

DataWhale & 趋动云任务四

通过免费的GPU算力，线上训练深度学习模型。

DataWhale & 趋动云任务四

创建项目

1. 填写项目名称
2. 选择镜像
3. 添加数据
4. 创建

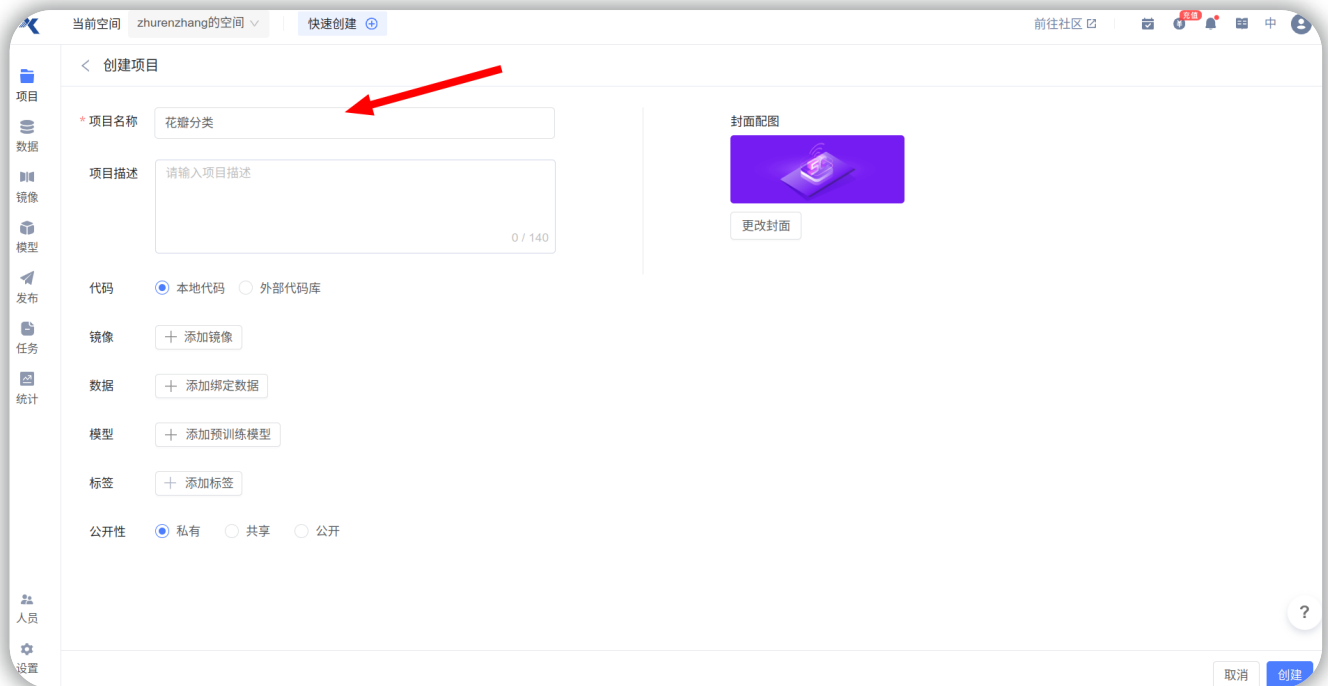
训练

预测

H3 创建项目

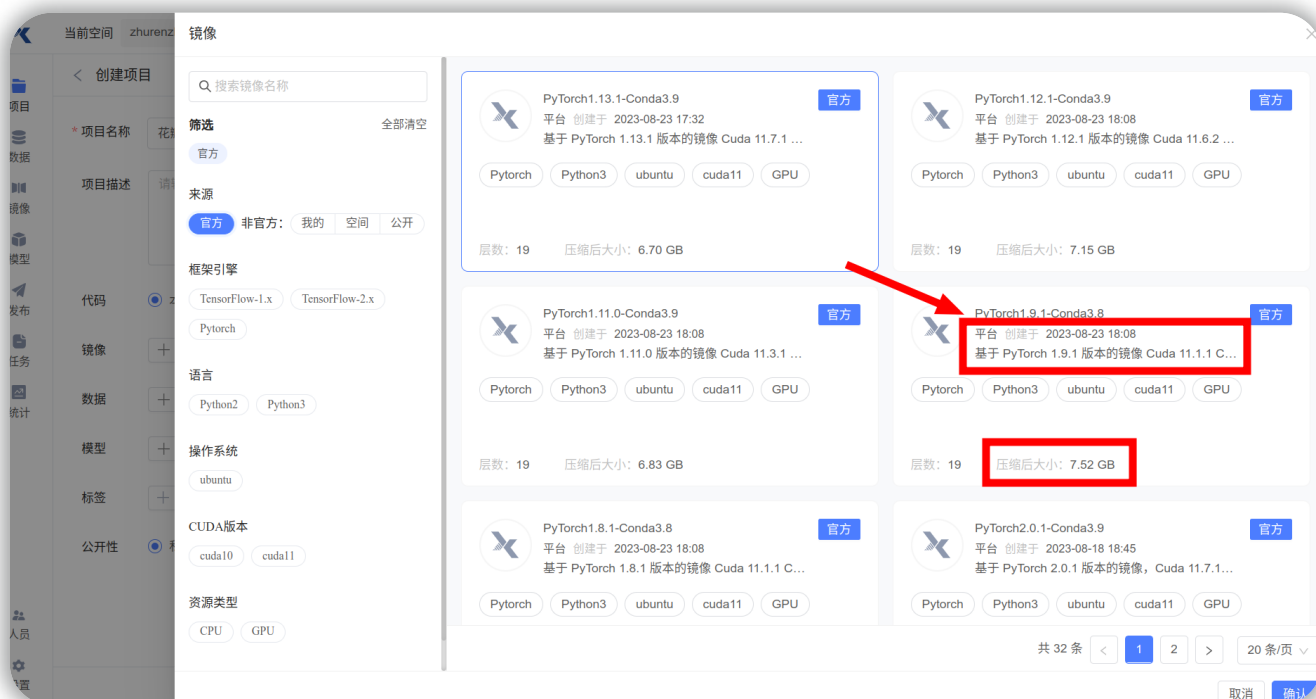
1. 填写项目名称

输入“花瓣分类”，也可以自定义名称



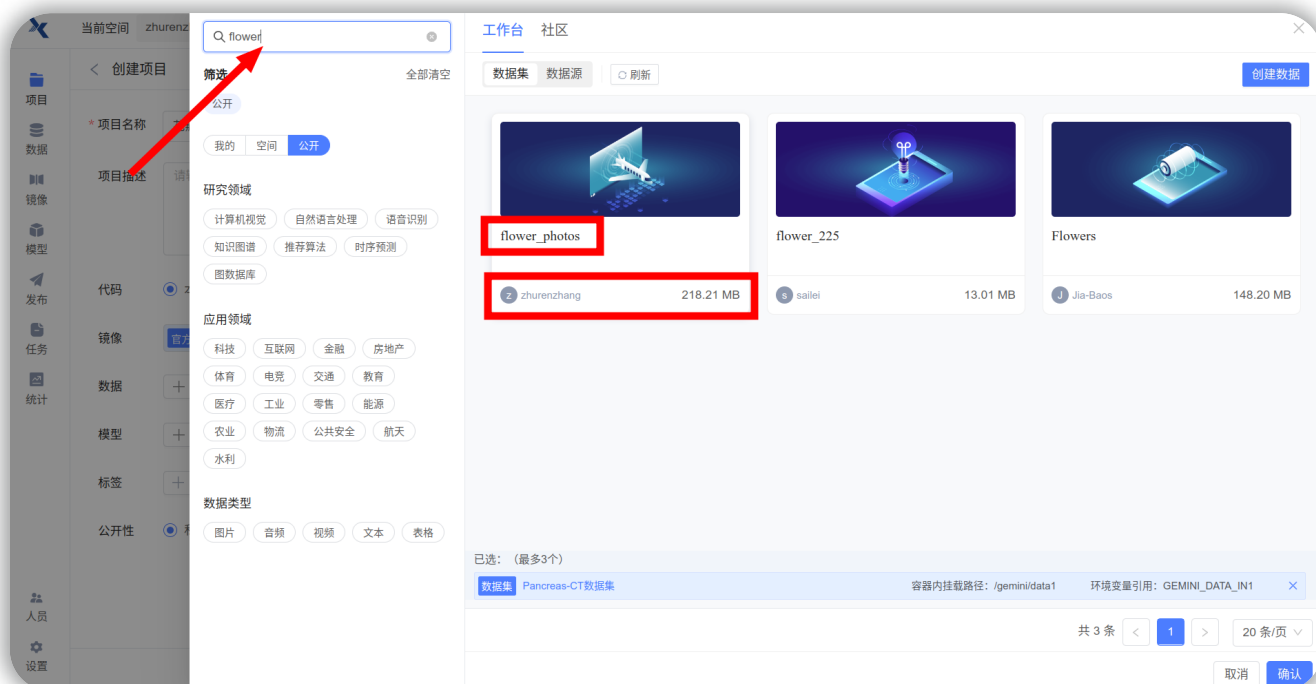
2. 选择镜像

官方镜像中，选择pytorch1.9.1和conda3.8的镜像即可。（结合镜像的创建时间，别选错了）



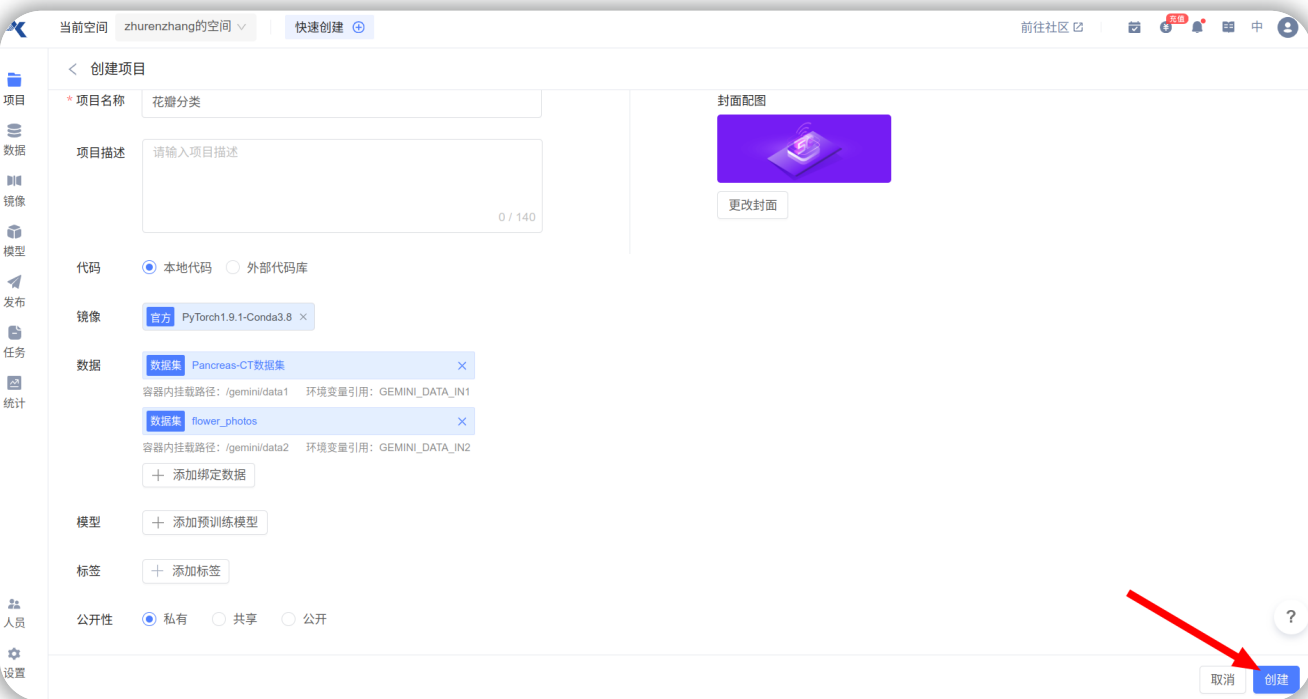
3. 添加数据

