

codeExplorer 使用说明书

最后修订时间: 25.10.10

目录

1 介绍	1
2 主要功能	1
2.1 模型仓库	1
2.2 代码数据集	2
2.3 任务列表	4
2.3.1 预处理任务列表	4
2.3.2 训练任务列表	6
2.3.3 测试任务列表	8

1 介绍

CodeExplorer 是一个汇集代码扫描工具的软件平台。使用这些代码扫描工具时，需要上传待分析的代码库，然后依次进行代码预处理、模型训练、模型使用和结果可视化等多个步骤（可分离使用）。通过扫描代码，可以发现隐藏在代码中的缺陷（如代码克隆、特征嫉妒等），用于评估和改善软件质量。

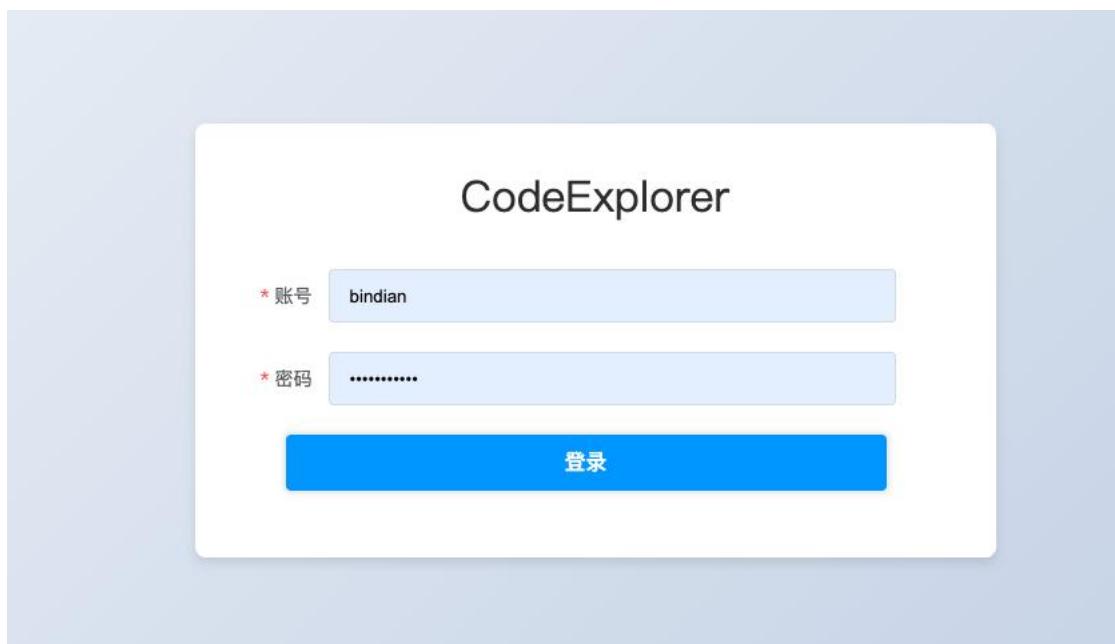
本文旨在帮助读者快速了解网页使用，详细介绍各部分具体内容及功能。接下来，将按照模型仓库、代码数据集、预处理任务列表、训练任务列表、测试任务列表的顺序依次进行介绍。

2 主要功能

2.1 登录

账号: bindian

密码: bindian2024



2.2 模型仓库

模型仓库用于展示当前所有的模型信息，除了表格中展示的基本信息外，点击“查看”按钮可以看到模型训练时设置的参数。点击“删除”按钮可以删除所选模型，**注：此操作不可逆**。模型会在训练任务结束时自动生成，用户不可自行导入，在测试阶段可选表格中任一模型进行代码测试。

CodeExplorer 首页 / 模型仓库

模型仓库

代码数据集

任务列表

编号	名称	模型	模型类型	描述	创建日期	操作
3	李生图匹配网络1	CodeGraph4CCDetector	代码克隆检测	一种基于 CFG/PDG 注意力机制的李生图匹配网络。它可以在 token、语句和图级别捕获关键信息，以充分学习代码语义。	2024-10-28 14:29:37	查看 删除
63	测试训练	测试训练	代码克隆检测	模型描述	2024-11-01 10:07:15	查看 删除
80	任务名称	输出模型名称	代码克隆检测	模型描述	2024-11-04 13:56:08	查看 删除
125	任务名称1	输出模型名称	代码克隆检测	模型描述	2024-12-18 23:04:22	查看 删除
126	任务名称	输出模型名称	代码克隆检测	模型描述	2024-12-19 10:54:16	查看 删除
127	任务名称1	输出模型名称	代码克隆检测	模型描述	2024-12-25 15:27:17	查看 删除
129	任务名称1	输出模型名称1	代码克隆检测	模型描述	2024-12-26 13:36:20	查看 删除
130	任务名称1	输出模型名称	代码克隆检测	模型描述	2025-01-20 22:04:30	查看 删除
131	任务名称1	输出模型名称	代码克隆检测	模型描述	2025-01-20 22:05:51	查看 删除

