# DeepLabV3模型自验报告

姓名： 蒲建伟

单位： 广东南方数码科技有限公司

邮箱： [jianwei.pu@southgis.com](mailto:jianwei.pu@southgis.com)

# 模型简介

本次基线实验选用经典的语义分割模型-DeepLabV3，因为该模型结构简单、高效，并且容易部署。为了解决语义分割中目标多尺度问题，该模型通过设计空洞卷积空间金字塔结构来获取多尺度特征，增加了模型感受野从而有效地提升了性能。下面针对DeepLabV3模型结构和基于MindSpore框架进行训练、测试与评估进行详细说明。

## 网络模型结构简介：

DeepLabV3模型结构主要分为两部分：1）使用CNN提取特征，本文使用的CNN网络是ResNet50,为了提升CNN的感受野，Deeplabv3使用了Multi-Grid策略，即在ResNet50网络的后端将不同卷积层替换为空洞卷积； 2）使用空洞卷积空间金字塔模块来获得丰富的上下文信息，该模块以并行的方式将不同扩张率的空洞卷积组合在一起，形成多尺度特征。具体地，一个卷积层和3个不同扩张率的空洞卷积层，每个卷积层通道数为256且后跟一个BN层，其中1x1卷积层用于获取图像的全局特征，不同扩张率的空洞卷积用于捕获多尺度信息，其扩张率分别为（6,12,18）。

## 数据集：

数据为1米分辨率的高分遥感影像，由广东南方数码科技股份有限公司采集、标注、构建；总共有5万张遥感影像耕地样本数据。其中训练集为2万张遥感影像和标注文件（公开），测试集为3万张遥感影像（非公开），共有2个类别，0表示为非耕地，255表示为耕地。

## 代码提交地址：

无

## 其它：

本次提供的代码是在Ascend910-ARM:24核96GB环境下进行训练及预测的.

