信息。	
dependenci es	否	dependen cy结构数 组	运行代码及模型需安装的依赖包,默认为空。如果已在模型配置文件中配置dependencies信息时,则可不填,后台自动从配置文件的dependencies字段中读取需要安装的依赖包。	
apis	否	String	模型提供的推理接口列表,默认为空。如果已在模型配置文件中配置apis信息时,则可不填,后台自动从配置文件中的apis字段读取所配置的推理接口信息。	

表 9-2 Environment 参数说明

参数	是否必 选	类型	说明
name	是	String	环境名称。
conda	否	CondaDepend encies	conda环境,具体请参见 表9-3 。

表 9-3 CondaDependencies 参数说明

参数	是否必 选	类型	说明
channels	否	List	python包的下载源。
pip_packages	否	List	conda虚拟环境需要使用的 python包,如tensorflow, pillow等。
conda_packages	否	List	conda虚拟环境需要使用的conda 包,如指定python版本。

表 9-4 params 结构

20 · Paramo 2113				
参数	是否 必选	参数类型	描述	
url	是	String	模型推理接口的请求路径。	
param_name	是	String	参数名,不超过64个字符。	
param_type	是	String	JSON Schema基本参数类型,有string、 object、array、boolean、number、integer。	
min	否	Double	当param_type为int或float时,可选填,默认为 空。	
max	否	Double	当param_type为int或float时,可选填,默认为 空。	
param_desc	否	String	参数描述,不超过100个字符,默认为空。	

表 9-5 dependency 结构

参数	是否 必选	参数类型	描述
installer	是	String	安装方式,当前只支持"pip"。

参数	是否 必选	参数类型	描述
packages	是	package结 构数组	依赖包集合。

表 9-6 package 结构

参数	是否 必选	参数类型	描述
package_nam e	是	String	依赖包名称。
package_versi on	否	String	依赖包版本。
restraint	否	String	版本过滤条件,当且仅当package_version存在时必填。取值为: EXACT:等于给定版本 ATLEAST:不小于给定版本 ATMOST:不大于给定版本

表 9-7 创建模型返回参数说明

参数	是否 必选	参数类型	描述
model	是	Model对 象	模型对象,可以调用本章节模型管理的所有接口。

9.2 导入模型

导入模型功能包括:

- 初始化已存在的模型,根据模型ID生成模型对象。
- 创建模型。模型对象的属性,请参见**查询模型详情**。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见**Session鉴权**。

from modelarts.session import Session from modelarts.model import Model from modelarts.config.model_config import ServiceConfig,Params,Dependencies,Packages session = Session()

• 方式1: 初始化已存在模型

model_instance = Model(session, model_id="input your model id")

● 方式2: 创建模型

```
model_instance = Model(
           session,
           model_name="input model name",
                                                # 模型名称
                                          # 模型版本
           model_version="1.0.0",
           source_location=model_location,
                                             # 模型文件路径
           model_type="MXNet",
                                           # 模型类型
           model_algorithm="image_classification", # 模型算法
           execution_code="OBS_PATH",
           input_params=input_params,
                                             #参考input_params格式描述
           output params=output params,
                                              #参考output params格式描
述
           dependencies=dependencies,
                                             #参考dependencies格式描述
           apis=apis)
```

- 方式2中关于input_params和output_params参数组的定义格式

SDK提供了Params类对输入输出参数组的定义。input_params和output_params为list,list中的元组对象是Params。

以input params为例说明:

- 方式2中关于dependencies参数组的定义格式

SDK提供了Dependencies类对其定义,dependencies为list,list中的元组对象是Dependencies。

定义代码如下:

- 方式2中关于package参数组的定义格式

SDK提供了Packages类对其定义,packages为list,list中的元组对象是Packages。

定义代码如下:

参数说明

表 9-8 初始化模型场景参数说明

参数	是否必选	参数类型	描述
session	是	Object	会话对象,初始化方法参考 Session鉴权 。
model_id	是	String	模型ID。

表 9-9 创建模型场景参数说明

参数	是否 必选	参数类型	描述
session	是	Object	会话对象,初始化方法见 Session鉴权 。
model_nam e	否	String	模型名称,名称只能字母,中文开头,为字母、数字、下划线、中文或者中划线组成的合法字符,支持1-64个字符。若未输入该参数,系统会自动生成模型name。
model_vers ion	是	String	模型版本,格式需为"数值.数值.数值",其中数值为1-2位正整数。版本不可以出现以0开头的版本号形式,如"01.01.01"等。
publish	否	Bool	是否发布模型。可选值: True: 发布模型。(默认值) False: 不发布模型,创建本地模型,可用来 调试相关代码。
source_loca tion_type	否	String	模型位置类型。可选值: OBS_SOURCE: source_location为OBS路
source_loca tion	是	String	模型文件所在路径,指定到模型文件的父目录。 • 当source_location_type为OBS_SOURCE时,模型文件所在路径为OBS路径,格式为"/obs_bucketname//model_file_parent_dir/"。 • 当source_location_type为LOCAL_SOURCE时,模型文件所在路径为本地路径,格式为"/local_path//model_file_parent_dir/"。
environme nt	否	Environme nt实例	描述模型正常运行需要的环境,如使用的 python版本、tensorflow版本等。 Environment实例的示例请参见 <mark>示例代码</mark> 。

参数	是否必选	参数类型	描述
source_job_ id	否	String	来源训练作业的ID,模型是从训练作业产生的可填写,用于溯源;如模型是从第三方元模型导入,则为空,默认值为空。
source_job_ version	否	String	来源训练作业的版本,模型是从训练作业产生的 可填写,用于溯源;如模型是从第三方元模型导 入,则为空,默认值为空。
source_type	否	String	模型来源的类型,当前仅可取值auto,用于区分通过自动学习部署过来的模型(不提供模型下载功能);用户通过训练作业部署的模型不设置此值。默认值为空。
model_type	是	String	模型类型,取值为:TensorFlow/MXNet/ Spark_MLlib/Scikit_Learn/XGBoost/ MindSpore/Image/PyTorch。
model_algo rithm	否	String	模型算法,表示模型的算法实现类型,如果已在模型配置文件中配置,则可不填。如: predict_analysis、object_detection 、 image_classification。
description	否	String	模型描述信息,不超过100个字符,且不能包含 特殊字符!<>=&'"。
execution_c ode	否	String	存放执行脚本的OBS路径。推理脚本必须放于模型所在路径(请参见"source_location"参数)的model目录下,名称固定为: "customize_service.py"。
input_para ms	否	params结 构数组	模型推理输入参数列表,默认为空。如果已在模型配置文件中配置apis信息时,则可不填,后台自动从配置文件的apis字段中读取输入参数信息。
output_par ams	否	params结 构数组	模型推理输出参数列表,默认为空。如果已在模型配置文件中配置apis信息时,则可不填,后台自动从配置文件的apis字段中读取输出参数信息。
dependenci es	否	dependen cy结构数 组	运行代码及模型需安装的依赖包,默认为空。如果已在模型配置文件中配置dependencies信息时,则可不填,后台自动从配置文件的dependencies字段中读取需要安装的依赖包。
apis	否	String	模型提供的推理接口列表,默认为空。如果已在模型配置文件中配置apis信息时,则可不填,后台自动从配置文件中的apis字段读取所配置的推理接口信息。

