ModelArts

AI 工程师用户指南

文档版本 01

发布日期 2022-06-16

华为技术有限公司

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

HUAWEI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。

目 录

1 AI 工程师如何使用 ModelArts.....1
2 免费体验 ModelArts.....4
2.1 免费体验自动学习.....4
2.2 免费体验 AI 全流程开发.....7
3 数据管理（New）.....10
3.1 数据管理简介.....10
3.2 创建数据集.....12
3.3 创建标注作业.....18
3.4 标注数据.....24
3.4.1 图像分类.....24
3.4.2 物体检测.....29
3.4.3 图像分割.....34
3.4.4 文本分类.....40
3.4.5 命名实体.....43
3.4.6 文本三元组.....46
3.4.7 声音分类.....50
3.4.8 语音内容.....53
3.4.9 语音分割.....55
3.4.10 视频标注.....57
3.5 导入数据.....59
3.5.1 导入操作简介.....59
3.5.2 OBS 目录导入数据规范说明.....67
3.5.3 Manifest 文件导入规范说明.....72
3.6 导出数据.....87
3.7 修改数据集.....89
3.8 发布数据集.....90
3.9 删除数据集.....92
3.10 管理数据集版本.....93
3.11 智能标注.....94
3.12 难例确认.....97
3.13 自动分组.....99
3.14 数据特征分析.....101

3.15 团队标注 106
3.15.1 团队标注简介 106
3.15.2 管理团队 108
3.15.3 管理成员 109
3.15.4 管理团队标注任务 111
3.16 数据处理 119
3.16.1 数据处理简介 119
3.16.2 创建数据处理任务 119
3.16.3 管理和查看数据处理任务 121
3.16.4 预置算子说明 122
3.16.4.1 数据校验 122
3.16.4.2 数据清洗 125
3.16.4.3 数据选择 128
3.16.4.4 数据选择（难例） 131
3.16.4.5 数据增强（数据扩增） 136
3.16.4.6 数据增强（图像生成） 140
4 数据管理（旧版即将下线） 144
4.1 数据管理简介 144
4.2 创建数据集（旧版） 147
4.3 标注数据 159
4.3.1 图像分类 159
4.3.2 物体检测 164
4.3.3 图像分割 170
4.3.4 文本分类 175
4.3.5 命名实体 179
4.3.6 文本三元组 182
4.3.7 声音分类 186
4.3.8 语音内容 188
4.3.9 语音分割 190
4.3.10 视频标注 192
4.4 导入数据 194
4.4.1 导入操作 194
4.4.2 从 OBS 目录导入的规范说明 197
4.4.3 导入 Manifest 文件的规范说明 202
4.5 导出数据 217
4.6 修改数据集 220
4.7 发布数据集 221
4.8 删除数据集 224
4.9 管理数据集版本 224
4.10 智能标注 226
4.11 难例确认 229
4.12 自动分组 231

4.13 数据特征 233
4.14 团队标注 238
4.14.1 团队标注简介 238
4.14.2 管理团队 240
4.14.3 管理成员 241
4.14.4 管理团队标注任务 243
4.15 数据处理 249
4.15.1 数据处理简介 249
4.15.2 创建数据处理任务 249
4.15.3 管理和查看数据处理任务 251
4.15.4 预置算子说明 252
4.15.4.1 数据校验 252
4.15.4.2 数据清洗 255
4.15.4.3 数据选择 258
4.15.4.4 数据选择（难例） 261
4.15.4.5 数据增强（数据扩增） 266
4.15.4.6 数据增强（图像生成） 270
5 开发环境（New） 274
5.1 开发环境简介 274
5.2 开发环境使用场景 276
5.3 CodeLab：免费体验 Notebook 277
5.4 管理 Notebook 实例 282
5.4.1 创建 Notebook 实例 282
5.4.2 打开 Notebook 实例 288
5.4.3 启动/停止/删除实例 288
5.4.4 变更 Notebook 实例运行规格 289
5.4.5 变更 Notebook 实例镜像 290
5.4.6 开发环境中如何选择存储 290
5.4.7 动态挂载 OBS 并行文件系统 292
5.4.8 动态扩充云硬盘 EVS 容量 294
5.4.9 修改 Notebook 远程连接 IP 地址 296
5.5 使用 JupyterLab 开发模型 296
5.5.1 JupyterLab 操作流程 297
5.5.2 JupyterLab 简介及常用操作 297
5.5.3 上传文件至 JupyterLab 306
5.5.3.1 场景介绍 306
5.5.3.2 上传本地文件至 JupyterLab 307
5.5.3.2.1 上传场景和入口介绍 307
5.5.3.2.2 上传本地小文件（100MB 以内）至 JupyterLab 308
5.5.3.2.3 上传本地大文件（100MB~5GB）至 JupyterLab 310
5.5.3.2.4 上传本地超大文件（5GB 以上）至 JupyterLab 312
5.5.3.3 GitHub 开源仓库 Clone 314

5.5.3.4 上传 OBS 文件到 JupyterLab....317
5.5.3.5 上传远端文件至 JupyterLab....321
5.5.4 从 JupyterLab 下载文件至本地....322
5.5.5 JupyterLab 插件....324
5.5.5.1 代码参数化插件....324
5.5.6 使用 ModelArts SDK....326
5.5.7 MindInsight 可视化作业....327
5.5.8 TensorBoard 可视化作业....333
5.6 使用本地 IDE 开发模型....339
5.6.1 本地 IDE 操作流程....339
5.6.2 配置本地 IDE（PyCharm ToolKit 连接）....340
5.6.3 配置本地 IDE（PyCharm 手动连接）....346
5.6.4 配置本地 IDE（VS Code ToolKit 连接）....352
5.6.5 配置本地 IDE（VS Code 手动连接）....361
5.6.6 配置本地 IDE（SSH 工具连接）....371
5.6.7 在本地 IDE 中上传数据至 Notebook....377
5.6.8 下载 Notebook 中的文件至本地....378
5.7 使用 MindStudio 开发 Ascend 算子....379
5.8 使用可视化工具 ML Studio 开发模型....384
5.9 算法开发套件....384
5.9.1 算法开发套件简介....384
5.9.2 准备开发环境....385
5.9.3 创建算法工程....386
5.9.4 使用算法套件快速完成水表读数识别....390
5.9.5 使用样例....399
5.9.5.1 使用现有模型进行推理....399
5.9.5.2 使用现有数据集测试现有模型....401
5.9.5.3 使用现有数据集训练预训练模型....402
5.9.5.4 使用自定义数据集进行训练....404
5.9.6 算法开发套件命令说明....409
5.9.6.1 列举....409
5.9.6.2 安装....409
5.9.6.3 拷贝....410
5.9.6.4 运行....410
5.9.6.5 转换....411
5.9.6.6 导出....411
5.9.6.7 部署....411
5.9.6.7.1 部署脚本参考....411
5.9.6.7.2 调用部署服务进行推理....417
5.9.6.7.3 删除已发布模型和在线服务....418
5.9.6.8 编排....418
5.9.6.9 查询....418

5.9.7 配置参数说明 419
5.9.7.1 全局参数 419
5.9.7.2 Runner 420
5.9.7.3 Converter 421
5.9.7.4 Exporter 422
5.9.7.5 Deployer 423
5.9.7.6 Adapter 423
5.9.7.7 Flow 424
6 开发环境（旧版即将下线） 425
6.1 Notebook 简介 425
6.2 管理 Notebook 实例 427
6.2.1 创建 Notebook 实例 427
6.2.2 创建带有 Git 存储库的 Notebook 实例 430
6.2.3 打开 Notebook 实例 435
6.2.4 启动或停止 Notebook 实例 436
6.2.5 删除 Notebook 实例 436
6.3 使用 Jupyter Notebook 436
6.3.1 Jupyter Notebook 简介 436
6.3.2 Jupyter Notebook 常用操作 437
6.3.3 Jupyter Notebook 环境配置 440
6.3.3.1 使用 Notebook Terminal 功能 440
6.3.3.2 GPU 类型实例，在 Terminal 中切换 cuda 版本 441
6.3.3.3 安装外部库和内核 442
6.3.4 数据的上传和下载 443
6.3.4.1 上传大文件至 Notebook 中 443
6.3.4.2 下载 Notebook 内的大文件到本地 445
6.3.5 使用 ModelArts 示例 445
6.3.6 使用 ModelArts SDK 447
6.3.7 与 OBS 同步文件 448
6.3.8 使用 Convert to Python File 功能 448
6.3.9 远程 GPU 运行（Notebook2.0） 449
6.4 使用 JupyterLab 450
6.4.1 JupyterLab 简介及常用操作 450
6.4.2 数据的上传和下载 456
6.4.2.1 数据上传至 JupyterLab 456
6.4.2.2 从 JupyterLab 下载至本地 458
6.4.3 使用 ModelArts 示例 459
6.4.4 使用 ModelArts SDK 460
6.4.5 与 OBS 同步文件 460
6.4.6 使用 Git 插件 461
6.4.7 设置初始化脚本 465
7 训练管理（New） 469

7.1 模型训练简介 469
7.2 数据准备 469
7.3 选择算法 471
7.3.1 选择算法简介 471
7.3.2 使用订阅算法 472
7.3.3 使用自定义脚本 475
7.3.3.1 使用自定义脚本简介 475
7.3.3.2 开发自定义脚本 477
7.3.3.3 创建算法 481
7.3.4 使用自定义镜像 487
7.3.5 发布算法到 AI Gallery 488
7.4 完成一次简单训练 488
7.4.1 创建训练作业 488
7.4.2 查看作业详情 492
7.4.2.1 训练作业详情 493
7.4.2.2 训练作业事件 494
7.4.2.3 训练日志详情 495
7.4.2.4 资源占用情况 498
7.4.2.5 评估结果 499
7.4.3 停止、重建或查找作业 500
7.4.4 清除训练作业资源 500
7.5 超参搜索训练 501
7.5.1 超参搜索简介 501
7.5.2 搜索算法 501
7.5.2.1 贝叶斯优化(SMAC) 501
7.5.2.2 TPE 算法 502
7.5.2.3 模拟退火算法(Anneal) 502
7.5.3 创建超参搜索作业 503
7.6 旧版训练与新版训练差异 506
8 训练管理（旧版即将下线） 509
8.1 模型训练简介 509
8.2 订阅算法 510
8.3 常用框架 512
8.4 创建训练作业 518
8.4.1 创建训练作业简介 518
8.4.2 使用已有算法训练模型 518
8.4.3 使用常用框架训练模型 522
8.4.4 使用自定义镜像训练模型 529
8.5 停止或删除作业 533
8.6 管理训练作业版本 533
8.7 查看作业详情 536
8.8 管理作业参数 538

8.9 添加评估结果 538
8.10 管理可视化作业 543
9 AI 应用管理（即将下线） 546
9.1 AI 应用管理简介 546
9.2 创建 AI 应用 547
9.2.1 从训练中选择元模型 547
9.2.2 从模板中选择元模型 549
9.2.3 从容器镜像中选择元模型 551
9.2.4 从 OBS 中选择元模型 554
9.3 管理 AI 应用版本 558
9.4 模型转换 558
9.4.1 模型转换操作 558
9.4.2 模型输入目录规范 561
9.4.3 模型输出目录说明 562
9.4.4 转换模板 563
9.5 我的订阅模型 572
9.6 云服务订阅 AI 应用 574
9.7 发布 AI 应用 577
10 部署服务（即将下线） 580
10.1 部署服务简介 580
10.2 在线服务 580
10.2.1 部署为在线服务 580
10.2.2 查看服务详情 583
10.2.3 测试服务 587
10.2.4 访问在线服务（Token 认证） 590
10.2.5 访问在线服务（AK/SK 认证) 594
10.2.6 访问在线服务（APP 认证） 601
10.2.7 集成在线服务 609
10.3 批量服务 609
10.3.1 部署为批量服务 609
10.3.2 查看批量服务预测结果 614
10.4 边缘服务 614
10.4.1 部署为边缘服务 614
10.4.2 访问边缘服务 618
10.5 升级服务 622
10.6 启动或停止服务 624
10.7 删除服务 624
11 资源池 625
12 使用自定义镜像 630
12.1 自定义镜像简介 630
12.2 制作和上传自定义镜像 631

12.3 用于训练模型（旧版即将下线）.....632
12.3.1 训练作业自定义镜像规范.....632
12.3.2 使用自定义镜像创建训练作业（GPU）.....636
12.3.3 使用自定义镜像训练模型（Ascend）.....640
12.3.4 示例：使用自定义镜像创建训练作业.....643
12.4 用于创建 AI 应用.....644
12.4.1 创建 AI 应用的自定义镜像规范.....644
12.4.2 使用自定义镜像创建 AI 应用.....646
12.4.3 使用自定义镜像创建的 AI 应用部署服务.....647
13 模型包规范（即将下线）.....652
13.1 模型包规范介绍.....652
13.2 模型配置文件编写说明.....654
13.3 模型推理代码编写说明.....667
14 模型模板（即将下线）.....673
14.1 模型模板简介.....673
14.2 模板说明.....674
14.2.1 TensorFlow 图像分类模板.....674
14.2.2 TensorFlow-py27 通用模板.....675
14.2.3 TensorFlow-py36 通用模板.....676
14.2.4 MXNet-py27 通用模板.....677
14.2.5 MXNet-py36 通用模板.....678
14.2.6 PyTorch-py27 通用模板.....678
14.2.7 PyTorch-py36 通用模板.....679
14.2.8 Caffe-CPU-py27 通用模板.....680
14.2.9 Caffe-GPU-py27 通用模板.....681
14.2.10 Caffe-CPU-py36 通用模板.....682
14.2.11 Caffe-GPU-py36 通用模板.....682
14.2.12 ARM-Ascend 模板.....683
14.3 输入输出模式说明.....684
14.3.1 预置物体检测模式.....684
14.3.2 预置图像处理模式.....686
14.3.3 预置预测分析模式.....687
14.3.4 未定义模式.....689
15 自定义脚本代码示例.....690
15.1 TensorFlow.....690
15.2 TensorFlow 2.1.....696
15.3 PyTorch.....697
15.4 Caffe.....700
15.5 XGBoost.....706
15.6 Pyspark.....707
15.7 Scikit Learn.....708

16 权限管理 710
16.1 创建并授权使用 ModelArts 710
16.2 创建 ModelArts 自定义策略 711
17 监控 714
17.1 ModelArts 支持的监控指标 714
17.2 设置告警规则 716
17.3 查看监控指标 717
18 审计日志 719
18.1 支持云审计的关键操作 719
18.2 查看审计日志 723
19 建议反馈 724
A 修订记录 727

1 AI 工程师如何使用 ModelArts

面向熟悉代码编写和调测，熟悉常见AI引擎的开发者，ModelArts不仅提供了在线代码开发环境，还提供了从数据准备、模型训练、模型管理到模型部署上线的端到端开发流程（即AI全流程开发），帮助您高效、快速的构建一个可用模型。

本文档介绍了如何在ModelArts管理控制台完成AI开发，如果您习惯使用API或者SDK进行开发，建议查看《**ModelArts SDK参考**》和《**ModelArts API参考**》获取帮助。

使用AI全流程开发的端到端示例，请参见**快速入门**和**最佳实践**。

AI 全流程开发

ModelArts提供的AI全流程开发，兼容开发者的使用习惯，支持多种引擎和用户场景，使用自由度较高。下文介绍使用ModelArts平台，从准备数据到完成模型开发上线的全流程。针对开发者的其他场景，建议参考**ModelArts使用流程详解**。

图 1-1 AI 工程师的使用流程

表 1-1 使用流程说明

流程	子任务	说明	详细指导
准备数据	创建数据集	基于您的业务数据，您可以在ModelArts中创建数据集管理和预处理您的数据。	创建数据集
	标注数据	针对您创建的数据集，基于业务逻辑标注数据，对数据进行预处理，方便后续训练使用。数据标注的情况将影响模型训练效果。	标注数据
	发布数据集	数据标注完成后，将数据集发布。即可生成一个可以用于模型训练的数据集版本。	发布数据集
开发	创建Notebook	创建一个Notebook作为开发环境。	创建Notebook实例
	编写调试代码	在已有的Notebook中编写代码直接构建模型。	JupyterLab简介及常用操作 使用本地IDE开发模型
训练模型	选择算法	创建训练作业前需要先选择算法，可以订阅ModelArts预置的算法，也可以使用自己的算法。	选择算法
	创建训练作业	创建一个训练作业，选择可用的数据集版本，并使用前面编写完成的训练脚本。训练完成后，将生成模型并存储至OBS中。	创建训练作业
管理模型	编写推理代码和配置文件	针对您生成的模型，建议您按照ModelArts提供的模型包规范，编写推理代码和配置文件，并将推理代码和配置文件存储至训练输出位置。	模型包规范介绍
	导入模型	将训练完成的模型导入至ModelArts，方便将模型部署上线。	导入模型
部署模型	部署服务	ModelArts支持将模型部署为在线服务、批量服务和边缘服务。	• 部署为在线服务 • 部署为批量服务 • 部署为边缘服务
	访问服务	服务部署完成后，针对在线服务和边缘服务，您可以访问并使用服务，针对批量服务，您可以查看其预测结果。	• 访问在线服务 • 访问批量服务 • 访问边缘服务

使用订阅算法构建模型

面向有一定AI基础的AI初学者，您可以不关注模型开发，使用自己的业务数据，在AIGallery上订阅ModelArts提供的算法，进行模型训练，从而得到新模型。

图 1-2 AI 初学者使用流程

ModelArts 使用流程详解

ModelArts平台提供了从数据准备到模型部署的AI全流程开发，针对每个环节，其使用是相对自由的。针对AI工程师，梳理了ModelArts使用流程详解，您可以选择其中一种方式完成AI开发。

图 1-3 使用流程详解

2 免费体验 ModelArts

2.1 免费体验自动学习

在ModelArts自动学习功能中，在训练模型和部署上线阶段，可选择免费的计算规格，端到端体验一个自动学习项目，大大降低您的体验成本。

单击此处进入ModelArts管理控制台，参考如下操作指导体验免费规格的使用。

使用场景

自动学习项目分为“数据标注”、“模型训练”和“部署上线”。针对“数据标注”步骤，ModelArts服务并不会产生费用，针对“模型训练”和“部署上线”步骤，不管是训练还是部署，都需要使用计算资源，需要根据您选择的资源进行计费。当前，ModelArts推出了限时免费的计算规格，让您免费体验自动学习功能。

图 2-1 自动学习提供的免费规格

说明

自动学习训练免费规格资源不包含OBS存储资源费用。存储在OBS的数据，按您使用的OBS规格进行计费。

免费规格声明（模型训练）

- 自动学习训练免费规格用于使用体验，训练作业会在1小时后自动停止（不包括模型发布时间），因此建议设置最大训练时长为1小时。

- 限时免费的规格，性能有限，如果您的数据量较大，或者训练时长会超过1小时，建议选择收费的计算规格用于模型训练。
- 免费规格的资源是有限的，当使用人数较多时，会出现长时间的排队。如果希望获得更佳的体验，请选择付费规格。
- 只能在1个训练作业中使用免费规格。等训练作业结束后，其他训练作业可重新使用免费规格。
- 仅在“华北-北京四”区域提供了免费规格，请注意您使用的区域。

免费规格声明（部署上线）

- 免费规格仅用于使用体验，部署的服务会在1小时后自动停止（不包括模型发布时间）。如果您还需要使用免费规格继续运行，可重新启动服务，可再运行1小时，1小时后仍然会自动停止。
- 免费规格的资源是有限的，当使用人数较多时，会出现长时间的排队。如果希望获得更佳的体验，建议选择付费规格。
- 使用“部署上线”功能时，可用的免费规格有“自动学习免费规格（CPU）”或“自动学习免费规格（GPU）”。
- 任意一种免费规格只能在1个服务中使用免费规格。如果一个自动学习项目下的部署上线已使用了一种免费规格，不管是运行中还是停止状态，其他部署上线任务都无法再使用这个免费规格。
- 仅在“华北-北京四”区域提供了免费规格，请注意您使用的区域。

免费使用自动学习

如下免费使用的教程，以“图像分类”类型为例，其他类型的自动学习项目，其操作步骤类似，不再赘述。在开始使用如下端到端流程前，请参考**准备工作**，完成帐号注册、全局配置等操作。本文重点指导用户如何免费使用，如需自动学习详细操作步骤，可参见**最佳实践**。

1. 登录ModelArts管理控制台，在左侧导航栏中选择“自动学习”。
2. 参考**准备数据**，准备用于模型训练的数据集，并按要求上传至OBS。
3. 参考**创建项目**，创建一个“图像分类”的自动学习项目，在项目中，选择步骤**2**中准备的数据。
4. 参考**标注数据**，将所有图片数据完成标注。
5. 参考**自动训练**，在“训练设置”窗口中，选择“自动学习免费规格（GPU）”的计算资源，勾选“我已阅读并同意以上内容”，然后单击“确定”，开始模型训练。

图 2-2 设置为免费规格

6. 在“自动学习>模型训练”页面，等待训练结束。当训练状态变为“已完成”时，表示训练结束，您可以在当前页面查看训练详情。

   如果训练结果满意，可执行下一步部署上线，如果不满意，建议您优化数据及其标注信息，然后再启动一次训练。

7. 参考部署上线，将训练所得的模型部署为在线服务。

   在“部署”弹出框中，您可以选择“自动学习免费规格（CPU）”或“自动学习免费规格（GPU）”规格，两个规格均为免费规格。计算节点个数只能设置为1。勾选“我已阅读并同意以上内容”，然后单击“确定”启动部署上线任务。

图 2-3 选择免费规格部署在线服务

8. 当部署上线任务状态变为“运行中”时，表示部署已完成。您可以在“自动学习>部署上线”中查看服务的URL接口，也可以直接在此页面，上传一张图片，进行预测识别。

   使用免费规格的服务，将在1个小时后自动停止，如果您还需要使用此服务，可在“版本管理”区域中，单击“启动”，系统将使用之前选择的免费规格启动服务。如果此免费规格正在被当前帐号下的其他服务所用，则需要排队等待。

2.2 免费体验 AI 全流程开发

在使用ModelArts完成AI全流程开发时，端到端过程中，提供免费规格，让您免费体验ModelArts的全流程开发。

单击此处进入ModelArts管理控制台，参考如下操作指导体验免费规格的使用。

使用场景

在AI全流程开发过程中，分为如下几个步骤：

- **数据管理**：ModelArts数据管理功能目前免费开放，但是由于数据需存储在OBS中，存储的数据按OBS规则进行计费。
- **算法开发或预置算法**：ModelArts官方发布在AI Gallery的算法，均可免费使用。如果您自己在本地开发算法，则不涉及费用。
- **模型训练**：在创建训练作业时，可选择免费规格，完成模型训练。
- **导入模型**：ModelArts提供的模型导入及管理功能，不收费。
- **部署上线**：将导入的模型部署为在线服务时，可选择免费规格（CPU或GPU）的资源，将模型部署为服务。

针对之前的ModelArts，如果您想端到端体验AI全流程开发，在“模型训练”和“部署上线”时，都需要付费体验。当前ModelArts提供免费规格后，整个AI开发过程，可端到端免费体验。

图 2-4 AI 全流程开发过程中的免费规格

说明

由于免费规格仅针对ModelArts服务，在使用过程中，需使用OBS存储数据或模型时，会根据OBS计费规则进行计费。

免费规格声明（模型训练）

- 免费规格用于使用体验，训练作业会在1小时后自动停止，因此建议设置最大训练时长为1小时。
- 限时免费的规格，性能有限，如果您的数据量较大，或者训练时长会超过1小时，建议选择收费的计算规格用于模型训练。

- 免费规格的资源是有限的，当使用人数较多时，会出现长时间的排队。如果希望获得更佳的体验，请选择付费规格。
- 训练作业功能，仅提供了GPU类型的免费规格。
- 同一个帐号只能在1个训练作业中使用免费规格。等训练作业结束后，其他训练作业可重新使用免费规格。
- “训练管理”中，只有“创建训练作业”功能支持使用免费规格，“可视化作业”暂时无法使用免费规格进行体验。
- 仅在“华北-北京四”区域提供了免费规格，请注意您使用的区域。

免费规格声明（部署上线）

- 免费规格用于使用体验，部署的服务会在1小时后自动停止。如果您还需要使用免费规格继续运行，可重新启动服务，可再运行1小时，1小时后仍然会自动停止。72小时内没有再次启动，会释放资源，请注意文件备份。
- 免费规格的资源是有限的，当使用人数较多时，会出现长时间的排队。如果希望获得更佳的体验，请选择付费规格。
- 使用“部署上线”功能时，可用的免费规格有2种，1个免费CPU规格，1个免费GPU规格。但是两种规格不能同时在一个服务中使用。
- 同一个帐号下，任意一种免费规格只能在1个服务中使用免费规格。如果一个部署上线已使用了一种免费规格，不管是运行中还是停止状态，其他部署上线任务都无法再使用这个免费规格。
- 只有部署为“在线服务”时，支持使用免费规格。“批量服务”和“边缘服务”暂时无法使用免费规格进行体验。
- 使用免费规格的服务，可以通过修改操作，将资源规格修改为收费的规格，也可以修改为另一种免费规格。但是已经在使用收费规格的服务，无法修改为免费规格。
- 仅在“华北-北京四”区域提供了免费规格，请注意您使用的区域。

免费体验 AI 全流程开发

如下免费使用的教程，以使用预置算法创建训练作业为例，其他类型的训练作业，操作步骤类似，不再赘述。在开始使用如下端到端流程前，请参考**准备工作**，完成帐号注册、全局配置等操作。本文重点指导用户如何免费使用，如需自动学习详细操作步骤，可参见**快速入门**。

1. 登录ModelArts管理控制台。
2. 参考**创建数据集（旧版）**，创建一个“图像分类”类型的数据集。同时，将需要进行标注的数据，上传至对应的OBS中。您可以使用**花卉数据集样例**完成体验。
3. 参考**图像分类**，将上传的数据全部完成标注。
4. 参考**发布数据集**，将标注好的数据集发布为可用版本。
5. 参考**使用已有算法训练模型**，上传一个可应用于图像分类的算法，如“ResNet_v1_50”算法，创建训练作业。

   在“规格”右侧的下拉框中，选择带有“限时免费”标识的规格，用于免费体验。然后单击“下一步”，根据界面提示完成训练作业的创建。选用免费规格时，其对应的计算节点个数，仅支持使用1个节点，无法修改。

图 2-5 选择限时免费的规格

等待训练作业运行结束，当训练作业的状态变为“运行成功”时，表示模型训练完成。

6. 参考**从训练中选择元模型**，将训练后得到的模型导入ModelArts进行管理。
7. 参考**部署为在线服务**，将导入的模型部署为在线服务。

   在“计算节点规格”中，选择带有“免费规格”标识的规格，部署在线服务。

图 2-6 选择免费规格部署为在线服务

8. 在服务管理列表中，使用免费规格创建的服务，名称右侧带“免费”标识，等待服务部署完成，当状态变为“运行中”时，表示服务已部署完成。

   您可以单击服务名称进入服务详情页面，在“调用指南”查看服务的URL接口，或者进入“预测”页签，上传一张图片，进行预测识别。

   使用免费规格的服务，将在1个小时后自动停止，如果您还需要使用此服务，可在“在线服务”页面，单击“启动”，系统将使用之前选择的免费规格启动服务。

图 2-7 使用免费规格运行的服务

3 数据管理（New）

3.1 数据管理简介

在ModelArts中，您可以在“数据管理”页面，完成数据导入、数据标注等操作，为模型构建做好数据准备。ModelArts以数据集为数据基础，进行模型开发或训练等操作。

数据集的类型

当前ModelArts支持如下格式的数据集，包含文件型和表格型。

- 文件型
  - 图片：对图像类数据进行处理，支持 .jpg、.png、.jpeg、.bmp四种图像格式，支持用户进行图像分类、物体检测、图像分割类型的标注。
  - 音频：对音频类数据进行处理，支持.wav格式，支持用户进行声音分类、语音内容、语音分割三种类型的标注。
  - 文本：对文本类数据进行处理，支持.txt、.csv格式，支持用户进行文本分类、命名实体、文本三元组三种类型的标注。
  - 视频：对视频类数据进行处理，支持.mp4格式，支持用户进行视频标注。
  - 自由格式：管理的数据可以为任意格式，目前不支持标注，适用于无需标注或开发者自行定义标注的场景。如果您的数据集需存在多种格式数据，或者您的数据格式不符合其他类型数据集时，可选择自由格式的数据集。

图 3-1 自由格式数据集示例

- 表格型

表格：适合表格等结构化数据处理。数据格式支持csv。不支持标注，支持对部分表格数据进行预览，但是最多支持100条数据预览。

规格限制

- 除表格类型之外的数据集（如视频、文本、音频等），单个数据集的最大样本数量限制：1000000，最大标签数量限制：10000。
- 除图片类型之外的数据集（如视频、文本、音频等），单个样本大小限制：5GB。
- 针对图片类数据集（物体检测、图像分类、图像分割），单个图片大小限制：25MB。
- 单个manifest文件大小限制：5GB。
- 文本文件单行大小限制：100KB。
- 数据管理标注结果文件大小限制：100MB。

数据集管理流程及功能简介

图 3-2 标注管理全流程

表 3-1 功能介绍

功能	说明
创建数据集	创建一个新的数据集。
导入操作	将本地Manifest文件或者OBS的数据导入数据集中。
数据标注	支持对数据集创建不同类型的数据标注作业，对数据进行标注。
智能标注	支持对未标注的数据快速完成数据标注，为您节省70%以上的标注时间。
自动分组	您可以针对您选中的数据，执行自动分组，提升您的数据标注效率。
数据特征分析	对数据进行特征分析，帮助您了解数据。

功能	说明
团队标注简介	支持多人共同完成同一个数据标注作业，且支持标注作业创建者统一管理标注任务。添加团队及其成员，参与到数据集的标注工作。
数据处理	为了保障数据质量，以免对后续操作（如数据标注、模型训练等）带来负面影响，开发过程通常需要进行数据处理。常见的数据处理类型有：数据校验、数据清洗、数据选择、数据增强。
导出数据	支持将部分数据导出为新的数据集或者将数据导出至OBS。同时支持对任务历史进行查看和管理。
修改数据集	修改数据集的基本信息。如数据集名称、描述或标签等信息。
发布数据集	将标注后的数据集发布为新版本，以便应用于后续的模型构建。
管理数据集版本	通过数据集版本查看演进过程。
删除数据集	删除数据集以释放资源。

不同类型数据集支持的功能列表

其中，不同类型的数据集，支持不同的功能，详细信息请参见表3-2。

表 3-2 不同类型数据集支持的功能

数据集类型	创建数据集	导入数据	导出数据	发布数据集	修改数据集	管理版本	数据标注	智能标注	团队标注	自动分组	数据特征
图像	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持
音频	支持	支持	-	支持	支持	支持	支持	-	支持	-	-
文本	支持	支持	-	支持	支持	支持	支持	-	支持	-	-
视频	支持	支持	-	支持	支持	支持	支持	-	-	-	-
自由格式	支持	支持	-	支持	支持	支持	-	-	-	-	-
表格	支持	支持	-	支持	支持	支持	-	-	-	-	-

3.2 创建数据集

在ModelArts进行数据管理时，首先您需要创建一个数据集，后续的操作，如标注数据、导入数据、数据集发布等，都是基于您创建的数据集。

说明

当前ModelArts同时存在新版数据集和旧版数据集。

新版数据集在旧版的基础上将创建数据集和创建标注任务进行了解耦，创建数据集和创建标注作业分别是独立的任务，使用更灵活。

本文档主要介绍新版数据集创建流程。旧版数据集创建，请参考创建数据集（旧版）。

前提条件

- 数据管理功能需要获取访问OBS权限，在未进行委托授权之前，无法使用此功能。在使用数据管理功能之前，请前往“全局配置”页面，使用委托完成访问授权。
- 已创建用于存储数据的OBS桶及文件夹。并且，数据存储的OBS桶与ModelArts在同一区域。当前不支持OBS并行文件系统，请选择OBS对象存储。

操作步骤

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据集”，单击“前往新版”，进入新版“数据集”管理页面。

   图 3-3 进入新版数据集

2. 单击“创建数据集”，进入“创建数据集”页面，根据数据类型以及数据标注要求，选择创建不同类型的数据集。

   a. 填写数据集基本信息，数据集的“名称”、“描述”、“数据格式”、“数据类型”和“数据集输出位置”。ModelArts目前支持的类型及其说明请参见数据集的类型。

   图 3-4 数据集基本信息

b. 基本信息填写完成后，单击“下一步”，填写数据集的数据输入信息。

i. OBS导入数据

用户在OBS中有准备好的数据时，选择“OBS”，“导入路径”、“数据标注状态”、和数据“标注格式”。针对不同类型的数据集，数据输入支持的标注格式不同，ModelArts目前支持的标注格式及其说明请参见导入数据。

图 3-5 选择 OBS 中的数据格式和数据类型

ii. 从AI Gallery下载数据

当用户没有准备数据时，可以从AI Gallery上下载数据创建数据集。选择“AI Gallery”并选中列表中的一个资产，在“下载至OBS桶位置（数据集输入位置）”选择一个空目录用来存储下载的数据集。

图 3-6 AI Gallery 下载数据

c. 参数填写无误后，单击页面右下角“提交”。

数据集创建完成后，系统自动跳转至数据集管理页面，针对创建好的数据集，您可以执行数据导入、发布、修改、删除、数据处理、数据标注、数据特征、版本管理和导出操作。

不同类型数据集，支持的操作请参见不同类型数据集支持的功能。

文件型（图片、音频、文本、视频、自由格式）

表 3-3 数据集的详细参数

参数名称	说明
数据集输出位置	选择数据集输出位置的OBS路径，此位置会存放输出的标注信息等文件。 **说明** “数据集输出位置”不能与“数据输入路径”为同一路径，且不能是“数据输入路径”的子目录。 “数据集输出位置”建议选择一个空目录。 “数据集输出位置”不支持OBS并行文件系统下的路径，请选择OBS对象桶。
导入路径	选择需要导入数据的OBS路径，此位置会作为数据集的数据存储路径。 **说明** “导入路径”不支持OBS并行文件系统下的路径，请选择OBS对象桶。 创建数据集时，此OBS路径下的数据会导入数据集，后续若直接在OBS中修改数据，会造成数据集的数据与OBS的数据不一致，可能导致部分数据不可用。如果需要在数据集中修改数据，建议使用**同步新数据**或4章节**导入数据**功能。 超出数据集的样本和标签配额，会导致数据无法正常导入。
数据标注状态	选择数据的标注状态，分为“未标注”和“已标注”。 选择“已标注”时，需指定标注格式，并保证数据文件满足相应的格式规范，否则可能存在导入失败的情况。 仅图片（物体检测、图像分类、图像分割）、音频（声音分类）、文本（文本分类）类型的标注任务支持导入已标注数据。

表格型（表格）

图 3-7 表格类型的参数

说明

使用CSV文件时，需要注意以下两点：

- 当数据类型选择String时，默认会把双引号内的数据当作一条，所以同一行数据需要保证双引号闭环，否则会导致数据过大，无法显示。
- 当CSV文件的某一行的列数与定义的Schema不同，则会忽略当前行。

表 3-4 数据集的详细参数

参数名称	说明
数据集输出位置	选择表格数据存储路径（OBS路径），此位置会存放由数据源导入的数据。此位置不能和OBS数据源中的文件路径相同或为其子目录。 创建表格数据集后，在存储路径下会自动生成以下4个目录。 ● annotation：版本发布目录，每次发布版本，会在此目录下生成和版本名称相同的子目录。 ● data：数据存放目录，导入的数据会放在此目录。 ● logs：日志存放目录。 ● temp：临时工作目录。
数据源（“OBS”）	● “文件路径”：单击输入框右侧按钮，可打开当前帐号下的所有OBS桶，请选择需要导入的数据文件所在目录。 ● “导入是否包含表头”：开启表示导入文件包含表头，此时会将导入文件的第一行作为列名，否则会添加默认列名，自动填写在Schema信息中。 OBS的详细功能说明，请参见《**OBS用户指南**》。

参数名称	说明
数据源（“DWS”）	● “集群名称”：系统自动将当前账号下的DWS集群展现在列表中，您可以在下拉框中选择您所需的DWS集群。 ● “数据库名称”：根据选择的DWS集群，填写数据所在的数据库名称。 ● “表名称”：根据选择的数据库，填写数据所在的表。 ● “用户名”：输入DWS集群管理员用户的用户名。 ● “密码”：输入DWS集群管理员用户的密码。 DWS的详细功能说明，请参见《**DWS用户指南**》。 **说明** 从DWS导入数据，需要借助DLI的功能，如果用户没有访问DLI服务的权限，需根据页面提示创建DLI的委托。
数据源（“DLI”）	● “队列名称”：系统自动将当前帐号下的DLI队列展现在列表中，您可以在下拉框中选择您所需的队列。 ● “数据库名称”：根据选择的队列展现所有的数据库，请在下拉框中选择您所需的数据库。 ● “表名称”：根据选择的数据库展现此数据库中的所有表。请在下拉框中选择您所需的表。 DLI的详细功能说明，请参见《**DLI用户指南**》。
数据源（“MRS”）	● “集群名称”：系统自动将当前帐号下的MRS集群展现在此列表中，但是流式集群不支持导入操作。请在下拉框中选择您所需的集群。 ● “文件路径”：根据选择的集群，输入对应的文件路径，此文件路径为HDFS路径。 ● “导入是否包含表头”：开启表示导入时将表头同时导入。 MRS的详细功能说明，请参见《**MRS用户指南**》。
Schema信息	表格的列名和对应类型，需要跟导入数据的列数保持一致。请根据您导入的数据输入“列名”，同时选择此列的“类型”。其中支持的类型见**表3-5**。 单击“添加Schema信息”，即可增加一行列。创建数据集时必须指定schema，且一旦创建不支持修改。 从OBS数据源导入数据，会自动获取文件路径下csv文件的schema，如果多个csv文件的schema不一致会报错。

表 3-5 Schema 数据类型说明

类型	描述	存储空间	范围
String	字符串	-	-
Short	有符号整数	2字节	-32768-32767
Int	有符号整数	4字节	-2147483648～2147483647

类型	描述	存储空间	范围
Long	有符号整数	8字节	-9223372036854775808～9223372036854775807
Double	双精度浮点型	8字节	-
Float	单精度浮点型	4字节	-
Byte	有符号整数	1字节	-128-127
Date	日期类型，描述了特定的年月日，格式：yyyy-MM-dd，例如2014-05-29	-	-
Timestamp	时间戳，表示日期和时间。格式：yyyy-MM-dd HH:mm:ss	-	-
Boolean	布尔类型	1字节	TRUE/FALSE

3.3 创建标注作业

由于模型训练过程需要大量有标签的数据，因此在模型训练之前需对没有标签的数据添加标签。您可以通过创建单人标注作业或团队标注作业对数据进行手工标注，或对任务启动智能标注添加标签，快速完成对图片的标注操作，也可以对已标注图片修改或删除标签进行重新标注。

标注作业支持的数据类型

对于不同类型的数据集，用户可以选择不同的标注任务，当前ModelArts支持如下类型的标注任务。

- 图片
  - 图像分类：识别一张图片中是否包含某种物体。
  - 物体检测：识别出图片中每个物体的位置及类别。
  - 图像分割：根据图片中的物体划分出不同区域。
- 音频
  - 声音分类：对声音进行分类。
  - 语音内容：对语音内容进行标注。
  - 语音分割：对语音进行分段标注。
- 文本
  - 文本分类：对文本的内容按照标签进行分类处理。
  - 命名实体：针对文本中的实体片段进行标注，如“时间”、“地点”等。
  - 文本三元组：针对文本中的实体片段和实体之间的关系进行标注。
- 视频

  视频标注：识别出视频中每个物体的位置及分类。目前仅支持mp4格式。

前提条件

在进行数据标注前，需要创建相应类型的数据集。具体步骤参考创建数据集。

操作步骤

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据标注”，进入“数据标注”管理页面。
2. 单击“创建标注作业”，进入“创建标注作业”页面，根据需求创建不同类型的标注作业。
   a. 填写标注作业基本信息，标注作业的“名称”和“描述”。

   图 3-8 标注作业基本信息

   b. 根据您的需求，选择“标注场景”和“标注类型”。

   图 3-9 选择标注场景和标注类型

   c. 针对不同类型的标注作业，需填写参数不同，请参考如下类型标注作业对应的参数介绍。
      - 图片（图像分类、物体检测、图像分割）
      - 音频（声音分类、语音内容、语音分割）
      - 文本（文本分类、命名实体、文本三元组）
      - 视频
   d. 参数填写无误后，单击页面右下角“创建”。

      标注作业创建完成后，系统自动跳转至数据标注管理页面，针对创建好的标注作业，您可以执行智能标注、发布、修改和删除等操作。

图片（图像分类、物体检测、图像分割）

图 3-10 图像分类和物体检测类型的参数

表 3-6 图片类型标注作业的详细参数

参数名称	说明
数据集名称	选择支持当前标注类型的数据集。
添加标签集	• **设置标签名称**：在标签名称文本框中，输入标签名称。长度为1～1024字符。 • **添加标签**：单击“添加标签”可增加多个标签。 • **设置标签颜色**：“物体检测”和“图像分割”类型标注作业需设置此参数。在每个标签右侧的标签颜色区域下，可在色板中选择颜色，或者直接输入十六进制颜色码进行设置。 • **设置标签属性**：针对“物体检测”类型标注作业，在设置完标签颜色后，可在右侧单击加号，增加对应的标签属性。标签属性用于区分同一标签物体的不同属性。例如，黄色小猫、黑色小猫。标签为cat，颜色为不同的标签属性。
启用团队标注	选择是否启用团队标注。图像分割暂不支持团队标注，当选择图像分割类型时，界面不显示此参数。 启用团队标注功能，需填写对应的团队标注任务“类型”，同时选择对应的“标注团队”及参与标注的“团队成员”。参数详细介绍请参见管理团队标注任务。 在启用“团队标注”前，需确保您已经在“标注团队”管理页面，添加相应的团队以及成员。如果没有标注团队，可直接从界面链接跳转至“标注团队”页面，添加您的团队并为其添加成员。详细指导请参见团队标注简介。 启用团队标注功能的数据集，在创建完成后，可以在“标注类型”中看到“团队标注”的标识。

音频（声音分类、语音内容、语音分割）

图 3-11 声音分类、语音内容、语音分割类型的参数

表 3-7 音频类型标注作业的详细参数

参数名称	说明
数据集名称	选择支持当前标注类型的数据集。
添加标签集（声音分类）	“声音分类”类型的标注作业可以添加标签集。 ● **设置标签名称**：在标签名称文本框中，长度为1～1024字符。 ● **添加标签**：单击“添加标签”可增加多个标签。
标签管理（语音分割）	“语音分割”类型的标注作业，支持标签管理。 ● **单标签** 单标签适用于一段音频标注只有一种类别的音频，通常标注一个标签。 – **设置标签名称**：在“标签名”列输入标签名称。长度为1～1024字符。 – **设置标签颜色**：在“标签颜色”列设置标签颜色。可在色板中选择颜色，或者直接输入十六进制颜色码进行设置。 ● **多标签** 多标签适用于多维度标注，例如在一段音频标注噪音与人说话的声音两种类别，其中说话的声音还可以标注为不同人的声音。单击“新建标签类别”可添加多个标签类别，一个标签类别可以包含多个标签。“标签类别”和“标签名”只能是中文、字母、数字、英文句号、下划线或中划线组成的合法字符串。长度为1～256字符。 – **设置标签类别**：在“标签类别”输入标签类别的名称。 – **设置标签名称**：在“标签名”输入标签名称。 – **添加标签**：单击“添加标签”可增加多个标签。
启用语音内容标注（语音分割）	仅“语音分割”类型数据集支持设置，默认关闭。如果启用此功能，支持针对语音内容进行标注。

参数名称	说明
启用团队标注（语音分割）	仅“语音分割”类型支持团队标注，因此选择创建语音分割类型时，支持设置是否启用团队标注。 启用团队标注功能，需填写对应的团队标注任务“类型”，同时选择对应的“标注团队”及参与标注的“团队成员”。参数详细介绍请参见创建团队标注任务。 在启用“团队标注”前，需确保您已经在“标注团队”管理页面，添加相应的团队以及成员。如果没有标注团队，可直接从界面链接跳转至“标注团队”页面，添加您的团队并为其添加成员。详细指导请参见团队标注简介。 启用团队标注功能的数据集，在创建完成后，可以在“标注类型”中看到“团队标注”的标识。

文本（文本分类、命名实体、文本三元组）

图 3-12 文本分类、命名实体、文本三元组类型的参数

表 3-8 文本类型标注作业的详细参数

参数名称	说明
数据集名称	选择支持当前标注类型的数据集。

参数名称	说明
添加标签集（文本分类、命名实体）	• **设置标签名称**：在标签名称文本框中，输入标签名称。长度为1～1024字符。 • **添加标签**：单击“添加标签”可增加多个标签。 • **设置标签颜色**：在每个标签右侧的标签颜色区域下，可在色板中选择颜色，或者直接输入十六进制颜色码进行设置。
添加标签集（文本三元组）	针对“文本三元组”类型的数据集，需要设置实体标签和关系标签。 • **实体标签**：需设置标签名以及标签颜色。可在颜色区域右侧单击加号增加多个标签。 • **关系标签**：关系标签为两个实体之间的关系。需设置起始实体和终止实体，您需要先添加至少2个实体标签后，再添加关系标签。
启用团队标注	选择是否启用团队标注。 启用团队标注功能，需填写对应的团队标注任务“类型”，同时选择对应的“标注团队”及参与标注的“团队成员”。参数详细介绍请参见**创建团队标注任务**。 在启用“团队标注”前，需确保您已经在“标注团队”管理页面，添加相应的团队以及成员。如果没有标注团队，可直接从界面链接跳转至“标注团队”页面，添加您的团队并为其添加成员。详细指导请参见**团队标注简介**。 启用团队标注功能的数据集，在创建完成后，可以在“标注类型”中看到“团队标注”的标识。

视频

图 3-13 视频类型的参数

表 3-9 视频类型标注作业的详细参数

参数名称	说明
数据集名称	选择支持当前标注类型的数据集。
添加标签集	• **设置标签名称**：在标签名称文本框中，输入标签名称。长度为1～1024字符。 • **添加标签**：单击“添加标签”可增加多个标签。 • **设置标签颜色**：在每个标签右侧的标签颜色区域下，可在色板中选择颜色，或者直接输入十六进制颜色码进行设置。

3.4 标注数据

3.4.1 图像分类

由于模型训练过程需要大量有标签的图片数据，因此在模型训练之前需对没有标签的图片添加标签。您可以通过手工标注或智能一键标注的方式添加标签，快速完成对图片的标注操作，也可以对已标注图片修改或删除标签进行重新标注。

针对图像分类场景，开始标注前，您需要了解：

- 图片标注支持多标签，即一张图片可添加多个标签。
- 标签名是由中文、大小写字母、数字、中划线、下划线或特殊符号组成，且不超过1024位的字符串。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据标注”，进入“数据标注”管理页面。

2. 在标注作业列表中，基于“标注类型”选择需要进行标注的标注作业，单击标注作业名称进入标注作业标注详情页。
3. 在标注作业标注详情中，展示此标注作业下全部数据。

同步新数据

ModelArts会自动将数据集中新增的数据同步至标注作业，包含数据及当前标注作业支持的标注信息。

为了快速获取数据集中最新数据，可在标注作业详情页的“全部”、“未标注”或“已标注”页签中，单击“同步新数据”，快速将数据集中的数据添加到标注作业中。

筛选数据

在标注作业详情页面，默认展示作业中全部数据，您可以在“全部”、“未标注”或页签下，在筛选条件区域，单击“展开”，添加筛选条件，快速过滤出您想要查看的数据。

支持的筛选条件如下所示，您可以设置一个或多个选项进行筛选。

- 难例集：难例或非难例。
- 标签：您可以选择全部标签，或者基于您指定的标签，选中其中一个或多个。
- 文件名或目录：根据文件名称或者文件存储目录筛选。
- 标注人：选择执行标注操作的帐号名称。
- 样本属性：表示自动分组生成的属性。只有启用了自动分组任务后才可使用此筛选条件。
- 数据属性：可筛选数据的来源，选择“全部”或“推理”。

图 3-14 筛选条件

标注图片（手工标注）

在标注作业详情页中，展示了此数据集中“全部”、“未标注”和“已标注”的图片，默认显示“全部”的图片列表。单击图片，即可进行图片的预览，对于已标注图片，预览页面下方会显示该图片的标签信息。

1. 在“未标注”页签，勾选需进行标注的图片。
   - 手工点选：在图片列表中，单击勾选图片左上角的选择框，进入选择模式，表示图片已勾选。可勾选同类别的多个图片，一起添加标签。
   - 批量选中：如果图片列表的当前页，所有图片属于一种类型，可以在图片列表的右上角单击“选择当前页”，则当前页面所有的图片将选中。

2. 为选中图片添加标签。
   a. 在右侧的“添加标签”区域中，单击“标签名”右侧的文本框中设置标签。
      单击“标签名”右侧的文本框，然后从下拉列表中选择已有的标签。如果已有标签无法满足要求时，直接在文本框中添加新标签。
   b. 单击“确定”。此时，选中的图片将被自动移动至“已标注”页签，且在“未标注”和“全部”页签中，标签的信息也将随着标注步骤进行更新，如增加的标签名称、各标签对应的图片数量。

图 3-15 添加标签

查看已标注图片

在标注任务详情页，单击“已标注”页签，您可以查看已完成标注的图片列表。图片缩略图下方默认呈现其对应的标签，您也可以勾选图片，在右侧的“选中文件标签”中了解当前图片的标签信息。

修改标注

当数据完成标注后，您还可以进入已标注页签，对已标注的数据进行修改。

- **基于图片修改**
  在标注作业详情页面，单击“已标注”页签，然后在图片列表中选中待修改的图片（选择一个或多个）。在右侧标签信息区域中对图片信息进行修改。

修改标签：在“选中文件标签”区域中，单击操作列的编辑图标，然后在文本框中输入正确的标签名，然后单击确定图标完成修改。

删除标签：在“选中文件标签”区域中，单击操作列的删除图标删除该标签。此操作仅删除选中图片中的标签。

图 3-16 编辑标签

- **基于标签修改**

  在标注作业详情页，单击“标签管理”页签，显示全部标签列表。

  – 修改标签：单击操作列的修改按钮，然后在弹出的对话框中输入修改后的标签名，然后单击“确定”完成修改。修改后，之前添加了此标签的图片，都将被标注为新的标签名称。

  – 删除标签：单击操作列的删除按钮，之前添加了此标签的图片，都将删除此标签。

图 3-17 全部标签的信息

添加数据

除了同步数据集中的新数据外，您还可以在标注作业中，直接添加图片，用于数据标注。添加的数据将先导入至标注任务关联的数据集中，然后标注任务会自动同步数据集中最新的数据。

1. 在标注作业详情页面，单击“全部”、“已标注”或“未标注”页签，然后单击左上角“添加数据”，选择添加数据。

图 3-18 添加数据

2. 在弹出的导入对话框中，选择数据来源和导入方式，选择导入的数据路径和数据标注状态。

图 3-19 添加图片

3. 在导入对话框中，单击“确定”，完成添加数据的操作。

   您添加的图片将自动呈现在“全部”的图片列表中，也可单击“添加数据>查看历史记录”，进入“任务历史”界面，可查看相应的导入历史。

图 3-20 查看历史数据

删除图片

通过数据删除操作，可将需要丢弃的图片数据快速删除。

在“全部”、“未标注”或“已标注”页面中，依次选中需要删除的图片，或者选择“选择当前页”选中该页面所有图片，然后单击左上角“删除图片”。在弹出的对话框中，根据实际情况选择是否勾选“同时删除源文件”，确认信息无误后，单击“确定”完成图片删除操作。

其中，被选中的图片，其左上角将显示为勾选状态。如果当前页面无选中图片时，“删除图片”按钮为灰色，无法执行删除操作。

说明

如果勾选了“同时删除源文件”，删除图片操作将删除对应OBS目录下存储的图片，此操作可能会影响已使用此源文件的其他数据集或数据集版本，有可能导致展示异常或训练/推理异常。删除后，数据将无法恢复，请谨慎操作。

3.4.2 物体检测

由于模型训练过程需要大量有标签的图片数据，因此在模型训练之前需对没有标签的图片添加标签。您可以通过手工标注或智能一键标注的方式添加标签，快速完成对图片的标注操作，也可以对已标注图片修改或删除标签进行重新标注。

针对物体检测场景，开始标注前，您需要了解：

- 图片中所有目标物体都要标注。
- 目标物体清晰无遮挡的，必须画框。
- 画框仅包含整个物体。框内包含整个物体的全部，画框边缘不可与待标注的物体的边缘轮廓相交，在此基础之上确保边缘和待标注物体间不要留着空隙，避免背景对模型训练造成干扰。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据标注”，进入“数据标注”管理页面。
2. 在标注作业列表中，基于“标注类型”选择需要进行标注的标注作业，单击标注作业名称进入标注作业标注详情页。
3. 在标注作业标注详情中，展示此标注作业下全部数据。

同步新数据

ModelArts会自动将数据集中新增的数据同步至标注作业，包含数据及当前标注作业支持的标注信息。

为了快速获取数据集中最新数据，可在标注作业详情页的“全部”、“未标注”或“已标注”页签中，单击“同步新数据”，快速将数据集中的数据添加到标注作业中。

筛选数据

在标注作业详情页面，默认展示作业中全部数据，您可以在“全部”、“未标注”或页签下，在筛选条件区域，单击“展开”，添加筛选条件，快速过滤出您想要查看的数据。

支持的筛选条件如下所示，您可以设置一个或多个选项进行筛选。

- 难例集：难例或非难例。
- 标签：您可以选择全部标签，或者基于您指定的标签，选中其中一个或多个。
- 文件名或目录：根据文件名称或者文件存储目录筛选。
- 标注人：选择执行标注操作的帐号名称。
- 样本属性：表示自动分组生成的属性。只有启用了自动分组任务后才可使用此筛选条件。
- 数据属性：可筛选数据的来源，选择“全部”或“推理”。

图 3-21 筛选条件

标注图片（手工标注）

标注作业详情页中，展示了此数据集中“全部”“未标注”和“已标注”的图片，默认显示“全部”的图片列表。

1. 在“未标注”页签图片列表中，单击图片，自动跳转到标注页面。在标注页面，常用按钮的使用可参见**表3-11**。
2. 在页面上方工具栏选择合适的标注图形，系统默认的标注图形为矩形。本示例使用矩形工具进行标注。

说明

页面左侧可以选择多种形状对图片进行标注。标注第一张图片时，一旦选择其中一种，其他图片默认使用此形状进行标注，用户可以根据自己需求再进行切换。

表 3-10 支持的标注框

图标	使用说明
□	矩形。鼠标单击标注对象左上角边缘位置，界面将出现矩形框，移动鼠标使得矩形框覆盖标注对象，然后单击完成标注。
⬠	多边形。在标注对象所在范围内，鼠标左键单击完成一个点的标注，沿着物体的形状边缘，通过鼠标指定多个点，最终单击到第一个点的位置，由所有的点组成一个多边形形状。使得需标注的对象在此标注框内。
○	圆形。在标注对象中，选择物体的中心点位置，单击鼠标确定圆心，然后移动鼠标，使得圆形框覆盖标注对象，然后再单击鼠标完成标注。
／	直线。在标注对象中，选择物体的起始点，单击鼠标确定直线的起始点，然后使得直线覆盖标注对象，然后再单击鼠标完成标注。
⋯	虚线。在标注对象中，选择物体的起始点，单击鼠标确定虚线的起始点，然后使得虚线覆盖标注对象，然后再单击鼠标完成标注。
⊙	点。单击图片中的物体所在位置，即可完成点的标注。

3. 在弹出的添加标签文本框中，直接输入新的标签名，在文本框前面选中标签颜色，然后单击“添加”。如果已存在标签，从下拉列表中选择已有的标签，单击“添加”。

   逐步标注图片中所有物体所在位置，一张图片可添加多个标签。完成一张图片标注后，可单击图片下方图片列表，快速选中其他未标注的图片，然后在标注页面中执行标注操作。

图 3-22 添加物体检测标签

4. 单击页面上方“返回数据标注预览”查看标注信息，在弹框中单击“确定”保存当前标注并离开标注页面。

   选中的图片被自动移动至“已标注”页签，且在“未标注”和“全部”页签中，标签的信息也将随着标注步骤进行更新，如增加的标签名称、标签对应的图片数量。

表 3-11 标注界面的常用按钮

按钮图标	功能说明
	撤销上一个操作。
	重做上一个操作。
	放大图片。
	缩小图片。
	删除当前图片中的所有标注框。
	显示或隐藏标注框。只有在已标注图片中可使用此操作。
	拖动，可将标注好的框拖动至其他位置，也可以选择框的边缘，更改框的大小。

按钮图标	功能说明
	复位，与上方拖动为同组操作，当执行了拖动后，可以单击复位按钮快速将标注框恢复为拖动前的形状和位置。
	全屏显示标注的图片。

查看已标注图片

在标注作业详情页，单击“已标注”页签，您可以查看已完成标注的图片列表。单击图片，可在右侧的“当前文件标签”中了解当前图片的标签信息。

修改标注

当数据完成标注后，您还可以进入已标注页签，对已标注的数据进行修改。

- **基于图片修改**

  在标注作业详情页面，单击“标注”页签，选择“已标注”页签，然后在图片列表中选中待修改的图片，单击图片跳转到标注页面，在右侧标签信息区域中对图片信息进行修改。

  - 修改标签：“标注”区域中，单击编辑图标，在文本框中输入正确的标签名，然后单击确定图标完成修改。也可以单击标签，在图片标注区域，调整标注框的位置和大小，完成调整后，单击其他标签即可保存修改。
  - 删除标签：在“标注”区域中，单击删除图标即可删除此图片中的标签。

    标签删除后，单击页面左上角的“返回数据标注预览”离开标注页面，在弹出对话框中保存标注。图标的标签全部删除后，该图片会重新回到“未标注”页签。

    图 3-23 编辑物体检测标签

- **基于标签修改**

  在标注作业详情页中，单击“已标注”页签，在图片列表右侧，显示全部标签的信息。

  - 修改标签：在已标注页签单击打开需要修改标签的图片，在标注区域右侧的“当前文件标签”单击操作列的编辑图标，然后在弹出的对话框中输入修改后的标签，然后单击确定图标完成修改。修改后，之前添加了此标签的图片，都将被标注为新的标签名称。
  - 删除标签：在已标注页签单击打开需要修改标签的图片，在标注区域右侧的“当前文件标签”，单击操作列的删除图标可删除此标签。

图 3-24 物体检测的全部标签

添加数据

除了同步数据集中的新数据外，您还可以在标注作业中，直接添加图片，用于数据标注。添加的数据将先导入至标注任务关联的数据集中，然后标注任务会自动同步数据集中最新的数据。

1. 在标注作业详情页面，单击“全部”、“已标注”或“未标注”页签，然后单击左上角“添加数据”，选择添加数据。

图 3-25 添加数据

2. 在弹出的导入对话框中，选择数据来源和导入方式，选择导入的数据路径和数据标注状态。

图 3-26 添加图片

3. 在导入对话框中，单击“确定”，完成添加数据的操作。

   您添加的图片将自动呈现在“全部”的图片列表中，也可单击“添加数据>查看历史记录”，进入“任务历史”界面，可查看相应的导入历史。

   图 3-27 查看历史数据

删除图片

通过数据删除操作，可将需要丢弃的图片数据快速删除。

在“全部”、“未标注”或“已标注”页面中，依次选中需要删除的图片，或者选择“选择当前页”选中该页面所有图片，然后单击左上角“删除图片”。在弹出的对话框中，根据实际情况选择是否勾选“同时删除源文件”，确认信息无误后，单击“确定”完成图片删除操作。

其中，被选中的图片，其左上角将显示为勾选状态。如果当前页面无选中图片时，“删除图片”按钮为灰色，无法执行删除操作。

说明

如果勾选了“同时删除源文件”，删除图片操作将删除对应OBS目录下存储的图片，此操作可能会影响已使用此源文件的其他数据集或数据集版本，有可能导致展示异常或训练/推理异常。删除后，数据将无法恢复，请谨慎操作。

3.4.3 图像分割

由于模型训练过程需要大量有标签的图片数据，因此在模型训练之前需对没有标签的图片添加标签。您可以通过在ModelArts控制台进行标注，也可以对已标注图片修改或删除标签进行重新标注。

针对图像分割场景，开始标注前，您需要了解：

- 图片中需要提取轮廓的物体都要标注。
- 支持使用多边形标注和极点标注。
  - 多边形标注，根据目标物体的轮廓绘制多边形。
  - 极点标注，在目标物体轮廓的最上、最左、最下、最右的位置分别标注四个极点，极点要在物体的轮廓上。系统将根据标注的极点推理出物体的轮廓。对于背景比较复杂的图片，极点标注效果不佳，推荐使用多边形标注。

- 多边形标注时，标注框或极点，必须在图片范围内，超出图片将导致后续作业异常。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据标注”，进入“数据标注”管理页面。
2. 在标注作业列表中，基于“标注类型”选择需要进行标注的标注作业，单击标注作业名称进入标注作业标注详情页。
3. 在标注作业标注详情中，展示此标注作业下全部数据。

同步新数据

ModelArts会自动将数据集中新增的数据同步至标注作业，包含数据及当前标注作业支持的标注信息。

为了快速获取数据集中最新数据，可在标注作业详情页的“全部”、“未标注”或“已标注”页签中，单击“同步新数据”，快速将数据集中的数据添加到标注作业中。

筛选数据

在标注作业详情页面，默认展示作业中全部数据，您可以在“全部”、“未标注”或页签下，在筛选条件区域，单击“展开”，添加筛选条件，快速过滤出您想要查看的数据。

支持的筛选条件如下所示，您可以设置一个或多个选项进行筛选。

- 难例集：难例或非难例。
- 标签：您可以选择全部标签，或者基于您指定的标签，选中其中一个或多个。
- 文件名或目录：根据文件名称或者文件存储目录筛选。
- 标注人：选择执行标注操作的帐号名称。
- 样本属性：表示自动分组生成的属性。只有启用了自动分组任务后才可使用此筛选条件。
- 数据属性：可筛选数据的来源，选择“全部”或“推理”。

图 3-28 筛选条件

标注图片（手工标注）

标注作业详情页中，展示了此标注作业中“未标注”和“已标注”的图片，默认显示“全部”的图片列表。

1. 在“未标注”页签图片列表中，单击图片，自动跳转到标注页面。在标注页面，常用按钮的使用可参见表3-13。
2. 选择标注方式。

   在标注页面，上方工具栏提供了常用的标注方式及常用按钮，系统默认的标注方式为多边形标注。选择多边形标注或极点标注。

   说明

   标注第一张图片时，一旦选择其中一种，其他所有图片都需要使用此方式进行标注。

图 3-29 工具栏

表 3-12 标注方式

图标	使用说明
	多边形。在标注对象所在范围内，鼠标左键单击完成一个点的标注，沿着物体的形状边缘，通过鼠标指定多个点，最终单击到第一个点的位置，由所有的点组成一个多边形形状。使得需标注的对象在此标注框内。
	极点标注。在目标物体轮廓的最上、最左、最下、最右的位置分别标注四个极点，极点要在物体的轮廓上。系统将根据标注的极点推理出物体的轮廓。

表 3-13 工具栏常用按钮

按钮图标	功能说明
	撤销上一个操作。
	重做上一个操作。
	放大图片。
	缩小图片。
	删除当前图片中的所有标注框。
	显示或隐藏标注框。只有在已标注图片中可使用此操作。
	拖动，可将标注好的框拖动至其他位置，也可以选择框的边缘，更改框的大小。
	复位，与上方拖动为同组操作，当执行了拖动后，可以单击复位按钮快速将标注框恢复为拖动前的形状和位置。

按钮图标	功能说明
	全屏显示标注的图片。

3. 标注物体。

   以极点标注为例。识别图片中的物体，单击左键分别定位物体的最上、最左、最下、最右的位置点。确定位置后，将弹出对话框，填入标签名称，单击确定添加物体的标签。确定后系统将自动推理出物体的轮廓。

   完成一张图片标注后，可单击图片下方图片列表，快速选中其他未标注的图片，然后在标注页面中执行标注操作。

   图 3-30 标注物体轮廓

4. 单击页面上方“返回数据标注预览”查看标注信息，在弹框中单击“确定”保存当前标注并离开标注页面。

   选中的图片被自动移动至“已标注”页签，且在“未标注”和“全部”页签中，标签的信息也将随着标注步骤进行更新，如增加的标签名称、标签对应的图片数量。

查看已标注图片

在标注作业详情页中，单击“已标注”页签，您可以查看已完成标注的图片列表。单击图片，可在右侧的“当前文件标签”中了解当前图片的标签信息。

修改标注信息

当数据完成标注后，您还可以进入已标注页签，对已标注的数据进行修改。

在数据标注详情页面，单击“已标注”页签，然后在图片列表中选中待修改的图片，单击图片跳转到标注页面，在右侧标签信息区域中单击此图片已添加的标注信息。

- 修改标签：“标注”区域中，单击编辑图标，在弹出框中输入正确的标签名或标签颜色，然后单击“确定”完成修改。也可以单击标签，在图片标注区域，调整标注框的位置和大小，完成调整后，单击其他标签即可保存修改。
- 修改图片标注信息：在图片展示区，显示物体边缘，可单击蓝色圆点，将标注框调整至物体边缘。

- 删除标签：在“标注”区域中，单击删除图标即可删除此图片中的标签。图片的标签全部删除后，该图片会重新回到“未标注”页签。

标注标注信息修改后，单击页面左上角的“返回数据标注预览”离开标注页面，在弹出对话框中单击“确定”保存修改。

图 3-31 编辑标注信息

添加数据

除了同步数据集中的新数据外，您还可以在标注作业中，直接添加图片，用于数据标注。添加的数据将先导入至标注任务关联的数据集中，然后标注任务会自动同步数据集中最新的数据。

1. 在标注作业详情页面，单击“全部”、“已标注”或“未标注”页签，然后单击左上角“添加数据”，选择添加数据。

   图 3-32 添加数据

2. 在弹出的导入对话框中，选择数据来源和导入方式，选择导入的数据路径和数据标注状态。

图 3-33 添加图片

3. 在导入对话框中，单击“确定”，完成添加数据的操作。

   您添加的图片将自动呈现在“全部”的图片列表中，也可单击“添加数据>查看历史记录”，进入“任务历史”界面，可查看相应的导入历史。

图 3-34 查看历史数据

删除图片

通过数据删除操作，可将需要丢弃的图片数据快速删除。

在“全部”、“未标注”或“已标注”页面中，依次选中需要删除的图片，或者选择“选择当前页”选中该页面所有图片，然后单击左上角“删除图片”。在弹出的对话框中，根据实际情况选择是否勾选“同时删除源文件”，确认信息无误后，单击“确定”完成图片删除操作。

其中，被选中的图片，其左上角将显示为勾选状态。如果当前页面无选中图片时，“删除图片”按钮为灰色，无法执行删除操作。

说明

如果勾选了“同时删除源文件”，删除图片操作将删除对应OBS目录下存储的图片，此操作可能会影响已使用此源文件的其他数据集或数据集版本，有可能导致展示异常或训练/推理异常。删除后，数据将无法恢复，请谨慎操作。

3.4.4 文本分类

由于模型训练过程需要大量有标签的数据，因此在模型训练之前需对没有标签的文本添加标签。您也可以对已标注文本进行修改、删除和重新标注。

针对文本分类场景，是对文本的内容按照标签进行分类处理，开始标注前，您需要了解：

- 文本标注支持多标签，即一个标注对象可添加多个标签。
- 标签名是由中文、大小写字母、数字、中划线、下划线或特殊符号组成，且不超过1024位的字符串。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据标注”，进入“数据标注”管理页面。
2. 在标注作业列表中，基于“标注类型”选择需要进行标注的标注作业，单击标注作业名称进入标注作业标注详情页。
3. 在标注作业标注详情中，展示此标注作业下全部数据。

同步新数据

ModelArts会自动将数据集中新增的数据同步至标注作业，包含数据及当前标注作业支持的标注信息。

为了快速获取数据集中最新数据，可在标注作业详情页的“未标注”页签中，单击“同步新数据”，快速将数据集中的数据添加到标注作业中。

标注文本

标注作业详情页中，展示了此标注作业中“未标注”和“已标注”的文本，默认显示“未标注”的文本列表。

1. 在“未标注”页签文本列表中，页面左侧罗列“标注对象列表”。在列表中单击需标注的文本对象，选择右侧“标签集”中的标签进行标注。一个标注对象可添加多个标签。

   以此类推，不断选中标注对象，并为其添加标签。

图 3-35 文本分类标注

2. 当所有的标注对象都已完成标注，单击页面下方“保存当前页”，完成“未标注”列表的文本标注。

添加标签

- 在“未标注”页签添加：单击页面中标签集右侧的加号，然后在弹出的“新增标签”页中，添加标签名称，选择标签颜色，单击“确定”完成标签的新增。

图 3-36 添加标签（1）

- 在“已标注”页签添加：在右侧单击页面中全部标签加号，然后在弹出的“新增标签”页中，添加标签名称，选择标签颜色，单击“确定”完成标签的新增。

图 3-37 添加标签（2）

图 3-38 新增标签

查看已标注文本

在标注作业详情页，单击“已标注”页签，您可以查看已完成标注的文本列表。您也可以在右侧的“全部标签”中了解当前标注作业支持的所有标签信息。

修改标注

当数据完成标注后，您还可以进入已标注页签，对已标注的数据进行修改。

- **基于文本修改**

  在标注作业详情页，单击“已标注”页签，然后在文本列表中选中待修改的文本。

  在文本列表中，单击文本，当文本背景变为蓝色时，表示已选择。当文本有多个标签时，可以单击文本标签上方的×删除单个标签。

- **基于标签修改**

  在标注作业详情页，单击“已标注”页签，在图片列表右侧，显示全部标签的信息。

  – 批量修改：在“全部标签”区域中，单击操作列的编辑图标，然后在文本框中修改标签名称，选择标签颜色，单击“确定”完成修改。
  – 批量删除：在“全部标签”区域中，单击操作列的删除图标，在弹出对话框中，可选择“仅删除标签”或“删除标签及仅包含此标签的标注对象”，然后单击“确定”。

添加文件

除了同步新数据外，您还可以在标注详情页面中，直接添加数据，用于数据标注。

1. 在标注作业详情页面，单击“未标注”页签，然后单击左上角“添加数据”。
2. 在弹出的导入对话框中，选择数据来源、导入方式、导入路径等参数，导入数据。单击确定。

   导入数据的详细操作介绍请参见导入操作简介。

图 3-39 导入数据

删除文件

通过数据删除操作，可将需要丢弃的文件数据快速删除。

- 在“未标注”页面中，单击选中需要删除的文本对象，然后单击左上角“删除”，即可完成文本的删除操作。
- 在“已标注”页面中，选中待删除的文本对象，然后单击“删除”，删除单个文本。或者选择“选择当前页”选中该页面所有文本，然后单击左上角“删除”，即可完成当前页所有文本的删除操作。

其中，被选中的文本，其背景将显示为蓝色。

3.4.5 命名实体

命名实体场景，是针对文本中的实体片段进行标注，如“时间”、“地点”等。开始标注前，您需要了解：

实体命名标签名是由中文、大小写字母、数字、中划线、下划线或特殊符号组成，且不超过1024位的字符串。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据标注”，进入“数据标注”管理页面。
2. 在标注作业列表中，基于“标注类型”选择需要进行标注的标注作业，单击标注作业名称进入标注作业标注详情页。
3. 在标注作业标注详情中，展示此标注作业下全部数据。

同步新数据

ModelArts会自动将数据集中新增的数据同步至标注作业，包含数据及当前标注作业支持的标注信息。

为了快速获取数据集中最新数据，可在标注作业详情页的“未标注”页签中，单击“同步新数据”，快速将数据集中的数据添加到标注作业中。

标注文本

标注作业详情页中，展示了此标注作业中“未标注”和“已标注”的文本，默认显示“未标注”的文本列表。

1. 在“未标注”页签文本列表中，页面左侧罗列“标注对象列表”。在列表中单击需标注的文本对象，在右侧标签集下显示的文本内容中选中需要标注的部分，然后选择右侧“标签集”中的标签进行标注。一个标注对象可添加多个标签。

   以此类推，不断选中标注对象，并为其添加标签。

   图 3-40 命名实体标注

2. 单击页面下方“保存当前页”完成文本标注。

添加标签

- 在“未标注”页签添加：单击页面中标签集右侧的加号，然后在弹出的“新增标签”页中，添加标签名称，选择标签颜色，单击“确定”完成标签的新增。

  图 3-41 添加命名实体标签（1）

- 在“已标注”页签添加：在右侧单击页面中全部标签加号，然后在弹出的“新增标签”页中，添加标签名称，选择标签颜色，单击“确定”完成标签的新增。

  图 3-42 添加命名实体标签（2）

图 3-43 新增命名实体标签

查看已标注文本

在数据集详情页，单击“已标注”页签，您可以查看已完成标注的文本列表。您也可以在右侧的“全部标签”中了解当前数据集支持的所有标签信息。

修改标注

当数据完成标注后，您还可以进入“已标注”页签，对已标注的数据进行修改。

在数据集详情页，单击“已标注”页签，在右侧标签信息区域中对文本信息进行修改。

- **基于文本修改**

  在数据集详情页，单击“已标注”页签，然后在文本列表中选中待修改的文本。

  手工点选删除：在文本列表中，单击文本，当文本背景变为蓝色时，表示已选择。在页面右侧，单击文本标签上方的×删除单个标签。

- **基于标签修改**

  在数据集详情页，单击“已标注”页签，在图片列表右侧，显示全部标签的信息。

  – 批量修改：在“全部标签”区域中，单击操作列的编辑按钮，然后在文本框中添加标签名称，选择标签颜色，单击“确定”完成修改。
  – 批量删除：在“全部标签”区域中，单击操作列的删除按钮，在弹出对话框中，可选择“仅删除标签”或“删除标签及仅包含此标签的标注对象”，然后单击“确定”。

添加文件

除了同步新数据外，您还可以在标注详情页面中，直接添加数据，用于数据标注。

1. 在标注作业详情页面，单击“未标注”页签，然后单击左上角“添加数据”。
2. 在弹出的导入对话框中，选择数据来源、导入方式、导入路径等参数，导入数据。单击确定。

   导入数据的详细操作介绍请参见导入操作简介。

图 3-44 导入数据

删除文件

通过数据删除操作，可将需要丢弃的文件数据快速删除。

- 在“未标注”页面中，单击选中需要删除的文本对象，然后单击左上角“删除”，即可完成文本的删除操作。
- 在“已标注”页面中，选中待删除的文本对象，然后单击“删除”，删除单个文本。或者选择“选择当前页”选中该页面所有文本，然后单击左上角“删除”，即可完成当前页所有文本的删除操作。

其中，被选中的文本，其背景将显示为蓝色。

3.4.6 文本三元组

三元组标注适用于标注出语句当中形如（主语/Subject，谓词/Predicate，宾语/Object）结构化知识的场景，标注时不但可以标注出语句当中的实体，还可以标注出实体之间的关系，其在依存句法分析、信息抽取等自然语言处理任务中经常用到。

文本三元组类型的数据标注，需要关注两种标签，“实体标签”和“关系标签”。“关系标签”需设置对应的“起始实体”和“终止实体”。

- 支持设置多个“实体标签”和“关系标签”。一个文本数据中，也可以标注多个“实体标签”和“关系标签”
- 创建数据集时定义的“实体标签”，不支持删除。

注意事项

在开始标注之前，需确保标注作业对应的“实体标签”和“关系标签”已定义好。“关系标签”需设置对应的“起始实体”和“终止实体”。“关系标签”只能添加至其设置好的“起始实体”和“终止实体”之间。

例如，如图3-45所示，当两个文本都被标注为“地点”，那么针对这两个实体，无法添加本示例中的任意一个关系标签。当无法添加某个关系标签时，界面将显示一个红色的叉号，如图3-46所示。

图 3-45 实体标签和关系标签的示例

图 3-46 无法添加关系标签

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据标注”，进入“数据标注”管理页面。
2. 在标注作业列表中，基于“标注类型”选择需要进行标注的标注作业，单击标注作业名称进入标注作业标注详情页。
3. 在标注作业标注详情中，展示此标注作业下全部数据。

同步新数据

ModelArts会自动将数据集中新增的数据同步至标注作业，包含数据及当前标注作业支持的标注信息。

为了快速获取数据集中最新数据，可在标注作业详情页的“未标注”页签中，单击“同步新数据”，快速将数据集中的数据添加到标注作业中。

标注文本

标注作业详情页中，展示了此标注作业中“未标注”和“已标注”的文本，默认显示“未标注”的文本列表。

1. 在“未标注”页签文本列表中，页面左侧罗列“标注对象列表”。在列表中单击需标注的文本对象，选中相应文本内容，在页面呈现的实体类型列表中选择实体名称，完成实体标注。

图 3-47 实体标注

2. 在完成多个实体标注后，鼠标左键依次单击起始实体和终止实体，在呈现的关系类型列表中选择一个对应的关系类型，完成关系标注。

图 3-48 关系标注

3. 当所有的标注对象都已完成标注，单击页面下方“保存当前页”完成“未标注”列表的文本标注。

说明

“文本三元组”类型的数据集，不支持在标注页面修改标签，需要进入“标签管理”页面，修改“实体标签”和“关系标签”。

修改标注

当数据完成标注后，您还可以进入已标注页签，对已标注的数据进行修改。

在标注作业详情页，单击“已标注”页签，在左侧文本列表中选中一行文本，右侧区域显示具体的标注信息。将鼠标移动至对应的实体标签或关系类型，单击鼠标右键，可删除此标注。单击鼠标左键，依次单击连接起始实体和终止实体，可增加关系类型，增加关系标注。

图 3-49 在文本中修改标签

您也可以在单击页面下方的“删除当前项标签”按钮，删除选中文本对象中的所有标签。

图 3-50 删除当前项标签

添加文件

除了同步新数据外，您还可以在标注详情页面中，直接添加数据，用于数据标注。

1. 在标注作业详情页面，单击“未标注”页签，然后单击左上角“添加数据”。
2. 在弹出的导入对话框中，选择数据来源、导入方式、导入路径等参数，导入数据。单击确定。

   导入数据的详细操作介绍请参见导入操作简介。

图 3-51 导入数据

删除文件

通过数据删除操作，可将需要丢弃的文件数据快速删除。

- 在“未标注”页面中，单击选中需要删除的文本，然后单击左上角“删除”，即可完成文本的删除操作。
- 在“已标注”页面中，选中待删除的文本，然后单击“删除”，删除单个文本。或者勾选“选择当前页”选中该页面所有文本，然后单击左上角“删除”，即可完成当前页所有文本的删除操作。

其中，被选中的文本，其背景将显示为蓝色。如果当前页面无选中文本时，“删除”按钮为灰色，无法执行删除操作。

3.4.7 声音分类

由于模型训练过程需要大量有标签的音频数据，因此在模型训练之前需对没有标签的音频添加标签。通过ModelArts您可对音频进行一键式批量添加标签，快速完成对音频的标注操作，也可以对已标注音频修改或删除标签进行重新标注。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据标注”，进入“数据标注”管理页面。
2. 在标注作业列表中，基于“标注类型”选择需要进行标注的标注作业，单击标注作业名称进入标注作业标注详情页。
3. 在标注作业标注详情中，展示此标注作业下全部数据。

同步新数据

ModelArts会自动将数据集中新增的数据同步至标注作业，包含数据及当前标注作业支持的标注信息。

为了快速获取数据集中最新数据，可在标注作业详情页的“未标注”和“已标注”页签中，单击“同步新数据”，快速将数据集中的数据添加到标注作业中。

标注音频

标注作业详情页中，展示了此标注作业中“未标注”和“已标注”的音频，默认显示“未标注”的音频列表。单击音频左侧 ，即可进行音频的试听。

1. 在“未标注”页签，勾选需进行标注的音频。
   - 手工点选：在音频列表中，单击音频，当右上角出现蓝色勾选框时，表示已勾选。可勾选同类别的多个音频，一起添加标签。
   - 批量选中：如果音频列表的当前页，所有音频属于一种类型，可以在列表的右上角单击“选择当前页”，则当前页面所有的音频将选中。
2. 添加标签。
   a. 在右侧的“标签”区域中，单击“标签”下侧的文本框中设置标签。

      方式一（已存在标签）：单击“标签”下方的文本框，在快捷键下拉列表中选择快捷键，然后在标签文本输入框中选择已有的标签名称，然后单击“确定”。

      方式二（新增标签）：在“标签”下方的文本框中，在快捷键下拉列表中选择快捷键，然后在标签文本输入框中输入新的标签名称，然后单击“确定”。

   b. 选中的音频将被自动移动至“已标注”页签，且在“未标注”页签中，标签的信息也将随着标注步骤进行更新，如增加的标签名称、各标签对应的音频数量。

说明

快捷键的使用说明：为标签指定快捷键后，当您选择一段音频后，在键盘中按一下快捷键，即可为此音频增加为此快捷键对应的标签。例如“aa”标签对应的快捷键是“1”，在数据标注过程中，选中1个或多个文件，按“1”，界面将提示是否需要将此文件标注为“aa”标签，单击确认即可完成标注。

快捷键对应的是标签，1个标签对应1个快捷键。不同的标签，不能指定为同一个快捷键。快捷键的使用，可以大大提升标注效率。

图 3-52 添加音频标签

查看已标注音频

在标注作业详情页，单击“已标注”页签，您可以查看已完成标注的音频列表。单击音频，可在右侧的“选中文件标签”中了解当前音频的标签信息。

修改标注

当数据完成标注后，您还可以进入“已标注”页签，对已标注的数据进行修改。

- **基于音频修改**

  在标注作业详情页面，单击“已标注”页签，然后在音频列表中选中待修改的音频（选择一个或多个）。在右侧标签信息区域中对标签进行修改。

  - 修改标签：在“选中文件标签”区域中，单击操作列的编辑图标，然后在文本框中输入正确的标签名，然后单击确定图标完成修改。
  - 删除标签：在“选中文件标签”区域中，单击操作列的删除图标删除该标签。

- **基于标签修改**

  在标注作业详情页面，单击“已标注”页签，在音频列表右侧，显示全部标签的信息。

  图 3-53 全部标签信息

  - 修改标签：单击操作列的编辑图标，然后在弹出的对话框中输入修改后的标签名，然后单击“确定”完成修改。修改后，之前添加了此标签的音频，都将被标注为新的标签名称。

– 删除标签：单击操作列的删除图标，在弹出的对话框中，根据提示框选择需要删除的对象，然后单击“确定”完成删除。

添加音频

除了同步新数据外，您还可以在标注详情页面中，直接添加数据，用于数据标注。

1. 在标注作业详情页面，单击“未标注”页签，然后单击左上角“添加数据”。
2. 在弹出的导入对话框中，选择数据来源、导入方式、导入路径等参数，导入数据。单击确定。

   导入数据的详细操作介绍请参见导入操作简介。

图 3-54 导入数据

删除音频

通过数据删除操作，可将需要丢弃的音频数据快速删除。

在“未标注”或“已标注”页面中，选中需要删除的音频，或者选择“选择当前页”选中该页面所有音频，然后单击左上角“删除音频”，在弹出的对话框中，根据实际情况选择是否勾选“同时删除源文件”，确认信息无误后，单击“确定”完成音频删除操作。

其中，被选中的音频，其右上角将显示为勾选状态。如果当前页面无选中音频时，“删除音频”按钮为灰色，无法执行删除操作。

说明

如果勾选了“同时删除源文件”，删除音频操作是将删除对应OBS目录下存储的音频。此操作可能会影响已使用此源文件的其他数据集或数据集版本，有可能导致展示异常或训练/推理异常。删除后，数据将无法恢复，请谨慎操作。

3.4.8 语音内容

由于模型训练过程需要大量有标签的音频数据，因此在模型训练之前需对没有标签的音频添加标签。通过ModelArts您可对音频进行一键式批量添加标签，快速完成对音频的标注操作，也可以对已标注音频修改或删除标签进行重新标注。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据标注”，进入“数据标注”管理页面。
2. 在标注作业列表中，基于“标注类型”选择需要进行标注的标注作业，单击标注作业名称进入标注作业标注详情页。
3. 在标注作业标注详情中，展示此标注作业下全部数据。

同步新数据

ModelArts会自动将数据集中新增的数据同步至标注作业，包含数据及当前标注作业支持的标注信息。

为了快速获取数据集中最新数据，可在标注作业详情页的“未标注”页签中，单击“同步新数据”，快速将数据集中的数据添加到标注作业中。

标注音频

标注作业详情页中，展示了此数据集中“未标注”和“已标注”的音频，默认显示“未标注”的音频列表。

1. 在“未标注”页签左侧音频列表中，单击目标音频文件，在右侧的区域中出现音频，单击音频下方，即可进行音频播放。
2. 根据播放内容，在下方“语音内容”文本框中填写音频内容。
3. 输入内容后单击下方的“确认标注”按钮完成标注。音频将被自动移动至“已标注”页签。

图 3-55 语音内容音频标注

查看已标注音频

在标注作业详情页，单击“已标注”页签，您可以查看已完成标注的音频列表。单击音频，可在右侧的“语音内容”文本框中了解当前音频的内容信息。

修改标注

当数据完成标注后，您还可以进入“已标注”页签，对已标注的数据进行修改。

在标注作业详情页，单击“已标注”页签，然后在音频列表中选中待修改的音频。在右侧标签信息区域中修改“语音内容”文本框中的内容，单击下方的“确认标注”按钮完成修改。

添加音频

除了同步新数据外，您还可以在标注详情页面中，直接添加数据，用于数据标注。

1. 在标注作业详情页面，单击“未标注”页签，然后单击左上角“添加数据”。
2. 在弹出的导入对话框中，选择数据来源、导入方式、导入路径等参数，导入数据。单击确定。

   导入数据的详细操作介绍请参见导入操作简介。

图 3-56 导入数据

删除音频

通过数据删除操作，可将需要丢弃的音频数据快速删除。

在“未标注”或“已标注”页面中，选中需要删除的音频，然后单击左上角“删除音频”，在弹出的对话框中，根据实际情况选择是否勾选“同时删除源文件”，确认信息无误后，单击“确定”完成音频删除操作。

说明

如果勾选了“同时删除源文件”，删除音频操作是将删除对应OBS目录下存储的音频。此操作可能会影响已使用此源文件的其他数据集或数据集版本，有可能导致展示异常或训练/推理异常。删除后，数据将无法恢复，请谨慎操作。

3.4.9 语音分割

由于模型训练过程需要大量有标签的音频数据，因此在模型训练之前需对没有标签的音频添加标签。通过ModelArts您可对音频添加标签，快速完成对音频的标注操作，也可以对已标注音频修改或删除标签进行重新标注。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据标注”，进入“数据标注”管理页面。
2. 在标注作业列表中，基于“标注类型”选择需要进行标注的标注作业，单击标注作业名称进入标注作业标注详情页。
3. 在标注作业标注详情中，展示此标注作业下全部数据。

同步数据源

ModelArts会自动从数据集输入位置同步数据至数据集详情页，包含数据及标注信息。

为了快速获取OBS桶中最新数据，可在数据集详情页的“未标注”页签中，单击“同步数据源”，快速将通过OBS上传的数据添加到数据集中。

标注音频

标注作业详情页中，展示了此标注作业中“未标注”和“已标注”的音频，默认显示“未标注”的音频列表。

1. 在“未标注”页签左侧音频列表中，单击目标音频文件，在右侧的区域中出现音频，单击音频下方，即可进行音频播放。
2. 根据播放内容，选取合适的音频段，在下方“语音内容”文本框中填写音频标签和内容。

图 3-57 语音标签音频标注

3. 输入内容后单击下方的“确认标注”按钮完成标注。音频将被自动移动至“已标注”页签。

查看已标注音频

在标注作业详情页，单击“已标注”页签，您可以查看已完成标注的音频列表。单击音频，可在右侧的“语音内容”文本框中了解当前音频的内容信息。

修改标注

当数据完成标注后，您还可以进入“已标注”页签，对已标注的数据进行修改。

- 修改标签：在数据集详情页，单击“已标注”页签，然后在音频列表中选中待修改的音频。在右侧标签信息区域中修改“语音内容”中的“标签”和“内容”，单击下方的“确认标注”按钮完成修改。
- 删除标签：单击目标编号操作列的 ，删除该段音频的标注。您也可以单击标注音频文件上方的叉号删除标注，然后单击“确认标注”。

添加音频

除了同步新数据外，您还可以在标注详情页面中，直接添加数据，用于数据标注。

1. 在标注作业详情页面，单击“未标注”页签，然后单击左上角“添加数据”。
2. 在弹出的导入对话框中，选择数据来源、导入方式、导入路径等参数，导入数据。单击确定。

   导入数据的详细操作介绍请参见**OBS导入数据**。

图 3-58 导入数据

删除音频

通过数据删除操作，可将需要丢弃的音频数据快速删除。

在“未标注”或“已标注”页面中，选中需要删除的音频，然后单击左上角“删除音频”，在弹出的对话框中，根据实际情况选择是否勾选“同时删除源文件”，确认信息无误后，单击“确定”完成音频删除操作。

说明

如果勾选了“同时删除源文件”，删除音频操作是将删除对应OBS目录下存储的音频。此操作可能会影响已使用此源文件的其他数据集或数据集版本，有可能导致展示异常或训练/推理异常。删除后，数据将无法恢复，请谨慎操作。

3.4.10 视频标注

由于模型训练过程需要大量有标签的视频数据，因此在模型训练之前需对没有标签的视频添加标签。通过ModelArts您可对视频添加标签，快速完成对视频的标注操作，也可以对已标注视频修改或删除标签进行重新标注。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据标注”，进入“数据标注”管理页面。
2. 在标注作业列表中，基于“标注类型”选择需要进行标注的标注作业，单击标注作业名称进入标注作业标注详情页。
3. 在标注作业标注详情中，展示此标注作业下全部数据。

同步数据源

ModelArts会自动从数据集输入位置同步数据至数据集详情页，包含数据及标注信息。

为了快速获取OBS桶中最新数据，可在数据集详情页的“全部”或“未标注”页签中，单击“同步数据源”，快速将通过OBS上传的数据添加到数据集中。

视频标注

标注作业详情页中，展示了此数据集中“未标注”和“已标注”的视频。

1. 在“未标注”页签左侧视频列表中，单击目标视频文件，打开标注页面。
2. 在标注页面中，播放视频，当视频播放至待标注时间时，单击进度条左侧的暂停按钮，将视频暂停至某一画面。
3. 在上方区域选择标注框，默认为矩形框。使用鼠标在视频画面中框出目标，然后在弹出的添加标签文本框中，直接输入新的标签名，在文本框前面选中标签颜色，单击“添加”完成1个物体的标注。如果已存在标签，从下拉列表中选择已有的标签，然后单击“添加”完成标注。逐步此画面中所有物体所在位置，一张画面可添加多个标签。

   支持的标注框与“物体检测”类型一致，详细描述请参见物体检测章节的**表3-10**。

图 3-59 视频标注

4. 上一个暂停画面标注完成后，在进度条处单击播放按钮继续播放，在需要标注的画面中暂停，然后重复执行步骤3完成整个视频的标注。

   单击界面右上角的“标签类别”，在“标签类别”的详情页将呈现当前视频带标注的时间点。

   图 3-60 当前文件标签信息

5. 单击页面左上角“返回数据标注预览”，页面将自动返回标注作业详情页面，同时，标注好的视频将呈现在“已标注”页签下。

修改标注

当数据完成标注后，您还可以进入“已标注”页签，删除标注数据。

在“已标注”页签下，单击目标视频文件，在标注页面右侧的“当前文件标签”下，可单击时间点右侧小三角展开详情，您可以修改或删除标签。

- 修改标签：单击标签右侧的编辑按钮，标签名称可进行修改。
- 删除标签：单击标签右侧的删除按钮，将直接删除此标签。如果单击画面时间右侧的删除按钮，将删除此画面下的所有标签。

图 3-61 修改标注

添加视频

除了同步新数据外，您还可以在标注详情页面中，直接添加数据，用于数据标注。

1. 在标注作业详情页面，单击“未标注”页签，然后单击左上角“添加数据”。
2. 在弹出的导入对话框中，选择数据来源、导入方式、导入路径等参数，导入数据。单击确定。

   导入数据的详细操作介绍请参见**OBS导入数据**。

图 3-62 导入数据

删除视频

通过数据删除操作，可将需要丢弃的视频数据快速删除。

在“全部”、“未标注”或“已标注”页面中，依次选中需要删除的视频，或者选择“选择当前页”选中该页面所有视频，然后单击左上角“删除视频”。在弹出的对话框中，根据实际情况选择是否勾选“同时删除源文件”，确认信息无误后，单击“确定”完成视频删除操作。

其中，被选中的视频，其左上角将显示为勾选状态。如果当前页面无选中视频时，“删除视频”按钮为灰色，无法执行删除操作。

说明

如果勾选了“同时删除源文件”，删除视频操作将删除对应OBS目录下存储的视频，此操作可能会影响已使用此源文件的其他数据集或数据集版本，有可能导致展示异常或训练/推理异常。删除后，数据将无法恢复，请谨慎操作。

3.5 导入数据

3.5.1 导入操作简介

前提条件

- 已存在创建完成的数据集。
- 至少已创建了一个OBS桶，用于数据存储，确保OBS桶与ModelArts在同一区域，并确保用户具有OBS桶的操作权限。

数据集中的数据导入入口

数据集中的数据导入有5个入口。

- 创建数据集时直接从设置的数据导入路径中自动同步数据。

图 3-63 创建数据集时导入数据

- 创建完数据集后，在数据集列表页面的操作栏单击“导入”，导入数据。

图 3-64 在数据集列表页导入数据

- 在数据集列表页面，单击某个数据集的名称，进入数据集详情页中，单击“导入”，导入数据。

图 3-65 在数据集详情页中导入数据

- 在数据集列表页面，单击某个数据集的名称，进入数据集详情页中，单击“同步数据源”，同步OBS中的数据。

图 3-66 在数据集详情页中同步数据源

- 在数据标注的标注作业详情中添加数据。

图 3-67 标注作业详情中添加数据

文件型数据来源

文件型数据集的导入支持两种数据来源：“OBS”和“本地上传”。导入后，导入目录下的数据会拷贝至数据集的数据源路径下。

- OBS：需要将导入的数据或Manifest文件提前存储至OBS目录中。
- 本地上传：将本地数据直接通过Internet上传至OBS指定目录后，再导入数据集。

文件型数据导入方式

导入方式分为“目录”和“manifest”两种。

- 目录：导入路径选择OBS目录，读取OBS目录下的数据并导入数据集。
- Manifest：按Manifest文件读取OBS上的数据并导入，需要满足Manifest的文件规范，同时每个Manifest文件仅包含一种标注类型。Manifest文件定义了标注对象和标注内容的对应关系。文件规范请参见**导入Manifest文件的规范说明**。当前“自由格式”的数据集不支持Manifest导入。

文件型数据标注状态

数据标注状态分为“未标注”和“已标注”。

- 未标注：仅导入标注对象（指待标注的图片，文本等），不导入标注内容（指标注结果信息）。
- 已标注：同时导入标注对象和标注内容，当前“自由格式”的数据集不支持导入标注内容。

  为了确保能够正确读取标注内容，要求用户严格按照规范存放数据：

  导入方式选择目录时，需要用户选择“标注格式”，并按照标注格式的要求存放数据，详细规范请参见**标注格式**章节。

  导入方式选择manifest时，需要满足manifest文件的规范。

  说明

  数据标注状态选择“已标注”，您需要保证目录或manifest文件满足相应的格式规范，否则可能存在导入失败的情况。

文件型数据标注格式

标注格式指标注对象和标注内容按照特定的格式存放，用于已标注数据的目录导入。不同类型的数据集支持不同的标注格式，用户需要根据数据集类型和标注类型，选择对应的标注格式。当前只有“图像分类”、“物体检测”、“图像分割”、“文本分类”和“声音分类”标注类型支持按标注格式导入。

表 3-14 不同类型数据集支持标注格式

数据集类型	标注类型	标注格式
图片	图像分类	1）ModelArts image classification 1.0 2）ModelArts imageNet 1.0 格式规范：**图像分类**
	物体检测	1) ModelArts PASCAL VOC 1.0 2) YOLO 格式规范：**物体检测**

数据集类型	标注类型	标注格式
图片	图像分割	ModelArts image segmentation 1.0 格式规范：图像分割
音频	声音分类	支持的标注格式： ModelArts audio classfication dir 1.0 格式规范：声音分类
文本	文本分类	1）ModelArts text classfication 1.0 2）ModelArts text classfication combine 1.0 格式规范：文本分类

文件型数据从 OBS 目录导入操作

不同类型的数据集，导入操作界面的示意图存在区别，请参考界面信息了解当前类型数据集的示意图。当前操作指导以图像分类的数据集为例。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理 >数据集”，进入“数据集”管理页面。
2. 在数据集所在行，单击操作列的“导入”。或者，您可以单击数据集名称，进入数据集“概览”页，在页面右上角单击“导入”。
3. 在“导入”对话框中，参考如下说明填写参数，然后单击“确定”。
   - “数据来源”：“OBS”
   - “导入方式”：“目录”。
   - “导入路径”：数据存储的OBS路径。
   - “数据标注状态”：已标注。
   - “高级特征选项 ”：默认关闭，可通过勾选高级选项提供增强功能。

     如“按标签导入”：系统将自动获取此数据集的标签，您可以单击“添加标签”添加相应的标签。此字段为可选字段，您也可以在导入数据集后，在标注数据操作时，添加或删除标签。

图 3-68 导入数据集-OBS

导入成功后，数据将自动同步到数据集中。您可以在“数据集”页面，单击数据集的名称，查看详细数据，并可以通过创建标注任务进行数据标注。

文件型数据从 Manifest 导入操作

不同类型的数据集，导入操作界面的示意图存在区别，请参考界面信息了解当前类型数据集的示意图。当前操作指导以图片数据集为例。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集所在行，单击操作列的“导入”。或者，您可以单击数据集名称，进入数据集“概览”页，在页面右上角单击“导入”。
3. 在“导入”对话框中，参考如下说明填写参数，然后单击“确定”。
   - “数据来源”：“OBS”
   - “导入方式”：“manifest”。
   - “Manifest文件”：存储Manifest文件的OBS路径。
   - “数据标注状态”：已标注。
   - “高级特征选项 ”：默认关闭，可通过勾选高级选项提供增强功能。

“按标签导入”：系统将自动获取此数据集的标签，您可以单击“添加标签”添加。此字段为可选字段，您也可以在导入数据集后，在标注数据操作时，添加或删除标签。

“只导入难例”：难例指manifest文件中的“hard”属性，勾选此参数，表示此导入操作，只导入manifest文件“hard”属性中数据信息。

图 3-69 导入 mainfest 文件

导入成功后，数据将自动同步到数据集中。您可以在“数据集”页面，单击数据集的名称，查看详细数据，并可以通过创建标注任务进行数据标注。

表格数据集的导入

表格支持从4种数据源导入数据，分别为对象存储服务（OBS）、数据仓库服务（DWS）、数据湖探索服务（DLI）、MapReduce服务（MRS）和本地上传。

导入说明：

- 导入成功的前提是，数据源的schema需要与创建数据集指定的schema保持一致。其中schema指表格的列名和类型，创建数据集时一旦指定，不支持修改。
- 数据格式不合法，会将数据置为null，详见**表3-5**。
- 从OBS或者MRS导入csv文件，不会校验数据类型，但是列数需要跟数据集的schema保持一致。

下面分别介绍如下几种数据源导入：

- 从OBS导入数据

  支持从OBS导入csv文件，需要选择文件所在目录，其中csv文件的列数需要跟数据集schema一致。支持自动获取csv文件的schema。

```
├──dataset-import-example
│      table_import_1.csv
│      table_import_2.csv
```

```
|     table_import_3.csv
|     table_import_4.csv
```

说明

针对“表格”类型的数据集，导入时，支持从OBS、DWS、DLI、MRS和本地上传类型的数据源。其导入时的设置和数据要求，与创建数据集相同，详细参数可参见创建数据集时**表格型（表格）**类型的参数说明。

- 从DWS导入数据

  从DWS导入数据，用户需要选择对应的DWS集群，并输入需要对应的数据库名、表名以及用户名和密码。所导入表的schema(列名和类型)需要跟数据集相同。

- 从DLI导入数据

  从DLI导入数据，用户需要选择DLI队列、数据库和表名称。所选择的表的schema(列名和类型)需与数据集一致，支持自动获取所选择表的schema。DLI的default队列只用作体验，不同帐号间可能会出现抢占的情况，需进行资源排队，不能保证每次都可以得到资源执行相关操作。DLI支持schema映射的功能，即导入的表的schema的字段名称可以不和数据集相同，但类型要保持一致。

- 从MRS导入数据

  只支持从分析集群导入数据，流式集群不支持导入。从MRS服务中导入存储在HDFS上的csv格式的数据，首先需要选择已有的MRS集群，并从HDFS文件列表选择文件名称或所在目录，导入文件的列数需与数据集schema一致。

本地上传

文件型和表格型数据均支持从本地上传。从本地上传的数据存储在OBS目录中，请先提前创建OBS桶。

从本地上传的数据单次最多支持100个文件同时上传，总大小不超过5GB。

图 3-70 从本地上传数据

3.5.2 OBS 目录导入数据规范说明

导入数据集时，使用存储在OBS的数据时，数据的存储目录以及文件名称需满足ModelArts的规范要求。

当前只有“图像分类”、“物体检测”、“图像分割”、“文本分类”和“声音分类”标注类型支持按标注格式导入。

其中，“表格”类型的数据集，支持从OBS、DWS、DLI和MRS等数据源导入数据。

说明

从OBS目录导入数据时，当前操作用户需具备此OBS路径的读取权限。

图像分类

图像分类的数据支持两种格式：

1）ModelArts imageNet 1.0：目录方式，只支持单标签

- 相同标签的图片放在一个目录里，并且目录名字即为标签名。当存在多层目录时，则以最后一层目录为标签名。

  示例如下所示，其中Cat和Dog分别为标签名。

```
dataset-import-example
├──Cat
│      10.jpg
│      11.jpg
│      12.jpg
│
└──Dog
      1.jpg
      2.jpg
      3.jpg
```

2）ModelArts image classification 1.0：txt标签文件，支持多标签

- 当目录下存在对应的txt文件时，以txt文件内容作为图像的标签。

  示例如下所示，import-dir-1和imort-dir-2为导入子目录。

```
dataset-import-example
├──import-dir-1
│      10.jpg
│      10.txt
│      11.jpg
│      11.txt
│      12.jpg
│      12.txt
└──import-dir-2
      1.jpg
      1.txt
      2.jpg
      2.txt
```

  单标签的标签文件示例，如1.txt文件内容如下所示：

```
Cat
```

  多标签的标签文件示例，如1.txt文件内容如下所示：

```
Cat
Dog
```

- 只支持JPG、JPEG、PNG、BMP格式的图片。单张图片大小不能超过5MB，且单次上传的图片总大小不能超过8MB。

物体检测

支持两种格式：

1)ModelArts PASCAL VOC 1.0

- 物体检测的简易模式要求用户将标注对象和标注文件存储在同一目录，并且一一对应，如标注对象文件名为“IMG_20180919_114745.jpg”，那么标注文件的文件名应为“IMG_20180919_114745.xml”。

  物体检测的标注文件需要满足PASCAL VOC格式，格式详细说明请参见**表3-22**。

  示例：

```
├──dataset-import-example
│      IMG_20180919_114732.jpg
│      IMG_20180919_114732.xml
│      IMG_20180919_114745.jpg
│      IMG_20180919_114745.xml
│      IMG_20180919_114945.jpg
│      IMG_20180919_114945.xml
```

  标注文件的示例如下所示：

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<annotation>
   <folder>NA</folder>
   <filename>bike_1_1593531469339.png</filename>
   <source>
      <database>Unknown</database>
   </source>
   <size>
      <width>554</width>
      <height>606</height>
      <depth>3</depth>
   </size>
   <segmented>0</segmented>
   <object>
      <name>Dog</name>
      <pose>Unspecified</pose>
```

```
        <truncated>0</truncated>
        <difficult>0</difficult>
        <occluded>0</occluded>
        <bndbox>
            <xmin>279</xmin>
            <ymin>52</ymin>
            <xmax>474</xmax>
            <ymax>278</ymax>
        </bndbox>
    </object>
    <object>
        <name>Cat</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <occluded>0</occluded>
        <bndbox>
            <xmin>279</xmin>
            <ymin>198</ymin>
            <xmax>456</xmax>
            <ymax>421</ymax>
        </bndbox>
    </object>
</annotation>
```

- 只支持JPG、JPEG、PNG、BMP格式的图片，单张图片大小不能超过5MB，且单次上传的图片总大小不能超过8MB。

2）YOLO：

- YOLO数据集目录应该遵循以下结构，格式不规范将导致导入任务失败。

```
└── yolo_dataset/
    |
    ├──── obj.names  # 标签集文件
    ├──── obj.data   # 记录数据集信息的文件及路径信息(相对路径)
    ├──── train.txt  # 训练集中各图片路径信息(相对路径)
    ├──── valid.txt  # 验证集中各图片路径信息(相对路径)
    |
    ├──── obj_train_data/  # 训练集的图片与对应的标注文件所在目录
    |     ├──── image1.txt  # image1对应的带标签bbox列表
    |     ├──── image1.jpg
    |     ├──── image2.txt
    |     ├──── image2.jpg
    |     ├──── ...
    |
    ├──── obj_valid_data/  # 验证集的图片与对应的标注文件所在目录
    |     ├──── image101.txt
    |     ├──── image101.jpg
    |     ├──── image102.txt
    |     ├──── image102.jpg
    |     ├──── ...
```

  YOLO数据集只支持train和valid子集。如果导入的数据集包括除了上述之外的子集，这些其他子集将被忽略。

- obj.data应包含以下内容，train和valid子集必许至少有一个，其中文件路径均为相对路径。

```
classes = 5 # 可选
names = <path/to/obj.names>
train = <path/to/train.txt>
valid = <path/to/valid.txt>
backup = backup/ # 可选
```

- obj.names文件记录标签列表。每一行的行标作为该标签的index。

```
label1  # label1的index：  0
label2  # label2的index：  1
label3
...
```

- train.txt和valid.txt文件结构如下，其中文件路径均为相对路径，且文件列表与目录下的文件需一一对应：

```
<path/to/image1.jpg>
<path/to/image2.jpg>
...
```

- obj_train_data/ 和 obj_valid_data/目录下的.txt文件应该包含对应图像的bbox标签信息，每行代表一个bbox标注。

```
# image1.txt:
# <label_index> <x_center> <y_center> <width> <height>
0 0.250000 0.400000 0.300000 0.400000
3 0.600000 0.400000 0.400000 0.266667
```

  其中x_center, y_center, width, and height分别表示归一化后的目标框中心点x坐标、归一化后的目标框中心点y坐标、归一化后的目标框宽度、归一化后的目标框高度。

- 只支持JPG、JPEG、PNG、BMP格式的图片，单张图片大小不能超过5MB，且单次上传的图片总大小不能超过8MB。

图像分割

ModelArts image segmentation 1.0：

- 要求用户将标注对象和标注文件存储在同一目录，并且一一对应，如标注对象文件名为“IMG_20180919_114746.jpg”，那么标注文件的文件名应为“IMG_20180919_114746.xml”。

  图像分割的标注文件基于PASCAL VOC格式增加了字段mask_source和mask_color，格式详细说明请参见**表3-18**。

  示例：

```
├──dataset-import-example
│      IMG_20180919_114732.jpg
│      IMG_20180919_114732.xml
│      IMG_20180919_114745.jpg
│      IMG_20180919_114745.xml
│      IMG_20180919_114945.jpg
│      IMG_20180919_114945.xml
```

  标注文件的示例如下所示：

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<annotation>
   <folder>NA</folder>
   <filename>image_0006.jpg</filename>
   <source>
      <database>Unknown</database>
   </source>
   <size>
      <width>230</width>
      <height>300</height>
      <depth>3</depth>
   </size>
   <segmented>1</segmented>
   <mask_source>obs://xianao/out/dataset-8153-Jmf5ylLjRmSacj9KevS/annotation/V001/
segmentationClassRaw/image_0006.png</mask_source>
   <object>
      <name>bike</name>
      <pose>Unspecified</pose>
      <truncated>0</truncated>
      <difficult>0</difficult>
      <mask_color>193,243,53</mask_color>
      <occluded>0</occluded>
      <polygon>
         <x1>71</x1>
         <y1>48</y1>
```

```
            <x2>75</x2>
            <y2>73</y2>
            <x3>49</x3>
            <y3>69</y3>
            <x4>68</x4>
            <y4>92</y4>
            <x5>90</x5>
            <y5>101</y5>
            <x6>45</x6>
            <y6>110</y6>
            <x7>71</x7>
            <y7>48</y7>
        </polygon>
    </object>
</annotation>
```

文本分类

文本分类支持导入“txt”和“csv”两种文件类型，文本的编码格式支持“UTF-8”和“GBK”。

文本分类的标注对象和标注文件有2种存放模式。

- ModelArts text classfication combine 1.0：文本和标注合并，文本分类的标注对象和标注内容在一个文本文件内，标注对象与标注内容之间，多个标注内容之间可分别指定分隔符。

  例如，文本文件的内容如下所示。标注对象与标注内容之间采用tab键分隔。

  ```
  手感很好，反应速度很快，不知道以后怎样	positive
  三个月前买了一个用的非常好果断把旧手机替换下来尤其在待机方面秒杀	positive
  没充一会电源怎么也会发热呢音量健不好用回弹不好	negative
  算是给自己的父亲节礼物吧物流很快下单不到24小时就到货了耳机更赞有些低音炮的感觉入耳很紧不会掉棒棒哒	positive
  ```

- ModelArts text classfication 1.0：文本和标注分离，文本分类的标注对象和标注文件均为文本文件，并且以行数进行对应，如标注文件中的第一行表示的是标注对象文件中的第一行的标注。

  例如，标注对象“COMMENTS_20180919_114745.txt”的内容如下所示。

  ```
  手感很好，反应速度很快，不知道以后怎样
  三个月前买了一个用的非常好果断把旧手机替换下来尤其在待机方面秒杀
  没充一会电源怎么也会发热呢音量健不好用回弹不好
  算是给自己的父亲节礼物吧物流很快下单不到24小时就到货了耳机更赞有些低音炮的感觉入耳很紧不会掉棒棒哒
  ```

  标注文件“COMMENTS_20180919_114745_result.txt”的内容。

  ```
  positive
  negative
  negative
  positive
  ```

  此数据格式要求将标注对象和标注文件存储在同一目录，并且一一对应，如标注对象文件名为“COMMENTS_20180919_114745.txt”，那么标注文件名为“COMMENTS _20180919_114745_result.txt”。

  数据文件存储示例：

  ```
  ├──dataset-import-example
  │      COMMENTS_20180919_114732.txt
  │      COMMENTS _20180919_114732_result.txt
  │      COMMENTS _20180919_114745.txt
  │      COMMENTS _20180919_114745_result.txt
  │      COMMENTS _20180919_114945.txt
  │      COMMENTS _20180919_114945_result.txt
  ```

声音分类

ModelArts audio classfication dir 1.0：要求用户将相同标签的声音文件放在一个目录里，并且目录名字即为标签名。

示例：

```
dataset-import-example
├──Cat
│      10.wav
│      11.wav
│      12.wav
│
└──Dog
      1.wav
      2.wav
      3.wav
```

表格

支持从OBS导入csv文件，需要选择文件所在目录，其中csv文件的列数需要跟数据集schema一致。支持自动获取csv文件的schema。

```
├──dataset-import-example
│      table_import_1.csv
│      table_import_2.csv
│      table_import_3.csv
│      table_import_4.csv
```

3.5.3 Manifest 文件导入规范说明

Manifest文件中定义了标注对象和标注内容的对应关系。此导入方式是指导入数据集时，使用Manifest文件。选择导入Manifest文件时，可以从OBS导入。当从OBS导入Manifest文件时，需确保当前用户具备Manifest文件所在OBS路径的权限。

说明

Manifest文件编写规范要求较多，推荐使用OBS目录导入方式导入新数据。一般此功能常用于不同区域或不同帐号下ModelArts的数据迁移，即当您已在某一区域使用ModelArts完成数据标注，发布后的数据集可从输出路径下获得其对应的Manifest文件。在获取此Manifest文件后，可将此数据集导入其他区域或者其他帐号的ModelArts中，导入后的数据已携带标注信息，无需重复标注，提升开发效率。

Manifest文件描述的是原始文件和标注信息，可用于标注、训练、推理场景。Manifest文件中也可以只有原始文件信息，没有标注信息，如用于推理场景，或用于生成未标注的数据集。Manifest文件需满足如下要求：

- Manifest文件使用UTF-8编码。文本分类的source数值可以包含中文，其他字段不建议使用中文。
- Manifest文件使用json lines格式（jsonlines.org），一行一个json对象。

  ```
  {"source": "/path/to/image1.jpg", "annotation": … }
  {"source": "/path/to/image2.jpg", "annotation": … }
  {"source": "/path/to/image3.jpg", "annotation": … }
  ```

  为了说明方便，下面的Manifest例子格式化为多行的json对象。
- Manifest文件可以由用户、第三方工具或ModelArts数据标注生成，其文件名没有特殊要求，可以为任意合法文件名。为了ModelArts系统内部使用方便，ModelArts数据标注功能生成的文件名由如下字符串组成：“DatasetName-VersionName.manifest”。例如，“animal-v20190123113030412 3.manifest”。

图像分类

```
{
   "source":"s3://path/to/image1.jpg",
   "usage":"TRAIN",
   "hard":"true",
   "hard-coefficient":0.8,
   "id":"0162005993f8065ef47eefb59d1e4970",
   "annotation": [
      {
         "type": "modelarts/image_classification",
         "name": "cat",
         "property": {
            "color":"white",
            "kind":"Persian cat"
         },
         "hard":"true",
         "hard-coefficient":0.8,
         "annotated-by":"human",
         "creation-time":"2019-01-23 11:30:30"
      },
      {
         "type": "modelarts/image_classification",
         "name":"animal",
         "annotated-by":"modelarts/active-learning",
         "confidence": 0.8,
         "creation-time":"2019-01-23 11:30:30"
      }],
   "inference-loc":"/path/to/inference-output"
}
```

表 3-15 字段说明

字段	是否必选	说明
source	是	被标注对象的URI。数据来源的类型及示例请参考**表3-16**。
usage	否	默认为空，取值范围： ● TRAIN：指明该对象用于训练。 ● EVAL：指明该对象用于评估。 ● TEST：指明该对象用于测试。 ● INFERENCE：指明该对象用于推理。 如果没有给出该字段，则使用者自行决定如何使用该对象。
id	否	此参数为系统导出的样本id，导入时可以不用填写。
annotation	否	如果不设置，则表示未标注对象。annotation值为一个对象列表，详细参数请参见**表3-17**。
inference-loc	否	当此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。

表 3-16 数据来源类型

类型	示例
OBS	“source”：“s3://path-to-jpg”
Content	“source”：“content://I love machine learning”

表 3-17 annotation 对象说明

字段	是否必选	说明
type	是	标签类型。取值范围为： ● image_classification：图像分类 ● text_classification：文本分类 ● text_entity：文本命名实体 ● object_detection：对象检测 ● audio_classification：声音分类 ● audio_content：声音内容 ● audio_segmentation：声音起止点
name	是/否	对于分类是必选字段，对于其他类型为可选字段，本示例为图片分类名称。
id	是/否	标签ID。对于三元组是必选字段，对于其他类型为可选字段。三元组的实体标签ID格式为“E+数字”，比如“E1”、“E2”，三元组的关系标签ID格式为“R+数字”，例如“R1”、“R2”。
property	否	包含对标注的属性，例如本示例中猫有两个属性，颜色（color）和品种（kind）。
hard	否	表示是否是难例。“True”表示该标注是难例，“False”表示该标注不是难例。
annotated-by	否	默认为“human”，表示人工标注。 ● human
creation-time	否	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。
confidence	否	表示机器标注的置信度。范围为0～1。

图像分割

```
{
    "annotation": [{
        "annotation-format": "PASCAL VOC",
        "type": "modelarts/image_segmentation",
        "annotation-loc": "s3://path/to/annotation/image1.xml",
        "creation-time": "2020-12-16 21:36:27",
        "annotated-by": "human"
```

```
    }],
    "usage": "train",
    "source": "s3://path/to/image1.jpg",
    "id": "16d196c19bf61994d7deccafa435398c",
    "sample-type": 0
}
```

- “source”、“usage”、“annotation”等参数说明与图像分类一致，详细说明请参见表3-15。
- “annotation-loc”：对于图像分割、物体检测是必选字段，对于其他类型是可选字段，标注文件的存储路径。
- “annotation-format”：描述标注文件的格式，可选字段，默认为“PASCAL VOC”。目前只支持“PASCAL VOC”。
- “**sample-type**”：样本格式，0表示图片，1表示文本，2表示语音，4表示表格，6表示视频

表 3-18 PASCAL VOC 格式说明

字段	是否必选	说明
folder	是	表示数据源所在目录。
filename	是	被标注文件的文件名。
size	是	表示图像的像素信息。 ● width：必选字段，图片的宽度。 ● height：必选字段，图片的高度。 ● depth：必选字段，图片的通道数。
segmented	是	表示是否用于分割。
mask_source	否	表示图像分割保存的mask路径
object	是	表示物体检测信息，多个物体标注会有多个object体。 ● name：必选字段，标注内容的类别。 ● pose：必选字段，标注内容的拍摄角度。 ● truncated：必选字段，标注内容是否被截断（0表示完整）。 ● occluded：必选字段，标注内容是否被遮挡（0表示未遮挡） ● difficult：必选字段，标注目标是否难以识别（0表示容易识别）。 ● confidence：可选字段，标注目标的置信度，取值范围0-1之间。 ● bndbox：必选字段，标注框的类型，可选值请参见表3-19。 ● mask_color：必选字段，标签的颜色，以RGB值表示

表 3-19 标注框类型描述

type	形状	标注信息
polygon	多边形	各点坐标。 <x1>100<x1> <y1>100<y1> <x2>200<x2> <y2>100<y2> <x3>250<x3> <y3>150<y3> <x4>200<x4> <y4>200<y4> <x5>100<x5> <y5>200<y5> <x6>50<x6> <y6>150<y6> <x7>100<x7> <y7>100<y7>

示例：

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<annotation>
    <folder>NA</folder>
    <filename>image_0006.jpg</filename>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>230</width>
        <height>300</height>
        <depth>3</depth>
    </size>
    <segmented>1</segmented>
    <mask_source>obs://xianao/out/dataset-8153-Jmf5ylLjRmSacj9KevS/annotation/V001/
segmentationClassRaw/image_0006.png</mask_source>
    <object>
        <name>bike</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <mask_color>193,243,53</mask_color>
        <occluded>0</occluded>
        <polygon>
            <x1>71</x1>
            <y1>48</y1>
            <x2>75</x2>
            <y2>73</y2>
            <x3>49</x3>
            <y3>69</y3>
            <x4>68</x4>
            <y4>92</y4>
            <x5>90</x5>
            <y5>101</y5>
            <x6>45</x6>
            <y6>110</y6>
```

```
            <x7>71</x7>
            <y7>48</y7>
        </polygon>
    </object>
</annotation>
```

文本分类

```
{
    "source": "content://I like this product ",
    "id":"XGDVGS",
    "annotation": [
        {
            "type": "modelarts/text_classification",
            "name": " positive",
            "annotated-by": "human",
            "creation-time": "2019-01-23 11:30:30"
        } ]
}
```

content字段是指被标注的文本（UTF-8编码，可以是中文），其他参数解释与图像分类相同，请参见表3-15。

文本命名实体

```
{
    "source":"content://Michael Jordan is the most famous basketball player in the world.",
    "usage":"TRAIN",
    "annotation":[
        {
            "type":"modelarts/text_entity",
            "name":"Person",
            "property":{
                "@modelarts:start_index":0,
                "@modelarts:end_index":14
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        },
        {
            "type":"modelarts/text_entity",
            "name":"Category",
            "property":{
                "@modelarts:start_index":34,
                "@modelarts:end_index":44
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        }
    ]
}
```

“source”、“usage”、“annotation”等参数说明与图像分类一致，详细说明请参见表3-15。

其中，property的参数解释如表3-20所示。例如，当“"source":"content://Michael Jordan"”时，如果要提取“Michael”，则对应的“start_index”为“0”，“end_index”为“7”。

表 3-20 property 参数说明

参数名	数据类型	说明
@modelarts:start_index	Integer	文本的起始位置，值从0开始，包括start_index所指的字符。
@modelarts:end_index	Integer	文本的结束位置，但不包括end_index所指的字符。

文本三元组

```
{
    "source":"content://"Three Body" is a series of long science fiction novels created by Liu Cix.",
    "usage":"TRAIN",
    "annotation":[
        {
            "type":"modelarts/text_entity",
            "name":"Person",
            "id":"E1",
            "property":{
                "@modelarts:start_index":67,
                "@modelarts:end_index":74
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        },
        {
            "type":"modelarts/text_entity",
            "name":"Book",
            "id":"E2",
            "property":{
                "@modelarts:start_index":0,
                "@modelarts:end_index":12
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        },
        {
            "type":"modelarts/text_triplet",
            "name":"Author",
            "id":"R1",
            "property":{
                "@modelarts:from":"E1",
                "@modelarts:to":"E2"
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        },
        {
            "type":"modelarts/text_triplet",
            "name":"Works",
            "id":"R2",
            "property":{
                "@modelarts:from":"E2",
                "@modelarts:to":"E1"
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        }
    ]
}
```

“source”、“usage”、“annotation”等参数说明与**图像分类**一致，详细说明请参见**表3-15**。

其中，property的参数解释如表5 property参数说明所示。其中，“@modelarts:start_index”和“@modelarts:end_index”和文本命名实体的参数说明一致。例如，当“source”："content://"Three Body" is a series of long science fiction novels created by Liu Cix."时，“Liu Cix”是实体Person（人物），“Three Body”是实体Book（书籍），Person指向Book的关系是Author（作者），Book指向Person的关系是Works（作品）。

表 3-21 property 参数说明

参数名	数据类型	说明
@modelarts:start_index	Integer	三元组实体的起始位置，值从0开始，包括start_index所指的字符。
@modelarts:end_index	Integer	三元组实体的结束位置，但不包括end_index所指的字符。
@modelarts:from	String	三元组关系的起始实体id
@modelarts:to	String	三元组关系的指向实体id

物体检测

```
{
    "source":"s3://path/to/image1.jpg",
    "usage":"TRAIN",
    "hard":"true",
    "hard-coefficient":0.8,
    "annotation": [
        {
            "type":"modelarts/object_detection",
            "annotation-loc": "s3://path/to/annotation1.xml",
            "annotation-format":"PASCAL VOC",
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        }]
}
```

- “source”、“usage”、“annotation”等参数说明与图像分类一致，详细说明请参见表3-15。
- “annotation-loc”：对于物体检测、图像分割是必选字段，对于其他类型是可选字段，标注文件的存储路径。
- “annotation-format”：描述标注文件的格式，可选字段，默认为“PASCAL VOC”。目前只支持“PASCAL VOC”。

表 3-22 PASCAL VOC 格式说明

字段	是否必选	说明
folder	是	表示数据源所在目录。
filename	是	被标注文件的文件名。

字段	是否必选	说明
size	是	表示图像的像素信息。 ● width：必选字段，图片的宽度。 ● height：必选字段，图片的高度。 ● depth：必选字段，图片的通道数。
segmented	是	表示是否用于分割。
object	是	表示物体检测信息，多个物体标注会有多个object体。 ● name：必选字段，标注内容的类别。 ● pose：必选字段，标注内容的拍摄角度。 ● truncated：必选字段，标注内容是否被截断（0表示完整）。 ● occluded：必选字段，标注内容是否被遮挡（0表示未遮挡） ● difficult：必选字段，标注目标是否难以识别（0表示容易识别）。 ● confidence：可选字段，标注目标的置信度，取值范围0-1之间。 ● bndbox：必选字段，标注框的类型，可选值请参见**表3-23**。

表 3-23 标注框类型描述

type	形状	标注信息
point	点	点的坐标。 <x>100<x> <y>100<y>
line	线	各点坐标。 <x1>100<x1> <y1>100<y1> <x2>200<x2> <y2>200<y2>
bndbox	矩形框	左上和右下两个点坐标。 <xmin>100<xmin> <ymin>100<ymin> <xmax>200<xmax> <ymax>200<ymax>

type	形状	标注信息
polygon	多边形	各点坐标。 <x1>100<x1> <y1>100<y1> <x2>200<x2> <y2>100<y2> <x3>250<x3> <y3>150<y3> <x4>200<x4> <y4>200<y4> <x5>100<x5> <y5>200<y5> <x6>50<x6> <y6>150<y6>
circle	圆形	圆心坐标和半径。 <cx>100<cx> <cy>100<cy> <r>50<r>

示例：

```
<annotation>
  <folder>test_data</folder>
  <filename>260730932.jpg</filename>
  <size>
     <width>767</width>
     <height>959</height>
     <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
     <name>point</name>
     <pose>Unspecified</pose>
     <truncated>0</truncated>
     <occluded>0</occluded>
     <difficult>0</difficult>
     <point>
        <x1>456</x1>
        <y1>596</y1>
     </point>
  </object>
  <object>
     <name>line</name>
     <pose>Unspecified</pose>
     <truncated>0</truncated>
     <occluded>0</occluded>
     <difficult>0</difficult>
     <line>
        <x1>133</x1>
        <y1>651</y1>
        <x2>229</x2>
        <y2>561</y2>
     </line>
  </object>
```

```
  <object>
    <name>bag</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <bndbox>
      <xmin>108</xmin>
      <ymin>101</ymin>
      <xmax>251</xmax>
      <ymax>238</ymax>
    </bndbox>
  </object>
  <object>
    <name>boots</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <hard-coefficient>0.8</hard-coefficient>
    <polygon>
      <x1>373</x1>
      <y1>264</y1>
      <x2>500</x2>
      <y2>198</y2>
      <x3>437</x3>
      <y3>76</y3>
      <x4>310</x4>
      <y4>142</y4>
    </polygon>
  </object>
  <object>
    <name>circle</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <circle>
      <cx>405</cx>
      <cy>170</cy>
      <r>100<r>
    </circle>
  </object>
</annotation>
```

声音分类

```
{
"source":
"s3://path/to/pets.wav",
   "annotation": [
      {
         "type": "modelarts/audio_classification",
         "name":"cat",
         "annotated-by":"human",
         "creation-time":"2019-01-23 11:30:30"
      }
   ]
}
```

“source”、“usage”、“annotation”等参数说明与图像分类一致，详细说明请参见表3-15。

语音内容

```
{
   "source":"s3://path/to/audio1.wav",
   "annotation":[
```

```
        {
            "type":"modelarts/audio_content",
            "property":{
                "@modelarts:content":"Today is a good day."
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        }
    ]
}
```

- “source”、“usage”、“annotation”等参数说明与图像分类一致，详细说明请参见表3-15。
- “property”中的“@modelarts:content”参数，数据类型为“String”，表示语音内容。

语音分割

```
{
    "source":"s3://path/to/audio1.wav",
    "usage":"TRAIN",
    "annotation":[
        {

"type":"modelarts/audio_segmentation",
            "property":{
                "@modelarts:start_time":"00:01:10.123",
                "@modelarts:end_time":"00:01:15.456",

                "@modelarts:source":"Tom",

                "@modelarts:content":"How are you?"
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        },
        {
            "type":"modelarts/audio_segmentation",
            "property":{
                "@modelarts:start_time":"00:01:22.754",
                "@modelarts:end_time":"00:01:24.145",
                "@modelarts:source":"Jerry",
                "@modelarts:content":"I'm fine, thank you."
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        }
    ]
}
```

- “source”、“usage”、“annotation”等参数说明与图像分类一致，详细说明请参见表3-15。
- “property”的参数解释如表3-24所示。

表 3-24 “property”参数说明

参数名	数据类型	描述
@modelarts:start_time	String	声音的起始时间，格式为“hh:mm:ss.SSS”。 其中“hh”表示小时，“mm”表示分钟，“ss”表示秒，“SSS”表示毫秒。

参数名	数据类型	描述
@modelarts:end_time	String	声音的结束时间，格式为“hh:mm:ss.SSS”。 其中“hh”表示小时，“mm”表示分钟，“ss”表示秒，“SSS”表示毫秒。
@modelarts:source	String	声音来源。
@modelarts:content	String	声音内容。

视频标注