共 1000 条 < 1 2 3 4 5 6 ... 10 > 前往 1 页

模型仓库列表

CodeExplorer 首页 / 模型仓库

模型仓库

代码数据集

任务列表

预处理任务列表

训练任务列表

代码克隆检测

代码异味检测

代码漏洞检测

测试任务列表

代码克隆检测

代码异味检测

代码漏洞检测

模型训练参数

任务名称 任务名称1

输出模型名称 输出模型名称

任务类型 代码克隆检测

数据集 googlejam4_src

模型描述 模型描述

参数设置 epochs 100

batch_size 64

学习率 0.0001

学习衰减因子 0.5

分类数 2

学习率衰减步长 50

X

2024-10-28 14:29:37 查看 删除

2024-11-01 10:07:15 查看 删除

2024-11-04 13:56:08 查看 删除

2024-12-18 23:04:22 查看 删除

2024-12-19 10:54:16 查看 删除

2024-12-25 15:27:17 查看 删除

2024-12-26 13:36:20 查看 删除

2025-01-20 22:04:30 查看 删除

2025-01-20 22:05:51 查看 删除

共 1000 条 < 1 2 3 >

关闭

模型仓库“查看”按钮

2.3 代码数据集

代码数据集用于展示当前所有的数据集信息，包括已预处理数据集和未预处理数据集。同样的，用户除了可以看到基本信息外，点击“查看”按钮可以看到此数据集中的具体内容。点击“新增原始数据集”可将本地数据上传至网站，用于检测与评估。

CodeExplorer 首页 / 已预处理数据集

模型仓库 代码数据集 任务列表

代码数据集信息表

+ 新增原始数据集

编号	数据集名称	描述	状态	创建日期	操作
7	第二数据集		● 已处理	2011-11-23 13:02:47	查看 删除
8	googlejam4_src		● 已处理	2011-11-23 13:02:47	查看 删除
44	googlejam4_src	数据集描述	● 已处理	2024-12-19 13:45:21	查看 删除
45	googlejam4_src	数据集描述	● 已处理	2024-12-19 16:11:44	查看 删除
54	googlejam4_src	数据集描述	● 已处理	2025-01-06 15:34:30	查看 删除
71	test	数据集描述	● 已处理	2025-01-12 23:12:26	查看 删除
83	googlejam4		● 未处理	2025-01-13 11:16:55	查看 删除
84	googlejam4		● 未处理	2025-01-13 11:18:00	查看 删除
85	test		● 未处理	2025-01-13 11:18:07	查看 删除

共 1000 条 < 1 2 3 4 5 6 ... 10 > 前往 1 页

代码数据集按钮

CodeExplorer 首页 / 已预处理数据集

模型仓库 代码数据集 任务列表

代码数据集信息表

+ 新增原始数据集

编号	数据集名称	描述	状态	创建日期	操作
7	数据集内容	<p>文件结构:</p> <ul style="list-style-type: none"> ▼ googlejam4_src <ul style="list-style-type: none"> ▼ 1 <ul style="list-style-type: none"> googlejam1.p019.Mushroom.java googlejam1.p021.mushroom.java googlejam1.p022.A.java googlejam1.p041.A.java ► 2 	<p>源代码:</p> <pre> package googlejam1.p019; import java.io.BufferedReader; import java.io.InputStreamReader; import java.util.StringTokenizer; public class A { public static void main(String[] args) throws Exception { BufferedReader bf = new BufferedReader(new InputStreamReader(System.in)); int ntets = Integer.parseInt(bf.readLine()); for (int tc = 1; tc <= ntets; tc++) { int n = Integer.parseInt(bf.readLine()); int[] arr = new int[n]; StringTokenizer st = new StringTokenizer(bf.readLine()); for (int i = 0; i < n; i++) { arr[i] = Integer.parseInt(st.nextToken()); } int firstMin = 0; int maxDiff = 0; for (int i = 1; i < n; i++) { int diff = -arr[i] + arr[i - 1]; if (diff > 0) { firstMin += diff; } } maxDiff = Math.max(maxDiff, diff); } } } </pre>		查看 删除 查看 删除

