

ModelArts

故障排除

文档版本

01

发布日期

2022-06-06



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。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <https://www.huawei.com>

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

目录

1 自动学习	1
1.1 准备数据	1
1.1.1 数据集版本发布失败	1
1.1.2 数据集版本不合格	3
1.2 模型训练	4
1.2.1 自动学习训练作业创建失败	4
1.2.2 自动学习训练作业失败	4
1.3 部署上线	7
1.3.1 部署上线任务提交失败	7
1.3.2 部署上线失败	8
1.4 模型发布	8
1.4.1 模型发布任务提交失败	9
1.4.2 模型发布失败	9
2 开发环境	11
2.1 OBS 操作相关故障	11
2.1.1 进行 OBS 操作时，出现 Error: 403 Forbidden 错误?	11
2.1.2 Terminal 重启后，其中安装的数据丢失如何处理?	12
2.1.3 使用 Sync OBS 从 OBS 同步数据报错，能同步的总文件大小是否有限制?	12
2.1.4 Notebook 中下载 OBS 文件时提示 Permission denied	13
2.2 环境配置故障	13
2.2.1 Terminal 环境无法正常访问，导入第三方安装包报错	13
2.2.2 如何在 Notebook 中导入 Python 库，解决 ModuleNotFoundError 错误?	14
2.2.3 在 Jupyter 页面新建 Terminal 之后找不到目录和文件	15
2.2.4 Notebook 提示磁盘空间已满	15
2.3 实例故障	16
2.3.1 创建 Notebook 实例后无法打开页面，如何处理?	16
2.3.2 使用 pip install 时出现“没有空间”的错误	17
2.3.3 出现“save error”错误，可以运行代码，但是无法保存	17
2.3.4 单击 Notebook 的打开按钮时报“请求超时”错误?	17
2.3.5 新建一个 ipynb 文件时，出现 Error loading notebook 的错误	17
2.3.6 Notebook Examples 加载失败，出现'_xsrf' argument missing from POST 错误	18
2.3.7 Notebook 无法引用同目录下的.py 文件	20
2.3.8 Notebook 保存“ipynb”文件报错	21

2.3.9 出现 ModelArts.6333 错误，如何处理？	22
2.4 代码运行故障.....	22
2.4.1 Notebook 运行代码报错，在'/tmp'中找不到文件.....	22
2.4.2 Notebook 无法执行代码，如何处理？	23
2.4.3 运行训练代码，出现 dead kernel，并导致实例崩溃.....	24
2.4.4 如何解决训练过程中出现的 cudaCheckError 错误？	24
2.4.5 开发环境提示空间不足，如何解决？	25
2.4.6 如何处理使用 opencv.imshow 造成的内核崩溃？.....	25
2.4.7 使用 Windows 下生成的文本文件时报错找不到路径？.....	25
3 训练作业.....	26
3.1 OBS 操作相关故障.....	26
3.1.1 读取文件报错，如何正确读取文件？	26
3.1.2 TensorFlow-1.8 作业连接 OBS 时反复出现提示错误.....	27
3.1.3 TensorFlow 在 OBS 写入 TensorBoard 到达 5GB 时停止.....	27
3.1.4 保存模型时出现 Unable to connect to endpoint 错误.....	28
3.1.5 训练作业日志中提示 “No such file or directory”，如何解决？	28
3.1.6 OBS 拷贝过程中提示 “BrokenPipeError: Broken pipe”	29
3.1.7 日志提示 “ValueError: Invalid endpoint: obs.xxxx.com”	30
3.1.8 日志提示 “errorMessage:The specified bucket does not exist”	31
3.2 云上迁移适配故障.....	32
3.2.1 无法导入模块.....	32
3.2.2 训练作业日志中提示 “No module named .*”	33
3.2.3 如何安装第三方包，安装报错的处理方法.....	34
3.2.4 下载代码目录失败.....	35
3.2.5 训练作业日志中提示 “No such file or directory”	36
3.2.6 训练过程中无法找到 so 文件.....	37
3.2.7 无法解析参数，日志报错.....	37
3.2.8 训练输出路径被其他作业使用.....	38
3.2.9 使用自定义镜像创建训练作业，找不到启动文件.....	38
3.2.10 Pytorch1.0 引擎提示 “RuntimeError: std::exception”	39
3.2.11 MindSpore 日志提示 “retCode=0x91, [the model stream execute failed]”	39
3.2.12 使用 moxing 适配 OBS 路径，pandas 读取文件报错.....	40
3.2.13 日志提示 “Please upgrade numpy to >= xxx to use this pandas version”	41
3.2.14 重装的包与镜像装 CUDA 版本不匹配.....	41
3.2.15 创建训练作业提示错误码 ModelArts.2763.....	42
3.3 内存限制故障.....	42
3.3.1 下载或读取文件报错，提示超时、无剩余空间.....	42
3.3.2 拷贝数据至容器中空间不足.....	44
3.3.3 Tensorflow 多节点作业下载数据到/cache 显示 No space left.....	44
3.3.4 日志文件的大小达到限制.....	44
3.3.5 日志提示“write line error”	45
3.3.6 日志提示 “No space left on device”	46

3.3.7 OOM 导致训练作业失败.....	47
3.4 外网访问限制.....	49
3.4.1 日志提示 “ Network is unreachable ”	49
3.4.2 运行训练作业时提示 URL 连接超时.....	49
3.5 权限问题.....	50
3.5.1 日志提示 “ reason:Forbidden ”	50
3.5.2 日志提示 “ Permission denied ”	50
3.6 GPU 相关问题.....	51
3.6.1 日志提示 “ No CUDA-capable device is detected ”	51
3.6.2 日志提示 “ RuntimeError: connect() timed out ”	52
3.6.3 日志提示 “ cuda runtime error (10) : invalid device ordinal at xxx ”	53
3.6.4 日志提示 “ RuntimeError: Cannot re-initialize CUDA in forked subprocess ”	54
3.6.5 训练作业找不到 GPU.....	55
3.7 业务代码问题.....	55
3.7.1 日志提示 “ pandas.errors.ParserError: Error tokenizing data. C error: Expected .* fields ”	55
3.7.2 日志提示 “ max_pool2d_with_indices_out_cuda_frame failed with error code 0 ”	56
3.7.3 训练作业失败，返回错误码 139.....	56
3.7.4 训练作业失败，如何使用云上环境调试训练代码？	57
3.7.5 日志提示 “ '(slice(0, 13184, None), slice(None, None, None))' is an invalid key ”	57
3.7.6 日志报错 “ DataFrame.dtypes for data must be int, float or bool ”	58
3.7.7 日志提示 “ CUDNN_STATUS_NOT_SUPPORTED. ”	59
3.7.8 日志提示 “ Out of bounds nanosecond timestamp ”	59
3.7.9 日志提示 “ Unexpected keyword argument passed to optimizer ”	60
3.7.10 日志提示 “ no socket interface found ”	60
3.7.11 分布式 Tensorflow 无法使用 “ tf.variable ”	61
3.7.12 MXNet 创建 kvstore 时程序被阻塞，无报错.....	62
3.7.13 日志出现 ECC 错误，导致训练作业失败.....	62
3.7.14 超过最大递归深度导致训练作业失败.....	63
3.7.15 使用预置算法训练时，训练失败，报 “ bndbox ” 错误.....	63
3.7.16 训练作业状态显示 “ 审核作业初始化 ”	63
3.7.17 训练作业进程异常退出.....	64
3.7.18 训练作业进程被 kill.....	64
4 模型管理.....	66
4.1 Caffe 模型转换不成功.....	66
4.2 TensorFlow 模型转换失败.....	67
4.3 自定义镜像模型部署为在线服务时出现异常.....	68
4.4 部署的在线服务状态为告警.....	69
5 MoXing.....	70
5.1 使用 MoXing 复制数据报错.....	70
5.2 如何关闭 Mox 的 warmup.....	71
5.3 Pytorch Mox 日志反复输出.....	72
5.4 moxing.tensorflow 是否包含整个 TensorFlow，如何对生成的 checkpoint 进行本地 Fine Tune?	72

5.5 训练作业使用 MoXing 拷贝数据较慢，重复打印日志..... 73

5.6 MoXing 如何访问文件夹并使用 get_size 读取文件夹大小? 74

6 修订记录.....75

1 自动学习

1.1 准备数据

1.1.1 数据集版本发布失败

出现此问题时，表示数据不满足数据管理模块的要求，导致数据集发布失败，无法执行自动学习的下一步流程。

请根据如下几个要求，检查您的数据，将不符合要求的数据排除后再重新启动自动学习的训练任务。

ModelArts.4710 OBS 权限问题

ModelArts在跟OBS交互时，由于权限相关的问题导致。当界面提示“OBS service Error Message”信息时，表示是由于OBS权限导致的问题，请参考如下步骤排除故障。如果界面错误提示不包含此信息，则是因为后台服务故障导致，建议[联系华为云技术支持](#)。

1. 检查当前帐号是否具备OBS权限。

如果当前帐号是个IAM用户（即子帐号），需确认当前帐号是否具备OBS服务操作权限。

请参考[OBS权限管理](#)，为当前IAM用户配置“作用范围”为“全局级服务”的“Tenant Administrator”策略，即拥有OBS服务所有操作权限。

如果需要限制此IAM用户操作，仅为该用户配置OBS相关的最小化权限项，具体操作请参见[创建ModelArts自定义策略](#)。

2. 检查OBS桶是否具备权限。

说明

下方步骤描述中所致的OBS桶，指创建自动学习项目时，指定的OBS桶，或者是创建项目时选择的数据集，其数据存储所在的OBS桶。

- 检查当前帐号具备OBS桶的读写权限（桶ACLs）

- 进入OBS管理控制台，选择当前自动学习项目使用的OBS桶，单击桶名称进入概览页。

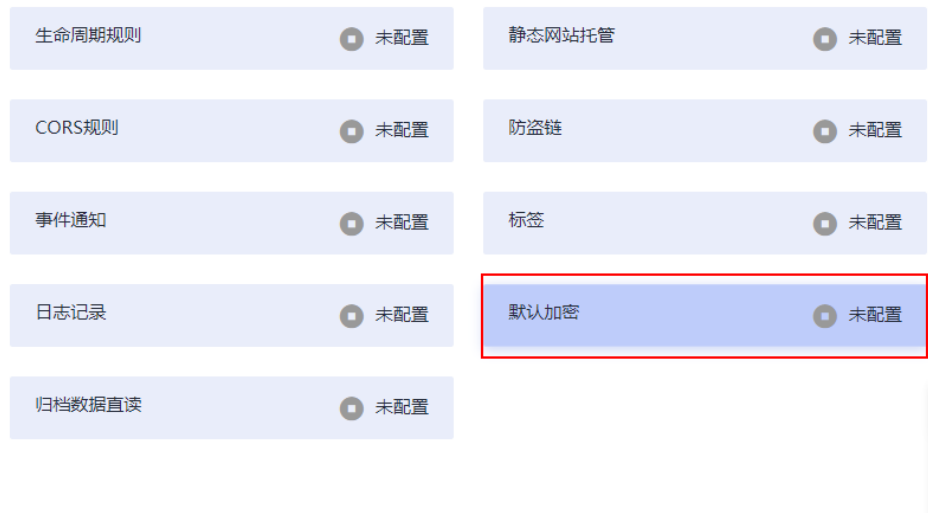
- 在左侧菜单栏选择“访问权限控制>桶ACL”，检查当前帐号是否具备读写权限，如果没有权限，请联系桶的拥有者配置权限。

图 1-1 桶 ACLs



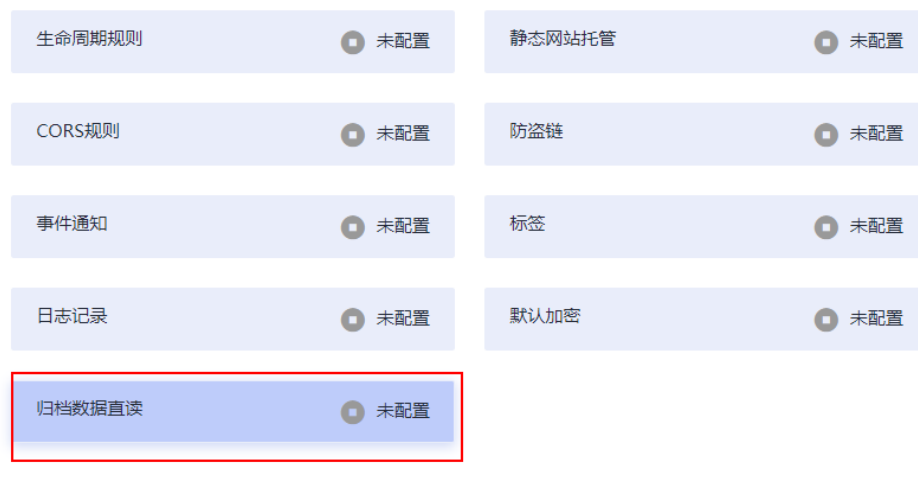
- 确保此OBS桶是非加密桶
 - i. 进入OBS管理控制台，选择当前自动学习项目使用的OBS桶，单击桶名称进入概览页。
 - ii. 确保此OBS桶的加密功能关闭。如果此OBS桶为加密桶，可单击“默认加密”选项进行修改。

图 1-2 OBS 桶是否加密



- 确保归档数据直读功能关闭
 - i. 进入OBS管理控制台，选择当前自动学习项目使用的OBS桶，单击桶名称进入概览页。
 - ii. 确保此OBS桶的归档数据直读功能关闭。如果此功能开启，可单击“归档数据直读”选项进行修改。

图 1-3 关闭归档数据直读功能



ModelArts.4711 数据集标注样本数满足算法要求

每个类别至少包含5张以上图片。

ModelArts.4342 标注信息不满足切分条件

出现此故障时，建议根据如下建议，修改标注数据后重试。

- 多标签的样本（即一张图片包含多个标签），至少需要有2张。如果启动训练时，设置了数据集切分功能，如果多标签的数据少于2张，会导致数据集切分失败。建议检查您的标注信息，保证标注多标签的图片，超过2张。
- 数据集切分后，训练集和验证集包含的标签类别不一样。出现这种情况的原因：多标签场景下时，做随机数据切分后，包含某一类标签的样本均被划分到训练集，导致验证集无该标签样本。由于这种情况出现的概率比较小，可尝试重新发布版本来解决。

ModelArts.4371 数据集版本已存在

出现此错误码时，表示数据集版本已存在，请重新发布数据集版本。

ModelArts.4712 数据集正在执行导入或同步等其他任务

如果自动学习中使用的数据集，正在执行导入或同步数据的任务时，此时进行训练将出现此错误。建议等待其他任务完成后，再启动自动学习的训练任务。

1.1.2 数据集版本不合格

出现此问题时，表示数据集版本发布成功，但是不满足自动学习训练作业要求，因此出现数据集版本不合格的错误提示。

标注信息不满足训练要求

针对不同类型的自动学习项目，训练作业对数据集的要求如下。

- 图像分类：用于训练的图片，至少有2种以上的分类（即2种以上的标签），每种分类的图片数不少于5张。

- 物体检测：用于训练的图片，至少有1种以上的分类（即1种以上的标签），每种分类的图片数不少于5张。
- 预测分析：由于预测分析任务的数据集不在数据管理中进行统一管理，即使数据不满足要求，不在此环节出现故障信息。
- 声音分类：用于训练的音频，至少有2种以上的分类（即2种以上的标签），每种分类的音频数不少于5个。
- 文本分类：用于训练的文本，至少有2种以上的分类（即2种以上的标签），每种分类的文本数不少于20个。

1.2 模型训练

1.2.1 自动学习训练作业创建失败

出现此问题，一般是因为后台服务故障导致的，建议稍等片刻，然后重新创建训练作业。如果重试超过3次仍无法解决，请[联系华为云技术支持](#)。

1.2.2 自动学习训练作业失败

训练作业创建成功，但是在运行过程中，由于一些故障导致作业运行失败。

首次请检查您的帐户是否欠费。如果帐号状态正常。请针对不同类型的作业进行排查。

- 针对图像分类、声音分类、文本分类的作业，排查思路请参见[确保OBS中的数据存在](#)、[检查OBS的访问权限](#)、[检查图片是否符合要求](#)。
- 针对物体检测作业，排查思路请参见[确保OBS中的数据存在](#)、[检查OBS的访问权限](#)、[检查图片是否符合要求](#)、[检查标注框是否符合要求（物体检测）](#)。
- 针对预测分析作业，排查思路请参见[确保OBS中的数据存在](#)、[检查OBS的访问权限](#)、[预测分析作业失败的排查思路](#)。

确保 OBS 中的数据存在

如果存储在OBS中的图片或数据被删除，且未同步至ModelArts自动学习或数据集中，则会导致任务失败。

建议前往OBS检查，确保数据存在。针对图像分类、声音分类、文本分类、物体检测等类型，可在自动学习的数据标注页面，单击“同步数据源”，将OBS中的数据重新同步至ModelArts中。

检查 OBS 的访问权限

如果OBS桶的访问权限设置无法满足训练要求时，将会出现训练失败。请排查如下几个OBS的权限设置。

- 当前帐号具备OBS桶的读写权限（桶ACLs）
 - a. 进入OBS管理控制台，选择当前自动学习项目使用的OBS桶，单击桶名称进入概览页。
 - b. 在左侧菜单栏选择“访问权限控制>桶ACLs”，检查当前帐号是否具备读写权限，如果没有权限，请联系桶的拥有者配置权限。

图 1-4 桶 ACLs



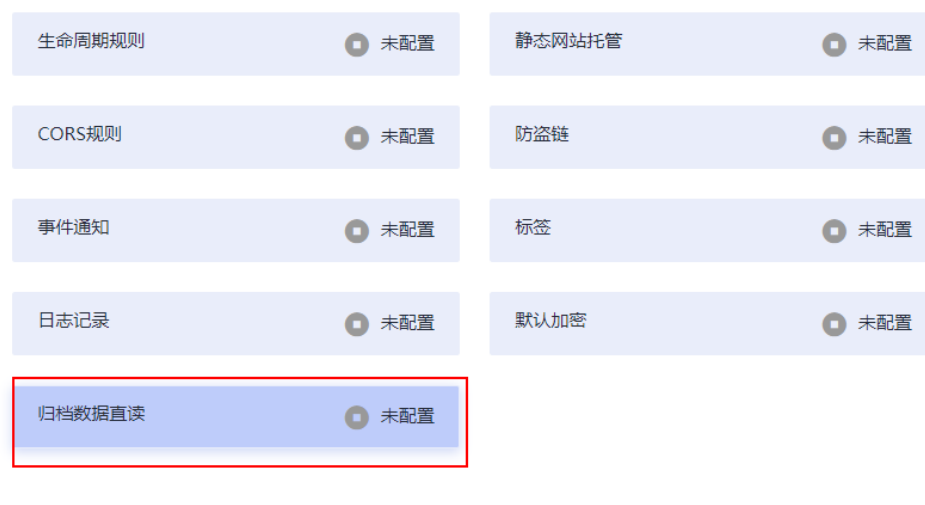
- 确保此OBS桶是非加密桶
 - a. 进入OBS管理控制台，选择当前自动学习项目使用的OBS桶，单击桶名称进入概览页。
 - b. 确保此OBS桶的加密功能关闭。如果此OBS桶为加密桶，可单击“默认加密”选项进行修改。

图 1-5 OBS 桶是否加密



- 确保归档数据直读功能关闭
 - a. 进入OBS管理控制台，选择当前自动学习项目使用的OBS桶，单击桶名称进入概览页。
 - b. 确保此OBS桶的归档数据直读功能关闭。如果此功能开启，可单击“归档数据直读”选项进行修改。

图 1-6 关闭归档数据直读功能



- 确保OBS中的文件是非加密状态
上传图片或文件时不要选择KMS加密，否则会导致数据集读取失败。文件加密无法取消，请先解除桶加密，重新上传图片或文件。

图 1-7 OBS 桶中的文件未加密

<input type="checkbox"/>	名称	存储类别	大小	加密状态	恢复状态
<input type="checkbox"/>	2.png	标准存储	1.05 KB	未加密	--

检查图片是否符合要求

目前自动学习不支持四通道格式的图片。请检查您的数据，排除或删除四通道格式的图片。

检查标注框是否符合要求（物体检测）

目前物体检测仅支持矩形标注框。请确保所有图片的标注框为矩形框。

如果使用非矩形框，可能存在以下报错：

Error: bandbox.