```
{
    "annotation": [{
        "annotation-format": "PASCAL VOC",
        "type": "modelarts/object_detection",
        "annotation-loc": "s3://path/to/annotation1_t1.473722.xml",
        "creation-time": "2020-10-09 14:08:24",
        "annotated-by": "human"
    }],
    "usage": "train",
    "property": {
        "@modelarts:parent_duration": 8,
        "@modelarts:parent_source": "s3://path/to/annotation1.mp4",
        "@modelarts:time_in_video": 1.473722
    },
    "source": "s3://input/path/to/annotation1_t1.473722.jpg",
    "id": "43d88677c1e9a971eeb692a80534b5d5",
    "sample-type": 0
}
```

- “source”、“usage”、“annotation”等参数说明与图像分类一致，详细说明请参见表3-15。
- “annotation-loc”：对于物体检测、是必选字段，对于其他类型是可选字段，标注文件的存储路径。
- “annotation-format”：描述标注文件的格式，可选字段，默认为“PASCAL VOC”。目前只支持“PASCAL VOC”。
- “**sample-type**”：样本格式，0表示图片，1表示文本，2表示语音，4表示表格，6表示视频。

表 3-25 property 参数说明

参数名	数据类型	说明
@modelarts:parent_duration	Double	标注视频的时长，单位：秒。
@modelarts:time_in_video	Double	标注的视频帧的时间戳，单位：秒。
@modelarts:parent_source	String	标注视频的OBS路径。

表 3-26 PASCAL VOC 格式说明

字段	是否必选	说明
folder	是	表示数据源所在目录。
filename	是	被标注文件的文件名。
size	是	表示图像的像素信息。 ● width：必选字段，图片的宽度。 ● height：必选字段，图片的高度。 ● depth：必选字段，图片的通道数。
segmented	是	表示是否用于分割。
object	是	表示物体检测信息，多个物体标注会有多个object体。 ● name：必选字段，标注内容的类别。 ● pose：必选字段，标注内容的拍摄角度。 ● truncated：必选字段，标注内容是否被截断（0表示完整）。 ● occluded：必选字段，标注内容是否被遮挡（0表示未遮挡） ● difficult：必选字段，标注目标是否难以识别（0表示容易识别）。 ● confidence：可选字段，标注目标的置信度，取值范围0-1之间。 ● bndbox：必选字段，标注框的类型，可选值请参见**表3-27**。

表 3-27 标注框类型描述

type	形状	标注信息
point	点	点的坐标。 <x>100<x> <y>100<y>
line	线	各点坐标。 <x1>100<x1> <y1>100<y1> <x2>200<x2> <y2>200<y2>

type	形状	标注信息
bndbox	矩形框	左上和右下两个点坐标。 <xmin>100<xmin> <ymin>100<ymin> <xmax>200<xmax> <ymax>200<ymax>
polygon	多边形	各点坐标。 <x1>100<x1> <y1>100<y1> <x2>200<x2> <y2>100<y2> <x3>250<x3> <y3>150<y3> <x4>200<x4> <y4>200<y4> <x5>100<x5> <y5>200<y5> <x6>50<x6> <y6>150<y6>
circle	圆形	圆心坐标和半径。 <cx>100<cx> <cy>100<cy> <r>50<r>

示例：

```
<annotation>
  <folder>test_data</folder>
  <filename>260730932_t1.473722.jpg.jpg</filename>
  <size>
     <width>767</width>
     <height>959</height>
     <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
     <name>point</name>
     <pose>Unspecified</pose>
     <truncated>0</truncated>
     <occluded>0</occluded>
     <difficult>0</difficult>
     <point>
        <x1>456</x1>
        <y1>596</y1>
     </point>
  </object>
  <object>
     <name>line</name>
     <pose>Unspecified</pose>
     <truncated>0</truncated>
```

```
    <occluded>0</occluded>
    <difficult>0</difficult>
    <line>
        <x1>133</x1>
        <y1>651</y1>
        <x2>229</x2>
        <y2>561</y2>
    </line>
  </object>
  <object>
    <name>bag</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <bndbox>
        <xmin>108</xmin>
        <ymin>101</ymin>
        <xmax>251</xmax>
        <ymax>238</ymax>
    </bndbox>
  </object>
  <object>
    <name>boots</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <hard-coefficient>0.8</hard-coefficient>
    <polygon>
        <x1>373</x1>
        <y1>264</y1>
        <x2>500</x2>
        <y2>198</y2>
        <x3>437</x3>
        <y3>76</y3>
        <x4>310</x4>
        <y4>142</y4>
    </polygon>
  </object>
  <object>
    <name>circle</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <circle>
        <cx>405</cx>
        <cy>170</cy>
        <r>100<r>
    </circle>
  </object>
</annotation>
```

3.6 导出数据

针对数据集中的数据，包含“已标注”和“未标注”的数据。您可以选中需要部分图片或者通过筛选条件筛选出需要的数据，导出成新的数据集，或者将数据导出至指定的OBS目录下。您可以通过任务历史查看数据导出的历史记录。

目前只有“图片”类型的数据集支持导出功能，“自由格式”、“音频”、“视频”、“文本”、“表格”类型的数据集均不支持导出功能。

“图片”支持导出“图像分类”、“物体检测”、“图像分割”三种标注结果。

- “图像分类”只支持导出txt格式的标注文件。

- “物体检测”只支持导出Pascal VOC格式的XML标注文件。
- “图像分割”只支持导出Pascal VOC格式的XML标注文件以及Mask图像。

导出新数据集

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，选择“图片”类型的数据集，单击数据集名称进入“数据集概览页”。
3. 在“数据集概览页”，单击右上角“导出 ”。在弹出的“导出”对话框中，填写相关信息，然后单击“确定”，开始执行导出操作。

   “导出方式”：选择新数据集或OBS。

   – **导出新数据集：**

     “名称”：当新数据集名称。

     “保存路径”：表示新数据集的输入路径，即当前数据导出后存储的OBS路径。

     “输出路径”：表示新数据集的输出路径，即新数据集在完成标注后输出的路径。“输出路径”不能与“保存路径”为同一路径，且“输出路径”不能是“保存路径”的子目录。

   – **导出OBS**

     “保存路径”：即导出数据存储的路径。建议不要将数据存储至当前数据集所在的输入路径或输出路径。

图 3-71 导出新数据集

4. 数据导出成功后，您可以前往您设置的保存路径，查看到存储的数据。当导出方式选择为新数据集时，在导出成功后，您可以前往“数据集”列表中，查看到新的数据集。

查看导出任务历史

当您导出新数据集或导出数据至OBS，您可以通过任务历史查看导出任务明细。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，选择“图片”类型的数据集，单击数据集名称进入“数据集概览页”。
3. 在“数据集概览页”，单击右上角“导出历史”。
4. 在弹出的“任务历史”对话框中，可以查看该数据集之前的导出任务历史。包括“任务ID”、“创建时间”、“导出方式”、“导出路径”、“导出样本总数”和“导出状态”。

图 3-72 任务历史

3.7 修改数据集

对于已创建的数据集，您可以修改数据集的基本信息以匹配业务变化。

前提条件

已存在创建完成的数据集。

修改数据集基本信息

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，单击操作列的“修改”。
3. 参考表3-28修改数据集基本信息，然后单击“确定”完成修改。

图 3-73 修改数据集

表 3-28 参数说明

参数	说明
名称	数据集的名称，名称只能是字母、数字、下划线或者中划线组成的合法字符串。
描述	数据集的简要描述。

3.8 发布数据集

ModelArts在数据集管理过程中，针对同一个数据源，对不同时间标注后的数据，按版本进行区分，方便后续模型构建和开发过程中，选择对应的数据集版本进行使用。数据标注完成后，您可以将数据集当前状态进行发布，生成一个新的数据集版本。

关于数据集版本

- 针对刚创建的数据集（未发布前），无数据集版本信息，必须执行发布操作后，才能应用于模型开发或训练。
- 数据集版本，默认按V001、V002递增规则进行命名，您也可以在发布时自定义设置。
- 您可以将任意一个版本设置为当前目录，即表示数据集列表中进入的数据集详情，为此版本的数据集标注信息。
- 针对每一个数据集版本，您可以通过“存储路径”参数，获得此版本对应的Manifest文件格式的数据集。可用于导入数据或难例筛选操作。
- 表格数据集暂不支持切换版本。

发布数据集

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，单击操作列的“发布”。或者，您可以单击数据集名称，进入数据集“概览”页，在页面右上角单击“发布”。
3. 在“发布新版本”弹出框中，填写发布数据集的相关参数，然后单击“确定”。

表 3-29 发布数据集的参数说明

参数	描述
“版本名称”	默认按V001、V002递增规则进行命名，您也可以自定义版本名称。版本名称只能包含字母、数字、中划线或下划线。
“版本格式”	仅“表格”类型数据集支持设置版本格式，支持“CSV”和“CarbonData”两种。 **说明** 如果导出的CSV文件中存在以“=”“+”“-”和“@”开头的命令时，为了安全考虑，ModelArts会自动加上Tab键，并对双引号进行转义处理。

参数	描述
“数据切分”	仅“图片”、“音频”、“文本”类型数据集支持进行数据切分功能。 默认不启用。启用后，需设置对应的训练验证比例。 输入“训练集比例”，数值只能是0~1区间内的数。设置好“训练集比例”后，“验证集比例”自动填充。“训练集比例”加“验证集比例”等于1。 “训练集比例”即用于训练模型的样本数据比例；“验证集比例”即用于验证模型的样本数据比例。“训练验证比例”会影响训练模板的性能。
“描述”	针对当前发布的数据集版本的描述信息。
“标注类型”	发布的新版本数据集用于的标注类型。

图 3-74 发布数据集

版本发布后，您可以前往版本管理查看详细信息。系统默认将最新的版本作为当前目录。

数据集发布后，相关文件的目录结构说明

由于数据集是基于OBS目录管理的，发布为新版本后，对应的数据集输出位置，也将基于新版本生成目录。

以图像分类为例，数据集发布后，对应OBS路径下生成，其相关文件的目录如下所示。

```
|-- user-specified-output-path
    |-- DatasetName-datasetId
```

```
    |-- annotation
        |-- VersionMame1
            |-- VersionMame1.manifest
        |-- VersionMame2
            ...
        |-- ...
```

以物体检测为例，如果数据集导入的是Manifest文件，在数据集发布后，其相关文件的目录结构如下。

```
|-- user-specified-output-path
   |-- DatasetName-datasetId
      |-- annotation
         |-- VersionMame1
            |-- VersionMame1.manifest
            |-- annotation
               |-- file1.xml
         |-- VersionMame2
            ...
         |-- ...
```

以视频标注为例，在数据集发布后，标注结果将标注结果文件（XML）存放在数据集输出目录下。

```
|-- user-specified-output-path
    |-- DatasetName-datasetId
       |-- annotation
          |-- VersionMame1
             |-- VersionMame1.manifest
             |-- annotations
               |-- images
                  |-- videoName1
                     |-- videoName1.timestamp.xml
                  |-- videoName2
                     |-- videoName2.timestamp.xml
          |-- VersionMame2
             ...
          |-- ...
```

视频标注的关键帧存在数据集的输入目录下。

```
|-- user-specified-input-path
    |-- images
       |-- videoName1
          |-- videoName1.timestamp.jpg
       |-- videoName2
          |-- videoName2.timestamp.jpg
```

3.9 删除数据集

如果数据集不再使用，您可以删除数据集释放资源。

说明

删除数据集后，数据集对应的数据集输入位置和数据集输出位置对应的OBS目录下，如果需要删除OBS目录下的数据释放资源，建议前往OBS管理控制台，删除对应的数据，然后再删除OBS文件夹。

操作步骤

1. 在“数据管理>数据集”页面中，单击数据集操作列的“更多 > 删除”。
2. 在弹出的对话框中，单击“确定”，确认删除此数据集。

说明

删除后，数据集的版本管理等功能无法恢复，请谨慎操作。但是，此数据集对应的原始数据和标注数据依然存储在OBS中。

3.10 管理数据集版本

数据标注完成后，您可以发布成多个版本对数据集进行管理。针对已发布生产的数据集版本，您可以通过查看数据集演进过程、设置当前版本、删除版本等操作，对数据集进行管理。数据集版本的相关说明，请参见关于数据集版本。

发布为新版本的说明，请参见发布数据集。

查看数据集演进过程

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，单击操作列的“更多 > 版本管理”，进入数据集“版本管理”页面。

   您可以查看数据集的基本信息，并在左侧查看版本演进信息及其发布时间。

   图 3-75 查看数据集版本

设置当前版本

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，单击操作列的“更多 > 版本管理”，进入数据集“版本管理”页面。
3. 在“版本管理”页面中，选择对应的数据集版本，在数据集版本基本信息区域，单击“设置为当前版本”。设置完成后，版本名称右侧将显示为“当前版本”。

图 3-76 设置当前版本

说明

只有状态为“正常”的版本，才能被设置为当前版本。

删除数据集版本

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，单击操作列的“更多 > 版本管理”，进入数据集“版本管理”页面。
3. 选择需删除的版本所在行，单击操作列的“删除”。在弹出的对话框中确认信息，然后单击“确定”完成删除操作。

说明

删除数据集版本不会删除原始数据，数据及其标注信息仍存在在对应的OBS目录下。但是，执行删除操作后，无法在ModelArts管理控制台清晰的管理数据集版本，请谨慎操作。

3.11 智能标注

除了人工标注外，ModelArts还提供了智能标注功能，快速完成数据标注，为您节省70%以上的标注时间。智能标注是指基于当前标注阶段的标签及图片学习训练，选中系统中已有的模型进行智能标注，快速完成剩余图片的标注操作。

背景信息

- 目前只有“图像分类”和“物体检测”类型的标注作业支持智能标注功能。
- 启动智能标注时，需标注作业存在至少1种标签，且每种标签已标注的图片不少于5张。
- 启动智能标注时，必须存在未标注图片。
- 启动智能标注前，保证当前系统中不存在正在进行中的智能标注任务。
- 检查用于标注的图片数据，确保您的图片数据中，不存在RGBA四通道图片。如果存在四通道图片，智能标注任务将运行失败，因此，请从数据集中删除四通道图片后，再启动智能标注。

启动智能标注作业

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理 > 数据标注”，进入“数据标注”管理页面。
2. 在标注作业列表中，选择“物体检测”或“图像分类”类型的标注作业，单击操作列的“智能标注”启动智能标注作业。
3. 在弹出的“启动智能标注”对话框中，选择智能标注类型，可选“主动学习”或者“预标注”，详见表3-30和表3-31。

表 3-30 主动学习

参数	说明
智能标注类型	“主动学习”。“主动学习”表示系统将自动使用半监督学习、难例筛选等多种手段进行智能标注，降低人工标注量，帮助用户找到难例。
算法类型	针对“图像分类”类型的数据集，您需要选择以下参数。 “快速型”：仅使用已标注的样本进行训练。 “精准型”：会额外使用未标注的样本做半监督训练，使得模型精度更高。

表 3-31 预标注

参数	说明
智能标注类型	“预标注”。“预标注”表示选择用户模型管理里面的模型，选择模型时需要注意模型类型和数据集的标注类型相匹配。预标注结束后，如果标注结果符合平台定义的标准标注格式，系统将进行难例筛选，该步骤不影响预标注结果。
选择模型及版本	● “我的模型”。您可以根据实际需求选择您的模型。您需要在目标模型的左侧单击下拉三角标，选择合适的版本。您的模型导入参见导入模型。 ● “我的订阅”。您可以根据实际需求选择AI Gallery中已订阅的模型。您需要在目标模型的左侧单击下拉三角标，选择合适的版本。查找模型参见我的订阅模型。
计算节点规格	在下拉框中，您可以选择目前ModelArts支持的节点规格选项。
计算节点个数	默认为1。您可以根据您的实际情况选择，最大为5。

说明

- 针对“物体检测”类型的标注作业，选择“主动学习”时，只支持识别和标注矩形框。
- 当系统中智能标注作业过多时，可能会出现排队的情况，导致作业一直处于“标注中”的状态。请您耐心等待，系统会按照顺序完成标注作业。

图 3-77 启动智能标注（图像分类）

图 3-78 启动智能标注（物体检测）

图 3-79 启动智能标注（预标注）

4. 完成参数设置后，单击“提交”，即可启动智能标注。
5. 在标注作业列表中，单击标注作业名称进入“标注作业详情”页。
6. 在“数据集概览页标注作业详情页”，选择“标注”页签，单击“待确认”页签，即可查看智能标注进度。

   您也可以在该页签，“启动智能标注”或者查看“智能标注历史”

   图 3-80 标注进度

7. 智能标注完成后，“待确认”页面将呈现所有标注后的图片列表。
   - 图像分类标注作业

     在“待确认”页面查看标签是否准确，勾选标注准确的图片，然后单击“确认”完成智能标注结果的确认。确认完成后的图片将被归类至“已标注”页面下。

     针对标为“难例”的图片，您可以根据实际情况判断，手工修正标签。详细操作及示例请参见**•针对“图像分类”数据集**。
   - 物体检测标注作业

     在“待确认”页面，单击图片查看标注详情，查看标签及目标框是否准确，针对标注准确的图片单击“确认标注”完成智能标注结果的确认。确认完成后的图片将被归类至“已标注”页面下。

     针对标为“难例”的图片，您可以根据实际情况判断，手工修正标签或目标框。详细操作及示例请参见**•针对“物体检测”数据集**。

3.12 难例确认

在数据量很大的标注任务中，标注初期由于已标注图片不足，智能标注的结果无法直接用于训练。若对所有的未标注数据一一进行调整确认仍然需要较大的人力和时间成本。为了更快地完成标注任务，在对未标注数据进行智能标注的任务中，ModelArts嵌入了自动难例发现功能。该功能会对剩余未标注图片的标注优先级给出建议。因为标注优先级高的图片的智能标注结果未达到预期，所以称之为难例。

ModelArts平台提供的自动难例发现功能，在智能标注以及数据采集筛选过程中，将自动标注出难例，建议对难例数据进一步确认标注，然后将其加入训练数据集中，使用此数据集训练模型，可得到精度更高的模型。首先，针对智能标注和采集筛选任务，难例的发现操作是系统自动执行的，无需人工介入，仅需针对标注后的数据进行确认和修改即可，提升数据管理和标注效率。其次，您可以基于难例的情况，补充类似数据，提升数据集的丰富性，进一步提升模型训练的精度。

在数据集管理中，对难例的管理有如下场景。

- 智能标注后，确认难例
- 将数据集中的数据标注为难例

说明

目前只有“图像分类”和“物体检测”类型的数据集支持难例发现功能。

智能标注后，确认难例

“智能标注”任务执行过程中，ModelArts将自动识别难例，并完成标注。当智能标注结束后，难例标注结果将呈现在“待确认”页签，建议您对难例数据进行人工修正，然后确认标注。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据标注”，进入“数据标注”管理页面。
2. 在标注作业列表中，选择“物体检测”或“图像分类”类型的标注作业，单击标注作业名称进入“标注作业详情”。
3. 在“标注作业详情页”，选择“标注>待确认”页签，查看并确认难例。

说明

只有当智能标注任务完成后，待确认页签才会显示标注数据。否则，此页签内容为空。智能标注操作请参见智能标注。

– 针对“物体检测”标注作业

在“待确认”页签中，单击图片展开标注详情，查看图片数据的标注情况，如标签是否准确、目标框位置添加是否准确。如果智能标注结果不准确，建议手工调整标签或目标框，然后单击“确认标注”。完成确认后，重新标注的数据将呈现在“已标注”页签下。

如图3-81所示的难例，dog标签的目标框位置不准确，使用标注框重新标注，如图中的“漏检”目标框，然后需要将原先标注错误的目标框删除，即“误检”标签框。手工调整后，单击“确认标注”完成难例确认。

图 3-81 物体检测的难例确认

– 针对“图像分类”标注作业

在“待确认”页签中，查看标注难例的图片，其添加的标签是否准确。勾选标注不准确的图片，删除错误标签，然后在右侧“标签名”处添加准确标签。单击“确认”，勾选的图片及其标注情况，将呈现在“已标注”页签下。

如图3-82所示，选中的图片为标注错误图片，在右侧删除错误标签，然后在标签名处添加“狗”的标签，然后单击“确认”，完成难例确认。

图 3-82 图像分类的难例确认

将数据集中的数据标注为难例

针对标注作业中，已标注或未标注数据，也可以将图片数据标注为难例。标注为难例的数据，对后续模型训练中，通过内置规则提升模型精度。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据标注”，进入“数据标注”管理页面。
2. 在标注作业列表中，选择“物体检测”或“图像分类”类型的标注作业，单击数据集名称进入“标注作业详情页”。
3. 在标注作业详情页，选择“标注”页签”，单击“已标注”、“未标注”或“全部”页签，勾选需标注为难例的图片，然后单击“难例批处理 > 确认为难例”。完成标注后，图片预览时，其右上角将显示为“难例”。

图 3-83 确认为难例

3.13 自动分组

为了提升智能标注算法精度，可以均衡标注多个类别，有助于提升智能标注算法精度。ModelArts内置了分组算法，您可以针对您选中的数据，执行自动分组，提升您的数据标注效率。

自动分组可以理解为数据标注的预处理，先使用聚类算法对未标注图片进行聚类，再根据聚类结果进行处理，可以分组打标或者清洗图片。

例如，用户通过搜索引擎搜索XX，将相关图片下载并上传到数据集，然后再使用自动分组，可以将XX图片分类，比如论文、宣传海报、确认为XX的图片、其他。用户可以根据分组结果，快速剔除掉不想要的，或者将某一类直接全选后添加标签。

说明

目前只有“图像分类”、“物体检测”和“图像分割”类型的数据集支持自动分组功能。

启动自动分组任务

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据标注”，进入“数据标注”管理页面。
2. 在标注作业列表中，选择“物体检测”或“图像分类”类型的标注作业，单击标注作业名称进入“标注作业详情页”。
3. 在数据集详情页的“标注>全部”页签中，单击“自动分组 > 启动任务”。

   说明

   只能在“全部”页签下启动自动分组任务或查看任务历史。
4. 在弹出的“自动分组”对话框中，填写参数信息，然后单击“确定”。
   - “分组数”：填写2~200之间的整数，指将图片分为多少组。
   - “结果处理方式”：“更新属性到当前样本中”，或者“保存到对象存储服务（OBS）”。
   - “属性名称”：当选择“更新属性到当前样本中”时，需输入一个属性名称。
   - “结果存储目录”：当选择“保存到对象存储服务（OBS）”时，需指定一个用于存储的OBS路径。
   - “高级特征选项”：启用此功能后，可选择“清晰度”、“亮度”、“图像色彩”等维度为自动分组功能增加选项，使得分组着重于图片亮度、色彩和清晰度等特征进行分组。支持多选。

图 3-84 自动分组

5. 启动任务提交成功后，界面右上角显示此任务的进度。等待任务执行完成后，您可以查看自动分组任务的历史记录，了解任务状态。

查看自动分组结果

在数据集详情页面的“全部”页签中，展开“筛选条件”，将“样本属性”设置为自动分组任务中的“属性名称”，并通过设置样本属性值，筛选出分组结果。

图 3-85 查看自动分组结果

查看自动分组的历史任务

在数据集详情页面的“全部”页签中，单击“自动分组 > 任务历史”。在弹出的“任务历史”对话框中，展示当前数据集之前执行的自动分组任务的基本信息。

图 3-86 自动分组任务历史

3.14 数据特征分析

基于图片或目标框对图片的各项特征，如模糊度、亮度进行分析，并绘制可视化曲线，帮助处理数据集。

您还可以选择数据集的多个版本，查看其可视化曲线，进行对比分析。

背景信息

- 只有“图片”的数据集，且版本标注类型为“物体检测”和“图像分类”的数据集版本支持数据特征分析。
- 只有发布后的数据集支持数据特征分析。发布后的Default格式数据集版本支持数据特征分析。
- 数据特征分析的数据范围，不同类型的数据集，选取范围不同：
  - 对于标注任务类型为“物体检测”的数据集版本，当已标注样本数为0时，发布版本后，数据特征页签版本置灰不可选，无法显示数据特征。否则，显示已标注的图片的数据特征。
  - 对于标注任务类型为“图像分类”的数据集版本，当已标注样本数为0时，发布版本后，数据特征页签版本置灰不可选，无法显示数据特征。否则，显示全部的图片的数据特征。
- 数据集中的图片数量要达到一定量级才会具有意义，一般来说，需要有大约1000+的图片。
- “图像分类”支持分析指标有：“分辨率”、“图片高宽比”、“图片亮度”、“图片饱和度”、“清晰度”和“图像色彩的丰富程度”。“物体检测”支持所有的分析指标。目前ModelArts支持的所有分析指标请参见**支持分析指标及其说明**。

数据特征分析

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 选择对应的数据集，单击操作列的“数据特征”，进入数据集概览页的数据特征页面。

   您也可以在单击数据集名称进入数据集概览页后，单击“数据特征”页签进入。
3. 由于发布后的数据集不会默认启动数据特征分析，针对数据集的各个版本，需手动启动特征分析任务。在数据特征页签下，单击“特征分析”。

   图 3-87 选择特征分析

4. 在弹出的对话框中配置需要进行特征分析的数据集版本，然后单击“确定”启动分析。

   “版本选择”，即选择当前数据集的已发布版本。

   图 3-88 启动数据特征分析任务

5. 数据特征分析任务启动后，需执行一段时间，根据数据量不同等待时间不同，请耐心等待。当您选择分析的版本出现在“版本选择”列表下，且可勾选时，即表示分析已完成。

图 3-89 可选择已执行特征分析的版本

6. 查看数据特征分析结果。

“版本选择”：在右侧下拉框中选择进行对比的版本。也可以只选择一个版本。

“类型”：选择需要分析的类型。支持“all”、“train”、“eval”和“inference”。

“数据特征指标”：在右侧下拉框中勾选需要展示的指标。详细指标说明请参见**支持分析指标及其说明**。

选择完成后，页面将自动呈现您选择对应版本及其指标数据，如**图3-90**所示，您可以根据呈现的图表了解数据分布情况，帮助您更好的处理您的数据。

图 3-90 数据特征分析

7. 查看分析任务的历史记录。

在数据特征分析后，您可以在“数据特征”页签下，单击右侧“任务历史”，可在弹出对话框中查看历史分析任务及其状态。

图 3-91 任务历史

支持分析指标及其说明

表 3-32 分析指标列表

名称	说明	分析说明
分辨率 Resolution	图像分辨率。此处使用面积值作为统计值。	通过指标分析结果查看是否有偏移点。如果存在偏移点，可以对偏移点做resize操作或直接删除。
图片高宽比 Aspect Ratio	图像高宽比，即图片的高度/图片的宽度。	一般呈正态分布，一般用于比较训练集和真实场景数据集的差异。
图片亮度 Brightness	图片亮度，值越大代表观感上亮度越高。	一般呈正态分布，可根据分布中心判断数据集整体偏亮还是偏暗。可根据使用场景调整，比如使用场景是夜晚，图片整体应该偏暗。

名称	说明	分析说明
图片饱和度 Saturation	图片的色彩饱和度，值越大表示图片整体色彩越容易分辨。	一般呈正态分布，一般用于比较训练集和真实场景数据集的差异。
清晰度 Clarity	图片清晰程度，使用拉普拉斯算子计算所得，值越大代表边缘越清晰，图片整体越清晰。	可根据使用场景判断清晰度是否满足需要。比如使用场景的数据采集来自高清摄像头，那么清晰度对应的需要高一些。可通过对数据集做锐化或模糊操作，添加噪声对清晰度做调整。
图像色彩的丰富程度 Colorfulness	横坐标：图像的色彩丰富程度，值越大代表色彩越丰富。 纵坐标：图片数量。	是观感上的色彩丰富程度，一般用于比较训练集和真实场景数据集的差异。
按单张图片中框的个数统计图片分布 Bounding Box Quantity	横坐标：单张图片中框的个数。 纵坐标：图片数量。	对模型而言一张图片的框个数越多越难检测，需要越多的这种数据用作训练。
按单张图片中框的面积标准差统计图片分布 Standard Deviation of Bounding Boxes Per Image	横坐标：单张图片中框的标准差。单张图片只有一个框时，标准差为0。标准差的值越大，表示图片中框大小不一程度越高。 纵坐标：图片数量。	对模型而言一张图中框如果比较多且大小不一，是比较难检测的，可以根据场景添加数据用作训练，或者实际使用没有这种场景可直接删除。
按高宽比统计框数量的分布 Aspect Ratio of Bounding Boxes	横坐标：目标框的高宽比。 纵坐标：框数量（统计所有图片中的框）。	一般呈泊松分布，但与使用场景强相关。多用于比较训练集和验证集的差异，如训练集都是长方形框的情况下，验证集如果是接近正方形的框会有比较大影响。
按面积占比统计框数量的分布 Area Ratio of Bounding Boxes	横坐标：目标框的面积占比，即目标框的面积占整个图片面积的比例，越大表示物体在图片中的占比越大。 纵坐标：框数量（统计所有图片中的框）。	主要判断模型中使用的anchor的分布，如果目标框普遍较大，anchor就可以选择较大。

名称	说明	分析说明
按边缘化程度统计框数量的分布 Marginalization Value of Bounding Boxes	横坐标：边缘化程度，即目标框中心点距离图片中心点的距离占图片总距离的比值，值越大表示物体越靠近边缘。（图片总距离表示以图片中心点为起点画一条经过标注框中心点的射线，该射线与图片边界交点到图片中心点的距离）。 纵坐标：框数量（统计所有图片中的框）。	一般呈正态分布。用于判断物体是否处于图片边缘，有一些只露出一部分的边缘物体，可根据需要添加数据集或不标注。
按堆叠度统计框数量的分布 Overlap Score of Bounding Boxes	横坐标：堆叠度，单个框被其他的框重叠的部分，取值范围为0～1，值越大表示被其他框覆盖的越多。 纵坐标：框数量（统计所有图片中的框）。	主要用于判断待检测物体的堆叠程度，堆叠物体一般对于检测难度较高，可根据实际使用需要添加数据集或不标注部分物体。
按亮度统计框数量的分布 Brightness of Bounding Boxes	横坐标：目标框的图片亮度，值越大表示越亮。 纵坐标：框数量（统计所有图片中的框）。	一般呈正态分布。主要用于判断待检测物体的亮度。在一些特殊场景中只有物体的部分亮度较暗，可以看是否满足要求。
按清晰度统计框数量的分布 Clarity of Bounding Boxes	横坐标：目标框的清晰度，值越大表示越清晰。 纵坐标：框数量（统计所有图片中的框）。	主要用于判断待检测物体是否存在模糊的情况。比如运动中的物体在采集中可能变得模糊，需要重新采集。

3.15 团队标注

3.15.1 团队标注简介

数据标注任务中，一般由一个人完成，但是针对数据集较大时，需要多人协助完成。ModelArts提供了团队标注功能，可以由多人组成一个标注团队，共同完成同一个标注作业。

说明

团队标注功能当前仅支持“图像分类”、“物体检测”、“文本分类”、“命名实体”、“文本三元组”、“语音分割”类型的标注作业。

如何创建团队标注作业

- 创建标注作业时，打开“启用团队标注”开关，同时指定一个标注团队，或者指定标注管理员。

图 3-92 创建团队标注作业

- 对于同一个数据集，可以创建多个团队标注任务。创建团队标注的操作详情请参见创建团队标注任务。

图 3-93 在数据集列表中创建团队标注任务

图 3-94 在数据集概览页中，选择标注标注任务页签，点击“新建标注任务”，创建团队标注任务

说明

- 当标注人员第一次收到标注邮件时，会告知标注的地址(labelConsole)以及默认初始密码。
- 只有当创建团队标注任务时，标注人员才会收到邮件。创建标注团队及添加标注团队的成员并不会发送邮件。此外，当所有样本都是已标注状态时，创建团队标注任务也不会收到邮件。

团队标注相关操作

- 管理团队
- 管理成员
- 管理团队标注任务

3.15.2 管理团队

团队标注功能是以团队为单位进行管理，数据集启用团队标注功能时，必须指定一个团队。一个团队可以添加多个成员。

背景说明

- 一个帐号最多可添加10个团队。
- 如果数据集需要启用团队标注功能，当前帐号至少拥有一个团队。如果没有，请执行添加团队操作添加。

添加团队

1. 在ModelArts管理控制台左侧导航栏中，选择“数据管理>标注团队”，进入“标注团队”管理页面。
2. 在“标注团队”管理页面，单击“添加团队”。
3. 在弹出的“添加团队”对话框中，填写团队“名称”和“描述”，然后单击“确定”。完成标注团队的添加。

图 3-95 添加团队

团队添加完成后，“标注团队”管理页面呈现新添加的团队，在页面右侧区域，可以查看团队详情。新添加的团队，其成员列表为空，请参考添加成员操作，为您的团队添加成员。

删除团队

当已有的团队不再使用，您可以执行删除操作。

在“标注团队”管理页面中，选中需删除的团队，然后单击“删除”。在弹出的对话框中，确认信息无误后，单击“确定”完成团队删除。

图 3-96 删除团队

3.15.3 管理成员

新添加的团队，其成员列表为空。您需要根据实际情况添加即将参与标注任务的成员信息。

一个团队最多支持添加100个成员，当超过100时，建议分为多个团队进行管理。

添加成员

1. 在ModelArts管理控制台左侧导航栏中，选择“数据管理>标注团队”，进入“标注团队”管理页面。

2. 在“标注团队”管理页面，从左侧团队列表中选择一个团队，单击团队，其右侧区域将呈现“团队详情”。
3. 在“团队详情”区域，单击“添加成员”。
4. 在弹出的“添加成员”对话框中，按照分类，可选择“当前账号”或“其他账号”。选择“当前账号”，需要填写成员的 “IAM用户名”、“角色”、“邮箱”、“描述”。选择“其他账号”，需要填写成员的“账号ID”、“IAM用户ID”、“角色”、“描述”，单击左下角“添加账号ID”。填写完成之后，然后单击“确定”。

   邮箱作为团队管理中的唯一标识，不同成员不能使用同一个邮箱。您填写的邮箱地址将被记录并保存在ModelArts中，仅用于ModelArts团队标注功能，当成员删除后，其填写的邮箱信息也将被一并删除。

   其中，“角色”支持“Labeler”、“Reviewer”和“Team Manager”，“Team Manager”只能设置为一个人。

图 3-97 当前账号添加成员

图 3-98 其他账号添加成员

成员添加完成后，团队详情区域中将呈现此成员的信息。

修改成员信息

团队中的成员，当其信息发生变化时，可以编辑其基本情况。

1. 在“团队详情”区域，选择需修改的成员。

2. 在成员所在行的“操作”列，单击“修改”。在弹出的对话框中，修改其“描述”或“角色”。

   成员的“邮箱”无法修改，如果需要修改邮箱地址，建议先删除此成员，然后再基于新的邮箱地址添加新成员。

   “角色”支持“Labeler”、“Reviewer”和“Team Manager”，“Team Manager”只能设置为一个人。

删除成员

- **删除单个成员**

  在“团队详情”区域，选择需要删除的成员， 单击“操作”列的“删除”。在弹出的对话框中，确认信息无误后，单击“确定”完成删除操作。

- **批量删除**

  在“团队详情”区域，勾选需删除的成员，然后单击“删除”。在弹出的对话框中，确认信息无误后，单击“确定”完成多个成员的删除操作。

图 3-99 批量删除

3.15.4 管理团队标注任务

数据标注支持创建团队标注任务，将标注任务指派给不同的团队，由多人完成标注任务。同时，在成员进行数据标注过程中，支持发起验收、继续验收以及查看验收报告等功能。

团队标注作业的创建方式

- 从控制台的“数据管理 >数据标注”页面进入，创建标注作业时，打开“启用团队标注”开关，同时指定一个标注团队，或者指定标注管理员。

图 3-100 创建团队标注作业

- 从控制台的“数据管理 >数据集”页面进入，对于同一个数据集，可以创建多个团队标注任务。

图 3-101 在数据集列表中创建团队标注任务

图 3-102 在数据集概览页中，选择标注任务页签，点击“新建标注任务”，创建团队标注任务

说明

- 当标注人员第一次收到标注邮件时，会告知标注的地址(labelConsole)以及默认初始密码。
- 只有当创建团队标注任务时，标注人员才会收到邮件。创建标注团队及添加标注团队的成员并不会发送邮件。此外，当所有样本都是已标注状态时，创建团队标注任务也不会收到邮件。
- 标注任务创建完成后，会将所有未标注状态的样本分配给标注人员。分配采用随机均分的策略，不支持重复分配。

创建团队标注任务

同一个数据集，支持创建多个团队标注作业，指派给同一团队的不同成员，或者指派给其他标注团队。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理 >数据集”，打开数据集列表。
2. 在数据集列表中，选择支持团队标注的数据集，单击数据集名称进入数据集概览页。
3. 在数据集概览页中，单击“标注任务”页签，可查看此数据集已有的标注任务。单击左上角的“新建标注任务”开始创建新任务。

图 3-103 标注任务

或者也可以从“数据管理 >数据标注”页面进入，单击“创建标注作业”进入创建标注作业页面。

4. 在弹出的“创建标注作业”页面中，填写相关参数，然后单击“确定”，完成任务创建。
   - “名称”：设置此任务的名称。
   - **“标注场景”**：选择标注作业的任务类型。
   - “标签集”：展示当前数据集已有的标签及标签属性。
   - **“启用团队标注”**：选择打开，并配置如下团队标注相关参数。
     - “类型”：设置任务类型，支持“指定标注团队”或“指定标注管理员”。
     - “选择标注团队”：任务类型设置为“指定标注团队”，需在此参数中指定一个团队，同时勾选此团队中某几个成员负责标注。下拉框中将罗列当前帐号下创建的标注团队及其成员。

- “选择标注接口人”：任务类型设置为“指定标注管理员”，需在所有团队的“Team Manager”中选择一人作为管理员。
- “自动将新增图片同步给标注团队”：根据需要选择是否将任务中新增的数据自动同步给标注人员。
- “团队标注的图片自动加载智能标注结果”：根据需要选择是否将任务中智能标注待确认的结果自动同步给标注人员。

说明

团队标注加载智能标注结果的处理步骤：

- 如果类型选择"指定标注团队"，需要先创建团队标注任务，然后执行智能标注任务。
- 如果类型选择"指定标注管理员"，需要登录labelconsole先执行"分配任务"，然后执行智能标注任务。

图 3-104 创建团队标注任务

任务创建完成后，您可以在“标注任务”页签下看到新建的任务。

进入标注（团队成员）

在标注任务创建后，被分配任务的团队成员将收到一封通知邮件，标题为“您有新的标注任务待查收”。

在邮件详情中，单击标注任务链接进入数据标注平台，使用您的“邮箱地址”及“初始密码”登录标注平台系统。首次登录时，需根据提示修改密码。关于标注平台的详细说明，请参见什么是数据标注平台。

说明

首次登录时，请务必修改密码。如果未修改，将导致下一次无法正常登录。

登录数据标注平台后，可查看到分配的标注任务，单击任务名称，可进入标注页面。不同类型的数据集，标注方式不同，详细请参见：

- 图像分类
- 物体检测
- 文本分类
- 命名实体
- 文本三元组

在标注平台中，每个成员可查看“未标注”、“待修正”、“已驳回”、“待审核”、“审核通过”、“验收通过”的图片信息。请及时关注管理员驳回以及待修正的图片。

当团队标注任务中，分配了Reviewer角色，则需要对标注结果进行审核，审核完成后，再提交给管理员验收。

图 3-105 成员标注平台

任务验收（管理员）

- **发起验收**

  当团队的成员已完成数据标注，标注作业的创建者可发起验收，对标注结果进行抽验。只有当标注成员存在标注完成的数据时，才可以发起验收，否则发起验收按钮为灰色。

  a. 录ModelArts管理控制台，在左侧菜单栏中选择“数据管理 >数据标注”，打开数据标注管理页。

  b. 在数据标注作业列表，选择团队标注作业，点击作业名称进入“标注作业详情页”，单击右上角“验收”，发起验收。

  c. 在弹出的对话框中，设置“抽样策略”，可设置为“按百分比”，也可以设置为“按数量”。设置好参数值后，单击“确定”启动验收。

     “按百分比”：按待验收图片总数的一定比例进行抽样验收。

     “按数量”：按一定数量进行抽样验收。

图 3-106 发起验收

d. 验收启动后，界面将展示实时验收报告，您可以在右侧选择“验收结果”（“通过”或“不通过”）。

当选择验收结果为“通过”时，需设置“验收评分”（分“A”、“B”、“C”、“D”四个选项，“A”表示最高分），如图3-108所示。当选择验收结果为“不通过”时，可以在文本框中写明驳回原因，如图3-109所示。

图 3-107 查看实时验收报告

图 3-108 设置验收结果为“通过”

图 3-109 设置验收结果为“不通过”

- **继续验收**

  针对未完成验收的任务，可以继续验收。针对未发起过验收流程的任务，不支持“继续验收”，按钮为灰色。

  在“标注任务进展”页签中，针对需继续验收的任务，单击“继续验收”。系统直接进入“实时验收报告”页面，您可以继续验收未验收的图片，设置其“验收结果”。

- **完成验收**

  在完成验收窗口，您可以查看本标注作业的验收情况，如抽样文件数等，同时设置如下参数，然后进行验收。只有完成验收，标注信息才会同步到标注作业的已标注页面中。

  一旦标注数据完成验收，团队成员无法再修改标注信息，只有数据集创建者可修改。

表 3-33 完成验收的参数设置

参数	说明
对已标注数据修改	• 不覆盖：针对同一个数据，不使用当前团队标注的结果覆盖已有数据。 • 覆盖：针对同一个数据，使用当前团队标注的结果覆盖已有数据。覆盖后无法恢复，请谨慎操作。
通过范围	• 全部：当前团队标注完成的所有数据。包含验收通过、未验收和验收不通过的。即本数据集的所有抽样文件数。 • 全部不通过：当前团队标注完成的所有数据不通过验收，即将所有标注数据驳回给标注人员。全部数据指验收通过、未验收和验收不通过的所有数据，即本数据集的所有抽样文件数。 • 验收通过和未验收的数据：针对抽样文件中验收通过和未验收的数据，通过验收。验收不通过的数据将驳回给标注人员。 • 验收通过的数据：针对抽样文件中验收通过的数据，通过验收。未验收和验收不通过的数据将驳回给标注人员。

图 3-110 完成验收

查看验收报告

针对进行中或已完成的标注任务，都可以查看其验收报告。在“标注任务进展”页签中，单击“验收报告”，即可在弹出的“验收报告”对话框中查看详情。

图 3-111 查看验收报告

删除标注任务

针对不再使用的标注任务，您可以在“标注任务进展”页签下，单击任务所在行的删除。任务删除后，未验收的标注详情将丢失，请谨慎操作。但是数据集中的原始数据以及完成验收的标注数据仍然存储在对应的OBS桶中。

3.16 数据处理

3.16.1 数据处理简介

ModelArts平台提供的数据处理功能，基本目的是从大量的、杂乱无章的、难以理解的数据中抽取或者生成对某些特定的人们来说是有价值、有意义的数据。当数据采集和接入之后，数据一般是不能直接满足训练要求的。为了保障数据质量，以免对后续操作（如数据标注、模型训练等）带来负面影响，开发过程通常需要进行数据处理。常见的数据处理类型有以下四种：

- 数据校验：通常数据采集后需要进行校验，保证数据合法。

  数据校验是指对数据可用性的基本判断和验证的过程。通常，我们采集的数据或多或少都会有很多格式问题，无法被进一步处理。以图像识别为例，用户经常会从网上找一些图片用于训练，但是其质量难以保证，有可能图片的名字、路径、后缀名都不满足训练算法的要求；图片也可能有部分损坏，造成无法解码、无法被算法处理的情况。因此，数据校验非常重要，可以帮助人工智能开发者提前发现数据问题，有效防止数据噪声造成的算法精度下降或者训练失败问题。

- 数据清洗：数据清洗是指对数据进行去噪、纠错或补全的过程。

  数据清洗是在数据校验的基础上，对数据进行一致性检查，处理一些无效值。例如在深度学习领域，可以根据用户输入的正样本和负样本，对数据进行清洗，保留用户想要的类别，去除用户不想要的类别。

- 数据选择：数据选择一般是指从全量数据中选择数据子集的过程。

  数据可以通过相似度或者深度学习算法进行选择。数据选择可以避免人工采集图片过程中引入的重复图片、相似图片等问题；在一批输入旧模型的推理数据中，通过内置规则的数据选择可以进一步提升旧模型精度。

- 数据增强：

  数据扩增通过简单的数据扩增例如缩放、裁剪、变换、合成等操作直接或间接的方式增加数据量。

  图像生成应用相关深度学习模型，通过对原数据集进行学习，训练生成新的数据集的方式增加数据量。

3.16.2 创建数据处理任务

您可以创建一个数据处理任务，对已有的数据进行数据校验、数据清洗、数据选择或者数据增强操作。

前提条件

- 数据已准备完成：已经创建数据集或者已经将数据上传至OBS。
- 确保您使用的OBS与ModelArts在同一区域。

创建数据处理任务

1. 登录ModelArts管理控制台，在左侧的导航栏中选择“数据管理>数据处理”，进入“数据处理”页面。
2. 在“数据处理”页面，单击“创建”进入“创建数据处理”页面。
3. 在创建数据处理页面，填写相关算法参数。

a. 填写基本信息。基本信息包括“名称”、“版本”和“描述”。其中“版本”信息由系统自动生成，按“V0001”、“V0002”规则命名，用户无法修改。

   您可以根据实际情况填写“名称”和“描述”信息。

   图 3-112 创建数据处理基本信息

b. 设置场景类别。场景类别当前支持“图像分类”和“物体检测”。

c. 设置数据处理类型。数据处理类型支持“数据清洗”、“数据校验”、“数据选择”和“数据增强”。

   针对不同的数据处理类型，您需要填写相应算子的设置参数，算子的详细参数参见**预置算子说明**。

   图 3-113 设置场景类别和数据处理类型

d. 设置输入与输出。需根据实际数据情况选择“数据集”或“OBS目录”。设置为“数据集”时，需填写“数据集名称”和“数据集版本”；设置为“OBS目录”时，需填写正确的OBS路径。

   图 3-114 输入输出设置-数据集

图 3-115 输入输出设置-OBS 目录

e. 确认参数填写无误后，单击“创建”，完成数据处理任务的创建。

3.16.3 管理和查看数据处理任务

删除数据处理任务

当已有的数据处理任务不再使用时，您可以删除数据处理任务。

处于“完成”、“失败”、“已停止”、“运行失败”、“部署中”状态的训练作业，您可以单击操作列的“删除”，删除对应的数据处理任务。

查看数据处理任务详情

1. 登录ModelArts管理控制台，在左侧的导航栏中选择“数据管理>数据处理”，进入“数据处理”页面。
2. 在数据处理列表中，单击数据处理任务名称，进入数据处理任务的版本管理页面。您可以在该页面进行数据处理任务的“修改”与“删除”。

图 3-116 数据处理版本管理页面

3. 您可以在版本管理页面，通过切换页签查看“配置信息”、“日志”和“结果展示”。

图 3-117 日志页面

图 3-118 结果展示页面

3.16.4 预置算子说明

3.16.4.1 数据校验

MetaValidation 算子概述

ModelArts的数据校验通过MetaValidation算子实现。当前ModelArts支持jpg、jpeg、bmp、png四种图片格式。物体检测场景支持xml标注格式，不支持“非矩形框”标注。针对您提供的数据集，MetaValidation算子支持对图片和xml文件进行数据校验：

表 3-34 图片类数据校验

异常情况	处理方案
图片本身损坏无法解码	过滤掉不能解码的图片
图片通道可能是1通道、2通道，不是常用的3通道	转换图片成RGB三通道
图片格式不在ModelArts支持的格式范围内	转换图片格式至jpg格式

异常情况	处理方案
图片后缀与实际格式不符，但格式在MA支持的格式内	后缀转换成与实际格式一致
图片后缀与实际格式不符，且格式不在MA支持的格式内	转换图片格式至jpg格式
图片分辨率过大	宽、高按指定大小同比例进行裁剪

表 3-35 标注类文件数据校验

异常情况	处理方案
xml结构残缺，无法解析	过滤xml文件
xml中没有标注“object”	过滤xml文件
xml中没有矩形框“bndbox”	过滤xml文件
某些标注“object”中没有矩形框“bndbox”	过滤标注“object”
图片经过裁剪后，xml文件中宽高不符	修改错误宽高参数为图片真实宽高
xml中没有“width”、“height”字段	根据图片真实宽高补全xml中的“width”、“height”字段和值
图片经过裁剪后，xml中矩形框“bndbox”大小不符	按图片裁剪比例缩放xml文件中“bnxbox”值
xml中矩形框“bndbox”宽或高值过小，显示为一条线	矩形框宽或高差值小于2，移除当前“object”
xml中矩形框“bndbox”最小值大于最大值	移除当前“object”
矩形框“bndbox”超出图片边界，且超出部分占框面积50%以上	移除当前“object”
矩形框“bndbox”超出图片边界，但超出部分小于框面积50%	矩形框“bndbox”拉回到图片边界

说明

数据校验过程不会改动原始数据，通过校验的图片或xml文件保存在指定的输出路径下。

参数说明

表 3-36 数据校验-MetaValidation 算子参数说明

参数名	是否必选	默认值	参数说明
image_max_width	否	-1	输入图片宽度最大值，若输入图片宽度超过设定值则按比例裁剪。单位为px。 默认值 -1 表示不做裁剪。
image_max_height	否	-1	输入图片长度最大值，若输入图片长度超过设定值则按比例裁剪。单位为px。 默认值 -1 表示不做裁剪。

输入要求

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在本任务中选择的场景类别一致。
- 选择“OBS目录”，存放结构又分两种情况，“仅包含图片”或“包含图片和标注信息”。
  - “仅包含图片”：当目录下全是图片时，支持jpg、jpeg、png、bmp格式，嵌套子目录的图片也将全部读入。
  - “包含图片和标注信息”：根据不同场景类型，结构不同。

    图像分类场景，其目录结构如下所示。如下目录结构，仅支持单标签场景。

    ```
    input_path/
      --label1/
          ----1.jpg
      --label2/
          ----2.jpg
      --../
    ```

    物体检测场景，其目录结构如下所示。支持jpg、jpeg、png、bmp格式的图片，xml为标准的PACAL VOC格式标注文件。

    ```
    input_path/
      --1.jpg
      --1.xml
      --2.jpg
      --2.xml
      ...
    ```

输出说明

- **图像分类**

  输出数据的目录结构如下所示。

  ```
  output_path/
    --Data/
        ----class1/  # 若输入数据有标注信息会一并输出，class1为标注类别
            ------1.jpg
            ------2_checked.jpg
        ----class2/
            ------3.jpg
  ```

```
        ------4_checked.jpg
      ----5_checked.jpg
  --output.manifest
```

其中manifest文件内容示例如下所示。会给每一条数据加上一个校验属性"property":{"@modelarts:data_checked":true}

```
{
  "id": "xss",
  "source": "obs://hard_example_path/Data/fc8e2688015d4a1784dcbda44d840307_14_checked.jpg",
  "property": {
    "@modelarts:data_checked": true
  },
  "usage": "train",
  "annotation": [
    {
      "name": "Cat",
      "type": "modelarts/image_classification"
    }
  ]
}
```

- **物体检测**

  在输出目录下，文件结构如下所示。

```
output_path/
  --Data/
      ----1_checked.jpg
      ----1_checked.xml  # 若输入数据在校验过程中经过了转换，文件名会加上'_checked'
      ----2.jpg          # 若输入数据未经过转换，则以原来的名字保存
      ----2.xml
  --output.manifest
```

其中manifest文件内容示例如下所示。会给每一条数据加上一个校验属性"property":{"@modelarts:data_checked":true}

```
{
  "source": "obs://hard_example_path/Data/be462ea9c5abc09f_checked.jpg",
  "property": {
    "@modelarts:data_checked": true
  },
  "annotation": [
    {
      "annotation-loc": "obs://hard_example_path/Data/be462ea9c5abc09f_checked.xml",
      "type": "modelarts/object_detection",
      "annotation-format": "PASCAL VOC",
      "annotated-by": "modelarts/hard_example_algo"
    }
  ]
}
```

3.16.4.2 数据清洗

PCC 算子概述

ModelArts的数据清洗通过PCC算子实现。图像分类或者物体检测的数据集中可能存在非所需类别的图像，需要将这些图像去除掉，以免对标注、模型训练造成干扰。

图 3-119 PCC 算子效果

参数说明

表 3-37 数据清洗-PCC 算子参数说明

参数名	是否必选	默认值	参数说明
prototype_sample_path	是	None	数据清洗正样例目录。目录应存放正样例图片文件，算法将这些图片为正样例，对输入中的数据进行过滤，即保留与“prototype_sample_path”目录下图片相似度高的数据。 请输入一个真实存在的OBS目录，且目录下已包含提供的正样例图片，且以obs://开头。如：*obs://obs_bucket_name/folder_name*
criticism_sample_path	否	None	数据清洗负样例目录。目录应存放负样例图片文件，算法将这些图片为负样例，对算法输入中的数据进行过滤， 即保留与“criticism_sample_path”目录下图片相似度差距较大的数据。 建议该参数和“prototype_sample_path”配合使用，可以提高数据清洗的准确性。 请输入一个真实存在的OBS目录，且以obs://开头。如：*obs://obs_bucket_name/folder_name*
n_clusters	否	auto	数据样本的种类数，默认值auto。您可以输入小于样本总数的整数或auto。auto表示使用正样本目录的图片个数作为数据样本的种类数。
simlarity_threshold	否	0.9	相似度阈值。两张图片相似程度超过阈值时，判定为相似图片，反之按非相似图片处理。输入取值范围为0~1。
embedding_distance	否	0.2	样本特征间距。两张图片样本特征间距小于设定值，判定为相似图片，反之按非相似图片处理。输入取值范围为0~1。

参数名	是否必选	默认值	参数说明
do_validation	否	True	是否做数据校验，可填True或者False。表示数据清洗前需要做数据校验，否则只做数据清洗。

输入要求

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在本任务中选择的场景类别一致。
- 选择“OBS目录”，存放结构又分两种情况，“仅包含图片”或“包含图片和标注信息”。
  - “仅包含图片”：当目录下全是图片时，支持jpg、jpeg、png、bmp格式，嵌套子目录的图片也将全部读入。
  - “包含图片和标注信息”：根据不同场景类型，结构不同。

    图像分类场景，其目录结构如下所示。如下目录结构，仅支持单标签场景。

    ```
    input_path/
        --label1/
            ----1.jpg
        --label2/
            ----2.jpg
        --../
    ```

    物体检测场景，其目录结构如下所示。支持jpg、jpeg、png、bmp格式的图片，xml为标准的PACAL VOC格式标注文件。

    ```
    input_path/
        --1.jpg
        --1.xml
        --2.jpg
        --2.xml
        ...
    ```

输出说明

- **图像分类**

  输出数据的目录结构如下所示。

  ```
  output_path/
      --Data/
          ----class1/  # 若输入数据有标注信息会一并输出，class1为标注类别
              ------1.jpg
          ----class2/
              ------2.jpg
          ----3.jpg
      --output.manifest
  ```

  其中manifest文件内容示例如下所示。

  ```
  {
      "id": "xss",
      "source": "obs://home/fc8e2688015d4a1784dcbda44d840307_14.jpg",
      "usage": "train",
      "annotation": [
          {
              "name": "Cat",
  ```

```
            "type": "modelarts/image_classification"
        }
    ]
}
```

- **物体检测**

  输出数据的目录结构如下所示。

```
output_path/
    --Data/
        ----1.jpg
        ----1.xml  # 若输入数据有标注信息会一并输出，xml为标注文件
        ----2.jpg
        ----3.jpg
    --output.manifest
```

  其中manifest文件内容示例如下所示。

```
{
    "source":"obs://fake/be462ea9c5abc09f.jpg",
    "annotation":[
        {
            "annotation-loc":"obs://fake/be462ea9c5abc09f.xml",
            "type":"modelarts/object_detection",
            "annotation-format":"PASCAL VOC",
            "annotated-by":"modelarts/hard_example_algo"
        }
    ]
}
```

3.16.4.3 数据选择

数据选择算子概述

ModelArts提供以下数据选择算子：

- SimDeduplication：可以依据用户设置的相似程度阈值完成图像去重处理。图像去重是图像数据处理常见的数据处理方法。图像重复指图像内容完全一样，或者有少量的尺度、位移、色彩、亮度变化，或者是添加了少量其他内容等。

图 3-120 SimDeduplication 效果图

表 3-38 高级参数说明

参数名	是否必选	默认值	参数说明
simlarity_threshold	否	0.9	相似程度阈值，两张图片间的相似度大于阈值时，其中一张会作为重复图片被过滤掉。取值范围为0~1。

参数名	是否必选	默认值	参数说明
do_validation	否	True	是否做数据校验，可填True或者False。表示数据去重前需要做数据校验，否则只做数据去重。

- RRD：可以依据用户设置的比例去除差异最大的数据。

图 3-121 RRD 效果图

表 3-39 高级参数说明

参数名	是否必选	默认值	参数说明
sample_ratio	否	0.9	数据留下的百分比。取值范围为0~1。例如0.9表示保留百分之90的原数据。
n_clusters	auto	auto	数据样本的种类数，默认为auto，即按照目录中图片个数取类别总数，可指定具体类别数，如 4
do_validation	否	True	是否做数据校验，可填True或者False。表示数据去冗余前需要做数据校验，否则只做数据去重。

输入要求

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在本任务中选择的场景类别一致。
- 选择“OBS目录”，存放结构又分两种情况，“仅包含图片”或“包含图片和标注信息”。
  - “仅包含图片”：当目录下全是图片时，支持jpg、jpeg、png、bmp格式，嵌套子目录的图片也将全部读入。
  - “包含图片和标注信息”：根据不同数据类型，结构不同。

    图像分类，其目录结构如下所示。如下目录结构，仅支持单标签场景。

```
input_path/
    --label1/
        ----1.jpg
    --label2/
        ----2.jpg
    --../
```

物体检测，其目录结构如下所示。支持jpg、jpeg、png、bmp格式的图片，xml为标准的PACAL VOC格式标注文件。

```
input_path/
    --1.jpg
    --1.xml
    --2.jpg
    --2.xml
    ...
```

输出说明

- **图像分类**

  输出数据的目录结构如下所示。

```
output_path/
    --Data/
        ----class1/  # 若输入数据有标注信息会一并输出，class1为标注类别
            ------1.jpg
        ----class2/
            ------2.jpg
            ------3.jpg
    --output.manifest
```

  其中manifest文件内容示例如下所示。

```
{
    "id": "xss",
    "source": "obs://home/fc8e2688015d4a1784dcbda44d840307_14.jpg",
    "usage": "train",
    "annotation": [
        {
            "name": "Cat",
            "type": "modelarts/image_classification"
        }
    ]
}
```

- **物体检测**

  输出数据的目录结构如下所示。

```
output_path/
    --Data/
        ----1.jpg
        ----1.xml  # 若输入数据有标注信息会一并输出，xml为标注文件
        ----2.jpg
        ----3.jpg
    --output.manifest
```

  其中manifest文件内容示例如下所示。

```
{
    "source":"obs://fake/be462ea9c5abc09f.jpg",
    "annotation":[
        {
            "annotation-loc":"obs://fake/be462ea9c5abc09f.xml",
            "type":"modelarts/object_detection",
            "annotation-format":"PASCAL VOC",
            "annotated-by":"modelarts/hard_example_algo"
            }
    ]
}
```

3.16.4.4 数据选择（难例）

算法概述

在实际业务场景中，模型维护是一个长期的过程，比如说按照每周、每月进行数据重训练，或者累计数据至一定量时进行定期的重训练。如果将全量的数据用于重训练，需要耗费较大的标注人力和训练耗时。为了提升模型维护效率，可以采用基于难例数据的重训练 。

难例筛选算法对全量数据进行分析并筛选，仅输出全量数据中少部分对于模型维护有价值的数据。基于筛选后的数据进行重训练，可以有效减少标注人力和训练耗时。

难例筛选算法中融合了多种方法，要达到最佳效果，需要根据实际数据选择部分或全部方法，并调整其权重。

参数说明

参数名	是否必选	默认值	参数说明
source_service	Y	inference	难例任务的前置数据来源，目前仅支持inference。此参数不可修改。
filter_func	Y	comprehensive_mining	难例筛选算法设置为“comprehensive_mining”。此参数不可修改。
checkpoint_path	Y	/home/work/user-job-dir/data_filter/resnet_v1_50	用于特征提取的模型目录，当前仅支持基于Imagenet预训练的resnet_v1_50模型，此参数目前暂时不可修改。
model_serving_url	N	None	推理模型路径。得出推理结果的模型文件路径, 即训练作业的输出路径。该模型会用于aug_consistent_mining算法中的数据增强再推理。 请输入一个真实存在的OBS目录，例如：obs://obs_bucket_name/folder_name/
train_data_path	N	None	训练数据集，model_serving_url模型使用的训练数据，需输入数据集版本生成的manifest。 请输入一个真实存在的OBS目录，例如：obs://obs_bucket_name/folder_name/v001.manifest

参数名	是否必选	默认值	参数说明
comprehensive_algo_config	N	clustering_mining: 0.2020+aug_consistent_mining: 0.4265+feature_distribution_mining: 0.0451+sequential_mining: 0.425+image_similarity_mining: 0.0949+predict_score_mining: 0.3900+anomaly_detection_mining:0.2020	使用的算法及其权重，默认使用系统实验后效果最佳参数，也可以根据不同的数据自行配置。 例如：predict_score_mining: 0.3900+anomaly_detection_mining: 0.2020
algo_hard_threshold	N	0.1	筛选系数的阈值设置。取值范围0~1。 由于过高阈值可能导致输出结果为0，建议合理填写。
aug_op_config	N	crop:0.1+fliplr: 0.1+gaussianblur:0.1	aug_consistent_mining算法中用到的增强手段，支持crop、fliplr、gaussianblur、flipud、scale、translate、shear、superpixels、sharpen、add、invert。

参数名	是否必选	默认值	参数说明
feature_op_config	N	image_aspect_ratio: 0.5+image_brightness: 1.0+image_saturation: 0.5+image_resolution: 0.5+image_colorfulness: 0.5+ambiguity: 1.0+bbox_num: 1.0+bbox_iou: 1.0+bbox_std: 0.5+bbox_bright: 0.5+bbox_ambiguity: 0.5+bbox_aspect_ratio: 1.0+bbox_area_ratio: 0.5+bbox_edge_value:0.5	feature_distribution_mining算法中定义的特征，可自行修改权重。
score_threshold_up	N	0.6	predict_score_mining算法中定义的置信度最高值。取值范围0~1。
score_threshold_low	N	0.3	predict_score_mining算法中定义的置信度最低值。取值范围0~1。
margin	N	0.8	top2置信度差值定义, 差值超过该阈值，则为难例。取值范围0~1，默认值为0.8。
similarity_sample_ratio	N	1.0	image_similarity_mining算法中的相似比例定义。取值范围0~1，默认值为1.0。
task_summary_file	N	None	算法简化的日志输出路径及日志文件。请填写一个真实存在的OBS目录，以“obs://”开头。文件名称可自行定义。 例如：obs://obs_bucket_name/folder_name/xxx.log

参数名	是否必选	默认值	参数说明
output_dataset_type	N	manifest	支持directory和manifest类型。 ● directory：将原始图片和标签输出到结果目录的Data文件夹下。 ● manifest：仅输出manifest文件。 该参数在数据处理模块将根据页面选择自动填充。

输入要求

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在本任务中选择的场景类别一致。
- 选择“OBS目录”，目录下需包含用于推理的原始图片以及推理结果文件

  其目录结构如下所示。

```
input_path/
   --images/ # 文件夹名称必须为images
       ----1.jpg
       ----2.jpg
   --inference_results/  # 文件夹名称必须为inference_results
       ----1.jpg_result.txt
       ----2.jpg_result.txt
```

  其中，推理结果文件txt的内容需满足如下格式。如果使用ModelArts预置算法训练得到的模型进行推理，默认的推理结果已满足要求。

  – **图像分类**

```
{
   "predicted_label": "dog",
   "scores": [
      [
         "dog",
         "0.589"
      ],
      [
         "cat",
         "0.411"
      ]
   ]
}
```

  – **物体检测**

```
{
   "detection_classes": [
      "cat",
      "cat"
      ],
   "detection_boxes": [
      [
         117.56356048583984,
         335.9902648925781,
         270.50848388671875,
         469.0136413574219
      ],
      [
         18.747316360473633,
         13.10757064819336,
```

```
            217.25146484375,
            108.3551025390625
        ]
    ],
    "detection_scores": [
            0.5179755091667175,
            0.46941104531288147
        ]
}
```

输出说明

- 目标检测

  输出目录结构如下所示。

```
output_path :
    --Data
        ----1.jpg
        ----1.xml      # 导出筛选结果至该目录下
    --output.manifest
```

  其中manifest文件示例如下所示。

```
{"source":"/tmp/test_out/object_detection/images/be462ea9c5abc09f.jpg",
"hard":"True",
"hard-reasons":"0",  # 判定该样本为难例的原因，具体原因目前只在智能标注模块展示
"hard-coefficient":"1.0", # 难例算法得到的难例系数，越大代表可能是难例的概率越大
"annotation":[
{"annotation-loc":"/tmp/test_out/object_detection/annotations/be462ea9c5abc09f.xml",
"type":"modelarts/object_detection",
"annotation-format":"PASCAL VOC",
"annotated-by":"modelarts/hard_example_algo"}]}
```

- 图像分类

  输出目录结构如下所示。

```
output_path :
    --Data
        ----class1
            ------1.jpg
        ----class2
            ------2.jpg
    --output.manifest
```

  其中manifest文件示例如下所示。

```
{"source":"obs://obs_bucket_name/folder_name/catDog/5.jpg",
"hard":true,
"hard-reasons":"1-20-2-19-21-3",
"hard-coefficient":1.0,
"annotation":[
{"name":"cat",
"type":"modelarts/image_classification",
"confidence":0.599,
"annotated-by":"modelarts/hard_example_algo"}]}
```

日志文件说明

task_summary_file 为简化日志的输出文件路径，内容如下：

```
{
"task_status": 'SUCCEED', # 算法执行状态
"total_sample": integer,  # 输入样本总数
"hard_sample": integer     # 输出样本总数
}
```

或者是

```
{
"task_status": 'FAILED',
```

```
"error_message": 'xxxxxx' # 导致算法执行失败的异常信息
}
```

3.16.4.5 数据增强（数据扩增）

数据扩增算子概述

数据扩增主要用于训练数据集不足或需要仿真的场景，能通过对已标注的数据集做变换操作来增加训练图片的数量，同时会生成相应的标签。在深度学习领域，增强有重要的意义，能提升模型的泛化能力，增加抗扰动的能力。数据扩增过程不会改动原始数据，扩增后的图片或xml文件保存在指定的输出路径下。

ModelArts提供以下数据扩增算子：

表 3-40 数据扩增算子介绍

算子	算子说明	高级
AddNoise	添加噪声，模拟常见采集设备在采集图片过程中可能会产生的噪声。	● noise_type：添加噪声的分布类型，Gauss为高斯噪声，Laplace为拉普拉斯噪声，Poisson是泊松噪声，Impulse是脉冲噪声，SaltAndPepper为椒盐噪声。默认值为Gauss ● loc：噪声分布的均值，仅在Gauss和Laplace生效。默认值为0 ● scale：噪声分布的标准差，仅在Gauss和Laplance生效。默认值为1 ● lam：泊松分布的lambda系数，仅在Poisson有效。默认值为2 ● p：对于每个像素点，出现脉冲噪声或椒盐噪声的概率，仅在Impulse和SaltAndPepper有效。默认值为0.01 ● do_validation：数据扩增前是否做数据校验。默认值为True。
Blur	模糊，使用滤波器对图像进行滤波操作，有时用于模拟成像设备的成像。	● blur_type：可选Gauss和Average两种模式，分别为高斯和均值滤波。默认值为Gauss ● do_validation：数据扩增前是否做数据校验。默认值为True。
Crop	图片裁剪，随机裁剪图片的一部分作为新的图片。	● crop_percent_min：各边裁剪占比的随机取值范围的最小值。默认值为0.0 ● crop_percent_max：各边裁剪占比的随机取值范围的最大值。默认值为0.2 ● do_validation：数据扩增前是否做数据校验。默认值为True。
CutOut	随机擦除，在深度学习中常用的方法，用于模拟物体被障碍物遮挡。	do_validation：数据扩增前是否做数据校验。默认值为True。

算子	算子说明	高级
Flip	翻转，沿图片水平轴或竖直轴做翻转，是非常常见的增强方法。	● lr_ud：选择翻转的方向,lr为水平翻转，ud为竖直翻转。默认值为lr ● flip_p：做翻转操作的概率。默认值为1。 ● do_validation：数据扩增前是否做数据校验。默认值为True。
Grayscale	图片灰度化，将三通道的彩色图像转换到三通道的灰度图像。	do_validation：数据扩增前是否做数据校验。默认值为True。
HistogramEqual	直方图均衡化，多半是使用于让图片的视觉效果更加好，在某些场景下会使用。	do_validation：数据扩增前是否做数据校验。默认值为True。
LightArithmetic	亮度增强 ，对亮度空间做线性增强操作。	do_validation：数据扩增前是否做数据校验。默认值为True。
LightContrast	亮度对比度增强，使用一定的非线性函数改变亮度空间的亮度值。	func：默认值为gamma ● gamma为常见方法伽马矫正，公式为255*((v/255)**gamma)') ● sigmoid为函数为S型曲线，公式为255*1/(1+exp(gain*(cutoff-I_ij/255)))') ● log为对数函数，公式为255*gain*log_2(1+v/255) ● linear为线性函数，公式为127 + alpha*(v-127)') do_validation：数据扩增前是否做数据校验。默认值为True。
MotionBlur	运动模糊，模拟物体运动时产生的残影现象。	do_validation：数据扩增前是否做数据校验。默认值为True。
Padding	图片填充，在边缘添加黑色的边。	● px_top：图像顶端增加的像素行数。默认值为1 ● px_right：图像右侧增加的像素行数。默认值为1 ● px_left：图像左侧增加的像素行数。默认值为1 ● px_bottom：图像底侧增加的像素行数。默认值为1 ● do_validation：数据扩增前是否做数据校验。默认值为True。

算子	算子说明	高级
Resize	调整图片大小。	● height：变换后的图片高度。默认值224 ● width：变换后的图片宽度。默认值224 ● do_validation：数据扩增前是否做数据校验。默认值为True。
Rotate	旋转，将图像围绕中心点旋转的操作，操作完成之后保持图片原本的形状不变，不足的部分用黑色填充。	● angle_min：旋转角度随机取值范围的最小值，每张图片会从范围中随机取值作为自己的参数。默认值为90° ● angle_max：旋转角度随机取值范围的最大值，每张图片会从范围中随机取值作为自己的参数。默认值为-90° ● do_validation：数据扩增前是否做数据校验。默认值为True。
Saturation	色度饱和度增强，对图片的HSV中的H和S空间做线性的变化，改变图片的色度和饱和度。	do_validation：数据扩增前是否做数据校验。默认值为True。
Scale	图片缩放，将图片的长或宽随机缩放到一定倍数。	● scaleXY：缩放方向，X为水平，Y为垂直。默认值为X ● scale_min：缩放比例随机取值范围的最小值。默认为0.5 ● scale_max：缩放比例随机取值范围的最大值。默认值为1.5 ● do_validation：数据扩增前是否做数据校验。默认值为True。
Sharpen	图像锐化，用于将边缘清晰化，让物体边缘更加明显。	do_validation：数据扩增前是否做数据校验。默认值为True。
Shear	图片错切，一般用于图片的几何变换，通过线性函数将像素点进行映射。	● shearXY：错切方向，X为水平，Y为竖直。默认值为X ● shear_min：错切角度随机取值范围的最小值。默认值为-30 ● shear_max：错切角度随机取值范围的最大值。默认值为30 ● do_validation：数据扩增前是否做数据校验。默认值为True。
Translate	图片平移，将图片整体像X轴或Y轴平移，超出原图部分舍弃，丢失部分用黑色填充。	● translateXY：平移的方向，X为水平，Y为竖直。默认值为X ● do_validation：数据扩增前是否做数据校验。默认值为True。

算子	算子说明	高级
Weather	添加天气，模拟天气效果。	weather_mode：添加天气的模式，默认值为Rain。 ● Rain：下雨 ● Fog：雾 ● Snow：雪 ● Clouds：云 do_validation：数据扩增前是否做数据校验。默认值为True。

输入要求

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在本任务中选择的场景类别一致。
- 选择“OBS目录”，存放结构支持“包含图片和标注信息”模式。

  “包含图片和标注信息”，根据不同场景类型，结构不同。

  图像分类场景，其目录结构如下所示。如下目录结构，仅支持单标签场景。

```
input_path/
    --label1/
        ----1.jpg
    --label2/
        ----2.jpg
    --../
```

  物体检测场景，其目录结构如下所示。支持jpg、jpeg、png、bmp格式的图片，xml为标准的PACAL VOC格式标注文件。

```
input_path/
    --1.jpg
    --1.xml
    --2.jpg
    --2.xml
    ...
```

输出说明

由于算法中有些操作将会舍弃一些数据，输出文件夹里可能不包含全量数据集。例如，“Rotate”会舍弃标注框超出原始图片边界的图片。

输出目录结构如下所示。其中“Data”文件夹用于存放新生成的图片和标注信息，“manifest”文件存储文件夹中图片的结构，可直接导入到数据管理的数据集中。

```
|----data_url
    |----Data
        |----xxx.jpg
        |----xxx.xml(xxx.txt)
    |----output.manifest
```

其中manifest文件内容示例如下所示。

```
{
    "id": "xss",
    "source": "obs://home/fc8e2688015d4a1784dcbda44d840307_14.jpg",
```

```
  "usage": "train",
  "annotation": [
    {
      "name": "Cat",
      "type": "modelarts/image_classification"
    }
  ]
}
```

3.16.4.6 数据增强（图像生成）

图像生成算子概述

图像生成算子利用Gan网络依据已知的数据集生成新的数据集。Gan是一个包含生成器和判别器的网络，生成器从潜在空间中随机取样作为输入，其输出结果需要尽量模仿训练集中的真实样本。判别器的输入则为真实样本或生成网络的输出，其目的是将生成网络的输出从真实样本中尽可能分辨出来。而生成网络则要尽可能地欺骗判别网络。两个网络相互对抗、不断调整参数，最终目的是使判别网络无法判断生成网络的输出结果是否真实。训练中获得的生成器网络可用于生成与输入图片相似的图片，用作新的数据集参与训练。基于Gan网络生成新的数据集不会生成相应的标签。图像生成过程不会改动原始数据，新生成的图片或xml文件保存在指定的输出路径下。

ModelArts提供两种类型的图像生成算子：

- CycleGan算子：基于CycleGAN用于生成域迁移的图像，即将一类图片转换成另一类图片，把X空间中的样本转换成Y空间中的样本。CycleGAN可以利用非成对数据进行训练。模型训练时运行支持两个输入，分别代表数据的原域和目标域，在训练结束时会生成所有原域向目标域迁移的图像。

图 3-122 CycleGan 算子

表 3-41 CycleGan 算子高级参数

参数名	默认值	参数说明
do_validation	True	是否做数据校验，默认为True，表示数据生成前需要做数据校验，否则只做数据生成。
image_channel	3	生成图像的通道数。
image_height	256	图像相关参数：生成图像的高，大小需要是2的次方。
image_width	256	图像相关参数：生成图像的宽，大小需要是2的次方

参数名	默认值	参数说明
batch_size	1	训练相关参数：批量训练样本个数。
max_epoch	100	训练相关参数：训练遍历数据集次数。
g_learning_rate	0.0001	训练相关参数：生成器训练学习率。
d_learning_rate	0.0001	训练相关参数：判别器训练学习率。
log_frequency	5	训练相关参数：日志打印频率（按step计数）。
save_frequency	5	训练相关参数：模型保存频率（按epoch计数）。
predict	False	是否进行推理预测，默认为False。如果设置True，需要在resume参数设置已经训练完成的模型的obs路径。
resume	empty	如果predict设置为True，需要填写Tensorflow模型文件的obs路径用于推理预测。当前仅支持“.pb”格式的模型。示例：obs://xxx/xxxx.pb。 默认值为empty。

- StyleGan算子：基于StyleGan2用于在数据集较小的情形下，随机生成相似图像。StyleGAN提出了一个新的生成器结构，能够控制所生成图像的高层级属性(high-level attributes)，如发型、雀斑等；并且生成的图像在一些评价标准上得分更好。而本算法又增加了数据增强算法，可以在较少样本的情况下也能生成较好的新样本，但是样本数尽量在70张以上，样本太少生成出来的新图像不会有太多的样式。

图 3-123 StyleGan 算子

表 3-42 StyleGan 算子高级参数

参数名	默认值	参数说明
resolution	256	生成正方形图像的高宽，大小需要是2的次方。

参数名	默认值	参数说明
batch-size	8	批量训练样本个数。
total-kimg	300	总共训练的图像数量为total_kimg*1000。
generate_num	300	生成的图像数量，如果是多个类的，则为每类生成的数量。
predict	False	是否进行推理预测，默认为False。如果设置True，需要在resume参数设置已经训练完成的模型的obs路径。
resume	empty	如果predict设置为True，需要填写Tensorflow模型文件的obs路径用于推理预测。当前仅支持“.pb”格式的模型。示例：obs://xxx/xxxx.pb。 默认值为empty。
do_validation	True	是否做数据校验，默认为True，表示数据生成前需要做数据校验，否则只做数据生成。

数据输入

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在本任务中选择的场景类别一致。
- 选择“OBS目录”，图像生成算子不需要标注信息，输入支持单层级或双层级目录，存放结构支持“单层级”或“双层级”模式。

单层级目录结构如下所示：

```
image_folder----0001.jpg
          ----0002.jpg
          ----0003.jpg
          ...
          ----1000.jpg
```

双层级目录结构如下所示：

```
image_folder----sub_folder_1----0001.jpg
                        ----0002.jpg
                        ----0003.jpg
                        ...
                        ----0500.jpg
          ----sub_folder_2----0001.jpg
                        ----0002.jpg
                        ----0003.jpg
                        ...
                        ----0500.jpg
                        ...
          ----sub_folder_100----0001.jpg
                          ----0002.jpg
                          ----0003.jpg
                          ...
                          ----0500.jpg
```

输出说明

输出目录的结构如下所示。其中“model”文件夹存放用于推理的“frozen pb”模型，“samples”文件夹存放训练过程中输出图像，“Data”文件夹存放训练模型生成的图像。

```
train_url----model----CYcleGan_epoch_10.pb
                 ----CYcleGan_epoch_20.pb
                 ...
                 ----CYcleGan_epoch_1000.pb
        ----samples----0000_0.jpg
                 ----0000_1.jpg
                 ...
                 ----0100_15.jpg
        ----Data----CYcleGan_0_0.jpg
                 ----CYcleGan_0_1.jpg
                 ...
                 ----CYcleGan_16_8.jpg
        ----output_0.manifest
```

其中manifest文件内容示例如下所示。

```
{
    "id": "xss",
    "source": "obs://home/fc8e2688015d4a1784dcbda44d840307_14.jpg",
    "usage": "train",
    "annotation": [
        {
            "name": "Cat",
            "type": "modelarts/image_classification"
        }
    ]
}
```

4 数据管理（旧版即将下线）

4.1 数据管理简介

说明

当前ModelArts同时存在新版数据集和旧版数据集。本文档主要介绍旧版数据集及数据管理相关功能。

旧版数据集即将下线，推荐用户使用新版数据集及相关功能，具体请参考数据准备与分析、数据标注、数据处理。

新版数据集在旧版的基础上将创建数据集和创建标注任务进行了解耦，创建数据集和创建标注作业分别是独立的任务，使用更灵活。

旧版数据集需要在创建数据集时创建标注任务，不支持分开单独创建数据集和数据标注任务。

在ModelArts中，您可以在“数据管理”页面，完成数据导入、数据标注等操作，为模型构建做好数据准备。ModelArts以数据集为数据基础，进行模型开发或训练等操作。

数据集的类型

当前ModelArts支持如下类型的数据集。包含图片、音频、文本、表格、视频和其他类别。

- 图片
  - 图像分类：识别一张图片中是否包含某种物体。
  - 物体检测：识别出图片中每个物体的位置及类别。
  - 图像分割：识别出图片中每个物体的轮廓。
- 音频
  - 声音分类：对声音进行分类。
  - 语音内容：对语音内容进行标注。
  - 语音分割：对语音进行分段标注。
- 文本
  - 文本分类：对文本的内容按照标签进行分类处理。
  - 命名实体：针对文本中的实体片段进行标注，如“时间”、“地点”等。
  - 文本三元组：针对文本中的实体片段和实体之间的关系进行标注。

- 表格
  - 表格：适合表格等结构化数据处理。文件格式支持csv。不支持标注，支持对部分表格数据进行预览，但是最多支持100条数据预览。
- 视频
  - 视频标注：识别出视频中每个物体的位置及分类。目前仅支持mp4格式。
- 其他
  - 自由格式：管理的数据可以为任意格式，目前不支持标注，适用于无需标注或开发者自行定义标注的场景。如果您的数据集需存在多种格式数据，或者您的数据格式不符合其他类型数据集时，可选择自由格式的数据集。

图 4-1 自由格式数据集示例

File	Size	Format
image003_1594105979724.jpeg	90.41 KB	jpeg
image005_1594105981161.gif	341.28 KB	gif
iris_1592881356613.csv	2.36 KB	csv
train_1592881356925.csv	8.26 KB	csv
vedio_1594106046159.MP4	430.71 KB	mp4

规格限制

- 除图片类型之外的数据集（如视频、文本、音频等），单个样本大小限制：5GB
- 针对图片类数据集（物体检测、图像分类、图像分割），单个图片大小限制：25MB
- 单个manifest文件大小限制：5GB
- 文本文件单行大小限制：100KB
- 数据管理标注结果文件大小限制：100MB

数据集管理流程及功能简介

图 4-2 标注管理全流程

表 4-1 功能介绍

功能	说明
创建数据集（旧版）	创建一个新的数据集。
图像分类 物体检测 文本分类 命名实体 文本三元组 声音分类 语音内容 语音分割 视频标注	针对不同类型的数据集，对数据进行标注。“自由格式”和“表格”类型的数据集暂不支持数据标注。
导入操作	将数据导入数据集中。
导出数据	支持将部分数据导出为新的数据集或者将数据导出至OBS。同时支持对任务历史进行查看和管理。
修改数据集	修改数据集的基本信息。如数据集名称、描述或标签等信息。
发布数据集	将标注后的数据集发布为新版本，以便应用于后续的模型构建。
管理数据集版本	通过数据集版本查看演进过程。
智能标注	支持对未标注的数据快速完成数据标注，为您节省70%以上的标注时间。
自动分组	您可以针对您选中的数据，执行自动分组，提升您的数据标注效率。
数据特征	对数据进行特征分析，帮助您了解数据。
团队标注简介	支持多人标注同一个数据集，且支持数据集创建者统一管理标注任务。添加团队及其成员，参与到数据集的标注工作。
数据处理	为了保障数据质量，以免对后续操作（如数据标注、模型训练等）带来负面影响，开发过程通常需要进行数据处理。常见的数据处理类型有：数据校验、数据清洗、数据选择、数据增强。
删除数据集	删除数据集以释放资源。

不同类型数据集支持的功能列表

其中，不同类型的数据集，支持不同的功能，详细信息请参见表4-2。

表 4-2 不同类型的数据集支持的功能

数据集类型	创建数据集	导入数据	导出数据	发布数据集	修改数据集	管理版本	智能标注	团队标注	自动分组	数据特征	一键模型上线
图像分类	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持
物体检测	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持
图像分割	支持	支持	支持	支持	支持	支持	-	-	支持	-	-
声音分类	支持	支持	-	支持	支持	支持	-	-	-	-	-
语音内容	支持	支持	-	支持	支持	支持	-	-	-	-	-
语音分割	支持	支持	-	支持	支持	支持	-	支持	-	-	-
文本分类	支持	支持	-	支持	支持	支持	-	支持	-	-	-
命名实体	支持	支持	-	支持	支持	支持	-	支持	-	-	-
文本三元组	支持	支持	-	支持	支持	支持	-	支持	-	-	-
表格	支持	支持	-	支持	支持	支持	-	-	-	-	-
视频	支持	支持	-	支持	支持	支持	-	-	-	-	-
自由格式	支持	-	支持	支持	支持	支持	-	-	-	-	-

4.2 创建数据集（旧版）

在ModelArts进行数据管理时，首先您需要创建一个数据集，后续的操作，如标注数据、导入数据、数据集发布等，都是基于您创建的数据集。

说明

当前ModelArts同时存在新版数据集和旧版数据集。

新版数据集在旧版的基础上将创建数据集和创建标注任务进行了解耦，创建数据集和创建标注作业分别是独立的任务，使用更灵活。

旧版数据集需要在创建数据集时创建标注任务，不支持分开单独创建数据集和数据标注任务。

本文档主要介绍旧版数据集创建流程。新版数据集创建，请参考**创建数据集（New）**。

前提条件

- 数据管理功能需要获取访问OBS权限，在未进行委托授权之前，无法使用此功能。在使用数据管理功能之前，请前往“全局配置”页面，使用委托完成访问授权。
- 已创建用于存储数据的OBS桶及文件夹。并且，数据存储的OBS桶与ModelArts在同一区域。
- 需要使用的数据已上传至OBS。详细指导请参见**如何上传数据至OBS**。

操作步骤

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据集”，进入“数据集”管理页面。
2. 单击“创建数据集”，进入“创建数据集”页面，根据数据类型以及数据标注要求，选择创建不同类型的数据集。
   a. 填写数据集基本信息，数据集的“名称”和“描述”。

   图 4-3 数据集基本信息

   b. 根据您的需求，选择“标注场景”和“标注类型”，ModelArts目前支持的类型及其说明请参见**数据集的类型**。

   图 4-4 选择标注场景和标注类型

   c. 针对不同类型的数据集，需填写参数不同，请参考如下类型数据集对应的参数介绍。
      - **图片（图像分类、物体检测、图像分割）**
      - **音频（声音分类、语音内容、语音分割）**
      - **文本（文本分类、命名实体、文本三元组）**
      - **表格**
      - **视频**

- 其他（自由格式）

d. 参数填写无误后，单击页面右下角“创建”。

数据集创建完成后，系统自动跳转至数据集管理页面，针对创建好的数据集，您可以执行标注数据、发布、版本管理、修改、导入和删除等操作。不同类型数据集，支持的操作请参见不同类型数据集支持的功能列表

图片（图像分类、物体检测、图像分割）

图 4-5 图像分类和物体检测类型的参数

表 4-3 数据集的详细参数

参数名称	说明
数据集输入位置	选择数据集输入位置的OBS路径。 **说明** 创建数据集时，此OBS路径下的数据会导入数据集，后续若直接在OBS中修改数据，会造成数据集的数据与OBS的数据不一致，可能导致部分数据不可用。如果需要在数据集中修改数据，建议使用同步数据源或导入操作功能。
数据集输出位置	选择数据集输出位置的OBS路径。 **说明** “数据集输出位置”不能与“数据集输入位置”为同一路径，且不能是“数据集输入位置”的子目录。“数据集输出位置”最好选择一个空目录。

参数名称	说明
添加标签集	• **设置标签名称**：在标签名称文本框中，输入标签名称。标签名称只能是中文、字母、数字、下划线或中划线组成的合法字符串。长度为1～32字符。 • **添加标签**：单击“添加标签”可增加多个标签。 • **设置标签颜色**：仅“物体检测”类型数据集需设置此参数。在每个标签右侧的标签颜色区域下，可在色板中选择颜色，或者直接输入十六进制颜色码进行设置。 • **设置标签属性**：针对“物体检测”类型数据集，在设置完标签颜色后，可在右侧单击加号，增加对应的标签属性。标签属性用于区分同一标签物体的不同属性。例如，黄色小猫、黑色小猫。标签为cat，颜色为不同的标签属性。
启用团队标注	选择是否启用团队标注。图像分割暂不支持团队标注，当选择图像分割类型时，界面不显示此参数。 启用团队标注功能，需填写对应的团队标注任务“名称”、“类型”，同时选择对应的“标注团队”及参与标注的“团队成员”。参数详细介绍请参见创建团队标注任务。 在启用“团队标注”前，需确保您已经在“标注团队”管理页面，添加相应的团队以及成员。如果没有标注团队，可直接从界面链接跳转至“标注团队”页面，添加您的团队并为其添加成员。详细指导请参见团队标注简介。 启用团队标注功能的数据集，在创建完成后，可以在“标注类型”中看到“团队标注”的标识。

音频（声音分类、语音内容、语音分割）

图 4-6 声音分类、语音内容、语音分割类型数据集的参数

参数名称	说明
数据集输入位置	选择数据集输入位置的OBS路径。
数据集输出位置	选择数据集输出位置的OBS路径。 **说明** “数据集输出位置”不能与“数据集输入位置”为同一路径，且不能是“数据集输入位置”的子目录。“数据集输出位置”最好选择一个空目录。
添加标签集（声音分类）	仅“声音分类”类型的数据集需设置标签。 ● **设置标签名称**：在标签名称文本框中，输入标签名称。标签名称只能是中文、字母、数字、下划线或中划线组成的合法字符串。长度为1～32字符。 ● **添加标签**：单击“添加标签”可增加多个标签。

参数名称	说明
标签管理（语音分割）	仅“语音分割”类型的数据集，支持多种标签。 ● **单标签** 单标签适用于一段音频标注只有一种类别的音频，通常标注一个标签。 – **设置标签名称**：在“标签名”列输入标签名称。标签名称只能是中文、字母、数字、下划线或中划线组成的合法字符串。长度为1～32字符。 – **设置标签颜色**：在“标签颜色”列设置标签颜色。可在色板中选择颜色，或者直接输入十六进制颜色码进行设置。 ● **多标签** 多标签适用于多维度标注，例如在一段音频标注噪音与人说话的声音两种类别，其中说话的声音还可以标注为不同人的声音。单击“新建标签类别”可添加多个标签类别，一个标签类别可以包含多个标签。“标签类别”和“标签名”只能是中文、字母、数字、下划线或中划线组成的合法字符串。长度为1～32字符。 – **设置标签类别**：在“标签类别”输入标签类别的名称。 – **设置标签名称**：在“标签名”输入标签名称。 – **添加标签**：单击“添加标签”可增加多个标签。
启用语音内容标注（语音分割）	仅“语音分割”类型数据集支持设置，默认关闭。如果启用此功能，支持针对语音内容进行标注。
启用团队标注	仅“语音分割”类型支持团队标注，因此选择创建语音分割类型时，支持设置是否启用团队标注。 启用团队标注功能，需填写对应的团队标注任务“名称”、“类型”，同时选择对应的“标注团队”及参与标注的“团队成员”。参数详细介绍请参见创建团队标注任务。 在启用“团队标注”前，需确保您已经在“标注团队”管理页面，添加相应的团队以及成员。如果没有标注团队，可直接从界面链接跳转至“标注团队”页面，添加您的团队并为其添加成员。详细指导请参见团队标注简介。 启用团队标注功能的数据集，在创建完成后，可以在“标注类型”中看到“团队标注”的标识。

文本（文本分类、命名实体、文本三元组）

图 4-7 文本分类、命名实体、文本三元组类型数据集的参数

表 4-4 数据集的详细参数

参数名称	说明
数据集输入位置	选择数据集输入位置的OBS路径。 **说明** 文本分类数据只能在执行导入数据操作时识别已标注数据，在此处创建数据集时不能识别已标注数据，建议此处的OBS设置为一个空目录，创建完数据集后再导入已标注数据，导入操作的数据格式要求请参见**从OBS目录导入的规范说明**。
数据集输出位置	选择数据集输出位置的OBS路径。 **说明** “数据集输出位置”不能与“数据集输入位置”为同一路径，且不能是“数据集输入位置”的子目录。“数据集输出位置”最好选择一个空目录。

参数名称	说明
添加标签集（文本分类、命名实体）	• **设置标签名称**：在标签名称文本框中，输入标签名称。标签名称只能是中文、字母、数字、下划线或中划线组成的合法字符串。长度为1～32字符。 • **添加标签**：单击“添加标签”可增加多个标签。 • **设置标签颜色**：在每个标签右侧的标签颜色区域下，可在色板中选择颜色，或者直接输入十六进制颜色码进行设置。
添加标签集（文本三元组）	针对“文本三元组”类型的数据集，需要设置实体标签和关系标签。 • **实体标签**：需设置标签名以及标签颜色。可在颜色区域右侧单击加号增加多个标签。 • **关系标签**：关系标签为两个实体之间的关系。需设置起始实体和终止实体，您需要先添加至少2个实体标签后，再添加关系标签。
启用团队标注	选择是否启用团队标注。 启用团队标注功能，需填写对应的团队标注任务“名称”、“类型”，同时选择对应的“标注团队”及参与标注的“团队成员”。参数详细介绍请参见创建团队标注任务。 在启用“团队标注”前，需确保您已经在“标注团队”管理页面，添加相应的团队以及成员。如果没有标注团队，可直接从界面链接跳转至“标注团队”页面，添加您的团队并为其添加成员。详细指导请参见团队标注简介。 启用团队标注功能的数据集，在创建完成后，可以在“标注类型”中看到“团队标注”的标识。

表格

图 4-8 表格类型的参数

说明

使用CSV文件时，需要注意以下两点：

- 当数据类型选择String时，默认会把双引号内的数据当作一条，所以同一行数据需要保证双引号闭环，否则会导致数据过大，无法显示。
- 当CSV文件的某一行的列数与定义的Schema不同，则会忽略当前行。

表 4-5 数据集的详细参数

参数名称	说明
存储路径	选择表格数据存储路径（OBS路径），此位置会存放由数据源导入的数据。此位置不能和OBS数据源中的文件路径相同或为其子目录。 创建表格数据集后，在存储路径下会自动生成以下4个目录。 ● annotation：版本发布目录，每次发布版本，会在此目录下生成和版本名称相同的子目录。 ● data：数据存放目录，导入的数据会放在此目录。 ● logs：日志存放目录。 ● temp：临时工作目录。
是否导入	如果您在其他云服务上存储了表格数据，可启用此功能，现支持将存储在对象存储服务(OBS)、数据湖探索(DLI)或MapReduce服务(MRS)的数据导入。

参数名称	说明
数据源（“OBS”）	● “文件路径”：单击输入框右侧按钮，可打开当前帐号下的所有OBS桶，请选择需要导入的数据文件所在目录。 ● “导入是否包含表头”：开启表示导入文件包含表头，此时会将导入文件的第一行作为列名，否则会添加默认列名，自动填写在Schema信息中。 OBS的详细功能说明，请参见《**OBS用户指南**》。
数据源（“DWS”）	● “集群名称”：系统自动将当前账号下的DWS集群展现在列表中，您可以在下拉框中选择您所需的DWS集群。 ● “数据库名称”：根据选择的DWS集群，填写数据所在的数据库名称。 ● “表名称”：根据选择的数据库，填写数据所在的表。 ● “用户名”：输入DWS集群管理员用户的用户名。 ● “密码”：输入DWS集群管理员用户的密码。 DWS的详细功能说明，请参见《**DWS用户指南**》。 **说明** 从DWS导入数据，需要借助DLI的功能，如果用户没有访问DLI服务的权限，需根据页面提示创建DLI的委托。
数据源（“DLI”）	● “队列名称”：系统自动将当前帐号下的DLI队列展现在列表中，您可以在下拉框中选择您所需的队列。 ● “数据库名称”：根据选择的队列展现所有的数据库，请在下拉框中选择您所需的数据库。 ● “表名称”：根据选择的数据库展现此数据库中的所有表。请在下拉框中选择您所需的表。 DLI的详细功能说明，请参见《**DLI用户指南**》。
数据源（“MRS”）	● “集群名称”：系统自动将当前帐号下的MRS集群展现在此列表中，但是流式集群不支持导入操作。请在下拉框中选择您所需的集群。 ● “文件路径”：根据选择的集群，输入对应的文件路径，此文件路径为HDFS路径。 ● “导入是否包含表头”：开启表示导入时将表头同时导入。 MRS的详细功能说明，请参见《**MRS用户指南**》。
Schema信息	表格的列名和对应类型，需要跟导入数据的列数保持一致。请根据您导入的数据输入“列名”，同时选择此列的“类型”。其中支持的类型见**表4-6**。 单击“添加Schema信息”，即可增加一行列。创建数据集时必须指定schema，且一旦创建不支持修改。 从OBS数据源导入数据，会自动获取文件路径下csv文件的schema，如果多个csv文件的schema不一致会报错。

表 4-6 Schema 数据类型说明

类型	描述	存储空间	范围
String	字符串	-	-
Short	有符号整数	2字节	-32768-32767
Int	有符号整数	4字节	-2147483648～2147483647
Long	有符号整数	8字节	-9223372036854775808～9223372036854775807
Double	双精度浮点型	8字节	-
Float	单精度浮点型	4字节	-
Byte	有符号整数	1字节	-128-127
Date	日期类型，描述了特定的年月日，格式：yyyy-MM-dd，例如2014-05-29	-	-
Timestamp	时间戳，表示日期和时间。格式：yyyy-MM-dd HH:mm:ss	-	-
Boolean	布尔类型	1字节	TRUE/FALSE

视频

图 4-9 视频类型的参数

表 4-7 数据集的详细参数

参数名称	说明
数据集输入位置	选择数据集输入位置的OBS路径。
数据集输出位置	选择数据集输出位置的OBS路径。 **说明** “数据集输出位置”不能与“数据集输入位置”为同一路径，且不能是“数据集输入位置”的子目录。“数据集输出位置”最好选择一个空目录。
添加标签集	• **设置标签名称**：在标签名称文本框中，输入标签名称。标签名称只能是中文、字母、数字、下划线或中划线组成的合法字符串。长度为1～32字符。 • **添加标签**：单击“添加标签”可增加多个标签。 • **设置标签颜色**：在每个标签右侧的标签颜色区域下，可在色板中选择颜色，或者直接输入十六进制颜色码进行设置。

其他（自由格式）

图 4-10 自由格式类型数据集的参数

表 4-8 数据集的详细参数

参数名称	说明
数据集输入位置	选择数据集输入位置的OBS路径。
数据集输出位置	选择数据集输出位置的OBS路径。 **说明** “数据集输出位置”不能与“数据集输入位置”为同一路径，且不能是“数据集输入位置”的子目录。“数据集输出位置”最好选择一个空目录。

4.3 标注数据

4.3.1 图像分类

由于模型训练过程需要大量有标签的图片数据，因此在模型训练之前需对没有标签的图片添加标签。您可以通过手工标注或智能一键标注的方式添加标签，快速完成对图片的标注操作，也可以对已标注图片修改或删除标签进行重新标注。

针对图像分类场景，开始标注前，您需要了解：

- 图片标注支持多标签，即一张图片可添加多个标签。
- 标签名是由中文、大小写字母、数字、中划线或下划线组成，且不超过32位的字符串。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据集”，进入“数据集”管理页面。
2. 在数据集列表中，基于“标注类型”选择需要进行标注的数据集，单击数据集名称进入数据集概览页。

   此操作默认进入数据集当前版本的概览页，如果需要对其他版本进行数据标注，请先在“版本管理”操作中，将需要进行数据标注的版本设置为“当前版本。”详细操作指导请参见管理数据集版本。
3. 在数据集概览页中，单击右上角“开始标注”，进入数据集详情页。数据集详情页默认展示此数据集下全部数据。

同步数据源

ModelArts会自动从数据集输入位置同步数据至数据集详情页，包含数据及标注信息。

- 对于图像分类数据集，同步数据源操作会以同级目录下的同名“txt”文件作为对应图像的标签。
- 对于物体检测、图像分割数据集，则以同级目录下的同名“xml”文件作为对应图像的标签。

为了快速获取OBS桶中最新数据，可在数据集详情页的“全部”或“未标注”页签中，单击“同步数据源”，快速将通过OBS上传的数据添加到数据集中。

筛选数据

在数据集概览页中，单击页面右上角的“开始标注”，进入数据集的详情页面，默认展示数据集中全部数据。在“全部”、“未标注”或“已标注”页签下，您可以在筛选条件区域，添加筛选条件，快速过滤出您想要查看的数据。

支持的筛选条件如下所示，您可以设置一个或多个选项进行筛选。

- 难例集：难例或非难例。
- 标签：您可以选择全部标签，或者基于您指定的标签，选中其中一个或多个。
- 样本创建时间：1个月内、1天内或自定义，如果选择自定义，可以在时间框中指定明确时间范围。

- 文件名或目录：根据文件名称或者文件存储目录筛选。
- 标注人：选择执行标注操作的帐号名称。
- 样本属性：表示自动分组生成的属性。只有启用了自动分组任务后才可使用此筛选条件。
- 数据属性：暂不支持。

图 4-11 筛选条件

标注图片（手工标注）

数据集详情页中，展示了此数据集中“全部”、“未标注”和“已标注”的图片，默认显示“全部”的图片列表。单击图片，即可进行图片的预览，对于已标注图片，预览页面下方会显示该图片的标签信息。

1. 在“未标注”页签，勾选需进行标注的图片。
   - 手工点选：在图片列表中，单击勾选图片左上角的选择框，进入选择模式，表示图片已勾选。可勾选同类别的多个图片，一起添加标签。
   - 批量选中：如果图片列表的当前页，所有图片属于一种类型，可以在图片列表的右上角单击“选择当前页”，则当前页面所有的图片将选中。
2. 为选中图片添加标签。
   a. 在右侧的“添加标签”区域中，单击“标签名”右侧的文本框中设置标签。

      单击“标签名”右侧的文本框，然后从下拉列表中选择已有的标签。如果已有标签无法满足要求时，可进入修改数据集页面，添加标签。

   b. 查看“选中文件标签”的信息，确认无误后，单击“确认”。此时，选中的图片将被自动移动至“已标注”页签，且在“未标注”和“全部”页签中，标签的信息也将随着标注步骤进行更新，如增加的标签名称、各标签对应的图片数量。

图 4-12 添加标签

查看已标注图片

在数据集详情页，单击“已标注”页签，您可以查看已完成标注的图片列表。图片缩略图下方默认呈现其对应的标签，您也可以勾选图片，在右侧的“选中文件标签”中了解当前图片的标签信息。

修改标注

当数据完成标注后，您还可以进入已标注页签，对已标注的数据进行修改。

- **基于图片修改**

  在数据集详情页面，单击“已标注”页签，然后在图片列表中选中待修改的图片（选择一个或多个）。在右侧标签信息区域中对图片信息进行修改。

  修改标签：在“选中文件标签”区域中，单击操作列的编辑图标，然后在文本框中输入正确的标签名，然后单击确定图标完成修改。

  删除标签：在“选中文件标签”区域中，单击操作列的删除图标删除该标签。此操作仅删除选中图片中的标签。

图 4-13 编辑标签

- **基于标签修改**

  在数据集详情页面，单击“已标注”页签，在图片列表右侧，显示全部标签的信息。

  – 修改标签：单击操作列的编辑按钮，然后在弹出的对话框中输入修改后的标签名，然后单击“确定”完成修改。修改后，之前添加了此标签的图片，都将被标注为新的标签名称。
  – 删除标签：单击操作列的删除按钮，在弹出的对话框中，选择“仅删除标签”、“删除标签及仅包含此标签的图片（不删除源文件）”或“删除标签及仅包含此标签的图片（同时删除源文件）”，然后单击“确定”。

图 4-14 全部标签的信息

添加图片

除了数据集输入位置自动同步的数据外，您还可以在ModelArts界面中，直接添加图片，用于数据标注。

1. 在数据集详情页面，单击“全部”或“未标注”页签，然后单击左上角“添加图片”。
2. 在弹出的“添加图片”对话框中，单击“添加图片”。

   选择本地环境中需要上传的图片，可以一次性选择多张图片。支持JPG、JPEG、PNG、BMP四种格式图片，单张图片大小不能超过5MB，单次上传的图片总大小不能超过8MB。

   图片选择完成后，“添加图片”对话框将显示上传图片的缩略图以及图片大小。

图 4-15 添加图片

3. 在添加图片对话框中，单击“确定”，完成添加图片的操作。

   您添加的图片将自动呈现在“未标注”的图片列表中。且图片将自动存储至此“数据集输入位置”对应的OBS目录中。

删除图片

通过数据删除操作，可将需要丢弃的图片数据快速删除。

在“全部”、“未标注”或“已标注”页面中，依次选中需要删除的图片，或者选择“选择当前页”选中该页面所有图片，然后单击左上角“删除图片”。在弹出的对话框中，根据实际情况选择是否勾选“同时删除源文件”，确认信息无误后，单击“确定”完成图片删除操作。

其中，被选中的图片，其左上角将显示为勾选状态。如果当前页面无选中图片时，“删除图片”按钮为灰色，无法执行删除操作。

说明

如果勾选了“同时删除源文件”，删除图片操作将删除对应OBS目录下存储的图片，此操作可能会影响已使用此源文件的其他数据集或数据集版本，有可能导致展示异常或训练/推理异常。删除后，数据将无法恢复，请谨慎操作。

4.3.2 物体检测

由于模型训练过程需要大量有标签的图片数据，因此在模型训练之前需对没有标签的图片添加标签。您可以通过手工标注或智能一键标注的方式添加标签，快速完成对图片的标注操作，也可以对已标注图片修改或删除标签进行重新标注。

针对物体检测场景，开始标注前，您需要了解：

- 图片中所有目标物体都要标注。
- 目标物体清晰无遮挡的，必须画框。
- 画框仅包含整个物体。框内包含整个物体的全部，画框边缘不可与待标注的物体的边缘轮廓相交，在此基础之上确保边缘和待标注物体间不要留着空隙，避免背景对模型训练造成干扰。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据集”，进入“数据集”管理页面。
2. 在数据集列表中，基于“标注类型”选择需要进行标注的数据集，单击数据集名称进入数据集概览页。

   此操作默认进入数据集当前版本的概览页，如果需要对其他版本进行数据标注，请先在“版本管理”操作中，将需要进行数据标注的版本设置为“当前版本。”详细操作指导请参见**管理数据集版本**。
3. 在数据集概览页中，单击右上角“开始标注”，进入数据集详情页。数据集详情页默认展示此数据集下全部数据。

同步数据源

ModelArts会自动从数据集输入位置同步数据至数据集详情页，包含数据及标注信息。

- 对于图像分类数据集，同步数据源操作会以同级目录下的同名“txt”文件作为对应图像的标签。
- 对于物体检测、图像分割数据集，则以同级目录下的同名“xml”文件作为对应图像的标签。

为了快速获取OBS桶中最新数据，可在数据集详情页的“全部”或“未标注”页签中，单击“同步数据源”，快速将通过OBS上传的数据添加到数据集中。

筛选数据

在数据概览页中，默认展示数据集的概览情况。在界面左上方，单击“开始标注”，进入数据集的详细数据页面，默认展示数据集中全部数据。在“全部”、“未标注”或“已标注”页签下，您可以在筛选条件区域，添加筛选条件，快速过滤出您想要查看的数据。

支持的筛选条件如下所示，您可以设置一个或多个选项进行筛选。

- 难例集：难例或非难例。
- 标签：您可以选择全部标签，或者基于您指定的标签，选中其中一个或多个。
- 样本创建时间：1个月内、1天内或自定义，如果选择自定义，可以在时间框中指定明确时间范围。

- 文件名或目录：根据文件名称或者文件存储目录筛选。
- 标注人：选择执行标注操作的帐号名称。
- 样本属性：表示自动分组生成的属性。只有启用了**自动分组**任务后才可使用此筛选条件。
- 数据属性：暂不支持。

图 4-16 筛选条件

标注图片（手工标注）

数据集详情页中，展示了此数据集中“未标注”和“已标注”的图片，默认显示“全部”的图片列表。

1. 在“未标注”页签图片列表中，单击图片，自动跳转到标注页面。在标注页面，常用按钮的使用可参见**表4-10**。
2. 在页面左侧工具栏选择合适的标注图形，系统默认的标注图形为矩形。本示例使用矩形工具进行标注。

说明

页面左侧可以选择多种形状对图片进行标注。标注第一张图片时，一旦选择其中一种，其他所有图片都需要使用此形状进行标注。

表 4-9 支持的标注框

图标	使用说明
□	矩形。鼠标单击标注对象左上角边缘位置，界面将出现矩形框，移动鼠标使得矩形框覆盖标注对象，然后单击完成标注。
⬠	多边形。在标注对象所在范围内，鼠标左键单击完成一个点的标注，沿着物体的形状边缘，通过鼠标指定多个点，最终单击到第一个点的位置，由所有的点组成一个多边形形状。使得需标注的对象在此标注框内。
○	圆形。在标注对象中，选择物体的中心点位置，单击鼠标确定圆心，然后移动鼠标，使得圆形框覆盖标注对象，然后再单击鼠标完成标注。

图标	使用说明
	直线。在标注对象中，选择物体的起始点，单击鼠标确定直线的起始点，然后使得直线覆盖标注对象，然后再单击鼠标完成标注。
	虚线。在标注对象中，选择物体的起始点，单击鼠标确定虚线的起始点，然后使得虚线覆盖标注对象，然后再单击鼠标完成标注。
	点。单击图片中的物体所在位置，即可完成点的标注。

3. 在弹出的添加标签文本框中，直接输入新的标签名，在文本框前面选中标签颜色，然后单击“添加”。如果已存在标签，从下拉列表中选择已有的标签，单击“添加”。

   逐步标注图片中所有物体所在位置，一张图片可添加多个标签。完成一张图片标注后，可单击图片下方图片列表，快速选中其他未标注的图片，然后在标注页面中执行标注操作。

图 4-17 添加物体检测标签

4. 单击页面上方“返回数据标注预览”查看标注信息，在弹框中单击“确定”保存当前标注并离开标注页面。

   选中的图片被自动移动至“已标注”页签，且在“未标注”和“全部”页签中，标签的信息也将随着标注步骤进行更新，如增加的标签名称、标签对应的图片数量。

表 4-10 标注界面的常用按钮

按钮图标	功能说明
	撤销上一个操作。
	重做上一个操作。

按钮图标	功能说明
	放大图片。
	缩小图片。
	删除当前图片中的所有标注框。
	显示或隐藏标注框。只有在已标注图片中可使用此操作。
	拖动，可将标注好的框拖动至其他位置，也可以选择框的边缘，更改框的大小。
	复位，与上方拖动为同组操作，当执行了拖动后，可以单击复位按钮快速将标注框恢复为拖动前的形状和位置。
	全屏显示标注的图片。

查看已标注图片

在数据集详情页，单击“已标注”页签，您可以查看已完成标注的图片列表。单击图片，可在右侧的“当前文件标签”中了解当前图片的标签信息。

修改标注

当数据完成标注后，您还可以进入已标注页签，对已标注的数据进行修改。

- **基于图片修改**

  在数据集详情页面，单击“已标注”页签，然后在图片列表中选中待修改的图片，单击图片跳转到标注页面，在右侧标签信息区域中对图片信息进行修改。

  - 修改标签：“标注”区域中，单击编辑图标，在文本框中输入正确的标签名，然后单击确定图标完成修改。也可以单击标签，在图片标注区域，调整标注框的位置和大小，完成调整后，单击其他标签即可保存修改。
  - 删除标签：在“标注”区域中，单击删除图标即可删除此图片中的标签。

    标签删除后，单击页面左上角的“返回数据标注预览”离开标注页面，在弹出对话框中保存标注。图标的标签全部删除后，该图片会重新回到“未标注”页签。

    图 4-18 编辑物体检测标签

- **基于标签修改**

  在数据集详情页面，单击“已标注”页签，在图片列表右侧，显示全部标签的信息。

- 修改标签：单击操作列的编辑图标，然后在弹出的对话框中输入修改后的标签和颜色，然后单击“确定”完成修改。修改后，之前添加了此标签的图片，都将被标注为新的标签名称。
- 删除标签：单击操作列的删除图标可删除此标签。

图 4-19 物体检测的全部标签

添加图片

除了数据集输入位置自动同步的数据外，您还可以在ModelArts界面中，直接添加图片，用于数据标注。

1. 在数据集详情页面，单击“全部”或“未标注”页签，然后单击左上角“添加图片”。
2. 在弹出的“添加图片”对话框中，单击“添加图片”。

   选择本地环境中需要上传的图片，可以一次性选择多张图片。支持JPG、JPEG、PNG、BMP四种格式图片，单张图片大小不能超过5MB，单次上传的图片总大小不能超过8MB。

   图片选择完成后，“添加图片”对话框将显示上传图片的缩略图以及图片大小。

图 4-20 添加图片

3. 在添加图片对话框中，单击“确定”，完成添加图片的操作。

   您添加的图片将自动呈现在“未标注”的图片列表中。且图片将自动存储至此“数据集输入位置”对应的OBS目录中。

删除图片

通过数据删除操作，可将需要丢弃的图片数据快速删除。

在“全部”、“未标注”或“已标注”页面中，依次选中需要删除的图片，或者选择“选择当前页”选中该页面所有图片，然后单击左上角“删除图片”。在弹出的对话框中，根据实际情况选择是否勾选“同时删除源文件”，确认信息无误后，单击“确定”完成图片删除操作。

其中，被选中的图片，其左上角将显示为勾选状态。如果当前页面无选中图片时，“删除图片”按钮为灰色，无法执行删除操作。

说明

如果勾选了“同时删除源文件”，删除图片操作将删除对应OBS目录下存储的图片，此操作可能会影响已使用此源文件的其他数据集或数据集版本，有可能导致展示异常或训练/推理异常。删除后，数据将无法恢复，请谨慎操作。

4.3.3 图像分割

由于模型训练过程需要大量有标签的图片数据，因此在模型训练之前需对没有标签的图片添加标签。您可以通过在ModelArts控制台进行标注，也可以对已标注图片修改或删除标签进行重新标注。

针对图像分割场景，开始标注前，您需要了解：

- 图片中需要提取轮廓的物体都要标注。
- 支持使用多边形标注和极点标注。
  - 多边形标注，根据目标物体的轮廓绘制多边形。
  - 极点标注，在目标物体轮廓的最上、最左、最下、最右的位置分别标注四个极点，极点要在物体的轮廓上。系统将根据标注的极点推理出物体的轮廓。对于背景比较复杂的图片，极点标注效果不佳，推荐使用多边形标注。
- 多边形标注时，标注框或极点，必须在图片范围内，超出图片将导致后续作业异常。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据集”，进入“数据集”管理页面。
2. 在数据集列表中，基于“标注类型”选择需要进行标注的数据集，单击数据集名称进入数据集概览页。

   此操作默认进入数据集当前版本的概览页，如果需要对其他版本进行数据标注，请先在“版本管理”操作中，将需要进行数据标注的版本设置为“当前版本。”详细操作指导请参见管理数据集版本。
3. 在数据集概览页中，单击右上角“开始标注”，进入数据集详情页。数据集详情页默认展示此数据集下全部数据。

同步数据源

ModelArts会自动从数据集输入位置同步数据至数据集详情页，包含数据及标注信息。

- 对于图像分类数据集，同步数据源操作会以同级目录下的同名“txt”文件作为对应图像的标签。
- 对于物体检测、图像分割数据集，则以同级目录下的同名“xml”文件作为对应图像的标签。

为了快速获取OBS桶中最新数据，可在数据集详情页的“全部”或“未标注”页签中，单击“同步数据源”，快速将通过OBS上传的数据添加到数据集中。

筛选数据

在数据概览页中，默认展示数据集的概览情况。在界面右上方，单击“开始标注”，进入数据集的详细数据页面，默认展示数据集中全部数据。在“全部”、“未标注”或“已标注”页签下，您可以在筛选条件区域，添加筛选条件，快速过滤出您想要查看的数据。

支持的筛选条件如下所示，您可以设置一个或多个选项进行筛选。

- 难例集：难例或非难例。
- 标签：您可以选择全部标签，或者基于您指定的标签，选中其中一个或多个。

- 样本创建时间：1个月内、1天内或自定义，如果选择自定义，可以在时间框中指定明确时间范围。
- 文件名或目录：根据文件名称或者文件存储目录筛选。
- 标注人：选择执行标注操作的帐号名称。
- 样本属性：表示自动分组生成的属性。只有启用了自动分组任务后才可使用此筛选条件。
- 数据属性：暂不支持

图 4-21 筛选条件

标注图片（手工标注）

数据集详情页中，展示了此数据集中“未标注”和“已标注”的图片，默认显示“全部”的图片列表。

1. 在“未标注”页签图片列表中，单击图片，自动跳转到标注页面。在标注页面，常用按钮的使用可参见表4-12。
2. 选择标注方式。

   在标注页面，上方工具栏提供了常用的标注方式及常用按钮，系统默认的标注方式为多边形标注。选择多边形标注或极点标注。

   说明

   标注第一张图片时，一旦选择其中一种，其他所有图片都需要使用此方式进行标注。

图 4-22 工具栏

表 4-11 标注方式

图标	使用说明
	多边形。在标注对象所在范围内，鼠标左键单击完成一个点的标注，沿着物体的形状边缘，通过鼠标指定多个点，最终单击到第一个点的位置，由所有的点组成一个多边形形状。使得需标注的对象在此标注框内。

图标	使用说明
	极点标注。在目标物体轮廓的最上、最左、最下、最右的位置分别标注四个极点，极点要在物体的轮廓上。系统将根据标注的极点推理出物体的轮廓。

表 4-12 工具栏常用按钮

按钮图标	功能说明
	撤销上一个操作。
	重做上一个操作。
	放大图片。
	缩小图片。
	删除当前图片中的所有标注框。
	显示或隐藏标注框。只有在已标注图片中可使用此操作。
	拖动，可将标注好的框拖动至其他位置，也可以选择框的边缘，更改框的大小。
	复位，与上方拖动为同组操作，当执行了拖动后，可以单击复位按钮快速将标注框恢复为拖动前的形状和位置。
	全屏显示标注的图片。

3. 标注物体。

   以极点标注为例。识别图片中的物体，单击左键分别定位物体的最上、最左、最下、最右的位置点。确定位置后，将弹出对话框，填入标签名称，单击确定添加物体的标签。确定后系统将自动推理出物体的轮廓。

   完成一张图片标注后，可单击图片下方图片列表，快速选中其他未标注的图片，然后在标注页面中执行标注操作。

图 4-23 标注物体轮廓

4. 单击页面上方“返回数据标注预览”查看标注信息，在弹框中单击“确定”保存当前标注并离开标注页面。

   选中的图片被自动移动至“已标注”页签，且在“未标注”和“全部”页签中，标签的信息也将随着标注步骤进行更新，如增加的标签名称、标签对应的图片数量。

查看已标注图片

在数据集详情页，单击“已标注”页签，您可以查看已完成标注的图片列表。单击图片，可在右侧的“当前文件标签”中了解当前图片的标签信息。

修改标注信息

当数据完成标注后，您还可以进入已标注页签，对已标注的数据进行修改。

在数据集详情页面，单击“已标注”页签，然后在图片列表中选中待修改的图片，单击图片跳转到标注页面，在右侧标签信息区域中单击此图片已添加的标注信息。

- 修改标签：“标注”区域中，单击编辑图标，在弹出框中输入正确的标签名或标签颜色，然后单击“确定”完成修改。也可以单击标签，在图片标注区域，调整标注框的位置和大小，完成调整后，单击其他标签即可保存修改。
- 修改图片标注信息：在图片展示区，显示物体边缘，可单击蓝色圆点，将标注框调整至物体边缘。
- 删除标签：在“标注”区域中，单击删除图标即可删除此图片中的标签。图片的标签全部删除后，该图片会重新回到“未标注”页签。

标注标注信息修改后，单击页面左上角的“返回数据标注预览”离开标注页面，在弹出对话框中单击“确定”保存修改。

图 4-24 编辑标注信息

添加图片

除了数据集输入位置自动同步的数据外，您还可以在ModelArts界面中，直接添加图片，用于数据标注。

1. 在数据集详情页面，单击“全部”或“未标注”页签，然后单击左上角“添加图片”。
2. 在弹出的“添加图片”对话框中，单击“添加图片”。

   选择本地环境中需要上传的图片，可以一次性选择多张图片。支持JPG、JPEG、PNG、BMP四种格式图片，单张图片大小不能超过5MB，单次上传的图片总大小不能超过8MB。

   图片选择完成后，“添加图片”对话框将显示上传图片的缩略图以及图片大小。

图 4-25 添加图片

3. 在添加图片对话框中，单击“确定”，完成添加图片的操作。

   您添加的图片将自动呈现在“未标注”的图片列表中。且图片将自动存储至此“数据集输入位置”对应的OBS目录中。

删除图片

通过数据删除操作，可将需要丢弃的图片数据快速删除。

在“全部”、“未标注”或“已标注”页面中，依次选中需要删除的图片，或者选择“选择当前页”选中该页面所有图片，然后单击左上角“删除图片”。在弹出的对话框中，根据实际情况选择是否勾选“同时删除源文件”，确认信息无误后，单击“确定”完成图片删除操作。

其中，被选中的图片，其左上角将显示为勾选状态。如果当前页面无选中图片时，“删除图片”按钮为灰色，无法执行删除操作。

说明

如果勾选了“同时删除源文件”，删除图片操作将删除对应OBS目录下存储的图片，此操作可能会影响已使用此源文件的其他数据集或数据集版本，有可能导致展示异常或训练/推理异常。删除后，数据将无法恢复，请谨慎操作。

4.3.4 文本分类

由于模型训练过程需要大量有标签的数据，因此在模型训练之前需对没有标签的文本添加标签。您也可以对已标注文本进行修改、删除和重新标注。

针对文本分类场景，是对文本的内容按照标签进行分类处理，开始标注前，您需要了解：

- 文本标注支持多标签，即一个标注对象可添加多个标签。
- 标签名是由中文、大小写字母、数字、中划线或下划线组成，且不超过32位的字符串。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据集”，进入“数据集”管理页面。
2. 在数据集列表中，基于“标注类型”选择需要进行标注的数据集，单击数据集名称进入数据集概览页。

   此操作默认进入数据集当前版本的概览页，如果需要对其他版本进行数据标注，请先在“版本管理”操作中，将需要进行数据标注的版本设置为“当前版本。”详细操作指导请参见**管理数据集版本**。
3. 在数据集概览页中，单击右上角“开始标注”，进入数据集详情页。数据集详情页默认展示此数据集下全部数据。

标注文本

数据集详情页中，展示了此数据集中“未标注”和“已标注”的文本，默认显示“未标注”的文本列表。

1. 在“未标注”页签文本列表中，页面左侧罗列“标注对象列表”。在列表中单击需标注的文本对象，选择右侧“标签集”中的标签进行标注。一个标注对象可添加多个标签。

   以此类推，不断选中标注对象，并为其添加标签。

   图 4-26 文本分类标注

2. 当所有的标注对象都已完成标注，单击页面下方“保存当前页”完成“未标注”列表的文本标注。

添加标签

- 在“未标注”页签添加：单击页面中标签集右侧的加号，然后在弹出的“新增标签”页中，添加标签名称，选择标签颜色，单击“确定”完成标签的新增。

图 4-27 添加标签（1）

- 在“已标注”页签添加：在右侧单击页面中全部标签加号，然后在弹出的“新增标签”页中，添加标签名称，选择标签颜色，单击“确定”完成标签的新增。

图 4-28 添加标签（2）

图 4-29 新增标签

查看已标注文本

在数据集详情页，单击“已标注”页签，您可以查看已完成标注的文本列表。您也可以在右侧的“全部标签”中了解当前数据集支持的所有标签信息。

修改标注

当数据完成标注后，您还可以进入已标注页签，对已标注的数据进行修改。

- **基于文本修改**

  在数据集详情页，单击“已标注”页签，然后在文本列表中选中待修改的文本。

  在文本列表中，单击文本，当文本背景变为蓝色时，表示已选择。当文本有多个标签时，可以单击文本标签上方的×删除单个标签。

- **基于标签修改**

  在数据集详情页，单击“已标注”页签，在图片列表右侧，显示全部标签的信息。

  – 批量修改：在“全部标签”区域中，单击操作列的编辑图标，然后在文本框中修改标签名称，选择标签颜色，单击“确定”完成修改。
  – 批量删除：在“全部标签”区域中，单击操作列的删除图标，在弹出对话框中，可选择“仅删除标签”或“删除标签及仅包含此标签的标注对象”，然后单击“确定”。

添加文件

除了数据集输入位置自动同步的数据外，您还可以在ModelArts界面中，直接添加文件，用于数据标注。

1. 在数据集详情页面，单击“未标注”页签，然后单击左上角“添加文件”。
2. 在弹出的“添加文件”对话框中，根据需上传文件的基本情况，完成设置后选择上传文件。

   选择本地环境中需要上传的文件，可以一次性选择多个文件。文件格式只支持“txt”或“csv”，且一次上传文件的总大小不能超过8MB。“文本与标签分割符”与“多标签分割符”不能选同一个。

   – “模式”：选择“文本和标注合并”或“文本和标注分离”模式。界面中已给出示例，请参考示例判断需添加的文件属于哪一种模式。
   – “文本与标签分隔符”：可设置为“Tab键”、“空格”、“分号”、“逗号”或“其他”。选择“其他”时，可以在右侧文本框中输入对应的分隔符。
   – “多标签分隔符”：可设置为“Tab键”、“空格”、“分号”、“逗号”或“其他”。选择“其他”时，可以在右侧文本框中输入对应的分隔符。

图 4-30 添加文件

3. 在添加文件对话框中，单击“上传文件”，完成添加文件的操作。您添加的文件内容将自动呈现在“未标注”或“已标注”的文本列表中。

删除文件

通过数据删除操作，可将需要丢弃的文件数据快速删除。

- 在“未标注”页面中，单击选中需要删除的文本对象，然后单击左上角“删除”，即可完成文本的删除操作。
- 在“已标注”页面中，选中待删除的文本对象，然后单击“删除”，删除单个文本。或者选择“选择当前页”选中该页面所有文本，然后单击左上角“删除”，即可完成当前页所有文本的删除操作。

其中，被选中的文本，其背景将显示为蓝色。

4.3.5 命名实体

命名实体场景，是针对文本中的实体片段进行标注，如“时间”、“地点”等。开始标注前，您需要了解：

- 实体命名标签名是由中文、大小写字母、数字、中划线或下划线组成，且不超过32位的字符串。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据集”，进入“数据集”管理页面。

2. 在数据集列表中，基于“标注类型”选择需要进行标注的数据集，单击数据集名称进入数据集概览页。

   此操作默认进入数据集当前版本的概览页，如果需要对其他版本进行数据标注，请先在“版本管理”操作中，将需要进行数据标注的版本设置为“当前版本。”详细操作指导请参见管理数据集版本。

3. 在数据集概览页中，单击右上角“开始标注”，进入数据集详情页。数据集详情页默认展示此数据集下全部数据。

标注文本

数据集详情页中，展示了此数据集中“未标注”和“已标注”的文本，默认显示“未标注”的文本列表。

1. 在“未标注”页签文本列表中，页面左侧罗列“标注对象列表”。在列表中单击需标注的文本对象，在右侧标签集下显示的文本内容中选中需要标注的部分，然后选择右侧“标签集”中的标签进行标注。一个标注对象可添加多个标签。

   以此类推，不断选中标注对象，并为其添加标签。

   图 4-31 命名实体标注

2. 单击页面下方“保存当前页”完成文本标注。

添加标签

- 在“未标注”页签添加：单击页面中标签集右侧的加号，然后在弹出的“新增标签”页中，添加标签名称，选择标签颜色，单击“确定”完成标签的新增。

  图 4-32 添加命名实体标签（1）

- 在“已标注”页签添加：在右侧单击页面中全部标签加号，然后在弹出的“新增标签”页中，添加标签名称，选择标签颜色，单击“确定”完成标签的新增。

图 4-33 添加命名实体标签（2）

图 4-34 新增命名实体标签

查看已标注文本

在数据集详情页，单击“已标注”页签，您可以查看已完成标注的文本列表。您也可以在右侧的“全部标签”中了解当前数据集支持的所有标签信息。

修改标注

当数据完成标注后，您还可以进入“已标注”页签，对已标注的数据进行修改。

在数据集详情页，单击“已标注”页签，在右侧标签信息区域中对文本信息进行修改。

- **基于文本修改**

  在数据集详情页，单击“已标注”页签，然后在文本列表中选中待修改的文本。

  手工点选删除：在文本列表中，单击文本，当文本背景变为蓝色时，表示已选择。在页面右侧，单击文本标签上方的×删除单个标签。

- **基于标签修改**

  在数据集详情页，单击“已标注”页签，在图片列表右侧，显示全部标签的信息。

  – 批量修改：在“全部标签”区域中，单击操作列的编辑按钮，然后在文本框中添加标签名称，选择标签颜色，单击“确定”完成修改。

– 批量删除：在“全部标签”区域中，单击操作列的删除按钮，在弹出对话框中，可选择“仅删除标签”或“删除标签及仅包含此标签的标注对象”，然后单击“确定”。

添加文件

除了数据集输入位置自动同步的数据外，您还可以在ModelArts界面中，直接添加文件，用于数据标注。

1. 在数据集详情页面，单击“未标注”页签，然后单击左上角“添加文件”。
2. 在弹出的“添加文件”对话框中，根据需上传文件的基本情况，选择上传文件。

   选择本地环境中需要上传的文件，可以一次性选择多个文件。文件格式只支持“txt”或“csv”，且一次上传文件的总大小不能超过8MB。

   图 4-35 添加文件

3. 在添加文件对话框中，单击“上传文件”，完成添加文件的操作。您添加的文件内容将自动呈现在“未标注”的文本列表中。

删除文件

通过数据删除操作，可将需要丢弃的文件数据快速删除。

- 在“未标注”页面中，单击选中需要删除的文本对象，然后单击左上角“删除”，即可完成文本的删除操作。
- 在“已标注”页面中，选中待删除的文本对象，然后单击“删除”，删除单个文本。或者选择“选择当前页”选中该页面所有文本，然后单击左上角“删除”，即可完成当前页所有文本的删除操作。

其中，被选中的文本，其背景将显示为蓝色。

4.3.6 文本三元组

三元组标注适用于标注出语句当中形如（主语/Subject，谓词/Predicate，宾语/Object）结构化知识的场景，标注时不但可以标注出语句当中的实体，还可以标注出实体之间的关系，其在依存句法分析、信息抽取等自然语言处理任务中经常用到。

文本三元组类型的数据标注，需要关注两种标签，“实体标签”和“关系标签”。“关系标签”需设置对应的“起始实体”和“终止实体”。

- 支持设置多个“实体标签”和“关系标签”。一个文本数据中，也可以标注多个“实体标签”和“关系标签”

- 创建数据集时定义的“实体标签”，不支持删除。

注意事项

在开始标注之前，需确保数据集对应的“实体标签”和“关系标签”已定义好。“关系标签”需设置对应的“起始实体”和“终止实体”。“关系标签”只能添加至其设置好的“起始实体”和“终止实体”之间。

例如，如图4-36所示，当两个文本都被标注为“地点”，那么针对这两个实体，无法添加本示例中的任意一个关系标签。当无法添加某个关系标签时，界面将显示一个红色的叉号，如图4-37所示。

图 4-36 实体标签和关系标签的示例

图 4-37 无法添加关系标签

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据集”，进入“数据集”管理页面。
2. 在数据集列表中，基于“标注类型”选择需要进行标注的数据集，单击数据集名称进入数据集概览页。

   此操作默认进入数据集当前版本的概览页，如果需要对其他版本进行数据标注，请先在“版本管理”操作中，将需要进行数据标注的版本设置为“当前版本。”详细操作指导请参见管理数据集版本。
3. 在数据集概览页中，单击右上角“开始标注”，进入数据集详情页。数据集详情页默认展示此数据集下全部数据。

标注文本

数据集详情页中，展示了此数据集中“未标注”和“已标注”的文本，默认显示“未标注”的文本列表。

1. 在“未标注”页签文本列表中，页面左侧罗列“标注对象列表”。在列表中单击需标注的文本对象，选中相应文本内容，在页面呈现的实体类型列表中选择实体名称，完成实体标注。

   图 4-38 实体标注

2. 在完成多个实体标注后，鼠标左键依次单击起始实体和终止实体，在呈现的关系类型列表中选择一个对应的关系类型，完成关系标注。

   图 4-39 关系标注

3. 当所有的标注对象都已完成标注，单击页面下方“保存当前页”完成“未标注”列表的文本标注。

说明

“文本三元组”类型的数据集，不支持在标注页面修改标签，需要进入“修改数据集”页面，修改“实体标签”和“关系标签”。

修改标注

当数据完成标注后，您还可以进入已标注页签，对已标注的数据进行修改。

在数据集详情页，单击“已标注”页签，在左侧文本列表中选中一行文本，右侧区域显示具体的标注信息。将鼠标移动至对应的实体标签或关系类型，单击鼠标右键，可删除此标注。单击鼠标左键，依次单击连接起始实体和终止实体，可增加关系类型，增加关系标注。

图 4-40 在文本中修改标签

您也可以在单击页面下方的“删除当前项标签”按钮，删除选中文本对象中的所有标签。

图 4-41 删除当前项标签

添加文件

除了数据集输入位置自动同步的数据外，您还可以在ModelArts界面中，直接添加文件，用于数据标注。

1. 在数据集详情页面，单击“未标注”页签，然后单击左上角“添加文件”。
2. 在弹出的“添加文件”对话框中，选择上传文件。

   选择本地环境中需要上传的文件，可以一次性选择多个文件。文件格式只支持“txt”或“csv”，且一次上传文件的总大小不能超过8MB。

   图 4-42 添加上传文件

3. 在添加文件对话框中，单击“上传文件”，完成添加文件的操作。您添加的文件内容将自动呈现在“未标注”的“标注对象列表”中。

删除文件

通过数据删除操作，可将需要丢弃的文件数据快速删除。

- 在“未标注”页面中，单击选中需要删除的文本，然后单击左上角“删除”，即可完成文本的删除操作。
- 在“已标注”页面中，选中待删除的文本，然后单击“删除”，删除单个文本。或者勾选“选择当前页”选中该页面所有文本，然后单击左上角“删除”，即可完成当前页所有文本的删除操作。

其中，被选中的文本，其背景将显示为蓝色。如果当前页面无选中文本时，“删除”按钮为灰色，无法执行删除操作。

4.3.7 声音分类

由于模型训练过程需要大量有标签的音频数据，因此在模型训练之前需对没有标签的音频添加标签。通过ModelArts您可对音频进行一键式批量添加标签，快速完成对音频的标注操作，也可以对已标注音频修改或删除标签进行重新标注。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据集”，进入“数据集”管理页面。
2. 在数据集列表中，基于“标注类型”选择需要进行标注的数据集，单击数据集名称进入数据集概览页。

   此操作默认进入数据集当前版本的概览页，如果需要对其他版本进行数据标注，请先在“版本管理”操作中，将需要进行数据标注的版本设置为“当前版本。”详细操作指导请参见管理数据集版本。
3. 在数据集概览页中，单击右上角“开始标注”，进入数据集详情页。数据集详情页默认展示此数据集下全部数据。

同步数据源

ModelArts会自动从数据集输入位置同步数据至数据集详情页，包含数据及标注信息。

为了快速获取OBS桶中最新数据，可在数据集详情页的“未标注”页签中，单击“同步数据源”，快速将通过OBS上传的数据添加到数据集中。

标注音频

数据集详情页中，展示了此数据集中“未标注”和“已标注”的音频，默认显示“未标注”的音频列表。单击音频左侧 ，即可进行音频的试听。

1. 在“未标注”页签，勾选需进行标注的音频。
   - 手工点选：在音频列表中，单击音频，当右上角出现蓝色勾选框时，表示已勾选。可勾选同类别的多个音频，一起添加标签。
   - 批量选中：如果音频列表的当前页，所有音频属于一种类型，可以在列表的右上角单击“选择当前页”，则当前页面所有的音频将选中。
2. 添加标签。
   a. 在右侧的“标签”区域中，单击“标签”下侧的文本框中设置标签。

      方式一（已存在标签）：单击“标签”下方的文本框，在快捷键下拉列表中选择快捷键，然后在标签文本输入框中选择已有的标签名称，然后单击“确定”。

      方式二（新增标签）：在“标签”下方的文本框中，在快捷键下拉列表中选择快捷键，然后在标签文本输入框中输入新的标签名称，然后单击“确定”。
   b. 选中的音频将被自动移动至“已标注”页签，且在“未标注”页签中，标签的信息也将随着标注步骤进行更新，如增加的标签名称、各标签对应的音频数量。

说明

快捷键的使用说明：为标签指定快捷键后，当您选择一段音频后，在键盘中按一下快捷键，即可为此音频增加为此快捷键对应的标签。例如“aa”标签对应的快捷键是“1”，在数据标注过程中，选中1个或多个文件，按“1”，界面将提示是否需要将此文件标注为“aa”标签，单击确认即可完成标注。

快捷键对应的是标签，1个标签对应1个快捷键。不同的标签，不能指定为同一个快捷键。快捷键的使用，可以大大提升标注效率。

图 4-43 添加音频标签

查看已标注音频

在数据集详情页，单击“已标注”页签，您可以查看已完成标注的音频列表。单击音频，可在右侧的“选中文件标签”中了解当前音频的标签信息。

修改标注

当数据完成标注后，您还可以进入“已标注”页签，对已标注的数据进行修改。

- **基于音频修改**

  在数据标注页面，单击“已标注”页签，然后在音频列表中选中待修改的音频（选择一个或多个）。在右侧标签信息区域中对标签进行修改。

  - 修改标签：在“选中文件标签”区域中，单击操作列的编辑图标，然后在文本框中输入正确的标签名，然后单击确定图标完成修改。
  - 删除标签：在“选中文件标签”区域中，单击操作列的删除图标删除该标签。

- **基于标签修改**

  在数据集详情页面，单击“已标注”页签，在音频列表右侧，显示全部标签的信息。

图 4-44 全部标签信息

  - 修改标签：单击操作列的编辑图标，然后在弹出的对话框中输入修改后的标签名，然后单击“确定”完成修改。修改后，之前添加了此标签的音频，都将被标注为新的标签名称。

- 删除标签：单击操作列的删除图标，在弹出的对话框中，根据提示框选择需要删除的对象，然后单击“确定”完成删除。

添加音频

除了数据集输入位置自动同步的数据外，您还可以在ModelArts界面中，直接添加音频，用于数据标注。

1. 在数据集详情页面，单击“未标注”页签，然后单击左上角“添加音频”。
2. 在弹出的“添加音频”对话框中，单击“添加音频”。

   选择本地环境中需要上传的音频，仅支持WAV格式音频文件，单个音频文件不能超过4MB，且单次上传的音频文件总大小不能超过8MB。
3. 在添加音频对话框中，单击“确定”，完成添加音频的操作。

   您添加的音频将自动呈现在“未标注”的音频列表中。且音频将自动存储至此“数据集输入位置”对应的OBS目录中。

删除音频

通过数据删除操作，可将需要丢弃的音频数据快速删除。

在“未标注”或“已标注”页面中，选中需要删除的音频，或者选择“选择当前页”选中该页面所有音频，然后单击左上角“删除音频”，在弹出的对话框中，根据实际情况选择是否勾选“同时删除源文件”，确认信息无误后，单击“确定”完成音频删除操作。

其中，被选中的音频，其右上角将显示为勾选状态。如果当前页面无选中音频时，“删除音频”按钮为灰色，无法执行删除操作。

说明

如果勾选了“同时删除源文件”，删除音频操作是将删除对应OBS目录下存储的音频。此操作可能会影响已使用此源文件的其他数据集或数据集版本，有可能导致展示异常或训练/推理异常。删除后，数据将无法恢复，请谨慎操作。

4.3.8 语音内容

由于模型训练过程需要大量有标签的音频数据，因此在模型训练之前需对没有标签的音频添加标签。通过ModelArts您可对音频进行一键式批量添加标签，快速完成对音频的标注操作，也可以对已标注音频修改或删除标签进行重新标注。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据集”，进入“数据集”管理页面。
2. 在数据集列表中，基于“标注类型”选择需要进行标注的数据集，单击数据集名称进入数据集概览页。

   此操作默认进入数据集当前版本的概览页，如果需要对其他版本进行数据标注，请先在“版本管理”操作中，将需要进行数据标注的版本设置为“当前版本。”详细操作指导请参见管理数据集版本。
3. 在数据集概览页中，单击右上角“开始标注”，进入数据集详情页。数据集详情页默认展示此数据集下全部数据。

同步数据源

ModelArts会自动从数据集输入位置同步数据至数据集详情页，包含数据及标注信息。

为了快速获取OBS桶中最新数据，可在数据集详情页的“未标注”页签中，单击“同步数据源”，快速将通过OBS上传的数据添加到数据集中。

标注音频

数据集详情页中，展示了此数据集中“未标注”和“已标注”的音频，默认显示“未标注”的音频列表。

1. 在“未标注”页签左侧音频列表中，单击目标音频文件，在右侧的区域中出现音频，单击音频下方，即可进行音频播放。
2. 根据播放内容，在下方“语音内容”文本框中填写音频内容。
3. 输入内容后单击下方的“确认标注”按钮完成标注。音频将被自动移动至“已标注”页签。

图 4-45 语音内容音频标注

查看已标注音频

在数据集详情页，单击“已标注”页签，您可以查看已完成标注的音频列表。单击音频，可在右侧的“语音内容”文本框中了解当前音频的内容信息。

修改标注

当数据完成标注后，您还可以进入“已标注”页签，对已标注的数据进行修改。

在数据集详情页，单击“已标注”页签，然后在音频列表中选中待修改的音频。在右侧标签信息区域中修改“语音内容”文本框中的内容，单击下方的“确认标注”按钮完成修改。

添加音频

除了数据集输入位置自动同步的数据外，您还可以在ModelArts界面中，直接添加音频，用于数据标注。

1. 在数据集详情页面，单击“未标注”页签，然后单击左上角“添加音频”。
2. 在弹出的“添加音频”对话框中，单击“添加音频”。

   选择本地环境中需要上传的音频，仅支持WAV格式音频文件，单个音频文件不能超过4MB，且单次上传的音频文件总大小不能超过8MB。
3. 在添加音频对话框中，单击“确定”，完成添加音频的操作。

   您添加的音频将自动呈现在“未标注”的音频列表中。且音频将自动存储至此“数据集输入位置”对应的OBS目录中。

删除音频

通过数据删除操作，可将需要丢弃的音频数据快速删除。

在“未标注”或“已标注”页面中，选中需要删除的音频，然后单击左上角“删除音频”，在弹出的对话框中，根据实际情况选择是否勾选“同时删除源文件”，确认信息无误后，单击“确定”完成音频删除操作。

说明

如果勾选了“同时删除源文件”，删除音频操作是将删除对应OBS目录下存储的音频。此操作可能会影响已使用此源文件的其他数据集或数据集版本，有可能导致展示异常或训练/推理异常。删除后，数据将无法恢复，请谨慎操作。

4.3.9 语音分割

由于模型训练过程需要大量有标签的音频数据，因此在模型训练之前需对没有标签的音频添加标签。通过ModelArts您可对音频添加标签，快速完成对音频的标注操作，也可以对已标注音频修改或删除标签进行重新标注。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据集”，进入“数据集”管理页面。
2. 在数据集列表中，基于“标注类型”选择需要进行标注的数据集，单击数据集名称进入数据集概览页。

   此操作默认进入数据集当前版本的概览页，如果需要对其他版本进行数据标注，请先在“版本管理”操作中，将需要进行数据标注的版本设置为“当前版本。”详细操作指导请参见管理数据集版本。
3. 在数据集概览页中，单击右上角“开始标注”，进入数据集详情页。数据集详情页默认展示此数据集下全部数据。

同步数据源

ModelArts会自动从数据集输入位置同步数据至数据集详情页，包含数据及标注信息。

为了快速获取OBS桶中最新数据，可在数据集详情页的“未标注”页签中，单击“同步数据源”，快速将通过OBS上传的数据添加到数据集中。

标注音频

数据集详情页中，展示了此数据集中“未标注”和“已标注”的音频，默认显示“未标注”的音频列表。

1. 在“未标注”页签左侧音频列表中，单击目标音频文件，在右侧的区域中出现音频，单击音频下方，即可进行音频播放。
2. 根据播放内容，选取合适的音频段，在下方“语音内容”文本框中填写音频标签和内容。

图 4-46 语音标签音频标注

3. 输入内容后单击下方的“确认标注”按钮完成标注。音频将被自动移动至“已标注”页签。

查看已标注音频

在数据集详情页，单击“已标注”页签，您可以查看已完成标注的音频列表。单击音频，可在右侧的“语音内容”文本框中了解当前音频的内容信息。

修改标注

当数据完成标注后，您还可以进入“已标注”页签，对已标注的数据进行修改。

- 修改标签：在数据集详情页，单击“已标注”页签，然后在音频列表中选中待修改的音频。在右侧标签信息区域中修改“语音内容”中的“标签”和“内容”，单击下方的“确认标注”按钮完成修改。
- 删除标签：单击目标编号操作列的 ，删除该段音频的标注。您也可以单击标注音频文件上方的叉号删除标注，然后单击“确认标注”。

添加音频

除了数据集输入位置自动同步的数据外，您还可以在ModelArts界面中，直接添加音频，用于数据标注。

1. 在数据集详情页面，单击“未标注”页签，然后单击左上角“添加音频”。
2. 在弹出的“添加音频”对话框中，单击“添加音频”。

   选择本地环境中需要上传的音频，仅支持WAV格式音频文件，单个音频文件不能超过4MB，且单次上传的音频文件总大小不能超过8MB。
3. 在添加音频对话框中，单击“确定”，完成添加音频的操作。

   您添加的音频将自动呈现在“未标注”的音频列表中。且音频将自动存储至此“数据集输入位置”对应的OBS目录中。

删除音频

通过数据删除操作，可将需要丢弃的音频数据快速删除。

在“未标注”或“已标注”页面中，选中需要删除的音频，然后单击左上角“删除音频”，在弹出的对话框中，根据实际情况选择是否勾选“同时删除源文件”，确认信息无误后，单击“确定”完成音频删除操作。

说明

如果勾选了“同时删除源文件”，删除音频操作是将删除对应OBS目录下存储的音频。此操作可能会影响已使用此源文件的其他数据集或数据集版本，有可能导致展示异常或训练/推理异常。删除后，数据将无法恢复，请谨慎操作。

4.3.10 视频标注

由于模型训练过程需要大量有标签的视频数据，因此在模型训练之前需对没有标签的视频添加标签。通过ModelArts您可对视频添加标签，快速完成对视频的标注操作，也可以对已标注视频修改或删除标签进行重新标注。

开始标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理> 数据集”，进入“数据集”管理页面。
2. 在数据集列表中，基于“标注类型”选择需要进行标注的数据集，单击数据集名称进入数据集概览页。

   此操作默认进入数据集当前版本的概览页，如果需要对其他版本进行数据标注，请先在“版本管理”操作中，将需要进行数据标注的版本设置为“当前版本。”详细操作指导请参见**管理数据集版本**。
3. 在数据集概览页中，单击右上角“开始标注”，进入数据集详情页。数据集详情页默认展示此数据集下全部数据。

同步数据源

ModelArts会自动从数据集输入位置同步数据至数据集详情页，包含数据及标注信息。

为了快速获取OBS桶中最新数据，可在数据集详情页的“未标注”页签中，单击“同步数据源”，快速将通过OBS上传的数据添加到数据集中。

视频标注

数据集详情页中，展示了此数据集中“未标注”和“已标注”的视频。

1. 在“未标注”页签左侧视频列表中，单击目标视频文件，打开标注页面。
2. 在标注页面中，播放视频，当视频播放至待标注时间时，单击进度条中的暂停按钮，将视频暂停至某一画面。
3. 在左侧区域选择标注框，默认为矩形框。使用鼠标在视频画面中框出目标，然后在弹出的添加标签文本框中，直接输入新的标签名，在文本框前面选中标签颜色，单击“添加”完成1个物体的标注。如果已存在标签，从下拉列表中选择已有的标签，然后单击“添加”完成标注。逐步此画面中所有物体所在位置，一张画面可添加多个标签。

   支持的标注框与“物体检测”类型一致，详细描述请参见**物体检测**章节的**表4-9**。

图 4-47 视频标注

4. 上一个暂停画面标注完成后，在进度条处单击播放按钮继续播放，在需要标注的画面中暂停，然后重复执行步骤3完成整个视频的标注。

   界面右侧将呈现当前视频带标注的时间点。

   图 4-48 当前文件标签信息

5. 单击页面左上角“返回数据标注预览”，页面将自动返回数据集详情页面，同时，标注好的视频将呈现在“已标注”页签下。

修改标注

当数据完成标注后，您还可以进入“已标注”页签，删除标注数据。

- 单击目标编号操作列的 ，删除该段视频的标注。您也可以单击标注视频文件上方的叉号删除标注，然后单击“确认标注”。

在“已标注”页签下，单击目标视频文件，在标注页面右侧的“当前文件标签”下，可单击时间点右侧小三角展开详情，您可以修改或删除标签。

- 修改标签：单击标签右侧的编辑按钮，标签名称可进行修改。
- 删除标签：单击标签右侧的删除按钮，将直接删除此标签。如果单击画面时间右侧的删除按钮，将删除此画面下的所有标签。

图 4-49 修改标注

删除视频

通过数据删除操作，可将需要丢弃的视频数据快速删除。

在“全部”、“未标注”或“已标注”页面中，依次选中需要删除的视频，或者选择“选择当前页”选中该页面所有视频，然后单击左上角“删除视频”。在弹出的对话框中，根据实际情况选择是否勾选“同时删除源文件”，确认信息无误后，单击“确定”完成视频删除操作。

其中，被选中的视频，其左上角将显示为勾选状态。如果当前页面无选中视频时，“删除视频”按钮为灰色，无法执行删除操作。

说明

如果勾选了“同时删除源文件”，删除视频操作将删除对应OBS目录下存储的视频，此操作可能会影响已使用此源文件的其他数据集或数据集版本，有可能导致展示异常或训练/推理异常。删除后，数据将无法恢复，请谨慎操作。

4.4 导入数据

4.4.1 导入操作

数据集创建完成后，一方面，可以直接从设置的数据集输入位置直接同步数据，另一方面，您还可以通过导入数据集的操作，导入更多数据。当前支持从OBS目录导入或从Manifest文件导入两种方式。

前提条件

- 已存在创建完成的数据集。
- 需导入的数据，已存储至OBS中。Manifest文件也需要存储至OBS。
- 确保数据存储的OBS桶与ModelArts在同一区域。

导入方式

导入方式分为“OBS目录”和“Manifest文件”两种。

- OBS目录：指需要导入的数据集已提前存储至OBS目录中。此时需选择用户具备权限的OBS路径，且OBS路径内的目录结构需满足规范，详细规范请参见**从OBS目录导入的规范说明**。当前只有“图像分类”、“物体检测”、“表格”、“文本分类”和“声音分类”类型的数据集，支持从OBS目录导入数据。其他类型只支持Manifest文件导入数据集的方式。
- Manifest文件：指数据集为Manifest文件格式，Manifest文件定义标注对象和标注内容的对应关系，且Manifest文件已上传至OBS中。Manifest文件的规范请参见**导入Manifest文件的规范说明**。

说明

导入“物体检测”类型数据集前，您需要保证标注文件的标注范围不超过原始图片大小，否则可能存在导入失败的情况。

表 4-13 不同类型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
图像分类	支持 格式规范：**图像分类**	支持 格式规范：**图像分类**
物体检测	支持 格式规范：**物体检测**	支持 格式规范：**物体检测**
图像分割	支持 格式规范：**图像分割**	支持 格式规范：**图像分割**
声音分类	支持 格式规范：**声音分类**	支持 格式规范：**声音分类**
语音内容	-	支持 格式规范：**语音分割**
语音分割	-	支持 格式规范：**语音内容**
文本分类	支持 格式规范：**文本分类**	支持 格式规范：**文本分类**
命名实体	-	支持 格式规范：**文本命名实体**
文本三元组	-	支持 格式规范：**文本三元组**
表格	支持 还支持从DWS、DLI、MRS导入数据。 格式规范：**表格**	-

数据集类型	OBS目录导入	Manifest文件导入
视频	-	支持 格式规范：视频标注
自由格式	-	-

从 OBS 目录导入

不同类型的数据集，导入操作界面的示意图存在区别，请参考界面信息了解当前类型数据集的示意图。当前操作指导以图像分类的数据集为例。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理 >数据集”，进入“数据集”管理页面。
2. 在数据集所在行，单击操作列的“更多 > 导入”。

   或者，您可以单击数据集名称，进入数据集“概览”页，在页面右上角单击“导入”。
3. 在“导入”对话框中，设置“导入方式”为“OBS目录”，然后在“对象存储服务（OBS）目录”中，设置数据存储的路径。然后单击“确定”。

   说明

   针对“表格”类型的数据集，导入时，支持从OBS、DWS、DLI和MRS等数据源。其导入时的设置和数据要求，与创建数据集相同，详细参数可参见创建数据集时表格类型的参数说明。

图 4-50 导入数据集-OBS 目录

导入成功后，数据将自动同步到数据集中。您可以在“数据集”页面，单击数据集的名称，查看详细数据并进行数据标注。

从 Manifest 文件导入

不同类型的数据集，导入操作界面的示意图存在区别，请参考界面信息了解当前类型数据集的示意图。当前操作指导以物体检测类型的数据集为例。其中，“表格”类型数据集不支持从Manifest文件导入。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集所在行，单击操作列的“更多 > 导入”。

   或者，您可以单击数据集名称，进入数据集“概览”页，在页面右上角单击“导入”。
3. 在“导入”对话框中，参考如下说明填写参数，然后单击“确定”。
   - “导入方式”：设置为“Manifest文件”。
   - “Manifest文件”：设置Manifest文件存储的OBS路径。
   - “按标签导入”：系统将自动获取此数据集的标签，您可以单击“添加标签”添加，也可以单击标签右侧的删除图标删除标签。此字段为可选字段，您也可以在导入数据集后，在标注数据操作时，添加或删除标签。
   - “同时导入标签”：勾选表示将Manifest文件中定义的标签一并导入ModelArts数据集中。
   - “只导入难例”：难列指Manifest文件中的“hard”属性，勾选此参数，表示此导入操作，只导入Manifest文件“hard”属性中数据信息。

   图 4-51 导入数据集

导入成功后，数据将自动同步到数据集中。您可以在“数据集”页面，单击数据集的名称进入数据集概览页，然后在概览页右上角单击“开始标注”进去数据集详情页，查看详细数据并进行数据标注。

4.4.2 从 OBS 目录导入的规范说明

导入数据集时，使用存储在OBS的数据时，数据的存储目录以及文件名称需满足ModelArts的规范要求。

当前只有“图像分类”、“物体检测”、“文本分类”、“表格”和“声音分类”类型的数据集，支持从OBS目录导入数据。其中，“表格”类型的数据集，支持从OBS、DWS、DLI和MRS等数据源导入数据。

说明

从OBS目录导入数据时，当前操作用户需具备此OBS路径的读取权限。

图像分类

- 图像分类的数据支持两种格式，第一种方式（目录方式）只支持单标签。第二种方式（txt标签文件）支持多标签。
  - 相同标签的图片放在一个目录里，并且目录名字即为标签名。当存在多层目录时，则以最后一层目录为标签名。

    示例如下所示，其中Cat和Dog分别为标签名。

    ```
    dataset-import-example
    ├──Cat
    │      10.jpg
    │      11.jpg
    │      12.jpg
    │
    └──Dog
          1.jpg
          2.jpg
          3.jpg
    ```

  - 当目录下存在对应的txt文件时，以txt文件内容作为图像的标签，优先级高于第一种格式。

    示例如下所示，import-dir-1和imort-dir-2为导入子目录。

    ```
    dataset-import-example
    ├──import-dir-1
    │      10.jpg
    │      10.txt
    │      11.jpg
    │      11.txt
    │      12.jpg
    │      12.txt
    └──import-dir-2
          1.jpg
          1.txt
          2.jpg
          2.txt
    ```

    单标签的标签文件示例，如1.txt文件内容如下所示：

    ```
    Cat
    ```

    多标签的标签文件示例，如1.txt文件内容如下所示：

    ```
    Cat
    Dog
    ```

- 只支持JPG、JPEG、PNG、BMP格式的图片。单张图片大小不能超过5MB，且单次上传的图片总大小不能超过8MB。

物体检测

- 物体检测的简易模式要求用户将标注对象和标注文件存储在同一目录，并且一一对应，如标注对象文件名为“IMG_20180919_114745.jpg”，那么标注文件的文件名应为“IMG_20180919_114745.xml”。

  物体检测的标注文件需要满足PASCAL VOC格式，格式详细说明请参见**表4-21**。

  示例：

