

长安大学

高性能计算平台 AI 模块
使用手册

V1.0

目 录

目 录.....	1
第一章 概述与架构.....	4
第二章 主要功能.....	5
2.1. 数据集.....	5
2.1.1. 数值数据.....	5
2.1.1.1. “数值数据-数据文件”数据集.....	5
2.1.1.2. “数值数据-数据库”数据集.....	8
2.1.1.3. “数值数据-HDFS”数据集.....	10
2.1.2. 图像数据.....	13
2.1.2.1. “图像数据-数据文件”数据集.....	13
2.1.2.2. “图像数据-HDFS”数据集.....	15
2.1.3. 其他.....	17
2.1.4. 共享数据集.....	19
2.1.4.1. 将我的数据集共享至共享数据集.....	20
2.1.4.2. 新建共享数据集.....	23
2.1.5. 图像标注.....	26
2.1.5.1. 编辑标签.....	26
2.1.5.2. 导入图像.....	27
2.1.5.3. 图像识别.....	28
2.1.5.4. 目标检测.....	30
2.2. 可视化建模.....	37
2.2.1. 方案设计.....	37
2.2.1.1. 构建方案设计流程.....	38
2.2.1.1.1. 新建.....	38
2.2.1.1.2. 设计.....	39
2.2.1.2. 操作.....	46
2.2.1.2.1. 方案保存.....	46

目录

2.2.1.2.2. 方案另存为.....	46
2.2.1.2.3. 方案运行.....	47
2.2.1.3. 管理.....	49
2.2.1.3.1. 查看描述.....	50
2.2.1.3.2. 删除.....	50
2.2.2. 方案实例.....	51
2.2.3. 模型库.....	54
2.2.3.1. 新建模型库.....	55
2.2.3.2. 模型训练历史列表.....	56
2.2.3.3. 模型卡片.....	60
2.2.3.4. 共享模型.....	61
2.3. 开发中心.....	63
2.3.1. 新建开发环境.....	64
2.3.2. 使用开发环境.....	67
2.3.2.1. 应用 Jupyter 服务.....	67
2.3.2.2. 应用 VSCode 服务.....	70
2.3.2.3. 应用 CentOs 桌面/Ubuntu 桌面	75
2.4. AI 作业提交	79
2.4.1. Tensorflow.....	79
2.4.1.1. 单机模式.....	79
2.4.1.2. PS 并行模式	82
2.4.1.3. Horovod 并行模式	83
2.4.2. Pytorch.....	83
2.4.2.1. 单机模式.....	83
2.4.2.2. Horovod 并行模式	85
2.4.3. Mxnet.....	86
2.4.3.1. 单机模式.....	86
2.4.3.2. Horovod 并行模式	87
2.4.4. PaddlePaddle.....	88

目录

2.4.4.1. 单机单卡.....	88
2.4.4.2. 单机多卡.....	90
2.5. 实验管理.....	90
2.5.1. 准备训练程序.....	91
2.5.2. 执行训练程序.....	92
2.5.2.1. 提交AI作业.....	92
2.5.2.2. 运行模型结构的方案.....	93
2.5.2.3. 从开发中心运行.....	94
2.6. 服务管理.....	97
2.6.1. 添加服务.....	98
2.6.1.1. “终端容器”类型的服务.....	98
2.6.1.2. “Tensorboard”类型的服务.....	99
2.6.1.3. “Tensorflow Serving”类型的服务.....	100
2.6.1.4. “Pytorch Serving”类型的服务.....	102
2.6.2. 重新添加服务.....	103
2.6.3. 查看服务信息.....	105
2.6.4. 应用服务.....	106
2.6.4.1. 访问服务.....	106
2.6.4.2. 保存为镜像.....	107
2.6.5. 删除服务.....	108
附录：常见问题及解决办法.....	110

第一章 概述与架构

人工智能平台提供了丰富的算法和建模框架，不仅提供了用户图形化、拖拽式的建模方式，支持用户“零编码建模”；也提供了开放式接口，支持高级用户自定义算法和模型。用户创建的模型可以用来创建自己的AI场景解决方案，也可以共享给其他用户使用。人工智能软件提供了数据集、模型共享功能，用户可以把自己的数据集和模型共享出去，帮助团队简单、快速地构建模型。人工智能软件还提供了常用的VSCode、Jupyter等Web IDE，灵活扩展用户AI实践过程。

系统的整体功能架构如下图所示：



系统功能架构图

整个系统主要包括Web应用门户、数据集管理、模型管理、方案管理、图像标注等模块。

第二章 主要功能

2.1. 数据集

数据集中包含“我的数据集”和“共享数据集”，在“我的数据集”下可以创建“数值数据”、“图像数据”和“其他”数据集。数据集为用户数据的集合，目前支持“数值数据”数据集、“图像数据”数据集和“其他”数据集的管理。同时支持数据集的共享，让不同用户之间可以相互共享数据。

方案设计中支持以可视化方式引用数据集，可以使用已有的数据集进行数据处理和模型训练。

2.1.1. 数值数据

2.1.1.1. “数值数据-数据文件”数据集

新建“数值数据-数据文件”数据集，操作步骤如下：依次点击“我的数据集” - “数值数据” - “数据文件”，然后在右侧的工作区，点击“新建数据集”卡片按钮，此时会弹出“新建数据集”窗口，如下图所示：



新建“数值数据-数据文件”数据集

图中每个参数的具体含义如下：

第二章

名称: 数据集名称, 输入任意符合命名规则的名称即可。

密级: 管理员开启密级功能后, 显示此选项, 默认为用户密级, 用于数据安全、保密。

选择文件: 点击蓝色的“文件夹图标”按钮, 然后在弹出的“选择文件”窗口中选择文件即可。目前仅支持 csv 格式的数据文件, 支持选择多个文件, 需要保证表头相同, 无表头的数据文件需要保证数据列数相同, 否则会导致新建数据集失败。

字符集: 选择文件的编码格式, 支持以下字符集:

- UTF-8
- GB2312
- GBK
- ISO-8859-1
- UTF-16
- Big5
- UTF-32
- US-ASCII

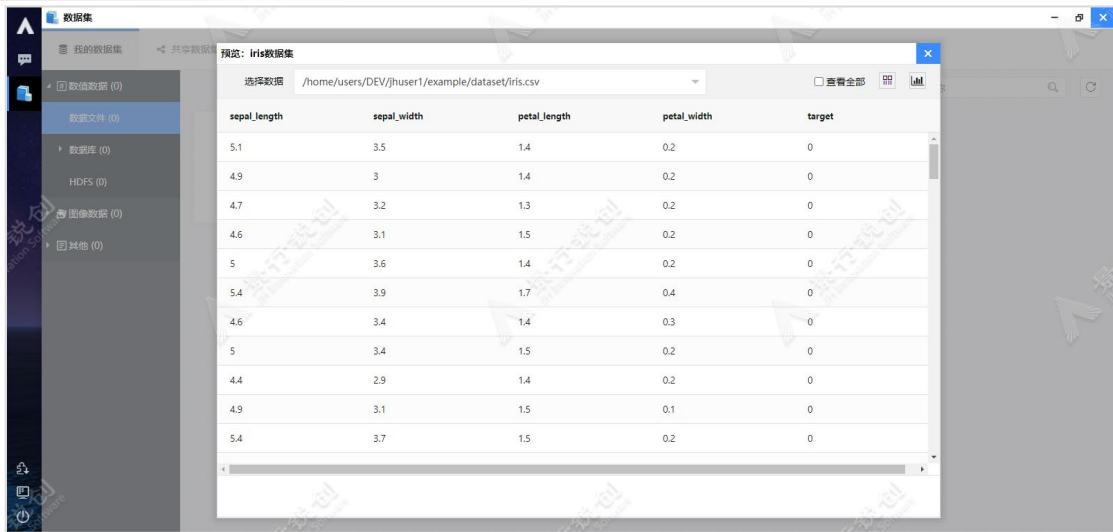
分隔符: 选择文件中使用的分隔符, 支持以下分隔符:

- 逗号
- 空格
- 制表符
- 下划线
- 中划线
- 分号

表头: 选择文件是否包含表头。

描述: 输入该数据集的描述信息, 可选。

填写完成后, 点击“预览”按钮, 即可预览数据集的数据文件内容, 如下图所示:



sepal_length	sepal_width	petal_length	petal_width	target
5.1	3.5	1.4	0.2	0
4.9	3	1.4	0.2	0
4.7	3.2	1.3	0.2	0
4.6	3.1	1.5	0.2	0
5	3.6	1.4	0.2	0
5.4	3.9	1.7	0.4	0
4.6	3.4	1.4	0.3	0
5	3.4	1.5	0.2	0
4.4	2.9	1.4	0.2	0
4.9	3.1	1.5	0.1	0
5.4	3.7	1.5	0.2	0

预览“数值数据-数据文件”数据集

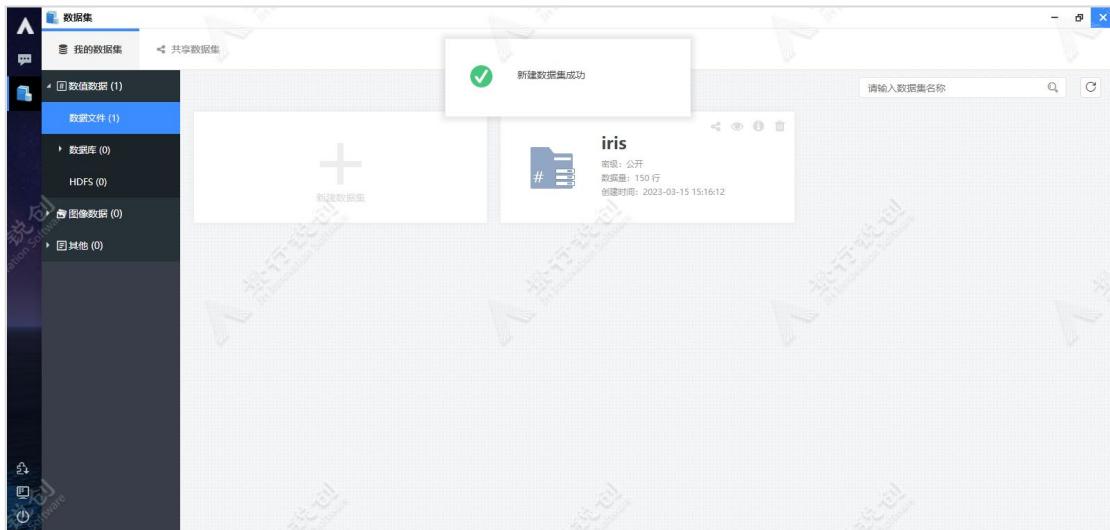
点击右上角的“数据可视化”图标按钮，将预览界面切换至数据可视化界面，如下图所示：



以可视化形式预览数据

点击“确定”按钮，新建数据集，如下图所示：

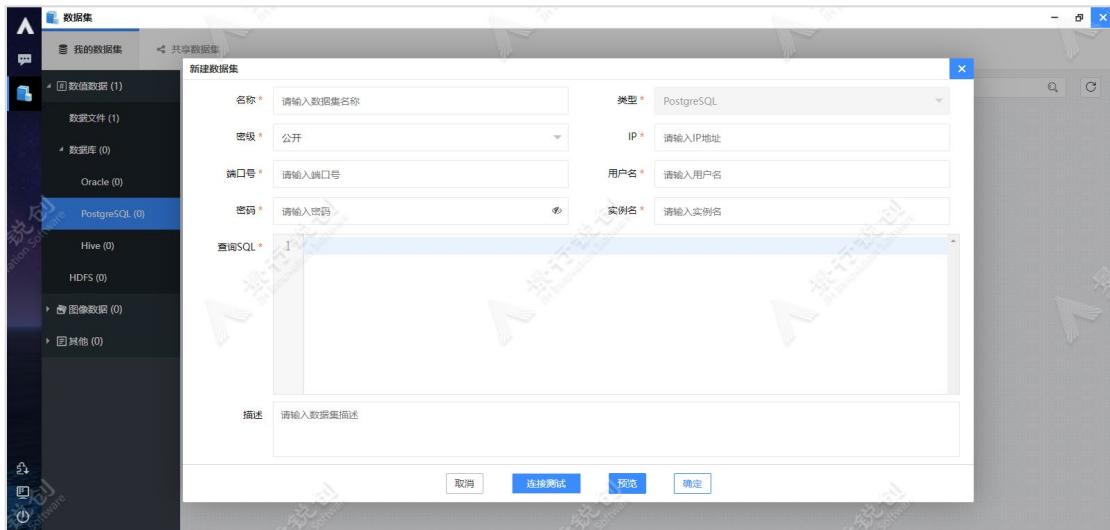
第二章



新建“数值数据-数据文件”数据集成功示意图

2.1.1.2. “数值数据-数据库”数据集

新建“PostgreSQL 数据库”数据集，操作步骤如下：依次点击“我的数据集” – “数值数据” – “数据库” – “PostgreSQL”，然后在右侧的工作区，点击“新建数据集”卡片按钮，此时会弹出“新建数据集”窗口，如下图所示：



新建“数值数据-PostgreSQL 数据库”数据集

图中每个参数的具体含义如下：

名称：数据集的名称，输入任意符合命名规则的名称即可。

类型：选择数据库的类型。

密级: 管理员开启密级功能后, 显示此选项, 默认为用户密级, 用于数据安全、保密。

IP: 输入数据库所在机器的 IP。

端口号: 输入数据库使用的端口号。

用户名: 输入数据库的用户名。

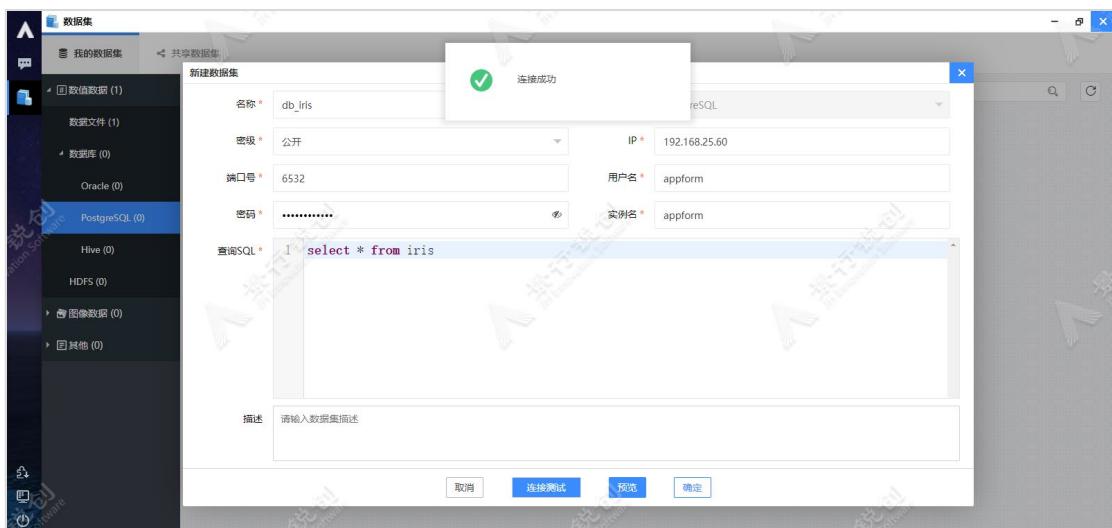
密码: 输入数据库用户对应的密码。

实例名: 输入数据库名。

查询 SQL: 输入 SQL 查询语句(注意: 不要在查询语句后面加分号, eg: select * from tb1)。

描述: 输入该数据集的描述信息, 可选。

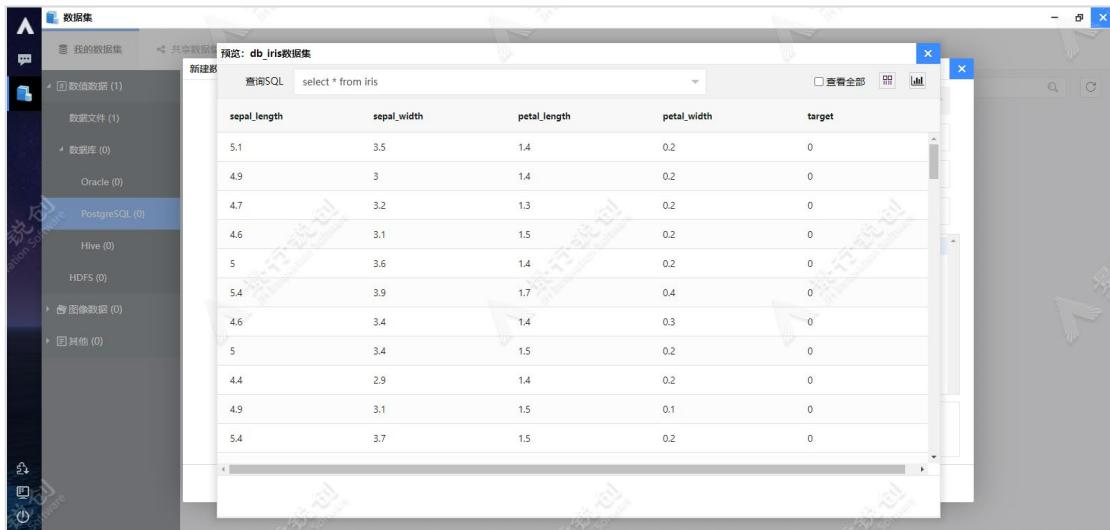
填写完对应信息后, 点击“连接测试”按钮, 即可测试数据库的连通性, 会有成功和失败提示。连接测试效果如下图所示:



数据库连接测试

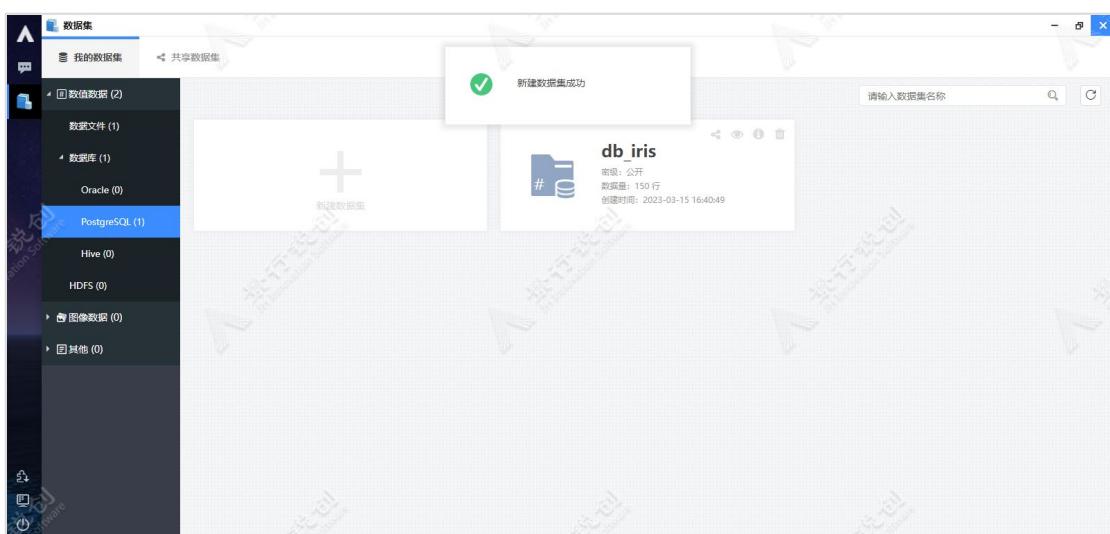
点击“预览”按钮, 即可查看“查询 SQL”的结果。如下图所示:

第二章



预览“数值数据-PostgreSQL 数据库”数据集

点击“确定”按钮，新建数据集。如下图所示：



新建“数值数据-PostgreSQL 数据库”数据集成功示意图

2. 1. 1. 3. “数值数据-HDFS” 数据集

新建“数值数据-HDFS”数据集，操作步骤如下：依次点击“我的数据集” - “数值数据” - “HDFS”，然后在右侧的工作区，点击“新建数据集”卡片按钮，此时会弹出“新建数据集”窗口，如下图所示：



新建“数值数据-HDFS”数据集

图中每个参数的具体含义如下：

名称：数据集的名称，输入任意符合命名规则的名称即可。

密级：管理员开启密级功能后，显示此选项，默认为用户密级，用于数据安全、保密。

IP：输入 HDFS 服务节点的 IP。

端口号：输入 HDFS 服务使用的端口号。

用户名：输入 HDFS 服务的用户名。

全路径：输入数据文件的绝对路径，目前仅支持 csv 格式的数据文件。

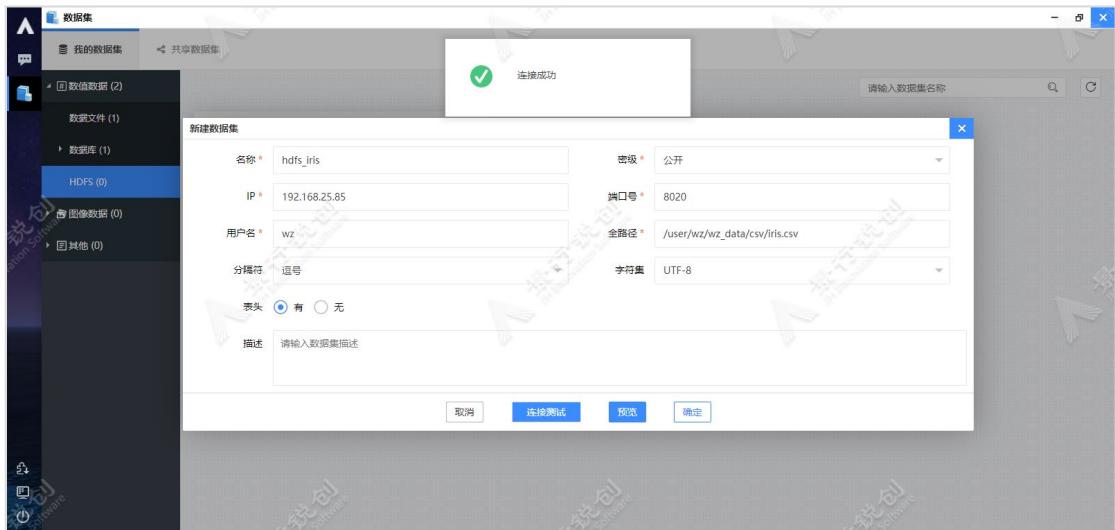
字符集：选择文件的编码格式。

分隔符：选择文件中使用的分隔符。

描述：输入该数据集的描述信息，可选。

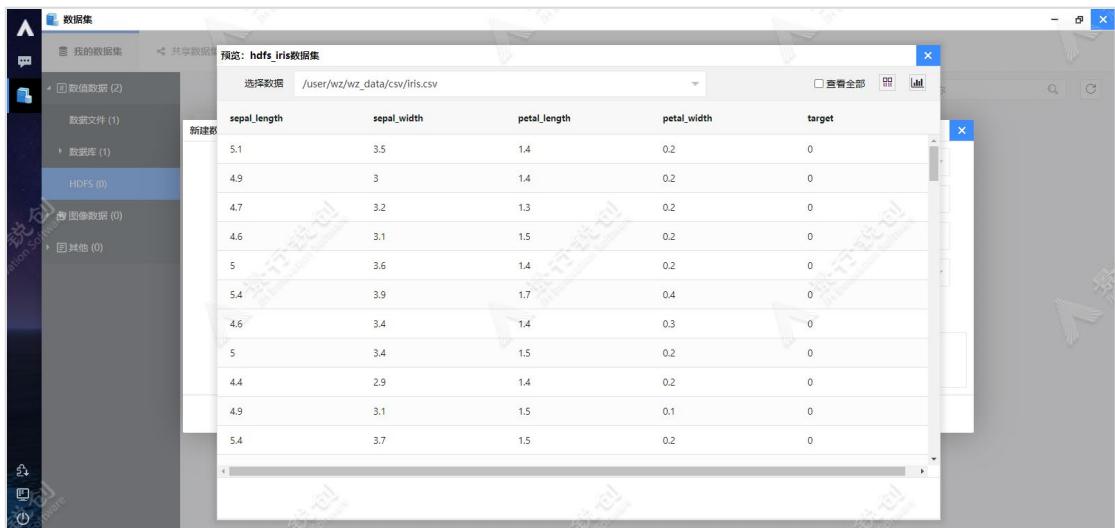
填写完对应信息后，点击“连接测试”按钮，即可测试 HDFS 服务的连通性，会有成功和失败提示。连接测试效果如下图所示：

第二章



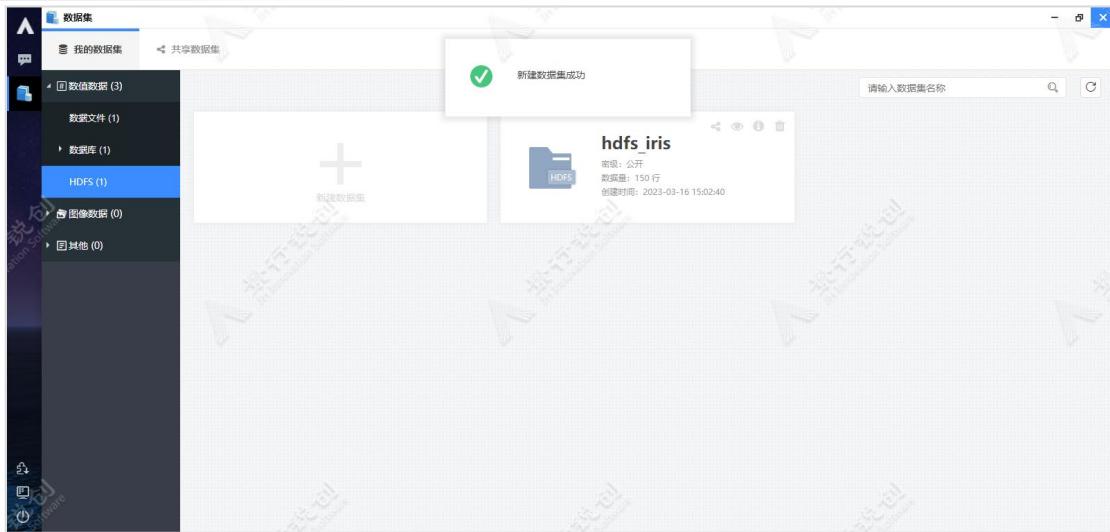
连接测试示意图

点击“预览”按钮，即可查看“查询 SQL”的结果。如下图所示：



预览“数值数据”示意图

点击“确定”按钮，新建数据集。如下图所示：



新建“数值数据-HDFS”数据集成功示意图

2.1.2. 图像数据

2.1.2.1. “图像数据-数据文件”数据集

新建“图像数据-数据文件”数据集，操作步骤如下：依次点击“我的数据集” – “图像数据” – “数据文件”，然后在右侧的工作区，点击“新建数据集”卡片按钮，此时会弹出“新建数据集”窗口，如下图所示：



新建“图像数据-数据文件”数据集

图中每个参数的具体含义如下：

第二章

名称：数据集的名称，输入任意符合命名规则的名称即可。

密级：管理员开启密级功能后，显示此选项，默认为用户密级，用于数据安全、保密。

选择文件夹：点击蓝色的“文件夹图标”按钮，然后在弹出的“选择文件夹”窗口中，选择图像文件夹即可。支持选择多个文件夹，程序仅支持1层目录，更多子文件夹内容自动忽略。

标注方式：有4种标注方式，如下所示：

- 1) “无”：表示没有使用任何标注方式，例如：测试集；
- 2) “文件标注”：标签文件项选择与选择文件项对应的标签文件，目前仅支持“csv”、“json”和“xml”格式的标签文件；
- 3) “文件夹名标注”：每个文件夹就是一个类别的集合，并且通过文件夹名来标注类别；
- 4) “文件名标注”：图像的类别通过图像文件名称标识，例如：1-pic.png，即该图像中的内容是1，此时使用了“标注信息-文件名”类别的标注方式；

描述：输入该数据集的描述信息，可选。

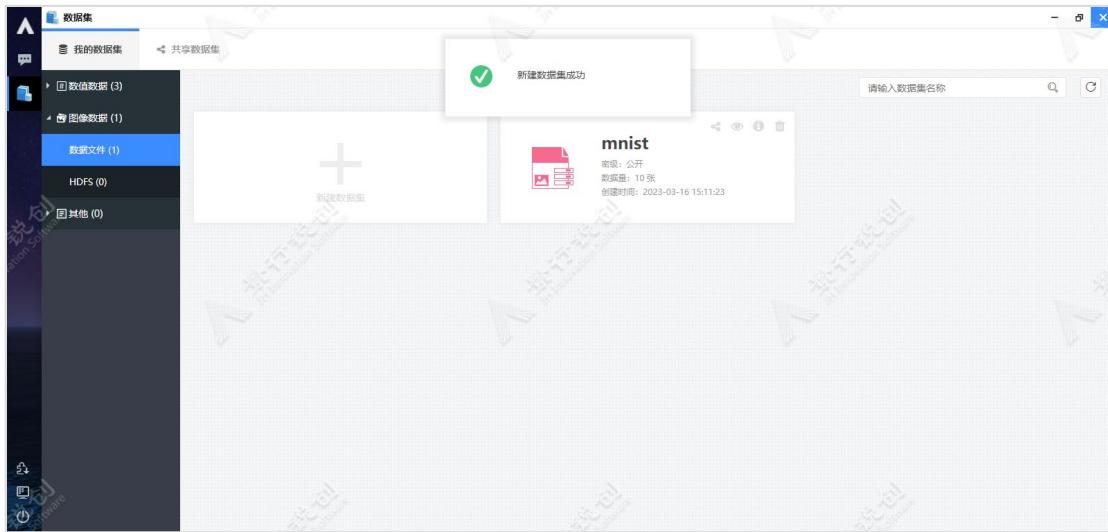
填写完对应的信息后，点击“预览”按钮，即可查看数据集的数据文件数据。

预览效果如下图所示：



预览“图像数据-数据文件”数据集

点击“确定”按钮，新建数据集。如下图所示：



2.1.2.2. “图像数据-HDFS”数据集

新建“图像数据-HDFS”数据集，操作步骤如下：依次点击“我的数据集” – “图像数据” – “数据文件”，然后在右侧的工作区，点击“新建数据集”卡片按钮，此时会弹出“新建数据集”窗口，如下图所示：



新建“图像数据-HDFS”数据集

图中每个参数的具体含义如下：

名称：数据集的名称，输入任意符合命名规则的名称即可。

密级：管理员开启密级功能后，显示此选项，默认为用户密级，用于数据安

第二章

全、保密。

IP: 输入 HDFS 服务节点的 IP。

端口号: 输入 HDFS 服务使用的端口号。

用户名: 输入 HDFS 服务的用户名。

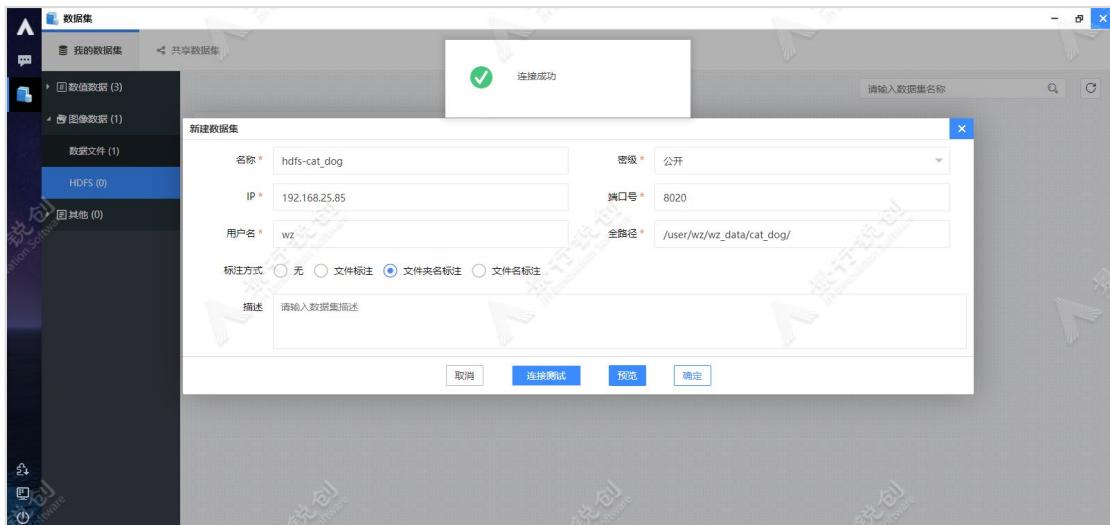
全路径: 输入数据文件目录的绝对路径。

标注方式: 有 4 种标注方式, 如下所示:

- “无”：表示没有使用任何标注方式，例如：测试集；
- “文件标注”：标签文件项选择与选择文件项对应的标签文件，目前仅支持“csv”、“json”和“xml”格式的标签文件；
- “文件夹名标注”：每个文件夹就是一个类别的集合，并且通过文件夹名来标注类别；
- “文件名标注”：图像的类别通过图像文件名称标识，例如：1-pic.png，即该图像中的内容是 1，此时使用了“标注信息-文件名”类别的标注方式；

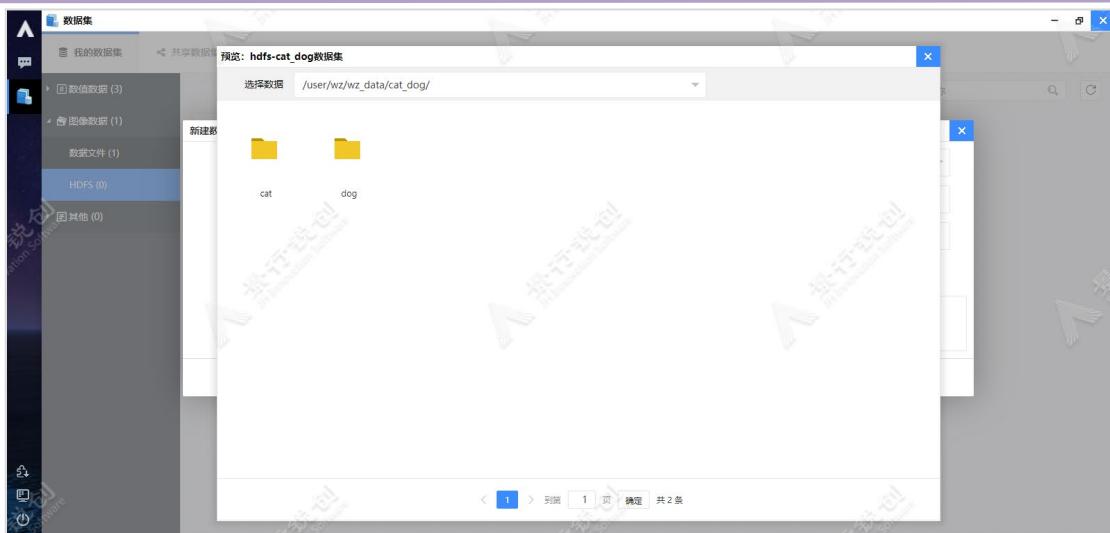
描述: 输入该数据集的描述信息，可选。

填写完对应信息后，点击“连接测试”按钮，即可测试 HDFS 服务的连通性，会有成功和失败提示。连接测试效果如下图所示：



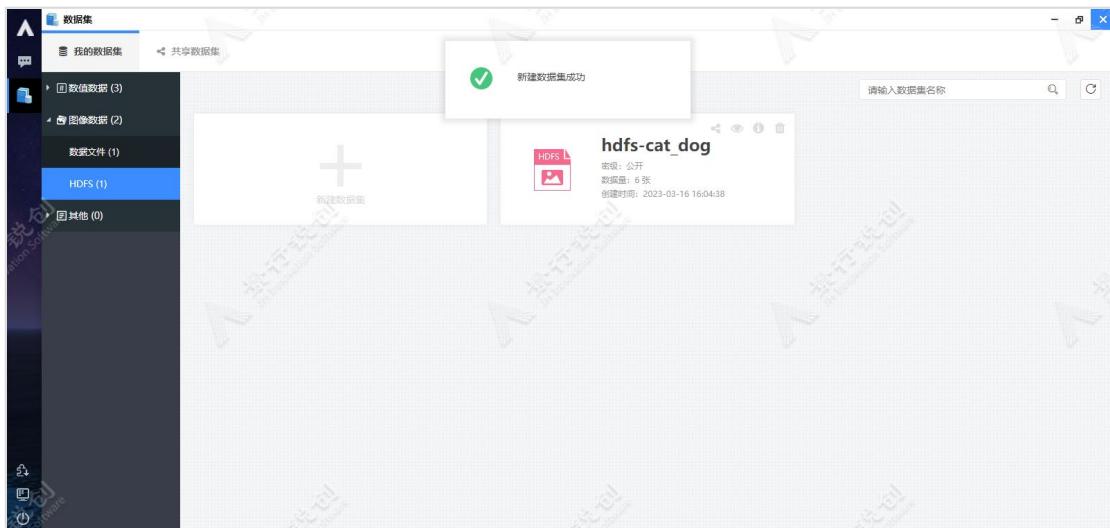
连接测试示意图

点击“预览”按钮，即可查看数据集的数据文件数据。如下图所示：



预览“图像数据-HDFS”数据集示意图

点击“确定”按钮，新建数据集。如下图所示：

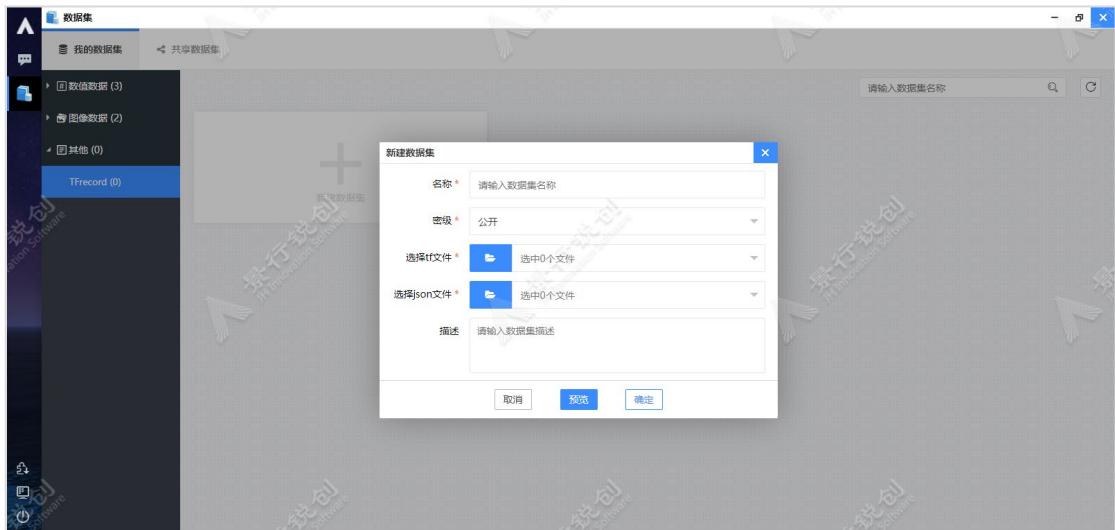


新建“图像数据-数据文件”数据集成功示意图

2.1.3. 其他

新建“其他-TFrecord”数据集，操作步骤如下：依次点击“我的数据集” – “其他” – “TFrecord”，然后在右侧的工作区，点击“新建数据集”卡片按钮，此时会弹出“新建数据集”窗口，如下图所示：

