机器学习PAI

常见问题

常见问题

算法组件常见问题

目录

运行格式转换组件出错

机器学习平台数据展示出现“blob”字符

x13-auto-arima 使用过程报错

DOC2VEC 报错 CallExecutorToParseTaskFail

如果以上内容无法解决您的问题，请首先查看机器学习知识库，若问题仍得不到解决请粘贴logview（Tensorflow日志中的蓝色链接）到机器学习工单系统进行提问。

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。

机器学习平台数据展示出现“blob”字符

现象描述

在机器学习平台右键查看数据时部分文本变成“blob”字符。

解决方法

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的训练数据的规模有限制，不能超过1200条。

DOC2VEC 报错 CallExecutorToParseTaskFail

DOC2VEC支持的规模是（doc个数+word个数）× vec长度，小于 2410000×10000。而用户的规模是42432500×7712293×300，超出了组件的支持范围，导致内存申请失败。
目前用户需要缩小数据的规模才能计算，并且输入的数据需要分词。

DSW常见问题

目录

DSW与机器学习PAI

DSW实例如何挂载自己的NAS

如何在DSW里使用第三方的库

DSW的深度学习简单代码例子

DSW跑机器学习代码，页面放置一段时间后提示重新登录

执行挂载mount命令出现报错mount:wrong fs type

DSW与OSS数据传输

DSW生成的模型如何部署

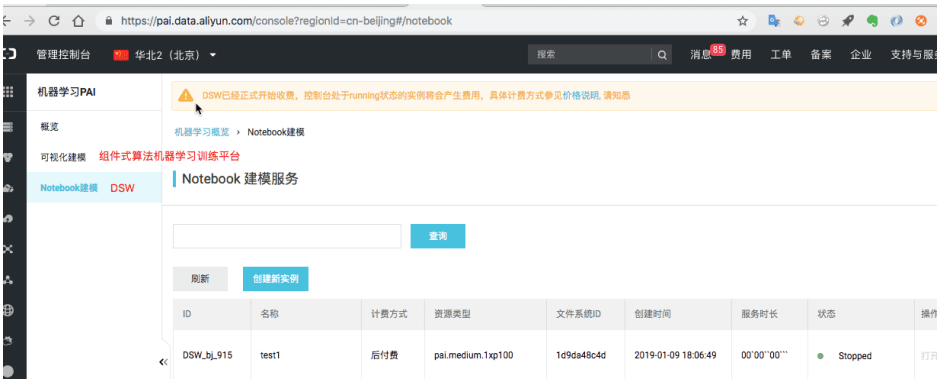
DSW如何收费

如何查看DSW账单

如果以上内容无法解决您的问题，请首先查看Notebook知识库，若问题仍得不到解决请粘贴到机器学习工单系统进行提问。

1. DSW平台与机器学习PAI

DSW(Data Science Workshop)是PAI团队新开发的在线深度学习开发平台，内置了深度优化后的tensorflow框架，底层有M40、P100的GPU卡支撑，用户可在平台在线编写、执行深度学习代码，并可下载生成模型到本地，与原先的可视化建模PAI是两个并列的平台。用户进入PAI控制台可通过切换左侧的tab访问这两个平台。



2. DSW实例如何挂载和使用自己的NAS文件系统

NAS是阿里云一款可共享访问、弹性扩展的文件存储产品，DSW与NAS是打通的，训练数据和代码等均存储于NAS。目前DSW实例分为两种：使用系统默认分配5G的NAS存储空间实例和挂载用户自己的NAS实例。如果训练数据量比较大，建议使用自己的NAS文件系统。创建文件系统后，记录文件系统的ID，创建DSW实例时输入该文件系统ID。挂载成功后，用户的nas文件都存放在/nas目录下，用户可使用DSW Terminal进入该目录查看和使用。更详细的操作步骤可参考DSW使用文档。

3. 如何在DSW里使用第三方的库

目前DSW是支持第三库的安装，可以进入DSW Terminal输入如下命令完成安装。

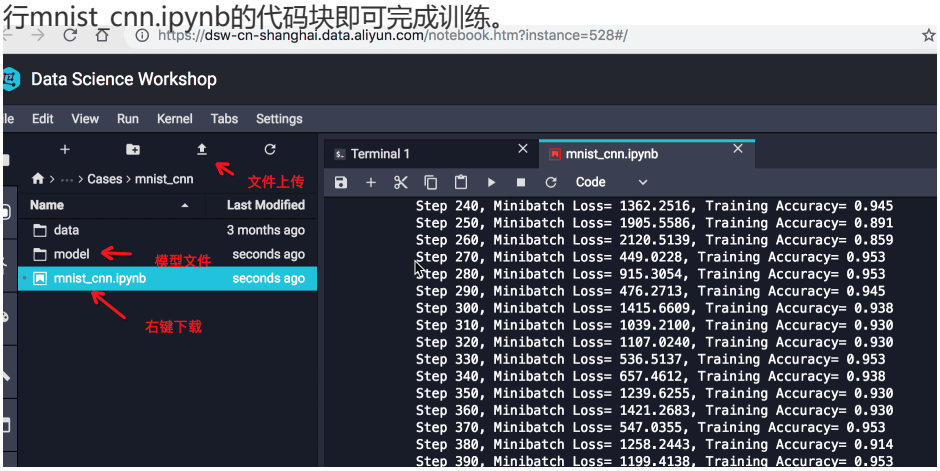
```
#python3版本：
pip install --user xxx

#python2版本：
source activate python2
pip install --user xxx
```

