

昇腾AI创新大赛2024昇思赛题比赛指导文档

1 华为云申请代金券指南

在华为云平台训练需要使用代金券，领取方式见下文。注意代金券数量有限，先到先得，代金券金额有限，请节约使用，并及时关注余额（余额更新有延迟，发现低于100元就要及时申请代金券），避免欠费。操作方式如下。

1.1 代金券申请

首先登陆华为云，链接：<https://auth.huaweicloud.com/authui/login.html?locale=zh-cn&service=https%3A%2F%2Fwww.huaweicloud.com%2F#/login>，如果已经有华为云账号可直接登陆，如果没有需要先注册账号，然后实名认证。注册完华为云账号之后，需要进行全局配置，操作如下图：



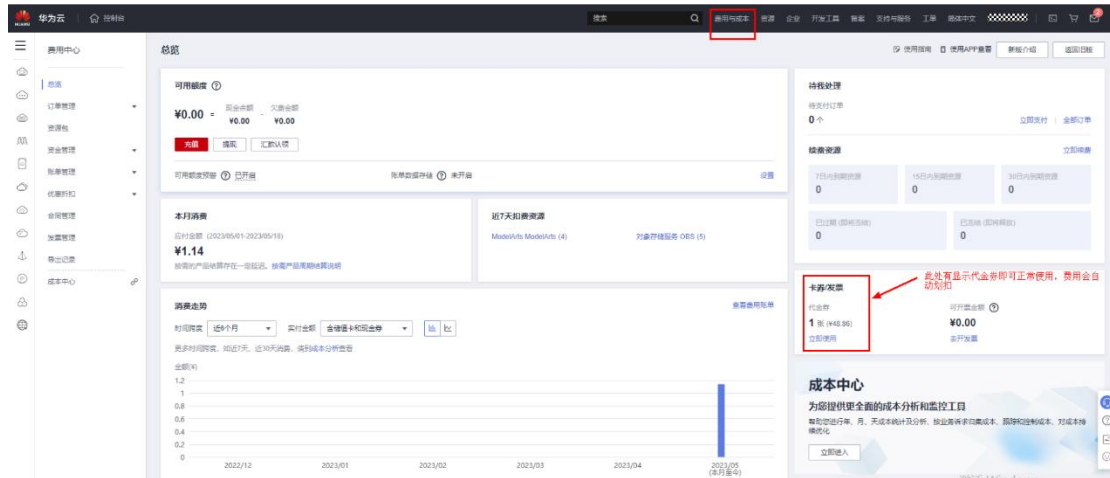
配置完成以后不要做其他操作（额外操作可能会收取费用导致账号欠费，需手动充值），去领取华为云代金券，注意代金券金额有限请谨慎使用。代金券领取链接详见比赛的各赛题官网页面，进入链接以后按照要求填写相关信息，提交申请。

1.2 代金券发放

审核标准：（1）选手需报名参加对应赛题；（2）申请选手需为队伍队长；

代金券到账时会进行短信提醒，同时可通过此链接查看代金券是否到账：<https://account.huaweicloud.com/usercenter/?agencyId=0e15c42d26c14cef994ead1af42648f9®ion=cn-north-4&locale=zh-cn#/userindex/allview>

打开界面如下图所示：



【特别提醒】

请参赛团队及时关注代金券额度，如发现额度较少，请先停止训练、删除服务。

1、由于比赛会用到昇腾算力、OBS存储等，会产生少量费用，因此在进行比赛操作前务必领取代金券，按照操作手册操作，以免账号欠费。代金券仅能在激活的账户上使用，参赛队员可与各自团队队长详细沟通代金券激活账户信息。

2、领取代金券资源后，请仔细了解代金券涵盖的资源类型，对于不包含的资源类型，或超出资源规格将会产生费用；

3、代金券到期后，如需继续使用相关服务，将产生相应费用。请在比赛结束后，及时删除不需要的项目，防止因资源到期产生不必要的扣费。释放资源请点击链接了解详情：

https://support.huaweicloud.com/usermanual-billing/renewals_topic_70000001.html

4、训练完成后，注意观察ModelArts首页是否还有计费中服务，并及时进行关闭；

5、您创建大赛所需资源时会优先扣除已领取的按需代金券，超出部分以按需付费的方式进行结算。如果您使用了其他类型规格的资源或其他云服务，将会产生费用。

2 华为云环境使用说明

2.1 注册镜像

赛题二模型微调和赛题三推理调优需选择指定镜像来进行开发。镜像需要注册后使用，操

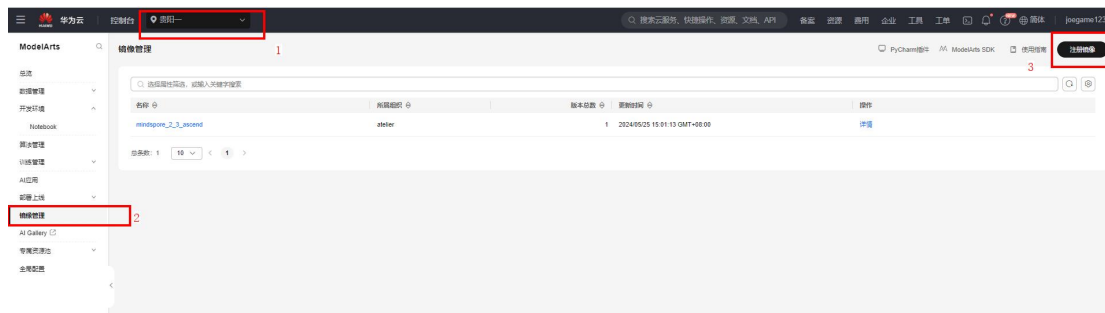
作流程如下：

2.1.1 进入 ModelArts 控制台

控制台链接：<https://console.huaweicloud.com/modelarts/?agencyId=0e15c42d26c14cef994ead1af42648f9®ion=cn-southwest-2&locale=zh-cn#/dev-container>，进入链接之后就会出现登录界面，如下图所示：

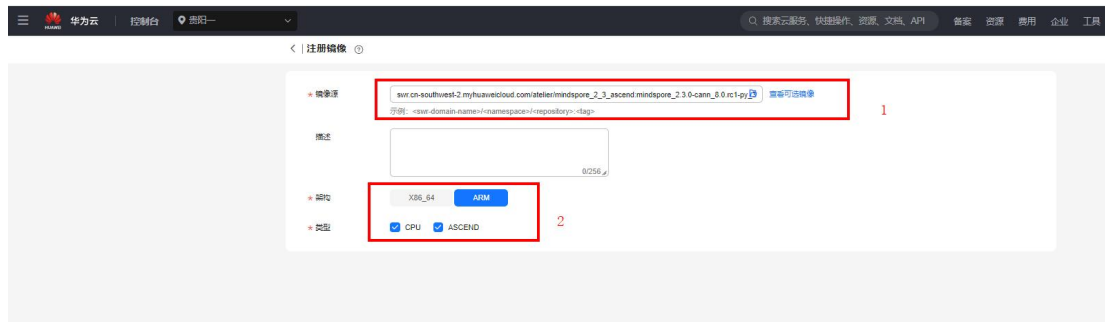


按照提示登录账号，进入ModelArts控制台如下图所示：



2.1.2 注册镜像

在上图1处，必须选择“贵阳一”节点，然后依次点击图中2“镜像管理”，图中3“注册镜像”，之后就会出现如下图所示界面：



上图1处需要填入镜像的SWR地址：swr.cn-southwest-2.myhuaweicloud.com/atelier/mindspore_2_3_ascend:mindspore_2.3.0-cann_8.0.rc1-py_3.9-euler_2.10.7-aarch64-snt9b-20240525100222-259922e；图中2处按截图“架构”和“类型”分别选择“ARM”和“CPU ASCEND”，然后点击界面右下角“立即注册”即可。

注意：

赛题二模型微调 and 赛题三模型推理都会用到这个镜像，同时需要额外安装指定的依赖（如 MindSpore、MindFormers等），详细操作请见对应赛题的指导；

镜像注册过之后就无需注册了，否则会出现如下图所示的错误：

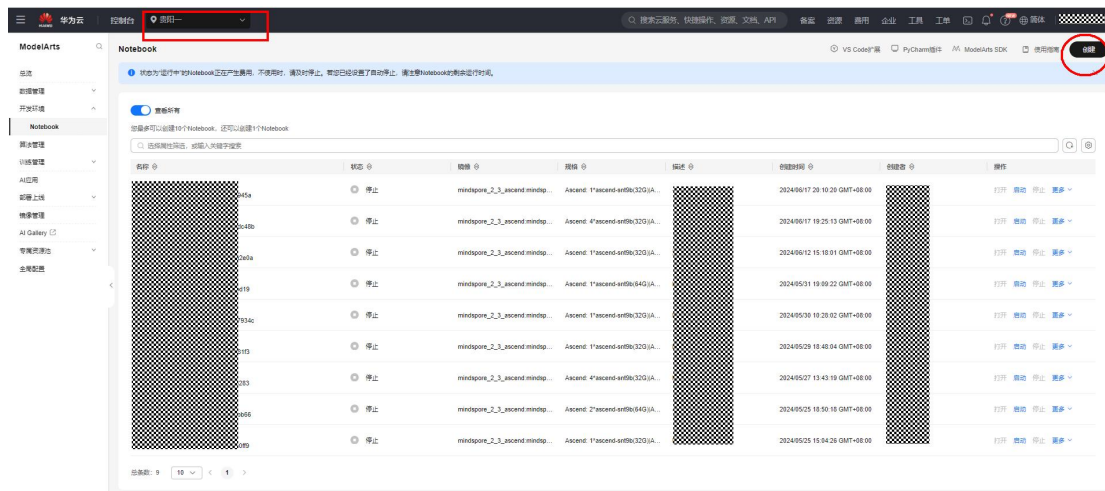


2.2 Notebook 环境

赛题一，二，三都可在华为云ModelArts的开发环境Notebook里面完成，进入该环境的操作如下所示。

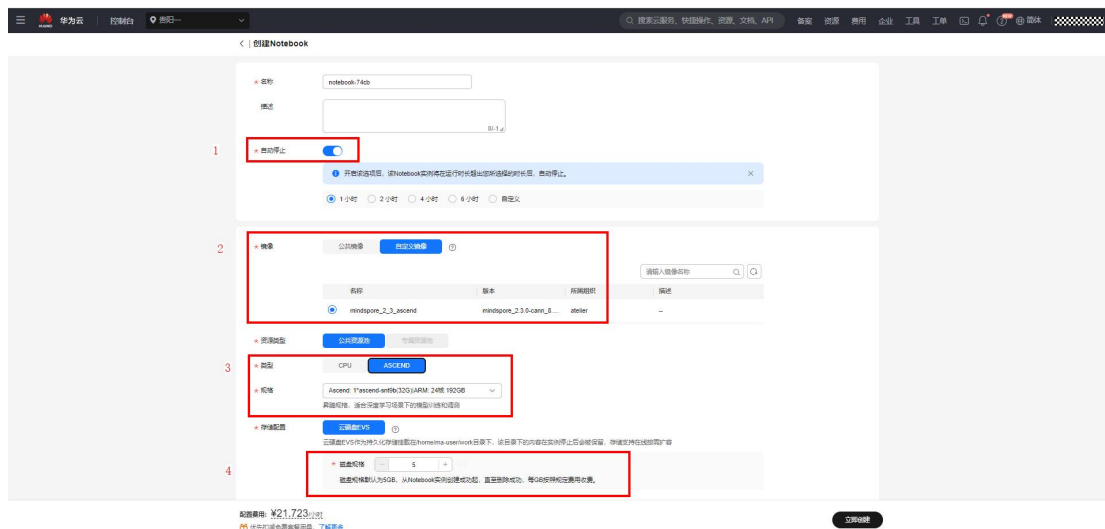
2.2.1 进入 ModelArts 控制台

按照1.2.1中“1. 进入ModelArts控制台”进入控制台，检查站点是否选择为“西南. 贵阳一”，然后选择“开发环境-Notebook”进入如下Notebook界面：



