ALIBABA CLOUD

阿里云

机器学习PAI

常见问题

文档版本：20201130

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.算法组件常见问题

本文为您介绍算法组件的相关问题。

格式转换组件运行报错

格式转换组件默认启动100个Worker，请检查数据量是否大于100条。

PAI平台数据展示出现“blob”字符

- 现象描述

  在画布中，右键单击组件，在快捷菜单，单击**查看数据**时，部分文本显示为blob字符。

- 解决方法

  因为部分字符不能转码，所以显示为blob，该现象并不影响下游节点读取并处理数据。

x13-auto-arima组件运行报错

x13-auto-arima的训练数据规模不能超过1200条。

Doc2Vec组件运行报错CallExecutorToParseTaskFail

Doc2Vec组件的数据规模 `（Doc个数+Word个数）× Vec长度` 必须小于2410000×10000，用户规模必须小于42432500×7712293×300。如果超出了该范围，则会导致内存申请失败。您可以先缩小数据规模，再计算，且输入的数据需要进行分词。

2.PAI-DSW常见问题

本文为您介绍PAI-DSW的相关问题。

什么是PAI-DSW?

PAI-DSW（Data Science Workshop）是PAI推出的在线深度学习开发平台，预置了深度优化后的TensorFlow框架，且支持M40和P100的GPU卡运行训练任务。您可以通过该平台在线编写及执行深度学习代码，并将生成的训练模型下载至本地。

PAI-DSW实例如何挂载和使用自己的NAS文件系统?

NAS是阿里云的文件存储产品，PAI-DSW的训练数据和代码等均存储于NAS。PAI-DSW实例分为使用系统默认分配5 GB NAS存储空间的实例和挂载自己NAS的实例。如果训练数据量较大，建议使用自己的NAS文件系统。创建PAI-DSW实例时，输入自己的NAS文件系统ID，即可挂载该NAS，详情请参见扩容实例。您所有的NAS文件均存储在*/nas*目录，可以通过PAI-DSW Terminal进入该目录查看并使用文件。

如何在PAI-DSW中使用第三方库

PAI-DSW支持安装第三库，可以使用PAI-DSW Terminal输入如下命令完成安装。

```
#Python 3版本。
pip install --user xxx
#Python 2版本。
source activate python2
pip install --user xxx
```

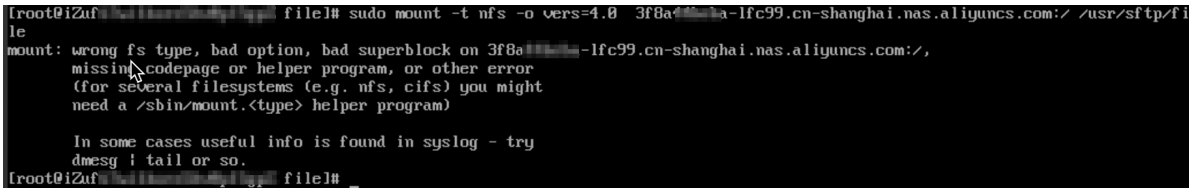
其中xxx需要替换为待安装的第三包名称。安装成功后，需要单击**kernel > restart kernel**，重启服务。

运行机器学习代码时，为什么页面放置一段时间后提示重新登录?

为安全考虑，PAI-DSW登录Session的有效期为3个小时，过期后需要重新登录，但是不会影响任务的执行。如果需要长时间运行任务，建议在PAI-DSW Terminal，使用 `nohup` 命令后台执行任务。

使用ECS搭建FTP上传下载文件到NAS，执行挂载（mount）命令报错mount:wrong fs type,bad option,bad superlock

- 现象描述



- 解决方法

  执行 `mount` 命令之前，先安装nfs-utils安装包。

  ```
  yum install nfs-utils
  ```

如何使用PAI-DSW读取OSS数据？

进入PAI-DSW Terminal，使用 `osscmd` 命令实现文件的上传和下载，示例如下。

```
#如果出现类似“Your configuration is saved into ”的提示，表示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
```

为什么安装的第三方包没有生效？

通过 `pip` 命令安装第三方包后，使用 `import` 命令导入时，如果出现无法查找到该包的问题，则先尝试重启服务。如果依然报错，则确认当前使用的环境。安装第三方包时，PAI-DSW默认安装到Python 3环境。如果需要安装到其他环境，则必须先手动切换环境再进行安装，示例如下。

```
安装到Python 2环境。
source activate python2
pip install --user xxx
安装到TensorFlow 2.0环境。
source activate tf2
pip install --user xxx
```