共 1000 条 < 1 2 3 4 5 6 ... 10 > 前往 1 页

代码数据集“查看”按钮

2.4 任务列表

任务分为三种：预处理任务、训练任务以及测试任务，三种任务相互独立，但 gpu、模型资源共享。

2.4.1 预处理任务列表

预处理任务会将未预处理的数据集变为已处理数据集。操作流程为：

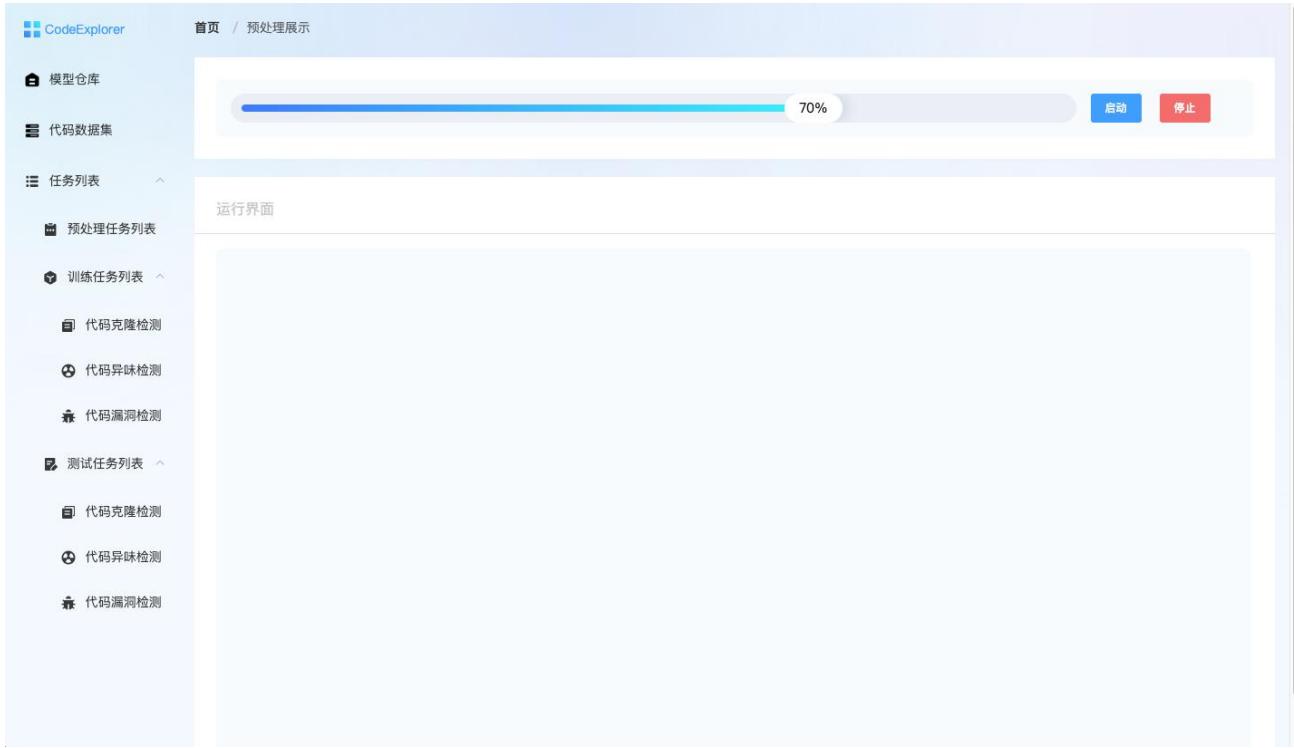
1. 进到任务列表-预处理任务列表，点击“新增预处理任务”，填入信息，选择想要预处理的代码。

The screenshot shows the 'Code Explorer' interface with the 'Pre-processing Task List' selected in the sidebar. A modal window titled 'Pre-processing Task Settings' is open, prompting for 'Task Name' (任务名称) and 'Data Set Description' (数据集描述). The main table lists 10 pre-processing tasks with columns for 'Task ID' (编号), 'Task Name' (任务名称), 'Creation Time' (创建时间), and 'Operations' (操作). Task 30 is currently running. Task 43 was created on 2025-02-25 at 10:00:50. Task 44 was created on 2024-12-16 at 10:14:25. Task 45 was created on 2024-12-16 at 10:16:26. Task 59 is a test task named 'googlejam4_src'. Task 60 and 61 are also test tasks. Task 62 is labeled 'test'. Task 63 has a value '1' in the task name field and 'test' in the code selection dropdown. Navigation controls at the bottom show page 1 of 1000.