2.2.2 创建 Notebook 环境

点击上图右上角“创建”可新创建Notebook环境，会出现如下截图：



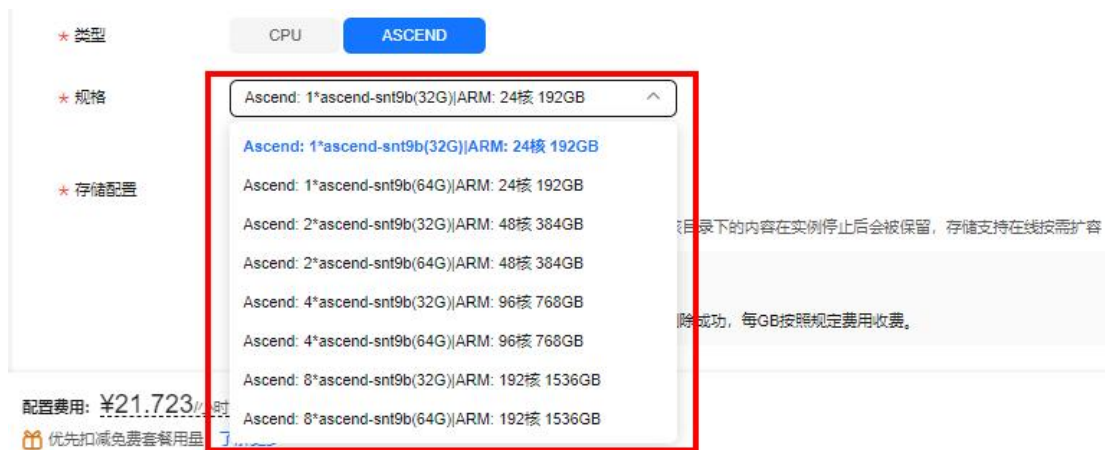
说明：

图中的1处：为了节省华为云代金券的使用，这里强烈建议打开“自动停止”。这个停止的时间在进入Notebook环境后也可自行设置，下文出现对应界面会进行说明；

图中的2处：镜像这里选择“自定义镜像”，就会看到1.2.1注册的自定义镜像；

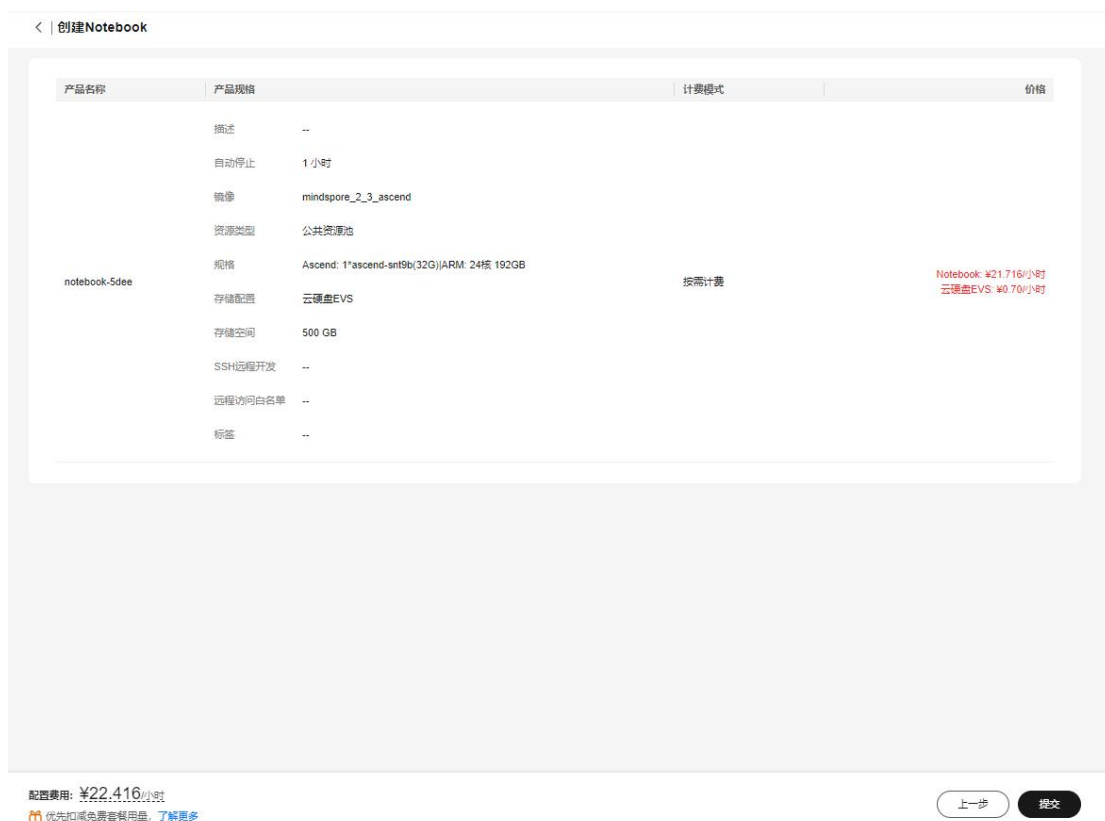
图中的4处：“磁盘规格”赛题二建议选择500G，赛题三可选择300G；

图中的3处：“类型”选择“Ascend”，“规格”点开可看到有8种选择，如下截图所示：



不同赛题的不同任务可有不同的选择，请选手选择32G显存的单卡或多卡资源，赛题二最低配置为“Ascend:4*ascend-snt9b(32G)ARM:96核768GB”，赛题三最低配置为第一个“Ascend:1*ascend-snt9b(32G)ARM:24核 192GB”，其他任务的最低配置会在后续描述中给出。不同规格对应的价格也有不同，选手可根据代金券使用情况酌情选择。