针对其他类型的项目（图像分类、声音分类等），无需关注此问题。

预测分析作业失败的排查思路

1. 检查用于预测分析的数据是否满足要求。

由于预测分析任务未使用数据管理的功能发布数据集，因此当数据不满足训练作业要求时，会出现训练作业运行失败的错误。

建议检查用于训练的数据，是否满足预测分析作业的要求。要求如下所示，如果数据满足要求，执行下一步检查。如果不满足要求，请根据要求仅需数据调整后重新训练。

- 文件规范：名称由以字母数字及中划线下划线组成，以'.csv'结尾，且文件不能直接放在OBS桶的根目录下，应该存放在OBS桶的文件夹内。如：“/obs-xxx/data/input.csv”。

- 文件内容：文件保存为“csv”文件格式，文件内容以换行符（即字符“\n”，或称为LF）分隔各行，行内容以英文逗号（即字符“,”）分隔各列。文件内容不能包含中文字符，列内容不应包含英文逗号、换行符等特殊字符，不支持引号语法，建议尽量以字母及数字字符组成。
 - 训练数据：训练数据列数一致，总数据量不少于100条不同数据（有一个特征取值不同，即视为不同数据）。训练数据列内容不能有时间戳格式（如：yy-mm-dd、yyyy-mm-dd等）的数据。确保指定标签列的取值至少有两个且无数据缺失，除标签列外数据集中至少还应包含两个有效特征列（列的取值至少有两个且数据缺失比例低于10%）。训练数据的csv文件不能包含表头，否则会导致训练失败。当前由于特征筛选算法限制，标签列建议放在数据集最后一列，否则可能导致训练失败。
2. 由于ModelArts会自动对数据进行一些过滤，过滤后再启动训练作业。当预处理后的数据不满足训练要求时，也会导致训练作业运行失败。
- 对于数据集中列的过滤策略如下所示：
- 如果某一列空缺的比例大于系统设定的阈值（0.9），此列数据在训练时将被剔除。
 - 如果某一列只有一种取值（即每一行的数据都是一样的），此列数据在训练时将被剔除。
 - 对于非纯数值列，如果此列的取值个数等于行数（即每一行的数值都是不一样的），此列数据在训练时将被剔除。
- 经过上述过滤后，如果数据集不再满足第一点中关于训练数据的要求，则会导致训练失败或无法进行。建议完善数据后，再启动训练。
3. 数据集文件有以下限制：
- a. 如果您使用2u8g规格，测试建议数据集文件应小于10MB。当文件大小符合限制要求，如果存在极端的数据规模（行数列数之积）时，仍可能会导致训练失败，建议的数据规模低于10000。
 - 如果您使用8u32g规格，测试建议数据集文件应小于100MB。当文件大小符合限制要求，如果存在极端的数据规模（行数列数之积）时，仍可能会导致训练失败，建议的数据规模低于1000000。
4. 如果上述排查操作仍无法解决，请[联系华为云技术支持](#)。

1.3 部署上线

1.3.1 部署上线任务提交失败

当出现此错误时，一般情况是由于帐号的配额受限导致的。

在自动学习项目中，启动部署后，会自动将模型部署为一个在线服务，如果由于配额限制（即在线服务的个数超出配额限制），导致无法将模型部署为服务。此时会在自动学习项目中提示“部署上线任务提交失败”的错误。

修改建议

- 方法1：进入“部署上线>在线服务”页面，将不再使用的服务删除，释放资源。
- 方法2：如果您部署的在线服务仍需继续使用，建议申请增加配额。

1.3.2 部署上线失败

出现此问题，一般是因为后台服务故障导致的，建议稍等片刻，然后重新部署在线服务。如果重试超过3次仍无法解决，请获取如下信息，并[联系华为云技术支持](#)协助解决故障。

- 获取服务ID。
进入“部署上线>在线服务”页面，在服务列表中找到自动学习任务中部署的在线服务，自动学习部署的服务都是以“exeML-”开头的。单击服务名称进入服务详情页面，在“基本信息”区域，获取“服务ID”的值。

图 1-8 获取服务 ID



- 获取在线服务事件信息。
进入服务详情页面后，单击“事件”页签，将事件信息表截图后反馈给技术支持人员。

图 1-9 获取事件信息

调用指南 预测 配置更新记录 难例筛选 监控信息 事件 日志 共享		
2020/11/09 15:14:47 — 202...X 图		
事件类型	事件信息	事件发生时间 下三
正常	automatically stop service that reached due time	2020/11/09 16:24:06 GMT+08:00
正常	start model success	2020/11/09 15:23:40 GMT+08:00
正常	pull image success	2020/11/09 15:23:40 GMT+08:00
正常	pulling model image	2020/11/09 15:20:29 GMT+08:00
正常	schedule resource success	2020/11/09 15:20:29 GMT+08:00
正常	prepare environment success	2020/11/09 15:20:28 GMT+08:00
正常	preparing environment	2020/11/09 15:20:07 GMT+08:00
正常	model (exeML-yunbao_ExeML_7ac5c286 0.0.1) build image success	2020/11/09 15:19:59 GMT+08:00
正常	building image for model [exeML-yunbao_ExeML_7ac5c286 0.0.1]	2020/11/09 15:14:49 GMT+08:00

1.4 模型发布

1.4.1 模型发布任务提交失败

出现此问题，一般是因为后台服务故障导致的，建议稍等片刻，然后重新创建训练作业。如果重试超过3次仍无法解决，请[联系华为云技术支持](#)。

1.4.2 模型发布失败

出现此问题，一般是因为后台服务故障导致的，建议稍等片刻，然后重新创建训练作业。如果重试超过3次仍无法解决，请获取如下信息，并[联系华为云技术支持](#)协助解决故障。

- 获取模型ID。

进入“模型管理>模型”页面，在模型列表中找到自动学习任务中自动创建的模型，自动学习产生的模型都是以“exeML-”开头的。单击模型名称进入模型详情页面，在“基本信息”区域，获取“ID”的值。

图 1-10 获取模型 ID

基本信息

名称	exeML-61a1_ExeML_7569f55d	标签	--
状态	✔ 正常	版本	0.0.3
ID	b6c718e0-8820-486d-a666-5362f3ae8049	大小	167.60 MB
运行环境	tf1.13-python3.7-cpu	AI引擎	TensorFlow
部署类型	在线服务/批量服务	描述	-- 
模型文档	--		

- 获取模型事件信息。

进入模型详情页面后，单击“事件”页签，将事件信息表截图后反馈给技术支持人员。

图 1-11 获取事件信息

参数配置 | 运行时依赖 | **事件**

2020/10/09 11:50:51 - 202...X 全部 C

事件类型	事件信息	事件发生时间 三
正常	Image built successfully.	2020/10/16 11:14:16 GMT+08:00
正常	The status of the image building task is READY.	2020/10/16 11:14:16 GMT+08:00
正常	The status of the image building task is CREATING.	2020/10/16 11:13:56 GMT+08:00
正常	The status of the image building task is CREATING.	2020/10/16 11:13:36 GMT+08:00
正常	The status of the image building task is CREATING.	2020/10/16 11:13:16 GMT+08:00
正常	The status of the image building task is CREATING.	2020/10/16 11:12:55 GMT+08:00
正常	The status of the image building task is CREATING.	2020/10/16 11:12:35 GMT+08:00
正常	The status of the image building task is CREATING.	2020/10/16 11:12:15 GMT+08:00
正常	Start the image building task.	2020/10/16 11:11:47 GMT+08:00
正常	Model imported successfully.	2020/10/09 11:51:03 GMT+08:00

2 开发环境

2.1 OBS 操作相关故障

2.1.1 进行 OBS 操作时，出现 Error: 403 Forbidden 错误?

问题现象

Notebook中，进行OBS操作时，如使用mox.file.copy_parallel时，出现Error: stat:403 错误。

图 2-1 错误现象

```
ERROR:root:
  stat:403
  errorCode:None
  errorMessage:None
  reason:Forbidden
  request-id:000001752610DE67600F295F15304A6C
  retry:0
```

原因分析

- 可能是ModelArts的全局配置使用是AK/SK访问密钥授权且AK/SK删除重建过。
- 可能是没有OBS桶的权限。

解决方法

- 如果ModelArts的全局配置使用是AK/SK访问密钥授权且AK/SK删除重建过。建议前往全局配置，重新配置访问密钥（AK/SK）
- 进入OBS管理控制台，查找对应的OBS桶，单击桶名称进入概览页。在左侧菜单栏选择“访问权限控制>桶ACLs”，检查当前帐号是否具备读写权限，如果没有权限，请联系桶的拥有者配置权限。

图 2-2 桶 ACLs



2.1.2 Terminal 重启后，其中安装的数据丢失如何处理？

Terminal重启，如果出现Terminal中安装的数据丢失，可能原因是该数据没有保存指定目录导致。为避免该问题出现，可参照如下步骤进行处理。

- EVS实例会自动同步到“~work”目录下，如果是相关代码数据，需要您将代码数据指定到该目录。
- OBS实例重启后“~work”目录下的数据会被删除。建议您在使用之前手动同步数据到该目录下。操作方式请参见[使用Sync OBS同步](#)。

为避免重启，请勿在开发环境中进行重型作业训练，如大量占用CPU、GPU或者内存的作业。

2.1.3 使用 Sync OBS 从 OBS 同步数据报错，能同步的总文件大小是否有限制？

问题现象

Notebook从OBS同步数据报错，导致无法正常使用：obs sync failed。

原因分析

如您在创建Notebook实例时，选择的“存储配置”为OBS的，则Notebook实例里的文件都是保存在您选择的OBS目录下，如您需使用该文件，则必须使用Sync OBS功能将文件同步到当前Notebook实例的“~/work”目录下。

- 单次同步文件个数最多是1024个。
- 同步的对象总大小不超过5GB，单次同步最大值为500MB。即当前容器目录“~/work”下已有200MB的文件了，那么用户使用Sync OBS单次能同步的最多就只有300MB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。

处理方法

根据您的报错提示，判断是单次同步文件内容超过使用限制。

1. 如果您单次同步文件个数超过1024，请您整理文件之后重新同步。
2. 如果同步的文件和当前“/home/ma-user/work”目录下的文件大于500MB，请您清理该目录下的文件空间后重新同步。

2.1.4 Notebook 中下载 OBS 文件时提示 Permission denied

问题现象

通过OBS下载数据到Notebook中时，提示：

```
Exception: ('Download file from OBS failed! ', PermissionError(13, 'Permission denied'))
```

原因分析

OBS文件访问权限问题

处理方法

1. 登录OBS控制台。
2. OBS桶是否和Notebook在同一个区域，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](#)。
3. 查看您的帐号是否有该OBS桶的访问权限。如没有权限，请参见[在Notebook中，如何访问其他帐号的OBS桶？](#)。
4. 查看OBS桶中待下载的文件是否加密。

2.2 环境配置故障

2.2.1 Terminal 环境无法正常访问，导入第三方安装包报错

问题现象

- 在Terminal中使用命令：source activate xxx 无法进入对应环境。
- 基础框架的包或者使用命令**pip install**安装的包在Notebook里无法导入。
- 在Notebook中导包时会遇到如下错误，导致无法正常使用：ImportError: No module named XXX。

原因分析

根据以上问题现象，可以判断是未激活Terminal环境导致的。

处理方法

1. 登录ModelArts管理控制台，选择“开发环境>Notebook”。
2. 在Notebook列表中，单击目标Notebook“操作”列的“打开”，进入“Jupyter”开发页面。
3. 在Jupyter页面的“Files”页签下，单击“New”，然后选择“Terminal”，进入到Terminal界面。

图 2-3 进入 Terminal 界面



4. 执行如下命令获取激活环境命令行，如图2-4所示。

```
cat /home/ma-user/README
```

例如，需要激活“TensorFlow-1.8”，可执行如下命令进入到TensorFlow-1.8的环境并进行开发。

```
source /home/ma-user/anaconda3/bin/activate TensorFlow-1.8
```

图 2-4 激活 Terminal 环境

```
sh-4.3$ cat /home/ma-user/README
Please use one of following command to start the specified framework environment.

for Conda-python3 ----- source /home/ma-user/anaconda3/bin/activate base
for MXNet-1.2.1 (CUDA 9.0) ----- 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 (CUDA 9.0) ----- source /home/ma-user/anaconda3/bin/activate Pytorch-1.0.0
for TensorFlow-1.13.1 (CUDA 10.0) ----- source /home/ma-user/anaconda3/bin/activate TensorFlow-1.13.1
for TensorFlow-1.8 (CUDA 9.0) ----- source /home/ma-user/anaconda3/bin/activate TensorFlow-1.8
for XGBoost-Sklearn ----- source /home/ma-user/anaconda3/bin/activate XGBoost-Sklearn
```

2.2.2 如何在 Notebook 中导入 Python 库，解决 ModuleNotFoundError 错误？

对于挂载 EVS 的 Notebook 实例导入 python 库

1. 获取需要导入的Python库的地址，然后参见Python3的“[将文件夹加入到sys.path](#)”的操作指导，完成python库的导入。导入操作有两种方式，比较常用的是使用PYTHONPATH环境变量来添加。
2. 导入之后，您可以在Notebook中查看您的PythonPath。
 - a. 在“ipynb”中查看PythonPath。在代码输入框中执行如下命令查看PythonPath，如果返回的地址与您设置的python库地址一致，表示导入成功。
!echo \$PYTHONPATH
 - b. 在“Terminal”中查看PythonPath。执行如下命令查看PythonPath，如果返回的地址与您设置的python库地址一致，表示导入成功。
echo \$PYTHONPATH

对于带 OBS 存储的 Notebook 实例导入 python 库

对于带OBS存储的Notebook实例导入python库，根据python库文件大小不同，使用方式有所不同。

1. 当python库的文件较小（小于100MB）时，您可以使用如下2种方式。
 - 首先，将python文件上传至OBS，然后使用[Sync OBS功能](#)方式将OBS中的python文件同步到Notebook中。最后参见Python3的“[将文件夹加入到sys.path](#)”指导（推荐使用PYTHONPATH环境变量来添加），完成python库的导入。
 - 首先，将python文件上传至OBS，然后使用[SDK](#)将OBS中的文件同步到Notebook，最后参见Python3的“[将文件夹加入到sys.path](#)”指导（推荐使用PYTHONPATH环境变量来添加），完成python库的导入。
2. 当python库的文件较大（大于100MB）时
首先，将python库的文件上传至OBS，然后使用[Moxing操作OBS文件](#)将OBS中的python文件同步到Notebook，最后参见Python3的“[将文件夹加入到sys.path](#)”指导（推荐使用PYTHONPATH环境变量来添加），完成python库的导入。

在python库完成导入后，您可以在Notebook中查看您的PythonPath。

1. 在“ipybn”中查看PythonPath。在代码输入框中执行如下命令查看PythonPath，如果返回的地址与您设置的pyhon库地址一致，表示导入成功。
!echo \$PYTHONPATH
2. 在“Terminal”中查看PythonPath。执行如下命令查看PythonPath，如果返回的地址与您设置的pyhon库地址一致，表示导入成功。
echo \$PYTHONPATH

2.2.3 在 Jupyter 页面新建 Terminal 之后找不到目录和文件

问题现象

创建Notebook进入Jupyter页面，选择进入到Terminal界面无法读取目录和文件。

原因分析

根据该现象，可以判断是文件未同步导致无法读取。

处理方法

针对这个问题，有两种情况：

- 如果您创建的Notebook使用OBS存储实例
 - a. 在Notebook列表中，打开目标Notebook。
 - b. 在Jupyter页面的“Files”页签下，勾选目标文件后，单击页面上方的“Sync OBS”，同步完成后，文件保存在实例的“~/work”目录下。
- 如果您创建的Notebook不使用OBS存储
使用MoXing将数据拷贝至Notebook，MoXing文件操作调用示例请参见[MoXing操作OBS文件](#)。

2.2.4 Notebook 提示磁盘空间已满

问题现象

在使用Notebook时，出现如下报错，提示磁盘空间已满：No Space left on Device。

原因分析

报错提示磁盘空间已满，需要查看具体的空间占用情况，删除不必要的大文件。

处理方法

根据报错提示，查看磁盘使用空间和处理方法如下：

1. 进入到Terminal界面，执行如下命令，查看所有盘的空间占用情况，比如“~work”目录挂载的盘，一般会显示已满。

```
df -Th
```

2. 再到Terminal界面，执行如下命令进入work目录。

```
cd work
```

3. 在“~work”目录下，执行如下命令，查看该目录下的空间占用情况，如果仍未查询到大文件，则可能是隐藏文件。

```
du -sh *
```

4. 在“~work”目录下，执行如下命令，查看该目录下的空间占用情况，包含隐藏文件，请删除不用的大文件即可。

```
du -h -d 1
```

- a. 删除示例文件“test.txt”：

```
rm -f /home/ma-user/work/data/test.txt
```

- b. 删除示例文件夹“data”：

```
rm -rf /home/ma-user/work/data/
```

建议与总结

建议在使用Notebook时注意磁盘空间大小，随时删除不需要的文件。以免因磁盘空间问题导致训练失败。

2.3 实例故障

2.3.1 创建 Notebook 实例后无法打开页面，如何处理？

如果您在创建Notebook实例之后，打开Notebook时，因报错导致无法打开页面，您可以根据以下对应的错误码来排查解决。

报错 404

如果是IAM用户在创建实例时出现此错误，表示此IAM用户不具备对应存储位置（OBS桶）的操作权限。

解决方法：

1. 使用帐号登录OBS，并将对应OBS桶的访问权限授予该IAM用户。详细操作指导请参见：[被授权用户](#)。
2. IAM用户获得权限后，登录ModelArts管理控制台，删除该实例，然后重新使用此OBS路径创建Notebook实例。