# 代码目录结构说明

## 代码目录

├── southgis-train //创建的桶

├── deeplabv3 // 训练模型

├── README.md // 所有模块的说明

├── src

│ ├──data

│ ├──build\_seg\_data.py // 数据集的格式转换

│ ├──dataset.py // 数据集的生成

│ ├──loss

├──loss.py // 梯度损失函数

│ ├──nets

│ ├──deeplab\_v3

│ ├──deeplab\_v3.py // deeplabv3模型

│ ├──net\_factory.py // deeplabv3模型的map

│ ├──tools

│ ├──get\_multicards\_json.py // 生成模型需要的json格式

│ ├──utils

│ ├──learning\_rates // 学习率函数生成

├──data\_prepro.py // 数据预处理

├──mydataset.py // 自定义数据集，可放在data中

├──train.py // 训练模型

├──test.py // 测试训练完的模型

├──myeval.py // 对测试的数据进行评估

## 上传数据目录

在OBS Browser中上传数据及需要创建的目录说明：

在整体建立的桶中需要用到的目录：创建的桶southgis-train中包括以下目录及数据存储

├── southgis-train //创建的桶

├── deeplabv3 //训练模型

├── gis // 存储所有数据的目录

├── labels // 存储原始标记的样本（初步准备数据，预处理后可删除）

├── label // 存储预处理后的数据样本（不需要准备）

├── image // 存储训练的原始数据（初步准备数据）

├── log // 运行结果日志总目录

├──data\_prepro\_log // 存储运行完成后的结果 ，根据需要创建日志目录

├── output\_train // 保存训练后的模型，在预测时可选择最高精度模型进行预测

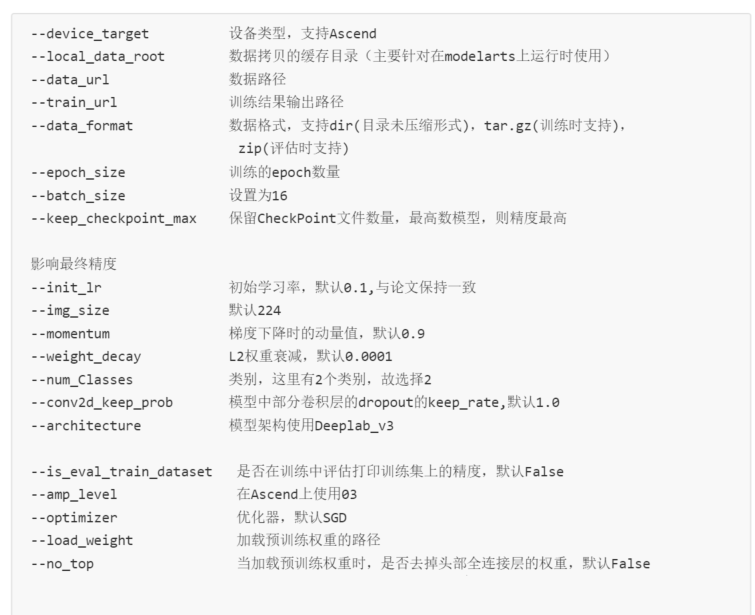
# 自验结果

## 自验环境：

华为云ModelArts平台下的Ascend910-ARM:24核96GB，mindspore1.0框架

## 训练超参数：

可通过train.py脚本中的参数修改训练行为。 train.py脚本中的参数如下：



## 数据预处理，模型训练：

### 前期准备工作：

1. 进入华为云[网站](https://bbs.huaweicloud.com/blogs/226041)，点击登录->点击控制台->点击服务列表->选择对象存储服务OBS->点击进入对象存储服务OBS->点击下载OBS Browser+，如图所示。



1. 进行桶的创建，点击右上角创建桶->区域选择北京四->数据冗余存储策略选择多AZ存储->按要求设置桶名称->桶策略选择私有->点击立即创建，如图所示。



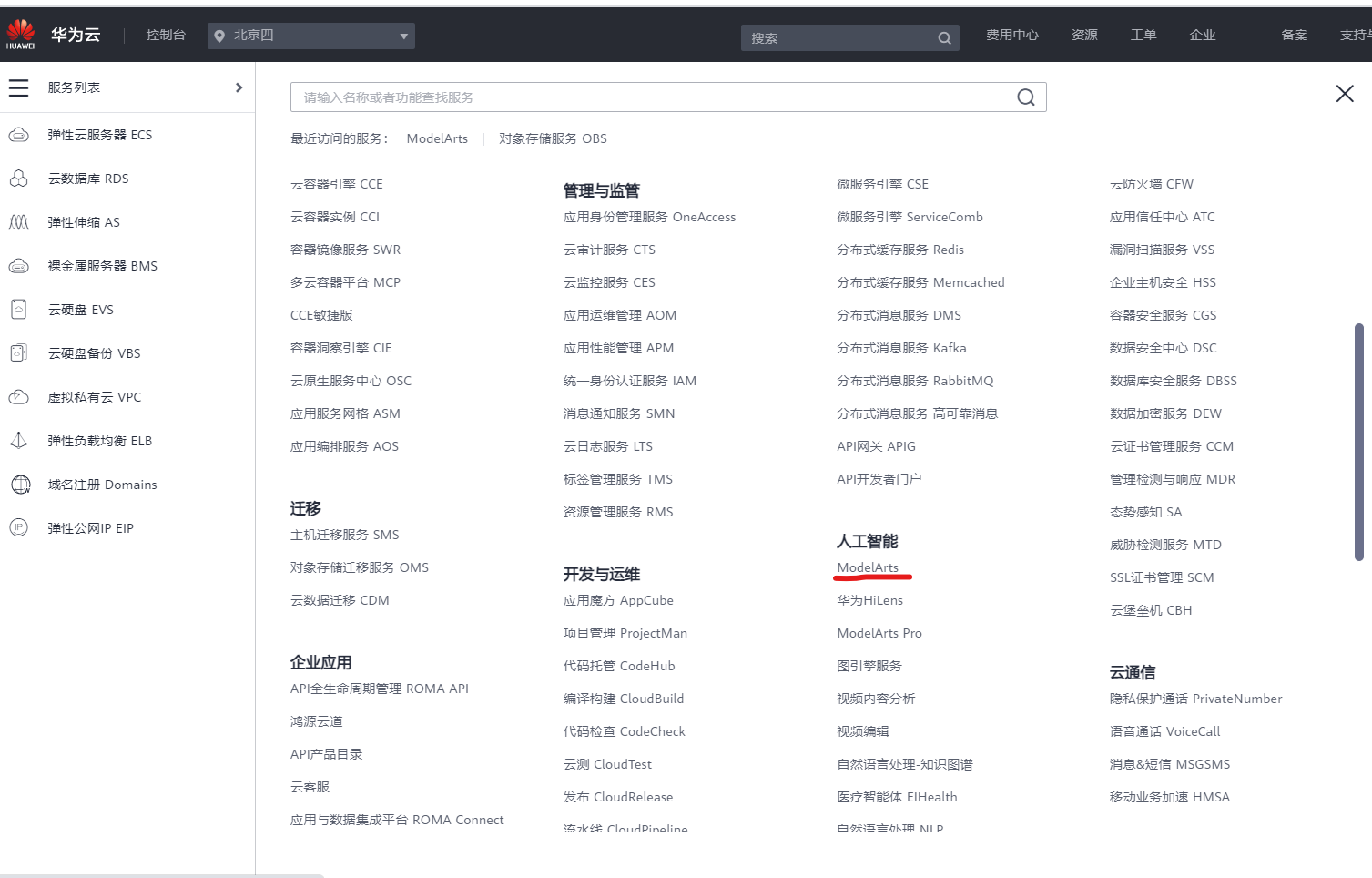
1. 登录下载好的OBS Browser+：然后可根据2.2讲述的上传数据目录完成目录创建，训练模型及准备的数据的上传，保证训练模型上传成功，可进行下一步操作。如图所示。