  ```
  ├──dataset-import-example
  │      IMG_20180919_114732.jpg
  │      IMG_20180919_114732.xml
  │      IMG_20180919_114745.jpg
  │      IMG_20180919_114745.xml
  ```

```
│       IMG_20180919_114945.jpg
│       IMG_20180919_114945.xml
```

标注文件的示例如下所示：

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<annotation>
    <folder>NA</folder>
    <filename>bike_1_1593531469339.png</filename>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>554</width>
        <height>606</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>Dog</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <occluded>0</occluded>
        <bndbox>
            <xmin>279</xmin>
            <ymin>52</ymin>
            <xmax>474</xmax>
            <ymax>278</ymax>
        </bndbox>
    </object>
    <object>
        <name>Cat</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <occluded>0</occluded>
        <bndbox>
            <xmin>279</xmin>
            <ymin>198</ymin>
            <xmax>456</xmax>
            <ymax>421</ymax>
        </bndbox>
    </object>
</annotation>
```

- 只支持JPG、JPEG、PNG、BMP格式的图片，单张图片大小不能超过5MB，且单次上传的图片总大小不能超过8MB。

图像分割

- 图像分割的简易模式要求用户将标注对象和标注文件存储在同一目录，并且一一对应，如标注对象文件名为“IMG_20180919_114746.jpg”，那么标注文件的文件名应为“IMG_20180919_114746.xml”。

  图像分割的标注文件基于PASCAL VOC格式增加了字段mask_source和mask_color，格式详细说明请参见**表4-17**。

  示例：

```
├──dataset-import-example
│       IMG_20180919_114732.jpg
│       IMG_20180919_114732.xml
│       IMG_20180919_114745.jpg
│       IMG_20180919_114745.xml
│       IMG_20180919_114945.jpg
│       IMG_20180919_114945.xml
```

  标注文件的示例如下所示：

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<annotation>
```

```
    <folder>NA</folder>
    <filename>image_0006.jpg</filename>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>230</width>
        <height>300</height>
        <depth>3</depth>
    </size>
    <segmented>1</segmented>
    <mask_source>obs://xianao/out/dataset-8153-Jmf5ylLjRmSacj9KevS/annotation/V001/
segmentationClassRaw/image_0006.png</mask_source>
    <object>
        <name>bike</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <mask_color>193,243,53</mask_color>
        <occluded>0</occluded>
        <polygon>
            <x1>71</x1>
            <y1>48</y1>
            <x2>75</x2>
            <y2>73</y2>
            <x3>49</x3>
            <y3>69</y3>
            <x4>68</x4>
            <y4>92</y4>
            <x5>90</x5>
            <y5>101</y5>
            <x6>45</x6>
            <y6>110</y6>
            <x7>71</x7>
            <y7>48</y7>
        </polygon>
    </object>
</annotation>
```

文本分类

文本分类支持导入“txt”和“csv”两种文件类型，文本的编码格式支持“UTF-8”和“GBK”。

文本分类的标注对象和标注文件有2种存放模式。

- 文本和标注合并：文本分类的标注对象和标注内容在一个文本文件内，标注对象与标注内容之间，多个标注内容之间可分别指定分隔符。

  例如，文本文件的内容如下所示。标注对象与标注内容之间采用tab键分隔。

```
手感很好，反应速度很快，不知道以后怎样	positive
三个月前买了一个用的非常好果断把旧手机替换下来尤其在待机方面秒杀	positive
没充一会电源怎么也会发热呢音量健不好用回弹不好	negative
算是给自己的父亲节礼物吧物流很快下单不到24小时就到货了耳机更赞有些低音炮的感觉入耳很紧不会掉棒棒哒	positive
```

- 文本和标注分离：文本分类的标注对象和标注文件均为文本文件，并且以行数进行对应，如标注文件中的第一行表示的是标注对象文件中的第一行的标注。

  例如，标注对象“COMMENTS_20180919_114745.txt”的内容如下所示。

```
手感很好，反应速度很快，不知道以后怎样
三个月前买了一个用的非常好果断把旧手机替换下来尤其在待机方面秒杀
没充一会电源怎么也会发热呢音量健不好用回弹不好
算是给自己的父亲节礼物吧物流很快下单不到24小时就到货了耳机更赞有些低音炮的感觉入耳很紧不会掉棒棒哒
```

  标注文件“COMMENTS_20180919_114745_result.txt”的内容。

```
positive
negative
negative
positive
```

此数据格式要求将标注对象和标注文件存储在同一目录，并且一一对应，如标注对象文件名为“COMMENTS_20180919_114745.txt”，那么标注文件名为“COMMENTS _20180919_114745_result.txt”。

数据文件存储示例：

```
├──dataset-import-example
│      COMMENTS_20180919_114732.txt
│      COMMENTS _20180919_114732_result.txt
│      COMMENTS _20180919_114745.txt
│      COMMENTS _20180919_114745_result.txt
│      COMMENTS _20180919_114945.txt
│      COMMENTS _20180919_114945_result.txt
```

声音分类

声音分类要求用户将相同标签的声音文件放在一个目录里，并且目录名字即为标签名。

示例：

```
dataset-import-example
├──Cat
│      10.wav
│      11.wav
│      12.wav
│
└──Dog
      1.wav
      2.wav
      3.wav
```

表格

表格支持从4种数据源导入数据，分别为对象存储服务（OBS）、数据仓库服务（DWS）、数据湖探索服务（DLI）、MapReduce服务（MRS）。

导入说明：

1. 导入成功的前提是，数据源的schema需要与创建数据集指定的schema保持一致。其中schema指表格的列名和类型，创建数据集时一旦指定，不支持修改。
2. 数据格式不合法，会将数据置为null，详见**表4-6**。
3. 从OBS或者MRS导入csv文件，不会校验数据类型，但是列数需要跟数据集的schema保持一致。

下面分别介绍如下几种数据源导入：

- 从OBS导入数据

  支持从OBS导入csv文件，需要选择文件所在目录，其中csv文件的列数需要跟数据集schema一致。支持自动获取csv文件的schema。

  ```
  ├──dataset-import-example
  │      table_import_1.csv
  │      table_import_2.csv
  │      table_import_3.csv
  │      table_import_4.csv
  ```

- 从DWS导入数据

从DWS导入数据，用户需要选择对应的DWS集群，并输入需要对应的数据库名、表名以及用户名和密码。所导入表的schema(列名和类型)需要跟数据集相同。

- 从DLI导入数据

  从DLI导入数据，用户需要选择DLI队列、数据库和表名称。所选择的表的schema(列名和类型)需与数据集一致，支持自动获取所选择表的schema。DLI的default队列只用作体验，不同帐号间可能会出现抢占的情况，需进行资源排队，不能保证每次都可以得到资源执行相关操作。DLI支持schema映射的功能，即导入的表的schema的字段名称可以不和数据集相同，但类型要保持一致。

- 从MRS导入数据

  只支持从分析集群导入数据，流式集群不支持导入。从MRS服务中导入存储在HDFS上的csv格式的数据，首先需要选择已有的MRS集群，并从HDFS文件列表选择文件名称或所在目录，导入文件的列数需与数据集schema一致。

4.4.3 导入 Manifest 文件的规范说明

Manifest文件中定义了标注对象和标注内容的对应关系。此导入方式是指导入数据集时，使用Manifest文件。选择导入Manifest文件时，可以从OBS导入。当从OBS导入Manifest文件时，需确保当前用户具备Manifest文件所在OBS路径的权限。

说明

Manifest文件编写规范要求较多，推荐使用OBS目录导入方式导入新数据。一般此功能常用于不同区域或不同帐号下ModelArts的数据迁移，即当您已在某一区域使用ModelArts完成数据标注，发布后的数据集可从输出路径下获得其对应的Manifest文件。在获取此Manifest文件后，可将此数据集导入其他区域或者其他帐号的ModelArts中，导入后的数据已携带标注信息，无需重复标注，提升开发效率。

Manifest文件描述的是原始文件和标注信息，可用于标注、训练、推理场景。Manifest文件中也可以只有原始文件信息，没有标注信息，如用于推理场景，或用于生成未标注的数据集。Manifest文件需满足如下要求：

- Manifest文件使用UTF-8编码。文本分类的source数值可以包含中文，其他字段不建议使用中文。
- Manifest文件使用json lines格式（jsonlines.org），一行一个json对象。

```
{"source": "/path/to/image1.jpg", "annotation": … }
{"source": "/path/to/image2.jpg", "annotation": … }
{"source": "/path/to/image3.jpg", "annotation": … }
```

  为了说明方便，下面的Manifest例子格式化为多行的json对象。

- Manifest文件可以由用户、第三方工具或ModelArts数据标注生成，其文件名没有特殊要求，可以为任意合法文件名。为了ModelArts系统内部使用方便，ModelArts数据标注功能生成的文件名由如下字符串组成：“DatasetName-VersionName.manifest”。例如，“animal-v20190123113030412 3.manifest”。

图像分类

```
{
   "source":"s3://path/to/image1.jpg",
   "usage":"TRAIN",
   "hard":"true",
   "hard-coefficient":0.8,
   "id":"0162005993f8065ef47eefb59d1e4970",
   "annotation": [
      {
         "type": "modelarts/image_classification",
         "name": "cat",
```

```
            "property": {
                "color":"white",
                "kind":"Persian cat"
            },
            "hard":"true",
            "hard-coefficient":0.8,
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        },
        {
            "type": "modelarts/image_classification",
            "name":"animal",
            "annotated-by":"modelarts/active-learning",
            "confidence": 0.8,
            "creation-time":"2019-01-23 11:30:30"
        }],
    "inference-loc":"/path/to/inference-output"
}
```

表 4-14 字段说明

字段	是否必选	说明
source	是	被标注对象的URI。数据来源的类型及示例请参考**表4-15**。
usage	否	默认为空，取值范围： ● TRAIN：指明该对象用于训练。 ● EVAL：指明该对象用于评估。 ● TEST：指明该对象用于测试。 ● INFERENCE：指明该对象用于推理。 如果没有给出该字段，则使用者自行决定如何使用该对象。
id	否	此参数为系统导出的样本id，导入时可以不用填写。
annotation	否	如果不设置，则表示未标注对象。annotation值为一个对象列表，详细参数请参见**表4-16**。
inference-loc	否	当此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。

表 4-15 数据来源类型

类型	示例
OBS	“source”：“s3://path-to-jpg”
Content	“source”：“content://I love machine learning”

表 4-16 annotation 对象说明

字段	是否必选	说明
type	是	标签类型。取值范围为： ● image_classification：图像分类 ● text_classification：文本分类 ● text_entity：文本命名实体 ● object_detection：对象检测 ● audio_classification：声音分类 ● audio_content：声音内容 ● audio_segmentation：声音起止点
name	是/否	对于分类是必选字段，对于其他类型为可选字段，本示例为图片分类名称。
id	是/否	标签ID。对于三元组是必选字段，对于其他类型为可选字段。三元组的实体标签ID格式为“E+数字”，比如“E1”、“E2”，三元组的关系标签ID格式为“R+数字”，例如“R1”、“R2”。
property	否	包含对标注的属性，例如本示例中猫有两个属性，颜色（color）和品种（kind）。
hard	否	表示是否是难例。“True”表示该标注是难例，“False”表示该标注不是难例。
annotated-by	否	默认为“human”，表示人工标注。 ● human
creation-time	否	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。
confidence	否	表示机器标注的置信度。范围为0～1。

图像分割

```
{
   "annotation": [{
      "annotation-format": "PASCAL VOC",
      "type": "modelarts/image_segmentation",
      "annotation-loc": "s3://path/to/annotation/image1.xml",
      "creation-time": "2020-12-16 21:36:27",
      "annotated-by": "human"
   }],
   "usage": "train",
   "source": "s3://path/to/image1.jpg",
   "id": "16d196c19bf61994d7deccafa435398c",
   "sample-type": 0
}
```

- “source”、“usage”、“annotation”等参数说明与图像分类一致，详细说明请参见表4-14。
- “annotation-loc”：对于图像分割、物体检测是必选字段，对于其他类型是可选字段，标注文件的存储路径。

- “annotation-format”：描述标注文件的格式，可选字段，默认为“PASCAL VOC”。目前只支持“PASCAL VOC”。
- “**sample-type**”：样本格式，0表示图片，1表示文本，2表示语音，4表示表格，6表示视频

表 4-17 PASCAL VOC 格式说明

字段	是否必选	说明
folder	是	表示数据源所在目录。
filename	是	被标注文件的文件名。
size	是	表示图像的像素信息。 ● width：必选字段，图片的宽度。 ● height：必选字段，图片的高度。 ● depth：必选字段，图片的通道数。
segmented	是	表示是否用于分割。
mask_source	否	表示图像分割保存的mask路径
object	是	表示物体检测信息，多个物体标注会有多个object体。 ● name：必选字段，标注内容的类别。 ● pose：必选字段，标注内容的拍摄角度。 ● truncated：必选字段，标注内容是否被截断（0表示完整）。 ● occluded：必选字段，标注内容是否被遮挡（0表示未遮挡） ● difficult：必选字段，标注目标是否难以识别（0表示容易识别）。 ● confidence：可选字段，标注目标的置信度，取值范围0-1之间。 ● bndbox：必选字段，标注框的类型，可选值请参见**表4-18**。 ● mask_color：必选字段，标签的颜色，以RGB值表示

表 4-18 标注框类型描述

type	形状	标注信息
polygon	多边形	各点坐标。 <x1>100<x1> <y1>100<y1> <x2>200<x2> <y2>100<y2> <x3>250<x3> <y3>150<y3> <x4>200<x4> <y4>200<y4> <x5>100<x5> <y5>200<y5> <x6>50<x6> <y6>150<y6> <x7>100<x7> <y7>100<y7>

示例：

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<annotation>
    <folder>NA</folder>
    <filename>image_0006.jpg</filename>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>230</width>
        <height>300</height>
        <depth>3</depth>
    </size>
    <segmented>1</segmented>
    <mask_source>obs://xianao/out/dataset-8153-Jmf5ylLjRmSacj9KevS/annotation/V001/
segmentationClassRaw/image_0006.png</mask_source>
    <object>
        <name>bike</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <mask_color>193,243,53</mask_color>
        <occluded>0</occluded>
        <polygon>
            <x1>71</x1>
            <y1>48</y1>
            <x2>75</x2>
            <y2>73</y2>
            <x3>49</x3>
            <y3>69</y3>
            <x4>68</x4>
            <y4>92</y4>
            <x5>90</x5>
            <y5>101</y5>
            <x6>45</x6>
            <y6>110</y6>
```

```
            <x7>71</x7>
            <y7>48</y7>
        </polygon>
    </object>
</annotation>
```

文本分类

```
{
    "source": "content://I like this product ",
    "id":"XGDVGS",
    "annotation": [
        {
            "type": "modelarts/text_classification",
            "name": " positive",
            "annotated-by": "human",
            "creation-time": "2019-01-23 11:30:30"
        } ]
}
```

content字段是指被标注的文本（UTF-8编码，可以是中文），其他参数解释与图像分类相同，请参见表4-14。

文本命名实体

```
{
    "source":"content://Michael Jordan is the most famous basketball player in the world.",
    "usage":"TRAIN",
    "annotation":[
        {
            "type":"modelarts/text_entity",
            "name":"Person",
            "property":{
                "@modelarts:start_index":0,
                "@modelarts:end_index":14
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        },
        {
            "type":"modelarts/text_entity",
            "name":"Category",
            "property":{
                "@modelarts:start_index":34,
                "@modelarts:end_index":44
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        }
    ]
}
```

“source”、“usage”、“annotation”等参数说明与图像分类一致，详细说明请参见表4-14。

其中，property的参数解释如表4-19所示。例如，当“"source":"content://Michael Jordan"”时，如果要提取“Michael”，则对应的“start_index”为“0”，“end_index”为“7”。

表 4-19 property 参数说明

参数名	数据类型	说明
@modelarts:start_index	Integer	文本的起始位置，值从0开始，包括start_index所指的字符。
@modelarts:end_index	Integer	文本的结束位置，但不包括end_index所指的字符。

文本三元组

```
{
    "source":"content://"Three Body" is a series of long science fiction novels created by Liu Cix.",
    "usage":"TRAIN",
    "annotation":[
        {
            "type":"modelarts/text_entity",
            "name":"Person",
            "id":"E1",
            "property":{
                "@modelarts:start_index":67,
                "@modelarts:end_index":74
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        },
        {
            "type":"modelarts/text_entity",
            "name":"Book",
            "id":"E2",
            "property":{
                "@modelarts:start_index":0,
                "@modelarts:end_index":12
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        },
        {
            "type":"modelarts/text_triplet",
            "name":"Author",
            "id":"R1",
            "property":{
                "@modelarts:from":"E1",
                "@modelarts:to":"E2"
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        },
        {
            "type":"modelarts/text_triplet",
            "name":"Works",
            "id":"R2",
            "property":{
                "@modelarts:from":"E2",
                "@modelarts:to":"E1"
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        }
    ]
}
```

“source”、“usage”、“annotation”等参数说明与**图像分类**一致，详细说明请参见**表4-14**。

其中，property的参数解释如表5 property参数说明所示。其中，“@modelarts:start_index”和“@modelarts:end_index”和文本命名实体的参数说明一致。例如，当“source”："content://"Three Body" is a series of long science fiction novels created by Liu Cix."时，“Liu Cix”是实体Person（人物），“Three Body”是实体Book（书籍），Person指向Book的关系是Author（作者），Book指向Person的关系是Works（作品）。

表 4-20 property 参数说明

参数名	数据类型	说明
@modelarts:start_index	Integer	三元组实体的起始位置，值从0开始，包括start_index所指的字符。
@modelarts:end_index	Integer	三元组实体的结束位置，但不包括end_index所指的字符。
@modelarts:from	String	三元组关系的起始实体id
@modelarts:to	String	三元组关系的指向实体id

物体检测

```
{
    "source":"s3://path/to/image1.jpg",
    "usage":"TRAIN",
    "hard":"true",
    "hard-coefficient":0.8,
    "annotation": [
        {
            "type":"modelarts/object_detection",
            "annotation-loc": "s3://path/to/annotation1.xml",
            "annotation-format":"PASCAL VOC",
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        }]
}
```

- “source”、“usage”、“annotation”等参数说明与图像分类一致，详细说明请参见表4-14。
- “annotation-loc”：对于物体检测、图像分割是必选字段，对于其他类型是可选字段，标注文件的存储路径。
- “annotation-format”：描述标注文件的格式，可选字段，默认为“PASCAL VOC”。目前只支持“PASCAL VOC”。

表 4-21 PASCAL VOC 格式说明

字段	是否必选	说明
folder	是	表示数据源所在目录。
filename	是	被标注文件的文件名。

字段	是否必选	说明
size	是	表示图像的像素信息。 ● width：必选字段，图片的宽度。 ● height：必选字段，图片的高度。 ● depth：必选字段，图片的通道数。
segmented	是	表示是否用于分割。
object	是	表示物体检测信息，多个物体标注会有多个object体。 ● name：必选字段，标注内容的类别。 ● pose：必选字段，标注内容的拍摄角度。 ● truncated：必选字段，标注内容是否被截断（0表示完整）。 ● occluded：必选字段，标注内容是否被遮挡（0表示未遮挡） ● difficult：必选字段，标注目标是否难以识别（0表示容易识别）。 ● confidence：可选字段，标注目标的置信度，取值范围0-1之间。 ● bndbox：必选字段，标注框的类型，可选值请参见表4-22。

表 4-22 标注框类型描述

type	形状	标注信息
point	点	点的坐标。 <x>100<x> <y>100<y>
line	线	各点坐标。 <x1>100<x1> <y1>100<y1> <x2>200<x2> <y2>200<y2>
bndbox	矩形框	左上和右下两个点坐标。 <xmin>100<xmin> <ymin>100<ymin> <xmax>200<xmax> <ymax>200<ymax>

type	形状	标注信息
polygon	多边形	各点坐标。 <x1>100<x1> <y1>100<y1> <x2>200<x2> <y2>100<y2> <x3>250<x3> <y3>150<y3> <x4>200<x4> <y4>200<y4> <x5>100<x5> <y5>200<y5> <x6>50<x6> <y6>150<y6>
circle	圆形	圆心坐标和半径。 <cx>100<cx> <cy>100<cy> <r>50<r>

示例：

```
<annotation>
  <folder>test_data</folder>
  <filename>260730932.jpg</filename>
  <size>
     <width>767</width>
     <height>959</height>
     <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
     <name>point</name>
     <pose>Unspecified</pose>
     <truncated>0</truncated>
     <occluded>0</occluded>
     <difficult>0</difficult>
     <point>
        <x1>456</x1>
        <y1>596</y1>
     </point>
  </object>
  <object>
     <name>line</name>
     <pose>Unspecified</pose>
     <truncated>0</truncated>
     <occluded>0</occluded>
     <difficult>0</difficult>
     <line>
        <x1>133</x1>
        <y1>651</y1>
        <x2>229</x2>
        <y2>561</y2>
     </line>
  </object>
```

```
  <object>
    <name>bag</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <bndbox>
      <xmin>108</xmin>
      <ymin>101</ymin>
      <xmax>251</xmax>
      <ymax>238</ymax>
    </bndbox>
  </object>
  <object>
    <name>boots</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <hard-coefficient>0.8</hard-coefficient>
    <polygon>
      <x1>373</x1>
      <y1>264</y1>
      <x2>500</x2>
      <y2>198</y2>
      <x3>437</x3>
      <y3>76</y3>
      <x4>310</x4>
      <y4>142</y4>
    </polygon>
  </object>
  <object>
    <name>circle</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <circle>
      <cx>405</cx>
      <cy>170</cy>
      <r>100<r>
    </circle>
  </object>
</annotation>
```

声音分类

```
{
"source":
"s3://path/to/pets.wav",
   "annotation": [
      {
         "type": "modelarts/audio_classification",
         "name":"cat",
         "annotated-by":"human",
         "creation-time":"2019-01-23 11:30:30"
      }
   ]
}
```

“source”、“usage”、“annotation”等参数说明与**图像分类**一致，详细说明请参见**表4-14**。

语音内容

```
{
   "source":"s3://path/to/audio1.wav",
   "annotation":[
```

```
        {
            "type":"modelarts/audio_content",
            "property":{
                "@modelarts:content":"Today is a good day."
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        }
    ]
}
```

- “source”、“usage”、“annotation”等参数说明与图像分类一致，详细说明请参见表4-14。
- “property”中的“@modelarts:content”参数，数据类型为“String”，表示语音内容。

语音分割

```
{
    "source":"s3://path/to/audio1.wav",
    "usage":"TRAIN",
    "annotation":[
        {

"type":"modelarts/audio_segmentation",
            "property":{
                "@modelarts:start_time":"00:01:10.123",
                "@modelarts:end_time":"00:01:15.456",

                "@modelarts:source":"Tom",

                "@modelarts:content":"How are you?"
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        },
        {
            "type":"modelarts/audio_segmentation",
            "property":{
                "@modelarts:start_time":"00:01:22.754",
                "@modelarts:end_time":"00:01:24.145",
                "@modelarts:source":"Jerry",
                "@modelarts:content":"I'm fine, thank you."
            },
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        }
    ]
}
```

- “source”、“usage”、“annotation”等参数说明与图像分类一致，详细说明请参见表4-14。
- “property”的参数解释如表4-23所示。

表 4-23 “property”参数说明

参数名	数据类型	描述
@modelarts:start_time	String	声音的起始时间，格式为“hh:mm:ss.SSS”。 其中“hh”表示小时，“mm”表示分钟，“ss”表示秒，“SSS”表示毫秒。

参数名	数据类型	描述
@modelarts:end_time	String	声音的结束时间，格式为“hh:mm:ss.SSS”。 其中“hh”表示小时，“mm”表示分钟，“ss”表示秒，“SSS”表示毫秒。
@modelarts:source	String	声音来源。
@modelarts:content	String	声音内容。

视频标注

```
{
    "annotation": [{
        "annotation-format": "PASCAL VOC",
        "type": "modelarts/object_detection",
        "annotation-loc": "s3://path/to/annotation1_t1.473722.xml",
        "creation-time": "2020-10-09 14:08:24",
        "annotated-by": "human"
    }],
    "usage": "train",
    "property": {
        "@modelarts:parent_duration": 8,
        "@modelarts:parent_source": "s3://path/to/annotation1.mp4",
        "@modelarts:time_in_video": 1.473722
    },
    "source": "s3://input/path/to/annotation1_t1.473722.jpg",
    "id": "43d88677c1e9a971eeb692a80534b5d5",
    "sample-type": 0
}
```

- “source”、“usage”、“annotation”等参数说明与图像分类一致，详细说明请参见表4-14。
- “annotation-loc”：对于物体检测、是必选字段，对于其他类型是可选字段，标注文件的存储路径。
- “annotation-format”：描述标注文件的格式，可选字段，默认为“PASCAL VOC”。目前只支持“PASCAL VOC”。
- “**sample-type**”：样本格式，0表示图片，1表示文本，2表示语音，4表示表格，6表示视频。

表 4-24 property 参数说明

参数名	数据类型	说明
@modelarts:parent_duration	Double	标注视频的时长，单位：秒。
@modelarts:time_in_video	Double	标注的视频帧的时间戳，单位：秒。
@modelarts:parent_source	String	标注视频的OBS路径。

表 4-25 PASCAL VOC 格式说明

字段	是否必选	说明
folder	是	表示数据源所在目录。
filename	是	被标注文件的文件名。
size	是	表示图像的像素信息。 ● width：必选字段，图片的宽度。 ● height：必选字段，图片的高度。 ● depth：必选字段，图片的通道数。
segmented	是	表示是否用于分割。
object	是	表示物体检测信息，多个物体标注会有多个object体。 ● name：必选字段，标注内容的类别。 ● pose：必选字段，标注内容的拍摄角度。 ● truncated：必选字段，标注内容是否被截断（0表示完整）。 ● occluded：必选字段，标注内容是否被遮挡（0表示未遮挡） ● difficult：必选字段，标注目标是否难以识别（0表示容易识别）。 ● confidence：可选字段，标注目标的置信度，取值范围0-1之间。 ● bndbox：必选字段，标注框的类型，可选值请参见**表4-26**。

表 4-26 标注框类型描述

type	形状	标注信息
point	点	点的坐标。 <x>100<x> <y>100<y>
line	线	各点坐标。 <x1>100<x1> <y1>100<y1> <x2>200<x2> <y2>200<y2>

type	形状	标注信息
bndbox	矩形框	左上和右下两个点坐标。 <xmin>100<xmin> <ymin>100<ymin> <xmax>200<xmax> <ymax>200<ymax>
polygon	多边形	各点坐标。 <x1>100<x1> <y1>100<y1> <x2>200<x2> <y2>100<y2> <x3>250<x3> <y3>150<y3> <x4>200<x4> <y4>200<y4> <x5>100<x5> <y5>200<y5> <x6>50<x6> <y6>150<y6>
circle	圆形	圆心坐标和半径。 <cx>100<cx> <cy>100<cy> <r>50<r>

示例：

```
<annotation>
  <folder>test_data</folder>
  <filename>260730932_t1.473722.jpg.jpg</filename>
  <size>
     <width>767</width>
     <height>959</height>
     <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
     <name>point</name>
     <pose>Unspecified</pose>
     <truncated>0</truncated>
     <occluded>0</occluded>
     <difficult>0</difficult>
     <point>
        <x1>456</x1>
        <y1>596</y1>
     </point>
  </object>
  <object>
     <name>line</name>
     <pose>Unspecified</pose>
     <truncated>0</truncated>
```

```
    <occluded>0</occluded>
    <difficult>0</difficult>
    <line>
        <x1>133</x1>
        <y1>651</y1>
        <x2>229</x2>
        <y2>561</y2>
    </line>
  </object>
  <object>
    <name>bag</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <bndbox>
        <xmin>108</xmin>
        <ymin>101</ymin>
        <xmax>251</xmax>
        <ymax>238</ymax>
    </bndbox>
  </object>
  <object>
    <name>boots</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <hard-coefficient>0.8</hard-coefficient>
    <polygon>
        <x1>373</x1>
        <y1>264</y1>
        <x2>500</x2>
        <y2>198</y2>
        <x3>437</x3>
        <y3>76</y3>
        <x4>310</x4>
        <y4>142</y4>
    </polygon>
  </object>
  <object>
    <name>circle</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <circle>
        <cx>405</cx>
        <cy>170</cy>
        <r>100<r>
    </circle>
  </object>
</annotation>
```

4.5 导出数据

针对数据集中的数据，包含“已标注”和“未标注”的数据。您可以选中需要部分图片或者通过筛选条件筛选出需要的数据，导出成新的数据集，或者将数据导出至指定的OBS目录下，您可以通过任务历史查看数据导出的历史记录。

说明

目前只有“图像分类”、“物体检测”、“图像分割”、“自由格式”类型的数据集支持导出功能。

- “图像分类”只支持导出txt格式的标注文件。
- “物体检测”只支持导出Pascal VOC格式的XML标注文件。
- “图像分割”只支持导出Pascal VOC格式的XML标注文件以及Mask图像。
- “自由格式”直接导出数据集的所有样本文件。

导出新数据集

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，选择“物体检测”或“图像分类”类型的数据集，单击数据集名称进入“数据集概览页”。

   说明

   针对“自由格式”类型的数据集，单击数据集名称即可进入数据集详情页，直接跳转至步骤4。

3. 在“数据集概览页”，单击右上角“开始标注”，进入数据集详情页。
4. 在数据集详情页面中，选中导出数据或者筛选出数据，然后单击“导出 > 新数据集”。

   图 4-52 选择图片或筛选图片

5. 在弹出的“导出新数据集”对话框中，填写相关信息，然后单击“确定”，开始执行导出操作。

   “名称”：新数据集名称。

   “保存路径”：表示新数据集的输入路径，即当前数据导出后存储的OBS路径。

   “输出路径”：表示新数据集的输出路径，即新数据集在完成标注后输出的路径。“输出路径”不能与“保存路径”为同一路径，且“输出路径”不能是“保存路径”的子目录。

   “导出范围”：“导出当前选中样本”，或者“导出当前筛选条件下的所有样本”。

   “开启难例属性”：设置是否开启难例。

图 4-53 导出新数据集

6. 数据集导出成功后，您可以前往“数据集”列表中，查看到新的数据集。

导出数据至 OBS

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，选择“物体检测”或“图像分类”类型的数据集，单击数据集名称进入“数据集概览页”。

   说明

   针对“自由格式”类型的数据集，单击数据集名称即可进入数据集详情页，直接跳转至步骤4。

3. 在“数据集概览页”，单击右上角“开始标注”，进入数据集详情页。
4. 在数据集详情页面中，选中导出数据或者筛选出数据，然后单击“导出 > 至对象存储服务（OBS）”。

图 4-54 选择图片或筛选图片导出

5. 在弹出的“导出至对象存储服务（OBS）”对话框中，填写相关信息，然后单击“确定”，开始执行导出操作。

   “保存路径”：即导出数据存储的路径。建议不要将数据存储至当前数据集所在的输入路径或输出路径。

   “导出范围”：“导出当前选中样本”，或者“导出当前筛选条件下的所有样本”。

   “开启难例属性”：设置是否开启难例。

图 4-55 导出至 OBS

6. 数据导出成功后，您可以前往您设置的保存路径，查看到存储的数据。

任务历史

当您导出新数据集或导出数据至OBS，您可以通过任务历史查看导出任务明细。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，选择“物体检测”或“图像分类”类型的数据集，单击数据集名称进入“数据集概览页”。

   说明

   针对“自由格式”类型的数据集，单击数据集名称即可进入数据集详情页，直接跳转至步骤4。

3. 在“数据集概览页”，单击右上角“开始标注”，进入数据集详情页。
4. 在数据集详情页面中，选中导出数据或者筛选出数据，然后单击“导出 > 任务历史”。
5. 在弹出的“任务历史”对话框中，可以查看该数据集之前的导出任务历史。包括“任务ID”、“创建时间”、“导出方式”、“导出路径”、“导出样本总数”和“导出状态”。

图 4-56 任务历史

4.6 修改数据集

对于已创建的数据集，您可以修改数据集的基本信息以匹配业务变化。

前提条件

已存在创建完成的数据集。

修改数据集基本信息

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。

2. 在数据集列表中，单击操作列的“更多 > 修改”。
   或者，您可以单击数据集名称，进入数据集“概览”页，在页面右上角单击“修改”。
3. 参考表4-27修改数据集基本信息，然后单击“确定”完成修改。

图 4-57 修改数据集

表 4-27 参数说明

参数	说明
名称	数据集的名称，名称只能是字母、数字、下划线或者中划线组成的合法字符串。
描述	数据集的简要描述。
标签集	针对不同类型数据集，标签集的设置不同，请参考创建数据集（旧版）中不同类型数据集的详细参数指导进行修改。当前对标签集个数没有限制。

4.7 发布数据集

ModelArts在数据集管理过程中，针对同一个数据源，对不同时间标注后的数据，按版本进行区分，方便后续模型构建和开发过程中，选择对应的数据集版本进行使用。数据标注完成后，您可以将数据集当前状态进行发布，生成一个新的数据集版本。

关于数据集版本

- 针对刚创建的数据集（未发布前），无数据集版本信息，必须执行发布操作后，才能应用于模型开发或训练。

- 数据集版本，默认按V001、V002递增规则进行命名，您也可以在发布时自定义设置。
- 您可以将任意一个版本设置为当前目录，即表示数据集列表中进入的数据集详情，为此版本的数据及标注信息。
- 针对每一个数据集版本，您可以通过“存储路径”参数，获得此版本对应的Manifest文件格式的数据集。可用于导入数据或难例筛选操作。
- 表格数据集暂不支持切换版本。

发布数据集

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，单击操作列的“发布”。

   或者，您可以单击数据集名称，进入数据集“概览”页，在页面右上角单击“发布”。
3. 在“发布新版本”弹出框中，填写发布数据集的相关参数，然后单击“确定”。

表 4-28 发布数据集的参数说明

参数	描述
“版本名称”	默认按V001、V002递增规则进行命名，您也可以自定义版本名称。版本名称只能包含字母、数字、中划线或下划线。
“版本格式”	仅“表格”类型数据集支持设置版本格式，支持“CSV”和“CarbonData”两种。 **说明** 如果导出的CSV文件中存在以“=”“+”“-”和“@”开头的命令时，为了安全考虑，ModelArts会自动加上Tab键，并对双引号进行转义处理。
“数据切分”	仅“图像分类”、“物体检测”、“文本分类”和“声音分类”类型数据集支持进行数据切分功能。 默认不启用。启用后，需设置对应的训练验证比例。 输入“训练集比例”，数值只能是0~1区间内的数。设置好“训练集比例”后，“验证集比例”自动填充。“训练集比例”加“验证集比例”等于1。 “训练集比例”即用于训练模型的样本数据比例；“验证集比例”即用于验证模型的样本数据比例。“训练验证比例”会影响训练模板的性能。
“描述”	针对当前发布的数据集版本的描述信息。
“开启难例属性”	仅“图像分类”和“物体检测”类型数据集支持难例属性。 默认不开启。启用后，会将此数据集的难例属性等信息写入对应的Manifest文件中。

图 4-58 发布数据集

版本发布后，您可以前往版本管理查看详细信息。系统默认将最新的版本作为当前目录。

数据集发布后，相关文件的目录结构说明

由于数据集是基于OBS目录管理的，发布为新版本后，对应的数据集输出位置，也将基于新版本生成目录。

以图像分类为例，数据集发布后，对应OBS路径下生成，其相关文件的目录如下所示。

```
|-- user-specified-output-path
   |-- DatasetName-datasetId
      |-- annotation
         |-- VersionMame1
            |-- VersionMame1.manifest
         |-- VersionMame2
            ...
         |-- ...
```

以物体检测为例，如果数据集导入的是Manifest文件，在数据集发布后，其相关文件的目录结构如下。

```
|-- user-specified-output-path
   |-- DatasetName-datasetId
      |-- annotation
         |-- VersionMame1
            |-- VersionMame1.manifest
            |-- annotation
               |-- file1.xml
         |-- VersionMame2
            ...
         |-- ...
```

以视频标注为例，在数据集发布后，标注结果将标注结果文件（XML）存放在数据集输出目录下。

```
|-- user-specified-output-path
    |-- DatasetName-datasetId
       |-- annotation
          |-- VersionMame1
```

```
            |-- VersionMame1.manifest
            |-- annotations
              |-- images
                |-- videoName1
                  |-- videoName1.timestamp.xml
                |-- videoName2
                  |-- videoName2.timestamp.xml
        |-- VersionMame2
            ...
        |-- ...
```

视频标注的关键帧存在数据集的输入目录下。

```
|-- user-specified-input-path
    |-- images
      |-- videoName1
          |-- videoName1.timestamp.jpg
      |-- videoName2
          |-- videoName2.timestamp.jpg
```

4.8 删除数据集

如果数据集不再使用，您可以删除数据集释放资源。

说明

删除数据集后，数据集对应的数据集输入位置和数据集输出位置对应的OBS目录下，如果需要删除OBS目录下的数据释放资源，建议前往OBS管理控制台，删除对应的数据，然后再删除OBS文件夹。

操作步骤

1. 在“数据管理>数据集”页面中，单击数据集操作列的“更多 > 删除”。
2. 在弹出的对话框中，单击“确定”，确认删除此数据集。

   说明

   删除后，数据集的版本管理等功能无法恢复，请谨慎操作。但是，此数据集对应的原始数据和标注数据依然存储在OBS中。

4.9 管理数据集版本

数据标注完成后，您可以发布成多个版本对数据集进行管理。针对已发布生产的数据集版本，您可以通过查看数据集演进过程、设置当前版本、删除版本等操作，对数据集进行管理。数据集版本的相关说明，请参见关于数据集版本。

发布为新版本的说明，请参见发布数据集。

查看数据集演进过程

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，单击操作列的“更多 > 版本管理”，进入数据集“版本管理”页面。

   您可以查看数据集的基本信息，并在左侧查看版本及其发布时间。

图 4-59 查看数据集版本

设置当前版本

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，单击操作列的“更多 > 版本管理”，进入数据集“版本管理”页面。
3. 在“版本管理”页面中，选择对应的数据集版本，在数据集版本基本信息区域，单击“设置为当前版本”。设置完成后，版本名称右侧将显示为“当前版本”。

说明

只有状态为“正常”的版本，才能被设置为当前版本。

图 4-60 设置当前版本

删除数据集版本

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，单击操作列的“更多 > 版本管理”，进入数据集“版本管理”页面。
3. 选择需删除的版本所在行，单击操作列的“删除”。在弹出的对话框中确认信息，然后单击“确定”完成删除操作。

说明

删除数据集版本不会删除原始数据，数据及其标注信息仍存在在对应的OBS目录下。但是，执行删除操作后，无法在ModelArts管理控制台清晰的管理数据集版本，请谨慎操作。

4.10 智能标注

除了人工标注外，ModelArts还提供了智能标注功能，快速完成数据标注，为您节省70%以上的标注时间。智能标注是指基于当前标注阶段的标签及图片学习训练，选中系统中已有的模型进行智能标注，快速完成剩余图片的标注操作。

背景信息

- 目前只有“图像分类”和“物体检测”类型的数据集支持智能标注功能。
- 启动智能标注时，需数据集存在至少2种标签，且每种标签已标注的图片不少于5张。
- 启动智能标注时，必须存在未标注图片。
- 启动智能标注前，保证当前系统中不存在正在进行中的智能标注任务。
- 检查用于标注的图片数据，确保您的图片数据中，不存在RGBA四通道图片。如果存在四通道图片，智能标注任务将运行失败，因此，请从数据集中删除四通道图片后，再启动智能标注。

智能标注

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理 > 数据集”，进入“数据集”管理页面。
2. 在数据集列表中，选择“物体检测”或“图像分类”类型的数据集，单击操作列的“智能标注”启动智能标注作业。
3. 在弹出的“启动智能标注”对话框中，选择智能标注类型，可选“主动学习”或者“预标注”，详见表4-29和表4-30。

表 4-29 主动学习

参数	说明
智能标注类型	“主动学习”。“主动学习”表示系统将自动使用半监督学习、难例筛选等多种手段进行智能标注，降低人工标注量，帮助用户找到难例。
算法类型	针对“图像分类”类型的数据集，您需要选择以下参数。 “快速型”：仅使用已标注的样本进行训练。 “精准型”：会额外使用未标注的样本做半监督训练，使得模型精度更高。

表 4-30 预标注

参数	说明
智能标注类型	“预标注”。“预标注”表示选择用户AI应用管理里面的AI应用,选择模型时需要注意模型类型和数据集的标注类型相匹配。预标注结束后，如果标注结果符合平台定义的标准标注格式，系统将进行难例筛选，该步骤不影响预标注结果。
选择模型及版本	• “我的AI应用”。您可以根据实际需求选择您的AI应用。您需要在目标AI应用的左侧单击下拉三角标，选择合适的版本。您的AI应用导入参见**创建AI应用**。 • “我的订阅”。您可以根据实际需求选择AI Gallery中已订阅的AI应用。您需要在目标AI应用的左侧单击下拉三角标，选择合适的版本。查找AI应用参见**我的订阅模型**。
计算节点规格	在下拉框中，您可以选择目前ModelArts支持的节点规格选项。
计算节点个数	默认为1。您可以根据您的实际情况选择，最大为5。

说明

- 针对“物体检测”类型的数据集，选择“主动学习”时，只支持识别和标注矩形框。
- 当系统中智能标注作业过多时，可能会出现排队的情况，导致作业一直处于“标注中”的状态。请您耐心等待，系统会按照顺序完成标注作业。

图 4-61 启动智能标注（图像分类）

图 4-62 启动智能标注（物体检测）

图 4-63 启动智能标注（预标注）

4. 完成参数设置后，单击“提交”，即可启动智能标注。
5. 在数据集列表中，单击数据集名称进入“数据集概览”页。
6. 在“数据集概览页”，单击右上角“开始标注”，进入数据集详情页。
7. 在数据集详情页，单击“待确认”页签，即可查看智能标注进度。
   您也可以在该页签，“启动智能标注”或者查看“智能标注历史”。

图 4-64 标注进度

8. 智能标注完成后，“待确认”页面将呈现所有标注后的图片列表。
   - 图像分类数据集

     在“待确认”页面查看标签是否准确，勾选标注准确的图片，然后单击“确认”完成智能标注结果的确认。确认完成后的图片将被归类至“已标注”页面下。

     针对标为“难例”的图片，您可以根据实际情况判断，手工修正标签。详细操作及示例请参见**•针对“图像分类”数据集**。
   - 物体检测数据集

     在“待确认”页面，单击图片查看标注详情，查看标签及目标框是否准确，针对标注准确的图片单击“确认标注”完成智能标注结果的确认。确认完成后的图片将被归类至“已标注”页面下。

     针对标为“难例”的图片，您可以根据实际情况判断，手工修正标签或目标框。详细操作及示例请参见**•针对“物体检测”数据集**。

4.11 难例确认

在数据量很大的标注任务中，标注初期由于已标注图片不足，智能标注的结果无法直接用于训练。若对所有的未标注数据一一进行调整确认仍然需要较大的人力和时间成本。为了更快地完成标注任务，在对未标注数据进行智能标注的任务中，ModelArts嵌入了自动难例发现功能。该功能会对剩余未标注图片的标注优先级给出建议。因为标注优先级高的图片的智能标注结果未达到预期，所以称之为难例。

ModelArts平台提供的自动难例发现功能，在智能标注以及数据采集筛选过程中，将自动标注出难例，建议对难例数据进一步确认标注，然后将其加入训练数据集中，使用此数据集训练模型，可得到精度更高的模型。首先，针对智能标注和采集筛选任务，难例的发现操作是系统自动执行的，无需人工介入，仅需针对标注后的数据进行确认和修改即可，提升数据管理和标注效率。其次，您可以基于难例的情况，补充类似数据，提升数据集的丰富性，进一步提升模型训练的精度。在数据集管理中，对难例的管理有如下场景。

- **智能标注后，确认难例**
- **将数据集中的数据标注为难例**

说明

目前只有“图像分类”和“物体检测”类型的数据集支持难例发现功能。

智能标注后，确认难例

“智能标注”任务执行过程中，ModelArts将自动识别难例，并完成标注。当智能标注结束后，难例标注结果将呈现在“待确认”页签，建议您对难例数据进行人工修正，然后确认标注。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，选择“物体检测”或“图像分类”类型的数据集，单击数据集名称进入“数据集概览页”。
3. 在“数据集概览页”，单击右上角“开始标注”，进入数据集详情页。
4. 在数据集详情页，单击“待确认”页签，查看并确认难例。

说明

只有当智能标注任务完成后，待确认页签才会显示标注数据。否则，此页签内容为空。智能标注操作请参见**智能标注**。

– 针对“物体检测”数据集

在“待确认”页签中，单击图片展开标注详情，查看图片数据的标注情况，如标签是否准确、目标框位置添加是否准确。如果智能标注结果不准确，建议手工调整标签或目标框，然后单击“确认标注”。完成确认后，重新标注的数据将呈现在“已标注”页签下。

如**图4-65**所示的难例，dog标签的目标框位置不准确，使用标注框重新标注，如图中的“漏检”目标框，然后需要将原先标注错误的目标框删除，即“误检”标签框。手工调整后，单击“确认标注”完成难例确认。

图 4-65 物体检测的难例确认

– 针对“图像分类”数据集

在“待确认”页签中，查看标注难例的图片，其添加的标签是否准确。勾选标注不准确的图片，删除错误标签，然后在右侧“标签名”处添加准确标签。单击“确认”，勾选的图片及其标注情况，将呈现在“已标注”页签下。

如**图4-66**所示，选中的图片为标注错误图片，在右侧删除错误标签，然后在标签名处添加“狗”的标签，然后单击“确认”，完成难例确认。

图 4-66 图像分类的难例确认

将数据集中的数据标注为难例

针对数据集中，已标注或未标注数据，也可以将图片数据标注为难例。标注为难例的数据，对后续模型训练中，通过内置规则提升模型精度。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，选择“物体检测”或“图像分类”类型的数据集，单击数据集名称进入“数据集概览页”。
3. 在“数据集概览页”，单击右上角“开始标注”，进入数据集详情页。
4. 在数据集详情页，单击“已标注”、“未标注”或“全部”页签，勾选需标注为难例的图片，然后单击“难例批处理 > 确认为难例”。完成标注后，图片预览时，其右上角将显示为“难例”。

图 4-67 确认为难例

4.12 自动分组

为了提升智能标注算法精度，可以均衡标注多个类别，有助于提升智能标注算法精度。ModelArts内置了分组算法，您可以针对您选中的数据，执行自动分组，提升您的数据标注效率。

自动分组可以理解为数据标注的预处理，先使用聚类算法对未标注图片进行聚类，再根据聚类结果进行处理，可以分组打标或者清洗图片。

例如，用户通过搜索引擎搜索XX，将相关图片下载并上传到数据集，然后再使用自动分组，可以将XX图片分类，比如论文、宣传海报、确认为XX的图片、其他。用户可以根据分组结果，快速剔除掉不想要的，或者将某一类直接全选后添加标签。

说明

目前只有“图像分类”、“物体检测”和“图像分割”类型的数据集支持自动分组功能。

启动自动分组任务

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 在数据集列表中，选择“物体检测”或“图像分类”类型的数据集，单击数据集名称进入“数据集概览页”。
3. 在“数据集概览页”，单击右上角“开始标注”，进入数据集详情页。
4. 在数据集详情页的“全部”页签中，单击“自动分组 > 启动任务”。

   说明

   只能在“全部”页签下启动自动分组任务或查看任务历史。

5. 在弹出的“自动分组”对话框中，填写参数信息，然后单击“确定”。
   - “分组数”：填写2~200之间的整数，指将图片分为多少组。
   - “结果处理方式”：“更新属性到当前样本中”，或者“保存到对象存储服务（OBS）”。
   - “属性名称”：当选择“更新属性到当前样本中”时，需输入一个属性名称。
   - “结果存储目录”：当选择“保存到对象存储服务（OBS）”时，需指定一个用于存储的OBS路径。
   - “高级特征选项”：启用此功能后，可选择“清晰度”、“亮度”、“图像色彩”等维度为自动分组功能增加选项，使得分组着重于图片亮度、色彩和清晰度等特征进行分组。支持多选。

图 4-68 自动分组

6. 启动任务提交成功后，界面右上角显示此任务的进度。等待任务执行完成后，您可以查看自动分组任务的历史记录，了解任务状态。

查看自动分组结果

在数据集详情页面的“全部”页签中，展开“筛选条件”，将“样本属性”设置为自动分组任务中的“属性名称”，并通过设置样本属性值，筛选出分组结果。

图 4-69 查看自动分组结果

查看自动分组的历史任务

在数据集详情页面的“全部”页签中，单击“自动分组 > 任务历史”。在弹出的“任务历史”对话框中，展示当前数据集之前执行的自动分组任务的基本信息。

图 4-70 自动分组任务历史

4.13 数据特征

基于图片或目标框对图片的各项特征，如模糊度、亮度进行分析，并绘制可视化曲线，帮助处理数据集。

您还可以选择数据集的多个版本，查看其可视化曲线，进行对比分析。

背景信息

- 只有“物体检测”和“图像分类”的数据集支持数据特征分析。
- 只有发布后的数据集支持数据特征分析。发布后的Default格式数据集版本支持数据特征分析。
- 数据特征分析的数据范围，不同类型的数据集，选取范围不同：
  - 在“物体检测”的数据集中，当已标注样本数为0时，发布版本后，数据特征页签版本置灰不可选，无法显示数据特征。有标注后，发布版本，显示已标注的图片的数据特征。
  - 在“图像分类”的数据集中，当已标注样本数为0时，发布版本后，数据特征页签版本置灰不可选，无法显示数据特征。有标注后，发布版本，显示全部的图片的数据特征。
- 数据集中的图片数量要达到一定量级才会具有意义，一般来说，需要有大约1000+的图片。
- “图像分类”支持分析指标有：“分辨率”、“图片高宽比”、“图片亮度”、“图片饱和度”、“清晰度”和“图像色彩的丰富程度”。“物体检测”支持所有的分析指标。目前ModelArts支持的所有分析指标请参见**支持分析指标及其说明**。

数据特征分析

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理>数据集”，进入“数据集”管理页面。
2. 选择对应的数据集，单击操作列的“数据特征”，进入数据集概览页的数据特征页面。

   您也可以在单击数据集名称进入数据集概览页后，单击“数据特征”页签进入。
3. 由于发布后的数据集不会默认启动数据特征分析，针对数据集的各个版本，需手动启动特征分析任务。在数据特征页签下，单击“特征分析”。

   图 4-71 选择特征分析

4. 在弹出的对话框中配置需要进行特征分析的数据集版本，然后单击“确定”启动分析。

   “版本选择”，即选择当前数据集的已发布版本。

   图 4-72 启动数据特征分析任务

5. 数据特征分析任务启动后，需执行一段时间，根据数据量不同等待时间不同，请耐心等待。当您选择分析的版本出现在“版本选择”列表下，且可勾选时，即表示分析已完成。

图 4-73 可选择已执行特征分析的版本

6. 查看数据特征分析结果。

“版本选择”：在右侧下拉框中选择进行对比的版本。也可以只选择一个版本。

“类型”：选择需要分析的类型。支持“all”、“train”、“eval”和“inference”。

“数据特征指标”：在右侧下拉框中勾选需要展示的指标。详细指标说明请参见**支持分析指标及其说明**。

选择完成后，页面将自动呈现您选择对应版本及其指标数据，如**图4-74**所示，您可以根据呈现的图表了解数据分布情况，帮助您更好的处理您的数据。

图 4-74 数据特征分析

7. 查看分析任务的历史记录。

在数据特征分析后，您可以在“数据特征”页签下，单击右侧“任务历史”，可在弹出对话框中查看历史分析任务及其状态。

图 4-75 任务历史

任务历史

数据集版本	任务ID	创建时间	运行时间（h...	状态
V016	rzGaEY2lQDZ...	2020/06/01 1...	00:00:19	成功
V015	fOPPZbgwdY...	2020/06/01 1...	00:00:17	成功
V014	hfRjPLx03w3...	2020/06/01 1...	00:00:15	成功
V013	xwSatuRsHLu...	2020/06/01 1...	00:00:16	成功
V012	ARQfHizkrGR...	2020/06/01 1...	00:00:13	成功
V011	fsDmMsPrtv...	2020/06/01 1...	00:00:18	成功
V010	uocIdNmR2B...	2020/06/01 1...	00:00:13	成功
V009	EFSJPawWlzu...	2020/06/01 1...	00:00:19	成功
V008	QBBflfFszv5...	2020/06/01 1...	00:00:23	成功
V006	LvNT9UKBx8...	2020/06/01 1...	00:00:27	成功

10 总条数: 10 1

支持分析指标及其说明

表 4-31 分析指标列表

名称	说明	分析说明
分辨率 Resolution	图像分辨率。此处使用面积值作为统计值。	通过指标分析结果查看是否有偏移点。如果存在偏移点，可以对偏移点做resize操作或直接删除。
图片高宽比 Aspect Ratio	图像高宽比，即图片的高度/图片的宽度。	一般呈正态分布，一般用于比较训练集和真实场景数据集的差异。
图片亮度 Brightness	图片亮度，值越大代表观感上亮度越高。	一般呈正态分布，可根据分布中心判断数据集整体偏亮还是偏暗。可根据使用场景调整，比如使用场景是夜晚，图片整体应该偏暗。

名称	说明	分析说明
图片饱和度 Saturation	图片的色彩饱和度，值越大表示图片整体色彩越容易分辨。	一般呈正态分布，一般用于比较训练集和真实场景数据集的差异。
清晰度 Clarity	图片清晰程度，使用拉普拉斯算子计算所得，值越大代表边缘越清晰，图片整体越清晰。	可根据使用场景判断清晰度是否满足需要。比如使用场景的数据采集来自高清摄像头，那么清晰度对应的需要高一些。可通过对数据集做锐化或模糊操作，添加噪声对清晰度做调整。
图像色彩的丰富程度 Colorfulness	横坐标：图像的色彩丰富程度，值越大代表色彩越丰富。 纵坐标：图片数量。	是观感上的色彩丰富程度，一般用于比较训练集和真实场景数据集的差异。
按单张图片中框的个数统计图片分布 Bounding Box Quantity	横坐标：单张图片中框的个数。 纵坐标：图片数量。	对模型而言一张图片的框个数越多越难检测，需要越多的这种数据用作训练。
按单张图片中框的面积标准差统计图片分布 Standard Deviation of Bounding Boxes Per Image	横坐标：单张图片中框的标准差。单张图片只有一个框时，标准差为0。标准差的值越大，表示图片中框大小不一程度越高。 纵坐标：图片数量。	对模型而言一张图中框如果比较多且大小不一，是比较难检测的，可以根据场景添加数据用作训练，或者实际使用没有这种场景可直接删除。
按高宽比统计框数量的分布 Aspect Ratio of Bounding Boxes	横坐标：目标框的高宽比。 纵坐标：框数量（统计所有图片中的框）。	一般呈泊松分布，但与使用场景强相关。多用于比较训练集和验证集的差异，如训练集都是长方形框的情况下，验证集如果是接近正方形的框会有比较大影响。
按面积占比统计框数量的分布 Area Ratio of Bounding Boxes	横坐标：目标框的面积占比，即目标框的面积占整个图片面积的比例，越大表示物体在图片中的占比越大。 纵坐标：框数量（统计所有图片中的框）。	主要判断模型中使用的anchor的分布，如果目标框普遍较大，anchor就可以选择较大。
按边缘化程度统计框数量的分布 Marginalization Value of Bounding Boxes	横坐标：边缘化程度，即目标框中心点距离图片中心点的距离占图片总距离的比值，值越大表示物体越靠近边缘。 纵坐标：框数量（统计所有图片中的框）。	一般呈正态分布。用于判断物体是否处于图片边缘，有一些只露出一部分的边缘物体，可根据需要添加数据集或不标注。

名称	说明	分析说明
按堆叠度统计框数量的分布 Overlap Score of Bounding Boxes	横坐标：堆叠度，单个框被其他的框重叠的部分，取值范围为0～1，值越大表示被其他框覆盖的越多。 纵坐标：框数量（统计所有图片中的框）。	主要用于判断待检测物体的堆叠程度，堆叠物体一般对于检测难度较高，可根据实际使用需要添加数据集或不标注部分物体。
按亮度统计框数量的分布 Brightness of Bounding Boxes	横坐标：目标框的图片亮度，值越大表示越亮。 纵坐标：框数量（统计所有图片中的框）。	一般呈正态分布。主要用于判断待检测物体的亮度。在一些特殊场景中只有物体的部分亮度较暗，可以看是否满足要求。
按清晰度统计框数量的分布 Clarity of Bounding Boxes	横坐标：目标框的清晰度，值越大表示越清晰。 纵坐标：框数量（统计所有图片中的框）。	主要用于判断待检测物体是否存在模糊的情况。比如运动中的物体在采集中可能变得模糊，需要重新采集。

4.14 团队标注

4.14.1 团队标注简介

数据标注任务中，一般由一个人完成，但是针对数据集较大时，需要多人协助完成。ModelArts提供了团队标注功能，可以由多人组成一个标注团队，针对同一个数据集进行标注管理。

说明

团队标注功能当前仅支持“图像分类”、“物体检测”、“文本分类”、“命名实体”、“文本三元组”、“语音分割”类型的数据集。

如何启用团队标注

- 创建数据集时，打开“启用团队标注”开关，同时指定一个标注团队，或者指定标注管理员。

图 4-76 创建数据集时启用

- 对于已创建，且未启用团队标注的数据集。可通过创建一个团队标注任务，直接启用团队标注功能。创建团队标注的操作详情请参见**创建团队标注任务**。

图 4-77 在数据集列表中创建团队标注任务

图 4-78 在数据集概览页中创建团队标注任务

图 4-79 在数据集详情页创建团队标注任务

说明

- 当标注人员第一次收到标注邮件时，会告知标注的地址(labelConsole)以及默认初始密码。
- 只有当创建团队标注任务时，标注人员才会收到邮件。创建标注团队及添加标注团队的成员并不会发送邮件。此外，当所有样本都是已标注状态时，创建团队标注任务也不会收到邮件。

团队标注相关操作

- 管理团队
- 管理成员
- 管理团队标注任务

4.14.2 管理团队

团队标注功能是以团队为单位进行管理，数据集启用团队标注功能时，必须指定一个团队。一个团队可以添加多个成员。

背景说明

- 一个帐号最多可添加10个团队。
- 如果数据集需要启用团队标注功能，当前帐号至少拥有一个团队。如果没有，请执行添加团队操作添加。

添加团队

1. 在ModelArts管理控制台左侧导航栏中，选择“数据管理>标注团队”，进入“标注团队”管理页面。
2. 在“标注团队”管理页面，单击“添加团队”。
3. 在弹出的“添加团队”对话框中，填写团队“名称”和“描述”，然后单击“确定”。完成标注团队的添加。

图 4-80 添加团队

团队添加完成后，“标注团队”管理页面呈现新添加的团队，在页面右侧区域，可以查看团队详情。新添加的团队，其成员列表为空，请参考添加成员操作，为您的团队添加成员。

删除团队

当已有的团队不再使用，您可以执行删除操作。

在“标注团队”管理页面中，选中需删除的团队，然后单击“删除”。在弹出的对话框中，确认信息无误后，单击“确定”完成团队删除。

图 4-81 删除团队

4.14.3 管理成员

新添加的团队，其成员列表为空。您需要根据实际情况添加即将参与标注任务的成员信息。

一个团队最多支持添加100个成员，当超过100时，建议分为多个团队进行管理。

添加成员

1. 在ModelArts管理控制台左侧导航栏中，选择“数据管理>标注团队”，进入“标注团队”管理页面。

2. 在“标注团队”管理页面，从左侧团队列表中选择一个团队，单击团队，其右侧区域将呈现“团队详情”。
3. 在“团队详情”区域，单击“添加成员”。
4. 在弹出的“添加成员”对话框中，填写成员的“邮箱”、“描述”、指定“角色”，然后单击“确定”。

   邮箱作为团队管理中的唯一标识，不同成员不能使用同一个邮箱。您填写的邮箱地址将被记录并保存在ModelArts中，仅用于ModelArts团队标注功能，当成员删除后，其填写的邮箱信息也将被一并删除。

   其中，“角色”支持“Labeler”、“Reviewer”和“Team Manager”，“Team Manager”只能设置为一个人。

   图 4-82 添加成员

   成员添加完成后，团队详情区域中将呈现此成员的信息。

修改成员信息

团队中的成员，当其信息发生变化时，可以编辑其基本情况。

1. 在“团队详情”区域，选择需修改的成员。
2. 在成员所在行的“操作”列，单击“修改”。在弹出的对话框中，修改其“描述”或“角色”。

   成员的“邮箱”无法修改，如果需要修改邮箱地址，建议先删除此成员，然后再基于新的邮箱地址添加新成员。

   “角色”支持“Labeler”、“Reviewer”和“Team Manager”，“Team Manager”只能设置为一个人。

删除成员

- **删除单个成员**

  在“团队详情”区域，选择需要删除的成员， 单击“操作”列的“删除”。在弹出的对话框中，确认信息无误后，单击“确定”完成删除操作。

- **批量删除**

  在“团队详情”区域，勾选需删除的成员，然后单击“删除”。在弹出的对话框中，确认信息无误后，单击“确定”完成多个成员的删除操作。

图 4-83 批量删除

4.14.4 管理团队标注任务

针对启用团队标注功能的数据集，支持创建团队标注任务，将标注任务指派给不同的团队，由多人完成标注任务。同时，在成员进行数据标注过程中，支持发起验收、继续验收以及查看验收报告等功能。

创建团队标注任务

如果您在创建数据集时，即启用团队标注，且指派了某一团队负责标注，系统将默认基于此团队创建一个标注任务。您可以在数据集创建后，在数据集的“标注任务进展”页面查看此任务。

您还可以重新创建一个团队标注任务，指派给同一团队的不同成员，或者指派给其他标注团队。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“数据管理 >数据集”，打开数据集列表。
2. 在数据集列表中，选择支持团队标注的数据集，单击数据集名称进入数据集概览页。
3. 在数据集概览页中，单击“标注任务进展”页签，可查看此数据集已有的标注任务。单击右上角的“创建团队标注任务”开始创建新任务。

图 4-84 标注任务

4. 在弹出的“创建团队标注任务”对话框中，填写相关参数，然后单击“确定”，完成任务创建。
   - “名称”：设置此任务的名称。
   - “类型”：设置任务类型，支持“指定标注团队”或“指定标注管理员”。
   - “选择标注团队”：任务类型设置为“指定标注团队”，需在此参数中指定一个团队，同时勾选此团队中某几个成员负责标注。下拉框中将罗列当前帐号下创建的标注团队及其成员，团队管理的操作指导请参见团队标注简介。

- “选择标注接口人”：任务类型设置为“指定标注管理员”，需在所有团队的“Team Manager”中选择一人作为管理员。
- “标签集”：展示当前数据集已有的标签及标签属性。在“标签集”下方也可以设置“自动将新增图片同步给标注团队”或“团队标注的图片自动加载智能标注结果”。

说明

团队标注加载智能标注结果的处理步骤：

- 如果类型选择"指定标注团队"，需要先创建团队标注任务，然后执行智能标注任务。
- 如果类型选择"指定标注管理员"，需要登录labelconsole先执行"分配任务"，然后执行智能标注任务。

图 4-85 创建团队标注任务

任务创建完成后，您可以在“标注任务进展”页签下看到新建的任务。

进入标注（团队成员）

在标注任务创建后，被分配任务的团队成员将收到一封通知邮件，标题为“您有新的标注任务待查收”。

在邮件详情中，单击标注任务链接进入数据标注平台，使用您的“邮箱地址”及“初始密码”登录标注平台系统。首次登录时，需根据提示修改密码。关于标注平台的详细说明，请参见什么是数据标注平台。

说明

首次登录时，请务必修改密码。如果未修改，将导致下一次无法正常登录。

登录数据标注平台后，可查看到分配的标注任务，单击任务名称，可进入标注页面。不同类型的数据集，标注方式不同，详细请参见：

- 图像分类
- 物体检测
- 文本分类
- 命名实体
- 文本三元组

在标注平台中，每个成员可查看“未标注”、“待修正”、“已驳回”、“待审核”、“审核通过”、“验收通过”的图片信息。请及时关注管理员驳回以及待修正的图片。

当团队标注任务中，分配了Reviewer角色，则需要对标注结果进行审核，审核完成后，再提交给管理员验收。

图 4-86 成员标注平台

任务验收（管理员）

- **发起验收**

  当团队的成员已完成数据标注，数据集的创建者可发起验收，对标注结果进行抽验。只有当标注成员存在标注完成的数据时，才可以发起验收，否则发起验收按钮为灰色。

  a. 在“标注任务进展”页签中，针对需发起验收的任务，单击“发起验收”。
  b. 在弹出的对话框中，设置“抽样策略”，可设置为“按百分比”，也可以设置为“按数量”。设置好参数值后，单击“确定”启动验收。

     “按百分比”：按待验收图片总数的一定比例进行抽样验收。

     “按数量”：按一定数量进行抽样验收。

图 4-87 发起验收

c. 验收启动后，界面将展示实时验收报告，您可以在右侧选择“验收结果”（“通过”或“不通过”）。

当选择验收结果为“通过”时，需设置“验收评分”（分“A”、“B”、“C”、“D”四个选项，“A”表示最高分），如图4-89所示。当选择验收结果为“不通过”时，可以在文本框中写明驳回原因，如图4-90所示。

图 4-88 查看实时验收报告

图 4-89 设置验收结果为“通过”

图 4-90 设置验收结果为“不通过”

- **继续验收**

  针对未完成验收的任务，可以继续验收。针对未发起过验收流程的任务，不支持“继续验收”，按钮为灰色。

  在“标注任务进展”页签中，针对需继续验收的任务，单击“继续验收”。系统直接进入“实时验收报告”页面，您可以继续验收未验收的图片，设置其“验收结果”。

- **完成验收**

  在完成验收窗口，您可以查看本数据集的验收情况，如抽样文件数等，同时设置如下参数，然后进行验收。只有完成验收，标注信息才会同步到数据集的已标注页面中。

  一旦标注数据完成验收，团队成员无法再修改标注信息，只有数据集创建者可修改。

表 4-32 完成验收的参数设置

参数	说明
对已标注数据修改	• 不覆盖：针对同一个数据，不使用当前团队标注的结果覆盖已有数据。 • 覆盖：针对同一个数据，使用当前团队标注的结果覆盖已有数据。覆盖后无法恢复，请谨慎操作。
通过范围	• 全部：当前团队标注完成的所有数据。包含验收通过、未验收和验收不通过的。即本数据集的所有抽样文件数。 • 全部不通过：当前团队标注完成的所有数据不通过验收，即将所有标注数据驳回给标注人员。全部数据指验收通过、未验收和验收不通过的所有数据，即本数据集的所有抽样文件数。 • 验收通过和未验收的数据：针对抽样文件中验收通过和未验收的数据，通过验收。验收不通过的数据将驳回给标注人员。 • 验收通过的数据：针对抽样文件中验收通过的数据，通过验收。未验收和验收不通过的数据将驳回给标注人员。

图 4-91 完成验收

查看验收报告

针对进行中或已完成的标注任务，都可以查看其验收报告。在“标注任务进展”页签中，单击“验收报告”，即可在弹出的“验收报告”对话框中查看详情。

图 4-92 查看验收报告

删除标注任务

针对不再使用的标注任务，您可以在“标注任务进展”页签下，单击任务所在行的删除。任务删除后，未验收的标注详情将丢失，请谨慎操作。但是数据集中的原始数据以及完成验收的标注数据仍然存储在对应的OBS桶中。

4.15 数据处理

4.15.1 数据处理简介

ModelArts平台提供的数据处理功能，基本目的是从大量的、杂乱无章的、难以理解的数据中抽取或者生成对某些特定的人们来说是有价值、有意义的数据。当数据采集和接入之后，数据一般是不能直接满足训练要求的。为了保障数据质量，以免对后续操作（如数据标注、模型训练等）带来负面影响，开发过程通常需要进行数据处理。常见的数据处理类型有以下四种：

- 数据校验：通常数据采集后需要进行校验，保证数据合法。

  数据校验是指对数据可用性的基本判断和验证的过程。通常，我们采集的数据或多或少都会有很多格式问题，无法被进一步处理。以图像识别为例，用户经常会从网上找一些图片用于训练，但是其质量难以保证，有可能图片的名字、路径、后缀名都不满足训练算法的要求；图片也可能有部分损坏，造成无法解码、无法被算法处理的情况。因此，数据校验非常重要，可以帮助人工智能开发者提前发现数据问题，有效防止数据噪声造成的算法精度下降或者训练失败问题。

- 数据清洗：数据清洗是指对数据进行去噪、纠错或补全的过程。

  数据清洗是在数据校验的基础上，对数据进行一致性检查，处理一些无效值。例如在深度学习领域，可以根据用户输入的正样本和负样本，对数据进行清洗，保留用户想要的类别，去除用户不想要的类别。

- 数据选择：数据选择一般是指从全量数据中选择数据子集的过程。

  数据可以通过相似度或者深度学习算法进行选择。数据选择可以避免人工采集图片过程中引入的重复图片、相似图片等问题；在一批输入旧模型的推理数据中，通过内置规则的数据选择可以进一步提升旧模型精度。

- 数据增强：

  数据扩增通过简单的数据扩增例如缩放、裁剪、变换、合成等操作直接或间接的方式增加数据量。

  图像生成应用相关深度学习模型，通过对原数据集进行学习，训练生成新的数据集的方式增加数据量。

4.15.2 创建数据处理任务

您可以创建一个数据处理任务，对已有的数据进行数据校验、数据清洗、数据选择或者数据增强操作。

前提条件

- 数据已准备完成：已经创建数据集或者已经将数据上传至OBS
- 确保您使用的OBS与ModelArts在同一区域

创建数据处理任务

1. 登录ModelArts管理控制台，在左侧的导航栏中选择“数据管理>数据处理”，进入“数据处理”页面。
2. 在“数据处理”页面，单击“创建”进入“创建数据处理”页面。
3. 在创建数据处理页面，填写相关算法参数。

a. 填写基本信息。基本信息包括“名称”、“版本”和“描述”。其中“版本”信息由系统自动生成，按“V0001”、“V0002”规则命名，用户无法修改。

您可以根据实际情况填写“名称”和“描述”信息。

图 4-93 创建数据处理基本信息

b. 设置场景类别。场景类别当前支持“图像分类”和“物体检测”。

c. 设置数据处理类型。数据处理类型支持“数据清洗”、“数据校验”、“数据选择”和“数据增强”。

针对不同的数据处理类型，您需要填写相应算子的设置参数，算子的详细参数参见**预置算子说明**。

图 4-94 设置场景类别和数据处理类型

d. 设置输入与输出。需根据实际数据情况选择“数据集”或“OBS目录”。设置为“数据集”时，需填写“数据集名称”和“数据集版本”；设置为“OBS目录”时，需填写正确的OBS路径。

图 4-95 输入输出设置-数据集

图 4-96 输入输出设置-OBS 目录

e. 确认参数填写无误后，单击“创建”，完成数据处理任务的创建。

4.15.3 管理和查看数据处理任务

删除数据处理任务

当已有的数据处理任务不再使用时，您可以删除数据处理任务。

处于“完成”、“失败”、“已停止”、“运行失败”、“部署中”状态的训练作业，您可以单击操作列的“删除”，删除对应的数据处理任务。

查看数据处理任务详情

1. 登录ModelArts管理控制台，在左侧的导航栏中选择“数据管理>数据处理”，进入“数据处理”页面。
2. 在数据处理列表中，单击数据处理任务名称，进入数据处理任务的版本管理页面。您可以在该页面进行数据处理任务的“修改”与“删除”。

图 4-97 数据处理版本管理页面

3. 您可以在版本管理页面，通过切换页签查看“配置信息”、“日志”和“结果展示”。

图 4-98 日志页面

图 4-99 结果展示页面

4.15.4 预置算子说明

4.15.4.1 数据校验

MetaValidation 算子概述

ModelArts的数据校验通过MetaValidation算子实现。当前ModelArts支持jpg、jpeg、bmp、png四种图片格式。物体检测场景支持xml标注格式，不支持“非矩形框”标注。针对您提供的数据集，MetaValidation算子支持对图片和xml文件进行数据校验：

表 4-33 图片类数据校验

异常情况	处理方案
图片本身损坏无法解码	过滤掉不能解码的图片
图片通道可能是1通道、2通道，不是常用的3通道	转换图片成RGB三通道
图片格式不在ModelArts支持的格式范围内	转换图片格式至jpg格式

异常情况	处理方案
图片后缀与实际格式不符，但格式在MA支持的格式内	后缀转换成与实际格式一致
图片后缀与实际格式不符，且格式不在MA支持的格式内	转换图片格式至jpg格式
图片分辨率过大	宽、高按指定大小同比例进行裁剪

表 4-34 标注类文件数据校验

异常情况	处理方案
xml结构残缺，无法解析	过滤xml文件
xml中没有标注“object”	过滤xml文件
xml中没有矩形框“bndbox”	过滤xml文件
某些标注“object”中没有矩形框“bndbox”	过滤标注“object”
图片经过裁剪后，xml文件中宽高不符	修改错误宽高参数为图片真实宽高
xml中没有“width”、“height”字段	根据图片真实宽高补全xml中的“width”、“height”字段和值
图片经过裁剪后，xml中矩形框“bndbox”大小不符	按图片裁剪比例缩放xml文件中“bnxbox”值
xml中矩形框“bndbox”宽或高值过小，显示为一条线	矩形框宽或高差值小于2，移除当前“object”
xml中矩形框“bndbox”最小值大于最大值	移除当前“object”
矩形框“bndbox”超出图片边界，且超出部分占框面积50%以上	移除当前“object”
矩形框“bndbox”超出图片边界，但超出部分小于框面积50%	矩形框“bndbox”拉回到图片边界

说明

数据校验过程不会改动原始数据，通过校验的图片或xml文件保存在指定的输出路径下。

参数说明

表 4-35 数据校验-MetaValidation 算子参数说明

参数名	是否必选	默认值	参数说明
image_max_width	否	-1	输入图片宽度最大值，若输入图片宽度超过设定值则按比例裁剪。单位为px。 默认值 -1 表示不做裁剪。
image_max_height	否	-1	输入图片长度最大值，若输入图片长度超过设定值则按比例裁剪。单位为px。 默认值 -1 表示不做裁剪。

输入要求

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在本任务中选择的场景类别一致。
- 选择“OBS目录”，存放结构又分两种情况，“仅包含图片”或“包含图片和标注信息”。
  - “仅包含图片”：当目录下全是图片时，支持jpg、jpeg、png、bmp格式，嵌套子目录的图片也将全部读入。
  - “包含图片和标注信息”：根据不同场景类型，结构不同。

    图像分类场景，其目录结构如下所示。如下目录结构，仅支持单标签场景。

    ```
    input_path/
      --label1/
          ----1.jpg
      --label2/
          ----2.jpg
      --../
    ```

    物体检测场景，其目录结构如下所示。支持jpg、jpeg、png、bmp格式的图片，xml为标准的PACAL VOC格式标注文件。

    ```
    input_path/
      --1.jpg
      --1.xml
      --2.jpg
      --2.xml
      ...
    ```

输出说明

- **图像分类**

  输出数据的目录结构如下所示。

  ```
  output_path/
    --Data/
        ----class1/  # 若输入数据有标注信息会一并输出，class1为标注类别
            ------1.jpg
            ------2_checked.jpg
        ----class2/
            ------3.jpg
  ```

```
          ------4_checked.jpg
      ----5_checked.jpg
  --output.manifest
```

其中manifest文件内容示例如下所示。会给每一条数据加上一个校验属性"property":{"@modelarts:data_checked":true}

```
{
  "id": "xss",
  "source": "obs://hard_example_path/Data/fc8e2688015d4a1784dcbda44d840307_14_checked.jpg",
  "property": {
    "@modelarts:data_checked": true
  },
  "usage": "train",
  "annotation": [
    {
      "name": "Cat",
      "type": "modelarts/image_classification"
    }
  ]
}
```

- **物体检测**

  在输出目录下，文件结构如下所示。

```
output_path/
  --Data/
      ----1_checked.jpg
      ----1_checked.xml  # 若输入数据在校验过程中经过了转换，文件名会加上'_checked'
      ----2.jpg          # 若输入数据未经过转换，则以原来的名字保存
      ----2.xml
  --output.manifest
```

其中manifest文件内容示例如下所示。会给每一条数据加上一个校验属性"property":{"@modelarts:data_checked":true}

```
{
  "source": "obs://hard_example_path/Data/be462ea9c5abc09f_checked.jpg",
  "property": {
    "@modelarts:data_checked": true
  },
  "annotation": [
    {
      "annotation-loc": "obs://hard_example_path/Data/be462ea9c5abc09f_checked.xml",
      "type": "modelarts/object_detection",
      "annotation-format": "PASCAL VOC",
      "annotated-by": "modelarts/hard_example_algo"
    }
  ]
}
```

4.15.4.2 数据清洗

PCC 算子概述

ModelArts的数据清洗通过PCC算子实现。图像分类或者物体检测的数据集中可能存在非所需类别的图像，需要将这些图像去除掉，以免对标注、模型训练造成干扰。

图 4-100 PCC 算子效果

参数说明

表 4-36 数据清洗-PCC 算子参数说明

参数名	是否必选	默认值	参数说明
prototype_sample_path	是	None	数据清洗正样例目录。目录应存放正样例图片文件，算法将这些图片为正样例，对输入中的数据进行过滤，即保留与“prototype_sample_path”目录下图片相似度高的数据。 请输入一个真实存在的OBS目录，且目录下已包含提供的正样例图片，且以obs://开头。如：*obs://obs_bucket_name/folder_name*
criticism_sample_path	否	None	数据清洗负样例目录。目录应存放负样例图片文件，算法将这些图片为负样例，对算法输入中的数据进行过滤， 即保留与“criticism_sample_path”目录下图片相似度差距较大的数据。 建议该参数和“prototype_sample_path”配合使用，可以提高数据清洗的准确性。 请输入一个真实存在的OBS目录，且以obs://开头。如：*obs://obs_bucket_name/folder_name*
n_clusters	否	auto	数据样本的种类数，默认值auto。您可以输入小于样本总数的整数或auto。auto表示使用正样本目录的图片个数作为数据样本的种类数。
simlarity_threshold	否	0.9	相似度阈值。两张图片相似程度超过阈值时，判定为相似图片，反之按非相似图片处理。输入取值范围为0~1。
embedding_distance	否	0.2	样本特征间距。两张图片样本特征间距小于设定值，判定为相似图片，反之按非相似图片处理。输入取值范围为0~1。

参数名	是否必选	默认值	参数说明
do_validation	否	True	是否做数据校验，可填True或者False。表示数据清洗前需要做数据校验，否则只做数据清洗。

输入要求

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在本任务中选择的场景类别一致。
- 选择“OBS目录”，存放结构又分两种情况，“仅包含图片”或“包含图片和标注信息”。
  - “仅包含图片”：当目录下全是图片时，支持jpg、jpeg、png、bmp格式，嵌套子目录的图片也将全部读入。
  - “包含图片和标注信息”：根据不同场景类型，结构不同。

    图像分类场景，其目录结构如下所示。如下目录结构，仅支持单标签场景。

    ```
    input_path/
        --label1/
            ----1.jpg
        --label2/
            ----2.jpg
        --../
    ```

    物体检测场景，其目录结构如下所示。支持jpg、jpeg、png、bmp格式的图片，xml为标准的PACAL VOC格式标注文件。

    ```
    input_path/
        --1.jpg
        --1.xml
        --2.jpg
        --2.xml
        ...
    ```

输出说明

- **图像分类**

  输出数据的目录结构如下所示。

  ```
  output_path/
      --Data/
          ----class1/  # 若输入数据有标注信息会一并输出，class1为标注类别
              ------1.jpg
          ----class2/
              ------2.jpg
          ----3.jpg
      --output.manifest
  ```

  其中manifest文件内容示例如下所示。

  ```
  {
      "id": "xss",
      "source": "obs://home/fc8e2688015d4a1784dcbda44d840307_14.jpg",
      "usage": "train",
      "annotation": [
          {
              "name": "Cat",
  ```

```
            "type": "modelarts/image_classification"
        }
    ]
}
```

- **物体检测**

  输出数据的目录结构如下所示。

```
output_path/
    --Data/
        ----1.jpg
        ----1.xml  # 若输入数据有标注信息会一并输出，xml为标注文件
        ----2.jpg
        ----3.jpg
    --output.manifest
```

  其中manifest文件内容示例如下所示。

```
{
    "source":"obs://fake/be462ea9c5abc09f.jpg",
    "annotation":[
        {
            "annotation-loc":"obs://fake/be462ea9c5abc09f.xml",
            "type":"modelarts/object_detection",
            "annotation-format":"PASCAL VOC",
            "annotated-by":"modelarts/hard_example_algo"
        }
    ]
}
```

4.15.4.3 数据选择

数据选择算子概述

ModelArts提供以下数据选择算子：

- SimDeduplication：可以依据用户设置的相似程度阈值完成图像去重处理。图像去重是图像数据处理常见的数据处理方法。图像重复指图像内容完全一样，或者有少量的尺度、位移、色彩、亮度变化，或者是添加了少量其他内容等。

图 4-101 SimDeduplication 效果图

表 4-37 高级参数说明

参数名	是否必选	默认值	参数说明
simlarity_threshold	否	0.9	相似程度阈值，两张图片间的相似度大于阈值时，其中一张会作为重复图片被过滤掉。取值范围为0~1。

参数名	是否必选	默认值	参数说明
do_validation	否	True	是否做数据校验，可填True或者False。表示数据去重前需要做数据校验，否则只做数据去重。

- RRD：可以依据用户设置的比例去除差异最大的数据。

图 4-102 RRD 效果图

表 4-38 高级参数说明

参数名	是否必选	默认值	参数说明
sample_ratio	否	0.9	数据留下的百分比。取值范围为0~1。例如0.9表示保留百分之90的原数据。
n_clusters	auto	auto	数据样本的种类数，默认为auto，即按照目录中图片个数取类别总数，可指定具体类别数，如 4
do_validation	否	True	是否做数据校验，可填True或者False。表示数据去冗余前需要做数据校验，否则只做数据去重。

输入要求

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在本任务中选择的场景类别一致。
- 选择“OBS目录”，存放结构又分两种情况，“仅包含图片”或“包含图片和标注信息”。
  - “仅包含图片”：当目录下全是图片时，支持jpg、jpeg、png、bmp格式，嵌套子目录的图片也将全部读入。
  - “包含图片和标注信息”：根据不同数据类型，结构不同。

    图像分类，其目录结构如下所示。如下目录结构，仅支持单标签场景。

```
input_path/
    --label1/
        ----1.jpg
    --label2/
        ----2.jpg
    --../
```

物体检测，其目录结构如下所示。支持jpg、jpeg、png、bmp格式的图片，xml为标准的PACAL VOC格式标注文件。

```
input_path/
    --1.jpg
    --1.xml
    --2.jpg
    --2.xml
    ...
```

输出说明

- **图像分类**

  输出数据的目录结构如下所示。

```
output_path/
    --Data/
        ----class1/  # 若输入数据有标注信息会一并输出，class1为标注类别
            ------1.jpg
        ----class2/
            ------2.jpg
            ------3.jpg
    --output.manifest
```

  其中manifest文件内容示例如下所示。

```
{
    "id": "xss",
    "source": "obs://home/fc8e2688015d4a1784dcbda44d840307_14.jpg",
    "usage": "train",
    "annotation": [
        {
            "name": "Cat",
            "type": "modelarts/image_classification"
        }
    ]
}
```

- **物体检测**

  输出数据的目录结构如下所示。

```
output_path/
    --Data/
        ----1.jpg
        ----1.xml  # 若输入数据有标注信息会一并输出，xml为标注文件
        ----2.jpg
        ----3.jpg
    --output.manifest
```

  其中manifest文件内容示例如下所示。

```
{
    "source":"obs://fake/be462ea9c5abc09f.jpg",
    "annotation":[
        {
            "annotation-loc":"obs://fake/be462ea9c5abc09f.xml",
            "type":"modelarts/object_detection",
            "annotation-format":"PASCAL VOC",
            "annotated-by":"modelarts/hard_example_algo"
            }
    ]
}
```

4.15.4.4 数据选择（难例）

算法概述

在实际业务场景中，模型维护是一个长期的过程，比如说按照每周、每月进行数据重训练，或者累计数据至一定量时进行定期的重训练。如果将全量的数据用于重训练，需要耗费较大的标注人力和训练耗时。为了提升模型维护效率，可以采用基于难例数据的重训练 。

难例筛选算法对全量数据进行分析并筛选，仅输出全量数据中少部分对于模型维护有价值的数据。基于筛选后的数据进行重训练，可以有效减少标注人力和训练耗时。

难例筛选算法中融合了多种方法，要达到最佳效果，需要根据实际数据选择部分或全部方法，并调整其权重。

参数说明

参数名	是否必选	默认值	参数说明
source_service	Y	inference	难例任务的前置数据来源，目前仅支持inference。此参数不可修改。
filter_func	Y	comprehensive_mining	难例筛选算法设置为“comprehensive_mining”。此参数不可修改。
checkpoint_path	Y	/home/work/user-job-dir/data_filter/resnet_v1_50	用于特征提取的模型目录，当前仅支持基于Imagenet预训练的resnet_v1_50模型，此参数目前暂时不可修改。
model_serving_url	N	None	推理模型路径。得出推理结果的模型文件路径, 即训练作业的输出路径。该模型会用于aug_consistent_mining算法中的数据增强再推理。 请输入一个真实存在的OBS目录，例如：obs://obs_bucket_name/folder_name/
train_data_path	N	None	训练数据集，model_serving_url模型使用的训练数据，需输入数据集版本生成的manifest。 请输入一个真实存在的OBS目录，例如：obs://obs_bucket_name/folder_name/v001.manifest

参数名	是否必选	默认值	参数说明
comprehensive_algo_config	N	clustering_mining: 0.2020+aug_consistent_mining: 0.4265+feature_distribution_mining: 0.0451+sequential_mining: 0.425+image_similarity_mining: 0.0949+predict_score_mining: 0.3900+anomaly_detection_mining:0.2020	使用的算法及其权重，默认使用系统实验后效果最佳参数，也可以根据不同的数据自行配置。 例如：predict_score_mining: 0.3900+anomaly_detection_mining: 0.2020
algo_hard_threshold	N	0.1	筛选系数的阈值设置。取值范围0~1。 由于过高阈值可能导致输出结果为0，建议合理填写。
aug_op_config	N	crop:0.1+fliplr: 0.1+gaussianblur:0.1	aug_consistent_mining算法中用到的增强手段，支持crop、fliplr、gaussianblur、flipud、scale、translate、shear、superpixels、sharpen、add、invert。

参数名	是否必选	默认值	参数说明
feature_op_config	N	image_aspect_ratio: 0.5+image_brightness: 1.0+image_saturation: 0.5+image_resolution: 0.5+image_colorfulness: 0.5+ambiguity: 1.0+bbox_num: 1.0+bbox_iou: 1.0+bbox_std: 0.5+bbox_bright: 0.5+bbox_ambiguity: 0.5+bbox_aspect_ratio: 1.0+bbox_area_ratio: 0.5+bbox_edge_value:0.5	feature_distribution_mining算法中定义的特征，可自行修改权重。
score_threshold_up	N	0.6	predict_score_mining算法中定义的置信度最高值。取值范围0~1。
score_threshold_low	N	0.3	predict_score_mining算法中定义的置信度最低值。取值范围0~1。
margin	N	0.8	top2置信度差值定义, 差值超过该阈值，则为难例。取值范围0~1，默认值为0.8。
similarity_sample_ratio	N	1.0	image_similarity_mining算法中的相似比例定义。取值范围0~1，默认值为1.0。
task_summary_file	N	None	算法简化的日志输出路径及日志文件。请填写一个真实存在的OBS目录，以“obs://”开头。文件名称可自行定义。 例如：obs://obs_bucket_name/folder_name/xxx.log

参数名	是否必选	默认值	参数说明
output_dataset_type	N	manifest	支持directory和manifest类型。 ● directory：将原始图片和标签输出到结果目录的Data文件夹下。 ● manifest：仅输出manifest文件。 该参数在数据处理模块将根据页面选择自动填充。

输入要求

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在本任务中选择的场景类别一致。
- 选择“OBS目录”，目录下需包含用于推理的原始图片以及推理结果文件

  其目录结构如下所示。

```
input_path/
   --images/ # 文件夹名称必须为images
       ----1.jpg
       ----2.jpg
   --inference_results/  # 文件夹名称必须为inference_results
       ----1.jpg_result.txt
       ----2.jpg_result.txt
```

  其中，推理结果文件txt的内容需满足如下格式。如果使用ModelArts预置算法训练得到的模型进行推理，默认的推理结果已满足要求。

  – **图像分类**

```
{
   "predicted_label": "dog",
   "scores": [
      [
         "dog",
         "0.589"
      ],
      [
         "cat",
         "0.411"
      ]
   ]
}
```

  – **物体检测**

```
{
   "detection_classes": [
      "cat",
      "cat"
      ],
   "detection_boxes": [
      [
         117.56356048583984,
         335.9902648925781,
         270.50848388671875,
         469.0136413574219
      ],
      [
         18.747316360473633,
         13.10757064819336,
```

```
            217.25146484375,
            108.3551025390625
        ]
    ],
    "detection_scores": [
            0.5179755091667175,
            0.46941104531288147
        ]
}
```

输出说明

- 目标检测

  输出目录结构如下所示。

```
output_path :
    --Data
        ----1.jpg
        ----1.xml       # 导出筛选结果至该目录下
    --output.manifest
```

  其中manifest文件示例如下所示。

```
{"source":"/tmp/test_out/object_detection/images/be462ea9c5abc09f.jpg",
"hard":"True",
"hard-reasons":"0",  # 判定该样本为难例的原因，具体原因目前只在智能标注模块展示
"hard-coefficient":"1.0", # 难例算法得到的难例系数，越大代表可能是难例的概率越大
"annotation":[
{"annotation-loc":"/tmp/test_out/object_detection/annotations/be462ea9c5abc09f.xml",
"type":"modelarts/object_detection",
"annotation-format":"PASCAL VOC",
"annotated-by":"modelarts/hard_example_algo"}]}
```

- 图像分类

  输出目录结构如下所示。

```
output_path :
    --Data
        ----class1
            ------1.jpg
        ----class2
            ------2.jpg
    --output.manifest
```

  其中manifest文件示例如下所示。

```
{"source":"obs://obs_bucket_name/folder_name/catDog/5.jpg",
"hard":true,
"hard-reasons":"1-20-2-19-21-3",
"hard-coefficient":1.0,
"annotation":[
{"name":"cat",
"type":"modelarts/image_classification",
"confidence":0.599,
"annotated-by":"modelarts/hard_example_algo"}]}
```

日志文件说明

task_summary_file 为简化日志的输出文件路径，内容如下：

```
{
"task_status": 'SUCCEED', # 算法执行状态
"total_sample": integer,  # 输入样本总数
"hard_sample": integer     # 输出样本总数
}
```

或者是

```
{
"task_status": 'FAILED',
```

```
"error_message": 'xxxxxx' # 导致算法执行失败的异常信息
}
```

4.15.4.5 数据增强（数据扩增）

数据扩增算子概述

数据扩增主要用于训练数据集不足或需要仿真的场景，能通过对已标注的数据集做变换操作来增加训练图片的数量，同时会生成相应的标签。在深度学习领域，增强有重要的意义，能提升模型的泛化能力，增加抗扰动的能力。数据扩增过程不会改动原始数据，扩增后的图片或xml文件保存在指定的输出路径下。

ModelArts提供以下数据扩增算子：

表 4-39 数据扩增算子介绍

算子	算子说明	高级
AddNoise	添加噪声，模拟常见采集设备在采集图片过程中可能会产生的噪声。	● noise_type：添加噪声的分布类型，Gauss为高斯噪声，Laplace为拉普拉斯噪声，Poisson是泊松噪声，Impulse是脉冲噪声，SaltAndPepper为椒盐噪声。默认值为Gauss ● loc：噪声分布的均值，仅在Gauss和Laplace生效。默认值为0 ● scale：噪声分布的标准差，仅在Gauss和Laplance生效。默认值为1 ● lam：泊松分布的lambda系数，仅在Poisson有效。默认值为2 ● p：对于每个像素点，出现脉冲噪声或椒盐噪声的概率，仅在Impulse和SaltAndPepper有效。默认值为0.01 ● do_validation：数据扩增前是否做数据校验。默认值为True。
Blur	模糊，使用滤波器对图像进行滤波操作，有时用于模拟成像设备的成像。	● blur_type：可选Gauss和Average两种模式，分别为高斯和均值滤波。默认值为Gauss ● do_validation：数据扩增前是否做数据校验。默认值为True。
Crop	图片裁剪，随机裁剪图片的一部分作为新的图片。	● crop_percent_min：各边裁剪占比的随机取值范围的最小值。默认值为0.0 ● crop_percent_max：各边裁剪占比的随机取值范围的最大值。默认值为0.2 ● do_validation：数据扩增前是否做数据校验。默认值为True。
CutOut	随机擦除，在深度学习中常用的方法，用于模拟物体被障碍物遮挡。	do_validation：数据扩增前是否做数据校验。默认值为True。

算子	算子说明	高级
Flip	翻转，沿图片水平轴或竖直轴做翻转，是非常常见的增强方法。	● lr_ud：选择翻转的方向,lr为水平翻转，ud为竖直翻转。默认值为lr ● flip_p：做翻转操作的概率。默认值为1。 ● do_validation：数据扩增前是否做数据校验。默认值为True。
Grayscale	图片灰度化，将三通道的彩色图像转换到三通道的灰度图像。	do_validation：数据扩增前是否做数据校验。默认值为True。
HistogramEqual	直方图均衡化，多半是使用于让图片的视觉效果更加好，在某些场景下会使用。	do_validation：数据扩增前是否做数据校验。默认值为True。
LightArithmetic	亮度增强 ，对亮度空间做线性增强操作。	do_validation：数据扩增前是否做数据校验。默认值为True。
LightContrast	亮度对比度增强，使用一定的非线性函数改变亮度空间的亮度值。	func：默认值为gamma ● gamma为常见方法伽马矫正，公式为255*((v/255)**gamma)') ● sigmoid为函数为S型曲线，公式为255*1/(1+exp(gain*(cutoff-I_ij/255)))') ● log为对数函数，公式为255*gain*log_2(1+v/255) ● linear为线性函数，公式为127 + alpha*(v-127)') do_validation：数据扩增前是否做数据校验。默认值为True。
MotionBlur	运动模糊，模拟物体运动时产生的残影现象。	do_validation：数据扩增前是否做数据校验。默认值为True。
Padding	图片填充，在边缘添加黑色的边。	● px_top：图像顶端增加的像素行数。默认值为1 ● px_right：图像右侧增加的像素行数。默认值为1 ● px_left：图像左侧增加的像素行数。默认值为1 ● px_bottom：图像底侧增加的像素行数。默认值为1 ● do_validation：数据扩增前是否做数据校验。默认值为True。

算子	算子说明	高级
Resize	调整图片大小。	• height：变换后的图片高度。默认值224 • width：变换后的图片宽度。默认值224 • do_validation：数据扩增前是否做数据校验。默认值为True。
Rotate	旋转，将图像围绕中心点旋转的操作，操作完成之后保持图片原本的形状不变，不足的部分用黑色填充。	• angle_min：旋转角度随机取值范围的最小值，每张图片会从范围中随机取值作为自己的参数。默认值为90° • angle_max：旋转角度随机取值范围的最大值，每张图片会从范围中随机取值作为自己的参数。默认值为-90° • do_validation：数据扩增前是否做数据校验。默认值为True。
Saturation	色度饱和度增强，对图片的HSV中的H和S空间做线性的变化，改变图片的色度和饱和度。	do_validation：数据扩增前是否做数据校验。默认值为True。
Scale	图片缩放，将图片的长或宽随机缩放到一定倍数。	• scaleXY：缩放方向，X为水平，Y为垂直。默认值为X • scale_min：缩放比例随机取值范围的最小值。默认为0.5 • scale_max：缩放比例随机取值范围的最大值。默认值为1.5 • do_validation：数据扩增前是否做数据校验。默认值为True。
Sharpen	图像锐化，用于将边缘清晰化，让物体边缘更加明显。	do_validation：数据扩增前是否做数据校验。默认值为True。
Shear	图片错切，一般用于图片的几何变换，通过线性函数将像素点进行映射。	• shearXY：错切方向，X为水平，Y为竖直。默认值为X • shear_min：错切角度随机取值范围的最小值。默认值为-30 • shear_max：错切角度随机取值范围的最大值。默认值为30 • do_validation：数据扩增前是否做数据校验。默认值为True。
Translate	图片平移，将图片整体像X轴或Y轴平移，超出原图部分舍弃，丢失部分用黑色填充。	• translateXY：平移的方向，X为水平，Y为竖直。默认值为X • do_validation：数据扩增前是否做数据校验。默认值为True。

算子	算子说明	高级
Weather	添加天气，模拟天气效果。	weather_mode：添加天气的模式，默认值为Rain。 ● Rain：下雨 ● Fog：雾 ● Snow：雪 ● Clouds：云 do_validation：数据扩增前是否做数据校验。默认值为True。

输入要求

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在本任务中选择的场景类别一致。
- 选择“OBS目录”，存放结构支持“包含图片和标注信息”模式。

  “包含图片和标注信息”，根据不同场景类型，结构不同。

  图像分类场景，其目录结构如下所示。如下目录结构，仅支持单标签场景。

```
input_path/
    --label1/
        ----1.jpg
    --label2/
        ----2.jpg
    --../
```

  物体检测场景，其目录结构如下所示。支持jpg、jpeg、png、bmp格式的图片，xml为标准的PACAL VOC格式标注文件。

```
input_path/
    --1.jpg
    --1.xml
    --2.jpg
    --2.xml
    ...
```

输出说明

由于算法中有些操作将会舍弃一些数据，输出文件夹里可能不包含全量数据集。例如，“Rotate”会舍弃标注框超出原始图片边界的图片。

输出目录结构如下所示。其中“Data”文件夹用于存放新生成的图片和标注信息，“manifest”文件存储文件夹中图片的结构，可直接导入到数据管理的数据集中。

```
|----data_url
    |----Data
        |----xxx.jpg
        |----xxx.xml(xxx.txt)
    |----output.manifest
```

其中manifest文件内容示例如下所示。

```
{
    "id": "xss",
    "source": "obs://home/fc8e2688015d4a1784dcbda44d840307_14.jpg",
```

```
  "usage": "train",
  "annotation": [
    {
      "name": "Cat",
      "type": "modelarts/image_classification"
    }
  ]
}
```

4.15.4.6 数据增强（图像生成）

图像生成算子概述

图像生成算子利用Gan网络依据已知的数据集生成新的数据集。Gan是一个包含生成器和判别器的网络，生成器从潜在空间中随机取样作为输入，其输出结果需要尽量模仿训练集中的真实样本。判别器的输入则为真实样本或生成网络的输出，其目的是将生成网络的输出从真实样本中尽可能分辨出来。而生成网络则要尽可能地欺骗判别网络。两个网络相互对抗、不断调整参数，最终目的是使判别网络无法判断生成网络的输出结果是否真实。训练中获得的生成器网络可用于生成与输入图片相似的图片，用作新的数据集参与训练。基于Gan网络生成新的数据集不会生成相应的标签。图像生成过程不会改动原始数据，新生成的图片或xml文件保存在指定的输出路径下。

ModelArts提供两种类型的图像生成算子：

- CycleGan算子：基于CycleGAN用于生成域迁移的图像，即将一类图片转换成另一类图片，把X空间中的样本转换成Y空间中的样本。CycleGAN可以利用非成对数据进行训练。模型训练时运行支持两个输入，分别代表数据的原域和目标域，在训练结束时会生成所有原域向目标域迁移的图像。

图 4-103 CycleGan 算子

表 4-40 CycleGan 算子高级参数

参数名	默认值	参数说明
do_validation	True	是否做数据校验，默认为True，表示数据生成前需要做数据校验，否则只做数据生成。
image_channel	3	生成图像的通道数。
image_height	256	图像相关参数：生成图像的高，大小需要是2的次方。
image_width	256	图像相关参数：生成图像的宽，大小需要是2的次方

参数名	默认值	参数说明
batch_size	1	训练相关参数：批量训练样本个数。
max_epoch	100	训练相关参数：训练遍历数据集次数。
g_learning_rate	0.0001	训练相关参数：生成器训练学习率。
d_learning_rate	0.0001	训练相关参数：判别器训练学习率。
log_frequency	5	训练相关参数：日志打印频率（按step计数）。
save_frequency	5	训练相关参数：模型保存频率（按epoch计数）。
predict	False	是否进行推理预测，默认为False。如果设置True，需要在resume参数设置已经训练完成的模型的obs路径。
resume	empty	如果predict设置为True，需要填写Tensorflow模型文件的obs路径用于推理预测。当前仅支持“.pb”格式的模型。示例：obs://xxx/xxxx.pb。 默认值为empty。

- StyleGan算子：基于StyleGan2用于在数据集较小的情形下，随机生成相似图像。StyleGAN提出了一个新的生成器结构，能够控制所生成图像的高层级属性(high-level attributes)，如发型、雀斑等；并且生成的图像在一些评价标准上得分更好。而本算法又增加了数据增强算法，可以在较少样本的情况下也能生成较好的新样本，但是样本数尽量在70张以上，样本太少生成出来的新图像不会有太多的样式。

图 4-104 StyleGan 算子

表 4-41 StyleGan 算子高级参数

参数名	默认值	参数说明
resolution	256	生成正方形图像的高宽，大小需要是2的次方。

参数名	默认值	参数说明
batch-size	8	批量训练样本个数。
total-kimg	300	总共训练的图像数量为total_kimg*1000。
generate_num	300	生成的图像数量，如果是多个类的，则为每类生成的数量。
predict	False	是否进行推理预测，默认为False。如果设置True，需要在resume参数设置已经训练完成的模型的obs路径。
resume	empty	如果predict设置为True，需要填写Tensorflow模型文件的obs路径用于推理预测。当前仅支持“.pb”格式的模型。示例：obs://xxx/xxxx.pb。 默认值为empty。
do_validation	True	是否做数据校验，默认为True，表示数据生成前需要做数据校验，否则只做数据生成。

数据输入

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在本任务中选择的场景类别一致。
- 选择“OBS目录”，图像生成算子不需要标注信息，输入支持单层级或双层级目录，存放结构支持“单层级”或“双层级”模式。

单层级目录结构如下所示：

```
image_folder----0001.jpg
          ----0002.jpg
          ----0003.jpg
          ...
          ----1000.jpg
```

双层级目录结构如下所示：

```
image_folder----sub_folder_1----0001.jpg
                        ----0002.jpg
                        ----0003.jpg
                        ...
                        ----0500.jpg
          ----sub_folder_2----0001.jpg
                        ----0002.jpg
                        ----0003.jpg
                        ...
                        ----0500.jpg
                        ...
          ----sub_folder_100----0001.jpg
                          ----0002.jpg
                          ----0003.jpg
                          ...
                          ----0500.jpg
```

输出说明

输出目录的结构如下所示。其中“model”文件夹存放用于推理的“frozen pb”模型，“samples”文件夹存放训练过程中输出图像，“Data”文件夹存放训练模型生成的图像。

```
train_url----model----CYcleGan_epoch_10.pb
                 ----CYcleGan_epoch_20.pb
                 ...
                 ----CYcleGan_epoch_1000.pb
        ----samples----0000_0.jpg
                 ----0000_1.jpg
                 ...
                 ----0100_15.jpg
        ----Data----CYcleGan_0_0.jpg
                 ----CYcleGan_0_1.jpg
                 ...
                 ----CYcleGan_16_8.jpg
        ----output_0.manifest
```

其中manifest文件内容示例如下所示。

```
{
    "id": "xss",
    "source": "obs://home/fc8e2688015d4a1784dcbda44d840307_14.jpg",
    "usage": "train",
    "annotation": [
        {
            "name": "Cat",
            "type": "modelarts/image_classification"
        }
    ]
}
```

5 开发环境（New）

5.1 开发环境简介

说明

本手册基于新版开发环境Notebook功能展开介绍。

软件开发的历史，就是一部降低开发者成本，提升开发体验的历史。在AI开发阶段，ModelArts也致力于提升AI开发体验，降低开发门槛。ModelArts开发环境，以云原生的资源使用和开发工具链的集成，目标为不同类型AI开发、探索、教学用户，提供更好云化AI开发体验。

ModelArts Notebook 云上云下，无缝协同

- 代码开发与调测。云化JupyterLab使用，本地IDE+ModelArts插件远程开发能力，贴近开发人员使用习惯
- 云上开发环境，包含AI计算资源，云上存储，预置AI引擎
- 运行环境自定义，将开发环境直接保存成为镜像，供训练、推理使用