配置完成后点击“立即创建”，就会进入如下界面：

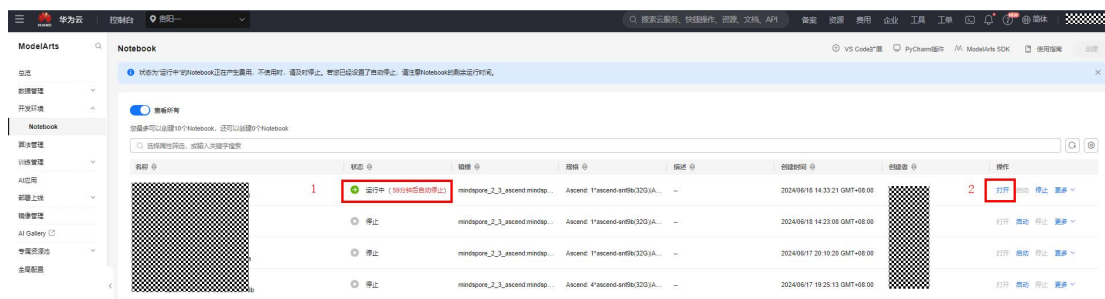


然后点击右下角的“提交”，出现下图：

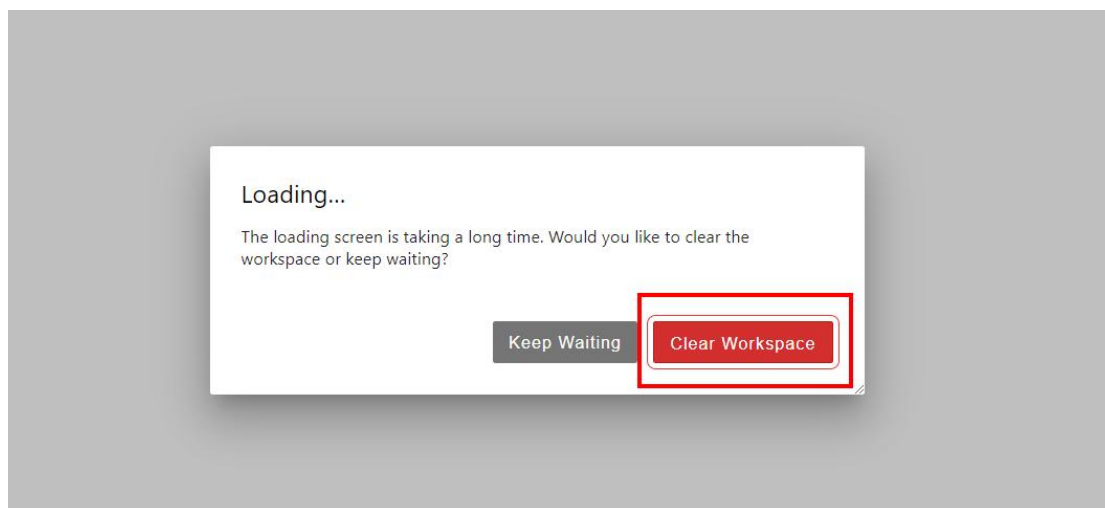


2.2.3 进入 Notebook 环境

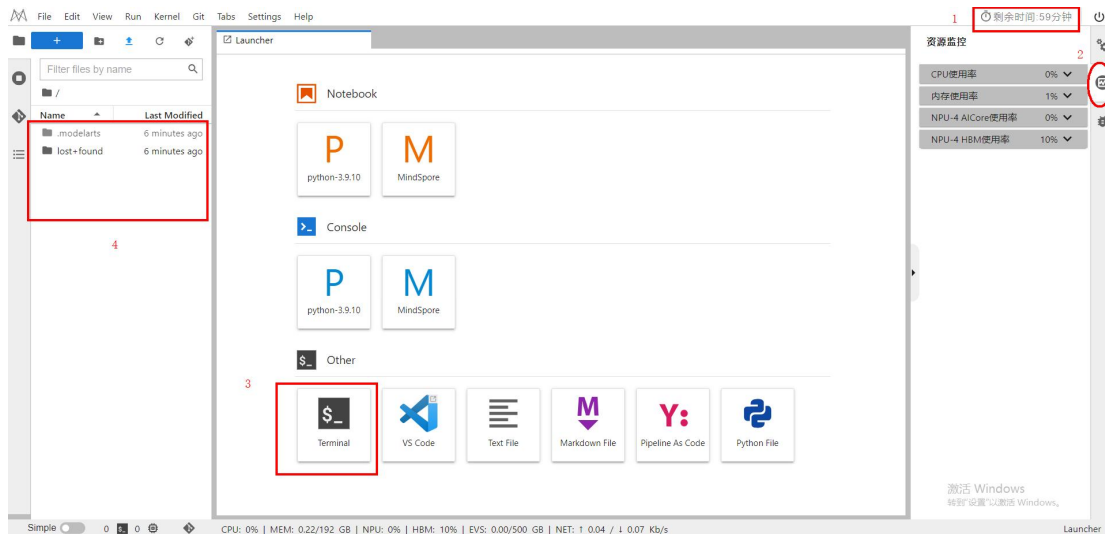
点击“立即返回”之后就会进入下面界面：



等待2分钟左右时间就会出现上图1处的“运行中”，然后点击图中2出的打开，等待1分钟左右时间，如果出现如下界面：



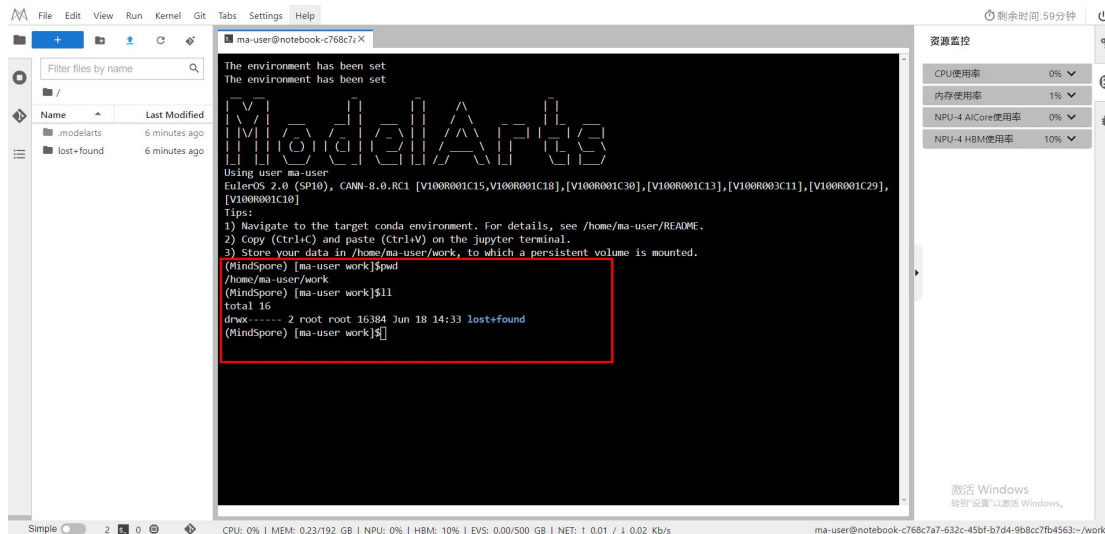
点击“Clear Workspace”，就会进入如下界面：



点击上图1处“剩余时间: XX”就可以手动修改自动停止Notebook的时间，在运行过程中可随时修改；

点击上图中2处可查看CPU和NPU的内存使用情况；

点击上图中3处可进入终端，如下图所示：



进入终端模型的虚拟环境是“MindSpore”，此为默认虚拟环境，必须使用这个。默认的目录位置是/home/ma-user/work，与截图左侧文件栏（上上张图中的4处）所在的目录位置一致。然后就可以在终端完成下面的赛题了。

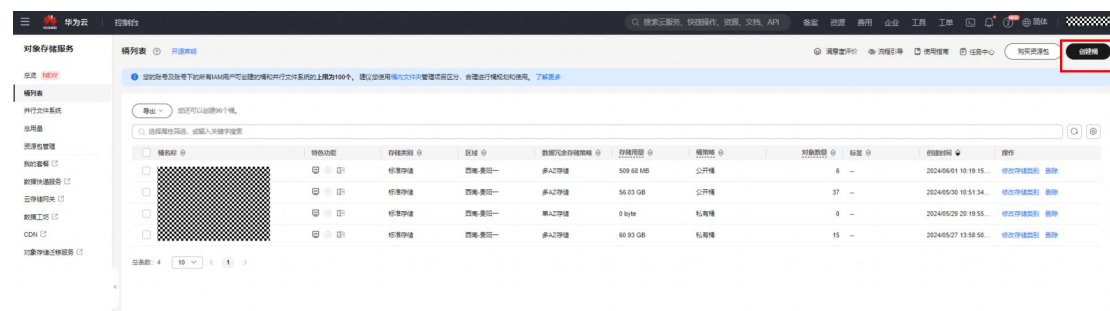
此外，华为云官方也提供了开发环境介绍，可参考链接：https://support.huaweicloud.com/devtool-modelarts/devtool-modelarts_0001.html；具体Notebook的使用可参考链接：https://support.huaweicloud.com/devtool-modelarts/devtool-modelarts_0004.html；有兴趣的开发者可以去浏览学习。

3 obs 数据传输指南

赛题二模型微调及赛题三推理调优的依赖包，数据集等将存储在华为云的obs桶里面，获取链接（URL）在比赛官网对应赛题的赛事详情页面，以及本指导书的各个赛题详细指导中展示，大家可以在Notebook终端用wget+URL命令进行文件下载。

此外，赛题二模型微调及赛题三推理调优在作品提交环节，会涉及较大文件的提交（如代码文件，保存的模型输出等），同样可以通过将文件上传obs桶，然后在作品提交报告中提供obs下载链接（URL）的方式完成提交，上传及获取URL的指南如下所示。

华为云OBS桶链接：<https://console.huaweicloud.com/console/?agencyId=0e15c42d26c14cef994ead1af42648f9®ion=cn-southwest-2&locale=zh-cn#/obs/manager/buckets>，点击链接之后跟Notebook环境一样登录账号，然后进入到如下界面：



点击右上角的“创建桶”，会出现如下画面：

< | 创建桶

复制桶配置 选择详情

该项可选。选择后可复制源桶的以下配置信息：区域 / 数据冗余策略 / 存储类别 / 桶策略 / 服务端加密 / 归档数据直读 / 企业项目 / 标签。

区域 西南-贵阳一

不同区域的资源之间内网互不相通，请选择靠近业务的区域，可以降低网络时延，提高访问速度。桶创建成功后不支持变更区域，请谨慎选择。 [如何选择区域](#) ①

桶名称 [查看命名规则](#) ①

① 不能和本用户已有桶重名 ② 不能和其他用户已有的桶重名 ③ 创建成功后不支持修改

数据冗余存储策略 多AZ存储 单AZ存储 ①

数据在同区域的多个AZ中存储，可用性更高。

⚠ 启用后不支持修改。多AZ存储采用相对较高的计费标准。 [价格详情](#)

默认存储类别 标准存储 低频访问存储 归档存储

标准存储：适合高性能、高可靠、高可用、频繁访问场景
低频访问存储：适合高可靠、低成本、较少访问场景
归档存储：适合长期存储，平均一年访问一次

创建桶时选择的存储类别会作为上传对象的默认存储类别。 [了解存储类别差异](#) ①

桶策略 私有 公共读 公共读写 复制桶策略 ①

任何用户都可以对桶内对象进行读操作，仅桶所有者可以进行写操作。

归档数据直读 开启 关闭 ①

关闭归档直读，归档存储类别的数据要先恢复才能访问。归档存储数据恢复和访问会收取相应的费用。 [价格详情](#)

服务端加密 SSE-KMS SSE-OBS 不开启加密 ①

开启服务端加密后，上传到当前桶的对象会被加密。您也可以在桶创建完成之后在桶概览页面调整服务端加密配置。

⚠ 建议开启加密，核心数据更安全，如果您使用KMS加密模式，超过免费配额会收取相应费用。 [价格详情](#)

创建阶段 使用阶段

OBS桶：创建免费 按需/资源包计费 [OBS计费说明](#)

立即创建

注意：

上图的区域需要选择“西南-贵阳一”，就是跟创建notebook的区域选择一样的；桶策略需要选择“公共读”，不然里面的数据别人下载不了，桶的大小不用设置，桶是自动扩容的。

obs桶存储详细操作，可参考如下说明：

在Notebook中上传下载OBS文操作件参考链接：https://support.huaweicloud.com/modelarts-faq/modelarts_05_0024.html

一些常见的问题处理方法参考链接：

https://support.huaweicloud.com/modelarts-faq/modelarts_05_0067.html

也可使用obsutil工具将本地的文件上传到obs桶，参考链接：

https://support.huaweicloud.com/util-tg-obs/obs_11_0001.html

4 赛题一：模型迁移指导

本赛题鼓励开发者基于昇思MindSpore、昇腾AI云服务开发模型，并丰富国内模型生态。

模型复现：选手需进行Configuration, Tokenizer, Model, Unit tests的复现

本地门禁自验：优先在自己的linux系统CPU下基于下方提供的门禁脚本（下称CI文件，获取链接详见下方-本地门禁自验）完成自验；必须在确保CI文件中的测试均通过后，再将迁移代码提交pr至MindNLP代码仓；

代码提交：提交PR时需附上自验通过的截图，并评论/model name触发MindNLP仓的CI测试；

结果检查：MindNLP仓的CI测试结果请自行查看，通过则视为有效作品，如通过，需在评论区回复通过链接，否则不进行代码合入，如未通过，请自行修改。

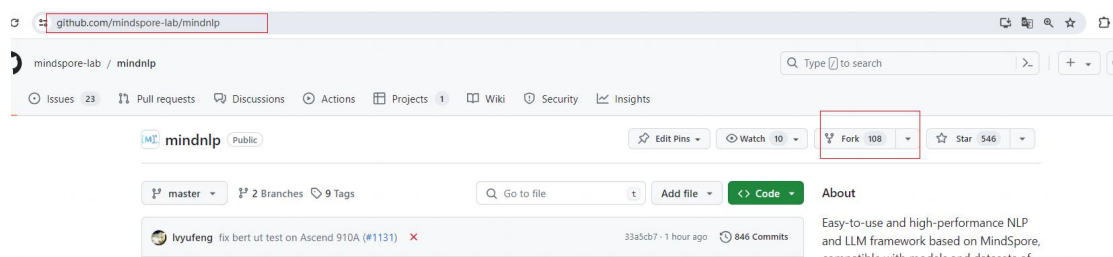
作品提交：和审核老师确认代码合入后，请在官网页面补充提交的PR链接及队伍信息，否则无法定位到获奖选手。

注意：

1. 模型完成复现后，需要在auto路径下注册模型。
2. CI文件严禁修改，通过修改CI伪造完成的，一经发现即刻取消资格。
3. 迁移Unit test（下称UT）测试时，禁止跳过测试精度的UT，即带slow的测试，否则视为未完成复现，本地如何进行slow的UT自验请参考下方-本地门禁自验。
4. CI要求Pylint语法检测必须通过，本地Pylint自验请参考下方-本地门禁自验。

4.1 模型复现

1. fork mindnlp的代码仓 <https://github.com/mindspore-lab/mindnlp>



Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).

Owner *

Repository name *

Choose an owner ▼

/ mindnlp

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Easy-to-use and high-performance NLP and LLM framework based on MindSpore, compatible with models and

☒ Copy the `master` branch only

Contribute back to mindsore-lab/mindnlp by adding your own branch. [Learn more.](#)

Create fork

2. 在个人仓库中找到刚才fork的mindnlp代码仓,并且 `git clone **mindnlp代码仓地址*`

```
MINGW64 ~/Desktop
$ git clone https://github.com/mindsore-lab/mindnlp.git
Cloning into 'mindnlp'...
remote: Enumerating objects: 14703, done.
remote: Counting objects: 100% (3523/3523), done.
remote: Compressing objects: 100% (1547/1547), done.
remote: Total 14703 (delta 2289), reused 2355 (delta 1946), pack-reused 11180
Receiving objects: 100% (14703/14703), 19.18 MiB | 4.94 MiB/s, done.
Resolving deltas: 100% (10093/10093), done.
Updating files: 100% (1371/1371), done.
```

3. 根据迁移指南完成模型迁移

Hugging Face大模型迁移至MindNLP有可参考的PDF文档和视频, 链接如下:

PDF文档链接: <https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic1/Huggingface%20Transformers%20to%20mindnlp.pptx>

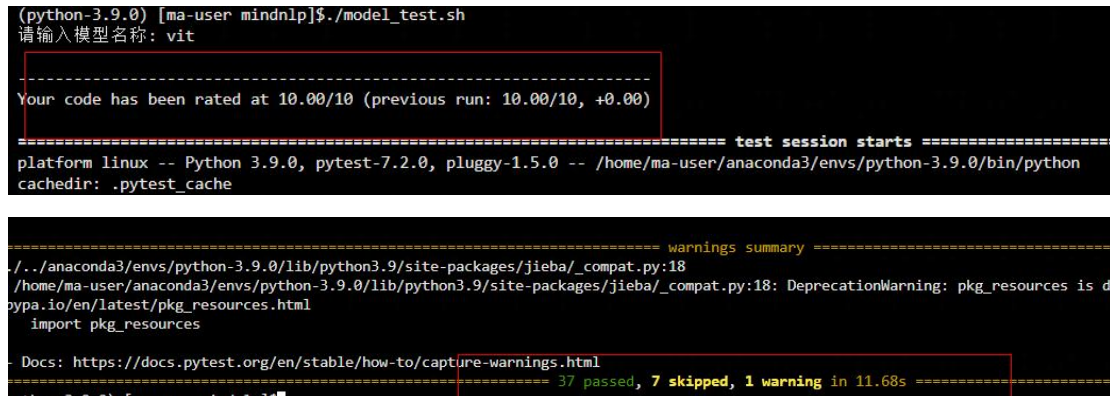
视频链接: <https://www.bilibili.com/video/BV1iC4y197hb/>