表 9-10 params 结构

参数	是否 必选	参数类型	描述
url	是	String	模型推理接口的请求路径。
param_name	是	String	参数名,不超过64个字符。
param_type	是	String	JSON Schema基本参数类型,有string、 object、array、boolean、number、integer。
min	否	Double	当param_type为int或float时,可选填,默认为 空。
max	否	Double	当param_type为int或float时,可选填,默认为 空。
param_desc	否	String	参数描述,不超过100个字符,默认为空。

表 9-11 dependency 结构

参数	是否 必选	参数类型	描述
installer	是	String	安装方式,当前只支持"pip"。
packages	是	package结 构数组	依赖包集合。

表 9-12 package 结构

参数	是否 必选	参数类型	描述
package_nam e	是	String	依赖包名称。
package_versi on	否	String	依赖包版本。
restraint	否	String	版本过滤条件,当且仅当package_version存在 时必填。取值为: • EXACT:等于给定版本 • ATLEAST:不小于给定版本 • ATMOST:不大于给定版本

表 9-13 create_model 返回参数说明

参数	是否 必选	参数类型	描述
model_instan ce	是	Model对 象	模型对象,可以调用本章节模型管理的所有接口。

□ 说明

9.3 查询模型列表

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• 场景1: 查询当前用户所有模型

from modelarts.session import Session from modelarts.model import Model session = Session() model_list = Model.get_model_list(session) print(model_list)

• **场景2**:按照检索条件查询当前用户模型

from modelarts.session import Session from modelarts.model import Model session = Session() model_list = Model.get_model_list(session, model_status="published", model_name="digit", order="desc") print(model_list)

参数说明

表 9-14 查询检索参数说明

参数	是否必 选	参数类型	说明
model_nam e	否	String	模型名称,可支持模糊匹配。
model_versi on	否	String	模型版本。

参数	是否必 选	参数类型	说明
model_statu s	否	String	模型状态,可根据模型的"publishing"、 "published"、"failed"三种状态执行查 询。
description	否	String	描述信息,可支持模糊匹配。
offset	否	Integer	指定要查询页的索引,默认为"0"。
limit	否	Integer	指定每一页返回的最大条目数,默认为 "280"。
sort_by	否	String	指定排序字段,可选"create_at"、 "model_version"、"model_size",默认 是可选"create_at"。
order	否	String	排序方式,可选"asc"或"desc",代表递增排序及递减排序,默认是"desc"。
workspace_i d	否	String	工作空间ID,默认为"0"。

表 9-15 get_model_list 打印参数说明

参数	参数类型	描述
total_count	Integer	不分页的情况下,符合查询条件的总模型数 量。
count	Integer	模型数量。
models	model结构数组	模型元数据信息。

表 9-16 model 结构

参数	参数类型	描述
model_id	String	模型ID。
model_name	String	模型名称。
model_versio n	String	模型版本。
model_type	String	模型类型, 取值为: TensorFlow/MXNet/ Spark_MLlib/Scikit_Learn/XGBoost/MindSpore/ Image/PyTorch。
model_size	Long	模型大小,单位为字节数。
tenant	String	模型归属租户。

参数	参数类型	描述
project	String	模型归属项目。
owner	String	模型归属用户。
create_at	Long	模型创建时间,距'1970.1.1 0:0:0 UTC'的毫秒数。
description	String	模型描述信息。
source_type	String	模型来源的类型。

9.4 查询模型对象列表

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• **场景1**: 查询当前用户所有模型对象

from modelarts.session import Session from modelarts.model import Model session = Session() model_object_list = Model.get_model_object_list(session) print(model_object_list)

● 场景2:按照检索条件查询当前用户模型对象

from modelarts.session import Session from modelarts.model import Model session = Session() model_object_list = Model.get_model_object_list(session, model_status="published", model_name="digit", order="desc") print(model_object_list)

参数说明

- 查询模型列表,返回list,list大小等于当前用户所有已经部署的模型个数,list中每个元素都是Model对象,对象属性和查询模型详情相同。查询模型列表返回说明:model_list = [model_instance1, model_instance2, model_instance3 ...],列表中元素model_instance对象即为本章节描述的模型管理,可调用模型接口。
- 支持按照检索参数查询模型列表,返回满足检索条件的模型list,检索参数如表 9-17所示。
- 在查询列表时,返回list的同时,会打印模型列表的详细信息,如**表9-18**和**表9-19** 所示。
- 当前支持最大获取150个模型对象。

表 9-17 查询检索参数说明

参数	是否必 选	参数类 型	说明
model_nam e	否	String	模型名称,可支持模糊匹配。

参数	是否必 选	参数类 型	说明
model_versi on	否	String	模型版本。
model_statu s	否	String	模型状态,可根据模型的"publishing"、 "published"、"failed"三种状态执行查 询。
description	否	String	描述信息,可支持模糊匹配。
offset	否	Integer	指定要查询页的索引,默认为"0"。
limit	否	Integer	指定每一页返回的最大条目数,默认为 "280"。
sort_by	否	String	指定排序字段,可选"create_at"、 "model_version"、"model_size",默认 是可选"create_at"。
order	否	String	排序方式,可选"asc"或"desc",代表递增排序及递减排序,默认是"desc"。
workspace_i d	否	String	工作空间ID,默认为"0"。

表 9-18 get_model_list 打印参数说明

参数	参数类型	描述
total_count	Integer	不分页的情况下,符合查询条件的总模型数 量。
count	Integer	模型数量。
models	model结构数组	模型元数据信息。

表 9-19 model 结构

参数	参数类型	描述
model_id	String	模型ID。
model_name	String	模型名称。
model_versio n	String	模型版本。
model_type	String	模型类型, 取值为: TensorFlow/MXNet/ Spark_MLlib/Scikit_Learn/XGBoost/MindSpore/ Image/PyTorch。

参数	参数类型	描述	
model_size	Long	模型大小,单位为字节数。	
tenant	String	模型归属租户。	
project	String	模型归属项目。	
owner	String	模型归属用户。	
create_at	Long	模型创建时间,距'1970.1.1 0:0:0 UTC'的毫秒数。	
description	String	模型描述信息。	
source_type	String	模型来源的类型,仅当模型为自动学习部署过来时 有值,取值为auto。	