ModelArts CodeLab（JupyterLab），让AI探索&教学更简单

- 云原生Notebook，案例内容秒级接入与分享
- Serverless化实例管理，资源自动回收
- 免费算力，规格按需切换

亮点特性 1：远程开发 - 支持本地 IDE 远程访问 Notebook

新版Notebook提供了远程开发功能，通过开启SSH连接，用户本地IDE可以远程连接到ModelArts的Notebook开发环境中，调试和运行代码。

对于使用本地IDE的开发者，由于本地资源限制，运行和调试环境大多使用团队公共搭建的CPU或GPU服务器，并且是多人共用，这带来一定的环境搭建和维护成本。

而ModelArts的Notebook的优势是即开即用，它预先装好了不同的AI引擎，并且提供了非常多的可选规格，用户可以独占一个容器环境，不受其他人的干扰。只需简单配置，用户即可通过本地IDE连接到该环境进行运行和调试。

图 5-1 本地 IDE 远程访问 Notebook 开发环境

ModelArts的Notebook可以视作是本地PC的延伸，均视作本地开发环境，其读取数据、训练、保存文件等操作与常规的本地训练一致。

对于习惯使用本地IDE的开发者，使用远程开发方式，不影响用户的编码习惯，并且可以方便快捷的使用云上的Notebook开发环境。

本地IDE当前支持VS Code、PyCharm、SSH工具。还有专门的插件PyCharm Toolkit和VS Code Toolkit，方便将云上资源作为本地的一个扩展。

亮点特性 2：预置镜像 - 即开即用，优化配置，支持主流 AI 引擎

每个镜像预置的AI引擎和版本是固定的，在创建Notebook实例时明确AI引擎和版本，包括适配的芯片。

ModelArts开发环境给用户提供了一组预置镜像，主要包括PyTorch、Tensorflow、MindSpore系列。用户可以直接使用预置镜像启动Notebook实例，在实例中开发完成后，直接提交到ModelArts训练作业进行训练，而不需要做适配。

ModelArts开发环境提供的预置镜像版本是依据用户反馈和版本稳定性决定的。当用户的功能开发基于ModelArts提供的版本能够满足的时候，比如用户开发基于Minspore1.5，建议用户使用预置镜像，这些镜像经过充分的功能验证，并且已经预置了很多常用的安装包，用户无需花费过多的时间来配置环境即可使用。

ModelArts开发环境提供的预置镜像主要包含：

- 常用预置包，基于标准的Conda环境，预置了常用的AI引擎，例如PyTorch、MindSpore；常用的数据分析软件包，例如Pandas、Numpy等；常用的工具软件，例如cuda、cudnn等，满足AI开发常用需求。
- 预置Conda环境：每个预置镜像都会创建一个相对应的Conda环境和一个基础Conda环境python（不包含任何AI 引擎），如预置Mindspore所对应的Conda环境如下：

用户可以根据是否使用AI引擎参与功能调试，选择不同的Conda环境。

- Notebook：是一款Web应用，能够使用户在界面编写代码，并且将代码、数学方程和可视化内容组合到一个文档中。
- JupyterLab插件：插件包括规格切换，分享案例到AI Gallery进行交流，停止实例等，提升用户体验。
- 支持SSH远程连接功能，通过SSH连接启动实例，在本地调试就可以操作实例，方便调试。
- ModelArts开发环境提供的预置镜像支持功能开发后，直接提到ModelArts训练作业中进行训练。

说明

- 为了简化操作，ModelArts的新版Notebook，同一个Notebook实例中不支持不同引擎之间的切换。
- 不同Region支持的AI引擎不一样，请以控制台实际界面为准。

亮点特性 3：提供在线的交互式开发调试工具 JupyterLab

ModelArts集成了基于开源的JupyterLab，可为您提供在线的交互式开发调试。您无需关注安装配置，在ModelArts管理控制台直接使用Notebook，编写和调测模型训练代码，然后基于该代码进行模型的训练。

JupyterLab是一个交互式的开发环境，是Jupyter Notebook的下一代产品，可以使用它编写Notebook、操作终端、编辑MarkDown文本、打开交互模式、查看csv文件及图片等功能。

5.2 开发环境使用场景

ModelArts提供灵活开放的开发环境，您可以根据实际情况选择。

- ModelArts提供了云化版本的Notebook，无需关注安装配置，即开即用，具体参见**使用JupyterLab开发模型**。
- ModelArts也提供了本地IDE的方式开发模型，通过开启SSH连接，用户本地IDE可以远程连接到ModelArts的Notebook开发环境中，调试和运行代码。本地IDE方式不影响用户的编码习惯，并且可以方便快捷的使用云上的Notebook开发环境。

  本地IDE当前支持VS Code、PyCharm、SSH工具。PyCharm和VS Code还分别有专门的插件PyCharm Toolkit、VS Code Toolkit，让远程连接操作更便捷。具体参见**使用本地IDE开发模型**。
- ModelArts的开发环境中集成了MindStudio工具，支持在ModelArts上使用MindStudio工具做算子开发，具体参考**使用MindStudio开发Ascend算子**。

- ModelArts还提供了可视化编排工具ML Studio，用户可以通过拖拉拽的方式开发基于ML Studio引擎的模型，具体参见**使用可视化工具ML Studio开发模型**。

5.3 CodeLab：免费体验 Notebook

面向众多开发者，ModelArts提供了CodeLab功能，一方面，一键进入开发环境，同时预置了免费的算力规格，可直接免费体验Notebook功能；另一方面，针对AIGallery社区发布的Notebook样例（.ipynb格式文件），可直接在CodeLab中打开，查看他人分享的样例代码。

功能亮点

- **免费算力**

  CodeLab内置了免费算力，包含CPU和GPU两种。您可以使用免费规格，端到端体验ModelArts Notebook能力。也可使用此免费算力，在线完成您的算法开发。

- **即开即用**

  无需创建Notebook实例，打开即可编码。

- **高效分享**

  ModelArts在AI Gallery中提供的Notebook样例，可以直接通过Run in ModelArts，一键打开运行和学习，并且可将样例修改后分享到AI Gallery中直接另存用于个人开发。

  同时，您开发的代码，也可通过CodeLab快速分享到AI Gallery中给他人使用学习。

使用限制

- CodeLab默认打开，使用的是CPU计算资源。如需切换为GPU，请在右侧窗口，更换GPU规格。
- 自启动后，免费规格默认可使用1小时，请注意右上角的剩余时长。超过1小时后，可执行续期操作，且系统每隔一段时间，将提醒确认下续期。
- 免费的CodeLab主要用于体验，72小时内未使用，将释放资源。保存在其中的代码文档将丢失，请注意备份文件以及使用时长。
- CodeLab中默认内置的AI框架及版本，请通过首页的Notebook下方获取，如下所示。

图 5-2 支持的框架及版本

CodeLab 入口

- ModelArts管理控制台的“总览”页

  在“开发工具”区域下方，展示“CodeLab”简介卡片，单击“立即体验”，即可进入。

  图 5-3 CodeLab 入口

- AI Gallery页面提供的Notebook样例

  a. 在**AI Gallery**页面，单击“资产集市 > 开发 > Notebook”栏目中的任意Notebook样例，进入详情页。

  b. 订阅该Notebook后，单击“Run in ModelArts”，进入CodeLab并打开该样例。等待右上角连接成功即可运行Notebook样例。

  图 5-4 运行 Notebook

说明

- 首次进入CodeLab时，会提醒先登录ModelArts控制台，由于需要进行环境准备，需要等待1~2分钟才可顺利进入开发页面，请耐心等待。
- CodeLab的内部环境依托Notebook功能，因此其工作环境，与JupyterLab界面相似。

体验 CodeLab

1. **进入CodeLab主页。**

   从管理控制台总览页进入，展示CodeLab首页。

图 5-5 CodeLab 首页

2. **常用功能。**

   CodeLab的界面依托于JupyterLab，其相关的常见功能，与JupyterLab相同。常用操作指导可参见JupyterLab操作指导：**JupyterLab简介及常用操作**。

   说明

   由于CodeLab的存储为系统默认路径，在使用“上传文件”或“下载文件至本地”时，只能使用JupyterLab页面提供的功能。

   如需使用大文件上传和下载的功能，建议您前往Notebook，创建一个收费的实例进行使用。

3. **切换规格。**

   CodeLab支持CPU和GPU两种规格在右侧区域，单击切换规格，修改规格类型。

图 5-6 切换规格

4. **资源监控。**

   在使用过程中，如果想了解资源使用情况，可在右侧区域选择“Resource Monitor”，展示“CPU使用率”和“内存使用率”。

图 5-7 资源监控

5. 分享副本到AI Gallery。单击右上角的 ，将修改后的Notebook样例保存分享到AI Gallery中，供自己或他人学习使用。

图 5-8 分享到 AI Gallery

分享成功后，通过分享链接可以打开分享的副本，也可以在AI Gallery中找到分享的Notebook。

图 5-9 发布成功

5.4 管理 Notebook 实例

5.4.1 创建 Notebook 实例

在开始进行模型开发前，您需要创建Notebook实例，并打开Notebook进行编码。

背景信息

- Notebook使用涉及到计费，具体收费项如下：
  - 处于“运行中”状态的Notebook，会消耗资源，产生费用。根据您选择的资源不同，收费标准不同，价格详情请参见**产品价格详情**。当您不需要使用Notebook时，建议停止Notebook，避免产生不必要的费用。
  - 创建Notebook时，如果选择使用云硬盘EVS存储配置，云硬盘EVS会一直收费，建议及时停止并删除Notebook，避免产品不必要的费用。小数据量建议优先使用免费的默认存储。
- 在创建Notebook时，默认会开启自动停止功能，在指定时间内停止运行Notebook，避免资源浪费。
- 只有处于“运行中”状态的Notebook，才可以执行打开、停止操作。
- 一个账户最多创建10个Notebook。

创建 Notebook 实例

1. 登录ModelArts管理控制台，在左侧导航栏中选择“全局配置”，检查是否配置了访问授权。若未配置，请先配置访问授权。参考使用委托授权完成操作。

   图 5-10 配置授权

2. 登录ModelArts管理控制台，在左侧导航栏中选择“开发环境 > Notebook”，进入“Notebook”新版管理页面。

   图 5-11 进入新版 Notebook

3. 单击“创建”，进入“创建Notebook”页面，请参见如下说明填写参数。

   a. 填写Notebook基本信息，包含名称、描述、是否自动停止，详细参数请参见表5-1。

   图 5-12 Notebook 基本信息

表 5-1 基本信息的参数描述

参数名称	说明
“名称”	Notebook的名称。只能包含数字、大小写字母、下划线和中划线，长度不能超过64位且不能为空。
“描述”	对Notebook的简要描述。
“自动停止”	默认开启，且默认值为“1小时”，表示该Notebook实例将在运行1小时之后自动停止，即1小时后停止规格资源计费。 开启自动停止功能后，可选择“1小时”、“2小时”、“4小时”、“6小时”或“自定义”几种模式。选择“自定义”模式时，可指定1~24小时范围内任意整数。

b. 填写Notebook详细参数，如镜像、资源规格等，详细参数请参见表5-2。

图 5-13 Notebook 实例的详细参数

表 5-2 Notebook 实例的详细参数说明

参数名称	说明
“镜像”	支持公共镜像和自定义镜像。 ● 公共镜像：即预置在ModelArts内部的AI引擎。 ● 自定义镜像：可以将基于公共镜像创建的实例保存下来，作为自定义镜像使用。自定义镜像功能暂未正式开放。 一个镜像对应支持一种AI引擎，创建Notebook实例时选择好了对应AI引擎的镜像。用户可以根据需要选择镜像。 Notebook运行停止后，可以在同一个Notebook实例中变更镜像。

参数名称	说明
“资源池”	支持公共资源池和专属资源池。 “公共资源池”无需单独购买，即开即用，按需付费，即按您的Notebook实例运行时长进行收费。 “专属资源池”按实际情况选择已创建的专属资源池。如果没有专属资源，需要单独购买并创建。
“类型”	芯片类型包括CPU、GPU和Ascend类型。 不同的镜像支持的芯片类型不同，根据实际需要选择。 GPU性能更佳，但是相对CPU而言，费用更高。
“规格”	根据选择的芯片类型不同，可选资源规格也不同。请根据界面实际情况和需要选择。 ● CPU规格 “2核8GB”：Intel CPU通用规格，用于快速数据探索和实验 “8核32GB”：Intel CPU算力增强型，适用于密集计算场景下运算 ● GPU规格 “GPU: 1*V100(32GB)|CPU: 8 核 64GB”：NVIDIA V100 GPU单卡规格，32GB显存，适合深度学习场景下的算法训练和调测 “GPU: 2*V100(64GB)|CPU: 16 核 128GB”：NVIDIA V100 GPU两卡规格，32GB显存，适合深度学习场景下的算法训练和调测 ● Ascend规格 “Ascend: 1*Ascend 910|CPU: 24 核 96GB”：昇腾910(32GB显存)单卡规格，配搭ARM处理器，适合深度学习场景下的模型训练和调测 “Ascend: 2*Ascend 910|CPU: 48 核 192GB”：昇腾910(32GB显存)两卡规格，配搭ARM处理器，适合深度学习场景下的模型训练和调测 “Ascend: 8*Ascend 910|CPU: 192 核 768GB”：昇腾910(32GB显存)八卡规格，配搭ARM处理器，适合深度学习场景下的模型训练和调测

参数名称	说明
“存储配置”	包括“默认存储”、“云硬盘EVS”和“弹性文件服务SFS”。请根据界面实际情况和需要选择。 ● 选择“默认存储”作为存储位置。 选择此模式，平台免费为每一个Notebook提供50GB的默认存储。 ● 选择“云硬盘EVS”作为存储位置。 根据实际使用量设置磁盘规格。磁盘规格默认5GB。磁盘规格的取值范围为5GB～4096GB。 从Notebook实例创建成功开始，直至实例删除成功，磁盘每GB按照规定费用收费。 ● 选择“弹性文件服务SFS”作为存储位置。 仅专属资源池支持。 “默认存储”、“硬盘EVS”和“弹性文件服务SFS”的存储路径挂载在/home/ma-user/work目录下。用户在Notebook实例中的所有文件读写操作都是针对该存储目录下的的内容操作，与OBS对象存储（OBS对象桶）无关。Notebook实例运行中，可以**动态挂载OBS并行文件系统**用来读取数据。 停止或重启Notebook实例时，内容会被保留，不丢失。 删除Notebook实例时，内容不保留。
“SSH远程开发”	开启此功能后，用户可以在本地开发环境中远程接入Notebook实例的开发环境。
“密钥对”	开启“SSH远程开发”功能后，需要设置此参数。 可以选择已有密钥对。 也可以单击密钥对右侧的“立即创建”，跳转到数据加密控制台，在“密钥对管理 > 私有密钥对”页面，单击“创建密钥对”。 **注意** 创建好的密钥对，请下载并妥善保存，使用本地IDE远程连接云上Notebook开发环境时，需要用到密钥对进行鉴权认证。

参数名称	说明
“远程访问白名单”	可选，开启“SSH远程开发”功能后，可以设置此参数。 设置为允许远程接入访问这个Notebook的IP地址（例如本地PC的IP地址或者访问机器的外网IP地址，最多配置5个，用英文逗号隔开），不设置则表示无接入IP地址限制。 如果用户使用的访问机器和ModelArts服务的网络有隔离，则访问机器的外网地址需要在主流搜索引擎中搜索“IP地址查询”获取，而不是使用ipconfig或ifconfigip命令在本地查询。 **图 5-14** 查询外网 IP 地址 创建完Notebook后，可以在Notebook详情页中修改白名单IP地址。

4. 参数填写完成后，单击“立即创建”进行规格确认。
5. 参数确认无误后，单击“提交”，完成Notebook的创建操作。

   进入Notebook列表，正在创建中的Notebook状态为“创建中”，创建过程需要几分钟，请耐心等待。当Notebook状态变为“运行中”时，表示Notebook已创建并启动完成。
6. 在Notebook列表，单击实例名称，进入实例详情页，查看Notebook实例配置信息。

图 5-15 查看 Notebook 实例详情

在白名单右侧单击修改，可以修改允许远程访问的白名单IP地址。

单击“添加数据存储”，可以挂载OBS并行文件系统，方便读取数据，具体操作参见**动态挂载OBS并行文件系统**。

如果存储使用的是云硬盘EVS，单击存储容量右侧的“扩容”，可以动态扩充云硬盘EVS的容量，具体操作参见**动态扩充云硬盘EVS容量**。

5.4.2 打开 Notebook 实例

针对创建好的Notebook实例（即状态为“运行中”的实例），可以打开Notebook并在开发环境中启动编码。

基于不同AI引擎创建的Notebook实例，打开方式不一样。

pytorch、tensorflow、mindspore、Tensorflow-mindspore、cylp-cbcpy、rlstudio-ray、mindquantum-mindspore镜像支持以下2种方式访问：

- 本地IDE使用PyCharm/VS Code/SSH工具，远程连接访问，具体参见**使用本地IDE开发模型**。
- 在线JupyterLab访问，具体参见**使用JupyterLab开发模型**。

mlstudio-pyspark镜像仅支持在线JupyterLab访问，具体参见**使用JupyterLab开发模型**

mindstudio镜像仅支持在本地使用MobaXterm通过SSH远程访问，具体参见**使用MindStudio开发Ascend算子**。

modelbox镜像仅支持在本地使用VS Code插件远程访问，具体参见**配置本地VSCode连接云上开发环境Modelbox镜像**。

ModelArts提供的Notebook实例是以ma-user启动的，用户进入实例后，工作目录默认是/home/ma-user。

```
sh-4.4$pwd
/home/ma-user
sh-4.4$
```

创建实例，持久化存储挂载路径为/home/ma-user/work目录。

```
sh-4.4$pwd
/home/ma-user
sh-4.4$cd work/
sh-4.4$pwd
/home/ma-user/work
sh-4.4$
```

存放在work目录的内容，在实例停止、重新启动后依然保留，其他目录下的内容不会保留，使用开发环境时建议将需要持久化的数据放在/home/ma-user/work目录。

5.4.3 启动/停止/删除实例

启动/停止实例

由于运行中的Notebook将一直耗费资源，您可以通过停止操作，停止资源消耗。对于停止状态的Notebook，可通过启动操作重新使用Notebook。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“开发环境 > Notebook”，进入新版Notebook管理页面。
2. 执行如下操作启动或停止Notebook。
   - **启动Notebook**：单击“操作”列的“启动”。只有处于“停止”状态的Notebook可以执行启动操作。

– **停止Notebook**：单击“操作”列的“停止”。只有处于“运行中”状态的Notebook可以执行停止操作。

注意

Notebook停止后：

- /home/ma-user/work目录下的数据会保存，其余目录下内容会被清理。例如：用户在开发环境中的其他目录下安装的外部依赖包等，在Notebook停止后会被清理。
- Notebook实例将停止计费，但如有EVS盘挂载，存储部分仍会继续计费。

删除实例

针对不再使用的Notebook实例，可以删除以释放资源。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“开发环境>Notebook”，进入新版Notebook管理页面。
2. 在Notebook列表中，单击操作列的“删除”，在弹出的确认对话框中，确认信息无误，然后单击“确定”，完成删除操作。

注意

Notebook删除后不可恢复，请谨慎操作。

无论Notebook使用默认存储还是云硬盘EVS存储，实例删除后，挂载目录下的数据也将一并删除，请谨慎操作。

5.4.4 变更 Notebook 实例运行规格

ModelArts允许用户在同一个Notebook实例中切换节点运行规格，方便用户灵活调整规格资源。

约束限制

Notebook实例状态必须在“停止”中。

变更实例规格操作

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“开发环境 > Notebook”，进入新版Notebook管理页面。
2. 在Notebook列表，单击某个Notebook实例规格栏的 ▾ ，在下拉框中选择对应规格资源。

图 5-16 变更规格

5.4.5 变更 Notebook 实例镜像

ModelArts允许用户在同一个Notebook实例中切换镜像，方便用户灵活调整实例的AI引擎。

约束限制

Notebook实例状态必须在“停止”中。

变更实例镜像操作

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“开发环境 > Notebook”，进入新版Notebook管理页面。
2. 在Notebook列表，单击某个Notebook实例操作栏的“更多 > 变更镜像”，在变更镜像窗口选择新的镜像，单击“确定”。

图 5-17 变更镜像

3. 在镜像窗口选择新的镜像，单击“确定”，变更成功后，在Notebook列表页的镜像栏，可以查看到变更后的镜像。

5.4.6 开发环境中如何选择存储

不同存储的实现都不同，在性能、易用性、成本的权衡中可以有不同的选择，没有一个存储可以覆盖所有场景，了解下云上开发环境中各种存储使用场景说明，更能提高使用效率。

表 5-3 云上开发环境中各种存储使用场景说明

存储类型	建议使用场景	优点	缺点
EFS文件存储（默认存储）	针对探索、实验等非正式生产场景，建议使用这种。 开发环境和训练环境可以同时挂载一块EFS存储，省去了每次训练作业下载数据的要求，一般来说重IO读写模型，超过32卡的大规模训练不适合。	提供免费的50GB共享网络存储。 实现为NFS，可以在多个开发环境、开发环境和训练之间共享，如果不需要重型分布式训练作业，特别是启动训练作业时，不需要额外再对数据进行下载，这种存储便利性可以作为首选。	缺点是生产环境下，需要额外申请并配置，以及重型训练作业时可能遇到IO瓶颈。
EVS云硬盘块存储	比较适合只在开发环境中做数据、算法探索，性能较好。	块存储SSD，可以理解为一个磁盘，整体IO性能比NFS要好，可以动态扩充，最大可以到4096GB。 云硬盘EVS作为持久化存储挂载在/home/ma-user/work目录下，该目录下的内容在实例停止后会被保留，存储支持在线按需扩容。	缺点是只能在单个开发环境中使用 。
OBS并行文件系统	数据集的存储。 ● 数据集预览和操作，将承载数据集的OBS并行文件系统挂载至Notebook中，可以像本地文件系统一样操作数据集。 ● 在Notebook中训练时，可直接使用挂载至Notebook容器中的OBS数据集。	廉价存储，大吞吐量，但是小文件读写较弱。建议上传时按照128MB或者64MB打包或者切分，使用时边下载边解压后在本地读取。	缺点是对象存储语义，和Posix语义有区别，需要进一步理解。
OBS对象存储	新版Notebook不支持直接存储在OBS桶或挂载OBS桶。 在开发环境中做大规模的数据上传下载时，可以通过OBS桶做中转。	-	-
本地存储	重型训练任务首选	容器运行所在虚拟机或者裸金属机器上自带的SSD高性能存储，文件读写的吞吐量大，建议对于重型训练任务先将数据准备到对应目录再启动训练。	缺点是存储生命周期和容器生命周期绑定，每次训练都要下载数据。

如何使用

1. 在开发环境中如何使用EFS文件存储（默认存储）？

   在**创建Notebook实例**时选择默认存储即可。

2. 在开发环境中如何使用云硬盘EVS块存储?

   在**创建Notebook实例**时选择云硬盘存储，Notebook运行过程中可以**动态扩充云硬盘EVS容量**。

3. 在开发环境中如何使用OBS并行文件系统?

   Notebook运行过程中可以**动态挂载OBS并行文件系统**。

5.4.7 动态挂载 OBS 并行文件系统

什么是动态挂载 OBS 并行文件系统

在运行态的Notebook容器中，采用动态挂载特性，将OBS对象存储模拟成本地文件系统。其本质是通过挂载工具，将对象协议转为POSIX文件协议。挂载后应用层可以在容器中正常操作OBS对象。

动态挂载适用于哪些使用场景

场景1：数据集预览和操作，将承载数据集的OBS挂载至Notebook中，可以像本地文件系统一样操作数据集。

场景2：在Notebook中训练时，可直接使用挂载至Notebook容器中的数据集。

动态挂载 OBS 并行文件系统有什么限制

OBS提供两种桶，对象存储（对象桶）和并行文件系统。

ModelArts的Notebook（新版）仅支持挂载OBS的**并行文件系统**，挂载至Notebook容器/data/的子目录下。

动态挂载 OBS 并行文件系统操作

方式1：通过ModelArts控制台操作

1. 登录ModelArts管理控制台，在左侧导航栏中选择“开发环境 > Notebook”，进入“Notebook”新版管理页面。
2. 选择运行中的Notebook实例，单击实例名称，进入Notebook实例详情页面，单击“添加数据存储”，设置挂载参数。
   a. 设置本地挂载目录，在/data/目录下输入一个文件夹名称，例如：demo。挂载时，后台自动会在Notebook容器的/data/目录下创建该文件夹，用来挂载OBS文件系统。
   b. 选择存放OBS并行文件系统下的文件夹，单击“确定”。

图 5-18 动态挂载 OBS

3. 挂载成功后，可以在Notebook实例详情页查看到挂载结果。

图 5-19 挂载成功

方式2：API模式

动态挂载API接口已发布至华北-北京四和华东-上海一站点。必须在新版Notebook中执行挂载脚本。

挂载脚本代码示例如下：

```
import os
from json import JSONEncoder
from modelarts.config.auth import auth_by_apig
from modelarts.session import Session

session = Session()
request_url = "/v1/{}/notebooks/{}/storage".format(os.environ['PROJECT_ID'], os.environ["INSTANCE_ID"])


# 查询动态挂载列表
auth_by_apig(session, 'GET', request_url)


# 动态挂载，OBS路径obs://obs-bucket-train/dir/需要根据实际修改，挂载路径mount_path取值为“/data/xxx/”，本示例以“/data/demo/”为例，“demo”可以自定义。
body = {
    "category": "OBS",
    "uri": "obs://obs-bucket-train/dir/",
    "mount_path": "/data/demo/"
}
auth_by_apig(session, 'POST', request_url, body=JSONEncoder().encode(body))

# 动态卸载
auth_by_apig(session, 'DELETE', request_url + "/ea217c4f-3282-4af2-98ea-d3c668d2fba9")
```

5.4.8 动态扩充云硬盘 EVS 容量

什么是动态扩容 EVS

存储配置采用云硬盘EVS的Notebook实例， 存储盘是挂载至容器/home/ma-user/work/目录下， 可以在实例运行中的状态下，动态扩充存储盘容量，最大动态扩容200GB。

动态扩容 EVS 适用于哪些使用场景

在Notebook开发过程中，初期存储使用量较小时， 创建Notebook可以选择小容量EVS， 比如5G大小； 开发完成后，需要大规模数据集训练，此时再将存储容量扩容至当前阶段所需容量，可以节约成本。

动态扩容 EVS 有什么限制

- Notebook实例的存储配置采用的是云硬盘EVS。

  图 5-20 创建 Notebook 实例时选择云硬盘 EVS 存储

- 单次最大可以扩容100GB，扩容后的总容量不超过4096GB。
- 云硬盘EVS存储容量最大支持4096GB，达到4096GB时，不允许再扩容。
- 实例停止后，扩容后的容量仍然有效。计费也是按照扩容后的云硬盘EVS容量进行计费。
- 云硬盘EVS只要使用就会计费，请在停止Notebook实例后，确认不使用就及时删除数据，释放资源，避免产生费用。

动态扩容 EVS 操作

1. 登录ModelArts管理控制台，在左侧导航栏中选择“开发环境 > Notebook”，进入“Notebook”新版管理页面。
2. 选择运行中的Notebook实例，单击实例名称，进入Notebook实例详情页面，单击“扩容”。

图 5-21 Notebook 实例详情页

3. 设置待扩充的存储容量大小，单击“确定”。系统显示“扩容中”，扩容成功后，可以看到扩容后的存储容量。

图 5-22 扩容

图 5-23 扩容中

5.4.9 修改 Notebook 远程连接 IP 地址

在创建Notebook实例时，如果设置了允许远程连接Notebook的白名单IP地址，当用户需要更换一个IP地址远程连接Notebook实例时，则可以在Notebook的实例详情页修改白名单IP地址。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“开发环境 > Notebook（New）”，进入新版Notebook管理页面。
2. 在Notebook列表，单击某个具体的实例名称，进入查看实例详情页面。单击白名单右侧的修改，可以修改允许远程访问Notebook的IP地址。修改后，单击保存即可。

   图 5-24 修改白名单 IP 地址

   此处的IP地址，请填写外网IP地址。如果用户使用的访问机器和华为云ModelArts服务的网络有隔离，则访问机器的外网地址需要在主流搜索引擎中搜索“IP地址查询”获取，而不是使用ipconfig或ifconfigip命令在本地查询。

   图 5-25 查询外网 IP 地址

3. 修改远程连接的可访问IP地址后， 原来已经建立的链接依然有效， 当链接关闭后失效；新打开建立的链接只允许当前设置的IP进行访问。

5.5 使用 JupyterLab 开发模型

5.5.1 JupyterLab 操作流程

ModelArts支持通过JupyterLab工具在线打开Notebook，开发基于PyTorch、TensorFlow和MindSpore引擎的AI模型。具体操作流程如下图所示。

图 5-26 使用 JupyterLab 在线开发调试代码

1. 创建Notebook实例

   在ModelArts控制台中创建一个Notebook开发环境实例，选择要使用的AI框架。具体参见**创建Notebook实例**。
2. 使用JupyterLab打开Notebook实例。具体参见**打开JupyterLab**。
3. 准备训练数据和代码文件，上传到JupyterLab中。具体参见**上传本地文件至JupyterLab**。
4. 在JupyterLab中编写代码文件，并运行调试。具体参见**JupyterLab简介及常用操作**。
5. 在JupyterLab中直接调用ModelArts提供的SDK，创建训练作业，上云训练。

   调用SDK创建训练作业的操作请参见**调用SDK创建训练作业**。

5.5.2 JupyterLab 简介及常用操作

JupyterLab是一个交互式的开发环境，是Jupyter Notebook的下一代产品，可以使用它编写Notebook、操作终端、编辑MarkDown文本、打开交互模式、查看csv文件及图片等功能。

可以说，JupyterLab是开发者们下一阶段更主流的开发环境。JupyterLab具有和Jupyter Notebooks一样的组件，但支持更加灵活和更加强大的项目操作方式。

打开 JupyterLab

下面介绍如何从运行中的Notebook实例打开JupyterLab。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“开发环境 > Notebook”，进入新版Notebook管理页面。
2. 选择状态为“运行中”的Notebook实例，单击操作列的“打开”，访问JupyterLab。

   图 5-27 打开 Notebook 实例

3. 进入JupyterLab页面后，自动打开Launcher页面，如下图所示。您可以使用开源支持的所有功能，详细操作指导可参见**JupyterLab官网文档**。

图 5-28 JupyterLab 主页

说明

不同AI引擎的Notebook，打开后Launcher页面呈现的Notebook和Console内核及版本均不同，图5-28仅作为示例，请以实际控制台为准。

- Notebook：选择运行Notebook的一个内核，例如TensorFlow、python
- Console：可调出终端进行命令控制
- Other：可编辑其他文件

在 JupyterLab 中新建 ipynb 文件

进入JupyterLab主页后，可在“Notebook”区域下，选择适用的AI引擎，单击后将新建一个对应框架的ipynb文件。

由于每个Notebook实例选择的工作环境不同，其支持的AI框架也不同，下图仅为示例，请根据实际显示界面选择AI框架。

图 5-29 选择 AI 引擎并新建一个 ipynb 文件

新建的ipynb文件将呈现在左侧菜单栏中。

图 5-30 新建文件

新建文件并打开 Console

Console的本质为Python终端，输入一条语句就会给出相应的输出，类似于Python原生的IDE。

进入JupyterLab主页后，可在“Console”区域下，选择适用的AI引擎，单击后将新建一个对应框架的Notebook文件。

由于每个Notebook实例选择的工作环境不同，其支持的AI框架也不同，下图仅为示例，请根据实际显示界面选择AI框架。

图 5-31 选择 AI 引擎并新建一个 Console

文件创建成功后，将直接呈现Console页面。

图 5-32 新建文件（Console）

在 JupyterLab 中编辑文件

JupyterLab可以在同一个窗口同时打开几个Notebook或文件（如HTML、TXT、Markdown等），以页签形式展示。

JupyterLab的一大优点是，可以任意排版多个文件。在右侧文件展示区，您可以拖动打开文件，随意调整文件展示位置，可以同时打开多个文件。

图 5-33 多文件任意编排

当在一个Notebook中写代码时，如果需要实时同步编辑文件并查看执行结果，可以新建该文件的多个视图。

打开ipynb文件，然后单击菜单栏“File > New View for Notebook”，即可打开多个视图。

图 5-34 同一个文件的多个视图

JupyterLab的ipynb文件代码栏中输入代码，需要在代码前加!符号。

例如：安装外部库Shapely

```
!pip install Shapely
```

例如：查看PythonPath

```
!echo $PYTHONPATH
```

图 5-35 运行代码

自动停止及续期

在创建或启动Notebook时，如果启用了自动停止功能，则在JupyterLab的右上角会显示当前实例停止的倒计时（剩余时长），在倒计时结束前可以单击剩余时间进行续期。

图 5-36 自动停止

图 5-37 续期

设置字体

在IOS系统里打开ModelArts的Notebook时，需要设置字体，否则可能会显示异常。

图 5-38 打开高级设置编辑器

图 5-39 设置“fontFamily”设置为“Menlo”

JupyterLab 常用快捷键和插件栏

图 5-40 JupyterLab 常用快捷键和插件栏

表 5-4 快捷键说明

快捷键	说明
+	打开Launcher页面，可快速创建新的Notebook、Console或其他文件。
	创建文件夹。
	上传文件。
	刷新文件目录。
	Git插件，可连接此Notebook实例关联的Github代码库。

表 5-5 插件栏常用插件说明

插件	说明
	文件列表。单击此处，将展示此Notebook实例下的所有文件列表。
	罗列ModelArts Examples。 不同Region支持的Examples不一样，请以控制台实际界面为准。
	当前实例中正在运行的Terminal和Kernel。
	Git插件，可以方便快捷的使用Github代码库。
	快速启动命令。
	属性检查器。
	查看正在打开的文件页签。
	文档结构图。

图 5-41 导航栏按钮

表 5-6 导航栏按钮介绍

按钮	说明
File	新建、关闭、保存、重新加载、、重命名、导出、打印Notebook等功能。
Edit	编辑ipynb文件中代码块的相关操作，包括撤销、重做、剪切、复制、粘贴、选择、移动、合并、清除、查找代码块等。
View	查看视图相关操作。
Run	运行代码块相关操作，例如：运行选中代码块、一键运行所有代码块等。
Kernel	中断、重启、关闭、改变Kernel相关操作。
Git	Git插件相关操作，可以方便快捷的使用Github代码库。
Tabs	同时打开多个ipynb文件时，通过Tabs激活或选择文件。

按钮	说明
Settings	JupyterLab工具系统设置。
Help	JupyterLab工具自带的帮助参考。

图 5-42 ipynb 文件菜单栏中的快捷键

表 5-7 ipynb 文件菜单栏中的快捷键

快捷键	说明
	保存文件。
+	添加新代码块。
	剪切选中的代码块。
	复制选中的代码块。
	粘贴选中的代码块。
	执行选中的代码块。
	终止kernel。
	重启kernel。
Markdown	此处下拉框有4个选项，分别是： Code（写python代码），Markdown（写Markdown代码，通常用于注释），Raw（一个转换工具），-（不修改）。
	查看代码历史版本。
git	git插件，图标显示灰色表示当前Region不支持。
2 vCPU + 4 GiB	当前的资源规格。
PyTorch-1.4	单击可以选择Kernel。
	表示代码运行状态，变为实心圆时，表示代码在运行中。
	分享到AI Gallery。

资源监控

在使用过程中，如果想了解资源使用情况，可在右侧区域选择“Resource Monitor”，展示“CPU使用率”和“内存使用率”。

图 5-43 资源监控

5.5.3 上传文件至 JupyterLab

5.5.3.1 场景介绍

在AI开发过程中，如何将文件方便快速地上传到Notebook几乎是每个开发者都会遇到的问题。

ModelArts之前对文件直接上传到Notebook的大小限制是100MB，超过限制的文件无法直接上传；其次需要上传的文件并不都在本地，可能是GitHub的开源仓库，可能是类似开源数据集（**https://nodejs.org/dist/v12.4.0/node-v12.4.0-linux-x64.tar.xz**）这样的远端文件，也可能是存放在OBS中的文件，ModelArts之前无法将这些文件直接上传到Notebook中；在文件上传过程中，用户无法获得更多的信息，例如上传进度和速度。

ModelArts上传文件特性主要解决了以上三个问题，不仅提供了更多上传文件的功能满足用户需求，而且展示了更多文件上传的细节，提升了用户的体验。

当前的文件上传功能：

- 支持上传本地文件；
- 支持Clone GitHub开源仓库；
- 支持上传OBS文件；

- 支持上传远端文件；
- 将文件上传详情可视化。

5.5.3.2 上传本地文件至 JupyterLab

5.5.3.2.1 上传场景和入口介绍

Notebook的JupyterLab中提供了多种方式上传文件。

上传文件要求

- 对于大小不超过100MB的文件直接上传，并展示文件大小、上传进度及速度等详细信息。
- 对于大小超过100MB不超过5GB的文件可以使用OBS中转，系统先将文件上传OBS（对象桶或并行文件系统），然后从OBS下载到Notebook，上传完成后，会将文件从OBS中删除。
- 5GB以上的文件上传通过调用ModelArts SDK或者Moxing完成。
- 对于Notebook当前目录下已经有同文件名称的文件，可以覆盖继续上传，也可以取消。
- 支持多个文件同时上传，但不支持文件夹。可以将文件夹压缩成压缩包上传至Notebook后， 在Terminal中解压压缩包。
  ```
  unzip xxx.zip  #在xxx.zip压缩包所在路径直接解压
  ```
  解压命令的更多使用说明可以在主流搜索引擎中查找Linux解压命令操作。

前提条件

使用JupyterLab打开一个运行中的Notebook环境。

上传文件入口 1：将文件直接拖拽至浏览器窗口中

直接将文件拖拽到JupyterLab窗口左边的空白处上传。

上传文件入口 2：通过上传图标传文件

单击窗口上方导航栏的 ModelArts Upload Files按钮，打开文件上传界面，拖拽或者选择本地文件上传。

图 5-44 上传文件图标

图 5-45 上传文件窗口

5.5.3.2.2 上传本地小文件（100MB 以内）至 JupyterLab

对于大小不超过100MB的文件直接上传，并展示文件大小、上传进度及速度等详细信息。下图展示了上传文件的三种方式，请参考。

图 5-46 将文件直接拖拽至左侧浏览器窗口中上传

图 5-47 通过上传图标上传（拖曳本地文件上传）

图 5-48 通过上传图标上传（选择本地文件上传）

文件上传完成后给出提示。

图 5-49 上传成功

5.5.3.2.3 上传本地大文件（100MB~5GB）至 JupyterLab

对于大小超过100MB不超过5GB的文件可以使用OBS中转，系统先将文件上传至OBS（对象桶或或并行文件系统），然后从OBS下载到Notebook。下载完成后，ModelArts会将文件自动从OBS中删除。

例如，对于下面这种情况，可以通过“OBS中转”上传，操作步骤如图5-50所示。

图 5-50 通过 OBS 中转上传大文件

若使用OBS中转需要提供一个OBS中转路径，可以通过以下三种方式提供：

图 5-51 OBS 中转路径

说明

仅第一次单击“OBS中转”需要提供OBS中转路径，以后默认使用该路径直接上传，可以通过上传文件窗口左下角的设置按钮 更新OBS中转路径。

- 方式一：在输入框中直接输入有效的OBS中转路径，然后单击“确定”完成。

图 5-52 输入有效的 OBS 中转路径

- 方式二：打开OBS File Browser选择一个OBS中转路径，然后单击“确定”完成。

图 5-53 打开 OBS File Browser

- 方式三：单击“使用默认路径”完成。

图 5-54 使用默认路径上传文件

解压缩文件包

将文件以压缩包形式上传至Notebook JupyterLab后， 可在Terminal中解压缩文件包。

```
unzip xxx.zip  #在xxx.zip压缩包所在路径直接解压
```

解压命令的更多使用说明可以在主流搜索引擎中查找Linux解压命令操作。

5.5.3.2.4 上传本地超大文件（5GB 以上）至 JupyterLab

不支持在Notebook的JupyterLab中直接上传大小超过5GB的文件。

图 5-55 不支持直接上传大小超过 5GB 的文件

5GB以上的文件需要先从本地上传到OBS中，再在Notebook中调用ModelArts的Moxing接口或者SDK接口读写OBS中的文件。

图 5-56 在 Notebook 中上传下载大文件

具体操作如下：

1. 从本地上传文件至OBS。具体操作请参见**上传文件至OBS桶**。
2. 将OBS中的文件下载到Notebook，可以通过在Notebook中运行代码的方式完成数据下载，具体方式有2种，ModelArts的SDK接口或者调用MoXing接口。
   - 方法一：使用ModelArts SDK对接OBS数据服务的接口将OBS中的文件下载到Notebook后进行操作。

     示例代码：

     ```
     from modelarts.session import Session
     session = Session()
     session.obs.copy("obs://bucket-name/obs_file.txt","/home/ma-user/work/")
     ```
   - 方法二：使用**Moxing操作OBS文件**将OBS中的文件同步到Notebook后进行操作。

     ```
     import moxing as mox

     #下载一个OBS文件夹sub_dir_0，从OBS下载至Notebook
     mox.file.copy_parallel('obs://bucket_name/sub_dir_0', '/home/ma-user/work/sub_dir_0')
     #下载一个OBS文件obs_file.txt，从OBS下载至Notebook
     mox.file.copy('obs://bucket_name/obs_file.txt', '/home/ma-user/work/obs_file.txt')
     ```

     如果下载到Notebook中的是zip文件，在Terminal中执行下列命令，解压压缩包。

     ```
     unzip xxx.zip  #在xxx.zip压缩包所在路径直接解压
     ```

     代码执行完成后，参考**图5-57**打开terminal后执行**ls /home/ma-user/work**命令查看下载到Notebook中的文件。或者在Jupyter左侧导航中显示下载的文件，如果没有显示，请刷新后查看，如**图5-58**所示。

     图 5-57 打开 terminal

图 5-58 查看下载到 Notebook 中的文件

异常处理

通过OBS下载文件到Notebook中时，提示Permission denied。请依次排查：

- 请确保读取的OBS桶和Notebook处于同一站点区域，例如：都在华北-北京四站点。不支持跨站点访问OBS桶。具体请参见**如何查看OBS桶与ModelArts是否在同一区域**。
- 请确认操作Notebook的帐号有权限读取OBS桶中的数据。如没有权限，请参见**在Notebook中，如何访问其他帐号的OBS桶？**。

5.5.3.3 GitHub 开源仓库 Clone

在Notebook的JupyterLab中，支持从GitHub开源仓库Clone文件。

1. 通过JupyterLab打开一个运行中的Notebook。
2. 单击JupyterLab窗口上方导航栏的 ModelArts Upload Files按钮，打开文件上传窗口，选择左侧的 进入GitHub开源仓库Clone界面。

图 5-59 上传文件图标

图 5-60 进入 GitHub 开源仓库 Clone 界面

3. 输入有效的GitHub开源仓库地址后会展示该仓库下的文件及文件夹，说明用户输入了有效的的仓库地址，同时给出该仓库下所有的分支供选择，选择完成后单击“克隆”开始Clone仓库。

   GitHub开源仓库地址：https://github.com/jupyterlab/extension-examples

图 5-61 输入有效的 GitHub 开源仓库地址

4. Clone仓库的过程中会将进度展示出来。

图 5-62 Clone 仓库的过程

5. Clone仓库成功。

图 5-63 Clone 仓库成功

从GitHub开源仓库Clone文件完整操作步骤如**图5-64**所示。

图 5-64 从 GitHub 开源仓库 Clone 文件

异常处理

- Clone仓库失败。可能是网络原因问题。可以在JupyterLab的Terminal中通过执行 git clone https://github.com/jupyterlab/extension-examples.git 测试网络连通情况。

图 5-65 Clone 仓库失败

- 若克隆时遇到Notebook当前目录下已有该仓库，系统给出提示仓库名称重复，此时可以单击“覆盖”继续克隆仓库，也可以单击 取消。

图 5-66 提示仓库名称重复

5.5.3.4 上传 OBS 文件到 JupyterLab

在Notebook的JupyterLab中，支持将OBS中的文件下载到Notebook。

1. 通过JupyterLab打开一个运行中的Notebook。
2. 单击JupyterLab窗口上方导航栏的 ModelArts Upload Files按钮，打开文件上传窗口，选择左侧的 进入OBS文件上传界面。

图 5-67 上传文件图标

图 5-68 OBS 文件上传界面

3. 需要提供OBS文件路径，可以通过以下两种方式提供：
   - 方式一：在输入框中直接输入有效的OBS文件路径，然后单击“上传”开始传文件。

图 5-69 输入有效的 OBS 文件路径

说明

此处输入的是具体的OBS文件路径，不是文件夹的路径，否则会导致上传失败。

- 方式二：打开OBS File Browser选择OBS文件路径，然后单击“上传”，开始上传文件。

图 5-70 上传 OBS 文件

图 5-71 文件上传成功

OBS文件上传到Jupyter的完整操作也可参考图5-72。

图 5-72 OBS 文件上传

异常处理

提示文件上传失败。

图 5-73 文件上传失败

可能原因：

- OBS路径没有设置为具体的文件路径，设置成了文件夹。
- OBS中的文件设置了加密。请前往OBS控制台查看，确保该文件未加密。

- OBS桶和Notebook不在同一个区域。请确保读取的OBS桶和Notebook处于同一站点区域，例如：都在华北-北京四站点。不支持跨站点访问OBS桶。具体请参见**如何查看OBS桶与ModelArts是否在同一区域**。
- 没有该OBS桶的访问权限。请确认操作Notebook的帐号有权限读取OBS桶中的数据。如没有权限，请参见**在Notebook中，如何访问其他帐号的OBS桶？**。

5.5.3.5 上传远端文件至 JupyterLab

在Notebook的JupyterLab中，支持通过远端文件地址下载文件。操作步骤如图5-74。

要求：远端文件的URL粘贴在浏览器的输入框中时，可以直接下载该文件。

图 5-74 上传远端文件至 JupyterLab

说明

输入有效的远端文件URL后，系统会自动识别上传文件名称。

图 5-75 远端文件上传成功

异常处理

远端文件上传失败。可能是网络原因。请先在浏览器中输入该远端文件的URL地址，测试该文件是否能下载。

图 5-76 远端文件上传失败

5.5.4 从 JupyterLab 下载文件至本地

在JupyterLab中开发的文件，可以下载至本地。无论创建Notebook时存储位置选择默认存储或云硬盘EVS，此处的下载操作都一样。

- 不大于100MB的文件，可以直接从JupyterLab中下载到本地，具体操作请参见**从JupyterLab中下载不大于100MB的文件至本地**。
- 大于100MB的文件，需要先从JupyterLab上传到OBS，再通过OBS下载到本地，具体操作请参见**从JupyterLab中下载大于100MB的文件到本地**。

从 JupyterLab 中下载不大于 100MB 的文件至本地

在JupyterLab文件列表中，选择需要下载的文件，单击右键，在操作菜单中选择“Download”下载至本地。下载的目的路径，为您本地浏览器设置的下载目录。

图 5-77 下载文件

从 JupyterLab 中下载大于 100MB 的文件到本地

大于100MB的文件需要先从Notebook中上传到OBS，再从OBS下载到本地，具体操作如下：

1. 在Notebook中，新建一个大于100MB的“ipynb”文件，使用MoXing先将该文件从Notebook上传到OBS中，示例代码如下：
```
import moxing as mox
mox.file.copy('/home/ma-user/work/obs_file.txt', 'obs://bucket_name/obs_file.txt')
```
   其中“/home/ma-user/work/obs_file.txt”为文件在Notebook中的存储路径，“obs://bucket_name/obs_file.txt”为该文件上传到OBS的存储路径。
2. 使用OBS或ModelArts SDK将OBS中的文件下载到本地。
   - 方式一：使用OBS进行下载
   - 在OBS中，可以将样例中的“obs_file.txt”下载到本地。如果您的数据较多，推荐OBS Browser+下载数据或文件夹。使用OBS下载文件的操作指导参见**下载文件**
   - 方式二：使用ModelArts SDK进行下载
     i. 在您的本地环境**下载并安装ModelArts SDK**。
     ii. 完成ModelArts SDK的**Session鉴权**。
     iii. 将OBS中的文件下载到本地，详请参见**从OBS下载数据**。示例代码如下：
```
from modelarts.session import Session
session=Session(access_key='***',secret_key='***',project_id='***',region_name='***')
```

```
session.download_data(bucket_path="/bucket_name/obs_file.txt",path="/home/user/
obs_file.txt")
```

5.5.5 JupyterLab 插件

5.5.5.1 代码参数化插件

代码参数化插件可以降低Notebook案例的复杂度，用户无需感知复杂的源码，按需调整参数快速进行案例复现、模型训练等。该插件可用于定制Notebook案例，适用于比赛、教学等场景。

使用介绍

- 仅对Code cell类型新增了Edit Form和Add Form功能，如果cell类型是Markdown或者Raw类型则不支持。如下图所示：

图 5-78 查看 Code cell

- 打开新的代码后，需先Add Form，再Edit Form。

图 5-79 Code 类型的 cell 右键选项

Add Form 介绍

“Add Form”按钮会将Code cell水平拆分为两种编辑区域，左侧为代码区域，右侧为表单区域。单击表单右侧的“Edit”可修改默认标题。

图 5-80 两种编辑区域

Edit Form 介绍

“Edit Form”按钮有四个子选项，分别是“Add new form field”、“Hide code”、“Hide form”和“show all”四个按钮，下文介绍这四个选项的功能。

- “Add new form field”按钮支持新增“dropdown”、“input”和“slider”类型的表单。如图5-81所示。每新增一个字段，会分别在代码和表单区域中增加对应的变量，修改表单区域的值也会同时修改代码变量值。

  说明

  创建dropdown类型的表单时，“ADD Item”至少创建2项。如图5-82所示。

图 5-81 “dropdown”，“input”，“slider”的表单样式

图 5-82 创建“dropdown”类型的表单

图 5-83 删除表单

  - 表单字段类型为“dropdown”时，支持的变量类型为“raw”和“string”。
  - 表单字段类型为“input”时，支持的变量类型有“boolean”、“date”、“integer”、“number”、“raw”和“string”。
  - 表单字段类型为“slider”时，支持输入滑动条的最小值、最大值和步长。
- “Hide code”按钮会隐藏代码区域。
- “Hide form”按钮会隐藏表单区域。
- “Show all”按钮会同时展示code和form区域。

5.5.6 使用 ModelArts SDK

在Notebook中，通过使用ModelArts SDK，可以完成OBS管理、训练作业管理、模型管理以及在线服务管理。

ModelArts SDK使用请参见《**ModelArts SDK参考**》。

在Notebook中，已承载了登录用户的鉴权信息（AK/SK）和区域信息，因此SDK session鉴权时，无需输入参数即可完成session鉴权。

示例代码

- 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='/obs-bucket-name/src/',                           # 训练脚本目录
                boot_file='/obs-bucket-name/src/pytorch_sentiment.py',      # 训练启动脚本目录
                log_url='/obs-bucket-name/log/',                            # 训练日志目录
                hyperparameters=[
                        {"label":"classes",
                         "value": "10"},
                        {"label":"lr",
                         "value": "0.001"}
                        ],
                output_path='/obs-bucket-name/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_instance = estimator.fit(inputs='/obs-bucket-name/data/train/', wait=False,
job_name='my_training_job')
```

- 查询模型列表

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

- 查询服务详情
```
from modelarts.session import Session
from modelarts.model import Predictor
session = Session()
predictor_instance = Predictor(session, service_id="input your service_id")
predictor_info_resp = predictor_instance.get_service_info()
```

5.5.7 MindInsight 可视化作业

ModelArts支持在新版开发环境中开启MindInsight可视化工具。在开发环境中通过小数据集训练调试算法，主要目的是验证算法收敛性、检查是否有训练过程中的问题，方便用户调测。

MindInsight能可视化展现出训练过程中的标量、图像、计算图以及模型超参等信息，同时提供训练看板、模型溯源、数据溯源、性能调试等功能，帮助您在更高效地训练调试模型。MindInsight当前支持基于MindSpore引擎的训练作业。MindInsight相关概念请参考**MindSpore官网**。

MindSpore支持将数据信息保存到Summary日志文件中，并通过可视化界面MindInsight进行展示。

前提条件

使用MindSpore引擎编写训练脚本时，为了保证训练结果中输出Summary文件，您需要在脚本中添加收集Summary相关代码。

将数据记录到Summary日志文件中的具体方式请参考**收集Summary数据**。

注意事项

- 在开发环境跑训练任务，在开发环境使用MindInsight，要求先启动MindInsight，后启动训练进程。
- 仅支持单机单卡训练。
- 运行中的可视化作业不单独计费，当停止Notebook实例时，计费停止。
- Summary文件如果存放在OBS中，由OBS单独收费。任务完成后请及时停止Notebook实例，清理OBS数据，避免产生不必要的费用。

在开发环境中创建 MidnightInsight 可视化作业流程

Step1 创建开发环境并在线打开

Step2 上传Summary数据

Step3 启动MindInsight

Step4 查看训练看板中的可视化数据

Step1 创建开发环境并在线打开

在ModelArts控制台，进入“开发环境> Notebook”新版页面，创建MindSpore引擎的开发环境实例。创建成功后，单击开发环境实例操作栏右侧的“打开 ”，在线打开运行中的开发环境。

MindInsight可视化训练作业，当前支持的镜像如下，请根据实际局点支持的镜像和资源规格选择使用。

- mindspore1.2.0版本，CPU/GPU规格的资源类型。
- mindspore1.5.x以上版本，Ascend规格的资源类型。

Step2 上传 Summary 数据

在开发环境中使用MindInsight可视化功能，需要用到Summary数据。

Summary数据可以直接传到开发环境的这个路径下/home/ma-user/work/，也可以放到OBS并行文件系统中。

- Summary数据上传到Notebook路径/home/ma-user/work/下的方式，请参见**上传数据至Notebook**。
- Summary数据如果是通过OBS并行文件系统挂载到Notebook中，请将模型训练时产生的Summary文件先上传到OBS并行文件系统，并确保OBS并行文件系统与ModelArts在同一区域。在Notebook中启动MindInsight时，Notebook会自动从挂载的OBS并行文件系统目录中读取Summary数据。

Step3 启动 MindInsight

在开发环境的JupyterLab中打开MindInsight有多种方法。可根据使用习惯选择。

图 5-84 JupyterLab 中打开 MindInsight 的方法

方式1：

1. 单击入口1

，进入JupyterLab开发环境，并自动创建.ipynb文件。

2. 在对话框中输入MindInsight相应命令，即可展示界面。

```
%reload_ext mindinsight
%mindinsight --port {PORT} --summary-base-dir {SUMMARY_BASE_DIR}
```

参数解释：

- --port {PORT}：指定Web可视化服务端口。可以不设置，默认使用8080端口。如果8080端口被占用了，需要在1~65535任意指定一个端口。
- --summary-base-dir {SUMMARY_BASE_DIR}：表示数据在开发环境中的存储路径。
  - 开发环境本地路径：“./work/xxx”（相对路径）或/home/ma-user/work/xxx（绝对路径）
  - OBS并行文件系统桶的路径：obs://xxx/

```
例如：
#Summary数据如果是在开发环境的这个路径下/home/ma-user/work/，执行下面这条命令
%mindinsight --summary-base-dir /home/ma-user/work/xxx
或者
#Summary数据如果是存在OBS并行文件系统中，执行下面这条命令，开发环境会自动挂载该OBS并行文件系统路径并读取数据
%mindinsight --summary-base-dir obs://xxx/
```

图 5-85 MindInsight 界面（1）

方式2：

单击入口2

，直接进入MindInsight可视化界面。

默认读取路径/home/ma-user/work/

当存在两个以及以上工程的log时，界面如下。通过Runs下选择查看相对应的log。

图 5-86 MindInsight 界面（2）

方式3：

1. 单击入口3 ，在搜索框中输入“MindInsight”，再单击“Create a new MindInsight”。

   图 5-87 Create a new MindInsight

2. 填写需要查看的Summary数据路径，或者OBS并行文件系统桶的路径，单击CREATE。
   - 开发环境本地路径：./summary（相对路径）或/home/ma-user/work/summary（绝对路径）
   - OBS并行文件系统的路径：obs://xxx/

图 5-88 输入 Summary 数据路径

图 5-89 MindInsight 界面（3）

说明

方式2及方式3最多可以启动10个MindInsight实例。

方式4：

单击入口4

，输入并执行以下命令，但启动后不能显示UI界面。

```
mindinsight start --summary-base-dir ./summary_dir
```

图 5-90 Terminal 方式打开 MindInsight

Step4 查看训练看板中的可视化数据

训练看板是MindInsight的可视化组件的重要组成部分，而训练看板的标签包含：标量可视化、参数分布图可视化、计算图可视化、数据图可视化、图像可视化和张量可视化等。

更多功能介绍请参见MindSpore官网资料：**查看训练看板中可视的数据**。

相关操作

关闭MindInsight方式如下：

- 方式1：在开发环境JupyterLab中的.ipynb文件窗口中输入命令，关闭MindInsight。端口号在**启动MindInsight**中设置，默认使用8080，需要替换为实际开启MindInsight时的端口。

```
!mindinsight stop --port 8080
```

- 方式2：单击下方 按钮进入MindInsight实例管理界面，该界面记录了所有启动的MindInsight实例，单击对应实例后面的SHUT DOWN即可停止该实例。

图 5-91 单击 SHUT DOWN 停止实例

- 方式3：单击下方红框中的的 按钮可以关闭所有启动的MindInsight实例。

图 5-92 关闭所有启动的 MindInsight 实例

- 方式4（不推荐）：直接在JupyterLab中上关闭MindInsight窗口，此方式仅是关闭MindInsight可视化窗口，并未关闭后台。

5.5.8 TensorBoard 可视化作业

ModelArts支持在开发环境中开启TensorBoard可视化工具。TensorBoard是TensorFlow的可视化工具包，提供机器学习实验所需的可视化功能和工具。

TensorBoard能够有效地展示TensorFlow在运行过程中的计算图、各种指标随着时间的变化趋势以及训练中使用到的数据信息。

前提条件

为了保证训练结果中输出Summary文件，在编写训练脚本时，您需要在脚本中添加收集Summary相关代码。

TensorFlow引擎的训练脚本中添加Summary代码，具体方式请参见**TensorFlow官方网站**。

注意事项

- TensorBoard可视化训练作业，当前仅支持基于TensorFlow2.1、Pytorch1.4/1.8以上版本镜像，CPU/GPU规格的资源类型。请根据实际局点支持的镜像和资源规格选择使用。
- 运行中的可视化作业不单独计费，当停止Notebook实例时，计费停止。
- Summary文件数据如果存放在OBS中，由OBS单独收费。任务完成后请及时停止Notebook实例，清理OBS数据，避免产生不必要的费用。

在开发环境中创建 TensorBoard 可视化作业流程

Step1 创建开发环境并在线打开

Step2 上传Summary数据

Step3 启动TensorBoard

Step4 查看训练看板中的可视化数据

Step1 创建开发环境并在线打开

在ModelArts控制台，进入“开发环境> Notebook”新版页面，创建TensorFlow或者PyTorch镜像的开发环境实例。创建成功后，单击开发环境实例操作栏右侧的“打开 > 打开Notebook”，在线打开运行中的开发环境。

TensorBoard可视化训练作业，当前仅支持基于TensorFlow2.1、Pytorch1.4/1.8以上版本镜像，CPU/GPU规格的资源类型。请根据实际局点支持的镜像和资源规格选择使用。

Step2 上传 Summary 数据

在开发环境中使用TensorBoard可视化功能，需要用到Summary数据。

Summary数据可以直接传到开发环境的这个路径下/home/ma-user/work/，也可以放到OBS并行文件系统中。

- Summary数据上传到Notebook路径/home/ma-user/work/下的方式，请参见**上传数据至Notebook**。
- Summary数据如果是通过OBS并行文件系统挂载到Notebook中，请将模型训练时产生的Summary文件先上传到OBS并行文件系统，并确保OBS并行文件系统与ModelArts在同一区域。在Notebook中启动TensorBoard时，Notebook会自动从挂载的OBS并行文件系统目录中读取Summary数据。

Step3 启动 TensorBoard

在开发环境的JupyterLab中打开TensorBoard有多种方法。可根据使用习惯选择。

图 5-93 JupyterLab 中打开 TensorBoard 的方法

方式1：

1. 单击入口1，进入JupyterLab开发环境中，并自动创建.ipynb文件。
2. 在对话框中输入TensorBoard相应命令，即可展示界面。

```
%reload_ext tensorboard
%tensorboard  --port {PORT} --logdir {BASE_DIR}
```

参数解释：

- --port {PORT}：指定Web可视化服务端口。可以不设置，默认使用8080端口。如果8080端口被占用了，需要在1~65535任意指定一个端口。
- --logdir {BASE_DIR}：表示数据在开发环境中的存储路径

- 开发环境本地路径：“./work/xxx”（相对路径）或/home/ma-user/work/xxx（绝对路径）
- OBS并行文件系统的路径：obs://xxx/

```
例如：
#Summary数据如果是在开发环境的这个路径下/home/ma-user/work/，执行下面这条命令。
%tensorboard  --port {PORT} --logdir /home/ma-user/work/xxx
或者
#Summary数据如果是存在OBS并行文件系统中，执行下面这条命令，开发环境会自动挂载该OBS并行文件系统路径并读取数据。
%tensorboard  --port {PORT} --logdir obs://xxx/
```

图 5-94 TensorBoard 界面（1）

方式2：

单击入口2

，直接进入TensorBoard可视化界面。

默认读取路径/home/ma-user/work/

当存在两个以及以上工程的log时，界面如下。可以在左侧Runs栏目下选择查看相对应的log。

图 5-95 TensorBoard 界面（2）

方式3：

1. 单击入口3 ，在搜索框中输入“TensorBoard”，再单击“Create a new TensorBoard”。

   图 5-96 Create a new TensorBoard

2. 填写需要查看的Summary数据路径，或者OBS并行文件系统桶的路径。
   - 开发环境本地路径：./summary（相对路径）或/home/ma-user/work/summary（绝对路径）

– OBS并行文件系统桶的路径：obs://xxx/

图 5-97 输入 Summary 数据路径

图 5-98 TensorBoard 界面（3）

方式4：

单击入口4

，输入命令执行，但启动后不能显示UI界面。

```
tensorboard --logdir ./log
```

图 5-99 Terminal 方式打开 TensorBoard

```
sh-4.4$pwd
/home/ma-user
sh-4.4$tensorboard --logdir ./log
2021-10-18 20:34:53.586976: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libnvinfer.so.6
2021-10-18 20:34:53.589272: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libnvinfer_plugin.so.6
Serving TensorBoard on localhost; to expose to the network, use a proxy or pass --bind_all
TensorBoard 2.1.1 at http://localhost:6006/ (Press CTRL+C to quit)
```

Step4 查看训练看板中的可视化数据

训练看板是TensorBoard的可视化组件的重要组成部分，而训练看板的标签包含：标量可视化、图像可视化和计算图可视化等。

更多功能介绍请参见TensorBoard官网资料：**开始使用 TensorBoard | TensorFlow (google.cn)**。

相关操作

关闭TensorBoard方式如下：

- 方式1：在开发环境JupyterLab中的.ipynb文件窗口中输入命令，关闭TensorBoard。PID在启动界面有提示或者通过 ps -ef | grep tensorboard 查看。
  ```
  !kill PID
  ```
- 方式2：单击下方 ，进入TensorBoard实例管理界面，该界面记录了所有启动的TensorBoard实例，单击对应实例后面的SHUT DOWN即可停止该实例。

  图 5-100 单击 SHUT DOWN 停该实例

- 方式3：单击下方红框中的的 按钮可以关闭所有启动的TensorBoard实例。

  图 5-101 关闭所有启动的 TensorBoard 实例

- 方式4（不推荐）：直接在JupyterLab中上关闭TensorBoard窗口，此方式仅关闭可视化窗口，并未关闭后台。

5.6 使用本地 IDE 开发模型

5.6.1 本地 IDE 操作流程

ModelArts支持通过本地IDE环境远程连接到Notebook中，开发基于Pytorch、Tensorflow和MindSpore引擎的AI模型。具体操作流程如下图所示。

图 5-102 使用本地 IDE 开发流程

1. 配置本地IDE

   在用户的PC端配置本地IDE环境。

   支持通过PyCharm和VS Code 2类本地IDE连接云上Notebook。PyCharm和VS Code可以使用插件自动化配置，也可以手工配置。

2. **创建Notebook实例**

   在ModelArts控制台中创建一个Notebook开发环境实例，选择要使用的AI框架，并开启SSH远程开发功能。

3. 使用本地IDE远程连接到ModelArts的开发环境中。
4. **上传数据和代码至开发环境中**，进行代码调试。
   - 代码直接拷贝至本地IDE中即可，本地IDE中会自动同步至云上开发环境。
   - 不大于500MB数据量直接拷贝至本地IDE中即可。
   - 大于500MB数据量请先上传到OBS中，从OBS上传到默认存储EFS或云硬盘EVS。
5. 将调试好的训练脚本和用于训练的数据集上传至OBS目录。
6. 提交训练作业。提交训练作业有如下2种方式：
   - 在本地IDE中提交训练作业
     - 可以基于PyCharm ToolKit直接提交训练作业，具体参考**使用PyCharm ToolKit提交训练作业**。
     - 也可以通过调用ModelArts提供的SDK，创建训练作业，上云训练，调用SDK创建训练作业的操作请参见**调用SDK创建训练作业**。
   - 在ModelArts的Console控制台页面中提交训练作业，具体参考**创建训练作业**。

5.6.2 配置本地 IDE（PyCharm ToolKit 连接）

ModelArts提供了一个PyCharm插件工具**PyCharm ToolKit**，协助用户完成代码上传、提交训练作业、将训练日志获取到本地展示等，用户只需要专注于本地的代码开发即可。

前提条件

本地已安装2019.2及以上版本的PyCharm专业版。SSH远程开发功能只限PyCharm专业版。

Step1 下载并安装 PyCharm ToolKit

插件工具包PyCharm ToolKit集成在ModelArts管理控制台中，请先下载并安装PyCharm ToolKit插件。

具体下载和安装过程请参见**安装PyCharm ToolKit工具插件包**。

Step2 登录插件

请创建插件访问密钥AK/SK（**参考链接**），并使用访问密钥完成登录认证（**参考链接**），使得ToolKit与公有云完成连接。

Step3 创建 Notebook 实例

创建一个Notebook实例，并开启远程SSH开发，配置远程访问IP白名单。该实例状态必须处于“运行中”，具体参见**创建Notebook实例**章节。

Step4 插件自动化配置

1. 在本地的PyCharm开发环境中，单击“ModelArts > Connect To Remote > Remote Config”，配置插件。

   图 5-103 配置插件

   说明

   此处如果看不到Connect to Remote选项，请先参考**创建Notebook实例**章节，创建Notebook实例，并开启该实例的SSH远程开发功能。

   也可能是PyCharm ToolKit的版本不正确，请按照文档要求下载新版本的PyCharm ToolKit。

   下载前请先清除浏览器缓存，如果之前下载过老版本的PyCharm ToolKit，浏览器会有缓存，可能会导致新版本下载失败。

2. 此时，会出现该帐号已创建的所有包含SSH功能的Notebook列表，下拉进行选择对应Notebook。

图 5-104 Notebook 列表

- KeyPair: 需要选择保存在本地的Notebook对应的keypair认证。即创建Notebook时创建的密钥对文件，创建时会直接保存到浏览器默认的下载文件夹中。
- PathMappings: 该参数为本地IDE项目和Notebook对应的同步目录，默认为/home/ma-user/work/project名称，可根据自己实际情况更改。

3. 单击“Apply”，配置完成后，重启IDE生效。

   重启后初次进行update python interpreter需要耗费20分钟左右。

Step5 使用插件连接云上 Notebook

与Notebook断开连接的状态下，单击Notebook名称，根据提示启动本地IDE与Notebook的连接(默认启动时间4小时)。

图 5-105 启动连接 Notebook

连接状态下，单击Notebook名称，根据提示断开本地IDE与云上Notebook的连接。

图 5-106 停止连接 Notebook

Step6 同步上传本地文件至 Notebook

本地文件中的代码直接拷贝至本地IDE中即可，本地IDE中会自动同步至云上开发环境。

初始化同步：

在本地IDE的Project目录下，单击右键，选择“Deployment”，单击“Upload to xxx”（Notebook名称），将本地工程文件上传至指定的Notebook。

图 5-107 同步本地文件至 Notebook

后续同步:

只需修改代码后保存（ctrl+s），即可进行自动同步。

插件安装完成后在本地IDE中开启了“Automatic Upload”，本地目录中的文件会自动上传至云端开发环境Notebook。如果未开启，请参考下图开启自动上传。

图 5-108 开启自动上传

Step7 远程调试

单击本地IDE右下角interpreter，选择Notebook的python解释器。

图 5-109 选择 Python 解释器

像本地运行代码一样，直接单击运行按钮运行代码即可，此时虽然是在本地IDE点的运行按钮，实际上运行的是云端Notebook里的代码，日志可以回显在本地的日志窗口。

图 5-110 查看运行日志

也可以单击本地IDE右上角的Run/Debug Configuration按钮来设置运行参数。

图 5-111 设置运行参数（1）

选择远程连接到云上开发环境实例对应的Python解释器。

图 5-112 设置运行参数（2）

当需要调试代码时，可以直接打断点，然后使用debug方式运行程序。

图 5-113 使用 debug 方式运行程序

此时可以进入debug模式，代码运行暂停在该行，且可以查看变量的值。

图 5-114 Debug 模式下查看变量值

5.6.3 配置本地 IDE（PyCharm 手动连接）

本地IDE环境支持Pycharm和VS Code。通过简单配置，即可用本地IDE远程连接到ModelArts的Notebook开发环境中，调试和运行代码。

本章节介绍基于PyCharm环境访问Notebook的方式。

前提条件

- 本地已安装2019.2及以上版本的PyCharm专业版。SSH远程调试功能只限PyCharm专业版。
- 创建一个Notebook实例，并开启远程SSH开发。该实例状态必须处于“运行中”，具体参见**创建Notebook实例**章节。
- 在Notebook实例详情页面获取开发环境IP地址（例如：dev-modelarts-cnnorth4.huaweicloud.com）和端口号。

  图 5-115 Notebook 实例详情页面

- 准备好密钥对。

  密钥对在用户第一次创建时，自动下载，之后使用相同的密钥时不会再有下载界面（用户一定要保存好），或者每次都使用新的密钥对。

Step1 配置 SSH

1. 在本地的PyCharm开发环境中，单击File -> Settings -> Tools -> SSH Configurations，单击+号，增加一个SSH连接配置。
   - Host: 云上开发环境的IP地址，即在开发环境实例页面远程访问模块获取的IP地址。例如：dev-modelarts-cnnorth4.huaweicloud.com

- Port: 云上开发环境的端口，即在开发环境实例页面远程访问模块获取的端口号。
- User name: 固定为ma-user
- Authentication type: Key pair方式
- Private key file: 存放在本地的云上开发环境私钥文件，即在创建开发环境实例时创建并保存的密钥对文件。

2. 单击 将连接重命名，可以自定义一个便于识别的名字，单击OK。
3. 配置完成后，单击Test Connection测试连通性。
4. 选择Yes，显示Successfully connected表示网络可以连通，单击OK。
5. 在最下方再单击OK保存配置。

图 5-116 配置 SSH

Step2 获取开发环境预置虚拟环境路径

1. 单击“Tools > Start SSH Session”，则可连接到云端开发环境内。
2. 执行如下命令可在/home/ma-user/下面的README文件查看当前环境内置的Python虚拟环境。
```
cat /home/ma-user/README
```
3. 执行source命令可以切换到具体的Python环境中。
4. 执行which python查看python路径并拷贝出来，以备后续配置云上Python Interpreter使用。

图 5-117 获取开发环境预置虚拟环境路径

Step3 配置云上 Python Interpreter

1. 单击“File > Settings > Project：PythonProject > Python Interpreter”，单击设置图标 ，再单击“Add”，添加一个新的interpreter。
2. 选择“Existing server configuration”，在下拉菜单中选择上一步配置好的SSH configuration，单击“Next”。
3. 配置Python Interpreter
   - Interpreter：填写第一步拷贝的python路径，例如：/home/ma-user/anaconda3/envs/Pytorch-1.0.0/bin/python

     如果路径为~/anaconda3/envs/Pytorch-1.0.0/bin/python 把~替换为/home/ma-user即可。
   - Sync folders: 需要配置本地的工程目录文件同步到云上开发环境环境中的某个目录，推荐配置为/home/ma-user下的某个目录中（其他目录可能没有访问权限），例如/home/ma-user/work/projects。
4. 单右侧文件夹图标!，勾选上“Automatically upload”选项，以便于本地修改的文件自动上传到容器环境中。
5. 单击“Finish”，结束配置。

   可以看到本地的工程文件已经自动往云上环境上传了。后续本地的文件每修改一下，都会自动的同步到云上的环境中。

   右下角可以看到当前的Interpreter为Remote Interpreter。

图 5-118 配置云上 Python Interpreter

Step4 云上环境依赖库安装

在进入开发环境后，可以使用不同的虚拟环境，例如TensorFlow、PyTorch等，但是实际开发中，通常还需要安装其他依赖包，此时可以通过terminal连接到环境里操作。

单击工具栏“Tools >Start SSH session”，选择SSH Configuration中配置的开发环境。可以执行pip install安装所需要的包。

Step5 在开发环境中调试代码

由于已经连接至云端开发环境，此时可以方便的在本地PyCharm中编码、调测并运行。运行实际环境为云上开发环境，资源为云上昇腾AI处理器资源。可以做到本地编写修改代码，直接在云上环境运行。

像本地运行代码一样，直接单击运行按钮运行代码即可，此时虽然是在本地IDE单击的运行按钮，实际上运行的是云端开发环境里的代码，日志可以回显在本地的日志窗口。

图 5-119 调试代码

也可以单击右上角的Run/Debug Configuration来设置运行的参数。

图 5-120 设置运行参数

当需要调试代码时，可以直接打断点，然后使用debug方式运行程序。

图 5-121 代码打断点

图 5-122 Debug 方式调试

此时可以进入debug模式，代码运行暂停在该行，且可以查看变量的值。

图 5-123 Debug 模式

使用debug方式调试代码的前提是本地的代码和云端的代码是完全一致的，如果不一致可能会导致在本地打断点的行和实际运行时该行的代码并不一样，会出现意想不到的错误。

因此在配置云上Python Interpreter时，推荐选择Automatically upload选项，以保证本地的文件修改能自动上传到云端。如果没有选择自动上传，则本地代码修改完后，也可以参考**Step6 同步上传本地文件至Notebook**手动上传目录或代码。

5.6.4 配置本地 IDE（VS Code ToolKit 连接）

本节介绍如何在本地安装Microsoft Visual Studio Code（简称VS Code）软件及ModelArts提供了一个VS Code插件工具，协助用户完成SSH远程连接Notebook。

前提条件

- 下载并安装VS Code。

  Windows用户直接单击此处下载：**https://update.code.visualstudio.com/1.57.1/win32-x64-user/stable**。

  其他系统用户的下载地址: **https://code.visualstudio.com/updates/v1_57**

  图 5-124 VS Code 的下载位置

  VS Code版本要求：建议用户使用VS Code 1.57.1版本或者最新版本进行远程连接。

图 5-125 VS Code 安装指导

- 设置禁止自动升级VS Code版本。
  a. 执行Ctrl+Shift+P
  b. 搜Open Settings（JSON）
  c. 在配置文件settings.json中添加如下参数
     "update.mode": "manual",
- 用户本地PC或服务器的操作系统中建议先安装Python环境，详见**VSCode官方指导**。
- 准备好密钥对。
  密钥对在用户第一次创建时，自动下载，之后使用相同的密钥时不会再有下载界面（用户一定要保存好），或者每次都使用新的密钥对。

Step1 下载 VS Code 插件

VS Code的插件工具包VSCode-ToolKit集成在ModelArts管理控制台中，请先下载并安装VSCode插件工具包，**下载链接**。

Step2 安装 VS Code 插件

1. 在本地的VS Code开发环境中，如**图5-126**所示，选择本地的VS Code插件并单击“安装”。

图 5-126 安装 VS Code 插件

2. 安装过程预计1~2分钟，如图5-127所示，请耐心等待。

图 5-127 安装过程

3. 安装完成后，系统右下角提示安装完成，导航左侧出现ModelArts图标 和SSH远程连接图标 ，表示VS Code插件安装完成。
当前网络不佳时SSH远程连接插件可能未安装成功，此时无需操作，之后在实例连接时点击Install and Reload即可.

图 5-128 Install and Reload

图 5-129 安装完成提示

图 5-130 安装完成

Step3 登录 VS Code 插件

1. 在本地的VS Code开发环境中，单击ModelArts图标，单击“User Settings”，配置用户登录信息。

图 5-131 登录插件

2. 输入如下用户登录信息，单击“登录”。
   - Name：自定义用户名，仅用于VS Code页面展示，不与任何华为云用户关联。
   - AK、SK：在华为云“帐号中心 > 我的凭证 > 访问密钥”中创建访问密钥，获取AK、SK（**参考链接**）。
   - 选择站点：此处的站点必须和远程连接的Notebook在同一个站点，否则会导致连接失败。

   也可以切换登录方式，采用如下图所示方式输入用户信息后，Ctrl+S保存。

图 5-132 配置用户登录信息

3. 登录成功后显示Notebook实例列表。

图 5-133 登录成功

Step4 创建 Notebook 实例

创建一个Notebook实例，并开启远程SSH开发，具体参见**创建Notebook实例**。

Step5 连接 Notebook 实例

1. 在本地的VS Code开发环境中，右键单击实例名称，单击“Connect to Instance”，启动并连接Notebook实例。

   Notebook实例状态处于“运行中”或“停止”状态都可以，如果Notebook实例是停止状态，连接Notebook时，VS Code插件会先启动实例再去连接。

   图 5-134 连接 Notebook 实例

   或者单击实例名称，在VS Code开发环境中显示Notebook实例详情页，单击“连接”，系统自动启动该Notebook实例并进行远程连接。

   图 5-135 查看 Notebook 实例详情页

2. 第一次连接Notebook时，系统右下角会提示需要先配置密钥文件。选择本地密钥pem文件（放在用户家目录下的.ssh文件夹下），根据系统提示单击“OK”。

图 5-136 配置密钥文件

3. 单击“确定”后，插件自动连接远端Notebook实例。首次连接大约耗时1~2分钟，取决于本地的网络情况。VS Code环境左下角显示类似下图即为连接成功。

图 5-137 连接成功

Step6 上传本地代码到云端开发环境

1. 在VS Code界面，单击“File > OpenFolder”打开云端路径。

图 5-138 Open Floder

2. 选择要打开的路径，单击“OK”。

图 5-139 选择文件路径

3. 此时，会在IDE左侧出现该开发环境下的目录结构，把想要上传的代码及其他文件直接拖拽至对应得文件夹内即完成本地代码上传至云端。

Step7 远程调试代码

在VS Code中打开要执行的代码文件，在执行代码之前需要选择合适的Python版本路径，单击下方默认的Python版本路径，此时在上方会出现该远程环境上所有的python版本，选择自己需要的版本即可。

图 5-140 选择 Python 版本

- 对于打开的代码文件，单击run按钮，即可执行，可以在下方的Terminal中看到代码输出信息。
- 如果要对代码进行debug调试，步骤如下：
  a. 单击左侧“Run > Run and Debug”。
  b. 选择当前打开的默认的python代码文件进行调试。
  c. 对当前代码进行打断点，即在代码左侧进行单击，就会出现小红点。
  d. 此时，即可按照正常的代码调试步骤对代码调试，在界面左边会显示debug信息，代码上方有相应的调试步骤。

相关操作

卸载VS Code插件操作如图5-141所示。

图 5-141 卸载 VS Code 插件

VSCode 使用常见问题

VSCode中查看远端日志

使用VSCode调试代码时不能进入源码

使用VSCode提交代码时弹出对话框提示用户名和用户邮箱配置错误

VSCode连接远端Notebook时报错如“XHR failed”

安装远端插件时不稳定，需尝试多次

Notebook实例重新启动后，需要删除本地known_hosts才能连接

更多VS Code使用相关FAQ，请参见**FAQ**。

5.6.5 配置本地 IDE（VS Code 手动连接）

本地IDE环境支持Pycharm和VS Code。通过简单配置，即可用本地IDE远程连接到ModelArts的Notebook开发环境中，调试和运行代码。

本章节介绍基于VS Code环境访问Notebook的方式。

前提条件

- 下载并安装VS Code。

  Windows用户直接单击此处下载：**https://update.code.visualstudio.com/1.57.1/win32-x64-user/stable**。

  其他系统用户的下载地址: **https://code.visualstudio.com/updates/v1_57**

  图 5-142 VS Code 的下载位置

  VS Code版本要求：建议用户使用VS Code 1.57.1版本或者最新版本进行远程连接。

图 5-143 VS Code 安装指导

- 用户本地PC或服务器的操作系统中建议先安装Python环境，详见**VSCode官方指导**。
- 创建一个Notebook实例，并开启远程SSH开发。该实例状态必须处于“运行中”，具体参见**创建Notebook实例**章节。
- 在Notebook实例详情页面获取开发环境访问地址（例如：dev-modelarts-cnnorth4.huaweicloud.com）和端口号。

图 5-144 Notebook 实例详情页面

- 准备好密钥对。

  密钥对在用户第一次创建时，自动下载，之后使用相同的密钥时不会再有下载界面（用户一定要保存好），或者每次都使用新的密钥对。

Step1 添加 Remote-SSH 插件

在本地的VS Code开发环境中，单击左侧列表的Extensions图标选项 ，在搜索框中输入SSH，单击Remote-SSH插件的install按钮，完成插件安装。

图 5-145 添加 Remote-SSH 插件

Step2 配置 SSH

1. 在本地的VS Code开发环境中，单击左侧Remote Explorer按钮，在上方的下拉列表中选择“SSH Target”，再单击页面上的设置按钮，此时会出现SSH配置文件路径。

图 5-146 配置 SSH Targets 页面

2. 单击列表中出现的SSH路径按钮，打开config文件，进行配置。

图 5-147 配置 SSH Config 文件

```
HOST remote-dev
    hostname <instance connection host>
    port <instance connection port>
    user ma-user
    IdentityFile ~/.ssh/test.pem
    UserKnownHostsFile=/dev/null
    StrictHostKeyChecking no
```

- Host: 自定义设置的云上开发环境名称
- HosName: 云上开发环境的访问地址，即在开发环境实例页面远程访问模块获取的访问地址。例如：dev-modelarts-cnnorth4.huaweicloud.com
- Port: 云上开发环境的端口，即在开发环境实例页面远程访问模块获取的端口号。
- User：登录用户只支持ma-user进行登录
- IdentityFile: 存放在本地的云上开发环境私钥文件，即前提条件**•准备好密钥对。**中准备的密钥对。

3. 配置云上开发环境系统平台，单击“File > Preference > Settings > Extensions > Remote-SSH”，在“Remote Platform”中，单击“Add Item”选项，设置“Item”和“Value”，配置完成后，单击“OK”。

图 5-148 配置云上开发环境系统平台

“Item”：在SSH Config中配置的Host的名称。

“Value”：在下拉选择框中选择远端开发环境平台。

4. 再回到SSH Targets页面，单击右侧的Connect to Host in New Window按钮，该按钮会显示远程开发环境名称，选中并打开。

图 5-149 打开开发环境

在新打开的页面中，看到下图所示界面，即表示连接成功。

图 5-150 开发环境远程连接成功

图 5-151 完整配置示例

Step3 安装云端 Python 插件

在新打开的的VS Code界面，单击左侧列表的Extensions选项 ，在搜索框中输入Python，在下拉列表中单击“Install”进行安装。

图 5-152 安装云端 Python 插件

如果安装云端的Python插件不成功时，建议通过离线包的方式安装。具体操作请参见**安装远端插件时不稳定，需尝试多次**。

Step4 云上环境依赖库安装

在进入容器环境后，可以使用不同的虚拟环境，例如TensorFlow、PyTorch等，但是实际开发中，通常还需要安装其他依赖包，此时可以通过Terminal连接到环境里操作。

1. 在VS Code环境中，执行Ctrl+Shift+P。
2. 搜Python: Select Interpreter，选择对应的Python环境。

3. 单击页面上方的“Terminal > New Terminal”，此时打开的的命令行界面即为远端容器环境命令行。
4. 进入引擎后，通过执行如下命令安装依赖包。
```
pip install spacy
```

Step5 上传本地代码到云端开发环境

1. 在VS Code界面，单击“File > OpenFolder”打开云端路径。

图 5-153 Open Floder

2. 选择要打开的路径，单击“OK”。

图 5-154 选择文件路径

3. 此时，会在IDE左侧出现该开发环境下的目录结构，把想要上传的代码及其他文件直接拖拽至对应得文件夹内即完成本地代码上传至云端。

图 5-155 上传本地代码

Step6 远程调试代码

在VS Code中打开要执行的代码文件，在执行代码之前需要选择合适的Python版本路径，单击下方默认的Python版本路径，此时在上方会出现该远程环境上所有的python版本，选择自己需要的版本即可。

图 5-156 选择 Python 版本

对于打开的代码文件，单击run按钮，即可执行，可以在下方的Terminal中看到代码输出信息。

图 5-157 运行代码文件

如果要对代码进行debug调试，步骤如下：

1. 单击左侧“Run > Run and Debug”。
2. 选择当前打开的默认的python代码文件进行调试。
3. 对当前代码进行打断点，即在代码左侧进行单击，就会出现小红点。
4. 此时，即可按照正常的代码调试步骤对代码调试，在界面左边会显示debug信息，代码上方有相应的调试步骤。

图 5-158 调试代码

VSCode 使用常见问题

VSCode中查看远端日志

使用VSCode调试代码时不能进入源码

使用VSCode提交代码时弹出对话框提示用户名和用户邮箱配置错误

VSCode连接远端Notebook时报错如“XHR failed”

安装远端插件时不稳定，需尝试多次

Notebook实例重新启动后，需要删除本地known_hosts才能连接

更多VS Code使用相关FAQ，请参见**FAQ**。

5.6.6 配置本地 IDE（SSH 工具连接）

本节操作介绍在Windows环境中使用PuTTY SSH远程登录云上Notebook实例的操作步骤。

前提条件

- 创建一个Notebook实例，并开启远程SSH开发，配置远程访问IP白名单。该实例状态必须处于“运行中”，具体参见**创建Notebook实例**章节。
- 在Notebook实例详情页面获取开发环境访问地址（例如：dev-modelarts-cnnorth4.huaweicloud.com）和端口号。

图 5-159 Notebook 实例详情页面

- 准备好密钥对文件。

  密钥对在用户第一次创建时，自动下载，之后使用相同的密钥时不会再有下载界面（用户一定要保存好），或者每次都使用新的密钥对。

Step1 安装 SSH 工具

下载并安装SSH远程连接工具，以PuTTY为例，**下载链接**。

Step2 使用 puttygen 将密钥对.pem 文件转成.ppk 文件

1. **下载puttygen**，并双击运行puttygen。
2. 单击“Load”，上传.pem密钥（即在创建Notebook实例时创建并保存的密钥对文件）。
3. 单击“Save private key”，保存生成的.ppk文件。.ppk文件的名字可以自定义，例如key.ppk。

图 5-160 将密钥对.pem 文件转成.ppk 文件

Step3 使用 SSH 工具连接云上 Notebook 实例

1. 运行PuTTY。
2. 单击“Session”，填写以下参数。
   a. Host Name (or IP address)：云上开发环境Notebook实例的访问地址，即在Notebook实例详情页获取的地址。例如：dev-modelarts-cnnorth4.huaweicloud.com。
   b. Port：云上Notebook实例的端口，即在Notebook实例详情页获取的端口号。例如：32701。

c. Connection Type：选择 SSH。

d. Saved Sessions：任务名称，在下一次使用PuTTY时就可以单击保存的任务名称，即可打开远程连接。

图 5-161 设置 Session

3. 选择“Window > Translation”，在“Remote character set:”中选择“UTF-8”。

图 5-162 设置字符格式

4. 选择“Connection > Data”， 在“Auto-login username” 中填写用户名“ma-user”。

图 5-163 填写用户名

5. 选择“Connection > SSH > Auth”，单击“Browse”，选择“.ppk文件”（由**Step2**密钥对.pem文件生成）。

6. 单击“Open”。如果首次登录，PuTTY会显示安全警告对话框，询问是否接受服务器的安全证书。单击“Accept”将证书保存到本地注册表中。

图 5-164 询问是否接受服务器的安全证书

7. 成功连接到云上Notebook实例。

图 5-165 连接到云上 Notebook 实例

5.6.7 在本地 IDE 中上传数据至 Notebook

不大于500MB数据量，直接拷贝至本地IDE中即可。

大于500MB数据量，请先上传到OBS中，再从OBS上传到云上开发环境的默认存储EFS或云硬盘EVS。

1. 上传数据至OBS，具体操作请参见**上传文件至OBS桶**。
2. 将OBS中的数据传至Notebook的EFS/EVS中，通过在本地IDE的Terminal中使用ModelArts提供的Moxing库的文件操作API（mox.file.copy_parallel）完成。

图 5-166 数据通过 OBS 中转上传到 Notebook

下图以PyCharm环境中开启Terminal为例，VS Code中操作类似。

图 5-167 PyCharm 环境开启 Terminal

在本地IDE的Terminal中使用Moxing下载OBS文件到开发环境的操作示例如下：

```
#手动source进入开发环境
cat /home/ma-user/README
#然后选择要source的环境
source /home/ma-user/miniconda3/bin/activate MindSpore-python3.7-aarch64
#使用moxing
import moxing as mox
#下载一个OBS文件夹，从OBS下载至EFS/EVS（OBS -> EFS/EVS）
mox.file.copy_parallel('obs://bucket_name/sub_dir_0', '/tmp/sub_dir_0')
```

5.6.8 下载 Notebook 中的文件至本地

在Notebook中开发的文件，可以下载至本地。无论创建Notebook时存储位置选择默认存储或云硬盘EVS，此处的下载操作都一样。

PyCharm 环境中下载 Notebook 中的文件至本地

在本地IDE的Project目录下，单击右键，选择“Deployment”，单击“Download from xxx”（Notebook名称），将Notebook2.0工程文件下载到本地。

图 5-168 PyCharm 环境中下载 Notebook 中的文件至本地

5.7 使用 MindStudio 开发 Ascend 算子

背景介绍

MindStudio是一套基于IntelliJ框架的开发工具链平台，提供了应用开发、调试、模型转换功能，同时还提供了网络移植、优化和分析功能，为用户开发应用程序带来了极大的便利。

MindStudio提供了基于TBE和AI CPU的算子编程开发的集成开发环境，让不同平台下的算子移植更加便捷，适配昇腾AI处理器的速度更快。

ModelArts集成了基于MindStudio镜像的Notebook实例，方便用户通过ModelArts平台使用MindStudio镜像进行算子开发。

想了解更多关于MindStudio相关内容，请参见**MindStudio资料**。

前提条件

- 已经创建基于MindStudio引擎的Notebook实例，并设置允许远程访问的白名单IP地址。且该Notebook实例状态必须为“运行中”。具体操作参考**创建Notebook实例**。

  图 5-169 创建基于 MindStudio 引擎的 Notebook 实例

- 在Notebook实例详情页面获取开发环境IP地址（例如：dev-modelarts-cnnorth4.huaweicloud.com）和端口号。

图 5-170 Notebook 实例详情页面

- 准备好密钥对。密钥对在用户第一次创建时，自动下载，之后使用相同的密钥时不会再有下载界面（用户一定要保存好），或者每次都使用新的密钥对。创建新的密钥对操作入口在创建Notebook实例页面中可以找到。

在本地使用 MobaXterm 连接并打开 MindStudio 环境

在本地使用MobaXterm远程连接MindStudio开发环境。

1. 下载并安装MobaXterm软件，下载地址**MobaXterm**。对MobaXterm软件版本不做要求。
2. 打开MobaXterm，单击“Sessions ”，在Session settings窗口设置SSH连接，单击“OK”，连接到Notebook实例环境中。

图 5-171 在 MobaXterm 中设置 SSH 连接

- Remote host：云上Notebook实例的IP地址，即在Notebook实例详情页获取的地址。例如：dev-modelarts-cnnorth4.huaweicloud.com
- Port：云上Notebook实例的端口，即在Notebook实例详情页获取的端口号。
- Specify username：固定为ma-user
- Use private key：在创建Notebook实例时创建并保存的密钥对文件。

说明

如果使用MobaXterm远程连接Notebook失败，有可能是Notebook状态不是在“运行中”，或者允许远程访问的IP地址配置问题。

3. 在MobaXterm中连接到Notebook实例后，执行如下命令打开MindStudio，弹出UI界面即为打开MindStudio开发环境镜像成功。

```
sh MindStudio/bin/MindStudio.sh
```

图 5-172 连接到 Notebook 实例

图 5-173 MindStudio UI 界面

4. 在图5-173的窗口中保持默认配置，单击“OK”，进入下一步。

5. 单击“CANN-Toolkit Path”右侧的，选择最新版本的“ascend-toolkit”，本示例中为“/home/ma-user/Ascend/ascend-toolkit/5.0.2.1”。“ascend-toolkit”持续在升级更新，版本号以实际界面为准。

图 5-174 选择 Ascend Toolkit Path

6. 单击“Finish”。进入MindStudio工程创建页面，单击新建工程，创建一个新工程。

图 5-175 选择 Ascend Toolkit Path

7. 在创建Ascend Operator工程页面，选择工程存放路径为“/home/ma-user/work”目录。

图 5-176 创建 Ascend Operator 工程

注意

创建工程时，建议选择“/home/ma-user/work”目录存放工程文件。

“/home/ma-user/work”目录为持久化目录，Notebook实例停止后该目录下的内容不会丢失。

8. 单击Next，进入算子模板配置界面，参考创建算子工程，完成算子信息配置。

9. 选择保存目录为“/home/ma-user/work”目，单击“Save”。

图 5-177 保存文件

10. 单击“Finish”，完成算子工程创建。

图 5-178 创建完成算子工程

11. 算子开发之前，开发者需要参考**设置Python库**章节设置算子工程依赖Python库。创建MindStudio算子更多操作介绍，请参见**MindStudio官方资料**。

停止 Notebook 实例前备份文件

Notebook实例停止时，后端对应的容器环境会被删除，只有“/home/ma-user/work”目录下的内容会持久化保存，其他目录下的修改都会丢失。

备份方法

可以在停止Notebook实例前手工拷贝文件到/home/ma-user/work目录下。

需要拷贝的目录内容包括：

1. /home/ma-user/ AscendProjects目录下的自建工程
2. /home/ma-user/modelzoo/目录下的模型转换后的om文件、配置文件、评估报告
3. /home/ma-user/.mindstudio目录下的ssh配置
4. 其他用户自己修改的内容

当Notebook实例再次启动时，用户将手工备份的目录内容拷贝回原始目录后即可正常继续使用。

5.8 使用可视化工具 ML Studio 开发模型

ML Studio简称MLS，是ModelArts中的一个支持可视化机器学习建模的企业级AI开发工具，支持用户通过浏览器以全代码、少代码甚至零代码的方式开发AI模型。

MLS为AI开发者提供可视化的操作界面来编排机器学习模型的训练、评估和预测的过程，无缝衔接数据分析和预测应用，为用户的数据挖掘分析业务提供易用、高效、高性能的工具。同时还提供了丰富的预置算子和预置算链，并支持编写自定义算子，可帮助开发者快速构建具有实用价值的机器学习应用。

MLS相关操作请参见《**ML Studio开发工具**》。

5.9 算法开发套件

5.9.1 算法开发套件简介

ModelArts算法开发套件提供了一个全流程和白盒化的云原生算法开发工具，支持通过本地VSCode/PyCharm远程连接到云上开发环境后使用。Notebook作为算法工程开发入口，便于用户无感调用云上计算存储资源进行开发，同时提供丰富的、可扩展的算法套件便于用户使用或二次开发。

算法开发套件支持的主要特性：

- 工程管理：用户可通过ma-cli createproject创建工程。

- 资产管理：用户可通过python manage.py list列举或python manage.py install来安装、升级、覆盖已发布的算法包、模型包、数据集。

- 数据拷贝：用户可通过python manage.py copy命令来实现OBS和本地数据的快速传输。

- 本地训练、验证、推理：用户可通过python manage.py run命令。

- 本地部署：用户可通过python manage.py deploy命令。

- ModelArts训练、模型转换：用户可通过python manage.py run --launch_remote命令提交远程训练作业。

- ModelArts部署：用户可通过python manage.py deploy --launch_remote命令提交远程部署任务。

- 模块化设计：用户可基于算法框架规范构建自定义的算法资产并发布到ModelArts。

5.9.2 准备开发环境

创建 Notebook 实例

1. 登录ModelArts控制台。选择控制台区域，以“华北-北京四”为例。
2. 在开发环境Notebook（New）中创建基于pytorch1.4-cuda10.1-cudnn7-ubuntu18.04镜像的Notebook，具体操作请参见**创建Notebook实例**章节。

   如果需要使用本地IDE（PyCharm或VSCode）远程连接Notebook，此处需要开启SSH远程开发。

   图 5-179 创建 Notebook 实例

打开 Notebook 实例

提供以下2种方式：

一、使用JupyterLab打开

1. 在“ModelArts控制台 > 开发环境 > Notebook”页面的列表中，单击操作栏的“打开”，进入JupyterLab页面。JupyterLab操作请参见**JupyterLab简介及常用操作**。

图 5-180 使用 JupyterLab 打开

2. 创建一个ipynb文件或者打开Terminal窗口，用于交互使用命令。

图 5-181 创建一个 ipynb 文件

图 5-182 打开 Terminal

二、使用本地IDE（PyCharm或VS Code）远程连接到Notebook。

具体操作请参见**配置本地IDE（PyCharm ToolKit连接）**或**配置本地IDE（VSCode ToolKit连接）**。

5.9.3 创建算法工程

ma-cli是基于cookiecutter开发的用于管理工程的命令行工具，它支持创建一个算法模板工程并一键式安装ModelArts算法套件等。

了解算法工程模板

算法工程模板结构如下：

```
your-project-name
├── algorithms          -- algorithm toolkit folder
│   ├──custom
├── data                -- data folder,
│   └── raw             -- the original, immutable data dump.
├── docs                -- doc for your project
├── model_zoo           -- pretrained model zoo folder
```

```
├─── project-slug          -- user's source code in this project.
├─── config_ma.py          -- ModelArts related configuration, authentications, temp obs
bucket, proxy, etc.
├─── manage.py             -- CLI entry
├─── README.md
├─── requirement.txt       -- The requirements file for reproducing the  environment
└─── setup.py             -- makes project pip installable (pip install -e .) so project-slug can
be imported
```

其中：

- algorithms：算法默认下载路径。

- data：数据集默认下载路径。

- docs：使用文档目录。

- model_zoo：模型默认下载路径。

- config_ma.py：创建工程时输入的配置信息，如果在算法内的config.py里填写了同样的信息，那么算法config.py的优先级高于config_ma.py。

- manage.py算法工程命令行入口。

使用 ma-cli 创建工程

场景一：在JupyterLab的Terminal里使用ma-cli创建工程

1. 打开JupyterLab的Terminal。

图 5-183 打开 Terminal

2. 执行ma-cli createproject命令创建工程，用户可按照提示交互式输入project_name，此处以water_meter为例。

```
ma-cli createproject
```

图 5-184 创建算法工程

3. 执行cd {project_name}命令切换到工程目录下。
```
cd {project_name}
```
4. 根据界面提示输入相关参数。
```
author_name [Your name/organization/company/team]: 自定义，例如：modelarts-algorithms-test
huaweicloud_region_name [cn-north-4]: 此参数可以忽略，系统会自动识别Notebook所在区域。
obs_bucket ： 自定义OBS桶路径，该OBS桶必须与Notebook在同一个区域。
Select install_asset:
```

图 5-185 创建算法工程参数设置

```
Terminal 3
(PyTorch-1.4) [ma-user work]$pwd
/home/ma-user/work
(PyTorch-1.4) [ma-user work]$ma-cli createproject
project_name [modelarts_algo_project]: water_meter
project_slug [water_meter]: cd water_meter
author_name [Your name/organization/company/team]: modelarts-algorithms-test
huaweicloud_region_name [cn-north-4]:
obs_bucket [Intermediate OBS destination. ModelArts Algo project will leverage this OBS bucket t
o manage the data, model, logs and other temp files for launching the remote training job or mod
el convertion.]: obs:// /test-data/water-meter
Select install_asset:
1 - algorithm-ivgPose|model-ivgPose:body/simplepose_resnet50_coco_256x192|dataset-coco2017_sampl
e
2 - algorithm-ivgClassification|model-ivgClassification:repvgg/repvggA0_imagenet_224x224|dataset
-imagenet2012_sample
3 - algorithm-ivgSegmentation|model-ivgSegmentation:deeplabv3p_resnet50_cityscapes_1024x512|data
set-cityscapes_custom_sample
4 - algorithm-ivgDetection|model-ivgDetection:yolo/yolov3_plate_416x416|dataset-coco2017_sample
5 - algorithm-mmdetection|model-mmdetection:faster_rcnn/faster_rcnn_r50_fpn_1x_coco|dataset-coco
2017_sample
6 - skip
Choose from 1, 2, 3, 4, 5, 6 [1]:
```

场景二：在JuputerLab的 Ipynb文件里使用ma-cli创建工程

说明

在JuputerLab的 Ipynb文件里执行命令，需要在命令前加！符号，例如：

!ma-cli createproject -sc

1. 输入如下命令并执行，获取需要填写的参数信息。
```
!ma-cli createproject -sc
```

图 5-186 创建算法工程

2. 在上方导航栏单击 ＋ ，新建代码行，并输入如下命令创建工程，样例代码如下。

```
context = {
    "project_name": "modelarts_algo_project",
    "project_slug": "modelarts_algo_project",
    "author_name": "Your name/organization/company/team",
    "huaweicloud_region_name": "cn-north-4",
    "obs_bucket": "obs://bucket_name/bucket_directory",
    "install_asset": "skip"
}
!ma-cli createproject --context f"{context}"
```

“author_name” 替换成自定义名称。

“obs_bucket”替换成实际的OBS桶路径。

图 5-187 创建工程

3. 切换工作目录为创建的工程目录。此命令没有结果输出。

```
import os
# set project_dir to your project path
project_dir = '/home/ma-user/work/modelarts_algo_project'
os.chdir(project_dir)
```

图 5-188 切换工作目录

```
[11] import os
     # set project_dir to your project path
     project_dir = '/home/ma-user/work/modelarts_algo_project'
     os.chdir(project_dir)
```

4. 通过python manage.py -h查看算法开发套件支持的命令及其用法。

```
!python manage.py -h
```

图 5-189 查看算法开发套件支持的命令及其用法

```
[ ] !python manage.py -h

Usage: manage.py [OPTIONS] COMMAND [ARGS]...

Options:
  -h, --help  Show this message and exit.

Commands:
  convert  Convert model.
  copy     Transfer data between OBS and ModelArts Notebook.
  deploy   deploy model.
  export   To export certain contents.
  flow     To execute the task by your flow config.
  install  Install algorithm/model/dataset.
  list     List algorithm/model/dataset.
  query    To query ModelArts info.
  run      Train Evaluate Infer.
```

查看如何使用 ma-cli

系统提供了内置的help，通过以下命令可以查看如何使用ma-cli和ma-cli createproject相关命令。

查看如何使用ma-cli

```
ma-cli -h
```

查看如何使用ma-cli createproject

```
ma-cli createproject -h
```

5.9.4 使用算法套件快速完成水表读数识别

本示例围绕真实AI需求场景，介绍算法开发套件在水表表盘读数识别算法开发任务上的使用流程。

算法开发套件中目前提供自研(ivg系列)和开源(mm系列)共两套算法资产，可应用于分类、检测、分割和OCR等任务中。本示例中将组合使用自研分割算法(ivgSegmentation)和开源OCR算法(mmOCR)完成水表读数识别项目，并使用算法开发套件将其部署为华为云在线服务。

说明

本案例教程仅适用于“华北-北京四”区域，新版Notebook。

准备数据

1. 登录OBS控制台，创建OBS对象桶，区域选择“华北-北京四”。
2. 登录ModelArts控制台，选择控制台区域为“华北-北京四”。
3. 在“全局配置”页面查看是否已经配置授权，允许ModelArts访问OBS。如果没有配置授权，请参考**配置访问授权（全局配置）**添加授权。
4. 分别下载本案例的数据集，**水表表盘分割数据集**和**水表表盘读数OCR识别数据集**到OBS桶中，OBS路径示例如下：

   obs://{OBS桶名称}/water_meter_segmentation 水表表盘分割数据集

   obs://{OBS桶名称}/water_meter_crop 水表表盘读数OCR识别数据集

   说明

   从AIGallery下载数据集免费，但是数据集存储在OBS桶中会收取少量费用，具体计费请参见**OBS价格详情页**，案例使用完成后请及时清除资源和数据。

准备开发环境

1. 在“ModelArts控制台 > 开发环境 > Notebook（New）”页面中，创建基于pytorch1.4-cuda10.1-cudnn7-ubuntu18.04镜像，类型为GPU的Notebook，具体操作请参见**创建Notebook实例**章节。

   如果需要使用本地IDE（PyCharm或VS Code）远程连接Notebook，需要开启SSH远程开发。本案例以在线的JupyterLab为例介绍整个过程。

图 5-190 创建 Notebook 实例

2. 在“ModelArts控制台 > 开发环境 > Notebook（New）”页面的列表中，单击操作栏的“打开”，进入JupyterLab页面。JupyterLab操作请参见**JupyterLab简介及常用操作**。

图 5-191 使用 JupyterLab 打开

3. 打开JupyterLab的Terminal。此处以Terminal为例介绍整个过程。

图 5-192 打开 Terminal

Step1 创建算法工程

1. 在JupyterLab的Terminal中，在work目录下执行**ma-cli createproject**命令创建工程，根据提示输入工程名称，例如：water_meter。然后直接回车选择默认参数，并选择跳过资产安装步骤（选择6）。

图 5-193 创建工程

```
(PyTorch-1.4) [ma-user work]$pwd
/home/ma-user/work
(PyTorch-1.4) [ma-user work]$ma-cli createproject
project_name [modelarts_algo_project]: water_meter
project_slug [water_meter]:
author_name [Your name/organization/company/team]:
huaweicloud_region_name [cn-north-4]:
obs_bucket [Intermediate OBS destination. ModelArts Algo project will leverage this OBS bucket to manage the data, model, lo
gs and other temp files for launching the remote training job or model convertion.]:
Select install_asset:
1 - algorithm-ivgPose|model-ivgPose:body/simplepose_resnet50_coco_256x192|dataset-coco2017_sample
2 - algorithm-ivgClassification|model-ivgClassification:repvgg/repvggA0_imagenet_224x224|dataset-imagenet2012_sample
3 - algorithm-ivgSegmentation|model-ivgSegmentation:deeplabv3p_resnet50_cityscapes_1024x512|dataset-cityscapes_custom_sample
4 - algorithm-ivgDetection|model-ivgDetection:yolo/yolov3_plate_416x416|dataset-coco2017_sample
5 - algorithm-mmdetection|model-mmdetection:faster_rcnn/faster_rcnn_r50_fpn_1x_coco|dataset-coco2017_sample
6 - skip
Choose from 1, 2, 3, 4, 5, 6 [1]: 6
============================ README ============================
current directory is /home/ma-user/work, please cd to your project directory /home/ma-user/work/water_meter

Execute command as follows for detailed functions.
>command: python manage.py -h
```

2. 执行以下命令进入工程目录。
```
cd water_meter
```

3. 执行以下命令拷贝项目数据到Notebook中。
```
python manage.py copy --source {obs_dataset_path} --dest ./data/raw/water_meter_crop
python manage.py copy --source {obs_dataset_path} --dest ./data/raw/
water_meter_segmentation
```

说明

{obs_dataset_path}路径为**Step1 准备数据**中下载到OBS中的数据集，比如“obs://{OBS桶名称}/water_meter_segmentation”和“obs://{OBS桶名称}/water_meter_crop”

图 5-194 拷贝数据集到 Notebook 中

```
(PyTorch-1.4) [ma-user work]$cd water_meter
(PyTorch-1.4) [ma-user water_meter1]$python manage.py copy --source
 obs://[redacted]ata/water_meter_segmentat
ion --dest ./data/raw/water_meter_segmentation
100%|[redacted]| 2009/2009 [00
:06<00:00, 294.53it/s]
(PyTorch-1.4) [ma-user water_meter1]$python manage.py copy --source
 obs://[redacted]ata/water_meter_crop --de
st ./data/raw/water_meter_crop
100%|[redacted]| 1004/1004 [00
:03<00:00, 280.32it/s]
(PyTorch-1.4) [ma-user water_meter1]$
```

Step2 使用 deeplabv3 完成水表区域分割任务

1. 首先安装ivgSegmentation套件。

   ```
   python manage.py install algorithm ivgSegmentation==1.0.2
   ```

   如果提示ivgSegmentation版本不正确，可以通过命令python manage.py list algorithm查询版本。

2. 安装ivgSegmentation套件后，在JupyterLab界面左侧的工程目录中进入“./algorithms/ivgSegmentation/config/sample”文件夹中查看目前支持的分割模型，以sample为例（sample默认的算法就是deeplabv3），文件夹中包括config.py（算法外壳配置）和deeplabv3_resnet50_standard-sample_1024x512.py（模型结构）。

3. 表盘分割只需要区分背景和读数区域，因此属于二分类，需要根据项目所需数据集对配置文件进行修改，如下所示：

   修改./algorithms/ivgSegmentation/config/sample/config.py文件。

   ```
   # config.py
   alg_cfg = dict(
       ...
       data_root='data/raw/water_meter_segmentation',  # 修改为真实路径本地分割数据集路径
       ...
   )
   ```

   修改完后按Ctrl+S保存。

4. 修改./algorithms/ivgSegmentation/config/sample/deeplabv3_resnet50_standard-sample_512x1024.py文件。

   ```
   # deeplabv3_resnet50_standard-sample_512x1024.py
   gpus=[0]
   ...
   data_cfg = dict(
       ...    num_classes=2,  # 修改为2类
       ...
       ...    train_scale=(512, 512),  # (h, w)#size全部修改为(512, 512)
       ...    train_crop_size=(512, 512),  # (h, w)
       ...    test_scale=(512, 512),  # (h, w)
       ...    infer_scale=(512, 512),  # (h, w)
   )
   ```

   修改完按Ctrl+S保存。

5. 在water_meter工程目录下，安装deeplabv3预训练模型。

```
python manage.py install model ivgSegmentation:deeplab/
deeplabv3_resnet50_cityscapes_1024x512
```

图 5-195 安装 deeplabv3 预训练模型

```
(PyTorch-1.4) [ma-user water_meter1]$python manage.py install model
 ivgSegmentation:deeplab/deeplabv3_resnet50_cityscapes_1024x512
INFO:ma_cau:Begin to install deeplabv3_resnet50_cityscapes_1024x512
_1.0.2 to /home/ma-user/work/water_meter1/model_zoo/ivgSegmentation
/deeplab
INFO:ma_cau:Extracting /home/ma-user/work/water_meter1/model_zoo/iv
gSegmentation/deeplab/e8d10766-4068-409c-9223-cf561b7a8e26qfxo8x557
80-code.zip to /home/ma-user/work/water_meter1/model_zoo/ivgSegment
ation/deeplab
INFO:ma_cau:Successfully installed deeplabv3_resnet50_cityscapes_10
24x512_1.0.2 to /home/ma-user/work/water_meter1/model_zoo/ivgSegmen
tation/deeplab
(PyTorch-1.4) [ma-user water_meter1]$
```

6. 训练分割模型。（推荐使用GPU进行训练）

```
# shell
python manage.py run --cfg algorithms/ivgSegmentation/config/sample/config.py
 --gpus 0
```

训练好的模型会保存在指定位置中，默认为output/deeplabv3_resnet50_standard-sample_1024x512/checkpoints/中。

7. 验证模型效果。

模型训练完成后，可以在验证集上计算模型的指标，首先修改配置文件的模型位置。

修改./algorithms/ivgSegmentation/config/sample/config.py。

```
# config.py
alg_cfg = dict(
    ...
    load_from='./output/deeplabv3_resnet50_standard-sample_1024x512/checkpoints/
checkpoint_best.pth.tar',  # 修改训练模型的路径
    ...
)
```

```
# shell
python manage.py run --cfg algorithms/ivgSegmentation/config/sample/config.py --
pipeline evaluate
```

8. 模型推理。

模型推理能够指定某一张图片，并且推理出图片的分割区域，并进行可视化，首先需要指定需要推理的图片路径。

修改./algorithms/ivgSegmentation/config/sample/config.py

```
alg_cfg = dict(
    ...
   img_file =  './data/raw/water_meter_segmentation/image/train_10.jpg'  # 指定需要推理的
图片路径
   ...
)
```

执行如下命令推理模型效果。

```
# shell
python manage.py run --cfg algorithms/ivgSegmentation/config/sample/config.py --
pipeline infer
```

推理输出的图片路径在./output/deeplabv3_resnet50_standard-sample_1024x512下。

图 5-196 水表表盘分割结果可视化

9. 导出SDK。

   算法开发套件支持将模型导出成一个模型SDK，方便进行模型部署等下游任务。

```
# shell
python manage.py export --cfg algorithms/ivgSegmentation/config/sample/config.py --
is_deploy
```

图 5-197 SDK 导出示意图

Step4 水表读数识别

1. 首先安装mmocr套件。
```
python manage.py install algorithm mmocr
```
2. 安装mmocr套件后，./algorithms/mmocr/config/textrecog文件夹中包括config.py（算法外壳配置），需要根据所需算法和数据集路径修改配置文件。以下以robust_scanner算法为例。

   修改./algorithms/mmocr/algorithm/configs/textrecog/robustscanner_r31_academic.py，
```
# robustscanner_r31_academic.py
...
train_prefix = 'data/raw/water_meter_crop/'  # 修改数据集路径改为水表ocr识别数据集路径
train_img_prefix1 = train_prefix + 'train'
train_ann_file1 = train_prefix + 'train.txt'

...
test_prefix = 'data/raw/water_meter_crop/'
test_img_prefix1 = test_prefix + 'val/'
test_ann_file1 = test_prefix + 'val.txt'
```
3. 安装robust_scanner预训练模型。
```
python manage.py install model mmocr:textrecog/robustscanner_r31_academic
```
4. 训练OCR模型。

   初次使用mmcv时需要编译mmcv-full，该过程较慢，可以直接使用官方预编译的依赖包。

   预编译包URL： **https://download.openmmlab.com/mmcv/dist/cu101/torch1.6.0/index.html**
```
pip install https://download.openmmlab.com/mmcv/dist/cu101/torch1.6.0/mmcv_full-1.3.8-cp37-cp37m-manylinux1_x86_64.whl
```
   将./algorithms/mmocr/config/textrecog/config.py中的epoch（迭代数量）改为2，如下图所示：
```
BATCH_SIZE = 128
EPOCHS = 2
LR = 1e-3

# /cache is recommended for data root
```
   训练好的模型会保存在指定位置中，默认为output/${algorithm}中。
5. 验证模型效果。

   模型训练完成后，可以在验证集上计算模型的指标，首先修改配置文件的模型位置。

   修改./algorithms/mmocr/config/textrecog/config.py
```
# config.py
...
model_path = './output/robustscanner_r31_academic/latest.pth'
...
```
```
# shell
python manage.py run --cfg algorithms/mmocr/config/textrecog/config.py --pipeline evaluate
```
6. 模型推理。

模型推理能够指定某一张图片，并且推理出图片的分割区域，并进行可视化。首先需要指定待推理的图片路径,修改algorithms/mmocr/config/textrecog/config.py文件，具体如下。

修改./algorithms/mmocr/algorithm/configs/textrecog/robust_scanner/config.py

```
...
infer_img_file='./data/raw/water_meter_crop/val/train_10.jpg'  # 指定需要推理的图片路径
...
```

```
# shell
python manage.py run --cfg algorithms/mmocr/config/textrecog/config.py --pipeline infer
```

推理输出的图片路径在output/robustscanner_r31_academic/vis下

图 5-198 表盘读数识别结果图

7. 导出SDK。

```
# shell
python manage.py export --cfg algorithms/mmocr/config/textrecog/config.py
```

部署为在线服务

本次展示仅部署OCR服务， 包括本地部署和线上部署， 部署上线后调用部署服务进行本地图片的推理，获取水表的预测读数。部署在线服务，需要指定OBS桶以便保存部署所需要的文件。

1.在algorithms/mmocr/config/textrecog/config.py文件中配置OBS桶，即obs_bucket=<please input your own bucket here>。

2.执行下述命令：

```
python manage.py export --cfg algorithms/mmocr/config/textrecog/config.py --is_deploy  # 导出部署模型
python manage.py deploy --cfg algorithms/mmocr/config/textrecog/config.py # 本地部署
python manage.py deploy --cfg algorithms/mmocr/config/textrecog/config.py --launch_remote# 在线部署
```

图 5-199 部署为在线服务

Step5 清除资源和数据

通过此示例学习完成创建算法套件流程后，如果不再使用，建议您清除相关资源，避免造成资源浪费和不必要的费用。

- 停止Notebook：在“Notebook”页面，单击对应实例操作列的“停止”。
- 删除数据：前往OBS，删除上传的数据，然后删除文件夹及OBS桶。

5.9.5 使用样例

5.9.5.1 使用现有模型进行推理

本小节以姿态估计ivgPose 的预训练模型simplepose_resnet50_coco_256x192为例进行图片人物的姿势估计推理。

Step0 准备环境

使用前需要先准备开发环境。

1. 参考**准备开发环境**章节创建并打开Notebook实例。
2. 参考**创建算法工程**章节创建算法工程。

Step1 安装算法套件

安装ivgPose算法套件。

```
python manage.py install algorithm ivgPose
```

Step2 安装预训练模型

默认下载到当前工程的./model_zoo/{算法类型}/{算法版本}目录下。

```
python manage.py install model ivgPose:body/simplepose_resnet50_coco_256x192
```

Step3 推理

1. 准备好待推理的图片，本小节以算法套件里内置的推理图片为例，原图片如下。

   图 5-200 待推理的图片

2. 在terminal中执行下述推理命令，其中--img_file的值为待推理图片的路径。

```
python manage.py run --cfg algorithms/ivgPose/config/sample/config.py --pipeline infer --demo_type image_demo --load_from model_zoo/ivgPose/body/simplepose_resnet50_coco_256x192.pth.tar --img_path algorithms/ivgPose/algorithm/examples/images/body/human36m_s_01_act_02_subact_01_ca_01_000001.jpg  --is_show
```

3. 运行完毕后，在当前目录的output/simplepose_resnet50_coco_256x192/下可以看到推理后的文件human36m_s_01_act_02_subact_01_ca_01_000001_vis.jpg，打开后显示如下。

图 5-201 推理后的文件

5.9.5.2 使用现有数据集测试现有模型

为了评估模型的准确性，通常会基于标准数据集来测试模型。算法套件当前发布了五种mini数据集，包括coco2017_sample，imagenet2012_sample，cityscapes_custom_sample，coco_stuff和icdar_det。

本章节将介绍如何基于已有数据集测试现有模型。

Step0 准备环境

使用前需要先准备开发环境。

1. 参考**准备开发环境**章节创建并打开Notebook实例。
2. 参考**创建算法工程**章节创建算法工程。

Step1 安装算法套件

安装ivgPose算法套件。

```
python manage.py install algorithm ivgPose
```

Step2 安装数据集

安装coco数据集。

```
python manage.py install dataset coco2017_sample
```

Step3 安装预训练模型

默认下载到当前工程的./model_zoo/{算法类型}目录下。

```
python manage.py install model ivgPose:body/simplepose_resnet50_coco_256x192
```

Step4 修改配置文件

打开预训练模型对应算法资产的配置文件路径，如./algorithms/ivgPose/config/sample/config.py。按需进行验证相关参数的修改，如修改数据集路径data_root为下载的mini数据集路径./data/raw/coco2017_sample、指定验证阶段使用的验证数据集路径等。如下述代码所示。

```
alg_cfg = dict(
    cfg=f'{work_dir}/config/sample/{alg_name}.py',
    data_root='data/raw/coco2017_sample',
    load_from=f'model_zoo/{alg_type}/body/{alg_name}.pth.tar',
    pretrained=f'model_zoo/{alg_type}/backbone/resnet50_imagenet_224x224.pth',
    img_file=f'{work_dir}/algorithm/examples/images/body/coco_000000013729.jpg',
    bbox_file=f'{work_dir}/algorithm/examples/images/body/bboxes.json'
)
    evaluate_args=dict(
        cfg=alg_cfg['cfg'],
        output_dir=run_dir,
        data_root=alg_cfg['data_root'],
        load_from=alg_cfg['load_from'],
        gpus=None,
        samples_per_gpu=None,
    ),
```

Step5 本地验证

在terminal中执行下述测试命令，其中：--cfg为该预训练模型对应算法资产的配置文件路径，--load_from的值为待测试的模型路径，config.py为：

```
python manage.py run --cfg algorithms/ivgPose/config/sample/config.py --pipeline evaluate --load_from ./model_zoo/ivgPose/body/simplepose_resnet50_coco_256x192.pth.tar
```

运行完毕后，在交互式输出界面或config.py配置的{run_dir}目录下的test.log可以看到验证过程的日志和结果。如下述信息所示：

```
ivgpose-utils.py 133: Test: AP:0.735012 | AP .5:0.925143 | AP .75:0.813647 | AP (M):0.705874 | AP (L):0.778985 | AR:0.764798 | AR .5:0.933879 | AR .75:0.832966 | AR (M):0.732641 | AR (L):0.813304 | elapsed:00:28
```

5.9.5.3 使用现有数据集训练预训练模型

本章节将介绍如何基于已有数据集训练一个预训练模型。

算法套件当前发布了五种mini数据集，包括coco2017_sample，imagenet2012_sample，cityscapes_custom_sample，coco_stuff和icdar_det。

Step0 准备环境

使用前需要先准备开发环境。

1. 参考**准备开发环境**章节创建并打开Notebook实例。
2. 参考**创建算法工程**章节创建算法工程。

Step1 安装算法套件

安装ivgPose算法套件。

```
python manage.py install algorithm ivgPose
```

Step2 安装数据集

安装coco数据集。

```
python manage.py install dataset coco2017_sample
```

Step3 安装预训练模型

默认下载到当前工程的./model_zoo/{算法类型}/{算法版本}目录下。

```
python manage.py install model ivgPose:body/simplepose_resnet50_coco_256x192
python manage.py install model ivgPose:backbone/resnet50_imagenet_224x224
```

Step4 修改配置文件

配置文件主要包括算法配置文件和算法参数文件，例如：

- 算法配置文件：./algorithms/ivgPose/config/sample/config.py
- 算法参数文件：./algorithms/ivgPose/config/sample/simplepose_resnet50_coco_256x192.py

根据需要修改训练相关参数，如数据集路径data_root、runner中的train_args参数以及train参数等。如下述代码所示。

config.py文件

```
train_args=dict(
    cfg=alg_cfg['cfg'],
    output_dir=run_dir,
    data_root=alg_cfg['data_root'],
    load_from=None,
    gpus=None,
    lr=None,
    max_epoch=5,
    samples_per_gpu=None,
    resume_from=None,
    pretrained=alg_cfg['pretrained'],
    train_data_root=None,
    val_data_root=None,
),
```

simplepose_resnet50_coco_256x192.py文件

```
train = dict(
```

```
    save_period=-1,
    val_period=5,
    best_saver=dict(key_indicator='AP', rule='max'),
    ema=dict(type='ExpEMA')
)
```

Step5 本地训练

在Terminal中执行下述训练命令，其中：--cfg为该预训练模型对应算法资产的配置文件路径。

```
python manage.py run --cfg algorithms/ivgPose/config/sample/config.py --gpus 0
```

运行完毕后，在交互式输出界面或config.py配置的{run_dir}目录下的train_{timestamp}.log可以看到训练过程的日志和结果。如下述信息所示。

```
Train: 000/003 | loss:0.002145 | acc_pose:0.029999 | lr:{'0.000001', '0.000005'}: 100%|
| 397/397 [00:36<00:00, 10.73it/s]
[07/19 15:36:30][INFO] ivgpose-utils.py  52: Train: 000/003 | loss:0.002145 | acc_pose:0.029999 | lr:
{'0.000001', '0.000005'} | elapsed:00:36
Val: 000/003 | loss:0.002451 | acc_pose:0.060385 | : 100%|
| 397/397 [00:16<00:00, 24.09it/s]
[07/19 15:36:53][INFO] ivgpose-utils.py 101: Val: 000/003 | loss:0.002451 | acc_pose:0.060385 | elapsed:00:16
[07/19 15:36:55][INFO] ivgpose-utils.py 167: ******************** Best: acc_pose:0.060385 | epoch:000
Train: 001/003 | loss:0.002144 | acc_pose:0.043690 | lr:{'0.000170', '0.000017'}: 100%|
| 397/397 [00:35<00:00, 11.10it/s]
[07/19 15:37:31][INFO] ivgpose-utils.py  52: Train: 001/003 | loss:0.002144 | acc_pose:0.043690 | lr:
{'0.000170', '0.000017'} | elapsed:00:35
Train: 002/003 | loss:0.002148 | acc_pose:0.053395 | lr:{'0.000034', '0.000335'}: 100%|
| 397/397 [00:35<00:00, 11.13it/s]
[07/19 15:38:13][INFO] ivgpose-utils.py  52: Train: 002/003 | loss:0.002148 | acc_pose:0.053395 | lr:
{'0.000034', '0.000335'} | elapsed:00:35
Val: 002/003 | loss:0.002407 | acc_pose:0.088161 | : 100%|
| 397/397 [00:16<00:00, 23.41it/s]
[07/19 15:38:36][INFO] ivgpose-utils.py 101: Val: 002/003 | loss:0.002407 | acc_pose:0.088161 | elapsed:00:16
[07/19 15:38:39][INFO] ivgpose-utils.py 167: ******************** Best: acc_pose:0.088161 | epoch:002
```

5.9.5.4 使用自定义数据集进行训练

ivgSegmentation支持cityscapes_custom_sample、[Pascal VOC](下载地址：**http://host.robots.ox.ac.uk/pascal/VOC/index.html**)和自定义数据集。

Step0 准备环境

使用前需要先准备开发环境。

1. 参考**准备开发环境**章节创建并打开Notebook实例。
2. 参考**创建算法工程**章节创建算法工程。

Step1 上传数据集到开发环境中

按照不同算法定义的数据结构和格式准备好数据集。

- 若数据集在OBS桶里，在Terminal里可执行下述命令将OBS里的数据下载到Notebook环境中。
  ```
  python manage.py copy --source obs://my_bucket/my_data_dir/ --dest /home/ma-user/work/my_data_dir
  ```
- 若数据集比较小且存放在PC上。

  若使用的本地IDE是VS Code，可以通过拖放方式复制到VS Code连接的Notebook中。

自定义数据集结构如下：

```
├── my_dataset
│   ├── image
│   │   ├── train
│   │   │   ├── xxx{img_suffix}
│   │   │   ├── yyy{img_suffix}
│   │   │   ├── zzz{img_suffix}
│   │   │   ├── ...
│   │   ├── val
│   ├── annotation
│   │   ├── train
│   │   │   ├── xxx{seg_map_suffix}
│   │   │   ├── yyy{seg_map_suffix}
│   │   │   ├── zzz{seg_map_suffix}
│   │   │   ├── ...
│   │   ├── val
```

Step2 修改配置文件

配置文件主要包括算法配置文件和算法参数文件，例如：

- 算法配置文件：./algorithms/ivgPose/config/sample/config.py
- 算法参数文件：./algorithms/ivgPose/config/sample/simplepose_resnet50_coco_256x192.py。

根据需要修改训练相关参数，如数据集路径data_root、runner中的train_args参数以及train参数等。如下述代码所示。

config.py文件

```
train_args=dict(
    cfg=alg_cfg['cfg'],
    output_dir=run_dir,
    data_root=alg_cfg['data_root'],
    load_from=None,
    gpus=None,
    lr=None,
    max_epoch=5,
    samples_per_gpu=None,
    resume_from=None,
    pretrained=alg_cfg['pretrained'],
    train_data_root=None,
    val_data_root=None,
),
```

simplepose_resnet50_coco_256x192.py文件

```
train = dict(
    save_period=-1,
    val_period=5,
    best_saver=dict(key_indicator='AP', rule='max'),
    ema=dict(type='ExpEMA')
)
```

Step3 本地训练

在Terminal中执行下述训练命令，其中：--cfg为该预训练模型对应算法资产的配置文件路径。

```
python manage.py run --cfg algorithms/ivgPose/config/sample/config.py --gpus 0
```

运行完毕后，在交互式输出界面或config.py配置的{run_dir}目录下的train_{timestamp}.log可以看到训练过程的日志和结果。如下述信息所示。

```
Train: 000/003 | loss:0.002145 | acc_pose:0.029999 | lr:{'0.000001', '0.000005'}: 100%|
| 397/397 [00:36<00:00, 10.73it/s]
[07/19 15:36:30][INFO] ivgpose-utils.py  52: Train: 000/003 | loss:0.002145 | acc_pose:0.029999 | lr:
{'0.000001', '0.000005'} | elapsed:00:36
Val: 000/003 | loss:0.002451 | acc_pose:0.060385 | : 100%|
| 397/397 [00:16<00:00, 24.09it/s]
[07/19 15:36:53][INFO] ivgpose-utils.py 101: Val: 000/003 | loss:0.002451 | acc_pose:0.060385 | elapsed:00:16
[07/19 15:36:55][INFO] ivgpose-utils.py 167: ******************* Best: acc_pose:0.060385 | epoch:000
Train: 001/003 | loss:0.002144 | acc_pose:0.043690 | lr:{'0.000170', '0.000017'}: 100%|
| 397/397 [00:35<00:00, 11.10it/s]
[07/19 15:37:31][INFO] ivgpose-utils.py  52: Train: 001/003 | loss:0.002144 | acc_pose:0.043690 | lr:
{'0.000170', '0.000017'} | elapsed:00:35
Train: 002/003 | loss:0.002148 | acc_pose:0.053395 | lr:{'0.000034', '0.000335'}: 100%|
| 397/397 [00:35<00:00, 11.13it/s]
[07/19 15:38:13][INFO] ivgpose-utils.py  52: Train: 002/003 | loss:0.002148 | acc_pose:0.053395 | lr:
{'0.000034', '0.000335'} | elapsed:00:35
Val: 002/003 | loss:0.002407 | acc_pose:0.088161 | : 100%|
| 397/397 [00:16<00:00, 23.41it/s]
[07/19 15:38:36][INFO] ivgpose-utils.py 101: Val: 002/003 | loss:0.002407 | acc_pose:0.088161 | elapsed:00:16
[07/19 15:38:39][INFO] ivgpose-utils.py 167: ******************* Best: acc_pose:0.088161 | epoch:002
```

Step4 本地验证

在Terminal中执行下述测试命令，其中：--cfg为该预训练模型对应算法资产的配置文件路径，--load_from的值为待测试的模型路径，config.py为：

```
python manage.py run --cfg algorithms/ivgPose/config/sample/config.py --pipeline evaluate --
load_from ./model_zoo/ivgPose/body/simplepose_resnet50_coco_256x192.pth.tar
```

运行完毕后，在交互式输出界面或config.py配置的{run_dir}目录下的test.log可以看到验证过程的日志和结果。如下述信息所示：

```
ivgpose-utils.py 133: Test: AP:0.735012 | AP .5:0.925143 | AP .75:0.813647 | AP (M):0.705874 | AP (L):
0.778985 | AR:0.764798 | AR .5:0.933879 | AR .75:0.832966 | AR (M):0.732641 | AR (L):0.813304 | elapsed:
00:28
```

Step5 推理

1. 准备好待推理的图片，本小节以算法套件里内置的推理图片为例，原图片如下：

图 5-202 待推理的图片

2. 在Terminal中执行下述推理命令，其中--img_file的值为待推理图片的路径：
```
python manage.py run --cfg algorithms/ivgPose/config/sample/config.py --pipeline infer --demo_type image_demo --load_from model_zoo/ivgPose/body/simplepose_resnet50_coco_256x192.pth.tar --img_path algorithms/ivgPose/algorithm/examples/images/body/human36m_s_01_act_02_subact_01_ca_01_000001.jpg  --is_show
```
3. 运行完毕后，在当前目录的export/exp_tmp下可以看到推理后的文件，打开后显示如下。

图 5-203 推理后的文件

Step6 远程训练

1. 修改配置文件

   a. 填写OBS桶信息

   ```
   obs_bucket = 'obs://my_bucket/my_object'
   ```

   b. 按需修改runner里的**Adapter**参数。

   ```
   adapter=dict(
       requirements=f'{work_dir}/algorithm/requirements.txt',

       framework_type='PyTorch',
       framework_version='PyTorch-1.4.0-python3.6',
       instance_type='modelarts.p3.large.public',
       pool_id=None,

       downloads=dict(
           src=[f'{obs_bucket}/{alg_cfg["data_root"]}',
               f'{obs_bucket}/{alg_cfg["pretrained"]}'],
           dst=[alg_cfg['data_root'], alg_cfg['pretrained']],
       ),
       uploads=dict(
           src=[run_dir],
           dst=[f'{obs_bucket}/{run_dir}'],
   ```

```
    ),
  ),
```

2. 在Terminal中输入下述命令来提交ModelArts训练作业完成训练。

```
python manage.py run --launch_remote --cfg algorithms/ivgPose/config/sample/config.py
--gpus 0
```

提交完训练作业后，可以在ModelArts控制台交互式界面看到当前训练作业的状态（如排队中、运行中等），可以在config.py配置的{run_dir}/{训练作业名称目录}下看到ModelArts上的训练日志。

5.9.6 算法开发套件命令说明

5.9.6.1 列举

每个内置的算法资产内包含了算法套件、数据集、预训练模型，可以通过list命令查看内置的资产。

查询内置算法

```
python manage.py list algorithm
```

查询内置数据集

```
python manage.py list dataset
```

查询内置预训练模型

```
# 列举所有的内置预训练模型
> python manage.py list model

# 根据关键字过滤
> python manage.py list model --filter mmDetection
```

5.9.6.2 安装

安装内置算法

```
# 安装ivgPose最新版本
> python manage.py install algorithm ivgPose

# 安装ivgPose==1.0.0
> python manage.py install algorithm ivgPose==1.0.0

# 强制安装最新版本的ivgPose,将覆盖已有的ivgPose目录
> python manage.py install algorithm ivgPose --force

# 升级ivgPose到最新版本，将新建ivgPose_{version}目录
> python manage.py install algorithm ivgPose --upgrade
```

安装内置数据集

```
安装最新版本的内置coco2017_sample数据集
> python manage.py install dataset coco2017_sample

强制安装coco2017_sample==1.0.0
> python manage.py install dataset coco2017_sample==1.0.0 --force
```

安装内置预训练模型

```
# 安装预训练模型ivgPose:backbone/resnet50_imagenet_224x224
> python manage.py install model ivgPose:backbone/resnet50_imagenet_224x224

# 强制安装预训练模型ivgPose:backbone/resnet50_imagenet_224x224
> python manage.py install model ivgPose:backbone/resnet50_imagenet_224x224 --force
```

5.9.6.3 拷贝

本地和 OBS 数据交互

```
# 拷贝本地目录到OBS
> python manage.py copy --source /home/ma-user/work/my_dir --dest obs://my_bucket/my_dir/

# 拷贝OBS目录到本地
> python manage.py copy --source obs://my_bucket/my_dir/ --dest /home/ma-user/work/my_dir

# 拷贝本地文件到OBS
> python manage.py copy --source /home/ma-user/work/my_file --dest obs://my_bucket/
my_dir/my_file

# 拷贝OBS文件到本地
> python manage.py copy --source obs://my_bucket/my_dir/my_file --dest /home/ma-user/
work/my_file

# 加速文件传输
当存在大量小文件时，逐个文件上传
> python manage.py copy --source obs://my_bucket/my_dir/my_file --dest /home/ma-user/
work/my_file
```

5.9.6.4 运行

本地模式

```
# 基于CPU训练
> python manage.py run --cfg algorithms/ivgSegmentation/config/sample/config.py

# 基于单卡GPU训练
> python manage.py run --cfg algorithms/ivgSegmentation/config/sample/config.py --gpus 0

# 基于单卡GPU验证模型
> python manage.py run --cfg algorithms/ivgSegmentation/config/sample/config.py --gpus 0 --
pipeline evaluate  --load_from output/deeplabv3_resnet50_standard-sample_1024x512/
checkpoints/checkpoint_best.pth.tar

# 基于模型simplepose_resnet50_coco_256x192.pth.tar进行图片推理
> python manage.py run --cfg algorithms/ivgPose/config/sample/config.py --pipeline infer --
demo_type image_demo --load_from model_zoo/ivgPose/body/
simplepose_resnet50_coco_256x192.pth.tar --img_path algorithms/ivgPose/algorithm/examples/
images/body/human36m_s_01_act_02_subact_01_ca_01_000001.jpg  --is_show
```

远程模式

```
# 基于四卡GPU进行训练
# 单机单进程
> python manage.py run --launch_remote --cfg algorithms/ivgPose/config/sample/config.py --
gpus 0,1,2,3
# 单机多进程
> python manage.py run --launch_remote --cfg algorithms/ivgPose/config/sample/config.py --
```

```
gpus 0,1,2,3 --local_size 4

# 基于双节点训练
# 两节点，每个节点4卡，单进程
> python manage.py run --cfg algorithms/ivgSegmentation/config/sample/config.py --gpus
0,1,2,3 --world_size 2
# 两节点，每个节点4卡，多进程
> python manage.py run --cfg algorithms/ivgSegmentation/config/sample/config.py --gpus
0,1,2,3 --local_size 4 --world_size 2
```

5.9.6.5 转换

本地模式

```
# 将torch转换为caffe
> python manage.py convert --cfg algorithms/ivgPose/config/sample/config.py --pipeline
torch2caffe
```

远程模式

```
# 在ModelArts训练平台将torch转换为caffe
> python manage.py convert --launch_remote --pipeline torch2onnx --cfg algorithms/
ivgSegmentation/config/sample/config.py
```

5.9.6.6 导出

```
# 导出源码、config.json和model
> python manage.py export --cfg algorithms/ivgSegmentation/config/sample/config.py

# 导出SDK和部署所需文件
> python manage.py export --cfg algorithms/ivgSegmentation/config/sample/config.py --
is_deploy
```

5.9.6.7 部署

5.9.6.7.1 部署脚本参考

模型部署主要由config.json和customize_service.py共同组成，其中，config.json的功能是管理模型运行所需的依赖环境，customize_service.py的功能是模型前向推理的逻辑。本篇文档将对于自研和第三方算法套件的customize_sevice.py进行介绍。

一、基础结构

如下所示是模型推理的基本逻辑，由_preprocess、_inference、_postprocess三部分组成。

表 5-8 模型推理函数解析

函数	输入	返回值	功能
_preprocess	request请求原始数据， Type(map)	numpy类型数据列表， Type(map)	接收request数据，并转换为模型可以接受的输入格式

函数	输入	返回值	功能
_preprocess	numpy类型数据列表， Type(map)	numpy类型推理结果列表，Type(map)	对于输入数据进行前向推理，得到推理结果
_postprocess	numpy类型数据列表， Type(map)	json类型输出结果， Type(map)	将推理的结果进行后处理，得到预期的输出格式，该结果就是最终的返回值

```
class SingleNodeService(ModelService):
    '''SingleNodeModel defines abstraction for model service which loads a
    single model.
    '''

    def inference(self, data):
        '''
        Wrapper function to run preprocess, inference and postprocess functions.