第二章



新建“其他-TFrecord”数据集示意图

图中每个参数的具体含义如下：

名称：数据集的名称，输入任意符合命名规则的名称即可。

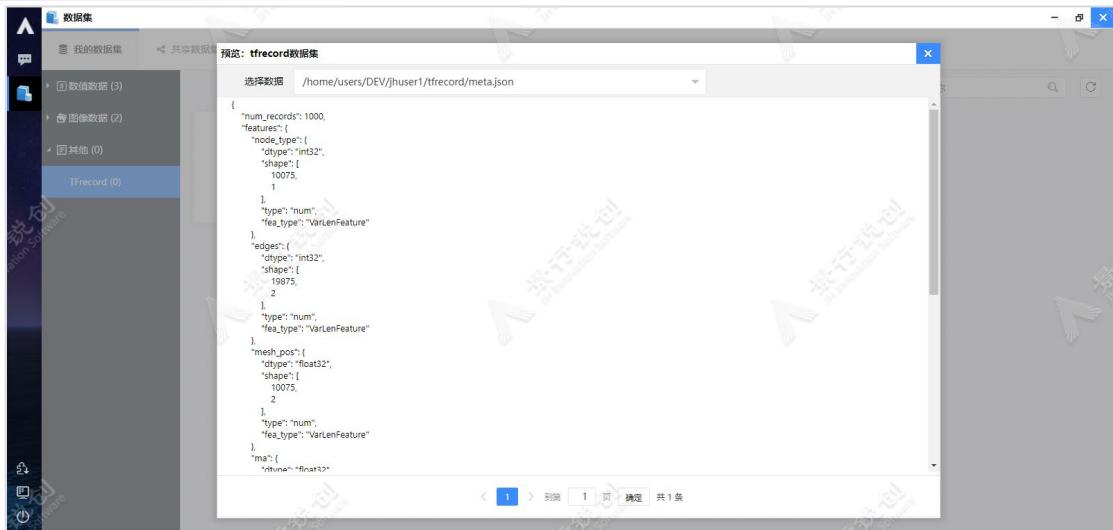
密级：管理员开启密级功能后，显示此选项，默认为用户密级，用于数据安全、保密。

选择 tf 文件：点击蓝色的“文件图标”按钮，然后在弹出的“选择文件”窗口中，选择 tfrecord 文件即可。只允许支持选择一个 tfrecord 文件。

选择 json 文件：点击蓝色的“文件图标”按钮，然后在弹出的“选择文件”窗口中，选择 json 文件即可。只允许支持选择一个 json 文件。

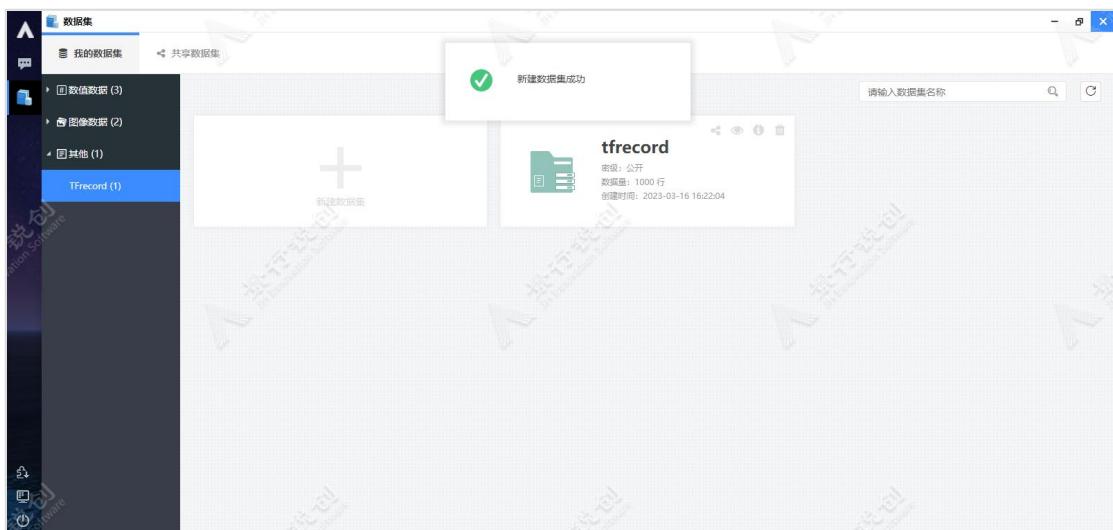
描述：输入该数据集的描述信息，可选。

点击“预览”按钮，即可查看数据集的数据文件数据。如下图所示：



预览“其他-TFrecord”数据集示意图

点击“确定”按钮，新建数据集。如下图所示：



新建“其他-TFrecord”数据集成功示意图

2. 1. 4. 共享数据集

共享数据集的产生有 2 种方式：

- 在我的数据集中，将我的数据集共享至共享数据集；
- 在共享数据集中，应用共享组中的数据文件，新建共享数据集；

各种角色对共享数据集的操作权限：

第二章

共享者：对共享数据集有修改、预览、使用和删除的权限；

共享组用户：对共享数据集有预览、使用的权限；

管理员：对共享数据集仅有删除权限；

其他用户：看不到共享数据集；

用途：共享组成员可以在方案设计和 AI 作业中使用共享数据集。

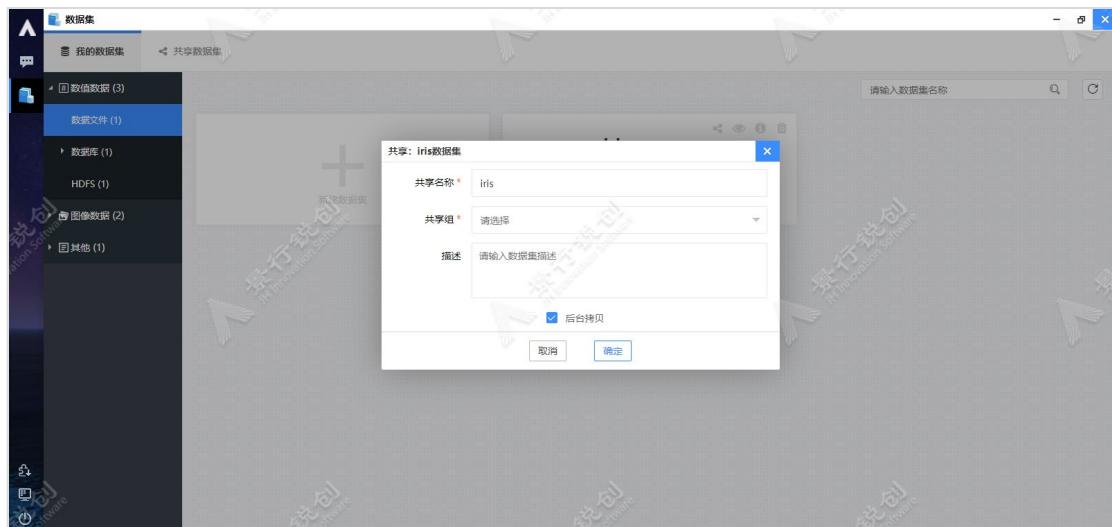
注意：进行共享操作前，需要先创建共享组。

2.1.4.1. 将我的数据集共享至共享数据集

共享数据文件类的数据集，如下所示：

- “数值数据-数据文件” 数据集
- “图像数据-数据文件” 数据集
- “其他-TFrecord” 数据集

以共享“数值数据-数据文件”数据集为例，步骤如下：依次点击“我的数据集” - “数值数据” - “数据文件”，然后在右侧的工作区，点击数据集卡片上的“共享”图标按钮，此时会弹出“共享”窗口，如下图所示：



数据文件共享至共享数值数据集

图中每个参数的具体含义如下：

共享名称：共享后的数据集名称，输入任意符合命名规则的名称即可。

共享组：选择可见的共享组，选项从我的数据->共享数据区获取。

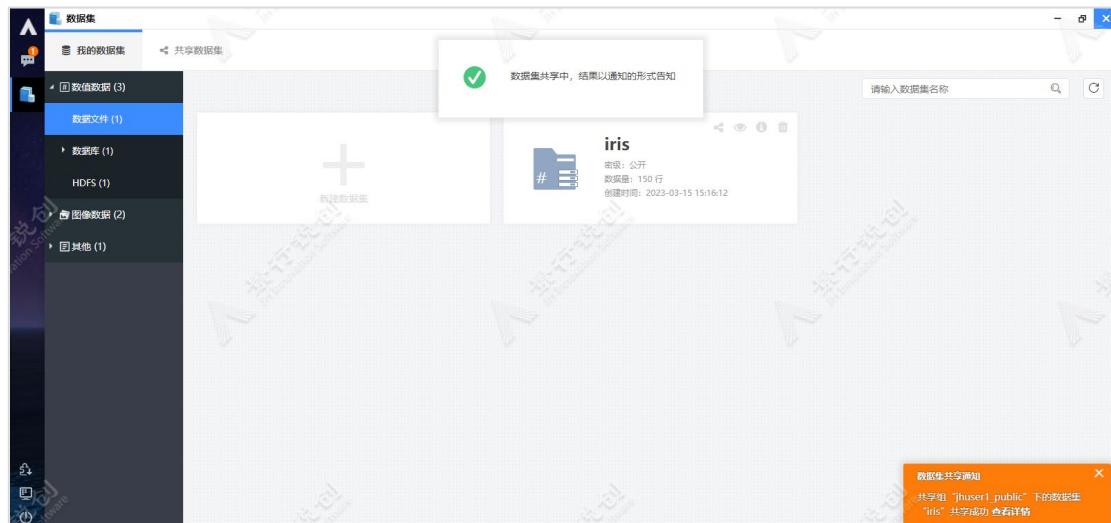
共享用户：选择共享组之后显示，显示为选择的共享组成员。

共享目录：选择共享组之后显示，只能在所选共享组的共享数据区选择文件夹，点击蓝色的“文件夹图标”按钮，然后在弹出的“选择文件夹”窗口中选择文件夹即可，用作存放源数据文件目录，只允许选择一个文件夹。

描述：共享后的数据集描述信息，可选。

后台拷贝：默认勾选，采用后台形式拷贝源数据集数据文件至共享目录，当前共享页面退出；不勾选则采用前台拷贝方式，当前共享页面直至共享结束退出；

填写完成后，点击“确定”按钮，进行数据集共享，如下图所示：



共享数据集示意图

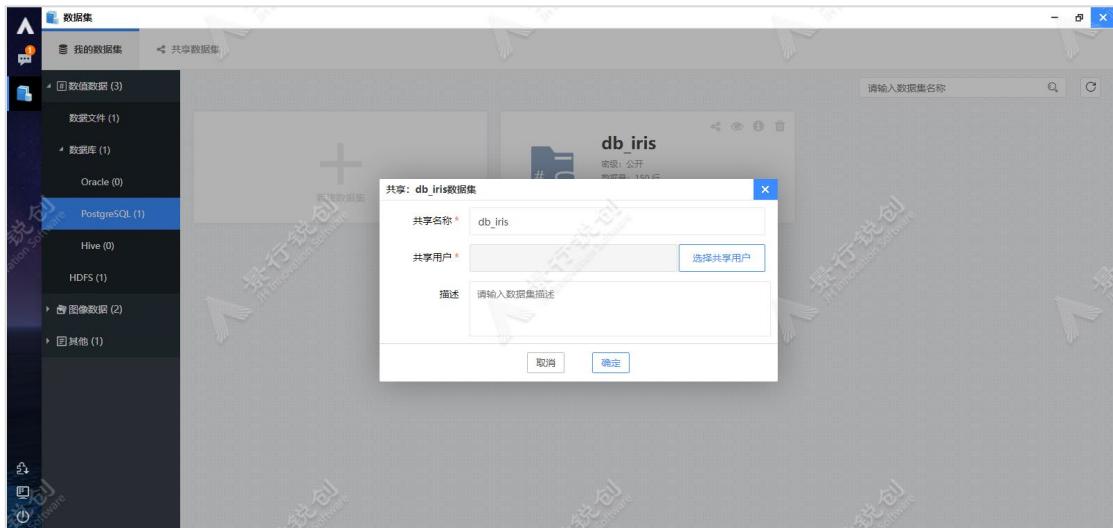
共享非“数据文件”类的数据集，如下所示：

- “数值数据-数据库”数据集
- “数值数据-HDFS”数据集
- “图像数据-HDFS”数据集

以共享“数值数据-数据库-PostgreSQL”数据集为例，操作步骤如下：依次点击“我的数据集” - “数值数据” - “数据库” - “PostgreSQL”，然后在右侧

第二章

的工作区，点击数据集卡片上的“共享”按钮，此时会弹出“共享”窗口，如下图所示：



PostgreSQL 共享至共享数值数据集

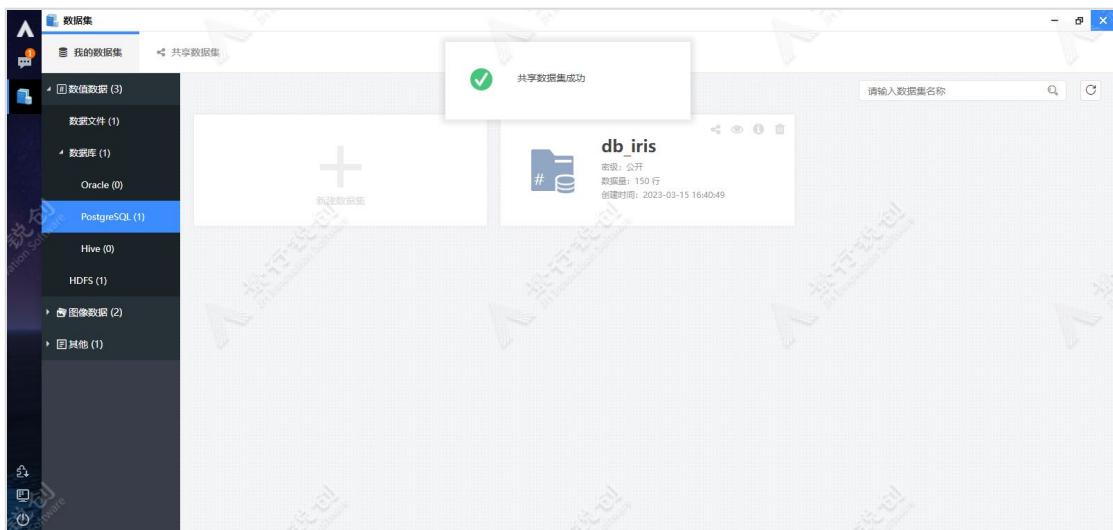
图中每个参数的具体含义如下：

共享名称：共享后的数据集名称，输入任意符合命名规则的名称即可。

共享用户：选择共享用户。

描述：共享后的数据集描述信息，可选。

填写完成后，点击“确定”按钮，共享数据集，如下图所示：



2.1.4.2. 新建共享数据集

新建数据文件类的共享数据集，如下所示：

- 数值数据->数据文件
- 图像数据->数据文件
- 其他->TFrecord

以新建“数值数据-数据文件”共享数据集为例，操作步骤如下：依次点击“共享数据集” - “数值数据” - “数据文件”，然后在右侧的工作区，点击“新建共享数据集”卡片按钮，此时会弹出“新建共享数据集”窗口，如下图所示：



使用数据文件添加共享数值数据集

图中每个参数的具体含义如下：

名称：同我的数据集中数值数据->数据文件。

密级：同我的数据集中数值数据->数据文件。

共享组：选择可见的共享组，选项从我的数据->共享数据区获取。

共享用户：选择共享组之后显示，显示为选择的共享组成员。

选择文件：选择共享组之后显示，只能在所选共享组的共享数据区选择文件，同我的数据集中数值数据->数据文件。

字符集：同我的数据集中数值数据->数据文件。

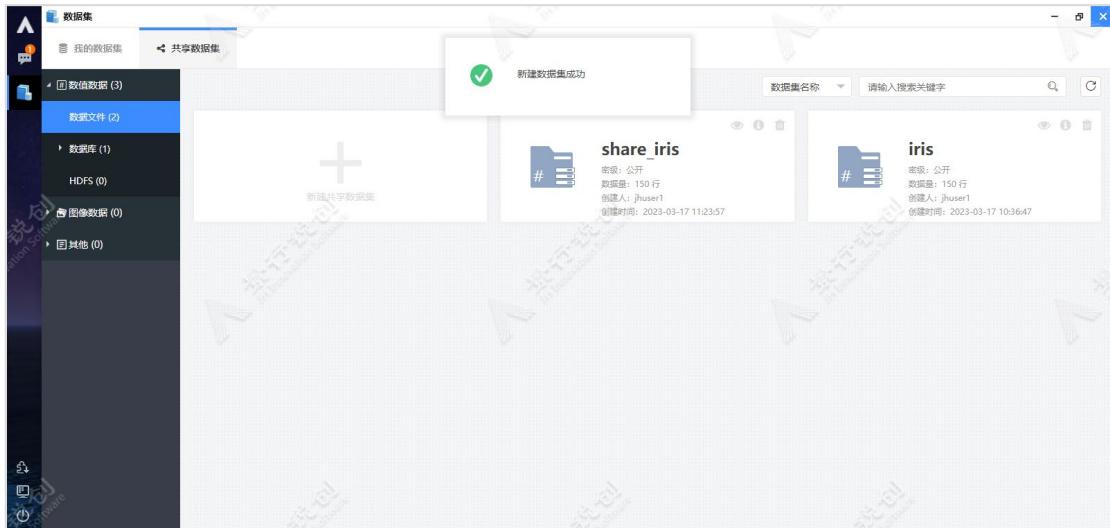
分隔符：同我的数据集中数值数据->数据文件。

第二章

表头: 同我的数据集中数值数据->数据文件。

描述: 同我的数据集中数值数据->数据文件。

填写完成后，点击“确定”按钮，新建共享数据集。如下图所示：

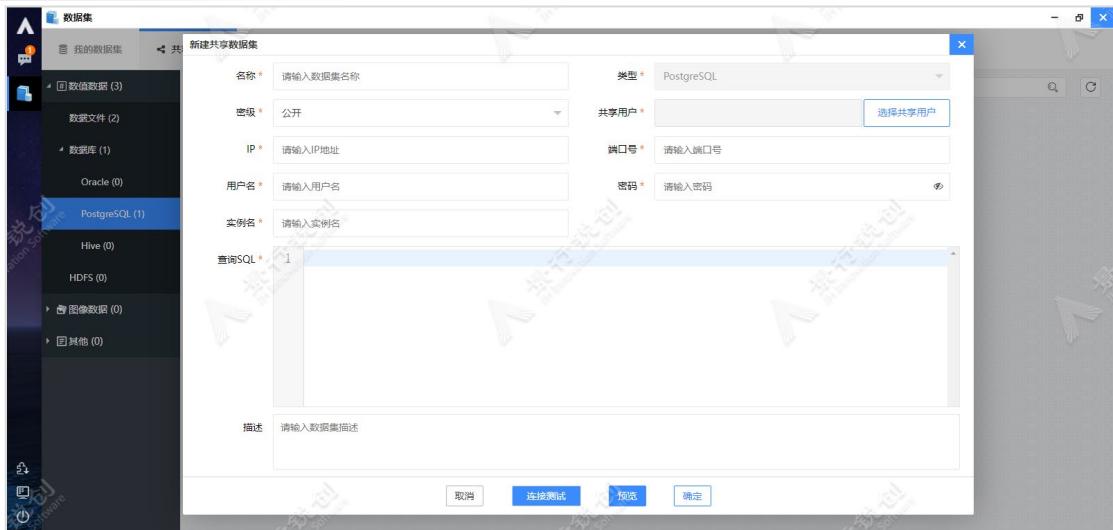


新建共享数据集示意图

共享非“数据文件”类的数据集，如下所示：

- “数值数据-数据库”数据集
- “数值数据-HDFS”数据集
- “图像数据-HDFS”数据集

以新建“数值数据-数据库-PostgreSQL”共享数据集为例，操作步骤如下：
依次点击“共享数据集” - “数值数据” - “数据库” - “PostgreSQL”，然后在右侧的工作区，点击“新建共享数据集”卡片按钮，此时会弹出“新建共享数据集”窗口，如下图所示：



使用数据库添加数值数据集

图中每个参数的具体含义如下：

名称： 同我的数据集中数值数据->数据库->PostgreSQL。

类型： 同我的数据集中数值数据->数据库->PostgreSQL。

密级： 同我的数据集中数值数据->数据库->PostgreSQL。

共享用户： 选择共享用户。

IP： 同我的数据集中数值数据->数据库->PostgreSQL。

端口号： 同我的数据集中数值数据->数据库->PostgreSQL。

用户名： 同我的数据集中数值数据->数据库->PostgreSQL。

密码： 同我的数据集中数值数据->数据库->PostgreSQL。

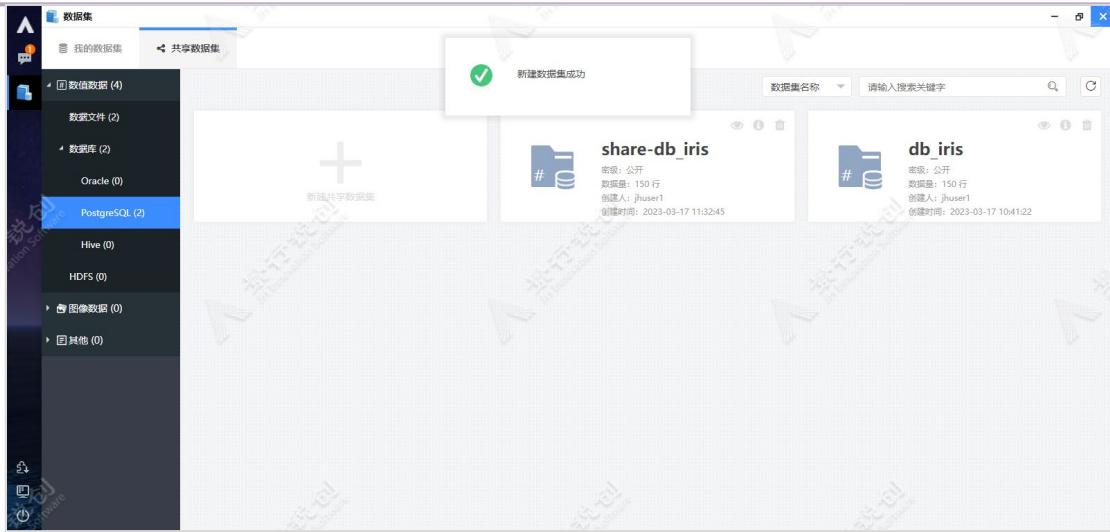
实例名： 同我的数据集中数值数据->数据库->PostgreSQL。

查询 SQL： 同我的数据集中数值数据->数据库->PostgreSQL。

描述： 同我的数据集中数值数据->数据库->PostgreSQL。

填写完成后，点击“确定”按钮，新建共享数据集。如下图所示：

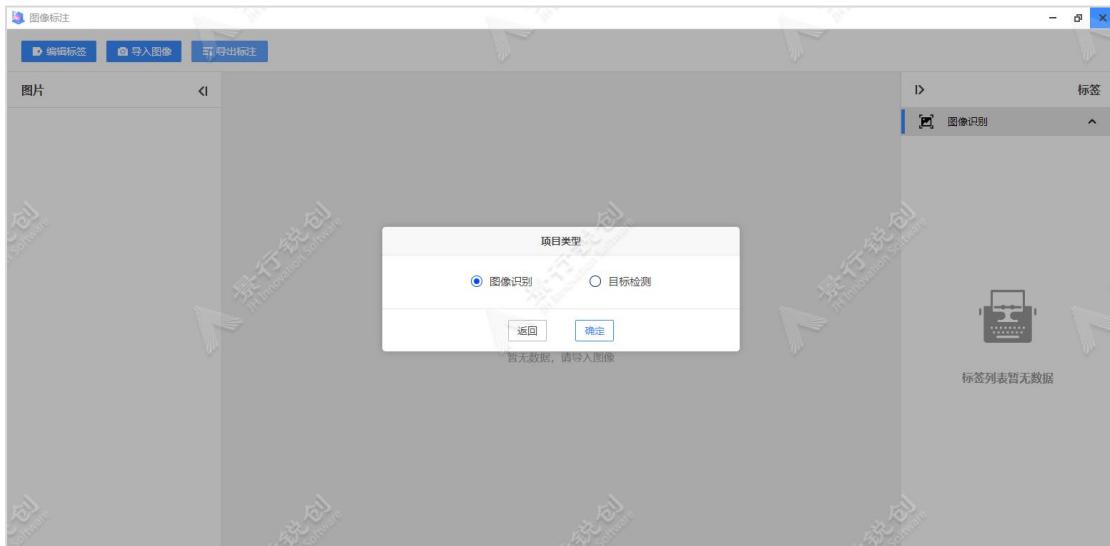
第二章



2.1.5. 图像标注

图像标注用于对图像进行手动标注，支持图像识别和目标检测两种类型的标注。目标检测中支持矩形标注、点标注、线标注以及多边形标注，标注结果可以导出为多种格式的标注文件。

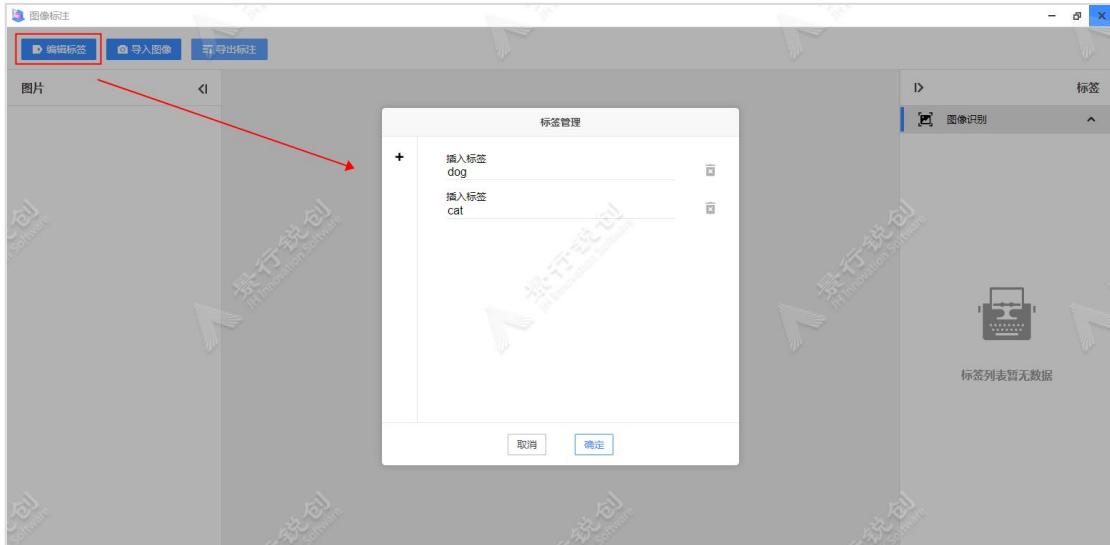
进入图像标注主界面，默认应用图像识别，如下图所示：



2.1.5.1. 编辑标签

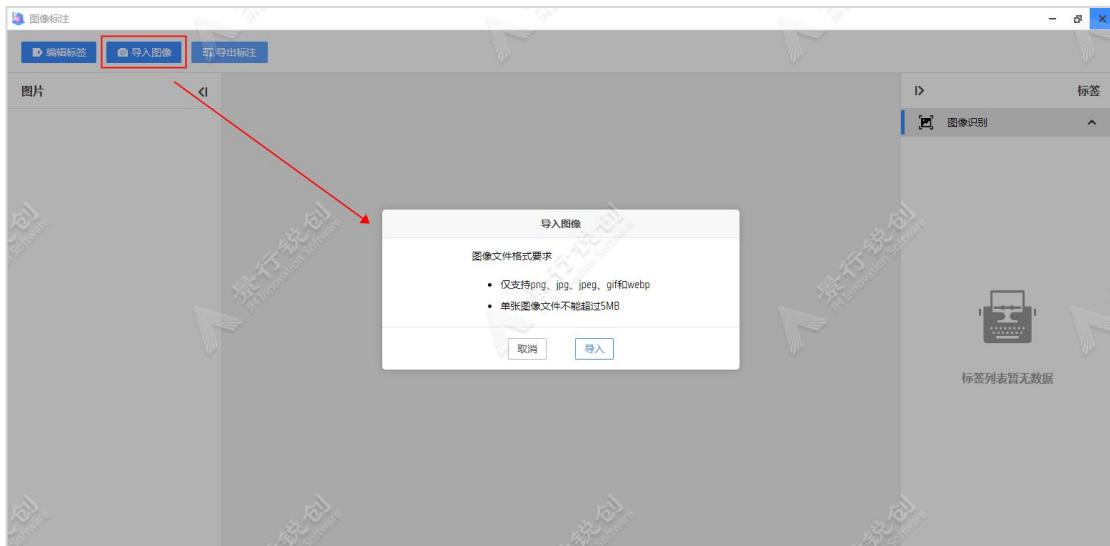
点击编辑标签按钮，弹出标签管理弹窗。点击加号（“+”）按钮，便可对

图像数据标签进行添加，标签添加成功后，可在图像标注主界面的右侧区域查看标签列表。以添加 cat 和 dog 两个标签为例，具体操作如下图所示：



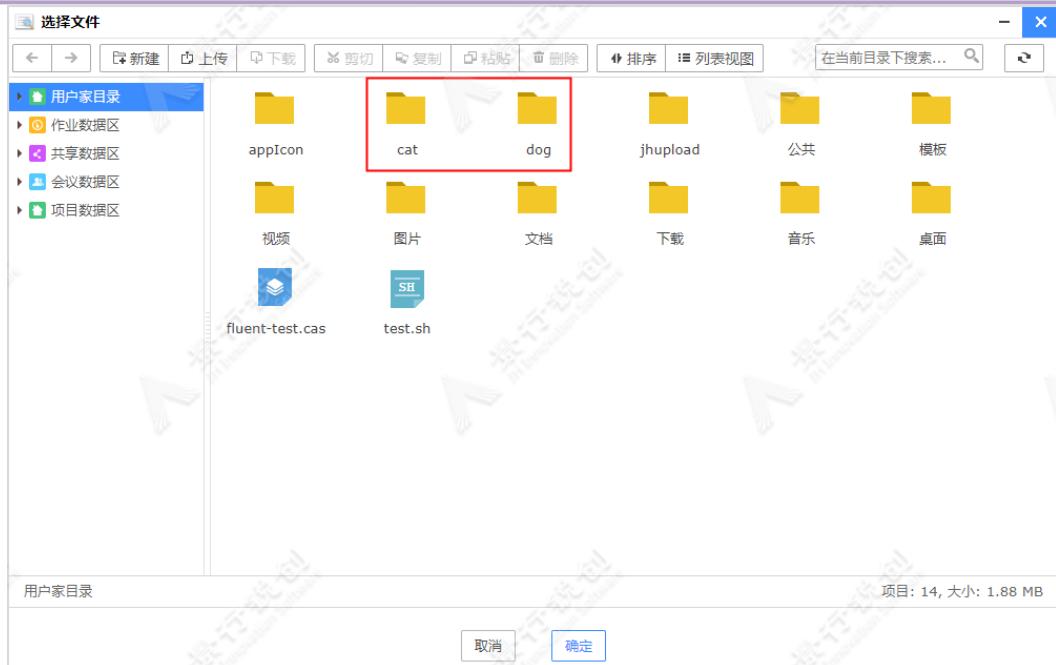
2.1.5.2. 导入图像

点击导入图像按钮，打开导入图像弹窗，用于告知导入图像的格式要求，如下图所示：



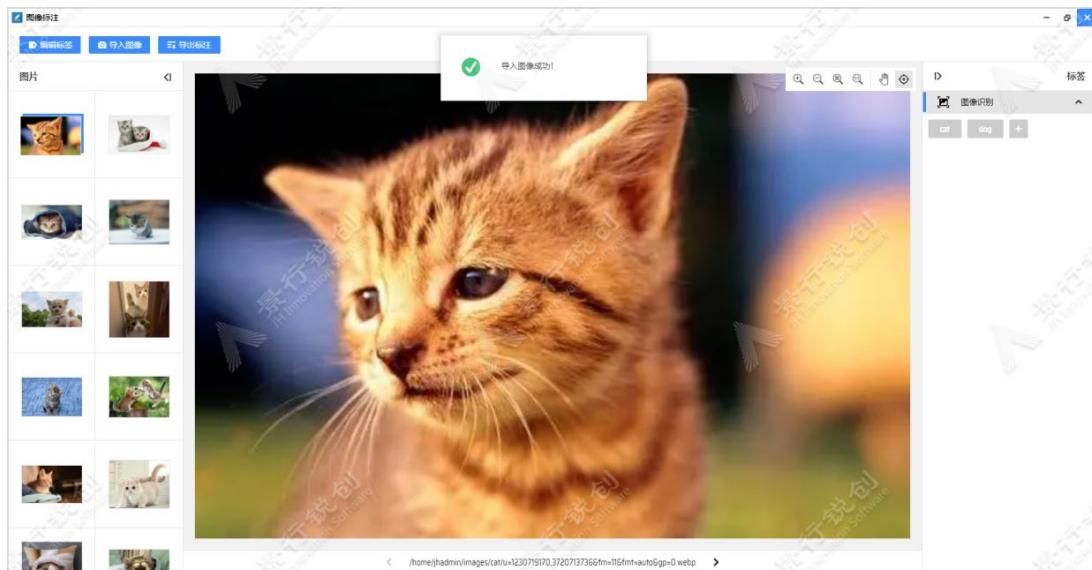
点击导入按钮，会弹出选择文件窗口。在选择文件弹窗中，可以选择图像文件或文件夹，选择一个包含各种格式的文件夹，会自动忽略非图像格式的文件。以导入用户家目录下的 cat 和 dog 目录为例，具体操作如下图所示：

