

JetAutoML 实验手册

目录

系统概述

流程概述

流程详细说明

常见问题

系统概述

本系统旨在提供实验过程的实验环境记录、实验过程记录、实验结果统计对比等功能。

流程概述

实验流程

整个实验流程分为如下几个步骤：

1. JetAutoML 构建项目流程
2. 下载运行 Tomcat，作为上传代码的应用
3. 编写本地代码，通过下载配置引用 python 模块 pzmllog 编写需要记录的 log、模型等相关代码
4. 运行本地代码
5. 在 JetAutoML 平台上查看实验的过程和结果

项目流程构建

为了实现上述功能，整个流程需要三个部分：代码仓库（用于与本地实验代码对应），实验开展 / 模型训练与测试（用于记录模型训练过程），实验开展 / 实验对比图表（用于展示和对比实验结果）

这三个模块是 JetAutoML 系统中项目的一部分，这些节点的类型分别是代码仓库 -> 实

验开展 / 模型训练与测试 -> 实验开展 / 实验对比图表。一个最简单的流程如下图所示。（注意三个节点之间的顺序需要在流程图中保持）



项目流程图

实验环境准备

与上述流程中三个部分分别对应，实验环境需要如下的工具

外部 python 第三方包 pzmllog：用于记录本地所输出的 log、实验的环境、模型等，[pzmllog 下载链接](#)

Tomcat：用于上传代码、过程文件、模型等，[windows 下载链接](#)，[Mac 下载链接](#)，[Linux 下载链接](#)

流程详细说明

实验流程

重新回顾上述整个实验的流程，下面对每一个部分进行详细介绍：

1. JetAutoML 构建项目流程
2. 运行 Tomcat，作为上传代码的应用
3. 编写本地代码，通过引用 python 模块 pzmllog 编写需要记录的 log、模型等相关代码
4. 运行本地代码
5. 在 JetAutoML 平台上查看实验的过程和结果

构建项目流程

在 JetAutoML 平台创建项目，并且为这个项目添加三个节点：

节点 1：项目仓库，选项如下

关联子流程

请选择

节点类型

☒ 定制节点 ☐ 内置节点

工作事项设置

设置

基础字段

☐ 单行文本

☐ 多行文本

☒ 计数器

☒ 多选框组

☐ 日期选择器

☒ 颜色选择器

☐ 开关

☐ 文字

☐ 多行文本

☒ 单选框组

☐ 时间选择器

☐ 评分

☒ 下拉选择框

☐ 滑块

☐ H 标题

高级字段

☒ 自定义区域

☐ 图片

☐ 视频上传

☐ 附件上传

☐ 编辑器

☐ 级联选择器

☐ 代码编辑器

☐ LaTeX

☐ 代码仓库

☐ 文献

预览

+ 代码仓库

仓库名称

创建仓库

节点 2：实验开展/模型训练与测试，选项如下

节点类型

☐ 定制节点 ☒ 内置节点

内置节点选择

实验开展 / 模型训练与测试 ^

<input type="radio"/> 实验开展 >	<input checked="" type="radio"/> 模型训练与测试
<input type="radio"/> 数据集准备	<input type="radio"/> 实验对比图表

节点 3：实验开展 / 实验对比图表，选项如下

节点类型

☐ 定制节点 ☒ 内置节点

内置节点选择

实验开展 / 实验对比图表 ^

<input type="radio"/> 实验开展 >	<input type="radio"/> 模型训练与测试
<input type="radio"/> 数据集准备	<input checked="" type="radio"/> 实验对比图表

最终形成如下的流程图



项目流程图

运行 Tomcat

安装 Tomcat

Windows: 直接安装即可

Linux: 安装好后, 需要以下命令赋权

```
Shell
chmod -R +x $HOME/pzmlab/apache-tomcat-9.0.62
```

创建文件目录

```
Shell
sudo mkdir /opt/jml_ai_result && sudo chmod -R 777
```

Mac: 除 Linux 流程外额外安装 JDK

运行 Tomcat

安装好 Tomcat 后, 首先需要配置 IP 地址, 来连接 JetAutoML 平台:

配置文件的地址:

Tomcat 安装目录\webapps\ml_client\WEB-INF\classes\application.yml

```
server:
  port: 5560
  servlet:
    context-path: /ml_client
logging:
  level:
    com.kingpoint: debug
    org.springframework: warn
spring:
  profiles:
    active: windows
jml:
  server:
    ##服务端远程地址
  #   url: http://192.168.30.143:8333/process-api/process-experiment/upload
  #   url: http://172.16.34.41:8333/process-api/process-experiment/upload
  #   url: http://localhost:8108/process-experiment/upload
```

application.yml

所需要修改的 IP 地址: 将上述 url: 中的 ip 修改为 JetAutoML 平台的 ip 地址, 如图

修改配置文件后, 运行 tomcat, tomcat 运行脚本所在位置如下:

Windows: Tomcat 安装目录\bin\startup.bat

Linux/Mac: Tomcat 安装目录\bin\startup.sh

Linux/Mac 运行脚本命令

Shell

```
sh $HOME/pzmlab/apache-tomcat-9.0.62/bin/startup.sh
```

编写本地代码

为了记录实验过程、上传实验相关信息，需要安装第三方包 pzmllog:

安装命令（需在 pzmllog-1.0.0.tar.gz 文件目录下运行下面的命令）

```
pip install psutil pyyaml nvidia-ml-py requests
pip install pzmllog-1.0.0.tar.gz
```

安装好上述包后，为连接 JetAutoML，配置 python 代码如下

Python

```
from pzmllog import NewLogger
log = NewLogger(
    config={
        #用户 ID
        'access_token':"gtta9c6tdvqsryo8ggnimg0",
        #项目 ID
        'project':"3711",
        #实验描述和说明信息
        "description":"MNIST ",
        #自定义实验名称
        "experiment_name":"MNIST TEST",
        # 仓库 ID
        "repository_id":"10da8fd5fa464b03ad68d92e56ae932a",
        # tomcat 的启动端口
        'port': "5560"
    },
    # 超参数集
    info = {
        "learning_rate": 0.001,
        "epoch": 3,
        "batch_size": 64
    }
)
```

上述配置中的参数解释：

access_token: 用户令牌，JetAutoML 右上角个人信息->账号安全中生成（注意不要使用过期的令牌/无权限令牌）

project: 对应实验开展/模型训练与测试节点中的任务 ID

负责人: 请选择负责人

执行者: 请选择执行者

设置元数据时间: 请选择时间

任务ID: 3711

代码仓库

查看实验指引

实验名称	标签	描述	创建者

仓库名称	仓库ID
	10da8fd5fa464b03ad68d92e56ae932a
	18af7af9628240caad0a90af6f1e0f50

repository_id: 仓库 ID，对应节点：（下面两个节点均可找到仓库 ID）

代码仓库：

代码仓库

负责人: 请选择负责人

执行者: 请选择执行者

代码仓库:

仓库名称

仓库ID

10da8fd5fa464b03ad68d92e56ae932a

实验开展/模型训练与测试：

负责人: 请选择负责人

执行者: 请选择执行者

设置元数据时间: 请选择时间

任务ID: 3711

代码仓库

查看实验指引

实验名称	标签	描述	创建者

仓库名称	仓库ID
	10da8fd5fa464b03ad68d92e56ae932a
	18af7af9628240caad0a90af6f1e0f50

tomcat 端口默认即可，其余参数可按需要填写，上面是以 MNIST 为例

编写好 Logger 后，可以用下面的代码做最基础的上传代码测试

```
Python
from pzmllog import NewLogger

log = NewLogger(
    config={
        #用户 ID
        'access_token':"gtta9c6tdvqsryo8ggnnimg0",
        #项目 ID
        'project':"3867",
        #实验描述和说明信息
```

```

        "description": "MNIST ",
        # 自定义实验名称
        "experiment_name": "MINIT EXAMPLE",
        # 仓库 ID
        "repository_id": "91932bc4ce5a41d4b60420e241ef9d85",
        # tomcat 的启动端口
        'port': "5560"
    },
# 超参数集
    info = {
        "learning_rate": 0.001,
        "epoch": 3,
        "batch_size": 64
    }
)

data_length = 10
epoch = 2
for e in range(epoch):
    # 开始实验
    log.Run()
    for i in range(data_length):
        # 实验部分
        loss = 0.1*(10 - i)
        accuracy = (0.1 * i)
        # 记录 Log
        log.Log({"epoch": e, "loss": loss, "accuracy": accuracy})
    # 实验模型路径
    model_path = "model.pt"
    # 记录模型
    log.Save([model_path])
    # 结束实验
    log.End()

# 结束整个过程
log.Submit()

```

运行本地代码&在 JetAutoML 平台上查看实验的过程和结果

将上述代码在本地运行（注意步骤 2 中已将 tomcat 开启），可以在实验开展/模型训练与测试节点看到类似如下的结果

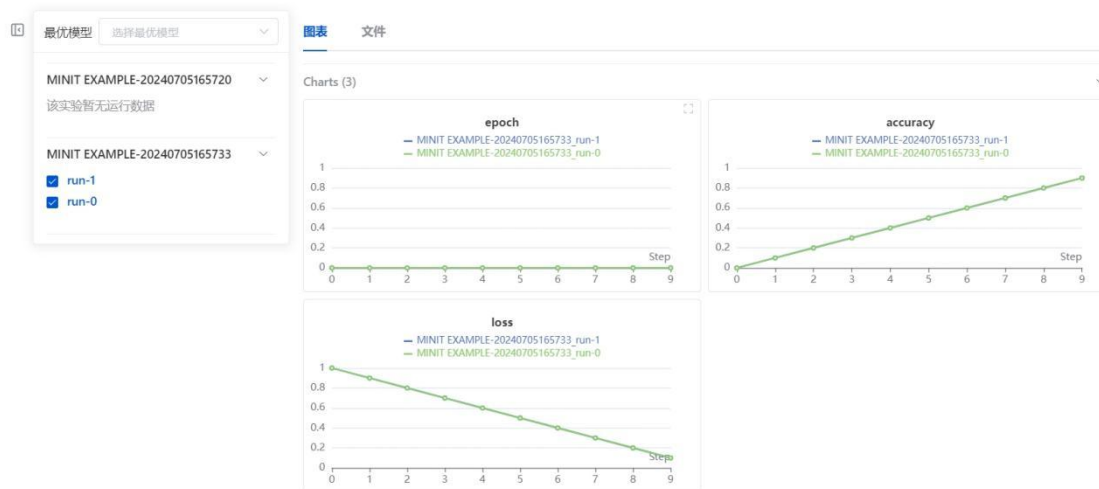
任务ID: 3867

代码仓库

查看实验指引

实验名称	标签	描述	创建者	创建时间	运行时间	模	操作
MINIT EXAMPLE-20240705165720	+	MNIST 2	高瑜琦	16秒前	26秒		

在实验开展 / 实验对比图表节点看到类似的结果



分别在这两个节点可以看到上述代码的实验结果，整个流程完成。

常见问题

Q: 是否允许这三个节点中间或前序插入其他节点

A: 允许，但要注意三个节点之间的顺序要保持，且不要使用其中一部分（例如只使用两个节点）

Q: 是否允许项目使用多个代码仓库

A: 允许，在项目流程中前序增加代码仓库节点

Q: 完成上述设置，但节点没有记录，且程序没有报错怎么办

A: 如确定环境配置无误，可能由于令牌过期/无权限，解决方法是按如下步骤操作：

1. 确保 python 进程结束
2. 删除 C:/jml_ai_result (windows 之外的平台： /opt/jml_ai_result)
3. 重启 tomcat

4. 运行 python 程序

Q: jml_ai_result 占用了大量磁盘空间，如何处理

A: 使用项目根目录下的 .path_ignore 文件（需创建）屏蔽过大/不想上传的文件（默认屏蔽 datasets）

Q: log.save 报错未找到文件

A: log.save 接受 list 存储的文件路径作为参数，请检查对应的路径是否存在这个文件