        Parameters
        ----------
        data : map of object
            Raw input from request.

        Returns
        -------
        list of outputs to be sent back to client.
            data to be sent back
        '''
        data = self._preprocess(data)
        ...
        data = self._inference(data)
        ...
        data = self._postprocess(data)
        ...
        return data

    @abstractmethod
    def _inference(self, data):
        '''
        Internal inference methods. Run forward computation and
        return output.

        Parameters
        ----------
        data : map of NDArray
            Preprocessed inputs in NDArray format.

        Returns
        -------
        list of NDArray
            Inference output.
        '''
        return data

    @abstractmethod
```

```
    def _preprocess(self, data):
        '''
        Internal preprocess methods. Do transformation on raw
        inputs and convert them to NDArray.

        Parameters
        ----------
        data : map of object
            Raw inputs from request.

        Returns
        -------
        list of NDArray
            Processed inputs in NDArray format.
        '''
        return data

    @abstractmethod
    def _postprocess(self, data):
        '''
        Internal postprocess methods. Do transformation on inference output
        and convert them to MIME type objects.

        Parameters
        ----------
        data : map of NDArray
            Inference output.

        Returns
        -------
        list of object
            list of outputs to be sent back.
        '''
        return data
```

二、自研套件

ivgDetection是自研的目标检测算法套件之一，此处以预置的yolov3车牌检测模型（pytorch）为例，介绍ivgDetection套件的推理文件书写方式。

2.1 模型初始化

模型初始化只会在模型初次部署时调用，主要包括模型准备工作，包含模型构建和模型导入等内容。

```
...
# PyTorch runtime
from model_service.pytorch_model_service import PTServingBaseService
from base.config import CfgNode
from infer import Controller
...

class ImageClassificationService(PTServingBaseService):
    def __init__(self, model_name, model_path, **kwargs):
        cfg = CfgNode()
        cfg.model_dir = osp.join(osp.dirname(osp.abspath(__file__)), 'res')
        cfg.devices = list(range(torch.cuda.device_count())) if torch.cuda.is_available() else []
        cfg.max_batch_size = 1
        cfg.is_vis = False

        controller = Controller(cfg=cfg)
```

```
        # 初始化模型并载入训练好的模型，模型结构和模型路径都是在算法开发套件的config.py中指定的
        self.infer = controller.create_ruinner(infer_type='ImageDetRunner')

        if torch.cuda.is_available():
            logger.info('Using GPU for inference')
        else:
            logger.info('Using CPU for inference')
```

2.2 _pre_process

```
    def _preprocess(self, data: dict) -> dict:
        imgs = []
        for k, v in data.items():
            if k == 'images':
                for file_name, file_content in v.items():

                    img = np.asarray(Image.open(file_content), dtype=np.uint8)

                    imgs.append(img)
            # 返回一个字典，其中字典的键可以是任意的，对应的值就是numpy类型的图像列表
        preprocess_data = dict(
            inputs=(imgs,)
        )
        return preprocess_data
```

2.3 _inference

```
    def _inference(self, data: dict) -> dict:
    # 这里的input就是_pre_process中返回的字典的键，需要进行对应
        inputs = data['inputs']
        # 前向推理
        results, _ = self.infer(*inputs)

        inference_data = dict(
            results=results
        )
        return inference_data
```

2.4 _post_process

```
    def _postprocess(self, data: dict) -> dict:
        results = data['results']
        # 将推理结果results进行后处理，如下的后处理方式可以直接在网页端显示bbox可视化结果
        bboxes, scores, classes = [], [], []
        work_dir = osp.dirname(osp.abspath(__file__))

        # 类别索引对照表，存放在模型导出后的res/config.json文件中
        label_list = json.load(open(osp.join(work_dir, 'res','config.json')))['infer']['visualizer']['label_list']
        for item in results['preds'][0]:
            bboxes.append([item[1], item[0], item[3], item[2]])
            scores.append(item[-2])
            classes.append(label_list[int(item[-1])])

        detection = dict(
            detection_classes=classes,
            detection_boxes=bboxes,
            detection_scores=scores
        )

        return detection
```

图 5-204 ivgDetection 在线部署

```
{
  "detection_classes": [
    "plate"
  ],
  "detection_boxes": [
    [
      264.01228502061633,
      311.3196867058076,
      343.07130951693404,
      546.5949329054499
    ]
  ],
  "detection_scores": [
    0.8014615178108215
  ]
}
```


上述可视化后处理方式还需要配合config.json一起指定输出的数据结构，用户需要在config.json中的response中写成：

```
...
...
    "response": {
            "Content-type": "multipart/form-data",
            "data": {
                "type": "object",
                "properties": {
                    "detection_classes": {
                        "type": "array",
                        "items": [{
                            "type": "string"
                        }]
                    },
                    "detection_boxes": {
                        "type": "array",
                        "items": [{
                            "type": "array",
                            "minItems": 4,
                            "maxItems": 4,
                            "items": [{
                                "type": "number"
                            }]
                        }]
                    },
                    "detection_scores": {
                        "type": "array",
                        "items": [{
                            "type": "number"
                        }]
                    }
                }
            }
        }
```

三、开源套件

mmdetection是算法开发套件支持的第三方目标检测算法套件之一，我们以预置的faster_rcnn目标检测模型（pytorch）为例，介绍mmdetection套件的推理文件书写方式。

3.1 模型初始化

构建模型结构，载入模型权值，完成模型初始化构建工作。

```
from model_service.pytorch_model_service import PTServingBaseService
from mmdet.apis import (inference_detector, init_detector)
...
class ImageClassificationService(PTServingBaseService):
    def __init__(self, model_name, model_path, **kwargs):
        model_dir = osp.join(osp.dirname(osp.abspath(__file__)), 'res')
        # 模型导出生成的config.py、训练模型、类别索引文件路径
        config = osp.join(model_dir, 'config.py')
        checkpoint = osp.join(model_dir, 'model.pth')
        class_txt_path = osp.join(model_dir, 'classes.txt')

        device = 'cuda:0' if torch.cuda.is_available() else 'cpu'
        cfg_options = {
            'data.test.classes':f'class_txt_path',
        }

        # 模型初始化，并载入训练模型
        self.model = init_detector(config, checkpoint, device=device, cfg_options=cfg_options)


        if torch.cuda.is_available():
            logger.info('Using GPU for inference')
        else:
            logger.info('Using CPU for inference')
```

3.2 _pre_process

从request请求中读取图片流，并转换为模型可以接受的输入。

```
    def _preprocess(self, data: dict) -> dict:
        imgs = []
        for k, v in data.items():
            if k == 'images':
                for file_name, file_content in v.items():
                    img = cv2.imdecode(np.array(bytearray(file_content.read()), dtype=np.uint8),
cv2.IMREAD_COLOR)
                    imgs.append(img)

        preprocess_data = dict(
            inputs=imgs
        )
        return preprocess_data
```

3.3 _inference

```
    def _inference(self, data: dict) -> dict:
        inputs = data['inputs']
        # 前向推理，得到推理结果
        result = inference_detector(self.model, inputs)

        inference_data = dict(
            results=result
        )
        return inference_data
```

3.4 _post_process

```
    def _postprocess(self, data: dict) -> dict:
        results = data['results']
        # 将推理结果转换成最终的json格式
        label_list = self.model.CLASSES
```

```
    classes = []
    bboxes = []
    scores = []
    for index, res in enumerate(results[0]):
        category = label_list[index]
        for item in res:
            if len(item) < 5:
                continue
            bbox = item[:4].tolist()
            score = float(item[4])
            if score >= SCORE_THR:
                classes.append(category)
                bboxes.append([bbox[1], bbox[0], bbox[3], bbox[2]])
                scores.append(score)

    detection = dict(
            detection_classes=classes,
            detection_boxes=bboxes,
            detection_scores=scores
    )
    return detection
```

图 5-205 mdetection 在线部署

```
{
  "detection_classes": [
    "person",
    "person",
    "person",
    "person",
    "person",
    "person",
    "cat",
    "dog"
  ],
  "detection_boxes": [
    [
      86.9727783203125,
      135.02552795410156,
      346.16168212890625,
      300.2494812011719
    ],
    [
      5.226272106170654,
      338.4970397949219,
      338.331787109375,
      518.6382446289062
    ],
```


四、用户自定义模型

如果用户需要部署的模型是基于自研套件或者第三方套件进行二次开发的，那么推理脚本和内置的算法是一致的，同一个套件中的所有算法理论上是可以共享同一个部署推理脚本的。如果是完全自创的或者不属于算法开发套件内置套件的任何一种，用户可以参考**模型推理代码编写说明**

5.9.6.7.2 调用部署服务进行推理

推理本地部署服务，本地服务的端口信息（如127.0.0.1:1234）将会在部署成功后打印在控制台中，推理本地服务，只需提供本地服务的地址即可

> python manage.py run --cfg algorithms/ivgSegmentation/config/sample/config.py --pipeline infer --demo_type image_service --url http://127.0.0.1:1234

推理在线服务，需要提供用户的AK和SK信息进行验证，URL可以在ModelArts的部署上线->在线服务->指定服务->调用指南中获取

```
> python manage.py run --cfg algorithms/ivgSegmentation/config/sample/
config.py --pipeline infer --demo_type image_service --url {URL} --auth
algorithms/ivgSegmentation/config/sample/config.py
```

服务请求成功后会在./output/deeplabv3_resnet50_standard-sample_1024x512/vis文件夹中保存请求的可视化结果。

5.9.6.7.3 删除已发布模型和在线服务

已经部署的模型和在线服务可以在ModelAers控制台上手动删除，也可以在Notebook中通过ModelArts SDK删除。

注意

执行删除操作不可逆，请谨慎删除模型和在线服务！

1. 根据模型ID删除已部署模型。
```
from modelarts.session import Session
from modelarts.model import Model
def delete_model_by_id(model_id):
    sess = Session()
    model_instance = Model(session=sess, model_id="").model_instance
    model_instance.delete_model(model_id)
if __name__ == '__main__':
    model_id = '*************************'
    delete_model_by_id(auth, model_id)
```
2. 根据部署服务ID删除已部署服务。
```
from modelarts.session import Session
from modelarts.model import Model
def delete_service_by_id(service_id):
    sess = Session()
    model_instance = Model(session=sess, model_id="").model_instance
    model_instance.delete_service(service_id)
if __name__ == '__main__':
    service_id = '*************************'
    delete_service_by_id(service_id)
```

5.9.6.8 编排

```
# 根据flow config，一键运行所有任务
> python manage.py flow --cfg algorithms/ivgSegmentation/config/sample/config.py
```

5.9.6.9 查询

查询训练参数

```
# 查询训练规格列表
> python manage.py query train flavor --cfg algorithms/ivgSegmentation/config/sample/
config.py

# 查询训练引擎列表
> python manage.py query train framework --cfg algorithms/ivgSegmentation/config/sample/
config.py

# 查询专属池列表
```

```
> python manage.py query train pool --cfg algorithms/ivgSegmentation/config/sample/config.py
```

查询已部署模型和服务

```
# 查询已发布的模型
> python manage.py query deploy model --cfg algorithms/ivgSegmentation/config/sample/
config.py

# 查询已发布的在线服务
> python manage.py query deploy service --cfg algorithms/ivgSegmentation/config/sample/
config.py
```

查询在线服务详情

```
from modelarts.session import Session
from modelarts.predictor import Predictor


def query_service(service_id):
    sess = Session()
    model_instance = Predictor(session=sess, service_id=service_id)
    return model_instance.get_service_info(service_id)


if __name__ == '__main__':
    service_id = '*************************'
    service_info = query_service(service_id)
    print(service_info)
```

5.9.7 配置参数说明

5.9.7.1 全局参数

本小节介绍了算法开发套件配置文件中涉及到的全局参数。

平台参数

- obs_bucket: 表示使用的远程存储根目录
- auth: 表示账户信息及配置。详见鉴权参数

鉴权参数

- hwc: 是否是华为云环境，默认是True
- region_name: 华为云区域，默认是cn-north-4

运行参数

- alg_type: 表示使用的算法类型，基于`algorithms/${alg_type}`得到算法的目录
- alg_name: 表示使用的算法名称
- run_dir: 表示runner的工作目录，放在此处便于runner内部复用

- convert_dir: 表示converter的工作目录，放在此处便于converter内部复用

- export_dir: 表示exporter的工作目录，放在此处便于exporter内部复用

复用参数

- alg_cfg: 表示算法使用的共性参数，便于多处共享

5.9.7.2 Runner

本小节介绍了基于算法开发套件进行算法的训练、测试和推理等阶段的相关参数。

其中，训练、测试支持单节点或多节点分布式运行，单节点分布式启动时只需给定local_size，即会在一个机器上启动多次算法脚本，从而实现单节点多进程；多节点分布式启动则需要在各个node上指定int_method（当前仅支持TCPmethod）、rank（节点序号）、world_size（节点数）、local_size（单节点进程数）。

启动ModelArt远程分布式作业时，仅需传递world_size和local_size参数，其余参数ModelArts平台会自动传递。

通过算法外壳启动分布式会给算法脚本传递如下分布式参数，其中rank和world_size分别表示总进程序号和总进程数，local_rank表示单节点进程序号，需要脚本自行解析和配置，配置方式可以参考领域套件代码。

```
--launcher=utility  # 分布式启动方法，基于算法外壳启动称为utility
--init_method=${init_method}  # TCP初始化方法，指定IP和Port
--rank=${rank}  # 总进程序号
--world_size=${world_size}  # 总进程数
--local_rank=${local_rank}  # 单节点进程序号
--local_size=${local_size}  # 单节点进程数
```

公共参数

- alg_type：表示使用的算法类型，继承Global参数。

- run_dir：表示runner的工作目录，继承Global参数。

- pipeline：表示不同算法管控模式构成的流水线，允许command传入`--pipeline mode1,mode2,...`进行覆盖。如train， evaluate，infer。

- pipeline_maps：表示算法支持的管控模式及其对应的算法入口脚本。

算法参数

- common_args：表示算法的共性参数，所有mode的算法参数会基于其进行update，不存在时默认为空字典。

- ${mode}_args：表示pipeline_maps中各mode对应的特性参数，在特定mode进行运行时，会和common_args进行组装（common_args.update(mode_args)）以构成mode最终的参数，并且允许command传入算法的任意参数进行覆盖（通过argparse解析时的无法识别参数进行传递，因此不能与common_algorithm_utility的参数重叠）。

说明

value为None时不传递，value为''（空字符串）时作为`store_arg`传递，采用`position_args=[arg1, arg2, ...]`的形式传递位置参数，采用`arg=[val1, val2, ...]`的形式传递列表参数。

远程训练参数

adapter：对算法运行过程中的所有环境切换、依赖安装、数据下载、结果上传等功能进行管理与实现。对本地启动远程作业的方式进行管控。

5.9.7.3 Converter

本小节基于算法开发套件进行模型转换的相关参数。

公共参数

- alg_type: 表示使用的算法类型，继承Global参数。

- convert_dir: 表示converter的工作目录，继承Global参数。

- pipeline: 表示不同模型转换模式构成的流水线，允许command传入`--pipeline mode1,mode2,...`进行覆盖。

- pipeline_maps: 表示算法支持的模型转换模式及对应的类型及脚本，所有涉及code-based model的模型转换均由用户实现custom脚本调用套件能力进行转换，其余采用default的套件能力即可。

转换参数

- common_args: 表示模型转换的共性参数，使用方式同runner；不识别的参数不会被使用。

- ${mode}_args: 表示pipeline_maps中各mode对应的特性参数，使用方式同runner；针对custom模式的模型转换，通过`script_args`传入算法脚本所需的参数，同样可以通过command中unknown_args的形式进一步传递算法脚本所需的其他参数。

- config_file：表示配置文件的路径，None表示`${convert_dir}/config.json`。会将转换onnx模型的配置写入config。

- do_comparison：表示是否进行转换前后模型的输出进行比较，部分情况不支持比较。

模型转换支持的类型有：

- torch2onnx (custom)

- script_args：表示传递给算法自定义脚本的命令行参数，算法自定义脚本中，先构建好torch model，而后调用功能外壳的能力进行模型转换。具体使用方式请参见领域套件。

- input_shapes: 表示模型输入的形状，list[tuple]

- input_names: 表示模型输入的名字，list[str]

- output_names: 表示模型输出的名字，list[str]

- onnx_path：表示生成的onnx模型路径，None表示`${convert_dir}/model.onnx`

- opset_version：表示转换onnx的版本，默认使用当前torch的默认版本

- onnx2rt (default， only support TensorRT5.1.5 for now)

- onnx_path：表示生成的onnx模型路径，None表示`${convert_dir}/model.onnx`

- rt_path: 表示生成的tensorRT模型路径，None表示`${convert_dir}/model.rt`

- mode: 表示生成的tensorRT模型的类型，目前支持`fp32`和`int8`(int8目标仅支持单图像输入)

- output_names: 表示模型输出的名字，list[str]。如果设置，则会将忽略list之外的输出

- calib_cfg：`int8`模式下进行参数校正的配置，详见后续说明

- torch2caffe (custom，support pytorch>=1.1.0,<1.6.0])

- script_args：表示传递给算法自定义脚本的命令行参数，算法自定义脚本中，先构建好torch model，而后调用功能外壳的能力进行模型转换

- input_shapes: 表示模型输入的形状，list[tuple]

- input_names: 表示模型输入的名字，list[str]

- caffe_prototxt：表示生成的caffe结构文件路径，None表示`${convert_dir}/model.prototxt`

- caffe_caffemodel：表示生成的caffe权重文件路径，None表示`${convert_dir}/model.caffemodel`

- caffe2ascend (default, only support low version for now)

- caffe_prototxt：表示生成的caffe结构文件路径，None表示`${convert_dir}/model.prototxt`

- caffe_caffemodel：表示生成的caffe权重文件路径，None表示`${convert_dir}/model.caffemodel`

- om_path: 表示生成的D模型路径，None表示`${convert_dir}/model.om`

- insert_op_cfg：表示插入新opetator（aipp）的配置文件，默认不使用

- input_shapes: 表示模型输入的形状，list[tuple]。只在进行输出比较时需要

远程训练参数

- adapter: 对算法运行过程中的所有环境切换、依赖安装、数据下载、结果上传等功能进行管理与实现。对本地启动远程作业的方式进行管控。

5.9.7.4 Exporter

本小节介绍了基于算法开发套件进行内容导出的相关参数。

公共参数

- type: 表示导出的平台，目前仅支持local模式导出。

- alg_type: 表示使用的算法类型，继承Global参数。

- export_dir: 表示exporter的工作目录，继承Global参数。

- pipeline: 表示导出的模式构成的流水线，允许command传入`--pipeline mode1,mode2,...`进行覆盖。

- pipeline_maps: 表示算法支持的内容导出模式及对应的类型及脚本，暂时仅支持导出模型SDK；支持用户实现custom脚本进行自定义导出，也可以采用default的形式进行runtime-based的模型sdk导出。`default`形式仅支持领域套件，自定义算法请采用`custom`形式并自行实现相应脚本。

导出参数

- common_args: 表示内容导出的共性参数，使用方式同runner。

- ${mode}_args: 表示pipeline_maps中各mode对应的特性参数，使用方式同runner；针对custom的模型转换，通过script_args传入算法脚本所需的参数，同样可以通过command中unknown_args的形式进一步传递算法脚本所需的其他参数。

5.9.7.5 Deployer

本小节介绍了基于算法开发套件进行模型部署的相关参数。

公共参数

- requirements: 表示部署模型的依赖列表文件路径。

- service_name: 表示需要部署在线服务名称及模型发布的名称。

- deploy_dir: 表示deployer的工作目录，继承Global参数。

- model_type: 表示需要部署的模型依赖的运行框架。

- pred_img_path: 表示模型部署完成后需要进行测试的图片路径，如果省略这个参数，将会跳过预测过程。

- instance_type: 表示在线部署使用的规格。

5.9.7.6 Adapter

本小节介绍了基于算法开发套件进行环境管控的相关参数。

adapter主要负责：

1) 对代码运行环境进行配置，目前支持的环境配置内容包括：conda环境切换、pip依赖安装。

2) 基于功能外壳进行无感远程作业的相关参数,可以自动完成启动前数据上传、远程作业启动、远程作业监控和运行中/后数据下载等。

参数说明

- requirements: 依赖项，可以为指定的requirements文件，也可以是一个list，list可以放置requirements路径或者指定的package名称。

- framework_type：训练作业选择的引擎规格，必选，如PyTorch。

- framework_version：训练作业选择的引擎版本，必选，如PyTorch-1.4.0-python3.6-v2。

- instance_type：训练作业选择的资源规格，如modelarts.p3.large.eco。

- world_size：训练作业计算节点个数。可选，默认为1。

- pool_id：专属池ID，默认为True。

- downloads: 表示数据下载内容的源路径（src）和目标路径（dst），其中src和dst需一一对应。不下载可以置为None。

- uploads：表示启动远程作业前数据上传的源路径和目标路径。不上传可以置为None。

5.9.7.7 Flow

本小节介绍了基于算法开发套件进行自动化流水线构建的相关参数。目前仅仅支持串行流，以list进行表示，其中的node以dict进行表示，node一般对应于tools中的一个脚本，运行node本质上是直接运行相应脚本。

参数说明

- type: 表示node的类型，目前支持launch和default形式。launch类型会启动远程作业运行脚本，default类型则直接本地运行脚本。

- mode: 表示node的运行模式，launch type会采用对应mode进行launch，default type则直接执行对应的脚本。

- args: 表示node的参数，参数解析方式同runner，所有参数必须在此进行设定。

6 开发环境（旧版即将下线）

6.1 Notebook 简介

说明

本手册基于旧版开发环境Notebook功能展开介绍，旧版Notebook即将下线，新版Notebook相关内容请参见**开发工具**。

ModelArts集成了基于开源的Jupyter Notebook和JupyterLab，可为您提供在线的交互式开发调试工具。您无需关注安装配置，在ModelArts管理控制台直接使用Notebook，编写和调测模型训练代码，然后基于该代码进行模型的训练。

- Jupyter Notebook是一个交互式笔记本。关于Jupyter Notebook的详细操作指导，请参见**Jupyter Notebook使用文档**。
- JupyterLab是一个交互式的开发环境，是Jupyter Notebook的下一代产品，可以使用它编写Notebook、操作终端、编辑MarkDown文本、打开交互模式、查看csv文件及图片等功能。关于JupyterLab的详细操作指导，请参见**JupyterLab官网文档**。

其中，ModelArts还提供了华为自研的**分布式训练加速框架MoXing**，您可以在Notebook中使用MoXing编写训练脚本，让您代码编写更加高效、代码更加简洁。

支持的 AI 引擎

每个工作环境多种AI引擎，可以在同一个Notebook实例中使用所有支持的AI引擎，不同的引擎之间可快速、方便的切换，并且有独立的运行环境。其中，Conda-python2不包含任何AI引擎的基础Python2.7环境，Conda-python3不包含任何AI引擎的基础python3.6环境。

说明

- ModelArts的Notebook支持多引擎，即同一个Notebook实例，可以使用所有支持的AI引擎，不同引擎之间可快速、方便切换。
- 创建Notebook时，选择一种工作环境，包含多引擎，分为Python2和Python3两个不同版本的环境，推荐使用Multi-Engine 1.0 (python3)。请参考**表6-1**选择适用的工作环境。
- 不同Region支持的AI引擎不一样，请以控制台实际界面为准。

表 6-1 AI 引擎

工作环境名称	预置的AI引擎及版本	适配芯片
Multi-Engine 1.0 (Python3, Recommended)	MXNet-1.2.1	CPU/GPU
	PySpark-2.3.2	CPU
	Pytorch-1.0.0	GPU
	TensorFlow-1.13.1	CPU/GPU
	TensorFlow-1.8	CPU/GPU
	XGBoost-Sklearn	CPU
Multi-Engine 1.0 (python2)	Caffe-1.0.0	CPU/GPU
	MXNet-1.2.1	CPU/GPU
	PySpark-2.3.2	CPU
	PyTorch1.0.0	GPU
	TensorFlow-1.13.1	CPU/GPU
	TensorFlow-1.8	CPU/GPU
	XGBoost-Sklearn	CPU
Multi-Engine 2.0 (python3)	Pytorch-1.4.0	GPU
	R-3.6.1	CPU/GPU
	TensorFlow-2.1.0	CPU/GPU
Ascend-Powered-Engine 1.0 (python3)	MindSpore-1.1.1	Ascend 910
	TensorFlow-1.15.0	Ascend 910

使用限制

- 出于安全因素考虑，ModelArts集成的Notebook暂不开放用户root权限，可使用非特权用户jovyan或者ma-user（Multi-Engine引擎）进行操作，因此暂不能使用apt-get安装操作系统软件。
- 针对当前的AI引擎框架，Notebook仅支持单机模式训练模型。如果需要使用分布式模式训练模型，建议使用ModelArts训练作业，资源池设置多节点的方式实现。
- 开发环境暂不支持apt-get。您可以使用**自定义镜像**的方式训练模型。
- 不支持GUI图形相关的库，如pyqt等。
- 使用Ascend规格创建的Notebook，不支持挂载EVS。
- 不支持连接DWS服务以及数据库服务。
- 不支持直接读取OBS中的文件， 需要先下载到本地。如需访问OBS中的数据，建议使用**MoXing**或者**SDK**进行交互。
- 开发环境中不支持TensorBoard，建议使用“训练作业”下的“可视化作业”功能。

- 针对已创建好的Notebook实例，在创建完成后无法修改规格。例如将CPU规格修改为GPU、修改工作环境。因此，建议在创建实例时，选择业务所需的规格配置，或者在开发过程中，及时保存您的代码和数据至OBS，方便快速上传至新的Notebook中使用。
- 在关闭页面后，再次打开时依然运行代码看到输出，建议使用Terminal。
- 自行升级PyTorch等AI框架的版本后，请注意是否和当前cuda版本兼容。预置在Multi-Engine 2.0和 Multi-Engine 1.0工作环境中的cuda版本不同。切换cuda版本请参见**GPU类型实例，在Terminal中切换cuda版本**。

6.2 管理 Notebook 实例

6.2.1 创建 Notebook 实例

在开始进行模型开发前，您需要创建Notebook实例，并打开Notebook进行编码。

背景信息

- 创建和使用Notebook需要消耗资源，需要收费。根据您选择的资源不同，收费标准不同，针对不同类型资源的价格，详情请参见**产品价格详情**。
- “运行中”的Notebook将一直收费，当您不需要使用时，建议停止Notebook，避免产生不必要的费用。在创建Notebook时，也可以选择开启自动停止功能，在指定时间内停止运行Notebook，避免产生不必要的费用。
- 只有处于“运行中”状态的Notebook，才可以执行打开操作。
- 一个账户最多创建10个Notebook。默认显示当前用户创建的Notebook实例，如果将“只显示自己”的开关关闭时，将显示当前帐号及其IAM用户创建的所有Notebook实例。
- 由于ModelArts支持多种AI引擎，在创建Notebook实例时，首先选择Python2或Python3的工作环境，Notebook实例创建完成后，再访问Notebook实例，在Jupyter页面创建对应AI引擎的工作环境。
- 如果设置使用OBS存储，确保您指定的OBS目录与ModelArts在同一区域。
- 创建Notebook实例前，可先了解ModelArts**支持的AI引擎**及其对应版本情况。

创建 Notebook

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“开发环境>Notebook”，进入“Notebook”管理页面。

图 6-1 Notebook 页面

2. 单击“创建”进入“创建Notebook”页面，请参见如下说明填写参数。

   a. 填写Notebook基本信息，包含名称、描述、是否自动停止，详细参数请参见表6-2。

图 6-2 Notebook 基本信息

表 6-2 基本信息的参数描述

参数名称	说明
“计费方式”	按需计费。当前仅支持按需计费，无需修改。
“名称”	Notebook的名称。只能包含数字、大小写字母、下划线和中划线，长度不能超过64位且不能为空。
“描述”	对Notebook的简要描述。
“自动停止”	默认开启，且默认值为“1小时后”，表示该Notebook实例将在运行1小时之后自动停止，即1小时后停止计费。 开启自动停止功能后，可选择“1小时后”、“2小时后”、“4小时后”、“6小时后”或“自定义”几种模式。选择“自定义”模式时，可指定1~24小时范围内任意整数。

   b. 填写Notebook详细参数，如工作环境、资源规格等，详细参数请参见表6-3。

图 6-3 Notebook 实例的详细参数

表 6-3 Notebook 实例的详细参数说明

参数名称	说明
“工作环境”	当前仅支持公共镜像，即预置在ModelArts内部的AI框架。 每个工作环境多种AI引擎，可以在同一个Notebook实例中使用所有支持的AI引擎，不同的引擎之间可快速、方便的切换，并且有独立的运行环境。您可以在Notebook实例创建完成后，进入Jupyter页面创建对应AI引擎的开发环境。不同工作环境其对应可使用的AI引擎不同，详细支持列表请参见**支持的AI引擎**。 推荐使用“Multi-Engine 1.0 (Python3, Recommended)”工作环境。 如果需要使用TensorFlow 2.X、PyTorch 1.4.0或者R语言版本的AI框架，建议使用“Multi-Engine 2.0 (python3)”工作环境。如果选择此类型的工作环境，暂时无法使用免费规格，请选择其他付费的规格。 **说明** ModelArts还支持Keras引擎，选择“Multi-Engine 1.0 (Python3, Recommended)”或“Multi-Engine 1.0 (python2)”工作环境即可使用，详细说明请参见**ModelArts是否支持Keras引擎?**
“资源池”	可选“公共资源池”和“专属资源池”，关于ModelArts专属资源池的介绍和购买，请参见**资源池**。 “公共资源池”无需购买，即开即用，按需付费，即按您的Notebook实例运行时长进行收费。“专属资源池”的特点是无需排队，当使用公共资源池人数较多时，可能出现排队等待，此时建议购买“专属资源池”提升开发效率，减少等待时间。
“类型”	支持CPU、GPU和Ascend类型。GPU性能更佳，但是相对CPU而言，费用更高。Ascend类型为公测资源，请提前完成**Ascend 910公测申请**。
“规格”	只有选择“公共资源池”时，需要选择规格。根据选择的类型不同，可选规格也不同。 ● CPU规格支持：“[限时免费]体验规格CPU版”、“2核8GiB”、“8核32GiB” ● GPU规格支持：“[限时免费]体验规格GPU版”、“GPU: 1*v100NV32 CPU: 8 核 64GiB” ● Ascend规格支持：“Ascend: 1*Ascend 910 CPU: 24 核 96GiB” 如果选择“限时免费”规格，请仔细阅读界面提示，并勾选“我已阅读并同意以上内容”。

参数名称	说明
“存储配置”	存储配置可选“云硬盘”和“对象存储服务”。 ● 选择“云硬盘”作为存储位置 根据实际使用量设置磁盘规格。磁盘规格默认5GB。ModelArts提供5GB容量供用户免费使用。超出5GB时，超出部分每GB按“超高IO”类型的收费标准进行按需收费。磁盘规格的取值范围为5GB～4096GB。 选择此模式，用户在Notebook列表的所有文件读写操作都是针对容器中的内容操作，与OBS无关；重启该实例，内容不丢失。 ● 选择“对象存储服务”作为存储位置 在“存储位置”右侧单击“选择”，设置用于存储Notebook数据的OBS路径。如果想直接使用已有的文件或数据，可将数据提前上传至对应的OBS路径下。“存储位置”不能设置为OBS桶的根目录，需设置为对应OBS桶下的具体目录。 选择此模式，用户在Notebook列表的所有文件读写操作是基于所选择的OBS路径下的内容操作，与当前实例空间无关。如果您需要将内容同步到实例空间，先选中该内容，单击“Sync OBS”，即可将所选内容同步到当前容器空间，详细操作可参见**与OBS同步文件**。重启该实例时，内容不丢失。
“Git存储库”	只有当“存储配置”选择“云硬盘”时，支持此参数。 开启此功能后，可创建一个带有Git存储库的Notebook实例，系统将自动从Github同步代码库。详细配置说明请参见**创建带有Git存储库的Notebook实例**。

3. 参数填写完成后，单击“下一步”进行规格确认。
4. 参数确认无误后，单击“提交”，完成Notebook的创建操作。

   进入Notebook列表，正在创建中的Notebook状态为“启动中”，创建过程需要几分钟，请耐心等待。当Notebook状态变为“运行中”时，表示Notebook已创建完成。

6.2.2 创建带有 Git 存储库的 Notebook 实例

ModelArts Notebook支持下载GitHub的公共仓库和私有仓库，且可基于JupyterLab的git插件进行图形化操作。

使用前必读

- 目前Git存储库功能仅支持EVS类型的Notebook实例。创建Notebook实例时，只有当“存储配置”选择“云硬盘”时，才可以看到“Git存储库”的开关，且该开关默认是关闭的。
- 启用“Git存储库”后，配置详情如**图6-4**所示。“Git存储库”分为“私有仓库”和“公共仓库”。

图 6-4 启用 Git 存储库

- 配置好的代码库将下载到对应Notebook实例对应路径下（“/home/ma-user/work”）。此时可以使用JupyterLab的git插件开启愉快的Git之旅，详细操作请参见**使用Git插件**。
- 停止Notebook实例，并再次启动时，如果“/home/ma-user/work”目录下已存在同工程名的目录，将不会重复下载，并在work目录下生成git_exception.log文件，提示已有同名文件。如果目录不存在时，重启后将重新下载。

下载 GitHub 公共仓库

当“仓库类型”选择“公共仓库”后，输入GitHub上的组织名称，如jupyter，按“Enter”键或者单击击右边的搜索按钮，即可看到代码库和分支内容。如果单击搜索后，响应比较慢，请稍等片刻。

如果该组织下有多个代码库或多个分支，可以从“代码库”列表中选择需要下载的仓库名称，然后从“分支”列表中选择需要下载的分支名称。设置完成后，可单击“下一步”创建Notebook实例。

图 6-5 选择代码仓库和分支

下载 GitHub 私有仓库

目前支持通过OAuth授权的方式下载GitHub的私有仓库。

1. 在“仓库类型”选择“私有仓库”后，单击“使用OAuth授权”。

图 6-6 单击使用 OAuth 授权

2. 根据您的当前会话是否登录GitHub，操作不同。
   - 如果已经登录了GitHub，系统将会直接跳转到授权页面。
   - 如果用户未登录GitHub，页面自动跳转至登录页面，请输入正确的帐号密码，然后单击“Sign in”。登录成功后，界面会提示认证信息，单击“Authorize ModelArts-Lab”按钮开启授权。

图 6-7 登录页面

当界面提示“新建授权成功”时，表示已为此GitHub帐号完成授权。您可以手工关闭授权页面，并跳转至“创建Notebook”页面。

3. 在“创建Notebook”页面中，会提示您确认是否已完成授权，单击“确定”。

说明

如果在此对话框中单击“取消”，则OAuth授权失败，无法下载私有仓库的代码。

图 6-8 确认是否完成授权

4. 确定已完成授权后，可以下拉框中选择授权GitHub帐号下的仓库和分支。请依次选择“用户名/组织”、“仓库名称”和“分支”，前一个参数的设置将影响后一个参数的参数值，当下拉框中无内容时，请等候1-2分钟后重新选择。

图 6-9 选择用户名、仓库名称和分支

5. 当仓库和分支选择完成后，单击“下一步”创建Notebook实例。

查看带 Git 存储库的 Notebook

不管是公共仓库还是私有仓库，当带Git存储库的Notebook创建完成后，支持在不同位置查看仓库。

- 在Notebook列表中查看仓库地址

  在Notebook列表中，单击名称左侧的小三角，展开实例详情，查看此Notebook对应的仓库地址。

图 6-10 查看仓库地址

- 打开Notebook实例查看代码库。当Notebook实例启动成功，并打开Notebook后，可以看到代码库。

如果代码库下载成功，打开运行中的Notebook实例，可在Jupyter页面查看到对应的代码库。

图 6-11 查看到对应的代码库

如果打开project时其展示界面如图6-12所示，含有“git.log”文件。打开“git.log”，如果文件内容如图10 git.log文件内容所示，表示仓库正在下载中，根据仓库大小不同网络情况不同，下载时间不定，请耐心等待。如果是其他内容，请根据对应的提示进行排查或者提交工单由专业工程师为您处理。

图 6-12 仓库正在加载中

图 6-13 git.log 文件内容

6.2.3 打开 Notebook 实例

针对创建好的Notebook实例（即状态为“运行中”的实例），可以打开Notebook并在开发环境中启动编码。

打开 Notebook 实例

- 进入“Jupyter Notebook”开发页面

  在Notebook列表中，选择需要打开的Notebook，单击“操作”列中的“打开”，或者单击Notebook实例名称。

- 进入“JupyterLab”开发页面

  在Notebook列表中，选择需要打开的Notebook，单击“操作”列中的“打开JupyterLab”。

图 6-14 打开 Notebook 实例

编码开发

ModelArts提供两种开发环境进行编码开发，一种是Jupyter Notebook，还有一个是新增的JupyterLab。

- **Jupyter Notebook**：是一款基于网页的用于交互计算的应用程序。其可被应用于全过程计算：开发、文档编写、运行代码和展示结果。
- **JupyterLab**：是一个交互式的开发环境，是Jupyter Notebook的下一代产品，可以使用它编写Notebook、操作终端、编辑MarkDown文本、打开交互模式、查看csv文件及图片等功能。

6.2.4 启动或停止 Notebook 实例

由于运行中的Notebook将一直耗费资源，您可以通过停止操作，停止资源消耗。对于停止状态的Notebook，可通过启动操作重新使用Notebook。

登录ModelArts管理控制台，在左侧菜单栏中选择“开发环境>Notebook”，进入“Notebook”管理页面。执行如下操作停止或启动Notebook。

- **停止Notebook**：单击“操作”列的“停止”。只有处于“运行中”状态的Notebook可以执行停止操作。
- **启动Notebook**：单击“操作”列的“启动”。只有处于“停止”状态的Notebook可以执行启动操作。

6.2.5 删除 Notebook 实例

针对不再使用的Notebook，您可以删除Notebook以释放资源。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“开发环境>Notebook”，进入“Notebook”管理页面。
2. 在Notebook列表中，单击操作列的删除，在弹出的确认对话框中，确认信息无误，然后单击确定完成删除操作。

说明

Notebook删除后不可恢复，请谨慎操作。但是Notebook中创建的文件仍然存储在创建Notebook时指定的OBS中。

如果Notebook使用EVS存储，那么删除后，数据也将一并删除，请谨慎操作。

6.3 使用 Jupyter Notebook

6.3.1 Jupyter Notebook 简介

Jupyter Notebook是基于网页的用于交互计算的应用程序。其可被应用于全过程计算：开发、文档编写、运行代码和展示结果。

ModelArts集成了基于开源的Jupyter Notebook，无需安装配置，在创建Notebook实例后，打开实例即可进行开发。

Notebook Kernel 是什么

- 可以理解是独立的代码执行环境，在ModelArts Notebook中，提供了多种Kernel类型，如TensorFlow1.13.1、PyTorch1.0等。该环境中包含已经预装并调试好的AI引擎以及依赖。
- 当选择对应kernel打开Notebook，在Notebook后端会对应启动一个Ipython的进程作为运行环境，用于执行用户在页面中的代码、命令输入。
- 每个Kernel类型中包含一个独立的Conda运行环境，以确保AI引擎相互之间是独立的，举例来说，在Tensorflow类型Kernel中更新了Keras库，不会影响到MindSpore的Kernel中。

Notebook kernel 和常规的交互式 Python 解释器有什么区别

Notebook Kernel是一个IPython运行环境，可以理解是增强版本的Python Shell，直观来说相比Python解释器，可以执行Shell脚本，更多的可视化工具集成、Magic Command等等。详细可参见**IPython Documentation**。

6.3.2 Jupyter Notebook 常用操作

本章节罗列Jupyter Notebook的常用操作介绍。

打开 Jupyter Notebook

在Notebook列表中，选择需要打开的Notebook，单击“操作”列中的“打开”，进入“Jupyter Notebook”开发页面。

在“Jupyter Notebook”页面中，有“Files”、“Running”、“ModelArts Examples”3个页签。

图 6-15 Jupyter Notebook 开发界面

选择不同 AI 引擎新建文件

打开Notebook实例后，进入“Jupyter Notebook”页面，在“Files”页签下，您可以单击右上角“New”，然后选择所需的AI引擎，创建一个用于编码的文件。

图 6-16 选择不同的 AI 引擎

上传文件

打开Notebook实例后，进入“Jupyter Notebook”页面，在“Files”页签下，您可以单击右上角“Upload”，可从本地选择一个文件上传。

此功能上传的文件大小有一定限制，如果您的文件大小超过限制，建议使用其他方式上传，详细请参见**上传大文件至Notebook中**。

图 6-17 上传文件

编写文件

新建文件后，单击文件名称即可进入编写文件页面。

图 6-18 编写文件

表 6-4 编写文件页面介绍

区域	区域名称	详细说明
1	文件名称	您可以在此区域填写自定义的文件名称，修改保存后，对应的File列表也将发生变化。
2	菜单栏	菜单栏中有File、Edit、View、Insert、Cell、Kernel、Help等丰富功能。详细说明建议参考**Jupyter Notebook使用文档**，下方工具栏提供了常用的功能，能够满足常见的Python运行文件编写。
3	工具栏	工具栏罗列了支持的常用快捷操作，从左到右分别为：保存文件、添加新Cell、剪切选中的Cell、复制选中的Cell、粘贴选中的Cell、将选中Cell上移、将选中Cell下移、运行选中的Cell、终止kernel、重启kernel、重启kernel并重新运行所有Cell。 Code下拉框中有四个选项： ● Code：写Python代码 ● MarkDown：写MarkDown代码，通常用于注释 ● Raw NBConvert：转换工具 ● Heading：快捷添加MarkDown标题
4	AI引擎及Python版本	显示当前文件对应的AI引擎以及Python版本。请参见**支持的AI引擎**了解目前ModelArts支持的所有AI引擎和Python版本。
5	代码Cell	代码单元格。每一个Cell有两种模式：命令模式和编辑模式。 最左侧为蓝色条时，是命令模式，绿色条表示编辑模式（此时Cell中有光标，可以进行代码编写）。在命令模式下，按下“Enter”键或者鼠标单击代码框可以进入编辑模式。在编辑模式下，按下“ESC”键或者鼠标单击代码框左侧区域即可进入命令模式。

删除文件或文件夹

如果需要在Jupyter Notebook中删除文件或文件夹，您可以在“Files”列表中勾选待删除的文件或文件夹，然后单击上方的红色删除按钮，即可删除选中的文件或文件夹。

说明

文件或文件夹删除完成后，需单击右上角的刷新按钮刷新Jupyter页面，清除缓存文件。

图 6-19 文件列表刷新功能

解压文件

将文件以压缩包形式上传至Notebook后， 可在Notebook Terminal中解压压缩包。

```
unzip xxx.zip  #在xxx.zip压缩包所在路径直接解压
```

解压命令的更多使用说明可以在主流搜索引擎中查找Linux解压命令操作。

如何打开Notebook Terminal，参见**打开Notebook的Terminal功能**。

6.3.3 Jupyter Notebook 环境配置

6.3.3.1 使用 Notebook Terminal 功能

对于习惯编码的开发者来讲，Terminal功能非常便捷实用。本章节介绍如何在Notebook实例中打开Terminal功能，同时，在Terminal中切换引擎环境。

打开 Notebook 的 Terminal 功能

1. 创建Notebook实例之后，在Notebook列表中，单击目标Notebook“操作”列的“打开”，进入“Jupyter Notebook”开发页面。
2. 在Jupyter页面的“Files”页签下，单击“New”，然后选择“Terminal”，进入到Terminal界面。

图 6-20 进入 Terminal 界面

在 Terminal 切换各引擎的环境

您可以在Jupyter的Terminal环境中切换到其他AI引擎环境。

1. 在Notebook列表中，创建并打开一个Notebook，或者直接打开已有的Notebook。

2. 在Jupyter页面的“Files”页签下，单击“New”，然后选择“Terminal”，进入到Terminal界面。
3. 您可以查看当前目录下提供的“README”切换环境，例如需要切换到“TensorFlow-1.8”，可执行命令**source /home/ma-user/anaconda3/bin/activate TensorFlow-1.8**进入到TensorFlow-1.8的环境并进行开发，若要退出则可执行命令**source deactivate**。

图 6-21 执行命令

```
sh-4.3$cat README
Please use one of following command to start the specifed framework environment.

for Conda-python3 ------------------------ source /home/ma-user/anaconda3/bin/activate base
for MXNet-1.2.1 -------------------------- source /home/ma-user/anaconda3/bin/activate MXNet-1.2.1
for PySpark-2.3.2 ------------------------ source /home/ma-user/anaconda3/bin/activate PySpark-2.3.2
for Pytorch-1.0.0 ------------------------ source /home/ma-user/anaconda3/bin/activate Pytorch-1.0.0
for TensorFlow-1.13.1 -------------------- source /home/ma-user/anaconda3/bin/activate TensorFlow-1.13.1
for TensorFlow-1.8 ----------------------- source /home/ma-user/anaconda3/bin/activate TensorFlow-1.8
for XGBoost-Sklearn ---------------------- source /home/ma-user/anaconda3/bin/activate XGBoost-Sklearn
```

6.3.3.2 GPU 类型实例，在 Terminal 中切换 cuda 版本

针对GPU类型的Notebook实例，您可以在Jupyter的Terminal环境中切换不同版本的cuda。

由于CPU类型的Notebook实例不涉及cuda，因此以下操作仅针对GPU类型的Notebook实例。

1. 在Notebook列表中，创建并打开一个Notebook，或者直接打开已有的Notebook。
2. 在Jupyter页面的“Files”页签下，单击“New”，然后选择“Terminal”，进入到Terminal界面。
3. 进入到目录“/usr/local”下。
```
cd /usr/local
```
4. 例如，需要切换到cuda10版本，可执行如下命令。
```
sudo ln -snf /usr/local/cuda-10.0 cuda
```

图 6-22 切换 cuda 版本的示例

```
sh-4.3$cd /usr/local/
sh-4.3$sudo ln -snf /usr/local/cuda-10.0 cuda
sh-4.3$ll
total 24
drwxr-xr-x 16 root    root     4096 May 12 14:41 ./
drwxr-xr-x 11 root    root      109 Mar  8 15:37 ../
drwxr-xr-x  2 root    root     4096 May 12 14:10 bin/
drwxr-xr-x  5 root    root       46 Apr 18 15:09 cloudguard/
lrwxrwxrwx  1 root    root       20 May 12 14:41 cuda -> /usr/local/cuda-10.0/
drwxr-xr-x 11 root    root     4096 Mar 12 00:39 cuda-10.0/
drwxr-xr-x 11 root    root     4096 Mar 12 00:30 cuda-8.0/
drwxr-xr-x 11 root    root     4096 Mar 12 00:32 cuda-9.0/
drwxr-xr-x  2 root    root        6 Aug  2  2017 etc/
drwxr-xr-x  2 root    root        6 Aug  2  2017 games/
drwxr-xr-x  2 root    root        6 Aug  2  2017 include/
drwxr-xr-x  4 root    root       38 Mar 11 23:17 lib/
lrwxrwxrwx  1 root    root        9 Aug  2  2017 man -> share/man/
drwxr-xr-x  9 root    root     4096 May  7 06:55 nvidia/
drwxr-xr-x  2 root    root        6 Aug  2  2017 sbin/
drwxr-x---  3 ma-user ma-group  141 Apr 18 15:19 seccrypto/
drwxr-xr-x  6 root    root       64 Apr 18 15:19 share/
drwxr-xr-x  2 root    root        6 Aug  2  2017 src/
```

6.3.3.3 安装外部库和内核

ModelArts Notebook中已安装Jupyter、Python程序包等多种环境，包括TensorFlow、MXNet、Caffe、PyTorch、Spark等。为了方便使用，您也可以使用pip install在Jupyter Notebook或Terminal中安装外部库。

在 Jupyter Notebook 中安装

如下操作以在Notebook中安装Shapely为例。

1. 在ModelArts管理控制台，进入“开发环境>Notebook”，并在Notebook列表中，打开一个Notebook。
2. 在Jupyter页面中，选择“New”，然后选择相应的AI引擎。
3. 在代码输入栏输入以下命令安装Shapely。

   pip install shapely

在 Terminal 中安装

如下操作以安装Shapely为例，在Notebook实例terminal中使用pip安装Shapely。

1. 在ModelArts管理控制台，进入“开发环境>Notebook”，并在Notebook列表中，打开一个Notebook。
2. 在Jupyter页面中，选择“New”（新建），然后选择“terminal”。
3. 如非Multi-Engine的实例在代码输入栏输入以下命令安装Shapely。

   /opt/conda/envs/python27_tf/bin/pip install Shapely
4. Multi-Engine的实例存在多个引擎，在“/home/ma-user/”路径下提供了README，可参照README切换到相应的引擎环境中安装包，如在TensorFlow-1.13.1中安装Shapely可按照如下步骤操作：

```
source /home/ma-user/anaconda3/bin/activate TensorFlow-1.13.1
pip install shapely
```

TensorFlow、MXNet、PyTorch、Caffe、Scikit-learn & XGBoost和Spark算法引擎在terminal中的Python路径请参见表6-5，其中pip也是在以下路径目录下。Multi-Engine的实例可参考提供的README。

表 6-5 AI 引擎及路径说明

AI引擎	版本	Python路径
TensorFlow	TF-1.8.0-python2.7	/opt/conda/envs/python27_tf/bin/python
TensorFlow	TF-1.8.0-python3.6	/opt/conda/envs/python36_tf/bin/python
MXNet	MXNet-1.2.1-python2.7	/opt/conda/envs/python27_mxnet/bin/python
MXNet	MXNet-1.2.1-python3.6	/opt/conda/envs/python36_mxnet/bin/python
PyTorch	PyTorch-1.0.0-python2.7	/opt/conda/envs/python27_pytorch/bin/python

AI引擎	版本	Python路径
PyTorch	PyTorch-1.0.0-python3.6	/opt/conda/envs/python36_pytorch/bin/python
Caffe	Caffe-1.0.0-python2.7	/opt/conda/envs/python27_caffe/bin/python
Scikit-learn & XGBoost	ML-1.0.0-python2.7	/opt/notebook/anaconda2/bin/python
Spark	Spark-2.2.0-python2.7	
Scikit-learn & XGBoost	ML-1.0.0-python3.6	/opt/notebook/anaconda3/bin/python
Spark	Spark-2.2.0-python3.6	

说明

由于在创建ModelArts训练作业时，是启动一个新的独立运行环境，不会关联在Notebook环境中安装的包；因此需要在启动代码中，导入安装包前添加：os.system('pip install xxx')。

例如，在训练作业中需要使用依赖包Shapely，在Notebook中安装完成后，需要在启动代码中添加以下代码：

```
os.system('pip install Shapely')
import Shapely
```

6.3.4 数据的上传和下载

6.3.4.1 上传大文件至 Notebook 中

在“Notebook”页面中，通过单击“Upload”按钮上传文件，详细操作可参见**Jupyter Notebook常用操作**章节的**上传文件**。当在Notebook或JupyterLab上传文件提示受限时，您可以先将大文件上传到OBS中，然后根据不同场景将大文件下载到Notebook中。

步骤 1：上传文件至 OBS

针对大文件场景，由于OBS管理控制台对文件大小和数量限制较多，所以推荐使用OBS工具上传大文件，如OBS Browser+或obsutil工具上传。OBS Browser+是一个比较常用的图形化工具，，支持完善的桶管理和对象管理操作。推荐使用此工具创建桶或上传对象。obsutil是一款用于访问管理OBS的命令行工具，对于熟悉命令行程序的用户，obsutil是执行批量处理、自动化任务的好的选择。OBS上传文件指导，请参见**《OBS工具指南》**。

您可以通过以下多种方式将文件上传至桶，OBS最终将这些文件以对象的形式存储在桶中。

表 6-6 不同访问方式上传对象的方法

访问方式	上传对象方法
控制台	**通过控制台上传对象**
OBS Browser+	**通过OBS Browser+上传对象**
obsutil	**通过obsutil上传对象**
SDK	**使用SDK上传对象**，具体参考各语言开发指南的上传对象章节
API	**PUT上传**、**POST上传**

步骤 2：从 OBS 中将文件下载至 Notebook

由于Notebook实例可以挂载OBS或EVS作为存储位置，针对不同的实例，请操作方式不同。

- 对于挂载EVS的Notebook实例下载文件
  - 使用**Moxing操作OBS文件**将OBS中的文件同步到Notebook后进行操作。

    读取一个OBS文件。例如读取“obs://bucket_name/obs_file.txt”文件内容，返回string（字符串类型）。

    ```
    file_str = mox.file.read('obs://bucket_name/obs_file.txt')
    ```

    也可以使用打开文件对象并读取的方式来实现，两者是等价的。

    ```
    with mox.file.File('obs://bucket_name/obs_file.txt', 'r') as f:
      file_str = f.read()
    ```
  - 使用ModelArts SDK的**从OBS下载数据**接口将OBS中的文件下载到Notebook后进行操作。

    说明

    当单个文件大小超过5GB时，无法使用此方式上传。推荐使用MoXing接口上传大文件。

    示例代码：

    ```
    from modelarts.session import Session
    session = Session()
    session.download_data(bucket_path="/bucket-name/dir1/sdk.txt", path="/home/user/sdk/obs.txt")
    ```
- 对于带OBS存储的Notebook实例下载文件

  将文件上传至创建Notebook实例时指定的OBS路径，然后使用Sync OBS功能将OBS中的文件同步到Notebook即可。

步骤 3：解压文件（可选）

如果是将文件以压缩包格式上传至Notebook， 可以在Terminal中解压压缩包。

```
unzip xxx.zip  #在xxx.zip压缩包所在路径直接解压
```

解压命令的更多使用说明可以在主流搜索引擎中查找Linux解压命令操作。

6.3.4.2 下载 Notebook 内的大文件到本地

Notebook内只允许下载100MB以内的文件到本地。您可以根据创建Notebook时选择存储位置的不同，分场景进行操作。

挂载 EVS 的 Notebook 实例

对于挂载EVS的Notebook实例，您可以执行以下操作将大文件下载到本地：

1. 在Notebook中，新建一个“ipynb”文件，使用MoXing先将大文件从Notebook上传到OBS中，示例代码如下：
```
import moxing as mox
mox.file.copy('/home/ma-user/work/obs_file.txt', 'obs://bucket_name/obs_file.txt')
```
其中“/home/ma-user/work/obs_file.txt”为文件在Notebook中的存储路径，“obs://bucket_name/obs_file.txt”为该文件上传到OBS的存储路径。
2. 使用OBS或ModelArts SDK将OBS中的文件下载到本地。
   - 方式一：使用OBS进行下载

     使用OBS下载文件的操作指导：**下载文件**，可以将样例中的“obs_file.txt”下载到本地。如果您的数据较多，推荐OBS Browser+下载数据或文件夹。
   - 方式二：使用ModelArts SDK进行下载
     i. 在您的本地环境**下载并安装ModelArts SDK**。
     ii. 完成ModelArts SDK的**Session鉴权**。
     iii. 将OBS中的文件下载到本地，详请参见**从OBS下载数据**。示例代码如下：
```
from modelarts.session import Session
session=Session(access_key='***',secret_key='***',project_id='***',region_name='***')
session.download_data(bucket_path="/bucket_name/obs_file.txt",path="/home/user/
obs_file.txt")
```

带 OBS 存储的 Notebook 实例

对于使用OBS存储的Notebook实例，您可以使用OBS或ModelArts SDK将OBS中的文件下载到本地。

- 方式一：使用OBS进行下载

  使用OBS下载文件的操作指导：**下载文件**，可以将OBS中的文件下载到本地。如果您的数据较多，推荐OBS Browser+下载数据或文件夹。
- 方式二：使用ModelArts SDK进行下载
  a. 在您的本地环境**下载并安装ModelArts SDK**。
  b. 完成ModelArts SDK的**Session鉴权**。
  c. 将OBS中的文件下载到本地，详请参见**从OBS下载数据**。示例代码如下：
```
from modelarts.session import Session
session=Session(access_key='***',secret_key='***',project_id='***',region_name='***')
session.download_data(bucket_path="/bucket_name/obs_file.txt",path="/home/user/
obs_file.txt")
```

6.3.5 使用 ModelArts 示例

ModelArts提供了Notebook示例，方便初学者通过示例，快速了解如何使用ModelArts Notebook。

说明

Jupyter Notebook示例使用nbexamples扩展，有关nbexamples扩展的更多信息，请参阅**nbexamples**。

预览 ModelArts Examples

1. 在Notebook列表中，创建并打开一个Notebook，或者直接打开已有的Notebook。
2. 在Jupyter页面中，单击“ModelArts Examples”页签，此页面罗列了“Basic ModelArts Python Sdk”、“Machine Learning Introduction”等示例。每个示例提供了详细的说明，您可以单击右侧的“Preview”预览示例。

图 6-23 进入 ModelArts Examples

使用 ModelArts Examples

1. 在Notebook列表中，创建并打开一个Notebook，或者直接打开已有的Notebook。
2. 在Jupyter页面中，单击“ModelArts Examples”页签，选择需要使用的示例，单击示例右侧的“Use”。
3. 在弹出的“Create a copy in your home directory”对话框中，设置新的“ipynb”文件名称，也可以直接使用默认文件，然后单击“Create copy”保存并打开新的“ipynb”文件。打开的示例文件如图6-25所示。

   使用示例是指将示例文件创建一个副本，其代码内容与示例一致。

图 6-24 创建示例副本

图 6-25 打开示例文件

6.3.6 使用 ModelArts SDK

在Notebook中，通过使用ModelArts SDK，可以完成OBS管理、训练作业管理、模型管理以及在线服务管理。

ModelArts SDK使用请参见《**ModelArts SDK参考**》。

在Notebook中，已承载了登录用户的鉴权信息（AK/SK）和区域信息，因此SDK session鉴权时，无需输入参数即可完成session鉴权。

示例代码

- 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='/obs-bucket-name/src/',                           # 训练脚本目录
                    boot_file='/obs-bucket-name/src/pytorch_sentiment.py',      # 训练启动脚本目录
                    log_url='/obs-bucket-name/log/',                            # 训练日志目录
                    hyperparameters=[
                                {"label":"classes",
                                 "value": "10"},
                                {"label":"lr",
                                 "value": "0.001"}
                                ],
                    output_path='/obs-bucket-name/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_instance = estimator.fit(inputs='/obs-bucket-name/data/train/', wait=False,
job_name='my_training_job')
```

- 查询模型列表
```
from modelarts.session import Session
from modelarts.model import Model
session = Session()
model_list_resp = Model.get_model_list(session, model_status="published", model_name="digit",
order="desc")
```
- 查询服务详情
```
from modelarts.session import Session
from modelarts.model import Predictor
session = Session()
predictor_instance = Predictor(session, service_id="input your service_id")
predictor_info_resp = predictor_instance.get_service_info()
```

6.3.7 与 OBS 同步文件

如您在创建Notebook实例时，选择了OBS的“存储位置”，您写的代码会自动存储到您选择的OBS目录下。如您需要不同ipynb文件进行代码的相互调用，则可以使用Sync OBS功能。

Sync OBS功能是将在Notebook实例文件list列表选中的对象从OBS桶路径下同步到当前容器目录“~/work”下。

注意事项

- 单次同步文件大小最大为500MB，个数最多是1024个。
- 同步的对象总大小不超过5GB。例如，当前容器目录“~/work”下已有2GB的文件时，用户使用Sync OBS最多只能同步3GB的文件。
- Sync OBS功能只在带有OBS存储的实例上存在，因为非OBS存储的Notebook实例，其所有的文件读写操作都在用户容器里，即在“~/work”容器目录。

操作步骤

Sync OBS功能支持在Notebook使用，其对应的操作指导如下所示。

例如，“Example1.ipynb”需要调用“Example2.ipynb”中的“module”。选中这两个“ipynb”文件，然后单击界面上的“Sync OBS”，同步成功后，即可在代码间相互调用。

图 6-26 使用 Sync OBS 功能

6.3.8 使用 Convert to Python File 功能

代码开发完成后，还支持将您输入的代码保存为“.py”文件，用于ModelArts训练作业的启动文件。

说明

JupyterLab目前暂时不支持Convert to Python File功能。

1. 在Notebook列表中，创建并打开一个Notebook，或者直接打开已有的Notebook。
2. 在Jupyter页面中，单击“New”，然后在列表下选择所需的AI引擎进入代码开发界面。
3. 在开发界面完成代码编写后，单击左上角的保存按钮。然后单击“Convert to Python File”将当前的“ipynb”转化为“python”文件，此功能可直接将您输入的代码保存为“.py”文件到工作目录。

   生成的“.py”文件可用于ModelArts训练作业的启动文件。

   图 6-27 Convert to Python File

4. 在弹出的对话框中，根据实际情况填写文件名称，然后勾选或去勾选“Force overwrite if file already exists.”，默认为不勾选，表示当目录下存在相同名称文件时，不会执行覆盖操作。然后单击“Convert”完成操作。

   图 6-28 设置并保存

6.3.9 远程 GPU 运行（Notebook2.0）

通过Notebook调试、训练一个AI模型，推荐按照如下逻辑步骤组织notebook的代码：

步骤1. 将训练数据下载到notebook工作目录下(/home/ma-user/work/)或者其子目录中；

步骤2. 对下载后的数据预处理；

步骤3. 定义机器/深度学习模型；

步骤4. 使用训练数据对模型进行训练；

步骤5. 训练完毕后将模型文件保存在当前目录下的子目录中；

步骤6. 加载模型文件，进行推理。

创建Notebook的时候不分配GPU资源，只有需要运行一些计算密集型的代码（如上步骤4）时，实时地通过远程GPU运行。

1. notebook编辑调试阶段，用户行为保持notebook原有行为一致；
2. 编辑结束，需要运行代码段的时候，如步骤1所涉及的代码没有在调试阶段执行，可先将步骤1中所涉及的代码块执行（选中相应Cell，点击**Run Selected Cells**按钮）；
3. 点击**GPU Training**开关，所有代码Cell上端将显示一个tag标记，默认都是**include**，将步骤2-5中所有的Cell保持**include**标记，其他Cell（包括步骤1所有Cell）点击切换成**skip**标记（远端会依次执行**include**标记的代码段）；
4. 点击**Submit**按钮，选择计算实例规格后点击**立即执行**；
5. 2-5步骤中的代码段将会在拥有GPU资源的远端服务器执行，每个Cell的日志将会在Cell的下方实时输出，直到所有Cell执行完毕；
6. 通过步骤5中生成的模型文件恢复模型，进行推理验证；
7. 重复如上流程，调试模型。

使用限制

- 通过远程GPU执行的代码块并不会在notebook中同时执行，如步骤6中的代码段依赖前面步骤的变量定义，需要在notebook中按需执行
- 如果当前平台GPU资源紧缺，远程执行时可能会有资源排队等待延迟

6.4 使用 JupyterLab

6.4.1 JupyterLab 简介及常用操作

JupyterLab是一个交互式的开发环境，是Jupyter Notebook的下一代产品，可以使用它编写Notebook、操作终端、编辑MarkDown文本、打开交互模式、查看csv文件及图片等功能。

可以说，JupyterLab是开发者们下一阶段更主流的开发环境。JupyterLab支持更加灵活和更加强大的项目操作方式，但具有和Jupyter Notebooks一样的组件。

ModelArts同时支持Jupyter Notebook和JupyterLab，您可以在同一个Notebook实例中，使用不同的工具进行代码开发。

打开 JupyterLab

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“开发环境 > Notebook”，进入Notebook管理列表。
2. 选择状态为“运行中”的Notebook实例，单击操作列的“打开”访问Notebook。

3. 在Jupyter页面中，单击右上角的“Open JupyterLab”，可直接打开此Notebook实例对应的JupyterLab页面。

图 6-29 进入 JupyterLab

4. 进入JupyterLab页面后，自动打开Launcher页面，如下图所示。您可以使用开源支持的所有功能，详细操作指导可参见**JupyterLab官网文档**。

图 6-30 JupyterLab 主页

新建并打开 Notebook

进入JupyterLab主页后，可在“Notebook”区域下，选择适用的AI引擎，单击后将新建一个对应框架的Notebook文件。

由于每个Notebook实例选择的工作环境不同，其支持的AI框架也不同，下图仅为示例，请根据实际显示界面选择AI框架，ModelArts支持的所有框架版本及Python版本请参见**支持的AI引擎**。

图 6-31 选择 AI 引擎并新建一个 Notebook

新建的Notebook文件将呈现在左侧菜单栏中。

图 6-32 新建文件

新建文件并打开 Console

Console的本质为Python终端，输入一条语句就会给出相应的输出，类似于Python原生的IDE。

进入JupyterLab主页后，可在“Console”区域下，选择适用的AI引擎，单击后将新建一个对应框架的Notebook文件。

由于每个Notebook实例选择的工作环境不同，其支持的AI框架也不同，下图仅为示例，请根据实际显示界面选择AI框架。

图 6-33 选择 AI 引擎并新建一个 Console

文件创建成功后，将直接呈现Console页面。

图 6-34 新建文件（Console）

上传文件

进入JupyterLab页面后，您可以单击左上角“Upload File”快捷键，从本地选择一个文件上传。

此功能上传的文件大小有一定限制，如果您的文件大小超过限制，建议使用其他方式上传，详细请参见**数据上传至JupyterLab**。

图 6-35 上传文件

编辑文件

JupyterLab可以在同一个窗口同时打开几个Notebook或文件（如HTML、TXT、Markdown等），以页签形式展示。

JupyterLab的一大优点是，可以任意排版多个文件。在右侧文件展示区，您可以拖动打开文件，随意调整文件展示位置，可以同时打开多个文件。

图 6-36 多文件任意编排

当在一个Notebook中写代码时，如果需要实时同步编辑文件并查看执行结果，可以新建该文件的多个视图。

打开此文件，然后单击菜单栏“File>New View for Notebook”，即可打开多个视图。

图 6-37 同一个文件的多个视图

下载文件至本地

在JupyterLab中开发的文件，可直接下载至本地。此功能对文件大小有一定限制，如果您的文件大小超过限制，建议使用其他方式下载，详细请参见**从JupyterLab下载至本地**。

在JupyterLab文件列表中，选择需要下载的文件，单击右键，在操作菜单中选择“Download”下载至本地。下载的目的路径，为您浏览器设置的下载目录。

图 6-38 下载文件

JupyterLab 常用快捷键和插件栏

图 6-39 JupyterLab 常用快捷键和插件栏

表 6-7 快捷键说明

快捷键	说明
+	打开Launcher页面，可快速创建新的Notebook、Console或其他文件。
	创建文件夹。
	上传文件。详细说明可参见上传文件
	更新文件夹。

快捷键	说明
	Git插件，可连接此Notebook实例关联的Github代码库。详细使用指导可参见**使用Git插件**。

表 6-8 插件栏常用插件说明

插件	说明
	文件列表。单击此处，将展示此Notebook实例下的所有文件列表。
	罗列ModelArts Examples。您可以单击列表中的任意一个示例，查看其代码及配套说明。 不同Region支持的Examples不一样，请以控制台实际界面为准。
	当前实例中正在运行的Terminal和Kernel。
	Git插件，可以方便快捷的使用Github代码库。详细指导请参见**使用Git插件**。
	快速启动命令。
	查看正在打开的文件页签。
	文档结构图。

6.4.2 数据的上传和下载

6.4.2.1 数据上传至 JupyterLab

在JupyterLab页面中，通过单击“Upload Files”快捷键上传文件，详细操作可参见**JupyterLab简介及常用操作**章节的**上传文件**。当在Notebook或JupyterLab上传文件提示受限时，您可以先将大文件上传到OBS中，然后根据不同场景将大文件下载到Notebook中。

步骤 1：上传文件至 OBS

针对大文件场景，由于OBS管理控制台对文件大小和数量限制较多，所以推荐使用OBS工具上传大文件，如OBS Browser+或obsutil工具上传。OBS Browser+是一个比较常用的图形化工具，支持完善的桶管理和对象管理操作。推荐使用此工具创建桶或上传对象。obsutil是一款用于访问管理OBS的命令行工具，对于熟悉命令行程序的用

户，obsutil是执行批量处理、自动化任务的好的选择。OBS上传文件指导，请参见《OBS工具指南》。

您可以通过以下多种方式将文件上传至桶，OBS最终将这些文件以对象的形式存储在桶中。

表 6-9 不同访问方式上传对象的方法

访问方式	上传对象方法
控制台	通过控制台上传对象
OBS Browser+	通过OBS Browser+上传对象
obsutil	通过obsutil上传对象
SDK	使用SDK上传对象，具体参考各语言开发指南的上传对象章节
API	PUT上传、POST上传

步骤 2：从 OBS 中将文件下载至 Notebook

由于Notebook实例可以挂载OBS或EVS作为存储位置，针对不同的实例，请操作方式不同。

- 对于挂载EVS的Notebook实例下载文件
  - 使用Moxing操作OBS文件将OBS中的文件同步到Notebook后进行操作。

    读取一个OBS文件。例如读取“obs://bucket_name/obs_file.txt”文件内容，返回string（字符串类型）。

    ```
    import moxing as mox
    file_str = mox.file.read('obs://bucket_name/obs_file.txt')
    ```

    也可以使用打开文件对象并读取的方式来实现，两者是等价的。

    ```
    import moxing as mox
    with mox.file.File('obs://bucket_name/obs_file.txt', 'r') as f:
      file_str = f.read()
    ```

  - 使用ModelArts SDK的从OBS下载数据接口将OBS中的文件下载到Notebook后进行操作。

    说明

    当单个文件大小超过5GB时，无法使用此方式上传。推荐使用MoXing接口上传大文件。

    示例代码：

    ```
    from modelarts.session import Session
    session = Session()
    session.download_data(bucket_path="/bucket-name/dir1/sdk.txt", path="/home/user/sdk/obs.txt")
    ```

- 对于带OBS存储的Notebook实例下载文件

  将文件上传至创建Notebook实例时指定的OBS路径，然后使用Sync OBS功能将OBS中的文件同步到Notebook即可。

步骤 3：解压文件（可选）

如果是将文件以压缩包格式上传至Notebook， 可以在Terminal中解压压缩包。

```
unzip xxx.zip  #在xxx.zip压缩包所在路径直接解压
```

解压命令的更多使用说明可以在主流搜索引擎中查找Linux解压命令操作。

6.4.2.2 从 JupyterLab 下载至本地

JupyterLab内只允许下载100MB以内的文件到本地。您可以根据创建Notebook时选择存储位置的不同，分场景进行操作。

挂载 EVS 的 Notebook 实例

对于挂载EVS的Notebook实例，您可以执行以下操作将大文件下载到本地：

1. 在Notebook中，新建一个“ipynb”文件，使用MoXing先将大文件从Notebook上传到OBS中，示例代码如下：
   ```
   import moxing as mox
   mox.file.copy('/home/ma-user/work/obs_file.txt', 'obs://bucket_name/obs_file.txt')
   ```
   其中“/home/ma-user/work/obs_file.txt”为文件在Notebook中的存储路径，“obs://bucket_name/obs_file.txt”为该文件上传到OBS的存储路径。
2. 使用OBS或ModelArts SDK将OBS中的文件下载到本地。
   - 方式一：使用OBS进行下载

     使用OBS下载文件的操作指导：**下载文件**，可以将样例中的“obs_file.txt”下载到本地。如果您的数据较多，推荐OBS Browser+下载数据或文件夹。
   - 方式二：使用ModelArts SDK进行下载
     i. 在您的本地环境**下载并安装ModelArts SDK**。
     ii. 完成ModelArts SDK的**Session鉴权**。
     iii. 将OBS中的文件下载到本地，详请参见**从OBS下载数据**。示例代码如下：
     ```
     from modelarts.session import Session
     session=Session(access_key='***',secret_key='***',project_id='***',region_name='***')
     session.download_data(bucket_path="/bucket_name/obs_file.txt",path="/home/user/
     obs_file.txt")
     ```

带 OBS 存储的 Notebook 实例

对于使用OBS存储的Notebook实例，您可以使用OBS或ModelArts SDK将OBS中的文件下载到本地。

- 方式一：使用OBS进行下载

  使用OBS下载文件的操作指导：**下载文件**，可以将OBS中的文件下载到本地。如果您的数据较多，推荐OBS Browser+下载数据或文件夹。
- 方式二：使用ModelArts SDK进行下载
  a. 在您的本地环境**下载并安装ModelArts SDK**。
  b. 完成ModelArts SDK的**Session鉴权**。
  c. 将OBS中的文件下载到本地，详请参见**从OBS下载数据**。示例代码如下：
  ```
  from modelarts.session import Session
  session=Session(access_key='***',secret_key='***',project_id='***',region_name='***')
  session.download_data(bucket_path="/bucket_name/obs_file.txt",path="/home/user/
  obs_file.txt")
  ```

6.4.3 使用 ModelArts 示例

ModelArts提供了Notebook示例，方便初学者通过示例，快速了解如何使用ModelArts Notebook。

预览 ModelArts Examples

在JupyterLab中，单击左侧第二个按钮，按钮上会提示“ModelArts Examples”。可选择任意Example双击打开，打开的Example会在JupyterLab右侧显示，打开后Example为只读模式，可以进行浏览。

图 6-40 打开 Examples

使用 ModelArts Examples

在Examples列表中，选择任意一个Example单击打开，打开的Example会在JupyterLab右侧显示，打开后Example为只读模式，可以进行浏览。如果需要在本地编辑或运行，请单击界面右上方“Create a copy”，输入保存路径，则会拷贝当前Example以及依赖文件到保存目录中，并打开一个新页签用于直接编辑以及运行。

6.4.4 使用 ModelArts SDK

在Notebook中，通过使用ModelArts SDK，可以完成OBS管理、训练作业管理、模型管理以及在线服务管理。

ModelArts SDK使用请参见《**ModelArts SDK参考**》。

在Notebook中，已承载了登录用户的鉴权信息（AK/SK）和区域信息，因此SDK session鉴权时，无需输入参数即可完成session鉴权。

示例代码

- 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='/obs-bucket-name/src/',                           # 训练脚本目录
                    boot_file='/obs-bucket-name/src/pytorch_sentiment.py',      # 训练启动脚本目录
                    log_url='/obs-bucket-name/log/',                            # 训练日志目录
                    hyperparameters=[
                                {"label":"classes",
                                 "value": "10"},
                                {"label":"lr",
                                 "value": "0.001"}
                                ],
                    output_path='/obs-bucket-name/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_instance = estimator.fit(inputs='/obs-bucket-name/data/train/', wait=False,
job_name='my_training_job')
```

- 查询模型列表

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

- 查询服务详情

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

6.4.5 与 OBS 同步文件

如您在创建Notebook实例时，选择了OBS的“存储位置”，您写的代码会自动存储到您选择的OBS目录下。如您需要不同ipynb文件进行代码的相互调用，则可以使用Sync OBS功能。

Sync OBS功能是将在Notebook实例文件list列表选中的对象从OBS桶路径下同步到当前容器目录“~/work”下。

注意事项

- 单次同步文件大小最大为500MB，个数最多是1024个。
- 同步的对象总大小不超过5GB。例如，当前容器目录“~/work”下已有2GB的文件时，用户使用Sync OBS最多只能同步3GB的文件。

- Sync OBS功能只在带有OBS存储的实例上存在，因为非OBS存储的Notebook实例，其所有的文件读写操作都在用户容器里，即在“~/work”容器目录。

操作步骤

Sync OBS功能支持在JupyterLab页面使用，其对应的操作指导如下所示。

例如，“Example-1.ipynb”需要调用“Example-2.ipynb”中的“module”。在左侧File Browser中选择两个“ipynb”文件，然后单击“Sync OBS”按钮，在弹出框中单击“YES”，即可完成同步，同步成功后，即可在代码间相互调用。

图 6-41 使用 Sync OBS 功能

6.4.6 使用 Git 插件

针对带Github代码库的Notebook实例，可以在JupyterLab页面中使用Git插件快速查看内容，并提交修改后的代码。

前提条件

已创建带Git存储库的Notebook，且Notebook处于运行中状态。

打开 JupyterLab 的 git 插件

1. 在Notebook列表中，选择一个带Git存储库的实例，单击名称打开Notebook实例。
2. 进入“Jupyter”页面后，单击右上角的“Open JupyterLab”进入“JupyterLab”页面。

图6-42所示图标，为JupyterLab的git插件。

图 6-42 git 插件

查看代码库信息

在Name下方列表中，选中您希望使用的文件夹，双击打开，然后单击左侧git插件图标进入此文件夹对应的代码库。

图 6-43 打开文件夹后打开 git 插件

即可看到当前代码库的信息，如仓库名称、分支、历史提交记录等。

图 6-44 查看代码库信息

查看修改的内容

如果修改代码库中的某个文件，在“Changes”页签的“Changed”下可以看到修改的文件，并点击修改文件名称右侧的“Diff this file”，可以看到修改的内容。

图 6-45 查看修改的内容

提交修改的内容

确认修改无误后，单击修改文件名称右侧的“Stage this change”，文件将进入“Staged”状态，相当于执行了**git add**命令。在左下方输入本次提交的Message，单击“Commit”，相当于执行了**git commit**命令。

图 6-46 提交修改内容

此时，可以在“History”页签下看到本地提交已成功。

图 6-47 查看是否提交成功

单击“push”按钮，相当于执行**git push**命令，即可提交代码到GitHub仓库中。提交成功后会提示“Git Push completed successfully”。若OAuth鉴权的token过期，则此时再push会弹框让输入用户的token或者帐户信息，按照提示输入即可。

图 6-48 提交代码至 GitHub 仓库

完成上述操作后，可以在JupyterLab的git插件页面的History页签，看到“origin/HEAD”和“orgin/master”已指向最新一次的提交。同时在GitHub对应仓库的commit记录中也可以查找到对应的信息。

6.4.7 设置初始化脚本

在使用Notebook进行开发时，用户经常会安装一些软件，但是notebook实例停止后再启动就需要重新安装，影响用户的使用效率；在开发环境中的jupyterlab功能里面增加了设置初始化启动脚本的功能，可以将安装包的操作固化到脚本里面，并将其设置为启动初始化脚本，每次Notebook实例启动会默认执行此脚本，提升用户开发效率。

前提条件

已在JupyterLab中完成脚本编码。

使用限制

必须要在root用户下安装包的操作不能被执行，例如apt-get安装包。

设置初始化脚本

1. 在JupyterLab中，选中一个已经编辑好的脚本，单击右键并选择“Set as Initialization Script”，将其设置为实例启动时的初始化脚本。

图 6-49 设置初始化脚本

2. 当界面弹框提示“Config Succeed”时表示已设置成功。

图 6-50 配置成功

查看和更新初始化脚本

1. 在上方菜单栏中选择“Settings>Advanced Setting Editor”。
   或者您可以使用默认快捷键“Ctrl+,”查看。

图 6-51 进入 JupyterLab 的高级设置界面

右侧视图中可以查看初始化脚本。

图 6-52 查看设置的初始化脚本

2. 可编辑“User Preferences”中的“scriptFile”内容，单击右上角的保存按钮，可更新初始化脚本。

图 6-53 更新初始化脚本的内容

User Preferences

```
1  {
2      // ModelArts
3      // @cloud/modelarts-extension:plugin
4      // ModelArts
5      // ********************************
6
7      // Init Script
8      // Init script config
9      "initScript": {
10         "scriptFile": "/home/ma-user/work/install_package.sh"
11     }
12 }
```

执行初始化脚本

设置初始化脚本后，Notebook实例启动时默认会执行设置的脚本。

- 如果执行成功，会在右下角给出提示。
- 如果执行失败，则会弹框提示，可单击“View logs”查看日志。

图 6-54 执行失败

7 训练管理（New）

7.1 模型训练简介

当前ModelArts同时存在新版训练和旧版训练。新版训练在旧版训练的基础上进行了功能增强、调度优化、API整改等设计，推荐用户使用新版训练进行模型训练。新旧训练的差异请参考常见问题。

说明

当前章节介绍了新版训练流程。旧版训练请参考训练管理（旧版即将下线）。

ModelArts提供了模型训练的功能，方便您查看训练情况并不断调整您的模型参数。您还可以基于不同的数据，选择不同规格的资源池用于模型训练。除支持用户自己开发的模型外，ModelArts还提供了从AI Gallery订阅算法，您可以不关注模型开发，直接使用AI Gallery的算法，通过算法参数的调整，得到一个满意的模型。

请参考以下指导在ModelArts上训练模型：

- 使用ModelArts训练模型全流程，请参考**AI工程师如何使用ModelArts**。
- 您可以将训练数据导入至数据管理模块进行数据标注或者数据预处理，也支持将已标注的数据上传至OBS服务使用，请参考**数据准备**。
- 训练模型的算法实现与指导请参考**选择算法**章节。
- 使用控制台创建训练作业请参考**创建训练作业**章节。使用订阅算法创建训练作业示例请参考**使用AI Gallery订阅的算法构建模型**。使用自定义算法构建模型示例请参考**使用自定义算法在ModelArts上构建模型**。
- 关于训练作业日志、训练资源占用等详情请参考**查看作业详情**。
- 停止或删除模型训练作业，请参考**停止、重建或查找作业**。
- 模型超参自动调优指南，请参考**超参搜索训练**。

7.2 数据准备

ModelArts使用对象存储服务（Object Storage Service，简称OBS）进行数据存储以及模型的备份和快照，实现安全、高可靠和低成本的存储需求。

- **OBS简介**

- 使用训练数据的两种方式

OBS 简介

对象存储服务OBS是一个基于对象的海量存储服务，为客户提供海量、安全、高可靠、低成本的数据存储能力。对象存储服务OBS的基本组成是桶和对象。桶是OBS中存储对象的容器，每个桶都有自己的存储类别、访问权限、所属区域等属性，用户在互联网上通过桶的访问域名来定位桶。对象是OBS中数据存储的基本单位。关于OBS更详细的介绍请参考《OBS用户指南》。

对ModelArts来说，obs服务是一个数据存储中心。AI 开发过程中的输入数据、输出数据、中间缓存数据都可以在obs桶中进行存储、读取。

因此，在使用ModelArts之前您需要创建一个OBS桶，然后在OBS桶中创建文件夹用于存放数据。

图 7-1 对象存储服务 OBS

使用训练数据的两种方式

ModelArts模型训练支持2种读取训练数据的方式：

- 使用OBS桶中存储的数据集

  如果您的数据集已完成数据标注和数据预处理，可以将数据上传至OBS桶。当创建训练作业时，在训练输入参数位置填写训练数据所在的OBS桶路径即可完成训练配置。

- 使用数据管理中的数据集

  如果您的数据集未标注或者需要进一步的数据预处理，可以将数据导入ModelArts数据管理模块进行数据预处理。创建数据集详细指导参考创建数据集。

图 7-2 准备数据

7.3 选择算法

7.3.1 选择算法简介

机器学习从有限的观测数据中学习一般性的规律，并利用这些规律对未知的数据进行预测。为了获取更准确的预测结果，用户需要选择一个合适的算法来训练模型。针对不同的场景，ModelArts提供大量的的算法样例。以下章节提供了关于业务场景、算法学习方式、算法实现方式的指导。

选择算法的实现方式

ModelArts提供如下方式实现模型训练。

- 使用订阅算法

  ModelArts的AI Gallery，发布了较多官方算法，同时管理了其他开发者分享的算法，不需要进行代码开发，即可使用现成的算法进行模型构建。订阅操作请参考使用订阅算法。

- 在支持的预置框架中使用自定义脚本

  如果您需要使用自己开发的算法，可以选择使用ModelArts预置训练引擎。在ModelArts中，这种方式称为“使用自定义脚本”。ModelArts支持了大多数主流的AI引擎，详细请参见预置训练引擎。这些预置引擎预加载了一些额外的python包，例如numpy等；也支持您通过在代码目录中使用“reqiurements.txt”文件安装依赖包。使用自定义脚本创建训练作业请参考使用自定义脚本指导。

- 使用自定义镜像（新版训练请参考使用自定义镜像训练模型）

  订阅算法和预置框架涵盖了大部分的训练场景。针对特殊场景，ModelArts支持用户构建自定义镜像用于模型训练。自定义镜像需上传至容器镜像服务（SWR），才能用于ModelArts上训练。由于自定义镜像的制作要求用户对容器相关知识有比较深刻的了解，除非订阅算法和预置引擎无法满足需求，否则不推荐使用。

选择算法的学习方式

ModelArts支持用户根据实际需求进行不同方式的模型训练。

- 离线学习

  离线学习是训练中最基本的方式。离线学习需要一次性提供训练所需的所有数据，在训练完成后，目标函数的优化就停止了。使用离线学习的优势是模型稳定性高，便于做模型的验证与评估。缺点是时间和空间成本效率低。

- 增量学习

  增量学习是一个连续不断的学习过程。相较于离线学习，增量学习不需要一次性存储所有的训练数据，缓解了存储资源有限的问题；另一方面，增量学习节约了重新训练中需要消耗大量算力、时间以及经济成本。

7.3.2 使用订阅算法

ModelArts的AI Gallery，发布了较多官方算法，可以帮助AI开发者快速开始训练和部署模型。对于不熟悉ModelArts的用户，可以快速订阅官方推荐算法实现模型训练全流程。

AI Gallery不仅可以订阅官方发布算法，也支持用户发布自定义算法和订阅其他开发者分享的算法。为了使用他人或者ModelArts官方分享的算法，您需要将AI Gallery的算法订阅至您的ModelArts中。

- 查找算法
- 订阅算法

查找算法

为了获得匹配您业务的算法，您可以通过多个入口区查找算法。

- 在ModelArts控制台，“算法管理>我的订阅”中，单击“订阅更多算法”，可跳转至“AI Gallery”页面，查找相应的算法。
- 在ModelArts控制台，直接在左侧菜单栏中选择“AI Gallery”，进入“AI Gallery”页面，在“资产集市 > 算法”页面查找相应的算法。

  在AI Gallery的算法模块，根据算法的业务场景，官方发布的算法如表7-1所示。

表 7-1 官方发布算法

应用场景	算法	AI引擎	支持服务类型	训练使用的资源类型	部署使用的资源类型
图像分类	ResNet_v1_50	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	ResNet_v1_101	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	ResNet_v1_50	TensorFlow	在线服务	Ascend 910	CPU、GPU、Ascend 310
	ResNet50	MindSpore	在线服务	Ascend 910	Ascend 310
	ResNet_v2_50	TensorFlow	在线服务	Ascend 910	Ascend 310

应用场景	算法	AI引擎	支持服务类型	训练使用的资源类型	部署使用的资源类型
	ResNet_v2_101	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	MobileNet_v1	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	MobileNet_v2	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	Inception_v3	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	ResNet50_AdaInfer	TensorFlow	在线服务	GPU	CPU、GPU
	Res2Net_50	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	Inception_ResNet_v2	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	EfficientNetB0/B4/B7/B8	PyTorch	在线服务	GPU	CPU、GPU
物体检测	RetinaNet_ResNet50	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	YOLOv3_ResNet18	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	YOLOv3_ResNet18	TensorFlow	在线服务	Ascend 910	Ascend 310
	YOLOv3_Darknet53	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	SSD_VGG	Caffe	在线服务	GPU	暂不支持
	EfficientDet	Tensorflow	在线服务	GPU	CPU、GPU
	YOLOv5	Pytorch	在线服务	GPU	CPU、GPU
文本分类	BERT	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	NEZHA	TensorFlow	在线服务	GPU	GPU
图像语义分割	DeepLabV3	TensorFlow	在线服务	GPU	Ascend 310

应用场景	算法	AI引擎	支持服务类型	训练使用的资源类型	部署使用的资源类型
	DeepLabV3	TensorFlow	在线服务	Ascend 910	Ascend 310
声音分类	声音分类	TensorFlow	在线服务	GPU	CPU、GPU
时序预测	时序预测	PyTorch	在线服务	GPU	暂不支持
强化学习	强化学习GameAI	TensorFlow	在线服务	GPU	GPU

订阅算法

1. 进入“AI Gallery”，选择“资产集市 > 算法”页签，查找您需要的算法并单击算法名称，进入算法详情页，单击算法详情页右侧的“订阅”，订阅您所需的算法。

   订阅后的算法，状态变为“已订阅”，并且将自动展现在“算法管理 > 我的订阅”页面中。

2. 单击“前往控制台”，选择云服务区域，进入“算法管理 > 我的订阅”页面进入此页面内，单击“产品名称”左侧的小三角，展开算法详情，在“版本列表”区域，单击“创建训练作业”即可进行后续操作。

   图 7-3 订阅算法

7.3.3 使用自定义脚本

7.3.3.1 使用自定义脚本简介

如果订阅算法不能满足需求或者用户希望迁移本地算法至ModelArts上训练，可以考虑使用ModelArts支持的预置训练引擎实现算法构建。这种方式在创建算法时被称为“使用自定义脚本”模式。

以下章节介绍了如何使用自定义脚本创建算法。

- 如果需要了解ModelArts模型训练支持的预置引擎和模型，请参考预置的训练引擎。
- 本地开发的算法迁移至ModelArts需要做代码适配，如何适配请参考开发自定义脚本章节。
- 通过ModelArts控制台界面使用自定义脚本创建算法可以参考创建算法章节。
- 完成算法开发后，您可以将个人开发算法免费分享给他人使用，请参考发布免费算法章节。

预置的训练引擎

当前ModelArts支持的训练引擎及对应版本如下所示。

表 7-2 新版训练作业支持的 AI 引擎

工作环境	适配芯片	系统架构	系统版本	AI引擎与版本	支持的cuda或Ascend版本
TensorFlow	CPU/GPU	x86_64	Ubuntu18.04	tensorflow_2.1.0-cuda_10.1-py_3.7-ubuntu_18.04-x86_64	cuda10.1
PyTorch	CPU/GPU	x86_64	Ubuntu18.04	pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-x86_64	cuda10.2
Ascend-Powered-Engine	Ascend910	aarch64	Euler2.8	mindspore_1.3.0-cann_5.0.2-py_3.7-euler_2.8.3-aarch64	cann_5.0.2
				mindspore_1.5.1-cann_5.0.3-py_3.7-euler_2.8.3-aarch64	cann_5.0.3
				tensorflow_1.15-cann_5.0.2-py_3.7-euler_2.8.3-aarch64	cann_5.0.2

工作环境	适配芯片	系统架构	系统版本	AI引擎与版本	支持的cuda或Ascend版本
				tensorflow_1.15-cann_5.0.3-py_3.7-euler_2.8.3-aarch64	cann_5.0.3
MPI	CPU/GPU	x86_64	Ubuntu18.04	mindspore_1.3.0-cuda_10.1-py_3.7-ubuntu_1804-x86_64	cuda_10.1
Horovod	GPU	x86_64	ubuntu_18.04	horovod_0.20.0-tensorflow_2.1.0-cuda_10.1-py_3.7-ubuntu_18.04-x86_64	cuda_10.1
				horovod_0.22.1-pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-x86_64	cuda_10.2

说明

- MoXing是ModelArts团队自研的分布式训练加速框架，它构建于开源的深度学习引擎TensorFlow、MXNet、PyTorch、Keras之上，详细说明请参见**MoXing使用说明**。如果您使用的是MoXing框架编写训练脚本，在创建训练作业时，请根据您选用的接口选择其对应的AI引擎和版本。
- “efficient_ai”是华为云ModelArts团队自研的加速压缩工具，它支持对训练作业进行量化、剪枝和蒸馏来加速模型推理速度，详细说明请参见**efficient_ai使用说明**。
- **Ascend-Powered-Engine**仅在“华北-北京四”区域支持。

新旧版训练预置引擎差异

- 新版的预置训练引擎默认安装Moxing2.0.0及以上版本。
- 新版的预置训练引擎统一使用了Python3.7及以上版本。
- 新版镜像修改了默认的HOME目录，由“/home/work”变为“/home/ma-user”，请注意识别训练代码中是否有“/home/work”的硬编码。
- 提供预置引擎类型有差异。新版的预置引擎在常用的训练引擎上进行了升级。

  如果您需要使用旧版训练引擎，单击显示旧版引擎即可选择旧版引擎。新旧版支持的预置引擎差异请参考**表7-3**。详细的训练引擎版本说明请参考**新版训练和旧版训练分别支持的AI引擎**。

表 7-3 新旧版预置引擎差异

工作环境	预置训练引擎与版本	旧版训练	新版训练
TensorFlow	Tensorflow-1.8.0	√	x
	Tensorflow-1.13.1	√	后续版本支持
	Tensorflow-2.1.0	√	√
MXNet	MXNet-1.2.1	√	x
Caffe	Caffe-1.0.0	√	x
Spark_MLlib	Spark-2.3.2	√	x
Ray	RAY-0.7.4	√	x
XGBoost-Sklearn	XGBoost-0.80-Sklearn-0.18.1	√	x
PyTorch	PyTorch-1.0.0	√	x
	PyTorch-1.3.0	√	x
	PyTorch-1.4.0	√	x
	PyTorch-1.8.0	x	√
Ascend-Powered-Engine	Mindspore-1.1.1	√	x
	Mindspore-1.3.0	x	√
	Tensorflow-1.15	√	√
MPI	MindSpore-1.3.0	x	√
Horovod	horovod_0.20.0-tensorflow_2.1.0	x	√
	horovod_0.22.1-pytorch_1.8.0	x	√
MindSpore-GPU	MindSpore-1.1.0	√	x
	MindSpore-1.2.0	√	x

7.3.3.2 开发自定义脚本

当您使用自定义脚本创建算法时，您需要提前完成算法的代码开发。本章详细介绍如何改造本地代码以适配ModelArts上的训练。

创建算法时，您需要在创建页面提供代码目录路径、代码目录路径中的启动文件、输入路径参数和训练输出路径参数。这四种输入搭建了用户代码和ModelArts后台交互的桥梁。

- 代码目录路径

  您需要在OBS桶中指定代码目录，并将训练代码、依赖安装包或者预生成模型等训练所需文件上传至该代码目录下。训练作业创建完成后，ModelArts会将代码目录及其子目录下载至后台容器中。

  请注意不要将训练数据放在代码目录路径下。训练数据比较大，训练代码目录在训练作业启动后会下载至后台，可能会有下载失败的风险。建议训练代码目录大小不超过50MB。

- 代码目录路径中的启动文件

  代码目录路径中的启动文件作为训练启动的入口，当前只支持python格式。

- 输入路径参数

  训练数据需上传至OBS桶或者存储至**数据集**中。在训练代码中，用户需解析**输入路径参数**。系统后台会自动下载输入参数路径中的训练数据至训练容器的本地目录。请保证您设置的桶路径有读取权限。在训练作业启动后，ModelArts会挂载硬盘至“/cache”目录，用户可以使用此目录来存储临时文件。“/cache”目录大小请参考**训练环境中不同规格资源“/cache”目录的大小**。

- 训练输出路径参数

  建议设置一个空目录为训练输出路径。在训练代码中，您需要解析**输出路径参数**。系统后台会自动上传训练输出至指定的训练输出路径，请保证您设置的桶路径有写入权限和读取权限。

在ModelArts中，训练代码需包含以下步骤：

（可选）引入依赖

1. 当您使用自定义脚本创建算法的时候，如果您的模型引用了其他依赖，您需要在创建算法的“代码目录”下放置相应的文件或安装包。
   - 安装python依赖包请参考**模型中引用依赖包时，如何创建训练作业?**
   - 安装C++的依赖库请参考**如何安装C++的依赖库?**
   - 在预训练模型中加载参数请参考**如何在训练中加载部分训练好的参数?**

   图 7-4 选择代码目录并指定模型启动文件

解析输入路径参数、输出路径参数

运行在ModelArts的模型读取存储在OBS服务的数据，或者输出至OBS服务指定路径，输入和输出数据需要配置3个地方：

1. 训练代码中需解析输入路径参数和输出路径参数。ModelArts推荐以下方式实现参数解析。

```
import argparse
# 创建解析
parser = argparse.ArgumentParser(description="train mnist",
                                 formatter_class=argparse.ArgumentDefaultsHelpFormatter)
# 添加参数
parser.add_argument('--train_url', type=str,
```

```
                    help='the path model saved')
parser.add_argument('--data_url', type=str, help='the training data')
# 解析参数
args, unkown = parser.parse_known_args()
```

完成参数解析后，用户使用“data_url”、“train_url”代替算法中数据来源和数据输出所需的路径。

2. 在使用自定义脚本创建自定义算法时，根据步骤1中的代码参数设置，您需要在创建算法页面配置已定义的输入输出参数。
   - 训练数据是算法开发中必不可少的输入。输入数据默认配置为“数据来源”，代码参数为“data_url”，也支持用户根据步骤1的算法代码自定义代码参数。

     图 7-5 解析输入路径参数“data_url”

   - 模型训练结束后，训练模型以及相关输出信息需保存在OBS路径。输出数据默认配置为“模型输出”，代码参数为“train_url”，也支持用户根据步骤1的算法代码自定义输出路径参数。

     图 7-6 解析输出路径参数“train_url”

3. 在创建训练作业时，填写输入路径和输出路径。

   训练输入选择对应的OBS路径或者数据集路径；训练输出选择对应的OBS路径。

   图 7-7 训练输入和输出设置

训练代码正文和保存模型

训练代码正文和保存模型涉及的代码与您使用的AI引擎密切相关。以下案例以Tensorflow框架为例，训练代码中解析参数方式采用tensorflow接口tf.flags.FLAGS接受命令行参数：

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import os
```

```
import tensorflow as tf
from tensorflow.examples.tutorials.mnist import input_data

import moxing as mox

tf.flags.DEFINE_integer('max_steps', 1000, 'number of training iterations.')
tf.flags.DEFINE_string('data_url', '/home/jnn/nfs/mnist', 'dataset directory.')
tf.flags.DEFINE_string('train_url', '/home/jnn/temp/delete', 'saved model directory.')

FLAGS = tf.flags.FLAGS


def main(*args):
    # Train model
    print('Training model...')
    mnist = input_data.read_data_sets(FLAGS.data_url, one_hot=True)
    sess = tf.InteractiveSession()
    serialized_tf_example = tf.placeholder(tf.string, name='tf_example')
    feature_configs = {'x': tf.FixedLenFeature(shape=[784], dtype=tf.float32),}
    tf_example = tf.parse_example(serialized_tf_example, feature_configs)
    x = tf.identity(tf_example['x'], name='x')
    y_ = tf.placeholder('float', shape=[None, 10])
    w = tf.Variable(tf.zeros([784, 10]))
    b = tf.Variable(tf.zeros([10]))
    sess.run(tf.global_variables_initializer())
    y = tf.nn.softmax(tf.matmul(x, w) + b, name='y')
    cross_entropy = -tf.reduce_sum(y_ * tf.log(y))

    tf.summary.scalar('cross_entropy', cross_entropy)

    train_step = tf.train.GradientDescentOptimizer(0.01).minimize(cross_entropy)

    correct_prediction = tf.equal(tf.argmax(y, 1), tf.argmax(y_, 1))
    accuracy = tf.reduce_mean(tf.cast(correct_prediction, 'float'))
    tf.summary.scalar('accuracy', accuracy)
    merged = tf.summary.merge_all()
    test_writer = tf.summary.FileWriter(FLAGS.train_url, flush_secs=1)

    for step in range(FLAGS.max_steps):
        batch = mnist.train.next_batch(50)
        train_step.run(feed_dict={x: batch[0], y_: batch[1]})
        if step % 10 == 0:
            summary, acc = sess.run([merged, accuracy], feed_dict={x: mnist.test.images, y_: mnist.test.labels})
            test_writer.add_summary(summary, step)
            print('training accuracy is:', acc)
    print('Done training!')

    builder = tf.saved_model.builder.SavedModelBuilder(os.path.join(FLAGS.train_url, 'model'))

    tensor_info_x = tf.saved_model.utils.build_tensor_info(x)
    tensor_info_y = tf.saved_model.utils.build_tensor_info(y)

    prediction_signature = (
        tf.saved_model.signature_def_utils.build_signature_def(
            inputs={'images': tensor_info_x},
            outputs={'scores': tensor_info_y},
            method_name=tf.saved_model.signature_constants.PREDICT_METHOD_NAME))

    builder.add_meta_graph_and_variables(
        sess, [tf.saved_model.tag_constants.SERVING],
        signature_def_map={
            'predict_images':
                prediction_signature,
        },
        main_op=tf.tables_initializer(),
        strip_default_attrs=True)

    builder.save()
```

```
    print('Done exporting!')


if __name__ == '__main__':
    tf.app.run(main=main)
```

新旧版训练代码适配的差异

旧版训练中，用户需要在输入输出数据上做如下配置：

```
#解析命令行参数
import argparse
parser = argparse.ArgumentParser(description='MindSpore Lenet Example')
parser.add_argument('--data_url', type=str, default="./Data",
                    help='path where the dataset is saved')
parser.add_argument('--train_url', type=str, default="./Model", help='if is test, must provide\
                    path where the trained ckpt file')
args = parser.parse_args()
...
#下载数据参数至容器本地，在代码中使用local_data_path代表训练输入位置
mox.file.copy_parallel(args.data_url, local_data_path)
...
#上传容器本地数据至obs路径
mox.file.copy_parallel(local_output_path, args.train_url)
```

7.3.3.3 创建算法

针对您在本地或使用其他工具开发的算法，支持上传至ModelArts中统一管理。在创建自定义算法过程中，您需要关注以下内容：

1. 前提条件
2. 进入创建算法页面
3. 设置算法基本信息
4. 设置创建方式
5. 配置输入输出数据
6. 定义超参
7. 支持的策略
8. 添加训练约束
9. 后续操作

前提条件

- 数据已完成准备：已在ModelArts中创建可用的数据集，或者您已将用于训练的数据集上传至OBS目录。
- 请准备好自定义脚本，并上传至OBS目录。训练脚本开发指导参见开发自定义脚本。
- 已在OBS创建至少1个空的文件夹，用于存储训练输出的内容。
- 由于训练作业运行需消耗资源，确保帐户未欠费。
- 确保您使用的OBS目录与ModelArts在同一区域。

进入创建算法页面

1. 登录ModelArts管理控制台，单击左侧菜单栏的“算法管理”。

2. 在“我的算法”管理页面，单击“创建”，进入“创建算法”页面。

设置算法基本信息

基本信息包含“名称”和“描述”。

图 7-8 设置算法基本信息

设置创建方式

选择“预置镜像”创建算法。

用户需根据实际算法代码情况设置“镜像”、“代码目录”和“启动文件”。选择的AI引擎和编写算法代码时选择的框架必须一致。例如编写算法代码使用的是TensorFlow，则在创建算法时也要选择TensorFlow。

表 7-4 创建方式参数说明

参数	说明
“预置镜像”	默认显示新版训练支持的AI引擎，勾选“显示旧版引擎”，可以选择旧版训练支持的引擎，具体请参考使用自定义脚本简介中的AI引擎介绍。
“代码目录”	算法代码存储的OBS路径。训练代码、依赖安装包或者预生成模型等训练所需文件上载至该代码目录下。 代码目录下不能存在他人上传的文件和目录，也不能存在无关的文件和目录，否则可能导致失败。 请注意不要将训练数据放在代码目录路径下。训练数据比较大，训练代码目录在训练作业启动后会下载至后台，可能会有下载失败的风险。 训练作业创建完成后，ModelArts会将代码目录及其子目录下载至后台容器中。 **说明** ● 编程语言不限。 ● 文件数（含文件、文件夹数量）不超过1024个。 ● 文件总大小不超过5GB。
“启动文件”	必须为“代码目录”下的文件，且以“.py”或“.pyc”结尾，即ModelArts目前只支持使用Python语言编写的启动文件。 代码目录路径中的启动文件为训练启动的入口。

图 7-9 使用自定义脚本创建算法

配置输入输出数据

训练过程中，自定义算法需要从OBS桶或者数据集中获取数据进行模型训练，训练产生的输出结果也需要存储至OBS桶中。用户的算法代码中需解析输入输出参数实现ModelArts后台与OBS的数据交互，用户可以参考开发自定义脚本完成适配ModelArts训练的代码开发。

创建自定义算法时，用户需要将算法代码中定义的输入输出参数进行配置。

- 输入数据配置

表 7-5 输入数据配置

参数	参数说明
映射名称	输入参数的说明，用户可以自定义描述。默认为“数据来源”。
代码路径参数	如果您的算法代码中使用argparse解析data_url为输入数据参数，则在创建的算法需要配置输入数据代码参数为data_url。根据实际代码中的输入数据参数定义此处的名称。 此处设置的代码路径参数必须与算法代码中解析的输入数据参数保持一致，否则您的算法代码无法获取正确的输入数据。
是否添加约束	用户可以根据实际情况限制数据来源的方式，可以设置为支持数据存储位置或者ModelArts数据集。 如果用户选择数据来源为ModelArts数据集，还可以约束以下三种： ● 标注类型。数据类型请参考标注数据。 ● 数据格式。可选“Default”和“CarbonData”，支持多选。其中“Default”代表Manifest格式。 ● 数据切分。仅“图像分类”、“物体检测”、“文本分类”和“声音分类”类型数据集支持进行数据切分功能。 可选“仅支持切分的数据集”、“仅支持未切分数据集”和“无限制”。数据切分详细内容可参考发布数据版本。
添加输入数据配置	用户可以根据实际算法确定多个输入数据来源。

图 7-10 输入数据配置

- 输出数据配置

表 7-6 输出数据配置

参数	参数说明
映射名称	输出参数的说明，用户可以自定义描述。默认为“输出数据”。
代码路径参数	如果您的算法代码中使用argparse解析train_url为训练输出参数，则在创建的算法需要配置输出数据参数为train_url。根据实际代码中的训练输出参数定义此处的名称。 此处设置的代码路径参数必须与算法代码中解析的训练输出参数保持一致，否则您的算法代码无法获取正确的输出路径。
添加输出数据配置	用户可以根据实际算法确定多个输出数据路径。

图 7-11 输出数据配置

定义超参

使用常用框架创建算法时，ModelArts支持用户自定义超参，方便用户查阅或修改。定义超参后会体现在启动命令中，以命令行参数的形式传入您的启动文件中。

1. 导入超参

   您可以单击“增加超参”手动添加超参。请注意超参输入的内容应该只包含大小写字母、中文、数字、空格、中划线、下划线、逗号和句号。

图 7-12 添加超参

2. 编辑超参。超参参数说明参见表7-7。

表 7-7 超参编辑参数

参数	说明
名称	填入超参名称。 超参名称支持64个以内字符，仅支持大小写字母、数字、下划线和中划线。
描述	填入超参的描述说明。 超参描述支持大小写字母、中文、数字、空格、中划线、下划线、逗号和句号。
类型	填入超参的数据类型。支持String、Integer、Float和Boolean。
默认值	填入超参的默认值。创建训练作业时，默认使用该值进行训练。
约束	单击约束。在弹出对话框中，支持用户设置默认值的取值范围或者枚举值范围。
必需	可选是或者否。如果您选择否，在使用该算法创建训练作业时，支持在创建训练作业页面删除该超参。如果您选择是，则不支持删除操作。

支持的策略

ModelArts支持用户使用自动化搜索功能。自动化搜索功能在零代码修改的前提下，自动找到最优的超参，有助于提高模型精度和收敛速度。详细的参数配置请参考超参搜索配置。

自动搜索目前仅支持pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-x86_64, tensorflow_2.1.0-cuda_10.1-py_3.7-ubuntu_18.04-x86_64引擎。

添加训练约束

用户可以根据实际情况定义此算法的训练约束。

- 资源类型：可选“CPU”、“GPU”和“Ascend”，支持多选。
- 多卡训练：可选“支持”和“不支持”。
- 分布式训练：可选“支持”和“不支持”。

图 7-13 算法训练约束

运行环境预览

创建训练作业时，可以打开创建页面右下方的运行环境预览窗口

，辅助您了解代码目录、启动文件、输入输出等数据配置在训练容器中的路径。

图 7-14 运行环境预览

后续操作

创建算法完成后，在“算法管理”页面，等待算法就绪。当新创建的算法状态变更为“就绪”时，即可执行其他操作。

可以使用算法快速创建训练作业，构建模型，详细操作请参见创建训练作业。

7.3.4 使用自定义镜像

订阅算法和预置框架涵盖了大部分的训练场景。针对特殊场景，ModelArts支持用户构建自定义镜像用于模型训练。自定义镜像需上传至容器镜像服务（SWR），才能用于ModelArts上训练。

自定义镜像的制作要求用户对容器相关知识有比较深刻的了解，除非订阅算法和预置框架无法满足需求，否则不推荐使用。

新版训练支持的自定义镜像使用说明，请参考使用自定义镜像训练模型。

7.3.5 发布算法到 AI Gallery

用户通过预置框架或自定义镜像创建的算法，支持发布到AI Gallery，并分享给其他用户使用。本章节介绍如何发布算法到AI Gallery的操作。

发布操作

1. 在ModelArts控制台“ 算法管理>我的算法”页面，单击待发布的算法名称，进入算法详情页。
2. 在算法详情页的右上角单击“发布”，可以将自己的算法发布到AI Gallery，分享给其他用户。
3. 在发布资产版本页面，输入发布参数后，单击“发布”。
   - 如果是已有资产，本次发布是为了更新资产版本。
     i. 在“资产名称”下拉框中选择已有资产名称，填写“资产版本”。已发布资产版本可在“资产版本”输入框右侧单击“查看该资产版本”查看。
     ii. 在发布资产版本页面，填写“版本描述”，单击“发布”。
   - 如果是新创建的资产。
     i. 在“资产名称”右侧单击“创建新资产”，填写名称和资产描述信息，单击“确定”。

     说明

     只有第一次在AI Gallery发布资产（包括算法、Notebook、数据集等）的用户，此处会出现勾选“同意《AI Gallery入驻协议》”选项。

     ii. 在发布资产版本页面，填写“资产版本”和“版本描述”，单击“发布”。
4. 提交资产发布申请后，AI Gallery侧会自动托管上架，可以前往AI Gallery查看资产上架情况。

7.4 完成一次简单训练

7.4.1 创建训练作业

训练管理模块是ModelArts不可或缺的功能模块，用于创建训练作业、查看训练情况以及管理训练版本。模型训练是一个不断迭代和优化的过程。在训练模块的统一管理

下，方便用户试验算法、数据和超参数的各种组合，便于追踪最佳的模型与输入配置，您可以通过不同版本间的评估指标比较，确定最佳训练作业。

前提条件

- 数据已完成准备：已在ModelArts中创建可用的数据集，或者您已将用于训练的数据上传至OBS目录。
- “算法管理”中，已完成使用自定义脚本或者已使用订阅算法。
- 已在OBS创建至少1个空的文件夹，用于存储训练输出的内容。
- 由于训练作业运行需消耗资源，确保帐户未欠费。
- 确保您使用的OBS目录与ModelArts在同一区域。

创建训练作业

1. 登录ModelArts管理控制台，在左侧导航栏中选择“全局配置”，检查是否配置了访问授权。若未配置，请参考使用委托授权完成操作。

   图 7-15 配置授权

2. 登录ModelArts管理控制台，在左侧导航栏中选择“训练管理 > 训练作业（New）”，默认进入“训练作业”列表。
3. 单击“创建训练作业”，进入“创建训练作业”页面，在该页面填写训练作业相关参数。

图 7-16 进入创建训练作业页面

4. 填写基本信息。基本信息包含“名称”和“描述”。
5. 填写算法配置参数。请根据您的算法类型选择算法来源。

表 7-8 作业参数说明

参数名称	子参数	说明
算法	自定义算法	选择“自定义算法”页签，设置“镜像来源”、“代码目录”和“启动文件”，详细操作指导参见创建算法。
算法	我的算法	选择“我的算法”页签，勾选“算法名称”左侧按钮，即选中该算法。 如果没有创建算法，请单击“创建”进入创建算法页面，详细操作指导参见创建算法。
	我的订阅	选择“我的订阅”页签，单击下拉三角标选择算法版本。勾选算法版本，即选中该算法。 如果没有订阅算法，请单击“订阅更多算法”进入算法订阅页面，查找算法并订阅。
训练输入	-	请选择OBS中数据存储位置作为训练输入。 此处训练输入与您选择算法的“输入数据配置”匹配，参见表7-5。例如您的训练代码中使用argparse解析data_url为输入数据超参，则在创建的算法需要配置输入数据代码参数为data_url。在此处，您可以选择数据集或者OBS存储位置作为输入的路径。训练启动时，会下载输入路径中的数据至运行容器。

参数名称	子参数	说明
	数据集	从ModelArts数据管理中选择可用的数据集及其版本。 单击“选择数据集”，在弹出的对话框中，勾选目标数据集并选择对应的版本。 **说明** 如果该选项置灰，表示该算法的输入不支持选择数据集。
	数据存储位置	从OBS桶中选择训练数据。 单击“数据存储位置”，从弹出的对话框中，选择数据存储的OBS桶及其文件夹。 **说明** 如果该选项置灰，表示该算法的输入不支持选择数据存储位置。
训练输出	-	选择训练结果的存储位置（OBS路径）。为避免出现错误，建议选择一个空目录用作“训练输出”。 模型输出参数与您选择算法的“输出数据配置”匹配，参见表7-6。例如您的训练代码中使用argparse解析train_url为输出路径参数，则在创建的算法需要配置输入数据代码参数为train_url。在此处，您可以选择OBS路径作为存储位置。训练过程中时，会上传训练输出至指定OBS路径。
超参	-	此参数根据您选择的算法不同而不同。 如果您在创建算法时已经定义超参，此处会显示该算法所有超参。超参可修改、可删除状态取决于您在创建算法时的超参约束设置，详见定义超参。 如果您在创建算法时设置支持自定义超参，您可以单击“增加超参”，添加多条超参用于调优。
环境变量	-	开启此参数用于增加用户自行设置的环境变量。
故障自动重启	-	开启此参数后，训练作业失败时会自动重新下发并运行训练作业。此处设置训练重启次数。

说明

根据您选择的算法不同，训练输入、训练输出和调优参数显示不同。

当“训练输入”显示为“当前算法没有输入通道”，表明您不需要在该页面配置输入数据。

当“训练输出”显示为“当前算法没有输出通道”，表明您不需要在该页面配置输出路径。

当“超参”显示为“当前所选算法不支持自定义超参”，表明您不需要在该页面配置超参用于调优。

6. 选择训练资源的规格。训练参数的可选范围与已有算法的使用约束保持一致。

表 7-9 资源参数说明

参数名称	说明
资源池	选择训练作业资源池。训练作业支持选择“公共资源池”和“专属资源池”。
资源类型	可选CPU、GPU、Ascend。
规格	针对不同的资源类型，选择资源规格。如果您的算法已定义使用CPU或GPU，根据已有算法约束条件，您可以在有效规格选择合适的资源规格，无效选项置灰不可选。 不同的资源类型的数据盘容量是不同的，为避免训练过程中出现内存不足的情况，请参考**训练环境中不同规格资源“/cache”目录的大小**。
计算节点个数	选择计算节点的个数。默认值为“1”。
作业日志路径	● 如果用户选择CPU或者GPU规格，可以选择是否打开永久保存按钮。 打开永久保存按钮后，用户需要选择作业运行中产生的日志文件存储路径。 ● 如果用户选择Ascend规格，用户需要选择作业运行中产生的日志文件存储路径。 请选择一个空的OBS目录用于存储日志文件。 **说明** 请注意选择的OBS目录有读写权限。
自动停止	使用付费资源时会出现此参数。 ● 启用该参数并设置时间后，训练作业将在指定时间自动停止。 ● 如果不启用此参数，训练作业将一直运行，同时一直收费，自动停止功能可以帮您避免产生不必要的费用。 自动停止时间支持设置为“1小时后”、“2小时后”、“4小时后”、“6小时后”、“自定义”。

7. 单击“提交”，完成训练作业的创建。

   训练作业一般需要运行一段时间。要查看训练作业实时情况，您可以前往训练作业列表，查看训练作业的基本情况。在训练作业列表中，刚创建的训练作业“状态”为“初始化”，当训练作业的“状态”变为“运行成功”时，表示训练作业运行结束，其生成的模型将存储至对应的“训练输出”中。当训练作业的“状态”变为“运行失败”时，您可以单击训练作业的名称，进入详情页面，通过查看日志等手段处理问题，详情请参考**查看作业详情**。

   说明

   训练作业创建完成后，将立即启动，运行过程中将按照您选择的资源按需计费。

7.4.2 查看作业详情

训练作业运行时，支持查看以下训练配置、训练过程日志、资源占用和模型评估结果四个场景，判断此训练作业是否满意。

7.4.2.1 训练作业详情

1. 登录ModelArts管理控制台，在左侧导航栏中选择“训练管理 >训练作业（New）”，默认进入“训练作业”列表。
2. 在训练作业列表中，您可以单击作业名称，查看该作业的基本信息与训练作业参数。

图 7-17 查看训练作业详情

3. 在训练作业详情页的左侧，可以查看此次训练作业的配置、算法的配置的相关信息。
   a. 训练作业基本信息

表 7-10 训练作业基本信息

参数	说明
作业ID	训练作业唯一标识。
状态	训练作业状态。
创建时间	记录训练作业创建时间。
运行时间	记录训练作业运行时长。

   b. 训练作业参数

表 7-11 训练作业参数

参数	说明
算法名称	本次训练作业使用的算法。
AI引擎	本次训练作业使用的AI引擎。
代码目录	训练作业脚本所在的代码目录。如果您是市场订阅算法，该参数不显示内容。

参数	说明
启动文件	训练作业启动文件位置。如果您是市场订阅算法，该参数不显示内容。
计算节点个数	本次训练作业设置的计算节点个数。
规格	本次训练作业使用的训练规格。
作业日志路径	设置的训练作业日志输出路径。
训练输入-输入路径	本次训练中，输入数据的OBS路径。
训练输入-训练参数名称	算法代码中，输入路径指代的参数。
训练输入-本地路径（训练参数值）	训练启动后，ModelArts将OBS路径中的数据下载至后台容器，本地路径指ModelArts后台容器中存储输入数据的路径。
训练输出-输出路径	本次训练中，输出数据的OBS路径。
训练输出-训练参数名称	算法代码中，输出路径指代的参数。
训练输出-本地路径（训练参数值）	ModelArts后台容器中存储训练输出的路径。
超参	本次训练作业使用的超参。
环境变量	本次训练作业设置的环境变量。

7.4.2.2 训练作业事件

训练作业的（从用户可看见训练任务开始）整个生命周期中，每一个关键事件点在系统后台均有记录，用户可随时在对应训练作业的详情页面进行查看。

方便用户更清楚的了解训练作业运行过程，遇到任务异常时，更加准确的排查定位问题。可查看的事件点包括：

- 训练作业准备中，查看当前训练作业都已完成了哪些准备动作或都有哪些关键事件点。
- 训练作业运行中，查看都有哪些关键阶段信息。

训练运行到结束的过程中，关键事件支持手动/自动刷新。

查看操作

1. 在ModelArts管理控制台的左侧导航栏中选择“训练管理 >训练作业”，在训练作业列表中，您可以单击作业名称，进入训练作业详情页面。
2. 在训练作业详情页面的右上角，单击“查看事件”，查看事件信息。

7.4.2.3 训练日志详情

在训练作业详情页，训练日志窗口提供日志预览、日志下载、日志中搜索关键字能力和训练故障识别能力。

- 预览

  系统日志窗口提供训练日志预览功能，并支持查看不同计算节点日志，您可以通过右侧下拉框选择目标节点预览。

  当日志文件过大时，系统日志窗口仅加载最新的部分日志，并在日志窗口上方提供全量日志访问链接。打开该链接可在新页面查看全部日志。

  图 7-18 查看全量日志

  说明

  - 如果全部日志超过500M，可能会引起浏览页面卡顿，建议您直接下载日志查看。
  - 预览链接在生成后的一小时内，支持任何人打开并查看。您可以分享链接至他人。请注意日志中不能包含隐私内容，否则会造成信息泄露。

- 下载

  训练日志仅保留30天，超过30天会被清理。如果用户需要永久保存日志，请单击系统日志窗口右上角下载按钮下载日志至本地保存，支持批量下载多节点日志。用户也可以在创建训练作业时打开永久保存日志按钮，保存训练日志至指定OBS路径。

  针对使用Ascend规格创建的训练作业，部分系统日志暂不支持直接在训练日志窗口下载，请用户在创建训练作业时指定OBS路径用于保存训练日志。

  图 7-19 下载日志

- 搜索关键字

  用户可以在系统日志右上角的搜索框搜索关键字。系统支持高亮关键字并实现搜索结果间的跳转。

  图 7-20 搜索关键字

说明

- 搜索功能仅支持搜索当前页面加载的日志，若日志加载不全（请关注页面提示）则需要下载或者通过打开全量日志访问链接进行搜索
- 全量日志访问链接打开的新页面可以通过Ctrl+F进行搜索

- 训练故障识别

  如果您的训练作业运行失败，我们会自动识别导致您作业失败的原因，在界面上给出提示。提示包括三部分：失败的可能原因、推荐的解决方案以及对应的日志（底色标红）。解决方案可能会有以下类型：