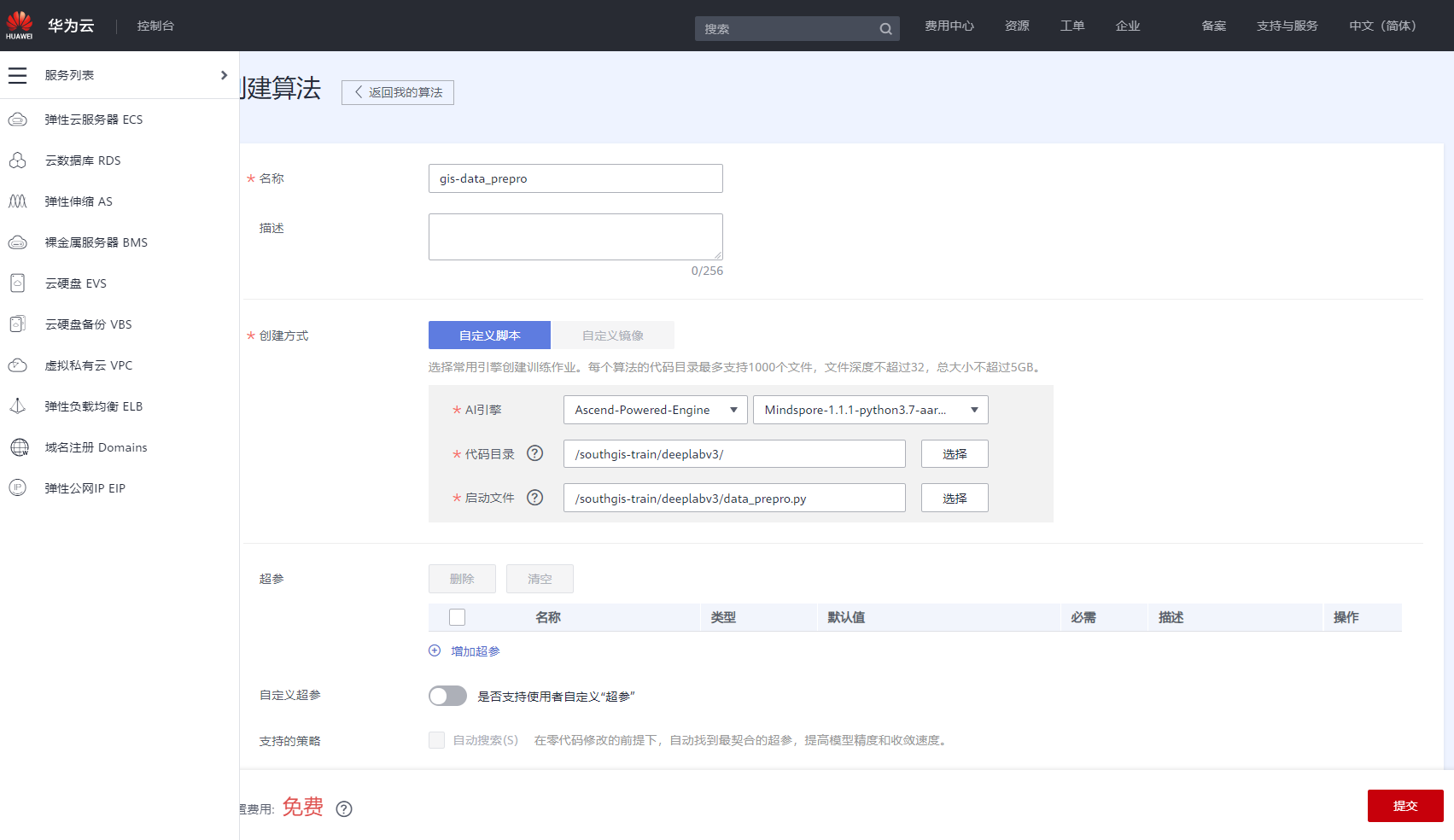
### 算法创建：

具体创建流程如下：

1.进入控制台界面->服务列表，选择ModerArts服务，点击进入，如图所示。



1. 点击算法管理->我的算法->点击创建，进入算法创建界面，填写相应的信息->点击提交即可，需要注意的部分已列出，请详读。如图所示。



注意：AI引擎：必须选择Ascend，版本选择默认的Mindspore1.1.1-python3.7版本；

代码目录：选择上传的训练模型DeepLabv3目录；

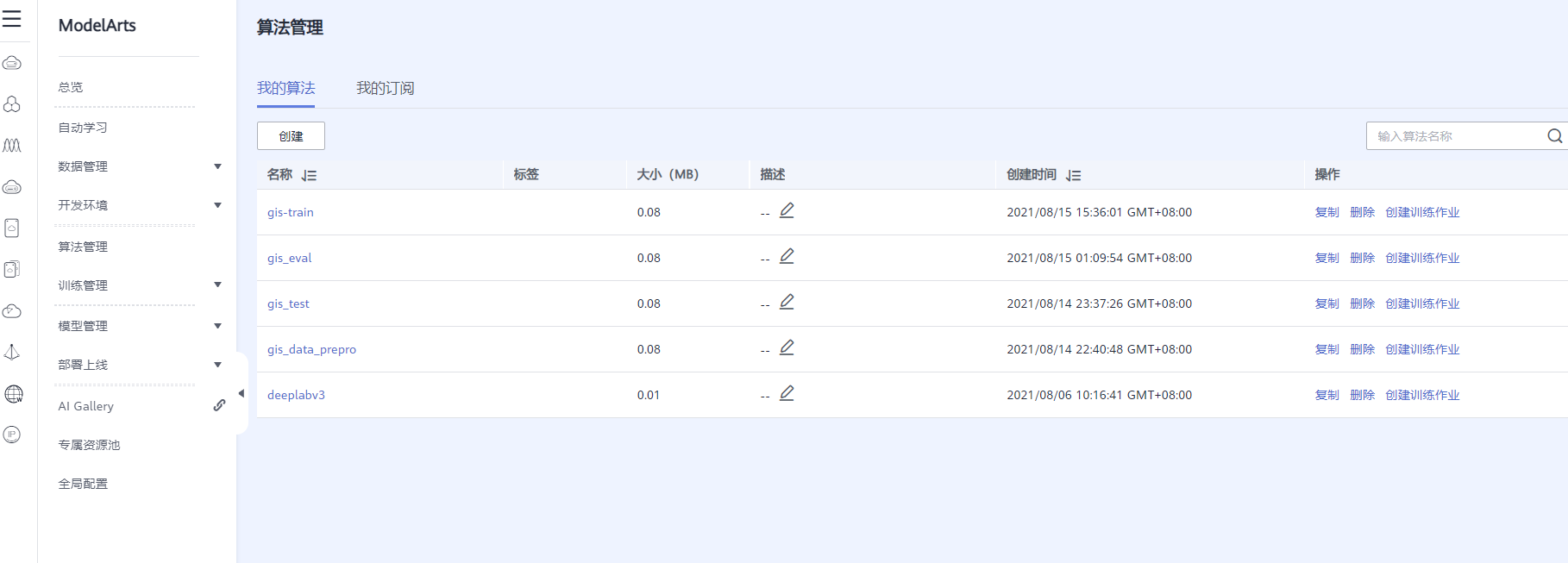
启动文件：选择需要执行的文件，此处如果对数据没有处理，可先选择数据预处理执行文件data\_prepro.py，如果已经处理完成的，可选择训练文件train.py.