报错 503

如果出现503错误，可能是由于该实例运行代码时比较耗费资源。建议先停止当前Notebook实例，然后重新启动。

报错 504

如果报此错误时，请提工单或拨打热线电话协助解决。提工单和热线电话请参见：
<https://www.huaweicloud.com/service/contact.html>。

2.3.2 使用 pip install 时出现“没有空间”的错误

问题现象

在Notebook实例中，使用**pip install**时，出现“No Space left...”的错误。

解决办法

建议使用**pip install --no-cache **** 命令安装，而不是使用**pip install ****。加上“--no-cache”参数，可以解决很多此类报错。

2.3.3 出现“save error”错误, 可以运行代码，但是无法保存

如果当前Notebook还可以运行代码，但是无法保存，保存时会提示“save error”错误。大多数原因是华为云WAF安全拦截导致的。

当前页面，即用户的输入或者代码运行的输出有一些字符被华为云拦截，认为有安全风险。出现此问题时，请提交工单，联系专业的工程师帮您核对并处理问题。

2.3.4 单击 Notebook 的打开按钮时报“请求超时”错误？

当Notebook容器因内存溢出等原因导致奔溃时，若此时单击Notebook的打开按钮时，将会出现“请求超时”错误。

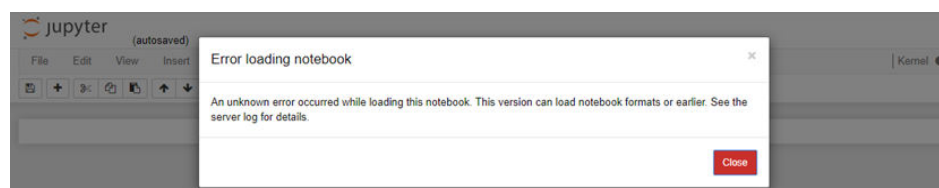
该种情况下，请耐心等待容器恢复，约几十秒，再重新单击打开按钮即可。

2.3.5 新建一个 ipynb 文件时，出现 Error loading notebook 的错误

问题现象

在Notebook Jupyter页面中，新建一个ipynb文件时，出现“Error loading notebook”错误提示，如何处理？

图 2-5 错误提示



原因分析

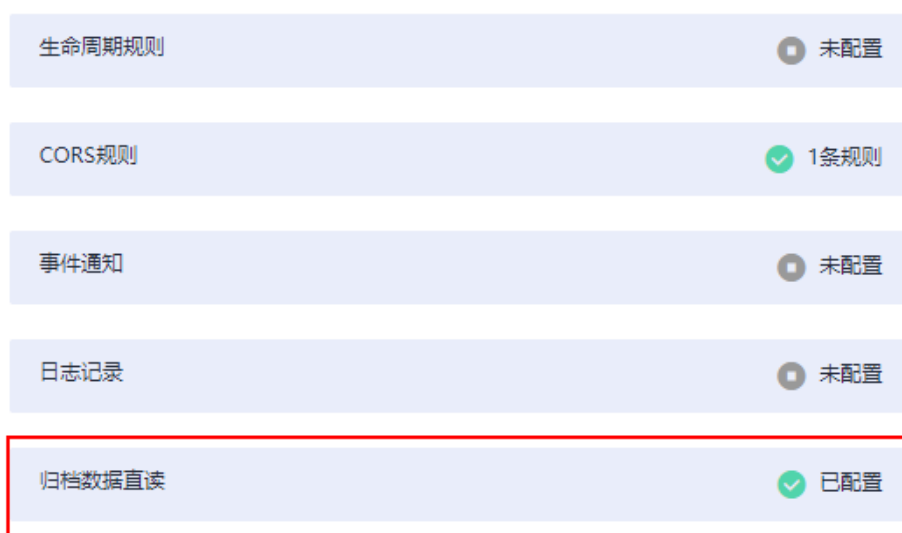
出现这个错误，其根本原因是您用于创建Notebook的OBS桶的属性。如果您使用的OBS桶，其“存储类别”为“归档存储”，且“归档数据直读”功能关闭，此时ModelArts Notebook会出现无法新建一个ipynb的错误。

解决方案

进入OBS管理控制台，选择此Notebook实例对应的桶，然后单击桶名称进入桶详情页面，在“基础配置”区域，找到“归档数据直读”功能设置，单击此功能，在弹出框中启用“归档数据直读”功能。设置完成后，Notebook实例可正常新建ipynb文件。

图 2-6 启用功能

基础配置



2.3.6 Notebook Examples 加载失败，出现'_xsrf' argument missing from POST 错误

问题现象

当打开一个使用GPU的Notebook后，jupyter页面的“ModelArts Examples”页签无法加载，引擎类型无法加载，并且在jupyter页面中创建文件夹失败。详情如图2-7和图2-8所示。

图 2-7 “ModelArts Examples” 页签和引擎类型未加载

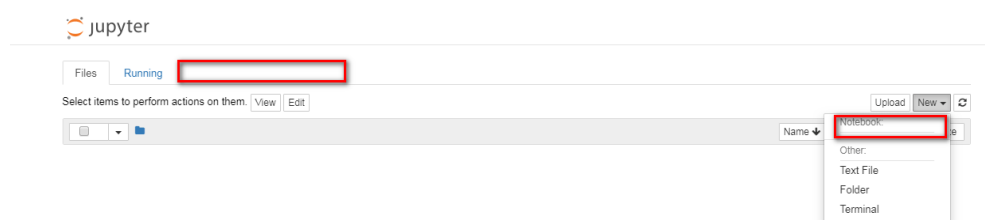
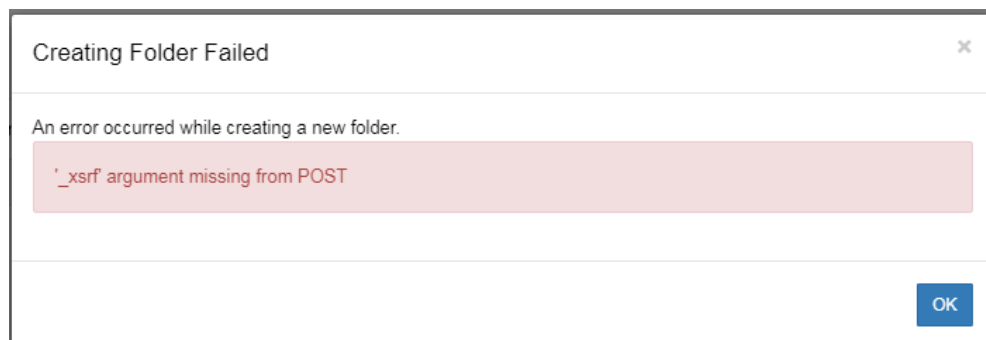


图 2-8 创建文件夹失败



原因分析

- 访问Notebook GPU需要用到第三方网站，该网站需要打开浏览器cookies功能才能正常访问。出现上述问题后，建议打开谷歌浏览器的cookies功能以解决问题。
- 当使用较高版本的Chrome浏览器访问时，由于一些默认设置（如“SameSite by default cookies”），导致无法打开此页签。建议修改浏览器的默认配置解决此问题。

解决方法

打开谷歌浏览器的cookies功能。


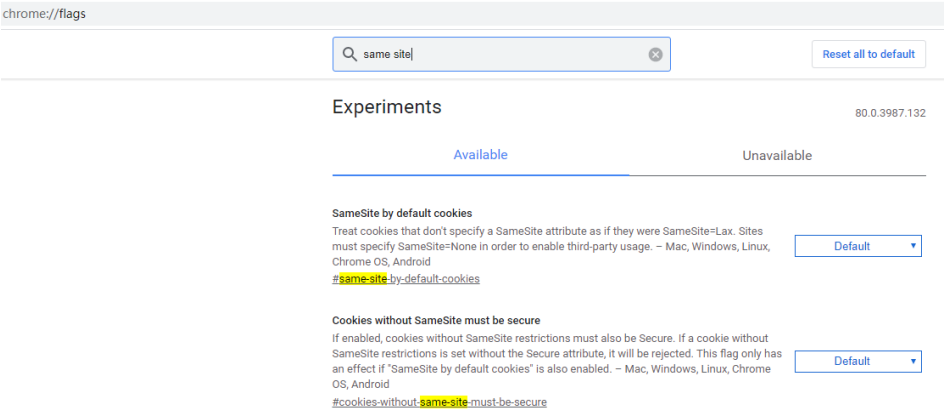
- 在谷歌浏览器的右上角，单击 选择“设置”。
- 在搜索框中输入“cookie”搜索，在搜索结果中，选择“网站设置”。
- 进入“网站设置”选项，单击“Cookie”进入详细设置。如图2-9所示，将“阻止第三方Cookie”的选项设置为不启用状态。

图 2-9 Cookie 设置



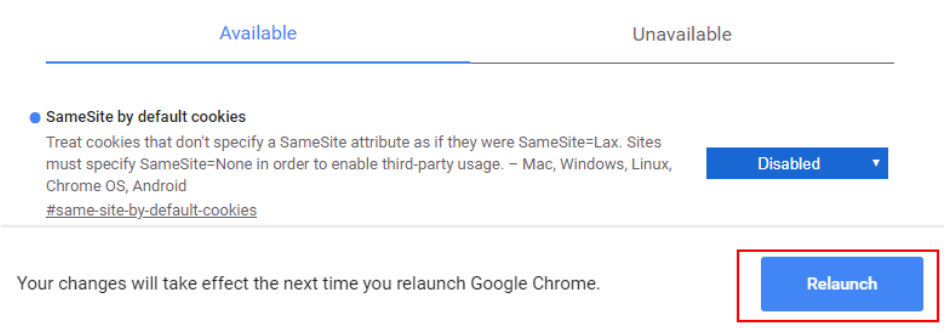
- 如果您使用的Chrome浏览器版本为80.XX及以上版本时，且上述步骤无法解决问题时，您可以尝试如下操作。
 - 在浏览器中输入“chrome://flags”。
 - 在打开的网页的搜索框中，输入“same site”搜索相关配置，然后在“SameSite by default cookies”选项中，其默认为“Default”，将其参数设置为“Disabled”。

图 2-10 设置 Same Site 配置



- c. 参数设置完成后，需要在浏览器下方，单击“Relaunch”，使配置在重启 Chrome 浏览器后生效。

图 2-11 重启 Chrome



- d. 重新启动浏览器后，进入ModelArts管理控制台，查看Example页签是否已展示。

说明

如果根据上述操作仍无法解决问题，请提交工单，联系专业技术人员为您服务。

2.3.7 Notebook 无法引用同目录下的.py 文件

问题现象

在Notebook的Terminal中执行命令python a.py无法引用同目录下的“.py”文件，出现如下报错：

ModuleNotFoundError: No module named 'b'。

目录结构如图2-14，操作和报错请参见图2-13。

图 2-12 目录结构

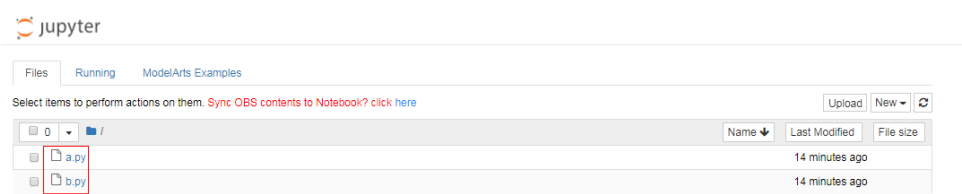


图 2-13 引用报错

```
sh-4.3$python a.py
Traceback (most recent call last):
  File "a.py", line 1, in <module>
    import b
ModuleNotFoundError: No module named 'b'
```

原因分析

在Terminal中执行，命令`ls`查看文件目录，“a.py”和“b.py”未在相同的目录下。

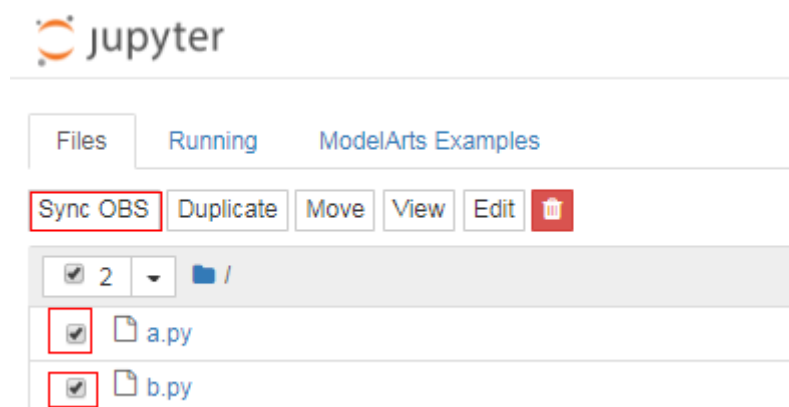
图 2-14 查看文件目录

```
sh-4.3$ls
a.py
```

处理方法

1. 登录ModelArts管理控制台，选择“开发环境>Notebook”。
2. 在Notebook列表中，单击目标Notebook“操作”列的“打开”，进入“Jupyter”开发页面。
3. 在Jupyter页面的“Files”页签下，选择“a.py”和“b.py”两个文件，然后点击“Sync OBS”将选中的对象从OBS桶路径下同步到当前容器目录“~/work”下，即可在代码中调用。

图 2-15 同步文件



2.3.8 Notebook 保存“ipynb”文件报错

问题现象

使用Notebook进入Jupyter页面，保存“ipynb”文件报错：The file has changed on disk since the last time we opened or saved it. xxxx。

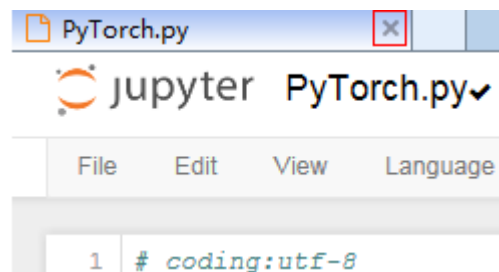
原因分析

根据该现象，可以判断是两个及以上的开发环境同时在改同一个文件导致保存失败。

处理方法

请关闭其他开发环境中打开的同一个文件，如图2-16所示。

图 2-16 关闭开发环境



2.3.9 出现 ModelArts.6333 错误，如何处理？

问题现象

在使用Notebook过程中，界面出现“ModelArts.6333”报错信息。

原因分析

可能由于实例过负载引起故障，Notebook正在自动恢复中，请刷新页面并等待几分钟。常见原因是内存占用满。

处理方法

当出现此错误时，Notebook会自动恢复，您可以刷新页面，等待几分钟。

由于出现此错误，常见原因是内存占用满导致的，您可以尝试使用如下方法，从根本上解决错误。

- 方法1：将Notebook更换为更高规格的资源。
- 方法2：可以参考如下方法调整代码中的参数，减少内存占用。如果代码调整后仍然出现内存不足的情况，请使用方法1。
 - a. 调用sklearn方法`silhouette_score(addr_1,siteskmeans.labels)`，可以指定参数`sample_size`来减少内存占用。
 - b. 调用`train`方法的时候可以尝试减少`batch_size`等参数。

2.4 代码运行故障

2.4.1 Notebook 运行代码报错，在'/tmp'中找不到文件

问题现象

使用Notebook运行代码，报错：

```
FileNotFoundError: [Error 2] No usable temporary directory found in ['/tmp', '/var/tmp', '/usr/tmp',  
'home/ma-user/work/SR/RDN_train_base']
```

图 2-17 运行代码报错

```
(PyTorch-1.0.0) sh-4.3$ python
Python 3.6.4 [Anaconda, Inc.] (default, Mar 13 2018, 01:15:57)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import mxing
INFO:root:Using MoXing-v1.13.0-de903ac9
INFO:root:Using OBS-Python-SDK-3.1.2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    from mxing.framework import *
  File "/home/ma-user/anaconda3/envs/PyTorch-1.0.0/lib/python3.6/site-packages/mxing/__init__.py", line 22, in <module>
    from mxing.framework import *
  File "/home/code/mxing/build/mxing/framework/_init_.py", line 31, in <module>
    File "/home/code/mxing/build/mxing/framework/file/_init_.py", line 28, in <module>
    File "/home/code/mxing/build/mxing/framework/file/file_io.py", line 119, in <module>
    File "/home/ma-user/anaconda3/envs/PyTorch-1.0.0/lib/python3.6/tempfile.py", line 296, in gettempdir
    tempdir = _get_default_tempdir()
  File "/home/ma-user/anaconda3/envs/PyTorch-1.0.0/lib/python3.6/tempfile.py", line 231, in _get_default_tempdir
    dirlist)
FileNotFoundError: [Errno 2] No usable temporary directory found in ['/tmp', '/var/tmp', '/usr/tmp', '/home/ma-user/work/SR
FROM_train_base']
>>>
[2]+  Stopped (SIGTSTP)      python
(PyTorch-1.0.0) sh-4.3$ df -hl
```

原因分析

根据报错提示，需要排查是否将大量数据被保存在“/tmp”中。

处理方法

- 进入到“Terminal”界面。在“/tmp”目录下，执行命令`du -sh *`，查看该目录下的空间占用情况。

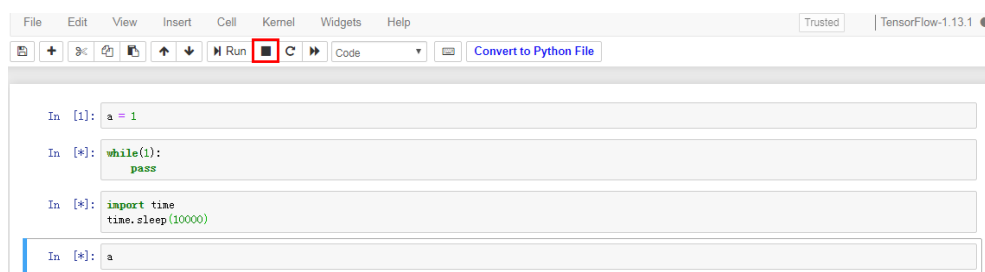
```
sh-4.3$ cd /tmp
sh-4.3$ du -sh *
4.0K  core-js-banners
0      npm-19-41ed4c62
6.7M  v8-compile-cache-1000
```
- 请删除不用的大文件。
 - 删除示例文件“test.txt”：`rm -f /home/ma-user/work/data/test.txt`
 - 删除示例文件夹“data”：`rm -rf /home/ma-user/work/data/`