  1.文档：会提供当前故障对应的一个指导文档，请按照文档提示修复问题。

  2.重建作业：建议重建作业进行重试，大概率能修复问题。若重试后仍然失败，建议提工单。

  3.提工单：建议提工单解决。

图 7-21 训练故障识别

说明

- 系统会对部分常见训练错误给出分析建议，目前还不能识别所有错误。
- 提供的失败可能原因仅供参考。
- 如果提供的解决方案不能解决您的问题，可以提工单进行人工咨询。
- 针对分布式作业，只会显示当前节点的一个分析结果，作业的失败需要综合各个节点的失败原因做一个综合判断。

训练日志说明

如果选择Ascend 910规格创建训练作业，日志结构举例说明如下：

```
obs://dgg-test-user/ascend910-test-cases/log-out/                    # 作业日志路径
├── modelarts-job-9ccf15f2-6610-42f9-ab99-059ba049a41e-proc-rank-0-device-0.txt    # proc log  单卡训练日志
└── modelarts-job-9ccf15f2-6610-42f9-ab99-059ba049a41e-worker-0.log          # training log 训练日志合集
```

```
└──── ascend                                    # Ascend日志
    └──── log                  # 默认不生成该文件
        ├──── device       # device log  默认不生成
        │     └──── ...
        └──── plog
              └──── ...
```

主要生成日志类型如下：

- **训练平台日志**

  training log是训练日志合集，proc log是单卡训练日志重定向文件，方便用户快速定位对应计算节点的日志。

- **Ascend 日志**

  默认设置 plog 日志打屏，“ascend/log” 文件夹及相关文件默认不生成日志文件。

  您可以通过ma-pre-start 脚本修改默认环境变量配置。

```
ASCEND_GLOBAL_LOG_LEVEL=3     # 日志级别设置  debug级别为0;info 级别为1;warning级别为 2;error级别为4
ASCEND_SLOG_PRINT_TO_STDOUT=1  # plog日志是否打屏
ASCEND_GLOBAL_EVENT_ENABLE=1   # 设置事件级别  不开启Event日志级别为0；开启Event日志级别为1
```

说明

ma-pre-start脚本

在与训练启动文件同级的目录下放置 ma-pre-start.sh or ma-pre-start.py 脚本。

在训练启动文件被执行前，系统会在 /home/work/user-job-dir/ 目录下执行上述 pre-start 脚本

使用该机制可以更新容器镜像内安装的 Ascend RUN 包，或者设置一些训练运行时额外需要的全局环境变量。

训练平台日志

```
样例：modelarts-job-9ccf15f2-6610-42f9-ab99-059ba049a41e-worker-0.log
统一日志格式：modelarts-job-[job id]-[task id].log
```

其中 task id 为分布式作业中的节点id，单机作业的 task id 默认为 worker-0。

training log中包括“pip-requirement.txt 安装日志”、“ma-pre-start 日志”、“davincirun 日志”、“训练进程日志”、“MA平台日志”等。

- **MA平台日志**

  类型一：[ModelArts Service Log] xxx

```
[ModelArts Service Log][init] download code_url: s3://dgg-test-user/ascend910-test-cases/mindspore/lenet/
```

  类型二：time=“xxx” level=“xxx” msg=“xxx” file=“xxx” Command=xxx Component=xxx Platform=xxx

```
time="2021-07-26T19:24:11+08:00" level=info msg="start the periodic upload task, upload period = 5 seconds " file="upload.go:46" Command=obs/upload Component=ma-training-toolkit Platform=ModelArts-Service
```

  单机作业只会生成一个 training log 文件。MA平台日志主要用于运维人员定位平台问题，如果不涉及可以通过上述类型模式规则进行筛除。

- **proc log**

  proc log是单卡训练日志重定向文件，方便用户快速定位对应计算节点的日志。

```
样例：modelarts-job-9b43aba3-ed9c-48c1-9f5e-2d61d20dce65-proc-rank-6-device-6.txt
日志文件格式：  [modelarts-job-uuid]-proc-rank-[rank id]-device-[device logic id].txt
```

device id 为顺序编号 id。

Ascend 日志

样例：

```
└────── ascend
   └────── log
      ├────── device        # device log
      │       └────── ...
      └────── plog        # plog
         └────── ...
```

表 7-12 Ascend 日志说明

日志说明	日志产生源	日志落盘位置
用户进程,在HOST侧产生的日志（例如:ACL /GE）	HOST侧	HOST侧: Linux: ~/ascend/log/plog
用户进程，在DEVICE侧产生的AICPU、HCCP的日志，回传到HOST侧	DEVICE侧	HOST侧: Linux:~/ascend/log/device-{device-id}

说明

由于 Device Log 在训练进程结束后获取；因此假若出现如下情况，Ascend Device Log 会获取不到。

- 节点异常重启
- 被主动停止的节点

7.4.2.4 资源占用情况

用户可以通过资源占用情况窗口查看计算节点的资源使用情况。

操作一：如果训练作业使用多个计算节点，可以通过实例名称的下拉框切换节点。

操作二：单击图例“cpuUsage”、“gpuMemUsage”、“gpuUtil”、“memUsage”“npuMemUsage”、“npuUtil”、可以添加或取消对应参数的使用情况图。

操作三：鼠标悬浮在图片上的时间节点，可查看对应时间节点的占用率情况。

图 7-22 资源占用情况

表 7-13 参数说明

参数	说明
cpuUsage	cpu使用率。
gpuMemUsage	gpu内存使用率。
gpuUtil	gpu使用情况
memUsage	内存使用率。
npuMemUsage	npu内存使用率。
npuUtil	npu使用情况

7.4.2.5 评估结果

针对使用ModelArts官方发布的算法创建训练作业时，其训练作业详情支持查看评估结果。根据模型训练的业务场景不同，会提供不同的评估结果。由于每个模型情况不同，系统将自动根据您的模型指标情况，给出一些调优建议，请仔细阅读界面中的建议和指导，对您的模型进行进一步的调优。

如果您的训练脚本中按照ModelArts规范添加了相应的评估代码，在训练作业运行结束后，也可在作业详情页面查看评估结果。

图 7-23 评估结果

7.4.3 停止、重建或查找作业

停止训练作业

在训练作业列表中，针对“创建中”、“等待中”、“运行中”的训练作业，您可以单击“操作”列的“终止”，停止正在运行中的训练作业。

训练作业停止后，ModelArts将停止计费。

运行结束的训练作业，如“已完成”、“运行失败”、“已终止”、“异常”的作业，不涉及“终止”操作。

重建训练作业

当对创建的训练作业不满意时，您可以单击操作列的重建，重新创建训练作业。在重创训练作业页面，会自动填入上一次训练作业设置的参数，您仅需在原来的基础上进行修改即可重新创建训练作业。

查找训练作业

当用户使用IAM账号登录时，训练作业列表会显示IAM账号下所有训练作业。ModelArst提供查找训练作业功能帮助用户快速查找训练作业。

操作一：打开“只显示自己”按钮。打开按钮后，训练作业列表仅显示当前子账号下创建的训练作业。

操作二：支持筛选训练作业状态。

操作三：支持查询训练作业名称。

操作四：支持高级搜索。高级搜索包括算法名称筛选。

图 7-24 查找训练作业

7.4.4 清除训练作业资源

如果不再需要使用此训练任务，建议清除相关资源，避免产生不必要的费用。

- 在“训练作业”页面，“删除”运行结束的训练作业。您可以单击“操作”列的“删除”，删除对应的训练作业。
- 进入OBS，删除本示例使用的OBS桶及文件。

完成资源清除后，您可以在总览页面的使用情况确认资源删除情况。

图 7-25 查看使用情况

7.5 超参搜索训练

7.5.1 超参搜索简介

ModelArts新版训练中新增了超参搜索功能，自动实现模型超参搜索，为您的模型匹配最优的超参。

在模型训练过程中，有很多超参需要根据任务进行调整，比如learning_rate、weight_decay等等，这一工作往往需要一个有经验的算法工程师花费一定精力和大量时间进行手动调优。ModelArts支持的超参搜索功能，在无需算法工程师介入的情况下，即可自动进行超参的调优，在速度和精度上超过人工调优。

ModelArts支持以下三种超参搜索算法：

- **贝叶斯优化(SMAC)**
- **TPE算法**
- **模拟退火算法(Anneal)**

7.5.2 搜索算法

7.5.2.1 贝叶斯优化(SMAC)

贝叶斯优化假设超参和目标函数存在一个函数关系。基于已搜索超参的评估值，通过高斯过程回归来估计其他搜索点处目标函数值的均值和方差。根据均值和方差构造采集函数（Acquisition Function），下一个搜索点为采集函数的极大值点。相比网格搜索，贝叶斯优化会利用之前的评估结果，从而降低迭代次数、缩短搜索时间；缺点是不容易找到全局最优解。

表 7-14 贝叶斯优化的参数说明

参数	说明	取值参考
num_samples	搜索尝试的超参组数	int，一般在10-20之间，值越大，搜索时间越长，效果越好
kind	采集函数类型	string，默认为'ucb'，可能取值还有'ei'、'poi'，一般不建议用户修改
kappa	采集函数ucb的调节参数，可理解为上置信边界	float，一般不建议用户修改
xi	采集函数poi和ei的调节参数	float，一般不建议用户修改

7.5.2.2 TPE 算法

TPE算法全称Tree-structured Parzen Estimator，是一种利用高斯混合模型来学习超参模型的算法。在每次试验中，对于每个超参，TPE为与最佳目标值相关的超参维护一个高斯混合模型l(x)，为剩余的超参维护另一个高斯混合模型g(x)，选择l(x)/g(x)最大化时对应的超参作为下一组搜索值。

表 7-15 TPE 算法的参数说明

参数	说明	取值参考
num_samples	搜索尝试的超参组数	int，一般在10-20之间，值越大，搜索时间越长，效果越好
n_initial_points	采用TPE接近目标函数之前，对目标函数的随机评估数	int，一般不建议用户修改
gamma	TPE算法的一定分位数，用于划分l(x)和g(x)	float，范围(0,1)，一般不建议用户修改

7.5.2.3 模拟退火算法(Anneal)

模拟退火算法即Anneal算法，是随机搜索中一个简单但有效的变体，它利用了响应曲面中的平滑度。退火速率不自适应。Anneal算法从先前采样的一个试验点作为起点，然后从与先验分布相似的分布中采样每组超参数，但其密度更集中在我们选择的试验点周围。随着时间推移，算法会倾向于从越来越接近最佳点处采样。在采样过程中，算法可能绘制一个次佳试验作为最佳试验，以一定概率跳出局部最优解。

表 7-16 模拟退火算法的参数说明

参数	说明	取值参考
num_samples	搜索尝试的超参组数	int，一般在10-20之间，值越大，搜索时间越长，效果越好
avg_best_idx	要探索试验的几何分布平均，从按照分数排序的试验中选择	float，一般不建议用户修改
shrink_coef	随着更多的点被探索，邻域采样大小的减少率	float，一般不建议用户修改

7.5.3 创建超参搜索作业

背景信息

对于用户希望优化的超参，需在“超参”设置中定义，可以给定名称、类型、默认值、约束等，具体设置方法可以参考定义超参。

如果用户使用的AI引擎为“PyTorch-1.4.0-python3.6-v2”或“TF-1.13.1-python3.6-v2”，并且优化的超参类型为float类型，ModelArts支持用户使用超参搜索功能。

在0代码修改的基础下，实现算法模型的超参搜索。需要完成以下步骤：

1. 准备工作
2. 创建算法
3. 创建训练作业（New）
4. 查看超参搜索作业详情

准备工作

- 数据已完成准备：已在ModelArts中创建可用的数据集，或者您已将用于训练的数据集上传至OBS目录。
- 请准备好训练脚本，并上传至OBS目录。训练脚本开发指导参见开发自定义脚本。
- 在训练代码中，用户需打印搜索指标参数。
- 已在OBS创建至少1个空的文件夹，用于存储训练输出的内容。
- 由于训练作业运行需消耗资源，确保帐户未欠费。
- 确保您使用的OBS目录与ModelArts在同一区域。

创建算法

进入ModelArts控制台，参考创建算法操作指导，创建自定义算法。在配置自定义算法参数时，需关注“超参”和“支持的策略”参数的设置。

图 7-26 超参搜索作业参数

对于用户希望优化的超参，需在“超参”设置中定义，可以给定名称、类型、默认值、约束等，具体设置方法可以参考**定义超参**。

单击勾选自动搜索，用户为算法设置算法搜索功能。自动搜索作业运行过程中，ModelArts后台通过指标正则表达式获取搜索指标参数，朝指定的优化方向进行超参优化。用户需要在代码中打印搜索参数并在控制台配置以下参数：

- 搜索指标

  搜索指标为目标函数的值，通常可以设置为loss、accuracy等。通过优化搜索指标的目标值超优化方向收敛，找到最契合的超参，提高模型精度和收敛速度。

表 7-17 搜索指标参数

参数	说明
名称	搜索指标的名称。需要与您在代码中打印的搜索指标参数保持一致。
优化方向	可选“最大化”或者“最小化”。
指标正则	填入正则表达式。您可以单击智能生成功能自动获取正则表达式。

图 7-27 搜索指标参数

- 设置自动化搜索参数

  从已设置的“超参”中选择可用于搜索优化的超参。优化的超参仅支持float类型，选中自动化搜索参数后，需设置取值范围。

  图 7-28 自动化参数设置

- 搜索算法配置

  ModelArts内置三种超参搜索算法，用户可以根据实际情况选择对应的算法，支持多选。对应的算法和参数解析请参考以下：

  – bayes_opt_search:**贝叶斯优化(SMAC)**
  – tpe_search:**TPE算法**
  – anneal_search:**模拟退火算法(Anneal)**

  图 7-29 搜索算法配置

创建算法完成后，在“算法管理”页面，等待算法就绪。当新创建的算法状态变更为“就绪”时，即可执行其他操作。

创建训练作业（New）

登录ModelArts控制台，参考**创建训练作业**操作指导，创建训练作业。用户需关注以下操作才能开启超参搜索。

当您选择支持超参搜索的算法，需单击超级参数的范围设置按钮才能开启超参搜索功能。

图 7-30 范围设置

开启超参搜索功能后，用户可以设置搜索指标、搜索算法和搜索算法参数。三个参数显示的支持值与**算法管理模块的超参设置**一一对应。

图 7-31 超参搜索参数设置

完成超参搜索作业的创建后，训练作业需要运行一段时间。

查看超参搜索作业详情

训练作业运行结束后，您可以通过查看训练作业详情判断此训练作业是否满意。

进入ModelArts控制台，您可以参考**查看作业详情**操作指导。如果训练作业是超参搜索作业，用户可以查看自动超参搜索结果。

图 7-32 超参搜索结果

	time_total_s	config	pid	reward_attr	worker_index	training_ite...	done	trial_id	acc	best_reward
☑ 0	4.17093276...	{'learning_r...	252	0.03125		6	True	a1996874	0.03125	0.03125
☑ 1	3.59161567...	{'learning_r...	253	0.0625		6	True	a47b6128	0.0625	0.0625
☐ 2	3.70686960...	{'learning_r...	359	0.09375		6	True	a7120b8a	0.09375	0.09375
☐ 3	3.55232930...	{'learning_r...	388	0.09375		6	True	aa04e060	0.09375	0.09375
☐ 4	3.67098760...	{'learning_r...	417	0.09375		6	True	acdbeb44	0.09375	0.09375
☐ 5	3.60248303...	{'learning_r...	446	0.125		6	True	afc26b62	0.125	0.125

7.6 旧版训练与新版训练差异

新版训练和旧版训练的差异主要体现在以下3点:

- **新旧版创建训练作业方式差异**
- **新旧版训练代码适配的差异**
- **新旧版训练预置引擎差异**

新旧版创建训练作业方式差异

旧版训练支持使用自定义算法、使用预置算法、使用常用框架、使用自定义镜像方式创建训练作业。

新版训练支持使用市场订阅算法和自定义算法创建训练作业。

新版训练的创建方式有了更明确的类别划分，不影响用户的原本的训练作业，只是选择方式存在区别。

旧版中使用自定义算法和使用常用框架创建训练作业的用户，可以在新版训练中**使用自定义算法方式**创建训练作业。

旧版中使用预置算法的用户，可以在新版训练中**使用订阅算法**创建训练作业。

旧版中使用自定义镜像创建训练作业的用户，可以在新版训练中**使用自定义镜像创建训练作业**。

新旧版训练代码适配的差异

旧版训练中，用户需要在输入输出数据上做如下配置：

```
#解析命令行参数
import argparse
parser = argparse.ArgumentParser(description='MindSpore Lenet Example')
parser.add_argument('--data_url', type=str, default="./Data",
                    help='path where the dataset is saved')
parser.add_argument('--train_url', type=str, default="./Model", help='if is test, must provide\
                    path where the trained ckpt file')
args = parser.parse_args()
...
#下载数据参数至容器本地，在代码中使用local_data_path代表训练输入位置
mox.file.copy_parallel(args.data_url, local_data_path)
...
#上传容器本地数据至obs路径
mox.file.copy_parallel(local_output_path, args.train_url)
```

新版训练中，用户配置输入输出数据，无需书写下载数据的代码，在代码中把arg.data_url和arg.train_url当做本地路径即可，详情参考**开发自定义脚本**。

```
#解析命令行参数
import argparse
parser = argparse.ArgumentParser(description='MindSpore Lenet Example')
parser.add_argument('--data_url', type=str, default="./Data",
                    help='path where the dataset is saved')
parser.add_argument('--train_url', type=str, default="./Model", help='if is test, must provide\
                    path where the trained ckpt file')
args = parser.parse_args()
...
# 下载的代码无需设置，后续涉及训练数据和输出路径数据使用data_url和train_url即可
#下载数据参数至容器本地，在代码中使用local_data_path代表训练输入位置
#mox.file.copy_parallel(args.data_url, local_data_path)
...
#上传容器本地数据至obs路径
#mox.file.copy_parallel(local_output_path, args.train_url)
```

新旧版训练预置引擎差异

- 新版的预置训练引擎默认安装Moxing2.0.0及以上版本。
- 新版的预置训练引擎统一使用了Python3.7及以上版本。
- 新版镜像修改了默认的HOME目录，由“/home/work”变为“/home/ma-user”，请注意识别训练代码中是否有“/home/work”的硬编码。

- 提供预置引擎类型有差异。新版的预置引擎在常用的训练引擎上进行了升级。

  如果您需要使用旧版训练引擎，单击显示旧版引擎即可选择旧版引擎。新旧版支持的预置引擎差异请参考**表7-18**。详细的训练引擎版本说明请参考**新版训练和旧版训练分别支持的AI引擎**。

表 7-18 新旧版预置引擎差异

工作环境	预置训练引擎与版本	旧版训练	新版训练
TensorFlow	Tensorflow-1.8.0	√	x
	Tensorflow-1.13.1	√	后续版本支持
	Tensorflow-2.1.0	√	√
MXNet	MXNet-1.2.1	√	x
Caffe	Caffe-1.0.0	√	x
Spark_MLlib	Spark-2.3.2	√	x
Ray	RAY-0.7.4	√	x
XGBoost-Sklearn	XGBoost-0.80-Sklearn-0.18.1	√	x
PyTorch	PyTorch-1.0.0	√	x
	PyTorch-1.3.0	√	x
	PyTorch-1.4.0	√	x
	PyTorch-1.8.0	x	√
Ascend-Powered-Engine	Mindspore-1.1.1	√	x
	Mindspore-1.3.0	x	√
	Tensorflow-1.15	√	√
MPI	MindSpore-1.3.0	x	√
Horovod	horovod_0.20.0-tensorflow_2.1.0	x	√
	horovod_0.22.1-pytorch_1.8.0	x	√
MindSpore-GPU	MindSpore-1.1.0	√	x
	MindSpore-1.2.0	√	x

8 训练管理（旧版即将下线）

8.1 模型训练简介

说明

当前ModelArts同时存在新版训练管理和旧版训练管理功能。旧版训练管理功能仅对部分存量用户可见，新用户不可见。

当前章节仅介绍了旧版训练，旧版训练即将下线，推荐用户使用新版训练，新版训练管理文档请参考模型开发。

新旧训练的差异请参考常见问题。

ModelArts提供了模型训练的功能，方便您查看训练情况并不断调整您的模型参数。您还可以基于不同的数据，选择不同规格的资源池用于模型训练。

模型训练功能说明

表 8-1 功能说明

功能	说明	详细指导
训练作业管理	支持创建训练作业、查看训练作业详情、管理训练作业版本、并且支持查看评估详情。	创建训练作业简介 管理训练作业版本 查看作业详情
作业参数管理	您可以将某一个训练作业的参数配置保存为作业参数，包含数据来源、算法来源、运行参数、资源池等参数信息，已保存的作业参数，可一键式应用到创建新的训练作业，大大提高效率。	管理作业参数
模型训练可视化	TensorBoard和MindInsight是可视化工具，能够有效地展示模型运行过程中的计算图、各种指标随着时间的变化趋势以及训练中使用到的数据信息。	管理可视化作业

8.2 订阅算法

ModelArts的AI Gallery，发布了较多官方算法，可以帮助AI开发者快速开始训练和部署模型。对于不熟悉ModelArts的用户，可以快速订阅官方推荐算法实现模型训练全流程。

AI Gallery不仅可以订阅官方发布算法，也支持用户发布自定义算法和订阅其他开发者分享的算法。为了使用他人或者ModelArts官方分享的算法，您需要将AI Gallery的算法订阅至您的ModelArts中。

- 查找算法
- 订阅算法

查找算法

为了获得匹配您业务的算法，您可以通过多个入口区查找算法。

- 在ModelArts控制台，“算法管理>我的订阅”中，单击“订阅更多算法”，可跳转至“AI Gallery”页面，查找相应的算法。
- 在ModelArts控制台，直接在左侧菜单栏中选择“AI Gallery”，进入“AI Gallery”页面，在“资产集市 > 算法”页面查找相应的算法。

  在AI Gallery的算法模块，根据算法的业务场景，官方发布的算法如表8-2所示。

表 8-2 官方发布算法

应用场景	算法	AI引擎	支持服务类型	训练使用的资源类型	部署使用的资源类型
图像分类	ResNet_v1_50	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	ResNet_v1_101	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	ResNet_v1_50	TensorFlow	在线服务	Ascend 910	CPU、GPU、Ascend 310
	ResNet50	MindSpore	在线服务	Ascend 910	Ascend 310
	ResNet_v2_50	TensorFlow	在线服务	Ascend 910	Ascend 310
	ResNet_v2_101	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	MobileNet_v1	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	MobileNet_v2	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310

应用场景	算法	AI引擎	支持服务类型	训练使用的资源类型	部署使用的资源类型
	Inception_v3	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	ResNet50_AdaInfer	TensorFlow	在线服务	GPU	CPU、GPU
	Res2Net_50	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	Inception_ResNet_v2	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	EfficientNetB0/B4/B7/B8	PyTorch	在线服务	GPU	CPU、GPU
物体检测	RetinaNet_ResNet50	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	YOLOv3_ResNet18	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	YOLOv3_ResNet18	TensorFlow	在线服务	Ascend 910	Ascend 310
	YOLOv3_Darknet53	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	SSD_VGG	Caffe	在线服务	GPU	暂不支持
	EfficientDet	Tensorflow	在线服务	GPU	CPU、GPU
	YOLOv5	Pytorch	在线服务	GPU	CPU、GPU
文本分类	BERT	TensorFlow	在线服务	GPU	CPU、GPU、Ascend 310
	NEZHA	TensorFlow	在线服务	GPU	GPU
图像语义分割	DeepLabV3	TensorFlow	在线服务	GPU	Ascend 310
	DeepLabV3	TensorFlow	在线服务	Ascend 910	Ascend 310
声音分类	声音分类	TensorFlow	在线服务	GPU	CPU、GPU
时序预测	时序预测	PyTorch	在线服务	GPU	暂不支持

应用场景	算法	AI引擎	支持服务类型	训练使用的资源类型	部署使用的资源类型
强化学习	强化学习GameAI	TensorFlow	在线服务	GPU	GPU

订阅算法

1. 进入“AI Gallery”，选择“资产集市 > 算法”页签，查找您需要的算法并单击算法名称，进入算法详情页，单击算法详情页右侧的“订阅”，订阅您所需的算法。

   订阅后的算法，状态变为“已订阅”，并且将自动展现在“算法管理 > 我的订阅”页面中。

2. 单击“前往控制台”，选择云服务区域，进入“算法管理 > 我的订阅”页面进入此页面内，单击“产品名称”左侧的小三角，展开算法详情，在“版本列表”区域，单击“创建训练作业”即可进行后续操作。

   图 8-1 订阅算法

8.3 常用框架

本章详细介绍ModelArts支持的常用AI框架以及使用AI框架编写创建训练作业的训练代码。

训练管理支持的 AI 常用框架

当前ModelArts支持的AI引擎及对应版本如下所示。

表 8-3 旧版训练作业支持的 AI 引擎

工作环境	适配芯片	系统架构	系统版本	AI引擎与版本	支持的cuda或Ascend版本
TensorFlow	CPU/GPU	x86_64	Ubuntu16.04	TF-1.8.0-python2.7	cuda9.0
				TF-1.8.0-python3.6	
				TF-1.13.1-python2.7	cuda10.0
				TF-1.13.1-python3.6	
				TF-2.1.0-python3.6	cuda10.1
MXNet	CPU/GPU	x86_64	Ubuntu16.04	MXNet-1.2.1-python2.7	cuda9.0
				MXNet-1.2.1-python3.6	
Spark_MLlib	CPU	x86_64	Ubuntu16.04	Spark-2.3.2-python2.7	-
				Spark-2.3.2-python3.6	-
Ray	CPU/GPU	x86_64	Ubuntu16.04	RAY-0.7.4-python3.6	cuda10.0
XGBoost-Sklearn	CPU	x86_64	Ubuntu16.04	XGBoost-0.80-Sklearn-0.18.1-python2.7	-
				XGBoost-0.80-Sklearn-0.18.1-python3.6	-
PyTorch	CPU/GPU	x86_64	Ubuntu16.04	PyTorch-1.0.0-python2.7	cuda9.0
				PyTorch-1.0.0-python3.6	
				PyTorch-1.3.0-python2.7	cuda10.0
				PyTorch-1.3.0-python3.6	

工作环境	适配芯片	系统架构	系统版本	AI引擎与版本	支持的cuda或Ascend版本
				PyTorch-1.4.0-python3.6	cuda10.1
Ascend-Powered-Engine	Ascend910	aarch64	Euler2.8	MindSpore-1.3-can5.0.2-python3.7-euleros2.8-aarch64	can5.0.2
				TF-1.15-can5.0.2-python3.7-euleros2.8-aarch64	can5.0.2
MindSpore-GPU	CPU/GPU	x86_64	Ubuntu18.04	MindSpore-1.1.0-python3.7 MindSpore-1.2.0-python3.7	cuda10.1
Caffe	CPU/GPU	x86_64	Ubuntu16.04	Caffe-1.0.0-python2.7	cuda8.0

说明

- MoXing是ModelArts团队自研的分布式训练加速框架，它构建于开源的深度学习引擎TensorFlow、MXNet、PyTorch、Keras之上，详细说明请参见**MoXing使用说明**。如果您使用的是MoXing框架编写训练脚本，在创建训练作业时，请根据您选用的接口选择其对应的AI引擎和版本。
- “efficient_ai”是华为云ModelArts团队自研的加速压缩工具，它支持对训练作业进行量化、剪枝和蒸馏来加速模型推理速度，详细说明请参见**efficient_ai使用说明**。
- **Ascend-Powered-Engine**仅在“华北-北京四”区域支持。

使用常见框架的训练代码开发

当您使用常用框架创建训练作业时，您需要在创建页面提供代码目录路径、代码目录路径中的启动文件、训练数据路径以及训练输出路径。这四种路径搭建了用户和ModelArts后台交互的桥梁。

- 代码目录路径

  您需要在OBS桶中指定代码目录，并将训练代码、依赖安装包或者预生成模型等训练所需文件上载至该代码目录下。训练作业创建完成后，ModelArts会将代码目录及其子目录下载至后台容器中。

- 代码目录路径中的启动文件

  代码目录路径中的启动文件作为训练启动的入口，当前只支持python格式。

- 训练数据路径

  请注意不要将训练数据路径放在代码目录路径下。训练数据比较大，训练代码目录在启动后会下载至后台，可能会有下载失败的风险。

在训练作业启动后，ModelArts会挂载硬盘至“/cache”目录，用户可以使用此目录来存储临时文件。“/cache”目录大小请参考**训练环境中不同规格资源“/cache”目录的大小**。

您需要将训练数据上传至OBS桶另外的路径，并在训练代码中需通过解析命令行参数“data_url”下载训练数据至“/cache”目录。请保证您设置的桶路径有读取权限。

- 训练输出路径

  建议设置一个空目录为训练输出路径。在训练代码中，您需要解析命令行参数“train_url”上载训练输出至指定的训练输出路径，请保证您设置的桶路径有写入权限和读取权限。

当您使用常用框架创建训练作业时，您需要实现训练代码的开发。在ModelArts中，训练代码需包含以下步骤：

图 8-2 训练代码开发说明

1. （可选）引入依赖

   当您使用常见框架创建训练作业的时候，如果您的模型引用了其他依赖，您需要在创建训练作业的“代码目录”下放置相应的文件或安装包。

- 安装python依赖包请参考**模型中引用依赖包时，如何创建训练作业?**
- 安装C++的依赖库请参考**如何安装C++的依赖库?**
- 在预训练模型中加载参数请参考**如何在训练中加载部分训练好的参数?**

图 8-3 选择常用框架并指定模型启动文件

2. 解析必选参数“data_url”、“train_url”

   在使用常见框架创建训练作业时，您需要在创建训练作业页面填写作业参数配置相关信息。

   “data_url”：训练数据是训练代码开发中必不可少的输入。在创建训练作业时，您需要在作业参数配置“数据来源”。训练代码中的“data_url”指代“数据来源”的路径。

   “train_url”：模型训练结束后，训练模型以及相关输出信息需保存在OBS路径。在创建训练作业时，您需要在作业参数配置“训练输出位置”。训练代码中的“train_url”指代“训练输出位置”的OBS路径。

图 8-4 作业参数配置相关信息

在训练代码中需解析“data_url”、“train_url”，ModelArts推荐以下方式实现参数解析。

```
import argparse
# 创建解析
parser = argparse.ArgumentParser(description="train mnist",
                                 formatter_class=argparse.ArgumentDefaultsHelpFormatter)
# 添加参数
parser.add_argument('--train_url', type=str, default='obs://obs-test/ckpt/mnist',
                    help='the path model saved')
parser.add_argument('--data_url', type=str, default='obs://obs-test/data/', help='the training data')
# 解析参数
args, unkown = parser.parse_known_args()
```

3. 从“data_url”导入训练数据

   已知训练数据路径为“data_url”，ModelArts推荐采用**Moxing接口**实现训练数据下载到“cache”目录。

```
import moxing as mox
mox.file.copy_parallel(args.data_url, “/cache”)
```

4. 训练代码正文和保存模型

   训练代码正文和保存模型涉及的代码与您使用的AI引擎密切相关。以下案例以Tensorflow框架为例，训练代码中解析参数方式采用tensorflow接口tf.flags.FLAGS接受命令行参数：

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import os

import tensorflow as tf
from tensorflow.examples.tutorials.mnist import input_data

import moxing as mox

tf.flags.DEFINE_integer('max_steps', 1000, 'number of training iterations.')
tf.flags.DEFINE_string('data_url', '/home/jnn/nfs/mnist', 'dataset directory.')
tf.flags.DEFINE_string('train_url', '/home/jnn/temp/delete', 'saved model directory.')

FLAGS = tf.flags.FLAGS


def main(*args):
    mox.file.copy_parallel(FLAGS.data_url, '/cache/data_url')

    # Train model
    print('Training model...')
    mnist = input_data.read_data_sets('/cache/data_url', one_hot=True)
    sess = tf.InteractiveSession()
    serialized_tf_example = tf.placeholder(tf.string, name='tf_example')
    feature_configs = {'x': tf.FixedLenFeature(shape=[784], dtype=tf.float32),}
    tf_example = tf.parse_example(serialized_tf_example, feature_configs)
    x = tf.identity(tf_example['x'], name='x')
    y_ = tf.placeholder('float', shape=[None, 10])
    w = tf.Variable(tf.zeros([784, 10]))
    b = tf.Variable(tf.zeros([10]))
    sess.run(tf.global_variables_initializer())
    y = tf.nn.softmax(tf.matmul(x, w) + b, name='y')
    cross_entropy = -tf.reduce_sum(y_ * tf.log(y))

    tf.summary.scalar('cross_entropy', cross_entropy)

    train_step = tf.train.GradientDescentOptimizer(0.01).minimize(cross_entropy)

    correct_prediction = tf.equal(tf.argmax(y, 1), tf.argmax(y_, 1))
    accuracy = tf.reduce_mean(tf.cast(correct_prediction, 'float'))
    tf.summary.scalar('accuracy', accuracy)
    merged = tf.summary.merge_all()
    test_writer = tf.summary.FileWriter('/cache/train_url', flush_secs=1)

    for step in range(FLAGS.max_steps):
        batch = mnist.train.next_batch(50)
        train_step.run(feed_dict={x: batch[0], y_: batch[1]})
        if step % 10 == 0:
            summary, acc = sess.run([merged, accuracy], feed_dict={x: mnist.test.images, y_:
mnist.test.labels})
            test_writer.add_summary(summary, step)
            print('training accuracy is:', acc)
    print('Done training!')

    builder = tf.saved_model.builder.SavedModelBuilder(os.path.join('/cache/train_url', 'model'))

    tensor_info_x = tf.saved_model.utils.build_tensor_info(x)
    tensor_info_y = tf.saved_model.utils.build_tensor_info(y)
```

```
    prediction_signature = (
        tf.saved_model.signature_def_utils.build_signature_def(
            inputs={'images': tensor_info_x},
            outputs={'scores': tensor_info_y},
            method_name=tf.saved_model.signature_constants.PREDICT_METHOD_NAME))

    builder.add_meta_graph_and_variables(
        sess, [tf.saved_model.tag_constants.SERVING],
        signature_def_map={
            'predict_images':
                prediction_signature,
        },
        main_op=tf.tables_initializer(),
        strip_default_attrs=True)

    builder.save()

    print('Done exporting!')

    mox.file.copy_parallel('/cache/train_url', FLAGS.train_url)


if __name__ == '__main__':
    tf.app.run(main=main)
```

5. 导出训练模型至“train_url”

   已知训练输出位置为“train_url”，ModelArts推荐采用**Moxing接口**实现输出结果从后台自定义目录“/cache/train_url”目录导出至“train_url”目录。

```
mox.file.copy_parallel(“/cache/train_url”, args.train_url)
```

8.4 创建训练作业

8.4.1 创建训练作业简介

在整个AI全流程开发过程中，ModelArts支持多种类型的训练作业。根据您的算法来源不同，选择不同的创建方式。

训练作业的几种算法来源

- **算法管理**

  算法管理中，管理了用户自己创建的算法和AI Gallery订阅的算法，您可以使用算法管理中的算法，快速创建训练作业，构建模型。操作指导请参见**使用已有算法训练模型**。

- **常用框架**

  如果您已在本地使用一些常用框架完成算法开发，您可以选择常用框架，创建训练作业来构建模型，操作指导请参见**使用常用框架训练模型**。

- **自定义**

  如果您开发算法时使用的框架并不是常用框架，您可以将算法构建为一个自定义镜像，通过自定义镜像创建训练作业。创建训练作业的操作指导请参见**使用自定义镜像训练模型**，自定义镜像的相关规范和说明请参见**训练作业自定义镜像规范**。

8.4.2 使用已有算法训练模型

针对您创建的算法，或者是从AI Gallery订阅的算法，支持快速使用此算法创建训练作业，构建模型。

前提条件

- 数据已完成准备：已在ModelArts中创建可用的数据集，或者您已将用于训练的数据上传至OBS目录。
- “算法管理”中，已创建算法或者**订阅算法**。新创建的算法仅在新版训练中支持，请参见**模型开发>创建算法**章节。
- ModelArts在AI Gallery提供了多个云端算法供用户选择，支持的算法列表及其参数说明请参见**官方发布的预置算法列表**。
- 已在OBS创建至少1个空的文件夹，用于存储训练输出的内容。
- 由于训练作业运行需消耗资源，确保帐户未欠费。
- 确保您使用的OBS目录与ModelArts在同一区域。

注意事项

训练作业指定的数据集目录中，用于训练的数据名称（如图片名称、音频文件名、标注文件名称等），名称长度限制为0～255英文字符。如果数据集目录下，部分数据的文件名称超过255英文字符，训练作业将不会使用此数据，使用符合要求的数据继续进行训练。如果数据集目录下，所有数据的文件名称都超过了255英文字符，导致训练作业无数据可用，则会导致训练作业失败。

创建训练作业

1. 登录ModelArts管理控制台，在左侧导航栏中选择“训练管理 > 训练作业”，默认进入“训练作业”列表。
2. 在训练作业列表中，单击左上角“创建”，进入“创建训练作业”页面。
3. 在创建训练作业页面，填写训练作业相关参数，然后单击“下一步”。
   a. 填写基本信息，包含“名称”和“描述”。“版本”信息由系统自动生成，按“V0001”、“V0002”规则命名，用户无法修改。
   b. 填写作业参数。包含数据来源、算法来源等关键信息，详情请参见**表8-4**。数据来源的可选范围与已有算法的使用约束保持一致。

图 8-5 算法来源为算法管理

表 8-4 作业参数说明

参数名称	子参数	说明
算法来源	算法管理	选择“算法管理”，在“算法名称”右侧单击“选择”，进入“算法管理”对话框。 ● 选择“我的算法”页签，您可以根据实际需求选择已创建成功的算法。新创建的算法仅在新版训练中支持，请参见模型开发>创建算法章节。 ● 选择“我的订阅”页签，您可以根据实际需求选择AI Gallery中已订阅的算法。您需要在目标算法的左侧单击下拉三角标，选择合适的版本用于训练作业的创建。如何订阅AI Gallery算法详见订阅算法。
训练输入	数据来源>数据集	从ModelArts数据管理中选择可用的数据集及其版本。 ● “选择数据集”：从右侧下拉框中选择ModelArts系统中已发布的数据集。当ModelArts无可用数据集时，此下拉框为空。 ● “选择版本”：根据“选择数据集”指定版本。
	数据来源>数据存储位置	从OBS桶中选择训练数据。在“数据存储位置”右侧，单击“选择”，从弹出的对话框中，选择数据存储的OBS桶及其文件夹。
训练输出	模型输出	选择训练结果的存储位置（OBS路径）。为避免出现错误，建议选择一个空目录用作“模型输出”，请勿将数据集存储的目录作为训练输出位置。
超参	-	此参数根据您选择的算法不同而不同。 如果创建的算法或订阅的算法，定义了相关的调优参数，则需在创建训练作业时，填写对应调优参数的参数值。您可以单击“增加超参”，添加多条。
作业日志路径	-	选择作业运行中产生的日志文件存储路径。

c. 选择用于训练作业的资源。训练参数的可选范围与已有算法的使用约束保持一致。

表 8-5 资源参数说明

参数名称	说明
资源池	选择训练作业资源池。训练作业支持选择“公共资源池”和“专属资源池”。

参数名称	说明
规格	针对不同的资源类型，选择资源规格。GPU资源性能更佳，CPU资源性价比更高。如果您的算法已定义使用CPU或GPU，根据已有算法约束条件，您可以在有效规格选择合适的资源规格，无效选项置灰不可选。 Ascend 910资源仅在“华北-北京四”可用。 不同的资源类型的数据盘容量是不同的，详细介绍参考**训练环境中不同规格资源“/cache”目录的大小**。
计算节点个数	选择计算节点的个数。默认值为“1”。

d. 配置订阅消息，并设置是否将当前训练作业中的参数保存为作业参数。

图 8-6 配置训练作业订阅消息

表 8-6 订阅消息及作业参数的参数说明

参数名称	说明
订阅消息	订阅消息使用消息通知服务，在事件列表中选择需要监控的资源池状态，在事件发生时发送消息通知。 此参数为可选参数，您可以根据实际情况设置是否打开开关。如果开启订阅消息，请根据实际情况填写如下参数。 ● “主题名”：订阅消息主题名称。您可以单击创建主题，在消息通知服务中创建主题。 ● “事件列表”：订阅事件。当前可选择“OnJobRunning”、“OnJobSucceeded”、“OnJobFailed”三种事件，分别代表作业运行中、运行成功、运行失败。
保存作业参数	勾选此参数，表示将当前训练作业设置的作业参数保存，方便后续一键复制使用。 勾选“保存作业参数”，然后填写“作业参数名称”和“作业参数描述”，即可完成当前参数配置的保存。训练作业创建成功后，您可以从ModelArts的作业参数列表中查看保存的信息，详细操作指导请参见**管理作业参数**。

e. 完成参数填写后，单击“下一步”。

4. 在“规格确认”页面，确认填写信息无误后，单击“提交”，完成训练作业的创建。训练作业一般需要运行一段时间，根据您选择的数据量和资源不同，训练将耗时几分钟到几十分钟不等。

说明

训练作业创建完成后，将立即启动，运行过程中将按照您选择的资源按需计费。

您可以前往训练作业列表，查看训练作业的基本情况。在训练作业列表中，刚创建的训练作业“状态”为“初始化”，当训练作业的“状态”变为“运行成功”时，表示训练作业运行结束，其生成的模型将存储至对应的“训练输出”中。当训练作业的“状态”变为“运行失败”时，您可以单击训练作业的名称，进入详情页面，通过查看日志等手段处理问题。

8.4.3 使用常用框架训练模型

如果您在本地使用一些常用框架完成算法开发，如TensorFlow、MindSpore等AI引擎，您可以选择常用框架，创建训练作业来构建模型。

前提条件

- 数据已完成准备：已在ModelArts中创建可用的数据集，或者您已将用于训练的数据集上传至OBS目录。
- 如果“算法来源”为“常用框架”，请准备好训练脚本，并上传至OBS目录。
- 已在OBS创建至少1个空的文件夹，用于存储训练输出的内容。
- 由于训练作业运行需消耗资源，确保帐户未欠费。
- 确保您使用的OBS目录与ModelArts在同一区域。

注意事项

- 训练作业指定的数据集目录中，用于训练的数据名称（如图片名称、音频文件名、标注文件名称等），名称长度限制为0～255英文字符。如果数据集目录下，部分数据的文件名称超过255英文字符，训练作业将不会使用此数据，使用符合要求的数据继续进行训练。如果数据集目录下，所有数据的文件名称都超过了255英文字符，导致训练作业无数据可用，则会最终导致训练作业失败。
- 训练脚本中，“数据来源”、“训练输出位置”，两个参数必须为OBS路径。当需要对OBS进行读写交互时，建议使用**MoXing接口**进行读写操作。

训练管理支持的 AI 常用框架

当前ModelArts支持的AI引擎及对应版本如下所示。

表 8-7 旧版训练作业支持的 AI 引擎

工作环境	适配芯片	系统架构	系统版本	AI引擎与版本	支持的cuda或Ascend版本
TensorFlow	CPU/GPU	x86_64	Ubuntu16.04	TF-1.8.0-python2.7	cuda9.0
				TF-1.8.0-python3.6	

工作环境	适配芯片	系统架构	系统版本	AI引擎与版本	支持的cuda或Ascend版本
				TF-1.13.1-python2.7	cuda10.0
				TF-1.13.1-python3.6	
				TF-2.1.0-python3.6	cuda10.1
MXNet	CPU/GPU	x86_64	Ubuntu16.04	MXNet-1.2.1-python2.7	cuda9.0
				MXNet-1.2.1-python3.6	
Spark_MLlib	CPU	x86_64	Ubuntu16.04	Spark-2.3.2-python2.7	-
				Spark-2.3.2-python3.6	-
Ray	CPU/GPU	x86_64	Ubuntu16.04	RAY-0.7.4-python3.6	cuda10.0
XGBoost-Sklearn	CPU	x86_64	Ubuntu16.04	XGBoost-0.80-Sklearn-0.18.1-python2.7	-
				XGBoost-0.80-Sklearn-0.18.1-python3.6	-
PyTorch	CPU/GPU	x86_64	Ubuntu16.04	PyTorch-1.0.0-python2.7	cuda9.0
				PyTorch-1.0.0-python3.6	
				PyTorch-1.3.0-python2.7	cuda10.0
				PyTorch-1.3.0-python3.6	
				PyTorch-1.4.0-python3.6	cuda10.1
Ascend-Powered-Engine	Ascend910	aarch64	Euler2.8	MindSpore-1.3-can5.0.2-python3.7-euleros2.8-aarch64	can5.0.2
				TF-1.15-can5.0.2-python3.7-euleros2.8-aarch64	can5.0.2

工作环境	适配芯片	系统架构	系统版本	AI引擎与版本	支持的cuda或Ascend版本
MindSpore-GPU	CPU/GPU	x86_64	Ubuntu18.04	MindSpore-1.1.0-python3.7 MindSpore-1.2.0-python3.7	cuda10.1
Caffe	CPU/GPU	x86_64	Ubuntu16.04	Caffe-1.0.0-python2.7	cuda8.0

说明

- MoXing是ModelArts团队自研的分布式训练加速框架，它构建于开源的深度学习引擎TensorFlow、MXNet、PyTorch、Keras之上，详细说明请参见**MoXing使用说明**。如果您使用的是MoXing框架编写训练脚本，在创建训练作业时，请根据您选用的接口选择其对应的AI引擎和版本。
- **Ascend-Powered-Engine**仅在“华北-北京四”区域支持。
- 如果您的训练脚本支持的版本与训练支持的AI引擎提供的版本存在差异，会出现训练失败的情况

创建训练作业

1. 登录ModelArts管理控制台，在左侧导航栏中选择“训练管理 > 训练作业”，默认进入“训练作业”列表。
2. 在训练作业列表中，单击左上角“创建”，进入“创建训练作业”页面。
3. 在创建训练作业页面，填写训练作业相关参数，然后单击“下一步”。
   a. 填写基本信息。您可以根据实际情况填写“名称”和“描述”信息。“版本”信息由系统自动生成，按“V0001”、“V0002”规则命名，用户无法修改。
   b. 填写作业参数。包含数据来源、算法来源等关键信息，详情请参见**表8-8**。

图 8-7 算法来源为常用框架

表 8-8 作业参数说明

参数名称	子参数	说明
一键式参数配置	-	如果您在ModelArts已保存作业参数，您可以根据界面提示，选择已有的作业参数，快速完成训练作业的参数配置。
算法来源	常用框架	选择“AI引擎”和“版本”，选择“代码目录”及“启动文件”。选择的AI引擎和编写训练代码时选择的框架必须一致。例如编写训练代码使用的是TensorFlow，则在创建训练作业时也要选择TensorFlow。 目前支持的AI引擎及其版本请参见**训练管理支持的AI常用框架**。 如果您的模型需要安装Python依赖包时，请按照ModelArts定义的要求将依赖包及其配置文件放置“代码目录”中，详细说明请参见**模型中引用依赖包时，如何创建训练作业?**

参数名称	子参数	说明
数据来源	数据集	从ModelArts数据管理中选择可用的数据集及其版本。 ● “选择数据集”：从右侧下拉框中选择ModelArts系统中已有的数据集。当ModelArts无可用数据集时，此下拉框为空。 ● “选择版本”：根据“选择数据集”指定的数据集选择其版本。 一个训练作业，支持选择多个数据集，单击增加一个数据集，单击删除当前行指定的数据集。
	数据存储位置	从OBS桶中选择训练数据。在“数据存储位置”右侧，单击“选择”，从弹出的对话框中，选择数据存储的OBS桶及其文件夹。 当“算法来源”选择“常用框架”时，一个训练作业，支持选择多个数据存储路径，单击增加一个数据存储路径，单击删除当前行指定的数据存储路径。
训练输出位置	-	选择训练结果的存储位置。 **说明** 为避免出现错误，建议选择一个空目录用作“训练输出位置”。请勿将数据集存储的目录作为训练输出位置。
运行参数	-	代码中的命令行参数设置值，请根据您编写的算法代码逻辑进行填写，确保参数名称和代码的参数名称保持一致。 例如：train_steps=10000，其中“train_steps”为代码中的某个传参。
作业日志路径	-	选择作业运行中产生的日志文件存储路径。

c. 选择用于训练作业的资源。

图 8-8 选择训练作业资源

表 8-9 资源参数说明

参数名称	说明
资源池	选择训练作业资源池。 训练作业支持选择“公共资源池”和“专属资源池”。公共资源池又可以选择CPU或GPU两种类型，不同类型的资源池，其收费标准不同，详情请参见**价格详情说明**。专属资源池的创建请参见**资源池**。
类型	当选择公共资源池时，需选择资源类型。目前支持CPU和GPU两种类型。 GPU资源性能更佳，CPU资源性价比更高。如果您选择的算法已定义使用CPU或GPU，界面将自动呈现此资源类型，请务必根据要求选择。 不同的资源类型的数据盘容量是不同的，详细介绍参考**训练环境中不同规格资源“/cache”目录的大小**。 **说明** 如果您在训练代码使用的是GPU资源，则在选择资源池时只能选择GPU集群，否则会导致训练作业失败。
规格	针对不同的资源类型，选择资源规格。 其中针对“已有算法”、“常用框架”或“自定义镜像”，ModelArts支持使用Ascend 910创建训练作业，且此资源仅在“华北-北京四”可用。 前缀带“[限时免费]”信息的为免费规格，您可以使用此规格免费体验ModelArts的训练作业功能。此规格的使用注意事项，请参见**免费体验AI全流程开发**。
计算节点个数	选择计算节点的个数。如果节点个数设置为1，表示后台的计算模式是单机模式；如果节点个数设置大于1，表示后台的计算模式为分布式的。请根据实际编码情况选择计算模式。 当“常用框架”选择Caffe时，只支持单机模式，即“计算节点个数”必须设置为“1”。针对其他“常用框架”，您可以根据业务情况选择单机模式或分布式模式。

d. 配置订阅消息，并设置是否将当前训练作业中的参数保存为作业参数。

图 8-9 配置训练作业订阅消息

表 8-10 订阅消息及作业参数的参数说明

参数名称	说明
订阅消息	订阅消息使用消息通知服务，在事件列表中选择需要监控的资源池状态，在事件发生时发送消息通知。 此参数为可选参数，您可以根据实际情况设置是否打开开关。如果开启订阅消息，请根据实际情况填写如下参数。 ● “主题名”：订阅消息主题名称。您可以单击创建主题，在消息通知服务中创建主题。 ● “事件列表”：订阅事件。当前可选择“OnJobRunning”、“OnJobSucceeded”、“OnJobFailed”三种事件，分别代表训练运行中、运行成功、运行失败。
保存作业参数	勾选此参数，表示将当前训练作业设置的作业参数保存，方便后续一键复制使用。 勾选“保存训练参数”，然后填写“作业参数名称”和“作业参数描述”，即可完成当前参数配置的保存。训练作业创建成功后，您可以从ModelArts的作业参数列表中查看保存的信息，详细操作指导请参见**管理作业参数**。

e. 完成参数填写后，单击“下一步”。

4. 在“规格确认”页面，确认填写信息无误后，单击“提交”，完成训练作业的创建。训练作业一般需要运行一段时间，根据您选择的数据量和资源不同，训练时间将耗时几分钟到几十分钟不等。

说明

训练作业创建完成后，将立即启动，运行过程中将按照您选择的资源按需计费。

您可以前往训练作业列表，查看训练作业的基本情况。在训练作业列表中，刚创建的训练作业“状态”为“初始化”，当训练作业的“状态”变为“运行成功”时，表示训练作业运行结束，其生成的模型将存储至对应的“训练输出位置”中。当训练作业的“状态”变为“运行失败”时，您可以单击训练作业的名称，进入详情页面，通过查看日志等手段处理问题。

8.4.4 使用自定义镜像训练模型

如果您开发算法时使用的框架并不是常用框架，您可以将算法构建为一个自定义镜像，通过自定义镜像创建训练作业。

前提条件

- 数据已完成准备：已在ModelArts中创建可用的数据集，或者您已将用于训练的数据集上传至OBS目录。
- 如果“算法来源”为“自定义”，请按照规范完成镜像制作，并上传至SWR服务，请参考**训练作业自定义镜像规范**。
- 训练脚本已上传至OBS目录。
- 已在OBS创建至少1个空的文件夹，用于存储训练输出的内容。
- 由于训练作业运行需消耗资源，确保帐户未欠费。
- 确保您使用的OBS目录与ModelArts在同一区域。

注意事项

- 训练作业指定的数据集目录中，用于训练的数据名称（如图片名称、音频文件名、标注文件名称等），名称长度限制为0～255英文字符。如果数据集目录下，部分数据的文件名称超过255英文字符，训练作业将不会使用此数据，使用符合要求的数据继续进行训练。如果数据集目录下，所有数据的文件名称都超过了255英文字符，导致训练作业无数据可用，则会最终导致训练作业失败。
- 训练脚本中，“数据来源”、“训练输出位置”，两个参数必须为OBS路径。当需要对路径中进行读写交互时，建议使用**MoXing接口**进行读写操作。

创建训练作业

1. 登录ModelArts管理控制台，在左侧导航栏中选择“训练管理 > 训练作业”，默认进入“训练作业”列表。
2. 在训练作业列表中，单击左上角“创建”，进入“创建训练作业”页面。
3. 在创建训练作业页面，填写训练作业相关参数，然后单击“下一步”。
   a. 填写基本信息。
      您可以根据实际情况填写“名称”和“描述”信息。
   b. 填写作业参数。包含数据来源、算法来源等关键信息，详情请参见**表8-11**。

表 8-11 作业参数说明

参数名称	子参数	说明
一键式参数配置	-	如果您在ModelArts已保存作业参数，您可以根据界面提示，选择已有的作业参数，快速完成训练作业的参数配置。

参数名称	子参数	说明
算法来源	自定义	自定义镜像的相关规范请参见训练作业自定义镜像规范。 ● “镜像地址”：镜像上传到SWR后生成的地址。 ● “代码目录”：训练代码文件存储的OBS路径。 ● “运行命令”：镜像启动后的运行命令，根据实际情况填写。
数据来源	数据集	从ModelArts数据管理中选择可用的数据集及其版本。 ● “选择数据集”：从右侧下拉框中选择ModelArts系统中已有的数据集。当ModelArts无可用数据集时，此下拉框为空。 ● “选择版本”：根据“选择数据集”指定的数据集选择其版本。
	数据存储位置	从OBS桶中选择训练数据。在“数据存储位置”右侧，单击“选择”，从弹出的对话框中，选择数据存储的OBS桶及其文件夹。
训练输出位置	-	选择训练结果的存储位置。 **说明** 为避免出现错误，建议选择一个空目录用作“训练输出位置”。请勿将数据集存储的目录作为训练输出位置。
环境变量	-	请根据您的镜像文件，添加环境变量，此参数为可选。单击“增加环境变量”可增加多个变量参数。
作业日志路径	-	选择作业运行中产生的日志文件存储路径。

c. 选择用于训练作业的资源。

表 8-12 资源参数说明

参数名称	说明
资源池	选择训练作业资源池。 训练作业支持选择“公共资源池”和“专属资源池”。公共资源池又可以选择CPU或GPU两种类型，不同类型的资源池，其收费标准不同，详情请参见价格详情说明。专属资源池的创建请参见资源池。

参数名称	说明
类型	当选择公共资源池时，选择资源类型。目前支持CPU和GPU两种类型。 GPU资源性能更佳，CPU资源性价比更高。如果您选择的算法已定义使用CPU或GPU，界面将自动呈现此资源类型，请务必根据要求选择。 不同的资源类型的数据盘容量是不同的，详细介绍参考**训练环境中不同规格资源“/cache”目录的大小**。 **说明** • 如果您在训练代码使用的是GPU资源，则在选择资源池时只能选择GPU集群，否则会导致训练作业失败。
规格	针对不同的资源类型，选择资源规格。 其中针对“已有算法”、“常用框架”或“自定义镜像”，ModelArts支持使用Ascend 910创建训练作业，且此资源仅在“华北-北京四”可用。 资源规格中，前缀带“[限时免费]”信息的为免费规格，您可以使用此规格免费体验ModelArts的训练作业功能。此规格的使用注意事项，请参见**免费体验AI全流程开发**。
计算节点个数	选择计算节点的个数。如果节点个数设置为1，表示后台的计算模式是单机模式；如果节点个数设置大于1，表示后台的计算模式为分布式的。请根据实际编码情况选择计算模式。

d. 配置订阅消息，并设置是否将当前训练作业中的参数保存为作业参数。

图 8-10 配置训练作业订阅消息

表 8-13 订阅消息及作业参数参数说明

参数名称	说明
订阅消息	订阅消息使用消息通知服务，在事件列表中选择需要监控的资源池状态，在事件发生时发送消息通知。 此参数为可选参数，您可以根据实际情况设置是否打开开关。如果开启订阅消息，请根据实际情况填写如下参数。 ● “主题名”：订阅消息主题名称。您可以单击创建主题，在消息通知服务中创建主题。 ● “事件列表”：订阅事件。当前可选择“OnJobRunning”、“OnJobSucceeded”、“OnJobFailed”三种事件，分别代表训练运行中、运行成功、运行失败。
保存作业参数	勾选此参数，表示将当前训练作业设置的作业参数保存，方便后续一键复制使用。 勾选“保存训练参数”，然后填写“作业参数名称”和“作业参数描述”，即可完成当前参数配置的保存。训练作业创建成功后，您可以从ModelArts的作业参数列表中查看保存的信息，详细操作指导请参见**管理作业参数**。

e. 完成参数填写后，单击“下一步”。

4. 在“规格确认”页面，确认填写信息无误后，单击“提交”，完成训练作业的创建。训练作业一般需要运行一段时间，根据您选择的数据量和资源不同，训练时间将耗时几分钟到几十分钟不等。

   完成自定义镜像作业创建后，系统默认授权ModelArts获取镜像运行。在第一次运行自定义镜像作业时，ModelArts进行自定义镜像审核，审核内容请参见**训练作业自定义镜像规范**，审核失败的原因可在日志查看，用户根据日志做相应的修改。

   图 8-11 审核镜像失败

   镜像审核成功后，后台就会开始启动用户自定义镜像容器，跑自定义镜像训练作业。您可以前往训练作业列表，查看训练作业的基本情况。在训练作业列表中，刚创建的训练作业“状态”为“初始化”，当训练作业的“状态”变为“运行成功”时，表示训练作业运行结束，其生成的模型将存储至对应的“训练输出位置”中。当训练作业的“状态”变为“运行失败”时，您可以单击训练作业的名称，进入详情页面，通过查看日志等手段处理问题。

图 8-12 运行日志

说明

- 训练作业创建完成后，将立即启动，运行过程中将按照您选择的资源按需计费。
- 审核成功后，再次使用相同镜像创建训练作业的时候，不会再次审核。
- 自定义镜像的默认用户必须为“uid”为“1101”的用户。

8.5 停止或删除作业

停止训练作业

在训练作业列表中，针对“运行中”的训练作业，您可以单击“操作”列的“停止”，停止正在运行中的训练作业。

训练作业停止后，ModelArts将停止计费。如果停止的训练作业已勾选“保存作业参数”，其设置的作业参数将继续保存至“作业参数管理”页面中。

运行结束的训练作业，如“运行成功”、“运行失败”的作业，不涉及“停止”操作。只有“运行中”的训练作业支持“停止”操作。

删除训练作业

当已有的训练作业不再使用时，您可以删除训练作业。

处于“运行中”、“运行成功”、“运行失败”、“已取消”、“部署中”状态的训练作业，您可以单击“操作”列的“删除”，删除对应的训练作业。

如果删除的训练作业已勾选“保存作业参数”，其设置的作业参数将继续保存至“作业参数管理”页面中。

8.6 管理训练作业版本

在模型构建过程中，您可能需要根据训练结果，不停的调整数据、训练参数或模型，以获得一个满意的模型。因此，ModelArts为了方便用户在调整内容后快速高效的训练模型，提供了管理训练作业版本的能力。每训练一次，生成一个版本，不同的作业版本之间，能快速进行对比，获得对比结果。

查看训练作业版本

1. 登录ModelArts管理控制台，在左侧导航栏中选择“训练管理 > 训练作业”，默认进入“训练作业”列表。
2. 在训练作业列表中，单击训练作业名称，进入训练作业的详情页面。

默认打开最近一个版本的基本信息。当版本较多时，您可以单击左上角“版本过滤”过滤某几个版本进行查看。单击版本左侧的小三角打开作业的详细信息。训练作业的详细信息说明请参见**训练作业详情**。

图 8-13 查看训练作业版本

版本对比

在“版本管理”页面中，针对当前训练作业的所有版本，或者使用过滤功能筛选后的版本，单击右侧“查看对比结果”，可查看训练版本之间的对比，包含“运行参数”、“F1值”、“召回率”、“精确率”、“准确率”。

说明

使用预置算法创建的训练作业，才会显示其对应的“F1值”、“召回率”、“精确率”、“准确率”。针对使用常用框架、或自定义镜像创建的训练作业，请在您的训练脚本代码中定义好这些参数的输出，暂不支持在界面中查看。

图 8-14 训练版本对比

训练版本对比

版本	运行参数	F1值	召回率	精确率	准确率
V0006	split_spec=train:0.8,eval:0.2 num_gpus=1 batch_size=32 eval_batch_size=32 learning_rate_strategy=0.002 evaluate_every_n_epochs=1 save_interval_secs=2000000 max_epoches=100 log_every_n_steps=10 save_summaries_steps=5	0.054877	0.998818	0.028213	0.944444
V0005	split_spec=train:0.8,eval:0.2 num_gpus=1 batch_size=32 eval_batch_size=32 learning_rate_strategy=0.002 evaluate_every_n_epochs=1 save_interval_secs=2000000 max_epoches=100 log_every_n_steps=10 save_summaries_steps=5	0.053715	0.998822	0.027599	0.944444

基于训练作业版本的快捷操作

在训练作业的版本管理页面，ModelArts提供了一些快捷操作的入口，方便您在模型训练结束后，快速进行下一步操作。

表 8-14 快捷操作说明

操作	说明
创建可视化作业	基于当前训练版本创建可视化作业，详细参见**管理可视化作业**。
创建模型	基于当前训练版本创建模型，详细参见**创建AI应用**。只有“运行成功”的训练作业，支持此操作。
修改训练作业	如果当前版本的训练结果不满足业务需求时，或者训练作业“运行失败”时，您可以单击“修改”，跳转至训练作业参数设置页面，训练作业的参数说明请参见“**创建训练作业**”。根据实际情况调整作业参数后，单击“确定”启动新版本的训练作业。
保存作业参数	将此版本的作业参数可保存为新的作业参数。单击“更多操作>保存作业参数”，进入“作业参数”页面，确认信息无误后的，单击确定完成操作。作业参数管理详情请参见**管理作业参数**。
停止	单击“更多操作>停止”可停止当前版本的训练作业。只有“运行中”的训练作业版本才支持停止操作。
删除	单击“更多操作>删除”可停止当前版本的训练作业。

图 8-15 快捷操作

8.7 查看作业详情

训练作业运行结束后，除了管理训练作业版本之外，您可以通过查看**训练作业详情**和**评估结果**，判断此训练作业是否满意。

训练作业详情

在ModelArts管理控制台，选择“训练管理 > 训练作业”，进入训练作业列表页面。在训练作业列表中，您可以单击作业名称，查看该作业的详情。

每个版本的训练作业，包含的信息如**表8-15**所示。

图 8-16 训练作业详情

表 8-15 训练作业详情

参数	说明
当前版本	训练作业版本，由系统自动定义，命名规则为V0001、V0002。
状态	训练作业的状态。
运行时间	训练作业的运行时长。
配置信息	指当前训练作业版本的参数详情。
日志	指当前训练作业版本的运行日志。如果您在参数中设置了“作业日志路径”，您可以在“日志”页签单击“下载”将存储在OBS桶中的日志下载到本地。
资源占用情况	指当前训练作业版本的资源使用情况，资源包括CPU、GPU、NPU和内存。

评估结果

针对使用ModelArts官方发布的预置算法创建训练作业时，其训练作业详情支持查看评估结果。如果您的训练脚本中按照ModelArts规范添加了相应的评估代码，在训练作业运行结束后，也可在作业详情页面查看评估结果，添加评估代码指导请参见**添加评估结果**。

说明

当前仅图像分类-ResNet_v1_50、物体检测-FasterRCNN_ResNet50、物体检测-EfficientDet预置算法支持模型评估结果查看功能。

当训练作业运行结束，且状态为“运行成功”时，进入作业详情页面，可在“评估结果”页签下查看详情，如**图8-17**所示。

评估结果包含“评估综述”、“精度评估”、“敏感度分析”等维度，包含了常用的模型指标。由于每个模型情况不同，系统将自动根据您的模型指标情况，给出一些调优建议，请仔细阅读界面中的建议和指导，对您的模型进行进一步的调优。

图 8-17 评估结果

8.8 管理作业参数

创建训练作业时，您可以将训练作业的参数保存在ModelArts中，再次创建训练作业时，可一键使用已存储的作业参数，使得训练作业的创建高效便捷。

在创建训练作业、编辑训练作业、查看训练作业等操作过程中，保存的作业参数都将存储在“作业参数管理”页面中。

使用作业参数

- 方式1：在“作业参数管理”页面使用

  登录ModelArts管理控制台，在左侧导航栏中选择“训练管理 > 训练作业”，然后单击“作业参数管理”页签。在已有的作业参数列表中，单击“创建训练”，可快速将此作业参数用于创建一个新的训练作业。

- 方式2：在创建训练作业页面使用

  在创建训练作业页面中，在“一键式参数配置参数”中，根据界面提示操作，选择需要使用的作业参数，快速创建一个可用的训练作业。

编辑作业参数

1. 登录ModelArts管理控制台，在左侧导航栏中选择“训练管理 > 训练作业”，然后单击“作业参数管理”页签。
2. 在作业参数列表中，单击“操作”列的“编辑”。
3. 在打开的作业参数页面，参见创建训练作业章节，修改相关参数，然后单击“确定”保存此作业参数。

   其中，作业参数的“名称”，不支持修改。

删除作业参数

1. 登录ModelArts管理控制台，在左侧导航栏中选择“训练管理 > 训练作业”，然后单击“作业参数管理”页签。
2. 在作业参数列表中，单击“操作”列的“删除”。
3. 确认弹出对话框的信息，单击“确定”，完成删除操作。

   说明

   作业参数删除后不可恢复，请谨慎操作。

8.9 添加评估结果

训练作业运行结束后，ModelArts将自动为您的模型进行评估，并且给出调优诊断和建议，详细功能描述请参见评估结果。

- 针对使用预置算法创建训练作业，无需任何配置，即可查看此评估结果。
- 针对用户自己编写训练脚本或自定义镜像方式创建的训练作业，则需要在您的训练代码中添加评估代码，才可以在训练作业结束后查看相应的评估诊断建议。

说明

1. 只支持验证集的数据格式为图片
2. 目前，仅如下常用框架的训练脚本支持添加评估代码。
   - TF-1.13.1-python3.6
   - TF-2.1.0-python3.6
   - PyTorch-1.4.0-python3.6

本章节介绍如何在训练中使用评估代码。对训练代码做一定的适配和修正，分为三个方面：**添加输出目录**、**拷贝数据集到本地**、**映射数据集路径到OBS**。

添加输出目录

添加输出目录的代码比较简单，即在代码中添加一个输出评估结果文件的目录，被称为train_url，也就是页面上的训练输出位置。并把train_url添加到使用的函数analysis中，使用save_path来获取train_url。示例代码如下所示：

```
FLAGS = tf.app.flags.FLAGS
tf.app.flags.DEFINE_string('model_url', '', 'path to saved model')
tf.app.flags.DEFINE_string('data_url', '', 'path to output files')
tf.app.flags.DEFINE_string('train_url', '', 'path to output files')
tf.app.flags.DEFINE_string('adv_param_json',
                           '{"attack_method":"FGSM","eps":40}',
                           'params for adversarial attacks')
FLAGS(sys.argv, known_only=True)

...

# analyse
res = analyse(
    task_type=task_type,
    pred_list=pred_list,
    label_list=label_list,
    name_list=file_name_list,
    label_map_dict=label_dict,
    save_path=FLAGS.train_url)
```

拷贝数据集到本地

拷贝数据集到本地主要是为了防止长时间访问OBS容易导致OBS连接中断使得作业卡住，所以一般先将数据拷贝到本地再进行操作。

数据集拷贝有两种方式，推荐使用OBS路径进行处理拷贝。

- OBS路径（推荐）

  直接使用moxing的copy_parallel接口，拷贝对应的OBS路径。

- ModelArts数据管理中的数据集（即manifest文件格式）

  使用moxing的copy_manifest接口将文件拷贝到本地并获取新的manifest文件路径，然后使用SDK解析新的manifest文件。

```
if data_path.startswith('obs://'):
    if '.manifest' in data_path:
        new_manifest_path, _ = mox.file.copy_manifest(data_path, '/cache/data/')
        data_path = new_manifest_path
    else:
        mox.file.copy_parallel(data_path, '/cache/data/')
        data_path = '/cache/data/'
    print('------------- download dataset success ------------')
```

映射数据集路径到 OBS

由于最终JSON体中需要填写的是图片文件的真实路径，也就是OBS对应的路径，所以在拷贝到本地做完分析和评估操作后，需要将原本的本地数据集路径映射到OBS路径，然后将新的list送入analysis接口。

如果使用的是OBS路径作为输入的data_url，则只需要替换本地路径的字符串即可。

```
if FLAGS.data_url.startswith('obs://'):
    for idx, item in enumerate(file_name_list):
        file_name_list[idx] = item.replace(data_path, FLAGS.data_url)
```

如果使用manifest文件，需要再解析一遍原版的manifest文件获取list，然后再送入analysis接口。

```
if or FLAGS.data_url.startswith('obs://'):
    if 'manifest' in FLAGS.data_url:
            file_name_list = []
            manifest, _ = get_sample_list(
                manifest_path=FLAGS.data_url, task_type='image_classification')
            for item in manifest:
                if len(item[1]) != 0:
                    file_name_list.append(item[0])
```

完整的适配了训练作业创建的图像分类样例代码如下：

```
import json
import logging
import os
import sys
import tempfile

import h5py
import numpy as np
from PIL import Image

import moxing as mox
import tensorflow as tf
from deep_moxing.framework.manifest_api.manifest_api import get_sample_list
from deep_moxing.model_analysis.api import analyse, tmp_save
from deep_moxing.model_analysis.common.constant import TMP_FILE_NAME

logging.basicConfig(level=logging.DEBUG)

FLAGS = tf.app.flags.FLAGS
tf.app.flags.DEFINE_string('model_url', '', 'path to saved model')
tf.app.flags.DEFINE_string('data_url', '', 'path to output files')
tf.app.flags.DEFINE_string('train_url', '', 'path to output files')
tf.app.flags.DEFINE_string('adv_param_json',
                           '{"attack_method":"FGSM","eps":40}',
                           'params for adversarial attacks')
FLAGS(sys.argv, known_only=True)


def _preprocess(data_path):
    img = Image.open(data_path)
    img = img.convert('RGB')
    img = np.asarray(img, dtype=np.float32)
    img = img[np.newaxis, :, :, :]
    return img


def softmax(x):
    x = np.array(x)
    orig_shape = x.shape
    if len(x.shape) > 1:
        # Matrix
        x = np.apply_along_axis(lambda x: np.exp(x - np.max(x)), 1, x)
```

```
        denominator = np.apply_along_axis(lambda x: 1.0 / np.sum(x), 1, x)
        if len(denominator.shape) == 1:
            denominator = denominator.reshape((denominator.shape[0], 1))
        x = x * denominator
    else:
        # Vector
        x_max = np.max(x)
        x = x - x_max
        numerator = np.exp(x)
        denominator = 1.0 / np.sum(numerator)
        x = numerator.dot(denominator)
    assert x.shape == orig_shape
    return x


def get_dataset(data_path, label_map_dict):
    label_list = []
    img_name_list = []
    if 'manifest' in data_path:
        manifest, _ = get_sample_list(
            manifest_path=data_path, task_type='image_classification')
        for item in manifest:
            if len(item[1]) != 0:
                label_list.append(label_map_dict.get(item[1][0]))
                img_name_list.append(item[0])
            else:
                continue
    else:
        label_name_list = os.listdir(data_path)
        label_dict = {}
        for idx, item in enumerate(label_name_list):
            label_dict[str(idx)] = item
            sub_img_list = os.listdir(os.path.join(data_path, item))
            img_name_list += [
                os.path.join(data_path, item, img_name) for img_name in sub_img_list
            ]
            label_list += [label_map_dict.get(item)] * len(sub_img_list)
    return img_name_list, label_list


def deal_ckpt_and_data_with_obs():
    pb_dir = FLAGS.model_url
    data_path = FLAGS.data_url

    if pb_dir.startswith('obs://'):
        mox.file.copy_parallel(pb_dir, '/cache/ckpt/')
        pb_dir = '/cache/ckpt'
        print('------------- download success -------------')
    if data_path.startswith('obs://'):
        if '.manifest' in data_path:
            new_manifest_path, _ = mox.file.copy_manifest(data_path, '/cache/data/')
            data_path = new_manifest_path
        else:
            mox.file.copy_parallel(data_path, '/cache/data/')
            data_path = '/cache/data/'
        print('------------- download dataset success -------------')
    assert os.path.isdir(pb_dir), 'Error, pb_dir must be a directory'
    return pb_dir, data_path


def evalution():
    pb_dir, data_path = deal_ckpt_and_data_with_obs()
    index_file = os.path.join(pb_dir, 'index')
    try:
        label_file = h5py.File(index_file, 'r')
        label_array = label_file['labels_list'][:].tolist()
        label_array = [item.decode('utf-8') for item in label_array]
    except Exception as e:
        logging.warning(e)
```

```
        logging.warning('index file is not a h5 file, try json.')
        with open(index_file, 'r') as load_f:
            label_file = json.load(load_f)
        label_array = label_file['labels_list'][:]
    label_map_dict = {}
    label_dict = {}
    for idx, item in enumerate(label_array):
        label_map_dict[item] = idx
        label_dict[idx] = item
    print(label_map_dict)
    print(label_dict)

    data_file_list, label_list = get_dataset(data_path, label_map_dict)

    assert len(label_list) > 0, 'missing valid data'
    assert None not in label_list, 'dataset and model not match'

    pred_list = []
    file_name_list = []
    img_list = []

    for img_path in data_file_list:
        img = _preprocess(img_path)
        img_list.append(img)
        file_name_list.append(img_path)

    config = tf.ConfigProto()
    config.gpu_options.allow_growth = True
    config.gpu_options.visible_device_list = '0'
    with tf.Session(graph=tf.Graph(), config=config) as sess:
        meta_graph_def = tf.saved_model.loader.load(
            sess, [tf.saved_model.tag_constants.SERVING], pb_dir)
        signature = meta_graph_def.signature_def
        signature_key = 'predict_object'
        input_key = 'images'
        output_key = 'logits'
        x_tensor_name = signature[signature_key].inputs[input_key].name
        y_tensor_name = signature[signature_key].outputs[output_key].name
        x = sess.graph.get_tensor_by_name(x_tensor_name)
        y = sess.graph.get_tensor_by_name(y_tensor_name)
        for img in img_list:
            pred_output = sess.run([y], {x: img})
            pred_output = softmax(pred_output[0])
            pred_list.append(pred_output[0].tolist())

    label_dict = json.dumps(label_dict)
    task_type = 'image_classification'

    if FLAGS.data_url.startswith('obs://'):
        if 'manifest' in FLAGS.data_url:
            file_name_list = []
            manifest, _ = get_sample_list(
                manifest_path=FLAGS.data_url, task_type='image_classification')
            for item in manifest:
                if len(item[1]) != 0:
                    file_name_list.append(item[0])
        for idx, item in enumerate(file_name_list):
            file_name_list[idx] = item.replace(data_path, FLAGS.data_url)
    # analyse
    res = analyse(
        task_type=task_type,
        pred_list=pred_list,
        label_list=label_list,
        name_list=file_name_list,
        label_map_dict=label_dict,
        save_path=FLAGS.train_url)

if __name__ == "__main__":
    evalution()
```

8.10 管理可视化作业

当前，您管理的ModelArts可视化作业支持创建TensorBoard类型和MindInsight两种类型。

TensorBoard和MindInsight能够有效地展示训练作业在运行过程中的变化趋势以及训练中使用到的数据信息。

- TensorBoard

  TensorBoard是一个可视化工具，能够有效地展示TensorFlow在运行过程中的计算图、各种指标随着时间的变化趋势以及训练中使用到的数据信息。TensorBoard当前只支持基于TensorFlow和MXNet引擎的训练作业。TensorBoard相关概念请参考**TensorBoard官网**。

- MindInsight

  MindInsight能可视化展现出训练过程中的标量、图像、计算图以及模型超参等信息，同时提供训练看板、模型溯源、数据溯源、性能调试等功能，帮助您在更高效地训练调试模型。MindInsight当前支持基于MindSpore引擎的训练作业。MindInsight相关概念请参考**MindSpore官网**。

您可以使用模型训练时产生的Summary文件来创建可视化作业。

前提条件

为了保证训练结果中输出Summary文件，在编写训练脚本时，您需要在代码中添加Summary相关代码。

- 使用TensorFlow引擎编写程序时

  使用基于TensorFlow的MoXing时，需要将“mox.run”中设置参数“save_summary_steps>0”，并且超参“summary_verbosity≥1”。

  如果您想显示其他指标，可以在“model_fn”的返回值类型“mox.ModelSpec”的“log_info”中添加张量（仅支持0阶张量，即标量），添加的张量会被写入到Summary文件中。如果您希望在Summary文件中写入更高阶的张量，只需要在“model_fn”中使用TensorFlow原生的“tf.summary”的方式添加即可。

- 使用MindSpore引擎编写程序时

  MindSpore支持将数据信息保存到Summary日志文件中，并通过可视化界面进行展示。将数据记录到Summary日志文件中的具体方式请参考**收集Summary数据**。

- 使用MXNet引擎编写程序时

  需要在代码里添加Summary相关代码，代码内容如下所示：

```
batch_end_callbacks.append(mx.contrib.tensorboard.LogMetricsCallback('OBS路径'))
```

注意事项

- 运行中的可视化作业会一直按需计费，当您不需要使用时，建议停止可视化作业，避免产生不必要的费用。可视化作业支持自动停止功能，即在指定时间后停止可视化作业，为避免产生不必要的费用，推荐启用此功能。
- 默认使用CPU资源运行可视化作业，且不支持修改为其他资源池。
- 确保您使用的OBS目录与ModelArts在同一区域。

创建可视化作业

1. 登录ModelArts管理控制台，在左侧导航栏中选择“训练作业”，然后单击“可视化作业”页签。
2. 在可视化作业列表中，单击左上方“创建”，进入“创建可视化作业”界面。
3. 其中，“计费模式”设置为“按需计费”，“作业类型”为“TensorBoard”和“MindInsight”两种类型。请根据实际情况填写可视化作业“名称”、“描述”，设置“训练输出位置”和“自动停止”参数。
   - “训练输出位置”：选择创建训练作业时的“训练输出位置”。
   - “自动停止”：设置是否开启自动停止功能。由于运行中的可视化作业会一直计费，为避免产生不必要的费用，您可以开启自动停止功能，在指定时间后自动停止可视化作业。目前支持设置为“1小时后”、“2小时后”、“4小时后”、“6小时后”、“自定义”。如果选择“自定义”的模式，可在右侧输入框中输入1~24范围内的任意整数。

图 8-18 创建可视化作业

4. 参数填写完成后，单击“下一步”进行规格确认。
5. 规格确认无误后，单击“立即创建”，完成可视化作业的创建。

   在可视化作业列表中，当状态变为“运行中”时，表示可视化作业已创建完成。您可以单击可视化作业名称进入查看详情。

打开可视化作业

在可视化作业列表中，单击可视化作业名称，即可打开可视化显示界面。只有“运行中”状态的可视化作业支持打开。

图 8-19 TensorBoard 界面

图 8-20 MindInsight 界面

运行或停止可视化作业

- **停止可视化作业**：由于“运行中”的可视化作业将一直按需计费，在不需要使用时，您可以停止可视化作业停止计费。在可视化作业列表中，单击“操作”列的“停止”，即可停止可视化作业。
- **运行可视化作业**：对于“已取消”状态的可视化作业，您可以重新运行并使用可视化作业。在可视化作业列表中，单击“操作”列的“运行”，即可运行可视化作业。

删除可视化作业

如果您的可视化作业不再使用，您可以删除可视化作业释放资源。在可视化作业列表中，单击“操作”列的“删除”，即可删除可视化作业。

说明

可视化作业删除后不可恢复，需重新创建可视化作业。请谨慎操作。

9 AI 应用管理（即将下线）

9.1 AI 应用管理简介

说明

为了更方便用户使用和清晰的获取文档帮助，《AI工程师用户指南》将按照AI开发流程调整拆分为多本文档，在一级导航中呈现，并从用户使用场景对各模块文档进行了优化和改进。“模型管理”的内容已拆分至《**推理部署**》,《AI工程师用户指南》里的“模型管理”即将下线。

AI开发和调优往往需要大量的迭代和调试，数据集、训练代码或参数的变化都可能会影响模型的质量，如不能统一管理开发流程元数据，可能会出现无法重现最优模型的现象。

ModelArtsAI应用管理可导入所有训练生成的元模型，可对所有迭代和调试的AI应用进行统一管理。

使用限制和说明

- 自动学习项目中，在完成模型部署后，其生成的模型也将自动上传至AI应用管理列表中。但是自动学习生成的AI应用无法下载，只能用于部署上线。
- 创建AI应用、管理AI应用版本、模型转换等功能目前是免费开放给所有用户，使用此功能不会产生费用。

创建 AI 应用的几种场景

- **从训练中选择**：在ModelArts中创建训练作业，并完成模型训练，在得到满意的模型后，可以将训练后得到的模型创建为AI应用，用于部署服务。
- **从模板中选择**：相同功能的模型配置信息重复率高，将相同功能的配置整合成一个通用的模板，通过使用该模板，可以方便快捷的导入模型，创建为AI应用，而不用编写config.json配置文件。
- **从容器镜像中选择**：针对ModelArts目前不支持的AI引擎，可以通过自定义镜像的方式将编写的模型镜像导入ModelArts，创建为AI应用，用于部署服务。
- **从OBS中选择**：如果您使用常用框架在本地完成模型开发和训练，可以将模型导入至ModelArts中，创建为AI应用，直接用于部署服务。

AI 应用管理的功能描述

表 9-1 AI 应用管理相关功能

支持的功能	说明
创建AI应用	将训练后的模型导入至ModelArts创建为AI应用，便于进行统一管理，支持如下几种场景的导入方式，不同场景对应的操作指导请参见： • 从训练中选择元模型 • 从模板中选择元模型 • 从容器镜像中选择元模型 • 从OBS中选择元模型
管理AI应用版本	为方便溯源和模型反复调优，在ModelArts中提供了AI应用版本管理的功能，您可以基于版本对AI应用进行管理。
发布AI应用	针对在ModelArts创建的AI应用，支持将参赛模型提交至比赛项目。

9.2 创建 AI 应用

9.2.1 从训练中选择元模型

在ModelArts中创建训练作业，并完成模型训练，在得到满意的模型后，可以将训练后得到的模型导入至模型管理，方便统一管理，同时支持将模型快速部署上线为服务。

背景信息

- 如果使用ModelArts训练作业生成的模型，请确保训练作业已运行成功，且模型已存储至训练输出的OBS目录下（输入参数为 train_url）。
- 针对使用订阅算法的训练作业，无需推理代码和配置文件，其生成的模型可直接导入ModelArts。
- 针对使用常用框架或自定义镜像创建的训练作业，需根据模型包规范介绍，将推理代码和配置文件上传至模型的存储目录中。
- 确保您使用的OBS目录与ModelArts在同一区域。

创建 AI 应用操作步骤

1. 登录ModelArts管理控制台，在左侧导航栏中选择“AI应用管理 > AI应用”，进入AI应用列表页面。
2. 单击左上角的“创建”，进入“创建AI应用”页面。
3. 在“创建AI应用”页面，填写相关参数。
   a. 填写AI应用基本信息，详细参数说明请参见表9-2。

表 9-2 AI 应用基本信息参数说明

参数名称	说明
名称	AI应用名称。支持1～64位可见字符（含中文），名称可以包含字母、中文、数字、中划线、下划线。
版本	设置所创建AI应用的版本。第一次导入时，默认为0.0.1。
标签	AI应用标签，最多支持5个。
描述	AI应用的简要描述。

b. 填写元模型来源及其相关参数。当“元模型来源”选择“从训练中选择”时，其相关的参数配置请参见**表9-3**。

图 9-1 从训练中选择元模型

表 9-3 元模型来源参数说明

参数	说明
“元模型来源”	选择“从训练中选择>训练作业”或者“从训练中选择>训练作业（New）”。 ● 在“选择训练作业”右侧下拉框中选择当前帐号下已完成运行的训练作业及其“版本”。 ● “动态加载”：若勾选动态加载，则模型文件和运行时依赖仅在实际部署时拉取，用于实现快速部署和快速更新模型。 **说明** 当前ModelArts同时存在新版训练管理和旧版训练管理功能。旧版训练管理功能仅对部分存量用户可见，新用户不可见。
“AI引擎”	元模型使用的推理引擎，选择训练作业后会自动匹配。
“推理代码”	推理代码自定义AI应用的推理处理逻辑。显示推理代码URL，您可以直接复制此URL使用。

参数	说明
“运行时依赖”	罗列选中模型对环境的依赖。例如依赖“tensorflow”，安装方式为“pip”，其版本必须为1.8.0及以上版本。
“AI应用说明”	为了帮助其他AI应用开发者更好的理解及使用您的AI应用，建议您提供AI应用的说明文档。单击“添加AI应用说明”，设置“文档名称”及其“URL”。AI应用说明最多支持3条。
“部署类型”	选择此AI应用支持部署服务的类型，部署上线时只支持部署为此处选择的部署类型，例如此处只选择在线服务，那您导入后只能部署为在线服务。当前支持“在线服务”、“批量服务”和“边缘服务”。

c. 确认信息填写无误，单击“立即创建”，完成AI应用的创建。

在AI应用列表中，您可以查看刚创建的AI应用及其对应的版本。当AI应用状态变更为“正常”时，表示AI应用导入成功。在此页面，您还可以创建新版本、快速部署服务、发布AI应用等操作。

后续操作

部署服务：在“AI应用列表”中，单击AI应用名称左侧的小三角，打开此AI应用下的所有版本。在对应版本所在行，单击“操作”列的“部署”，在下拉框中选择部署类型，可以将AI应用部署上线为创建AI应用时所选择的部署类型。

9.2.2 从模板中选择元模型

相同功能的模型配置信息重复率高，将相同功能的配置整合成一个通用的模板，通过使用该模板，可以方便快捷的创建AI应用，而不用编写config.json配置文件。

背景信息

- 由于相同功能的模型配置信息重复率高，ModelArts将相同功能的配置整合成一个通用的模板，用户通过使用该模板，可以方便快捷的创建AI应用。模板的详细说明请参见模型模板简介。
- 目前支持的模板请参见支持的模板，各模板相应的输入输出模式说明，请参见支持的输入输出模式。
- 确保您已按照相应模板的模型包规范要求将模型上传至OBS。
- 确保您使用的OBS与ModelArts在同一区域。
- 创建和管理AI应用是免费的，不会产生费用。

创建 AI 应用操作步骤

1. 登录ModelArts管理控制台，在左侧导航栏中选择“AI应用管理 > AI应用”，进入AI应用列表页面。
2. 单击左上角的“创建”，进入“创建AI应用”页面。
3. 在“创建AI应用”页面，填写相关参数。
   a. 填写AI应用基本信息，详细参数说明请参见表9-4。

表 9-4 AI 应用基本信息参数说明

参数名称	说明
名称	AI应用名称。支持1～64位可见字符（含中文），名称可以包含字母、中文、数字、中划线、下划线。
版本	设置所创建AI应用的版本。第一次导入时，默认为0.0.1。
标签	AI应用标签，最多支持5个。
描述	AI应用的简要描述。

b. 填写元模型来源及其相关参数。当“元模型来源”选择“从模板中选择”时，其相关的参数配置请参见表9-5。

图 9-2 从模板中选择元模型

表 9-5 元模型来源参数说明

参数	说明
“模型模板”	从已有的ModelArts模板列表中选择模板。例如，“TensorFlow图像分类模板”。 ModelArts还提供“类型”、“引擎”、“环境”三个筛选条件，帮助您更快找到想要的模板。如果这个三个筛选条件不能满足您的要求，可以使用关键词搜索，找到目标模板。支持的模板请参见支持的模板。

参数	说明
“模型目录”	指定模型存储的OBS路径。请根据您选择的“模型模板”，按照模板的模型输入要求，选择对应的模型存储的OBS路径。 OBS路径不能含有空格，否则创建AI应用会失败。 **说明** ● 选择加密桶或者加密文件，会导入失败。 ● 当训练作业执行多次时，将基于V001、V002等规则生成不同的版本目录，且生成的模型将存储在不同版本目录下的model文件夹。此处选择模型文件时，需指定对应版本目录下的model文件夹。
“动态加载”	若勾选动态加载，则模型文件和运行时依赖仅在实际部署时拉取，用于实现快速部署和快速更新模型。
“输入输出模式”	如果您选择的模板允许覆盖其中的默认输入输出模式，您可以根据AI应用功能或业务场景在“输入输出模式”中，选择相应的输入输出模式。“输入输出模式”是对“config.json”中API（apis）的抽象，描述AI应用对外提供推理的接口。一个“输入输出模式”描述一个或多个API接口，每个模板对应于一个“输入输出模式”。 例如，“TensorFlow图像分类模板”，其支持的“输入输出模式”为“预置图像处理模式”，但该模板不允许修改其中的输入输出模式，所以您在页面上只能看到模板默认的输入输出模式，而不能选择其他模式。 支持的输入输出模式请参见**支持的输入输出模式**。
“AI应用说明”	为了帮助其他AI应用开发者更好的理解及使用您的AI应用，建议您提供AI应用的说明文档。单击“添加AI应用说明”，设置“文档名称”及其“URL”。AI应用说明支持增加3条。
“部署类型”	选择此AI应用支持部署服务的类型，部署上线时只支持部署为此处选择的部署类型，例如此处只选择在线服务导入后只能部署为在线服务。当前支持“在线服务”、“批量服务”和“边缘服务”。

c. 确认信息填写无误，单击“立即创建”，完成AI应用导入。

在AI应用列表中，您可以查看刚创建的AI应用及其对应的版本。当AI应用状态变更为“正常”时，表示AI应用创建成功。在此页面，您还可以创建新版本、快速部署服务、发布AI应用等操作。

后续操作

部署服务：在“AI应用列表”中，单击AI应用名称左侧的小三角，打开此AI应用下的所有版本。在对应版本所在行，单击“操作”列的“部署”，在下拉框中选择部署类型，可以将AI应用部署上线为创建AI应用时所选择的部署类型。

9.2.3 从容器镜像中选择元模型

针对ModelArts目前不支持的AI引擎，您可以通过自定义镜像的方式将编写的模型导入ModelArts。

使用前必读

- 关于自定义镜像规范和说明，请参见**模型镜像规范**。
- 确保您使用的OBS目录与ModelArts在同一区域。

创建 AI 应用操作步骤

1. 登录ModelArts管理控制台，在左侧导航栏中选择“AI应用管理 > AI应用”，进入AI应用列表页面。
2. 单击左上角的“创建”，进入“创建AI应用”页面。
3. 在“创建AI应用”页面，填写相关参数。
   a. 填写AI应用基本信息，详细参数说明请参见**表9-6**。

   表 9-6 AI 应用基本信息参数说明

参数名称	说明
名称	AI应用名称。支持1～64位可见字符（含中文），名称可以包含字母、中文、数字、中划线、下划线。
版本	设置所创建AI应用的版本。第一次导入时，默认为0.0.1。
标签	AI应用标签，最多支持5个。
描述	AI应用的简要描述。

   b. 填写元模型来源及其相关参数。当“元模型来源”选择“从容器镜像中选择”时，其相关的参数配置请参见**表9-7**。

   图 9-3 从容器镜像中选择 AI 应用

表 9-7 元模型来源参数说明

参数	说明
“容器镜像所在的路径”	单击 从容器镜像中导入模型的镜像，其中，模型均为Image类型，且不再需要用配置文件中的“swr_location”来指定您的镜像位置。 制作自定义镜像的操作指导及规范要求，请参见模型镜像规范。 **说明** 您选择的模型镜像将共享给管理员，请确保具备共享该镜像的权限（不支持导入其他帐户共享给您的镜像），部署上线时，ModelArts将使用该镜像部署成推理服务，请确保您的镜像能正常启动并提供推理接口。
“容器调用接口”	可选参数，用于指定模型启动的协议和端口号。
“健康检查”	可选参数，用于指定模型的健康检查。仅当自定义镜像中配置了健康检查接口，才能配置“健康检查”，否则会导致AI应用创建失败。 ● **健康检查URL**：填写健康检查的URL。 ● **健康检查周期**：填写大于0的整数，单位为秒。 ● **健康检查最大失败次数**：填写大于0的整数。在服务启动阶段，当健康检查请求连续失败达到所填次数后，服务会进入异常状态；在服务运行阶段，当健康检查请求连续失败达到所填次数后，服务会进入告警状态。 **说明** 当AI应用配置了健康检查，部署的服务在收到停止指令后，会延后3分钟才停止。
“AI应用说明”	为了帮助其他AI应用开发者更好的理解及使用您的AI应用，建议您提供AI应用的说明文档。单击“添加AI应用说明”，设置“文档名称”及其“URL”。AI应用说明支持增加3条。
“部署类型”	选择此AI应用支持部署服务的类型，部署上线时只支持部署为此处选择的部署类型，例如此处只选择在线服务，那您导入后只能部署为在线服务。当前支持“在线服务”、“批量服务”和“边缘服务”。
“apis定义”	提供AI应用对外Restfull api数据定义，用于定义AI应用的输入、输出格式。

c. 确认信息填写无误，单击“立即创建”，完成AI应用创建。

在AI应用列表中，您可以查看刚创建的AI应用及其对应的版本。当AI应用状态变更为“正常”时，表示AI应用创建成功。在此页面，您还可以进行创建新版本、快速部署服务、发布AI应用等操作。

后续操作

部署服务：在“AI应用列表”中，单击AI应用名称左侧的小三角，打开此AI应用下的所有版本。在对应版本所在行，单击“操作”列的“部署”，在下拉框中选择部署类型，可以将AI应用部署上线为创建AI应用时所选择的部署类型。

9.2.4 从 OBS 中选择元模型

针对使用常用框架完成模型开发和训练的场景，可以将您的模型导入至ModelArts中，创建为AI应用，并进行统一管理。

使用前必读

- 已完成模型开发和训练，使用的AI引擎为ModelArts支持的类型和版本。ModelArts支持如下常用引擎及其支持的Runtime范围。

表 9-8 支持的常用引擎及其 Runtime

模型使用的引擎类型	支持的运行环境（Runtime）	注意事项
TensorFlow	python3.6 python2.7 tf1.13-python2.7-gpu tf1.13-python2.7-cpu tf1.13-python3.6-gpu tf1.13-python3.6-cpu tf1.13-python3.7-cpu tf1.13-python3.7-gpu tf2.1-python3.7	• python2.7、python3.6的运行环境搭载的TensorFlow版本为1.8.0。 • python3.6、python2.7、tf2.1-python3.7，表示该模型可同时在CPU或GPU运行。其他Runtime的值，如果后缀带cpu或gpu，表示该模型仅支持在CPU或GPU中运行。 • 默认使用的Runtime为python2.7。
MXNet	python3.7 python3.6 python2.7	• python2.7、python3.6、python3.7的运行环境搭载的MXNet版本为1.2.1。 • python2.7、python3.6、python3.7，表示该模型可同时在CPU或GPU运行。 • 默认使用的Runtime为python2.7。

模型使用的引擎类型	支持的运行环境（Runtime）	注意事项
Caffe	python2.7 python3.6 python3.7 python2.7-gpu python3.6-gpu python3.7-gpu python2.7-cpu python3.6-cpu python3.7-cpu	● python2.7、python3.6、python3.7、python2.7-gpu、python3.6-gpu、python3.7-gpu、python2.7-cpu、python3.6-cpu、python3.7-cpu的运行环境搭载的Caffe版本为1.0.0。 ● python2.7、python3.6、python3.7只能用于运行适用于CPU的模型。其他Runtime的值，如果后缀带cpu或gpu，表示该模型仅支持在CPU或GPU中运行。推荐使用python2.7-gpu、python3.6-gpu、python3.7-gpu、python2.7-cpu、python3.6-cpu、python3.7-cpu的Runtime。 ● 默认使用的Runtime为python2.7。
Spark_MLlib	python2.7 python3.6	● python2.7以及python3.6的运行环境搭载的Spark_MLlib版本为2.3.2。 ● 默认使用的Runtime为python2.7。 ● python2.7、python3.6只能用于运行适用于CPU的模型。
Scikit_Learn	python2.7 python3.6	● python2.7以及python3.6的运行环境搭载的Scikit_Learn版本为0.18.1。 ● 默认使用的Runtime为python2.7。 ● python2.7、python3.6只能用于运行适用于CPU的模型。
XGBoost	python2.7 python3.6	● python2.7以及python3.6的运行环境搭载的XGBoost版本为0.80。 ● 默认使用的Runtime为python2.7。 ● python2.7、python3.6只能用于运行适用于CPU的模型。

模型使用的引擎类型	支持的运行环境（Runtime）	注意事项
PyTorch	python2.7 python3.6 python3.7 pytorch1.4-python3.7 pytorch1.5-python3.7	• python2.7、python3.6、python3.7的运行环境搭载的PyTorch版本为1.0。 • python2.7、python3.6、python3.7、pytorch1.4-python3.7、pytorch1.5-python3.7，表示该模型可同时在CPU或GPU运行。 • 默认使用的Runtime为python2.7。

- 针对创建AI应用的模型，需符合ModelArts的模型包规范，推理代码和配置文件也需遵循ModelArts的要求，详细说明请参见**模型包规范介绍**、**模型配置文件编写说明**、**模型推理代码编写说明**。
- 已完成训练的模型包，及其对应的推理代码和配置文件，且已上传至OBS目录中。
- 确保您使用的OBS与ModelArts在同一区域。

创建 AI 应用操作步骤

1. 登录ModelArts管理控制台，在左侧导航栏中选择“AI应用管理 > AI应用”，进入AI应用列表页面。
2. 单击左上角的“创建”，进入“创建AI应用”页面。
3. 在“创建AI应用”页面，填写相关参数。
   a. 填写AI应用基本信息，详细参数说明请参见**表9-9**。

   表 9-9 AI 应用基本信息参数说明

参数名称	说明
名称	AI应用名称。支持1～64位可见字符（含中文），名称可以包含字母、中文、数字、中划线、下划线。
版本	设置所创建AI应用的版本。第一次导入时，默认为0.0.1。
标签	AI应用标签，最多支持5个。
描述	AI应用的简要描述。

   b. 填写元模型来源及其相关参数。当“元模型来源”选择“从对象存储（OBS）中选择”时，其相关的参数配置请参见**表9-10**。

   针对从OBS导入的元模型，ModelArts要求根据**模型包规范**，编写推理代码和配置文件，并将推理代码和配置文件放置元模型存储的“model”文件夹下。如果您选择的目录下不符合模型包规范，将无法创建AI应用。

图 9-4 从 OBS 中选择元模型

表 9-10 元模型来源参数说明

参数	说明
“选择元模型”	选择元模型存储的OBS路径。 OBS路径不能含有空格，否则创建AI应用会失败。
“AI引擎”	根据您选择的元模型存储路径，将自动关联出元模型使用的“AI引擎”。
“动态加载”	若勾选动态加载，则模型文件和运行时依赖仅在实际部署时拉取，用于实现快速部署和快速更新模型。
“运行时依赖”	罗列选中模型对环境的依赖。例如依赖“tensorflow”，安装方式为“pip”，其版本必须为1.8.0及以上版本。
“AI应用说明”	为了帮助其他AI应用开发者更好的理解及使用您的AI应用，建议您提供AI应用的说明文档。单击“添加AI应用说明”，设置“文档名称”及其“URL”。AI应用说明支持增加3条。
“配置文件”	系统默认关联您存储在OBS中的配置文件。打开开关，您可以直接在当前界面查看或编辑模型配置文件。 **说明** 该功能即将下线，后续请根据“AI引擎”、“运行时依赖”和“apis定义”修改模型的配置信息。
“部署类型”	选择此AI应用支持部署服务的类型，部署上线时只支持部署为此处选择的部署类型，例如此处只选择在线服务，那您导入后只能部署为在线服务。当前支持“在线服务”、“批量服务”和“边缘服务”。
“apis定义”	提供AI应用对外Restfull api数据定义，用于定义AI应用的输入、输出格式。

c. 确认信息填写无误，单击“立即创建”，完成AI应用创建。

在AI应用列表中，您可以查看刚创建的AI应用及其对应的版本。当AI应用状态变更为“正常”时，表示AI应用创建成功。在此页面，您还可以创建新版本、快速部署服务、发布AI应用等操作。

后续操作

部署服务：在“AI应用列表”中，单击AI应用名称左侧的小三角，打开此AI应用下的所有版本。在对应版本所在行，单击“操作”列的“部署”，在下拉框中选择部署类型，可以将AI应用部署上线为导入AI应用时所选择的部署类型。

9.3 管理 AI 应用版本

为方便溯源和AI应用反复调优，在ModelArts中提供了AI应用版本管理的功能，您可以基于版本对AI应用进行管理。

前提条件

已在ModelArts中创建AI应用。

创建新版本

在“AI应用管理 > AI应用”页面，单击操作列的“创建新版本”进入“创建新版本”页面，参见创建AI应用操作中的参数说明填写相关参数，单击“立即创建”，完成新版本的创建操作。

删除版本

在“AI应用管理 > AI应用”页面，单击AI应用名称左侧的小三角展开版本列表，在版本列表中，单击“操作”列“删除”，即可删除对应的版本。

说明

版本删除后不可恢复，请谨慎操作。

9.4 模型转换

9.4.1 模型转换操作

针对您在ModelArts或者本地构建的模型，为获得更高的算力，希望将模型应用于Ascend芯片、ARM或GPU上，此时，您需要将已有模型压缩/转换成相应的格式后，再应用至不同的芯片类型。

ModelArts提供了模型转换功能，即将已有的模型转换成所需格式，以便应用于算力和性能更高的芯片上。

模型转换主要应用场景如下所示：

- 使用Caffe（.caffemodel格式）或者Tensorflow框架（“frozen_graph”或“saved_model”格式）训练的模型，使用转换功能可转换成om格式，转换后的模型可在昇腾（Ascend）芯片上部署运行。
- 使用Tensorflow框架训练模型（frozen_graph或“saved_model”格式）， 使用转换功能可以将模型转换量化成tflite格式，转换后的模型可以在ARM上部署运行。
- 使用Tensorflow框架训练模型（frozen_graph或“saved_model”格式）， 使用转换功能可以将模型转换量化成tensorRT格式， 转换后的模型可以在nvidia p4 GPU上部署运行。

约束限制

- 模型转换当前只支持三种芯片类型，分别为：Ascend、ARM、GPU。
- 模型转换当前仅支持原始框架类型为Caffe和Tensorflow的模型转换。当原始框架类型为Caffe时，输入数据类型为FLOAT；当原始框架类型为Tensorflow时，输入数据类型为INT32、BOOL、UINT8、FLOAT。
- ModelArts提供了转换模板供用户选择，只能选择对应模板进行转换，支持的模板描述，请参见**转换模板**。
- 现阶段由于tflite和tensorRT支持的算子和量化算子有限，可能存在部分模型转换失败的情况。如果出现转换失败，可以将页面的日志框滑到最后或者去转换输出目录下查看错误日志。
- 针对用于Ascend芯片的模型转换，其转换限制说明可参见“昇腾开发者社区”的**约束及参数说明**。
- 压缩/转换任务指定的OBS路径，需确保OBS目录与ModelArts在同一区域。
- 转换后的模型，再导入ModelArts时，需**从模板中选择元模型**。
- 支持模型文件类型为onnx的模型转换，转换时会先将其转换为Tensorflow框架的FrozenGraphDef格式，然后再转换为om格式。转换工具要求onnx版本为1.6.0，opset为9+。
- 当原始框架类型为Caffe时，模型文件（.prototxt）和权重文件（.caffemodel）的op name、op type必须保持名称一致（包括大小写）。
- 当原始框架类型为Caffe时，除了top与bottom相同的layer以外（例如BatchNorm，Scale，ReLU等），其他layer的top名称需要与其name名称保持一致。
- 当原始框架类型为TensorFlow时，支持FrozenGraphDef格式和SavedModel 格式。如果是SavedModel格式，转换时会先将其转换为FrozenGraphDef格式，然后再转换为om格式。
- 不支持动态shape的输入，例如：NHWC输入为[?,?,?,3]多个维度可任意指定数值。模型转换时需指定固定数值。
- 输入数据最大支持四维，转维算子（reshape、expanddim等）不能输出五维。
- 模型中的所有层算子除const算子外，输入和输出需要满足“dim!=0”。
- 模型转换不支持含有训练算子的模型。
- 量化（uint8）后的模型不支持模型转换。
- 模型中的算子只支持2D卷积，暂不支持3D卷积。暂不支持多批量转换batch_normalization_1算子和FusedBatchNorm算子。
- 只支持**Caffe算子清单**和**Tensorflow算子**清单中的算子，并需满足算子限制条件。

创建模型转换任务

1. 登录ModelArts管理控制台，在左侧导航栏中选择“AI应用管理 > 模型转换”，进入模型转换列表页面。
2. 单击左上角的“创建任务”，进入任务创建任务页面。
3. 在“创建任务”页面，参考以下说明，填写转换任务的详细参数。
   a. 填写转换任务的“名称”和“描述”信息。
   b. 填写转换任务的具体参数，详细描述请参见**表9-11**。

表 9-11 参数说明

参数	说明
输入框架	目前支持转换的框架有“Caffe”和“TensorFlow”。
转换输入目录	用于转换的模型所在目录，此目录必须为OBS目录，且模型文件的目录需符合ModelArts规范，详情请参见**模型输入目录规范**。
输出框架	当“输入框架”选择“Caffe”时，“输出框架”支持“MindSpore”。 当“输入框架”选择“TensorFlow”时，“输出框架”支持“TFLite”、“MindSpore”、“TensorRT”。
转换输出目录	模型转换完成后，根据此参数设置的目录存储模型。输出目录需符合ModelArts规范要求，详情请参见**模型输出目录说明**。
转换模板	ModelArts提供了一系列的模板，定义转换功能以及转换过程中所需的参数。当前支持的转换模板详细描述请参见**转换模板**。转换模板右侧下拉框，将根据您选择的“输入框架”和“输出框架”，展现匹配的模板。 在下拉框选择模板后，下方将呈现此模板对应的高级参数。例如量化精度，方便您可以对模型转换任务进行更高阶的设置。 不同的转换模板，其对应的高级选项支持的参数不同，每个模板支持的详细参数，请参见**转换模板**。

图 9-5 创建模型转换任务

4. 任务信息填写完成后，单击右下角“立即创建”。

   创建完成后，系统自动跳转至“模型转换”列表中。刚创建的转换任务将呈现在界面中，其“任务状态”为“初始化”。任务执行过程预计需要几分钟到十几分钟不等，请耐心等待，当“任务状态”变为“成功”时，表示任务运行完成并且模型转换成功。

   如果“任务状态”变为“失败”，建议单击任务名称进入详情页面，查看日志信息，根据日志信息调整任务的相关参数并创建新的转换任务。

   模型转换成功后，有如下几个使用场景：

- 可以在HiLens管理控制台，导入转换后的模型，导入后可将模型安装部署至HiLens Kits设备上。
- 登录“转换输出目录”对应的OBS路径，下载目录中的模型（.om格式），部署至您的设备中。

删除模型转换任务

针对运行结束的任务，如果不需要再使用，您可以删除转换任务。其中，“运行中”或“初始化”状态中的任务不支持删除操作。

说明

任务删除后，将无法恢复，请谨慎操作。

- **删除单个**

  在“模型转换列表”中，针对需要删除的单个任务，您可以在此任务所在行，单击操作列的“删除”，完成删除操作。

- **批量删除**

  在“模型转换列表”中，勾选多个待删除的任务，然后单击左上角“删除”，完成批量任务的删除操作。

9.4.2 模型输入目录规范

模型转换后，应用于不同的芯片，针对不同的芯片，其模型输入目录的要求不同。ModelArts当前对模型输入目录的要求为**Ascend芯片**和**ARM或GPU**两种。

Ascend 芯片

用于Ascend芯片的模型，其转换要求如下所示：

- 针对基于Caffe框架的模型，执行模型转换时，其输入目录需符合如下规范。

```
|
|---xxxx.caffemodel        模型参数文件，输入目录下有且只能有一个，必填。
|---xxxx.prototxt          模型网络文件，输入目录下有且只能有一个，必填。
|---insert_op_conf.cfg     插入算子配置文件，输入目录下有且只有一个，可选。
|---plugin                 自定义算子目录，输入目录下有且只能有一个plugin文件夹，可选。仅支持基于TE（Tensor Engine）开发的自定义算子。
```

- 针对基于TensorFlow框架的模型（“frozen_graph”或“saved_model”格式），执行模型转换时，其输入目录需符合如下规范。

  “frozen_graph”格式

```
|
|---xxxx.pb                模型网络文件，输入目录下有且只能有一个，必填。支持以frozen_graph或saved_model格式保存的模型。
|---insert_op_conf.cfg     插入算子配置文件，输入目录下有且只有一个，可选。
|---plugin                 自定义算子目录，输入目录下有且只能有一个plugin文件夹，可选。仅支持基于TE（Tensor Engine）开发的自定义算子。
```

  “saved_model”格式

```
|
|---saved_model.pb         模型网络文件，输入目录下有且只能有一个，必填。支持以frozen_graph或saved_model格式保存的模型。
|---variables              固定子目录名称，包含模型的权重偏差等信息，必选
    |---variables.index      必选
    |---variables.data-00000-of-00001 必选
|---insert_op_conf.cfg     插入算子配置文件，输入目录下有且只有一个，可选。
|---plugin                 自定义算子目录，输入目录下有且只能有一个plugin文件夹，可选。仅支持基于TE（Tensor Engine）开发的自定义算子。
```

ARM 或 GPU

用于ARM或GPU的模型，当前只支持TensorFlow框架的模型，包含两种格式“frozen_graph”和“saved_model”。

“frozen_graph”格式如下所示：

```
|
|---model                 模型存放目录，必须以model命名，有且只能有一个， 目录下只能放一个模型相关文件。
    |----xxx.pb           模型文件。必须是tensorflow的frozen_graph格式的文件。
|---calibration_data      校准数据集存放目录，必须以calibration_data命名，8bit和32bit转换都需要。输入目录下有且只能有一个。
32bit转换时可以直接使用python工具生成的test.npy文件。
    |---xx.npy            校准数据集。可以是多个npy格式文件，需要确保npy是在预处理后直接输入模型的数据，其输入的tensor需要与模型输入保持一致。
```

“saved_model”格式如下所示：

```
|
|---model                 模型存放目录，必须以model命名，有且只能有一个， 目录下只能放一个模型相关文件。
    |----saved_model.pb   模型文件。必须是tensorflow的saved_model格式的文件。
    |----variables        变量存储文件夹。
        |----variables.data-******-of-*****   saved_model格式文件需要的数据。
        |----variables.index                  saved_model格式文件需要的索引。
|---calibration_data      校准数据集存放目录，必须以calibration_data命名，8bit和32bit转换都需要。输入目录下有且只能有一个。32bit转换时可以直接使用python工具生成的test.npy文件。
    |---xx.npy            校准数据集。可以是多个npy格式文件，需要确保npy是在预处理后直接输入模型的数据，其输入的tensor需要与模型输入保持一致。
```

说明

32bit转换时，可以直接使用python工具生成的test.npy文件，并将此文件放置在上述两种格式的calibration_data目录下。

```
import numpy as np
a = np.arange(4)
np.save("test.npy", a)
```

9.4.3 模型输出目录说明

转换模型任务执行完成后，ModelArts将转换后的模型输出至指定的OBS路径。针对不同的转换任务，基于不同的芯片，其对应的目录有所区别，分为**Ascend芯片**和**ARM或GPU**两种。

Ascend 芯片

用于Ascend芯片的模型，其转换后输出目录说明如下所示：

- 针对基于Caffe框架的模型，执行模型转换时，其输出目录说明如下所示。

```
|
|---xxxx.om          转换输出的模型，可用于Ascend芯片，模型文件后缀统一为“.om”。
|---job_log.txt      转换过程的日志文件。
```

- 针对基于TensorFlow框架的模型，执行模型转换时，其输出目录说明如下所示。

```
|
|---xxxx.om          转换输出的模型，可用于Ascend芯片，模型文件后缀统一为“.om”。
|---job_log.txt      转换过程的日志文件。
```

ARM 或 GPU

用于ARM或GPU的模型，其转换后输出目录说明如下所示：

GPU格式如下所示：

```
|
|---model
      |---xxx.pb          GPU转换后模型后缀为“.pb”。
|---job_log.txt           转换过程的日志文件。
```

ARM格式如下所示：

