

目录

目录	1
SDK简介	2
产品特性	2
参数说明	2
操作步骤	3
Tensorflow 提交模型训练任务	3
Pytorch提交模型训练任务	3

SDK简介

King SDK 是金山云人工智能平台 KingAI 提供的开源软件包，让用户可以通过SDK代码向 KingAI 提交机器学习和深度学习的训练任务。用户无需关注底层的系统配置，更专注于模型和算法等业务逻辑。用户直接调用KingAI SDK即可轻松管理数据集、启动AI训练以及生成模型并将其部署上线。

KingAI SDK目前只提供Python语言的SDK。

产品特性

- 内置了深度优化的 TensorFlow 和 PyTorch 等多种流行的深度学习框架。
- KingAI SDK分为local模式和remote模式。local模式下，会开启一个独立的conda执行环境；remote模式，会对接模型训练服务。
- SDK当前支持tensorflow、pytorch、xgboost框架，同时支持自定义框架或者pip源。
- 对接数据管理，支持基于datasetid为训练指定数据源。
- 对接模型管理，可以注册训练好的模型。
- 用机器学习平台能力，支持 CPU 和 GPU 等多种算力类型。

参数说明

参数：	说明：	备注：
framework_name	框架名称，比如 tensorflow/pytorch等。	在自定义训练中，需要指定框架名称。具体训练中，无需指定，比如基于Tensorflow训练， <code>tf_estimator = Tensorflow(xxxx)</code> 。参数类型：字符串
framework_version	框架版本	local模型下，可以指定pip源支持的版本。remote模式，必须是模型训练服务支持的版本 参数类型：字符串，必需指定
entry_point	入口执行文件	参数类型：字符串，必需指定
source_dir	执行代码所属的目录	训练时，训练代码的完成路径为： <code>source_dir/entry_point</code> 参数类型：字符串，如果不指定，代表当前目录
hyperparameters	超参	参数类型：dict形式，可选 如： <code>{ "epochs": 1, "batch-size": 200 }</code>
envs	训练使用的环境变量信息	参数类型：dict形式，可选 如： <code>{ "ADDRESS": "beijing" }</code>
image_name	使用的镜像名称	参数类型：字符串，可选。remote模式，基于框架名称和版本，训练服务会拼成使用的镜像。
train_instance_type	训练的类型	可选参数：local和remote local：本地执行 remote：远程执行 参数类型：字符串，可选，默认值是local
train_instance_local_mode	本地模式下，训练的方式	当前仅支持conda，后续会支持container。仅在train_instance_type设为local模式下才生效。参数类型：字符串，可选，默认值是conda
base_job_name	作业名称的前缀	参数类型：字符串，可选
output_path	模型输出目录	当前仅支持指定本地，后续会支持输出到ks3 参数类型：字符串，可选
train_engine	训练使用的环境对象	参数类型：TrainEngine类型，可选
dependencies	训练代码依赖的pip包	参数类型：列表，可选 如： <code>dependencies=['numpy==1.15.1']</code>
python_version	使用的python版本	默认是3.7，当前仅在local模式下生效。远程模式下，依赖镜像中的python版本。参数类型：字符串，可选
kai_conf	训练上下文配置	参数类型：KaiConf，可选，默认None
is_registered	是否注册到模型管理	仅在local模式下生效
register_model_config	模型注册时，使用的配置信息	当remote模式训练或者local模式is_register为True时，此参数必须设置。dict形式 <code>register_model_config={ 'experiment_id': '4a4be455-bb0a-4ac9-9bef-d7811abe9b30', 'model_name': 'remote-train-test' }</code> ，设置关联实验以及模型名称
model_metadata_path	Model.py所在的路径	如果模型文件和服务发布使用的Model.py不在一个目录，那么可以执行此参数
resource_dict	训练使用的资源	仅在remote模式下生效，dict形式 如： <code>resource_dict={ 'cpu': 12, 'memory': 24, "gpu": 1, "type": "p40" }</code>

preprocess 训练前置处理 参数类型：字符串, 可选
 postprocess 训练后处理 参数类型：字符串, 可选

操作步骤

1. 在容器开发中新建任务，进入jupyter Notebook编译环境。
2. 准备训练的代码文件。
3. SDK提交训练任务分为remote和local两种模式
 - 如果为remote模式，需要将训练的代码文件上传到KS3指定目录。
 - 如果为local模式，需要将训练的代码文件放在jupyter Notebook编译环境的主目录下（/home/kai）。
4. 输入示例代码，模型训练模块会同步该条训练任务。