选择“公开”，同时搜索“flower”，找到这个218.21MB的数据集，选择并确定。



4. 创建

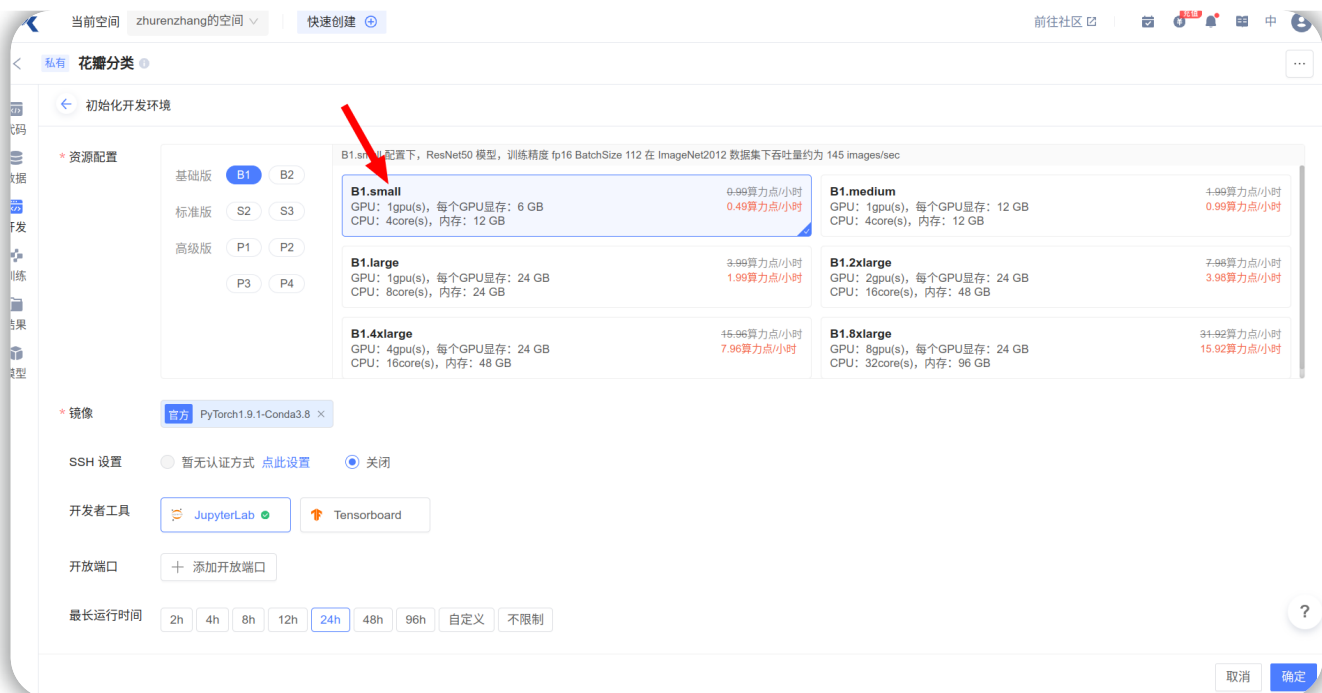
点击“确定”之后，弹出窗口，选择“代码暂不上传”。



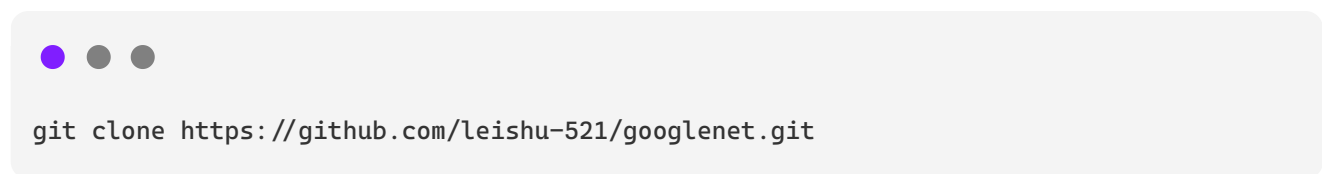
H3 训练

1. 先“运行环境”，选择资源“B1.small”：GPU：6GB，内存：12GB够用了





2. 进入开发环境，拉取代码。（由于网络原因，如果拉取失败，多试几次即可）



当前空间zhurenzhang的空间快速创建

花瓣分类JupyterLab网页终端通过代理源加速安装包过程

全屏

将当前环境制作作为镜像提交训练任务

用户须知：请提前阅读本文档

| 存储 | 路径 | 环境变量 (\$+变量) | 权限 | 大小 | 备注 |