4.2 本地门禁自验

门禁脚本获取链接: https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic1/model_test.sh, 操作流程如下:

1. 上传门禁检查脚本model_test.sh到mindnlp同级目录下

2. 执行脚本 `./model_test.sh`，根据错误提示修改对应语法以及用例报错信息，确保没有语法错误并且所有测试用例执行通过



```
(python-3.9.0) [ma-user mindnlp]$ ./model_test.sh
请输入模型名称: vit

-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
-----
===== test session starts =====
platform linux -- Python 3.9.0, pytest-7.2.0, pluggy-1.5.0 -- /home/ma-user/anaconda3/envs/python-3.9.0/bin/python
cachedir: .pytest_cache

----- warnings summary -----
../anaconda3/envs/python-3.9.0/lib/python3.9/site-packages/jieba/_compat.py:18
/home/ma-user/anaconda3/envs/python-3.9.0/lib/python3.9/site-packages/jieba/_compat.py:18: DeprecationWarning: pkg_resources is deprecated as an API. See https://pypi.io/en/latest/pkg_resources.html
import pkg_resources

Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 37 passed, 7 skipped, 1 warning in 11.68s =====
(python-3.9.0) [ma-user mindnlp]$
```

3. 带slow的UT为精度测试，不允许跳过，本地跑UT自验时需先配置以下环境变量

```
export RUN_SLOW=1
```

```
pytest -vs tests/ut/transformers/models/name
```

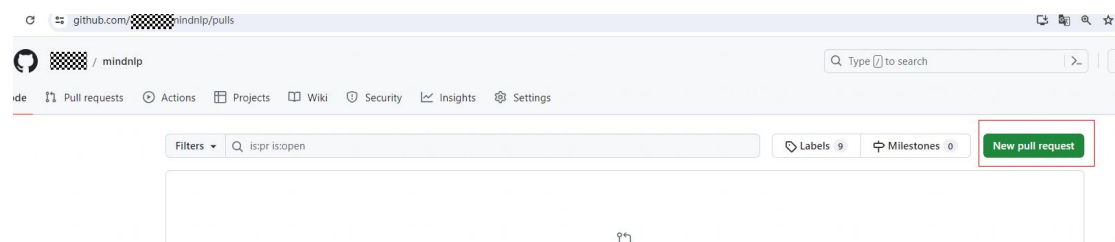
4. 本地Pylint自验方法:

```
cd mindnlp
```

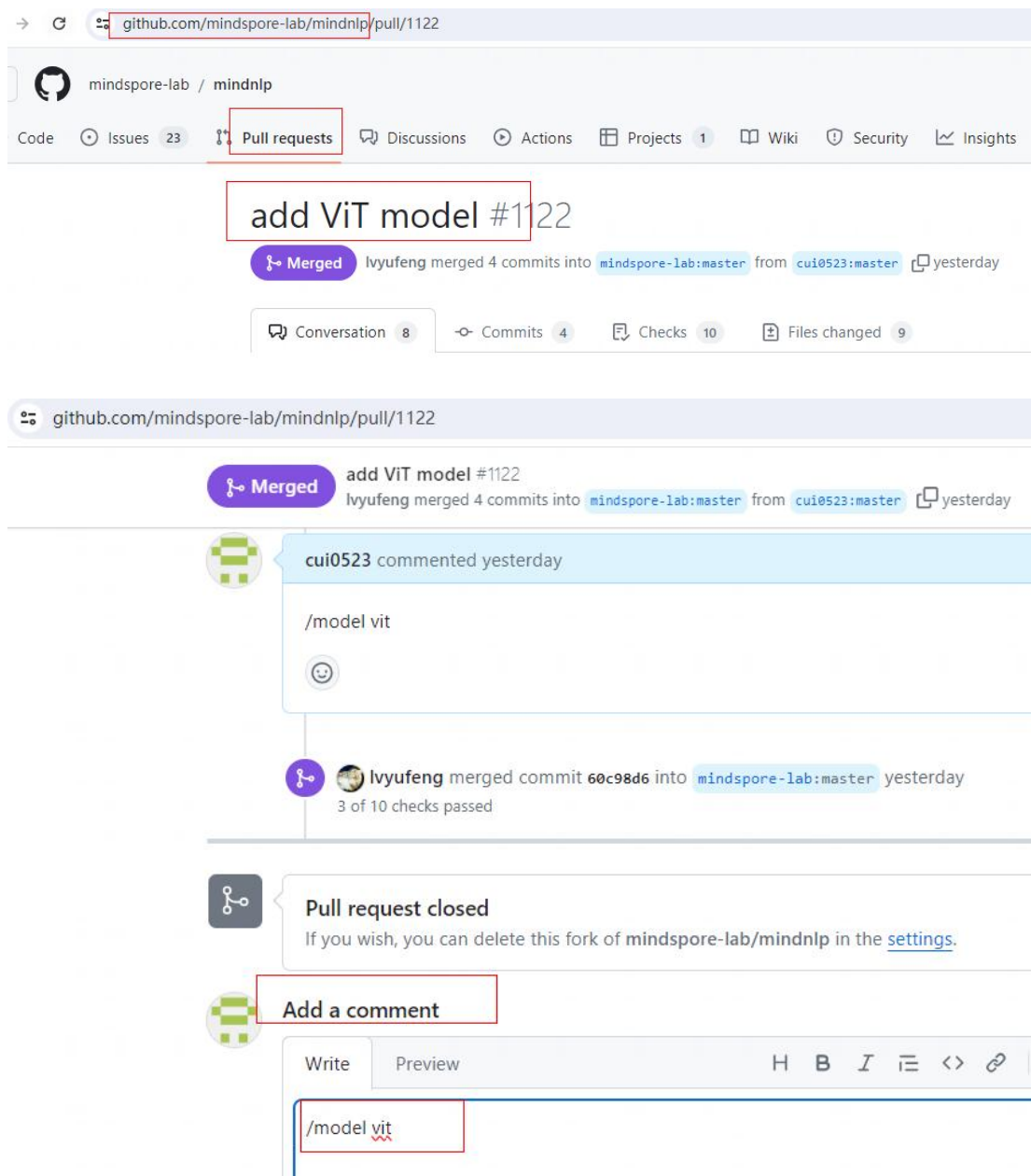
```
bash scripts/pylint_check.sh
```

5. 代码提交

如下图所示提交代码，并且create pull request

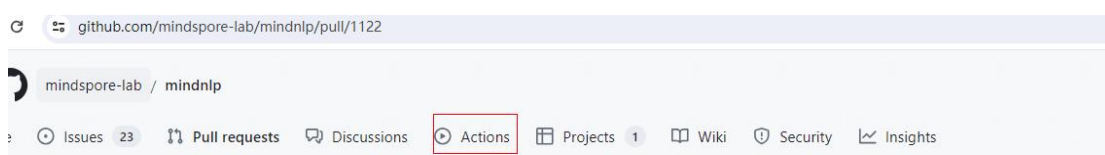


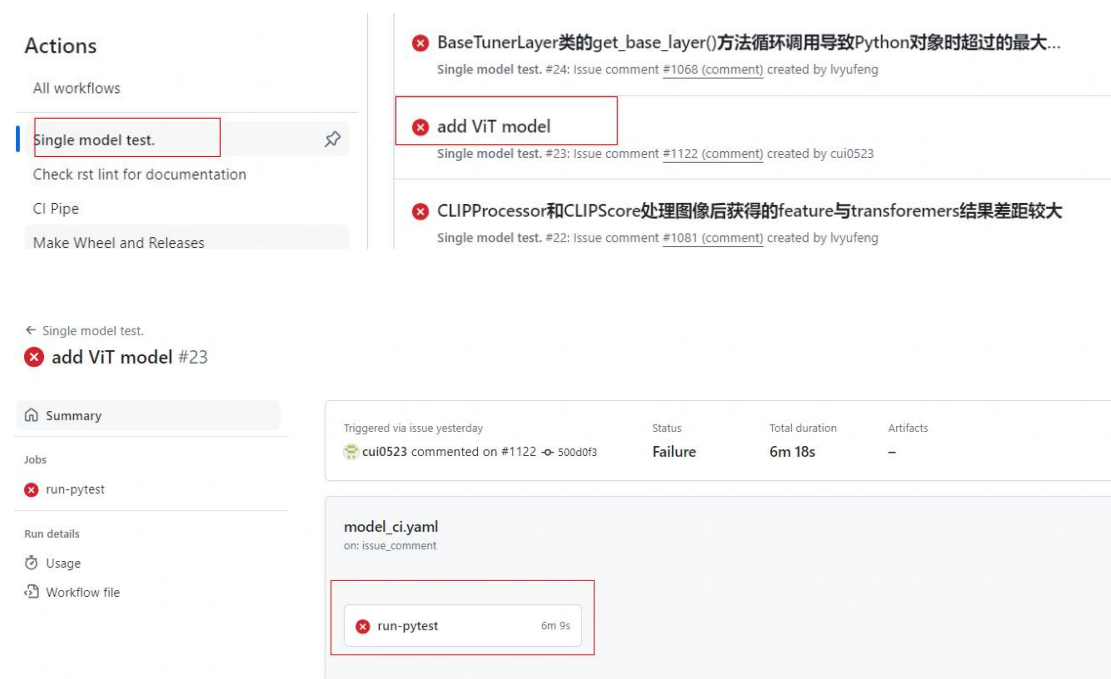
6. 在mindnlp主代码仓“pull requests”中找到刚才提交的代码，并在“Add a comment”中写入“/model 模型名称”（eg. /model vit）



4.3 结果检查

在mindnlp主代码仓“Actions”-----> “Single model test”中找到刚才提交的代码，查看“run-pytest”是否执行成功。根据错误提示修改对应语法以及用例报错信息，修改完成后重复步骤d，直至run-pytest 执行成功。





The screenshot displays the GitHub Actions interface for a workflow named "Single model test.". On the left, a sidebar lists actions: "All workflows", "Single model test." (highlighted), "Check rst lint for documentation", "CI Pipe", and "Make Wheel and Releases". The main area shows a list of failed jobs with red 'x' icons. The first job is "add ViT model", which failed due to a "BaseTunerLayer类的get_base_layer()方法循环调用导致Python对象时超过的最大..." error. The second job is "CLIPProcessor和CLIPScore处理图像后获得的feature与transformers结果差距较大", which failed due to a "Single model test. #22: Issue comment #1081 (comment) created by lvyufeng" error. Below the job list, a summary section for "add ViT model #23" is shown, including a table with the following data:

Triggered via issue yesterday	Status	Total duration	Artifacts
cui0523 commented on #1122 -> 500d0f3	Failure	6m 18s	-

Below the table, a "model_ci.yaml" file is shown, triggered on "issue_comment". A job named "run-pytest" is listed with a duration of "6m 9s".