9.5 查询模型详情

查询当前模型对象的信息。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• 方式1:根据导入模型生成的模型对象进行模型详情查询

from modelarts.session import Session
from modelarts.model import Model
session = Session()
model_instance = Model(session, model_id="input your model_id")
model_info = model_instance.get_model_info()
print(model_info)

• 方式2:根据**查询模型对象列表**返回的模型对象进行模型详情查询

from modelarts.session import Session from modelarts.model import Model session = Session() model_object_list = Model.get_model_object_list(session) model_instance = model_object_list[0] model_info = model_instance.get_model_info() print(model_info)

参数说明

表 9-20 get_model_info 返回参数说明

参数	参数类型	描述
model_id	String	模型ID。
model_nam e	String	模型名称。
model_versi on	String	模型版本。

参数	参数类型	描述
tenant	String	租户。
project	String	项目。
owner	String	用户。
create_at	Long	模型创建时间,距'1970.1.1 0:0:0 UTC'的毫秒数。
source_locat ion	String	模型所在的OBS路径。
source_job_i d	String	来源训练作业的ID。
source_job_v ersion	String	来源训练作业的版本。
source_type	String	模型来源的类型。 • 当模型为自动学习部署过来时,取值为 "auto"。 • 当模型是用户通过训练作业或OBS模型文件部署 时,此值为空。
model_type	String	模型类型,取值为:TensorFlow/MXNet/ Spark_MLlib/Scikit_Learn/XGBoost/MindSpore/ Image/PyTorch。
model_size	Long	模型大小,单位为字节数。
model_statu s	String	模型状态,取值为:publishing/published/failed。
description	String	模型描述信息。
execution_c ode	String	执行代码存放的OBS地址,名称固定为 "customize_service.py"。
schema_doc	String	模型schema文档的下载地址。
image_addr ess	String	模型的执行镜像地址,镜像未构建之前(即当前模型从未发布成服务),显示为空。
input_param s	params结构数 组	模型的输入参数集,默认为空
output_para ms	params结构数 组	模型的输出参数集,默认为空
dependencie s	dependency结 构数组	运行代码及模型需安装的包。
model_metri cs	String	模型评测参数,仅当source_job_id和 source_job_version有值且对应的训练作业有评测 结果时会返回该结果。

参数	参数类型	描述
apis	String	模型所有的apis入参出参信息。

表 9-21 params 结构

参数	参数类型	描述
url	String	api代表的url路径。
param_nam e	String	参数名,不超过64个字符。
param_type	String	参数类型,取值为:int/string/float/timestamp/ date/file。
min	Number	当param_type为int或float时创建模型时,有配置 min则返回,默认为空。
max	Number	当param_type为int或float时创建模型时,有配置 max则返回,默认为空。
param_desc	String	参数描述,不超过100个字符,默认为空。

表 9-22 dependency 结构

参数	参数类型	描述
installer	String	安装器名称。
packages	package结构数 组	依赖包集合。

表 9-23 package 结构

参数	参数类型	描述	
package_na me	String	依赖包名称。	
package_ver sion	String	依赖包版本。	
restraint	String	版本过滤条件,取值为: • EXACT:等于给定版本 • ATLEAST:不小于给定版本 • ATMOST:不大于给定版本	

表 9-24 metric 参数说明

参数	是否 必选	参数类型	描述
f1	是	Double	平均数。
recall	是	Double	召回率。
precision	是	Double	精确率。
accuracy	是	Double	准确率。

9.6 删除模型

删除模型对象。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• 方式1:根据导入模型或模型调试生成的模型对象进行模型对象删除

from modelarts.session import Session from modelarts.model import Model session = Session() model_instance = Model(session, model_id="input your model_id") model_instance.delete_model()

• 方式2:根据**查询模型对象列表**返回的模型对象进行模型删除

from modelarts.session import Session
from modelarts.model import Model
session = Session()
model_object_list = Model.get_model_object_list(session)
model_instance = model_object_list[0]
model_instance.delete_model()

10 服务管理

10.1 服务管理概述

服务管理,包括将已创建成功的模型部署为在线服务或本地服务。可以实现在线预测、本地预测、服务详情查询、查看服务日志等功能。

这里的在线服务包括"predictor"和"transformer"两类,都包括下文描述的功能,本章节以"predictor"服务为例进行说明。

□ 说明

本章节的示例代码都是在ModelArts Notebook中实现的,若在其它开发环境使用,需要进行 Session鉴权,请参见**Session鉴权**。

10.2 服务调试

可以通过部署本地服务来进行调试,即在**导入模型或模型调试**后,在本地部署 Predictor进行本地推理,不再依赖线上资源。

推荐使用tensorflow_mlp_mnist_local_mode案例学习,单击如下链接直接进入 ModelArts使用。

Run in ModelArts

□ 说明

目前仅支持在Linux平台上部署本地服务Predictor,Windows平台不支持。推荐使用ModelArts Notebook部署本地服务。

● 本地服务Predictor和在线服务Predictor说明

- 部署本地服务Predictor,即将模型文件部署在本地,其环境规格取决于本地;例如在一个modelarts.vm.cpu.2u的Notebook中,部署本地Predictor,其运行环境就是cpu.2u。
- <mark>部署在线服务</mark>Predictor,即将存储在OBS中的模型文件部署到线上服务管理模块提供的容器中运行,其环境规格(如CPU规格,GPU规格)由**表3** predictor configs结构决定。

- 部署在线服务Predictor需要线上服务端根据AI引擎创建容器,较耗时;本地 Predictor部署较快,最长耗时10s,可用以测试模型,不建议进行模型的工业 应用。
- 当前版本支持部署本地服务Predictor的AI引擎为: "XGBoost"、 "Scikit_Learn"、"PyTorch"、"TensorFlow"和"Spark_MLlib"。具体版本 信息可参考**支持的常用引擎及其Runtime**。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

TensorFlow1.8本地推理示例代码

需要在环境中配置"tensorflow_model_server",可调用SDK接口快速配置,请参考如下示例代码。

- CPU环境,调用Model.configure_tf_infer_environ(device_type="CPU")完成配置,环境中只需配置运行一次。
- GPU环境,调用Model.configure_tf_infer_environ(device_type="GPU")完成配置,环境中只需配置运行一次。

```
from modelarts.session import Session
from modelarts.model import Model
from modelarts.config.model_config import ServiceConfig
session = Session()
# GPU环境推理配置
Model.configure_tf_infer_environ(device_type="GPU")
# CPU环境推理配置
#Model.configure_tf_infer_environ(device_type="CPU")
model_instance = Model(
            session,
            model name="input model name",
                                                   # 模型名称
            model_version="1.0.0",
                                              # 模型版本
            source location=model location,
                                                # 模型文件路径
            model_type="MXNet",
                                               #模型类型
            model_algorithm="image_classification", # 模型算法
            execution_code="OBS_PATH",
            input_params=input_params,
                                                #参考input_params格式描述
            output_params=output_params,
                                                  #参考output_params格式描述
            dependencies=dependencies,
                                                #参考dependencies格式描述
            apis=apis)
configs = [ServiceConfig(model_id=model.get_model_id(), weight="100", instance_count=1,
              specification="local")]
predictor_instance = model_instance.deploy_predictor(configs=configs)
predict_result = predictor_instance.predict(data=data_path, data_type=data_type) # 本地推理预测
print(predict_result)
```