2.4.2 Notebook 无法执行代码，如何处理？

当Notebook出现无法执行时，您可以根据如下几种情况判断并处理。

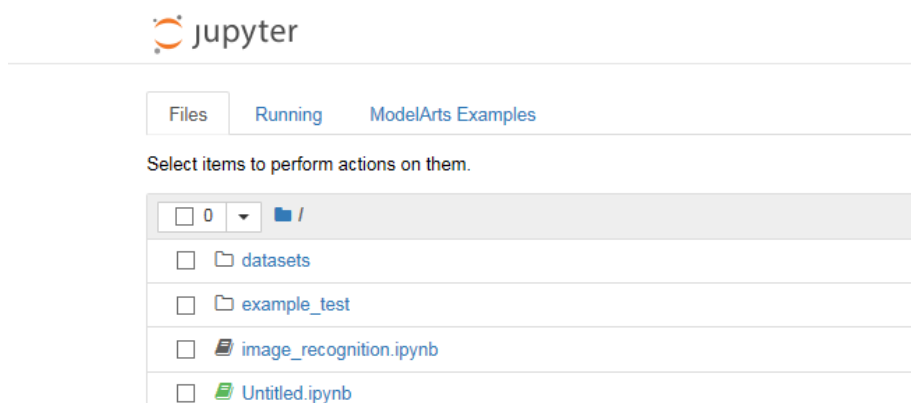
- 如果只是Cell的执行过程卡死或执行时间过长，如图2-18中的第2个和第3个Cell，导致第4个Cell无法执行，但整个Notebook页面还有反应，其他Cell也还可以点击，则直接单击下图红色方框处的“interrupt the kernel”，停止所有Cell的执行，同时会保留当前Notebook中的所有变量空间。

图 2-18 停止所有 Cell



- 如果整个Notebook页面也已经无法使用，单击任何地方都无反应，则关闭Notebook页面，关闭ModelArts管理控制台页面。然后，重新打开管理控制台，打开之前无法使用的Notebook，此时的Notebook仍会保留无法使用之前的所有变量空间。

图 2-19 重新打开 Notebook



3. 如果重新打开的Notebook仍然无法使用，则进入ModelArts管理控制台页面的Notebook列表页面，“停止”此无法使用的Notebook。待Notebook处于“停止”状态后，再单击“启动”，重新启动此Notebook，并打开Notebook。此时，Notebook仍会保留无法使用之前的所有变量空间。

图 2-20 停止 Notebook

创建	您最多可以创建10个Notebook，还可以创建7个Notebook。						全部状态	请输入名称查询	Q	C
名称	状态	AI引擎	描述	创建时间	更新时间	操作				
common_notebook_env_gpu	停止	Multi-Engine-python3.6	~	2019/07/02 16:49:47 GMT+08:00	2019/07/02 17:02:00 GMT+08:00	打开 启动 删除				
common_notebook_envs	停止	Multi-Engine-python3.6	~	2019/06/29 15:31:22 GMT+08:00	2019/07/05 11:49:38 GMT+08:00	打开 启动 删除				
common_notebook	运行中	Multi-Engine-python3.6	使用OBS	2019/06/29 11:19:55 GMT+08:00	2019/07/08 17:58:39 GMT+08:00	打开 停止 删除				

2.4.3 运行训练代码，出现 dead kernel，并导致实例崩溃

在Notebook实例中运行训练代码，如果数据量太大或者训练层数太多，亦或者其他原因，导致出现“内存不够”问题，最终导致该容器实例崩溃。

出现此问题后，系统将自动重启Notebook，来修复实例崩溃的问题。此时只是解决了崩溃问题，如果重新运行训练代码仍将失败。如果您需要解决“内存不够”的问题，建议您创建一个新的Notebook，使用更高规格的资源池，比如GPU或专属资源池来运行此训练代码。已经创建成功的Notebook不支持选用更高规格的资源规格进行扩容。

2.4.4 如何解决训练过程中出现的 cudaCheckError 错误？

问题现象

Notebook中，运行训练代码出现如下错误。

```
cudaCheckError() failed : no kernel image is available for execution on the device
```

原因分析

因为编译的时候需要设置setup.py中编译的参数arch和code和电脑的显卡匹配。

解决方法

对于Tesla V100的显卡，GPU算力为-gencode
arch=compute_70,code=[sm_70,compute_70]，设置setup.py中的编译参数即可解决。

2.4.5 开发环境提示空间不足，如何解决？

当提示空间不足时，推荐使用EVS类型的Notebook实例。

参考[如何在Notebook中读写OBS文件？](#)操作指导，针对原有的Notebook，首先将代码和数据上传至OBS桶中。然后创建一个EVS类型的Notebook，将此OBS中的文件下载至Notebook本地（指新建的EVS类型Notebook）。

2.4.6 如何处理使用 opencv.imshow 造成的内核崩溃？

问题现象

当在Notebook中使用opencv.imshow后，会造成Notebook崩溃。

原因分析

opencv的cv2.imshow在jupyter这样的client/server环境下存在问题。而matplotlib不存在这个问题。

解决方法

参考如下示例进行图片显示。注意opencv加载的是BGR格式，而matplotlib显示的是RGB格式。

Python语言：

```
from matplotlib import pyplot as plt
import cv2
img = cv2.imread('图片路径')
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.title('my picture')
plt.show()
```

2.4.7 使用 Windows 下生成的文本文件时报错找不到路径？

问题现象

当在Notebook中使用Windows下生成的文本文件时，文本内容无法正确读取，可能报错找不到路径。

原因分析

Notebook是Linux环境，和Windows环境下的换行格式不同，Windows下是CRLF，而Linux下是LF。

解决方法

可以在Notebook中转换文件格式为Linux格式。

shell语言：

```
dos2unix 文件名
```

3 训练作业

3.1 OBS 操作相关故障

3.1.1 读取文件报错，如何正确读取文件？

问题现象

- 创建训练作业如何读取“json”和“npz”文件。
- 训练作业如何使用cv2库读取文件。
- 如何在MXNet环境下使用torch包。
- 训练作业读取文件，出现如下报错：
NotFoundError (see above for traceback): Unsuccessful TensorSliceReader constructor: Failed to find any matching files for xxx://xxx

原因分析

在ModelArts中，用户的数据都是存放在OBS桶中，而训练作业运行在容器中，无法通过访问本地路径的方式访问OBS桶中的文件。

处理方法

读取文件报错，您可以使用Moxing将数据拷贝至容器中，再直接访问容器中的数据。请参见步骤1。

您也可以根据不同的文件类型，进行读取。请参见[读取“json”文件](#)、[读取“npz”文件](#)、[使用cv2库读取文件](#)和[在MXNet环境下使用torch包](#)。

1. 读取文件报错，您可以使用Moxing将数据拷贝至容器中，再直接访问容器中的数据。具体方式如下：

```
import moxing as mox
mox.file.make_dirs('/cache/data_url')
mox.file.copy_parallel('obs://bucket-name/data_url', '/cache/data_url')
```
2. 读取“json”文件，请您在代码中尝试如下方法：

```
json.loads(mox.file.read(json_path, binary=True))
```
3. 使用“numpy.load”读取“npz”文件，请您在代码中尝试如下方法：

- 使用MoXing API读取OBS中的文件
`np.load(mox.file.read(_SAMPLE_PATHS['rgb'], binary=True))`
- 使用MoXing的file模块对OBS文件进行读写
with `mox.file.File(_SAMPLE_PATHS['rgb'], 'rb')` as f:
`np.load(f)`
- 4. 使用cv2库读取文件，请您尝试如下方法：
`cv2.imdecode(np.fromstring(mox.file.read(img_path), np.uint8), 1)`
- 5. 在MXNet环境下使用torch包，请您尝试如下方法先进行导包：
`import os`
`os.system('pip install torch')`
`import torch`

3.1.2 TensorFlow-1.8 作业连接 OBS 时反复出现提示错误

问题现象

基于TensorFlow-1.8启动训练作业，并在代码中使用“tf.gfile”模块连接OBS，启动训练作业后会频繁打印如下日志信息：

```
Connection has been released. Continuing.  
Found secret key
```

原因分析

这是TensorFlow-1.8中会出现的情况，该日志是Info级别的，并不是错误信息，可以通过设置环境变量来屏蔽INFO级别的日志信息。环境变量的设置一定要在import tensorflow或者import moxing之前。

处理方法

您需要通过在代码中设置环境变量“TF_CPP_MIN_LOG_LEVEL”来屏蔽INFO级别的日志信息。具体操作如下：

```
import os  
  
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'  
  
import tensorflow as tf  
import moxing.tensorflow as mox
```

“TF_CPP_MIN_LOG_LEVEL”与日志等级对应关系为：

```
import os  
os.environ["TF_CPP_MIN_LOG_LEVEL"]="1" # 默认的显示等级，显示所有信息  
os.environ["TF_CPP_MIN_LOG_LEVEL"]="2" # 只显示warning和Error  
os.environ["TF_CPP_MIN_LOG_LEVEL"]="3" # 只显示Error
```

3.1.3 TensorFlow 在 OBS 写入 TensorBoard 到达 5GB 时停止

问题现象

ModelArts训练作业出现如下报错：

```
Encountered Unknown Error EntityTooLarge  
Your proposed upload exceeds the maximum allowed object size.:  
If the signature check failed. This could be because of a time skew. Attempting to adjust the signer
```

原因分析

OBS限制单次上传文件大小为5GB，TensorFlow保存summary可能是本地缓存，在每次触发flush时将该summary文件覆盖OBS上的原文件。当超过5GB后，由于达到了OBS单次导入文件大小的上限，导致无法继续写入。

处理方法

如果在运行训练作业的过程中出现该问题，建议处理方法如下：

1. 推荐使用本地缓存的方式来解决，使用如下方法：

```
import mxing.tensorflow as mx
mx.cache()
```

3.1.4 保存模型时出现 Unable to connect to endpoint 错误

问题现象

训练作业保存模型时日志报错，具体信息如下：

InternalError (see above for traceback): : Unable to connect to endpoint

原因分析

OBS连接不稳定可能会出现报错，“Unable to connect to endpoint”。

处理方法

对于OBS连接不稳定的现象，通过增加代码来解决。您可以在代码最前面增加如下代码，让TensorFlow对ckpt和summary的读取和写入可以通过本地缓存的方式中转解决：

```
import mxing.tensorflow as mx
mx.cache()
```

3.1.5 训练作业日志中提示 “No such file or directory”，如何解决？

使用ModelArts时，用户数据是存放在自己OBS桶中，OBS桶中数据都有对应的路径，例如“bucket_name/dir/image.jpg”。ModelArts训练作业是运行在容器中，如果要访问OBS数据，需要通过数据对应的路径。此提示说明该文件或路径不存在，可能由于您在创建训练作业时，选择的“数据存储位置”有误，或者您编写的代码文件中，访问的路径不正确。

用户请按照以下思路进行逐步排查：

1. [检查报错的路径是否为OBS路径](#)
2. [检查报错的路径是否存在](#)

检查报错的路径是否为 OBS 路径

使用ModelArts时，用户数据需要存放在自己OBS桶中，但是训练代码运行过程中不能使用OBS路径读取数据。

原因：

训练作业创建成功后，由于在运行容器直连OBS服务进行训练性能很差，系统会自动下载训练数据至运行容器的本地路径。所以，在训练代码中直接使用OBS路径会报错。

如果报错路径为训练数据路径，需要在以下两个地方完成适配，具体适配方法请参考自定义算法适配章节的[输入输出配置部分](#)：

1. 在创建算法时，您需要在输入路径配置中设置[代码路径参数](#)，默认为“data_url”。
2. 您需要在训练代码中添加超参，默认为“data_url”。使用“data_url”当做训练数据输入的本地路径。

检查报错的路径是否存在

由于用户本地开发的代码需要上传至ModelArts后台，训练代码中涉及到依赖文件的路径时，用户设置有误的场景较多。

推荐通用的解决方案：使用os接口得到依赖文件的绝对路径，避免报错。

示例：

```
!---project_root      #代码根目录
|---BootfileDirectory #启动文件所在的目录
|---bootfile.py       #启动文件
|---otherfileDirectory #其他依赖文件所在的目录
|---otherfile.py       #其他依赖文件
```

在启动文件中，建议用户参考以下方式获取依赖文件所在路径，即示例中的otherfile_path。

```
import os
current_path = os.path.dirname(os.path.realpath(__file__)) # BootfileDirectory, 启动文件所在的目录
project_root = os.path.dirname(current_path) # 工程的根目录，对应ModelArts训练控制台上设置的代码目录
otherfile_path = os.path.join(project_root, "otherfileDirectory", "otherfile.py")
```

3.1.6 OBS 拷贝过程中提示 “BrokenPipeError: Broken pipe”

问题现象

训练作业在使用moxing拷贝数据时出现如下报错。

图 3-1 错误日志

```
readable=readable)
File "/home/work/anaconda/lib/python3.6/site-packages/moxing/framework/file/src/obs/client.py", line 358, in _make_put_request
  chunkedMode, methodName=methodName, readable=readable)
File "/home/work/anaconda/lib/python3.6/site-packages/moxing/framework/file/src/obs/client.py", line 390, in _make_request_with_retry
  raise e
File "/home/work/anaconda/lib/python3.6/site-packages/moxing/framework/file/src/obs/client.py", line 369, in _make_request_with_retry
  _redirectLocation, skipAuthentication=skipAuthentication)]
File "/home/work/anaconda/lib/python3.6/site-packages/moxing/framework/file/src/obs/client.py", line 436, in _make_request_internal
  conn = self._send_request(connect_server, method, path, header_config, entity, port, scheme, redirect, chunkedMode)
File "/home/work/anaconda/lib/python3.6/site-packages/moxing/framework/file/src/obs/client.py", line 586, in _send_request
  entity(util.conn_delegate(conn))
File "/home/work/anaconda/lib/python3.6/site-packages/moxing/framework/file/src/obs/util.py", line 250, in entity
  conn.send(chunk)
File "/home/work/anaconda/lib/python3.6/site-packages/moxing/framework/file/src/obs/util.py", line 154, in send
  self.conn.send(data)
File "/home/work/anaconda/lib/python3.6/http/client.py", line 986, in send
  self.sock.sendall(data)
File "/home/work/anaconda/lib/python3.6/ssl.py", line 972, in sendall
  v = self.send(byte_view(count))
File "/home/work/anaconda/lib/python3.6/ssl.py", line 941, in send
  return self._sslobj.write(data)
File "/home/work/anaconda/lib/python3.6/ssl.py", line 642, in write
  return self._sslobj.write(data)
BrokenPipeError: [Errno 32] Broken pipe
```

原因分析

出现该问题的可能原因如下。：

- 在大规模分布式作业上，每个节点都在拷贝同一个桶的文件，导致OBS桶限流。
- OBS Client连接数过多，进程/线程之间的轮询，导致一个OBS Client与服务端连接30S内无响应，超过超时时间，服务端断开了连接。

处理方法

1. 如果是限流问题，日志中还会有如下错误，OBS相关的错误码解释请看 [OBS官方文档](#)，这种情况建议提工单。

图 3-2 错误日志

```
[ModelArts Service Log]2021-01-21 11:35:42,178 - file_io.py[line:652] - ERROR: Fail  
func= <bound method ObsClient.getObjectMetadata of <moxing.frame  
args=('bucket-816', 'AIRAW_AJ/c00454567/TeleQtj/23_zyl_J_quad_Tele  
kwargs={}  
[ModelArts Service Log]2021-01-21 11:35:42,178 - file_io.py[line:658] - ERROR:  
stat:503  
errorCode:None  
errorMessage:None  
reason:Service Unavailable  
request-id:000001772302B34C9019B2408F9FF1B2  
retry:0
```

2. 如果是client数太多，尤其对于5G以上文件，OBS接口不支持直接调用，需要分多个线程分段拷贝，目前OBS侧服务端超时时间是30S，可以通过如下设置减少进程数。

```
import moxing as mox  
  
mox.file.set_auth(is_secure=False)  
from moxing.framework.file import file_io  
file_io._NUMBER_OF_PROCESSES=1  
mox.file.copy_parallel(threads=0, is_processing=False)
```

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.1.7 日志提示“ValueError: Invalid endpoint: obs.xxxx.com”

问题现象

训练作业中使用Tensorboard直接写入到OBS路径，出现如下类似报错。

图 3-3 错误日志

```
Traceback (most recent call last):
  File "/home/work/anaconda/lib/python3.6/threading.py", line 916, in _bootstrap_inner
    self.run()
  File "/home/work/anaconda/lib/python3.6/site-packages/tensorboardX/event_file_writer.py", line 219, in run
    self._record_writer.flush()
  File "/home/work/anaconda/lib/python3.6/site-packages/tensorboardX/event_file_writer.py", line 69, in flush
    self._py_recordio_writer.flush()
  File "/home/work/anaconda/lib/python3.6/site-packages/tensorboardX/record_writer.py", line 187, in flush
    self._writer.flush()
  File "/home/work/anaconda/lib/python3.6/site-packages/tensorboardX/record_writer.py", line 89, in flush
    s3 = boto3.client('s3', endpoint_url=os.environ.get('S3_ENDPOINT'))
  File "/home/work/anaconda/lib/python3.6/site-packages/boto3/_init_.py", line 91, in client
    return _get_default_session().client(*args, **kwargs)
  File "/home/work/anaconda/lib/python3.6/site-packages/boto3/session.py", line 263, in client
    aws_session_token=aws_session_token, config=config)
  File "/home/work/anaconda/lib/python3.6/site-packages/botocore/session.py", line 835, in create_client
    client_config=config, api_version=api_version)
  File "/home/work/anaconda/lib/python3.6/site-packages/botocore/client.py", line 85, in create_client
    verify, credentials, scoped_config, client_config, endpoint_bridge)
  File "/home/work/anaconda/lib/python3.6/site-packages/botocore/client.py", line 287, in _get_client_args
    verify, credentials, scoped_config, client_config, endpoint_bridge)
  File "/home/work/anaconda/lib/python3.6/site-packages/botocore/args.py", line 107, in get_client_args
    client_cert=new_config.client_cert)
  File "/home/work/anaconda/lib/python3.6/site-packages/botocore/endpoint.py", line 261, in create_endpoint
    raise ValueError("Invalid endpoint: %s" % endpoint_url)
ValueError: Invalid endpoint: obs.myhuaweicloud.com
```

原因分析

出现该问题的可能原因：

直接往OBS上写tensorboard文件，存在不稳定的风险。

处理方法

建议先将Tensorboard文件写到本地，然后再拷贝回OBS。

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.1.8 日志提示 “errorMessage:The specified bucket does not exist”

问题现象

在用moxing访问OBS路径的时候，出现如下错误

```
ERROR:root:
stat:404
errorCode:NoSuchKey
errorMessage:The specified key does not exist.
```