第二章



注：按住 **ctrl** 键，选择需要的文件或目录，便可完成多选。

导入图像成功后，会在左侧的图片区域加载图片列表，并且在画布区域默认加载第一张图片，如下图所示：

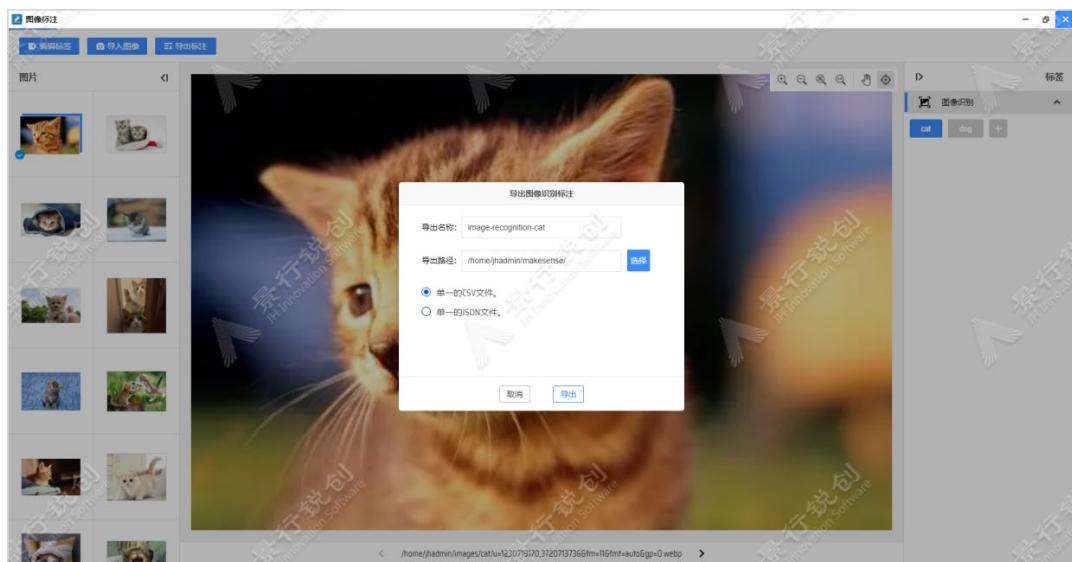


2.1.5.3. 图像识别

图像标注工具支持图像识别，标注图像数据文件属于某一分类或多个分类，点击右侧标签列表中的标签对当前选中的图片进行标注，如下图所示：



点击左上角的“导出标注”按钮，会弹出“导出图像识别标注”窗口，如下图所示：



图中每个参数的具体含义如下所示：

导出名称：默认为 my-project-name，可修改。

导出路径：选择标注文件的存储路径。

导出文件类型：支持导出为 csv 和 json 格式的标注文件。

在上图中填写完对应信息后，点击导出按钮，将标注结果导出。在选择的存储路径中查看标注文件内容，如下图所示：

第二章

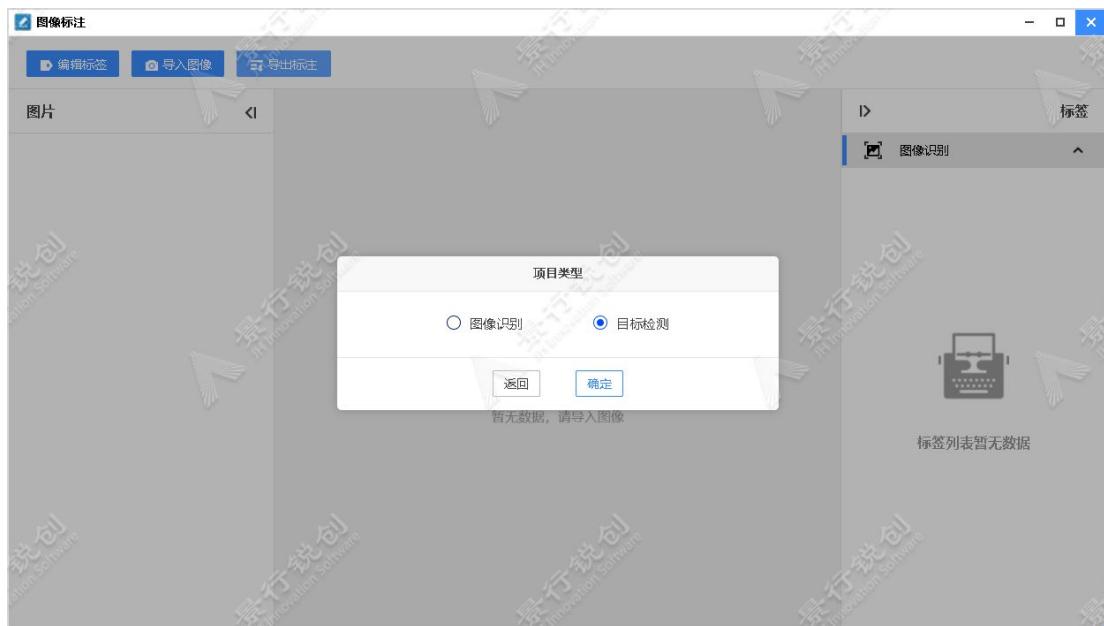
```
pic_path, label
/home/jhadmin/images/cat/u=1230719170,3720713736&fm=11&fmt=auto&gp=0.webp, "[cat]"
```

数据详情：

- pic_path: 第一列表示该图片的路径。
- label: 第二列表示该图片选中的标签集合

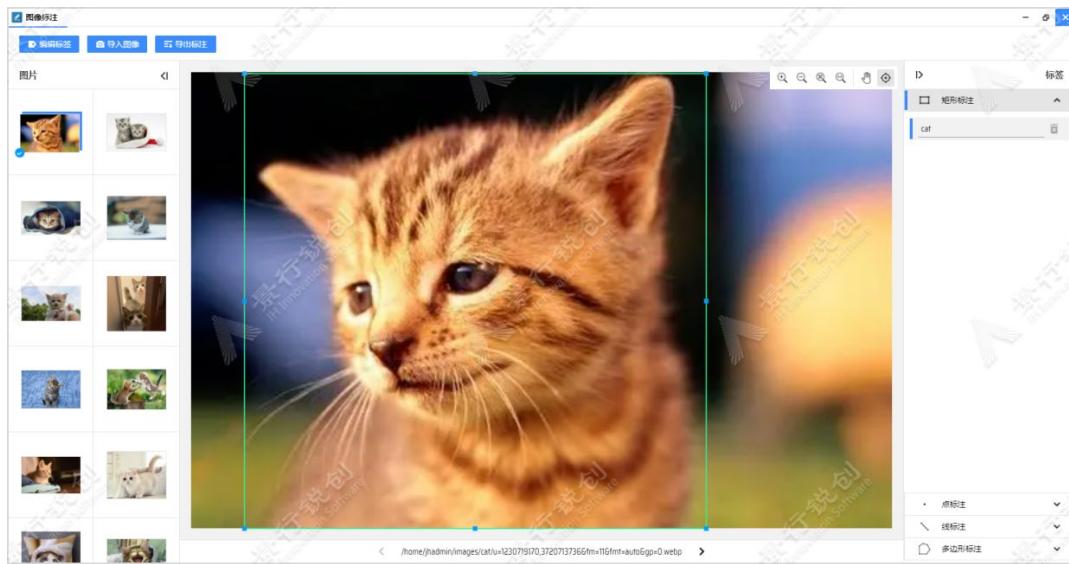
2.1.5.4. 目标检测

选中目标检测单选项，进入目标检测主页：

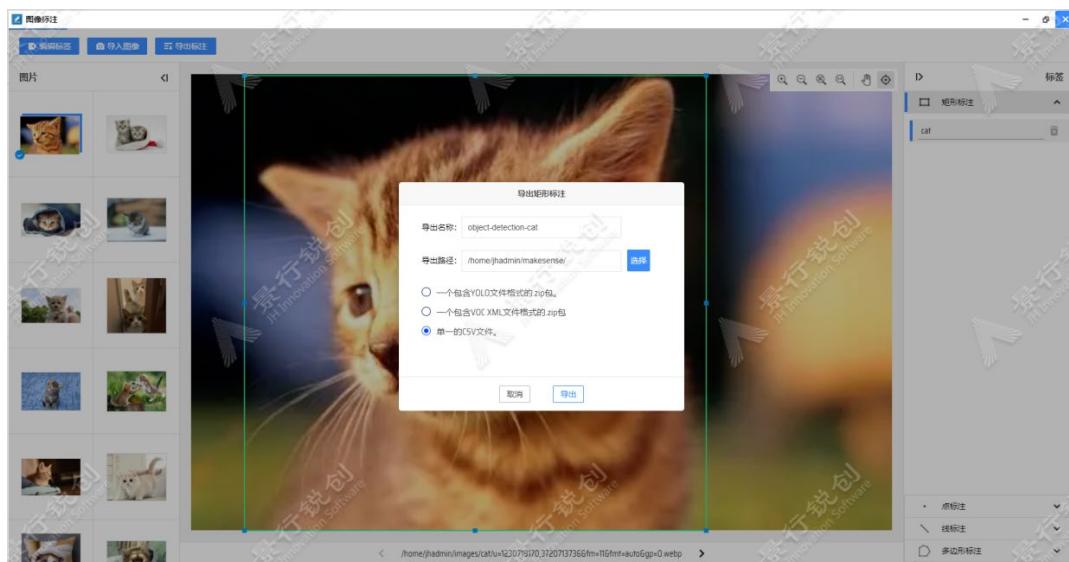


(1) 矩形标注

默认使用矩形标注，对图像文件中的矩形区域进行标注，使用矩形框将所要标注的局部图像框起来，然后选择相应的标签，如下图所示：



点击左上角的“导出标注”按钮，会弹出“导出矩形标注”窗口，如下图所示：



图中每个参数的具体含义如下：

导出名称：默认为 my-project-name，可修改。

导出路径：选择标注文件的存储路径。

导出文件类型：支持导出为 YOLO、VOC XML 和 CSV 格式的标注文件。

在上图中填写完对应信息后，点击导出按钮，将标注结果导出。在选择的存储路径中查看标注文件内容，如下图所示：

```
label, xmin, ymin, boxw, boxh, pic_path, width, height
cat, 37, 2, 325, 323, /home/jhadmin/images/cat/u=1230719170,3720713736&fm=11&fmt=auto&gp=0.webp, 500, 325
```

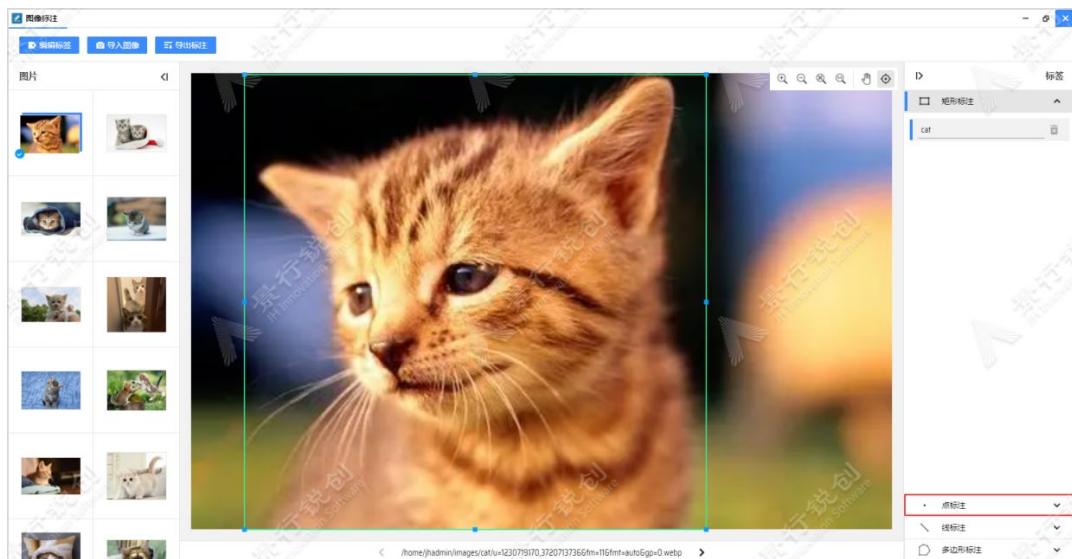
第二章

标注详情：

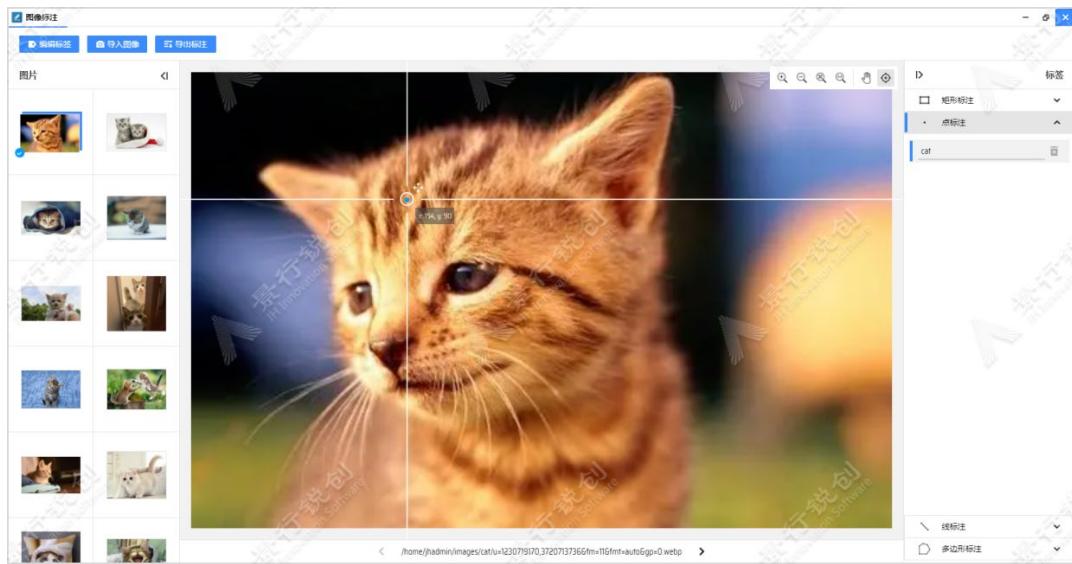
- label: 第一列表示对该图片局部区域的图像选中的标签
- xmin: 第二列表示矩形框左上角的 x 轴坐标
- ymin: 第三列表示矩形框左上角的 y 轴坐标
- boxw: 第四列表示矩形框的宽度
- boxh: 第五列表示矩形框的高度
- pic_path: 第六列表示该图片的路径
- width: 第七列表示该图片的宽度
- height: 第八列表示该图片的高度

(2) 点标注

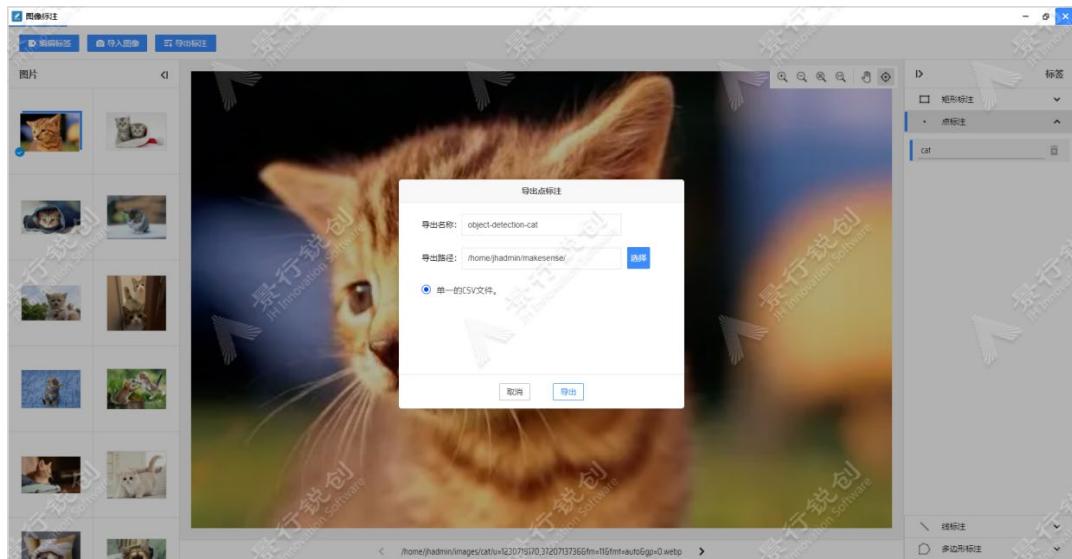
点击点标注切换至点标注模式，如下图所示：



对图像文件中的点域进行标注，在画布图片上进行点击，便可进行点标注，如下图所示：



点击左上角“导出标注”按钮，会弹出“导出点标注”窗口，如下图所示：



图中每个参数的具体含义如下：

导出名称：默认为 my-project-name，可修改。

导出路径：选择标注文件存储的文件夹。

导出文件类型：仅支持导出 csv 格式的标注文件。

在上图中填写完对应信息后，点击导出按钮，将标注结果导出。在选择的存储路径中查看标注文件内容，如下图所示：

```
label, x, y, pic_path, width, height
cat, 180, 93, /home/jhadmin/images/cat/u=1230719170,3720713736&fm=11&fmt=auto&gp=0.webp, 500, 325
```

第二章

标注详情：

- label: 第一列表示对该点标注选中的标签
- x: 第二列表示该点标注的 x 轴坐标
- y: 第三列表示该点标注的 y 轴坐标
- pic_path: 第四列表示该图片的路径。
- width: 第五列表示该图片的宽度
- height: 第六列表示该图片的高度

(3) 线标注

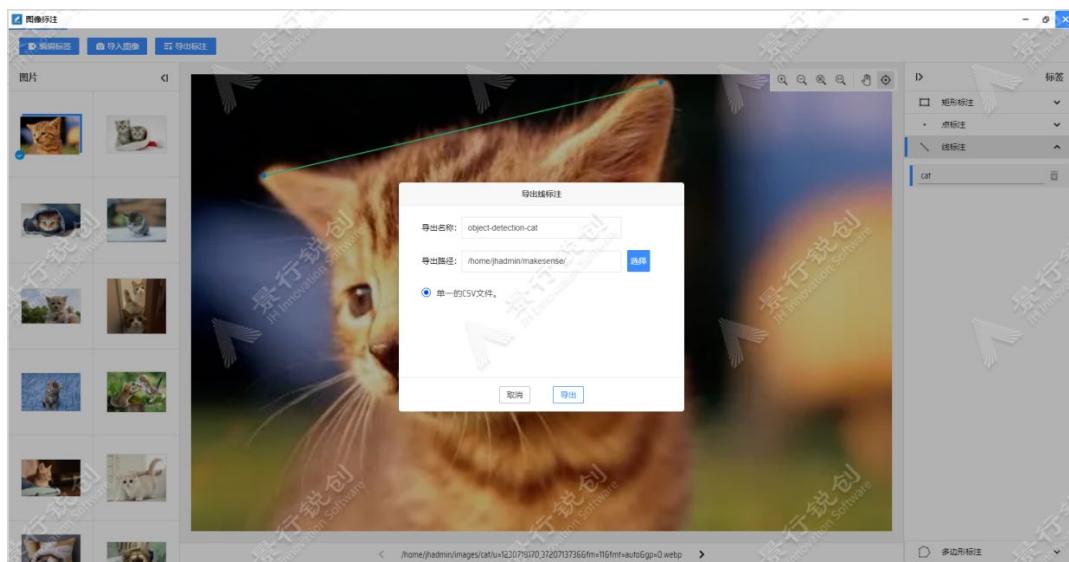
点击线标注切换至线标注模式，如下图所示：



通过鼠标在画布区域中的图片中点击两次，两点一线，便可进行线标注，如下图所示：



点击左上角的“导出标注”按钮，会弹出“导出线标注”窗口，如下图所示：



图中每个参数的具体含义如下：

导出名称：默认为 my-project-name，可修改。

导出路径：选择标注文件存储的文件夹。

导出文件类型：仅支持导出 csv 格式的标注文件。

在上图中填写完对应信息后，点击导出按钮，将标注结果导出。在选择的存储路径中查看标注文件内容，如下图所示：

```
label, xmin, ymin, xmax, ymax, pic_path, width, height
cat, 49, 73, 338, 5, /home/jhadmin/images/cat/u=1230719170,3720713736&fm=11&fmt=auto&gp=0.webp, 500, 325
```

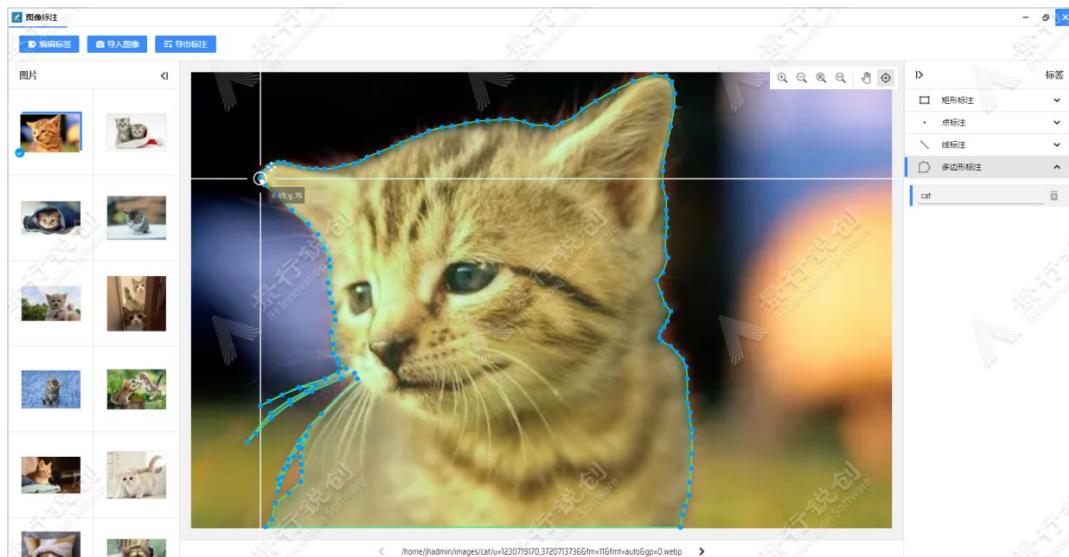
第二章

标注详情：

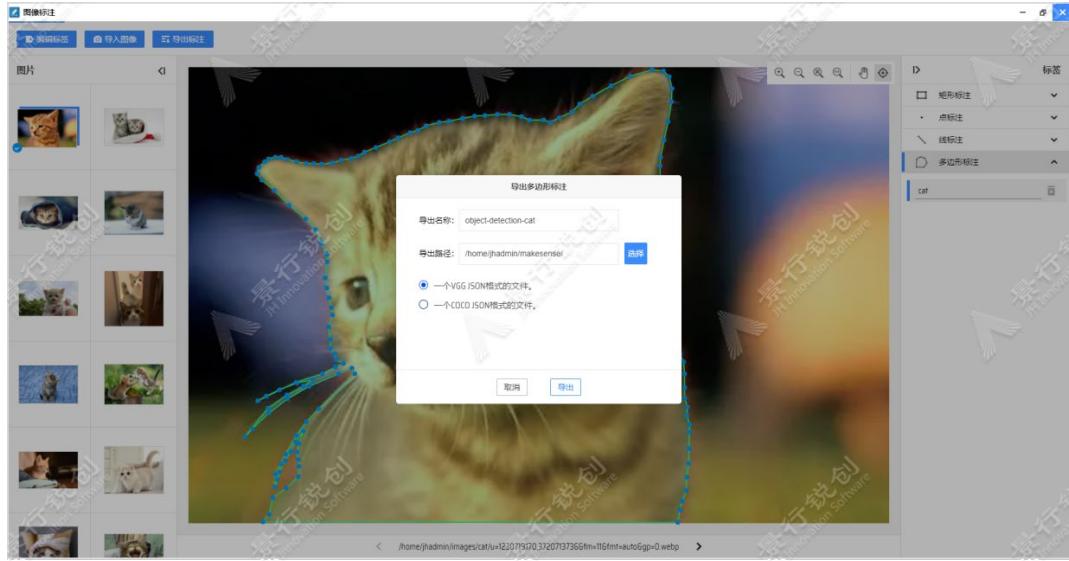
- label: 第一列表示对该线标注选中的标签
- xmin: 第二列表示起始点的 x 轴坐标
- ymin: 第三列表示起始点的 y 轴坐标
- xmax: 第四列表示终止点的 x 轴坐标
- ymax: 第五列表示终止点的 y 轴坐标
- pic_path: 第六列表示该图片的路径。
- width: 第七列表示该图片的宽度
- height: 第八列表示该图片的高度

(4) 多边形标注

点击多边形标注切换至多边形标注模式，通过鼠标在画布区域的图片上，多次点击产生的点连成一条线且首尾呼应，圈出一个不规则的区域，从而进行更加精确的标注，便可进行多边形标注，如下图所示：



点击左上角的“导出标注”按钮，会弹出“导出多边形标注”窗口，如下图所示：



图中每个参数的具体含义如下：

导出名称: 默认为 my-project-name, 可修改。

导出路径: 选择标注文件存储的文件夹。

导出文件类型: 支持导出为 VGG JSON 和 COCO JSON 格式的标注文件。

在上图中填写完对应信息后，点击导出按钮，将标注结果导出。在选择的存储路径中查看标注文件内容，如下图所示：

2.2 可视化建模

2.2.1. 方案设计

方案设计提供人工智能的流程可视化设计，方案提供了数据接入、数据处理、

第二章

机器学习和深度学习等组件，通过拖拽组件和连线的方式可以实现数据处理和人工智能建模、训练、评估等复杂流程。



方案设计页面

2.2.1.1. 构建方案设计流程

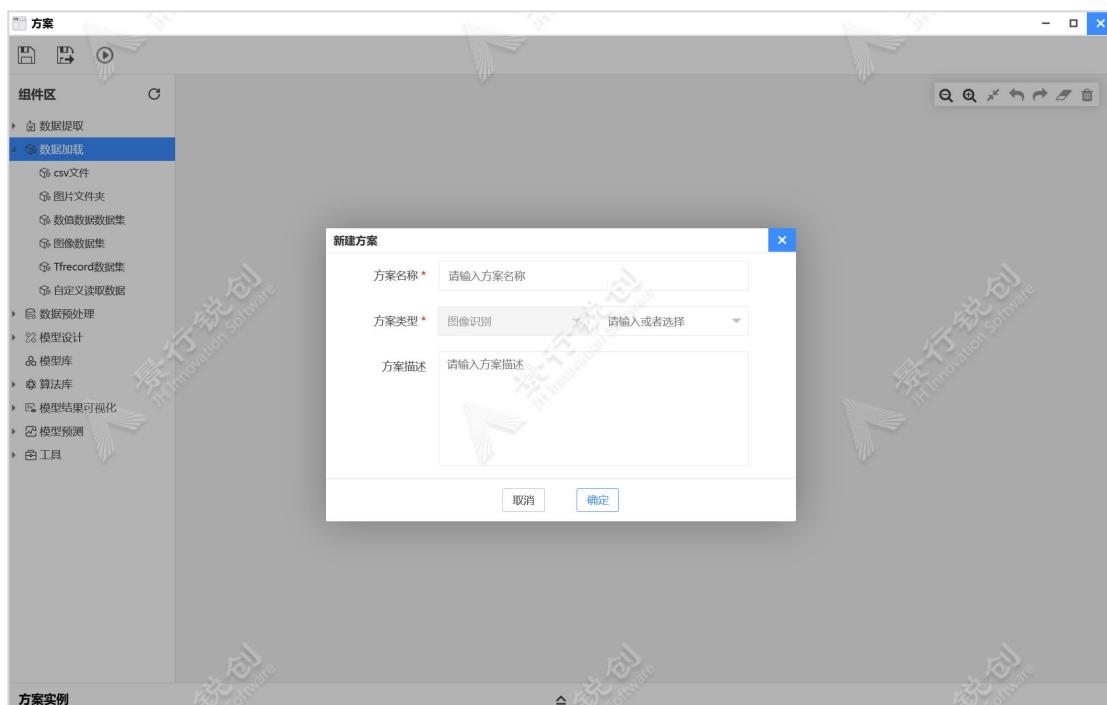
2.2.1.1.1. 新建

点击“新建方案”卡片，弹出“新建方案”窗口，如下图所示：



新建方案

再点击“新建空白方案”卡片后，页面跳转至“方案设计”页面，如下图所示：



新建方案页面

图中每个参数的具体含义如下：

方案名称：用户自定义，但是不能与现有方案名称一致。

密级：管理员开启密级功能后，显示此选项，默认为用户密级，用于数据安全、保密。

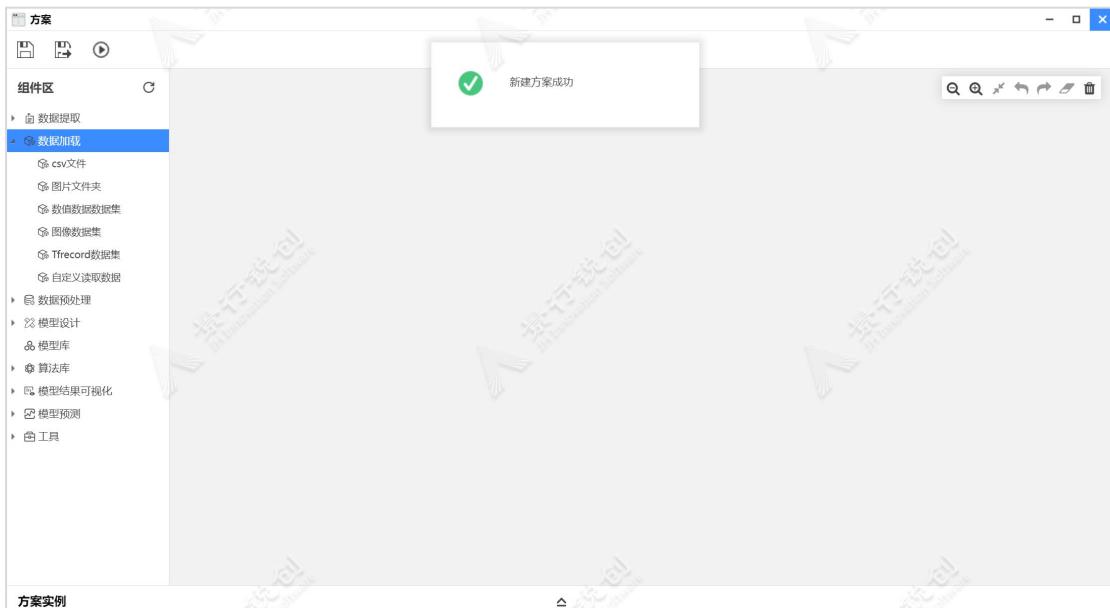
方案类型：默认为已选的方案类型，也可以自定义方案类型。

方案描述：用于描述方案设计的相关内容，选填。

2.2.1.1.2. 设计

填写完成新建方案的表单后，点击“确定”按钮，进入方案设计页面。

第二章



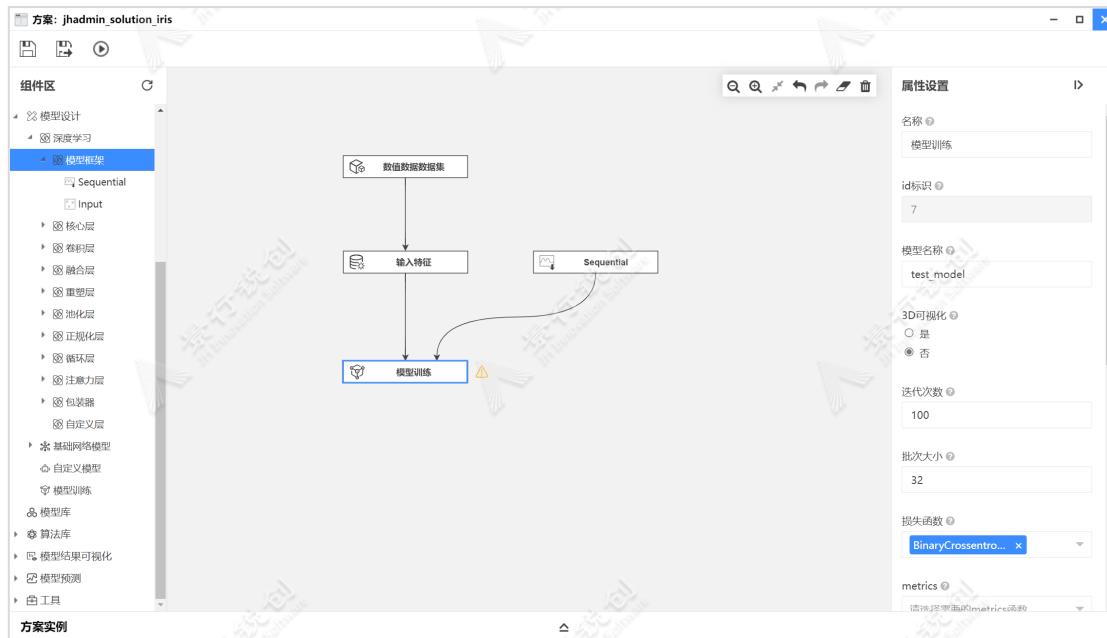
方案设计页面

“方案设计”左侧区域为组件区，组件区又根据组件属性分为“数据提取”，“数据加载”，“数据预处理”，“模型设计”，“模型库”，“算法库”，“模型结果可视化”，“模型预测”和“工具”。每一个组件都是一个模块的封装。

注：各组件对应的属性说明见附录一（数据加载处理组件属性说明）。

“方案设计”右侧区域为属性区，组件拖拽到画布区域后，属性设置面板就会从画布右侧滑出，用户可自行配置参数。当属性设置有误或未设置属性时，设计区的组件右侧会显示黄色感叹号的图标，当鼠标移至此图标上时会显示出错误信息，必须纠正错误后才能够开始运行此方案。

画布右上角的工具栏可对画布区域和画布中的组件进行快捷操作，功能包含缩小、放大、实际比例、后退、前进、删除和清空。其中“删除”按钮仅会删除画布中选中的任意组件，“清空”则会删除整个画布中的内容，长按鼠标右键可拖动整个画布。



画布工具及组件属性配置

自定义函数管理

在自定义函数管理中可以创建和管理“损失函数”和“Metrics 函数”，选择相应的分类并点击新建卡片即可新建自定义函数。

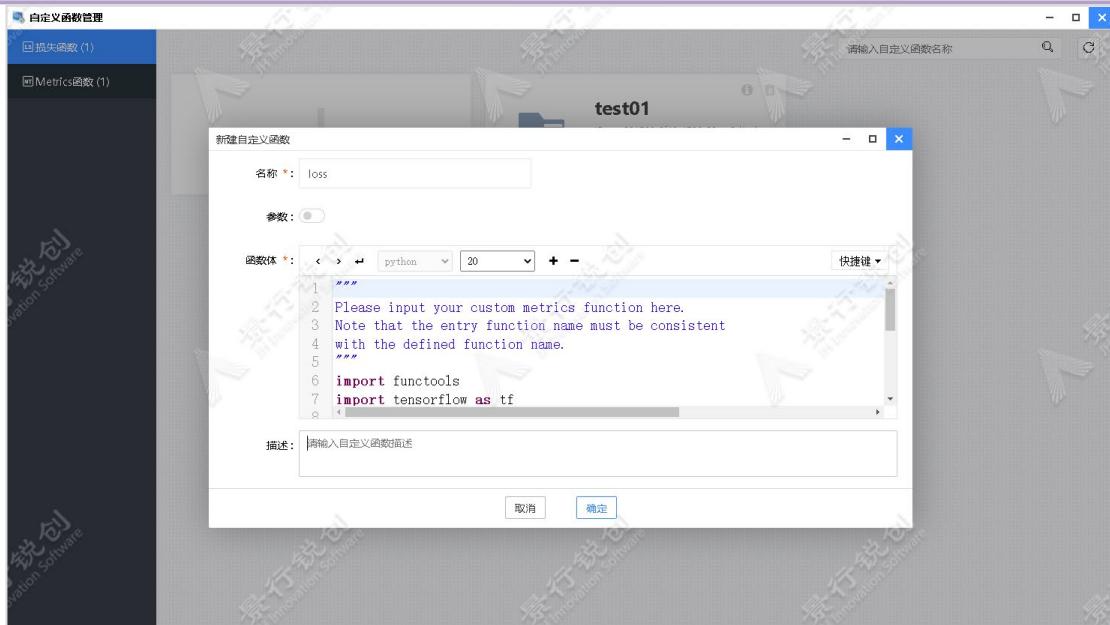
创建成功的自定义函数会以卡片的形式显示在右侧区域中，卡片中的信息包括函数名称、ID 和创建时间，当鼠标停留在卡片右上角的“描述”按钮处时，会显示此函数的描述信息，点击卡片右上角的删除按钮即可删除当前自定义函数，删除操作不可恢复。

注意：在此创建的自定义函数可在方案设计中被引用。

● 损失函数

进入“自定义函数管理”页面，选择左侧“损失函数”，点击“新建自定义函数”卡片，弹出新建窗口，如下图所示：

第二章



自定义损失函数

输入名称，并根据函数是否包含参数选择是否打开“参数”开关，在函数体部分输入自定义函数的内容，在描述部分输入对函数的描述或者备注信息，点击“确定”按钮，即可完成自定义损失函数的创建。