其中xxx需要替换为待安装的第三方包名称。

如何部署PAI-DSW生成的模型？

- 使用PAI-EAS模型部署服务

  使用PAI-DSW预置的EASCMD，在Terminal中使用命令行部署模型服务，详情请参见部署模型。

- 下载模型到本地部署

  您可以通过右键单击PAI-DSW生成的模型将其下载至本地。如果模型较大，则可以通过ECS服务器搭建FTP的方式实现模型下载，详情请参见上传下载数据。

PAI-DSW如何收费？

PAI-DSW支持预付费和后付费，您可以根据自己的实际需要选择付费方式，计费详情请参见PAI-DSW计费说明。

如何查看PAI-DSW账单？

对于后付费用户，可以进入用户中心查看账单明细，详情请参见查看账单与用量明细。

3.模型数据常见问题

本文为您介绍模型数据的相关问题。

- 为什么实验生成的模型为空？
- 如何下载实验生成的模型？
- 如何上传数据？

为什么实验生成的模型为空？

- 现象描述。

  右键单击画布中的模型训练组件，在快捷菜单，单击**查看模型**，其结果为空。

- 解决方法：

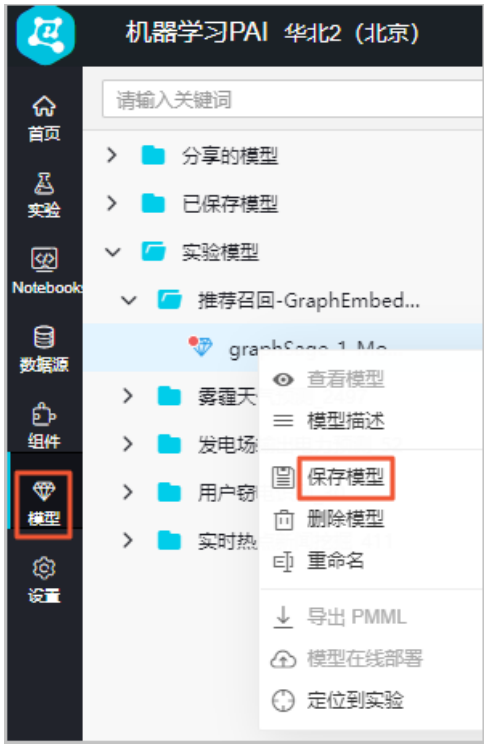
  i. 在PAI-Studio控制台页面的左侧导航栏，单击**设置**。

  ii. 在**基本设置**页面，选中**自动生成PMML**复选框。

  iii. 重新运行实验，即可查看模型。

如何下载实验生成的模型？

1. 在PAI-Studio控制台页面的左侧导航栏，单击**模型**。
2. 在模型列表，右键单击待下载模型，在快捷菜单，单击**保存模型**。



如何上传数据？

上传数据视频请参见如何上传数据。

上传数据文档请参见准备数据。

4.在线预测功能常见问题

本文为您介绍在线预测的相关问题。

在线预测的说明在哪里？

在线预测仅针对模型的在线预测处理，并非针对全流程的在线预测，详情请参见PAI-Studio模型部署及预测。

为什么出现AuthorizationFailed错误？

RAM用户调用导致的报错，在线预测调用仅支持主账号。

为什么出现kInvalidArgument错误？

Body字段输入错误。

为什么出现CanNotVisitTheRouter错误？

在线预测请求URL错误。

5.TensorFlow常见问题

本文为您介绍TensorFlow的相关问题。

如何开通深度学习功能？

PAI提供的深度学习组件包括TensorFlow、Caffe及MXNet，需要进行GPU资源和OSS访问授权，详情请参见授权。

如何支持多Python文件引用？

您可以通过Python文件组织训练脚本。通常首先将数据预处理逻辑存放在某个Python文件中，然后将模型定义在另一个Python文件中，最后通过一个Python文件串联整个训练过程。例如，在test1.py中定义函数，如果test2.py文件需要使用test1.py中的函数，且将test2.py作为程序入口文件，则只需要将test1.py和test2.py打包为.tar.gz包并上传即可，如下图所示。

其中：

- **Python代码文件**：.tar.gz包。
- **Python主文件**：入口程序文件。

如何上传数据到OSS？

上传数据的视频请参见如何上传数据。

深度学习算法的数据存储在OSS的Bucket中，因此需要先创建OSS Bucket。建议您将OSS Bucket创建在与深度学习GPU集群相同的地域，从而使用阿里云经典网络进行数据传输，进而使算法运行免收流量费。创建OSS Bucket后，可以在OSS管理控制台创建文件夹、组织数据目录或上传数据。