原因分析

出现该问题的可能原因如下：

桶中的对象不存在，请检查OBS路径中的内容是否存在。具体错误码可以参考 [OBS官方文档](#)。

处理方法

1. 检查OBS路径及内容格式是否正常。
2. 必现的问题，使用本地Pycharm远程连接Notebook调试。

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.2 云上迁移适配故障

3.2.1 无法导入模块

问题现象

ModelArts训练作业导入模块时日志报错：

```
Traceback (most recent call last):File "project_dir/main.py", line 1, in <module>from module_dir import module_file
ImportError: No module named module_dir
ImportError: No module named xxx
```

原因分析

- 训练作业导入模块时日志出现前两条报错信息，可能原因如下：
代码如果在本地运行，需要将“project_dir”加入到PYTHONPATH或者将整个“project_dir”安装到“site-package”中才能运行。但是在ModelArts可以将“project_dir”加入到“sys.path”中解决该问题。
使用**from module_dir import module_file**来导包，代码结构如下：

```
project_dir
|- main.py
|- module_dir
|  |- __init__.py
|  |- module_file.py
```
- 训练作业导入模块时日志出现“ImportError: No module named xxx”的报错，可以判断是环境中没有包含用户依赖的python包。

处理方法

- 训练作业导入模块时日志出现前两条报错信息，处理方法如下：
 - a. 首先保证被导入的module中有“__init__.py”存在，创建“module_dir”的“__init__.py”，如[原因分析](#)中的结构所示。
 - b. 由于无法知晓“project_dir”在容器中的位置，所以利用绝对路径，在“main.py”中将“project_dir”添加到“sys.path”中，再导入：

```
import os
import sys
# __file__ 为获取当前执行脚本main.py的绝对路径
# os.path.dirname(__file__)获取main.py的父目录，即project_dir的绝对路径
current_path = os.path.dirname(__file__)
sys.path.append(current_path)
# 在sys.path.append执行完毕之后再导入其他模块
from module_dir import module_file
```

- 训练作业导入模块时日志出现 “ImportError: No module named xxx” 的报错，请添加如下代码安装依赖包：

```
import os
os.system('pip install xxx')
```

3.2.2 训练作业日志中提示 “No module named .*”

用户请按照以下思路进行逐步排查：

1. [检查依赖包是否存在](#)
2. [检查依赖包路径是否能被识别](#)
3. 建议与总结

检查依赖包是否存在

如果依赖包不存在，您可以使用以下两种方式完成依赖包的安装。

- 方式一（推荐使用）：在[创建我的算法](#)时，需要在“代码目录”下放置相应的文件或安装包。

请根据依赖包的类型，在代码目录下放置对应文件：

- 依赖包为开源安装包时

在“代码目录”中创建一个命名为“pip-requirements.txt”的文件，并且在文件中写明依赖包的包名及其版本号，格式为“包名==版本号”。

例如，“代码目录”对应的OBS路径下，包含模型文件，同时还存在“pip-requirements.txt”文件。“代码目录”的结构如下所示：

```
|--模型启动文件所在OBS文件夹
|   |--model.py           #模型启动文件。
|   |--pip-requirements.txt #定义的配置文件，用于指定依赖包的包名及版本号。
```

“pip-requirements.txt”文件内容如下所示：

```
alembic==0.8.6
bleach==1.4.3
click==6.6
```

- 依赖包为whl包时

如果训练后台不支持下载开源安装包或者使用用户编译的whl包时，由于系统无法自动下载并安装，因此需要在“代码目录”放置此whl包，同时创建一个命名为“pip-requirements.txt”的文件，并且在文件中指定此whl包的包名。依赖包必须为“.whl”格式的文件。

例如，“代码目录”对应的OBS路径下，包含模型文件、whl包，同时还存在“pip-requirements.txt”文件。“代码目录”的结构如下所示：

```
|--模型启动文件所在OBS文件夹
|   |--model.py           #模型启动文件。
|   |--XXX.whl            #依赖包。依赖多个时，此处放置多个。
|   |--pip-requirements.txt #定义的配置文件，用于指定依赖包的包名。
```

“pip-requirements.txt”文件内容如下所示：

```
numpy-1.15.4-cp36-cp36m-manylinux1_x86_64.whl
tensorflow-1.8.0-cp36-cp36m-manylinux1_x86_64.whl
```


- 方式二：可以在启动文件添加如下代码安装依赖包：

```
import os
os.system('pip install xxx')
```

方式一在训练作业启动前即可完成相关依赖包的下载与安装，而方式二是运行启动文件过程中进行依赖包的下载与安装。

检查依赖包路径是否能被识别

代码如果在本地运行，需要将“project_dir”加入到PYTHONPATH或者将整个“project_dir”安装到“site-package”中才能运行。但是在ModelArts可以将“project_dir”加入到“sys.path”中解决该问题。

使用from module_dir import module_file来导包，代码结构如下：

```
project_dir
|- main.py
|- module_dir
|  |- __init__.py
|  |- module_file.py
```

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.2.3 如何安装第三方包，安装报错的处理方法

问题现象

- ModelArts[如何安装自定义库函数](#)，例如“apex”。
- ModelArts训练环境安装第三方包时出现如下报错：
xxx.whl is not a supported wheel on this platform

原因分析

由于安装的文件名格式不支持，导致出现“xxx.whl is not a supported wheel on this platform”报错，具体解决方法请参见[2](#)。

处理方法

1. 安装第三方包

- a. pip中存在的包，使用如下代码：

```
import os
os.system('pip install xxx')
```

- b. pip源中不存在的包，此处以“apex”为例，请您用如下方式将安装包上传到OBS桶中。该样例已将安装包上传至“obs://cnnorth4-test/codes/mox_benchmarks/apex-master/”中，将在启动文件中添加以下代码进行安装。

```
try:
    import apex
except Exception:
```



```
import os
import moxing as mox
mox.file.copy_parallel('obs://cn-north4-test/codes/mox_benchmarks/apex-master/', '/cache/
apex-master')
os.system('pip --default-timeout=100 install -v --no-cache-dir --global-option="--cpp_ext" --
global-option="--cuda_ext" /cache/apex-master')
```

2. 安装报错

“xxx.whl”文件无法安装，需要您按照如下步骤排查：

- a. 当出现“xxx.whl”文件无法安装，在启动文件中添加如下代码，查看当前pip命令支持的文件名和版本。

```
import pip
print(pip.pep425tags.get_supported())
```

获取到支持的文件名和版本如下：

```
[('cp36', 'cp36m', 'manylinux1_x86_64'), ('cp36', 'cp36m', 'linux_x86_64'), ('cp36', 'abi3',
'manylinux1_x86_64'), ('cp36', 'abi3', 'linux_x86_64'), ('cp36', 'none', 'manylinux1_x86_64'),
('cp36', 'none', 'linux_x86_64'), ('cp35', 'abi3', 'manylinux1_x86_64'), ('cp35', 'abi3',
'linux_x86_64'), ('cp34', 'abi3', 'manylinux1_x86_64'), ('cp34', 'abi3', 'linux_x86_64'), ('cp33',
'abi3', 'manylinux1_x86_64'), ('cp33', 'abi3', 'linux_x86_64'), ('cp32', 'abi3', 'manylinux1_x86_64'),
('cp32', 'abi3', 'linux_x86_64'), ('py3', 'none', 'manylinux1_x86_64'), ('py3', 'none', 'linux_x86_64'),
('cp36', 'none', 'any'), ('cp3', 'none', 'any'), ('py36', 'none', 'any'), ('py3', 'none', 'any'), ('py35',
'none', 'any'), ('py34', 'none', 'any'), ('py33', 'none', 'any'), ('py32', 'none', 'any'), ('py31', 'none',
'any'), ('py30', 'none', 'any')]
```

- b. 将“faiss_gpu-1.5.3-cp36-cp36m-manylinux2010_x86_64.whl”更改为“faiss_gpu-1.5.3-cp36-cp36m-manylinux1_x86_64.whl”，并安装，执行命令如下：

```
import moxing as mox
import os

mox.file.copy('obs://wolfros-net/zip/AI/code/faiss_gpu-1.5.3-cp36-cp36m-
manylinux2010_x86_64.whl', '/cache/faiss_gpu-1.5.3-cp36-cp36m-manylinux1_x86_64.whl')
os.system('pip install /cache/faiss_gpu-1.5.3-cp36-cp36m-manylinux1_x86_64.whl')
```

3.2.4 下载代码目录失败

问题现象

训练作业运行时下载失败，出现如下报错，请参见图3-4：

```
ERROR: modelarts-downloader.py: Get object key failed: 'Contents'
```

图 3-4 获取内容失败

```
2019-07-04 14:12:37,678 - modelarts-downloader.py[line:90] - ERROR: modelarts-downloader.py: Get object key failed: 'Contents'
[Modelarts Service Log][modelarts_logger] modelarts-pipe found
[Modelarts Service Log]App download error:
2019-07-04 14:12:36,574 - modelarts-downloader.py[line:471] - INFO: Main: modelarts-downloader starting with Namespace(dst='./', recursive=True,
6538/1a2ych1u/code/honovod/pretrain/, trace=False, verbose=False)
```

原因分析

在创建训练作业时指定的代码目录不存在导致训练失败。

处理方法

请您根据报错原因排查创建训练作业时指定的代码目录，即OBS桶的路径是否正确。有两种方法判断是否存在。

- 使用当前帐户登录OBS管理控制台，去查找对应的OBS桶、文件夹、文件是否存在。

- 通过接口判断路径是否存在。在代码中执行如下命令，检查路径是否存在。

```
import os
os.path.exists('obs://obs-test/ModelArts/examples/')
```

3.2.5 训练作业日志中提示 “No such file or directory”

用户请按照以下思路进行逐步排查：

1. [检查报错的路径是否为OBS路径](#)
2. [检查报错的路径是否存在](#)
3. 建议与总结

检查报错的路径是否为 OBS 路径

使用ModelArts时，用户数据需要存放在自己OBS桶中，但是训练代码运行过程中不能使用OBS路径读取数据。

原因：

训练作业创建成功后，由于在运行容器直连OBS服务进行训练性能很差，系统会自动下载训练数据至运行容器的本地路径。所以，在训练代码中直接使用OBS路径会报错。

如果报错路径为训练数据路径，需要在以下两个地方完成适配，具体适配方法请参考自定义算法适配章节的[输入输出配置部分](#)：

1. 在创建算法时，您需要在输入路径配置中设置[代码路径参数](#)，默认为“data_url”。
2. 您需要在训练代码中添加超参，默认为“data_url”。使用“data_url”当做训练数据输入的本地路径。

检查报错的路径是否存在

由于用户本地开发的代码需要上传至ModelArts后台，训练代码中涉及到依赖文件的路径时，用户设置有误的场景较多。

推荐通用的解决方案：使用os接口得到依赖文件的绝对路径，避免报错。

示例：

```
!--project_root      #代码根目录
!--BootfileDirectory #启动文件所在的目录
!--bootfile.py       #启动文件
!--otherfileDirectory #其他依赖文件所在的目录
!--otherfile.py       #其他依赖文件
```

在启动文件中，建议用户参考以下方式获取依赖文件所在路径，即示例中的otherfile_path。

```
import os
current_path = os.path.dirname(os.path.realpath(__file__)) # BootfileDirectory, 启动文件所在的目录
project_root = os.path.dirname(current_path) # 工程的根目录，对应ModelArts训练控制台上设置的代码目录
otherfile_path = os.path.join(project_root, "otherfileDirectory", "otherfile.py")
```

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.2.6 训练过程中无法找到 so 文件

问题现象

ModelArts训练作业运行时，日志中遇到如下报错，导致训练失败：

```
libcudart.so.9.0 cannot open shared object file no such file or directory
```

原因分析

编译生成so文件的cuda版本与训练作业的cuda版本不一致。

处理方法

编译环境的cuda版本与训练环境不一致，训练作业运行就会报错。例如：使用cuda版本为10的开发环境tf-1.13中编译生成的so包，在cuda版本为9.0训练环境中tf-1.12训练会报该错。

编译环境和训练环境的cuda版本不一致时，可参考如下处理方法：

1. 在业务执行前加如下命令，检查是否能找到so文件。如果已经找到so文件，执行[2](#)；如果没有找到，执行[3](#)。

```
import os;
os.system(find /usr -name *libcudart.so*);
```

2. 设置环境变量LD_LIBRARY_PATH，设置完成后，重新下发作业即可。

例如so文件的存放路径为：/usr/local/cuda/lib64，LD_LIBRARY_PATH设置如下：

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda/lib64 )
```

3. 执行如下命令，查看训练环境的cuda版本，确认当前cuda版本是否支持so文件。

```
os.system("cat /usr/local/cuda/version.txt")
```

 - a. 支持。当前cuda版本无so文件，需外部导入so文件（自行在浏览器下载），再设置LD_LIBRARY_PATH，具体见[2](#)。
 - b. 不支持。尝试更换引擎，重新下发作业。或者使用自定义镜像创建作业，可参考[使用自定义镜像创建作业](#)。

3.2.7 无法解析参数，日志报错

问题现象

ModelArts训练作业无法解析参数，遇到如下报错，导致无法正常运行：

```
error: unrecognized arguments: --data_url=xxx://xxx/xxx
absl.flags._exceptions.UnrecognizedFlagError:Unknown command line flag 'task_index'
```

原因分析

在训练环境中，系统可能会传入在Python脚本里没有定义的其他参数名称，导致参数无法解析，日志报错。

处理方法

您需要通过使用解析方式`args, unparsed = parser.parse_known_args()`代替`args = parser.parse_args()`解决该问题。代码示例如下：

```
import argparse
parser = argparse.ArgumentParser()
parser.add_argument('--data_url', type=str, default=None, help='obs path of dataset')
args, unparsed = parser.parse_known_args()
```

3.2.8 训练输出路径被其他作业使用

问题现象

在创建训练作业时出现如下报错：操作失败！ Other running job contain train_url: / bucket-20181114/code_hxm/

原因分析

根据报错信息判断，在创建训练作业时，同一个“训练输出路径”在被其他作业使用。

处理方法

一个“训练输出路径”只能被一个处于“运行中”、“排队中”或“初始化”状态的作业使用。

当出现此报错时，建议检查并重新训练作业的“训练输出路径”，以避免创建作业失败。

3.2.9 使用自定义镜像创建训练作业，找不到启动文件

问题现象

使用旧版训练的自定义镜像创建训练作业，出现如下报错，提示找不到运行的主文件：no such file or directory。

原因分析

根据报错提示可以判断是运行命令的启动文件目录不正确导致运行失败。

处理方法

需要排查执行命令的启动文件目录是否正确，具体操作如下：

在ModelArts管理控制台，使用训练的自定义镜像创建训练作业时，“算法来源”选择“自定义”页签。

若训练代码启动脚本在OBS路径为“obs://bucket-name/app/code/train.py”，创建作业时配置代码目录为“/bucket-name/app/code/”。

代码目录配置完成后，执行如下命令，那么“run_train.sh”将选中的“code”文件夹下载到旧版训练容器的“/home/work/user-job-dir”目录中。

```
bash /home/work/run_train.sh #旧版训练命令，run_train.sh训练启动引导脚本，打包在ModelArts提供的基础镜像中。
```

运行命令就可以设置为：

```
bash /home/work/run_train.sh python /home/work/user-job-dir/code/train.py {python_file_parameter} #旧版训练
```

3.2.10 Pytorch1.0 引擎提示 “RuntimeError: std::exception”

问题现象

在使用pytorch1.0镜像的时候，必现如下报错
“RuntimeError: std::exception”

原因分析

出现该问题的可能原因如下：

pytorch1.0镜像中的libmkldnn软连接与原生torch的冲突，具体可参看这篇[文档](#)。

处理方法

1. 按照issues中的说明，应该是环境中的库冲突了，因此在启动脚本最开始之前，添加如下代码。

```
import os
os.system("rm /home/work/anaconda3/lib/libmkldnn.so")
os.system("rm /home/work/anaconda3/lib/libmkldnn.so.0")
```
2. 必现的问题，使用本地Pycharm远程连接Notebook调试。

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.2.11 MindSpore 日志提示 “retCode=0x91, [the model stream execute failed]”

问题现象

使用mindspore进行训练时，出现如下报错：
[ERROR] RUNTIME(3002)model execute error, retCode=0x91, [the model stream execute failed]

原因分析

出现该问题的可能原因如下：

数据读入的速度跟不上模型迭代的速度。

处理方法

1. 减少预处理shuffle操作。

```
dataset = dataset.shuffle(buffer_size=x)
```

2. 关闭数据预处理开关，可能会影响性能。
`NPURunConfig(enable_data_pre_proc=False)`

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.2.12 使用 moxing 适配 OBS 路径，pandas 读取文件报错

问题现象

使用moxing适配OBS路径，然后用较高版本的pandas读取OBS文件报出如下错误。

1. 'can't decode byte xxx in position xxx'
2. 'OSError:File isn't open for writing'

原因分析

出现该问题的可能原因如下：

moxing对高版本的pandas兼容性不够

处理方法

1. 在适配OBS路径后，读取文件模式从‘r’改成‘rb’，然后将mox.file.File的‘_write_check_passed’属性值改为‘True’，参考如下代码。

```
import pandas as pd
import moxing as mox

mox.file.shift('os', 'mox') # 将os的open操作替换为mox.file.File适配OBS路径的操作

param = {'encoding': 'utf-8'}
path = 'xxx.csv'
with open(path, 'rb') as f:
    f._write_check_passed = True
    df = pd.read_csv(f, **param)
```

2. 必现的问题，使用本地Pycharm远程连接Notebook调试。

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.2.13 日志提示 “Please upgrade numpy to >= xxx to use this pandas version”

问题现象

在安装其他包的时候，有依赖冲突，对numpy库有其他要求，但是发现numpy卸载不了。出现如下类似错误

```
your numpy version is 1.14.5.Please upgrade numpy to >= 1.15.4 to use this pandas version
```

原因分析

出现该问题的可能原因如下：

conda和pip包混装，有一些包卸载不掉。

处理方法

参考如下代码，三步走。

1. 先卸载numpy中可以卸载的组件
2. 删除你环境中site-packages路径下的numpy文件夹
3. 重新进行安装需要的版本

```
import os
os.system("pip uninstall -y numpy")
os.system('rm -rf /home/work/anaconda/lib/python3.6/site-packages/numpy/')
os.system("pip install numpy==1.15.4")
```

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.2.14 重装的包与镜像装 CUDA 版本不匹配

问题现象

在现有镜像基础上，重新装了引擎版本，或者编译了新的CUDA包，出现如下错误。

1. “RuntimeError: cuda runtime error (11) : invalid argument at /pytorch/aten/src/THC/THCCachingHostAllocator.cpp:278
2. “libcudart.so.9.0 cannot open shared object file no such file or directory”
3. “Make sure the device specification refers to a valid device, The requested device appears to be a GPU, but CUDA is not enabled”

原因分析

出现该问题的可能原因如下。

新安装的包与镜像中带的CUDA版本不匹配。

处理方法

1. 必现的问题，使用本地Pycharm远程连接Notebook调试安装。
 - a. 先远程登录到所选的镜像，使用“nvcc -V”查看目前镜像自带的CUDA版本。
 - b. 重装torch这些，需要注意选择与上一步版本相匹配的版本。

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.2.15 创建训练作业提示错误码 ModelArts.2763

问题现象

创建训练作业时，提示ModelArts.2763：选择的支持实例无效，请检查请求中信息的合法性。

原因分析

用户选择的训练规格资源和算法不匹配。

例如：算法支持的是GPU规格，创建训练作业时选择了ASCEND规格的资源类型。

处理方法

1. 查看算法代码中设置的训练资源规格。
2. 检查创建训练作业时所选的资源规格是否正确，重新创建训练作业选择正确的资源规格。

3.3 内存限制故障

3.3.1 下载或读取文件报错，提示超时、无剩余空间

问题现象

训练过程中拷贝数据/代码/模型时出现如下报错。

图 3-5 错误日志