注意：函数“名称”的命名需要和函数入口名称一致。

损失函数内容如下：

```
"""
Please input your custom metrics function here.
Note that the entry function name must be consistent
with the defined function name.
"""

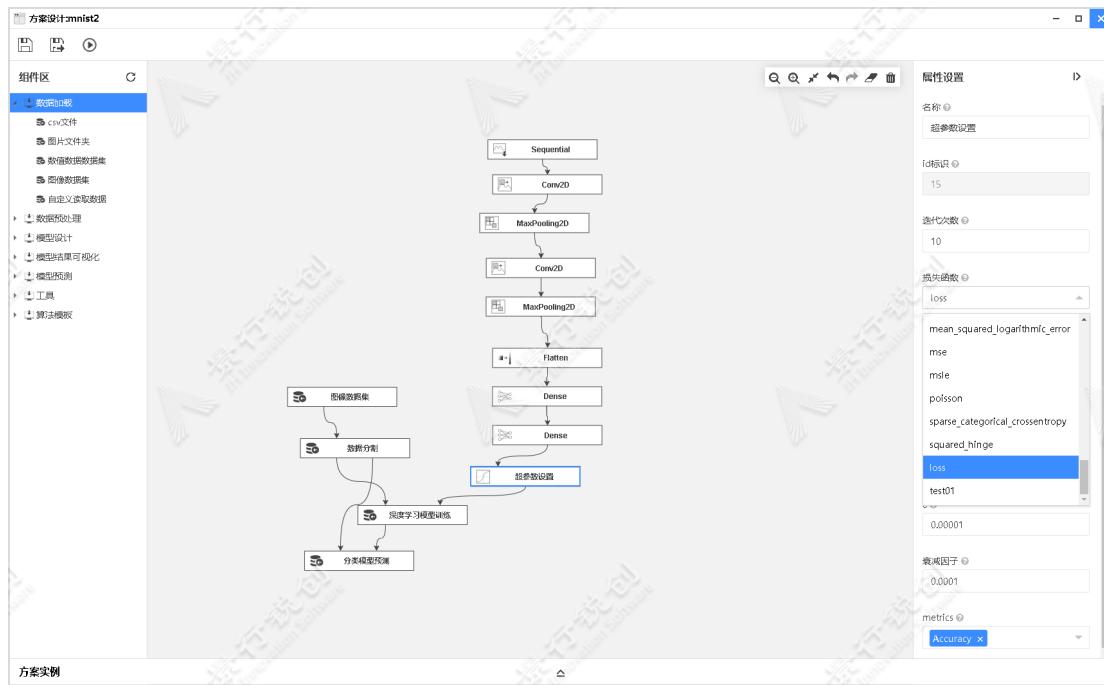
import functools
import tensorflow as tf
from tensorflow.python.keras.losses import LossFunctionWrapper

def scaled_mean_square_error(y_true, y_pred, mean=0., std=1.):
    y_pred = (tf.convert_to_tensor(y_pred) - mean) / std
    y_true = (tf.cast(y_true, y_pred.dtype) - mean) / std
    return tf.keras.metrics.mean_squared_error(y_true, y_pred)

class my_loss_1(LossFunctionWrapper):
    def __init__(self, mean=0., std=1.,
name='scaled_mean_square_error'):
        super(my_loss_1,
self).__init__(functools.partial(scaled_mean_square_error,
mean=mean, std=std), name=name)
```

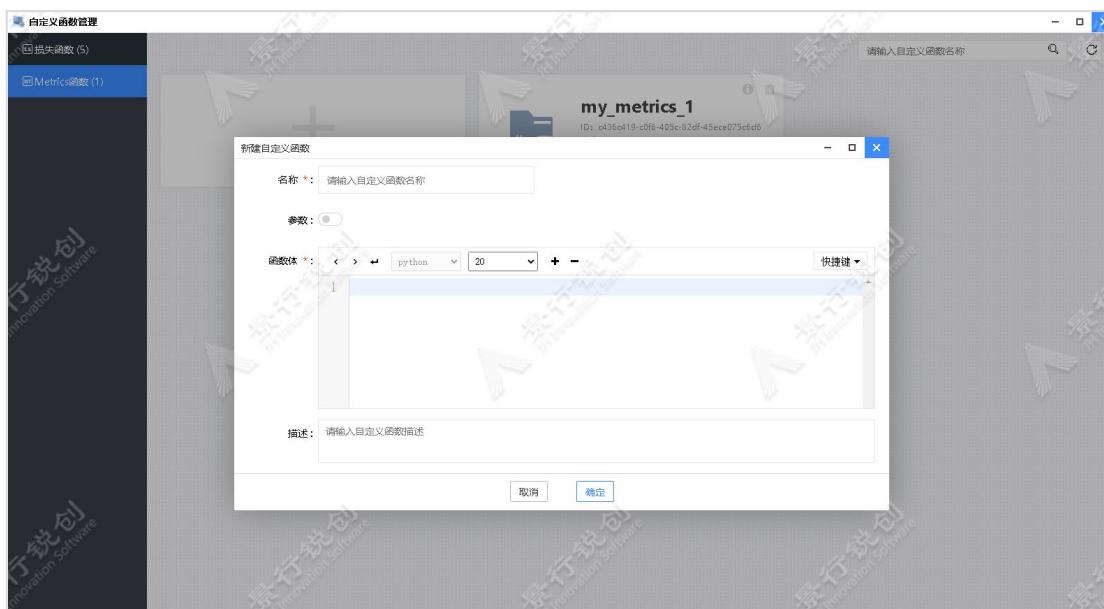
损失函数创建完成后在方案设计中训练深度学习方案结构时可以引用此函数。

第二章



● Metrics 函数

选择左侧“Metrics 函数”，点击“新建自定义函数”卡片，弹出新建窗口，如下图所示：



自定义 Metrics 函数

输入名称，根据函数是否包含参数选择是否打开“参数”开关，在函数体部分输入自定义函数的内容，在描述部分输入对函数的描述或者备注信息，点击“确

定”按钮，即可完成自定义 metrics 函数的创建。

注意：函数“名称”的命名需要和函数入口名称一致。

metrics 函数内容如下：

```
"""
Please input your custom metrics function here.

Note that the entry function name must be consistent
with the defined function name.

"""

import functools

import tensorflow as tf

from tensorflow.python.keras.metrics import MeanMetricWrapper

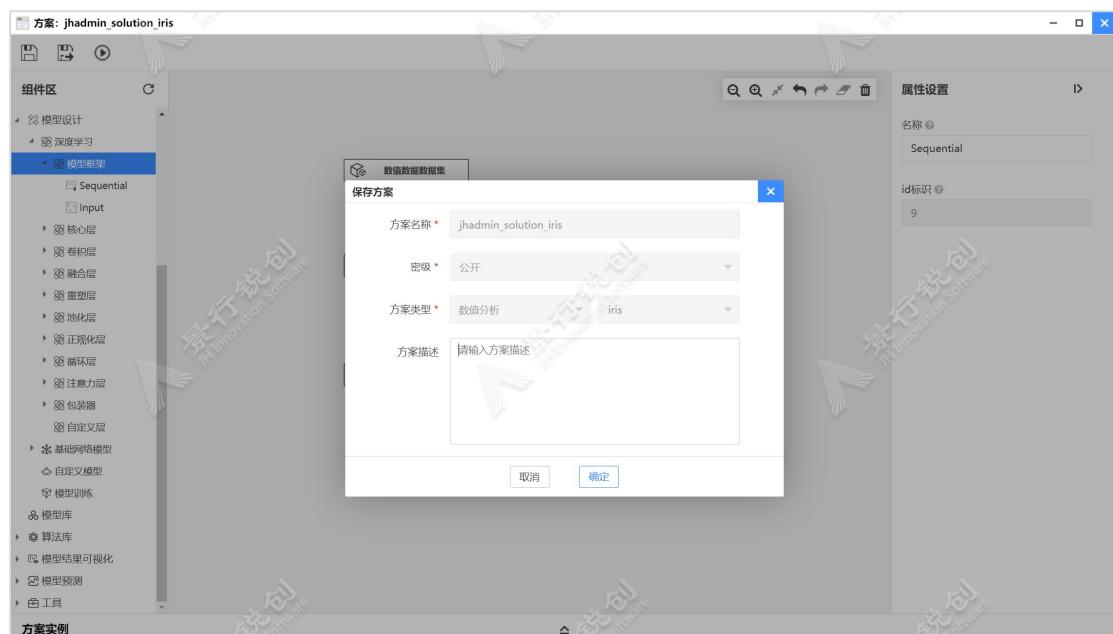

def mean_scaled_relative_error(y_true, y_pred, epsilon=1.):
    y_pred = tf.convert_to_tensor(y_pred)
    y_true = tf.cast(y_true, y_pred.dtype)
    return tf.keras.backend.mean(tf.abs(y_pred - y_true) /
        (tf.abs(y_true) + epsilon), axis=-1)


class my_metrics_1(MeanMetricWrapper):
    def __init__(self, epsilon=0.1,
name='mean_scaled_relative_error', dtype=None):
        super(my_metrics_1,
self).__init__(functools.partial(mean_scaled_relative_error,
epsilon=epsilon), name, dtype=dtype)
```

2.2.1.2. 操作

2.2.1.2.1. 方案保存

点击工具栏中的“保存”按钮，即可将方案描述和画布上的方案设计保存至数据库，当再次打开该方案时，画布上会显示最近一次保存的方案。“保存方案”页面，如下图所示：



保存方案页面

图中每个参数的具体含义如下：

模型名称：任何状态下均不可修改。

密级：任何状态下均不可修改。

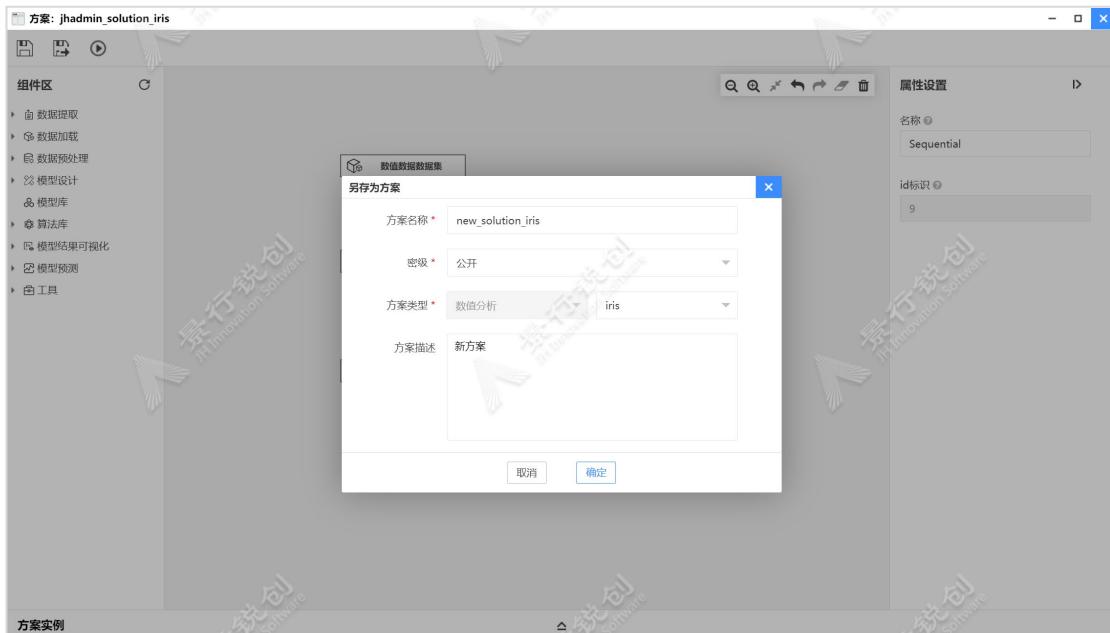
模型类型：任何状态下均不可修改。

模型描述：用于描述方案的相关内容，选填。

2.2.1.2.2. 方案另存为

点击工具栏中的“另存为”按钮，即可将方案另存为一个新的方案，方案可

以是已保存的和未保存的。“另存为方案”页面。如下图所示：



另存为方案页面

图中每个参数的具体含义如下：

方案名称：用户自定义，但是不能与现有的模型名称一致，必填。

密级：对于另存的方案可根据需要修改密级。

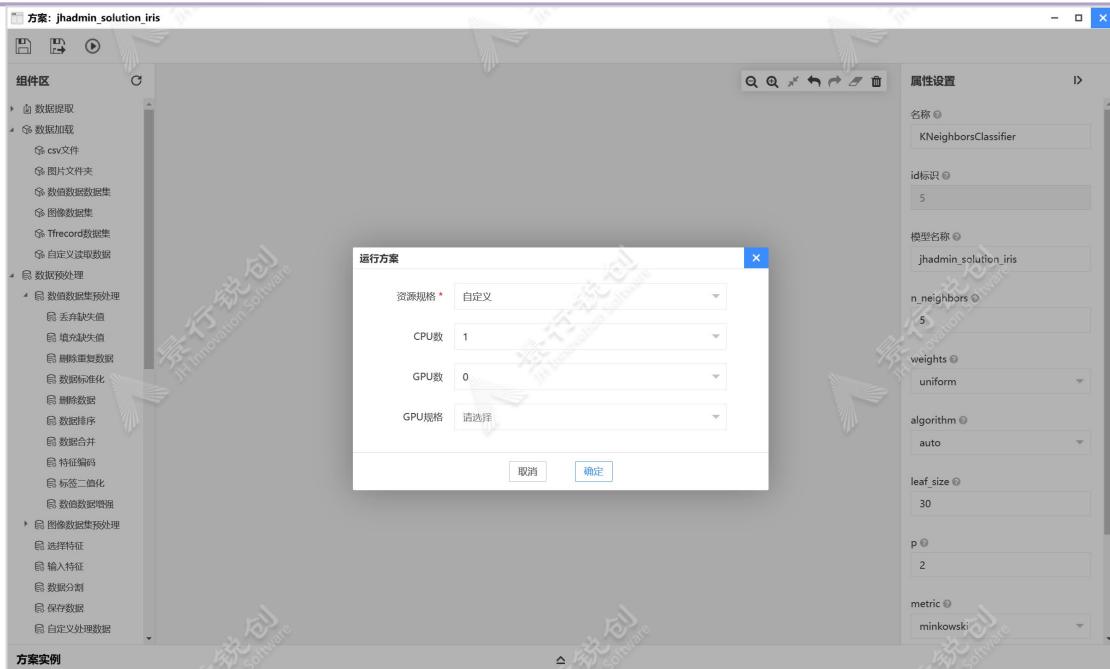
方案类型：默认为当前方案的方案类型，可以选择，也可以自定义方案类型，必填。

方案描述：用于描述方案的相关内容，选填。

2.2.1.2.3. 方案运行

方案设计完成后，点击“运行”按钮，弹出“运行方案”窗口，选择训练方案需要使用的资源，资源选择后，点击“确定”按钮，即可开始训练。如下图所示：

第二章



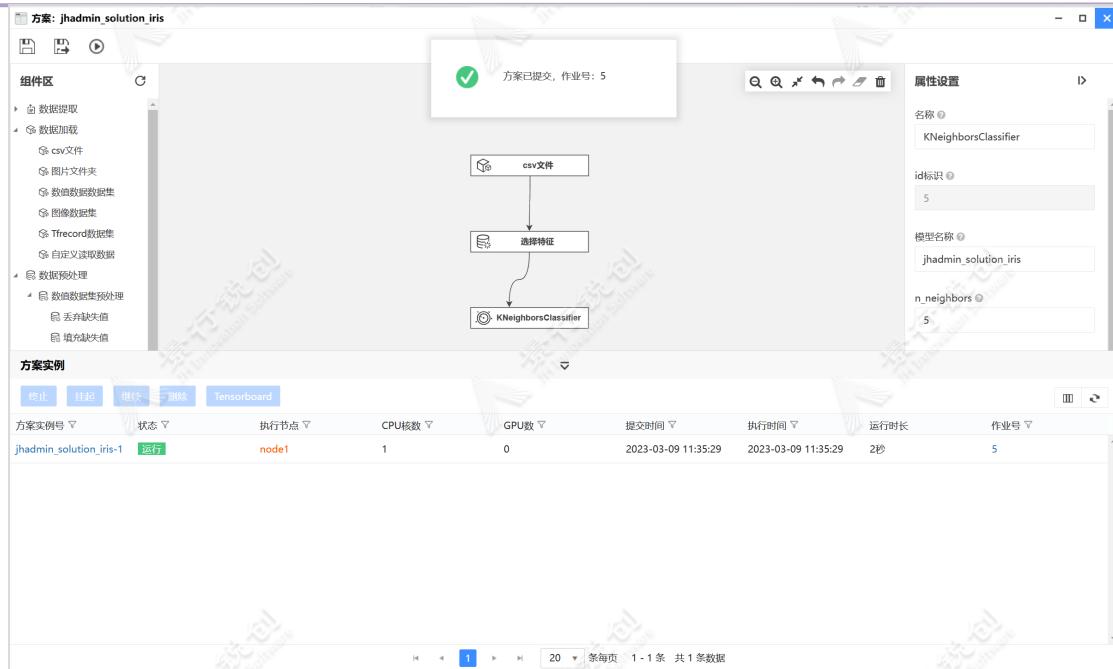
资源配置页面

点击运行按钮，打开选择资源页面，同时进行方案保存。选择资源后，点击确定按钮，即可开始训练方案。

注意：在方案设计页面中，点击“运行”按钮时，错误和问题对应如下：

- 画布中不存在组件时，提示“组件不能为空”。
- 组件的锚点没有连线、必填项属性为空时，提示“请检查异常组件并清除异常”。

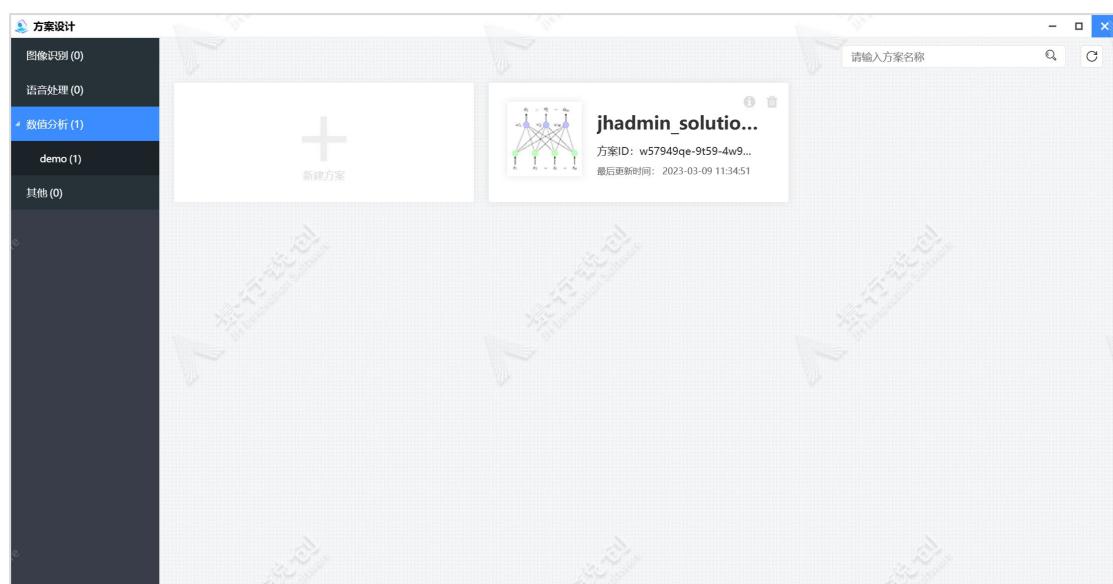
开始训练方案后，会自动打开底部方案实例滑块，如下图所示：



底部方案实例滑块除了只能显示该方案的所有实例外，功能操作与方案实例一致，具体操作详情请看“方案实例”章节。

2.2.1.3. 管理

方案新建成功后，在方案设计应用中会自动生成了一个方案卡片，卡片内容包含方案名称、方案 ID、最后更新时间，点击方案卡片的右上角功能图标，支持对方案进行“发布”，“查看描述”和“删除”操作。



方案管理

2.2.1.3.1. 查看描述

鼠标移至方案卡片上的“描述”按钮时，可以查看方案的描述信息，如下图所示：



点击“描述”的图标按钮，可在弹出窗口中修改方案的描述信息，如下图所示：



2.2.1.3.2. 删除

点击“删除”按钮，可以把该方案卡片删除。



2.2.2. 方案实例

方案实例中以列表的形式记录了所有方案的历史运行记录，用户可以通过实例号、名称等筛选功能快速过滤实例。

普通用户可在方案实例应用中查看自己的所有实例，或在具体的方案设计页面中，通过点击画布下方的方案实例滑块，在展开的面板中查看当前方案实例的信息，亦可对其进行终止、挂起等操作。管理员可在方案实例应用中查看所有用户的方案实例并对其进行操作。

方案实例号	方案名称	用户	状态	执行节点	CPU核数	GPU数	提交时间	执行时间	运行时长	作业号
jhadmin_so...	jhadmin_so...	jhadmin	等待		1	1	2023-03-09...	-	-	7
jhadmin_so...	jhadmin_so...	jhadmin	完成	2*node1	2	0	2023-03-09...	2023-03-09...	7秒	6
jhadmin_so...	jhadmin_so...	jhadmin	完成	node1	1	0	2023-03-09...	2023-03-09...	36秒	5

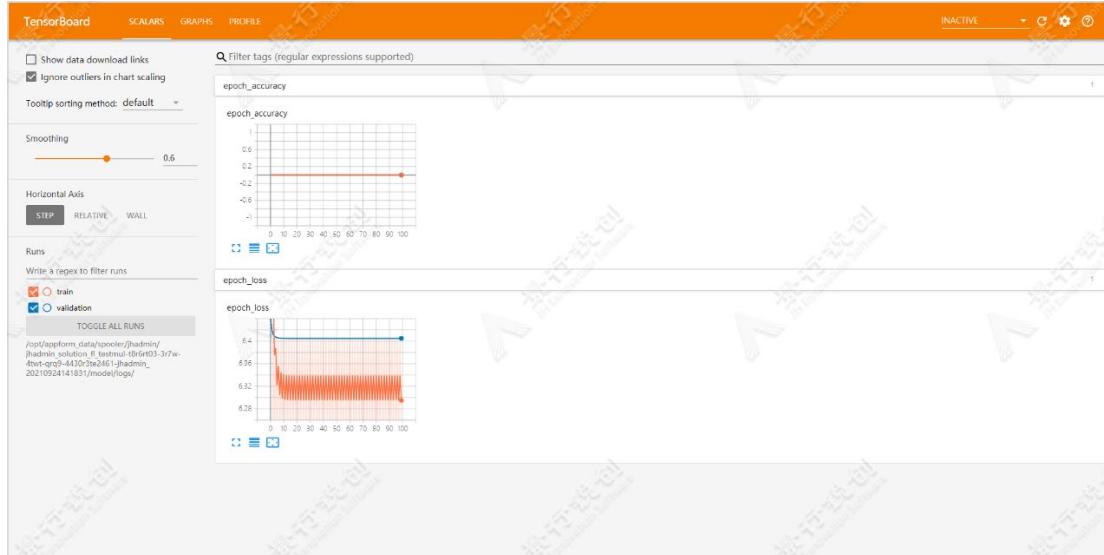
- 功能按钮：

方案实例支持对方案实例进行“终止”、“挂起”、“继续”、“删除”以

第二章

及打开 Tensorboard 操作。

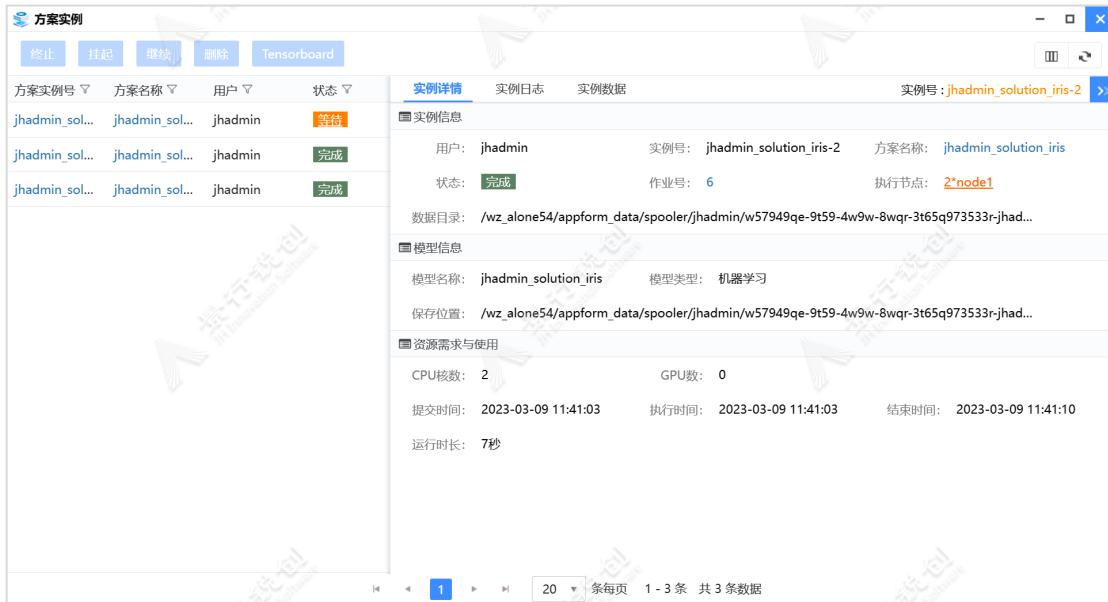
当深度学习的方案运行完成后，点击“Tensorboard”按钮可以以图表形式查看当前实例运行结果。



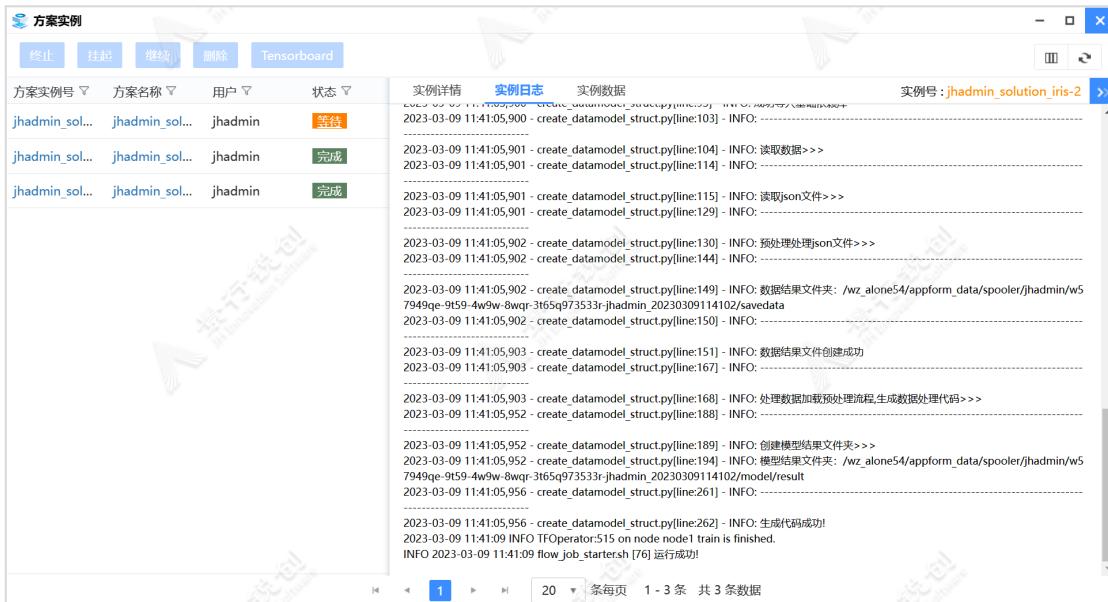
- 方案实例滑块：

点击“方案实例号”或者双击方案实例列表中的某条记录，会打开右侧滑块。滑块有三个标签页，分别为实例详情、实例日志和实例数据。

实例详情：显示方案实例的基本信息、资源信息。点击作业号，可以查看该实例关联的作业信息。实例详情中包含方案实例关联的数据集信息，包括数据集名称、数据集类型、字符集、分隔符以及数据保存目录。实例详情中包含方案实例关联的模型信息，实例详情中会显示模型信息，包括模型名称、模型类型、超参数(仅深度学习模型)。



实例日志: 点击实例日志按钮, 打开实例日志界面。方案实例的状态处于正在运行时, 该实例日志会实时输出。方案实例的状态处于完成时, 会显示该方案实例的所有日志。



实例数据: 点击实例数据按钮, 打开实例数据界面, 显示该方案实例包含的所有相关文件。

第二章

The screenshot shows a software interface for managing project examples. At the top, there are buttons for 'Stop', 'Resume', 'Continue', 'Delete', and 'Tensorboard'. Below this is a table with columns: 'Project Example ID', 'Project Name', 'User', 'Status', and 'Actions'. There are three entries: 'jhadmin_sol...' (Status: Pending), 'jhadmin_sol...' (Status: Completed), and 'jhadmin_sol...' (Status: Completed). The 'Completed' entries have a green background. The 'Completed' row is expanded to show a list of files with details like name, owner, permissions, modification date, size, and type. The files are: 'mlruns' (jhadmin, 读取和写入, 2023-03-09 11: 3 KB, 文件夹), 'model' (jhadmin, 读取和写入, 2023-03-09 11: 3 KB, 文件夹), 'savedata' (jhadmin, 读取和写入, 2023-03-09 11: 3 KB, 文件夹), 'jhadmin_sol' (jhadmin, 读取和写入, 2023-03-09 11: 3 KB, 文本文件), 'jhadmin_sol' (jhadmin, 读取和写入, 2023-03-09 11: 4 KB, 程序), and 'output.6.txt' (jhadmin, 读取和写入, 2023-03-09 11: 6 KB, 文本文件). The bottom of the interface shows a search bar 'w57949qe-9t59-4w9w-8wqr-3t65q...', a page number '1', a page size '20', and a total count '1 - 3 条 共 3 条数据'.

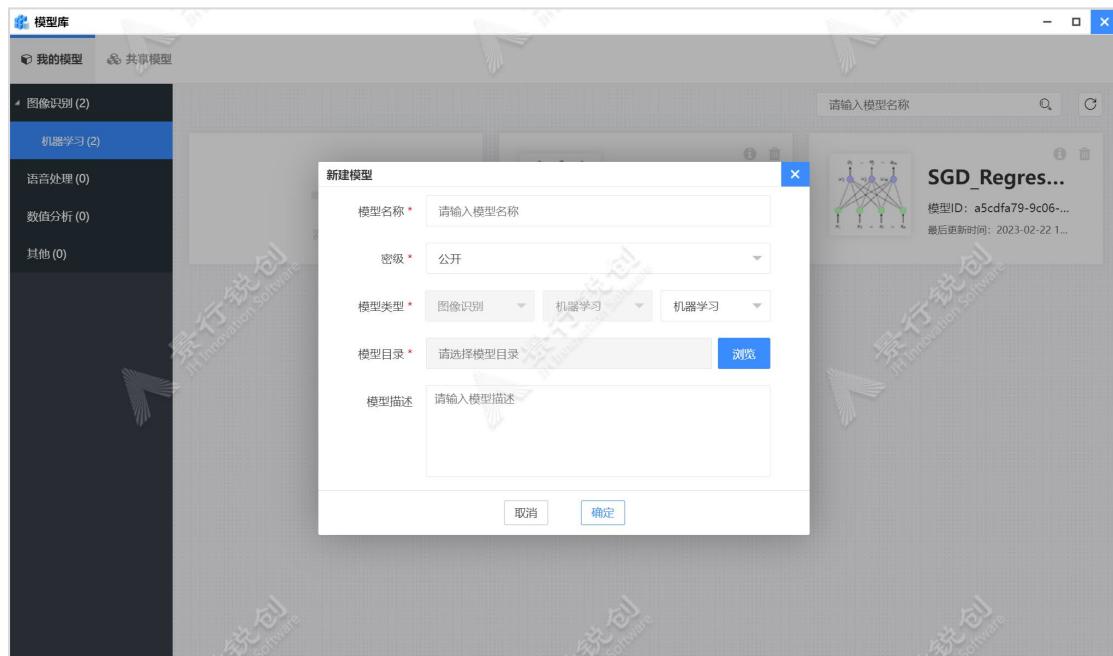
2.2.3. 模型库

模型库方便用户管理所有程序训练产生的模型，通过新建模型卡片来新建自定义模型。点击桌面上的“模型库”桌面图标，进入模型库界面，如下图所示：

The screenshot shows the 'Model Library' interface. On the left, a sidebar lists categories: '我的模型' (My Models) and '共享模型' (Shared Models). Under '我的模型', there are sub-categories: '图像识别 (2)' (Image Recognition), '机器学习 (2)' (Machine Learning), '语音处理 (0)' (Speech Processing), '数值分析 (0)' (Numerical Analysis), and '其他 (0)' (Others). The main area displays a list of models. A search bar at the top right says '请输入模型名称'. There are two visible model cards: 'Spectral_Cl...' (Model ID: b1e73667-1043..., Last updated: 2023-02-22 1...) and 'SGD_Regres...' (Model ID: a5cdfa79-9c06..., Last updated: 2023-02-22 1...). Each card shows a small neural network diagram.

