机器学习PAI

常见问题

常见问题

PAI算法组件Q&A

目录

如果以上内容无法解决您的问题，请首先查看PAI知识库:https://help.aliyun.com/product/30347.html,若问题仍得不到解决请粘贴logview（Tensorflow日志中的蓝色长链接）到PAI工单系统进行提问，工单系统地址：https://workorder.console.aliyun.com/console.htm?spm=5176.product30347.4.8.ZvFh1j#/ticket/add?productCode=pai&commonQuestionId=660

运行格式转换组件出错

格式转换组件默认起100个worker，请检查数据量是否大于100条。

PAI数据展示出现blob字符

在PAI平台显示文本的时候，右键查看数据的时候部分文本变成blob，因为有部分字符不可转码，所以显示成为了blob，不影响下游节点的读取和处理。

x13-auto-arima使用过程报错

x13-auto-arima的训练数据条数有限制，不能超过1200条

DOC2VEC报错CallExecutorToParseTaskFail

我们现在的doc2vec支持的规模是 (doc个数+word个数)Xvec长度 <2410000X10000; 而用户的规模是42432500X7712293X300，超出了我们的支持范围，内存申请失败，导致core掉，这个后边会fix这个算法，明确的报出规模限制，目前用户需要缩小数据的规模，才能计算，并且看起来用户的输入需要分词

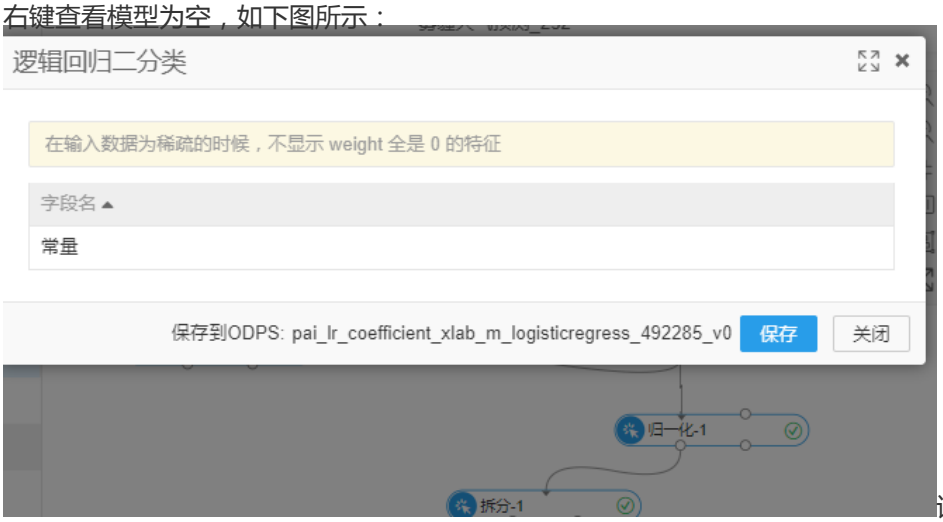
PAI模型数据相关Q&A

目录

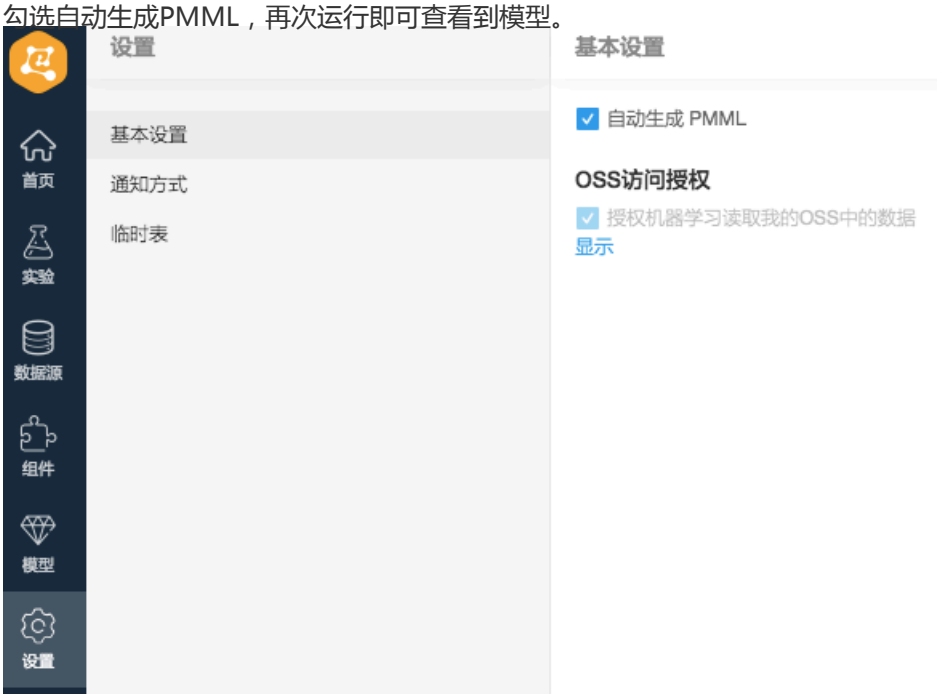
如果以上内容无法解决您的问题，请首先查看PAI知识库:https://help.aliyun.com/product/30347.html,若问题仍得不到解决请粘贴logview（Tensorflow日志中的蓝色长链接）到PAI工单系统进行提问，工单系统地址：https://workorder.console.aliyun.com/console.htm?spm=5176.product30347.4.8.ZvFh1j#/ticket/add?productCode=pai&commonQuestionId=660