```
|
|---model
      |---xxx.tflite        ARM转换后模型后缀为“.tflite”。
      |---config.json       8bit转换后，用户需要使用tflite时需要的参数。
|---job_log.txt       转换过程的日志文件。
```

9.4.4 转换模板

基于不同的AI框架，ModelArts提供的转换模板如下所示：

- **Caffe转Ascend**
- **Tensorflow frozen_graph转TFLite**
- **Tensorflow saved_model转TFLite**
- **Tensorflow frozen_graph转TensorRT**
- **Tensorflow saved_model转TensorRT**
- **Tensorflow frozen graph 转 Ascend**
- **TF-FrozenGraph-To-Ascend**
- **TF-SavedModel-To-Ascend**
- **Onnx-To-Ascend-TBE**
- **TF-FrozenGraph-To-Ascend-C32**
- **TF-SavedModel-To-Ascend-C32**
- **TF-FrozenGraph-To-Ascend-HiLens**
- **TF-SavedModel-To-Ascend-HiLens**

Caffe 转 Ascend

转换Caffe框架训练出来的模型， 转换后模型可在Ascend芯片上运行。

此模板无高级选项。

Tensorflow frozen_graph 转 TFLite

转换Tensorflow框架训练并以“frozen_graph”格式保存的模型，转换后模型可在ARM上运行。

表 9-12 Tensorflow frozen_graph 转 TFLite 的高级选项

参数名称	参数解释
“模型输入tensor名称”	以字符串形式输入模型输入张量名称，以“input1:input2”形式表示。
“模型输出tensor名称”	以字符串形式输入模型输出张量名称，以“output1:output2”形式表示。

参数名称	参数解释
“量化精度”	可选择8bit或32bit。32bit表示直接转换模型，8bit表示模型进行量化。
“量化批大小”	以数值形式输入量化批大小。必须为正整数。

Tensorflow saved_model 转 TFLite

转换Tensorflow框架训练并以“saved_model”格式保存的模型，转换后模型可在ARM上运行。

表 9-13 Tensorflow saved_model 转 TFLite 的高级选项

参数名称	参数解释
“模型签名”	以字符串形式输入模型输入tensor签名，默认会选择第一个签名。
“传入模型标签”	以字符串形式输入模型输出标签，默认会选择第一个标签。
“量化精度”	可选择8bit或32bit。32bit表示直接转换模型，8bit表示模型进行量化。
“量化批大小”	以数值形式输入量化批大小。必须为正整数。

Tensorflow frozen_graph 转 TensorRT

转换Tensorflow框架训练并以“frozen_graph”格式保存的模型，转换后模型可在GPU上运行。

表 9-14 Tensorflow frozen_graph 转 TensorRT 的高级选项

参数名称	参数解释
“模型输入tensor名称”	以字符串形式输入模型输入张量名称，以“input1:input2”形式表示。
“模型输出tensor名称”	以字符串形式输入模型输出张量名称，以“output1:output2”形式表示。
“量化精度”	可选择8bit或32bit。32bit表示直接转换模型，8bit表示模型进行量化。
“量化批大小”	以数值形式输入量化批大小。必须为正整数。

Tensorflow saved_model 转 TensorRT

转换Tensorflow框架训练并以“saved_model”格式保存的模型，转换后模型可在GPU上运行。

表 9-15 Tensorflow saved_model 转 TensorRT 的高级选项

参数名称	参数解释
“模型签名”	以字符串形式输入模型输入tensor签名，默认会选择第一个签名。
“传入模型标签”	以字符串形式输入模型输出标签，默认会选择第一个标签。
“量化精度”	可选择8bit或32bit。32bit表示直接转换模型，8bit表示模型进行量化。
“量化批大小”	以数值形式输入量化批大小。必须为正整数。

Tensorflow frozen graph 转 Ascend

转换Tensorflow框架训练并以“frozen_graph”格式保存的模型，转换后模型可在Ascend上运行。

表 9-16 Tensorflow frozen graph 转 Ascend 的高级选项

参数名称	参数解释
“输入张量形状”	模型输入数据的shape，输入数据格式为NHWC，如“input_name:1,224,224,3”，必填项。“input_name”必须是转换前的网络模型中的节点名称。当模型存在动态shape输入时必须提供。例如“input_name1:?,h,w,c”，该参数必填，其中“?”为batch数，表示1次处理的图片数量，需要根据实际情况填写，用于将动态shape的原始模型转换为固定shape的离线模型。目前不支持批量特性，转换输入张量形状batch只能为1。

TF-FrozenGraph-To-Ascend

转换Tensorflow框架训练并以“frozen_graph”格式保存的模型，转换后模型可在Ascend上运行。转换时支持使用基于TE（Tensor Engine）开发的自定义算子(TE算子)。

表 9-17 支持自定义算子转换模板的高级选项

参数名称	参数解释
“输入张量形状”	模型输入数据的shape，如“input_name1:n1,c1,h1,w1;input_name2:n2,c2,h2,w2”。“input_name”必须是转换前的网络模型中的节点名称。当模型存在动态shape输入时必须提供。例如“input_name1:?,h,w,c”，该参数必填，其中“?”为batch数，表示1次处理的图片数量，需要根据实际情况填写，用于将动态shape的原始模型转换为固定shape的离线模型。目前不支持批量特性，转换输入张量形状batch只能为1。转换时系统会解析输入模型获取输入张量并打印在日志中，如果不了解所使用模型的输入张量，可参考日志中的解析结果。
“输入数据格式”	支持NCHW和NHWC，默认是NHWC。当原始框架是TensorFlow时，默认是NHWC。如果实际是NCHW的话，需要通过此参数指定NCHW。原始框架为Caffe时，只支持NCHW格式。
“转换输出节点”	指定输出节点，例如“node_name1:0;node_name1:1;node_name2:0”，其中“node_name”必须是模型转换前的网络模型中的节点名称，冒号后的数字表示第几个输出，例如“node_name1:0”，表示节点名称为“node_name1”的第0个输出。如果不指定输出节点，则模型的输出默认为最后一层的算子信息，某些情况下，用户想要查看某层算子参数是否合适，则需要将该层算子的参数输出，即可以在模型转换时通过该参数指定输出某层算子。转换时系统会解析输入模型获取输出节点并打印在日志中，如果不了解所使用模型的输入张量，可以参考日志中的解析结果。
“优选数据格式”	指定网络算子优先选用的数据格式，“ND（N<=4）”和“5D”。仅在网络中算子的输入数据同时支持“ND”和“5D”两种格式时，指定该参数才会生效。“ND”表示模型中算子按NCHW转换成通用格式，“5D”表示模型中算子按华为自研的5维转换成华为格式。“5D”为默认值。
“生成高精度模型”	指定是否生成高精度FP16 Davinci模型。“0”为默认值，表示生成普通FP16 Davinci模型，推理性能更好。“1”表示生成高精度FP16 Davinci模型，推理精度更好。高精度当前仅支持Caffe算子（Convolution、Pooling、FullConnection）和TensorFlow算子（tf.nn.conv2d、tf.nn.max_poo）。
“网络输出数据类型”	FP32为默认值，推荐分类网络、检测网络使用。图像超分辨率网络，推荐使用UINT8，推理性能更好。

TF-SavedModel-To-Ascend

转换Tensorflow框架训练并以“saved_model”格式保存的模型，转换后模型可在Ascend上运行。转换时支持使用基于TE（Tensor Engine）开发的自定义算子(TE算子)。

表 9-18 支持自定义算子转换模板的高级选项

参数名称	参数解释
“输入张量形状”	模型输入数据的shape，如“input_name1:n1,c1,h1,w1;input_name2:n2,c2,h2,w2”。“input_name”必须是转换前的网络模型中的节点名称。当模型存在动态shape输入时必须提供。例如“input_name1:?,h,w,c”，该参数必填，其中“?”为batch数，表示1次处理的图片数量，需要根据实际情况填写，用于将动态shape的原始模型转换为固定shape的离线模型。目前不支持批量特性，转换输入张量形状batch只能为1。转换时系统会解析输入模型获取输入张量并打印在日志中，如果不了解所使用模型的输入张量，可参考日志中的解析结果。
“输入数据格式”	支持NCHW和NHWC，默认是NHWC。当原始框架是TensorFlow时，默认是NHWC。如果实际是NCHW的话，需要通过此参数指定NCHW。原始框架为Caffe时，只支持NCHW格式。
“转换输出节点”	指定输出节点，例如“node_name1:0;node_name1:1;node_name2:0”，其中“node_name”必须是模型转换前的网络模型中的节点名称，冒号后的数字表示第几个输出，例如“node_name1:0”，表示节点名称为“node_name1”的第0个输出。如果不指定输出节点，则模型的输出默认为最后一层的算子信息，某些情况下，用户想要查看某层算子参数是否合适，则需要将该层算子的参数输出，即可以在模型转换时通过该参数指定输出某层算子。转换时系统会解析输入模型获取输出节点并打印在日志中，如果不了解所使用模型的输入张量，可以参考日志中的解析结果。
“优选数据格式”	指定网络算子优先选用的数据格式，“ND（N<=4）”和“5D”。仅在网络中算子的输入数据同时支持“ND”和“5D”两种格式时，指定该参数才会生效。“ND”表示模型中算子按NCHW转换成通用格式，“5D”表示模型中算子按华为自研的5维转换成华为格式。“5D”为默认值。
“生成高精度模型”	指定是否生成高精度FP16 Davinci模型。“0”为默认值，表示生成普通FP16 Davinci模型，推理性能更好。“1”表示生成高精度FP16 Davinci模型，推理精度更好。高精度当前仅支持Caffe算子（Convolution、Pooling、FullConnection）和TensorFlow算子（tf.nn.conv2d、tf.nn.max_poo）。
“网络输出数据类型”	FP32为默认值，推荐分类网络、检测网络使用。图像超分辨率网络，推荐使用UINT8，推理性能更好。

Onnx-To-Ascend-TBE

转换onnx格式保存的模型，模型首先被转换为frozen_graph格式，然后再转换为可在Ascend上运行的模型。转换时支持使用基于TBE（Tensor Based Engine）开发的自定义算子(TBE算子)。

表 9-19 支持自定义算子转换模板的高级选项

参数名称	参数解释
“输入张量形状”	模型输入数据的shape，如“input_name1:n1,c1,h1,w1;input_name2:n2,c2,h2,w2”。“input_name”必须是转换前的网络模型中的节点名称。当模型存在动态shape输入时必须提供。例如“input_name1:?,h,w,c”，该参数必填，其中“?”为batch数，表示1次处理的图片数量，需要根据实际情况填写，用于将动态shape的原始模型转换为固定shape的离线模型。目前不支持批量特性，转换输入张量形状batch只能为1。转换时系统会解析输入模型获取输入张量并打印在日志中，如果不了解所使用模型的输入张量，可参考日志中的解析结果。
“输入数据格式”	支持NCHW和NHWC，默认是NHWC。当原始框架是TensorFlow时，默认是NHWC。如果实际是NCHW的话，需要通过此参数指定NCHW。原始框架为Caffe时，只支持NCHW格式。
“转换输出节点”	指定输出节点，例如“node_name1:0;node_name1:1;node_name2:0”，其中“node_name”必须是模型转换前的网络模型中的节点名称，冒号后的数字表示第几个输出，例如“node_name1:0”，表示节点名称为“node_name1”的第0个输出。如果不指定输出节点，则模型的输出默认为最后一层的算子信息，某些情况下，用户想要查看某层算子参数是否合适，则需要将该层算子的参数输出，即可以在模型转换时通过该参数指定输出某层算子。转换时系统会解析输入模型获取输出节点并打印在日志中，如果不了解所使用模型的输入张量，可以参考日志中的解析结果。
“优选数据格式”	指定网络算子优先选用的数据格式，“ND（N<=4）”和“5D”。仅在网络中算子的输入数据同时支持“ND”和“5D”两种格式时，指定该参数才会生效。“ND”表示模型中算子按NCHW转换成通用格式，“5D”表示模型中算子按华为自研的5维转换成华为格式。“5D”为默认值。
“生成高精度模型”	指定是否生成高精度FP16 Davinci模型。“0”为默认值，表示生成普通FP16 Davinci模型，推理性能更好。“1”表示生成高精度FP16 Davinci模型，推理精度更好。高精度当前仅支持Caffe算子（Convolution、Pooling、FullConnection）和TensorFlow算子（tf.nn.conv2d、tf.nn.max_poo）。
“网络输出数据类型”	FP32为默认值，推荐分类网络、检测网络使用。图像超分辨率网络，推荐使用UINT8，推理性能更好。

TF-FrozenGraph-To-Ascend-C32

转换Tensorflow框架训练并以“frozen_graph”格式保存的模型，转换后模型可在Ascend上运行。转换时支持使用基于TBE（Tensor Based Engine）开发的自定义算子(TBE算子)。

表 9-20 支持自定义算子转换模板的高级选项

参数名称	参数解释
“输入张量形状”	模型输入数据的shape，如“input_name1:n1,c1,h1,w1;input_name2:n2,c2,h2,w2”。“input_name”必须是转换前的网络模型中的节点名称。当模型存在动态shape输入时必须提供。例如“input_name1:?,h,w,c”，该参数必填，其中“?”为batch数，表示1次处理的图片数量，需要根据实际情况填写，用于将动态shape的原始模型转换为固定shape的离线模型。目前不支持批量特性，转换输入张量形状batch只能为1。转换时系统会解析输入模型获取输入张量并打印在日志中，如果不了解所使用模型的输入张量，可参考日志中的解析结果。
“输入数据格式”	支持NCHW和NHWC，默认是NHWC。当原始框架是TensorFlow时，默认是NHWC。如果实际是NCHW的话，需要通过此参数指定NCHW。原始框架为Caffe时，只支持NCHW格式。
“转换输出节点”	指定输出节点，例如“node_name1:0;node_name1:1;node_name2:0”，其中“node_name”必须是模型转换前的网络模型中的节点名称，冒号后的数字表示第几个输出，例如“node_name1:0”，表示节点名称为“node_name1”的第0个输出。如果不指定输出节点，则模型的输出默认为最后一层的算子信息，某些情况下，用户想要查看某层算子参数是否合适，则需要将该层算子的参数输出，即可以在模型转换时通过该参数指定输出某层算子。转换时系统会解析输入模型获取输出节点并打印在日志中，如果不了解所使用模型的输入张量，可以参考日志中的解析结果。
“优选数据格式”	指定网络算子优先选用的数据格式，“ND（N<=4）”和“5D”。仅在网络中算子的输入数据同时支持“ND”和“5D”两种格式时，指定该参数才会生效。“ND”表示模型中算子按NCHW转换成通用格式，“5D”表示模型中算子按华为自研的5维转换成华为格式。“5D”为默认值。
“生成高精度模型”	指定是否生成高精度FP16 Davinci模型。“0”为默认值，表示生成普通FP16 Davinci模型，推理性能更好。“1”表示生成高精度FP16 Davinci模型，推理精度更好。高精度当前仅支持Caffe算子（Convolution、Pooling、FullConnection）和TensorFlow算子（tf.nn.conv2d、tf.nn.max_poo）。
“网络输出数据类型”	FP32为默认值，推荐分类网络、检测网络使用。图像超分辨率网络，推荐使用UINT8，推理性能更好。

TF-SavedModel-To-Ascend-C32

转换Tensorflow框架训练并以“saved_model”格式保存的模型，转换后模型可在Ascend上运行。转换时支持使用基于TE（Tensor Engine）开发的自定义算子(TE算子)。

表 9-21 支持自定义算子转换模板的高级选项

参数名称	参数解释
“输入张量形状”	模型输入数据的shape，如“input_name1:n1,c1,h1,w1;input_name2:n2,c2,h2,w2”。“input_name”必须是转换前的网络模型中的节点名称。当模型存在动态shape输入时必须提供。例如“input_name1:?,h,w,c”，该参数必填，其中“?”为batch数，表示1次处理的图片数量，需要根据实际情况填写，用于将动态shape的原始模型转换为固定shape的离线模型。目前不支持批量特性，转换输入张量形状batch只能为1。转换时系统会解析输入模型获取输入张量并打印在日志中，如果不了解所使用模型的输入张量，可参考日志中的解析结果。
“输入数据格式”	支持NCHW和NHWC，默认是NHWC。当原始框架是TensorFlow时，默认是NHWC。如果实际是NCHW的话，需要通过此参数指定NCHW。原始框架为Caffe时，只支持NCHW格式。
“转换输出节点”	指定输出节点，例如“node_name1:0;node_name1:1;node_name2:0”，其中“node_name”必须是模型转换前的网络模型中的节点名称，冒号后的数字表示第几个输出，例如“node_name1:0”，表示节点名称为“node_name1”的第0个输出。如果不指定输出节点，则模型的输出默认为最后一层的算子信息，某些情况下，用户想要查看某层算子参数是否合适，则需要将该层算子的参数输出，即可以在模型转换时通过该参数指定输出某层算子。转换时系统会解析输入模型获取输出节点并打印在日志中，如果不了解所使用模型的输入张量，可以参考日志中的解析结果。
“优选数据格式”	指定网络算子优先选用的数据格式，“ND（N<=4）”和“5D”。仅在网络中算子的输入数据同时支持“ND”和“5D”两种格式时，指定该参数才会生效。“ND”表示模型中算子按NCHW转换成通用格式，“5D”表示模型中算子按华为自研的5维转换成华为格式。“5D”为默认值。
“生成高精度模型”	指定是否生成高精度FP16 Davinci模型。“0”为默认值，表示生成普通FP16 Davinci模型，推理性能更好。“1”表示生成高精度FP16 Davinci模型，推理精度更好。高精度当前仅支持Caffe算子（Convolution、Pooling、FullConnection）和TensorFlow算子（tf.nn.conv2d、tf.nn.max_poo）。
“网络输出数据类型”	FP32为默认值，推荐分类网络、检测网络使用。图像超分辨率网络，推荐使用UINT8，推理性能更好。

TF-FrozenGraph-To-Ascend-HiLens

主要提供给华为HiLens使用，当HiLens Kit系统固件版本为2.2.200.011时建议使用此模板进行转换。支持将TensorFlow frozen_graph模型转换成可在ascend芯片上运行的模型，转换时支持使用基于TE(Tensor Engine)开发的自定义算子。

表 9-22 支持自定义算子转换模板的高级选项

参数名称	参数解释
“输入张量形状”	模型输入数据的shape，如“input_name1:n1,c1,h1,w1;input_name2:n2,c2,h2,w2”。“input_name”必须是转换前的网络模型中的节点名称。当模型存在动态shape输入时必须提供。例如“input_name1:?,h,w,c”，该参数必填，其中“?”为batch数，表示1次处理的图片数量，需要根据实际情况填写，用于将动态shape的原始模型转换为固定shape的离线模型。目前不支持批量特性，转换输入张量形状batch只能为1。转换时系统会解析输入模型获取输入张量并打印在日志中，如果不了解所使用模型的输入张量，可参考日志中的解析结果。
“输入数据格式”	支持NCHW和NHWC，默认是NHWC。当原始框架是TensorFlow时，默认是NHWC。如果实际是NCHW的话，需要通过此参数指定NCHW。原始框架为Caffe时，只支持NCHW格式。
“转换输出节点”	指定输出节点，例如“node_name1:0;node_name1:1;node_name2:0”，其中“node_name”必须是模型转换前的网络模型中的节点名称，冒号后的数字表示第几个输出，例如“node_name1:0”，表示节点名称为“node_name1”的第0个输出。如果不指定输出节点，则模型的输出默认为最后一层的算子信息，某些情况下，用户想要查看某层算子参数是否合适，则需要将该层算子的参数输出，即可以在模型转换时通过该参数指定输出某层算子。转换时系统会解析输入模型获取输出节点并打印在日志中，如果不了解所使用模型的输入张量，可以参考日志中的解析结果。
“优选数据格式”	指定网络算子优先选用的数据格式，“ND（N<=4）”和“5D”。仅在网络中算子的输入数据同时支持“ND”和“5D”两种格式时，指定该参数才会生效。“ND”表示模型中算子按NCHW转换成通用格式，“5D”表示模型中算子按华为自研的5维转换成华为格式。“5D”为默认值。
“生成高精度模型”	指定是否生成高精度FP16 Davinci模型。“0”为默认值，表示生成普通FP16 Davinci模型，推理性能更好。“1”表示生成高精度FP16 Davinci模型，推理精度更好。高精度当前仅支持Caffe算子（Convolution、Pooling、FullConnection）和TensorFlow算子（tf.nn.conv2d、tf.nn.max_poo）。
“网络输出数据类型”	FP32为默认值，推荐分类网络、检测网络使用。图像超分辨率网络，推荐使用UINT8，推理性能更好。

TF-SavedModel-To-Ascend-HiLens

主要提供给华为HiLens使用，当HiLens Kit系统固件版本为2.2.200.011时建议使用此模板进行转换。支持将TensorFlow saved_model模型转换成可在ascend芯片上运行的模型，转换时支持使用基于TE(Tensor Engine)开发的自定义算子。

表 9-23 支持自定义算子转换模板的高级选项

参数名称	参数解释
“输入张量形状”	模型输入数据的shape，如“input_name1:n1,c1,h1,w1;input_name2:n2,c2,h2,w2”。“input_name”必须是转换前的网络模型中的节点名称。当模型存在动态shape输入时必须提供。例如“input_name1:?,h,w,c”，该参数必填，其中“?”为batch数，表示1次处理的图片数量，需要根据实际情况填写，用于将动态shape的原始模型转换为固定shape的离线模型。目前不支持批量特性，转换输入张量形状batch只能为1。转换时系统会解析输入模型获取输入张量并打印在日志中，如果不了解所使用模型的输入张量，可参考日志中的解析结果。
“输入数据格式”	支持NCHW和NHWC，默认是NHWC。当原始框架是TensorFlow时，默认是NHWC。如果实际是NCHW的话，需要通过此参数指定NCHW。原始框架为Caffe时，只支持NCHW格式。
“转换输出节点”	指定输出节点，例如“node_name1:0;node_name1:1;node_name2:0”，其中“node_name”必须是模型转换前的网络模型中的节点名称，冒号后的数字表示第几个输出，例如“node_name1:0”，表示节点名称为“node_name1”的第0个输出。如果不指定输出节点，则模型的输出默认为最后一层的算子信息，某些情况下，用户想要查看某层算子参数是否合适，则需要将该层算子的参数输出，即可以在模型转换时通过该参数指定输出某层算子。转换时系统会解析输入模型获取输出节点并打印在日志中，如果不了解所使用模型的输入张量，可以参考日志中的解析结果。
“优选数据格式”	指定网络算子优先选用的数据格式，“ND（N<=4）”和“5D”。仅在网络中算子的输入数据同时支持“ND”和“5D”两种格式时，指定该参数才会生效。“ND”表示模型中算子按NCHW转换成通用格式，“5D”表示模型中算子按华为自研的5维转换成华为格式。“5D”为默认值。
“生成高精度模型”	指定是否生成高精度FP16 Davinci模型。“0”为默认值，表示生成普通FP16 Davinci模型，推理性能更好。“1”表示生成高精度FP16 Davinci模型，推理精度更好。高精度当前仅支持Caffe算子（Convolution、Pooling、FullConnection）和TensorFlow算子（tf.nn.conv2d、tf.nn.max_poo）。
“网络输出数据类型”	FP32为默认值，推荐分类网络、检测网络使用。图像超分辨率网络，推荐使用UINT8，推理性能更好。

9.5 我的订阅模型

在AI Gallery中，支持订阅官方发布或者他人分享的模型，订阅后的模型，可推送至ModelArts模型管理中，进行统一管理。

说明

我的订阅模型与云服务订阅模型的区别：

- 在管理控制台中，模型管理所在位置不同。我的订阅统一管理在“AI应用管理>AI应用>我的订阅”页面中，而云服务订阅模型管理在“AI应用管理>云服务订阅AI应用”页面中。
- 模型来源不同。我的订阅，模型来源于AI Gallery；云服务订阅模型，模型来源于其他AI服务开发的模型。

我的订阅模型列表

在ModelArts的“AI应用管理>AI应用>我的订阅”页面中，罗列了从AI Gallery订阅的所有模型。如果需要更多模型，可单击“查找AI应用”，跳转至“AI Gallery”选择更多。

图 9-6 我的订阅模型列表

我的订阅模型，可通过如下几个操作获得：

1. 订阅

   在“AI Gallery”中，作为买家，完成模型订阅。订阅模型的操作指导请参见**买家指导（订阅模型）**。

2. **将订阅模型部署为服务**

将订阅模型部署为服务

针对我的订阅模型，支持将模型一键部署为服务。

1. 在“我的订阅”列表中，单击模型名称左侧的小三角，展开模型的详情和模型版本。
2. 在“版本列表”中，单击“部署”，选择对应的服务类型。

图 9-7 部署

– 若您选择部署的是商用模型，则选择服务类型后会弹出“修改配额”窗口，根据需要选择配额后单击“确定”即可跳转至“部署”页面。

说明

由于商用模型支持同时购买多种配额模式的资产，所以仅部署商用模型时需要进行配额选择。免费模型仅一种配额模式无需选择。

图 9-8 修改配额

– 若您选择部署的非商业模型，系统自动跳转至“部署”页面。

3. 在部署页面中，无需再选择模型及其版本，参考部署服务的操作指导完成其他参数填写，即可部署为您需要的服务。

9.6 云服务订阅 AI 应用

针对其他AI服务开发的AI应用，您可以订阅AI应用，并在ModelArts管理控制台中，快速部署为边缘服务。

在“云服务订阅AI应用”中，AI应用具备几种状态：“正常”、“未支付”、“已过期”。

- “正常”：表示已付费，可以使用的AI应用。请关注其剩余可用天数。
- “未支付”：表示已购买未支付费用，或者未购买，此AI应用暂时无法部署。购买后，AI应用才能变为正常状态。
- “已过期”：表示已经超过可用天数，AI应用暂时无法部署。续费后，AI应用才能变为正常状态。

使用限制

- 只支持部署为边缘服务。
- 当前仅支持部分服务的AI应用，将持续不断开放新的服务。

购买 AI 应用

当在其他AI服务完成AI应用订阅，订阅后的AI应用将呈现在ModelArts的“AI应用管理>AI应用>云服务订阅AI应用”中。在完成购买操作后，AI应用才可用于部署。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“AI应用管理>AI应用>云服务订阅AI应用”。
2. 在“云服务订阅AI应用”页面，选择“未支付”状态的AI应用，单击操作列的“购买”。
3. 在“购买AI应用”页面，填写“高级配置”、设置“购买时长”，然后单击“下一步”。
4. 在“规格确认”页面，确认信息无误后，单击“去支付”。
5. 在“支付”页面，确认需支付的“金额”，选择“支付方式”，然后单击“确认支付”。

   支付成功后，您可以通过界面提示链接，快速返回订阅模型的AI服务控制台（如人脸识别服务控制台），或者返回订阅列表页面。

部署 AI 应用

对于状态为“正常”的AI应用，可将AI应用部署为边缘服务。

1. 在“订阅列表”页面，选择“正常”状态的AI应用，单击操作列的“部署”。
2. 在部署页面，设置边缘服务的名称和描述，同时，选择AI应用配置以及边缘节点。

表 9-24 参数说明

参数名称	说明
“名称”	边缘服务的名称。
“描述”	边缘服务的简要描述。
“选择AI应用及配置”	此处显示在“订阅列表”中对应的AI应用及其版本号。

参数名称	说明
“计算节点规格”	支持如下几种规格。 ● “CPU：2核 8GiB” ● “CPU：2核 8GiB GPU：1*P4” ● “自定义规格”，如果选择自定义规格，可以在参数下方设置您所需的“CPU”、“Memory”、“GPU”或“Ascend”。其中，“GPU”和“Ascend”只能二选一。
“环境变量”	设置环境变量，注入环境变量到容器实例。
“选择边缘节点”	边缘节点是您自己的边缘计算设备，用于运行边缘应用，处理您的数据，并安全、便捷地和云端应用进行协同。 单击选择边缘节点“添加”，在弹出的“添加节点”对话框中选择节点。选择您已创建的节点后，单击“确定”。

图 9-9 部署为边缘服务

3. 确认信息填写无误后，单击“立即创建”。

   边缘服务创建完成后，可跳转至“部署上线 > 边缘服务”，查看服务的“状态”，当“状态”变为“运行中”时，表示服务已部署成功。如何访问边缘服务，操作指导请参见**访问边缘服务**。

续费

针对“已过期”或“正常”的AI应用，您都可以为其续费。

1. 在“订阅列表”页面，选择“已过期”或“正常”状态的AI应用，单击操作列的“续费”。
2. 在“续费”页面，选择续费时长，然后单击“去支付”。
3. 在“支付”页面，确认需支付的“金额”，选择“支付方式”，然后单击“确认付款”。

支付成功后，您可以通过界面提示链接，快速返回订阅AI应用的AI服务控制台（如人脸识别服务控制台），或者返回订阅列表页面。

退订

针对“正常”的AI应用，如果您不再需要订阅此AI应用，可以执行退订操作。如果AI应用已部署至边缘服务，不支持退订操作。如果需要执行退订，请先删除使用此AI应用部署的边缘服务。

1. 在“订阅列表”页面，选择“正常”状态的AI应用，单击操作列的“退订”。
2. 在“退订资源”页面，选择退订原因，然后单击“退订”。
3. 在弹出的对话框中，确认信息无误后，单击“是”，完成退订操作。

   退订操作提交成功后，您可以通过界面提示链接，查看退订的处理进度。

9.7 发布 AI 应用

针对ModelArts中创建的AI应用，支持以下三种发布方式：

- **发布至AI Gallery**

  AI Gallery是在ModelArts的基础上构建的开发者生态社区，提供算法、模型、数据集等内容的共享，为高校科研机构、AI应用开发商、解决方案集成商、企业级个人开发者等群体，提供安全、开放的共享，加速AI资产的开发与落地。

  发布至AI Gallery的资产是免费的，只需要支付在使用过程中消耗的硬件资源，硬件资源费用将根据实际使用情况由华为云ModelArts管理控制台向使用方收取。

- **发布至AI云市场**

  华为云云市场是软件及服务交易交付平台。云市场AI专区汇聚优质的人工智能服务提供商，提供丰富的人工智能解决方案、应用、API及算法模型，助力用户快速部署、接入、调用相关应用，方便地购买和使用算法模型。

  发布至云市场AI专区的模型为商业售卖资产。买家需购买商品的使用配额进行有偿使用。

  说明

  云市场仅支持企业级卖家发布商品，请在首次发布商品前入驻华为云市场成为企业级卖家。

- **发布至AI大赛**

  华为云人工智能大赛面向开发者组织了一些开发者大赛，您可以在ModelArts开发模型，并将参赛模型提交至对应的比赛项目。

前提条件

已在ModelArts中导入模型。且至少存在一个版本。

发布至 AI Gallery

ModelArts提供了“AI Gallery”功能，方便将个人的模型等共享给所有ModelArts用户，您也可以从“AI Gallery”获取他人共享的内容，快速完成构建。在您完成模型的训练和导入之后，您可以将自己的模型分享至“AI Gallery”，进行知识共享。

1. 登录ModelArts管理控制台，在左侧导航栏中选择“AI应用管理 > AI应用”，进入AI应用列表页面。

2. 单击AI应用左侧的小三角展开版本列表，单击“操作”列的“发布 ”进入发布页面。
3. 在发布弹出框中，单击“前往AI Gallery”进入AI Gallery。

图 9-10 前往 AI Gallery

4. 进入AI Gallery后，请参考**免费分享模型**。

发布至 AI 云市场

华为云云市场有完整的售前、交易、售后的保障体系，用户无需担忧交易，省心省力。在您完成模型的训练和导入之后，您可以将自己的模型发布至AI 云市场进行商品售卖。请在首次发布商品前**入驻华为云市场**成为企业级卖家。

1. 登录ModelArts管理控制台，在左侧导航栏中选择“AI应用管理 > AI应用”，进入AI应用列表页面。
2. 单击AI应用左侧的小三角展开版本列表，单击“操作”列的“发布”进入发布页面。
3. 在发布弹出框中，单击“前往AI 云市场”进入云市场。

图 9-11 前往云市场

4. 进入云市场后，请参考**发布AI资产类商品操作指导**。

发布至 AI 大赛

华为云人工智能大赛面向开发者组织了一些开发者大赛，您可以在ModelArts开发AI应用，并将参赛AI应用提交至对应的比赛项目。

1. 登录ModelArts管理控制台，在左侧导航栏中选择“AI应用管理 > AI应用”，进入AI应用列表页面。
2. 单击AI应用左侧的小三角展开版本列表，单击“操作”列的“发布”进入发布页面。
3. 在发布弹出框中，填写“比赛项目”，确认AI应用信息无误后，单击“提交作品”完成提交。

图 9-12 提交参赛 AI 应用

10 部署服务（即将下线）

10.1 部署服务简介

说明

为了更方便用户使用和清晰的获取文档帮助，《AI工程师用户指南》将按照AI开发流程调整拆分为多本文档，在一级导航中呈现，并从用户使用场景对各模块文档进行了优化和改进。“部署服务”的内容已拆分至《**推理部署**》的“部署AI应用（在线服务）”、“部署AI应用（批量服务）”和“部署AI应用（边缘服务）”,《AI工程师用户指南》里的“部署服务”即将下线。

在完成训练作业并创建为AI应用后，可在“部署上线”页面对AI应用进行部署，您也可以将从OBS导入的元模型进行部署。ModelArts当前支持如下几种部署类型：

- **在线服务**
  将AI应用部署为一个Web Service，并且提供在线的测试UI与监控功能。
- **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。
- **边缘服务**
  通过智能边缘平台，在边缘节点将AI应用部署为一个Web Service。

10.2 在线服务

10.2.1 部署为在线服务

AI应用准备完成后，您可以将AI应用部署为在线服务，对在线服务进行预测和调用。

说明

单个用户最多可创建20个在线服务。

前提条件

- 数据已完成准备：已在ModelArts中创建状态“正常”可用的AI应用。
- 由于在线运行需消耗资源，确保帐户未欠费。

操作步骤

1. 登录ModelArts管理控制台，在左侧导航栏中选择“部署上线 > 在线服务”，默认进入“在线服务”列表。
2. 在“在线服务”列表中，单击左上角“部署”，进入“部署”页面。
3. 在“部署”页面，填写在线服务相关参数。
   a. 填写基本信息，详细参数说明请参见表10-1。

表 10-1 基本信息参数说明

参数名称	说明
“名称”	在线服务的名称，请按照界面提示规则填写。
“是否自动停止”	启用该参数并设置时间后，服务将在指定时间后自动停止。如果不启用此参数，在线服务将一直运行，同时一直收费，自动停止功能可以帮您避免产生不必要的费用。默认开启自动停止功能，且默认值为“1小时后”。 目前支持设置为“1小时后”、“2小时后”、“4小时后”、“6小时后”、“自定义”。如果选择“自定义”的模式，可在右侧输入框中输入1~24范围内的任意整数。
“描述”	在线服务的简要说明。

   b. 填写资源池和AI应用配置等关键信息，详情请参见表10-2。

表 10-2 参数说明

参数名称	子参数	说明
“资源池”	“公共资源池”	公共资源池有CPU或GPU两种规格，不同规格的资源池，其收费标准不同，详情请参见价格详情说明。当前仅支持按需付费模式。
“资源池”	“专属资源池”	创建专属资源池请参见创建专属资源池。您可以在资源池规格中选择对应的规格进行使用。
“选择AI应用及配置”	“AI应用来源”	根据您的实际情况选择“我的AI应用”或者“我的订阅”。
	“选择AI应用及版本”	选择状态“正常”的AI应用及版本。
	“分流”	设置当前实例节点的流量占比，服务调用请求根据该比例分配到当前版本上。 如您仅部署一个版本的AI应用，请设置为100%。如您添加多个版本进行灰度发布，多个版本分流之和设置为100%。

参数名称	子参数	说明
	“计算节点规格”	请根据界面显示的列表，选择可用的规格，置灰的规格表示当前环境无法使用。
	“计算节点个数”	设置当前版本AI应用的实例个数。如果节点个数设置为1，表示后台的计算模式是单机模式；如果节点个数设置大于1，表示后台的计算模式为分布式的。请根据实际编码情况选择计算模式。
	“环境变量”	设置环境变量，注入环境变量到容器实例。为确保您的数据安全，在环境变量中，请勿输入敏感信息，如明文密码。
	“添加AI应用版本进行灰度发布”	当选择的AI应用有多个版本时，您可以添加多个AI应用版本，并配置其分流占比，完成多版本和灵活流量策略的灰度发布，实现AI应用版本的平滑过渡升级。 **说明** 当前免费计算规格不支持多版本灰度发布。
“服务流量限制”	-	服务流量限制是指每秒内一个服务能够被访问的次数上限。您可以根据实际需求设置每秒流量限制。
“支持APP认证”	APP授权配置	默认关闭。如需开启此功能，请参见**访问在线服务（APP认证）**了解详情并根据实际情况进行设置。
“订阅消息”	-	订阅消息使用消息通知服务，在事件列表中选择需要监控的资源池状态，在事件发生时发送消息通知。

图 10-1 设置 AI 应用相关信息

4. 确认填写信息无误后，根据界面提示完成在线服务的部署。部署服务一般需要运行一段时间，根据您选择的数据量和资源不同，部署时间将耗时几分钟到几十分钟不等。

说明

在线服务部署完成后，将立即启动，运行过程中将按照您选择的资源按需计费。

您可以前往在线服务列表，查看在线服务的基本情况。在在线服务列表中，刚部署的服务“状态”为“部署中”，当在线服务的“状态”变为“运行中”时，表示服务部署完成。

10.2.2 查看服务详情

当AI应用部署为在线服务成功后，您可以进入“在线服务”页面，来查看服务详情。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“部署上线>在线服务”，进入“在线服务”管理页面。
2. 单击目标服务名称，进入服务详情页面。
   您可以查看服务的“名称”、“状态”等信息，详情说明请参见表10-3。

   表 10-3 在线服务配置

参数	说明
名称	在线服务名称。
状态	在线服务当前状态。
来源	在线服务的来源。
服务ID	在线服务的ID。
调用失败次数/总次数	服务调用次数从创建后开始统计。 如修改AI应用数量，或在AI应用处于“未就绪”时对服务发起调用，不计入统计。
网络配置	如果您使用专属资源池，显示专属资源池的自定义网络配置。
描述	您可以单击编辑按钮，添加服务描述。
个性化配置	您可以为在线服务的不同版本设定不同配置条件，并支持携带自定义运行参数，丰富版本分流策略或同一版本内的不同运行配置。您可以打开个性化配置按钮，单击“查看配置”修改服务个性化配置。
服务流量限制	服务流量限制是指每秒内一个服务能够被访问的次数上限。

3. 您可以在在线服务的详情页面，通过切换页签查看更多详细信息，详情说明请参见表10-4。

   表 10-4 在线服务详情

参数	说明
调用指南	展示API接口地址、AI应用信息、输入参数、输出参数。您可以通过复制API接口地址，调用服务。如果您支持APP认证方式，可以在调用指南查看API接口地址和授权管理详情，包括“应用名称”、“AppKey”、“AppSecret”等信息。您也可以在此处对APP应用进行“添加授权”或“解除授权”的操作。
预测	对在线服务进行预测测试。具体操作请参见测试服务。

参数	说明
配置更新记录	展示“当前配置”详情和“历史更新记录”。 ● “当前配置”：AI应用名称、版本、状态、分流、 ● “历史更新记录”：展示历史AI应用相关信息。
监控信息	展示当前服务的“资源统计信息”和“AI应用调用次数统计”。 ● “资源统计信息”：包括CPU、内存、GPU的可用和已用信息。 ● “AI应用调用次数统计”：当前AI应用的调用次数，从AI应用状态为“已就绪”后开始统计。
事件	展示当前服务使用过程中的关键操作，比如服务部署进度、部署异常的详细原因、服务被启动、停止、更新的时间点等。
日志	展示当前服务下每个AI应用的日志信息。包含最近5分钟、最近30分钟、最近1小时和自定义时间段。 自定义时间段您可以选择开始时间和结束时间。

修改服务个性化配置

服务个性化配置规则由配置条件、访问版本、自定义运行参数（包括配置项名称和配置项值）组成。

您可以为在线服务的不同版本设定不同配置条件，并支持携带自定义运行参数。

个性化配置规则的优先级与顺序相对应，从高到低设置。您可以通过拖动个性化配置规则的顺序更换优先级。

当匹配了某一规则后就不再继续下一规则的判断，最多允许配置10个条件。

表 10-5 个性化配置参数

参数	是否必选	说明
配置条件	必选	SPEL（Spring Expression Language）规则的表达式，当前仅支持字符型的“相等”和“matches”。
访问版本	必选	服务个性化配置规则对应的访问版本。当匹配到规则时，请求该版本的在线服务。
配置项名称	可选	自定义运行参数的Key值，不超过128个字符。 当需要通过Header（http消息头）携带自定义运行参数至在线服务时，可以配置。
配置项值	可选	自定义运行参数的Value值，不超过256个字符。 当需要通过Header（http消息头）携带自定义运行参数至在线服务时，可以配置。

可以设置以下三种场景：

- 如果在线服务部署多个版本用于灰度发布，可以使用个性化配置实现按用户分流。

表 10-6 按内置变量配置条件

内置变量	说明
DOMAIN_NAME	调用预测请求的帐号名。
DOMAIN_ID	调用预测请求的帐号ID。
PROJECT_NAME	调用预测请求的项目名。
PROJECT_ID	调用预测请求的项目ID。
USER_NAME	调用预测请求的用户名。
USER_ID	调用预测请求的用户ID。

“#”表示引用变量，匹配的字符串需要用单引号。

```
#{内置变量} == '字符串'
#{内置变量} matches '正则表达式'
```

– 示例一：

  当调用预测请求的帐号名为“zhangsan”时，匹配至指定版本。

  ```
  #DOMAIN_NAME == 'zhangsan'
  ```

– 示例二：

  当调用预测请求的帐号名以“op”开头时，匹配至指定版本。

  ```
  #DOMAIN_NAME matches 'op.*'
  ```

表 10-7 常用的正则匹配表达式

字符	描述
“.”	匹配除“\n”之外的任何单个字符串。需匹配包括“\n”在内的任何字符，请使用“(.|\n)”的模式。
“*”	匹配前面的子表达式零次或多次。例如，“zo*”能匹配“z”以及“zoo”。
“+”	匹配前面的子表达式一次或多次。例如，“zo+”能匹配“zo”以及“zoo”，但不能匹配“z”。
“?”	匹配前面的子表达式零次或一次。例如，“do(es)?”可以匹配“does”或“does”中的“do”。
“^”	匹配输入字符串的开始位置。
“$”	匹配输入字符串的结束位置。

字符	描述
“{n}”	*n*是一个非负整数。匹配确定的*n*次。例如，“o{2}”不能匹配“Bob”中的“o”，但是能匹配“food”中的两个“o”。
“x|y”	匹配x或y。例如，“z|food”能匹配“z”或“food”。“(z|f)ood”则匹配“zood”或“food”。
“[xyz]”	字符集合。匹配所包含的任意一个字符。例如，“[abc]”可以匹配“plain”中的“a”。

更多正则表达式请参考**表达式全集**。

图 10-2 按用户分流

- 如果在线服务部署多个版本用于灰度发布，可以使用个性化配置实现通过Header来访问不同版本。

  您需要通过"#HEADER_"开头说明引用header作为条件

  ```
  #HEADER_{key} == '{value}'
  #HEADER_{key} matches '{value}'
  ```

  - 示例一：

    当预测的http请求的header中存在version，且值为0.0.1则符合条件。不存在此header或者值不为0.0.1都不符合条件。

    ```
    #HEADER_version == '0.0.1'
    ```

  - 示例二：

    当预测的http请求的header中存在testheader且值符合正以mock开头时，可匹配到这条规则。

    ```
    #HEADER_testheader matches 'mock.*'
    ```

图 10-3 通过 Header 访问不同版本

- 如果在线服务部署的版本支持使用不同的运行配置，您可以通过“配置项名称”和“配置项值”携带自定义运行参数至在线服务，实现不同用户使用不同运行配置。

示例：

用户zhangsan访问时，AI应用使用配置A；用户lisi访问时，AI应用使用配置B。当匹配到运行配置条件时，ModelArts会在请求里增加一个Header，传入自定义运行参数，其中Key是“配置项名称”，Value是“配置项值”。

图 10-4 个性化配置规则支持传入自定义运行参数。

10.2.3 测试服务

AI应用部署为在线服务成功后，您可以在“预测”页签进行代码调试或添加文件测试。根据AI应用定义的输入请求不同（JSON文本或文件），测试服务包括如下两种方式：

1. **JSON文本预测**：如当前部署服务的AI应用，其输入类型指定的为JSON文本类，即不含有文件类型的输入，可以在“预测”页签输入JSON代码进行服务预测。
2. **文件预测**：如当前部署服务的AI应用，其输入类型指定为文件类，可包含图片、音频或视频等场景，可以在“预测”页签添加图片进行服务预测。

说明

- 如果您的输入类型为图片，请注意测试服务单张图片输入应小于8MB。
- 图片支持以下类型：“png”、“psd”、“jpg”、“jpeg”、“bmp”、“gif”、“webp”、“psd”、“svg”、“tiff”。
- 若服务部署时使用的是“Ascend”规格，则无法预测含有透明度的PNG图片，因为Ascend仅支持RGB-3通道的图片。
- 该功能为调测使用，实际生产建议使用API调用。根据鉴权方式的不同，可以根据实际情况选择**访问在线服务（Token认证）**、**访问在线服务（AK/SK认证)**或者**访问在线服务（APP认证）**。

了解服务的输入参数

针对您部署上线的服务，您可以在服务详情页面的“调用指南”中，了解本服务的输入参数，即上文提到的输入请求类型。

图 10-5 查看服务的调用指南

调用指南中的输入参数取决于您选择的AI应用来源：

- 如果您的元模型来源于自动学习或预置算法，其输入输出参数由ModelArts官方定义，请直接参考“调用指南”中的说明，并在预测页签中输入对应的JSON文本或文件进行服务测试。
- 如果您的元模型是自定义的，即推理代码和配置文件是自行编写的（**配置文件编写说明**），“调用指南”只是将您编写的配置文件进行了可视化展示。调用指南的输入参数与配置文件对应关系如下所示。

图 10-6 配置文件与调用指南的对应关系

- 如果您的元模型是采用模型模板导入，不同的模板指定了其对应的输入输出模式，请参见**模型模板简介**的相关说明。

JSON 文本预测

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“部署上线>在线服务”，进入“在线服务”管理页面。
2. 单击目标服务名称，进入服务详情页面。在“预测”页签的预测代码下，输入预测代码，然后单击“预测”即可进行服务的预测，如**图10-7**所示，attr_7为需要目标列，predictioncol为目标列attr_7的预测结果。

   JSON文本类的预测代码和返回结果样例，可参见**银行存款预测样例**。此样例是使用自动学习功能训练的元模型，其输入类型为ModelArts官方定义，不可更改。

图 10-7 预测代码

请求路径： /

预测代码

```
{
  "data":
  {
    "count": 1,
    "req_data":
    [
      {
        "attr_1": "58",
        "attr_2": "management",
        "attr_3": "married",
        "attr_4": "tertiary",
        "attr_5": "yes",
        "attr_6": "no",
        "attr_7": ""
      }
    ]
  }
}
```

返回结果

```
{
    "result": {
        "count": 1,
        "resp_data": [
            {
                "probabilitycol": 1,
                "attr_6": "no",
                "attr_5": "yes",
                "attr_4": "tertiary",
                "attr_3": "married",
                "attr_2": "management",
                "attr_1": 58,
                "predictioncol": "yes"
            }
        ]
    }
}
```

预测

说明

输入数据中attr_7的值可任意填写，或为空，不会影响预测结果。

文件预测

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“部署上线>在线服务”，进入“在线服务”管理页面。
2. 单击目标服务名称，进入服务详情页面。在“预测”页签，单击“上传”，然后选择测试文件。文件上传成功后，单击“预测”即可进行服务的预测，如图10-8所示，输出标签名称，以及位置坐标和检测的评分。

   文件类的预测代码和返回结果样例，可参见花卉识别样例。此样例是使用预置算法训练的元模型，其输入类型为ModelArts官方定义，不可更改，如需自定义的元模型，请参见手写数字识别样例。

图 10-8 图片预测

请求路径： / 选择预测图片文件 上传 重新预测 难例反馈

预测图片预览

预测结果显示

预测成功

```
{
    "detection_classes": [
        "person",
        "person",
        "person",
        "mask",
        "mask",
        "face",
        "face",
        "face"
    ],
    "detection_boxes": [
        [
            358,
            791,
            482,
            900
        ],
        [
            431,
            366,
            566,
            505
```

10.2.4 访问在线服务（Token 认证）

若在线服务的状态处于“运行中”，则表示在线服务已部署成功，部署成功的在线服务，将为用户提供一个可调用的API，此API为标准Restful API。在集成至生产环境之前，需要对此API进行调测，您可以使用以下方式向在线服务发起预测请求：

- **方式一：使用图形界面的软件进行预测（以Postman为例）**。Windows系统建议使用Postman。
- **方式二：使用curl命令发送预测请求**。Linux系统建议使用curl命令。
- **方式三：使用python脚本发送预测请求**。

方式一：使用图形界面的软件进行预测（以 Postman 为例）

1. 下载Postman软件并安装，您也可以直接在Chrome浏览器添加Postman扩展程序（也可使用其它支持发送post请求的软件）。Postman推荐使用7.24.0版本。
2. 打开Postman，如**图10-9**所示。

图 10-9 Postman 界面

3. 在Postman界面填写参数，以图像分类举例说明。
   - 选择POST任务，将在线服务的调用地址（通过在线服务详情界面-调用指南页签查看）复制到POST后面的方框。Headers页签的Key值填写为“X-Auth-Token”，Value值为您获取到的Token。

     关于如何获取token，请参考**获取用户Token**。获取Token认证时，由于ModelArts生成的在线服务API不支持domain范围的token，因此需获取使用范围为project的Token信息，即scope参数的取值为project。

     说明

     您也可以通过AK（Access Key ID）/SK（Secret Access Key)加密调用请求，具体可参见**SDK参考>Session鉴权>用户AK-SK认证模式**。

     图 10-10 参数填写

   - 在Body页签，根据AI应用的输入参数不同，可分为2种类型：文件输入、文本输入。

- **文件输入**

  选择“form-data”。在“KEY”值填写AI应用的入参，比如本例中预测图片的参数为“images”。然后在“VALUE”值，选择文件，上传一张待预测图片（当前仅支持单张图片预测），如图10-11所示。

  图 10-11 填写 Body

- **文本输入**

  选择“raw”，选择JSON(application/json)类型，在下方文本框中填写请求体，请求体样例如下：

  ```
  {
    "meta": {
      "uuid": "10eb0091-887f-4839-9929-cbc884f1e20e"
    },
    "data": {
      "req_data": [
        {
          "sepal_length": 3,
          "sepal_width": 1,
          "petal_length": 2.2,
          "petal_width": 4
        }
      ]
    }
  }
  ```

  其中，“meta”中可携带“uuid”，调用时传入一个“uuid”，返回预测结果时回传此“uuid”用于跟踪请求，如无此需要可不填写meta。“data”包含了一个“req_data”的数组，可传入单条或多条请求数据，其中每个数据的参数由AI应用决定，比如本例中的“sepal_length”、“sepal_width”等。

4. 参数填写完成，单击“send”发送请求，结果会在“Response”下的对话框里显示。
   - 文件输入形式的预测结果样例如图10-12所示，返回结果的字段值根据不同AI应用可能有所不同。
   - 文本输入形式的预测结果样例如图10-13所示，请求体包含“meta”及“data”。如输入请求中包含“uuid”，则输出结果中回传此“uuid”。如未输入，则为空。“data”包含了一个“resp_data”的数组，返回单条或多条输入数据的预测结果,其中每个结果的参数由AI应用决定，比如本例中的“sepal_length”、“predictresult”等。

图 10-12 文件输入预测结果

图 10-13 文本输入预测结果

方式二：使用 curl 命令发送预测请求

使用curl命令发送预测请求的命令格式也分为文件输入、文本输入两类。

- 文件输入

```
curl -kv -F 'images=@图片路径' -H 'X-Auth-Token:Token值' -X POST 在线服务地址
```

  - “-k”是指允许不使用证书到SSL站点。
  - “-F”是指上传数据的是文件，本例中参数名为“images”，这个名字可以根据具体情况变化，@后面是图片的存储路径。
  - “-H”是post命令的headers，Headers的Key值为“X-Auth-Token”，这个名字为固定的， Token值是用户获取到的token值（关于如何获取token，请参见**获取请求认证**）。
  - “POST”后面跟随的是在线服务的调用地址。

  curl命令文件输入样例：

```
curl -kv -F 'images=@/home/data/test.png' -H 'X-Auth-Token:MIISkAY***80T9wHQ==' -X POST https://modelarts-infers-1.xxx/v1/infers/eb3e0c54-3dfa-4750-af0c-95c45e5d3e83
```

- 文本输入

```
curl -kv -d '{"data":{"req_data":[{"sepal_length":3,"sepal_width":1,"petal_length":2.2,"petal_width":4}]}}' -H 'X-Auth-Token:MIISkAY***80T9wHQ==' -H 'Content-type: application/json' -X POST https://modelarts-infers-1.xxx/v1/infers/eb3e0c54-3dfa-4750-af0c-95c45e5d3e83
```

  “-d”是Body体的文本内容。

方式三：使用 python 脚本发送预测请求

使用python脚本发送预测请求，也可以处理文件输入和文本输入。

- 以常用的图片文件输入为例：

```
# -*- coding:utf-8 -*-
import requests
import threading
import json

def inference(file_name, num):
  while num > 0:
    num -= 1

    try:
        method = "POST"
        headers = {
            "X-Auth-Token": "MIISkAY***80T9wHQ=="
        }
        url = "https://modelarts-infers-1.xxx/v1/infers/eb3e0c54-3dfa-4750-af0c-95c45e5d3e83"
        files = {
            'images': (file_name, open(file_name, 'rb'), "multipart/form-data")
        }
        resp = requests.request(method, url, headers=headers, files=files, verify=False)
        if 200 != int(resp.status_code):
            raise Exception(
                'inference failed. the status code is:' + str(resp.status_code) + str(resp.content) +
str(resp.headers))
    except Exception as e:
        raise e

if __name__ == '__main__':
    thread = threading.Thread()
    predict = threading.Thread(target=inference, args=("./test.png", 20))  #预测
    predict.start()
    predict.join()  #等待线程结束
```

  - “X-Auth-Token”后面填写的是token值，如何获取token请参见**获取请求认证**。
  - “url”后面填写的是在线服务的调用地址。
  - “args”后面填写需要输入的图片以及执行多少次后停止预测。

- 以文本输入为例：

```
    body = {
        "input": xxx
    }
    data = json.dumps(body)
    resp = requests.request(method, url, data=data, headers=headers, verify=False)
```

“input”后面填写文本内容。

10.2.5 访问在线服务（AK/SK 认证）

若在线服务的状态处于“运行中”，则表示在线服务已部署成功。部署成功的在线服务，将为用户提供一个可调用的API，此API为标准Restful API。用户可以通过AK/SK签名认证方式调用API。

使用AK/SK认证时，您可以通过APIG SDK 访问，也可以通过ModelArts SDK 访问。使用ModelArts SDK 访问参见**用户AK-SK认证模式**。本文档详细介绍如何通过APIG SDK 访问在线服务，具体操作流程如下：

1. **获取AK/SK**
2. **获取URI**
3. **以Python语言为例准备环境**
4. **调用API示例**

说明

1. AK/SK签名认证方式，仅支持Body体12M以内，12M以上的请求，需使用Token认证。
2. 客户端须注意本地时间与时钟服务器的同步，避免请求消息头X-Sdk-Date的值出现较大误差
   API网关除了校验时间格式外，还会校验该时间值与网关收到请求的时间差，如果时间差超过15分钟，API网关将拒绝请求。

获取 AK/SK

如果已生成过AK/SK，则可跳过此步骤，找到原来已下载的AK/SK文件，文件名一般为：credentials.csv。

如下图所示，文件包含了租户名（User Name），AK（Access Key Id），SK（Secret Access Key）。

图 10-14 credential.csv 文件内容

	A	B	C
1	User Name	Access Key Id	Secret Access Key
2	hu[redacted]dg	QTWA[redacted]UT2QVKYUC	MFyfvK41ba2[redacted]npdUKGpownRZlmVmHc

AK/SK生成步骤：

1. 注册并登录管理控制台。
2. 单击右上角的用户名，在下拉列表中单击“我的凭证”。
3. 单击“访问密钥”。
4. 单击“新增访问密钥”，进入“身份验证”页面。

图 10-15 访问密钥获取页面示意

5. 输入验证码，单击“确定”，下载密钥，请妥善保管。

获取 URI

在调用接口时，需获取API接口地址。步骤如下：

1. 登录ModelArts管理控制台，在左侧导航栏中选择“部署上线 > 在线服务”，默认进入“在线服务”列表。
2. 单击目标服务名称，进入服务详情页面。
3. 在“在线服务”的详情页面，可以获取该服务的API接口。

图 10-16 获取 API 接口

以 Python 语言为例准备环境

此章节内容以Python语言为例介绍APIG SDK使用方式，其他语言的SDK下载与使用示例请参见《API网关 APIG开发指南》。

1. 从Python官网获取并安装Python安装包（支持使用2.7.9或3.X版本）。Python安装完成后，您可以执行命令**pip install requests**，通过Python通用包管理工具pip安装“requests”库。

   说明

   如果使用pip安装requests库遇到证书错误，请下载并使用Python执行此文件，升级pip，然后再执行以上命令安装。

2. 从IntelliJ IDEA官网获取并安装IntelliJ IDEA。在IntelliJ IDEA中安装Python插件， 如图10-17所示。

图 10-17 安装 Python 插件

3. 获取SDK。下载SDK，获取“ApiGateway-python-sdk.zip”压缩包，解压后目录结构如下：

表 10-8 SDK 压缩包解压目录

名称	说明
apig_sdk__init__.py	SDK代码
apig_sdk\signer.py	
main.py	示例代码
backend_signature.py	后端签名示例代码
licenses\license-requests	第三方库license文件

4. 新建工程。
   a. 打开IDEA，选择菜单“File > New > Project”。在弹出的“New Project”对话框中选择“Python”，单击“Next”。

图 10-18 新建工程

   b. 再次单击“Next”，弹出以下对话框。单击“...”，在弹出的对话框中选择解压后的SDK路径，单击“Finish”完成工程创建。

图 10-19 选择解压后的 SDK 路径

5. 完成工程创建后，目录结构如下。其中“main.py”为示例代码，请根据实际情况修改参数后使用。

图 10-20 新建工程的目录结构

调用 API 示例

1. 在工程中引入“apig_sdk”。
```
from apig_sdk import signer
import requests
```
2. 生成一个新的Signer，填入“AK”和“SK”，获取方式请参见**获取AK/SK**。
```
sig = signer.Signer()
sig.Key = "UATBQ1PQO1D5ORNVCDAA"
sig.Secret = "ap6H7L******0QvHCNk"
```
3. 生成一个Request对象，指定方法名、请求uri、header和body。
```
r= signer.HttpRequest(method, uri, header, body)
```

表 10-9 HttpRequest 参数说明

参数	子参数	是否必填	说明
method	-	是	可填“GET”、“POST”、“PUT”、“DELETE”
uri	-	是	填入在线服务的API接口，获取接口方式参见**获取URI**。
header	x-stage	是	接口发布环境，必填，当前仅支持“RELEASE”。
	Content-Type	否	内容类型，当前支持“application/json”。“multipart/form-data”形式请求体请参考**表10-10**。
	x-sdk-content-sha256	否	签名方式，可填“UNSIGNED-PAYLOAD”，表示不对body进行签名认证。 当body为文件时，该参数必填。
body	-	否	支持json格式，示例："{\"xxx\":\"xxx\"}"

a. 当请求体为json格式

```
r = signer.HttpRequest("POST",
                    "https://
1684994b180244de9d141c00d3e52c73.apig.exampleRegion.huaweicloudapis.com/v1/infers/
exampleServiceId",
                    {"x-stage": "RELEASE","Content-Type":"application/json"},"{\"xxx\":\"xxx\"}")
```

b. 当请求体为文件格式，您需要构造multipart/form-data形式的请求体。

请求体样式：*files={"请求参数":("文件路径"，文件内容，“文件类型”，请求头)}*

表 10-10 files 参数

参数	说明
请求参数	在线服务输入参数名称。
文件路径	上传文件的路径。
文件内容	上传文件的内容。
文件类型	上传文件类型。当前支持以下类型： ● txt类型：text/plain ● jpg/jpeg类型：image/jpeg ● png类型：image/png
请求头	建议填“{}”，请求头在“HttpRequest”参数“header”中填入。

如果您访问请求参数为images在线服务，示例如下。

图 10-21 访问在线服务

```
r = signer.HttpRequest("POST","https://63fb035aeef34368880448a94cb7f440.apig.cn-north-4.huaweicloudapis.com/v1/infers/76c41384-23ab-45f9-a66e-892e7bc2be53", {"x-stage": "RELEASE", "x-sdk-content-sha256": "UNSIGNED-PAYLOAD"})
files = {"images": ("flower.png", open("flower.png", "rb"), "image/png", {})}
```

4. 进行签名，执行此函数会在请求参数中添加用于签名的“X-Sdk-Date”头和“Authorization”头。

```
sig.Sign(r)
```

5. 调用API，查看访问结果。

```
resp = requests.request(method,url, headers, data, files)
```

表 10-11 request 参数说明

参数	说明
method	填入签名后Request对象的请求方法。
url	填入签名后Request对象的请求地址。
headers	填入签名后Request对象的headers。
data	填入Request对象的body请求体，仅支持Json格式。
files	填入multipart/form-data形式的请求体。

a. 当请求体为json格式

```
resp = requests.request(r.method, r.scheme + "://" + r.host + r.uri, headers=r.headers, data=r.body)
print(resp.status_code, resp.reason)
print(resp.content)
```

b. 当请求体为图片格式

```
resp = requests.request(r.method, r.scheme + "://" + r.host + r.uri, headers=r.headers, data={}, files=files)
print(resp.status_code, resp.reason)
print(resp.content)
```

10.2.6 访问在线服务（APP 认证）

部署在线服务支持开启APP认证，即ModelArts会为服务注册一个支持APP认证的接口，为此接口配置APP授权后，用户可以使用授权应用的AppKey+AppSecret或AppCode调用该接口。

针对在线服务的APP认证，具体操作流程如下。

- **开启支持APP认证功能**：开启支持APP认证并创建应用。
- **在线服务授权管理**：对创建的应用进行管理，包括查看、重置或删除应用，绑定或解绑应用对应的在线服务，获取“AppKey/AppSecret”或“AppCode”。
- **APP认证鉴权**：调用支持APP认证的接口需要进行认证鉴权，支持两种鉴权方式，您可以选择其中一种进行认证鉴权。
- **以Python语言为例准备环境**：在APP认证鉴权完成之后，您可以准备调用环境进行接口调用。
- **调用API示例**：使用授权应用信息调用该接口进行预测。

前提条件

- 数据已完成准备：已在ModelArts中创建状态“正常”可用的AI应用。
- 由于在线运行需消耗资源，确保帐户未欠费。

开启支持 APP 认证功能

在部署为在线服务时，您可以开启支持APP认证功能。或者针对已部署完成的在线服务，您可以修改服务，开启支持APP认证功能。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“部署上线 > 在线服务”，进入在线服务管理页面。
2. 开启支持APP认证功能。
   - 在部署为在线服务时，即“部署”页面，填写部署服务相关参数时，开启支持APP认证功能。
   - 针对已部署完成的在线服务，进入在线服务管理页面，单击目标服务名称“操作”列的“修改”按钮，进入修改服务页面开启支持APP认证功能。

   图 10-22 部署页面开启支持 APP 认证功能

3. 选择APP授权配置。从下拉列表中选择您需要配置的APP应用，如果没有可选项，您可以通过如下方式创建应用。
   - 单击右侧“创建应用”，填写应用名称和描述之后单击“确定”完成创建。其中应用名称默认以“app_”开头，您也可以自行修改。
   - 进入“部署上线>在线服务”页面，单击“授权管理”，进入“在线服务授权管理”页面，选择“创建应用”，详请参见**在线服务授权管理**。
4. 开启支持APP认证功能后，将支持APP认证的服务授权给应用，用户可以使用创建的“AppKey/AppSecret”或“AppCode”调用服务的支持APP认证的接口。

在线服务授权管理

如果您需要使用支持APP认证功能，建议您在部署在线服务之前进行授权管理操作完成应用创建。进入“部署上线>在线服务”页面，单击“授权管理”，进入“在线服务授权管理”页面。在此页面您可以实现应用的创建和管理，包括查看、重置或删除应用，解绑应用对应的在线服务，获取“AppKey/AppSecret”或“AppCode”。

图 10-23 在线服务授权管理

- **创建应用**

  选择“创建应用”，填写应用名称和描述之后单击“确定”完成创建。其中应用名称默认以“app_”开头，您也可以自行修改。

- **查看、重置或删除应用**

  您可以单击目标应用名称操作列的按钮完成应用的查看、重置或删除。创建完成后自动生成“AppKey/AppSecret”以供您后续调取接口进行APP鉴权使用。

- **解绑服务**

  您可以单击目标应用名称前方的∨，在下拉列表中展示绑定的服务列表，即该应用对应的在线服务列表。单击操作列的“解绑”取消绑定，将不再支持调用该接口。

- **获取AppKey/AppSecret或AppCode**

  调用接口需要进行APP鉴权，在创建应用时自动生成“AppKey/AppSecret”，“AppCode”也可以通过单击“+添加AppCode”自动生成。

  图 10-24 添加 AppCode

APP 认证鉴权

当支持APP认证功能的在线服务运行成功处于“运行中”状态，就可以对服务进行调用 。在调用之前您需要进行APP认证鉴权。

调用支持APP认证的接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。推荐使用AppKey/AppSecret认证，其安全性比AppCode认证要高。

- AppCode认证：通过AppCode认证通用请求。
- AppKey/AppSecret认证：通过AppKey（APP访问密钥ID）与AppSecret（APP私有访问密钥），对请求进行加密签名，可标识发送方，并防止请求被修改。

您可以在服务详情页的“调用指南”页签（如**图10-25**）或者在线服务授权管理页面（如**图10-23**）获取API接口和AppKey/AppSecret和AppCode。请注意使用图中红框所示的API接口地址。

图 10-25 获取 API 的接口地址

AppCode认证

当使用APP认证，且开启了简易认证模式，API请求既可以选择使用Appkey和AppSecret做签名和校验，也可以选择使用AppCode进行简易认证。ModelArts默认启用简易认证。

AppCode认证就是在调用API的时候，在HTTP请求头部消息增加一个参数“X-Apig-AppCode”（参数值为“AppCode”），而不需要对请求内容签名，API网关也仅校验AppCode，不校验请求签名，从而实现快速响应。示例代码如下：

```
import requests
url = 'https://e27faf2cc66a42fbad2dce18747962e5.apig.cn-north-4.huaweicloudapis.com/v1/infers/xxxxxx/goodbye'
headers={'Content-Type':'application/json',
         'X-Apig-Appcode':'dd19xxxxxx'}
response=requests.request('GET', url, headers=headers)
print(response.text)
```

说明

- 当APP不支持AppCode时，可使用AppKey进行简易认证，即在调用API的时候，在HTTP请求头部消息增加一个参数“apikey”（参数值为“AppKey”），实现快速认证。
- 使用AppCode认证访问的更多方式可参考**访问在线服务（Token认证）**。

AppKey/AppSecret认证

AppKey/AppSecret认证就是通过应用的AppKey和AppSecret进行签名认证。

- AppKey：APP访问密钥ID。与私有访问密钥关联的唯一标识符；访问密钥ID和私有访问密钥一起使用，对请求进行加密签名。
- AppSecret：与访问密钥ID结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。

使用AppKey/AppSecret认证时，您需要使用专门的签名SDK对请求进行签名。

以 Python 语言为例准备环境

在APP认证鉴权完成之后，您可以准备调用环境进行接口调用。此章节内容以Python语言为例介绍SDK使用方式，其他语言的SDK下载与使用示例请参见《API网关 APIG开发指南》。

1. 获取API的接口地址、请求方法、AppKey和AppSecret。获取方式请参见APP认证鉴权。
2. 从Python官网获取并安装Python安装包（支持使用2.7.9或3.X版本）。Python安装完成后，您可以执行命令**pip install requests**，通过Python通用包管理工具pip安装“requests”库。

   说明

   如果使用pip安装requests库遇到证书错误，请下载并使用Python执行此文件，升级pip，然后再执行以上命令安装。

3. 从IntelliJ IDEA官网获取并安装IntelliJ IDEA。在IntelliJ IDEA中安装Python插件， 如图10-26所示。

   图 10-26 安装 Python 插件

4. 获取SDK。下载SDK，获取“ApiGateway-python-sdk.zip”压缩包，解压后目录结构如下：

   表 10-12 SDK 压缩包解压目录

名称	说明
apig_sdk__init__.py	SDK代码
apig_sdk\signer.py	
main.py	示例代码
backend_signature.py	后端签名示例代码
licenses\license-requests	第三方库license文件

5. 新建工程。
   a. 打开IDEA，选择菜单“File > New > Project”。在弹出的“New Project”对话框中选择“Python”，单击“Next”。

图 10-27 新建工程

   b. 再次单击“Next”，弹出以下对话框。单击“...”，在弹出的对话框中选择解压后的SDK路径，单击“Finish”完成工程创建。

图 10-28 选择解压后的 SDK 路径

6. 完成工程创建后，目录结构如下。其中“main.py”为示例代码，请根据实际情况修改参数后使用。

图 10-29 新建工程的目录结构

调用 API 示例

1. 在工程中引入“apig_sdk”。
```
from apig_sdk import signer
import requests
```
2. 生成一个新的Signer，填入“AppKey”和“AppSecret”，获取方式请参见**APP认证鉴权**。
```
sig = signer.Signer()
sig.Key = "4f5f626b-073f-402f-a1e0-e52171c6100c"
sig.Secret = "******"
```
3. 生成一个Request对象，指定方法名、请求uri、header和body。
```
r= signer.HttpRequest(method, uri, header, body)
```

表 10-13 HttpRequest 参数说明

参数	子参数	是否必填	说明
method	-	是	可填“GET”、“POST”、“PUT”、“DELETE”
uri	-	是	填入在线服务的API接口，获取接口方式参见**APP认证鉴权**
header	x-stage	是	接口发布环境，必填，当前仅支持“RELEASE”。
	Content-Type	否	内容类型，当前支持“application/json”。“multipart/form-data”形式请求体请参考**表10-14**。
	x-sdk-content-sha256	否	签名方式，可填“UNSIGNED-PAYLOAD”，表示不对body进行签名认证。 当body为文件时，该参数必填。

参数	子参数	是否必填	说明
body	-	否	支持json格式，示例 ： "{\"xxx\":\"xxx\"}"。

a. 当请求体为json格式

```
r = signer.HttpRequest("POST",
                       "https://
1684994b180244de9d141c00d3e52c73.apig.exampleRegion.huaweicloudapis.com/v1/infers/
exampleServiceId",
                       {"x-stage": "RELEASE","Content-Type":"application/json"},"{\"xxx\":\"xxx\"}")
```

b. 当请求体为图片格式，您需要构造multipart/form-data形式的请求体。

请求体样式： *files={"请求参数":("文件路径"，文件内容，“文件类型”，请求头)}*

表 10-14 files 参数

参数	说明
请求参数	在线服务输入参数名称。
文件路径	上传文件的路径。
文件内容	上传文件的内容。
文件类型	上传文件类型。当前支持以下类型： ● txt类型：text/plain ● jpg/jpeg类型：image/jpeg ● png类型：image/png
请求头	建议填“{}”，请求头在“HttpRequest”参数“header”中填入。

如果您访问请求参数为images，输入格式为file类型的在线服务，示例如下。

图 10-30 访问在线服务

```
r = signer.HttpRequest("POST","https://63fb035aeef34368880448a94cb7f440.apig.cn-
north-4.huaweicloudapis.com/v1/infers/76c41384-23ab-45f9-a66e-892e7bc2be53",
{"x-stage": "RELEASE", "x-sdk-content-sha256": "UNSIGNED-PAYLOAD"})
files = {"images": ("flower.png", open("flower.png", "rb"), "image/png", {})}
```

4. 进行签名，执行此函数会在请求参数中添加用于签名的“X-Sdk-Date”头和“Authorization”头。

```
sig.Sign(r)
```

5. 调用API，查看访问结果。

```
resp = requests.request(method,url, headers, data, files)
```

表 10-15 request 参数说明

参数	说明
method	填入签名后Request对象的请求方法。
url	填入签名后Request对象的请求地址。
headers	填入签名后Request对象的headers。
data	填入Request对象的body请求体，仅支持Json格式。
files	填入multipart/form-data形式的请求体。

a. 当请求体为json格式

```
resp = requests.request(r.method, r.scheme + "://" + r.host + r.uri, headers=r.headers, data=r.body)
print(resp.status_code, resp.reason)
print(resp.content)
```

b. 当请求体为图片格式

```
resp = requests.request(r.method, r.scheme + "://" + r.host + r.uri, headers=r.headers, data={},
files=files)
print(resp.status_code, resp.reason)
print(resp.content)
```

10.2.7 集成在线服务

针对已完成调测的API，可以将在线服务API集成至生产环境中应用。

前提条件

确保在线服务一直处于“运行中”状态，否则会导致生产环境应用不可用。

集成方式

ModelArts在线服务提供的API是一个标准的Restful API，可使用HTTPS协议访问。ModelArts提供了SDK用于调用在线服务API，SDK调用方式请参见《**SDK参考**》>“场景1：部署在线服务Predictor的推理预测”。

除此之外，您还可以使用常见的开发工具及开发语言调用此接口，本文不再赘述，建议通过互联网搜索并获取调用标准Restful API的指导。

10.3 批量服务

10.3.1 部署为批量服务

AI应用准备完成后，您可以将AI应用部署为批量服务。在“部署上线>批量服务”界面，列举了用户所创建的批量服务。您可以在右上方搜索框中输入服务名称，单击 进行查询。

前提条件

- 数据已完成准备：已在ModelArts中创建状态“正常”可用的AI应用。
- 准备好需要批量处理的数据，并上传至OBS目录。
- 已在OBS创建至少1个空的文件夹，用于存储输出的内容。

背景信息

- 用户最多可创建1000个批量服务。
- 根据AI应用定义的输入请求不同（JSON文本或文件），不同的AI应用输入，需要填写的参数不同。当AI应用输入为JSON文件时，则需要根据配置文件生成映射文件；如果AI应用输入为文件时，则不需要。
- 批量服务只支持使用公共资源池，暂不支持使用专属资源池。

操作步骤

1. 登录ModelArts管理控制台，在左侧导航栏中选择“部署上线 > 批量服务”，默认进入“批量服务”列表。
2. 在批量服务列表中，单击左上角“部署”，进入“部署”页面。
3. 在部署页面，填写批量服务相关参数。
   a. 填写基本信息。基本信息包含“名称”、“描述”。其中“名称”默认生成。例如：service-bc0d，您也可以根据实际情况填写“名称”和“描述”信息。
   b. 填写服务参数。包含资源池、AI应用配置等关键信息，详情请参见**表10-16**。

表 10-16 参数说明

参数名称	说明
“AI应用来源”	根据您的实际情况选择“我的AI应用”或者“我的订阅”。
“选择AI应用及版本”	选择状态“正常”的AI应用及版本。
“输入数据目录位置”	选择输入数据的OBS路径，即您上传数据的OBS目录。只能选择文件夹或“.manifest”文件。“.manifest”文件规范请参见**Manifest文件规范**。 **说明** ● 输入数据为图片时，建议单张图片小于10MB。 ● 输入数据格式为csv时，建议不要包含中文。如需使用中文，请将文件编码格式设置为UTF-8编码。您可以使用代码方式转换文件编码格式，也可以将csv文件用记事本方式打开，在另存为弹出的窗口页面设置编码格式。
“请求路径”	批量服务中调用AI应用的接口URl，表示服务的请求路径，此值来自AI应用配置文件中apis的url字段。
“映射关系”	如果AI应用输入是json格式时，系统将根据此AI应用对应的配置文件自动生成映射关系。如果AI应用的输入是文件，则不需要映射关系。 自动生成的映射关系文件，填写每个参数对应到csv单行数据的字段索引，索引index从0开始计数。 映射关系生成规则：映射规则来源于模型配置文件“config.json”中输入参数（request）。当“type”定义为“string/number/integer/boolean”基本类型时，需要配置映射规则参数，即index参数。请参见**映射关系示例**了解其规则。 index必须从0开始的正整数，当index设置不符合规则时,最终的请求将忽略此参数。配置映射规则后，其对应的csv数据必须以英文半角逗号分隔。
“输出数据目录位置”	选择批量预测结果的保存位置，可以选择您创建的空文件夹。

参数名称	说明
“计算节点规格”	系统将根据您的AI应用匹配提供可用的计算资源。请下拉框中选择可用资源，如果资源标识为售罄，表示此暂无此资源。 例如，模型来源于自动学习项目，则计算资源将自动关联自动学习规格供使用。
“计算节点个数”	设置当前版本AI应用的实例个数。如果节点个数设置为1，表示后台的计算模式是单机模式；如果节点个数设置大于1，表示后台的计算模式为分布式的。请根据实际编码情况选择计算模式。
“环境变量”	设置环境变量，注入环境变量到容器实例。为确保您的数据安全，在环境变量中，请勿输入敏感信息，如明文密码。

4. 完成参数填写后，根据界面提示完成批量服务的部署。部署服务一般需要运行一段时间，根据您选择的数据量和资源不同，部署时间将耗时几分钟到几十分钟不等。

   说明

   批量服务部署完成后，将立即启动，运行过程中将按照您选择的资源按需计费。

   您可以前往批量服务列表，查看批量服务的基本情况。在批量服务列表中，刚部署的服务“状态”为“部署中”，当批量服务的“状态”变为“运行完成”时，表示服务部署完成。

Manifest 文件规范

推理平台批量服务支持使用manifest文件，manifest文件可用于描述数据的输入输出。

输入manifest文件样例

- 文件名：“test.manifest”
- 文件内容：

```
{"source": "<obs path>/test/data/1.jpg"}
{"source": "https://infers-data.obs.cn-north-1.myhuaweicloud.com:443/xgboosterdata/data.csv?AccessKeyId=2Q0V0TQ461N26DDL18RB&Expires=1550611914&Signature=wZBttZj5QZrReDhz1uDzwve8GpY%3D&x-obs-security-token=gQpzb3V0aGNoaW5hixvY8V9a1SnsxmGoHYmB1SArYMyqnQT-ZaMSxHvl68kKLAy5feYvLDM...”}
```

- 文件要求：
  a. 文件名后缀需为“.manifest”；
  b. 文件内容是多行JSON，每行JSON描述一个输入数据，需精确到文件，不能是文件夹；
  c. JSON内容需定义一个source字段，字段值是OBS的文件地址，有2种表达形式：
     i. “<obs path>/{{桶名}}/{{对象名}}”，适用于访问自己名下的OBS数据；您可以访问OBS服务的对象获取路径。

图 10-31 数据的 OBS 路径

ii. OBS生成的分享链接，包含签名信息。适用于访问其他人的OBS数据。

输出manifest文件样例

批量服务的输出结果目录会有一个manifest文件。

- 假设用户输出结果路径为//test-bucket/test/，则结果存放位置如下：

```
OBS桶/目录名
├─── test-bucket
│   ├─── test
│   │   ├─── infer-result-{{index}}.manifest
│   │   ├─── infer-result
│   │   │   ├─── 1.jpg_result.txt
│   │   │   ├─── 2.jpg_result.txt
```

- infer-result-0.manifest文件内容：

```
{"source": "<obs path>/obs-data-bucket/test/data/1.jpg","inference-loc": "<obs path>/test-bucket/test/infer-result/1.jpg_result.txt"}
{"source ": "https://infers-data.obs.cn-north-1.myhuaweicloud.com:443/xgboosterdata/2.jpg?AccessKeyId=2Q0V0TQ461N26DDL18RB&Expires=1550611914&Signature=wZBttZj5QZrReDhz1uDzwve8GpY%3D&x-obs-security-token=gQpzb3V0aGNoaW5hixvY8V9a1SnsxmGoHYmB1SArYMyqnQT-ZaMSxHvl68kKLAy5feYvLDMNZWxzhBZ6Q-3HcoZMh9gISwQOVBwm4ZytB_m8sg1fL6isU7T3CnoL9jmvDGgT9VBC7dC1EyfSJrUcqfB...",  "inference-loc": "obs://test-bucket/test/infer-result/2.jpg_result.txt"}
```

- 文件格式：
  a. 文件名为“infer-result-{{index}}.manifest”，index为实例序号，批量服务运行多少个实例就会产生多少个manifest文件。
  b. manifest同一目录下会创建infer-result目录存放文件处理结果。
  c. 文件内容是多行JSON，每行JSON描述一个输入数据的对应输出结果。
  d. JSON内容包含多个字段。
     i. source：输入数据描述，与输入的manifest一致。
     ii. inference-loc：输出结果路径，格式为“<obs path>/{{桶名}}/{{对象名}}”。

映射关系示例

如下示例展示了配置文件、映射规则、csv数据以及最终推理请求的关系。

假设，您的模型所用配置文件，其apis参数如下所示：

```
[
  {
    "protocol": "http",
    "method": "post",
    "url": "/",
    "request": {
      "type": "object",
      "properties": {
        "data": {
          "type": "object",
          "properties": {
            "req_data": {
```

```
                    "type": "array",
                    "items": [
                        {
                            "type": "object",
                            "properties": {
                                "input_1": {
                                    "type": "number"
                                },
                                "input_2": {
                                    "type": "number"
                                },
                                "input_3": {
                                    "type": "number"
                                },
                                "input_4": {
                                    "type": "number"
                                }
                            }
                        }
                    ]
                }
            }
        }
    }
}
]
```

此时，其对应的映射关系如下所示。ModelArts管理控制台将从配置文件中自动解析映射关系，如果您调用ModelArts API时，需要自行根据规则编写映射关系。

```
{
    "type": "object",
    "properties": {
        "data": {
            "type": "object",
            "properties": {
                "req_data": {
                    "type": "array",
                    "items": [
                        {
                            "type": "object",
                            "properties": {
                                "input_1": {
                                    "type": "number",
                                    "index": 0
                                },
                                "input_2": {
                                    "type": "number",
                                    "index": 1
                                },
                                "input_3": {
                                    "type": "number",
                                    "index": 2
                                },
                                "input_4": {
                                    "type": "number",
                                    "index": 3
                                }
                            }
                        }
                    ]
                }
            }
        }
    }
}
```

用户需要进行推理的数据，即CSV数据，格式如下所示。数据必须以英文逗号隔开。

```
5.1,3.5,1.4,0.2
4.9,3.0,1.4,0.2
4.7,3.2,1.3,0.2
```

根据定义好的映射关系，最终推理请求样例如下所示，与在线服务使用的格式类似：

```
{
    "data": {
        "req_data": [{
            "input_1": 5.1,
            "input_2": 3.5,
            "input_3": 1.4,
            "input_4": 0.2
        }]
    }
}
```

10.3.2 查看批量服务预测结果

当您在部署批量服务时，会选择输出数据目录位置，您可以查看“运行完成”状态的批量服务运行结果。

操作步骤

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“部署上线>批量服务”，进入“批量服务”管理页面。
2. 单击状态为“运行完成”的目标服务名称，进入服务详情页面。
   - 您可以查看服务的“名称”、“状态”、“服务ID”、“输入数据目录位置”、“输出数据目录位置”和“描述”。
   - 您也可以通过单击描述右侧的 ，对描述信息进行编辑。
3. 从“输出数据目录位置”参数右侧获取详细OBS地址，前往此OBS目录，可以获取批量服务预测结果，包括预测结果文件和AI应用预测结果。

   若预测成功，目录下有预测结果文件和AI应用预测结果；若预测失败，目录下只有预测结果文件。
   - 预测结果文件：文件格式为“xxx.manifest”，里面包含文件路径、预测结果等信息。
   - AI应用预测结果输出：
     - 当输入为图片时，每张图片输出一个结果，输出结果格式为“图片名_result.txt”。例如：IMG_20180919_115016.jpg_result.txt。
     - 当输入为音频时，每个音频输出一个结果，输出结果格式为“音频名_result.txt”。例如：1-36929-A-47.wav_result.txt。
     - 当输入为表格数据时，输出结果格式为“表格名_result.txt”。例如：train.csv_result.txt。

10.4 边缘服务

10.4.1 部署为边缘服务

AI应用准备完成后，您可以将AI应用部署为边缘服务。在“部署上线>边缘服务”界面，列举了用户所创建的边缘服务。您可以在右上方搜索框中输入服务名称，单击

进行查询。边缘服务依赖智能边缘平台（IEF），部署前需要在智能边缘平台上创建边缘节点。

前提条件

- 数据已完成准备：已在ModelArts中创建状态“正常”可用的AI应用。
- 已在IEF上创建边缘节点。如果您未创建边缘节点，具体操作请参见创建边缘节点。
- 由于在线运行需消耗资源，确保帐户未欠费。

背景信息

- 边缘服务目前还处于限时免费阶段，运行中的边缘服务，并不会产生费用。
- 用户最多可创建1000个边缘服务。

部署边缘服务

1. 登录ModelArts管理控制台，在左侧导航栏中选择“部署上线>边缘服务”，默认进入“边缘服务”列表。
2. 在边缘服务列表中，单击左上角“部署”，进入“部署”页面。
3. 在部署页面，填写边缘服务相关参数。
   a. 填写基本信息。基本信息包含“名称”、“描述”。其中“名称”默认生成。例如：service-bc0d，您也可以根据实际情况填写“名称”和“描述”信息。
   b. 填写服务参数。包含资源池、AI应用配置等关键信息，详情请参见表10-17。

表 10-17 参数说明

参数名称	说明
“AI应用来源”	根据您的实际情况选择“我的AI应用”或者“我的订阅”。
“选择AI应用及版本”	选择状态“正常”的AI应用及版本。
“计算节点规格”	支持如下几种规格。 ● “CPU：2核 8GiB”：适合纯CPU类型的负载运行的AI应用。 ● “CPU：2核 8GiB GPU：1*P4”：适合CPU+GPU类型AI应用的运行，带有1个Nvidia P4卡。 ● “自定义规格”，如果选择自定义规格，可以在参数下方设置您所需的“CPU”、“内存配额”、“GPU”或“Ascend”。其中，“GPU”和“Ascend”只能二选一。

参数名称	说明
“环境变量”	设置环境变量，注入环境变量到容器实例。为确保您的数据安全，在环境变量中，请勿输入敏感信息，如明文密码。 默认设置外部接口协议为https，用户可以通过修改“MODELARTS_SSL_ENABLED”环境变量设置接口协议为http。 MODELARTS_SSL_ENABLED = false
“部署方式”	可选择“节点”或“节点组”。 ● 如果您在IEF创建的是边缘节点，则选择“节点”。IEF相关说明请参见**边缘节点**。 ● 如果您在IEF创建的是铂金版实例和边缘节点组，则选择“节点组”。需指定对应的铂金版“资源实例”和“部署实例个数”。IEF相关说明请参见**边缘节点组**。
“选择边缘节点”	边缘节点是您自己的边缘计算设备，用于运行边缘应用，处理您的数据，并安全、便捷地和云端应用进行协同。 单击选择边缘节点“添加”，在弹出的“添加节点”对话框中选择节点。选择您已创建的节点后，单击“确定”。

4. 完成参数填写后，根据界面提示完成边缘服务的部署。部署服务一般需要运行一段时间，根据您选择的数据量和资源不同，部署时间将耗时几分钟到几十分钟不等。

   您可以前往边缘服务列表，查看边缘服务的基本情况。在边缘服务列表中，刚部署的服务“状态”为“部署中”，当边缘服务的“状态”变为“运行中”时，表示服务部署完成。

部署边缘服务（Atlas 500）

如果您纳管至IEF的设备是Atlas 500智能小站，则需要将训练好的模型部署至Atlas 500设备中。在开始操作之前，您需要了解如下几点要求。

- 对AI应用的要求：仅支持om模型或tflite模型，即支持部署在Ascend或ARM资源的模型。针对不满足格式的模型，必须经过模型转换操作，将模型转换成对应格式。模型转换操作和限制，请参见**模型转换操作**。
- 关于固件升级：如果您使用AI Gallery中新版预置算法训练得到的模型。当前此算法仅适配C32固件,不支持低级版本和C7X版本。因此将此模型部署至Atlas 500设备时，需下载并升级固件，请参考《Atlas500产品C32固件升级操作指导》升级Atlas 500设备。如果您使用部署的模型适配Atlas 500原有的固件，则无需升级固件。
- 固件下载升级仅适用于Atlas 500。
- 对于预置算法训练所得的模型，要求使用AI Gallery中的预置算法，且算法支持Ascend 310进行推理。

将AI应用部署至Atlas 500，请参考如下步骤进行操作。

1. 登录ModelArts管理控制台，在左侧导航栏中选择“部署上线>边缘服务”，默认进入“边缘服务”列表。
2. 在边缘服务列表中，单击左上角“部署”，进入“部署”页面。
3. 在部署页面，填写边缘服务相关参数，然后单击“下一步”。
   a. 填写基本信息。基本信息包含“名称”、“描述”。其中“名称”默认生成，建议根据实际业务填写有意义的名称。
   b. 填写边缘服务参数，详情请参见表10-18。

表 10-18 部署至 Atlas 500 的参数说明

参数名称	说明
“AI应用来源”	根据您的实际情况选择“我的AI应用”或者“我的订阅”。
“选择AI应用及版本”	从下拉列表中选择可用的AI应用及版本。 **说明** 选择的模型是om或tflite格式的，即经过模型转换，然后使用“ARM-Ascend模板”导入至ModelArts创建为AI应用。
“计算节点规格”	选择符合要求的AI应用后，计算节点规格默认支持如下两种： ● ARM：3 核 3 GiB Ascend: 1 * Ascend 310 ● 自定义规格：可自行设置CPU、内存和Ascend个数。由于Atlas 500只有1个Ascend，设置为Ascend后，数量需设置为1。
“环境变量”	设置环境变量，注入环境变量到容器实例。为确保您的数据安全，在环境变量中，请勿输入敏感信息，如明文密码。
“部署方式”	选择“节点”。
“选择边缘节点”	边缘节点是您自己的边缘计算设备，用于运行边缘应用，处理您的数据，并安全、便捷地和云端应用进行协同。 单击选择边缘节点右侧的“添加”，在弹出的“添加节点”对话框中，选择在IEF中纳管的Atlas 500节点，然后单击“确定”。 ModelArts系统会进行自动识别和匹配，如果纳管的设备未升级至符合要求的固件，则需根据界面提示完成C32固件升级。反之，则不需要进行固件升级。

图 10-32 选择 AI 应用及边缘节点

4. （可选）Atlas 500设备升级C32固件。
   a. 如图10-32提示，在节点列表下方单击“升级C32固件”，在弹出的对话框中，仔细阅读升级说明，勾选“我已阅读并同意以上内容”，然后单击“下载”，将固件版本及升级指导下载至本地。文件名称为“atlas500_C32_Firmware.zip”。
   b. 解压“atlas500_C32_Firmware.zip”文件，打开Atlas500产品C32固件升级操作指导.doc文件，根据指导完成Atlas 500的固件升级操作。
   c. 待Atlas 500升级后，重新部署边缘服务。

      刷新ModelArts管理控制台页面，根据1~3步骤，重新填写部署边缘服务的信息，此时选择升级后的Atlas 500，则不会再出现升级提示。
5. 完成参数填写后，单击“立即创建”，完成边缘服务的部署。部署服务一般需要运行一段时间，根据您选择的数据量和资源不同，部署时间将耗时几分钟到几十分钟不等。

   您可以前往边缘服务列表，查看边缘服务的基本情况。在边缘服务列表中，刚部署的服务“状态”为“部署中”，当边缘服务的“状态”变为“运行中”时，表示服务部署完成。部署完成后，您可以登录Atlas 500查看部署完成的应用。

10.4.2 访问边缘服务

访问边缘服务

当边缘服务和边缘节点的状态都处于“运行中”状态，表示边缘服务已在边缘节点成功部署。

您可以通过以下两种方式，在能够访问到边缘节点的网络环境中，对部署在边缘节点上的边缘服务发起预测请求。

- **方式一：使用图形界面的软件进行预测（以Postman为例）**
- **方式二：使用curl命令发送预测请求**

说明

通过方式一和方式二，边缘服务无需安全认证即可访问。如果用户需要设置安全认证，请选择使用自定义镜像创建的AI应用，具体操作请参考**官方指导**。在制作自定义镜像时，可参考**Flask文档**使用Flask设计RESTful的认证。

方式一：使用图形界面的软件进行预测（以 Postman 为例）

1. 下载**Postman**软件并安装，您可以直接在Chrome浏览器添加Postman扩展程序（也可使用其它支持发送post请求的软件）。
2. 打开Postman，如**图10-33**所示。

图 10-33 Postman 软件界面

3. 在Postman界面填写参数，以图像分类举例说明。
   - 选择POST任务，将某个边缘节点的调用地址（通过边缘服务详情界面-节点信息页签查看）复制到POST后面的方框。

     图 10-34 POST 参数填写

   - 在Body页签，根据AI应用的输入参数不同，可分为“文件输入”或“文本输入”。
     - **文件输入**

       选择“form-data”。在“KEY”值填写AI应用的入参，比如本例中预测图片的参数为“images”。然后在“VALUE”值，选择文件，上传一张待预测图片（当前仅支持单张图片预测）。

       图 10-35 填写 Body 配置

- **文本输入**

  选择“raw”，选择JSON(application/json)类型，在下方文本框中填写请求体，请求体样例如下。

```
{
"meta": {
"uuid": "10eb0091-887f-4839-9929-cbc884f1e20e"
},
"data": {
"req_data": [
{
"sepal_length": 3,
"sepal_width": 1,
"petal_length": 2.2,
"petal_width": 4
}
]
}
}
```

  其中，“meta”中可携带“uuid”，返回预测结果时回传此“uuid”用于跟踪请求，**如无此需要可不填写meta。**“data”包含了一个“req_data”的数组，可传入单条或多条请求数据，其中每个数据的参数由AI应用决定，比如本例中的“sepal_length”、“sepal_width”等。

4. 参数填写完成，点击“Send”发送请求，结果会在Response下的对话框里显示。
   - 文件输入形式的预测结果样例如图10-36所示，返回结果的字段值根据不同AI应用可能有所不同。

图 10-36 边缘服务文件输入预测结果

   - 文本输入形式的预测结果样例如图10-37所示，请求体包含“meta”及“data”。如输入请求中包含“uuid”，则输出结果中回传此“uuid”。如

未输入，则为空。“data”包含了一个“req_data”的数组，可传入单条或多条请求数据，其中每个数据的参数由AI应用决定，比如本例中的“sepal_length”、“sepal_width”等。

图 10-37 边缘服务文本输入预测结果

方式二：使用 curl 命令发送预测请求

使用curl命令发送预测请求的命令格式也分为文件输入、文本输入两类

1. 文件输入

```
curl -F 'images=@图片路径'-X POST 边缘节点服务地址 -k
```

   - “-F”是指上传数据的是文件，本例中参数名为**images**，这个名字可以根据具体情况变化，@后面是图片的存储路径。
   - “POST”后面跟随的是边缘节点的调用地址。

   curl命令文件输入预测样例：

```
curl -F 'images=@/home/data/cat.jpg' -X POST https://192.168.0.158:1032 -k
```

   预测结果如图10-38所示。

   图 10-38 curl 命令文件输入预测结果

2. 文本输入

```
curl -d '{
"meta": {
"uuid": "10eb0091-887f-4839-9929-cbc884f1e20e"
},
```

```
"data": {
"req_data": [
{
"sepal_length": 3,
"sepal_width": 1,
"petal_length": 2.2,
"petal_width": 4
}
]
}
}' -X POST <边缘节点服务地址> -k
```

– “-d”是Body体的文本内容，如AI应用为文本输入，则需要用此参数。

curl命令文本输入预测样例：

```
curl -d '{
"meta": {
"uuid": "10eb0091-887f-4839-9929-cbc884f1e20e"
},
"data": {
"req_data": [
{
"sepal_length": 3,
"sepal_width": 1,
"petal_length": 2.2,
"petal_width": 4
}
]
}
}' -X POST https://192.168.0.158:1033 -k
```

预测结果如图10-39所示。

图 10-39 curl 命令文本输入预测结果

```
root@modelarts006:/# curl -X POST \
> https://192.168.0.158:1033/ \
>   -d '{
>   "meta": {
>     "uuid": "10eb0091-887f-4839-9929-cbc884f1e20e"
>   },
>   "data": {
>     "req_data": [
>       {
>         "sepal_length": 3,
>         "sepal_width": 1,
>         "petal_length": 2.2,
>         "petal_width": 4
>       }
>     ]
>   }
> }
> '  -k
{"meta": {"uuid": "10eb0091-887f-4839-9929-cbc884f1e20e"}, "data": {"res_data": [{"sepal_length": 3, "sepal_width": 1, "
petal_length": 2.2, "petal_width": 4, "predictresult": 0}]}}root@modelarts006:/#
```

10.5 升级服务

对于已部署的服务，您可以修改服务的基本信息以匹配业务变化，实现服务升级。您可以通过如下两种方式修改服务的基本信息：

方式一：通过服务管理页面修改服务信息

方式二：通过服务详情页面修改服务信息

前提条件

已存在部署完成的服务。

约束限制

- 服务升级关系着业务实现，不当的升级操作会导致升级期间业务中断的情况，请谨慎操作。
- ModelArts支持部分场景下在线服务进行无损滚动升级。按要求进行升级前准备，做好验证，即可实现业务不中断的无损升级。

表 10-19 支持无损滚动升级的场景

创建AI应用的元模型来源	服务使用的是公共资源池	服务使用的是专属资源池
从训练中选择元模型	不支持	不支持
从模板中选择元模型	不支持	不支持
从容器镜像中选择元模型	不支持	支持，创建AI应用的自定义镜像需要满足**创建AI应用的自定义镜像规范**。
从OBS中选择元模型	不支持	不支持

方式一：通过服务管理页面修改服务信息

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“部署上线”，进入目标服务类型管理页面。
2. 在服务列表中，单击目标服务操作列的“修改”，修改服务基本信息，然后单击“确定”完成修改。
   - 在线服务参数说明请参见**部署为在线服务**。
   - 批量服务参数说明请参见**部署为批量服务**。
   - 边缘服务参数说明请参见**部署为边缘服务**。

   说明

   “部署中”状态的服务无法进行修改。

方式二：通过服务详情页面修改服务信息

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“部署上线”，进入目标服务类型管理页面。
2. 单击目标服务名称，进入服务详情页面。
3. 您可以通过单击页面右上角“修改”，修改服务基本信息，然后单击“确定”完成修改。
   - 在线服务参数说明请参见**部署为在线服务**。
   - 批量服务参数说明请参见**部署为批量服务**。
   - 边缘服务参数说明请参见**部署为边缘服务**。

10.6 启动或停止服务

启动服务

您可以对处于“运行完成”、“异常”和“停止”状态的服务进行启动操作，“部署中”状态的服务无法启动。启动服务，当服务处于“运行中”状态后，ModelArts将开始计费。您可以通过如下方式启动服务：

- 登录ModelArts管理控制台，在左侧菜单栏中选择“部署上线”，进入目标服务类型管理页面。您可以单击“操作”列的“启动”（在线服务在操作列选择“更多 > 启动”），启动服务。
- 登录ModelArts管理控制台，在左侧菜单栏中选择“部署上线”，进入目标服务类型管理页面。单击目标服务名称，进入服务详情页面。您可以单击页面右上角“启动”，启动服务。

停止服务

停止服务，ModelArts将停止计费。您可以通过如下方式停止服务：

- 登录ModelArts管理控制台，在左侧菜单栏中选择“部署上线”，进入目标服务类型管理页面。您可以单击“操作”列的“停止”（在线服务在操作列选择“更多 > 停止”），停止服务。
- 登录ModelArts管理控制台，在左侧菜单栏中选择“部署上线”，进入目标服务类型管理页面。单击目标服务名称，进入服务详情页面。您可以单击页面右上角“停止”，停止正在运行中服务。

10.7 删除服务

如果服务不再使用，您可以删除服务释放资源。

登录ModelArts管理控制台，在左侧菜单栏中选择“部署上线”，进入目标服务类型管理页面。

- 在线服务，您可以单击“操作”列的“更多>删除”删除服务。
- 批量服务和边缘服务，

说明

删除操作无法恢复，请谨慎操作。

11 资源池

ModelArts 资源池说明

在使用ModelArts进行AI全流程开发时，您可以选择使用两种不同的资源池。

- **公共资源池**：公共资源池提供公共的大规模计算集群，根据用户作业参数分配使用，资源按作业隔离。按资源规格、使用时长及实例数计费，不区分任务（训练作业、部署、开发）。公共资源池是ModelArts默认提供，不需另行创建或配置，您可以直接在AI开发过程中，直接选择公共资源池进行使用。
- **专属资源池**：提供独享的计算资源，可用于Notebook、训练作业、部署模型。专属资源池不与其他用户共享，更加高效。

  在使用专属资源池之前，您需要先购买一个专属资源池，然后在AI开发过程中选择此专属资源池。专属资源池的详细介绍和操作请参见：

  专属资源池介绍

  创建专属资源池

  扩缩容专属资源池

  删除专属资源池

专属资源池介绍

- 专属资源池可以在如下作业和任务中使用：Notebook、训练作业、TensorBoard、部署上线。
- 专属资源池分为“开发环境/训练专用”和“部署上线专用”两种类型。“开发环境/训练专用”类型的专属资源池只能用于Notebook、训练作业、TensorBoard等功能，“部署上线专用”类型的专属资源池只能用于AI应用的部署上线。
- 只有处于“运行中”状态的专属资源池才是可用的。如果专属资源池状态为“不可用”或“异常”，请排除故障后再使用。
- 创建专属资源池后，就会基于选择的规格开始计费。
- 专属资源池的收费支持“按需计费”和“包年包月”两种。

创建专属资源池

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“专属资源池”。
2. 在专属资源池管理页面，您可以选择通过“开发环境/训练专用”和“部署上线专用”页签选择两种不同类型的专属资源池。

3. 单击左上角“创建”，进入创建专属资源池界面。
4. 在“创建专属资源池”界面填写参数，参数填写请参见表11-1和表11-2。

表 11-1 “开发环境/训练专用”专属资源池的参数说明

参数名称	说明
资源类型	系统默认为“开发环境/训练专用”， 不可修改。
计费模式	选择计费模式，“包年/包月”或“按需计费”。
名称	专属资源池的名称。 名称由大小写字母、数字、中划线和下划线组成。
描述	专属资源池的简要描述。
节点数	选择专属资源池的节点数，选择的节点数越多，计算性能越强，同时费用越高。
节点规格	请根据界面提示选择需要使用的规格，GPU性能更好，CPU更加实惠。如果某一规格出现“售罄”字样，表示此规格已用完，需要等待其他用户删除资源池后，您才可以重新购买。
购买时长	● 选择购买时长。只有选择“包年/包月”计费模式时才需填写。 最少为1个月，最长为1年。其中，ModelArts推出套餐包优惠，购买10个月赠送2个月，直接选择1年的购买时长即可完成买10个月送2个月的套餐包购买。 ● 自动续费。开通自动续费后，系统将在产品到期前自动续费，无需用户手动操作。

表 11-2 “部署上线专用”专属资源池的参数说明

参数名称	说明
资源类型	系统默认为“部署上线专用”， 不可修改。
计费模式	选择计费模式，“包年/包月”或“按需计费”。“包年/包月”仅在北京四支持。
名称	专属资源池的名称。 名称只能以小写字母开头，由小写字母、数字、中划线组成，不能以中划线结尾，长度为4~24个字符。
描述	专属资源池的简要描述。

参数名称	说明
自定义网络配置	启用自定义配置，则服务实例运行在指定的网络中，可以与该网络中的其它云服务资源实例互通；不启用自定义配置，ModelArts会为每个用户分配一个专属的网络，用户之间隔离。 如果启用自定义网络配置，请设置对应的“虚拟私有云”、“子网”和“安全组”。如果没有可用网络，请前往虚拟私有云服务创建。
可用区	您可以根据实际情况选择“随机分配”、“可用区1”、“可用区2”、“可用区3”。可用区是在同一区域下，电力、网络隔离的物理区域。可用区之间内网互通，不同可用区之间物理隔离。如果您需要提高工作负载的高可靠性，建议您将云服务器创建在不同的可用区。
节点数	选择专属资源池的节点数，选择的节点数越多，计算性能越强，同时费用越高。
节点规格	请根据界面提示选择需要使用的规格，GPU性能更好，CPU更加实惠。如果某一规格出现“售罄”字样，表示此规格已用完，需要等待其他用户删除资源池后，您才可以重新购买。
购买时长	● 选择购买时长。只有选择“包年/包月”计费模式时才需填写。最少为1个月，最长为11个月。 ● 自动续费。开通自动续费后，系统将在产品到期前自动续费，无需用户手动操作。

5. 规格确认无误后，根据界面提示完成专属资源池的创建。当专属资源池创建成功后，其状态将变为“运行中”。

扩缩容专属资源池

当专属资源池使用一段时间后，由于AI开发业务的变化，您可以通过扩容或缩容操作，增加或减少节点数量。

扩缩容的操作步骤如下所示：

1. 进入专属资源池管理页面，在专属资源池所在行，单击操作列“扩缩容”。
2. 在扩缩容页面，增加或减少节点数量。增加节点数量表示扩容，减少节点数量表示缩容。请根据本身业务诉求进行调整。
   - 扩容时，请务必选择当前帐号的配额，增加节点数量，否则会导致扩容失败。
   - 缩容时，您需要在操作列单击开关删除减少的节点。如图11-1所示，减少1个节点，需在“节点列表”中，单击删除节点对应操作列的开关，删除此节点。

注意

部署上线类资源池用于运行推理部署在线服务，缩容节点时请务必保证该节点上无运行实例，否则可能影响到您的线上业务。如果不确定待缩容节点上是否有实例在运行，请提咨询单咨询。

图 11-1 缩容时选择删除节点

3. 单击“提交”完成修改。提交完成后系统自动返回专属资源池管理页面。

说明

缩容节点时，提交缩容操作后，删除节点操作不是立即成功的，还需要后台做处理，此时请勿去专属资源池列表中删除节点，可能会导致删除失败。

提交缩容操作后，可以在专属资源池的详情页中查看事件列表。Begin to delete resource node %s表示开始删除节点；当出现事件Resource node %s deleted，表示缩容节点在后台删除成功。

专属资源池计费模式转换

- “按需计费”的专属资源池转换为“包年包月”的专属资源池

  ModelArts支持训练的专属资源池从“按需计费”转换为“包年包月”。前提是您需要使用按需计费的专属资源池完成一次以上的消费。如果购买的按需计费专属资源池没有消费记录，转周期操作会报错。

  转包周期的操作步骤如下所示：

  a. 进入专属资源池管理页面，在专属资源池所在行，单击操作列“转包周期”，进入“按需转包年按月”页面。

  b. 根据页面提示选择包年包月的时长并勾选是否自动续费，完成支付后即可转换成功。

- “包年包月”的专属资源池转换为“按需计费”的专属资源池

  针对用于训练的“包年包月”专属资源池，支持在包周期到期后，将专属资源池设置为按需计费。

  转按需的操作步骤如下：

  a. 进入ModelArts控制台，单击上方设置栏的“费用中心>续费管理”，进入续费管理页面。

图 11-2 进入费用中心

b. 进入续费管理页面，在专属资源池所在行，单击操作列“更多>到期转按需”。

删除专属资源池

当AI业务开发不再需要使用专属资源池时，您可以删除专属资源池，释放资源。

说明

- 专属资源池删除后，将导致使用此资源的训练作业、Notebook、在线服务和批量服务等不可用，且删除后不可恢复，请谨慎操作。

1. 进入专属资源池管理页面，释放资源。
   - 如果是“包年/包月”类型的专属资源池，请单击操作列“退订”完成订单退费，退订完成后将自动删除该资源。
   - 如果是“按需”计费类型的专属资源池，单击操作列的“删除”即可。
   - 对于创建失败的专属资源池，请单击操作列“删除”完成资源删除。
2. 在弹出的确认对话框中，单击“确定”，完成资源删除。

12 使用自定义镜像

12.1 自定义镜像简介

ModelArts为用户提供了多种常见的预置引擎，但是当用户对深度学习引擎、开发库有特殊需求场景的时候，预置AI引擎已经不能满足用户需求。ModelArts提供自定义镜像功能支持用户自定义运行引擎。

ModelArts底层采用容器技术，自定义镜像指的是用户自行制作容器镜像并在ModelArts上运行。自定义镜像功能支持自由文本形式的命令行参数和环境变量，灵活性比较高，便于支持任意计算引擎的作业启动需求。

关联服务介绍

使用自定义镜像功能可能涉及以下云服务：容器镜像服务、对象存储服务、弹性云服务器。

- 容器镜像服务

  容器镜像服务（Software Repository for Container，SWR）是一种支持镜像全生命周期管理的服务， 提供简单易用、安全可靠的镜像管理功能，帮助您快速部署容器化服务。您可以通过界面、社区CLI和原生API上传、下载和管理容器镜像。

  ModelArts训练和创建AI应用使用的自定义镜像需要从SWR服务管理列表获取。您制作的自定义镜像需要上传至SWR服务。

  图 12-1 获取镜像列表

- 对象存储服务

  对象存储服务（Object Storage Service，OBS）是一个基于对象的海量存储服务，为客户提供海量、安全、高可靠、低成本的数据存储能力。

  在创建训练作业和创建AI应用时往往存在数据交互，您需要的数据可以存储至OBS服务。

- 弹性云服务器

  弹性云服务器（Elastic Cloud Server，ECS）是由CPU、内存、操作系统、云硬盘组成的基础的计算组件。弹性云服务器创建成功后，您就可以像使用自己的本地PC或物理服务器一样，在云上使用弹性云服务器。

  在制作自定义镜像时，您可以在本地环境或者ECS上完成自定义镜像制作。

说明

在您使用自定义镜像功能时，ModelArts可能需要访问您的容器镜像服务SWR、对象存储服务OBS等依赖服务，若没有授权，这些功能将不能正常使用。建议您使用委托授权功能，将依赖服务操作权限委托给ModelArts服务，让ModelArts以您的身份使用依赖服务，代替您进行一些资源操作。详细操作参见使用委托授权。

自定义镜像使用场景

- **用于训练模型**

  如果您已经在本地完成模型开发或训练脚本的开发，且您使用的AI引擎是ModelArts不支持的框架。您可以基于ModelArts提供的基础镜像包制作自定义镜像，并上传至SWR服务。您可以在ModelArts使用此自定义镜像创建训练作业，使用ModelArts提供的资源训练模型。

- **用于创建AI应用**

  如果您使用了ModelArts不支持的AI引擎开发模型，也可通过制作自定义镜像，导入ModelArts创建为AI应用，并支持进行统一管理和部署为服务。

自定义镜像的制作流程

1. 购买弹性云服务器或者应用本地主机搭建Docker环境。
2. 在本地环境拉取基础镜像。
3. 根据您的实际需求编写Dockerfile文件构建自定义镜像。如何高效编写Dockerfile指导可参考**SWR服务最佳实践**。
   - 如果您使用自定义镜像用于训练作业请参考示例**训练作业自定义镜像规范**。
   - 如果您使用自定义镜像用于创建AI应用请参考示例**创建AI应用的自定义镜像规范**。
4. 当完成自定义镜像制作后，请参考**上传镜像至容器镜像服务**将镜像上传到自己的SWR中。

12.2 制作和上传自定义镜像

ModelArts支持您使用自定义镜像创建训练作业和创建AI应用。在制作和上传自定义镜像之前，您需要了解以下内容：

- SWR服务。

  容器镜像服务（Software Repository for Container，简称SWR）是一种支持镜像全生命周期管理的服务， 提供简单易用、安全可靠的镜像管理功能，帮助您快速部署容器化服务。您可以通过界面、社区CLI和原生API上传、下载和管理容器镜像。

  ModelArts训练和创建AI应用使用的自定义镜像需要从SWR服务管理列表获取。您制作的自定义镜像需要上传至SWR服务。

- 自定义镜像规范。如果您使用自定义镜像用于训练作业请参考**训练作业自定义镜像规范**；如果您使用自定义镜像用于创建AI应用请参考**创建AI应用的自定义镜像规范**。

制作并上传自定义镜像

1. 购买华为云ECS或者应用本地主机搭建Docker环境。
2. 在本地环境拉取基础镜像。
3. 根据您的实际需求编写Dockerfile文件构建自定义镜像。如何高效编写Dockerfile指导可参考**SWR服务最佳实践**。
   - 如果您使用自定义镜像用于训练作业请参考示例**制作自定义镜像**。
   - 如果您使用自定义镜像用于创建AI应用请参考示例**本地构建镜像**。
4. 当完成自定义镜像制作后，请参考**上传镜像至容器镜像服务**将镜像上传到自己的SWR中。

12.3 用于训练模型（旧版即将下线）

12.3.1 训练作业自定义镜像规范

说明

本章节介绍的是基于旧版训练的自定义镜像训练模型，旧版训练仅对部分存量用户可见，新用户不可见，新用户推荐使用**新版训练功能**。

针对您本地开发的模型及训练脚本，在制作镜像时，需满足ModelArts定义的规范。

规范要求

- 自定义镜像中不能包含恶意代码。
- 基础镜像中的部分内容不能改变，包括“/bin”、“/sbin”、“/usr”、“/lib(64)”下的所有文件，“/etc”下的部分重要配置文件，以及“$HOME”下的ModelArts小工具。
- 不可以新增属主为“root”且权限包含“setuid”或“setgid”位的文件。
- 自定义镜像大小不能超过9.5GB。
- 日志文件输出，为保证日志内容可以正常显示，日志信息需要打印到标准输出。
- 自定义镜像的默认用户必须为“uid”为“1101”的用户。
- 自定义镜像可以基于ModelArts官方提供的基础镜像制作，基础镜像请参考**基础镜像包概述**。

基础镜像包概述

为了方便用户实现代码下载、训练日志输出、上传日志文件至OBS等功能，ModelArts提供基础镜像包用于自定义镜像的制作。ModelArts提供的基础镜像有以下特点：

- 基础镜像中有一些必要的工具，用户需要基于ModelArts官方提供的基础镜像来制作自定义镜像。
- ModelArts将持续更新基础镜像版本，基础镜像更新后，对于兼容性更新，用户还可以继续使用旧的镜像；对于不兼容性更新，基于旧版本制作的自定义镜像将不能在ModelArts上运行，但已经审核过的自定义镜像可以继续使用。

- 当用户发现自定义镜像审核不通过，并且审核日志中出现基础镜像不匹配的错误信息时，需要使用新的基础镜像重新制作镜像。

您可以通过以下命令获取ModelArts镜像：

```
docker pull <基础镜像地址>
```

完成自定义镜像制作后，执行以下命令推送镜像至SWR服务。前提条件是已经创建组织并获取SWR登录指令，可参考上传镜像至SWR服务。

```
docker push  swr.<region>.myhuaweicloud.com/<用户镜像所属组织>/<镜像名称>
```

根据芯片不同的需求，您可以根据实际需求获取不同的基础镜像：

- 适用于CPU的基础镜像
- 适用于GPU的基础镜像
- 适用于Ascend芯片基础镜像

适用于 CPU 的基础镜像

基础镜像地址

```
swr.<region>.myhuaweicloud.com/modelarts-job-dev-image/custom-cpu-base:1.3
```

表 12-1 可选参数范围

参数	可选值	说明
<region>	• cn-north-1 • cn-north-4	镜像所在的区域。支持的值中，分别表示： • 北京一 • 北京四

基础镜像包含表12-2和表12-3

表 12-2 组件列表

名称	说明
run_train.sh	训练启动引导脚本。实现了代码目录下载，执行训练命令、重定向训练日志输出、以及训练命令结束后上传日志文件至OBS的功能。

表 12-3 工具列表

工具名称	说明
utils.sh	工具脚本。“run_train.sh”脚本依赖此脚本。 提供了SK解密，代码目录下载，日志文件上传等方法。

工具名称	说明
ip_mapper.py	网卡地址获取脚本。 默认获取ib0网卡地址IP，训练代码可以使用ib0网卡的IP加速网络通信。
dls-downloader.py	OBS下载脚本。“utils.sh”脚本依赖此脚本。

适用于 GPU 的基础镜像

- cuda10.0/10.1/10.2版本的镜像，以ubuntu18.04为基础镜像默认预装moxing
```
swr.<region>.myhuaweicloud.com/modelarts-job-dev-image/custom-base-<cuda version>-<python version>-<os>-<arch>:<image tag>
```
- cuda8/9/92版本的镜像，默认预装Moxing
```
swr.<region>.myhuaweicloud.com/modelarts-job-dev-image/custom-gpu-<cuda version>-inner-moxing-<python version>:<image tag>
```
- cuda8/9/92版本的镜像
```
swr.<region>.myhuaweicloud.com/modelarts-job-dev-image/custom-gpu-<cuda version>-base:<image tag>
```

表 12-4 名称可选参数范围

参数	支持的值	说明
<region>	• cn-north-1 • cn-north-4	镜像所在的区域。支持的值中，分别表示： • 北京一 • 北京四
<cuda version>	• cuda92 • cuda9 • cuda8 • cuda10.0 • cuda10.1 • cuda10.2	镜像中已安装的CUDA版本。 **说明** 请您确认对应的CUDA版本，版本指定之后不支持更换， 否则会导致训练失败。
<image tag>	• 1.1 • 1.3	镜像版本。 • cuda8/9/92版本的镜像，支持1.3。 • cuda10.0/10.1/10.2版本的镜像支持1.1。
python version	• cp27 • cp36	python 环境。
os	ubuntu18.04	操作系统。
arch	x86	架构。

例如，在“华北-北京一”区域，ModelArts支持的基础镜像列表如下，您可根据个人需求选择相应的镜像。

cuda10之前的版本：

- swr.cn-north-1.myhuaweicloud.com/modelarts-job-dev-image/custom-cpu-base:1.3
- swr.cn-north-1.myhuaweicloud.com/modelarts-job-dev-image/custom-gpu-cuda92-base:1.3
- swr.cn-north-1.myhuaweicloud.com/modelarts-job-dev-image/custom-gpu-cuda9-base:1.3
- swr.cn-north-1.myhuaweicloud.com/modelarts-job-dev-image/custom-gpu-cuda8-base:1.3
- swr.cn-north-1.myhuaweicloud.com/modelarts-job-dev-image/custom-gpu-cuda9-inner-moxing-cp36:1.3
- swr.cn-north-1.myhuaweicloud.com/modelarts-job-dev-image/custom-gpu-cuda8-inner-moxing-cp27:1.3
- swr.cn-north-1.myhuaweicloud.com/modelarts-job-dev-image/custom-gpu-cuda9-inner-moxing-cp27:1.3
- ...

cuda10之后的版本：

- swr.cn-north-1.myhuaweicloud.com/modelarts-job-dev-image/custom-base-cuda10.0-cp36-ubuntu18.04-x86:1.1
- swr.cn-north-1.myhuaweicloud.com/modelarts-job-dev-image/custom-base-cuda10.1-cp36-ubuntu18.04-x86:1.1
- swr.cn-north-1.myhuaweicloud.com/modelarts-job-dev-image/custom-base-cuda10.2-cp36-ubuntu18.04-x86:1.1

基础镜像包含**表12-2**和**表12-3**

表 12-5 组件列表

名称	说明
run_train.sh	训练启动引导脚本。实现了代码目录下载，执行训练命令、重定向训练日志输出、以及训练命令结束后上传日志文件至OBS的功能。

表 12-6 工具列表

工具名称	说明
utils.sh	工具脚本。“run_train.sh”脚本依赖此脚本。 提供了SK解密，代码目录下载，日志文件上传等方法。

工具名称	说明
ip_mapper.py	网卡地址获取脚本。 默认获取ib0网卡地址IP，训练代码可以使用ib0网卡的IP加速网络通信。
dls-downloader.py	OBS下载脚本。“utils.sh”脚本依赖此脚本。

适用于 Ascend 芯片基础镜像

基础镜像地址

```
swr.cn-north-4.myhuaweicloud.com/modelarts-job-dev-image/mindspore-ascend910-cp37-euleros2.8-aarch64-training:1.3.0-3.3.0-roma
swr.cn-north-4.myhuaweicloud.com/modelarts-job-dev-image/tensorflow-ascend910-cp37-euleros2.8-aarch64-training:1.15.0-3.3.0-roma
```

基础镜像包含的组件、工具如表12-7和表12-8所示

表 12-7 组件列表

名称	说明
run_train.sh	可自动完成OBS代码路径下载至本地、Ascend HCCL RANK_TABLE_FILE v0.1 格式转 v1.0 格式、多卡训练进程拉起功能。

表 12-8 工具列表

工具名称	说明
utils.sh	工具脚本。“run_train.sh”脚本依赖此脚本。 提供了SK解密，代码目录下载，日志文件上传等方法。
modelarts-downloader.py	OBS下载脚本。“utils.sh”脚本依赖此脚本。

12.3.2 使用自定义镜像创建训练作业（GPU）

说明

基于自定义镜像训练模型仅适用于旧版训练模块（仅对部分存量用户可见，新用户不可见），新版训练请参见使用自定义镜像训练模型（新版训练）。

用户将自定义镜像制作完成并上传至SWR后，可在ModelArts管理控制台，使用自定义镜像创建训练作业，完成模型训练操作。

前提条件

- 已按照ModelArts规范制作自定义镜像包，使用自定义镜像创建训练作业需遵守的规范请参见**训练作业自定义镜像规范**。
- 已将自定义镜像上传至SWR服务，详细操作指导请参见**制作并上传自定义镜像**。

创建训练作业

进入ModelArts管理控制台，创建训练作业。在使用自定义镜像创建作业时，需关注“算法来源”、“环境变量”和“资源池”参数的设置。

- **“算法来源”**

  选择“自定义”页签。

  – “镜像地址”：镜像上传到SWR后生成的地址。

  图 12-2 SWR 镜像地址

  – “代码目录”：训练代码文件存储的OBS路径。

  – “运行命令”：镜像启动后的运行命令，基本格式如下所示：

  bash /home/work/run_train.sh {UserCommand}

  bash /home/work/run_train.sh [python/bash/..] {file_location} {file_parameter}

  “run_train.sh”为训练启动引导脚本。执行该脚本后，ModelArts将“代码目录”下的所有内容递归下载到容器本地路径，下载后的容器本地路径为“/home/work/user-job-dir/${“代码目录”的最后一级名称}/”。

  例如，训练代码文件的OBS路径为“obs://obs-bucket/new/train.py”，“代码目录”选择“obs://obs-bucket/new/”时，则下载后的容器本地代码目录对应为“/home/work/user-job-dir/new/”。下载后的容器本地训练代码路径为“/home/work/user-job-dir/new/train.py”，运行命令可以设置为：**bash /home/work/run_train.sh python /home/work/user-job-dir/new/train.py {python_file_parameter}**

  说明

  使用自定义镜像创建训练作业过程中，ModelArts 允许用户完全自定义“运行命令”。“运行命令”中提到以下两种基本格式：

  bash /home/work/run_train.sh {UserCommand}

  bash /home/work/run_train.sh [python/bash/..] {file_location} {file_parameter}

  其中，“run_train.sh，”为训练启动引导脚本。用户在制作自定义镜像过程中，可以自主实现训练启动引导脚本，也可以提前将训练代码置于自定义镜像环境中，无需遵循上述两种格式，实现完全的自定义“运行命令”。

- **“环境变量”**

  容器启动后，除了用户在训练作业中自行增加的“环境变量”外，其它加载的环境变量如表12-9所示。用户可以根据需求来确认在自己训练脚本的python中是否要使用这些环境变量，也可以通过运行命令中的“{python_file_parameter}”传入相关参数。

  表 12-9 可选环境变量说明

环境变量	说明
DLS_TASK_INDEX	当前容器索引，容器从0开始编号。
DLS_TASK_NUMBER	容器总数。对应“计算节点个数”。
DLS_APP_URL	代码目录。对应界面上“代码目录”配置，会加上协议名。比如，可直接使用“$DLS_APP_URL/*.py”来读取OBS下的文件。
DLS_DATA_URL	数据集位置。对应界面上“数据来源”，会加上协议名。
DLS_TRAIN_URL	训练输出位置。对应界面上“训练输出位置”，会加上协议名。
BATCH_{jobName}.0_HOSTS（单机）	当选择单机时，即计算节点个数为1时，此环境变量为“BATCH_{jobName}.0_HOSTS”。 HOSTS环境变量的格式为“hostname:port”。一个容器可以看到同一个作业中所有容器的HOSTS，根据索引的不同，分别为“BATCH_CUSTOM0_HOSTS”、“BATCH_CUSTOM1_HOSTS”等。当资源池为8GPU规格的专属资源池时， 容器的网络类型为主机网络，并且可以使用主机IB网络加速通信。当使用其他资源池时为容器网络。 **说明** 使用主机IB网络加速通信时，IPoIB特性需要ip_mapper.py工具获取ib0网卡IP地址。

- **“资源池”**

  当用户选择GPU类型的资源池时，ModelArts会挂载高速固态硬盘（NVME SSD）至“/cache”目录，用户可以使用此目录来储存临时文件。

图 12-3 创建训练作业

运行自定义镜像训练作业

用户上传自定义镜像到SWR后，在创建自定义镜像作业时，默认已经授权ModelArts去获取镜像运行。自定义审核镜像第一次运行的时候，先审核镜像，审核内容请参见**训练作业自定义镜像规范**，审核失败的原因见于日志，用户根据日志做相应的修改。

图 12-4 审核镜像失败

镜像审核成功后，后台就会开始启动用户自定义镜像容器，开始跑自定义镜像训练作业，用户可根据日志来查看训练情况。

说明

审核成功后，再次使用相同镜像创建训练作业的时候，不会再次审核。

图 12-5 运行日志

12.3.3 使用自定义镜像训练模型（Ascend）

如果 Ascend-Powered-Engine 常用框架无法满足您的需求，您可以将算法构建为一个自定义镜像，通过自定义镜像创建训练作业。

说明

基于自定义镜像训练模型仅适用于旧版训练模块（仅对部分存量用户可见，新用户不可见），新版训练请参见**使用自定义镜像训练模型（新版训练）**。

前提条件

- 数据已完成准备：已在ModelArts中创建可用的数据集，或者您已将用于训练的数据集上传至OBS目录。
- 如果“算法来源”为“自定义”，请完成镜像制作，自定义镜像制作规范参见**训练作业自定义镜像规范**。
- 制作完成的自定义镜像需上传至SWR服务，请参考“《容器镜像服务用户指南》>镜像管理>客户端上传镜像”。
- 已在OBS创建至少1个空的文件夹，用于存储训练输出的内容。
- 确保您使用的OBS目录与ModelArts在同一区域。

注意事项

- 训练作业指定的数据集目录中，用于训练的数据名称（如图片名称、音频文件名、标注文件名称等），名称长度限制为0～255英文字符。如果数据集目录下，部分数据的文件名称超过255英文字符，训练作业将不会使用此数据，使用符合要求的数据继续进行训练。如果数据集目录下，所有数据的文件名称都超过了255英文字符，导致训练作业无数据可用，则会最终导致训练作业失败。
- 训练脚本中，“数据来源”、“训练输出位置”，两个参数必须为OBS路径。当需要对路径中进行读写交互时，建议使用**MoXing接口**进行读写操作。

创建训练作业

进入ModelArts管理控制台，参考**创建训练作业**操作指导，创建训练作业。在使用自定义镜像创建作业时，需关注“算法来源”、“环境变量”和“资源池”参数的设置。

您需要特别关注以下作业参数的设置：

- “镜像地址”

  镜像上传到SWR后生成的地址。

  图 12-6 SWR 镜像地址

- “运行命令”

  若使用 ModelArts Ascend 基础镜像，运行命令参考如下：

  ```
  /bin/bash /home/work/run_train.sh ${obs-code-path} ${the-base-name-of-obs-code-path}/${boot-file} '/tmp/log/train.log' ${python_file_parameter}
  ```

  run_train.sh 为ModelArts提供的启动脚本，可自动完成OBS代码路径下载至本地、Ascend HCCL RANK_TABLE_FILE v0.1 格式转 v1.0 格式、多P训练进程拉起功能，详细描述请参见**适用于Ascend芯片基础镜像**。

  说明

  - “obs-code-path”：OBS 代码路径，例如 obs://training-bucket/ascend-tf-1.15/resnet50/
  - “the-base-name-of-obs-code-path”：OBS 代码路径的最后一级目录，例如 resnet50
  - “boot-file”：以 .py 结尾的训练启动文件，例如 train.py
  - '/tmp/log/train.log': 默认值，日志重定向至该文件
  - “python_file_parameter”：传入训练启动文件的参数，例如 --param1=value1 --param2=value2

  运行命令示例：

  ```
  /bin/bash /home/work/run_train.sh 'obs://training-bucket/ascend-tf-1.15/resnet50/' 'resnet50/train.py' '/tmp/log/train.log' --'data_url'='obs://training-bucket/cifar-10/' --'train_url'='obs://training-bucket/model/'
  ```

- **“环境变量”**

  表 12-10 必选环境变量说明

环境变量	说明
RANK_TABLE_FILE	该参数可以指定“jobstart_hccl.json” 文件的生成路径。 建议配置为 /user/config，则“jobstart_hccl.json ”文件路径为“/user/config/jobstart_hccl.json”。 算法开发者自行开发启动脚本时，可通过 “${RANK_TABLE_FILE}/jobstart_hccl.json”，获取文件。“jobstart_hccl.json”是 v0.1 版本的，用于分布式通信，会在运行过程中被Ascend910 芯片的集合通信库解析。

  说明

  若未添加上述环境变量，则系统不会生成 RANK TABLE FILE，训练作业日志会停留在

  ```
  Wait for Rank table file ready
  ```

  容器启动后，除了用户在训练作业中自行增加的“环境变量”外，其它加载的环境变量如**表12-11**所示。用户可以根据需求来确认在自己训练脚本的python中是否要使用这些环境变量，也可以通过运行命令中的“{python_file_parameter}”传入相关参数。

表 12-11 可选环境变量说明

环境变量	说明
DLS_TASK_INDEX	当前容器索引，容器从0开始编号。
DLS_TASK_NUMBER	容器总数。对应“计算节点个数”。
DLS_APP_URL	代码目录。对应界面上“代码目录”配置，会加上协议名。比如，可直接使用“$DLS_APP_URL/*.py”来读取OBS下的文件。
DLS_DATA_URL	数据集位置。对应界面上“数据来源”，会加上协议名。
DLS_TRAIN_URL	训练输出位置。对应界面上“训练输出位置”，会加上协议名。
BATCH_{jobName}.0_HOSTS（单机）	当选择单机时，即计算节点个数为1时，此环境变量为“BATCH_{jobName}.0_HOSTS”。 HOSTS环境变量的格式为“hostname:port”。一个容器可以看到同一个作业中所有容器的HOSTS，根据索引的不同，分别为“BATCH_CUSTOM0_HOSTS”、“BATCH_CUSTOM1_HOSTS”等。

– jobstart_hccl.json 文件格式（v0.1）示例

```
{
    "group_count": "1",
    "group_list": [{
        "device_count": "1",
        "group_name": "job-trainjob",
        "instance_count": "1",
        "instance_list": [{
            "devices": [{
                "device_id": "4",
                "device_ip": "192.1.10.254"
            }],
            "pod_name": "jobxxxxxxxx-job-trainjob-0",
            "server_id": "192.168.0.25"
        }]
    }],
    "status": "completed"
}
```

– jobstart_hccl.json 文件格式（v1.0）示例

```
{
    "server_count": "1",
    "server_list": [{
        "device": [{
            "device_id": "4",
            "device_ip": "192.1.10.254",
            "rank_id": "0"
        }],
        "server_id": "192.168.0.25"
    }],
    "status": "completed",
    "version": "1.0"
}
```

12.3.4 示例：使用自定义镜像创建训练作业

说明

基于自定义镜像训练模型仅适用于旧版训练模块（仅对部分存量用户可见，新用户不可见），新版训练暂不支持。

本示例所需的文件存储在**Github仓库**中。本示例使用**MNIST数据集**。端到端示例请参见**最佳实践**。

- “mnist_softmax.py”为单机脚本。

制作并上传自定义镜像

本示例使用Dockerfile文件定制自定义镜像。

以linux x86_x64架构的主机为例，您可以购买相同规格的ECS或者应用本地已有的主机进行自定义镜像的制作。

1. 安装Docker，可参考**Docker官方文档**。

   以linux x86_64架构的操作系统为例，获取Docker安装包。使用以下指令安装Docker：

   ```
   curl -fsSL get.docker.com -o get-docker.sh
   sh get-docker.sh
   ```

   如果**docker images**命令可以执行成功，表示Docker已安装，该步骤可跳过。

2. 获取自定义镜像的基础镜像。

   训练作业的自定义镜像需要以基础镜像为基础。基础镜像名称格式参见**基础镜像包概述**。使用以下指令获取自定义镜像的基础镜像：

   ```
   docker pull swr.<region>.myhuaweicloud.com/<image org>/<image name>
   ```

   另外，您还可以使用**docker images**命令可查看本地的镜像列表。

3. 编写构建自定义镜像的Dockerfile文件。

   本示例构建tensorflow 1.13.2版本镜像。文件命名为“tf-1.13.2.dockerfile”。执行*vi tf-1.13.2.dockerfile*命令，进入文件中。

   Dockerfile文件编写的更多指导内容参见**官方指导说明**。

   ```
   FROM swr.cn-north-4.myhuaweicloud.com/modelarts-job-dev-image/custom-base-cuda10.0-cp36-ubuntu18.04-x86:1.1
   # 配置华为云的源，安装tensorflow
   RUN cp -a /etc/apt/sources.list /etc/apt/sources.list.bak && \
   sed -i "s@http://.*archive.ubuntu.com@http://repo.myhuaweicloud.com@g"  /etc/apt/sources.list && \
   sed -i "s@http://.*security.ubuntu.com@http://repo.myhuaweicloud.com@g"  /etc/apt/sources.list && \
   pip install  --trusted-host https://repo.huaweicloud.com -i https://repo.huaweicloud.com/repository/pypi/simple  tensorflow==1.13.2
   # 配置环境变量
   ENV PATH=/root/miniconda3/bin/:$PATH
   ```

4. 构建自定义镜像。

   下列例子中镜像所在的区域为cn-north-4，镜像所属组织为deep-learning-diy，在“tf-1.13.2.dockerfile”文件所在的目录执行：

   ```
   docker build -f tf-1.13.2.dockerfile . -t swr.cn-north-4.myhuaweicloud.com/deep-learning-diy/tf-1.13.2:latest
   ```

5. 推送镜像至SWR，上传镜像的详细操作可参考**SWR用户指南**。

   前提条件是已经**创建组织**并**获取SWR登录指令**。下列例子中镜像所在的区域为cn-north-4，镜像所属组织为deep-learning-diy，执行以下命令推送镜像至SWR。

   ```
   docker push swr.cn-north-4.myhuaweicloud.com/deep-learning-diy/tf-1.13.2:latest
   ```

“swr.cn-north-4.myhuaweicloud.com/deep-learning-diy/tf-1.13.2:latest”即为此自定义镜像的“SWR_URL”。

单机训练

1. 将训练代码“mnist_softmax.py”和训练数据上传至OBS。将代码和数据都放在同一代码根目录下，以便直接下载到容器中。

   以根目录“obs://deep-learning/new/mnist/”为例：

   训练代码文件为“obs://deep-learning/new/mnist/mnist_softmax.py”；

   数据存储路径为“obs://deep-learning/new/mnist/mnist_data”。

2. 创建自定义镜像训练作业，“镜像地址”、“代码目录”和“运行命令”参考如下信息填写，“数据存储位置”和“训练输出位置”请根据实际情况填写。
   - “镜像地址”：填写已上传镜像的“SWR_URL”。
   - “代码目录”：训练代码存储的OBS路径，即为步骤1中的代码根目录。

     在训练作业实际启动之前，ModelArts自动将“代码目录”下的所有内容递归下载到容器本地路径。下载后的容器本地路径为“/home/work/user-job-dir/${代码根目录的最后一级名称}/”。例如“代码目录”选择“obs://deep-learning/new/mnist”时，下载后的本地路径为“/home/work/user-job-dir/mnist/”，代码启动文件为“/home/work/user-job-dir/mnist/mnist_softmax.py”。
   - “运行命令”：**bash /home/work/run_train.sh python /home/work/user-job-dir/mnist/mnist_softmax.py --data_url /home/work/user-job-dir/mnist/mnist_data**

     其中，“/home/work/user-job-dir/mnist/mnist_softmax.py”为代码启动文件，“--data_url /home/work/user-job-dir/mnist/mnist_data”为数据存储路径。

3. 训练作业创建完成后，后台完成代码目录下载、自定义镜像审核以及自定义镜像的训练作业。训练作业一般需要运行一段时间，根据您选择的数据量和资源不同，训练时间将耗时几分钟到几十分钟不等。程序执行成功后，日志信息如下所示。

   图 12-7 运行日志信息

12.4 用于创建 AI 应用

12.4.1 创建 AI 应用的自定义镜像规范

针对您本地开发的模型，在制作AI应用的自定义镜像时，需满足ModelArts定义的规范。

- 自定义镜像中不能包含恶意代码。
- 自定义镜像大小不超过10GB。

- **镜像对外接口**

  镜像的对外服务接口需要为8080，推理接口需与config.json文件中apis定义的url一致，当镜像启动时可以直接访问。下面是mnist镜像的访问示例，该镜像内含mnist数据集训练的模型，可以识别手写数字。其中listen_ip为容器IP。

  – 请求示例
  ```
  curl -X POST \ http://{listen_ip}:8080/ \ -F images=@seven.jpg
  ```
  – 返回示例
  ```
  {"mnist_result": 7}
  ```

- **（可选）健康检查接口**

  如果在滚动升级时要求不中断业务，那么必需提供健康检查的接口供ModelArts调用，在config.json文件中配置。当业务可提供正常服务时，健康检查接口返回健康状态，否则返回异常状态。

  须知

  如果要实现无损滚动升级，必需配置健康检查接口。

  健康检查接口示例如下。

  – URI
  ```
  GET /health
  ```
  – 请求示例**curl -X GET \ http://{listen_ip}:8080/health**
  – 响应示例
  ```
  {"health": "true"}
  ```
  – 状态码

  表 12-12 状态码

状态码	编码	状态码说明
200	OK	请求成功

- **日志文件输出**

  为保证日志内容可以正常显示，日志信息需要打印到标准输出。

- **镜像启动入口**

  如果需要部署批量服务，镜像的启动入口文件需要为“/home/run.sh”，采用CMD设置默认启动路径，例如Dockerfile如下：

  CMD ["sh", "/home/run.sh"]

- **镜像依赖组件**

  如果需要部署批量服务，镜像内需要安装python、jre/jdk、zip等组件包。

- **（可选）保持Http长链接，无损滚动升级**

  如果需要支持滚动升级的过程中不中断业务，那么需要将服务的Http的“keep-alive”参数设置为200s。以gunicorn服务框架为例，gunicorn缺省情形下不支持keep-alive，需要同时安装gevent并配置启动参数“--keep-alive 200 -k gevent”。不同服务框架参数设置有区别，请以实际情况为准。

- **（可选）处理SIGTERM信号，容器优雅退出**

如果需要支持滚动升级的过程中不中断业务，那么需要在容器中捕获SIGTERM信号，并且在收到SIGTERM信号之后等待60秒再优雅退出容器。提前优雅退出容器可能会导致在滚动升级的过程中业务概率中断。要保证容器优雅退出，从收到SIGTERM信号开始，业务需要将收到的请求全部处理完毕再结束，这个处理时长最多不超过90秒。例如run.sh如下所示：

```
#!/bin/bash
gunicorn_pid=""

handle_sigterm() {
  echo "Received SIGTERM, send SIGTERM to $gunicorn_pid"
  if [ $gunicorn_pid != "" ]; then
      sleep 60
      kill -15 $gunicorn_pid  # 传递 SIGTERM 给gunicorn进程
      wait $gunicorn_pid          # 等待gunicorn进程完全终止
  fi
}

trap handle_sigterm TERM
```

12.4.2 使用自定义镜像创建 AI 应用

用户将自定义镜像制作完成并上传至SWR后，可在ModelArts管理控制台，使用自定义镜像导入AI应用，并将AI应用部署上线。

前提条件

- 已按照ModelArts规范制作自定义镜像包，自定义镜像包规范请参见**创建AI应用的自定义镜像规范**。镜像构建可参见**本地构建镜像**。
- 已将自定义镜像上传至SWR服务，操作指导可参见**本地验证镜像并上传镜像至SWR服务**。

创建 AI 应用

参考**从容器镜像中选择元模型**操作指导，将自定义镜像创建为AI应用。需关注如下参数的设置：

- “元模型来源”

  选择“从容器镜像中选择”，然后在“容器镜像所在的路径”右侧输入框中，单击 选择您的镜像。系统将自动罗列出当前帐号上传至SWR服务的所有镜像。请根据实际情况选择您使用的镜像。

  容器调用接口：选填，用于指定模型启动的协议和端口号。

  健康检查URL：选填，用于指定模型的健康检查。
- apis定义：选填，用于编辑自定义镜像的apis定义。模型apis定义需要遵循ModelArts的填写规范，参见**模型配置文件编写说明**。

图 12-8 使用自定义镜像导入 AI 应用

部署上线

待使用自定义镜像创建AI应用成功后，即AI应用状态显示为正常时，您可以将AI应用部署为服务。在“AI应用列表”中，单击“部署”，选择服务类型，如“在线服务”。

您可以基于您自定义镜像的业务逻辑，将AI应用部署为在线服务、批量服务或边缘服务。

12.4.3 使用自定义镜像创建的 AI 应用部署服务

针对ModelArts目前不支持的AI引擎，您可以通过自定义镜像的方式将编写的模型导入ModelArts，创建为AI应用。本文详细介绍如何使用自定义镜像完成AI应用的创建，并部署成在线服务。

操作流程如下：

1. **本地构建镜像**：在本地制作自定义镜像包，镜像包规范可参考**创建AI应用的自定义镜像规范**。
2. **本地验证镜像并上传镜像至SWR服务**：验证自定义镜像的API接口功能，无误后将自定义镜像上传至SWR服务。
3. **将自定义镜像创建为AI应用**：将上传至SWR服务的镜像导入ModelArts模型管理。
4. **将AI应用部署为在线服务**：将导入的模型部署上线。

本地构建镜像

以linux x86_x64架构的主机为例，您可以购买相同规格的ECS或者应用本地已有的主机进行自定义镜像的制作。

1. 安装Docker，可参考**Docker官方文档**。可参考以下方式安装docker。
```
curl -fsSL get.docker.com -o get-docker.sh
sh get-docker.sh
```
2. 获取基础镜像。本示例以Ubuntu18.04为例。
```
docker pull ubuntu:18.04
```
3. 新建文件夹“self-define-images”，在该文件夹下编写自定义镜像的“Dockerfile”文件和应用服务代码“test_app.py”。本样例代码中，应用服务代码采用了flask框架。

   文件结构如下所示
```
self-define-images/
   --Dockerfile
   --test_app.py
```
   – “Dockerfile”
```
From ubuntu:18.04
# 配置华为云的源，安装 python、python3-pip 和 Flask
RUN cp -a /etc/apt/sources.list /etc/apt/sources.list.bak && \
  sed -i "s@http://.*security.ubuntu.com@http://repo.huaweicloud.com@g" /etc/apt/sources.list
&& \
  sed -i "s@http://.*archive.ubuntu.com@http://repo.huaweicloud.com@g" /etc/apt/sources.list
&& \
  apt-get update && \
  apt-get install -y python3 python3-pip && \
  pip3 install  --trusted-host https://repo.huaweicloud.com -i https://repo.huaweicloud.com/
repository/pypi/simple  Flask

# 拷贝应用服务代码进镜像里面
COPY test_app.py /opt/test_app.py
```

```
# 指定镜像的启动命令
CMD python3  /opt/test_app.py
```

– “test_app.py”

```
from flask import Flask, request
import json
app = Flask(__name__)

@app.route('/greet', methods=['POST'])
def say_hello_func():
    print("----------- in hello func ----------")
    data = json.loads(request.get_data(as_text=True))
    print(data)
    username = data['name']
    rsp_msg = 'Hello, {}!'.format(username)
    return json.dumps({"response":rsp_msg}, indent=4)

@app.route('/goodbye', methods=['GET'])
def say_goodbye_func():
    print("----------- in goodbye func ----------")
    return '\nGoodbye!\n'


@app.route('/', methods=['POST'])
def default_func():
    print("----------- in default func ----------")
    data = json.loads(request.get_data(as_text=True))
    return '\n called default func !\n {} \n'.format(str(data))

# host must be "0.0.0.0", port must be 8080
if __name__ == '__main__':
    app.run(host="0.0.0.0", port=8080)
```

说明

modelarts 平台会将请求转发至自定义镜像起的服务的8080端口，故容器内的服务监听的端口必须是8080，如 test_app.py文件所示。

4. 进入“self-define-images”文件夹，执行以下命令构建自定义镜像“test:v1”。

```
docker build -t test:v1 .
```

5. 您可以使用“docker image”查看您构建的自定义镜像。

本地验证镜像并上传镜像至 SWR 服务

1. 在本地环境执行以下命令启动自定义镜像

```
docker run -it -p 8080:8080 test:v1
```

图 12-9 启动自定义镜像

2. 另开一个终端，执行以下命令验证自定义镜像的三个API接口功能。

```
curl -X POST -H "Content-Type: application/json" --data '{"name":"Tom"}'  127.0.0.1:8080/
curl -X POST -H "Content-Type: application/json" --data '{"name":"Tom"}' 127.0.0.1:8080/greet
curl -X GET 127.0.0.1:8080/goodbye
```

如果验证自定义镜像功能成功，结果如下图所示。

图 12-10 校验接口

```
root@                :~# curl -X POST -H "Content-Type: application/json" --data '{"name":"Tom"}'  127.0.0.1:8080/

 called default func !
 {'name': 'Tom'}
root@                :~# curl -X POST -H "Content-Type: application/json" --data '{"name":"Tom"}' 127.0.0.1:8080/greet
{
    "response": "Hello, Tom!"
}root@                :~#curl -X GET 127.0.0.1:8080/goodbye

Goodbye!
```

3. 上传自定义镜像至SWR服务。上传镜像的详细操作可参考**SWR用户指南**。
4. 完成自定义镜像上传后，您可以在“容器镜像服务>我的镜像>自有镜像”列表中看到已上传镜像。

图 12-11 上传镜像列表

将自定义镜像创建为 AI 应用

参考**从容器镜像中选择元模型**导入元模型，您需要特别关注以下参数：

- 元模型来源：选择“从容器镜像中选择”
  - 容器镜像所在的路径：选择已制作好的自有镜像

    图 12-12 选择已制作好的自有镜像

  - 容器调用接口：选填，用于指定模型启动的协议和端口号。
  - 健康检查：选填，用于指定模型的健康检查。仅当自定义镜像中配置了健康检查接口，才能配置“健康检查”，否则会导致AI应用创建失败。
- apis定义：选填，用于编辑自定义镜像的apis定义。模型apis定义需要遵循ModelArts的填写规范，参见**模型配置文件说明**。

  本样例的配置文件如下所示：