2.2.3.1. 新建模型库

用户可以通过“新建模型”卡片，导入代码训练产生的AI模型。点击“新建模型”卡片，弹出“新建模型”窗口，如下图所示：



新建模型示意图

模型名称：用户自定义，但是不能与现有模型名称一致，必填项。

密级：管理员开启密级功能后，显示此选项，默认为用户密级，用于数据安全、保密。

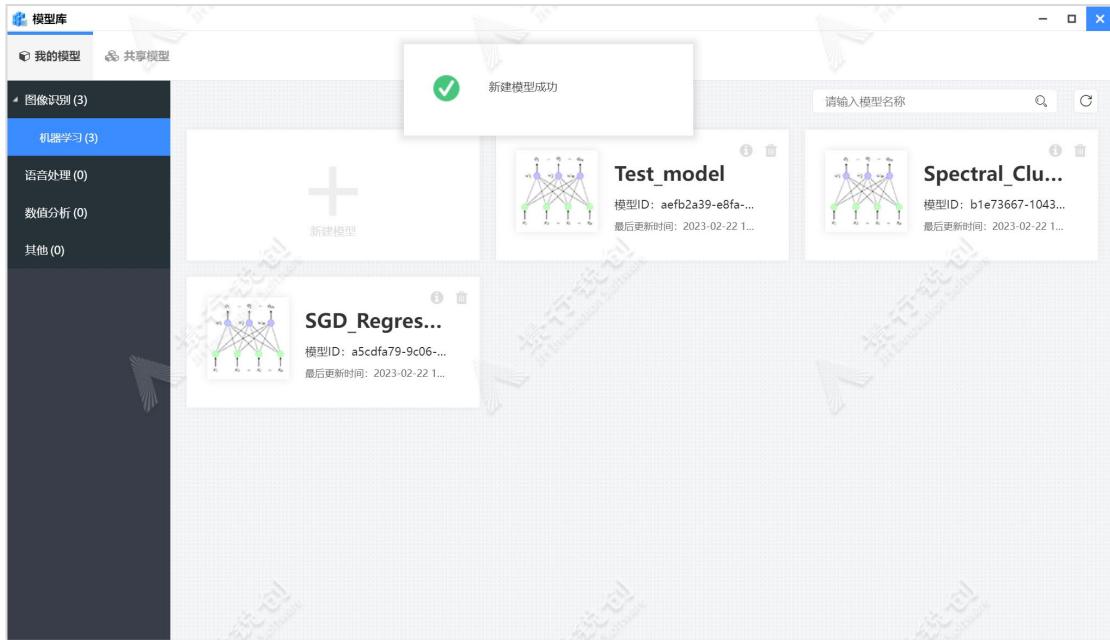
模型类型：默认为已选的模型类型，也可以自定义模型类型，必填项。

模型目录：选择模型文件所在的目录，必填项。

模型描述：用于描述模型结果的相关内容，选填项。

点击“确定”按钮后，创建模型卡片。模型卡片创建成功后，会提示“新建模型成功”，如下图所示：

第二章



新建模型示例图

2.2.3.2. 模型训练历史列表

点击模型卡片，打开模型训练历史列表，方案运行实例产生的模型结果都存储于模型训练历史列表中，点击模型版本号或双击模型所在行则会展开滑块，滑块中包含“模型详情”和“模型结果”两个页签，在对应的板块中可以查看模型相关的信息，如下图所示：

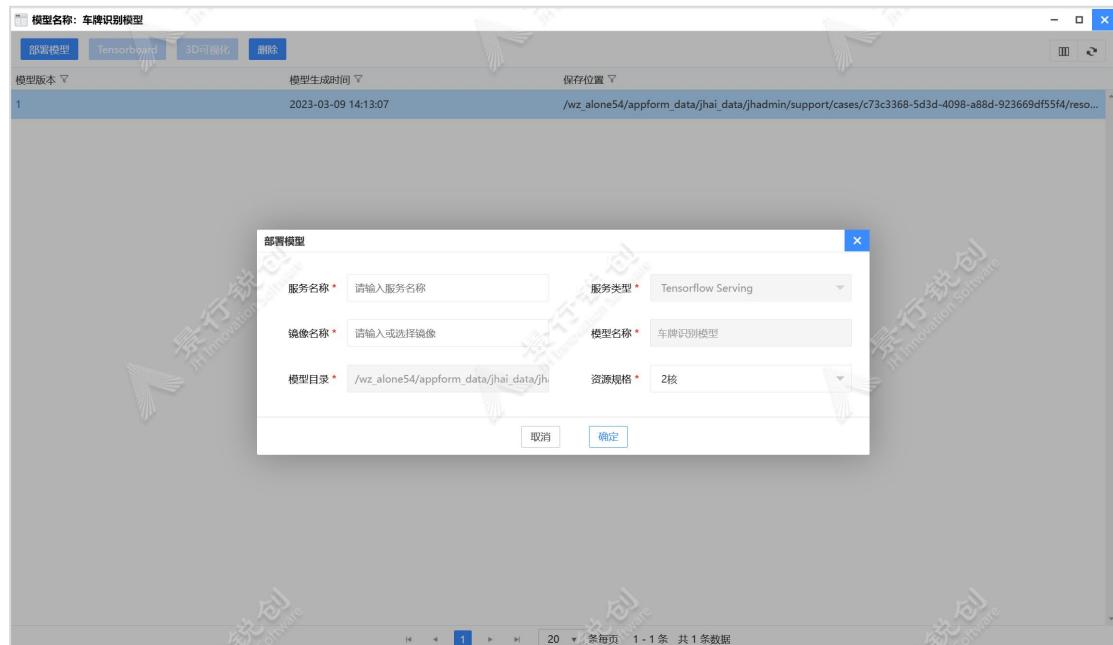


模型训练历史列表中可以进行部署模型、Tensorboard、3D 可视化、删除和

共享操作。

1. 模型部署:

模型部署将训练好的 AI 模型部署为 HTTP 服务, 可以供外部程序调用实现推理功能, 目前模型部署仅支持深度学习模型。点击模型部署按钮, 会打开部署模型窗口, 如下图所示:

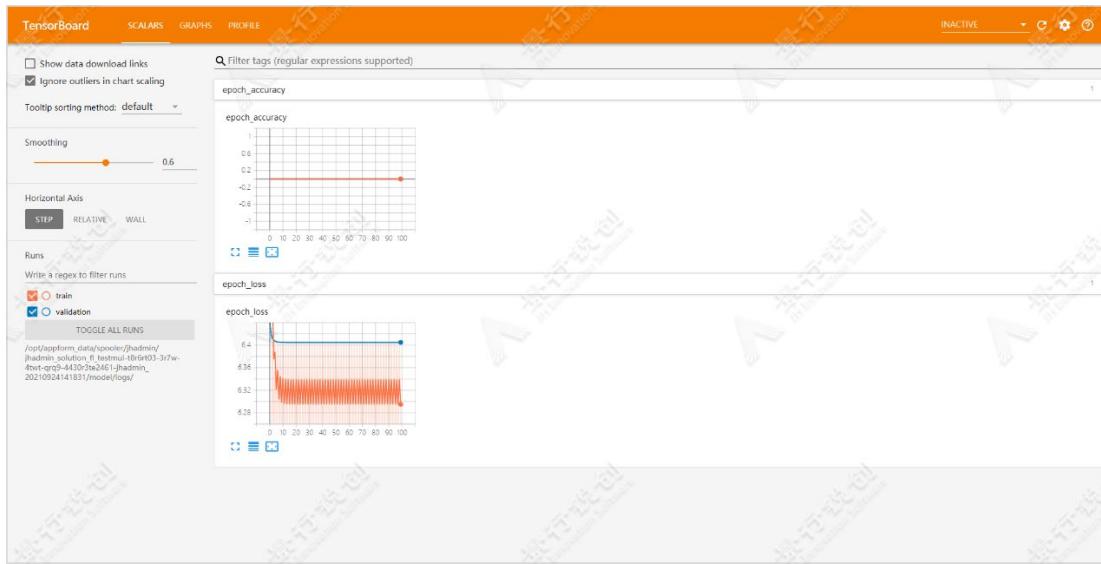


填写服务名称、选择资源。点击确定按钮, 提交模型服务的作业。



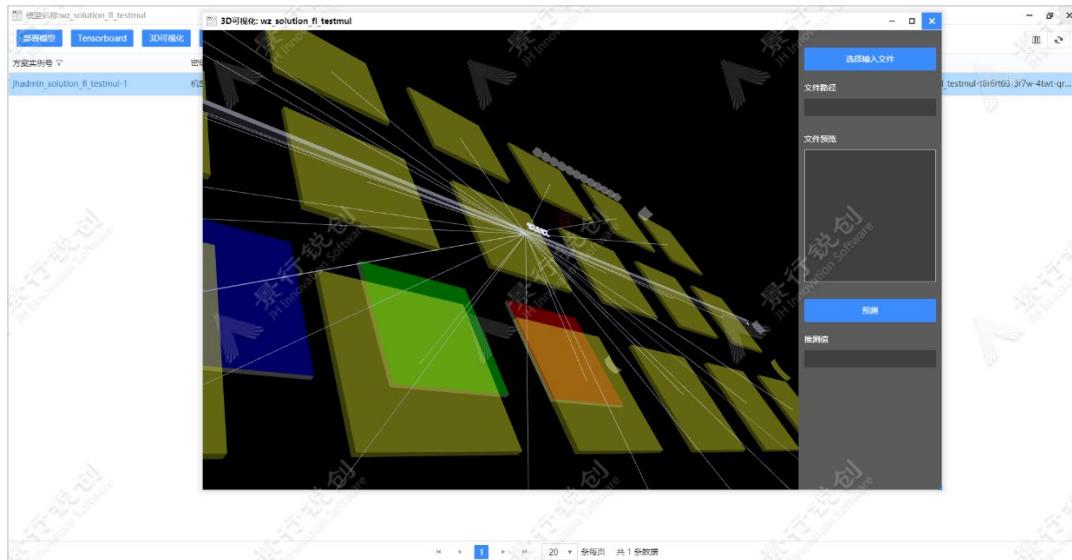
2. Tensorboard

打开 Tensorboard 查看模型的指标和结构, 目前仅支持深度学习模型。选中训练历史中某条记录, 点击 Tensorboard 按钮, 即可打开 Tensorboard 服务页面。



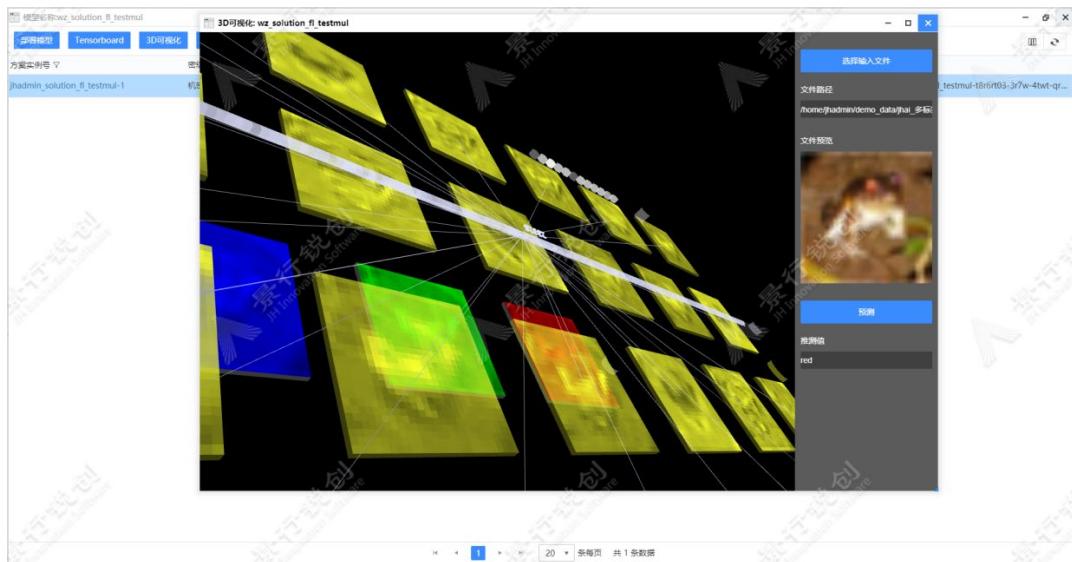
3. 3D 可视化

3D 可视化功能支持在 Web 浏览器中以 3D 的形式显示模型结构，目前仅支持部分深度学习模型。点击 3D 可视化按钮，弹出该模型结果的 3D 可视化页面，点击窗口的彩色块，可查看模型结构和配置参数等。



深度学习模型-3D 可视化

页面右侧，点击“选择输入文件”选择需要预测的数据，接着点击“预测”，可实现模型预测，在“推预测值”位置显示预测的具体值。可视化页面如下：



3D 可视化模型预测

4. 删除:

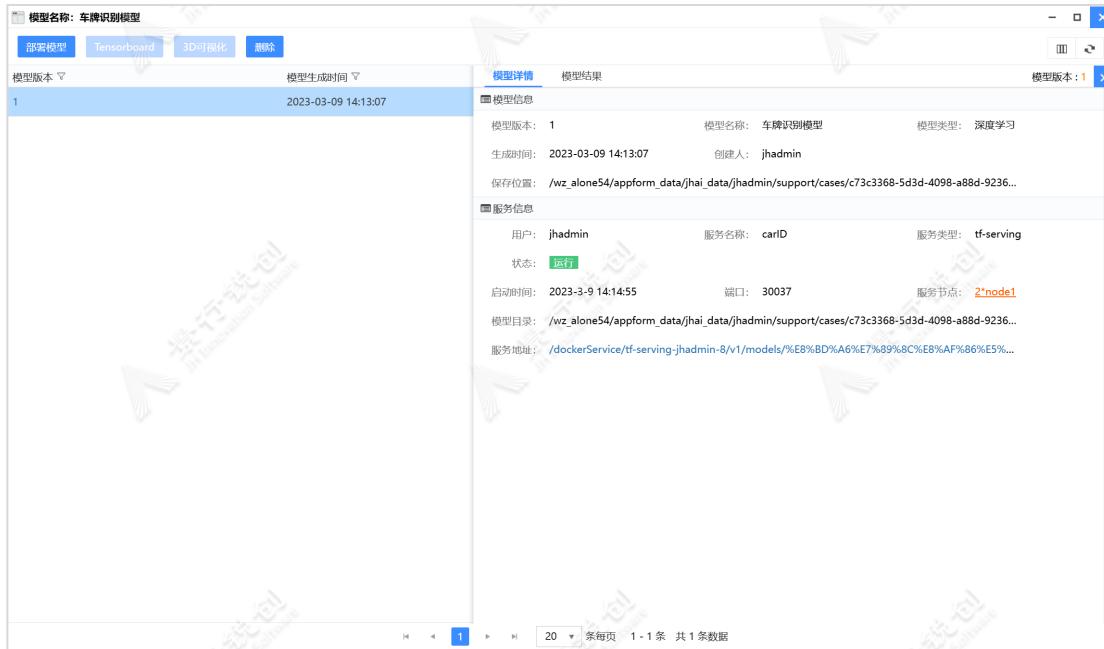
点击删除按钮，可以删除该模型训练历史记录。



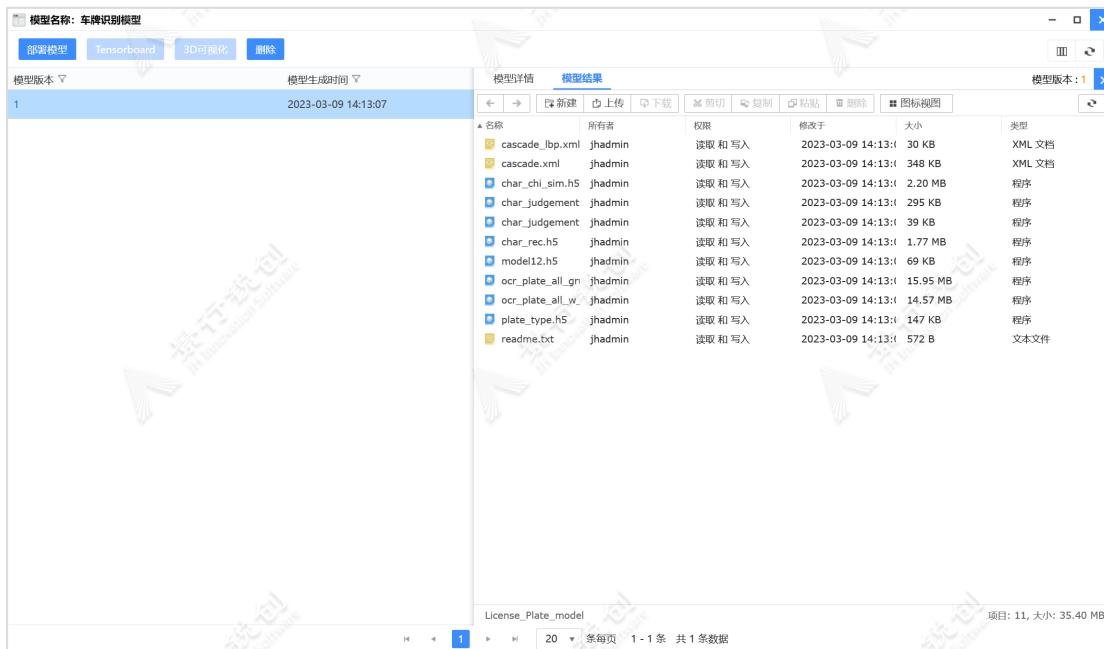
点击训练历史列表中某条记录的方案实例号或者双击某条记录，会打开右侧滑块。滑块中有两个标签页，分别为“模型详情”和“模型结果”。

➤ **模型详情:** 显示模型的基本信息和关联的服务信息。如下图所示:

第二章



➤ **模型结果:** 点击模型结果按钮, 打开模型结果界面, 显示模型的结果文件与日志文件, 如下图所示:



2.2.3.3. 模型卡片

方案训练成功后, 模型库自动生成模型卡片, 模型卡片包含模型名称、模型ID、最后更新时间等信息, 模型支持“修改描述”和“删除”操作。



2.2.3.4. 共享模型

共享模型用以给共享组成员或指定成员在方案设计和服务发布中使用。

我的模型共享至共享模型

注意：添加共享模型之前需要先创建共享组。

以车牌识别模型为例，步骤如下：

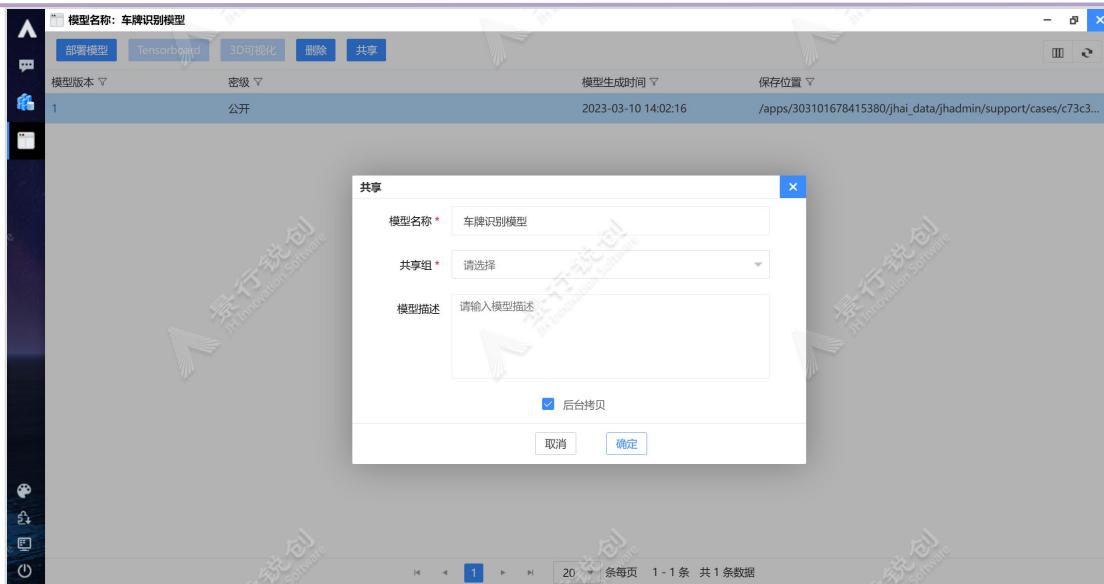
依次点击“我的模型” – “图像识别” – “车牌识别”，然后在右侧的工作区，点击“车牌识别模型”卡片，打开模型版本列表，如下图所示：

第二章



模型版本

选中某一个模型版本，点击顶部“共享”按钮，弹出共享窗口，如下图所示：



共享模型

图中每个参数的具体含义如下：

模型名称：共享后的模型名称，输入任意符合命名规则的名称即可。

共享组：选择可见的共享组，选项从我的数据->共享数据区获取。

共享用户：选择共享组之后显示，显示为选择的共享组成员。

共享目录：选择共享组之后显示，只能在所选共享组的共享数据区选择文件夹，点击蓝色的“文件夹图标”按钮，然后在弹出的“选择文件夹”窗口中选择文件夹即可，用作存放源数据文件目录，只允许选择一个文件夹。

模型描述：共享后的数据集描述信息。

后台拷贝：勾选后采用后台形式拷贝源模型文件至共享目录，当前共享页面退出；不勾选则采用前台拷贝方式，当前共享页面不退出。

在共享窗口中填写完成后，点击“确定”按钮，可以保存此次共享的模型。

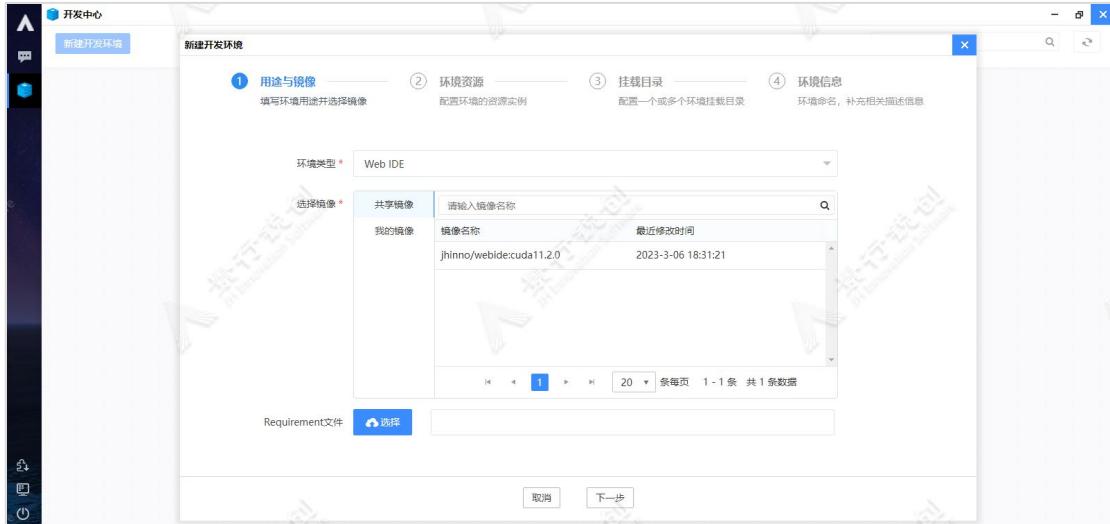
2.3. 开发中心

开发中心支持人工智能模块在线安全编程环境，通过 web 方式调试和编写 AI 代码，集成了“Web IDE”、“CentOS 桌面”、“Ubuntu 桌面”和“SSH”四种环境类型。满足用户在开发中心编写和调试模型训练代码，然后基于该代码

进行模型的训练。

2.3.1. 新建开发环境

以新建“Web IDE”类型的开发环境为例，操作步骤如下：点击“新建开发环境”按钮，弹出“新建开发环境”窗口，如下图所示：



新建开发环境

图中每个参数的具体含义如下：

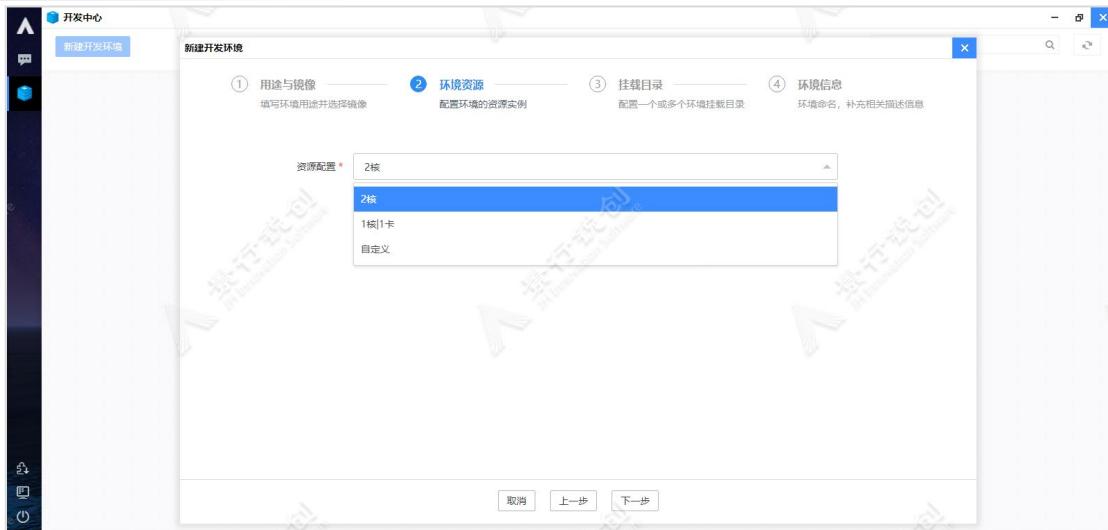
环境类型：选择新建环境的类型，支持四种类型的环境，如下所示：

- Web IDE：该环境集成了 Jupyter 和 VScode 开发工具。
- CentOS 桌面：提供带 GUI 的操作系统环境。
- Ubuntu 桌面：提供带 GUI 的操作系统环境。
- SSH：仅提供 WEB 终端。

选择镜像：选择创建环境实例所用的模板镜像。

Requirement 文件：根据选择的配置文件，快速配置环境实例，可选。

选择完环境类型和镜像后，点击“下一步”按钮进入环境资源配置页面：

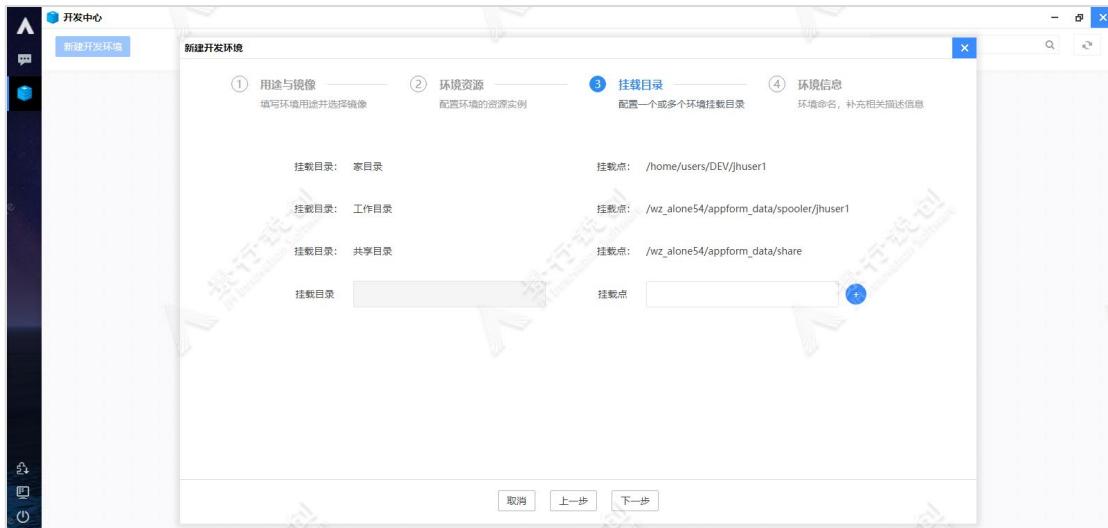


环境资源

图中每个参数的具体含义如下：

资源配置：默认为 2 核，选择新建环境实例所需要的资源配置。

选择合适资源规格后，点击“下一步”按钮，进入挂载目录页面：



挂载目录

默认会把“家目录”、“工作目录”、“共享目录”挂载到容器中，也可以额外指定目录，挂载到开发环境容器的中。

完成挂载目录配置，点击“下一步”按钮配置环境名称和描述，如下图所示：

第二章



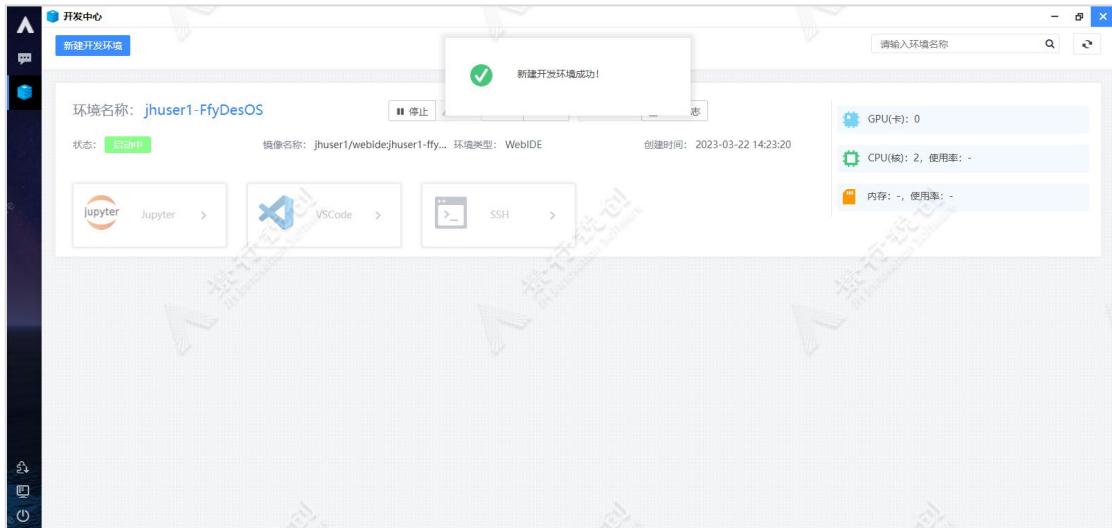
环境信息

图中每个参数的具体含义如下：

环境名称：默认系统会回填一个可用的环境名称。

环境描述：输入环境描述信息，可选。

点击“创建”按钮，即可完成开发环境的创建，如下图所示：



开发环境列表

2.3.2. 使用开发环境

2.3.2.1. 应用 Jupyter 服务

点击“Web IDE”类型的开发环境实例上的“Jupyter”卡片按钮，访问 Jupyter 服务，运行程序的操作方式有 2 种：

- 直接运行。
 - 以提交作业的方式运行程序。

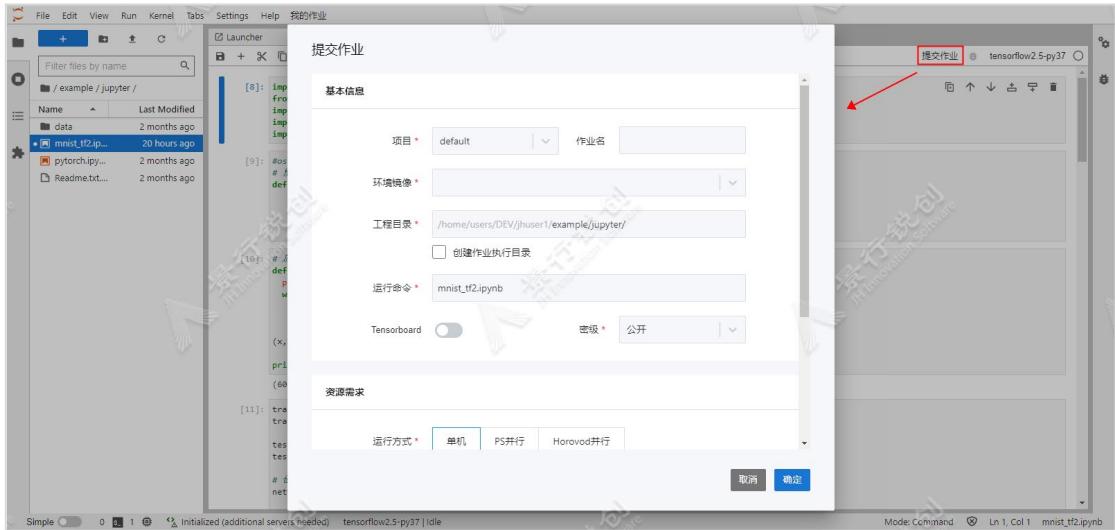
（一）直接运行程序

点击右上角的“运行”按钮，运行程序，如下图所示：

（二）以提交作业的方式运行程序

选择“ipynb”格式的训练程序后，点击“提交作业”按钮，弹出“提交作业”窗口，如下图所示：

第二章



提交 AI 作业示意图

作业提交参数：

项目：指定项目，默认为：default。

作业名：指定作业名。

环境镜像：必选项，选择程序运行环境的镜像，需要根据运行程序选择合适的镜像。

工程目录：指定代码的工程目录，默认回填运行程序的父目录。

创建工作执行目录：默认不勾选，直接在工程目录中运行程序。手动勾选，会在“我的作业-作业数据区”中创建临时执行目录，将工程目录中的数据复制至其中后，进行运行程序操作。

运行命令：必填项，指定运行程序的入口文件和程序参数，例如：train.py --data-path=/data --batch=54。

Tensorboard：选择是否允许访问 tensorboard 服务。若启动，作业详情页面，点击“访问 Tensorboard”按钮，即可查看模型训练情况。

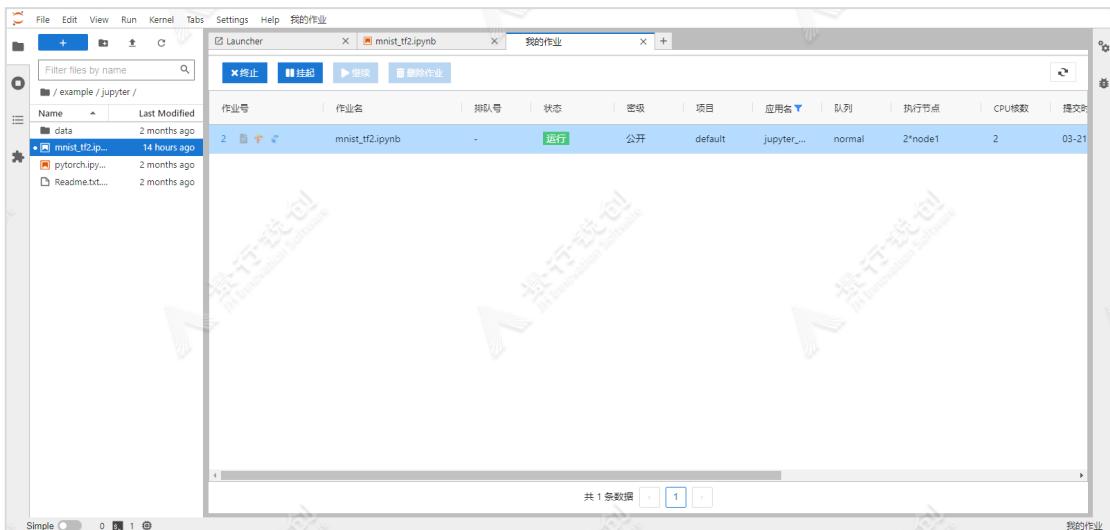
密级：管理员开启密级功能后，显示此选项，默认为用户密级，用于数据安全、保密。

运行方式：选择作业运行方式，如下所示：

- **单机：**默认选择单机，仅单节点上运行训练程序。
- **PS 并行：**提交以 Parameter Server 方式实现的训练代码，支持跨节点，需填写并行相关参数，如下所示：

- **ParamServer 数:** 指定参数服务的数量。
 - **Worker 数:** 指定训练服务数量。
 - **实例规格:** 分别指定每个 ParamServer 和 Worker 使用的资源。
 - **Horovod 并行:** 提交以 Horovod 实现的并行 AI 训练程序，需要指定 Horovod 相关参数。Horovod 并行模式需要模型训练代码以 Horovod 库的并行方式进行编码，如下所示：
 - **并行实例数:** 指定并行训练的节点数量。
 - **实例规格:** 指定每个并行节点使用的资源。
- 环境变量:** 指定程序运行需要的额外环境变量。

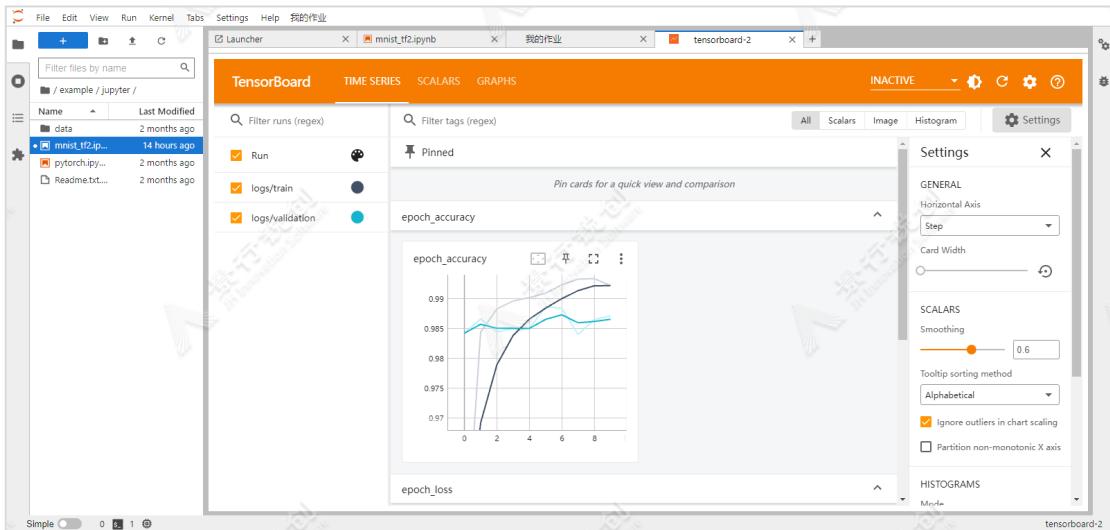
点击“我的作业”按钮，打开我的作业 tab 页，如下图所示：



我的作业示意图

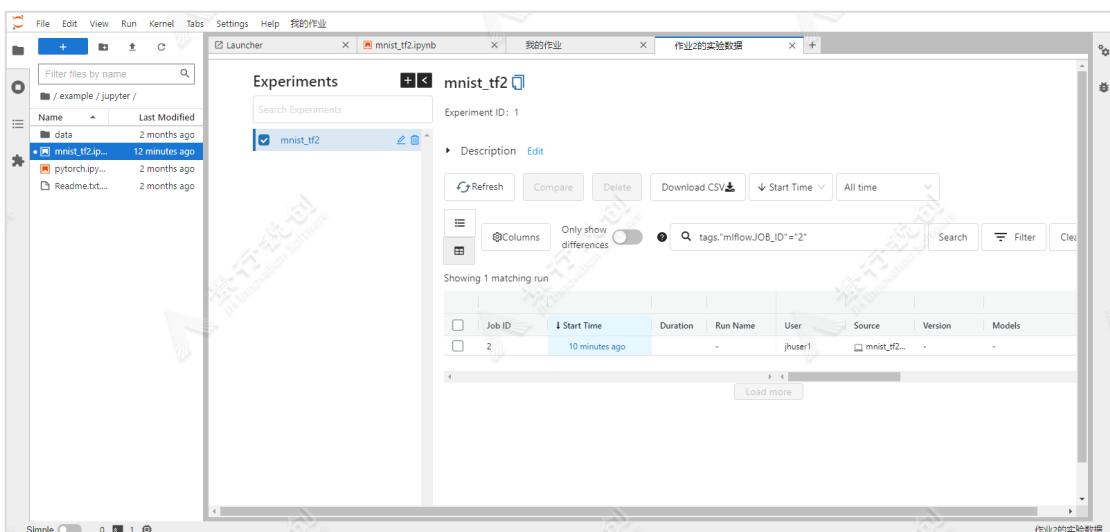
点击作业实例上的“Tensorboard”图标按钮，访问 Tensorboard 服务查看模型数据，如下图所示：