为什么使用PAI生成的模型为空

右键查看模型为空，如下图所示：

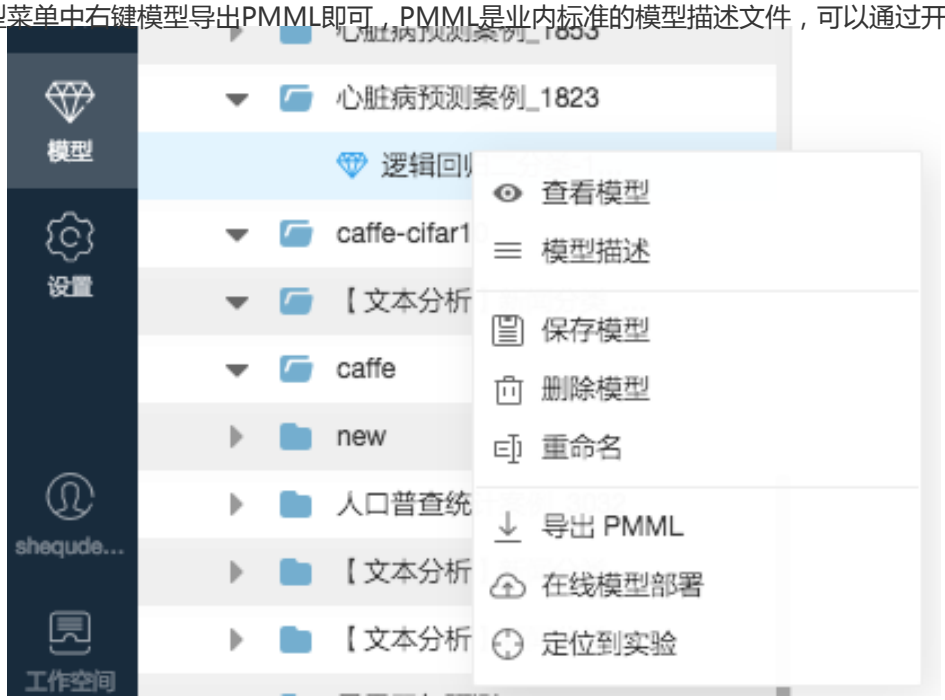


请在PAI界面点击设置，并且勾选自动生成PMML，再次运行即可查看到模型。



如何下载PAI实验生成的模型

在模型菜单中右键模型导出PMML即可，PMML是业内标准的模型描述文件，可以通过开源工具解析PMML并使用。



如何上传或下载PAI上面的数据

数据下载请参见：https://help.aliyun.com/document_detail/52260.html
https://help.aliyun.com/video_detail/54945.html

PAI在线预测 Q&A

目录

如果以上内容无法解决您的问题，请首先查看PAI知识库:https://help.aliyun.com/product/30347.html,若问题仍得不到解决请粘贴logview（Tensorflow日志中的蓝色长链接）到PAI工单系统进行提问，工单系统地址：https://workorder.console.aliyun.com/console.htm?spm=5176.product30347.4.8.ZvFh1j#/ticket/add?productCode=pai&commonQuestionId=660