```
INFO:root:RawImageIterAsync: Loading image list...
Traceback (most recent call last):
  File "test.py", line 142, in <module>
    val_path, args.batch_size)
  File "test.py", line 59, in get_data
    val_img_list=val_list)
  File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/data_factory.py", line 134, in get_data_iter
  File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/imageraw_dataset_async.py", line 486, in get_data_iter
  File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/imageraw_dataset_async.py", line 184, in __init__
  File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/imageraw_dataset_async.py", line 184, in <listcomp>
  File "/home/work/anaconda3/lib/python3.6/multiprocessing/context.py", line 129, in RawArray
    return RawArray(typecode or type, size or initializer)
  File "/home/work/anaconda3/lib/python3.6/multiprocessing/sharedctypes.py", line 60, in RawArray
    obj = new_value(type)
  File "/home/work/anaconda3/lib/python3.6/multiprocessing/sharedctypes.py", line 40, in _new_value
    wrapper = heap.BufferWrapper(size)
  File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 248, in __init__
    block = BufferWrapper(heap.malloc(size))
  File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 230, in malloc
    (arena, start, stop) = self._malloc(size)
  File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 128, in _malloc
    arena = Arena(length)
  File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 77, in __init__
    f.write(zeros)
OSError: [Errno 28] No space left on device
Exception ignored in: <bound method RawImageIterAsync.__del__ of <moxing.mxnet.data.imageraw_dataset_async.RawImageIterAsync object at 0x7fa18588f9b0>>
Traceback (most recent call last):
  File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/imageraw_dataset_async.py", line 222, in __del__
```

原因分析

出现该问题的可能原因如下。

- 磁盘空间不足。
- 分布式作业时，有些节点的docker base size配置未生效，容器内“/”根目录空间未达到50G，只有默认的10GB，导致作业训练失败。
- 实际存储空间足够，却依旧报错“No Space left on device”。

同一目录下创建较多文件时，为了加快文件检索速度，内核会创建一个索引表，短时间内创建较多文件时，会导致索引表达到上限，进而报错。

说明

触发条件和下面的因素有关：

- 文件名越长，文件数量的上限越小
- blocksize越小，文件数量的上限越小。（blocksize，系统默认 4096B。总共有三种大小：1024B、2048B、4096B）
- 创建文件越快，越容易触发（机制大概是：有一个缓存，这块大小和上面的1和2有关，目录下文件数量比较大时会启动，使用方式是边用边释放）

处理方法

1. 可以参照[日志提示"write line error"](#)文档进行修复。
2. 如果是分布式作业有的节点有错误，有的节点正常，建议提工单请求隔离有问题的节点。
3. 如果是触发了欧拉操作系统的限制，有如下建议措施。
 - 分目录处理，减少单个目录文件量。
 - 减慢创建文件的速度。
 - 关闭 ext4 文件系统的 dir_index 属性，具体可参考：<https://access.redhat.com/solutions/29894>，（可能会影响文件检索性能）。

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.3.2 拷贝数据至容器中空间不足

问题现象

ModelArts训练作业运行时，日志中遇到如下报错，导致数据无法拷贝至容器中。

```
OSError: [Errno 28] No space left on device
```

原因分析

数据下载至容器的位置空间不足。

处理方法

1. 请排查是否将数据下载至/cache目录下，GPU规格资源的每个节点会有一个“/cache”目录，空间大小为4TB。
2. 请排查是否使用的是GPU资源。如果使用的是CPU规格的资源，/cache与代码目录共用10G，会造成内存不足，请更改为使用GPU资源。
3. 请在代码中添加环境变量来解决。

```
import os  
os.system('export TMPDIR=/cache')
```

3.3.3 Tensorflow 多节点作业下载数据到/cache 显示 No space left

问题现象

创建训练作业，Tensorflow多节点作业下载数据到/cache显示：“No space left”。

原因分析

TensorFlow多节点任务会启动parameter server（简称ps）和worker两种角色，ps和worker会被调度到相同的机器上。由于训练数据对于ps没有用，因此在代码中ps相关的逻辑不需要下载训练数据。如果ps也下载数据到“/cache”实际下载的数据会翻倍。例如只下载了2.5TB的数据，程序就显示空间不够而失败，因为/cache只有4TB的可用空间。

处理方法

在使用Tensorflow多节点作业下载数据时，正确的下载逻辑如下：

```
import argparse  
parser = argparse.ArgumentParser()  
parser.add_argument("--job_name", type=str, default="")  
args = parser.parse_known_args()  
  
if args[0].job_name != "ps":  
    copy.....
```

3.3.4 日志文件的大小达到限制

问题现象

ModelArts训练作业在运行过程中报错，提示日志文件的大小已达到限制：

```
modelarts-pope: log length overflow(max:1073741824; already: 107341771; new:90), process will continue
running silently
```

原因分析

根据报错信息，可以判断是日志文件的大小已达到限制。出现该报错之后，日志不再增加，后台将继续运行。

处理方法

请您在启动文件中减少无用日志输出。

3.3.5 日志提示"write line error"

问题现象

在程序运行过程中，刷出大量错误日志"write line error"。并且问题是必现问题，每次运行到同一地方的时候，出现错误，具体见下面截图。

图 3-6 错误日志

[illegible]

原因分析

出现该问题的可能原因如下：

- 程序运行过程中，产生了core文件，core文件占满了"/"根目录空间。
- 本地数据、文件保存将"/cache"目录3.5T空间用满了。

说明

云上训练磁盘空间满一般是两个地方

1. 一个是“/”根目录，这个是docker中配置项“base size”，默认是10G，云上统一改成50G了。
2. 一个是‘/cache’目录满了，一般是3.5T存储空间满了，具体规格的空间大小见如下[文档](#)。

处理方法

1. 如果有core文件生成，可以在启动脚本最前面加上如下代码，来关闭core文件产生。

```
import os
os.system("ulimit -c 0")
```
2. 排查数据集大小，checkpoint保存文件大小，是否占满了磁盘空间。
3. 必现的问题，使用本地Pycharm远程连接Notebook调试。

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

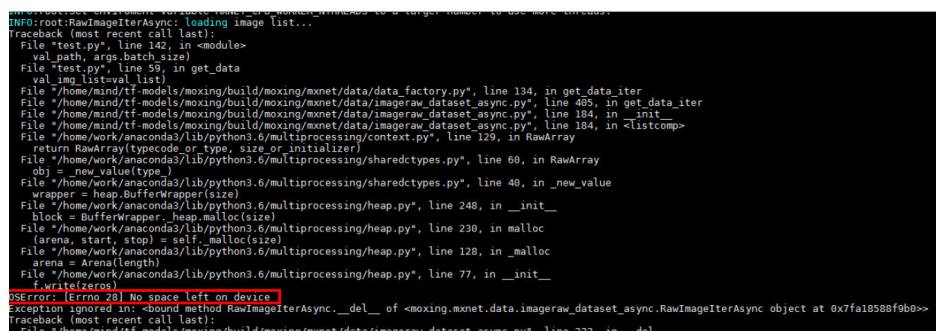
- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.3.6 日志提示 “No space left on device”

问题现象

训练过程中拷贝数据/代码/模型时出现如下报错。

图 3-7 错误日志



```
INFO:root:RawImageIterAsync: loading image list...
Traceback (most recent call last):
  File "test.py", line 142, in <module>
    val_path, args.batch_size)
  File "test.py", line 59, in get_data
    val_img_list=val_list)
  File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/data_factory.py", line 134, in get_data_iter
  File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/imageraw_dataset_async.py", line 485, in get_data_iter
  File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/imageraw_dataset_async.py", line 184, in __init__
  File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/imageraw_dataset_async.py", line 184, in <listcomp>
  File "/home/work/anaconda3/lib/python3.6/multiprocessing/context.py", line 129, in RawArray
    return RawArray(typecode_or_type, size_or_initializer)
  File "/home/work/anaconda3/lib/python3.6/multiprocessing/sharedctypes.py", line 68, in RawArray
    obj = new_value(type_)
  File "/home/work/anaconda3/lib/python3.6/multiprocessing/sharedctypes.py", line 48, in _new_value
    wrapper = heap.BufferWrapper(size)
  File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 248, in __init__
    block = BufferWrapper._heap.malloc(size)
  File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 230, in malloc
    (arena, start, stop) = self._malloc(size)
  File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 128, in _malloc
    arena = Arena(length)
  File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 77, in __init__
    f.write(zeros)
OSError: [Errno 28] No space left on device
Exception ignored in: <bound method RawImageIterAsync.__del__ of <moxing.mxnet.data.imageraw_dataset_async.RawImageIterAsync object at 0x7fa18588f9b0>>
Traceback (most recent call last):
  File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/imageraw_dataset_async.py", line 222, in __del__
```

原因分析

出现该问题的可能原因如下。

- 磁盘空间不足。
- 分布式作业时，有些节点的docker base size配置未生效，容器内“/”根目录空间未达到50G，只有默认的10GB，导致作业训练失败。
- 实际存储空间足够，却依旧报错“No Space left on device”。

同一目录下创建较多文件时，为了加快文件检索速度，内核会创建一个索引表，短时间内创建较多文件时，会导致索引表达到上限，进而报错。

📖 说明

触发条件和下面的因素有关：

- 文件名越长，文件数量的上限越小
- blocksize越小，文件数量的上限越小。（blocksize，系统默认 4096B。总共有三种大小：1024B、2048B、4096B）
- 创建文件越快，越容易触发（机制大概是：有一个缓存，这块大小和上面的1和2有关，目录下文件数量比较大时会启动，使用方式是边用边释放）

处理方法

1. 可以参照[日志提示"write line error"](#)文档进行修复。
2. 如果是分布式作业有的节点有错误，有的节点正常，建议提工单请求隔离有问题的节点。
3. 如果是触发了欧拉操作系统的限制，有如下建议措施。
 - 分目录处理，减少单个目录文件量。
 - 减慢创建文件的速度。
 - 关闭 ext4 文件系统的 dir_index 属性，具体可参考：<https://access.redhat.com/solutions/29894>，（可能会影响文件检索性能）。

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.3.7 OOM 导致训练作业失败

问题现象

因为OOM导致的训练作业失败，会有如下几种现象。

1. 错误码返回137，如下图所示。

图 3-8 错误日志

```
[Modelarts Service Log] Training end with return code: 137
[Modelarts Service Log]handle outputs of training job
[Modelarts Service Log][modelarts_logger] modelarts-pipe found
INFO:root:Using MoXing-v1.17.1-47ce498c
INFO:root:Using OBS-Python-SDK-3.1.2
[ModelArts Service Log][INFO][2021/04/28 18:52:27]: env MA_OUTPUTS is not found, skip the outputs handler
[Modelarts Service Log]Training completed.
```

2. 日志中有报错，含有“killed”相关字段，例如如下截图。

图 3-9 错误日志信息

```
Traceback (most recent call last):
  File "/home/ma-user/modelarts/user-job-dir/addernet-firstlast/main-imgnet.py", line 261, in <module>
    main()
  File "/home/ma-user/modelarts/user-job-dir/addernet-firstlast/main-imgnet.py", line 251, in main
    loss, acc = train_and_test(e, opt.alpha_start)
  File "/home/ma-user/modelarts/user-job-dir/addernet-firstlast/main-imgnet.py", line 243, in train_and_test
    acc = test(epoch, alpha_start, False)
  File "/home/ma-user/modelarts/user-job-dir/addernet-firstlast/main-imgnet.py", line 222, in test
    output = net(images, epoch, alpha_start)
  File "/home/ma-user/anaconda/lib/python3.6/site-packages/torch/nn/modules/module.py", line 541, in __call__
    result = self.forward(*input, **kwargs)
  File "/home/ma-user/anaconda/lib/python3.6/site-packages/torch/nn/parallel/data_parallel.py", line 152, in forward
    outputs = self.parallel_apply(replicas, inputs, kwargs)
  File "/home/ma-user/anaconda/lib/python3.6/site-packages/torch/nn/parallel/data_parallel.py", line 162, in parallel_apply
    return parallel_apply(replicas, inputs, kwargs, self.device_ids[:len(replicas)])
  File "/home/ma-user/anaconda/lib/python3.6/site-packages/torch/nn/parallel/parallel_apply.py", line 75, in parallel_apply
    thread.start()
  File "/home/ma-user/anaconda/lib/python3.6/threading.py", line 851, in start
    self._started.wait()
  File "/home/ma-user/anaconda/lib/python3.6/threading.py", line 551, in wait
    signaled = self._cond.wait(timeout)
  File "/home/ma-user/anaconda/lib/python3.6/threading.py", line 295, in wait
    waiter.acquire()
  File "/home/ma-user/anaconda/lib/python3.6/site-packages/torch/utils/data/_utils/signal_handling.py", line 66, in handler
    error if any worker fails()
RuntimeError: DataLoader worker (pid 38077) is killed by signal: Killed.
```

3. 日志中有报错 “RuntimeError: CUDA out of memory.”

图 3-10 错误日志信息

```
Traceback (most recent call last):
  File "memory_test.py", line 47, in <module>
    tmp_tensor = torch.empty(int(total_memory * 0.45), dtype=torch.int8, device='cuda')
RuntimeError: CUDA out of memory. Tried to allocate 14.29 GiB (GPU 0: 14.29 GiB total capacity; 0 bytes
already allocated; 14.29 GiB free; 0 bytes reserved in total by PyTorch)
```

4. Tensorflow引擎日志中出现 “Dst tensor is not initialized”

原因分析

按照之前支撑的经验，出现该问题的可能原因如下。：

- 绝大部分都是确实是显存不够用
- 还有较少数原因是节点故障，跑到特定节点必现OOM，其他节点正常

处理方法

1. 如果是正常的OOM，就需要修改一些超参，释放一些不需要的tensor。
 - a. 修改网络参数，比如batch_size、hide_layer、cell_nums等。
 - b. 释放一些不需要的tensor，使用过的，如下：

```
del tmp_tensor
torch.cuda.empty_cache()
```
2. 必现的问题，使用本地Pycharm远程连接Notebook调试超参。
3. 如果还存在问题，可能需要提工单进行定位，甚至需要隔离节点修复。

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)

- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.4 外网访问限制

3.4.1 日志提示 “ Network is unreachable ”

问题现象

在使用pytorch时，将torchvision.models中的pretrained置为了True，日志中出现如下报错

```
'OSError: [Errno 101] Network is unreachable'
```

原因分析

出现该问题的可能原因如下：

因为安全性问题，ModelArts内部训练机器不能访问外网。

处理方法

- 将pretrained改成false,提前下载好预训练模型，加载下载好的预训练模型位置即可，可参考如下代码。

```
import torch
import torchvision.models as models

model1 = models.resnet34(pretrained=False, progress=True)
checkpoint = '/xxx/resnet34-333f7ec4.pth'
state_dict = torch.load(checkpoint)
model1.load_state_dict(state_dict)
```

- 必现的问题，使用本地Pycharm远程连接Notebook调试。

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.4.2 运行训练作业时提示 URL 连接超时

问题现象

训练作业在运行时提示URL连接超时，具体报错如下：

```
urllib.error.URLError:<urlopen error [Errno 110] Connection timed out>
```

原因分析

由于安全性问题在ModelArts上不能联网下载。

处理方法

如果在运行训练作业时提示连接超时，请您将需要联网下载的数据提前下载至本地，并上传至OBS中。

3.5 权限问题

3.5.1 日志提示“reason:Forbidden”

问题现象

训练作业访问OBS时，出现如下报错

图 3-11 报错信息

```
ERROR:root:Failed to call:
  func=<bound method ObsClient.getObjectMetadata of <moxing.framework.file.src.obs.client.ObsClient object at 0x7fdb4ad06d0>>
  args=('bucket-cv-competition-bj4', 'fangjiemin/output/')
  kwargs={}
ERROR:root:
stat:403
errorCode:None
errorMessage:None
reason:Forbidden
request-id:00000179D5ACCAC445CAA1A71019C9D0
retry:0
```

原因分析

出现该问题的可能原因如下。具体错误码可以参考 [OBS官方文档](#)：

- 镜像中OBS-SDK版本过低，需要升级OBS-SDK版本
- 排查子账户是否拥有该桶的读写权限
- 环境中AK/SK设置有误

处理方法

一般如果账户有该桶的访问权限，那就需要提工单找OBS的同事来一起定位，更新环境变量了。

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.5.2 日志提示"Permission denied"

问题现象

训练作业访问挂载的EFS，或者是执行.sh启动脚本时，出现如下错误。

图 3-12 错误日志

```
Traceback (most recent call last):  
  File "codes/prepare_listdir.py", line 11, in <module>  
    rec_file_list = os.listdir(recurrent_path)  
OSError: [Errno 13] Permission denied: '/data/recurrent'
```

原因分析

出现该问题的可能原因如下。

- 上传数据时文件所属与文件权限未修改，导致训练作业以work用户组访问时没有权限了。
- 在代码目录中的.sh拷贝到容器之后，需要添加“x”可执行权限

处理方法

1. 对挂载盘的数据加权限，可以改为与训练容器内相同的用户组（1101），假如/nas盘是挂载路径，执行如下代码
chown -R 1101: 1101 /nas
或者
chmod 777 -R /nas
2. 如果是自定义镜像中拉取的.sh脚本没有执行权限，可以在自定义脚本启动前执行"chmod +x xxx.sh"添加可执行权限。

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.6 GPU 相关问题

3.6.1 日志提示"No CUDA-capable device is detected"

问题现象

在程序运行过程中，出现如下类似错误。

1. 'failed call to cuInit: CUDA_ERROR_NO_DEVICE: no CUDA-capable device is detected'
2. 'No CUDA-capable device is detected although requirements are installed'

原因分析

出现该问题的可能原因如下：

- 用户/训练系统，将CUDA_VISIBLE_DEVICES传错了，检查一下CUDA_VISIBLE_DEVICES变量是否正常。

- 用户选择了1/2/4卡这些规格的作业，然后设置了CUDA_VISIBLE_DEVICES= ‘1’ 这种类似固定的卡ID号，与实际选择的卡ID不匹配。

处理方法

1. 尽量代码里不要去修改CUDA_VISIBLE_DEVICES变量，用系统默认里面自带的。
2. 如果必须指定卡ID，需要注意一下1/2/4规格下，指定的卡ID与实际分配的卡ID不匹配的情况。
3. 如果上述方法还出现了错误，可以去notebook里面调试打印CUDA_VISIBLE_DEVICES变量，或者用以下代码测试一下,查看结果是否返回的是True。