联系相关工作人员merge代码。

4.4 作品提交

提交作品前，选手需完成赛题报名，然后在赛题页面的banner点击提交作品的按钮，进行提交。



The screenshot shows a competition banner for the "【推理调优赛题】昇思MindSpore&昇腾AI云服务大模型开发挑战赛". The banner includes the following information:

- 标题:** 【推理调优赛题】昇思MindSpore&昇腾AI云服务大模型开发挑战赛
- 奖金:** ¥ 460,000
- 主办方:** 华为技术有限公司
- 提交作品按钮:** 提交作品 (highlighted with a red box)
- 作品提交截止时间:** 2024/07/31 23:59

将Word文档压缩成Zip文件上传提交（文件命名规则：团队名称.zip），参赛者可多次提交，最新一次提交将覆盖上一次提交作品，以最后提交的作品为准。提交Word示例如下：

团队名: 姓名: 

以下为提交且已经合入的 PR 链接:

x_clip: <https://github.com/mindspore-lab/mindnlp/pull/1135>mobilevit: <https://github.com/mindspore-lab/mindnlp/pull/1138>owlvit: <https://github.com/mindspore-lab/mindnlp/pull/1142>imagegpt: <https://github.com/mindspore-lab/mindnlp/pull/1149>poolformer: <https://github.com/mindspore-lab/mindnlp/pull/1158>

5 赛题二：模型微调指导

5.1 赛题介绍

本赛题要求基于开源中英文混合数学运算数据集，跑通baseline，并对MindFormers中Llama3-8b模型进行微调（LoRA或其他微调算法）。微调后的模型在原有能力不丢失的前提下（需保持在原能力的90%及以上），回答数学运算准确率相对baseline有所提升，按照低参比例及准确率进行综合排名。

模型原有能力以其在SQUAD数据集上的阅读理解能力为准，评价标准为F1 Score和Em Score，要求微调后两项评价指标需要在给定阈值以上方可算作有效作品。具体如何进行原有能力评估，以及F1 Score和Em Score的参考阈值，请参考下方1.5.8微调后模型原有能力评估。

数学运算准确率评价标准：模型基于测试数据集（不公开，与训练数据集格式相同，为数道中英文数学运算题）进行推理，生成数学运算结果。如计算结果（数值）与正确答案相同，则视为本题正确，最终统计在测试数据集上回答正确的题目数量占比。

$$\text{运算准确率} = \text{正确运算题目数} / \text{测试集总题目数}$$

注：baseline的数学运算准确率为20%，请以此为参考进行微调。

低参比例：低参比例为微调参数量在总参数量的占比，选手在提交作品时需提供低参比例的计算结果，低参比例运算公式如下。

低参比例 = 参与微调的参数数量 / 模型总参数量

低参比例和运算准确率综合排名：低参比例越低越好，数学运算准确率越高越好，最终按照如下加权进行运算。

$(100\% - \text{低参比例}) * 0.3 + \text{运算准确率} * 0.7$

本题目共提供80万条中英文混合题目作为训练数据集，选手可根据自己的实际情况调整数据集规模，建议综合微调、推理时长、算力需求，维持模型原有能力及模型运算准确率提升等多方面因素进行训练数据集规模的评估。

参考：9万条数据集在4卡的LoRA微调（微调参数量大概3million）下的运行时长为6个小时（seq_len为256，batch_size为64，epoch为5）。

本赛题基础流程共分为8个环节：环境配置、模型权重和tokenizer文件准备、数据集准备、修改配置文件并启动微调、微调参数比例计算、微调后多卡的模型权重保存与合并、微调后模型原有能力评估、微调后模型数学计算结果推理，下方会针对每个环节进行完整说明。

5.2 环境配置

本赛题在默认基础环境下，即指定的华为云自定义镜像下，需按照要求额外安装指定的MindSpore和MindFormers依赖。

5.2.1 Notebook 环境创建

本赛题配置最低可使用华为云modelarts-开发环境-Notebook 4卡NPU（32G显存）环境运行，硬盘规格推荐使用500G，如下图所示设置：



★ 资源类型 **公共资源池** 专属资源池

★ 类型 **CPU** **ASCEND**

★ 规格 Ascend: 4*ascend-9t9b(32G)ARM: 96核 768GB

昇腾规格，适合深度学习场景下的模型训练和推理

★ 存储配置 **云硬盘EVS**

云硬盘EVS作为持久化存储挂载在/home/ma-user/work目录下，该目录下的内容在实例停止后会被保留，存储支持在线按需扩容

★ 磁盘规格 - 500 +

磁盘规格默认为50GB，从Notebook实例创建成功起，直至删除成功，每GB按照标准费用收费。

自定义镜像获取

请参考上述1.2.1 注册镜像章节进行操作。

5.2.2 MindSpore 安装

MindSpore可用如下命令安装：

```
pip install mindspore==2.3.0rc2
```

如果上面安装命令出现问题，可通过如下命令安装：

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic2-finetune/mindspore-2.3.0rc2-cp39-cp39-linux_aarch64.whl
```

```
pip install mindspore-2.3.0rc2-cp39-cp39-linux_aarch64.whl
```

5.2.3 MindFormers 安装

MindFormers包必须使用赛事组提供的，使用其他版本出现问题，选手自己负责。可使用以下命令下载安装MindFormers包：

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic2-finetune/mindformers.zip
```

```
unzip mindformers.zip
```

```
cd mindformers/
```

```
bash build.sh
```

5.2.4 环境变量及其他依赖

环境变量设置命令如下（环境变量中的绝对路径要与你本地文件的路径一致）：

```
export PYTHONPATH="${PYTHONPATH}:/home/ma-user/work/mindformers/"
```

安装其他依赖，代码如下所示：

```
pip install tiktoken
```

5.3 模型权重和 tokenizer 文件准备

为了比赛的公平公正，选手必须在比赛提供的权重文件和tokenizer文件的基础上进行微调，不可在其他的权重上进行微调，如有发现立刻取消比赛资格。

权重文件下载命令：

```
cd /home/ma-user/work/
```

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic2-finetune/llama3-8B.ckpt
```

tokenizer.model文件的下载命令:

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic2-finetune/tokenizer.model
```

5.4 数据集准备

本赛题数据集获取链接已同步更新至赛题官网页面，具体下载方式见本手册1.3 obs数据传输指南。本赛题提供的数据集为模型微调数据集，同样的数据来源，面向不同数据预处理阶段共保存了三个版本以供参考，具体如下文描述。

5.4.1 原始数据集

该数据集为最原始的数据集，只有问题和答案的数据对，选手可在此数据集上自行选择数据前处理操作进行模型微调。数据集下载命令如下：

```
cd /home/ma-user/work/
```

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic2-finetune/train.json
```

5.4.2 添加 prompt 模板后的数据集

该数据集是参考MindFormers官网的数据前处理，使用fastchat工具添加了prompts模板，这里只是给出预处理的参考，选手可自行发挥对数据进行适当预处理。这里提供两种获取预处理数据的方法，如下所示：

第一种，选手可以直接通过链接下载使用，数据集下载命令如下：

```
cd /home/ma-user/work/
```

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic2-finetune/train-data-conversation.json
```

第二种，选手可通过代码生成预处理的数据。MindFormers官网的数据前处理介绍链接：<https://gitee.com/mindspore/mindformers/blob/r1.1.0/research/llama3/llama3.md#%E>

6%95%B0%E6%8D%AE%E9%9B%86%E5%87%86%E5%A4%87，如下图中的“数据集准备”下面的“step 1”：

- Llama 3
 - 模型描述
 - 仓库介绍
 - 前期准备
 - 环境要求
 - 数据集准备
 - 模型权重准备
 - 模型权重转换
- Llama3-8B
 - 全参微调
 - MindSpore推理
 - 基于高阶接口推...

数据集准备

目前提供alpaca数据集的预处理脚本用于全参微调任务。

数据集下载链接如下：

- [alpaca_data](#)

alpaca数据集原始格式样例：

```
# alpaca examples:
{
  "instruction": "Describe a time when you had to make a difficult decision.",
  "input": "",
  "output": "I had to make a difficult decision when I was working as a project manager at a construction company. I w
},
{
  "instruction": "Identify the odd one out.",
  "input": "Twitter, Instagram, Telegram",
  "output": "Telegram"
},
}
```

- step 1** 执行 `alpaca_converter.py`，使用fastchat工具添加prompts模板，将原始数据集转换为多轮对话格式。

```
# 脚本路径: tools/dataset_preprocess/llama/alpaca_converter.py
# 执行转换脚本
python alpaca_converter.py \
--data_path {path}/alpaca_data.json \
--output_path {path}/alpaca-data-conversation.json
```

参数说明
data_path: 存放alpaca数据的路径
output_path: 输出转换后对话格式的数据路径

此处提供的数据集是参考上图alpaca_converter.py文件的代码根据train.json数据的特点做了一些代码调整生成出来的，修改后的代码下载命令如下：

```
cd /home/ma-user/work/
```

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic2-finetune/data_converter.py
```

在终端运行如下命令（路径可根据自己数据集实际路径修改）进行数据生成：

```
cd /home/ma-user/work/
```

```
python data_converter.py --data_path /home/ma-user/work/train.json --output_path /home/ma-user/work/train-data-conversation.json
```

5.4.3 MindRecord 格式数据集

该数据集是参考MindFormers官网生成MindRecord格式数据的指引生成的，教程见下文。这里也提供选手可直接用来运行模型微调的转换完成的数据集，下载并解压命令如下：

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic2-finetune/train-fastchat256-mindrecord.zip
```

```
unzip train-fastchat256-mindrecord.zip
```

生成MindRecord格式的数据教程见链接<https://gitee.com/mindspore/mindformers/blob/r1.1.0/research/llama3/llama3.md#%E6%95%B0%E6%8D%AE%E9%9B%86%E5%87%86%E5%A4%87>中的“数据集准备”下的step 2，选手可根据自己数据实际的本地路径修改数据集路径和tokenizer文件路径，需要修改内容如下截图：

- step 2. 执行 llama_preprocess.py，进行数据预处理、Mindrecord数据生成，将带有prompt模板的数据转换为mindrecord格式。

```
# 脚本路径: research/llama3/llama_preprocess.py
# 由于此工具依赖fschat工具包解析prompt模板，请提前安装fschat >= 0.2.13 python = 3.9
python llama_preprocess.py \
--dataset_type qa \
--input_glob /{path}/alpaca-data-conversation.json \
--model_file /{path}/tokenizer.model \
--seq_length 8192 \
--output_file /{path}/alpaca-fastchat8192.mindrecord
```

其中seq_length参数需要跟模型微调的时候一致，此超参数选手可适当调整，上文提供的mindrecord格式数据是设置seq_length=256运行得到的。此处也提供直接转换数据的运行命令，如下：

```
cd /home/ma-user/work/mindformers/research/llama3

python llama_preprocess.py \
--dataset_type qa \
--input_glob /home/ma-user/work/train-data-conversation.json \
--model_file /home/ma-user/work/tokenizer.model \
--seq_length 256 \
--output_file /home/ma-user/work/train-fastchat256.mindrecord
```

提醒：此处使用硬件规格为：Ascend:1*ascend-snt9b(32G)|ARM:24核192GB 的Notebook环境，数据集为9万条，seq_length设置为256，完成数据转换花费了50分钟左右的时间。