安装成功后，点击kernel->restart kernel重启服务。也可以通过导航右上角加入钉钉群联系我们或者提交工单工单系统进行提问，考虑后续平台内置安装常用的第三方包。

4. DSW的深度学习简单代码例子

在DSW里内置了一个基于mnist数据集的tensorflow简单例子，包含数据和训练代码，用户可直接下载试用。

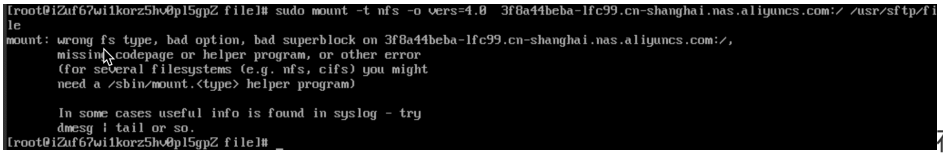
切换左侧tab至Demo例子文件夹，点击mnist cnn下载，完成下载后切换左侧tab至自己的Demo文件夹，进入/Demo/Cases 文件夹，可看到新增了名为mnist_cnn的文件夹。mnist_cnn是一个手写体识别的例子，逐个运行mnist_cnn.ipynb的代码块即可完成训练。



5. DSW跑机器学习代码，页面放置一段时间后提示重新登录，怎么办？

目前DSW由于安全上的需要，登录的Session的有效期是3个小时，过期需要重新登录，不会影响任务的执行。如果有长时间运行任务，建议使用DSW中的Terminal，使用nohup命令后台执行。

6. 使用ECS搭建ftp上传下载文件到NAS，执行挂载mount命令出现报错mount:wrong fs type,bad option,bad superlock



在执行mount命令之前，需要先安装nfs-utils安装包

```
yum install nfs-utils
```

7. DSW与OSS数据传输

在DSW上如果需要与OSS打通，读取OSS里的数据，可以进入DSW Terminal使用osscmd命令来实现文件的上传和下载。

```
#如果出现类似“Your configuration is saved into ”的提示，表明ID和KEY已经保存成功。
$ osscmd config --id=accessid --key=accesskey --host=your_endpoint
#文件上传
$ osscmd put local_existed_file oss://mybucketname/test_object
#文件下载
$ osscmd get oss://mybucketname/test_object download_file
```

数据从OSS下载到DSW的步骤：

数据从DSW上传到OSS的步骤：

更详细的操作步骤请参见文档osscmd命令的使用

8. DSW生成的模型如何部署

用户在DSW生成模型用户可通过右键的方式下载到本地，对于比较大的模型可通过ecs服务器搭建ftp的方式实现下载，具体的操作步骤可参考文档：数据上传和下载

9. DSW如何收费？

在使用DSW之前需要先购买服务，DSW的付费方式包含预付费和后付费。对于选择后付费的用户，如果已经开通了PAI的后付费用户，不需要再次开通，可直接使用DSW（不创建实例不会产生费用），M40的GPU卡定价是8.4元/卡/小时，P100的GPU卡定价是12元/卡/小时。选择预付费即包年包月，目前是2000元/月/GPU，用户可根据自己的实际需要来选择付费方式。**（2019年1月31号之前有活动优惠）**

在实际使用过程中，用户在控制台可以看到每个实例的运行时长。实例停止状态不计入运行时长、也不会产生费用，对于暂时不用的实例可执行暂停操作，之后可以重新启动，数据和代码均不会丢失。

10. 如何查看DSW生成的账单

对于选择后付费的用户，可以进入费用中心查看账单。

产品栏选择 机器学习（pai），选定账期即可看到账单详情，包含实例ID，每个实例运行的时长以及产生的费用。用户可在次日出账后，自行查看详细账单。

模型数据常见问题

目录

为什么实验生成的模型为空

如何下载实验生成的模型

如何在机器学习平台上传数据

如果以上内容无法解决您的问题，请首先查看**机器学习知识库**，若问题仍得不到解决请粘贴logview（Tensorflow日志中的蓝色链接）到机器学习**工单系统**进行提问。