PAI在线预测说明在哪里

在线预测相关的问题可以参考以下两篇文章：

https://help.aliyun.com/document_detail/45395.html

https://help.aliyun.com/document_detail/58275.html

PAI在线预测只是针对模型的在线预测处理，并不是针对全部流程的在线预测。

AuthorizationFailed错误

在线预测调用目前只支持主账号，这个错误是子账号调用造成的报错。

kInvalidArgument错误

用户body字段输入错误，请仔细参看在线预测案例文档

CanNotVisitTheRouter错误

在线预测请求URL错误，请仔细参看在线预测案例文档

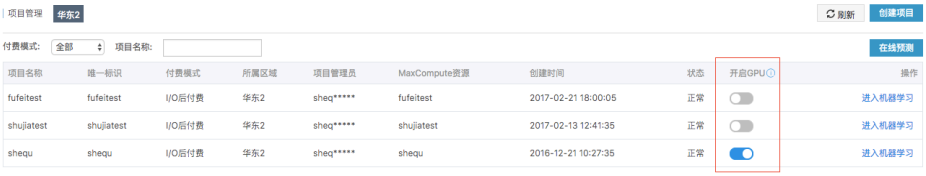
PAI 深度学习 Q&A

目录

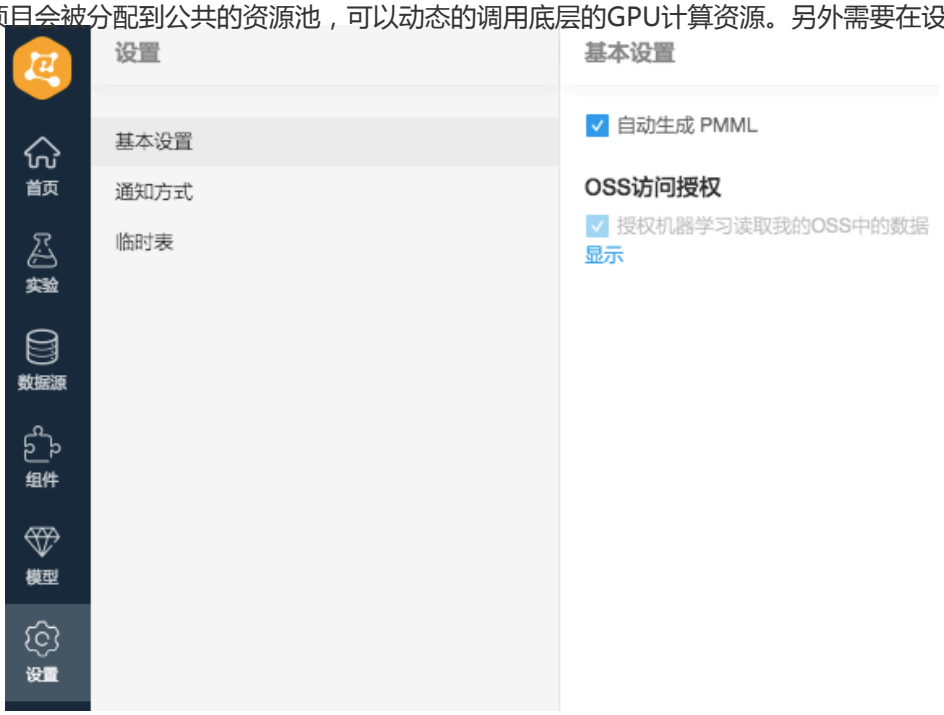
如果以上内容无法解决您的问题，请首先查看PAI知识库:https://help.aliyun.com/product/30347.html,若问题仍得不到解决请粘贴logview（Tensorflow日志中的蓝色长链接）到PAI工单系统进行提问，工单系统地址：https://workorder.console.aliyun.com/console.htm?spm=5176.product30347.4.8.ZvFh1j#/ticket/add?productCode=pai&commonQuestionId=660

如何开通PAI的深度学习功能

目前机器学习平台深度学习相关功能处于公测阶段，深度学习组件包含TensorFlow、Caffe、MXNet三个框架，开通方式如下，进入机器学习控制台，在相应项目下勾选GPU资源即可使用。