第二章



访问 Tensorboard 服务示意图

点击作业实例上的“实验管理”图标按钮，查看实验数据，如下图所示：



查看实验数据示意图

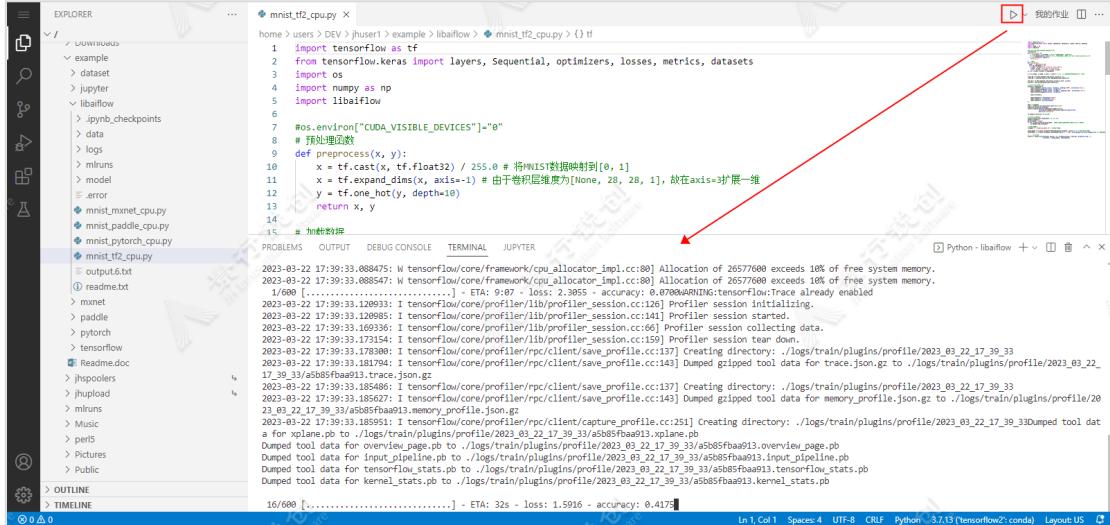
2.3.2.2. 应用 VSCode 服务

点击“Web IDE”类型的开发环境实例上的“VSCode”卡片按钮，访问 VSCode 服务，运行程序的操作方式有 2 种：

- 直接运行。
- 以提交作业的方式运行程序。

（一）直接运行程序

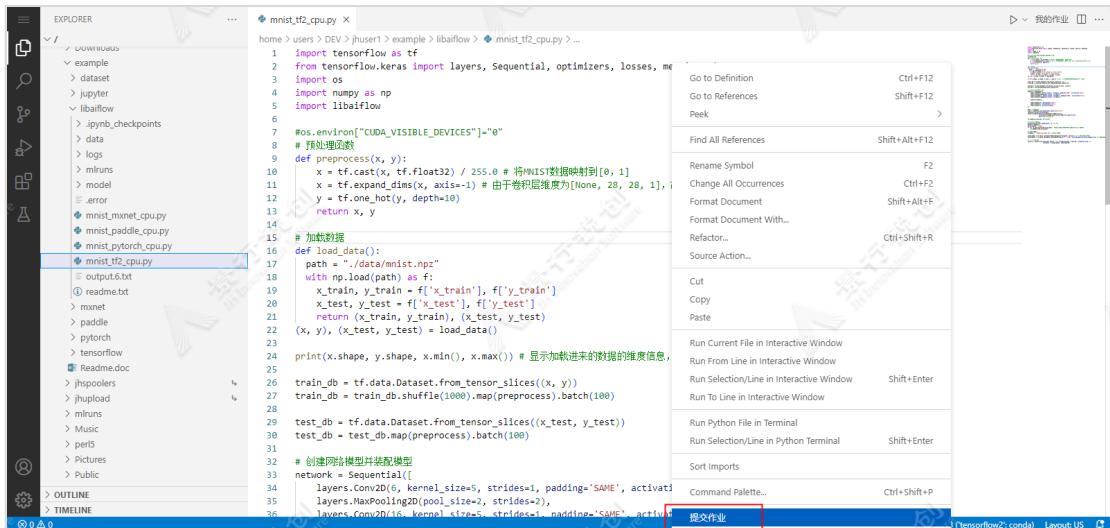
打开执行程序后，点击右上角的“运行”按钮，运行程序，如下图所示：



运行程序示意图

（二）以提交作业的方式运行程序

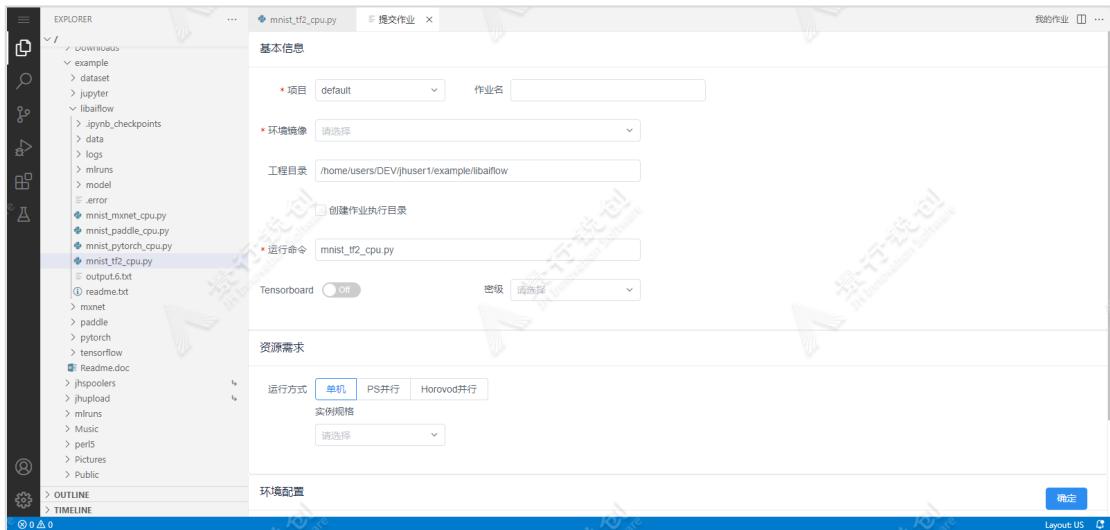
Explorer 中选中文件或者打开文件，右键打开右键菜单，如下图所示：



右键菜单示意图

点击“提交作业”选项，弹出“提交作业”弹窗，如下图所示：

第二章



作业提交页面示意图

作业提交参数:

项目: 指定项目, 默认为: default。

作业名: 指定作业名。

环境镜像: 必选项, 选择程序运行环境的镜像, 需要根据运行程序选择合适的镜像。

工程目录: 指定代码的工程目录, 默认回填运行程序的父目录。

创建工作执行目录: 默认不勾选, 直接在工程目录中运行程序。手动勾选, 会在“我的作业-作业数据区”中创建临时执行目录, 将工程目录中的数据复制至其中后, 进行运行程序操作。

运行命令: 必填项, 指定运行程序的入口文件和程序参数, 例如: train.py --data-path=/data --batch=54。

Tensorboard: 选择是否允许访问 tensorboard 服务。若启动, 作业详情页面, 点击“访问 Tensorboard”按钮, 即可查看模型训练情况。

密级: 管理员开启密级功能后, 显示此选项, 默认为用户密级, 用于数据安全、保密。

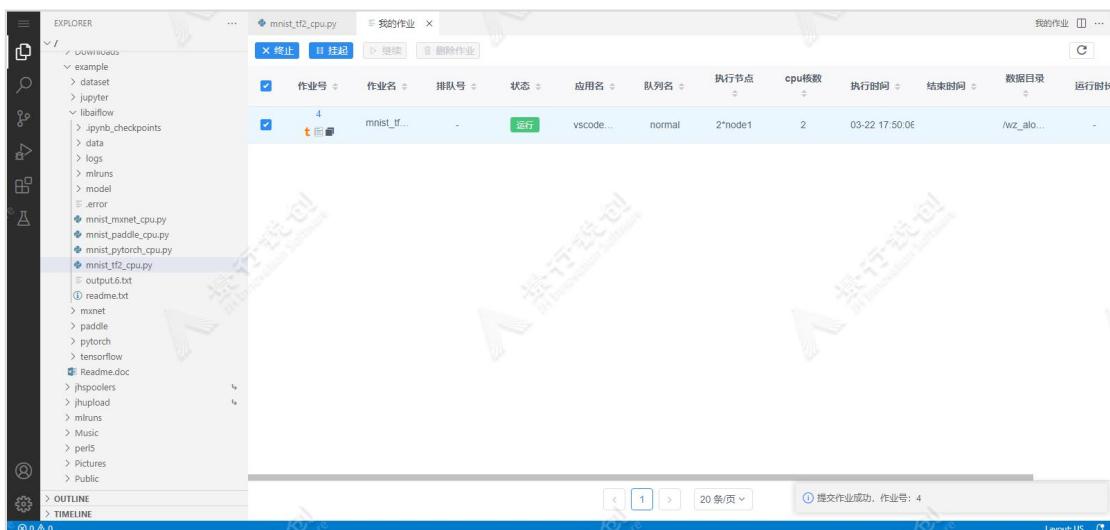
运行方式: 选择作业运行方式, 如下所示:

- **单机:** 默认选择单机, 仅单节点上运行训练程序。
- **PS 并行:** 提交以 Parameter Server 方式实现的训练代码, 支持跨节点, 需填写并行相关参数, 如下所示:

- **ParamServer 数:** 指定参数服务的数量。
- **Worker 数:** 指定训练服务数量。
- **实例规格:** 分别指定每个 ParamServer 和 Worker 使用的资源。
- **Horovod 并行:** 提交以 Horovod 实现的并行 AI 训练程序，需要指定 Horovod 相关参数。Horovod 并行模式需要模型训练代码以 Horovod 库的并行方式进行编码，如下所示：
 - **并行实例数:** 指定并行训练的节点数量。
 - **实例规格:** 指定每个并行节点使用的资源。

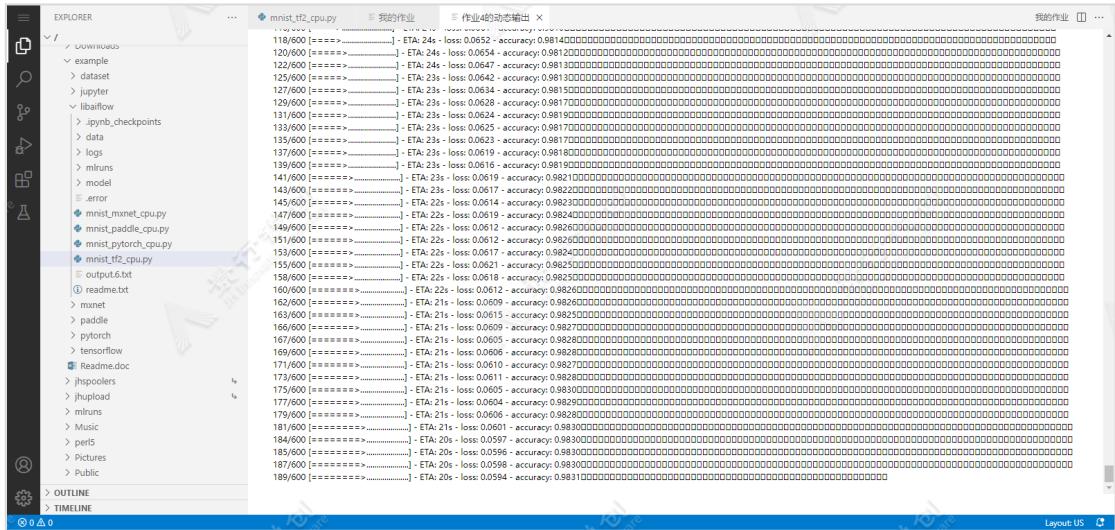
环境变量: 指定程序运行需要的额外环境变量。

作业提交后，会自动打开“我的作业”页面，方便用户在此查看提交作业的信息，如下图所示：

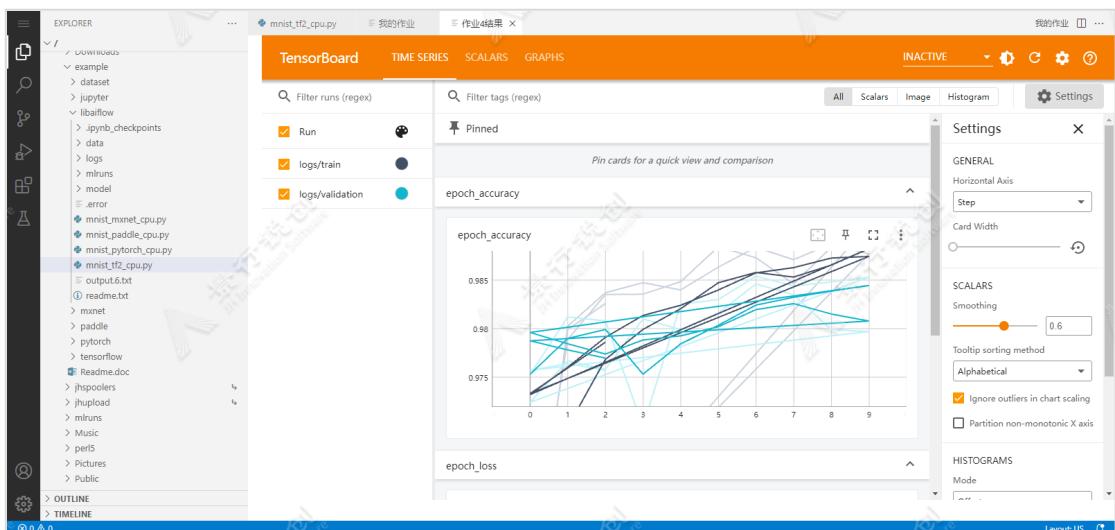


点击作业实例上的“动态输出”图标按钮，查看作业日志的动态输出，如下图所示：

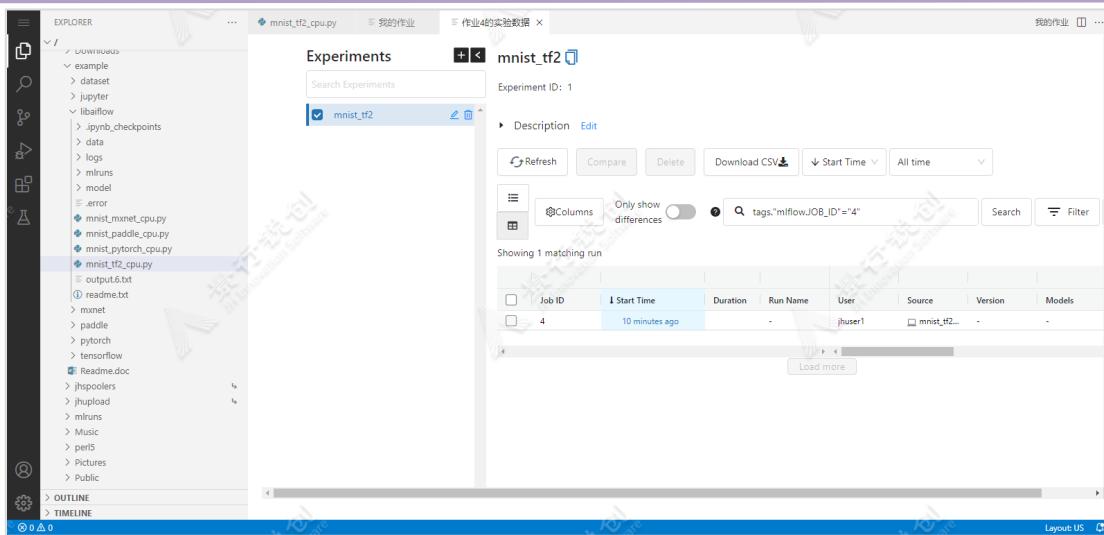
第二章



点击作业实例上的“Tensorboard”图标按钮，访问Tensorboard服务查看模型数据，如下图所示：



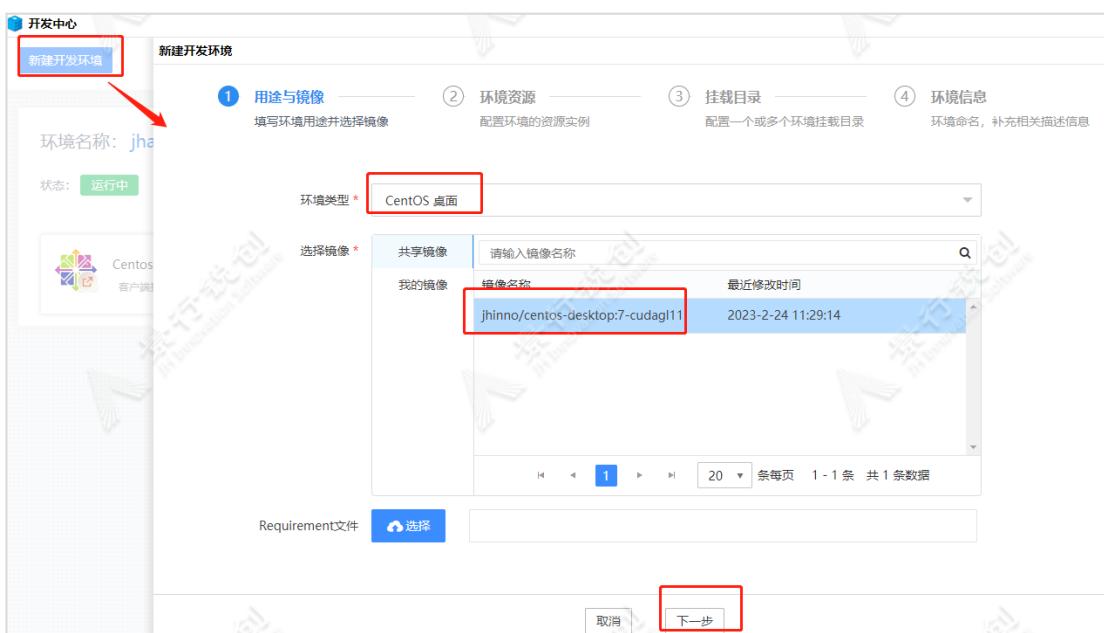
点击作业实例上的“实验管理”图标按钮，查看实验数据，如下图所示：



查看实验数据示意图

2.3.2.3. 应用 CentOS 桌面/Ubuntu 桌面

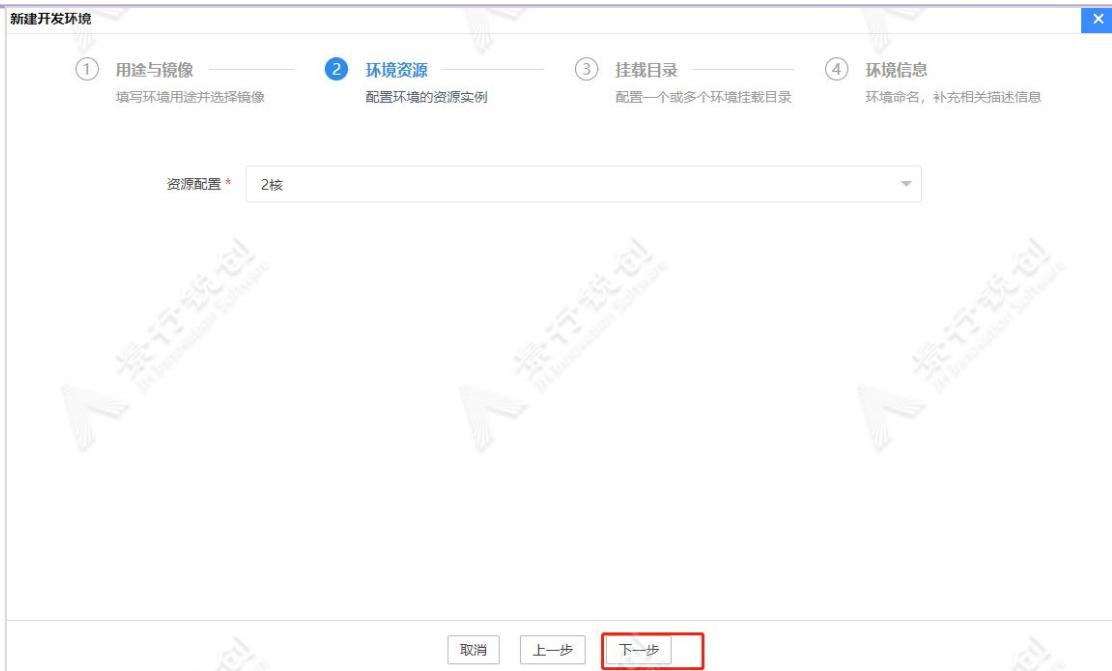
创建 CentOS 桌面/Ubuntu 桌面开发环境，以 CentOS 为例，具体步骤如下：



新建开发环境

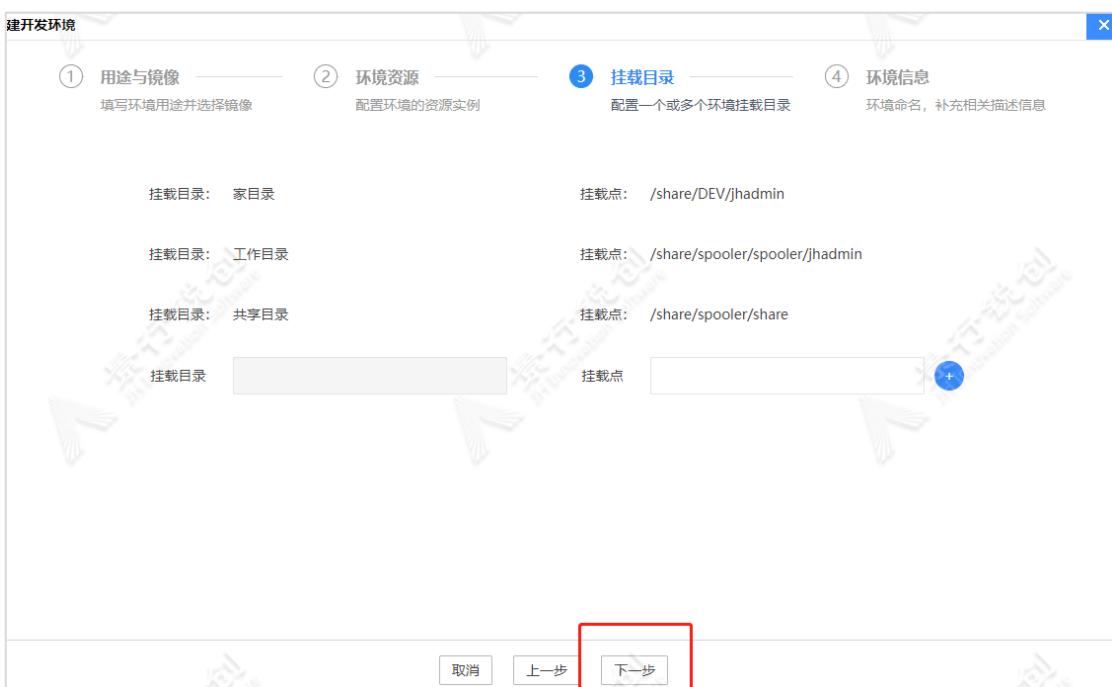
在“资源配置”中选择合适的资源规格，然后点击“下一步”：

第二章



环境资源

在挂载目录页面选择需要挂载的目录，正常情况下使用默认挂载即可，点击“下一步”：



挂载目录

输入环境名称，点击“创建”。



环境信息

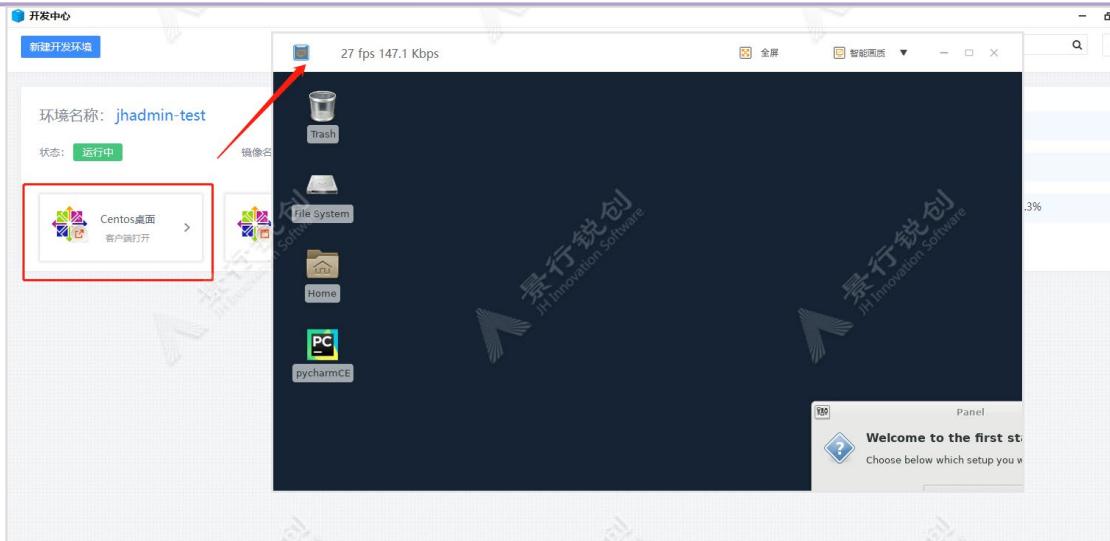
此时，在开发中心下可以看到 CentOS 桌面开发环境卡片：



CentOs 桌面卡片

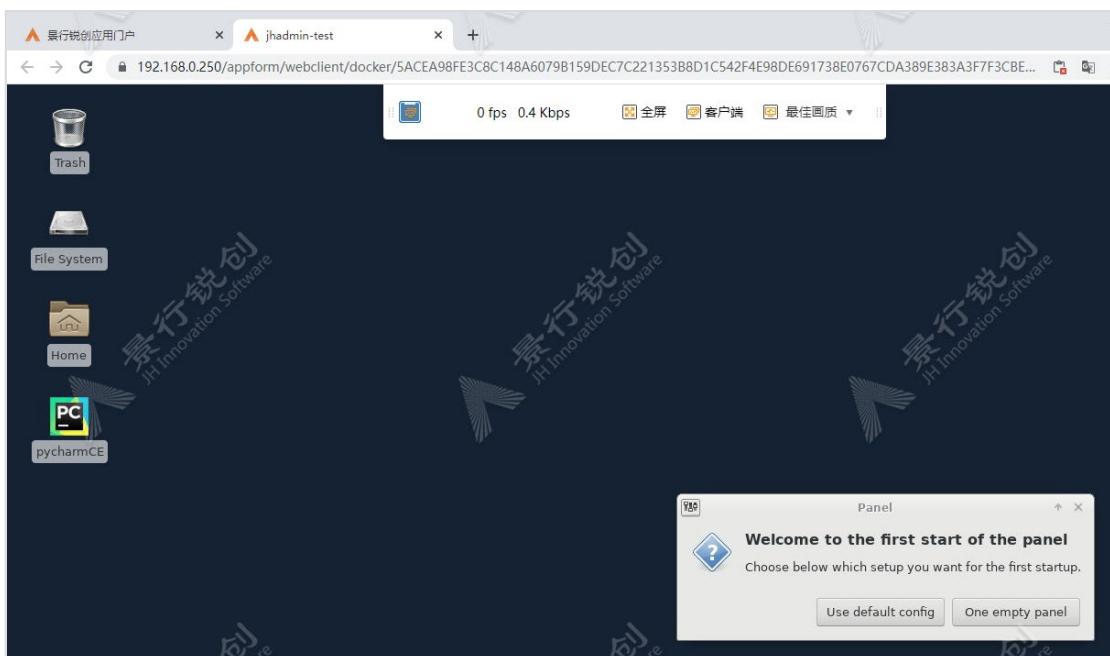
点击“CentOs 桌面 客户端打开”可以用客户端形式打开 CentOs 桌面，前提：机器上需要安装 JH_Client_V5.4_Win_x86_r3****.exe 软件包：

第二章



Centos 桌面客户端界面

点击“Centos 桌面 WEB 打开”，此时会在 WEB 上打开 Centos 桌面：



Centos 桌面 WEB 页面

在打开的 Centos 桌面中正常使用即可。

注意：此 Centos 桌面和普通的 Centos 物理机桌面稍有不同，例如：该 Centos 桌面不支持 mount 文件，不支持粘贴文件、图片等。

2.4. AI 作业提交

AI 平台集成了各类主流的深度学习框架，包括 Tensorflow、Pytorch、Mxnet 和 PaddlePaddle，并提供相应的 Web 页面以提交深度学习作业。

提供 Tensorflow、Pytorch、Mxnet 和 Paddle 等框架编写模型的 GPU 或 CPU 并行训练功能，支持单机多 GPU，多机多 GPU，多机多 CPU 的训练方式。

支持 Tensorflow、Pytorch、Mxnet 和 Paddle 等机器学习作业的容器封装，用户可在 Web 门户中提交容器封装的训练作业，提交时可指定训练作业所需的 CPU、GPU 个数以及内存数量，支持从 web 门户打开该训练作业的屏幕输出、日志和 Tensorboard 界面以查看训练进展，并支持作业的插队、重新提交、暂停和终止等控制功能。

2.4.1. Tensorflow

2.4.1.1. 单机模式

点击“Tensorflow”图标，进入作业提交参数设置页面，如下图所示：

TensorFlow(tensorflow)

目前集群中可用于计算【tensorflow】作业的CPU核数为6

项目： default 作业名：

镜像 *： jhinno/tensorflow:py37-2.5 运行方式： 单机

工程目录 *： 本地 0个目录

挂载目录： 远端 0个目录

运行命令 *： python train.py --batch=64 --data-path="/data/minst_test"

环境变量： ENV1=test1,ENV2=test2

资源规格 *： 2核

Tensorboard： 关 密级： 公开

取消 确定

Tensorflow 单机模式作业提交页面

第二章

作业提交参数：

项目：指定项目，默认为： default。

作业名：指定作业名，默认为： tensorflow。

镜像：必选项，选择程序运行环境的镜像，需要根据运行程序选择合适的镜像，下拉列表中仅显示包含“tensorflow”关键字的镜像。

运行方式：默认选择单机，仅单节点上运行训练程序。

工程目录：指定代码的工程目录，可以从本地上传或选择服务端已存在的目录。

创建作业执行目录：默认不勾选，直接在工程目录中运行程序。手动勾选，会在“我的作业-作业数据区”中创建临时执行目录，将工程目录中的数据复制到临时目录后，进行运行程序操作。

挂载目录：指定挂载额外的目录到运行容器中，例如：可以选择挂载数据集。

挂载点：挂载目录在容器中对应的目录，未填写时挂载点和挂载目录路径保持一致。

运行命令：必填项，指定运行程序的入口文件和程序参数，例如： train.py --data-path=/data --batch=54。

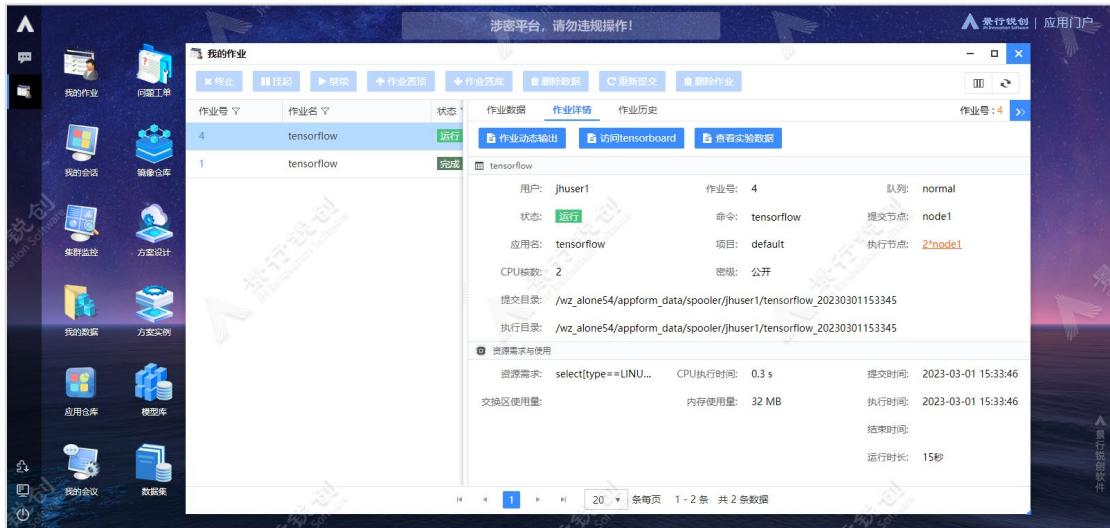
环境变量：指定程序运行需要的额外环境变量。

资源规格：默认为 2 核，选择运行程序所需要的资源配置。

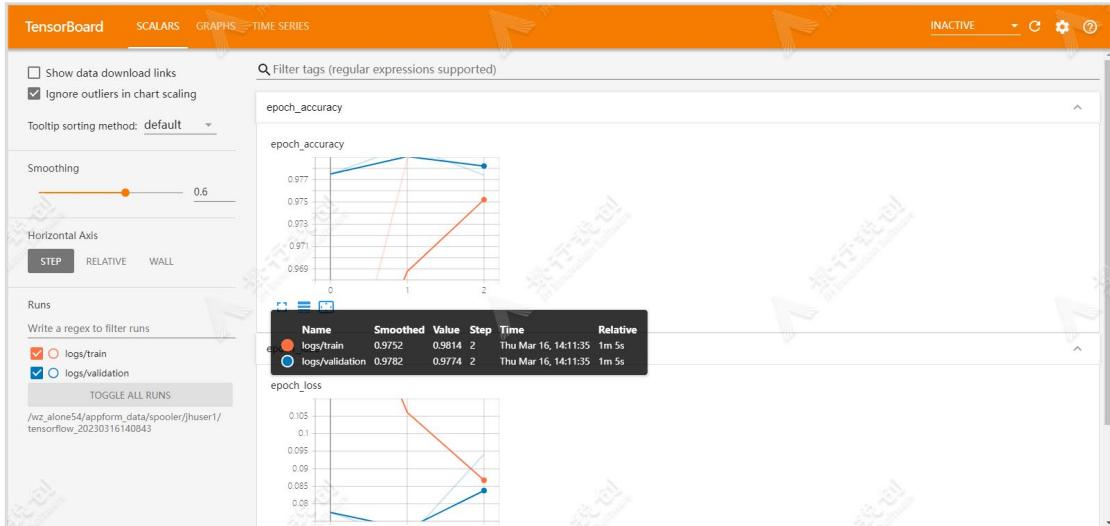
Tensorboard：选择是否允许访问 tensorboard 服务。若启动，作业详情页面，点击“访问 Tensorboard”按钮，即可查看模型训练情况。

密级：管理员开启密级功能后，显示此选项，默认为用户密级，用于数据安全、保密。

Tensorflow 作业提交后，会自动打开“我的作业”页面，方便用户在此查看提交作业的信息，如下图所示：



点击“访问 tensorboard”按钮，可以查看训练情况，如下图所示：



运行程序中集成了 libaiflow api，点击“查看实验数据”按钮，可以查看实验数据，如下图所示：

2.4.1.2. PS 并行模式

提交以 Parameter Server 方式实现的训练代码，支持跨节点，需填写并行相关参数，如下所示：

- **ParamServer 数：**指定参数服务的数量。
- **Worker 数：**指定训练服务数量。
- **资源规格：**分别指定每个 ParamServer 和 Worker 使用的资源。

TensorFlow(tensorflow)

目前集群中可用于计算【tensorflow】作业的CPU核数为6

项目： default 作业名：

镜像 *： jhinno/tensorflow:py37-2.5 运行方式： ps并行

工程目录 *： 本地 远端 0个目录 创建作业执行目录

挂载目录： 远端 0个目录 挂载点：

运行命令 *： python

环境变量：

ParamServer数 *： 资源规格 *： 2核

Worker数 *： 资源规格 *： 2核

Tensorboard： 关 密级： 公开

取消 确定

Tensorflow PS 并行模式作业提交页面

2.4.1.3. Horovod 并行模式

提交以 Horovod 实现的并行 AI 训练程序，需要指定 Horovod 相关参数。Horovod 并行模式需要模型训练代码以 Horovod 库的并行方式进行编码，如下所示：

- **并行实例数：**指定并行训练的节点数量。
- **资源规格：**指定每个并行节点使用的资源。



Tensorflow Horovod 并行模式作业提交页面

2.4.2. Pytorch

2.4.2.1. 单机模式

点击“Pytorch”图标，进入作业提交参数设置页面，如下图所示：

第二章



Pytorch(pytorch) - X

目前集群中可用于计算【pytorch】作业的CPU核数为5 历史表单

项目: 作业名:

镜像 *: 运行方式:

工程目录 *: 本地 远端 创建作业执行目录

挂载目录: 远端 挂载点:

运行命令 *:

环境变量:

资源规格 *:

Tensorboard: 密级:

Pytorch 单机模式作业提交页面

作业提交参数:

项目: 指定项目, 默认为: default。

作业名: 指定作业名, 默认为: pytorch。

镜像: 必选项, 选择程序运行环境的镜像, 需要根据运行程序选择合适的镜像, 下拉列表中仅显示包含“pytorch”关键字的镜像。

运行方式: 默认选择单机, 仅单节点上运行训练程序。

工程目录: 指定代码的工程目录, 可以从本地上传或选择服务端已存在的目录。

创建作业执行目录: 默认不勾选, 直接在工程目录中运行程序; 手动勾选, 会在“我的作业-作业数据区”中创建临时执行目录, 将工程目录中的数据复制至其中后, 进行运行程序操作。

挂载目录: 指定挂载额外的目录到运行容器中, 例如: 可以选择挂载数据集。

挂载点: 挂载目录在容器中对应的目录, 未填写时挂载点和挂载目录路径保持一致。

运行命令: 必填项, 指定运行程序的入口文件和程序参数, 例如: train.py --data-path=/data --batch=54。

环境变量: 指定程序运行需要的额外环境变量。

资源规格：默认为 2 核，选择运行程序所需要的资源配置。

Tensorboard：选择是否启动 tensorboard 服务。若启动，作业详情页面，点击“访问 Tensorboard”按钮，即可查看模型训练情况。

密级：管理员开启密级功能后，显示此选项，默认为用户密级，用于数据安全、保密。

2.4.2.2. Horovod 并行模式

提交以 Horovod 实现的并行 AI 训练程序，需要指定 Horovod 相关参数。

Horovod 并行模式需要模型训练代码以 Horovod 库的并行方式进行编码，如下所示：

- **并行节点数：**指定并行训练的节点数量。
- **资源规格：**指定每个并行节点使用的资源。

Pytorch(ptorch)