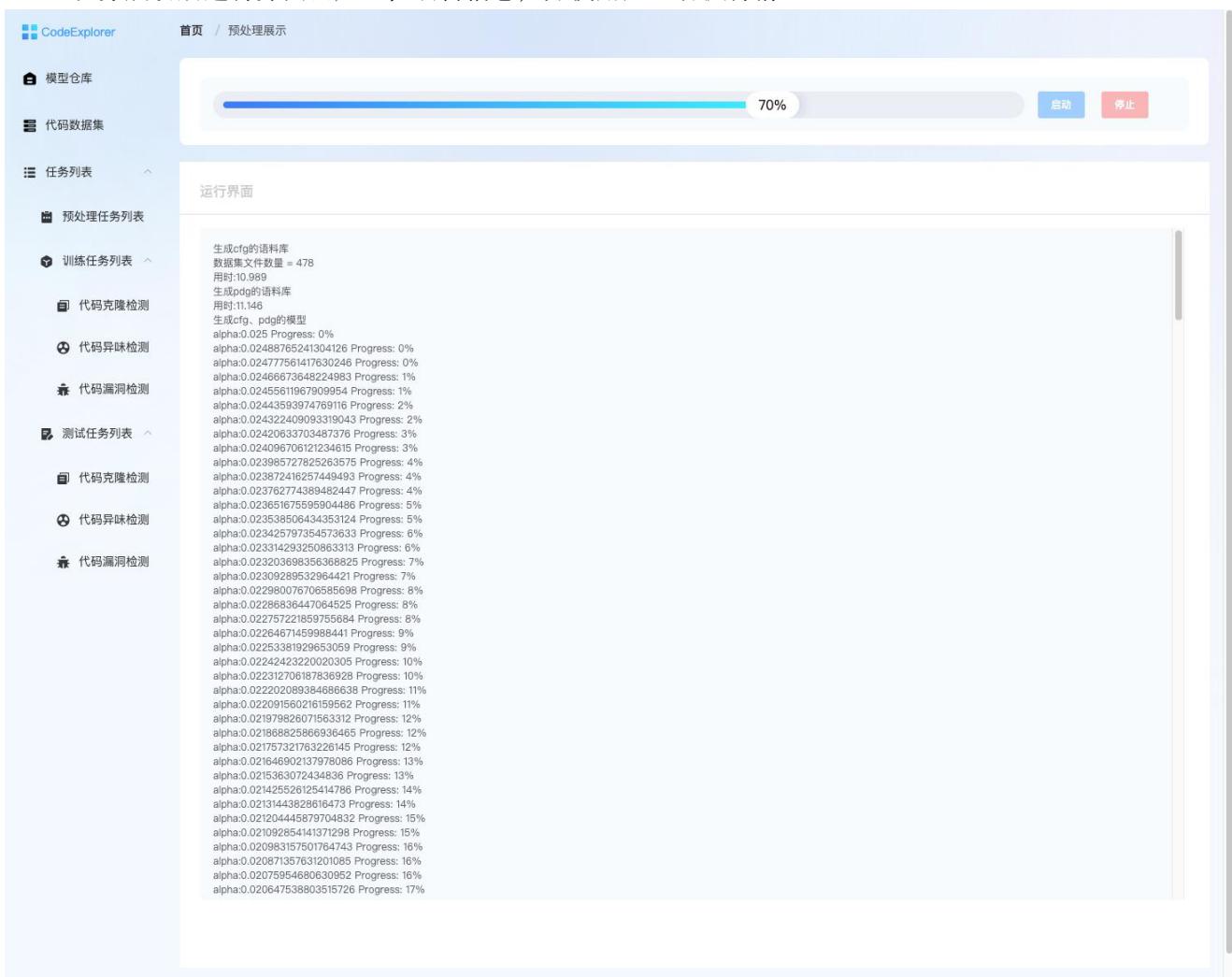
编号	任务名称	创建时间	操作
30	任务名称	运行 2024-12-06 14:11:24	启动 编辑 删除
43	任务名称	运行 2025-01-12 23:14:18	查看 编辑 删除
44	任务名称2	运行 2024-12-16 10:14:25	查看 编辑 删除
45	任务名称3	运行 2024-12-16 10:16:26	查看 编辑 删除
59	测试任务名称	进行中 2025-01-06 15:20:12	查看 编辑 删除
60	测试任务名称	未运行 2025-01-06 15:15:15	启动 编辑 删除
61	测试任务名称	未运行 2025-01-06 15:15:15	启动 编辑 删除
62	测试	已运行 2025-01-06 17:35:15	查看 编辑 删除
63	1	未运行 2025-01-13 14:52:56	启动 编辑 删除

“新增预处理任务”窗口

2. 新建完成后可在预处理任务表中看到，任务在未启动前都是可以进行编辑的。
3. 点击“启动”按钮，会跳转到预处理展示界面，点击启动按钮任务，选择空闲的 GPU 后开始执行任务，顶部的进度条会显示总体进行流程，并且用户可随时点击“停止”按钮停止任务。