5.5 修改配置文件并启动微调

5.5.1 配置文件修改

本赛题提供可供选手直接运行微调的配置文件，下载命令如下：

```
cd /home/ma-user/work
```

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/2024-ascend-innovation-contest-mindspore/obs.cn-southwest-2.myhuaweicloud.com/2024-6-21
```

eicloud.com/topic2-finetune/run_llama3_8b_8k_800T_A2_64G_lora_dis_256.yaml

此配置文件可直接运行微调。修改的内容有主要是参考文件https://gitee.com/mindspore/mindformers/blob/dev/research/llama3/run_llama3_8b_8k_800T_A2_64G.yaml，具体内容如下：

(1) 增加pet_config配置，位置在model下的model_config下，具体内容如下图所示：

pet_config:

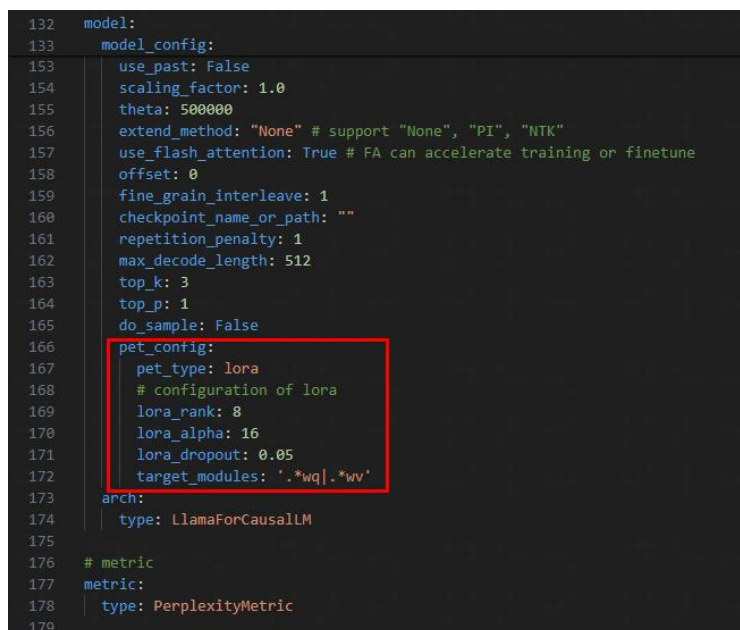
```
pet_type: lora

lora_rank: 8

lora_alpha: 16

lora_dropout: 0.05

target_modules: '.*wq|.*wv'
```



```
132 model:
133   model_config:
153     use_past: False
154     scaling_factor: 1.0
155     theta: 500000
156     extend_method: "None" # support "None", "PI", "NTK"
157     use_flash_attention: True # FA can accelerate training or finetune
158     offset: 0
159     fine_grain_interleave: 1
160     checkpoint_name_or_path: ""
161     repetition_penalty: 1
162     max_decode_length: 512
163     top_k: 3
164     top_p: 1
165     do_sample: False
166     pet_config:
167       pet_type: lora
168       # configuration of lora
169       lora_rank: 8
170       lora_alpha: 16
171       lora_dropout: 0.05
172       target_modules: '.*wq|.*wv'
173   arch:
174     type: LlamaForCausalLM
175
176 # metric
177 metric:
178   type: PerplexityMetric
179
```

(2) 其他需要修改的参数如下：

```
load_checkpoint: 'path/to/llama3_8b.ckpt'    # 填写权重路径

auto_trans_ckpt: False                      # 关闭自动权重转换

use_past: False                             # 关闭增量推理

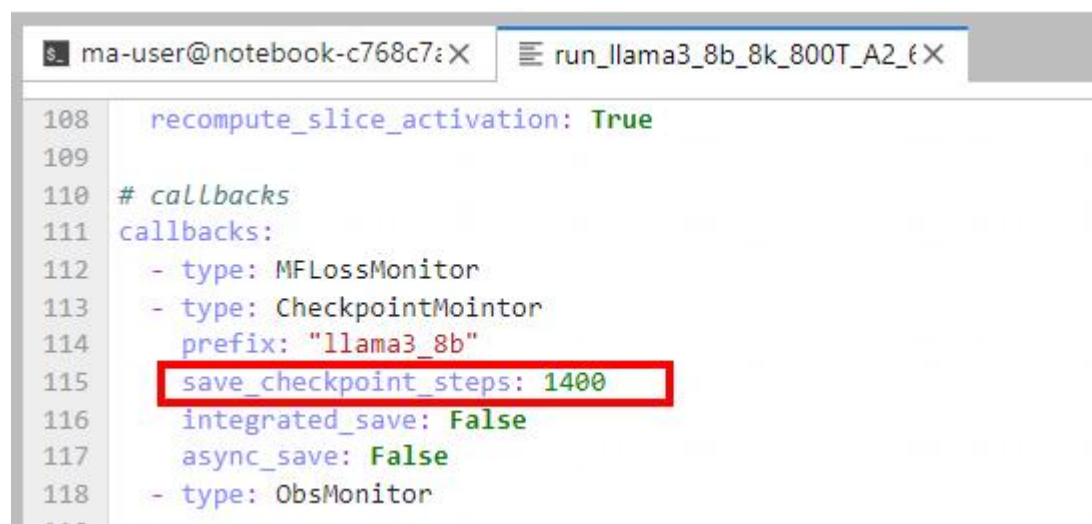
vocab_file: 'path/to/tokenizer.model'       # 配置词表路径

use_parallel: False                         # 关闭并行模式（单卡），多卡
需要设置True
```


only_save_strategy: True

max_device_memory: "26GB" # 这是选择显存为32G的NPU，如果选择64G的可设置为56G
为保证比赛公平公开，请选手基于比赛给定的预训练权重进行微调，并在微调过程中按照平均间隔保存5份权重（权重需为合并后的完整权重），比赛将以在测试数据集上效果最好的权重得分计为选手最终成绩。平均间隔：保存权重的时刻需均匀分布在微调过程中，假设微调了10个epoch，可以提供第2、4、6、8、10个epoch的权重；如微调了20个epoch，可以提供第4、8、12、16、20个epoch的权重，以此类推。

选择保存权重的超参数在配置文件 run_llama3_8b_8k_800T_A2_64G_lora_dis_256.yaml 的第115行的 save_checkpoint_steps，如下图所示：



```
108 recompute_slice_activation: True
109
110 # callbacks
111 callbacks:
112   - type: MFLossMonitor
113   - type: CheckpointMointor
114     prefix: "llama3_8b"
115   save_checkpoint_steps: 1400
116   integrated_save: False
117   async_save: False
118   - type: ObsMonitor
```

save_checkpoint_steps: 1400 表示每隔 1400 step 保存一个权重文件。

5.5.2 启动4卡微调

启动4卡微调任务，脚本如下（涉及到绝对路径的地方请选手检查自己文件的实际路径）：

```
cd /home/ma-user/work/mindformers/research/
bash ../scripts/msrun_launcher.sh \
"llama3/run_llama3.py \
--config /home/ma-user/work/run_llama3_8b_8k_800T_A2_64G_lora_dis_256.yaml \
--load_checkpoint /home/ma-user/work/llama3-8B.ckpt \
--auto_trans_ckpt False \
--use_parallel True \
```

```
--run_mode finetune \
```

```
--train_data /home/ma-user/work/train-fastchat256.mindrecord" 4
```

微调过程脚本会自动保存日志，日志路径在：/home/ma-user/work/mindformers/research/output/msrun_log/下，每张卡都有独立的日志记录文件。

5.5.3 启动 8 卡微调

如果需要使用八卡服务器启动微调任务，需要在新建Notebook的时候选择规格为“Ascend: 8*ascend-snt9b(32G)ARM: 192核 1536GB ”或者 “Ascend:8*ascend-snt9b(64G)ARM: 192核 1536GB”。教程baseline选择的是“Ascend:8*ascend-snt9b(32G)ARM: 192核 1536GB ”。

可下载另外一个可直接运行配置文件，下载命令：

```
cd /home/ma-user/work/
```

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic2-finetune/run_llama3_8b_8k_800T_A2_64G_lora_dis8_256.yaml
```

此配置文件是在 run_llama3_8b_8k_800T_A2_64G_lora_dis_256.yaml 的基础上做了一些修改，主要修改的内容有：

第93行中的参数 model_parallel 由4修改为8，如下图所示：

```
90 # default parallel of device num = 8 for Atlas 800T A2
91 parallel_config:
92   data_parallel: 1
93   model_parallel: 8
94   pipeline_stage: 1
95   use_seq_parallel: False
96   micro_batch_num: 1
97   vocab_emb_dp: True
98   gradient_aggregation_group: 4
```

第18行，47行，147行的batch_size参数修改为16；

第127行的 max_device_memory 需要修改26G；

将启动微调脚本最后的数字 4 修改为 8。

最终启动8卡微调的命令如下：

```
cd /home/ma-user/work/mindformers/research/
```

```
bash ../scripts/msrun_launcher.sh \
```

```
"llama3/run_llama3.py \  
--config /home/ma-user/work/run_llama3_8b_8k_800T_A2_64G_lora_dis8_256.yaml \  
--load_checkpoint /home/ma-user/work/llama3-8B.ckpt \  
--auto_trans_ckpt False \  
--use_parallel True \  
--run_mode finetune \  
--train_data /home/ma-user/work/train-fastchat256.mindrecord" 8  
日志文件目录同4卡微调。
```

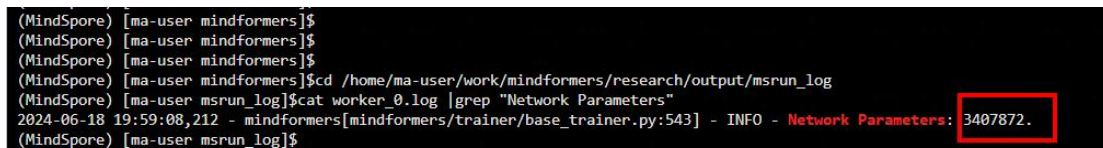
5.6 微调参数比例计算

模型微调参数的数量可在运行日志中获取，日志目录见1.5.5，如果感觉某张卡的日志文件信息不全，可查看其他卡的日志文件。

可通过如下命令在终端打印出微调参数的数量（以worker0为例）：

```
cd /home/ma-user/work/mindformers/research/output/msrun_log  
cat worker_0.log |grep "Network Parameters"
```

具体操作和显示结果如下截图：



```
(MindSpore) [ma-user mindformers]$  
(MindSpore) [ma-user mindformers]$  
(MindSpore) [ma-user mindformers]$  
(MindSpore) [ma-user mindformers]$cd /home/ma-user/work/mindformers/research/output/msrun_log  
(MindSpore) [ma-user msrun_log]$cat worker_0.log |grep "Network Parameters"  
2024-06-18 19:59:08,212 - mindformers[mindformers/trainer/base_trainer.py:543] - INFO - Network Parameters: 3407872.  
(MindSpore) [ma-user msrun_log]$
```

图中 3407872 即为微调的参数数量，用该数值除以llama3-8B的参数数量 8030000000（选手统一使用这个数值做分母） 即可获得微调参数比例。

5.7 微调后权重合并

微调完成之后权重会分别分布在四个rank文件夹内，此时需要将权重文件进行合并，权重合并教程参考链接：https://gitee.com/mindspore/mindformers/blob/r1.1.0/docs/feature_cards/Transform_Ckpt.md，具体的操作如下所示：

这里推荐使用“方案1：源码执行”方式进行权重合并，具体如下：

5.7.1 获取分布式策略文件

在你微调使用的yaml文件中配置参数 `only_save_strategy: True`，正常启动分布式微调任务，自动生成对应的分布式策略文件后，任务将会主动退出。

分布式策略文件会保存为`output/strategy/ckpt_strategy_rank_x.ckpt`，`ckpt_strategy_rank_x.ckpt`数量和卡数相同。

5.7.2 权重合并

运行离线转换脚本获得目标权重，脚本如下：

```
python mindformers/tools/transform_ckpt.py \  
--src_ckpt_strategy src_strategy_path_or_dir \  
--dst_ckpt_strategy dst_strategy_path_or_dir \  
--src_ckpt_dir src_ckpt_dir \  
--dst_ckpt_dir dst_ckpt_dir \  
--prefix "checkpoint_"
```

各个参数的说明可见上面的参考链接，推荐使用如下命令合并权重（如命令中涉及到绝对路径，仅供参考，请确认自己实际路径是否正确），此处提供可直接执行的脚本仅供示例参考：

```
cd /home/ma-user/work/mindformers/  
python mindformers/tools/transform_ckpt.py \  
--src_ckpt_strategy /home/ma-user/work/mindformers/research/output/strategy/ \  
--src_ckpt_dir /home/ma-user/work/mindformers/research/output/checkpoint/ \  
--dst_ckpt_dir /home/ma-user/work/mindformers/research/output/checkpoint/ \  
--prefix "new_lora_checkpoint_"
```