|------|--|---|------------------------|---|---|
| 容器储存 | 代码、数据集、模型、结果集路径以外的所有路径 | none | 【开发环境】可读写 【离线训练】可读写 | small: 20G midium: 30G large: 50G xlarge: 100G | 1. [临时保存]把容器保存成新镜像 (包含容器存储的数据) 2. 容器关闭或重启, 会销毁容器, 容器存储的数据不会保留 |
| 代码 | /gemini/code | GEMINI_CODE | 【开发环境】可读写 【离线训练】只读 | 不限制大小 | 1. 在项目内挂载, 归属于所在的项目 2. 启动容器后, 如果开启了SSH或注入了JupyterLab, 可以通过SSH工具或JupyterLab上传下载 |
| 数据集 | /gemini/data-1 /gemini/data-2 /gemini/data-3 | GEMINI_DATA_IN1 GEMINI_DATA_IN2 GEMINI_DATA_IN3 | 只读 Read Only | 不限制大小 | 在【数据】栏内上传数据, 保存在数据目录下, 创建项目时选择会挂载到容器内 |
| 模型 | /gemini/pretrain /gemini/pretrain2 /gemini/pretrain3 | GEMINI_PRETRAIN GEMINI_PRETRAIN2 GEMINI_PRETRAIN3 | 只读 Read Only | 不限制大小 | none |
| 结果集 | /gemini/output | GEMINI_DATA_OUT | 仅【离线训练】有此功能可读写 | 不限制大小 | 挂载在项目内, 归属于所在的项目 |

(base) root@379630697427505152-taskrole1-0:/gemini/code# git clone https://github.com/leishu-521/googlenet.git
Cloning into 'googlenet'...

环境信息

环境状态: 运行中

资源配置

GPU 型B1.small
GPU: 1gpu(s), 每个GPU显存: 6 GB
CPU: 4core(s), 内存: 12 GB

工作配置

工作镜像: 官方 PyTorch1.9.1-Conda3.8

数据集: [Pancreas-CT数据](#) 复制
[flower_photos](#) 复制

关联代码: latest

环境变量: 查看

不建议额外安装其它版本的框架和 Cuda, 版本不兼容时将引起 GPU 不可用。[查看兼容信息](#)