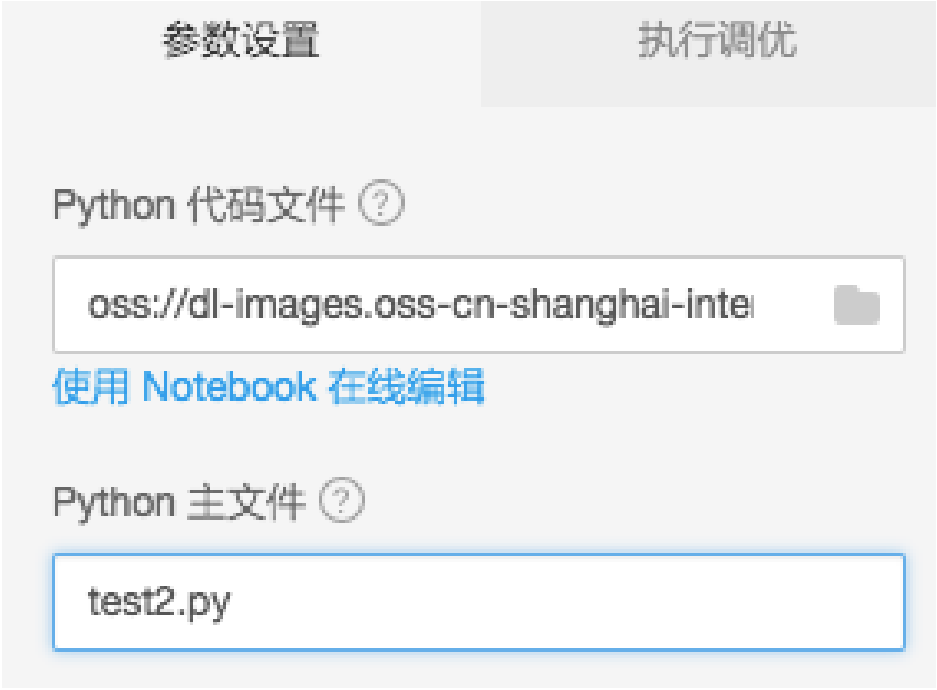


开通GPU资源的项目会被分配到公共的资源池，可以动态的调用底层的GPU计算资源。另外需要在设置中设置OSS的访问权限：



如何支持多python文件脚本引用

很多时候我们通过python 模块文件组织训练脚本。可能将模型定义在不同的Python文件里，将数据的预处理逻辑放在另外一个Python文件中，最后有一个Python文件将整个训练过程串联起来。例如：在test1.py中定义了一些函数，需要在test2.py文件使用test1.py中的函数，并且将test2.py作为程序入口函数，只需要将test1.py和test2.py打包成tar.gz文件上传即可。



- Python代码文件为定义的tar.gz包
- Python主文件定义入口程序文件

如何上传数据到OSS

可以观看视频：https://help.aliyun.com/video_detail/54945.html

使用深度学习处理数据时，数据先存储到OSS的bucket中。第一步要创建OSS Bucket。 由于深度学习的GPU集群在华东2，建议您创建 OSS Bucket 时选择华东2地区。这样在数据传输时就可以使用阿里云经典网络，算法运行时不需要收取流量费用。Bucket 创建好之后，可以在OSS管理控制台 来创建文件夹，组织数据目录，上传数据了。

OSS支持多种方式上传数据， API或SDK详细见：
https://help.aliyun.com/document_detail/31848.html?spm=5176.doc31848.6.580.a6es2a

OSS还提供了大量的常用工具用来帮助用户更加高效的使用OSS。工具列表请参见：
https://help.aliyun.com/document_detail/44075.html?spm=5176.doc32184.6.1012.XlMMUx

建议您使用 ossutil 或 osscmd ，这是两个命令行工具，通过命令的方式来上传、下载文件，还支持断点续传。

注：在使用工具时需要配置 AccessKey 和ID，登录后，可以在Access Key 管理控制台创建或查看。

如何使用PAI读取OSS数据

Python不支持读取oss的数据, 故所有调用python **Open() os.path.exist()** 等文件, 文件夹操作的函数的代码都无法执行.如**Scipy.misc.imread(),numpy.load()** 等

那如何在PAI读取数据呢, 通常我们采用两种办法.

方法一

如果只是简单的读取一张图片, 或者一个文本等, 可以使用**tf.gfile**下的函数, 具体成员函数如下

```
tf.gfile.Copy(oldpath, newpath, overwrite=False) # 拷贝文件
tf.gfile.DeleteRecursively(dirname) # 递归删除目录下所有文件
tf.gfile.Exists(filename) # 文件是否存在
tf.gfile.FastGFile(name, mode='r') # 无阻塞读取文件
tf.gfile.GFile(name, mode='r') # 读取文件
tf.gfile.Glob(filename) # 列出文件夹下所有文件, 支持pattern
tf.gfile.IsDirectory(dirname) # 返回dirname是否为一个目录
tf.gfile.ListDirectory(dirname) # 列出dirname下所有文件
tf.gfile.MakeDirs(dirname) # 在dirname下创建一个文件夹, 如果父目录不存在, 会自动创建父目录. 如果
文件夹已经存在, 且文件夹可写, 会返回成功
tf.gfile.MkDir(dirname) # 在dirname处创建一个文件夹
tf.gfile.Remove(filename) # 删除filename
tf.gfile.Rename(oldname, newname, overwrite=False) # 重命名
tf.gfile.Stat(dirname) # 返回目录的统计数据
tf.gfile.Walk(top, inOrder=True) # 返回目录的文件树
```