数据预处理文件讲解：本公司提供的数据为训练数据2w张，和标注数据2w张，其中需要处理的数据是将2w张标注数据labels(0,255)处理为2w张标注数据label(0,1),处理完成后可进行训练操作。

1. 如果创建了数据预处理启动文件算法，还需要创建训练启动文件算法，train.py，步骤和步骤4中讲述的一致，如图所示。



1. 算法创建完成，根据自己的需求进行创建，如果创建错误，可点击删除，如图所示。



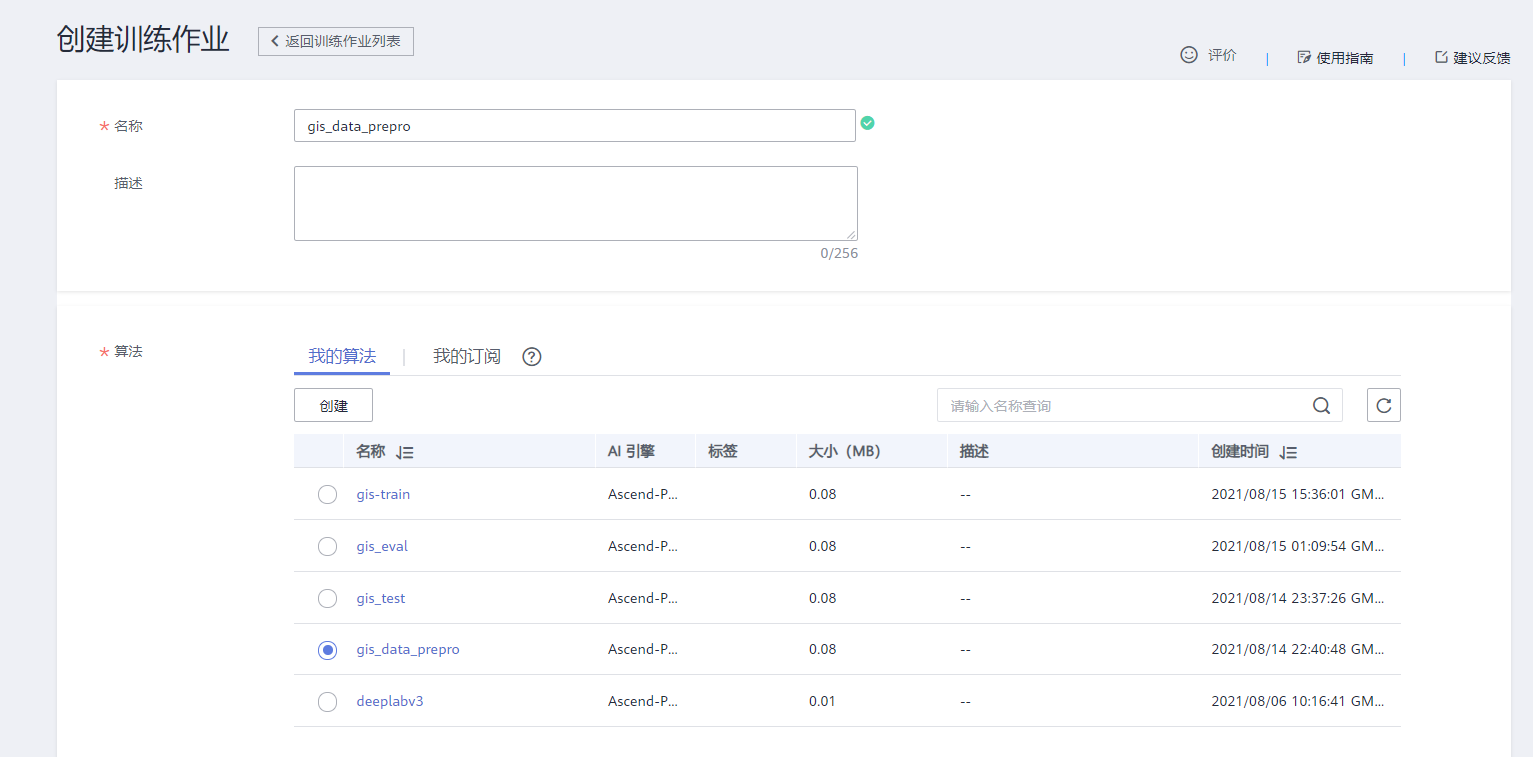