参数说明

表 10-1 部署本地服务 predictor 参数说明

参数	是否 必选	参数类型	描述
service_na me	否	String	服务名称,支持1-64位可见字符(含中文),只能以英文大小写字母或者中文字符开头,名称可以包含字母、中文、数字、中划线、下划线。
configs	是	JSON Array	本地服务运行配置。

表 10-2 predictor configs 结构

参数	是否必 选	参数类型	描述
model_ id	睼	String	模型ID。"model_id"可以通过 <mark>查询模型列</mark> 表 或者ModelArts管理控制台获取。
weight	是	Integer	权重百分比,分配到此模型的流量权重,部署 本地服务Predictor时,取值100。
specific ation	是	String	部署本地服务时,取值为为"local"。
instanc e_coun t	是	Integer	模型部署的实例数,当前限制最大实例数为 5,部署本地服务Predictor时,取值为1。
envs	否	Map <string, String></string, 	运行模型需要的环境变量键值对,可选填,默 认为空。

表 10-3 部署本地服务 predictor 返回参数说明

参数	是否必 选	参数类型	描述
predicto r	是	Predictor对象	Predictor对象,其属性只包括 推理服务测 试。

10.3 部署在线服务

部署在线服务包括:

- 已部署为在线服务的初始化。
- 部署在线服务predictor。

部署批量服务transformer。

部署服务返回服务对象Predictor,其属性包括服务管理章节下的所有功能。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

● 方式1: 已部署为在线服务predictor的初始化

```
from modelarts.session import Session
from modelarts.model import Predictor
session = Session()
predictor_instance = Predictor(session, service_id="input your service_id")
```

• 方式2: 部署在线服务predictor

参数"model_id"代表将部署成在线服务的模型。"model_id"可以通过<mark>查询模</mark> <mark>型列表</mark>或者ModelArts管理控制台获取。

• 方式3: 部署批量服务transformer

• 关于部署在线服务predictor和批量服务transformer时,configs参数格式说明

- 部署在线服务predictor时:

SDK提供了ServiceConfig类对其定义,configs为list,list中的元组对象是 ServiceConfig.定义代码如下:

部署批量服务transformer时:

SDK提供了TransformerConfig类对其定义,configs都是list,list中的元组对象是TransformerConfig.定义代码如下:

参数说明

表 10-4 参数说明

参数	是否 必选	参数类 型	描述
service_id	是	String	服务ID,可从ModelArts前端在线服务中获取。
session	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。

表 10-5 部署在线服务 predictor 和 transformer 参数说明

参数	是否 必选	参数类 型	描述
service_na me	否	String	服务名称,支持1-64位可见字符(含中文),只能 以英文大小写字母或者中文字符开头,名称可以包 含字母、中文、数字、中划线、下划线。
descriptio n	否	String	服务备注,默认为空,不超过100个字符。
infer_type	否	String	推理方式,取值为real-time/batch/edge。默认为 real-time。 real-time代表在线服务,将模型部署为一个 Web Service,并且提供在线的测试UI与监控能力,服务一直保持运行。 batch为批量服务,批量服务可对批量数据进行推理,完成数据处理后自动停止。 edge表示边缘服务,通过华为云智能边缘平台,在边缘节点将模型部署为一个Web Service,需提前在IEF(智能边缘服务)创建好节点。

参数	是否 必选	参数类型	描述
vpc_id	否	String	在线服务实例部署的虚拟私有云ID,默认为空,此时ModelArts会为每个用户分配一个专属的VPC,用户之间隔离;如需要在服务实例中访问名下VPC内的其他服务组件,则可配置此参数为对应VPC的ID。
			VPC一旦配置,不支持修改。当vpc_id与cluster_id 一同配置时,只有专属集群参数生效。
subnet_ne twork_id	否	String	子网的网络ID,默认为空,当配置了vpc_id则此参数必填。需填写虚拟私有云控制台子网详情中显示的"网络ID"。通过子网可提供与其他网络隔离的、可以独享的网络资源。
security_g roup_id	否	String	安全组,默认为空,当配置了vpc_id则此参数必填。安全组起着虚拟防火墙的作用,为服务实例提供安全的网络访问控制策略。安全组须包含至少一条入方向规则,对协议为TCP、源地址为0.0.0.0/0、端口为8080的请求放行。
configs	是	包括 predicto r configs 结构和 transfor mer configs	模型运行配置。 • 当推理方式为batch/edge时仅支持配置一个模型。 • 当推理方式为real-time时,可根据业务需要配置多个模型并分配权重,但多个模型的版本号不能相同
schedule	否	schedul e结构数 组	服务调度配置,仅在线服务可配置,默认不使用, 服务长期运行。请参见 <mark>表10-9</mark> 。

表 10-6 predictor configs 结构

参数	是否必选	参数类型	描述
model_ id	是	String	模型ID。"model_id"可以通过 <mark>查询模型列表</mark> 或者 ModelArts管理控制台获取。

参数	是否必选	参数类型	描述
weight	是	Integer	权重百分比,分配到此模型的流量权重,仅当infer_type为real-time时需要配置,多个权重相加必须等于100;当在一个在线服务中同时配置了多个模型版本且设置不同的流量权重比例时,持续地访问此服务的预测接口,ModelArts会按此权重比例将预测请求转发到对应的模型版本实例。 { "service_name": "mnist", "description": "mnist service", "infer_type": "real-time", "config": [{ "model_id": "xxxmodel-idxxx", "weight": "70", "specification": "modelarts.vm.cpu.2u", "instance_count": 1, "envs": { "model_name": "mxnet-model-1", "load_epoch": "0" } }, { "model_id": "xxxxxxx", "weight": "30", "specification": "modelarts.vm.cpu.2u", "instance_count": 1 }] }
specific ation	是	String	资源规格,当前版本可选modelarts.vm.cpu.2u/modelarts.vm.gpu.p4(需申请)/modelarts.vm.ai1.a310(需申请),需申请权限才能使用的规格请在华为云创建工单,由ModelArts运维工程师添加权限。当取值为"local"时,即为在本地推理,请参见部署本地Predictor说明。
instanc e_coun t	是	Integer	模型部署的实例数,当前限制最大实例数为5,如需使用更多的实例数,需提交工单申请。
envs	否	Map <stri ng, String></stri 	运行模型需要的环境变量键值对,可选填,默认为 空。