端口信息

暂未添加

性能监控

实时数据

3. 开始训练，等待一段时间，让它训练

#进入根目录

cd googlenet

#开始训练

python train.py

googlenetJupyterLab网页终端通过代理源加速安装包过程

全屏

将当前环境制作作为镜像提交训练任务

[epoch 20] train_loss: 0.915 val_accuracy: 0.822
train epoch[21/30] loss:0.145: 100%
[epoch 21] train_loss: 0.824 val_accuracy: 0.820
train epoch[22/30] loss:1.400: 100%
[epoch 22] train_loss: 0.839 val_accuracy: 0.808
train epoch[23/30] loss:1.768: 100%
[epoch 23] train_loss: 0.882 val_accuracy: 0.818
train epoch[24/30] loss:0.312: 100%
[epoch 24] train_loss: 0.839 val_accuracy: 0.785
train epoch[25/30] loss:0.129: 100%
[epoch 25] train_loss: 0.769 val_accuracy: 0.804
train epoch[26/30] loss:1.631: 100%
[epoch 26] train_loss: 0.793 val_accuracy: 0.788
train epoch[27/30] loss:0.196: 100%
[epoch 27] train_loss: 0.756 val_accuracy: 0.799
train epoch[28/30] loss:0.882: 100%
[epoch 28] train_loss: 0.776 val_accuracy: 0.759
train epoch[29/30] loss:3.810: 100%
[epoch 29] train_loss: 0.780 val_accuracy: 0.820
train epoch[30/30] loss:0.688: 100%
[epoch 30] train_loss: 0.798 val_accuracy: 0.820
Finished Training

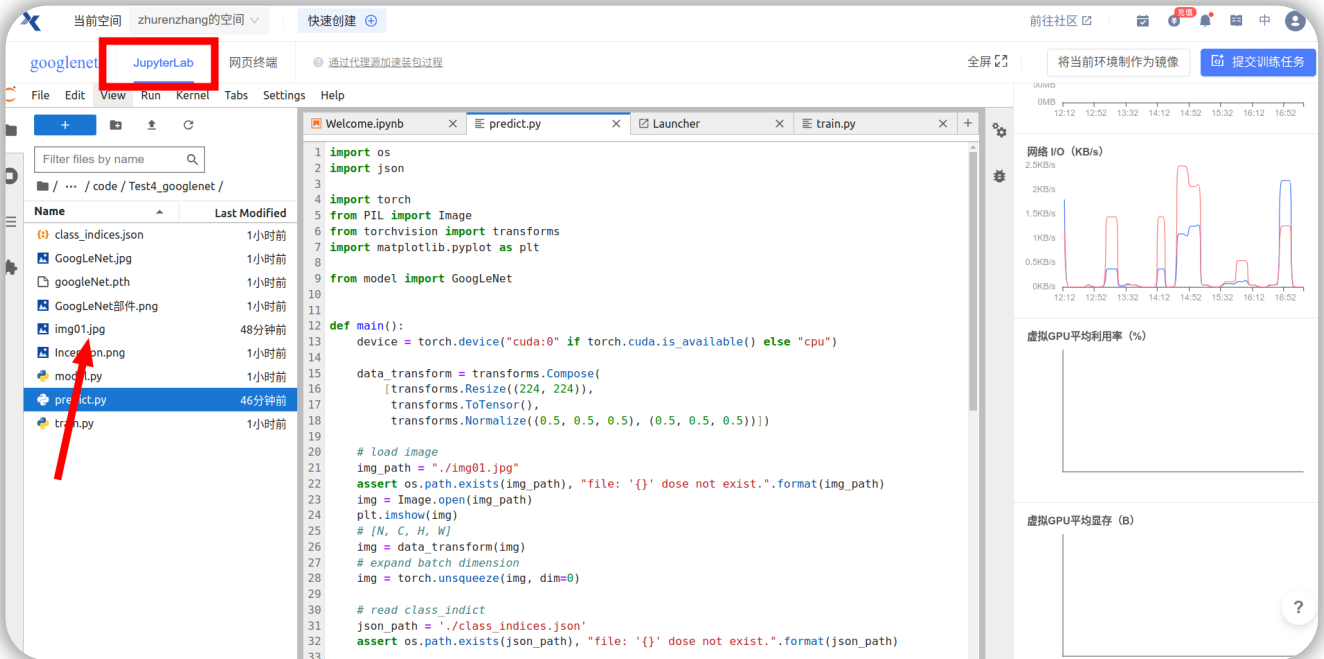
网络 I/O (KB/s)

虚拟GPU平均利用率 (%)

虚拟GPU平均显存 (GB)

H3 预测

- 在网上找一张花的图片，可以用截图工具截取一张240*240的花瓣图片，修改名称：img01.jpg，拖动到代码目录中，如图：



- 预测，得到结果