### 如何启动训练脚本：

**（1）Ascend进行数据预处理执行流程如下**：

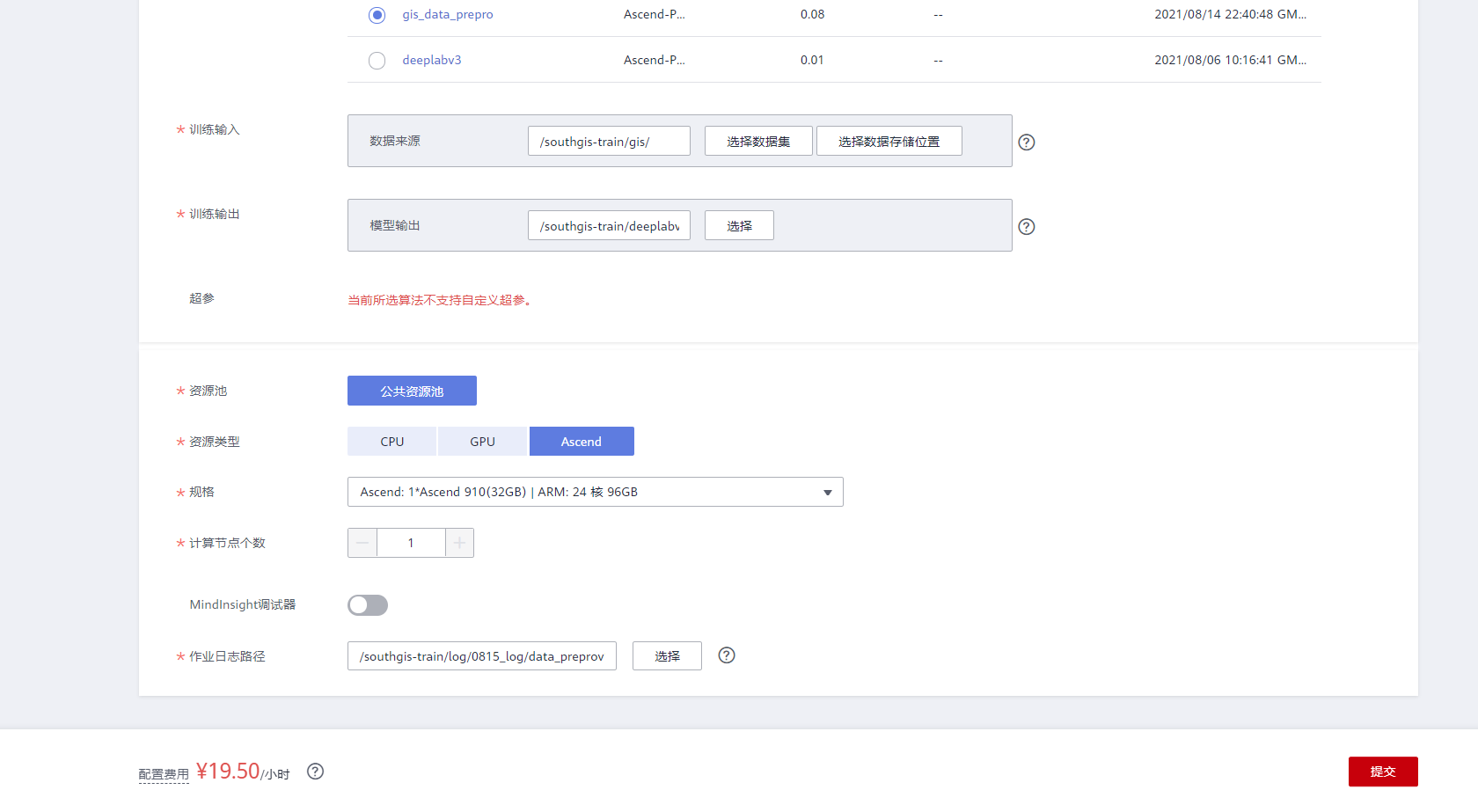
1. 点击训练管理->点击训练作业->点击创建训练作业，如图所示。



2.进入创建训练作业->填写训练名称->选择需要运行的启动文件算法(这里选择预处理算法文件)，如图所示。



3.选择数据来源（这里选择存储数据集目录gis）->训练模型目录（这里选择deeplabv3）->资源类型选择Ascend->规格选择24核96GB即可->计算节点数为1->选择运行后的结果日志路径（根据自己的需求进行创建即可）->点击提交->点击确定，进行训练，训练前期需要等待大约8分钟才可执行文件进行运行。