具体的文档可以参照这里(可能需要翻墙)

方法二

如果是一批一批的读取文件, 一般会采用**tf.WhoFileReader()** 和 **tf.train.batch()** 或**tf.train.shuffer_batch()**

接下来会重点介绍常用的 **tf.gfile.Glob**, **tf.gfile.FastGFile**, **tf.WhoFileReader()** 和**tf.train.shuffer_batch()**

读取文件一般有两步：

1. 获取文件列表
2. 读取文件

如果是批量读取, 还有第三步：创建batch

从代码上手:在使用PAI的时候, 通常需要在右侧设置读取目录, 代码文件等参数, 这些参数都会通过—XXX的形式传入，**tf.flags**可以提供了这个功能

```
import tensorflow as tf
FLAGS = tf.flags.FLAGS
tf.flags.DEFINE_string('buckets', 'oss://{OSS Bucket}/', '训练图片所在文件夹')
tf.flags.DEFINE_string('batch_size', '15', 'batch大小')
files = tf.gfile.Glob(os.path.join(FLAGS.buckets,'*.jpg')) # 如我想列出buckets下所有jpg文件路径
```

接下来就分两种情况了

- 小规模读取时建议：tf.gfile.FastGfile()

```
for path in files:
file_content = tf.gfile.FastGFile(path, 'rb').read() # 一定记得使用rb读取, 不然很多情况下都会报错
image = tf.image.decode_jpeg(file_content, channels=3) # 本教程以JPG图片为例
```

- 大批量读取时建议：tf.WhoFileReader()

```
reader = tf.WholeFileReader()  # 实例化一个reader
fileQueue = tf.train.string_input_producer(files) # 创建一个供reader读取的队列
file_name, file_content = reader.read(fileQueue) # 使reader从队列中读取一个文件
image_content = tf.image.decode_jpeg(file_content, channels=3) # 讲读取结果解码为图片
label = XXX # 这里省略处理label的过程
batch = tf.train.shuffle_batch([label, image_content], batch_size=FLAGS.batch_size, num_threads=4,
capacity=1000 + 3 * FLAGS.batch_size, min_after_dequeue=1000)

sess = tf.Session() # 创建Session
tf.train.start_queue_runners(sess=sess) # 重要!!! 这个函数是启动队列, 不加这句线程会一直阻塞
labels, images = sess.run(batch) # 获取结果
```

现在解释下其中重要的部分

1. tf.train.string_input_producer, 这个是把files转换成一个队列, 并且需要 **tf.train.start_queue_runners** 来启动队列
2. tf.train.shuffle_batch 参数解释

- **batch_size** 批大小, 每次运行这个batch, 返回多少个数据
- **num_threads** 运行线程数, 在PAI上4个就好
- **capacity** 随机取文件范围, 比如你的数据集有10000个数据, 你想从5000个数据中随机取, capacity就设置成5000.
- **min_after_dequeue** 维持队列的最小长度, 这里只要注意不要大于**capacity**即可

如何使用PAI写入数据到OSS

- 直接使用tf.gfile.FastGFile()写入

```
tf.gfile.FastGFile(FLAGS.checkpointDir + 'example.txt', 'wb').write('hello world')
```

- 通过tf.gfile.Copy()拷贝

```
tf.gfile.Copy('./example.txt', FLAGS.checkpointDir + 'example.txt')
```

通过这两种方法, 文件都会出现在 ‘输出目录/model/example.txt’ 下

PAI平台关于Tensorflow的案例有哪些

案例一:如何使用TensorFlow实现图像分类

视频地址：https://help.aliyun.com/video_detail/54948.html

文档介绍：https://yq.aliyun.com/articles/72841

代码下载：https://help.aliyun.com/document_detail/51800.html

案例二:如何使用TensorFlow自动写歌

文档介绍：https://yq.aliyun.com/articles/134287

代码下载：https://help.aliyun.com/document_detail/57011.html

如何查看Tensorflow的相关日志

具体请参考：https://yq.aliyun.com/articles/72841

model_average_iter_interval这个参数在我设置两个GPU的时候起到什么作用

model_average_iter_interval这个参数在两个GPU时不设置的话，就相当于是运行标准的parallel-sgd，每个iteration都会交换梯度更新，model_average_iter_interval大于1的话，就是使用model Average 方法，间隔若干轮（model_average_iter_interval设置的数值轮数）两个平均模型参数。两卡带来的增益是训练速度的提升.