运行完成以后，合并后的权重文件会在`/home/ma-user/work/mindformers/research/output/checkpoint/rank_0`目录下，结合`--prefix`参数的设置，就可以找到合并后的权重文件。

5.8 微调后模型原有能力评估

原有能力评估，可在微调的4卡环境上直接运行。不过原有能力评估运行的最低配置环境是单卡NPU（显存32G），建议选手重新建立单卡环境运行评估，可节省代金券的使用。

评测步骤如下：

5.8.1 获取数据集

SQuAD 1.1包含针对500+文章的10万+问答对,是一个阅读理解数据集,由维基百科文章上提出的问题组成,其中每个问题的答案都是相应文章中的一段文本。

数据集获取链接: <https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic2-finetune/squad1.1.zip>; 可通过如下命令进行获取

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic2-finetune/squad1.1.zip
```

```
unzip squad1.1.zip
```

解压后的文件夹中包含两个json文件: train-v1.1.json 和 dev-v1.1.json,本次评估中我们将使用dev-v1.1.json文件。

5.8.2 处理数据成 mindrecord 格式

使用/home/ma-user/work/mindformers/mindformers/tools/dataset_preprocess/llama/squad_data_process.py进行数据预处理+Mindrecord数据生成

```
cd /home/ma-user/work/mindformers/mindformers/tools/dataset_preprocess/llama/
```

```
python squad_data_process.py \
```

```
--input_file /{path}/dev-v1.1.json \
```

```
--output_file /{path}/squad8192.mindrecord \
```

```
--mode eval \
```

```
--max_length 8192 \
```

```
--tokenizer_type "llama3-8B" > test_eval_base.log 2>&1 &
```

注: {path}为解压后的数据集dev-v1.1.json所在的绝对路径; squad8192.mindrecord为生成的mindrecord格式的数据集文件。> test_eval_base.log 2>&1 &为日志重定向脚本,可将日志保存到本地。

5.8.3 按照如下步骤修改配置文件

注: 以下具体参数设置,在"./mindformers/research/llama3/predict_llama3_8b_800T_A

2_64G.yaml”文件中修改。

eval_dataset的input_columns中增加labels,

metric type设为EmF1Metric,

修改seq_length为8192,

最大解码长度（max_decode_length）设为700

设置最大生成长度（max_new_tokens）为20,

因运行环境的显存大小为32G，需设置max_device_memory: “28GB”避免空间不足的问题。

修改好的配置文件内容如下方所示

```
# eval dataset

eval_dataset: &eval_dataset

  data_loader:

    type: MindDataset

    dataset_dir: ""

    shuffle: False

    input_columns: ["input_ids", "labels"]      # 增加"labels"

# metric

metric:

  type: EmF1Metric      # metric type设为EmF1Metric

# model config

model:

  model_config:

    type: LlamaConfig

    batch_size: 1 # add for increase predict

    seq_length: 8192      # seq_length设为8192

    max_decode_length: 700 # 最大解码长度设为700

    max_new_tokens: 20    # 设置最大生成长度

  context:

    max_device_memory: "28GB" # max_device_memory设为28GB
```

step 4. 开启评测，指标为Em/F1：进入run_mindformer.py文件所在路径，运行以下代码

```
python run_mindformer.py \  
--config research/llama3/predict_llama3_8b_800T_A2_64G.yaml \  
--eval_dataset_dir /{path}/squad8192.mindrecord \  
--run_mode eval \  
--load_checkpoint /{path}/llama3-8B.ckpt \  
--epochs 1 \  
--batch_size 1 \  
--use_parallel False \  
--device_id 0
```

注：{path}为llama3-8B.ckpt文件所在的实际路径，其中参数--device_id 只能设置为 0 不可设置其他值，否则会报错。

运行以上步骤后，得到的最终评测结果如下：

F1 score: 59.87023775108988, Em score: 44.17029511369134

最终评测结果90%的得分值如下，选手需要保证微调后的模型的原有能力得分大于等于下述数值，作品方可算作有效作品：

F1 score: 53.88321397598089, Em score: 39.75326560232221

5.9 微调后模型数学计算结果推理

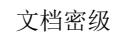
此处提供模型数学计算的推理的实现参考，选手可以参考下述方法得到数学计算的结果来自行对模型微调效果进行评估。

5.9.1 运行推理

此处提供另外一个生成推理结果的运行脚本，下载命令如下(命令中有绝对路径请注意根据自己实际路径来修改)：

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic2-finetune/run\_llama3\_test.py -P /home/ma-user/work/mindformers/research/llama3/
```

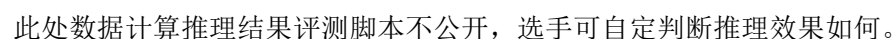
运行推理的命令（如命令中涉及到绝对路径，经供参考，请确认自己实际路径是否正确）：



其中“`--load_checkpoint`”参数为你模型微调完成之后合并之后的新的权重文件的路径，

“`--input_dir`”为输入数据的路径。

最后结果保存在目录 `/home/ma-user/work/mindformers/research` 下,保存的路径可修改,具体见`/home/ma-user/work/mindformers/research/llama3/run_llama3_test.py`文件的183行,如下图所示:



5.10 作品提交

所有作答文件汇总后打包成zip压缩包，以团队名称命名压缩包（如：团队名称.zip）参赛者可多次提交，最新一次提交将覆盖上一次提交作品，以最后提交的作品为准。

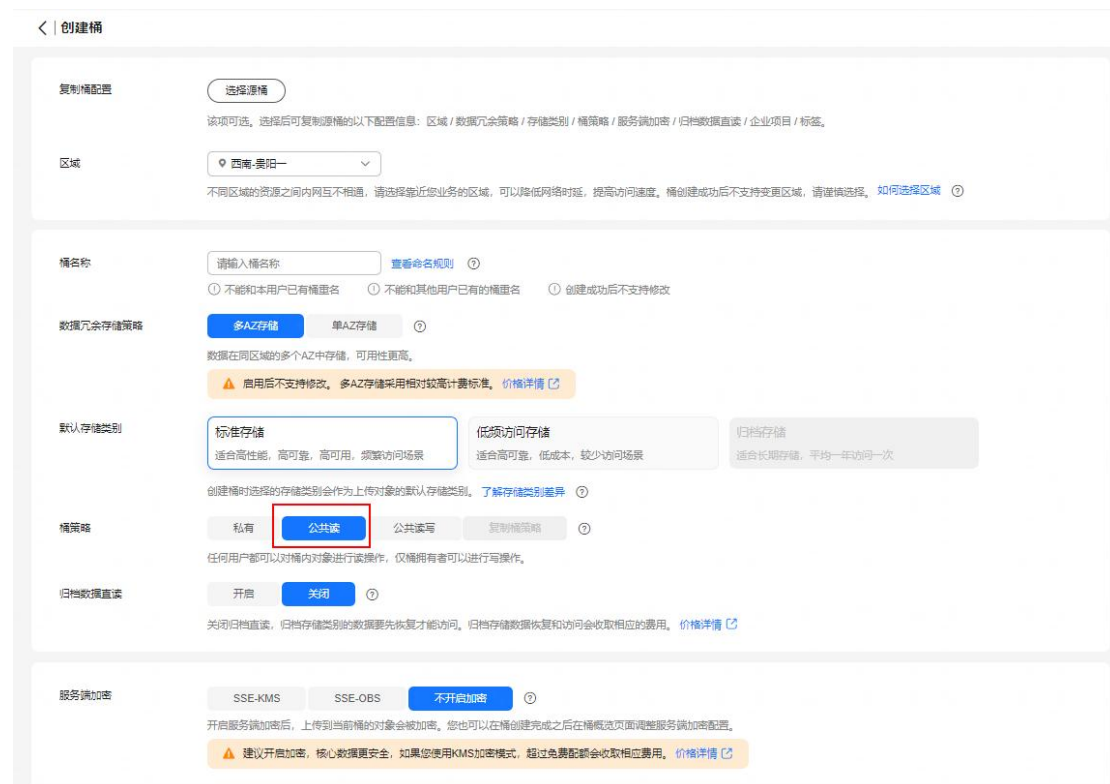
作答文件需包含以下内容：

1. 提供作品报告(word、pdf、markdown等格式)，模板如下：

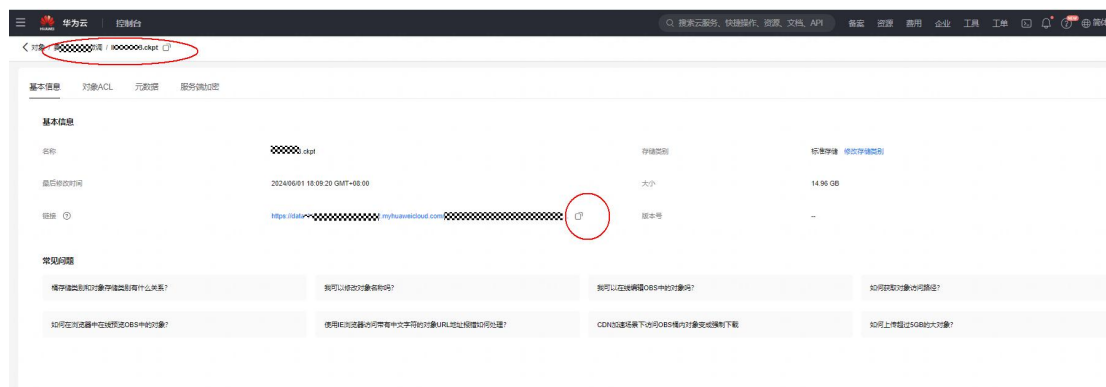
（1）微调算法介绍，包含使用的微调数据集规模的预处理方式

（2）超参配置介绍说明

（3）微调后的权重文件链接，权重文件可上传到自己的obs桶（注意桶需要读权限，选“公共读”，具体如下图）里面，然后将权重文件的下载链接（获取见下图）放入到作品报告里面：



The screenshot shows the 'Create Bucket' (创建桶) interface in the Huawei OBS console. The 'Bucket Policy' (桶策略) section is expanded, and the 'Public Read' (公共读) option is selected and highlighted with a red box. Other options include 'Private' (私有), 'Public Write' (公共读写), and 'Replication Policy' (复制桶策略). The 'Public Read' option is described as allowing any user to read objects in the bucket, while only bucket owners can perform write operations.



(4) 运行环境说明，即除了1.5.2 环境配置中提及的操作外，是否有进行额外的配置，如有请写出配置命令；

(5) 模型微调后原有能力评估得分:

(6) 作品验收时将以1.5.9 数学计算结果推理章节的方式获取测试集的推理结果，如选手对该推理方式有修改，请详细说明模型推理的操作步骤：

2. 提供模型微调的完整日志、yaml格式的配置文件；
3. 提供能保障从数据预处理到模型推理全流程跑通的mindformers源码包(可提供zip压缩文件，如文件过大可上传至自己的obs桶，并在作品报告中附上获取链接同权重文件)；
4. 原有能力评估的完整日志文件。

6 赛题三：推理调优指导

6.1 赛题介绍

基于给定数据集及后处理方法，跑通baseline，并对MindFormers中LLaMA2-7b模型进行推理调优，调优算法不限，在精度无损前提下（对比输出logits的误差，千分之五以内），推理性能相比baseline有提升，对推理总时间进行排名，推理时间越短排名越靠前。

精度无损：此评价方法以对比推理单个 token 的 logits 为准，要求绝对偏差值在千分之五以内的作品方可视为有效作品，请选手根据 1.6.6 的操作指引将模型推理的 logits 保存为 npy 文件。

推理总时间：因上述保存logits文件会增加额外耗时，所以建议选手运行两次：一次保存logits文件；一次不进行保存文件操作，仅作推理，推理总时间以后者为准，如何进行两次

运行的配置，请参考下方文档说明。

选手提交作品后，审核老师会检查代码是否包含前处理-推理-后处理全流程，且选手并没有通过如事先保存推理结果文件，然后直接读取文件进行推理等不正当方式缩短推理时间，或者发现选手修改官方提供的评测推理时间的脚本，一经发现有不正当手段即刻取消参赛资格。本赛题基础流程共分为以下5个环节：环境准备、模型权重准备、启动llm-serving、启动推理及推理时长获取、logits文件保存，下方会针对每个环节进行完整说明。

6.2 环境准备

本赛题指定使用华为云modelarts-开发环境-Notebook，使用32G显存的NPU，硬盘规格推荐使用300G，如下图所示设置：



The screenshot shows the configuration interface for a Huawei Cloud ModelArts Notebook. The 'Resource Type' is set to 'Public Resource Pool'. The 'Type' is set to 'ASCEND'. The 'Specification' is set to 'Ascend: 1*ascend-910b(32G)/ARM: 24核 192GB'. The 'Storage Configuration' is set to 'Cloud Disk EVS'. The 'Disk Specification' is set to '300' GB.