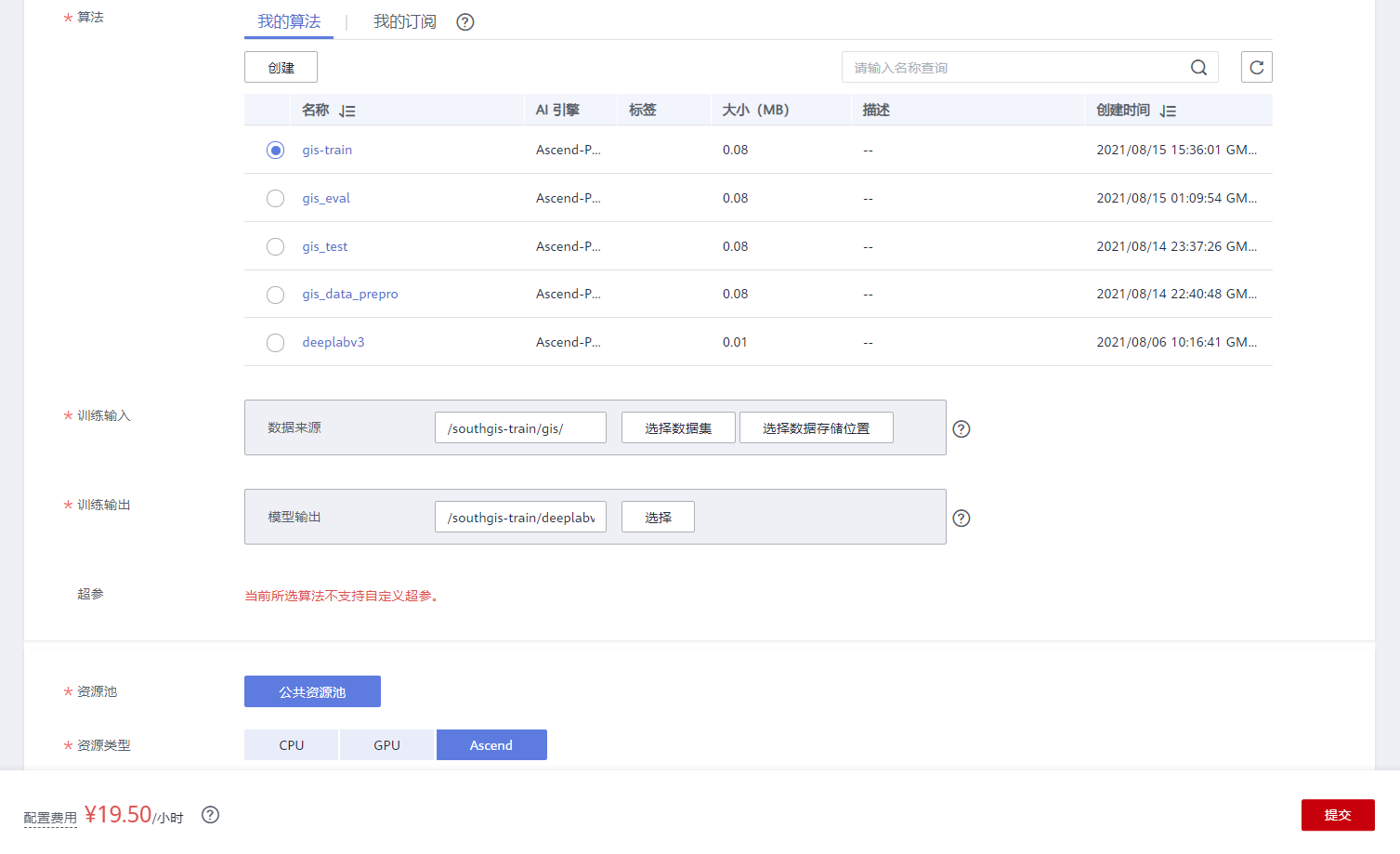
4.进入训练界面，如果运行错误，请查看日志，很有可能是执行文件中的路径未写正确。如图所示。