您可以通过API或SDK上传数据至OSS，详情请参见简单上传。同时，OSS提供了大量工具（工具列表请参见OSS常用工具汇总。）帮助您更高效地完成任务，建议使用ossutil或osscmd工具上传下载文件。

> **说明** 使用工具上传文件时，需要配置AccessKey ID和AccessKey Secret，您可以登录阿里云管理控制台创建或查看该信息。

如何读取OSS数据？

Python不支持读取OSS数据，因此所有调用Python `Open()` 及 `os.path.exist()` 等文件和文件夹操作函数的代码都无法执行。例如 `Scipy.misc.imread()` 及 `numpy.load()` 等。

通常采用以下两种方式在PAI中读取数据：

- 使用tf.gfile下的函数，适用于简单读取一张图片或一个文本等。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, 则自动创建父目录。如果文件夹已经存在, 且文件夹可写, 则返回成功。
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。
```

- 使用 `tf.gfile.Glob` 、 `tf.gfile.FastGFile` 、 `tf.WhoFileReader()` 及 `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。如果批量读取，还需要创建Batch）。

使用PAI-Studio搭建深度学习实验时，通常需要在界面右侧设置读取目录及代码文件等参数。tf.flags支持通过-XXX（XXX表示字符串）的形式传入该参数。

```
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')) # 列出buckets下所有JPG文件路径。
```

批量读取文件时，对于不同规模的文件，建议分别使用如下方式：

- 读取小规模文件时，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, 否则很多情况下都会报错。
  image = tf.image.decode_jpeg(file_content, channels=3) # 以JPG图片为例。
```

- 读取大规模文件时，建议使用 `tf.WhoFileReader()` 。

```
reader = tf.WholeFileReader() # 实例化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) # 将读取结果解码为图片。
label = XXX # 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) # 启动队列。如果未执行该命令，则线程会一直阻塞。
labels, images = sess.run(batch) # 获取结果。
```

核心代码解释如下：

- `tf.train.string_input_producer` ：将files转换为队列，且需要使用 `tf.train.start_queue_runners` 启动队列。
- `tf.train.shuffle_batch` 参数如下：
  - batch_size：批处理大小，即每次运行Batch返回的数据数量。
  - num_threads：运行线程数，一般设置为4。
  - capacity：随机取文件范围。例如，数据集有10000个数据，如果需要从5000个数据中随机抽取，则将capacity配置为5000。
  - min_after_dequeue：维持队列的最小长度，不能大于**capacity**。

如何为OSS写入数据？

您可以通过以下任意一种方式将数据写入OSS中，生成的文件存储在输出目录*/model/example.txt*：

- 通过 `tf.gfile.FastGFile()` 写入，示例如下。

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

- 通过 `tf.gfile.Copy()` 拷贝，示例如下。

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

为什么运行过程中出现OOM？

使用的内存达到上线30 GB，建议通过gfile读取OSS数据，详情请参见如何读取OSS数据？ 。

TensorFlow有哪些案例？

- 使用TensorFlow实现图像分类，详情请参见使用TensorFlow实现图片分类视频、使用TensorFlow实现图片分类及TensorFlow案例相关代码。

- 使用TensorFlow自动写歌，详情请参见TensorFlow自动写歌词及TensorFlow自动写歌词案例相关代码。

如何查看TensorFlow相关日志？

查看TensorFlow相关日志请参见查看训练中的日志。

配置两个GPU时，model_average_iter_interval有什么作用？

如果未配置model_average_iter_interval参数，则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，则使用Model Average方法，训练迭代间隔若干轮（model_average_iter_interval表示数值轮数）计算两个平均模型参数。

6.计费常见问题

本文为您介绍计费相关问题。

什么项目会产生费用？

- PAI-Studio运行中的实验，其算法组件在计算过程中会产生对应费用。
- PAI-DSW运行中的实例会占用计算资源，从而产生费用。对于不需要使用的实例，建议及时停止该实例，避免产生不必要的费用。
- PAI-EAS运行在公共资源组的服务会产生费用（无论是否调用该服务）。运行在后付费专属资源组中的服务，后付费专属资源组会产生费用。运行在预付费专属资源组中的服务，预付费资源组在购买时已付费，使用中不再产生费用。