目前集群中可用于计算【pytorch】作业的CPU核数为5

项目： default 作业名：

镜像 *： jhinno/pytorch:py37-1.10.0 运行方式： Horovod并行

工程目录 *： 本地 远端 0个目录 创建作业执行目录

挂载目录： 远端 0个目录 挂载点：

运行命令 *： python train.py --batch=64 --data-path="/data/minst_test"

环境变量： ENV1=test1,ENV2=test2

并行实例数 *： 2核

Tensorboard： 关 密级： 公开

取消 确定

Pytorch Horovod 并行模式作业提交页面

2.4.3. Mxnet

Mxnet 是开源深度学习框架，允许用户在多种设备上定义、训练和部署深度神经网络。该框架具备高度可扩展性，可以进行快速的模型训练，并支持灵活的编程模型。

2.4.3.1. 单机模式

点击“Mxnet”图标，进入作业提交参数设置页面，如下图所示：



Mxnet 单机模式作业提交页面

作业提交参数：

项目：指定项目，默认为：default。

作业名：指定作业名，默认为：mxnet。

镜像：必选项，选择程序运行环境的镜像，需要根据运行程序选择合适的镜像，下拉列表中仅显示包含“mxnet”关键字的镜像。

运行方式：默认选择单机，仅单节点上运行训练程序。

工程目录：指定代码的工程目录，可以从本地上传或选择服务端已存在的目

录。

创建作业执行目录: 默认不勾选, 直接在工程目录中运行程序; 手动勾选, 会在“我的作业-作业数据区”中创建临时执行目录, 将工程目录中的数据复制至其中后, 进行运行程序操作。

挂载目录: 指定挂载额外的目录到运行容器中, 例如: 可以选择挂载数据集。

挂载点: 挂载目录在容器中对应的目录, 未填写时挂载点和挂载目录路径保持一致。

运行命令: 必填项, 指定运行程序的入口文件和程序参数, 例如: `train.py --data-path=/data --batch=54`。

环境变量: 指定程序运行需要的额外环境变量。

资源规格: 默认为 2 核, 选择运行程序所需要的资源配置。

密级: 管理员开启密级功能后, 显示此选项, 默认为用户密级, 用于数据安全、保密。

2. 4. 3. 2. Horovod 并行模式

提交以 Horovod 实现的并行 AI 训练程序, 需要指定 Horovod 相关参数。Horovod 并行模式需要模型训练代码以 Horovod 库的并行方式进行编码, 如下所示:

- **并行节点数:** 指定并行训练的节点数量。
- **资源规格:** 指定每个并行节点使用的资源。

第二章

Mxnet(mxnet)

目前集群中可用于计算【mxnet】作业的CPU核数为5

项目： default 作业名：

镜像 *： jhinno/mxnet:py37-1.9.1 运行方式： Horovod并行

工程目录 *： 本地 远端 0个目录

挂载目录： 远端 0个目录 挂载点：

运行命令 *： python train.py --batch=64 --data-path="/data/minst_test"

环境变量： ENV1=test1,ENV2=test2

并行实例数 *： 资源规格 *： 2核

密级： 公开

取消 确定

Mxnet Horovod 并行模式作业提交页面

2.4.4. PaddlePaddle

飞桨(PaddlePaddle)是集深度学习核心框架、工具组件和服务平台为一体的
技术先进、功能完备的开源深度学习平台。

2.4.4.1. 单机单卡

点击“PaddlePaddle”图标，进入作业提交参数设置页面，如下图所示：



PaddlePaddle 单机模式作业提交页面

作业提交参数：

项目：指定项目，默认为：default。

作业名：指定作业名，默认为：paddle。

镜像：必选项，选择程序运行环境的镜像，需要根据运行程序选择合适的镜像，下拉列表中仅显示包含“paddle”关键字的镜像。

运行方式：默认选择单机，仅单节点上运行训练程序。

工程目录：指定代码的工程目录，可以从本地上传或选择服务端已存在的目录。

创建作业执行目录：默认不勾选，直接在工程目录中运行程序；手动勾选，会在“我的作业-作业数据区”中创建临时执行目录，将工程目录中的数据复制到临时目录后，进行运行程序操作。

挂载目录：指定挂载额外的目录到运行容器中，例如：可以选择挂载数据集。

挂载点：挂载目录在容器中对应的目录，未填写时挂载点和挂载目录路径保持一致。

运行命令：必填项，指定运行程序的入口文件和程序参数，例如：train.py --data-path=/data --batch=54。

环境变量：指定程序运行需要的额外环境变量。

第二章

资源规格：默认为 2 核，选择运行程序所需要的资源配置。

密级：管理员开启密级功能后，显示此选项，默认为用户密级，用于数据安全、保密。

2.4.4.2. 单机多卡

提交单机多卡 PaddlePaddle 作业，需要在运行命令处额外配置“-m paddle.distributed.launch”参数，具体操作详情如下图所示：



PaddlePaddle 单机多卡模式作业提交页面

2.5. 实验管理

实验管理为人工智能软件平台提供跟踪、比较和记录模型训练指标的工具和平台，是保存每个运行实验所关注的所有实验相关信息的过程。实验训练过程中的参数，超参数，指标，数字，图像，文件等信息会记录到数据库中，通过可视化界面供用户查看。

涉及的一些关键字：

实验 (experiment)：相当于项目名，所有的调参都是针对这个实验。

运行 (Run)：每次调参的运行记录。

参数 (parameter)：每次调参所记录的参数。

指标 (metric)：记录每次调参之后的指标结果。

文件 (artifact)：记录文件。

2.5.1. 准备训练程序

准备实验训练程序，并在实验训练程序中集成 libaiflow API。

方式一：通过 libaiflow.tensorflow.autolog 和 libaiflow.pytorch.autolog 提供的 `Callback` 机制支持快速简易的收集 tensorflow 和 pytorch 实验的参数，指标等多种实验训练数据。下面仅展示 autolog 自动收集数据的关键代码，（完整样例代码参考附录七样例一）。

```
import libaiflow
expt = libaiflow.init(experiment_name="mnist_tf2_cpu")
libaiflow.tensorflow.autolog()
```

方式二：通过指标和超参数相关 api 记录和收集数据，（完整样例代码参考附录七样例二）。

```
import libaiflow
expt = libaiflow.init(experiment_name="pytorch-sample")
.....
with libaiflow.start_run(experiment_id=expt.experiment_id):
    lr = ElasticNet(alpha=alpha, l1_ratio=l1_ratio, random_state=42)
    lr.fit(train_x, train_y)
    predicted_qualities = lr.predict(test_x)
    (rmse, mae, r2) = eval_metrics(test_y, predicted_qualities)
    print("Elasticnet model (alpha=%f, l1_ratio=%f):" % (alpha,
```

第二章

```
11_ratio))

    print(" RMSE: %s" % rmse)
    print(" MAE: %s" % mae)
    print(" R2: %s" % r2)

    libaiflow.log_param("alpha", alpha)
    libaiflow.log_param("11_ratio", 11_ratio)
    libaiflow.log_metric("rmse", rmse)
    libaiflow.log_metric("r2", r2)
    libaiflow.log_metric("mae", mae)
```

2.5.2. 执行训练程序

人工智能平台的多个功能已集成 libaiflow 库，如：提交 AI 作业、运行模型结构的方案、Jupyter 和 VSCode 等。

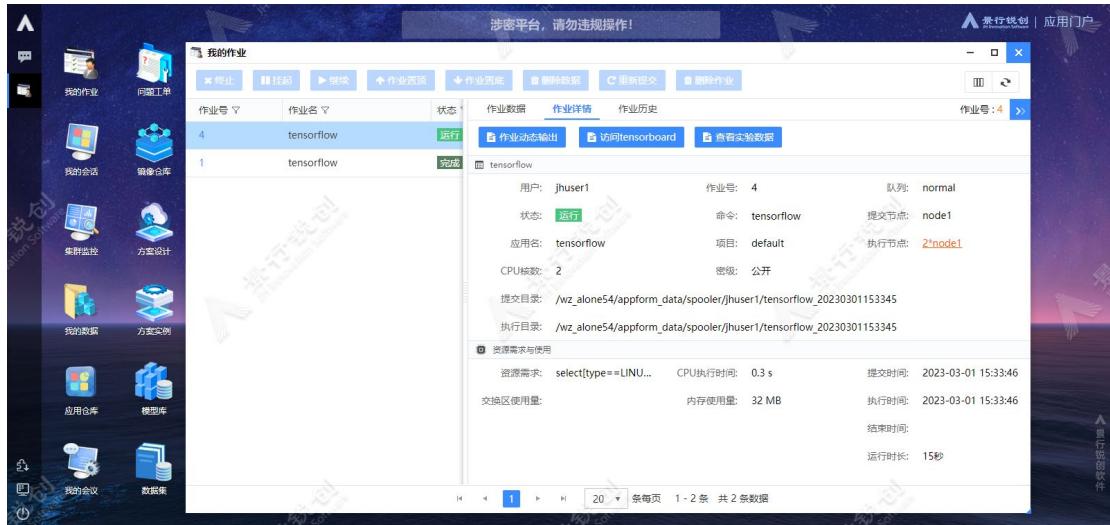
2.5.2.1. 提交 AI 作业

AI 作业默认集成 libaiflow 库，应用集成 libaiflow API 的程序，提交 AI 作业，进行模型训练。以提交 Tensorflow 作业为例，如下图所示：



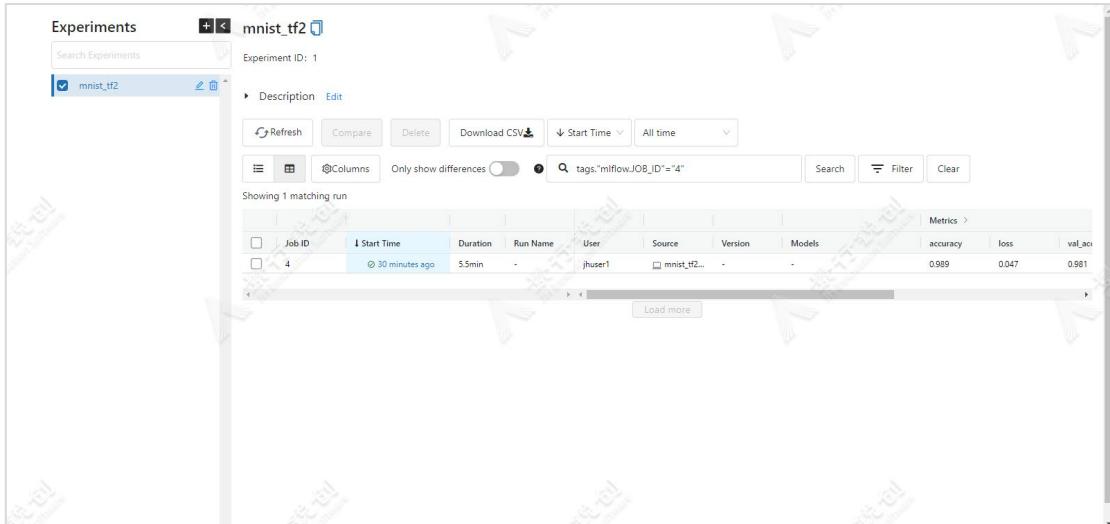
提交 AI 作业示意图

Tensorflow 作业提交后，会自动打开“我的作业”页面，方便用户在此查看提交作业的信息，如下图所示：



注意：选择集成 libaiflow api 的训练程序，才能查看实验数据。

点击“查看实验数据”按钮，可以查看实验数据，如下图所示：

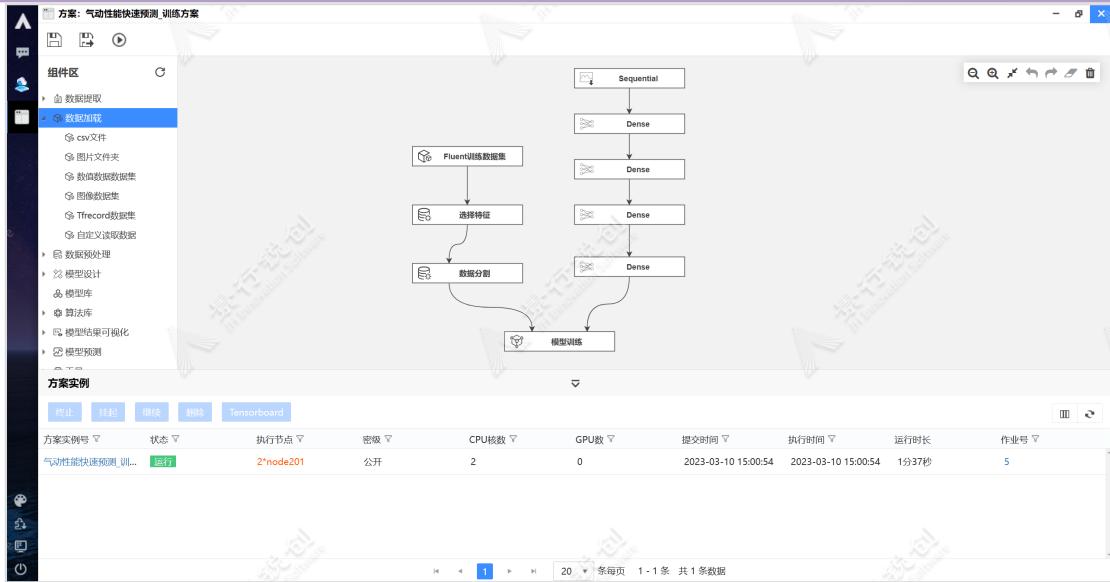


查看实验数据示意图

2.5.2.2. 运行模型结构的方案

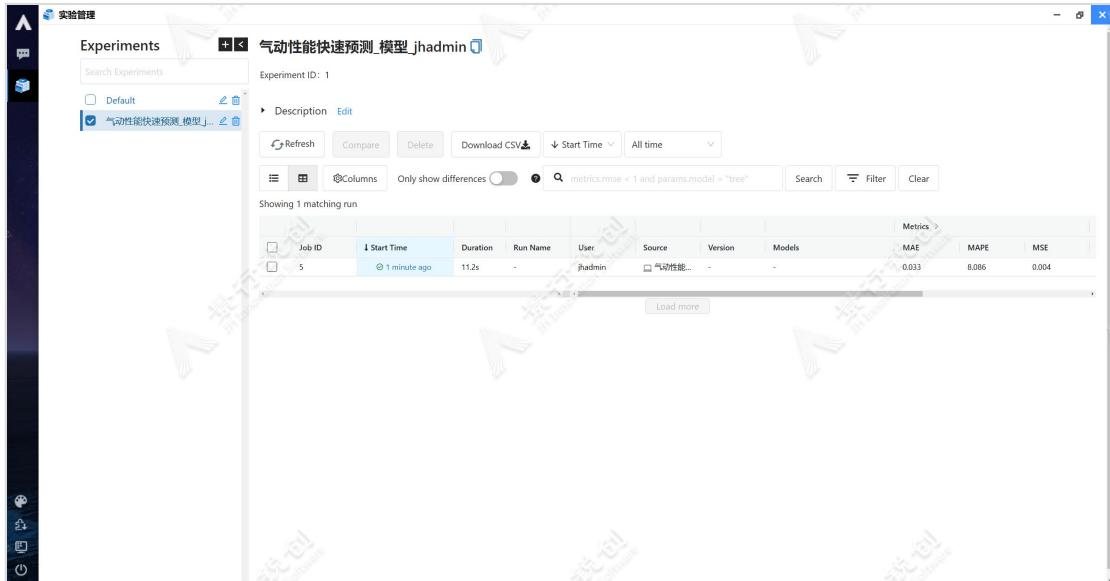
打开方案设计管理，进入方案设计器界面，构建模型结构流程、训练模型，如下图所示：

第二章



运行方案示意图

在门户桌面，双击“实验管理”桌面图标，可以查看实验数据，如下图所示：



查看实验数据

2.5.2.3. 从开发中心运行

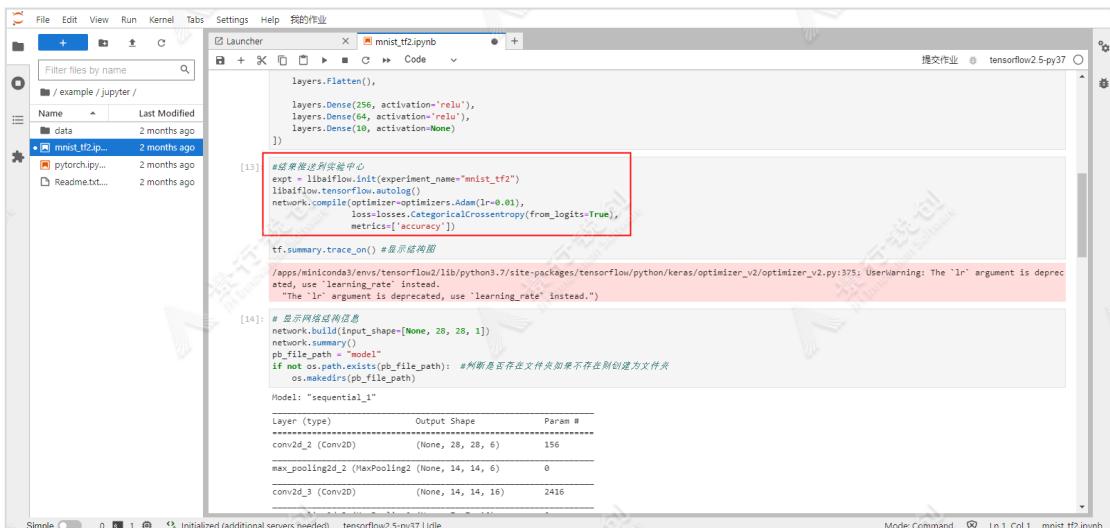
WebIDE 类型的开发环境默认集成 libaiflow 库，访问“Jupyter”和“VSCode”服务训练程序或者提交 AI 作业，进行模型训练。以通过 Jupyter 服务应用 libaiflow API 为例，操作步骤如下：双击“开发中心”桌面图标，打开“开发

“中心管理”界面，如下图所示：



开发中心管理示意图

点击“Jupyter”卡片按钮，访问 Jupyter 服务，选择集成 libaiflow API 的训练程序，执行训练程序，或通过提交作业方式训练程序。如下图所示：



访问“Jupyter”服务示意图

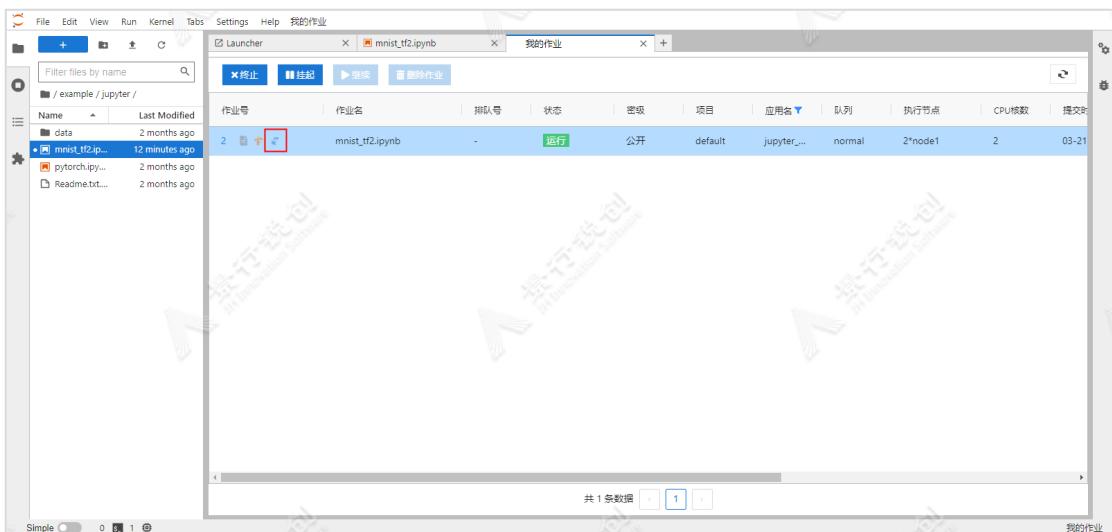
点击“提交作业”按钮，提交作业，如下图所示：

第二章



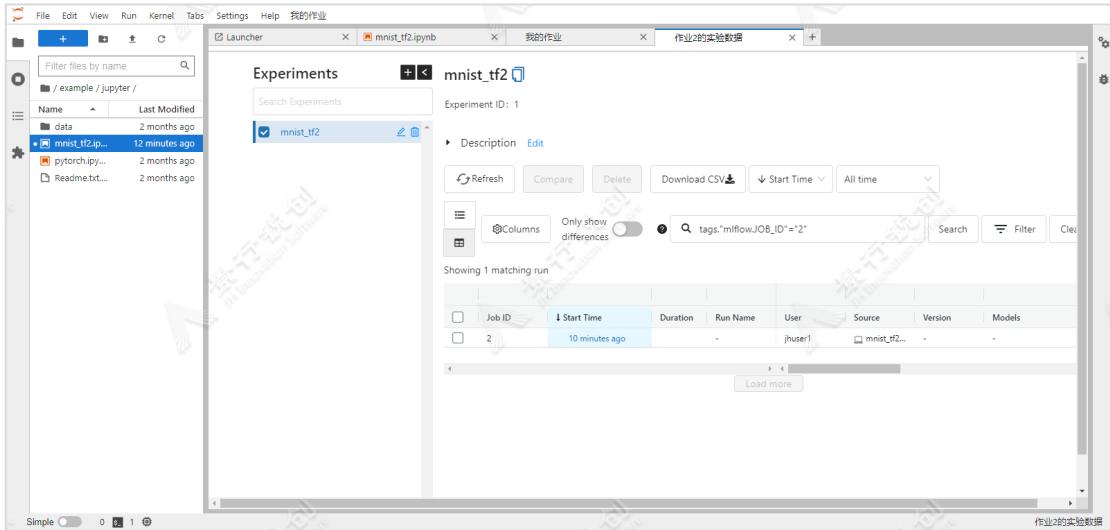
提交 AI 作业示意图

点击“我的作业”按钮，打开我的作业 tab 页，如下图所示：



我的作业示意图

点击作业实例上的“实验管理”图标按钮，查看实验数据，如下图所示：



查看实验数据示意图

2.6. 服务管理

服务管理支持“容器终端”、“Tensorboard”、“Tensorflow Serving”和“Torch Serving”类型的服务。如下所示：

- **容器终端**：仅启动一个 web ssh 服务。
- **Tensorboard**：用于可视化查看模型数据。
- **Tensorflow Serving**：将训练的 Tensorflow 模型发布为 Web 服务。
- **Torch Serving**：将训练的 Pytorch 模型发布为 Web 服务。

各个角色对服务实例的操作权限：

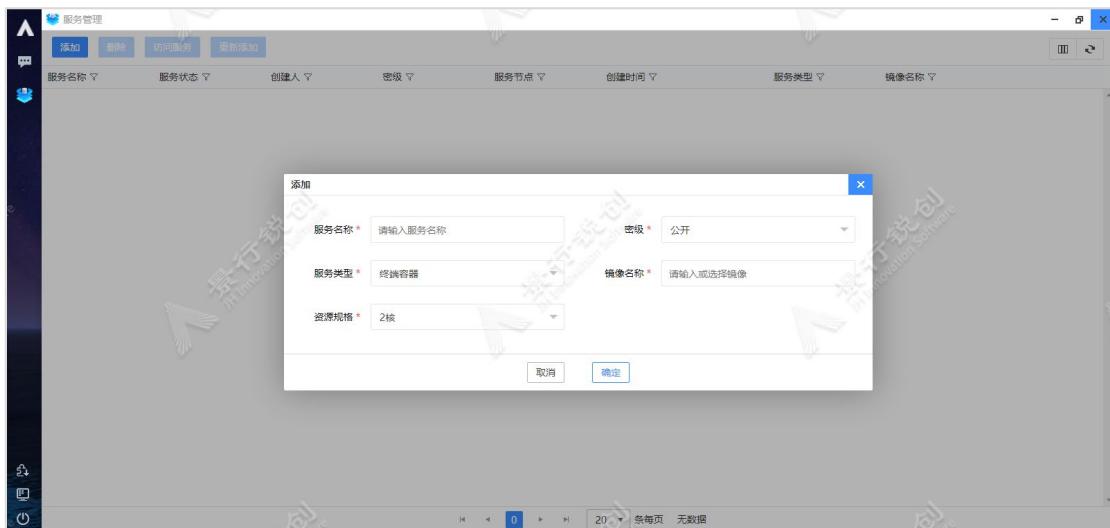
添加者：对服务实例具有添加、重新添加、访问服务以及删除服务的权限；

管理员：对其他人添加的服务实例仅有查看服务实例信息和删除服务的权限。

2.6.1. 添加服务

2.6.1.1. “终端容器”类型的服务

点击“添加”按钮，弹出“添加”窗口，选择启动的服务类型和填写相关参数，如下图所示：



添加“终端容器”类型的服务配置示意图

图中每个参数的具体含义如下：

服务名称：用户自定义。

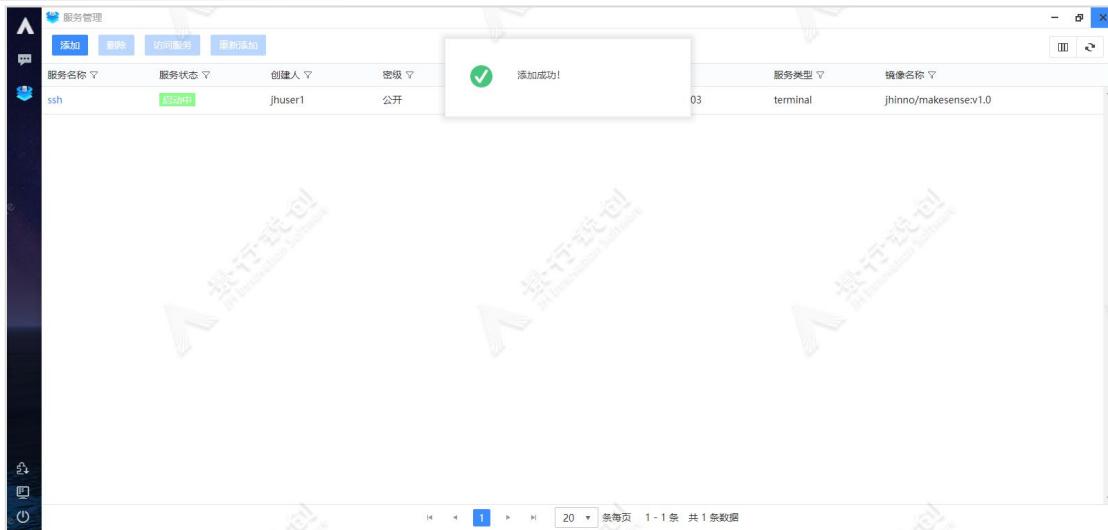
密级：管理员开启密级功能后，显示此选项，默认为用户密级，用于数据安全、保密。

服务类型：选择启动的服务类型，默认为“终端容器”。

镜像名称：通过下拉列表选择需要的镜像。

资源规格：选择启动服务所需要的资源配置，默认为 2 核。

填写完成后，点击“确定”按钮，添加服务实例，如下图所示：



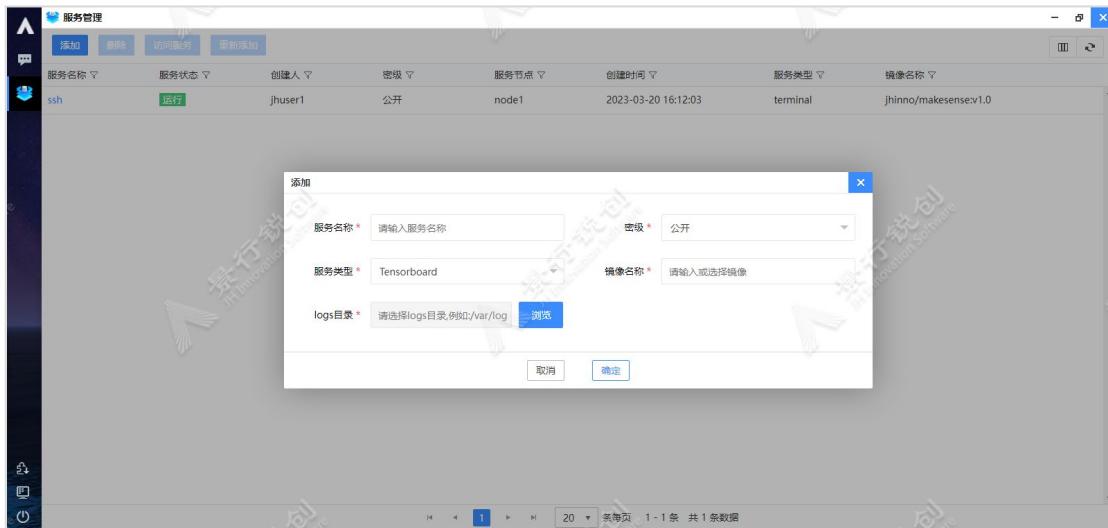
添加“终端容器”类型的服务示意图

2.6.1.2. “Tensorboard”类型的服务

添加“Tensorboard”类型的服务有2种方式，如下所示：

- 当服务管理中没有“Tensorboard”类型的服务时，在我的作业、方案设计和模型库中，可以通过点击“访问Tensorboard”按钮，添加“Tensorboard”类型的服务。
- 进入服务管理，手动添加“Tensorboard”类型的服务。

以手动添加“Tensorboard”类型的服务为例，操作步骤如下：点击“添加”按钮，弹出“添加”窗口，选择启动的服务类型和填写相关参数，如下图所示：



第二章

添加“Tensorboard”类型的服务配置示意图

图中每个参数的具体含义如下：

服务名称：用户自定义。

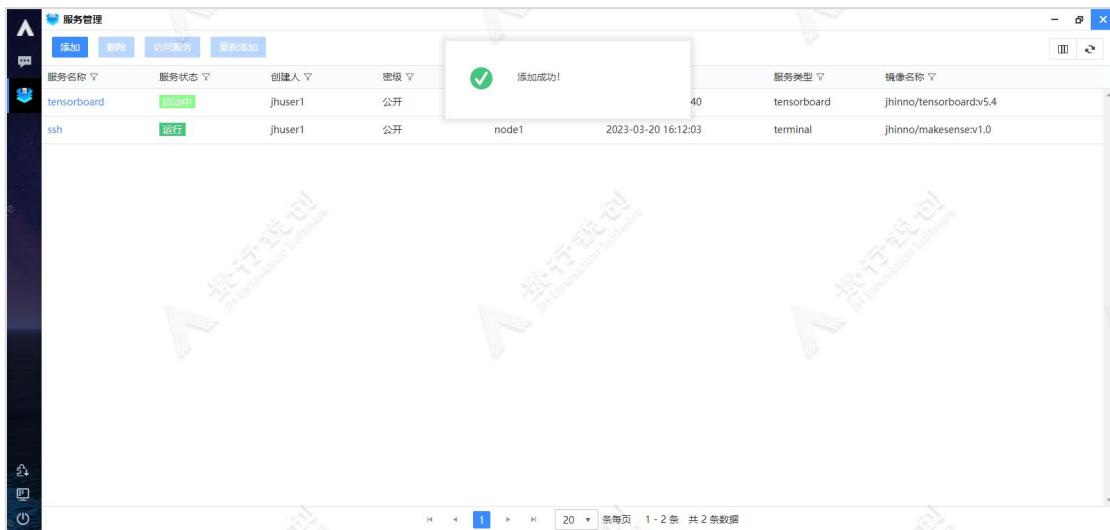
密级：管理员开启密级功能后，显示此选项，默认为用户密级，用于数据安全、保密。

服务类型：选择“Tensorboard”。

镜像名称：通过下拉列表选择镜像。

Logs 地址：单击浏览按钮选择路径。

填写完成后，点击“确定”按钮，添加服务实例，如下图所示：



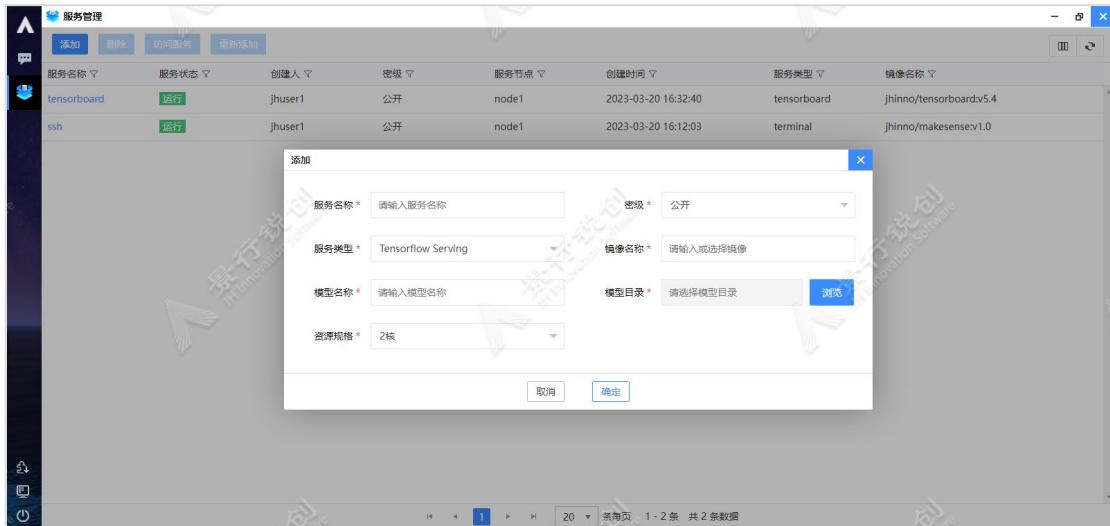
添加“Tensorboard”类型的服务示意图

2.6.1.3. “Tensorflow Serving”类型的服务

添加“Tensorflow Serving”类型的服务有2种方式，如下所示：

- 进入模型库，点击模型卡片，进入模型实例列表，选中一个模型实例后，点击“模型部署”按钮，添加“Tensorflow Serving”类型的服务。
- 进入服务管理，手动添加“Tensorflow Serving”类型的服务。

以手动添加“Tensorflow Serving”类型的服务为例，操作步骤如下：点击“添加”按钮，弹出“添加”窗口，选择启动的服务类型和填写相关参数，如下图所示：