注意：如果预处理完成后，则在gis/label目录下可以看到预处理后的数据。

**（2）Ascend进行数据训练执行流程如下：**

1. 步骤与上边操作相同，进入训练管理->训练作业->点击创建训练作业->完成训练名称->选择我的算法（此时选择需要训练的启动文件算法）-> 选择数据存储目录->选择训练模型目录->选择Ascend->选择Ascend-24核96GB->选择需要存储的训练日志目录->点击提交->点击确定。如图所示。



训练20个epoch需要80分钟。注意，训练完成后，则在southgis-train/output\_train目录下会存储相应的训练模型

**（2）Ascend进行预测与评估执行流程如下：（此处讲解仅提供操作步骤,参赛人员可以省略）**

1.预测需要的数据及需要的数据目录：

├── gis // 存储所有数据的目录

├──test\_labels // 存储原始标记的预测数据样本（初步准备数据）

├──test\_label // 存储预处理后的预测数据样本（不需要准备）

├──test // 存储训练的原始数据（初步准备数据）

├──pre // 存储预测数据（不需要数据准备）

├──pre\_test // 存储预测可视化数据，可与test\_labels人工做比对

├── log // 运行结果日志总目录

├──test\_log //日志目录

├──eval\_log //行预测

2..对提供的预测的数据test\_labels做数据预处理，具体流程与6.1.1-(1)相同，但需要注意的是，在预处理deeplabv3/data\_prepro.py中修改相应的执行路径。

2.执行测试，需要创建算法->启动文件选择deeplabv3/test.py，详见3.3.2。然后进入训练管理->进入训练学习->填写相应的信息(选择刚才创建的测试算法)->点击提交进行测试训练，详见6.1.1-(2)。

注意：在执行文件deeplabv3/mydataset.py和自定义数据集deeplabv3/mydataset.py中需要查看训练路径是否正确，如不正确，可按照相关路径修改完成即可。测试完成后，在git/pre与git/pre\_test中均会生成相应的测试数据。

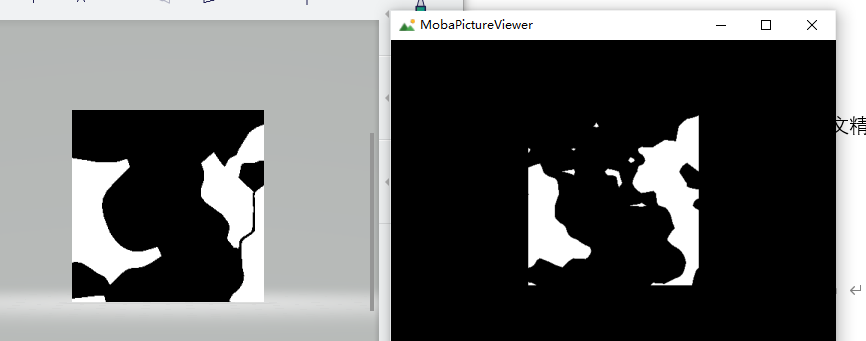
3.对测试后的数据进行评估->首先仍然需要创建算法->启动文件选择deeplabv3/myeval.py，详见3.3.2。然后进入训练管理->进入训练学习->填写相关信息(选择刚才创建的评估算法)->点击提交进行评估，评估完成后->进入选择的日志log/eval\_log中进行查看评估结果。

### 预测精度结果：

训练后对模型进行预测评估，评估结果如下：



效果图如图所示，左侧为真实图像的语义分割，右图为预测的语义分割图像：



# 参考资料

## 参考论文：

参考论文：[Rethinking Atrous Convolution for Semantic Image Segmentation](https://arxiv.org/pdf/1706.05587.pdf)

网址：<https://arxiv.org/pdf/1706.05587.pdf>

## 参考git项目：

<https://www.mindspore.cn/>

<https://gitee.com/mindspore/mindspore/tree/master/model_zoo/official/cv/deeplabv3>