4. 任务启动后运行界面会产生控制台信息，方便用户查看执行情况。



- 启动成功后用户可随时点击该任务的“查看”按钮查看运行情况。

2.4.2 训练任务列表

训练任务有三种，分别是代码克隆检测、代码异味检测以及代码漏洞检测。这里将着重对代码克隆检测流程的描述。

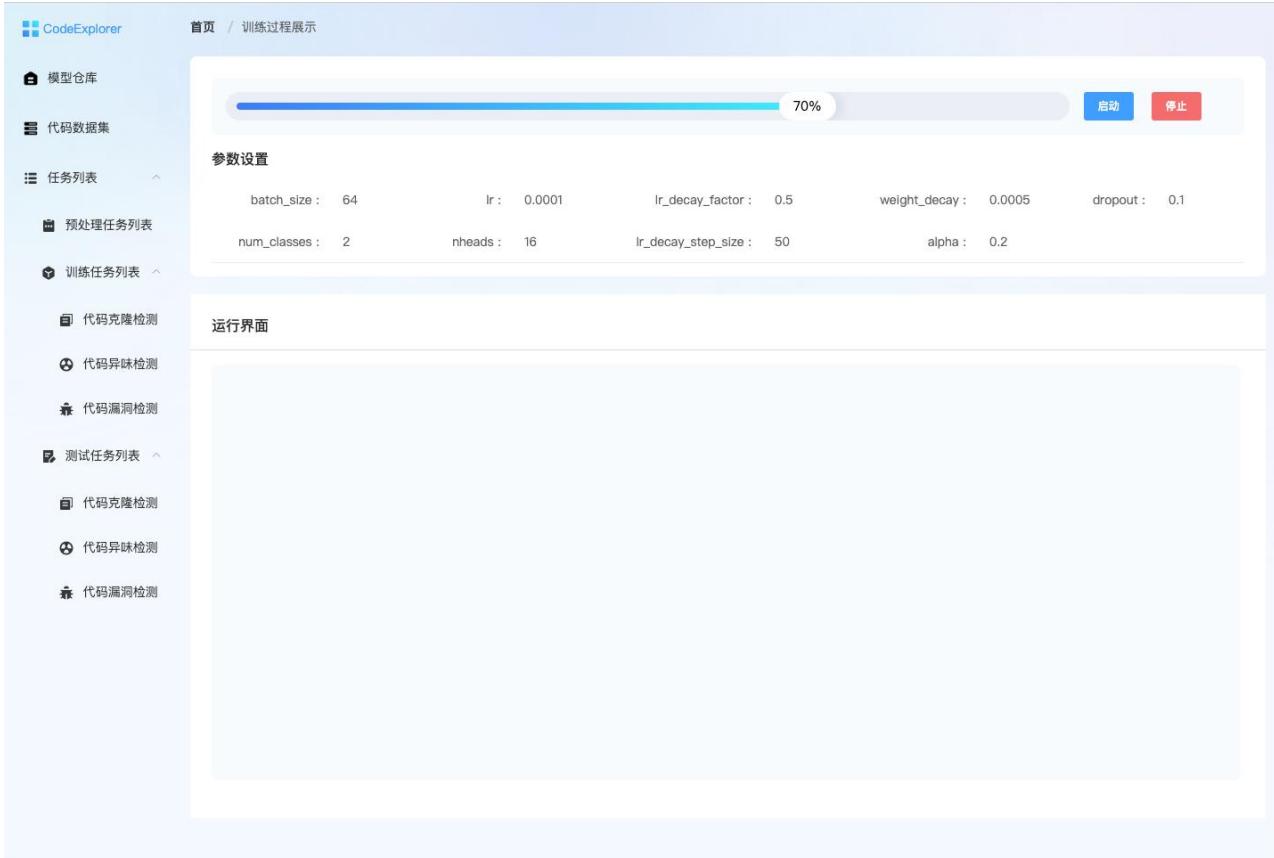
训练任务会对选择的不同数据集训练出不同模型，操作流程为：

- 进入任务列表-训练任务列表-代码克隆检测，点击“新增训练任务”，输入各项信息，并且可以自定义修改参数设置。注意：此处数据集必须是已处理好的数据集才可在此处展示。

状态	ID	操作
进行中	1	查看 编辑 删除
进行中	1	查看 编辑 删除
未运行	2	启动 编辑 删除
未运行	2	启动 编辑 删除
已运行	模	查看 编辑 删除
进行中	模	查看 编辑 删除
已运行	模	查看 编辑 删除
进行中	模	查看 编辑 删除
进行中	模	查看 编辑 删除