添加“Tensorflow Serving”类型的服务配置示意图

图中每个参数的具体含义如下：

服务名称：用户自定义。

密级：管理员开启密级功能后，显示此选项，默认为用户密级，用于数据安全、保密。

服务类型：选择“Tensorflow Serving”。

镜像名称：通过下拉列表选择镜像。

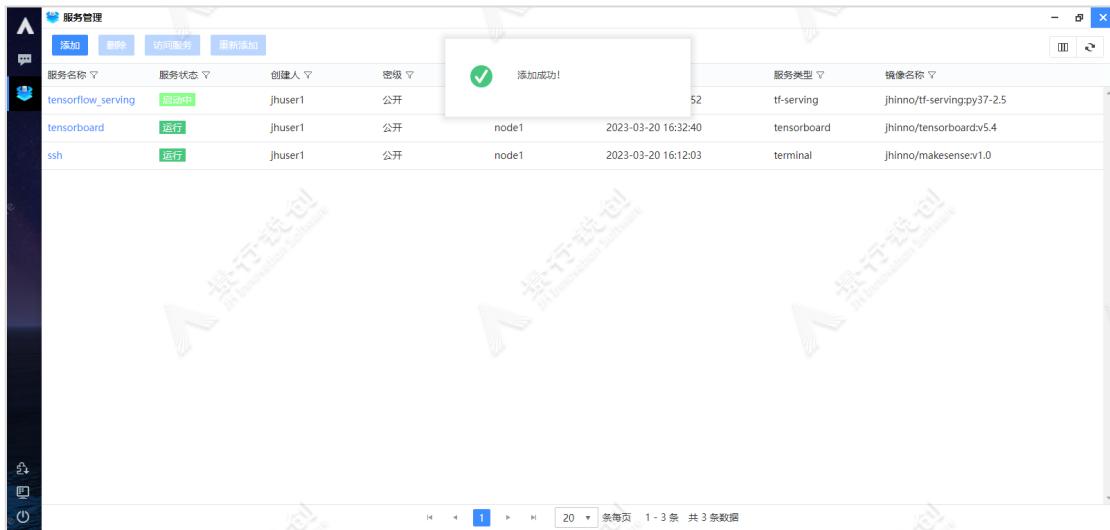
模型名称：用户自定义。

模型目录：选择模型结果所在的目录。

资源规格：选择启动服务所需要的资源配置，默认为2核。

填写完成后，点击“确定”按钮，添加服务实例，如下图所示：

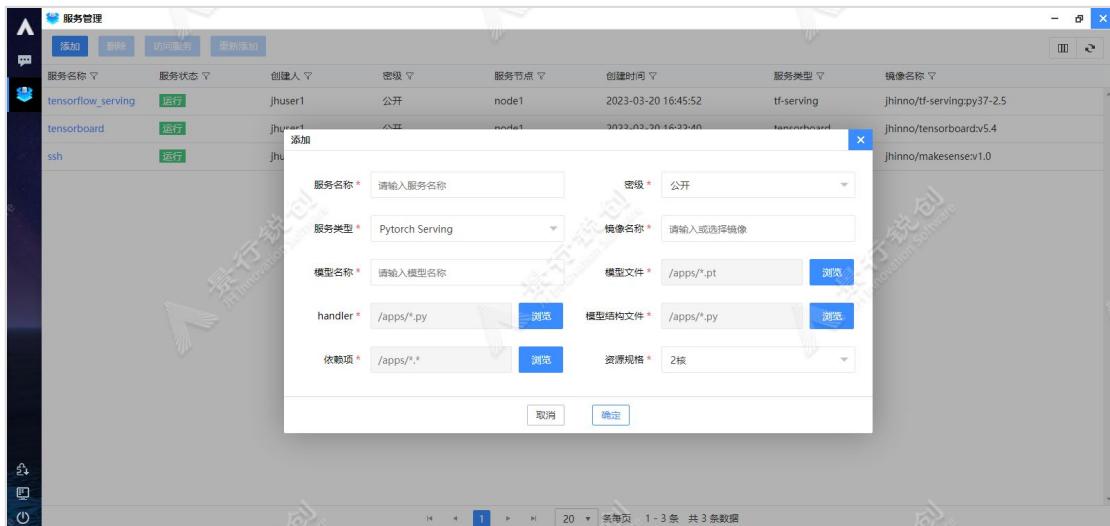
第二章



添加“Tensorflow Serving”类型的服务示意图

2.6.1.4. “Pytorch Serving”类型的服务

点击“添加”按钮，弹出“添加”窗口，选择启动的服务类型和填写相关参数，如下图所示：



添加“Pytorch Serving”类型的服务配置示意图

图中每个参数的具体含义如下：

服务名称：用户自定义。

密级：管理员开启密级功能后，显示此选项，默认为用户密级，用于数据安全、保密。

服务类型: 选择“Pytorch Serving”。

镜像名称: 通过下拉列表选择镜像。

模型名称: 用户自定义。

模型文件: 选择*.pt 模型文件。

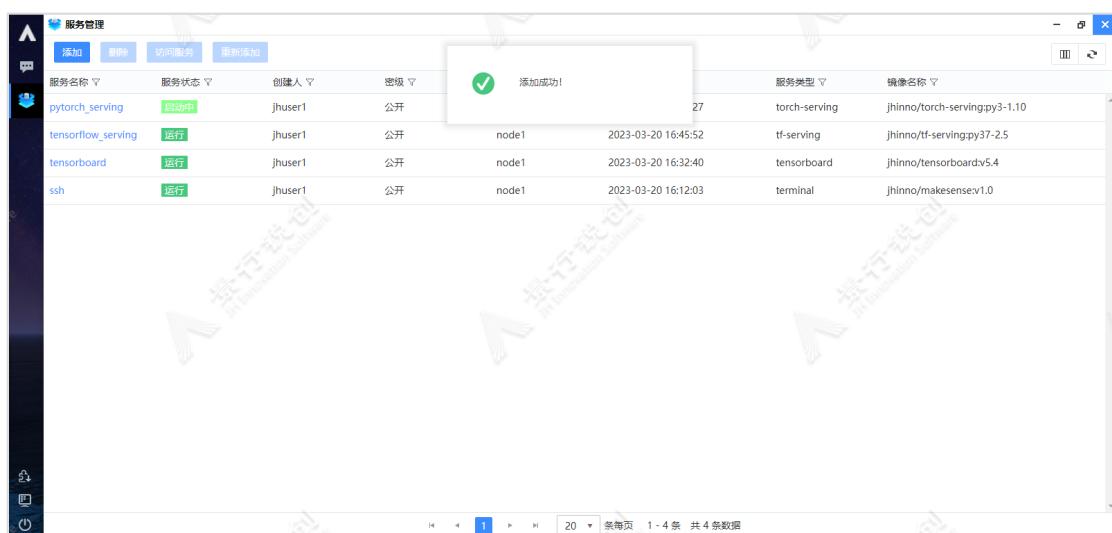
handler: 选择*.py 文件。

模型结构文件: 选择*.py 模型结构文件。

依赖项: 选择依赖项。

资源规格: 选择启动服务所需要的资源配置，默认为 2 核。

填写完成后，点击“确定”按钮，添加服务实例，如下图所示：

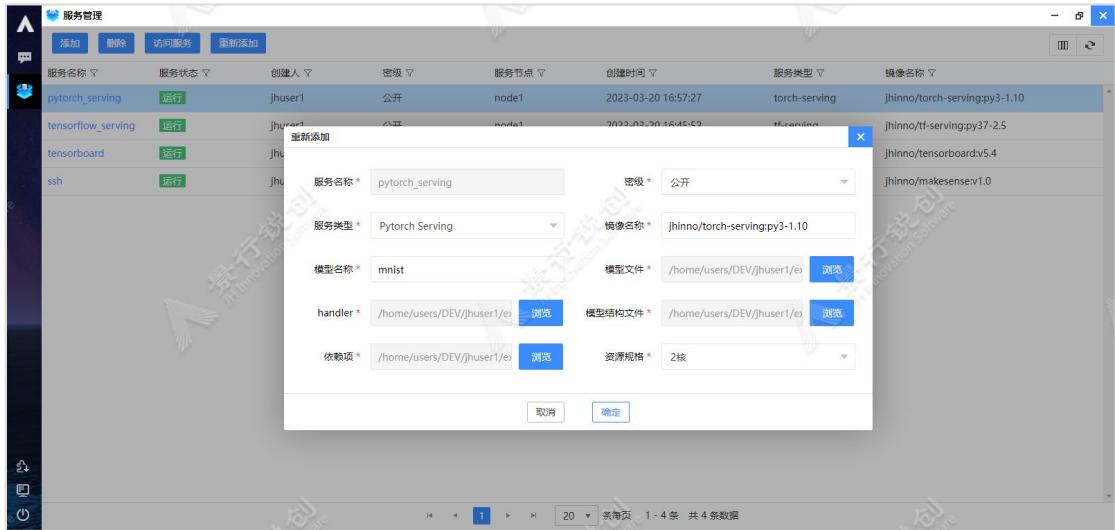


添加“Pytorch Serving”类型的服务示意图

2.6.2. 重新添加服务

以重新添加“Tensorflow Serving”类型的服务为例，操作步骤如下：选中一条服务实例后，点击“重新添加”按钮，弹出“重新添加”窗口，如下图所示：

第二章



重新添加“Pytorch Serving”类型的服务配置示意图

图中每个参数的具体含义如下：

服务名称：服务名称只读，默认回填该服务实例的服务名称。

密级：管理员开启密级功能后，显示此选项，默认为该服务实例的密级。

服务类型：Pytorch Serving。

镜像名称：通过下拉列表选择镜像。

模型名称：用户自定义。

模型文件：选择*.pt 模型文件。

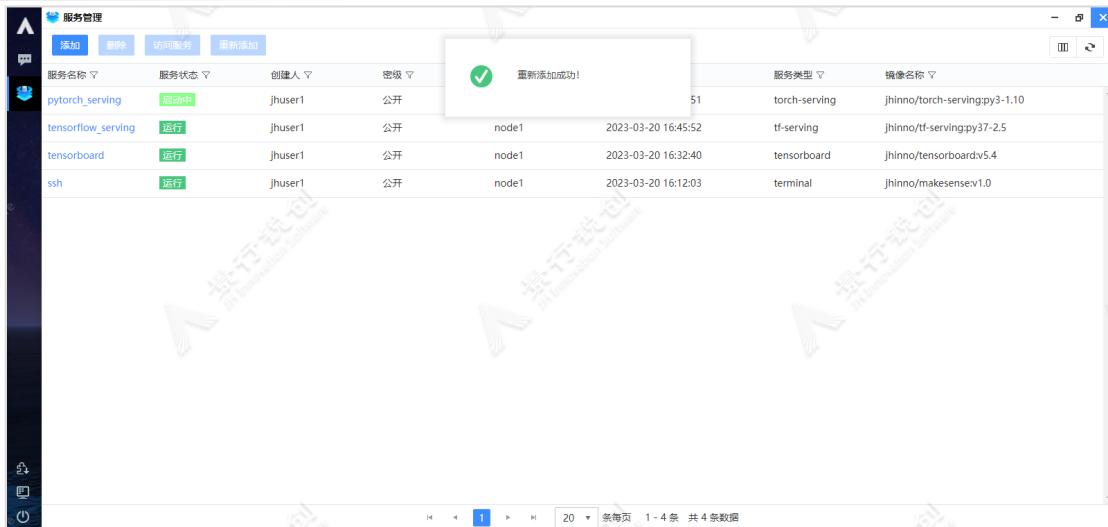
handler：选择*.py 文件。

模型结构文件：选择*.py 模型结构文件。

依赖项：选择依赖项。

资源规格：选择启动服务所需要的资源配置，默认为 2 核。

填写完成后，点击“确定”按钮，重新添加服务实例，如下图所示：

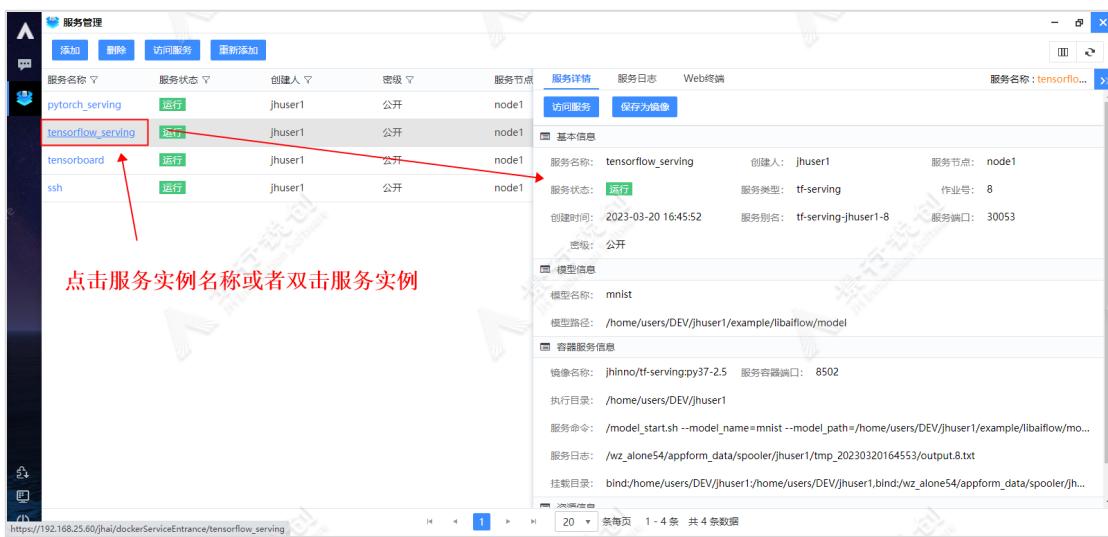


重新添加“Pytorch Serving”类型的服务示意图

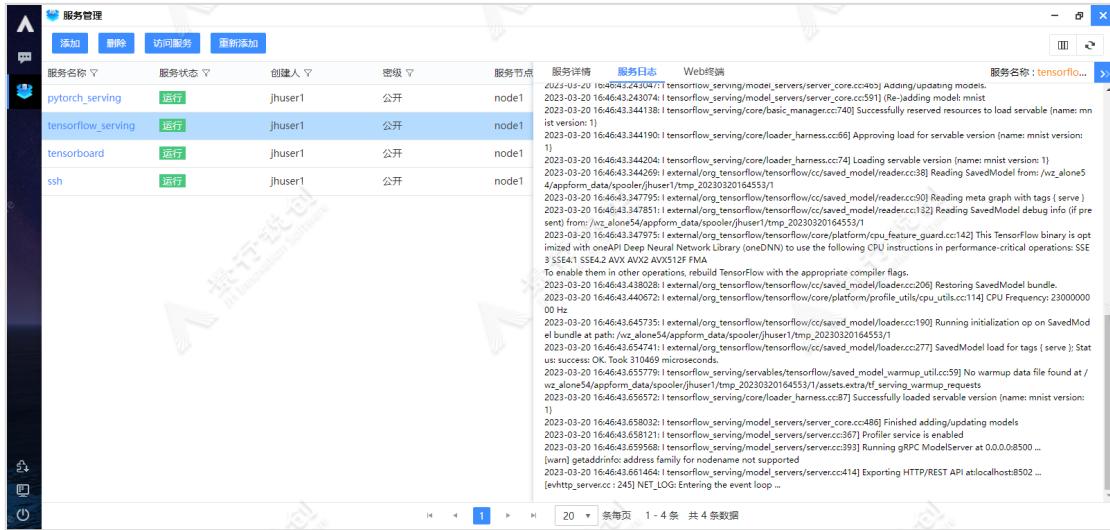
2. 6. 3. 查看服务信息

点击服务列表中的某一条服务实例的服务名称或者双击某一条服务实例，会打开右侧滑块。滑块中有三个标签页：服务详情、服务日志以及 Web 终端。

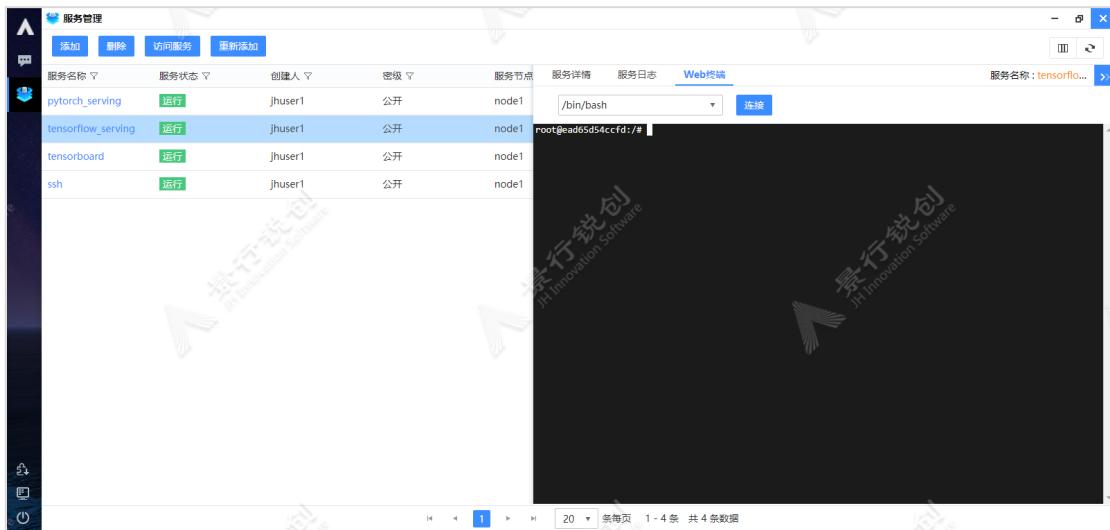
服务详情：显示服务基本信息、模型信息、容器服务信息和资源信息，如下图所示：



服务日志：点击“服务日志”按钮，切换至服务日志页面，查看服务日志。当该服务的状态处于“启动中”和“运行”状态时，会动态输出该服务的日志，否则会静态输出所有日志，具体详情如下图所示：



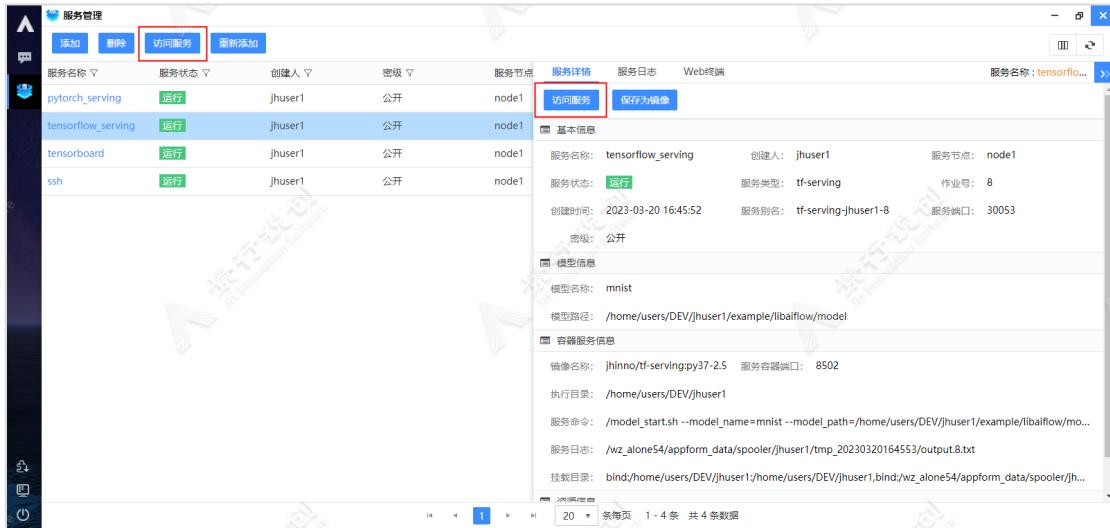
Web 终端：点击“web 终端”按钮，切换至 Web 终端页面，可以在页面模拟 ssh 终端访问操作容器内系统，如下图所示：



2.6.4. 应用服务

2.6.4.1. 访问服务

点击选中服务列表中的某一条服务实例后，点击“访问服务”按钮，会在一个新的窗口打开服务地址，如下图所示：



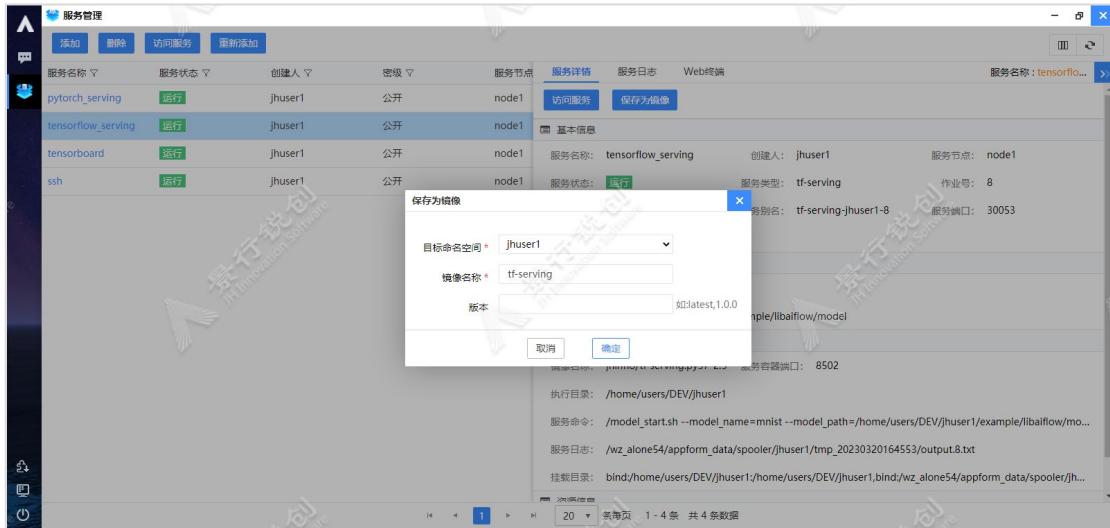
注意：访问服务功能仅在服务实例的状态为“运行”状态时可用。



2.6.4.2. 保存为镜像

点击服务列表中的某一条服务实例的服务名称或者双击某一条服务实例，会打开右侧滑块。点击“保存为镜像”按钮，弹出“保存为镜像”窗口，如下图所示：

第二章



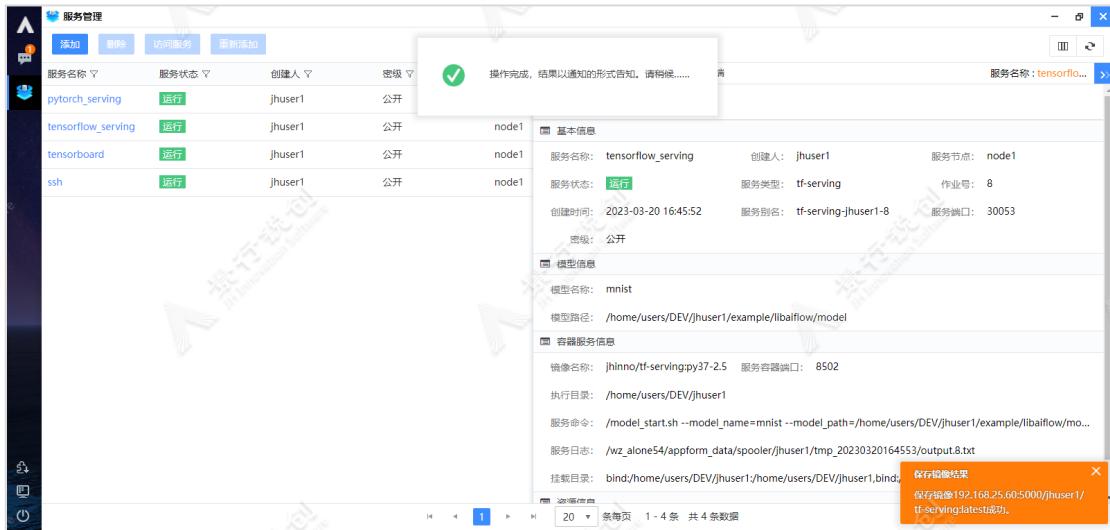
图中每个参数的具体含义如下：

目标命名空间：选择镜像的保存位置。

镜像名称：输入保存后的镜像名称。

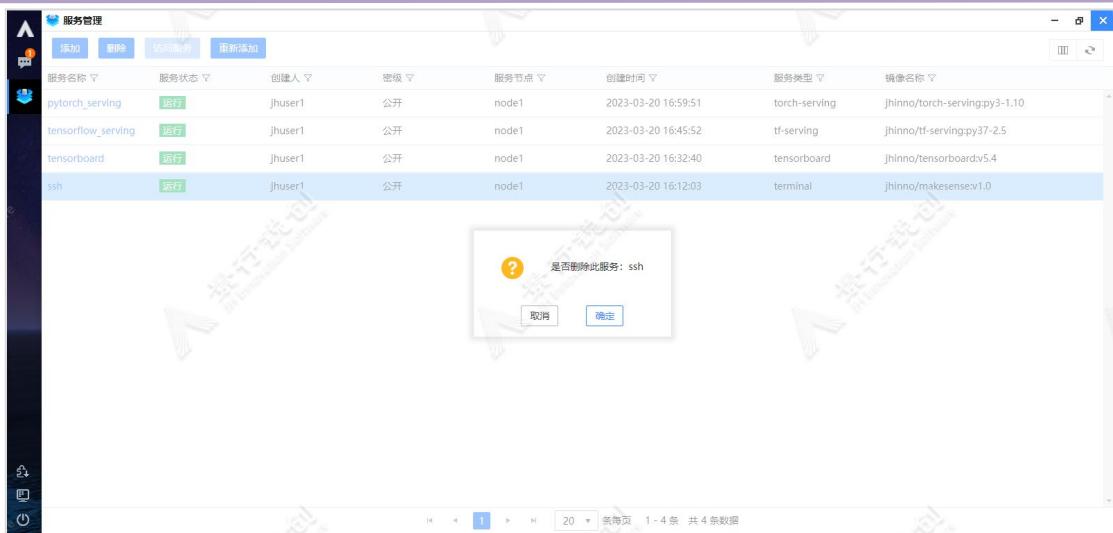
版本：输入镜像版本，默认为 latest。

点击“确定”按钮，将服务保存为镜像，如下图所示：

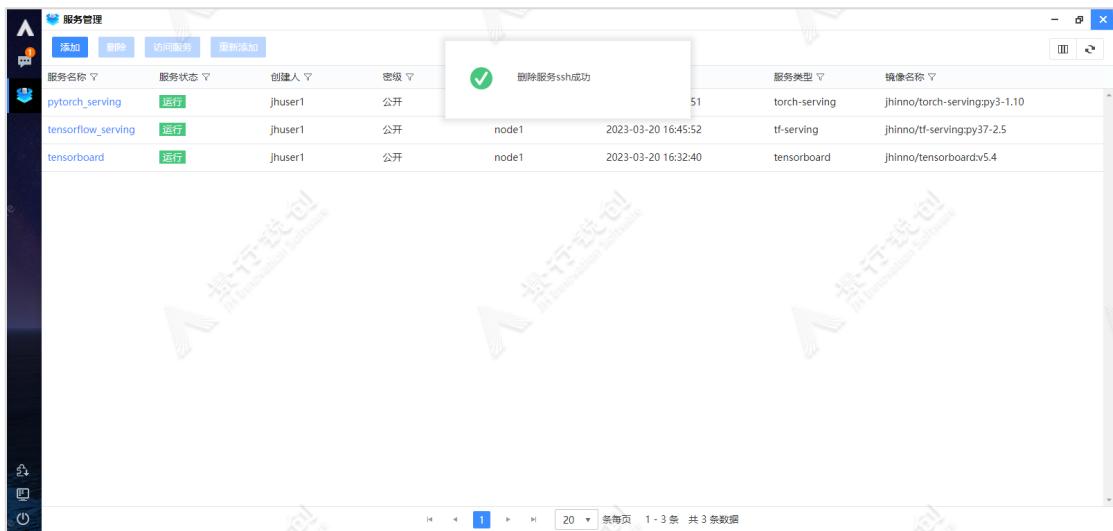


2.6.5. 删除服务

选中服务列表中某一条服务实例后，点击“删除”按钮，弹出“是否删除此服务”提示窗口，如下图所示：



点击“确定”按钮，删除该条服务实例，如下图所示：



附录：常见问题及解决办法

1. 应用 3D 可视化功能，报：“浏览器未开启 WebGL 功能，请检查浏览器设置”

人工智能平台支持对卷积神经网络模型进行 3D 可视化，但是 3D 可视化功能依赖浏览器的 WebGL 功能。如果使用 Google Chrome84 以上版本，却无法使用 3D 可视化功能，需要在 Google Chrome 中打开 WebGL2 功能。

解决方案：

首先，测试下浏览器是否支持 WebGL2。在线检测，请在浏览器地址栏中输入以下网址：

```
https://webglreport.com/?v=2
```

当出现下列字样时，说明你的浏览器支持 WebGL2，如下所示：

✓ This browser supports WebGL 2

启用 WebGL，在 Google Chrome 中设置 WebGL2 相关参数，浏览器输入

```
chrome://flags
```

搜索参数并设置

WebGL Draft Extensions 设置为 Enable

Choose ANGLE graphics backend 设置为 OpenGL

2. 如何在容器服务中使用用户自己的数据

解决方案：

容器服务启动时会默认挂载用户家目录、作业数据区、共享数据区，可

以在启动的容器服务中直接访问用户家目录、作业数据区、共享数据区下的数据文件，目录绝对路径和宿主机保持一致。

3. 如何保存在容器服务中创建的文件

解决方案：

容器服务启动时会默认挂载用户家目录、作业数据区和共享数据区，可以将需要保存的文件拷贝到用户家目录、作业数据区或者共享数据区下。

4. 如何在离线内网环境安装额外的 pip 或 conda 依赖

解决方案：

可以从外网下载 pip、conda 的离线包，导入内网安装。

5. 如何在非 root 启动的镜像中安装依赖包

解决方案：

使用 sudo 执行 yum 或 apt-get 命令。

6. 模型训练任务异常退出，显示 out of memory

解决方案：

调整训练程序代码或数据大小。

7. 容器服务启动失败报错，output.txt 中提示：“service start failed, current status is: rejected, reason is: No such image: ahaha/aaa:v_testal”

原因：

创建容器服务时，选择/输入的镜像名是不存在的，或者是浏览器页面缓存的镜像名实际上已经不存在。

解决方案：

清除浏览器缓存，输入/选择真正存在的镜像。

8. pytorch、tensorflow 等 AI 页面加载失败，没有正常显示

原因：

appform session 过期，在 portal.log 和 jhai.log 中可以看到过期信息。

解决方案：

重新登录门户。

9. 挂起 TensorFlow、Pytorch 等 AI 作业。作业状态显示挂起，但后台进程仍然运行

原因：

作业容器启动前无法被挂起。

10. 计算节点无法访问应用门户节点外网 IP 时，在 jupyter 容器中，无法使用模型部署的容器服务做推理。

原因：

jupyter 容器需要能访问应用门户节点外网 IP。

解决办法：

允许计算节点访问应用门户节点外网 IP 或在 jupyter 中使用门户内网 IP 访问模型部署的服务，自行调整容器服务 url 来使用。内网 IP 的获取需要联系管理员。

11. 数值数据集列名中包含符号. 的数据集，预览无法显示数据。

原因：

不支持列名中包含符号. 的数据集。

解决办法：

不要在列名中包含符号。

12. 容器桌面中，执行 mount，报错：“mount: /ubuntuxwf/: mount failed: Operation not permitted.”

原因：

容器桌面安全考虑，没有开放 privileged 权限。

解决方法：

把需要挂载的文件解压后放到容器中使用。

13. tensorboard 有时打开后为空页面，没有加载出 tensorboard 主页面。

原因：

tensorboard 需要的 json 文件没有加载完。

解决方案：

关闭页面重新打开。

14. 开发环境容器桌面中启动 vscode 闪退。

原因：

vscode 在容器桌面中启动需要使用非隔离模式。

解决办法：

启动参数增加--no-sandbox，例如：./code --no-sandbox

15. 使用 libaiflow.log_figure 保存并记录图表 html 时，在实验管理界面中 html 无法显示。

原因：

plotly 生成的图表 html 文件中对于 plotly 库的引用使用的是 cdn 模式，需要能够联网加载 plotly.js 能正常显示图表。

解决办法：

建议保存和记录 plotly 时使用 .png, .jpeg 等图片格式替代 html 格式

16. 桌面容器在 firefox 中获取剪切板内容失败，导致无法粘贴从本地复制的内容

原因：

Firefox 存在剪贴板访问安全限制。

解决方案 A：

可使用门户登录页推荐的浏览器。

解决方案 B：

步骤 1、打开火狐浏览器地址栏输入：about:config

步骤 2、点击接收风险并继续，搜索：

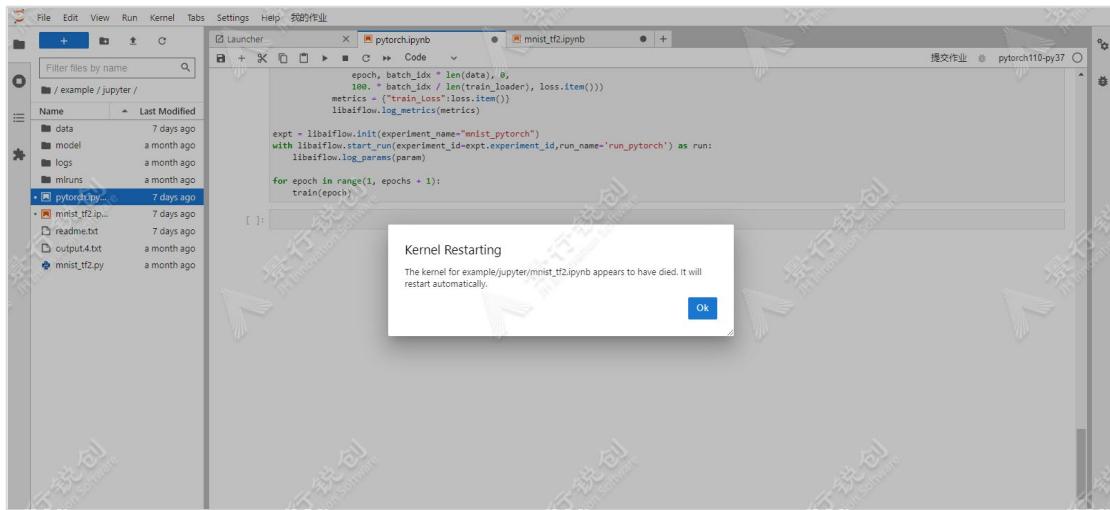
dom.events.testing.asyncClipboard、

dom.events.asyncClipboard.readText 将其值修改为 true; (注：如果 dom.events.asyncClipboard.readText 不存在可以不用修改)

步骤 3、重启浏览器即可成功。

17. 开发环境的内存耗尽后可能会引起下面一系列问题

1) Jupyter 中 kernel Restarting



原因：

可能是因为内存占满而导致 Kernel 重启。

解决方案：

可以尝试切换更高版本内存的硬件规格。

2) Jupyter 报错内存已满或者 “CUDA error: out of memory”

原因：

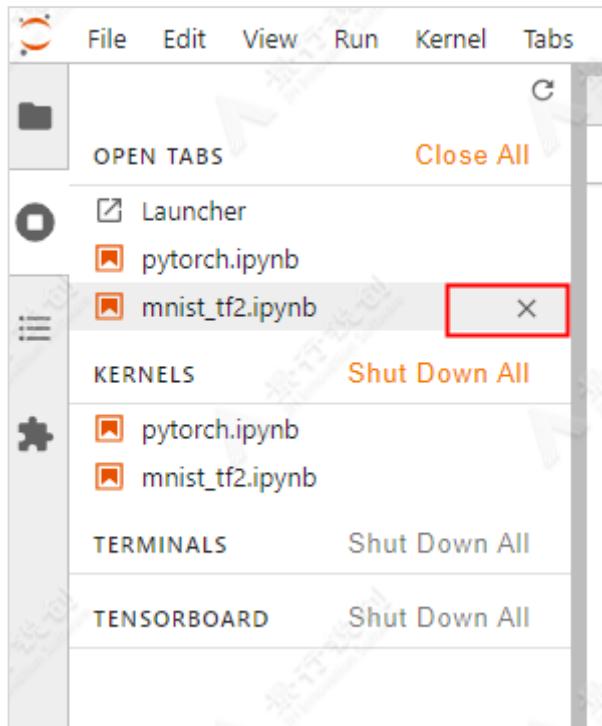
这类内存或显存满了的错误，原因是正在运行中的案例脚本太多。

解决方案：

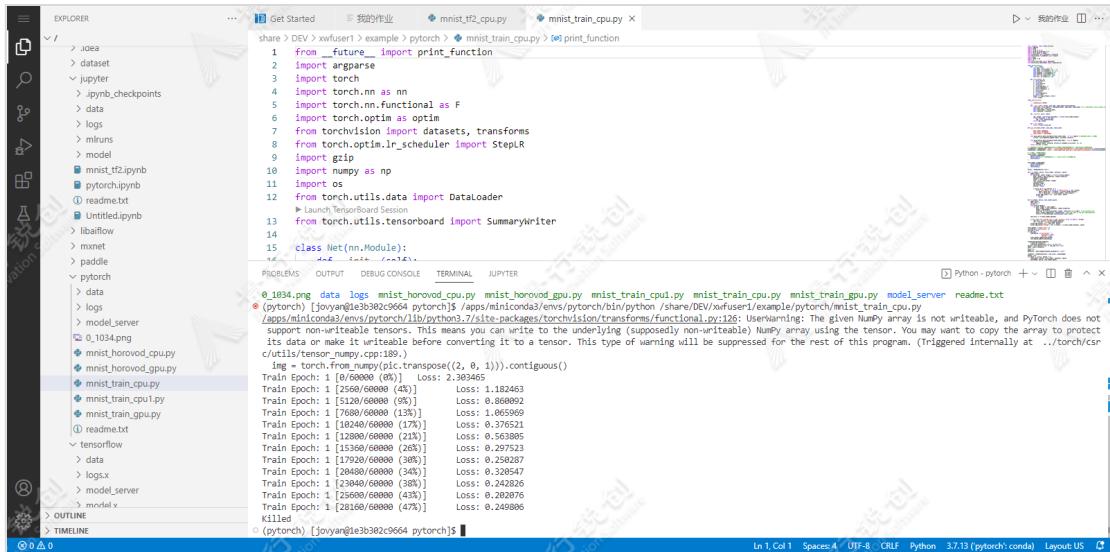
关掉一些正在运行的脚本即可释放内存或显存，操作方法如下：

进入到如下页面，检查后台正在运行的脚本，点击“×”关闭不需要运行的脚本。

附录八



3) 问题: vscode 中运行的程序直接被 Killed。



原因：

可能是内存占满导致。

解决方法：

检查开发环境内存是否用完，释放内存，或者可以尝试切换更高版本内存的硬件规格，或调整程序减少内存使用。

4) 问题:CentOs/Ubuntu 桌面开发环境连不上,提示错误:“会话登录超时!”。

原因:

可能是内存占满导致。

解决方法:

检查 CentOs/Ubuntu 桌面开发环境内存是否用完, 可以等待一段时间, CentOs/Ubuntu 桌面中使用的内存释放后就会恢复正常, 或者可以尝试切换更高版本内存的硬件规格。