表 10-7 transformer configs 结构

参数	是否 必选	参数类型	描述	
model_i d	是	String	模型ID。	
specific ation	是	String	资源规格,当前版本可选modelarts.vm.cpu.2u/ modelarts.vm.gpu.p4。	
instanc e_count	是	Integer	模型部署的实例数,邀测阶段取值范围[1, 2]。	
envs	否	Map <stri ng, String></stri 	运行模型需要的环境变量键值对,可选填,默认为 空。	
src_pat h	是	String	批量任务输入数据的OBS路径。	
dest_pa th	是	String	批量任务输出结果的OBS路径。	
req_uri	是	String	批量任务中调用的推理接口,即模型镜像中暴露的 REST接口,需要从模型的config.json文件中选取一 个api路径用于此次推理;如使用ModelArts提供的 预置推理镜像,则此接口为/	

参数	是否必选	参数类型	描述
mappin	是	String	输入数据的映射类型,可选"file"或"csv"。
g_type			file指每个推理请求对应到输入数据目录下的一个 文件,当使用此方式时,此模型对应req_uri只能 有一个输入参数且此参数的类型是file。
			• csv指每个推理请求对应到csv里的一行数据,当使用此方式时,输入数据目录下的文件只能以.csv 为后缀,且需配置mapping_rule参数,以表达推 理请求体中各个参数对应到csv的索引。
			创建批量服务且输入数据映射方式为file的样例
			"service_name": "batchservicetest", "description": "", "infer_type": "batch",
			"config": [{ "model_id": "598b913a-af3e-41ba-a1b5-bf065320f1e2", "specification": "modelarts.vm.cpu.2u",
			"instance_count": 1, "src_path": "https://infers-data.obs.myhuaweicloud.com/ xgboosterdata/",
			"dest_path": "https://infers-data.obs.myhuaweicloud.com/ output/", "req_uri": "/",
			"mapping_type": "file" }] }
			创建批量服务且输入数据映射方式为csv的样例 { "service_name": "batchservicetest",
			"description": "", "infer_type": "batch",
			"config": [{ "model_id": "598b913a-af3e-41ba-a1b5-bf065320f1e2", "specification": "modelarts.vm.cpu.2u", ""
			"instance_count": 1, "src_path": "https://infers-data.obs.myhuaweicloud.com/ xgboosterdata/",
			"dest_path": "https://infers-data.obs.myhuaweicloud.com/ output/", "req_uri": "/",
			"mapping_type": "csv", "mapping_rule": { "type": "object",
			"properties": { "data": { "type": "object",
			"properties": { "req_data": { "type": "array",
			"items": [{ "type": "object", "properties": {
			"input5": { "type": "number", "index": 0
			}, "input4": { "type": "number", "index": 1
			index : i
			type . Hullibel ,

参数	是否 必选	参数类型	描述
			"index": 2 }, "input2": { "type": "number", "index": 3 }, "input1": { "type": "number", "index": 4 } } } } } }
mappin g_rule	否	Мар	输入参数与csv数据的映射关系,仅当mapping_type为csv时需要填写。映射规则与模型配置文件config.json中输入参数的定义方式相似,只需要在每一个基本类型(string/number/integer/boolean)的参数下配置index参数,指定使用csv数据中对应索引下标的数据作为此参数的值去发送推理请求,csv数据必须以英文半角逗号分隔,index从0开始计数,特殊地,当index为-1时忽略此参数,具体请参见部署transformer的示例代码的样例。 样例中mapping_rule描述的推理请求体格式为: { "req_data": { "req_data": {{ "req_data": {{ "input1": 1, "input2": 2, "input3": 3, "input4": 4, "input5": 5 }} }

表 10-8 部署 predictor 和 transformer 返回参数说明

参数	是否必选	参数类型	描述
predicto r	是	Predictor 对象	Predictor对象,其属性描述包括服务管理章节全部功能。

表 10-9 schedule 结构

参数	是否必选	参数类型	说明
op_type	是	String	调度类型,当前仅支持取值为"stop"。
time_unit	是	String	调度时间单位,可选: • DAYS • HOURS • MINUTES
duration	是	Integer	对应时间单位的数值,比如2小时后停止, 则 "time_unit"填"HOURS", "duration"填"2"。

□ 说明

• 给出MXNet实现手写数字识别项目中部署在线predictror实例:

给出MXNet实现手写数字识别项目中部署transformer实例(批推理):

```
from modelarts.session import Session
from modelarts.model import Model
from modelarts.config.model_config import ServiceConfig,TransformerConfig
model_instance = Model(session, model_id = "input your model id")
configs = []
config1 = TransformerConfig(model_id="input your model id",
                  specification="modelarts.vm.cpu.2u",
                  instance_count=1,
envs={"input_data_name":"images","input_data_shape":"0,1,28,28","output_data_shape":"0,10"},
                  src_path="/w0403/testdigitrecognition/inferimages/",
                  dest_path="/w0403/testdigitrecognition/",
                  req_uri = "/"
                  mapping_type = "file")
configs.append(config1)
predictor = model_instance.deploy_transformer(service_name="DigitRecognition",
infer_type="batch", configs=configs)
```

10.4 查询服务详情

查询当前服务对象的详细信息。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• 方式1:根据部署在线服务生成的服务对象进行服务详情查询

from modelarts.session import Session from modelarts.model import Predictor session = Session() predictor_instance = Predictor(session, service_id="input your service_id") predictor_info = predictor_instance.get_service_info() print(predictor_info)

• 方式2:根据查询服务对象列表返回的服务对象进行服务详情查询

from modelarts.session import Session from modelarts.model import Predictor session = Session() predictor_object_list = Predictor.get_service_object_list(session) predictor_instance = predictor_object_list[0] predictor_info = predictor_instance.get_service_info() print(predictor_info)

参数说明

表 10-10 get_service_info 返回参数说明

参数	参数类型	描述
service_id	String	服务ID。
service_name	String	服务名称。
description	String	服务描述。
tenant	String	服务归属租户。
project	String	服务归属项目。
owner	String	服务归属用户。
publish_at	Number	服务最新的发布时间,距'1970.1.1 0:0:0 UTC'的毫秒数。
infer_type	String	推理方式,取值为real-time/batch/ edge。
vpc_id	String	服务实例所在的虚拟私有云ID,服务自定 义网络配置时返回。
subnet_netwo rk_id	String	服务实例所在的子网的网络ID,服务自定 义网络配置时返回。
security_grou p_id	String	服务实例所在的安全组,服务自定义网络配置时返回。
status	String	服务状态,取值为: running/deploying/ concerning/failed/stopped/finished。
error_msg	String	错误信息,当status为failed时,返回注 明部署失败原因。