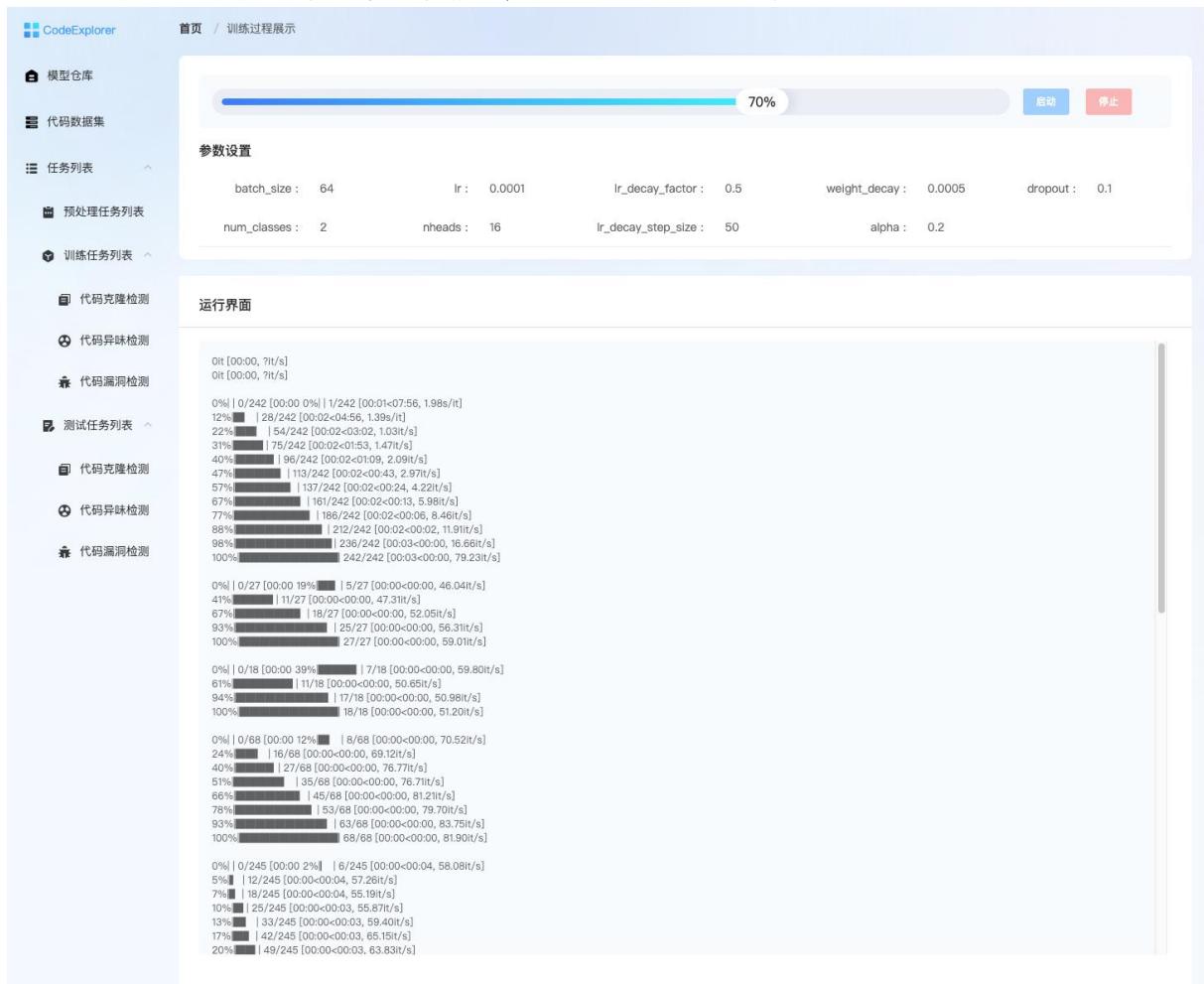
“新增训练任务”按钮界面

- 新建完成后可在代码克隆检测任务表中看到，任务在未启动前都是可以进行编辑的。
- 点击“启动”按钮，会跳转到训练过程展示界面，点击启动按钮任务，选择空闲的 GPU 后开始执行任务，顶部的进度条会显示总体进行流程，并且用户可随时点击“停止”按钮停止任务。