  简单的计费规则如下，计费规则详情请参见PAI-EAS计费说明：

  - 后付费专属资源组处于**运行中**，系统计费。

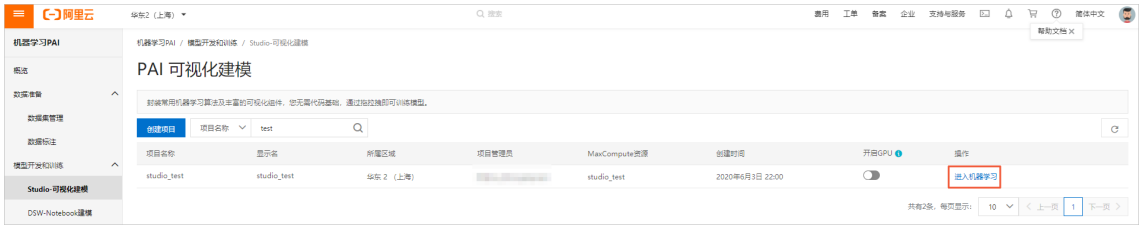
    > **说明** 运行中的专属资源组进行扩容或缩容时，会有中间状态（扩容或缩容中），该状态下系统也会对已占用的资源收费。

  - 部署在公共资源组的服务处于**运行中**，系统计费。

    > **说明** 已部署的服务进行扩容时，会有中间状态（等待），该状态下系统也会对已占用的资源收费。

如何停止正在计费的项目？

- 停止PAI-Studio可视化建模计费项目：
  i. 登录PAI控制台。
  ii. 在左侧导航栏，选择**模型开发和训练 > Studio-可视化建模**。
  iii. 在**PAI可视化建模**页面，单击**进入机器学习**。



  iv. 在PAI-Studio控制台的左侧导航栏，单击**实验**。
  v. 打开待停止的实验，并单击画布上方的**停止**。
- 停止PAI-DSW Notebook建模计费项目：
  - 登录PAI控制台。
  - 在左侧导航栏，选择**模型开发和训练 > DSW-Notebook建模**。
  - 在**Notebook建模服务**页面，单击待停止实例**操作**列下的**停止**。

    实例停止后，其**状态**变为**停止**，此时对于后付费实例，系统停止计费。退出PAI-DSW时，确保您的实例处于**停止**状态，否则可能产生不必要的费用。
- 停止PAI-EAS模型在线服务计费项目：

- 停止运行在后付费专属资源组中的项目：

  将专属资源组的服务器数量降为0，即可停止计费。具体操作如下：

  a. 在**PAI EAS 模型在线服务**页面的资源组区域，单击后付费资源组下的**缩容/停用**。

  b. 确认待释放的资源组无需再使用后，单击**缩容/停用**。

  c. 在**降配**页面，配置**机器数量**为0，并选中**PAI-EAS后付费服务协议**复选框，其他参数使用默认值。

  d. 单击**立即购买**。

- 停止运行在公共资源组中的项目：

  在**PAI EAS 模型在线服务**页面的服务列表区域，单击待停止服务**操作**列下的**停止**，即可停止模型服务和计费。

  > **说明** 请确保被停止的服务不需要再使用，以免造成不必要的业务损失。

如何查询扣款项及明细？

在**费用账单**页面，可以通过筛选查看机器学习的账单明细，详情请参见查看账单与用量明细。其中**产品明细**列表示该费用产生于哪个子产品模块，取值的含义如下：

- **PAI-EAS资源组后付费**：PAI-EAS后付费专属资源组产生的费用。
- **机器学习（PAI）**：该费用包括了PAI-Studio实验训练产生的费用、PAI-DSW后付费实例产生的费用及PAI-EAS公共资源组部署服务产生的费用。需要根据**计费项**区分具体的费用，对应关系如下。

产品明细	计费项	实例ID	费用来源
机器学习（PAI）	使用量	○ text_analysis ○ data_analysis ○ data_manipulation ○ deep_learning ○ default	PAI-Studio实验训练产生的费用
机器学习（PAI）	DSW_CPU_Large使用量	无	PAI-DSW后付费实例产生的费用
机器学习（PAI）	EAS CPU使用量	无	PAI-EAS公共资源组部署服务产生的费用

- **PAI-EAS预付费**：PAI-EAS预付费专属资源组产生的费用。
- **机器学习预付费**：PAI-DSW中购买计算资源的预付费费用。

为什么停止计费项目后仍有扣费？

阿里云并非即时扣费，需要在出账后进行扣费，而实际出账与用户的前台操作存在时延。例如10:00-11:00使用的资源费用，可能在几小时后才会出账扣款。因此即使在11:00执行了停止操作，仍会在几小时后收到扣款信息，造成“多收费”的错觉。实际上，只是收费延迟了，并未多收取费用。