参数	参数类型	描述
config	不同infer_type的 config结构数组	不同infer_type的config结构数组 服务配置(如果是共享过来的服务,只返 回model_id ,model_name, model_version)
access_addres s	String	推理请求的访问地址,当infer_type为 real-time时会返回此值
invocation_ti mes	Number	服务的总调用次数。
failed_times	Number	服务调用失败次数。
is_shared	Boolean	是否是订阅的服务。
shared_count	Number	订阅的服务数。
progress	Integer	部署进度,当状态是deploying时,返 回。

表 10-11 real-time config 结构

参数	参数类型	描述
model_id	String	模型ID。 "model_id"可以通过 <mark>查询模型列表</mark> 或者ModelArts管理控制台获取。
model_name	String	模型名称。
model_versio n	String	模型版本。
source_type	String	模型来源,当模型是由自动学习产生时,返回 此字段,取值为:auto。
status	String	模型实例运行状态,取值为:
		• ready:已就绪(所有实例已启动)
		• concerning: 部分就绪(部分实例已启动、 部分实例未启动)
		• notReady:未就绪(所有实例都没启动)
weight	Integer	权重,分配到此模型的流量权重。
specification	String	资源规格,取值为:modelarts.vm.cpu.2u/ modelarts.vm.gpu.p4/ modelarts.vm.ai1.a310。
envs	Map <string, String></string, 	运行模型需要的环境变量键值对。
instance_cou nt	Integer	模型部署的实例数。

参数	参数类型	描述
scaling	Boolean	是否启用弹性伸缩。

表 10-12 batch config 结构

参数	参数类型	描述
model_id String		模型ID。"model_id"可以通过 <mark>查询模型</mark> 列表 或者ModelArts管理控制台获取。
model_name	String	模型名称。
model_versio String		模型版本。
specification	String	资源规格,取值为:modelarts.vm.cpu. 2u/modelarts.vm.gpu.p4。
envs	Map <string, string=""></string,>	运行模型需要的环境变量键值对。
instance_cou nt	Integer	模型部署的实例数。
src_path	String	批量任务输入数据的OBS路径。
dest_path	String	批量任务输出结果的OBS路径。
req_uri	String	批量任务中调用的推理路径。
mapping_typ e	String	输入数据的映射类型,取值为: file或 csv。
mapping_rule	Мар	输入参数与csv数据的映射关系,仅当 mapping_type为csv时,会返回。

10.5 推理服务测试

推理服务在线测试支持文件、图片、json三种格式。通过部署为在线服务Predictor可 以完成在线推理预测。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session 鉴权请参见Session鉴权。

场景: 部署在线服务Predictor的推理预测 from modelarts.session import Session from modelarts.model import Predictor session = Session() predictor_instance = Predictor(session, service_id="input your service_id") predict_result = predictor_instance.predict(data=data_path, data_type=data_type)

参数说明

表 10-13 参数说明

参数	是否必 选	参数类型	描述	
data_ty pe	是	String	当前支持三种格式:files、images、json, 即文 本、图片、json格式。	
data	是	String	 针对files、images类型的数据, 该参数为其本地路径,如: data = "/home/ma-user/work/test.jpg" 针对json类型的数据, 该参数可以是其本地路径,如: data = "/home/ma-user/work/test.json" 同时也可以为 "dict" 类型的变量,如: data = { "is_training": "False", "observations": [[1,2,3,4]], "default_policy/eps:0": "0.0" } 	
path	否	String	服务内的推理路径,默认为"/"。	

表 10-14 predict 返回参数说明

参数	描述
返回消息体	输出的参数和值,平台只做转发,不做识别。

10.6 查询服务列表

获取当前用户服务列表。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• 场景1: 查询当前用户所有服务

from modelarts.session import Session from modelarts.model import Predictor session = Session() predictor_list = Predictor.get_service_list(session) print(predictor_list)

• 场景2: 按照检索条件查询当前用户服务

from modelarts.session import Session from modelarts.model import Predictor session = Session() predictor_list = Predictor.get_service_list(session, service_name="digit", order="asc", offset="0", infer_type="real-time") print(predictor_list)

参数说明

表 10-15 查询检索参数说明

参数	是否必 选	参数类型	描述
session	是	Object	会话对象,初始化方法见 <mark>Session鉴权</mark> 。
service_id	否	String	服务ID,默认不过滤服务ID。
service_name	否	String	服务名称,默认不过滤服务名。
infer_type	否	String	推理方式,取值为:real-time/batch/ edge,默认不过滤推理方式。
offset	否	Integer	分页列表的起始页,默认为: "0"。
limit	否	Integer	指定每一页返回的最大条目数,默认为: "1000"。
service_statu s	否	String	服务状态,默认不过滤服务状态。可根据服务状态查询,取值如下。 • running: 运行中,服务正常运行,正在计费。 • deploying: 部署中,服务正在部署,调度资源部署等。 • concerning: 告警,后端实例异常,可能正在计费。例如多实例的情况下,有的实例正常,有的实例异常。正常的实例会产生费用,此时服务状态是concerning。 • failed: 失败,服务部署失败,失败原因可以查看事件和日志标签。 • stopped: 停止。 • finished: 只有批量服务会有这个状态,表示运行完成。
sort_by	否	String	指定排序字段,可选"publish_at"、 "service_name",默认可选 "publish_at"。
order	否	String	排序方式,可选"asc"或"desc",代表 递增排序及递减排序,默认为:"desc"。
model_id	否	String	模型ID,默认不过滤模型ID。

表 10-16 get_service_list 返回参数说明

参数	参数类型	描述
total_count	Integer	不分页的情况下,符合查询条件的总服务数量。
count	Integer	当前查询结果的服务数量,不设置offset、limit 查询参数时,count与total相同。
services	service结构数 组	查询到的服务集合。

表 10-17 service 结构

参数	参数类型	描述
service_id	String	服务ID。
service_name	String	服务名称。
description	String	服务描述。
tenant	String	服务归属租户。
project	String	服务归属项目。
owner	String	服务归属用户。
publish_at	Number	服务最新的发布时间,距'1970.1.1 0:0:0 UTC'的 毫秒数。
infer_type	String	推理方式,取值为:real-time/batch/edge。
status	String	服务状态,取值为:running/deploying/ concerning/failed/stopped/finished。
progress	Integer	部署进度,当状态是deploying时,返回。
invocation_tim es	Number	服务的总调用次数。
failed_times	Number	服务调用失败次数。
is_shared	Boolean	是否是订阅的服务。
shared_count	Number	订阅的服务数。

10.7 查询服务对象列表

获取当前用户服务对象列表。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

场景1:查询当前用户所有服务对象

from modelarts.session import Session
from modelarts.model import Predictor
session = Session()
predictor_list_object_resp = Predictor.get_service_object_list(session)
print(predictor_list_object_resp)