训练任务“启动”按钮界面

4. 任务启动后运行界面会产生控制台信息，方便用户查看执行情况。



训练任务运行界面

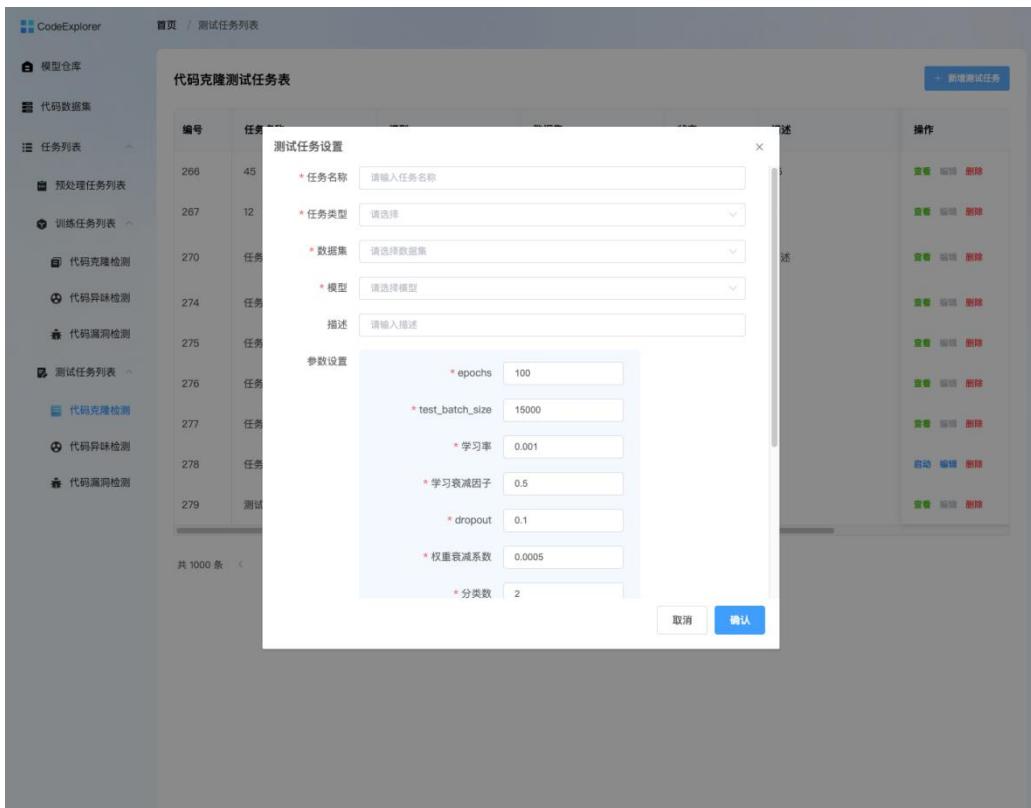
- 启动成功后用户可随时点击该任务的“查看”按钮查看运行情况。

2.4.3 测试任务列表

对应训练任务列表，测试任务同样有三种，这里将着重进行代码克隆检测流程的描述。

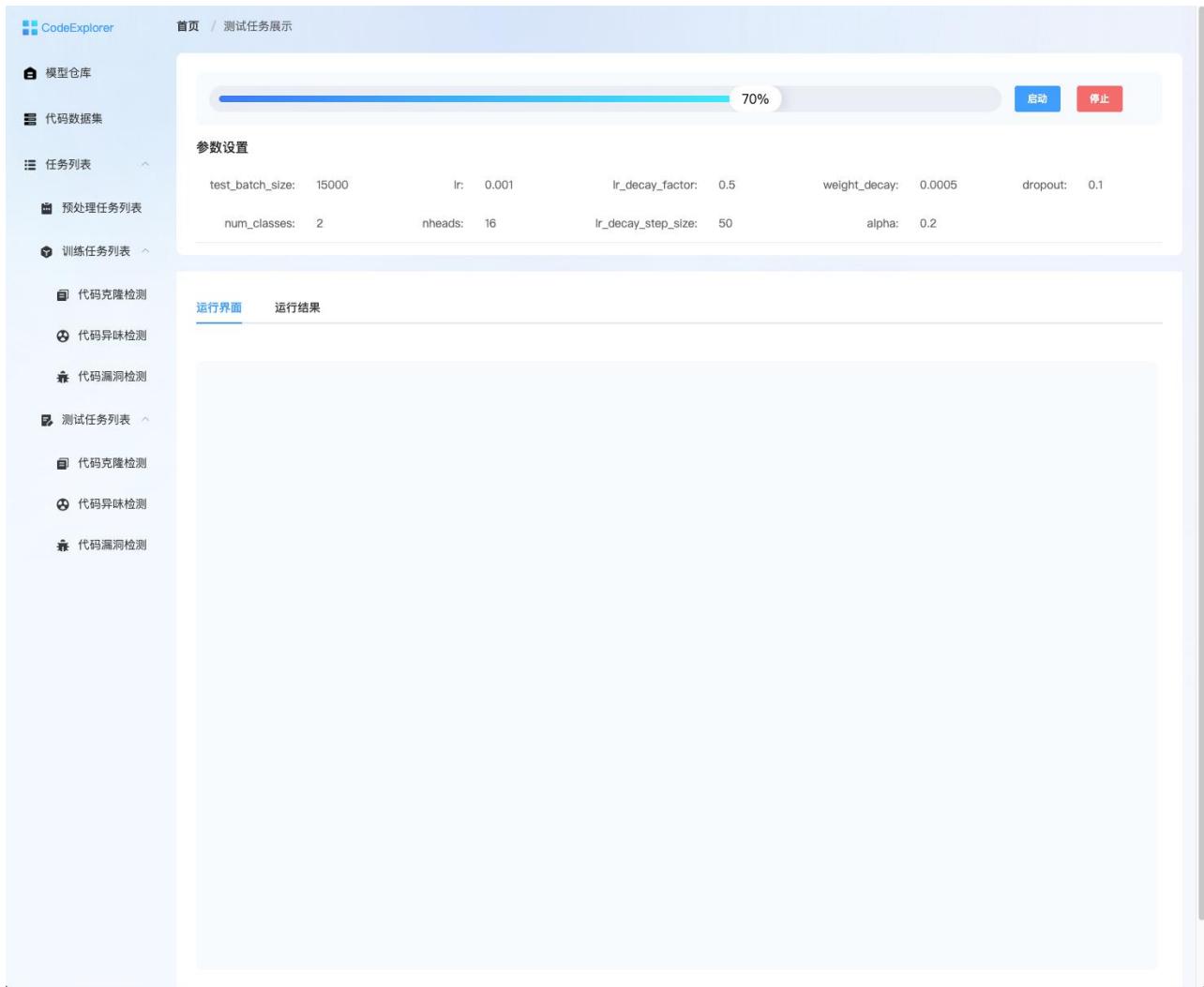
测试任务会对生成的模型与用户选择的代码进行检测，分析与评估模型的好坏。操作流程为：

- 进入任务列表-测试任务列表-代码克隆检测，点击“新增测试任务”，输入各项信息，并且可以自定义修改参数设置。注意：此处数据集必须是已处理好的数据集才可在此处展示。



“新增测试任务”界面

- 新建完成后可在代码克隆测试任务表中看到，任务在未启动前都是可以进行编辑的。
- 点击“启动”按钮，会跳转到测试过程展示界面，同样的，点击启动按钮任务，选择空闲的 GPU 后开始执行任务，顶部的进度条会显示总体进行流程，并且用户可随时点击“停止”按钮停止任务。



测试任务“启动”按钮

4. 任务启动后运行界面会产生控制台信息，方便用户查看执行情况。

CodeExplorer 首页 / 测试任务展示

参数设置

test_batch_size:	15000	lr:	0.001	lr_decay_factor:	0.5	weight_decay:	0.0005	dropout:	0.1
num_classes:	2	nheads:	16	lr_decay_step_size:	50	alpha:	0.2		