7.TensorFlow模型如何导出为SavedModel

本文为您介绍如何将TensorFlow模型导出为SavedModel格式。

SavedModel格式

使用PAI-EAS预置官方Processor将TensorFlow模型部署为在线服务，必须先将模型导出为官方定义的SavedModel格式（TensorFlow官方推荐的导出模型格式）。SavedModel模型格式的目录结构如下。

```
assets/
variables/
  variables.data-00000-of-00001
  variables.index
saved_model.pb|saved_model.pbtxt
```

其中：

- `assets` 表示一个可选目录，用于存储预测时的辅助文档信息。
- `variables` 存储tf.train.Saver保存的变量信息。
- `saved_model.pb` 或 `saved_model.pbtxt` 存储MetaGraphDef（存储训练预测模型的程序逻辑）和SignatureDef（用于标记预测时的输入和输出）。

导出SavedModel

使用TensorFlow导出SavedModel格式的模型请参见Saving and Restoring。如果模型比较简单，则可以使用如下方式快速导出SavedModel。

```
tf.saved_model.simple_save(
 session,
 "./savedmodel/",
 inputs={"image": x}, ## x表示模型的输入变量。
 outputs={"scores": y} ## y表示模型的输出。
)
```

请求在线预测服务时，请求中需要指定模型signature_name，使用 `simple_save()` 方法导出的模型中，signature_name默认为**serving_default**。

如果模型比较复杂，则可以使用手工方式导出SavedModel，代码示例如下。

```
print 'Exporting trained model to', export_path
 builder = tf.saved_model.builder.SavedModelBuilder(export_path)
 tensor_info_x = tf.saved_model.utils.build_tensor_info(x)
 tensor_info_y = tf.saved_model.utils.build_tensor_info(y)
 prediction_signature = (
   tf.saved_model.signature_def_utils.build_signature_def(
     inputs={'images': tensor_info_x},
     outputs={'scores': tensor_info_y},
     method_name=tf.saved_model.signature_constants.PREDICT_METHOD_NAME))
 legacy_init_op = tf.group(tf.tables_initializer(), name='legacy_init_op')
 builder.add_meta_graph_and_variables(
   sess, [tf.saved_model.tag_constants.SERVING],
   signature_def_map={
     'predict_images':
       prediction_signature,
   },
   legacy_init_op=legacy_init_op)
 builder.save()
 print 'Done exporting!'
```

其中：

- `export_path` 表示导出模型的路径。
- `prediction_signature` 表示模型为输入和输出构建的SignatureDef，详情请参见SignatureDef。示例中的signature_name为predict_images
- `builder.add_meta_graph_and_variables` 方法表示导出模型的参数。

> **说明**
> - 导出预测所需的模型时，必须指定导出模型的Tag为tf.saved_model.tag_constants.SERVING。
> - TensorFlow导出SavedModel格式模型的完整代码请参见saved_model.tar.gz，有关TensorFlow模型的更多信息，请参见TensorFlow SavedModel。

Keras模型转换为SavedModel

使用Keras的 `model.save()` 方法会将Keras模型导出为H5格式，需要将其转换为SavedModel才能进行在线预测。您可以先调用 `load_model()` 方法加载H5模型，再将其导出为SavedModel格式，代码示例如下。

```
import tensorflow as tf
with tf.device("/cpu:0"):
  model = tf.keras.models.load_model('./mnist.h5')
  tf.saved_model.simple_save(
    tf.keras.backend.get_session(),
    "./h5_savedmodel/",
    inputs={"image": model.input},
    outputs={"scores": model.output}
  )
```

Checkpoint转换为Savedmodel

训练过程中使用 `tf.train.Saver()` 方法保存的模型格式为checkpoint，需要将其转换为SavedModel才能进行在线预测。您可以先调用 `saver.restore()` 方法将Checkpoint加载为tf.Session，再将其导出为SavedModel格式，代码示例如下。

```
import tensorflow as tf
# variable define ...
saver = tf.train.Saver()
with tf.Session() as sess:
 # Initialize v1 since the saver will not.
  saver.restore(sess, "./lr_model/model.ckpt")
  tensor_info_x = tf.saved_model.utils.build_tensor_info(x)
  tensor_info_y = tf.saved_model.utils.build_tensor_info(y)
  tf.saved_model.simple_save(
    sess,
    "./savedmodel/",
    inputs={"image": tensor_info_x},
    outputs={"scores": tensor_info_y}
  )
```