为什么实验生成的模型为空

现象描述

右键单击模型，选择**查看模型**，结果为空，如下图所示。

解决方法

在机器学习界面单击**设置**，勾选**自动生成PMML**，如下图所示，再次运行即可查看到模型。

如何下载实验生成的模型

在模型菜单中右键单击模型，选择**导出PMML**（PMML是业内标准的模型描述文件，可以通过开源工具解析并使用），如下图所示。

如何在机器学习平台上传数据

数据上传视频：如何上传数据

数据上传文档：数据准备

在线预测功能常见问题

目录

机器学习在线预测说明在哪里

AuthorizationFailed错误

kInvalidArgument错误

CanNotVisitTheRouter错误

如果以上内容无法解决您的问题，请首先查看机器学习知识库，若问题仍得不到解决请粘贴logview（Tensorflow日志中的蓝色链接）到机器学习工单系统进行提问。

机器学习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：

在线预测部署功能介绍

模型在线预测

机器学习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

子账号调用造成的报错，在线预测调用目前只支持主账号。

kInvalidArgument错误

用户body字段输入错误，请仔细查看模型在线预测文档。

CanNotVisitTheRouter错误

在线预测请求URL错误，请仔细查看模型在线预测文档。

TensorFlow常见问题

目录

如何开通深度学习功能

如何支持多python文件脚本引用

如何上传数据到OSS

如何读取OSS数据

如何写入数据到OSS

运行出现OOM原因

Tensorflow案例有哪些

其它问题

如果以上内容无法解决您的问题，请首先查看机器学习知识库，若问题仍得不到解决请粘贴logview（Tensorflow日志中的蓝色链接）到机器学习工单系统进行提问。

如何开通深度学习功能

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。开通方式如下图，进入机器学习控制台，在相应项目下开启GPU资源即可。

开通GPU资源的项目会被分配到公共的资源池，可以动态地调用底层的GPU计算资源。另外需要设置OSS的访问权限，如下图所示。

如何支持多python文件脚本引用

可以通过python模块文件组织训练脚本。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作为程序入口文件，只需要将test1.py和test2.py打包成tar.gz文件上传即可，如下图所示。

- **Python代码文件**为定义的tar.gz包
- **Python主文件**为定义的入口程序文件

如何上传数据到OSS

详细步骤可参考**如何上传数据**视频。

使用深度学习算法处理数据时，数据先存储到OSS的bucket中。首先要创建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管理控制台创建文件夹、组织数据目录、上传数据。

> OSS支持多种方式上传数据，API或SDK请参见 https://help.aliyun.com/document_detail/31848.html?spm=5176.doc31848.6.580.a6es2a 。
>
> OSS提供了大量工具来帮助用户更加高效地使用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命令行工具，通过命令的方式来上传下载文件，同时支持断点续传。

> **注意**：在使用工具时需要配置 AccessID 和 AccessKey，请登录阿里云管理控制台，并在 Access Key 管理界面创建或查看。

如何读取OSS数据

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

通常采用以下两种办法在机器学习平台读取数据。

使用**tf.gfile**下的函数，适用于简单地读取一张图片，或者一个文本等，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
```

具体请参考tf.gfile模块。

使用tf.gfile.Glob、tf.gfile.FastGFile、 tf.WhoFileReader() 、tf.train.shuffer_batch()，适用于批量读取文件（读取文件之前需要获取文件列表，如果是批量读取，还需要创建batch）。

使用机器学习搭建深度学习实验时，通常需要在界面右侧设置读取目录、代码文件等参数。这些参数通过“—XXX”（XXX代表字符串）的形式传入，**tf.flags**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
```

小规模读取文件时建议使用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
```

大批量读取文件时建议使用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
```

部分代码解释如下：

- tf.train.string_input_producer：把**files**转换成一个队列，并且需要 **tf.train.start_queue_runners** 来启动队列。
- tf.train.shuffle_batch参数解释如下：
  - **batch_size**：批处理大小。即每次运行这个batch，返回的数据个数。
  - **num_threads**：运行线程数，一般设置为4。
  - **capacity**：随机取文件范围。比如数据集有10000个数据，需要从5000个数据中随机抽取，那么**capacity**就设置成5000。
  - **min_after_dequeue**：维持队列的最小长度，不能大于**capacity**。

如何写入数据到OSS

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')
```

通过以上两种方法，将数据写入OSS中，生成的文件存储在“输出目录/model/example.txt”下。

运行出现OOM原因

内存使用达到30G上限，建议通过gfile读取OSS，参考如何读取OSS数据

Tensorflow案例有哪些

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

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

其它问题

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参数在设置两个GPU的时候起到什么作用？

- 如果没有设置**model_average_iter_interval**参数，GPU会运行标准的parallel-sgd，每个迭代都会交换梯度更新。
- 如果**model_average_iter_interval**大于1，就是使用 model Average 方法，间隔若干轮（**model_average_iter_interval**设置数值轮数）两个平均模型参数。

两卡带来的增益是训练速度的提升。