运行界面 运行结果

0it [00:00, ?it/s]
0it [00:00, ?it/s]

0% | 0/242 [0:00:00 7%] | 18/242 [0:02<00:25, 8.79it/s]
66% | 160/242 [0:02<00:06, 12.51it/s]
100% | 242/242 [0:02<00:00, 109.18it/s]

0% | 0/27 [0:00:00 100%] | 27/27 [0:00<00:00, 599.17it/s]

0% | 0/18 [0:00:00 100%] | 18/18 [0:00<00:00, 151601.35it/s]

0% | 0/68 [0:00:00 100%] | 68/68 [0:00<00:00, 986.97it/s]

0% | 0/245 [0:00:00 31%] | 75/245 [0:00<00:00, 686.17it/s]
45% | 110/245 [0:00<00:00, 524.64it/s]
68% | 167/245 [0:00<00:00, 522.88it/s]
96% | 235/245 [0:00<00:00, 556.06it/s]
100% | 245/245 [0:00<00:00, 574.86it/s]

0% | 0/2 [0:00:00 100%] | 2/2 [0:00<00:00, 11066.77it/s]

0% | 0/4 [0:00:00 100%] | 4/4 [0:00<00:00, 22826.14it/s]

0% | 0/478 [0:00:27 27%] | 130/478 [0:00<00:00, 1222.48it/s]
58% | 277/478 [0:00<00:00, 1277.14it/s]
73% | 351/478 [0:00<00:00, 1038.68it/s]
100% | 478/478 [0:00<00:00, 1259.07it/s]

0% | 0/435 [0:00:35 35%] | 154/435 [0:00<00:00, 1498.24it/s]
71% | 309/435 [0:00<00:00, 1495.20it/s]
100% | 435/435 [0:00<00:00, 1540.73it/s]

0% | 0/20 [0:00:00 100%] | 20/20 [0:00<00:00, 177349.01it/s]

0% | 0/88 [0:00:00 100%] | 88/88 [0:00<00:00, 1561.32it/s]

0% | 0/38 [0:00:00 100%] | 38/38 [0:00<00:00, 643.50it/s]

0% | 0/157 [0:00:02<06:03, 2.33it/s]
3% | 4/157 [0:02<04:11, 1.64it/s]
4% | 17/157 [0:02<02:54, 1.16it/s]

测试任务运行界面

- 启动成功后用户可随时点击该任务的“查看”按钮查看运行情况。
- 当程序运行完成后，点击运行结果，可查看检测结果。注：目录中红色字体为疑似克隆文件名，点击出现源代码、疑似克隆代码以及克隆代码所在目录。

CodeExplorer 首页 / 测试任务结果

参数设置

test_batch_size:	15000	lr:	0.001	lr_decay_factor:	0.5	weight_decay:	0.0005	dropout:	0.1
num_classes:	2	nheads:	16	lr_decay_step_size:	50	alpha:	0.2		

运行界面 运行结果

文件结构:

- googlejam4_src
 - 1
 - googlejam1.p019.Mt
 - googlejam1.p021.mt
 - googlejam1.p022.Aj
 - googlejam1.p041.Aj
 - googlejam1.p042.mi
 - googlejam1.p046.A.
 - googlejam1.p271.Mt
 - googlejam1.p273.Pr
 - googlejam1.p846.mi
 - googlejam1.p849.A.
 - 2

源代码:

```

package googlejam1.p022;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.StringTokenizer;
public class A {
    public static void main(String[] args) {
        BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(bf.readLine());
        int n = Integer.parseInt(st.nextToken());
        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            arr[i] = Integer.parseInt(st.nextToken());
        }
        int firstMin = 0;
        int maxDiff = 0;
        for (int i = 1; i < n; i++) {
            int diff = arr[i] - arr[firstMin];
            if (diff > 0) {
                firstMin = i;
            }
        }
        int sum2 = 0;
        for (int i = 0; i < n; i++) {
            sum2 += arr[i];
        }
        System.out.println("Case #");
        System.out.println("sum2 = " + sum2);
        int sumrd = 0;
        for (int i = 1; i < n; i++) {
            if (arr[i] < arr[firstMin]) {
                int rd = arr[i];
                sumrd += rd;
                if (maxDiff < rd) {
                    maxDiff = rd;
                }
            }
        }
        System.out.println("sumrd = " + sumrd);
        System.out.println("maxDiff = " + maxDiff);
    }
}

```

克隆代码所在目录:

- googlejam4_src
 - 1
 - googlejam1.p019.Mt
 - googlejam1.p021.mt
 - googlejam1.p022.Aj
 - googlejam1.p041.Aj
 - googlejam1.p042.mi
 - googlejam1.p046.A.
 - googlejam1.p271.Mt
 - googlejam1.p273.Pr
 - googlejam1.p846.mi
 - googlejam1.p849.A.
 - 2

测试任务运行结果