```
[{
        "url": "/",
        "method": "post",
        "request": {
            "Content-type": "application/json"
        },
        "response": {
            "Content-type": "application/json"
        }
    },
{
        "url": "/greet",
        "method": "post",
        "request": {
```

```
            "Content-type": "application/json"
        },
        "response": {
            "Content-type": "application/json"
        }
    },
{
        "url": "/goodbye",
        "method": "get",
        "request": {
            "Content-type": "application/json"
        },
        "response": {
            "Content-type": "application/json"
        }
    }
]
```

将 AI 应用部署为在线服务

1. 参考**部署为在线服务**将AI应用部署为在线服务。
2. 在线服务创建成功后，您可以在服务详情页查看服务详情。

图 12-13 调用指南

3. 您可以通过“预测”页签访问在线服务。

图 12-14 访问在线服务

13 模型包规范（即将下线）

13.1 模型包规范介绍

说明

为了更方便用户使用和清晰的获取文档帮助，《AI工程师用户指南》将按照AI开发流程调整拆分为多本文档，在一级导航中呈现，并从用户使用场景对各模块文档进行了优化和改进。“模型包规范”的内容已拆分至《推理部署》,《AI工程师用户指南》里的“模型包规范介绍”即将下线。

在AI应用管理创建AI应用时，如果是从OBS中导入元模型，则需要符合一定的模型包规范。

说明

模型包框架适用于单模型场景，若是多模型场景（例如含有多个权重文件）推荐使用自定义镜像方式。

模型包里面必需包含“model”文件夹，“model”文件夹下面放置模型文件，模型配置文件，模型推理代码。

- 模型文件在不同模型包结构中要求不同，具体请参见模型包示例。
- 模型配置文件必需存在，文件名固定为“config.json”，有且只有一个，模型配置文件编写请参见模型配置文件编写说明。
- 模型推理代码文件是可选的。如果需要此文件，则文件名固定为“customize_service.py”，此文件有且只能有一个，模型推理代码编写请参见模型推理代码编写说明。
  - customize_service.py依赖的py文件可以直接放model目录下，推荐采用相对导入方式（Python Import）导入自定义包。
  - customize_service.py依赖的其他文件可以直接放model目录下，需要采用绝对路径方式访问。绝对路径获取请参考绝对路径如何获取。

ModelArts针对多种引擎提供了样例及其示例代码，您可以参考样例编写您的配置文件和推理代码，详情请参见**ModelArts样例列表**。ModelArts也提供了常用AI引擎对应的自定义脚本示例，请参见自定义脚本代码示例。

模型包示例

- TensorFlow模型包结构

  发布该模型时只需要指定到“ocr”目录。

```
OBS桶/目录名
|—— ocr
|   |—— model 必选： 固定子目录名称，用于放置模型相关文件
|   |   |—— <<自定义python包>> 可选：用户自有的Python包，在模型推理代码中可以直接引用
|   |   |—— saved_model.pb 必选: protocol buffer格式文件，包含该模型的图描述
|   |   |—— variables 对*.pb模型主文件而言必选；固定子目录名称，包含模型的权重偏差等信息
|   |   |   |—— variables.index 必选
|   |   |   |—— variables.data-00000-of-00001 必选
|   |   |——config.json 必选：模型配置文件，文件名称固定为config.json, 只允许放置一个
|   |   |——customize_service.py 可选：模型推理代码，文件名称固定为customize_service.py, 只允许放置一个，customize_service.py依赖的文件可以直接放model目录下
```

- MXNet模型包结构

  发布该模型时只需要指定到“resnet”目录。

```
OBS桶/目录名
|—— resnet
|   |—— model 必选： 固定子目录名称，用于放置模型相关文件
|   |   |—— <<自定义python包>> 可选：用户自有的Python包，在模型推理代码中可以直接引用
|   |   |—— resnet-50-symbol.json 必选，模型定义文件，包含模型的神经网络描述
|   |   |—— resnet-50-0000.params 必选；模型变量参数文件，包含参数和权重信息
|   |   |——config.json 必选：模型配置文件，文件名称固定为config.json, 只允许放置一个
|   |   |——customize_service.py 可选：模型推理代码，文件名称固定为customize_service.py, 只允许放置一个，customize_service.py依赖的文件可以直接放model目录下
```

- Image模型包结构

  发布该模型时只需要指定到“resnet”目录。

```
OBS桶/目录名
|—— resnet
|   |—— model 必选： 固定子目录名称，用于放置模型相关文件
|   |   |——config.json 必选：模型配置文件（需要配置swr镜像地址），文件名称固定为config.json, 只允许放置一个
```

- pyspark模型包结构

  发布该模型时只需要指定到“resnet”目录。

```
OBS桶/目录名
|—— resnet
|   |—— model 必选： 固定子目录名称，用于放置模型相关文件
|   |   |—— <<自定义Python包>> 可选：用户自有的Python包，在模型推理代码中可以直接引用
|   |   |—— spark_model 必选: 模型文件夹，包含pyspark保存的模型内容
|   |   |——config.json 必选：模型配置文件，文件名称固定为config.json, 只允许放置一个
|   |   |——customize_service.py 可选：模型推理代码，文件名称固定为customize_service.py, 只允许放置一个，customize_service.py依赖的文件可以直接放model目录下
```

- PyTorch模型包结构

  发布该模型时只需要指定到“resnet”目录。

```
OBS桶/目录名
|—— resnet
|   |—— model 必选： 固定子目录名称，用于放置模型相关文件
|   |   |—— <<自定义Python包>> 可选：用户自有的Python包，在模型推理代码中可以直接引用
|   |   |—— resnet50.pth 必选，pytorch模型保存文件，保存为“state_dict”，存有权重变量等信息。
|   |   |——config.json 必选：模型配置文件，文件名称固定为config.json, 只允许放置一个
|   |   |——customize_service.py 必选：模型推理代码，文件名称固定为customize_service.py, 只允许放置一个，customize_service.py依赖的文件可以直接放model目录下
```

- Caffe模型包结构

  用户发布该模型时只需要指定到“resnet”目录

```
OBS桶/目录名
|—— resnet
|   |—— model 必选： 固定子目录名称，用于放置模型相关文件
|   |   |—— <<自定义python包>> 可选：用户自有的Python包，在模型推理代码中可以直接引用
```

```
|   |   |—— deploy.prototxt 必选，caffe模型保存文件，存有模型网络结构等信息
|   |   |—— resnet.caffemodel 必选，caffe模型保存文件，存有权重变量等信息
|   |   |—— config.json 必选：模型配置文件，文件名称固定为config.json, 只允许放置一个
|   |   |—— customize_service.py  可选：模型推理代码，文件名称固定为customize_service.py, 只允许放置一个，customize_service.py依赖的文件可以直接放model目录下
```

- XGBoost模型包结构

  用户发布该模型时只需要指定到“resnet”目录

```
OBS桶/目录名
|—— resnet
|   |—— model  必选： 固定子目录名称，用于放置模型相关文件
|   |   |—— <<自定义python包>>  可选：用户自有的Python包，在模型推理代码中可以直接引用
|   |   |—— *.m必选: 模型文件，命名以.m结尾
|   |   |—— config.json 必选：模型配置文件，文件名称固定为config.json, 只允许放置一个
|   |   |—— customize_service.py  可选：模型推理代码，文件名称固定为customize_service.py, 只允许放置一个，customize_service.py依赖的文件可以直接放model目录下
```

- Scikit_Learn模型包结构

  用户发布该模型时只需要指定到“resnet”目录

```
OBS桶/目录名
|—— resnet
|   |—— model  必选： 固定子目录名称，用于放置模型相关文件
|   |   |—— <<自定义python包>>  可选：用户自有的Python包，在模型推理代码中可以直接引用
|   |   |—— *.m必选: 模型文件，命名以.m结尾
|   |   |—— config.json 必选：模型配置文件，文件名称固定为config.json, 只允许放置一个
|   |   |—— customize_service.py  可选：模型推理代码，文件名称固定为customize_service.py, 只允许放置一个，customize_service.py依赖的文件可以直接放model目录下
```

13.2 模型配置文件编写说明

模型开发者发布模型时需要编写配置文件。模型配置文件描述模型用途、模型计算框架、模型精度、推理代码依赖包以及模型对外API接口。

配置文件格式说明

配置文件为JSON格式，参数说明如表13-1所示。

表 13-1 参数说明

参数	是否必选	参数类型	描述
model_algorithm	是	String	模型算法，表示该模型的用途，由模型开发者填写，以便使用者理解该模型的用途。只能以英文字母开头，不能包含中文以及&!'\"<>=，不超过36个字符。常见的模型算法有image_classification（图像分类）、object_detection（物体检测）、predict_analysis（预测分析）等。
model_type	是	String	模型AI引擎，表明模型使用的计算框架，可选的框架有TensorFlow、MXNet、Caffe、MindSpore、Image。 其中，Image并不是一个常用AI框架，当model_type设置为Image，表示以自定义镜像方式创建AI应用，此时swr_location为必填参数。Image镜像制作规范可参见模型镜像规范。

参数	是否必选	参数类型	描述
runtime	否	String	模型运行时环境，系统默认使用 python2.7。runtime可选值与model_type相关，当model_type设置为Image时，不需要设置runtime，当model_type设置为其他常用框架时，请选择您使用的引擎及其对应开发环境。目前支持的运行环境列表请参见**支持的常用引擎及其Runtime**。 需要注意的是，如果您的模型需指定CPU或GPU上运行时，请根据runtime的后缀信息选择，当runtime中未包含cpu或gpu信息时，请仔细阅读**支持的常用引擎及其Runtime**中每个runtime的说明信息。
swr_location	否	String	SWR镜像地址。 ● 如果您使用“从容器镜像中选择”的方式导入自定义镜像元模型，“swr_location”参数无需配置。 ● 如果您使用“从对象存储服务(OBS)中选择”的方式导入自定义镜像元模型（不推荐该方式），且model_type设置为Image时，“swr_location”参数必填。“swr_location”为docker镜像在SWR上的地址，表示直接使用SWR的docker镜像发布模型。
metrics	否	object数据结构	模型的精度信息，包括平均数、召回率、精确率、准确率，metrics object数据结构说明如**表13-2**所示。 结果会显示在AI应用详情页面的“模型精度”模块。
apis	否	api数据结构数组	表示模型接收和返回的请求样式，为结构体数据。 即模型可对外提供的Restful API数组，API数据结构如**表13-3**所示。 ● “model_type”为“Image”时，即自定义镜像的模型场景，“apis”可根据镜像实际对外暴露的请求路径在“apis”中声明不同路径的API。 ● “model_type”不为“Image”时，“apis”只能声明一个请求路径为“/”的API，因为系统预置的AI引擎仅暴露一个请求路径为“/”的推理接口。

参数	是否必选	参数类型	描述
dependencies	否	dependency结构数组	表示模型推理代码需要依赖的包，为结构体数据。 模型开发者需要提供包名、安装方式、版本约束。目前只支持pip安装方式。dependency结构数组说明如表13-6所示。 如果模型包内没有推理代码customize_service.py文件，则该字段可不填。自定义镜像模型不支持安装依赖包。 **说明** “dependencies”参数支持多个“dependency”结构数组以ist格式填入，适用于安装包存在先后依赖关系。示例请参考**使用OBS或者容器镜像导入模型时安装包存在前后依赖关系，模型配置文件应该如何说明?**
health	否	health数据结构	镜像健康接口配置信息，只有“model_type”为“Image”时才需填写。 如果在滚动升级时要求不中断业务，那么必需提供健康检查的接口供ModelArts调用。health数据结构如表13-8所示。

表 13-2 metrics object 数据结构说明

参数	是否必选	参数类型	描述
f1	否	Number	平均数。精确到小数点后17位，超过17位时，取前17位数值。
recall	否	Number	召回率。精确到小数点后17位，超过17位时，取前17位数值。
precision	否	Number	精确率。精确到小数点后17位，超过17位时，取前17位数值。
accuracy	否	Number	准确率。精确到小数点后17位，超过17位时，取前17位数值。

表 13-3 api 数据结构说明

参数	是否必选	参数类型	描述
protocol	否	String	请求协议。填写http或者https均可，请根据您的自定义镜像填写此参数。如果您使用OBS导入元模型，默认协议为https。其他字段的详细指导可参考**目标检测模型配置文件示例**解读。

参数	是否必选	参数类型	描述
url	否	String	请求路径。默认值为“/”。自定义镜像的模型（即model_type为Image时）需要根据镜像内实际暴露的请求路径填写“url”。非自定义镜像模型（即model_type不为Image时）时，“url”只能为“/”。
method	否	String	请求方法。默认值为“POST”。
request	否	Object	请求体，request结构说明如表13-4所示。
response	否	Object	响应体，response结构说明如表13-5所示。

表 13-4 request 结构说明

参数	是否必选	参数类型	描述
Content-type	在线服务-非必选 批量服务-必选	String	data以指定内容类型发送。默认值为“application/json”。 一般情况包括如下两种内容类型： ● “application/json”，发送json数据。 ● “multipart/form-data”，上传文件。 **说明** 针对机器学习类模型，仅支持“application/json”
data	在线服务-非必选 批量服务-必选	String	请求体以json schema描述。参数说明请参考官方指导。

表 13-5 response 结构说明

参数	是否必选	参数类型	描述
Content-type	在线服务-非必选 批量服务-必选	String	data以指定内容类型发送。默认值为“application/json”。 **说明** 针对机器学习类模型，仅支持“application/json”
data	在线服务-非必选 批量服务-必选	String	响应体以json schema描述。参数说明请参考官方指导。

表 13-6 dependency 结构数组说明

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，package结构数组说明如表13-7所示。

表 13-7 package 结构数组说明

参数	是否必选	参数类型	描述
package_name	是	String	依赖包名称。不能含有中文及特殊字符&!'"<>=。
package_version	否	String	依赖包版本，如果不强依赖于版本号，则该项不填。不能含有中文及特殊字符&!'"<>=。
restraint	否	String	版本限制条件，当且仅当“package_version”存在时必填，可选“EXACT/ATLEAST/ATMOST”。 ● “EXACT”表示安装给定版本。 ● “ATLEAST”表示安装版本不小于给定版本。 ● “ATMOST”表示安装包版本不大于给定版本。 **说明** ● 如果对版本有明确要求，优先使用“EXACT”；如果使用“EXACT”与系统安装包有冲突，可以选择“ATLEAST” ● 如果对版本没有明确要求，推荐不填写“restraint”、“package_version”，只保留“package_name”参数

表 13-8 health 数据结构说明

参数	是否必选	参数类型	描述
url	是	String	健康检查接口请求路径。
protocol	否	String	健康检查接口请求协议，当前仅支持http。

参数	是否必选	参数类型	描述
initial_delay_seconds	否	String	实例启动后，延迟initial_delay_seconds秒再执行健康检查。
timeout_seconds	否	String	健康检查超时时间

目标检测模型配置文件示例

如下代码以TensorFlow引擎为例，您可以根据实际使用的引擎类型修改model_type参数后使用。

- 模型输入

  key：images

  value：图片文件

- 模型输出

```
{
    "detection_classes": [
        "face",
        "arm"
    ],
    "detection_boxes": [
        [
            33.6,
            42.6,
            104.5,
            203.4
        ],
        [
            103.1,
            92.8,
            765.6,
            945.7
        ]
    ],
    "detection_scores": [0.99, 0.73]
}
```

- 配置文件

```
{
    "model_type": "TensorFlow",
    "model_algorithm": "object_detection",
    "metrics": {
        "f1": 0.345294,
        "accuracy": 0.462963,
        "precision": 0.338977,
        "recall": 0.351852
    },
    "apis": [{
        "protocol": "http",
        "url": "/",
        "method": "post",
        "request": {
            "Content-type": "multipart/form-data",
            "data": {
                "type": "object",
                "properties": {
```

```
                "images": {
                    "type": "file"
                }
            }
        }
    },
    "response": {
        "Content-type": "multipart/form-data",
        "data": {
            "type": "object",
            "properties": {
                "detection_classes": {
                    "type": "array",
                    "items": [{
                        "type": "string"
                    }]
                },
                "detection_boxes": {
                    "type": "array",
                    "items": [{
                        "type": "array",
                        "minItems": 4,
                        "maxItems": 4,
                        "items": [{
                            "type": "number"
                        }]
                    }]
                },
                "detection_scores": {
                    "type": "array",
                    "items": [{
                        "type": "number"
                    }]
                }
            }
        }
    }
}],
"dependencies": [{
    "installer": "pip",
    "packages": [{
            "restraint": "EXACT",
            "package_version": "1.15.0",
            "package_name": "numpy"
        },
        {
            "restraint": "EXACT",
            "package_version": "5.2.0",
            "package_name": "Pillow"
        }
    ]
}]
}
```

图像分类模型配置文件示例

如下代码以TensorFlow引擎为例，您可以根据实际使用的引擎类型修改model_type参数后使用。

- 模型输入

  key：images

  value：图片文件

- 模型输出

  ```
  {
  ```

```
    "predicted_label": "flower",
    "scores": [
      ["rose", 0.99],
      ["begonia", 0.01]
    ]
}
```

- 配置文件

```
{
    "model_type": "TensorFlow",
    "model_algorithm": "image_classification",
    "metrics": {
        "f1": 0.345294,
        "accuracy": 0.462963,
        "precision": 0.338977,
        "recall": 0.351852
    },
    "apis": [{
        "protocol": "http",
        "url": "/",
        "method": "post",
        "request": {
            "Content-type": "multipart/form-data",
            "data": {
                "type": "object",
                "properties": {
                    "images": {
                        "type": "file"
                    }
                }
            }
        },
        "response": {
            "Content-type": "multipart/form-data",
            "data": {
                "type": "object",
                "properties": {
                    "predicted_label": {
                        "type": "string"
                    },
                    "scores": {
                        "type": "array",
                        "items": [{
                            "type": "array",
                            "minItems": 2,
                            "maxItems": 2,
                            "items": [
                                {
                                    "type": "string"
                                },
                                {
                                    "type": "number"
                                }
                            ]
                        }]
                    }
                }
            }
        }
    }],
    "dependencies": [{
        "installer": "pip",
        "packages": [{
                "restraint": "ATLEAST",
                "package_version": "1.15.0",
                "package_name": "numpy"
            },
            {
```

```
            "restraint": "",
            "package_version": "",
            "package_name": "Pillow"
        }
    ]
  }]
}
```

预测分析模型配置文件示例

如下代码以TensorFlow引擎为例，您可以根据实际使用的引擎类型修改model_type参数后使用。

- 模型输入

```
{
  "data": {
    "req_data": [
      {
        "buying_price": "high",
        "maint_price": "high",
        "doors": "2",
        "persons": "2",
        "lug_boot": "small",
        "safety": "low",
        "acceptability": "acc"
      },
      {
        "buying_price": "high",
        "maint_price": "high",
        "doors": "2",
        "persons": "2",
        "lug_boot": "small",
        "safety": "low",
        "acceptability": "acc"
      }
    ]
  }
}
```

- 模型输出

```
{
  "data": {
    "resp_data": [
      {
        "predict_result": "unacc"
      },
      {
        "predict_result": "unacc"
      }
    ]
  }
}
```

- 配置文件

```
{
  "model_type": "TensorFlow",
  "model_algorithm": "predict_analysis",
  "metrics": {
    "f1": 0.345294,
    "accuracy": 0.462963,
    "precision": 0.338977,
    "recall": 0.351852
  },
```

```
  "apis": [
    {
        "protocol": "http",
        "url": "/",
        "method": "post",
        "request": {
            "Content-type": "application/json",
            "data": {
                "type": "object",
                "properties": {
                    "data": {
                        "type": "object",
                        "properties": {
                            "req_data": {
                                "items": [
                                    {
                                        "type": "object",
                                        "properties": {
                                        }
                                    }],
                                "type": "array"
                            }
                        }
                    }
                }
            }
        },
        "response": {
            "Content-type": "multipart/form-data",
            "data": {
                "type": "object",
                "properties": {
                    "data": {
                        "type": "object",
                        "properties": {
                            "resp_data": {
                                "type": "array",
                                "items": [
                                    {
                                        "type": "object",
                                        "properties": {
                                        }
                                    }]
                            }
                        }
                    }
                }
            }
        }
    }],
  "dependencies": [
    {
        "installer": "pip",
        "packages": [
            {
                "restraint": "EXACT",
                "package_version": "1.15.0",
                "package_name": "numpy"
            },
            {
                "restraint": "EXACT",
                "package_version": "5.2.0",
                "package_name": "Pillow"
            }]
    }]
}
```
```
```

自定义镜像类型的模型配置文件示例

模型输入和输出与目标检测模型配置文件示例类似。

- 模型预测输入为**图片类型**时，request请求示例如下：

  该实例表示模型预测接收一个参数名为images、参数类型为file的预测请求，在推理界面会显示文件上传按钮，以文件形式进行预测。

```
{
        "Content-type": "multipart/form-data",
        "data": {
            "type": "object",
            "properties": {
                "images": {
                    "type": "file"
                }
            }
        }
}
```

- 模型预测输入为**json数据类型**时，request请求示例如下：

  该实例表示模型预测接收json请求体，只有一个参数名为input、参数类型为string的预测请求，在推理界面会显示文本输入框，用于填写预测请求。

```
  {
        "Content-type": "application/json",
        "data": {
            "type": "object",
            "properties": {
                "input": {
                    "type": "string"
                }
            }
        }
    }
```

完整请求示例如下：

```
{
    "model_algorithm": "image_classification",
    "model_type": "Image",

    "metrics": {
        "f1": 0.345294,
        "accuracy": 0.462963,
        "precision": 0.338977,
        "recall": 0.351852
    },
    "apis": [{
        "protocol": "http",
        "url": "/",
        "method": "post",
        "request": {
            "Content-type": "multipart/form-data",
            "data": {
                "type": "object",
                "properties": {
                    "images": {
                        "type": "file"
                    }
                }
            }
        },
        "response": {
            "Content-type": "multipart/form-data",
            "data": {
                "type": "object",
                "required": [
```

```
                "predicted_label",
                "scores"
            ],
            "properties": {
                "predicted_label": {
                    "type": "string"
                },
                "scores": {
                    "type": "array",
                    "items": [{
                        "type": "array",
                        "minItems": 2,
                        "maxItems": 2,
                        "items": [{
                                "type": "string"
                            },
                            {
                                "type": "number"
                            }
                        ]
                    }]
                }
            }
        }
    }
    }]
}
```

机器学习类型的模型配置文件示例

以XGBoost为例：

- 模型输入：

```
{
    "data": {
        "req_data": [{
            "sepal_length": 5,
            "sepal_width": 3.3,
            "petal_length": 1.4,
            "petal_width": 0.2
        }, {
            "sepal_length": 5,
            "sepal_width": 2,
            "petal_length": 3.5,
            "petal_width": 1
        }, {
            "sepal_length": 6,
            "sepal_width": 2.2,
            "petal_length": 5,
            "petal_width": 1.5
        }]
    }
}
```

- 模型输出：

```
{
    "data": {
        "resp_data": [{
            "predict_result": "Iris-setosa"
        }, {
            "predict_result": "Iris-versicolor"
        }]
    }
}
```

- 配置文件：

```
{
  "model_type": "XGBoost",
```

```
  "model_algorithm": "xgboost_iris_test",
  "runtime": "python2.7",
  "metrics": {
    "f1": 0.345294,
    "accuracy": 0.462963,
    "precision": 0.338977,
    "recall": 0.351852
  },
  "apis": [
    {
      "protocol": "http",
      "url": "/",
      "method": "post",
      "request": {
        "Content-type": "application/json",
        "data": {
          "type": "object",
          "properties": {
            "data": {
              "type": "object",
              "properties": {
                "req_data": {
                  "items": [
                    {
                      "type": "object",
                      "properties": {}
                    }
                  ],
                  "type": "array"
                }
              }
            }
          }
        }
      },
      "response": {
        "Content-type": "applicaton/json",
        "data": {
          "type": "object",
          "properties": {
            "resp_data": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "predict_result": {
                      "type": "number"
                    }
                  }
                }
              ]
            }
          }
        }
      }
    }
  ]
}
```

使用自定义依赖包的模型配置文件示例

如下示例中，定义了1.16.4版本的numpy的依赖环境。

```
{
    "model_algorithm": "image_classification",
    "model_type": "TensorFlow",
    "runtime": "python3.6",
```

```
  "apis": [{
        "procotol": "http",
        "url": "/",
        "method": "post",
        "request": {
            "Content-type": "multipart/form-data",
            "data": {
                "type": "object",
                "properties": {
                    "images": {
                        "type": "file"
                    }
                }
            }
        },
        "response": {
            "Content-type": "applicaton/json",
            "data": {
                "type": "object",
                "properties": {
                    "mnist_result": {
                        "type": "array",
        "item": [{
           "type": "string"
                    }]
                    }
                }
            }
        }
    }
  ],
  "metrics": {
      "f1": 0.124555,
      "recall": 0.171875,
      "precision": 0.0023493892851938493,
      "accuracy": 0.00746268656716417
  },
 "dependencies": [{
     "installer": "pip",
     "packages": [{
            "restraint": "EXACT",
            "package_version": "1.16.4",
            "package_name": "numpy"
        }
     ]
  }]
}
```

13.3 模型推理代码编写说明

本章节介绍了在ModelArts中模型推理代码编写的通用方法及说明，针对常用AI引擎的自定义脚本代码示例（包含推理代码示例），请参见自定义脚本代码示例。本文在编写说明下方提供了一个TensorFlow引擎的推理代码示例以及一个在推理脚本中自定义推理逻辑的示例。

推理代码编写指导

1. 所有的自定义Python代码必须继承自BaseService类，不同类型的模型父类导入语句如表13-9所示。

表 13-9 BaseService 类导入语句

模型类型	父类	导入语句
TensorFlow	TfServingBaseService	from model_service.tfserving_model_service import TfServingBaseService
MXNet	MXNetBaseService	from mms.model_service.mxnet_model_service import MXNetBaseService
PyTorch	PTServingBaseService	from model_service.pytorch_model_service import PTServingBaseService
Caffe	CaffeBaseService	from model_service.caffe_model_service import CaffeBaseService

2. 可以重写的方法有以下几种。

表 13-10 重写方法

方法名	说明
__init__(self, model_name, model_path)	初始化方法，适用于深度学习框架模型。该方法内加载模型及标签等（pytorch和caffe类型模型必须重写，实现模型加载逻辑）。
__init__(self, model_path)	初始化方法，适用于机器学习框架模型。该方法内初始化模型的路径（self.model_path）。在Spark_MLlib中，该方法还会初始化SparkSession（self.spark）。
_preprocess(self, data)	预处理方法，在推理请求前调用，用于将API接口用户原始请求数据转换为模型期望输入数据。
_inference(self, data)	实际推理请求方法（不建议重写，重写后会覆盖ModelArts内置的推理过程，运行自定义的推理逻辑）。
_postprocess(self, data)	后处理方法，在推理请求完成后调用，用于将模型输出转换为API接口输出。

说明

- 用户可以选择重写preprocess和postprocess方法，以实现数据的API输入的预处理和推理结果输出的后处理。
- 重写BaseService继承类的初始化方法init可能导致AI应用“运行异常”。

3. 可以使用的属性为模型所在的本地路径，属性名为“self.model_path”。另外pyspark模型在“customize_service.py”中可以使用“self.spark”获取SparkSession对象。

说明

推理代码中读取文件需要通过绝对路径，绝对路径可以通过self.model_path属性获得模型所在的本地路径。

- 当使用TensorFlow、Caffe、MXNet时，self.model_path为模型文件目录路径，读取文件示例如下：
```
# model目录下放置label.json文件，此处读取
with open(os.path.join(self.model_path, 'label.json')) as f:
    self.label = json.load(f)
```
- 当使用PyTorch、Scikit_Learn、pyspark时，self.model_path为模型文件路径，读取文件示例如下：
```
# model目录下放置label.json文件，此处读取
dir_path = os.path.dirname(os.path.realpath(self.model_path))
with open(os.path.join(dir_path, 'label.json')) as f:
    self.label = json.load(f)
```

4. 预处理方法、实际推理请求方法和后处理方法中的接口传入“data”当前支持两种content-type，即“multipart/form-data”和“application/json”。
   - “multipart/form-data”请求
```
curl -X POST \
  <modelarts-inference-endpoint> \
  -F image1=@cat.jpg \
  -F images2=@horse.jpg
```
     对应的传入data为
```
[
  {
    "image1":{
      "cat.jpg":"<cat..jpg file io>"
    }
  },
  {
    "image2":{
      "horse.jpg":"<horse.jpg file io>"
    }
  }
]
```
   - “application/json”请求
```
curl -X POST \
  <modelarts-inference-endpoint> \
  -d '{
   "images":"base64 encode image"
   }'
```
     对应的传入data为python dict
```
{
   "images":"base64 encode image"

}
```

TensorFlow 的推理脚本示例

TensorFlow MnistService示例如下。更多tensorflow推理代码示例请参考**TensorFlow**、**TensorFlow 2.1**。其他引擎推理代码请参考**PyTorch**、**Caffe**。

- 推理代码
```
from PIL import Image
import numpy as np
from model_service.tfserving_model_service import TfServingBaseService

class mnist_service(TfServingBaseService):

    def _preprocess(self, data):
        preprocessed_data = {}
```

```
        for k, v in data.items():
            for file_name, file_content in v.items():
                image1 = Image.open(file_content)
                image1 = np.array(image1, dtype=np.float32)
                image1.resize((1, 784))
                preprocessed_data[k] = image1

        return preprocessed_data

    def _postprocess(self, data):

        infer_output = {}

        for output_name, result in data.items():

            infer_output["mnist_result"] = result[0].index(max(result[0]))

        return infer_output
```

- 请求
  ```
  curl -X POST \ 在线服务地址 \ -F images=@test.jpg
  ```
- 返回
  ```
  {"mnist_result": 7}
  ```

在上面的代码示例中，完成了将用户表单输入的图片的大小调整，转换为可以适配模型输入的shape。首先通过Pillow库读取“32×32”的图片，调整图片大小为“1×784”以匹配模型输入。在后续处理中，转换模型输出为列表，用于Restful接口输出展示。

XGBoost 的推理脚本示例

更多机器学习引擎的推理代码请参考**Pyspark**、**Scikit Learn**。

```
# coding:utf-8
import collections
import json
import xgboost as xgb
from model_service.python_model_service import XgSklServingBaseService


class user_Service(XgSklServingBaseService):

    # request data preprocess
    def _preprocess(self, data):
        list_data = []
        json_data = json.loads(data, object_pairs_hook=collections.OrderedDict)
        for element in json_data["data"]["req_data"]:
            array = []
            for each in element:
                array.append(element[each])
                list_data.append(array)
        return list_data

    #   predict
    def _inference(self, data):
        xg_model = xgb.Booster(model_file=self.model_path)
        pre_data = xgb.DMatrix(data)
        pre_result = xg_model.predict(pre_data)
        pre_result = pre_result.tolist()
        return pre_result

    # predict result process
    def _postprocess(self, data):
        resp_data = []
        for element in data:
            resp_data.append({"predict_result": element})
        return resp_data
```

自定义推理逻辑的推理脚本示例

首先，需要在配置文件中，定义自己的依赖包，详细示例请参见**使用自定义依赖包的模型配置文件示例**。然后通过如下所示示例代码，实现了“saved_model”格式模型的加载推理。

```
# -*- coding: utf-8 -*-
import json
import os
import threading

import numpy as np
import tensorflow as tf
from PIL import Image

from model_service.tfserving_model_service import TfServingBaseService
import logging

logger = logging.getLogger(__name__)


class MnistService(TfServingBaseService):

    def __init__(self, model_name, model_path):
        self.model_name = model_name
        self.model_path = model_path
        self.model_inputs = {}
        self.model_outputs = {}

        # label文件可以在这里加载,在后处理函数里使用
        # label.txt放在OBS和模型包的目录

        # with open(os.path.join(self.model_path, 'label.txt')) as f:
        #     self.label = json.load(f)

        # 非阻塞方式加载saved_model模型，防止阻塞超时
        thread = threading.Thread(target=self.get_tf_sess)
        thread.start()

    def get_tf_sess(self):
        # 加载saved_model格式的模型

        # session要重用，建议不要用with语句
        sess = tf.Session(graph=tf.Graph())
        meta_graph_def = tf.saved_model.loader.load(sess, [tf.saved_model.tag_constants.SERVING],
self.model_path)
        signature_defs = meta_graph_def.signature_def

        self.sess = sess

        signature = []

        # only one signature allowed
        for signature_def in signature_defs:
            signature.append(signature_def)
        if len(signature) == 1:
            model_signature = signature[0]
        else:
            logger.warning("signatures more than one, use serving_default signature")
            model_signature = tf.saved_model.signature_constants.DEFAULT_SERVING_SIGNATURE_DEF_KEY

        logger.info("model signature: %s", model_signature)

        for signature_name in meta_graph_def.signature_def[model_signature].inputs:
            tensorinfo = meta_graph_def.signature_def[model_signature].inputs[signature_name]
            name = tensorinfo.name
            op = self.sess.graph.get_tensor_by_name(name)
            self.model_inputs[signature_name] = op
```

```
    logger.info("model inputs: %s", self.model_inputs)

    for signature_name in meta_graph_def.signature_def[model_signature].outputs:
        tensorinfo = meta_graph_def.signature_def[model_signature].outputs[signature_name]
        name = tensorinfo.name
        op = self.sess.graph.get_tensor_by_name(name)

        self.model_outputs[signature_name] = op

    logger.info("model outputs: %s", self.model_outputs)

def _preprocess(self, data):
    # https两种请求形式
    # 1. form-data文件格式的请求对应：data = {"请求key值":{"文件名":<文件io>}}
    # 2. json格式对应：data = json.loads("接口传入的json体")
    preprocessed_data = {}

    for k, v in data.items():
        for file_name, file_content in v.items():
            image1 = Image.open(file_content)
            image1 = np.array(image1, dtype=np.float32)
            image1.resize((1, 28, 28))
            preprocessed_data[k] = image1

    return preprocessed_data

def _inference(self, data):

    feed_dict = {}
    for k, v in data.items():
        if k not in self.model_inputs.keys():
            logger.error("input key %s is not in model inputs %s", k, list(self.model_inputs.keys()))
            raise Exception("input key %s is not in model inputs %s" % (k, list(self.model_inputs.keys())))
        feed_dict[self.model_inputs[k]] = v

    result = self.sess.run(self.model_outputs, feed_dict=feed_dict)
    logger.info('predict result : ' + str(result))

    return result

def _postprocess(self, data):
    infer_output = {"mnist_result": []}
    for output_name, results in data.items():

        for result in results:
            infer_output["mnist_result"].append(np.argmax(result))

    return infer_output

def __del__(self):
    self.sess.close()
```

14 模型模板（即将下线）

14.1 模型模板简介

说明

为了更方便用户使用和清晰的获取文档帮助，《AI工程师用户指南》将按照AI开发流程调整拆分为多本文档，在一级导航中呈现，并从用户使用场景对各模块文档进行了优化和改进。“模型模板”的内容已拆分至《**推理部署**》,《AI工程师用户指南》里的“模型模板”即将下线。

相同功能的模型配置信息重复率高，将相同功能的配置整合成一个通用的模板，通过使用该模板，可以方便快捷的导入模型创建AI应用，而不用编写config.json配置文件。简单来说，模板将AI引擎以及模型配置模板化，每种模板对应于1种具体的AI引擎及1种推理模式，借助模板，可以快速导入模型到ModelArts创建AI应用。

背景信息

模板分两大类型：通用类型，非通用类型。

- 非通用类型模板，针对特定的场景所定制的，固定输入输出模式，不可覆盖，如“TensorFlow图像分类模板”，固定使用预置图像处理模式。
- 通用模板，搭载特定的AI引擎以及运行环境，内置的输入输出模式为未定义模式，即不定义具体的输入输出格式，用户需根据模型功能或业务场景重新选择新的输入输出模式来覆盖内置的未定义模式，如图像分类模型应选择预置图像处理模式，而目标检测模型则应选择预置物体检测模式。

  说明

  使用未定义模式的模型将无法部署批量服务。

如何使用模板

您需要先将模型包上传至OBS。模型文件应存放在model目录下，通过该模板创建AI应用时，您需要选择到model这一目录。具体使用方式请参见**从模板中选择元模型**。

支持的模板

- **TensorFlow图像分类模板**

- TensorFlow-py27通用模板
- TensorFlow-py36通用模板
- MXNet-py27通用模板
- MXNet-py36通用模板
- PyTorch-py27通用模板
- PyTorch-py36通用模板
- Caffe-CPU-py27通用模板
- Caffe-GPU-py27通用模板
- Caffe-CPU-py36通用模板
- Caffe-GPU-py36通用模板
- ARM-Ascend模板

支持的输入输出模式

- 预置物体检测模式
- 预置图像处理模式
- 预置预测分析模式
- 未定义模式

14.2 模板说明

14.2.1 TensorFlow 图像分类模板

简介

搭载TensorFlow1.8引擎，运行环境为“python2.7”，适合导入以“SavedModel”格式保存的TensorFlow图像分类模型。该模板使用平台预置的图像处理模式，模式详情参见预置图像处理模式，推理时向模型输入一张“key”为“images”的待处理图片，所以需确保您的模型能处理“key”为“images”的输入。使用该模板导入模型时请选择到包含模型文件的“model”目录。

模板输入

存储在OBS上的TensorFlow模型包，确保您使用的OBS目录与ModelArts在同一区域。模型包的要求请参见模型包示例。

对应的输入输出模式

预置图像处理模式，不可覆盖，即创建模型时不支持选择其他输入输出模式。

模型包规范

- 模型包必须存储在OBS中，且必须以“model”命名。“model”文件夹下面放置模型文件、模型推理代码。

- 模型推理代码文件不是必选文件，如果有，其文件名必须为“customize_service.py”，“model”文件夹下有且只能有1个推理代码文件，模型推理代码编写请参见**模型推理代码编写说明**。
- 使用模板导入的模型包结构如下所示：

```
model/
 |
 |—— 模型文件              //必选，不同的框架，其模型文件格式不同，详细可参考模型包示例。
 |—— 自定义Python包        //可选，用户自有的Python包，在模型推理代码中可以直接引用。
 |—— customize_service.py  //可选，模型推理代码，文件名称必须为“customize_service.py”，否则不视为推理代码。
```

模型包示例

TensorFlow模型包结构

发布该模型时只需要指定到“model”目录。

```
OBS桶/目录名
|—— model    必选，文件夹名称必须为“model”，用于放置模型相关文件。
    |—— <<自定义python包>>    可选，用户自有的Python包，在模型推理代码中可以直接引用。
    |—— saved_model.pb        必选，protocol buffer格式文件，包含该模型的图描述。
    |—— variables             对“*.pb”模型主文件而言必选。文件夹名称必须为“variables”，包含模型的权重偏差等信息。
        |—— variables.index                 必选
        |—— variables.data-00000-of-00001     必选
    |——customize_service.py   可选，模型推理代码，文件名称必须为“customize_service.py”，有且只有1个推理代码文件。“customize_service.py”依赖的“py”文件可以直接放“model”目录下。
```

14.2.2 TensorFlow-py27 通用模板

简介

搭载TensorFlow1.8 AI引擎，运行环境为“python2.7”，内置输入输出模式为未定义模式，请根据模型功能或业务场景重新选择合适的输入输出模式。使用该模板导入模型时请选择到包含模型文件的model目录。

模板输入

存储在OBS上的TensorFlow模型包，确保您使用的OBS目录与ModelArts在同一区域。模型包的要求请参见**模型包示例**。

对应的输入输出模式

未定义模式，可覆盖，即创建模型时支持选择其他输入输出模式。

模型包规范

- 模型包必须存储在OBS中，且必须以“model”命名。“model”文件夹下面放置模型文件、模型推理代码。
- 模型推理代码文件不是必选文件，如果有，其文件名必须为“customize_service.py”，“model”文件夹下有且只能有1个推理代码文件，模型推理代码编写请参见**模型推理代码编写说明**。
- 使用模板导入的模型包结构如下所示：

```
model/
 |
 |—— 模型文件              //必选，不同的框架，其模型文件格式不同，详细可参考模型包示例。
 |—— 自定义Python包        //可选，用户自有的Python包，在模型推理代码中可以直接引用。
```

```
├── customize_service.py  //可选，模型推理代码，文件名称必须为“customize_service.py”，否则不视为推理代码。
```

模型包示例

TensorFlow模型包结构

发布该模型时只需要指定到“model”目录。

```
OBS桶/目录名
├── model    必选，文件夹名称必须为“model”，用于放置模型相关文件。
    ├── <<自定义python包>>    可选，用户自有的Python包，在模型推理代码中可以直接引用。
    ├── saved_model.pb        必选，protocol buffer格式文件，包含该模型的图描述。
    ├── variables             对“*.pb”模型主文件而言必选。文件夹名称必须为“variables”，包含模型的权重偏差等信息。
        ├── variables.index                 必选
        ├── variables.data-00000-of-00001     必选
    ├──customize_service.py   可选，模型推理代码，文件名称必须为“customize_service.py”，有且只有1个推理代码文件。“customize_service.py”依赖的“py”文件可以直接放“model”目录下。
```

14.2.3 TensorFlow-py36 通用模板

简介

搭载TensorFlow1.8 AI引擎，运行环境为“python3.6”，内置输入输出模式为未定义模式，请根据模型功能或业务场景重新选择合适的输入输出模式。使用该模板导入模型时请选择到包含模型文件的model目录。

模板输入

存储在OBS上的TensorFlow模型包，确保您使用的OBS目录与ModelArts在同一区域。模型包的要求请参见**模型包示例**。

对应的输入输出模式

未定义模式，可覆盖，即创建模型时支持选择其他输入输出模式。

模型包规范

- 模型包必须存储在OBS中，且必须以“model”命名。“model”文件夹下面放置模型文件、模型推理代码。
- 模型推理代码文件不是必选文件，如果有，其文件名必须为“customize_service.py”，“model”文件夹下有且只能有1个推理代码文件，模型推理代码编写请参见**模型推理代码编写说明**。
- 使用模板导入的模型包结构如下所示：

```
model/
│
├── 模型文件              //必选，不同的框架，其模型文件格式不同，详细可参考模型包示例。
├── 自定义Python包        //可选，用户自有的Python包，在模型推理代码中可以直接引用。
├── customize_service.py  //可选，模型推理代码，文件名称必须为“customize_service.py”，否则不视为推理代码。
```

模型包示例

TensorFlow模型包结构

发布该模型时只需要指定到“model”目录。

```
OBS桶/目录名
├── model    必选，文件夹名称必须为“model”，用于放置模型相关文件。
    ├── <<自定义python包>>    可选，用户自有的Python包，在模型推理代码中可以直接引用。
    ├── saved_model.pb        必选，protocol buffer格式文件，包含该模型的图描述。
    ├── variables             对“*.pb”模型主文件而言必选。文件夹名称必须为“variables”，包含模型的权重偏差等信息。
        ├── variables.index                 必选
        ├── variables.data-00000-of-00001     必选
    ├──customize_service.py   可选，模型推理代码，文件名称必须为“customize_service.py”，有且只有1个推理代码文件。“customize_service.py”依赖的“py”文件可以直接放“model”目录下。
```

14.2.4 MXNet-py27 通用模板

简介

搭载MXNet1.2.1 AI引擎，运行环境为“python2.7”，内置输入输出模式为未定义模式，请根据模型功能或业务场景重新选择合适的输入输出模式。使用该模板导入模型时请选择到包含模型文件的model目录。

模板输入

存储在OBS上的MXNet模型包，确保您使用的OBS目录与ModelArts在同一区域。模型包的要求请参见模型包示例。

对应的输入输出模式

未定义模式，可覆盖，即创建模型时支持选择其他输入输出模式。

模型包规范

- 模型包必须存储在OBS中，且必须以“model”命名。“model”文件夹下面放置模型文件、模型推理代码。
- 模型推理代码文件不是必选文件，如果有，其文件名必须为“customize_service.py”，“model”文件夹下有且只能有1个推理代码文件，模型推理代码编写请参见模型推理代码编写说明。
- 使用模板导入的模型包结构如下所示：

```
model/
 |
 ├── 模型文件              //必选，不同的框架，其模型文件格式不同，详细可参考模型包示例。
 ├── 自定义Python包        //可选，用户自有的Python包，在模型推理代码中可以直接引用。
 ├── customize_service.py  //可选，模型推理代码，文件名称必须为“customize_service.py”，否则不视为推理代码。
```

模型包示例

MXNet模型包结构

发布该模型时只需要指定到“model”目录。

```
OBS桶/目录名
├── model    必选，文件夹名称必须为“model”，用于放置模型相关文件。
    ├── <<自定义python包>>      可选，用户自有的Python包，在模型推理代码中可以直接引用。
    ├── resnet-50-symbol.json    必选，模型定义文件，包含模型的神经网络描述，
    ├── resnet-50-0000.params    必选，模型变量参数文件，包含参数和权重信息。
    ├──customize_service.py      可选，模型推理代码，文件名称必须为“customize_service.py”，有且只有1个推理代码文件。“customize_service.py”依赖的“py”文件可以直接放“model”目录下。
```

14.2.5 MXNet-py36 通用模板

简介

搭载MXNet1.2.1 AI引擎，运行环境为“python3.6”，内置输入输出模式为未定义模式，请根据模型功能或业务场景重新选择合适的输入输出模式。使用该模板导入模型时请选择到包含模型文件的model目录。

模板输入

存储在OBS上的MXNet模型包，确保您使用的OBS目录与ModelArts在同一区域。模型包的要求请参见模型包示例。

对应的输入输出模式

未定义模式，可覆盖，即创建模型时支持选择其他输入输出模式。

模型包规范

- 模型包必须存储在OBS中，且必须以“model”命名。“model”文件夹下面放置模型文件、模型推理代码。
- 模型推理代码文件不是必选文件，如果有，其文件名必须为“customize_service.py”，“model”文件夹下有且只能有1个推理代码文件，模型推理代码编写请参见模型推理代码编写说明。
- 使用模板导入的模型包结构如下所示：

```
model/
 |
 |─── 模型文件              //必选，不同的框架，其模型文件格式不同，详细可参考模型包示例。
 |─── 自定义Python包        //可选，用户自有的Python包，在模型推理代码中可以直接引用。
 |─── customize_service.py  //可选，模型推理代码，文件名称必须为“customize_service.py”，否则不视为推理代码。
```

模型包示例

MXNet模型包结构

发布该模型时只需要指定到“model”目录。

```
OBS桶/目录名
|─── model    必选，文件夹名称必须为“model”，用于放置模型相关文件。
    |─── <<自定义python包>>      可选，用户自有的Python包，在模型推理代码中可以直接引用。
    |─── resnet-50-symbol.json    必选，模型定义文件，包含模型的神经网络描述，
    |─── resnet-50-0000.params    必选，模型变量参数文件，包含参数和权重信息。
    |───customize_service.py      可选，模型推理代码，文件名称必须为“customize_service.py”，有且只有1个推理代码文件。“customize_service.py”依赖的“py”文件可以直接放“model”目录下。
```

14.2.6 PyTorch-py27 通用模板

简介

搭载PyTorch1.0AI引擎，运行环境为“python2.7”，内置输入输出模式为未定义模式，请根据模型功能或业务场景重新选择合适的输入输出模式。使用该模板导入模型时请选择到包含模型文件的model目录。

模板输入

存储在OBS上的PyTorch模型包，确保您使用的OBS目录与ModelArts在同一区域。模型包的要求请参见模型包示例。

对应的输入输出模式

未定义模式，可覆盖，即创建模型时支持选择其他输入输出模式。

模型包规范

- 模型包必须存储在OBS中，且必须以“model”命名。“model”文件夹下面放置模型文件、模型推理代码。
- 模型推理代码文件不是必选文件，如果有，其文件名必须为“customize_service.py”，“model”文件夹下有且只能有1个推理代码文件，模型推理代码编写请参见模型推理代码编写说明。
- 使用模板导入的模型包结构如下所示：

```
model/
 |
 |—— 模型文件              //必选，不同的框架，其模型文件格式不同，详细可参考模型包示例。
 |—— 自定义Python包        //可选，用户自有的Python包，在模型推理代码中可以直接引用。
 |—— customize_service.py  //可选，模型推理代码，文件名称必须为“customize_service.py”，否则不视为推理代码。
```

模型包示例

PyTorch模型包结构

发布该模型时只需要指定到“model”目录。

```
OBS桶/目录名
|—— model    必选，文件夹名称必须为“model”，用于放置模型相关文件。
   |—— <<自定义Python包>>    可选，用户自有的Python包，在模型推理代码中可以直接引用。
   |—— resnet50.pth          必选，pytorch模型保存文件，存有权重变量等信息。
   |——customize_service.py   可选，模型推理代码，文件名称必须为“customize_service.py”，有且只有1个推理代码文件。“customize_service.py”依赖的“py”文件可以直接放“model”目录下。
```

14.2.7 PyTorch-py36 通用模板

简介

搭载PyTorch1.0AI引擎，运行环境为“python3.6”，内置输入输出模式为未定义模式，请根据模型功能或业务场景重新选择合适的输入输出模式。使用该模板导入模型时请选择到包含模型文件的model目录。

模板输入

存储在OBS上的PyTorch模型包，确保您使用的OBS目录与ModelArts在同一区域。模型包的要求请参见模型包示例。

对应的输入输出模式

未定义模式，可覆盖，即创建模型时支持选择其他输入输出模式。

模型包规范

- 模型包必须存储在OBS中，且必须以“model”命名。“model”文件夹下面放置模型文件、模型推理代码。
- 模型推理代码文件不是必选文件，如果有，其文件名必须为“customize_service.py”，“model”文件夹下有且只能有1个推理代码文件，模型推理代码编写请参见模型推理代码编写说明。
- 使用模板导入的模型包结构如下所示：

```
model/
 |
 |—— 模型文件              //必选，不同的框架，其模型文件格式不同，详细可参考模型包示例。
 |—— 自定义Python包        //可选，用户自有的Python包，在模型推理代码中可以直接引用。
 |—— customize_service.py  //可选，模型推理代码，文件名称必须为“customize_service.py”，否则不视为推理代码。
```

模型包示例

PyTorch模型包结构

发布该模型时只需要指定到“model”目录。

```
OBS桶/目录名
|—— model    必选，文件夹名称必须为“model”，用于放置模型相关文件。
   |—— <<自定义Python包>>    可选，用户自有的Python包，在模型推理代码中可以直接引用。
   |—— resnet50.pth          必选，pytorch模型保存文件，存有权重变量等信息。
   |——customize_service.py   可选，模型推理代码，文件名称必须为“customize_service.py”，有且只有1个推理代码文件。“customize_service.py”依赖的“py”文件可以直接放“model”目录下。
```

14.2.8 Caffe-CPU-py27 通用模板

简介

搭载Caffe1.0 CPU版 AI引擎，运行环境为“python2.7”，内置输入输出模式为未定义模式，请根据模型功能或业务场景重新选择合适的输入输出模式。使用该模板导入模型时请选择到包含模型文件的model目录那一层。

模板输入

存储在OBS上的Caffe模型包，确保您使用的OBS目录与ModelArts在同一区域。模型包的要求请参见模型包示例。

对应的输入输出模式

未定义模式，可覆盖，即创建模型时支持选择其他输入输出模式。

模型包规范

- 模型包必须存储在OBS中，且必须以“model”命名。“model”文件夹下面放置模型文件、模型推理代码。
- 模型推理代码文件不是必选文件，如果有，其文件名必须为“customize_service.py”，“model”文件夹下有且只能有1个推理代码文件，模型推理代码编写请参见模型推理代码编写说明。
- 使用模板导入的模型包结构如下所示：

```
model/
 |
 |—— 模型文件              //必选，不同的框架，其模型文件格式不同，详细可参考模型包示例。
```

```
 ├── 自定义Python包          //可选，用户自有的Python包，在模型推理代码中可以直接引用。
 ├── customize_service.py  //可选，模型推理代码，文件名称必须为“customize_service.py”，否则不视为推理代码。
```

模型包示例

Caffe模型包结构

发布该模型时只需要指定到“model”目录。

```
OBS桶/目录名
├── model    必选，文件夹名称必须为“model”，用于放置模型相关文件。
   ├── <<自定义python包>>      可选，用户自有的Python包，在模型推理代码中可以直接引用。
   ├── deploy.prototxt         必选，caffe模型保存文件，存有模型网络结构等信息。
   ├── resnet.caffemodel       必选，caffe模型保存文件，存有权重变量等信息。
   ├── customize_service.py    可选，模型推理代码，文件名称必须为“customize_service.py”，有且只有1个推理代码文件。“customize_service.py”依赖的“py”文件可以直接放“model”目录下。
```

14.2.9 Caffe-GPU-py27 通用模板

简介

搭载Caffe1.0 GPU版 AI引擎，运行环境为“python2.7”，内置输入输出模式为未定义模式，请根据模型功能或业务场景重新选择合适的输入输出模式。使用该模板导入模型时请选择到包含模型文件的model目录。

模板输入

存储在OBS上的Caffe模型包，确保您使用的OBS目录与ModelArts在同一区域。模型包的要求请参见模型包示例。

对应的输入输出模式

未定义模式，可覆盖，即创建模型时支持选择其他输入输出模式。

模型包规范

- 模型包必须存储在OBS中，且必须以“model”命名。“model”文件夹下面放置模型文件、模型推理代码。
- 模型推理代码文件不是必选文件，如果有，其文件名必须为“customize_service.py”，“model”文件夹下有且只能有1个推理代码文件，模型推理代码编写请参见模型推理代码编写说明。
- 使用模板导入的模型包结构如下所示：

```
model/
 │
 ├── 模型文件              //必选，不同的框架，其模型文件格式不同，详细可参考模型包示例。
 ├── 自定义Python包          //可选，用户自有的Python包，在模型推理代码中可以直接引用。
 ├── customize_service.py  //可选，模型推理代码，文件名称必须为“customize_service.py”，否则不视为推理代码。
```

模型包示例

Caffe模型包结构

发布该模型时只需要指定到“model”目录。

```
OBS桶/目录名
├── model    必选，文件夹名称必须为“model”，用于放置模型相关文件。
   ├── <<自定义python包>>      可选，用户自有的Python包，在模型推理代码中可以直接引用。
```

```
    |—— deploy.prototxt        必选，caffe模型保存文件，存有模型网络结构等信息。
    |—— resnet.caffemodel      必选，caffe模型保存文件，存有权重变量等信息。
    |—— customize_service.py   可选，模型推理代码，文件名称必须为“customize_service.py”，有且只有1个推理代码文件。“customize_service.py”依赖的“py”文件可以直接放“model”目录下。
```

14.2.10 Caffe-CPU-py36 通用模板

简介

搭载Caffe1.0 CPU版 AI引擎，运行环境为“python3.6”，内置输入输出模式为未定义模式，请根据模型功能或业务场景重新选择合适的输入输出模式。使用该模板导入模型时请选择到包含模型文件的model目录。

模板输入

存储在OBS上的Caffe模型包，确保您使用的OBS目录与ModelArts在同一区域。模型包的要求请参见**模型包示例**。

对应的输入输出模式

未定义模式，可覆盖，即创建模型时支持选择其他输入输出模式。

模型包规范

- 模型包必须存储在OBS中，且必须以“model”命名。“model”文件夹下面放置模型文件、模型推理代码。
- 模型推理代码文件不是必选文件，如果有，其文件名必须为“customize_service.py”，“model”文件夹下有且只能有1个推理代码文件，模型推理代码编写请参见**模型推理代码编写说明**。
- 使用模板导入的模型包结构如下所示：

```
model/
 |
 |——— 模型文件              //必选，不同的框架，其模型文件格式不同，详细可参考模型包示例。
 |——— 自定义Python包        //可选，用户自有的Python包，在模型推理代码中可以直接引用。
 |——— customize_service.py  //可选，模型推理代码，文件名称必须为“customize_service.py”，否则不视为推理代码。
```

模型包示例

Caffe模型包结构

发布该模型时只需要指定到“model”目录。

```
OBS桶/目录名
|—— model    必选，文件夹名称必须为“model”，用于放置模型相关文件。
    |—— <<自定义python包>>     可选，用户自有的Python包，在模型推理代码中可以直接引用。
    |—— deploy.prototxt        必选，caffe模型保存文件，存有模型网络结构等信息。
    |—— resnet.caffemodel      必选，caffe模型保存文件，存有权重变量等信息。
    |—— customize_service.py   可选，模型推理代码，文件名称必须为“customize_service.py”，有且只有1个推理代码文件。“customize_service.py”依赖的“py”文件可以直接放“model”目录下。
```

14.2.11 Caffe-GPU-py36 通用模板

简介

搭载Caffe1.0 GPU版 AI引擎，运行环境为“python3.6”，内置输入输出模式为未定义模式，请根据模型功能或业务场景重新选择合适的输入输出模式。使用该模板导入模型时请选择到包含模型文件的model目录。

模板输入

存储在OBS上的Caffe模型包，确保您使用的OBS目录与ModelArts在同一区域。模型包的要求请参见模型包示例。

对应的输入输出模式

未定义模式，可覆盖，即创建模型时支持选择其他输入输出模式。

模型包规范

- 模型包必须存储在OBS中，且必须以“model”命名。“model”文件夹下面放置模型文件、模型推理代码。
- 模型推理代码文件不是必选文件，如果有，其文件名必须为“customize_service.py”，“model”文件夹下有且只能有1个推理代码文件，模型推理代码编写请参见模型推理代码编写说明。
- 使用模板导入的模型包结构如下所示：

```
model/
 |
 |—— 模型文件                //必选，不同的框架，其模型文件格式不同，详细可参考模型包示例。
 |—— 自定义Python包          //可选，用户自有的Python包，在模型推理代码中可以直接引用。
 |—— customize_service.py  //可选，模型推理代码，文件名称必须为“customize_service.py”，否则不视为推理代码。
```

模型包示例

Caffe模型包结构

发布该模型时只需要指定到“model”目录。

```
OBS桶/目录名
|—— model    必选，文件夹名称必须为“model”，用于放置模型相关文件。
   |—— <<自定义python包>>      可选，用户自有的Python包，在模型推理代码中可以直接引用。
   |—— deploy.prototxt         必选，caffe模型保存文件，存有模型网络结构等信息。
   |—— resnet.caffemodel       必选，caffe模型保存文件，存有权重变量等信息。
   |—— customize_service.py    可选，模型推理代码，文件名称必须为“customize_service.py”，有且只有1个推理代码文件。“customize_service.py”依赖的“py”文件可以直接放“model”目录下。
```

14.2.12 ARM-Ascend 模板

简介

搭载MindSpore AI引擎，运行环境为“python3.5”，内置输入输出模式为未定义模式，请根据模型功能或业务场景重新选择合适的输入输出模式。使用该模板导入模型时请选择到包含模型文件的model目录。

模板输入

存储在OBS上的om模型包，确保您使用的OBS目录与ModelArts在同一区域。模型包的要求请参见模型包示例。

对应的输入输出模式

未定义模式，可覆盖，即创建模型时支持选择其他输入输出模式。

模型包规范

- 模型包必须存储在OBS中，且必须以“model”命名。“model”文件夹下面放置模型文件、模型推理代码。
- 模型推理代码文件不是必选文件，如果有，其文件名必须为“customize_service.py”，“model”文件夹下有且只能有1个推理代码文件，模型推理代码编写请参见**模型推理代码编写说明**。
- 使用模板导入的模型包结构如下所示：

```
model/
 |
 |—— 模型文件              //必选，不同的框架，其模型文件格式不同，详细可参考模型包示例。
 |—— 自定义Python包        //可选，用户自有的Python包，在模型推理代码中可以直接引用。
 |—— customize_service.py  //可选，模型推理代码，文件名称必须为“customize_service.py”，否则不视为推理代码。
```

模型包示例

om模型包结构

发布该模型时只需要指定到“model”目录。

```
OBS桶/目录名
|—— model    必选，文件夹名称必须为“model”，用于放置模型相关文件。
    |—— <<自定义python包>>    可选，用户自有的Python包，在模型推理代码中可以直接引用。
    |—— model.om              必选，protocol buffer格式文件，包含该模型的图描述。
    |——customize_service.py   可选，模型推理代码，文件名称必须为“customize_service.py”，有且只有1个推理代码文件。“customize_service.py”依赖的“py”文件可以直接放“model”目录下。
```

14.3 输入输出模式说明

14.3.1 预置物体检测模式

输入

系统预置物体检测输入输出模式，适用于物体检测的模型，使用该模式的模型被标识为物体检测模型。预测请求路径“/”，请求协议为“HTTP”，请求方法为“POST”，调用方需采用“multipart/form-data”内容类型，以“key”为“images”，“type”为“file”的格式输入待处理图片。选择该模式时需确保您的模型能处理key为images的输入数据。

输出

推理结果以“JSON”体的形式返回，具体字段请参见**表14-1**。

表 14-1 参数说明

字段名	类型	描述
detection_classes	字符串数组	输出物体的检测类别列表，如["yunbao","cat"]

字段名	类型	描述
detection_boxes	数组，元素为浮点数数组	输出物体的检测框坐标列表，坐标表示为 $[y_{min}, x_{min}, y_{max}, x_{max}]$
detection_scores	浮点数数组	输出每种检测列表的置信度，用来衡量识别的准确度。

推理结果的“JSON Schema”表示如下：

```
{
    "type": "object",
    "properties": {
        "detection_classes": {
            "items": {
                "type": "string"
            },
            "type": "array"
        },
        "detection_boxes": {
            "items": {
                "minItems": 4,
                "items": {
                    "type": "number"
                },
                "type": "array",
                "maxItems": 4
            },
            "type": "array"
        },
        "detection_scores": {
            "items": {
                "type": "string"
            },
            "type": "array"
        }
    }
}
```

请求样例

该模式下的推理方式均为输入一张待处理图片，推理结果以“JSON”格式返回。示例如下：

- 页面预测

  在服务详情的“预测”页签，上传需要检测的图片，单击“预测”即可获取检测结果。

- Postman调REST接口预测

  部署上线成功后，您可以从服务详情页的调用指南中获取预测接口地址，预测步骤如下：

  – 选择“Headers”设置请求头部，“Content-Type”的值设为“multipart/form-data”，“X-Auth-Token”的值设为用户实际获取的token值。

图 14-1 设置请求头部

– 选择“Body”设置请求体，“key”选择为“images”，选择为“File”类型，接着通过选择文件按钮选择需要处理的图片，最后单击“send”，发送您的预测请求。

图 14-2 设置请求体

14.3.2 预置图像处理模式

输入

系统预置图像处理输入输出模式，适用于图像分类、物体检测和图像语义分割等图像处理模型。预测请求路径“/”，请求协议为“HTTPS”，请求方法为“POST”，调用方需采用“multipart/form-data”内容类型，以“key”为“images”，“type”为“file”的格式输入待处理图片。选择该模式时需确保您的模型能处理key为images的输入数据。

输出

推理结果以“JSON”体的形式返回，“JSON”的具体字段由模型决定。

请求样例

该模式下的推理方式均为输入一张待处理图片，响应的“JSON”根据模型改变而改变。示例如下：

- 页面预测
- Postman调REST接口预测

  部署上线成功后，您可以从服务详情页的调用指南中获取预测接口地址。选择“Body”设置请求体，“key”选择为“images”，选择为“File”类型，接着通过选择文件按钮选择需要处理的图片，最后单击“send”，发送您的预测请求。

图 14-3 调用 REST 接口

14.3.3 预置预测分析模式

输入

系统预置预测分析输入输出模式，适用于预测分析的模型，使用该模式的模型将被标识为预测分析模型。预测请求路径“/”，请求协议为“HTTP”，请求方法为“POST”，调用方需采用“application/json”内容类型，发送预测请求，请求体以“JSON”格式表示，“JSON”字段说明请参见表14-2。选择该模式时需确保您的模型能处理符合该输入“JSON Schema”格式的输入数据。“JSON Schema”格式说明请参考官方指导。

表 14-2 JSON 字段说明

字段名	类型	描述
data	Data结构	包含预测数据。“Data结构”说明请参见表14-3。

表 14-3 Data 结构说明

字段名	类型	描述
req_data	ReqData结构数组	预测数据列表。

“ReqData”，是“Object”类型，表示预测数据，数据的具体组成结构由业务场景决定。使用该模式的模型，其自定义的推理代码中的预处理逻辑应能正确处理模式所定义的输入数据格式。

预测请求的“JSON Schema”表示如下：

```
{
    "type": "object",
    "properties": {
        "data": {
            "type": "object",
            "properties": {
                "req_data": {
                    "items": [{
                        "type": "object",
                        "properties": {}
                    }],
                    "type": "array"
                }
            }
        }
    }
}
```

输出

预测结果以“JSON”格式返回，“JSON”字段说明请参见表14-4。

表 14-4 JSON 字段说明

字段名	类型	描述
data	Data结构	包含预测数据。“Data结构”说明请参见表14-5。

表 14-5 Data 结构说明

字段名	类型	描述
resp_data	RespData结构数组	预测结果列表。

与“ReqData”一样，“RespData”也是“Object”类型，表示预测结果，其具体组成结构由业务场景决定。建议使用该模式的模型，其自定义的推理代码中的后处理逻辑应输出符合模式所定义的输出格式的数据。

预测结果的“JSON Schema”表示如下：

```
{
    "type": "object",
    "properties": {
        "data": {
            "type": "object",
            "properties": {
                "resp_data": {
                    "type": "array",
                    "items": [{
                        "type": "object",
                        "properties": {}
                    }]
                }
            }
        }
    }
}
```

请求样例

该模式下的推理方式均为输入“JSON”格式的待预测数据，预测结果以“JSON”格式返回。示例如下：

- 页面预测

  在服务详情的“预测”页签，输入预测代码，单击“预测”即可获取检测结果。

- Postman调REST接口预测

  部署上线成功后，您可以从服务详情页的调用指南中获取预测接口地址，预测步骤如下：

  – 选择“Headers”设置请求头部，“Content-Type”的值设为“application/json”，“X-Auth-Token”的值设为用户实际获取的token值。

图 14-4 预测设置请求头部

– 选择“Body”设置请求体，编辑需要预测的数据，最后单击“send”，发送您的预测请求。

14.3.4 未定义模式

描述

未定义的模式，即不定义具体的输入输出格式，请求的输入输出完全由模型决定。当现有的输入输出模式不适合模型的场景时，才考虑选择该模式。使用未定义模式导入的模型无法部署批量服务，同时服务的预测界面可能无法正常工作。

输入

不限。

输出

不限。

请求样例

未定义模式没有特定的请求样例，请求的输入输出完全由模型决定。

15 自定义脚本代码示例

15.1 TensorFlow

TensorFlow存在两种接口类型，keras接口和tf接口，其训练和保存模型的代码存在差异，但是推理代码编写方式一致。

训练模型（keras 接口）

```
from keras.models import Sequential
model = Sequential()
from keras.layers import Dense
import tensorflow as tf

# 导入训练数据集
mnist = tf.keras.datasets.mnist
(x_train, y_train),(x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

print(x_train.shape)

from keras.layers import Dense
from keras.models import Sequential
import keras
from keras.layers import Dense, Activation, Flatten, Dropout

# 定义模型网络
model = Sequential()
model.add(Flatten(input_shape=(28,28)))
model.add(Dense(units=5120,activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(units=10, activation='softmax'))

# 定义优化器，损失函数等
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
# 训练
model.fit(x_train, y_train, epochs=2)
# 评估
model.evaluate(x_test, y_test)
```

保存模型（keras 接口）

```
from keras import backend as K

# K.get_session().run(tf.global_variables_initializer())

# 定义预测接口的inputs和outputs
# inputs和outputs字典的key值会作为模型输入输出tensor的索引键
# 模型输入输出定义需要和推理自定义脚本相匹配
predict_signature = tf.saved_model.signature_def_utils.predict_signature_def(
    inputs={"images" : model.input},
    outputs={"scores" : model.output}
)

# 定义保存路径
builder = tf.saved_model.builder.SavedModelBuilder('./mnist_keras/')

builder.add_meta_graph_and_variables(

    sess = K.get_session(),
    # 推理部署需要定义tf.saved_model.tag_constants.SERVING标签
    tags=[tf.saved_model.tag_constants.SERVING],
    """
    signature_def_map：items只能有一个，或者需要定义相应的key为
    tf.saved_model.signature_constants.DEFAULT_SERVING_SIGNATURE_DEF_KEY
    """
    signature_def_map={
        tf.saved_model.signature_constants.DEFAULT_SERVING_SIGNATURE_DEF_KEY:
            predict_signature
    }

)
builder.save()
```

训练模型（tf 接口）

```
from __future__ import print_function

import gzip
import os
import urllib

import numpy
import tensorflow as tf
from six.moves import urllib

# 训练数据来源于yann lecun官方网站http://yann.lecun.com/exdb/mnist/
SOURCE_URL = 'http://yann.lecun.com/exdb/mnist/'
TRAIN_IMAGES = 'train-images-idx3-ubyte.gz'
TRAIN_LABELS = 'train-labels-idx1-ubyte.gz'
TEST_IMAGES = 't10k-images-idx3-ubyte.gz'
TEST_LABELS = 't10k-labels-idx1-ubyte.gz'
VALIDATION_SIZE = 5000


def maybe_download(filename, work_directory):
    """Download the data from Yann's website, unless it's already here."""
    if not os.path.exists(work_directory):
        os.mkdir(work_directory)
    filepath = os.path.join(work_directory, filename)
    if not os.path.exists(filepath):
        filepath, _ = urllib.request.urlretrieve(SOURCE_URL + filename, filepath)
        statinfo = os.stat(filepath)
        print('Successfully downloaded %s %d bytes.' % (filename, statinfo.st_size))
    return filepath


def _read32(bytestream):
    dt = numpy.dtype(numpy.uint32).newbyteorder('>')
```

```
    return numpy.frombuffer(bytestream.read(4), dtype=dt)[0]


def extract_images(filename):
    """Extract the images into a 4D uint8 numpy array [index, y, x, depth]."""
    print('Extracting %s' % filename)
    with gzip.open(filename) as bytestream:
        magic = _read32(bytestream)
        if magic != 2051:
            raise ValueError(
                'Invalid magic number %d in MNIST image file: %s' %
                (magic, filename))
        num_images = _read32(bytestream)
        rows = _read32(bytestream)
        cols = _read32(bytestream)
        buf = bytestream.read(rows * cols * num_images)
        data = numpy.frombuffer(buf, dtype=numpy.uint8)
        data = data.reshape(num_images, rows, cols, 1)
        return data


def dense_to_one_hot(labels_dense, num_classes=10):
    """Convert class labels from scalars to one-hot vectors."""
    num_labels = labels_dense.shape[0]
    index_offset = numpy.arange(num_labels) * num_classes
    labels_one_hot = numpy.zeros((num_labels, num_classes))
    labels_one_hot.flat[index_offset + labels_dense.ravel()] = 1
    return labels_one_hot


def extract_labels(filename, one_hot=False):
    """Extract the labels into a 1D uint8 numpy array [index]."""
    print('Extracting %s' % filename)
    with gzip.open(filename) as bytestream:
        magic = _read32(bytestream)
        if magic != 2049:
            raise ValueError(
                'Invalid magic number %d in MNIST label file: %s' %
                (magic, filename))
        num_items = _read32(bytestream)
        buf = bytestream.read(num_items)
        labels = numpy.frombuffer(buf, dtype=numpy.uint8)
        if one_hot:
            return dense_to_one_hot(labels)
        return labels


class DataSet(object):
    """Class encompassing test, validation and training MNIST data set."""

    def __init__(self, images, labels, fake_data=False, one_hot=False):
        """Construct a DataSet. one_hot arg is used only if fake_data is true."""

        if fake_data:
            self._num_examples = 10000
            self.one_hot = one_hot
        else:
            assert images.shape[0] == labels.shape[0], (
                    'images.shape: %s labels.shape: %s' % (images.shape,
                                                           labels.shape))
            self._num_examples = images.shape[0]

            # Convert shape from [num examples, rows, columns, depth]
            # to [num examples, rows*columns] (assuming depth == 1)
            assert images.shape[3] == 1
            images = images.reshape(images.shape[0],
                                    images.shape[1] * images.shape[2])
            # Convert from [0, 255] -> [0.0, 1.0].
            images = images.astype(numpy.float32)
```

```
            images = numpy.multiply(images, 1.0 / 255.0)
        self._images = images
        self._labels = labels
        self._epochs_completed = 0
        self._index_in_epoch = 0

    @property
    def images(self):
        return self._images

    @property
    def labels(self):
        return self._labels

    @property
    def num_examples(self):
        return self._num_examples

    @property
    def epochs_completed(self):
        return self._epochs_completed

    def next_batch(self, batch_size, fake_data=False):
        """Return the next `batch_size` examples from this data set."""
        if fake_data:
            fake_image = [1] * 784
            if self.one_hot:
                fake_label = [1] + [0] * 9
            else:
                fake_label = 0
            return [fake_image for _ in range(batch_size)], [
                fake_label for _ in range(batch_size)
            ]
        start = self._index_in_epoch
        self._index_in_epoch += batch_size
        if self._index_in_epoch > self._num_examples:
            # Finished epoch
            self._epochs_completed += 1
            # Shuffle the data
            perm = numpy.arange(self._num_examples)
            numpy.random.shuffle(perm)
            self._images = self._images[perm]
            self._labels = self._labels[perm]
            # Start next epoch
            start = 0
            self._index_in_epoch = batch_size
            assert batch_size <= self._num_examples
        end = self._index_in_epoch
        return self._images[start:end], self._labels[start:end]


def read_data_sets(train_dir, fake_data=False, one_hot=False):
    """Return training, validation and testing data sets."""

    class DataSets(object):
        pass

    data_sets = DataSets()

    if fake_data:
        data_sets.train = DataSet([], [], fake_data=True, one_hot=one_hot)
        data_sets.validation = DataSet([], [], fake_data=True, one_hot=one_hot)
        data_sets.test = DataSet([], [], fake_data=True, one_hot=one_hot)
        return data_sets

    local_file = maybe_download(TRAIN_IMAGES, train_dir)
    train_images = extract_images(local_file)

    local_file = maybe_download(TRAIN_LABELS, train_dir)
```

```
    train_labels = extract_labels(local_file, one_hot=one_hot)

    local_file = maybe_download(TEST_IMAGES, train_dir)
    test_images = extract_images(local_file)

    local_file = maybe_download(TEST_LABELS, train_dir)
    test_labels = extract_labels(local_file, one_hot=one_hot)

    validation_images = train_images[:VALIDATION_SIZE]
    validation_labels = train_labels[:VALIDATION_SIZE]
    train_images = train_images[VALIDATION_SIZE:]
    train_labels = train_labels[VALIDATION_SIZE:]

    data_sets.train = DataSet(train_images, train_labels)
    data_sets.validation = DataSet(validation_images, validation_labels)
    data_sets.test = DataSet(test_images, test_labels)
    return data_sets

training_iteration = 1000

modelarts_example_path =  './modelarts-mnist-train-save-deploy-example'

export_path = modelarts_example_path + '/model/'
data_path = './'

print('Training model...')
mnist = read_data_sets(data_path, one_hot=True)
sess = tf.InteractiveSession()
serialized_tf_example = tf.placeholder(tf.string, name='tf_example')
feature_configs = {'x': tf.FixedLenFeature(shape=[784], dtype=tf.float32), }
tf_example = tf.parse_example(serialized_tf_example, feature_configs)
x = tf.identity(tf_example['x'], name='x')  # use tf.identity() to assign name
y_ = tf.placeholder('float', shape=[None, 10])
w = tf.Variable(tf.zeros([784, 10]))
b = tf.Variable(tf.zeros([10]))
sess.run(tf.global_variables_initializer())
y = tf.nn.softmax(tf.matmul(x, w) + b, name='y')
cross_entropy = -tf.reduce_sum(y_ * tf.log(y))
train_step = tf.train.GradientDescentOptimizer(0.01).minimize(cross_entropy)
values, indices = tf.nn.top_k(y, 10)
table = tf.contrib.lookup.index_to_string_table_from_tensor(
    tf.constant([str(i) for i in range(10)]))
prediction_classes = table.lookup(tf.to_int64(indices))
for _ in range(training_iteration):
    batch = mnist.train.next_batch(50)
    train_step.run(feed_dict={x: batch[0], y_: batch[1]})
correct_prediction = tf.equal(tf.argmax(y, 1), tf.argmax(y_, 1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, 'float'))
print('training accuracy %g' % sess.run(
    accuracy, feed_dict={
        x: mnist.test.images,
        y_: mnist.test.labels
    }))
print('Done training!')
```

保存模型（tf 接口）

```
# 导出模型
# 模型需要采用saved_model接口保存
print('Exporting trained model to', export_path)
builder = tf.saved_model.builder.SavedModelBuilder(export_path)

tensor_info_x = tf.saved_model.utils.build_tensor_info(x)
tensor_info_y = tf.saved_model.utils.build_tensor_info(y)

# 定义预测接口的inputs和outputs
# inputs和outputs字典的key值会作为模型输入输出tensor的索引键
# 模型输入输出定义需要和推理自定义脚本相匹配
prediction_signature = (
```

```
    tf.saved_model.signature_def_utils.build_signature_def(
        inputs={'images': tensor_info_x},
        outputs={'scores': tensor_info_y},
        method_name=tf.saved_model.signature_constants.PREDICT_METHOD_NAME))

legacy_init_op = tf.group(tf.tables_initializer(), name='legacy_init_op')
builder.add_meta_graph_and_variables(
    # tag设为serve/tf.saved_model.tag_constants.SERVING
    sess, [tf.saved_model.tag_constants.SERVING],
    signature_def_map={
        'predict_images':
            prediction_signature,
    },
    legacy_init_op=legacy_init_op)

builder.save()

print('Done exporting!')
```

推理代码（keras 接口和 tf 接口）

推理代码需要继承自BaseService类，不同类型的模型父类导入语句如请参考**表13-9**。

```
from PIL import Image
import numpy as np
from model_service.tfserving_model_service import TfServingBaseService


class mnist_service(TfServingBaseService):

    # 预处理中处理用户HTTPS接口输入匹配模型输入
    # 对应上述训练部分的模型输入为{"images":<array>}
    def _preprocess(self, data):

        preprocessed_data = {}
        images = []
        # 对输入数据进行迭代
        for k, v in data.items():
            for file_name, file_content in v.items():
                image1 = Image.open(file_content)
                image1 = np.array(image1, dtype=np.float32)
                image1.resize((1,784))
                images.append(image1)
        # 返回numpy array
        images = np.array(images,dtype=np.float32)
        # 对传入的多个样本做batch处理，shape保持和训练时输入一致
        images.resize((len(data), 784))
        preprocessed_data['images'] = images
        return preprocessed_data

    # inference调用父类处理逻辑

    # 对应的上述训练部分保存模型的输出为{"scores":<array>}
    # 后处理中处理模型输出为HTTPS的接口输出
    def _postprocess(self, data):
        infer_output = {"mnist_result": []}
        # 迭代处理模型输出
        for output_name, results in data.items():
            for result in results:
                infer_output["mnist_result"].append(result.index(max(result)))
        return infer_output
```

15.2 TensorFlow 2.1

训练并保存模型

```
from __future__ import absolute_import, division, print_function, unicode_literals

import tensorflow as tf

mnist = tf.keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(256, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    # 对输出层命名output，在模型推理时通过该命名取结果
    tf.keras.layers.Dense(10, activation='softmax', name="output")
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=10)

tf.keras.models.save_model(model, "./mnist")
```

推理代码

推理代码需要继承自BaseService类，不同类型的模型父类导入语句如请参考**表13-9**。

```
import logging
import threading

import numpy as np
import tensorflow as tf
from PIL import Image

from model_service.tfserving_model_service import TfServingBaseService

logger = logging.getLogger()
logger.setLevel(logging.INFO)


class mnist_service(TfServingBaseService):

    def __init__(self, model_name, model_path):
        self.model_name = model_name
        self.model_path = model_path
        self.model = None
        self.predict = None

        # label文件可以在这里加载,在后处理函数里使用
        # label.txt放在obs和模型包的目录

        # with open(os.path.join(self.model_path, 'label.txt')) as f:
        #     self.label = json.load(f)
        # 非阻塞方式加载saved_model模型，防止阻塞超时
        thread = threading.Thread(target=self.load_model)
        thread.start()

    def load_model(self):
```

```
        # load saved_model 格式的模型
        self.model = tf.saved_model.load(self.model_path)

        signature_defs = self.model.signatures.keys()

        signature = []
        # only one signature allowed
        for signature_def in signature_defs:
            signature.append(signature_def)

        if len(signature) == 1:
            model_signature = signature[0]
        else:
            logging.warning("signatures more than one, use serving_default signature from %s", signature)
            model_signature = tf.saved_model.DEFAULT_SERVING_SIGNATURE_DEF_KEY

        self.predict = self.model.signatures[model_signature]

    def _preprocess(self, data):
        images = []
        for k, v in data.items():
            for file_name, file_content in v.items():
                image1 = Image.open(file_content)
                image1 = np.array(image1, dtype=np.float32)
                image1.resize((28, 28, 1))
                images.append(image1)

        images = tf.convert_to_tensor(images, dtype=tf.dtypes.float32)
        preprocessed_data = images

        return preprocessed_data

    def _inference(self, data):

        return self.predict(data)

    def _postprocess(self, data):

        return {
            "result": int(data["output"].numpy()[0].argmax())
        }
```

15.3 PyTorch

训练模型

```
from __future__ import print_function
import argparse
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from torchvision import datasets, transforms

# 定义网络结构
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        # 输入第二维需要为784
        self.hidden1 = nn.Linear(784, 5120, bias=False)
        self.output = nn.Linear(5120, 10, bias=False)

    def forward(self, x):
        x = x.view(x.size()[0], -1)
        x = F.relu((self.hidden1(x)))
        x = F.dropout(x, 0.2)
        x = self.output(x)
```

```
        return F.log_softmax(x)

def train(model, device, train_loader, optimizer, epoch):
    model.train()
    for batch_idx, (data, target) in enumerate(train_loader):
        data, target = data.to(device), target.to(device)
        optimizer.zero_grad()
        output = model(data)
        loss = F.cross_entropy(output, target)
        loss.backward()
        optimizer.step()
        if batch_idx % 10 == 0:
            print('Train Epoch: {} [{}/{} ({:.0f}%)]\tLoss: {:.6f}'.format(
                epoch, batch_idx * len(data), len(train_loader.dataset),
                       100. * batch_idx / len(train_loader), loss.item()))

def test( model, device, test_loader):
    model.eval()
    test_loss = 0
    correct = 0
    with torch.no_grad():
        for data, target in test_loader:
            data, target = data.to(device), target.to(device)
            output = model(data)
            test_loss += F.nll_loss(output, target, reduction='sum').item()  # sum up batch loss
            pred = output.argmax(dim=1, keepdim=True)  # get the index of the max log-probability
            correct += pred.eq(target.view_as(pred)).sum().item()

    test_loss /= len(test_loader.dataset)

    print('\nTest set: Average loss: {:.4f}, Accuracy: {}/{} ({:.0f}%)\n'.format(
        test_loss, correct, len(test_loader.dataset),
        100. * correct / len(test_loader.dataset)))

device = torch.device("cpu")

batch_size=64

kwargs={}

train_loader = torch.utils.data.DataLoader(
    datasets.MNIST('.', train=True, download=True,
                   transform=transforms.Compose([
                       transforms.ToTensor()
                   ])),
    batch_size=batch_size, shuffle=True, **kwargs)
test_loader = torch.utils.data.DataLoader(
    datasets.MNIST('.', train=False, transform=transforms.Compose([
        transforms.ToTensor()
    ])),
    batch_size=1000, shuffle=True, **kwargs)

model = Net().to(device)
optimizer = optim.SGD(model.parameters(), lr=0.01, momentum=0.5)
optimizer = optim.Adam(model.parameters())

for epoch in range(1, 2 + 1):
    train(model, device, train_loader, optimizer, epoch)
    test(model, device, test_loader)
```

保存模型

```
# 必须采用state_dict的保存方式，支持异地部署
torch.save(model.state_dict(), "pytorch_mnist/mnist_mlp.pt")
```

推理代码

推理代码需要继承自BaseService类，不同类型的模型父类导入语句如请参考**表13-9**。

```
from PIL import Image
import log
from model_service.pytorch_model_service import PTServingBaseService
import torch.nn.functional as F

import torch.nn as nn
import torch
import json

import numpy as np

logger = log.getLogger(__name__)

import torchvision.transforms as transforms

# 定义模型预处理
infer_transformation = transforms.Compose([
    transforms.Resize((28,28)),
    # 需要处理成pytorch tensor
    transforms.ToTensor()
])


import os


class PTVisionService(PTServingBaseService):

    def __init__(self, model_name, model_path):
        # 调用父类构造方法
        super(PTVisionService, self).__init__(model_name, model_path)
        # 调用自定义函数加载模型
        self.model = Mnist(model_path)
        # 加载标签
        self.label = [0,1,2,3,4,5,6,7,8,9]
        # 亦可通过文件标签文件加载
        # model目录下放置label.json文件，此处读取
        dir_path = os.path.dirname(os.path.realpath(self.model_path))
        with open(os.path.join(dir_path, 'label.json')) as f:
            self.label = json.load(f)


    def _preprocess(self, data):

        preprocessed_data = {}
        for k, v in data.items():
            input_batch = []
            for file_name, file_content in v.items():
                with Image.open(file_content) as image1:
                    # 灰度处理
                    image1 = image1.convert("L")
                    if torch.cuda.is_available():
                        input_batch.append(infer_transformation(image1).cuda())
                    else:
                        input_batch.append(infer_transformation(image1))
            input_batch_var = torch.autograd.Variable(torch.stack(input_batch, dim=0), volatile=True)
            print(input_batch_var.shape)
            preprocessed_data[k] = input_batch_var

        return preprocessed_data

    def _postprocess(self, data):
        results = []
        for k, v in data.items():
            result = torch.argmax(v[0])
            result = {k: self.label[result]}
            results.append(result)
        return results
```

```
    def _inference(self, data):

        result = {}
        for k, v in data.items():
            result[k] = self.model(v)

        return result

class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.hidden1 = nn.Linear(784, 5120, bias=False)
        self.output = nn.Linear(5120, 10, bias=False)

    def forward(self, x):
        x = x.view(x.size()[0], -1)
        x = F.relu((self.hidden1(x)))
        x = F.dropout(x, 0.2)
        x = self.output(x)
        return F.log_softmax(x)



def Mnist(model_path, **kwargs):
    # 生成网络
    model = Net()
    # 加载模型
    if torch.cuda.is_available():
        device = torch.device('cuda')
        model.load_state_dict(torch.load(model_path, map_location="cuda:0"))
    else:
        device = torch.device('cpu')
        model.load_state_dict(torch.load(model_path, map_location=device))
    # CPU或者GPU映射
    model.to(device)
    # 声明为推理模式
    model.eval()

    return model
```

15.4 Caffe

训练并保存模型

“lenet_train_test.prototxt”文件

```
name: "LeNet"
layer {
  name: "mnist"
  type: "Data"
  top: "data"
  top: "label"
  include {
    phase: TRAIN
  }
  transform_param {
    scale: 0.00390625
  }
  data_param {
    source: "examples/mnist/mnist_train_lmdb"
    batch_size: 64
    backend: LMDB
  }
}
layer {
  name: "mnist"
```

```
  type: "Data"
  top: "data"
  top: "label"
  include {
    phase: TEST
  }
  transform_param {
    scale: 0.00390625
  }
  data_param {
    source: "examples/mnist/mnist_test_lmdb"
    batch_size: 100
    backend: LMDB
  }
}
layer {
  name: "conv1"
  type: "Convolution"
  bottom: "data"
  top: "conv1"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  convolution_param {
    num_output: 20
    kernel_size: 5
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "pool1"
  type: "Pooling"
  bottom: "conv1"
  top: "pool1"
  pooling_param {
    pool: MAX
    kernel_size: 2
    stride: 2
  }
}
layer {
  name: "conv2"
  type: "Convolution"
  bottom: "pool1"
  top: "conv2"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  convolution_param {
    num_output: 50
    kernel_size: 5
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
```

```
    }
  }
}
layer {
  name: "pool2"
  type: "Pooling"
  bottom: "conv2"
  top: "pool2"
  pooling_param {
    pool: MAX
    kernel_size: 2
    stride: 2
  }
}
layer {
  name: "ip1"
  type: "InnerProduct"
  bottom: "pool2"
  top: "ip1"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  inner_product_param {
    num_output: 500
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "relu1"
  type: "ReLU"
  bottom: "ip1"
  top: "ip1"
}
layer {
  name: "ip2"
  type: "InnerProduct"
  bottom: "ip1"
  top: "ip2"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  inner_product_param {
    num_output: 10
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "accuracy"
  type: "Accuracy"
  bottom: "ip2"
  bottom: "label"
  top: "accuracy"
  include {
```

```
    phase: TEST
  }
}
layer {
  name: "loss"
  type: "SoftmaxWithLoss"
  bottom: "ip2"
  bottom: "label"
  top: "loss"
}
```

“lenet_solver.prototxt”文件

```
# The train/test net protocol buffer definition
net: "examples/mnist/lenet_train_test.prototxt"
# test_iter specifies how many forward passes the test should carry out.
# In the case of MNIST, we have test batch size 100 and 100 test iterations,
# covering the full 10,000 testing images.
test_iter: 100
# Carry out testing every 500 training iterations.
test_interval: 500
# The base learning rate, momentum and the weight decay of the network.
base_lr: 0.01
momentum: 0.9
weight_decay: 0.0005
# The learning rate policy
lr_policy: "inv"
gamma: 0.0001
power: 0.75
# Display every 100 iterations
display: 100
# The maximum number of iterations
max_iter: 1000
# snapshot intermediate results
snapshot: 5000
snapshot_prefix: "examples/mnist/lenet"
# solver mode: CPU or GPU
solver_mode: CPU
```

执行训练

```
./build/tools/caffe train --solver=examples/mnist/lenet_solver.prototxt
```

训练后生成“caffemodel”文件，然后将“lenet_train_test.prototxt”文件改写成部署用的lenet_deploy.prototxt，修改输入层和输出层。

```
name: "LeNet"
layer {
  name: "data"
  type: "Input"
  top: "data"
  input_param { shape: { dim: 1 dim: 1  dim: 28 dim: 28 } }
}
layer {
  name: "conv1"
  type: "Convolution"
  bottom: "data"
  top: "conv1"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  convolution_param {
    num_output: 20
    kernel_size: 5
    stride: 1
    weight_filler {
      type: "xavier"
```

```
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "pool1"
  type: "Pooling"
  bottom: "conv1"
  top: "pool1"
  pooling_param {
    pool: MAX
    kernel_size: 2
    stride: 2
  }
}
layer {
  name: "conv2"
  type: "Convolution"
  bottom: "pool1"
  top: "conv2"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  convolution_param {
    num_output: 50
    kernel_size: 5
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "pool2"
  type: "Pooling"
  bottom: "conv2"
  top: "pool2"
  pooling_param {
    pool: MAX
    kernel_size: 2
    stride: 2
  }
}
layer {
  name: "ip1"
  type: "InnerProduct"
  bottom: "pool2"
  top: "ip1"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  inner_product_param {
    num_output: 500
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
```

```
    }
  }
}
layer {
  name: "relu1"
  type: "ReLU"
  bottom: "ip1"
  top: "ip1"
}
layer {
  name: "ip2"
  type: "InnerProduct"
  bottom: "ip1"
  top: "ip2"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  inner_product_param {
    num_output: 10
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "prob"
  type: "Softmax"
  bottom: "ip2"
  top: "prob"
}
```

推理代码

推理代码需要继承自BaseService类，不同类型的模型父类导入语句如请参考**表13-9**。

```
from model_service.caffe_model_service import CaffeBaseService

import numpy as np

import os, json

import caffe

from PIL import Image


class LenetService(CaffeBaseService):

    def __init__(self, model_name, model_path):
        # 调用父类推理方法
        super(LenetService, self).__init__(model_name, model_path)

        # 设置预处理
        transformer = caffe.io.Transformer({'data': self.net.blobs['data'].data.shape})
        # 转换为NCHW格式
        transformer.set_transpose('data', (2, 0, 1))
        # 归一化处理
        transformer.set_raw_scale('data', 255.0)

        # batch size设为1，只支持一张图片的推理
        self.net.blobs['data'].reshape(1, 1, 28, 28)
        self.transformer = transformer
```

```
    # 设置类别标签
    self.label = [0,1,2,3,4,5,6,7,8,9]


def _preprocess(self, data):

    for k, v in data.items():
        for file_name, file_content in v.items():
            im = caffe.io.load_image(file_content, color=False)
            # 图片预处理
            self.net.blobs['data'].data[...] = self.transformer.preprocess('data', im)

            return

def _postprocess(self, data):

    data = data['prob'][0, :]
    predicted = np.argmax(data)
    predicted = {"predicted" : str(predicted) }

    return predicted
```

15.5 XGBoost

训练并保存模型

```
import pandas as pd
import xgboost as xgb
from sklearn.model_selection import train_test_split

# Prepare training data and setting parameters
iris = pd.read_csv('/home/ma-user/work/iris.csv')
X = iris.drop(['variety'],axis=1)
y = iris[['variety']]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1234565)
params = {
    'booster': 'gbtree',
    'objective': 'multi:softmax',
    'num_class': 3,
    'gamma': 0.1,
    'max_depth': 6,
    'lambda': 2,
    'subsample': 0.7,
    'colsample_bytree': 0.7,
    'min_child_weight': 3,
    'silent': 1,
    'eta': 0.1,
    'seed': 1000,
    'nthread': 4,
}
plst = params.items()
dtrain = xgb.DMatrix(X_train, y_train)
num_rounds = 500
model = xgb.train(plst, dtrain, num_rounds)
model.save_model('/tmp/xgboost.m')
```

训练前请先下载iris.csv数据集，解压后上传至Notebook本地路径/home/ma-user/work/。iris.csv数据集下载地址：**https://gist.github.com/netj/8836201**。Notebook上传文件操作请参见**上传本地文件至Notebook中**。

保存完模型后，需要上传到OBS目录才能发布。发布时需要带上config.json配置和推理代码customize_service.py。config.json编写请参考**模型配置文件编写说明**，推理代码请参考**推理代码**。

推理代码

推理代码需要继承自BaseService类，不同类型的模型父类导入语句如请参考**表13-9**。

```
# coding:utf-8
import collections
import json
import xgboost as xgb
from model_service.python_model_service import XgSklServingBaseService
class user_Service(XgSklServingBaseService):

    # request data preprocess
    def _preprocess(self, data):
        list_data = []
        json_data = json.loads(data, object_pairs_hook=collections.OrderedDict)
        for element in json_data["data"]["req_data"]:
            array = []
            for each in element:
                array.append(element[each])
            list_data.append(array)
        return list_data

    #   predict
    def _inference(self, data):
        xg_model = xgb.Booster(model_file=self.model_path)
        pre_data = xgb.DMatrix(data)
        pre_result = xg_model.predict(pre_data)
        pre_result = pre_result.tolist()
        return pre_result

    # predict result process
    def _postprocess(self,data):
        resp_data = []
        for element in data:
            resp_data.append({"predictresult": element})
        return resp_data
```

15.6 Pyspark

训练并保存模型

```
from pyspark.ml import Pipeline, PipelineModel
from pyspark.ml.linalg import Vectors
from pyspark.ml.classification import LogisticRegression

# 创建训练数据，此处通过tuples创建
# Prepare training data from a list of (label, features) tuples.
training = spark.createDataFrame([
    (1.0, Vectors.dense([0.0, 1.1, 0.1])),
    (0.0, Vectors.dense([2.0, 1.0, -1.0])),
    (0.0, Vectors.dense([2.0, 1.3, 1.0])),
    (1.0, Vectors.dense([0.0, 1.2, -0.5]))], ["label", "features"])

# 创建训练实例，此处使用逻辑回归算法进行训练
# Create a LogisticRegression instance. This instance is an Estimator.
lr = LogisticRegression(maxIter=10, regParam=0.01)

# 训练逻辑回归模型
# Learn a LogisticRegression model. This uses the parameters stored in lr.
model = lr.fit(training)

# 保存模型到本地目录
# Save model to local path.
model.save("/tmp/spark_model")
```

保存完模型后，需要上传到OBS目录才能发布。发布时需要带上config.json配置和推理代码customize_service.py。config.json编写请参考**模型配置文件编写说明**，推理代码请参考**推理代码**。

推理代码

推理代码需要继承自BaseService类，不同类型的模型父类导入语句如请参考**表13-9**。

```
# coding:utf-8
import collections
import json
import traceback

import model_service.log as log
from model_service.spark_model_service import SparkServingBaseService
from pyspark.ml.classification import LogisticRegression

logger = log.getLogger(__name__)


class user_Service(SparkServingBaseService):
    # 数据预处理
    def _preprocess(self, data):
        logger.info("Begin to handle data from user data...")
        # 读取数据
        req_json = json.loads(data, object_pairs_hook=collections.OrderedDict)
        try:
            # 将数据转换成spark dataframe格式
            predict_spdf = self.spark.createDataFrame(pd.DataFrame(req_json["data"]["req_data"]))
        except Exception as e:
            logger.error("check your request data does meet the requirements ?")
            logger.error(traceback.format_exc())
            raise Exception("check your request data does meet the requirements ?")
        return predict_spdf

    # 模型推理
    def _inference(self, data):
        try:
            # 加载模型文件
            predict_model = LogisticRegression.load(self.model_path)
            # 对数据进行推理
            prediction_result = predict_model.transform(data)
    except Exception as e:
            logger.error(traceback.format_exc())
            raise Exception("Unable to load model and do dataframe transformation.")
        return prediction_result

    # 数据后处理
    def _postprocess(self, pre_data):
        logger.info("Get new data to respond...")
        predict_str = pre_data.toPandas().to_json(orient='records')
        predict_result = json.loads(predict_str)
        return predict_result
```

15.7 Scikit Learn

训练并保存模型

```
import json
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.externals import joblib
iris = pd.read_csv('/home/ma-user/work/iris.csv')
```

```
X = iris.drop(['variety'],axis=1)
y = iris[['variety']]
# Create a LogisticRegression instance and train model
logisticRegression = LogisticRegression(C=1000.0, random_state=0)
logisticRegression.fit(X,y)
# Save model to local path
joblib.dump(logisticRegression, '/tmp/sklearn.m')
```

训练前请先下载iris.csv数据集，解压后上传至Notebook本地路径/home/ma-user/work/。iris.csv数据集下载地址：**https://gist.github.com/netj/8836201**。Notebook上传文件操作请参见**上传本地文件至Notebook中**。

保存完模型后，需要上传到OBS目录才能发布。发布时需要带上“config.json”配置以及“customize_service.py”，定义方式参考**模型包规范介绍**。

推理代码

推理代码需要继承自BaseService类，不同类型的模型父类导入语句如请参考**表13-9**。

```
# coding:utf-8
import collections
import json
from sklearn.externals import joblib
from model_service.python_model_service import XgSklServingBaseService

class user_Service(XgSklServingBaseService):

    # request data preprocess
    def _preprocess(self, data):
        list_data = []
        json_data = json.loads(data, object_pairs_hook=collections.OrderedDict)
        for element in json_data["data"]["req_data"]:
            array = []
            for each in element:
                array.append(element[each])
                list_data.append(array)
        return list_data

    # predict
    def _inference(self, data):
        sk_model = joblib.load(self.model_path)
        pre_result = sk_model.predict(data)
        pre_result = pre_result.tolist()
        return pre_result

    # predict result process
    def _postprocess(self,data):
        resp_data = []
        for element in data:
            resp_data.append({"predictresult": element})
        return resp_data
```

16 权限管理

16.1 创建并授权使用 ModelArts

如果您需要对您所拥有的ModelArts进行精细的权限管理，您可以使用**统一身份认证服务**（Identity and Access Management，简称IAM），通过IAM，您可以：

- 根据企业的业务组织，在您的华为云帐号中，给企业中不同职能部门的员工创建IAM用户，让员工拥有唯一安全凭证，并使用ModelArts资源。
- 根据企业用户的职能，设置不同的访问权限，以达到用户之间的权限隔离。
- 将ModelArts资源委托给更专业、高效的其他华为云帐号或者云服务，这些帐号或者云服务可以根据权限进行代运维。

如果华为云帐号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用ModelArts服务的其它功能。

本章节为您介绍对用户授权的方法，操作流程如**图16-1**所示。

前提条件

- 给用户组授权之前，请您了解用户组可以添加的ModelArts权限，并结合实际需求进行选择，ModelArts支持的系统权限，请参见：**ModelArts系统权限**。
- 由于ModelArts的使用权限依赖OBS服务的授权，您需要为用户授予OBS的系统权限，详细说明请参见**OBS权限管理**。
- 若您需要对除ModelArts和OBS之外的其它服务授权，IAM支持服务的所有策略请参见**权限策略**。

示例流程

图 16-1 给用户授权 ModelArts 权限流程

1. **创建用户组并授权**

   在IAM控制台创建用户组，并授予“ModelArts CommonOperations”权限。

   由于ModelArts依赖OBS权限，请为用户组授予“作用范围”为“全局级服务”的“Tenant Administrator”策略。

2. **创建用户并加入用户组**

   在IAM控制台创建用户，并将其加入**1**中创建的用户组。

3. **用户登录**并验证权限

   新创建的用户登录控制台，切换至授权区域，验证权限：

   - 在“服务列表”中选择ModelArts，进入ModelArts主界面，单击“专属资源池>创建”，如果无法进行创建（假设当前权限仅包含ModelArts CommonOperations），表示“ModelArts CommonOperations”已生效。
   - 在“服务列表”中选择除ModelArts外（假设当前策略仅包含ModelArts CommonOperations）的任一服务，若提示权限不足，表示“ModelArts CommonOperations”已生效。
   - 在“服务列表”中选择ModelArts，进入ModelArts主界面，单击“数据管理>数据集>创建数据集”，如果可以成功访问对应的OBS路径，表示全局级服务的“Tenant Administrator”已生效。

16.2 创建 ModelArts 自定义策略

如果系统预置的ModelArts权限，不满足您的授权要求，可以创建自定义策略。自定义策略中可以添加的授权项（Action）请参考**《ModelArts API参考》>权限策略和授权项**。

目前华为云支持以下两种方式创建自定义策略：

- 可视化视图创建自定义策略：无需了解策略语法，按可视化视图导航栏选择云服务、操作、资源、条件等策略内容，可自动生成策略。
- JSON视图创建自定义策略：可以在选择策略模板后，根据具体需求编辑策略内容；也可以直接在编辑框内编写JSON格式的策略内容。

具体创建步骤请参见：**创建自定义策略**。本章为您介绍常用的**ModelArts依赖的OBS权限自定义策略样例**和**ModelArts自定义策略样例**。

注意事项

- 由于ModelArts的使用权限依赖OBS服务的授权，您需要为用户授予OBS的系统权限。
- 如果一个自定义策略中包含多个服务的授权语句，这些服务必须是同一属性，即都是“全局级服务”或者“项目级服务”。
- 如果需要同时设置全局级服务和项目级服务的自定义策略，请创建两条自定义策略，“作用范围”别为“全局级服务”以及“项目级服务”，然后给用户同时授予这两条自定义策略。
- 针对当前帐号下创建的IAM用户，默认具备ModelArts的所有操作权限，当需要对IAM用户进行权限控制，如拒绝某用户使用某功能的权限时，可通过创建ModelArts自定义策略实现。

ModelArts 依赖的 OBS 权限自定义策略样例

由于ModelArts为“项目级服务”，OBS为“全局级服务”，因此需要分别创建自定义策略然后授予用户。如下示例为ModelArts依赖OBS服务的最小化权限项，包含OBS桶和OBS对象的权限。授予示例中的权限您可以通过ModelArts正常访问OBS不受限制。

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "obs:object:PutObjectAcl",
                "obs:bucket:PutBucketAcl",
                "obs:bucket:PutBucketPolicy",
                "obs:bucket:HeadBucket",
                "obs:bucket:ListAllMyBuckets",
                "obs:bucket:ListBucket",
                "obs:object:DeleteObjectVersion",
                "obs:object:AbortMultipartUpload",
                "obs:object:DeleteObject",
                "obs:object:PutObject",
                "obs:bucket:CreateBucket",
                "obs:object:GetObject",
                "obs:bucket:GetBucketLocation",
                "obs:object:GetObjectVersionAcl",
                "obs:bucket:GetBucketAcl",
                "obs:object:ListMultipartUploadParts",
                "obs:bucket:ListBucketVersions",
                "obs:object:GetObjectVersion",
                "obs:object:GetObjectAcl",
                "obs:bucket:GetBucketPolicy"
            ]
        }
    ]
}
```

ModelArts 自定义策略样例

针对当前帐号下创建的IAM用户，默认具备ModelArts的所有操作权限，当需要对IAM用户进行权限控制，如拒绝某用户使用某功能的权限时，可参考如下样例进行配置。

- 示例：拒绝用户删除自动学习项目

  拒绝策略需要同时配合其他策略使用，否则没有实际作用。用户被授予的策略中，一个授权项的作用如果同时存在Allow和Deny，则遵循Deny优先。

  如果您给用户授予ModelArts FullAccess的系统策略，但不希望用户拥有ModelArts FullAccess中定义的删除自动学习项目权限，您可以创建一条拒绝删除自动学习项目的自定义策略，然后同时将ModelArts FullAccess和拒绝策略授予用户，根据Deny优先原则，则用户可以对ModelArts执行除了删除自动学习项目外的所有操作。拒绝策略示例如下：

```
{
    "Version": "1.1",
    "Statement": [
        {
       "Effect": "Deny",
              "Action": [
                    "modelarts:exemlProject:delete"
              ]
        }
    ]
}
```

- 示例：拒绝用户使用开发环境功能

  此用户的策略配置示例如下所示：

```
{
   "Version": "1.1",
   "Statement": [

      {
          "Effect": "Deny",
          "Action": [
              "modelarts:notebook:list",
              "modelarts:notebook:create" ,
              "modelarts:notebook:get" ,
              "modelarts:notebook:update" ,
              "modelarts:notebook:delete" ,
              "modelarts:notebook:action" ,
              "modelarts:notebook:access"
          ]
      }
   ]
}
```

17 监控

17.1 ModelArts 支持的监控指标

功能说明

为使用户更好地掌握自己的ModelArts在线服务和对应模型负载的运行状态，云服务平台提供了云监控。您可以使用该服务监控您的ModelArts在线服务和对应模型负载，执行自动实时监控、告警和通知操作，帮助您更好地了解服务和模型的各项性能指标。

命名空间

SYS.ModelArts

监控指标

表 17-1 ModelArts 支持的监控指标

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期
cpu_usage	CPU使用率	该指标用于统计ModelArts用户服务的CPU使用率。 单位：百分比。	≥ 0%	ModelArts模型负载	1分钟
mem_usage	内存使用率	该指标用于统计ModelArts用户服务的内存使用率。 单位：百分比。	≥ 0%	ModelArts模型负载	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期
gpu_util	GPU使用率	该指标用于统计ModelArts用户服务的GPU使用情况。 单位：百分比。	≥ 0%	ModelArts模型负载	1分钟
gpu_mem_usage	GPU显存使用率	该指标用于统计ModelArts用户服务的GPU显存使用情况。 单位：百分比。	≥ 0%	ModelArts模型负载	1分钟
successfully_called_times	调用成功次数	统计ModelArts用户调用服务的成功次数。 单位：次/分钟。	≥Count/min	ModelArts模型负载 ModelArts在线服务	1分钟
failed_called_times	调用失败次数	统计ModelArts用户调用服务的失败次数。 单位：次/分钟。	≥Count/min	ModelArts模型负载 ModelArts在线服务	1分钟
total_called_times	调用次数	统计ModelArts用户调用服务的次数。 单位：次/分钟。	≥Count/min	ModelArts模型负载 ModelArts在线服务	1分钟

对于有多个测量维度的测量对象，使用接口查询监控指标时，所有测量维度均为必选。

- 查询单个监控指标时，多维度dim使用样例：dim.0=service_id,530cd6b0-86d7-4818-837f-935f6a27414d&dim.1="model_id,3773b058-5b4f-4366-9035-9bbd9964714a。
- 批量查询监控指标时，多维度dim使用样例：
  "dimensions": [
  {
  "name": "service_id",
  "value": "530cd6b0-86d7-4818-837f-935f6a27414d"
  }
  {
  "name": "model_id",
  "value": "3773b058-5b4f-4366-9035-9bbd9964714a"
  }
  ],

维度

表 17-2 维度说明

Key	Value
service_id	在线服务ID。
model_id	模型负载ID。

17.2 设置告警规则

操作场景

通过设置ModelArts在线服务和模型负载告警规则，用户可自定义监控目标与通知策略，及时了解ModelArts在线服务和模型负载状况，从而起到预警作用。

设置ModelArts服务和模型的告警规则包括设置告警规则名称、监控对象、监控指标、告警阈值、监控周期和是否发送通知等参数。本节介绍了设置ModelArts服务和模型告警规则的具体方法。

说明

只有“运行中”的在线服务，支持对接CES监控。

前提条件

已创建ModelArts在线服务。

操作步骤

1. 登录管理控制台。
2. 在“服务列表”中选择“管理与部署 > 云监控”。
3. 在“云监控 > 自定义监控”页面，根据界面提示创建ModelArts监控服务。
4. 在左侧导航栏，选择“云服务监控 > ModelArts”。
5. 选择需要添加告警规则的在线服务名称，单击操作列的“创建告警规则”。
6. 在“创建告警规则”界面，根据界面提示设置ModelArts在线服务和模型负载的告警规则。

图 17-1 创建告警规则

7. 设置完成后，单击“立即创建”。当符合规则的告警产生时，系统会自动进行通知。

17.3 查看监控指标

操作场景

云服务平台提供的云监控，可以对ModelArts在线服务和模型负载运行状态进行日常监控。您可以通过管理控制台，直观地查看ModelArts在线服务和模型负载的各项监控指标。由于监控数据的获取与传输会花费一定时间，因此，云监控显示的是当前时间5～10分钟前的状态。如果您的在线服务刚刚创建完成，请等待5～10分钟后查看监控数据。

前提条件

- ModelArts在线服务正常运行。
- 已在云监控页面设置告警规则，具体操作请参见**设置告警规则**。
- 在线服务已正常运行一段时间（约10分钟）。
- 对于新创建的在线服务，需要等待一段时间，才能查看上报的监控数据和监控视图。
- 故障、删除状态的在线服务，无法在云监控中查看其监控指标。当在线服务再次启动或恢复后，即可正常查看。

对接云监控之前，用户无法查看到未对接资源的监控数据。具体操作，请参见**设置告警规则**。

操作步骤

1. 登录管理控制台。
2. 在“服务列表”中选择“管理与部署 > 云监控”。
3. 在左侧导航栏，选择“云服务监控 > ModelArts”。

4. 查看监控图表。
   - 查看在线服务监控图表：单击目标在线服务“操作”列的“查看监控图表”。
   - 查看模型负载监控图标：单击目标在线服务左侧的，在下拉列表中选择模型负载“操作”列的“查看监控图表”。
5. 在监控区域，您可以通过选择时长，查看对应时间的监控数据。

   当前支持查看近1小时、近3小时和近12小时的监控数据，查看更长时间范围监控曲线，请在监控视图中单击进入大图模式查看。

18 审计日志

18.1 支持云审计的关键操作

公有云平台提供了云审计服务。通过云审计服务，您可以记录与ModelArts相关的操作事件，便于日后的查询、审计和回溯。

前提条件

已开通云审计服务。

数据管理支持审计的关键操作列表

表 18-1 数据管理支持审计的关键操作列表

操作名称	资源类型	事件名称
创建数据集	dataset	createDataset
删除数据集	dataset	deleteDataset
更新数据集	dataset	updateDataset
发布数据集版本	dataset	publishDatasetVersion
删除数据集版本	dataset	deleteDatasetVersion
同步数据源	dataset	syncDataSource
导出数据集	dataset	exportDataFromDataset
创建自动标注任务	dataset	createAutoLabelingTask
创建自动分组任务	dataset	createAutoGroupingTask
创建自动部署任务	dataset	createAutoDeployTask
导入样本到数据集	dataset	importSamplesToDataset
创建数据集标签	dataset	createLabel

操作名称	资源类型	事件名称
更新数据集标签	dataset	updateLabel
删除数据集标签	dataset	deleteLabel
删除数据集标签和对应的样本	dataset	deleteLabelWithSamples
添加样本	dataset	uploadSamples
删除样本	dataset	deleteSamples
停止自动标注任务	dataset	stopTask
创建团队标注任务	dataset	createWorkforceTask
删除团队标注任务	dataset	deleteWorkforceTask
启动团队标注验收的任务	dataset	startWorkforceSamplingTask
通过/驳回/取消验收任务	dataset	updateWorkforceSamplingTask
提交验收任务的样本评审意见	dataset	acceptSamples
给样本添加标签	dataset	updateSamples
发送邮件给团队标注任务的成员	dataset	sendEmails
接口人启动团队标注任务	dataset	startWorkforceTask
更新团队标注任务	dataset	updateWorkforceTask
给团队标注样本添加标签	dataset	updateWorkforceTaskSamples
团队标注审核	dataset	reviewSamples
创建标注成员	workforce	createWorker
更新标注成员	workforce	updateWorker
删除标注成员	workforce	deleteWorker
批量删除标注成员	workforce	batchDeleteWorker
创建标注团队	workforce	createWorkforce
更新标注团队	workforce	updateWorkforce
删除标注团队	workforce	deleteWorkforce
自动创建IAM委托	IAM	createAgency
标注成员登录labelConsole标注平台	labelConsoleWorker	workerLoginLabelConsole

操作名称	资源类型	事件名称
标注成员登出labelConsole标注平台	labelConsoleWorker	workerLogOutLabelConsole
标注成员修改labelConsole平台密码	labelConsoleWorker	workerChangePassword
标注成员忘记labelConsole平台密码	labelConsoleWorker	workerForgetPassword
标注成员通过url重置labelConsole标注密码	labelConsoleWorker	workerResetPassword

开发环境支持审计的关键操作列表

表 18-2 开发环境支持审计的关键操作列表

操作名称	资源类型	事件名称
创建Notebook	Notebook	create_instance
删除Notebook	Notebook	delete_instance
启动Notebook	Notebook	change_instance_status
停止Notebook	Notebook	change_instance_status

训练作业支持审计的关键操作列表

表 18-3 训练作业支持审计的关键操作列表

操作名称	资源类型	事件名称
创建训练作业	ModelArtsTrainJob	createModelArtsTrainJob
创建训练作业版本	ModelArtsTrainJob	createModelArtsTrainVersion
停止训练作业	ModelArtsTrainJob	stopModelArtsTrainVersion
更新训练作业描述	ModelArtsTrainJob	updateModelArtsTrainDesc
删除训练作业版本	ModelArtsTrainJob	deleteModelArtsTrainVersion
删除训练作业	ModelArtsTrainJob	deleteModelArtsTrainJob
创建训练作业参数	ModelArtsTrainConfig	createModelArtsTrainConfig

操作名称	资源类型	事件名称
更新训练作业参数	ModelArtsTrainConfig	updateModelArtsTrainConfig
删除训练作业参数	ModelArtsTrainConfig	deleteModelArtsTrainConfig
创建可视化作业	ModelArtsTensorboardJob	createModelArtsTensorboardJob
删除可视化作业	ModelArtsTensorboardJob	deleteModelArtsTensorboardJob
更新可视化作业描述	ModelArtsTensorboardJob	updateModelArtsTensorboardDesc
停止可视化作业	ModelArtsTensorboardJob	stopModelArtsTensorboardJob
重启可视化作业	ModelArtsTensorboardJob	restartModelArtsgTensorboardJob

AI 应用管理支持审计的关键操作列表

表 18-4 AI 应用管理支持审计的关键操作列表

操作名称	资源类型	事件名称
创建AI应用	model	addModel
更新AI应用	model	updateModel
删除AI应用	model	deleteModel
添加转换任务	convert	addConvert
更新转换任务	convert	updateConvert
删除转换任务	convert	deleteConvert

服务管理支持审计的关键操作列表

表 18-5 服务管理支持审计的关键操作列表

操作名称	资源类型	事件名称
部署服务	service	addService
删除服务	service	deleteService
更新服务	service	updateService

操作名称	资源类型	事件名称
启停服务	service	startOrStopService
启停边缘服务节点	service	startOrStopNodesService
添加用户访问秘钥	service	addAkSk
删除用户访问秘钥	service	deleteAkSk
创建专属资源池	cluster	createCluster
删除专属资源池	cluster	deleteCluster
添加专属资源池节点	cluster	addClusterNode
删除专属资源池节点	cluster	deleteClusterNode
获取专属资源池创建结果	cluster	createClusterResult

18.2 查看审计日志

在您开启了云审计服务后，系统会记录ModelArts的相关操作，且控制台保存最近7天的操作记录。本节介绍如何在云审计服务管理控制台查看最近7天的操作记录。

操作步骤

1. 登录云审计服务管理控制台。
2. 在管理控制台左上角单击 图标，选择区域。
3. 在左侧导航栏中，单击“事件列表”，进入“事件列表”页面。
4. 事件列表支持通过筛选来查询对应的操作事件。当前事件列表支持四个维度的组合查询，详细信息如下：
   - 事件来源、资源类型和筛选类型。

     在下拉框中选择查询条件。

     其中筛选类型选择事件名称时，还需选择某个具体的事件名称。

     选择资源ID时，还需输入某个具体的资源ID。

     选择资源名称时，还需选择或手动输入某个具体的资源名称。
   - 操作用户：在下拉框中选择某一具体的操作用户，此操作用户指用户级别，而非租户级别。
   - 事件级别：可选项为“所有事件级别”、“normal”、“warning”、“incident”，只可选择其中一项。
   - 时间范围：可选择查询最近七天内任意时间段的操作事件。
5. 在需要查看的事件左侧，单击 展开该事件的详细信息。
6. 单击需要查看的事件“操作”列的“查看事件”，可以在弹窗中查看该操作事件结构的详细信息。

   更多关于云审计服务事件结构的信息，请参见《云审计服务用户指南》。

19 建议反馈

ModelArts建议反馈功能旨在收集、跟踪、解决用户问题。当前ModelArts提供一种实时的截图发帖反馈方式，帮助用户高效、快捷地进行建议反馈。ModelArts建议反馈功能仅在华为云中国站中文上使用。

在 Chrome 安装插件

ModelArts建议反馈功能需要插件支持。当前建议反馈插件仅支持Chrome浏览器，建议您在Chrome浏览器环境下完成插件的安装。

1. 登录ModelArts控制台，进入任意业务页面，单击页面右上角的“建议反馈>建议反馈”。

   图 19-1 建议反馈

2. 当您未安装ModelArts建议反馈插件时，弹窗存在提醒，如图19-2所示。单击图19-2中的链接可跳转至资料页面，资料页面提供插件下载地址。单击下载地址，即可完成ModelArts建议反馈插件的下载。

   图 19-2 建议反馈提醒

3. 解压ModelArts建议反馈插件压缩文件modelarts-screenshot-feedback.zip至任意路径，得到modelarts-screenshot-feedback文件夹。
4. 打开Chrome浏览器，单击Chrome浏览器的设置-扩展程序选项，进入扩展程序页面。
5. 在扩展页面中，打开右上角的开发者模式开关，如图19-3所示。

图 19-3 开发者模式开关

6. 拖动已解压的ModelArts建议反馈插件文件夹modelarts-screenshot-feedback到Chrome浏览器扩展程序页面，即可完成插件的安装。

截图发帖反馈

ModelArts建议反馈功能提供实时截图能力并支持截图编辑操作。

1. 登录ModelArts平台，进入任意业务页面，单击右上角“建议反馈>建议反馈”。
2. 在弹窗页面选择“是”进入页面截图编辑状态。您也可以选择“否”直接进行建议发帖。

图 19-4 截图反馈确认

3. 进入当前页面截图可编辑状态。用户可以通过左侧的图形编辑功能进行截图的编辑。

图 19-5 建议反馈可编辑状态页面

插件功能的图标从左到右分别为：

- 高亮
- 直线/曲线
- 喷枪

- 文字
- 矩形框/矩形方块/圆框
- 箭头
- 裁剪
- 撤销
- 恢复
- 发帖

4. 完成截图编辑后，用户可以通过单击右侧发帖反馈键跳转至图 发帖界面进行帖子的编辑和发布。

进入我的反馈

1. 登录ModelArts控制台，进入任意业务页面，单击页面右上角“建议反馈>我的反馈”。

图 19-6 我的反馈

2. 页面跳转至“个人中心>我的论坛”。您可以在此页面跟踪您的建议反馈进展。您可以根据实际情况查看您的“最新发布”和“最新回复”。

图 19-7 我的反馈详细页

A 修订记录

发布日期	修改说明
2022-06-16	修改创建AI应用的自定义镜像大小要求，**创建AI应用的自定义镜像规范**。
2022-05-19	推理服务支持无损滚动升级：**升级服务**
2022-01-26	下线部署服务章节的“难例筛选”。
2022-01-04	下线数据管理章节中的“一键模型”上线功能。
2021-12-20	旧版训练中下线预置算法。 下线模型评估章节，转为邀测特性。
2021-12-15	体验优化：“模型管理”更名为“AI应用”。
2021-11-25	资源池中优化部分描述内容。
2021-10-21	新增11.7-旧版训练与新版训练差异
2021-9-27	更新开发环境New中的VSCode插件下载地址。 新增开发环境New中的变更Notebook实例运行规格。
2021-09-18	新增管理数据（New），数据集与数据标注任务解耦。 新增配置SSH工具远程连接开发环境Notebook。 新增配置VSCode插件远程连接开发环境Notebook 新增开发环境New中的动态挂载OBS并行文件系统。
2021-08-30	开发环境（New）中优化部分描述内容。
2021-07-17	新增发布开发环境（New）。
2021-06-26	新增发布训练管理（New）。
2021-04-23	新增发布CodeLab。
2021-03-26	● 在线服务新增**个性化配置**。 ● “图像分割”类型数据集支持从OBS导入数据**图像分割**

发布日期	修改说明
2021-01-14	• 使用自定义镜像导入模型新增示例。 **使用自定义镜像创建的AI应用部署服务** • 智能标注功能优化。 **智能标注** • AI市场更名。 • 模型包规范新增机器学习推理代码样例和配置文件说明。 **模型配置文件编写说明** **模型推理代码编写说明** • 自定义脚本代码示例优化Pytorch代码示例。 **PyTorch**
2020-12-10	• 团队标注功能优化。 **管理团队标注任务** • 创建数据集时，随软件界面更新。 **创建数据集（旧版）** • 创建或者扩容推理专属资源池支持指定可用区。 **创建专属资源池**
2020-11-10	• 创建Notebook时，优化工作环境的展示。 **创建Notebook实例** • 增加模型评估诊断后的优化建议。 • 新增数据处理功能。 **数据处理**
2020-10-15	• 新增建议反馈功能。 **建议反馈** • 在线服务新增AK/SK认证介绍。 **访问在线服务（AK/SK认证)**
2020-09-21	• 在线服务新增支APP认证方式的功能。 **访问在线服务（APP认证）**
2020-09-11	• 新增支持模型评估诊断的功能。 • 查看训练作业时，支持查看评估结果。 **评估结果** **添加评估结果**
2020-08-06	• Notebook功能描述优化。重新优化大纲结构，同时增加JupyterLab常用操作指导。 **Jupyter Notebook简介** **JupyterLab简介及常用操作** • “表格”类型数据集，支持DWS数据源输入。 **表格** • 部署在线服务API增加集成API的指导。 **集成在线服务** • 旧版“AI市场”及“数据管理”下线。

发布日期	修改说明
2020-07-06	数据集管理新增支持视频数据。 ● 创建数据集（旧版） ● 视频标注 自动化搜索作业优化增强，同时增加1个示例。 开发环境增加支持GitHub代码库功能。 ● 创建带有Git存储库的Notebook实例 ● 使用Git插件’ 开发环境、模型导入支持PyTorch 1.4.0引擎。 ● 创建Notebook实例 ● 从OBS中选择元模型 增加TensorFlow 2.X的自定义脚本代码示例。 ● TensorFlow 2.1 模型转换任务创建页面优化。 模型转换操作
2020-06-08	数据集管理新增支持表格型数据，支持导入和发布csv。 ● 创建数据集（旧版） ● 从OBS目录导入的规范说明 数据特征分析，增加特征分析任务启动和管理功能。 ● 数据特征 开发环境增强，引入支持JupyterLab、引擎新增TF2.1，对ModelArts多种基础能力的集成以及若干体验改进。 ● 创建Notebook实例 ● 使用ModelArts示例 ● 与OBS同步文件 ● JupyterLab简介及常用操作 ● 上传大文件至Notebook中 ● 下载Notebook内的大文件到本地 自动化搜索作业优化：新增fix_norm超参搜索、adv_aug数据增强、betanas等多种NAS方法，达到ImageNet上mobile setting下最佳精度，支持最低5行代码即可享用多元搜索能力。 预置框架支持TenorFlow 2.1。创建Notebook实例或使用常用框架创建训练作业时，可选择TenorFlow 2.1。 ● 使用常用框架训练模型 ● Notebook简介 模型管理新增支持python3.7的runtime环境。 ● 从OBS中选择元模型

发布日期	修改说明
2020-04-07	• 支持Ascend 910训练，AI市场提供了可用于Ascend 910训练的云端算法。 • 部署上线，开放支持Ascend 310推理能力。 部署为在线服务 部署为批量服务 部署为边缘服务
2020-03-25	• 提供限时免费的资源规格，可在自动学习、训练作业以及Notebook中使用，增加相应说明。 – 免费体验自动学习 – 免费体验AI全流程开发 – 5.2-免费体验Notebook • 在AI市场，ModelArts官方提供了云端算法，无需编码，快速启动模型训练。
2020-03-20	• 优化训练作业相关说明，按不同算法来源进行描述。 – 使用已有算法训练模型 – 使用常用框架训练模型 – 使用自定义镜像训练模型 • 文本三元组、文本分类、命名实体类型的数据集支持团队标注，刷新数据标注功能说明。 数据管理简介 • ModelArts UI风格升级，刷新本文档所有相关截图。
2020-02-21	• 新增两个的转换模板。 转换模板
2020-01-07	• 旧版数据管理功能隐藏。
2019-12-03	• 对模型导入功能，按场景拆分为4个子章节。 从训练中选择元模型 从模板中选择元模型 从容器镜像中选择元模型 从OBS中选择元模型 • 针对支持团队标注的数据集，支持管理团队标注任务。 管理团队标注任务 • 增加数据集类型“文本三元组” 文本三元组 • 数据管理的整体章节进行编辑优化。 数据管理简介 ~ 删除数据集 • 增加自动化搜索作业管理的功能。

发布日期	修改说明
2019-11-06	● 增加模型订阅管理功能。 云服务订阅AI应用 ● 部署边缘服务时，在计算节点规格中，增加“自定义规格”。 部署为边缘服务
2019-10-17	● 数据管理：随软件变更，优化并新增功能。所有相关章节刷新描述，并且新增如下章节。 数据特征 团队标注简介 ● 增加自定义脚本代码示例（包含常用引擎）。 自定义脚本代码示例 ● 增加监控相关描述 ModelArts支持的监控指标
2019-09-30	● 对模型模板的描述进行丰富和优化。 模型模板简介 ● 优化模型包规范的描述，并丰富了模型包示例。 模型包规范介绍 ● 新增提供模型转换功能。 模型转换
2019-08-08	● 在线服务的部署增加自动停止功能，并在详情页面增加事件的信息。 部署为在线服务 查看服务详情
2019-07-18	修改Notebook中选择AI引擎的方式。修改如下章节： Notebook简介 创建Notebook实例
2019-06-30	新增Notebook上传大文件的功能。 上传大文件至Notebook中
2019-06-20	新增关于模型模板的输入输出模式说明。 ● 预置物体检测模式 ● 预置预测分析模式 ● 未定义模式
2019-06-03	导入模板时，新增从模板中导入的功能。 ● 新增：模型模板（即将下线） ● 修改：创建AI应用 开发环境中，增加“Multi-Engine(Recommend)”引擎。 ● 修改：Notebook简介、创建Notebook实例

发布日期	修改说明
2019-05-31	将用户指南拆分成三本文档，分别为自动学习用户指南、AI初学者用户指南、AI工程师用户指南。 本文档命名为AI工程师用户指南，除自动学习之外，包含了ModelArts管理控制台的功能操作指导。 本次刷新，修改了所有大纲，并优化了每个章节的描述语言。
2019-04-16	修改 ● 准备工作章节内容。 ● 开发环境章节内容。
2019-04-12	新增 ● 图像分类简介章节内容。 ● 物体检测简介章节内容。 ● 数据标注-文本标注章节内容。 ● 数据标注-语音内容章节内容。 修改 ● 修改了模型管理章节内容。 ● 图像分类-模型训练章节内容。 ● 物体检测-模型训练章节内容。 ● 优化了开发环境章节内容。 ● 优化了在线服务章节内容。
2019-04-04	新增 ● 数据标注-声音分类章节内容。
2019-04-01	修改 ● 文档导读。 ● 准备工作。 ● 自动学习。 ● 数据标注-Beta。 ● 训练作业。 ● 模型管理。 ● 部署上线。 ● AI市场。
2019-03-25	新增 ● 新增数据标注-Beta章节内容。
2019-03-18	修改 ● 修改了图像分类、物体检测和预测分析章节内容。 ● 调整专属资源池位置。

发布日期	修改说明
2019-02-22	新增 ● 新增了自动学习声音分类，包括构建声音分类模型流程、声音分类。
2019-01-21	新增 ● 新增了模型包规范。 ● 新增Notebook中**与OBS同步文件**介绍、**使用ModelArts SDK**等内容。
2018-12-21	修改 ● 修改了**导入模型**中导入模型内容。 ● 修改了**在线服务**、**批量服务**、**边缘服务**章节内容。
2018-12-03	修改 ● 调整第三章节结构、修改部分内容。
2018-11-15	修改 ● 修改了使用预置算法快速生成模型快速入门样例数据下载路径及步骤。
2018-11-08	第一次正式发布。