Tensorflow 提交模型训练任务

本文档将向您介绍如何在Notebook通过SDK将训练的Tensorflow 模型发布到模型训练模块中。示例代码及注释如下：

示例：local模式

```
from kai_sdk.train.tensorflow.estimator import Tensorflow
from kai_sdk.conf.kai_conf import KaiConf

def local_train():
    kai_conf = KaiConf()
    tf_estimator = Tensorflow(
        framework_version='2.4.0',           # 框架为Tensorflow
        entry_point='tf_mnist.py',           # 输入Tensorflow的框架版本
        source_dir='/home/kai',              # 入口执行文件
        hyperparameters={                    # 代码文件存放的目录
            "epochs": 1,                     # 训练任务的参数设置
            "batch-size": 200
        },
        envs={                                # region地址
            "ADDRESS": "beijing"
        },
        train_instance_type='local',          # 训练任务的方式为local模式
        train_instance_local_mode='conda',   # 本地模式下，训练的方式，目前只支持conda
        base_job_name='base_job',           # 作业名称的前缀
        dependencies=[                        # 训练代码依赖的pip包
            'pymysql'
        ],
        # output_path='model',                # 模型输出目录, 当前仅支持指定本地, 后续会支持输出到ks3
        kai_conf=kai_conf,                   # 训练上下文配置
        is_register=True,                    # 是否注册到模型管理
        model_meta_path='/home/kai/Model.py', # Model.py所在的路径
        register_model_config={              # 模型注册时, 使用的配置信息
            'experiment_id': '439642fb-4cbd-4af0-8ee2-490e282b3f00',
            'model_name': 'my-test'
        }
    )
    """
    inputs: "" or dict("train": "", "test": "")
    dataset
    """
    tf_estimator.fit({'train': '/home/kai'})
if __name__ == '__main__':
    local_train()
```

注意：每次创建一个工作区会生成一个experiment_id。查看方式：在实验管理列表中，选择要连接的实验，单击实验名称，跳转到实验管理详情页中，该页面生成的url后缀有一行id码，复制该id替换到容器开发中的experiment_id。

Pytorch提交模型训练任务

本文档将向您介绍如何在Notebook通过SDK将训练的Pytorch模型发布到模型训练模块中。示例代码及注释如下：

示例：remote模式

```
from kai_sdk.train.pytorch.estimator import Pytorch
from kai_sdk.conf.kai_conf import KaiConf
```

```
def remote_train():
    kai_conf = KaiConf()
    pytorch_estimator = Pytorch(
        framework_version='1.6.0-torchvision0.7-py3.6', #框架为Pytorch
        entry_point='train.py', #输入Pytorch的框架版本
        source_dir='ks3://xc-train-ksc/yolo3-pytorch-master', #入口执行文件
        hyperparameters={ #代码文件存放的目录
            # "epochs": 1, #训练任务的参数设置
            # "batch-size": 200
        },
        envs={ #训练使用的环境变量信息
            # "ADDRESS": "beijing"
        },
        train_instance_type='remote', #训练的类型
        # base_job_name='base_job', #作业名称的前缀
        dependences=[ #训练代码依赖的pip包
            # 'numpy==1.15.1',
            # 'Pillow == 5.3.0',
            # 'scipy==1.1.0',
            # 'wget==3.2',
            # 'seaborn==0.9.0',
            # 'opencv-python',
            # 'easydict',
            # 'tqdm'
        ],
        # output_path='model', #模型输出目录
        kai_conf=kai_conf,
        register_model_config={ #模型注册时,使用的配置信息
            'experiment_id': '439642fb-4cbd-4af0-8ee2-490e282b3f00',
            'model_name': 'remote-train-test-pytorch'
        },
        resource_dict={ #训练使用的资源
            'cpu': 4,
            'memory': 8,
            'gpu': 0,
            'type': "p40"
        },
        preprocess = "cp sources.list /etc/apt/sources.list && apt-get update && apt-get install sudo && sudo apt-get install -y
        libgl2.0-dev libsm6 libxext6 libxrender-dev libgl1-mesa-glx" #训练前置处理
    )
    pytorch_estimator.fit({'data_path': 'ks3://xc-train-ksc/VOC_test'})

if __name__ == '__main__':
    remote_train()
```