```
import torch
torch.cuda.is_available()
```

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.6.2 日志提示 “RuntimeError: connect() timed out”

问题现象

使用pytorch进行分布式训练时，出现如下错误。

图 3-13 错误日志

```
INFO - 03/23/21 17:20:50 - 0:00:04 - Building data done with 1331166 images loaded.
Traceback (most recent call last):
  File "swav-master/main_swav.py", line 500, in <module>
    main()
  File "swav-master/main_swav.py", line 191, in main
    mp.spawn(main_worker, nprocs=args.ngpu, args=())
  File "/home/work/anaconda/lib/python3.6/site-packages/torch/multiprocessing/spawn.py", line 171, in spawn
    while not spawn_context.join():
  File "/home/work/anaconda/lib/python3.6/site-packages/torch/multiprocessing/spawn.py", line 118, in join
    raise Exception(msg)
Exception:

-- Process 2 terminated with the following error:
Traceback (most recent call last):
  File "/home/work/anaconda/lib/python3.6/site-packages/torch/multiprocessing/spawn.py", line 19, in _wrap
    fn(i, *args)
  File "/cache/user-job-dir/swav-master/main_swav.py", line 231, in main_worker
    rank=args.rank)
  File "/home/work/anaconda/lib/python3.6/site-packages/torch/distributed/distributed_c10d.py", line 397, in init_process_group
    store, rank, world_size = next(rendezvous_iterator)
  File "/home/work/anaconda/lib/python3.6/site-packages/torch/distributed/rendezvous.py", line 168, in _env_rendezvous_handler
    store = TCPStore(master_addr, master_port, world_size, start_daemon)
RuntimeError: connect() timed out.
```

原因分析

出现该问题的可能原因如下。：

如果在此之前是有进行数据拷贝的，每个节点拷贝的速度不是同一个时间完成的，然后有的节点没有拷贝完，其他节点进行torch.distributed.init_process_group（）导致超时。

处理方法

1. 如果是多个节点拷贝不同步，并且没有barrier的话导致的超时，可以在拷贝数据之前，先进行torch.distributed.init_process_group（），然后再根据local_rank()==0去拷贝数据，之后再调用torch.distributed.barrier()等待所有rank完成拷贝。可以参考如下代码

```
import mox as mox
import torch

torch.distributed.init_process_group()
if local_rank == 0:
    mox.file.copy_parallel(src,dst)

torch.distributed.barrier()
```

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.6.3 日志提示“cuda runtime error (10) : invalid device ordinal at xxx”

问题现象

训练作业失败，日志报出如下错误

图 3-14 错误日志

```
main()
File "train.py", line 278, in main
  torch.cuda.set_device(args.local_rank)
File "/home/work/anaconda/lib/python3.6/site-packages/torch/cuda/_init_.py", line 300, in set_device
  torch.C. cuda_setDevice(device)
RuntimeError: cuda runtime error (10) : invalid device ordinal at /pytorch/torch/csrc/cuda/Module.cpp:37
```

原因分析

可以从以下角度排查：

- 请检查CUDA_VISIBLE_DEVICES设置的值是否与作业规格匹配。例如您选择4卡规格的作业，实际可用的卡ID为0、1、2、3，但是您在进行cuda相关的运算时，例如"tensor.to(device="cuda:7")"，将张量搬到了7号gpu卡上，超过了实际可用的ID号。
- 如果cuda相关运算设置的卡ID号在所选规格范围内，但是依旧出现了上述报错。可能是该资源节点中存在GPU卡损坏的情况，导致实际能检测到的卡少于所选规格。

处理方法

1. 建议直接根据系统分卡情况下传进去的CUDA_VISIBLE_DEVICES去设置，不用手动指定默认的。

2. 如果发现资源节点中存在GPU卡损坏，请联系技术支持处理。

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.6.4 日志提示 “RuntimeError: Cannot re-initialize CUDA in forked subprocess”

问题现象

在使用pytorch启动多进程的时候，出现如下报错
RuntimeError: Cannot re-initialize CUDA in forked subprocess

原因分析

出现该问题的可能原因如下。

multiprocessing启动方式有误

处理方法

可以参考[官方文档](#)，如下

```
"""run.py: """
#!/usr/bin/env python
import os
import torch
import torch.distributed as dist
import torch.multiprocessing as mp

def run(rank, size):
    """ Distributed function to be implemented later. """
    pass

def init_process(rank, size, fn, backend='gloo'):
    """ Initialize the distributed environment. """
    os.environ['MASTER_ADDR'] = '127.0.0.1'
    os.environ['MASTER_PORT'] = '29500'
    dist.init_process_group(backend, rank=rank, world_size=size)
    fn(rank, size)

if __name__ == "__main__":
    size = 2
    processes = []
    mp.set_start_method("spawn")
    for rank in range(size):
        p = mp.Process(target=init_process, args=(rank, size, run))
        p.start()
        processes.append(p)

    for p in processes:
        p.join()
```

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.6.5 训练作业找不到 GPU

问题现象

训练作业运行出现如下报错：

```
failed call to cuInit: CUDA_ERROR_NO_DEVICE: no CUDA-capable device is detected
```

原因分析

根据错误信息判断，报错原因为训练作业运行程序读取不到GPU。

处理方法

根据报错提示，请您排查代码，是否已添加以下配置，设置该程序可见的GPU：

```
os.environ['CUDA_VISIBLE_DEVICES'] = '0,1,2,3,4,5,6,7'
```

其中，0为服务器的GPU编号，可以为0, 1, 2, 3等，表明对程序可见的GPU编号。若未进行添加配置则该编号对应的GPU不可用。

3.7 业务代码问题

3.7.1 日志提示“pandas.errors.ParserError: Error tokenizing data. C error: Expected .* fields”

问题现象

使用pandas读取csv数据表时，日志报出如下错误导致训练作业失败
pandas.errors.ParserError: Error tokenizing data. C error: Expected 4 field

原因分析

csv中文件的每一行的列数不相等。

处理方法

可以使用以下方法处理：

- 校验csv文件，将多出字段的行删除。
- 在代码中忽略错误行，参考如下：

```
import pandas as pd
pd.read_csv(filePath,error_bad_lines=False)
```

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.7.2 日志提示 “max_pool2d_with_indices_out_cuda_frame failed with error code 0”

问题现象

pytorch1.3镜像中，去升级了1.4的版本，导致之前在1.3跑通的代码报错如下
“RuntimeError:max_pool2d_with_indices_out_cuda_frame failed with error code 0”

原因分析

出现该问题的可能原因如下：

pytorch1.4引擎与之前pytorch1.3版本兼容性问题。

处理方法

1. 在images之后添加contiguous。

```
images = images.cuda()  
pred = model(images.permute(0, 3, 1, 2).contiguous())
```
2. 将版本回退至pytorch1.3
3. 必现的问题，使用本地Pycharm远程连接Notebook调试

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.7.3 训练作业失败，返回错误码 139

问题现象

训练作业运行失败，返回错误码139，如下截图。

图 3-15 错误码信息

```
[Modelarts Service Log] Training end with return code: 139  
INFO:root:Using MoXing-v1.17.2-c806a92f  
INFO:root:Using OBS-Python-SDK-3.1.2  
[ModelArts Service Log]2020-09-27 20:57:19,264 - modelarts  
[ModelArts Service Log]2020-09-27 20:57:19,352 - modelarts  
[Modelarts Service Log]Training completed.
```

原因分析

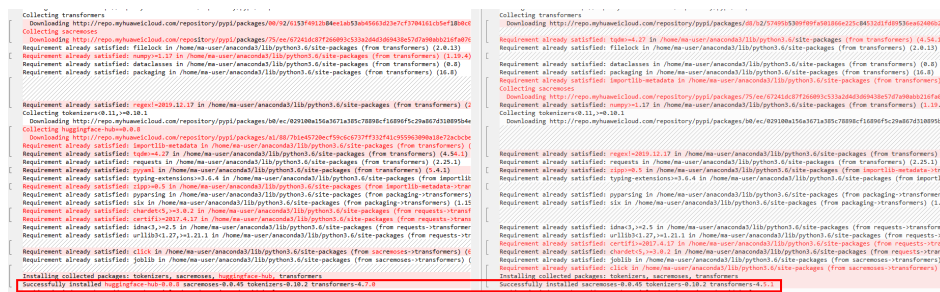
出现该问题的可能原因如下

- pip源中的pip包更新了，之前能跑通的代码，在包更新之后产生了不兼容的情况，例如 transformers包，导致import的时候出现了错误。
- 用户代码问题，出现了内存越界、踩内存的情况。
- 未知系统问题导致，建议先尝试重建作业，重建后仍然失败，建议提工单定位。

处理方法

1. 如果存在之前能跑通，什么都没修改，过了一阵跑不通的情况，先去排查跑通和跑不通的日志是否存在pip源更新了依赖包，如下图，安装之前跑通的老版本即可。

图 3-16 PIP 安装对比图



2. 推荐您使用本地Pycharm远程连接Notebook调试。
3. 如果上述情况都解决不了，请联系技术支持工程师。

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.7.4 训练作业失败，如何使用云上环境调试训练代码？

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.7.5 日志提示 “'(slice(0, 13184, None), slice(None, None, None))' is an invalid key”

问题现象

训练过程中出现如下报错


```
TypeError: '(slice(0, 13184, None), slice(None, None, None))' is an invalid key
```

原因分析

出现该问题的可能原因如下：

切分数据时，选择的数据不对。

处理方法

尝试如下代码

```
X = dataset.iloc[:, :-1].values
```

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.7.6 日志报错 “DataFrame.dtypes for data must be int, float or bool”

问题现象

训练过程中出现如下报错

```
DataFrame.dtypes for data must be int, float or bool
```

原因分析

出现该问题的可能原因如下：

训练数据中出现了非int,float,bool类型数据。

处理方法

可参考如下代码，将错误列进行转换

```
from sklearn import preprocessing  
lbl = preprocessing.LabelEncoder()  
train_x['acc_id1'] = lbl.fit_transform(train_x['acc_id1'].astype(str))
```

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.7.7 日志提示 “CUDNN_STATUS_NOT_SUPPORTED.”

问题现象

在pytorch训练时，出现如下报错

```
RuntimeError: cuDNN error: CUDNN_STATUS_NOT_SUPPORTED. This error may appear if you passed in a non-contiguous input.
```

原因分析

出现该问题的可能原因如下：

数据输入不连续，cuDNN不支持的类型。

处理方法

1. 禁用cuDNN，在训练前加入如下代码
`torch.backends.cudnn.enabled = False`
2. 将输入数据转换成contiguous
`images = images.cuda()`
`images = images.permute(0, 3, 1, 2).contiguous()`

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.7.8 日志提示 “Out of bounds nanosecond timestamp”

问题现象

在使用pandas.to_datetime转换时间时，出现如下报错

```
pandas._libs.tslibs.np_datetime.OutOfBoundsDatetime: Out of bounds nanosecond timestamp: 1-01-02 13:20:00
```

原因分析

出现该问题的可能原因如下：

时间值越界，请参考[官方文档](#)

处理方法

校验时间数据，pandas以纳秒表示时间戳。

最小时间：1677-09-22 00:12:43.145225，

最大时间：2262-04-11 23:47:16.854775807，需注意上下界限

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.7.9 日志提示“Unexpected keyword argument passed to optimizer”

问题现象

在使用keras时，升级版本 $\geq 2.3.0$ 之后，之前跑通的代码出现如下报错
TypeError: Unexpected keyword argument passed to optimizer: learning_rate

原因分析

出现该问题的可能原因如下：

请参考[官方文档](#)：参数在keras中，参数重命名了。

图 3-17 API 修改手册

- Rename `lr` to `learning_rate` for all optimizers.

处理方法

将learning_rate参数更改为lr。

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.7.10 日志提示“no socket interface found”

问题现象

在pytorch镜像运行分布式作业时，设置NCCL日志级别，代码如下。

```
import os
os.environ["NCCL_DEBUG"] = "INFO"
```

会出现如下错误

图 3-18 错误日志

```
job0879f61e-job-base-pda-2-0:712:712 [0] bootstrap.cc:37 NCCL WARN Bootstrap : no socket interface found
job0879f61e-job-base-pda-2-0:712:712 [0] NCCL INFO init.cc:128 -> 3
job0879f61e-job-base-pda-2-0:712:712 [0] NCCL INFO bootstrap.cc:76 -> 3
job0879f61e-job-base-pda-2-0:712:712 [0] NCCL INFO bootstrap.cc:245 -> 3
job0879f61e-job-base-pda-2-0:712:712 [0] NCCL INFO bootstrap.cc:266 -> 3
Traceback (most recent call last):
  File "train_net.py", line 1923, in <module>
    main_worker(args)
  File "train_net.py", line 355, in main_worker
    network = torch.nn.parallel.DistributedDataParallel(network, device_ids=device_ids, find_unused_parameters=True)
  File "/home/work/anaconda/lib/python3.6/site-packages/torch/nn/parallel/distributed.py", line 298, in __init__
    self.broadcast_bucket_size)
  File "/home/work/anaconda/lib/python3.6/site-packages/torch/nn/parallel/distributed.py", line 480, in _distributed_broadcast_coalesced
    dist._broadcast_coalesced(self.process_group, tensors, buffer_size)
RuntimeError: NCCL error in: /pytorch/torch/lib/c10d/ProcessGroupNCCL.cpp:374, internal error
Traceback (most recent call last):
```

原因分析

出现该问题的可能原因如下：

用户修改了云上已经设置过的NCCL_SOCKET_IFNAME环境变量，将其网卡指向了云上机器不存在的网卡，导致出现了如下错误。云上默认设置了如下配置。

```
import os
os.environ[NCCL_SOCKET_IFNAME]=ib0,bond0,eth0
```

处理方法

1. 去除用户代码侧NCCL_SOCKET_IFNAME相关的修改代码，使用云上系统自带的环境变量。
2. 必现的问题，使用本地Pycharm远程连接Notebook调试

建议与总结

在创建训练作业前，推荐您先使用ModelArts开发环境调试训练代码，避免代码迁移过程中的错误。

- 直接使用线上notebook环境调试请参考[使用JupyterLab开发模型](#)
- 配置本地IDE（Pycharm或者VsCode）联接云上环境调试请参考[使用本地IDE开发模型](#)

3.7.11 分布式 Tensorflow 无法使用 “tf.variable”

问题现象

多机或多卡使用“tf.variable”会造成以下错误：WARNING:tensorflow:Gradient is None for variable:v0/tower_0/UNET_v7/sub_pixel/Variable:0. Make sure this variable is used in loss computation.

图 3-19 分布式 Tensorflow 无法使用

```
WARNING:tensorflow:Gradient is None for variable: v0/tower_0/UNET_v7/sub_pixel/Variable:0. Make sure this variable is used in loss computation.
WARNING:tensorflow:Gradient is None for variable: v0/tower_0/UNET_v7/sub_pixel/Variable:1:0. Make sure this variable is used in loss computation.
WARNING:tensorflow:Gradient is None for variable: v0_1/tower_1/UNET_v7/sub_pixel/Variable:0. Make sure this variable is used in loss computation.
WARNING:tensorflow:Gradient is None for variable: v0_1/tower_1/UNET_v7/sub_pixel/Variable:1:0. Make sure this variable is used in loss computation.
WARNING:tensorflow:Gradient is None for variable: v0_2/tower_2/UNET_v7/sub_pixel/Variable:0. Make sure this variable is used in loss computation.
WARNING:tensorflow:Gradient is None for variable: v0_2/tower_2/UNET_v7/sub_pixel/Variable:1:0. Make sure this variable is used in loss computation.
WARNING:tensorflow:Gradient is None for variable: v0_3/tower_3/UNET_v7/sub_pixel/Variable:0. Make sure this variable is used in loss computation.
WARNING:tensorflow:Gradient is None for variable: v0_3/tower_3/UNET_v7/sub_pixel/Variable:1:0. Make sure this variable is used in loss computation.
WARNING:tensorflow:Gradient is None for variable: v0_4/tower_4/UNET_v7/sub_pixel/Variable:0. Make sure this variable is used in loss computation.
WARNING:tensorflow:Gradient is None for variable: v0_4/tower_4/UNET_v7/sub_pixel/Variable:1:0. Make sure this variable is used in loss computation.
WARNING:tensorflow:Gradient is None for variable: v0_5/tower_5/UNET_v7/sub_pixel/Variable:0. Make sure this variable is used in loss computation.
WARNING:tensorflow:Gradient is None for variable: v0_5/tower_5/UNET_v7/sub_pixel/Variable:1:0. Make sure this variable is used in loss computation.
WARNING:tensorflow:Gradient is None for variable: v0_6/tower_6/UNET_v7/sub_pixel/Variable:0. Make sure this variable is used in loss computation.
WARNING:tensorflow:Gradient is None for variable: v0_6/tower_6/UNET_v7/sub_pixel/Variable:1:0. Make sure this variable is used in loss computation.
```

原因分析

分布式Tensorflow不能使用“tf.variable”要使用“tf.get_variable”。

处理方法

请您将“启动文件”中的“tf.variable”替换为“tf.get_variable”。

3.7.12 MXNet 创建 kvstore 时程序被阻塞，无报错

问题现象

使用`kv_store = mxnet.kv.create('dist_async')`方式创建“kvstore”时程序被阻塞。如，执行如下代码，如果无法输出“end”，表明程序阻塞。

```
print('start')
kv_store = mxnet.kv.create('dist_async')
print('end')
```

原因分析

worker阻塞的原因可能是连不上server。

处理方法

将如下代码放在“启动文件”里“import mxnet”之前可以看到节点间相互通信状态，同时ps能够重新发送。

```
import os
os.environ['PS_VERBOSE'] = '2'
os.environ['PS_RESEND'] = '1'
```

其中，“os.environ['PS_VERBOSE'] = '2'”为打印所有的通信信息。
“os.environ['PS_RESEND'] = '1'”为在“PS_RESEND_TIMEOUT”毫秒后没有收到ACK消息，Van实例会重发消息。

3.7.13 日志出现 ECC 错误，导致训练作业失败

问题现象

训练作业日志运行出现如下报错：RuntimeError: CUDA error: uncorrectable ECC error encountered

原因分析

由于ECC错误，导致作业运行失败，该作业节点会被自动隔离，需要重启作业。

处理方法

如果出现此报错，请您重新创建训练作业。

3.7.14 超过最大递归深度导致训练作业失败

问题现象

ModelArts训练作业报错：