● 场景2:按照检索条件查询当前用户服务对象

from modelarts.session import Session from modelarts.model import Predictor session = Session() predictor_object_list = Predictor.get_service_object_list(session, service_name="digit", order="asc", offset="0", infer_type="real-time") print(predictor_object_list)

参数说明

● 查询服务列表,返回list,list大小等于当前用户所有已经部署的服务个数,list中每个元素都是Predictor对象,对象属性同本章初始化服务。

查询服务列表返回说明: service_list_resp = [service_instance1, service_instance2, service_instance3 ...],列表中元素 "service_instance"对象即为服务管理章节描述的可调用服务接口。

- 支持按照检索参数查询服务列表,返回满足检索条件的服务list,检索参数如表 10-18所示。
- 在查询列表时,返回list的同时,默认会打印模型列表的详细信息,如表10-19和表10-20所示。

表 10-18 查询检索参数说明

参数	是否必 选	参数类型	描述
session	是	Object	会话对象,初始化方法见 Session<mark>鉴权</mark>。
is_show	否	Boolean	是否打印出服务对象信息,默认为 "True"。
service_id	否	String	服务ID,默认不过滤服务ID。
service_name	否	String	服务名称,默认不过滤服务名。
infer_type	否	String	推理方式,取值为:real-time/batch/edge, 默认不过滤推理方式。
offset	否	Integer	分页列表的起始页,默认为"0"。
limit	否	Integer	指定每一页返回的最大条目数,默认为 "1000"。
sort_by	否	String	指定排序字段,可选"publish_at"、 "service_name",默认可选 "publish_at"。
order	否	String	排序方式,可选"asc"或"desc",代表递增排序及递减排序,默认为:"desc"。
model_id	否	String	模型ID,默认不过滤模型ID。

表 10-19 get_service_list 返回参数说明

参数	参数类型	描述
total_count	Integer	不分页的情况下,符合查询条件的总服务数 量。
count	Integer	当前查询结果的服务数量,不设置offset、 limit查询参数时,count与total相同。
services	service结构数组	查询到的服务集合。

表 10-20 service 结构

参数	参数类型	描述
service_id	String	服务ID。
service_name	String	服务名称。
description	String	服务描述。
tenant	String	服务归属租户。
project	String	服务归属项目。
owner	String	服务归属用户。
publish_at	Number	服务最新的发布时间,距'1970.1.1 0:0:0 UTC'的毫秒数。
infer_type	String	推理方式,取值为:real-time/batch/ edge。
status	String	服务状态,取值为: running/deploying/ concerning/failed/stopped/finished。
progress	Integer	部署进度,当状态是deploying时,返回。
invocation_tim es	Number	服务的总调用次数。
failed_times	Number	服务调用失败次数。
is_shared	Boolean	是否是订阅的服务。
shared_count	Number	订阅的服务数。

10.8 更新服务配置

更新当前服务对象配置。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数;其它平台的Session鉴权请参考Session鉴权。

• 方式1:根据部署在线服务生成的服务对象进行更新服务配置

方式2:根据查询服务对象列表返回的服务对象进行更新服务配置

参数说明

表 10-21 部署 predictor 参数说明

参数	是否 必选	参数类型	描述
descri ption	否	String	服务描述,不超过100个字符,不设置此参数时,表示不更新。
status	否	String	服务状态,可设置状态为running或stopped来启动、 停止服务,不设置此参数则不修改状态。status不可 跟configs同时修改,同时存在则只修改status。
config s	否	包括 predictor configs结构 和 transforme r configs	服务配置,不设置此参数时,表示不更新。关于 configs如何生成,请参见 <mark>部署在线服务</mark> 。

表 10-22 predictor configs 结构

参数	是否 必选	参数类型	描述
model_i d	是	String	模型ID。"model_id"可以通过 <mark>查询模型列表</mark> 或 者ModelArts管理控制台获取。
weight	是	Integer	权重百分比,分配到此模型的流量权重,仅当 infer_type为real-time时需要配置,多个权重相加 必须等于100;当在一个在线服务中同时配置了多 个模型版本且设置不同的流量权重比例时,持续 地访问此服务的预测接口,ModelArts会按此权重 比例将预测请求转发到对应的模型版本实例。
specifica tion	是	String	资源规格,当前版本可选modelarts.vm.cpu.2u/modelarts.vm.gpu.p4(需申请)/modelarts.vm.ai1.a310(需申请),需申请权限才能使用的规格请在华为云创建工单,由ModelArts运维工程师添加权限。
instance _count	是	Integer	模型部署的实例数,当前限制最大实例数为5,如 需使用更多的实例数,需提交工单申请。
envs	否	Map <strin g, String></strin 	运行模型需要的环境变量键值对,可选填,默认 为空。

表 10-23 transformer configs 结构

参数	是否必选	参数类型	描述
model_i d	是	String	模型ID。"model_id"可以通过 查询模型列表 或 者ModelArts管理控制台获取。
specific ation	是	String	资源规格,当前版本可选modelarts.vm.cpu.2u/ modelarts.vm.gpu.p4。
instanc e_count	是	Integer	模型部署的实例数,当前限制最大实例数为5,如 需使用更多的实例数,需提交工单申请。
envs	否	Map <strin g, String></strin 	运行模型需要的环境变量键值对,可选填,默认 为空。
src_pat h	是	String	批量任务输入数据的OBS路径。
dest_pa th	是	String	批量任务输出结果的OBS路径。
req_uri	是	String	批量任务中调用的推理接口,需要从模型的 config.json文件中选取一个api路径用于此次推理

参数	是否必 选	参数类型	描述
mappin g_type	是	String	输入数据的映射类型,可选"file"或"csv"。 file指每个推理请求对应到输入数据目录下的一个文件,当使用此方式时,此模型对应req_uri只能有一个输入参数且此参数的类型是file。 csv指每个推理请求对应到csv里的一行数据,当使用此方式时,输入数据目录下的文件只能以.csv为后缀,且需配置mapping_rule参数,以表达推理请求体中各个参数对应到csv的索引。
mappin g_rule	否	Мар	输入参数与csv数据的映射关系,仅当mapping_type为csv时需要填写。映射规则与模型配置文件config.json中输入参数的定义方式相似,只需要在每一个基本类型(string/number/integer/boolean)的参数下配置index参数,指定使用csv数据中对应索引下标的数据作为此参数的值去发送推理请求,csv数据必须以英文半角逗号分隔,index从0开始计数,特殊地,当index为-1时忽略此参数。

表 10-24 update_service_config 返回参数说明

参数	是否必选	参数类型	描述
error_co de	是	String	调用失败时,的错误码。 调用成功时,无此字段。
error_m sg	是	String	调用失败时,错误信息。 调用成功时,无此字段。

10.9 查询服务监控信息

查询当前服务对象监控信息。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• 方式1:根据部署在线服务生成的服务对象进行查询服务监控

from modelarts.session import Session from modelarts.model import Predictor session = Session() predictor_instance = Predictor(session, service_id="input your service_id") predictor_monitor = predictor_instance.get_service_monitor() print(predictor_monitor) • 方式2:根据查询服务对象列表返回的服务对象进行查询服务监控

from modelarts.session import Session from modelarts.model import Predictor session = Session() predictor_object_list = Predictor.get_service_object_list(session) predictor_instance = predictor_object_list[0] predictor_monitor = predictor_instance.get_service_monitor() print(predictor_monitor)

参数说明

表 10-25 get_service_monitor 返回参数说明

参数	参数类型	描述
service_id	String	服务ID。
service_name	String	服务名称。
monitors	服务的infer_type对应的 monitor结构数组	监控信息详情。

表 10-26 real-time monitor 结构

参数	参数类型	描述
model_id	String	模型ID。
model_name	String	模型名称。
model_versio n	String	模型版本。
invocation_ti mes	Number	模型实例的总调用次数。
failed_times	Number	模型实例调用失败次数。
cpu_core_usa ge	Float	已使用CPU核数。
cpu_core_tot al	Float	总CPU核数。
cpu_memory _usage	Integer	已使用内存,单位MB。
cpu_memory _total	Integer	总内存,单位MB。
gpu_usage	Float	已使用GPU个数。
gpu_total	Float	总GPU个数。

表 10-27 edge monitor 结构

参数	参数类型	描述
node_id	String	待查询的边缘节点ID,仅当infer_type为edge时 存在。
node_name	String	节点名称。
cpu_core_usa ge	Float	已使用CPU核数。
cpu_core_tot al	Float	总CPU核数。
cpu_memory _usage	Integer	已使用内存,单位MB。
cpu_memory _total	Integer	总内存,单位MB。
gpu_usage	Float	已使用GPU个数。
gpu_total	Float	总GPU个数。

10.10 查询服务日志

查询当前服务对象的日志信息。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• 方式1:根据<mark>部署在线服务</mark>生成的服务对象进行查询服务日志

from modelarts.session import Session from modelarts.model import Predictor session = Session() predictor_instance = Predictor(session, service_id="input your service_id") predictor_log = predictor_instance.get_service_logs() print(predictor_log)

• 方式2: 根据**查询服务对象列表**返回的服务对象进行查询服务日志

from modelarts.session import Session from modelarts.model import Predictor session = Session() predictor_object_list = Predictor.get_service_object_list(session) predictor_instance = predictor_object_list[0] predictor_log = predictor_instance.get_service_logs() print(predictor_log)

参数说明

表 10-28 get_service_logs 返回参数说明

参数	参数类型	描述
service_id	String	服务ID。
service_name	String	服务名称。
logs	log结构数组	服务的更新记录。

表 10-29 log 结构

参数	参数类型	描述
update_time	Long	更新时间,距'1970.1.1 0:0:0 UTC'的毫秒数。
result	String	更新结果,取值为:SUCCESS/FAIL/ RUNNING。
config	Config结构数组	更新后的服务配置,当infer_type为real- time时,会返回此值。
success_num	Number	操作成功的节点数,当infer_type为edge 时,会返回此值。
failed_num	Number	操作失败的节点数,当infer_type为edge 时,会返回此值。
result_detail	Result结构数组	操作结果详情,当infer_type为edge时, 会返回此值。

表 10-30 config 结构

参数	参数类型	描述
model_id	String	模型ID。
model_name	String	模型名称。
model_version	String	模型版本。
weight	Integer	权重,分配到此模型的流量权重。
specification	String	资源规格。
instance_count	Integer	模型部署的实例数。
envs	Map <string, String></string, 	运行模型需要的环境变量键值对。

表 10-31 result 结构

参数	参数类型	描述
node_id	String	待查询的边缘节点ID,仅当infer_type为 edge时存在。
node_name	String	边缘节点名称。
operation	String	操作类型,取值deploy/delete。
result	Boolean	操作结果,true代表成功,false表示操 作失败。

10.11 删除服务

删除服务存在如下两种删除方式。

- 根据**部署在线服务**或**服务调试**生成的服务对象删除服务。
- 根据查询服务对象列表返回的服务对象删除服务。

示例代码

在ModelArts notebook平台中,Session鉴权无需输入鉴权参数。其它平台的Session鉴权请参见Session鉴权。

• 方式1:根据部署在线服务或服务调试生成的服务对象删除服务

from modelarts.session import Session from modelarts.model import Predictor session = Session() predictor_instance = Predictor(session, service_id="input your service_id") predictor_instance.delete_service()

• 方式2:根据查询服务对象列表返回的服务对象删除服务

from modelarts.session import Session
from modelarts.model import Predictor
session = Session()
predictor_object_list = Predictor.get_service_object_list(session)
predictor_instance = predictor_object_list[0]
predictor_instance.delete_service()

1 1 修订记录

发布日期	修订记录
2022-03-29	新增训练管理(推荐)的SDK参考: 训练管理(推荐)
2021-11-18	优化内容,(可选)安装ModelArts SDK 。
2021-10-18	内容优化,独立章节 获取访问密钥 。
2021-08-30	新增Manifest管理的SDK参考: Manifest管理
2021-07-30	新增数据管理SDK参考: 数据管理
2021-01-15	新增SDK参考: • 训练作业调试 • 模型调试 • 服务调试
2020-12-14	 下线开发环境相关接口。 下线老版OBS管理的相关接口,建议使用新版的OBS管理接口。 针对config.json认证模式,即将下线,更改相应描述。
2020-04-10	新增: OBS管理: 新增OBS相关操作指导功能。 服务调试: 支持TensorFlow本地推理。
2019-08-13	模型管理的dependency结构中不再支持conda安装器。更新如下接口:

发布日期	修订记录
2019-08-08	新增:
	• 训练管理: 新增功能本地训练功能。
	模型管理: 新增获取模型对象列表功能。
	服务管理:新增部署本地服务和本地推理功能。
	● OBS管理: 新增获取OBS Client功能。
	刷新:
	● Session鉴权 :优化Session鉴权内容。
2019-04-23	新增V1.1.1版本SDK,刷新 SDK简介 和(可选)安装ModelArts SDK 内容,其他操作与V1.1.0一致。
2019-03-20	新增v1.1.0版本的SDK,主要包含Session鉴权、OBS操作、作业管理、模型管理、服务管理 。
2019-02-21	新增在线服务模型推理。
2018-11-21	优化和修改训练作业部分描述。
2018-11-06	第一次正式发布。