```
RuntimeError: maximum recursion depth exceeded in __instancecheck__
```

原因分析

递归深度超过了Python默认的递归深度，导致训练失败。

处理方法

如果超过最大递归深度，建议您在启动文件中增大递归调用深度，具体操作如下：

```
import sys
sys.setrecursionlimit(1000000)
```

3.7.15 使用预置算法训练时，训练失败，报“bndbox”错误

问题现象

使用预置算法创建训练作业，训练失败，日志中出现如下报错。

```
KeyError: 'bndbox'
```

原因分析

用于训练的数据集中，使用了“非矩形框”标注。而预置使用算法不支持“非矩形框”标注的数据集。

处理方法

此问题有两种解决方法：

- 方法1：使用常用框架自行编码开发模型，支持“多边形”标注的数据集。
- 方法2：修改数据集，使用矩形标注。然后再启动训练作业。

3.7.16 训练作业状态显示“审核作业初始化”

问题现象

当创建训练作业的“算法来源”选择“自定义”镜像创建训练作业时，训练作业状态显示审核作业初始化。

原因分析

自定义镜像首次运行时，需要先审核镜像，通过审核之后才可创建作业，即当前状态为审核作业初始化。

3.7.17 训练作业进程异常退出

问题现象

训练失败，日志中出现如下报错：

```
[Modelarts Service Log]Training end with return code: 137
```

原因分析

日志显示训练进程的退出码为137。训练进程表示用户的代码启动后的进程，所以这里的退出码是用户的训练作业代码返回的。常见的错误码还包括247、139等。

- 退出码137或者247
可能是内存溢出造成的。请减少数据量、减少batch_size，优化代码，合理聚合、复制数据。

说明

数据文件大小不等于内存占用大小，需仔细评估内存使用情况。

- 退出码139
请排查安装包的版本，可能存在包冲突的问题。

排查办法

根据错误信息判断，报错原因来源于用户代码。

您可以通过以下两种方式排查：

- 线上环境调试代码（仅适用于非分布式代码）
 - a. 在开发环境（notebook）申请相同规格的开发环境实例。
 - b. 在notebook调试用户代码，并找出问题的代码段。
 - c. 通过关键代码段 + 退出码尝试去搜索引擎寻找解决办法。
- 通过训练日志排查问题
 - a. 通过日志判断出问题的代码范围。
 - b. 修改代码，在问题代码段添加打印，输出更详细的日志信息。
 - c. 再次运行作业，判断出问题的代码段。

3.7.18 训练作业进程被 kill

问题现象

用户进程被Kill表示用户进程因外部因素被Kill或者中断，表现为日志中断。

原因分析

- CPU软锁
在解压大量文件可能会出现此情况并造成节点重启。可以适当在解压大量文件时，加入sleep。比如每解压1w个文件，就停止1s。
- 存储限制

根据规格情况合理使用数据盘，数据盘大小请参考[训练环境中不同规格资源大小](#)。

- CPU过载
减少线程数。

排查办法

根据错误信息判断，报错原因来源于用户代码。

您可以通过以下两种方式排查：

- 线上环境调试代码（仅适用于非分布式代码）
 - a. 在开发环境（notebook）申请相同规格的开发环境实例。
 - b. 在notebook调试用户代码，并找出问题的代码段。
 - c. 通过关键代码段 + 退出码尝试去搜索引擎寻找解决办法。
- 通过训练日志排查问题
 - a. 通过日志判断出问题的代码范围。
 - b. 修改代码，在问题代码段添加打印，输出更详细的日志信息。
 - c. 再次运行作业，判断出问题的代码段。

4 模型管理

4.1 Caffe 模型转换不成功

问题现象

用户提交的Caffe模型出现转换不成功。

转换失败后，您可以在模型转换任务详情页面获得相应日志。如果出现如下类似日志，表示算子不支持导致转换失败。

'Error your model contain ddk not supoort operators, please refer to [\[指向faq连接\]](#)'

原因分析

由于海思DDK当前只支持部分算子，如果用户定义的模型包含不支持的算子，则会出现转换失败。

解决方案

1. 在转换模型任务的“模型输出目录”中存在算子评估结果文件“eval_report.json”，从对应的OBS目录获取该文件，并使用json格式化工具将评估结果文件进行格式化。
文件格式化之后，您可以在文件中查看哪个算子不支持，建议可以使用哪个算子做替换，示例如下：

```
{
  "fail": 1,
  "name": "SSD_VGG_640x640",
  "op": [{
    "name": "conv1_1",
    "result": "success",
    "type": "Convolution"
  }, {
    "cause": [{
      "code": 8,
      "message": "The type is ambiguous. Please choose from the following candidate list [FSRDetectionOutput, SSDDetectionOutput, YoloDetectionOutput].",
    }],
    "name": "detection_out",
    "result": "failed",
    "type": "DetectionOutput"
  }],
}
```



```
"pass": 86,  
"result": "failed",  
"total": 87  
}
```

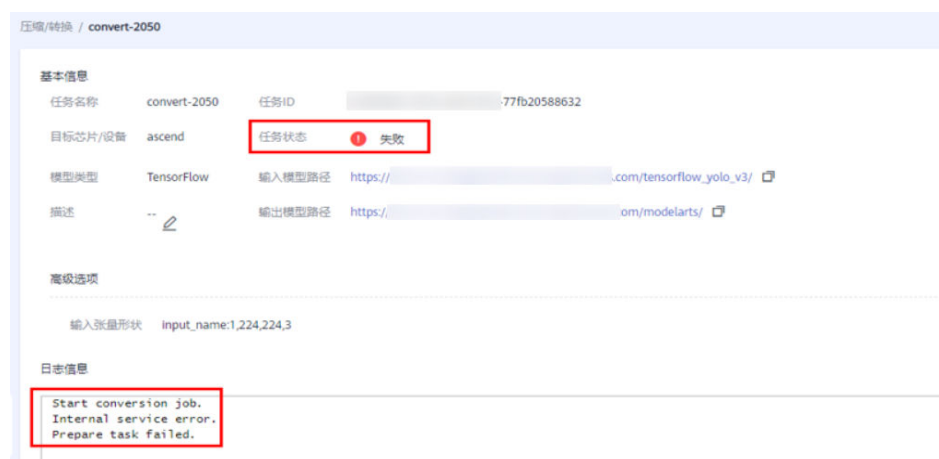
2. 如果需要继续转换模型，您需要完成算子映射。
在“模型输入目录”下添加算子映射文件，文件必须以“opmap.txt”命名，在这个文件里面写入算子映射，格式为“不支持算子:替换算子”，如下所示：
DetectionOutput:SSDDetectionOutput
3. 在ModelArts管理控制台，再次提交模型转换任务。

4.2 TensorFlow 模型转换失败

问题现象

使用TensorFlow框架编写的模型，在运行模型转换任务时，任务失败，且日志信息如下所示。

图 4-1 模型转换任务失败



解决方法

针对模型转换失败的任务，请根据如下排除指导进行排查。

1. 检查当前帐号是否具备转换任务中“转换输入目录”和“转换输出目录”的权限。
针对模型转换任务中，选择的“转换输入目录”和“转换输出目录”，需确保当前帐号具备此OBS桶的权限。

图 4-2 转换任务中的 OBS 目录配置



检查方法：

- 进入OBS管理控制台，在OBS桶列表中找到对应的OBS桶，单击OBS桶名称进入详情页。
- 选择“访问权限控制>桶ACLs”，检查当前帐号是否具备“读取权限”和“写入权限”。
 - 有，执行下一步。
 - 没有，建议更换OBS桶，或者联系OBS桶的拥有者为您赋予权限。

图 4-3 查看权限



- 检查用于转换的模型，是否包含不支持转换的算子。

当前，模型转换功能仅支持Caffe算子清单和Tensorflow算子清单中的算子，且需要满足算子的限制条件。请根据昇腾社区中罗列的算子清单检查您的模型，确保模型中使用的算子，在支持列表范围内。

排除上述原因后，请重新运行模型转换任务。

4.3 自定义镜像模型部署为在线服务时出现异常

问题现象

在部署在线服务时，部署失败。进入在线服务详情页面，“事件”页签，提示“failed to pull image, retry later”，同时在“日志”页签中，无任何信息。

图 4-4 部署在线服务异常



解决方法

出现此问题现象，通常是因为您部署的模型过大导致的。解决方法如下：

- 精简模型，重新导入模型和部署上线。

- 购买专属资源池，在部署上线为在线服务时，使用专属资源池进行部署。

4.4 部署的在线服务状态为告警

问题现象

在部署在线服务时，状态显示为“告警”。

解决方法

使用状态为告警的服务进行预测，可能存在预测失败的风险，请从以下3个角度进行排查，并重新部署。

1. 后台预测请求过多。
如果您使用API接口进行预测，请检查是否预测请求过多。大量的预测请求会导致部署的在线服务进入告警状态。
2. 业务内存不正常。
请检查推理代码是否存在内存溢出或者内存泄漏的问题。
3. 模型运行异常。
请检查您的模型是否能正常运行。例如模型依赖的资源是否故障，需要排查推理日志。

5 MoXing

5.1 使用 MoXing 复制数据报错

问题现象

1. 调用`moxing.file.copy_parallel()`将文件从开发环境拷贝到桶里，但是桶内没有出现目标文件。
2. 使用MoXing复制数据不成功，出现报错。如：
 - ModelArts开发环境使用MoXing复制OBS数据报错：keyError: 'request-id'
 - ModelArts使用MoXing拷贝报错：No files to copy
 - socket.gaierror: [Errno -2] Name or service not known
 - ERROR:root:Failed to call:
func=<bound method ObsClient.getObject of <obs.client.ObsClient object at 0x7fd705939710>>
args=('bucket', 'data/TFRecord/HY_all_inside/
no_adjust_light_3/09_06_6x128x128_0000000212.tfrecord')
3. 使用MoXing复制数据报错，提示超时。如：
 - 报错：TimeoutError: [Errno 110] Connection timed out
 - WARNING:root:Retry=9,Wait=0.1, Timestamp = 1567152567.5327423

原因分析

当使用MoXing复制数据不成功，可能原因如下：

- 源文件不存在。
- 复制的两个OBS路径不正确或者不在同一区域。
- 训练作业空间不足。

处理方法

按照报错提示，需要排查以下几个问题：

1. 检查`moxing.file.copy_parallel()`的第一个参数中是否有文件，否则会出现报错：No files to copy

- 文件存在，请执行2。
 - 文件不存在，请忽略报错继续执行后续操作。
2. 检查复制的OBS的路径是否与开发环境或训练作业在同一个区域。
进入ModelArts管理控制台，查看其所在区域。然后再进入OBS管理控制台，查看您使用的OBS桶所在的区域。查看是否在同一区域。
 - 是，请执行3。
 - 否，请在ModelArts同一区域的OBS中新建桶和文件夹，并将所需的数据上传至此OBS桶中。
 3. 检查OBS的路径是否正确，是否写为了“obs://xxx”。可使用如下方式判断OBS路径是否存在。
mox.file.exists('obs://bucket_name/sub_dir_0/sub_dir_1')
 - 路径存在，请执行4。
 - 路径不存在，请在更换为一个可用的OBS路径。
 4. 检查使用的资源是否为CPU，CPU的“/cache”与代码目录共用10G，可能是空间不足导致，可在代码中使用如下命令查看磁盘大小。
os.system('df -hT')
 - 磁盘空间满足，请执行5。
 - 磁盘空间不足，请您使用GPU资源。
 5. 如果是在Notebook使用MoXing复制数据不成功，可以在Terminal界面中使用**df -hT**命令查看空间大小，排查是否因空间不足导致，可在创建Notebook时使用EVS挂载。

如果代码写作正确，仍然无法解决该问题，请提交工单，由专业工程师为您分析并解决问题。

5.2 如何关闭 Mox 的 warmup

问题现象

训练作业mox的Tensorflow版本在运行的时候，会先执行“50steps”4次，然后才会开始正式运行。

warmup即先用一个小的学习率训练几个epoch（warmup），由于网络的参数是随机初始化的，如果一开始就采用较大的学习率会出现数值不稳定的问题，这是使用warmup的原因。等到训练过程基本稳定之后就可以使用原先设定的初始学习率进行训练。

原因分析

Tensorflow分布式有多种执行模式，mox会通过4次执行50 step记录执行时间，选择执行时间最少的模型。

处理方法

创建训练作业时，在“运行参数”中增加参数“variable_update=parameter_server”来关闭Mox的warmup。

5.3 Pytorch Mox 日志反复输出

问题现象

ModelArts训练作业算法来源选用常用框架的Pytorch引擎，在训练作业运行时Pytorch Mox日志会每个epoch都打印Mox版本，具体日志如下：

```
INFO:root:Using MoXing-v1.13.0-de803ac9
INFO:root:Using OBS-Python-SDK-3.1.2
INFO:root:Using MoXing-v1.13.0-de803ac9
INFO:root:Using OBS-Python-SDK-3.1.2
```

原因分析

Pytorch通过spawn模式创建了多个进程，每个进程会调用多进程方式使用Mox下载数据。此时子进程会不断销毁重建，Mox也就会不断的被导入，导致打印很多Mox的版本信息。

处理方法

为避免训练作业Pytorch Mox日志反复输出的问题，需要您在“启动文件”中添加如下代码，当“MOX_SILENT_MODE = “1””时，可在日志中屏蔽mox的版本信息：

```
import os
os.environ["MOX_SILENT_MODE"] = "1"
```

5.4 moxing.tensorflow 是否包含整个 TensorFlow，如何对生成的 checkpoint 进行本地 Fine Tune？

问题现象

使用MoXing训练模型，“global_step”放在Adam名称范围下，而非MoXing代码中没有Adam名称范围，如图5-1所示。其中1为使用MoXing代码，2代表非MoXing代码。

图 5-1 代码示例

```
1 ('Adam/betal_power', .[])
2 ('Adam/beta2_power', .[])
3 ('global_step', .[])
4 ('p2p/conv_lstm/LayerNorm/beta', .[8

<tf.Variable.'p2p/conv_lstm/LayerNorm_4/beta:0'.s
<tf.Variable.'p2p/conv_lstm/LayerNorm_4/gamma:0'.s
<tf.Variable.'p2p/output/weights:0'.shape=(7, 7, .
<tf.Variable.'Variable:0'.shape=() .dtype=int32_re
<tf.Variable.'betal_power:0'.s2ape=() .dtype=float
<tf.Variable.'beta2_power:0'.shape=() .dtype=float
<tf.Variable.'p2p/ds_x2/weights/Adam:0'.shape=(3,
<tf.Variable.'p2p/ds_x2/weights/Adam_1:0'.shape=(
<tf.Variable.'p2p/ds_x2/instance_norm/scale/Adam:
```

处理方法

Fine Tune就是用别人训练好的模型，加上自己的数据，来训练新的模型。相当于使用别人的模型的前几层，来提取浅层特征，然后在最后再落入我们自己的分类中。

由于一般新训练模型准确率都会从很低的价值开始慢慢上升，但是Fine Tune能够让我们在比较少的迭代次数之后得到一个比较好的效果。Fine Tune的好处在于不用完全重新训练模型，从而提高效率，在数据量不是很大的情况下，Fine Tune会是一个比较好的选择。

moxing.tensorflow包含所有的接口，对TensorFlow做了优化，里面的实际接口还是TensorFlow的原生接口。

当非MoXing代码中没有Adam名称范围时，需要修改非MoXing代码，在其中增加如下内容：

```
with tf.variable_scope("Adam"):
```

在增加代码时不建议使用自定义“global_step”，推荐使用
`tf.train.get_or_create_global_step()`。

5.5 训练作业使用 MoXing 拷贝数据较慢，重复打印日志

问题现象

- ModelArts训练作业使用MoXing拷贝数据较慢。
- 重复打印日志 INFO:root:Listing OBS。

图 5-2 重复打印日志

```
INFO:root:Listing OBS: 77000  
INFO:root:Listing OBS: 78000  
INFO:root:Listing OBS: 79000  
INFO:root:Listing OBS: 80000  
INFO:root:Listing OBS: 81000  
INFO:root:Listing OBS: 82000  
INFO:root:Listing OBS: 83000  
INFO:root:Listing OBS: 84000  
INFO:root:Listing OBS: 85000  
INFO:root:Listing OBS: 86000  
INFO:root:Listing OBS: 87000  
INFO:root:Listing OBS: 88000  
INFO:root:Listing OBS: 89000
```

原因分析

1. 拷贝数据慢的可能原因如下：
 - 直接从OBS上读数据会造成读数据变成训练的瓶颈，导致迭代缓慢。
 - 由于环境或网络问题，读OBS时遇到读取数据失败情况，从而导致整个作业失败。
2. 重复打印日志，该日志表示正在读取远端存在的文件，当文件列表读取完成以后，开始下载数据。如果文件比较多，那么该过程会消耗较长时间。

处理方法

在创建训练作业时，数据可以保存到OBS上。不建议使用TensorFlow、MXNet、PyTorch的OBS接口直接从OBS上读取数据。

- 如果文件较小，可以将OBS上的数据保存成“.tar”包。训练开始时从OBS上下载到“/cache”目录，解压以后使用。
- 如果文件较大，可以保存成多个“.tar”包，在入口脚本中调用多进程进行并行解压数据。不建议把散文件保存到OBS上，这样会导致下载数据很慢。
- 在训练作业中，使用如下代码进行“.tar”包解压：

```
import moxing as mox
import os
mox.file.copy_parallel("obs://donotdel-modelarts-test/AI/data/PyTorch-1.0.1/tiny-imagenet-200.tar", '/cache/tiny-imagenet-200.tar')
os.system('cd /cache; tar -xvf tiny-imagenet-200.tar > /dev/null 2>&1')
```

5.6 MoXing 如何访问文件夹并使用 get_size 读取文件夹大小？

问题现象

- 使用MoXing无法访问文件夹。
- 使用MoXing的“get_size”读取文件夹大小，显示为0。

原因分析

使用MoXing访问文件夹，需添加参数：“recursive=True”，默认为False。

处理方法

获取一个OBS文件夹的大小：

```
import moxing as mox
mox.file.get_size('obs://bucket_name/sub_dir_0/sub_dir_1', recursive=True)
```

获取一个OBS文件的大小：

```
import moxing as mox
mox.file.get_size('obs://bucket_name/obs_file.txt')
```


6 修订记录

发布日期	修订记录
2022-01-04	增加 OBS下载权限案例
2021-12-15	增加 ModelArts.2763案例
2021-09-15	训练作业模块新增若干故障排查案例。
2021-07-16	训练模块大纲整改。 删除一条训练模块故障排查，内容已过时。 增加训练模块故障排查。 训练作业进程异常退出 训练作业进程被kill
2020-12-10	增加自动学习故障排除指导。 数据集版本发布失败 数据集版本不合格 自动学习训练作业创建失败 自动学习训练作业失败 模型发布任务提交失败 模型发布失败 部署上线任务提交失败 部署上线失败
2019-11-25	第一次正式发布。