在默认基础环境下，即指定的华为云自定义镜像下，需按照要求额外安装指定的MindSpore和MindFormers依赖。注意，以下的命令强烈建议在终端运行。

6.2.1 模块卸载

在安装之前需要手动卸载两个镜像自带的两个模块，卸载命令如下：

```
pip uninstall mindformers mindspore-lite
```

6.2.2 MindSpore 安装

MindSpore可用如下命令安装：

```
pip install mindspore==2.3.0RC2
```

如果上面安装命令出现问题，可通过如下命令安装：

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic2-finetune/mindspore-2.3.0rc2-cp39-cp39-linux_aarch64.whl  
pip install mindspore-2.3.0rc2-cp39-cp39-linux\_aarch64.whl
```

6.2.3 MindFormers 包及 llm-serving 包下载

MindFormers包下载解压，命令及相关链接如下：

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic3-infer/mindformers.zip  
unzip mindformers.zip
```

注意：此处的MindFormers包不可以通过命令 `bash build.sh` 命令进行安装。

llm-serving包下载解压，命令及相关链接如下：

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic3-infer/llm-serving.zip  
unzip llm-serving.zip
```

注意：MindFormers和llm-serving不用额外安装，通过wget命令下载到当前目录后，可设置环境变量来直接使用。

6.2.4 环境变量配置

环境变量配置命令如下（环境变量的路径在设置的过程中请注意，以自己实际的路径为准）：

```
export PYTHONPATH=/home/ma-user/work/mindformers:$PYTHONPATH  
export PYTHONPATH=/home/ma-user/work/llm-serving:$PYTHONPATH  
export GRAPH_OP_RUN=1  
export MS_ENABLE_INTERNAL_KERNELS=on
```

下面两个环境变量也是运行llm-serving需要的，请一起设置。

设置完环境变量之后可通过命令：`echo $PYTHONPATH`，查看是否设置正确，正确结果如下所示（环境变量中的路径要与你实际文件的路径一致）：



还有另外其他的依赖需要安装，安装命令如下：

```
pip install tiktoken
```

注意：每次Notebook重新启动之后都需要重新卸载自带的mindformers和mindspore-lite包、安装MindSpore、设置环境变量一遍，依赖也需要重新安装一遍，之前下载过的文件会保留的。

6.3 模型权重准备

要运行起来需要先将权重文件和tokenizer文件下载到指定文件夹内，具体操作如下。

在与mindformers同级目录下（这里是 /home/ma-user/work/）创建目录，在终端输入命令如下：

```
mkdir -p checkpoint download/llama2/
```

下载11ama2-7b基础权重文件到该目录下，命令如下：

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic3-infer/llama2_7b.ckpt -P checkpoint download/llama2/
```

下载llama2-7b的tokenizer文件到该目录下，命令如下：

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic3-infer/tokenizer.model -P checkpoint/download/llama2/
```

6.4 启动 llm-serving

llm-serving的使用方法可参考链接：<https://gitee.com/mindspore/llm-serving>，也可参考serving仓库，链接为：<https://gitee.com/mindspore/serving>，还有MindSpore官网的介绍教程，链接：<https://www.mindspore.cn/serving/docs/zh-CN/master/index.html>。

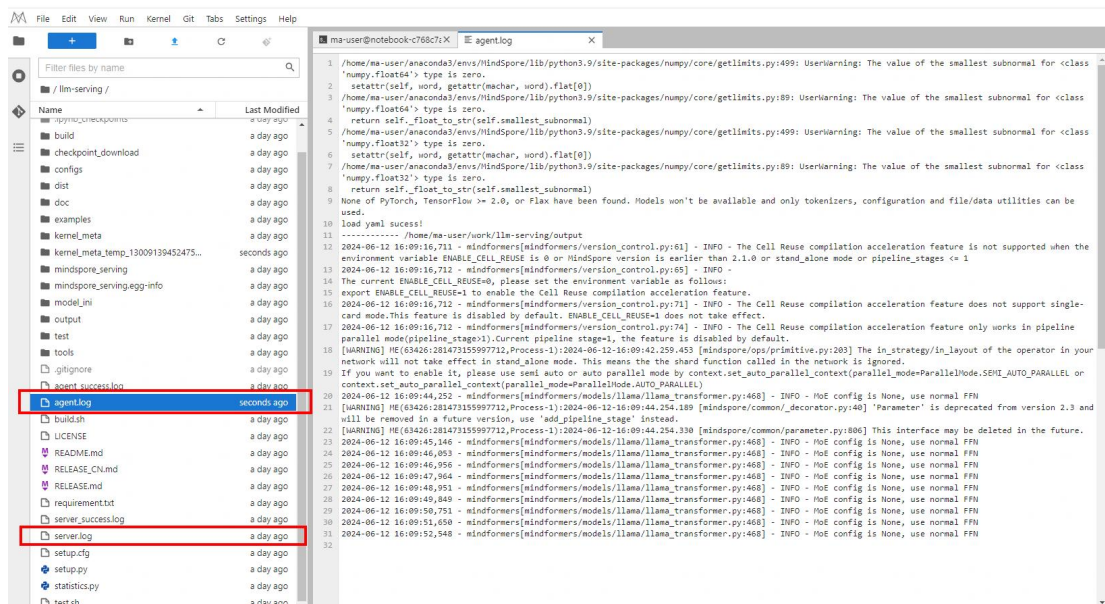
具体使用指导如下步骤。

使用 `start.py` 启动推理服务，命令如下：

```
cd /home/ma-user/work/llm-serving/  
python examples/start.py --config /home/ma-user/work/llm-serving/configs/llama/  
llama_7b_kbk_pa_dyn.yaml
```

此处配置文件可使用包中自带配置文件，如需修改请谨慎，以上命令中的路径以你本地实际路径为准。

运行成功serving服务拉起一般需要5分钟左右，请耐心等待。如果时间过长可查看运行中的日志情况，运行过程的日志文件保存可在 `/home/ma-user/work/llm-serving/` 目录下的 `agent.log` 和 `server.log` 文件里，具体如下截图：



运行成功之后终端显示如下图所示：

```
(MindSpore) [ma-user llm-serving]$python examples/start.py --config /home/ma-user/work/demo/llm-serving/co  
nfigs/llama/llama_7b_kbk_pa_dyn.yaml  
----starting agents----  
/home/ma-user/anaconda3/envs/MindSpore/lib/python3.9/subprocess.py:941: RuntimeWarning: line buffering (bu  
ffering=1) isn't supported in binary mode, the default buffer size will be used  
  self.stdout = io.open(c2pread, 'rb', bufsize)  
----agents are ready----  
----starting server----  
----server is ready----
```

另外说明：后续如果有其他操作需要关闭服务可见1.6.6说明。

6.5 启动推理及推理时长获取

此处提供两种推理方式。

第一种是快速推理，主要用于测试能否正常推理，实际推理时间检测主要通过第二种方式。

在serving服务启动成功的情况下，在终端运行如下代码可启动快速单条推理：

```
curl 127.0.0.1:8835/models/llama2/generate \
-X POST \
-d '{"inputs": " I love Beijing, because", "parameters": {"max_new_tokens": 56, "do_sample": "True", "return_full_text": "True"}, "stream": "True"}' \
-H 'Content-Type: application/json'
```

注意：此处的127.0.0.1:8835，中的8835要跟配置文件“/home/ma-user/work/demo/llm-serving/configs/llama/llama_7b_kbk_pa_dyn.yaml”中的 serving_config:下的 server_port:8835 一样；包中自带 llama_7b_kbk_pa_dyn.yaml 配置文件可直接运行。

成功之后如下图所示：

```
(MindSpore) [ma-user llm-serving]$
(MindSpore) [ma-user llm-serving]$ curl 127.0.0.1:8835/models/llama2/generate \
> -X POST \
> -d '{"inputs": " I love Beijing, because", "parameters": {"max_new_tokens": 16, "do_sample": "True", "return_full_text": "True"}, "stream": "True"}' \
> -H 'Content-Type: application/json'
{"generated_text": "it is the most beautiful city in the world. It is a city with", "finish_reason": "length", "generated_tokens": 16, "prefill": [{"id": 372, "logprob": 17.203125, "special": false, "text": "it"}], "seed": 0, "tokens": [{"id": 372, "logprob": 17.203125, "special": false, "text": "it"}, {"id": 338, "logprob": 16.25, "special": false, "text": " is"}, {"id": 278, "logprob": 13.0390625, "special": false, "text": " the"}, {"id": 1556, "logprob": 14.9296875, "special": false, "text": " most"}, {"id": 9560, "logprob": 14.890625, "special": false, "text": " beautiful"}, {"id": 4272, "logprob": 17.921875, "special": false, "text": " city"}, {"id": 297, "logprob": 21.96875, "special": false, "text": " in"}, {"id": 278, "logprob": 22.90625, "special": false, "text": " the"}, {"id": 3186, "logprob": 22.09375, "special": false, "text": " world"}, {"id": 29889, "logprob": 22.171875, "special": false, "text": "."}, {"id": 739, "logprob": 13.921875, "special": false, "text": " It"}, {"id": 338, "logprob": 17.515625, "special": false, "text": " is"}, {"id": 263, "logprob": 13.1953125, "special": false, "text": " a"}, {"id": 4272, "logprob": 15.8046875, "special": false, "text": " city"}, {"id": 411, "logprob": 13.828125, "special": false, "text": " with"}, {"id": 1784, "logprob": 12.328125, "special": true, "text": ""}], "top_tokens": [{"id": 372, "logprob": 17.203125, "special": false, "text": "it"}], "details": null}
(MindSpore) [ma-user llm-serving]$
```

第二种批量推理服务，这种方式也是主要用来测试推理时长的。

6.5.1 脚本获取

测试脚本下载解压命令如下：

```
cd /home/ma-user/work/
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic3-infer/performance_serving.zip
unzip performance_serving.zip
```

在目录 llm-serving/mindspore_serving/agent/ 下有两个文件，一个命名为：agent_multi_post_method.py，一个命名为：agent_multi_post_method_save_logits.py，推理运行会默认使用命名为“agent_multi_post_method.py”的文件，此文件也是用来收集推理时长的。

6.5.2 推理数据集说明

为了比赛的公平公正，选手必须使用指定测试推理时长的数据集，此数据集为 performance_serving/ 目录下的 alpaca_5010.json，此数据集是随 performance_serving.zip 包下载的，数据集路径的修改在 /home/ma-user/work/performance_serving 目录下的 test_serving_performance.py 文件的第211行，如下图所示：



```
202
203
204 if __name__ == '__main__':
205     parser = argparse.ArgumentParser(description="test serving performance")
206     parser.add_argument("-X", "--qps", help='x req/s', required=True, type=float)
207     parser.add_argument("-P", "--port", help='port, default is 8000', required=True)
208     parser.add_argument("-O", "--out_dir", help='dir for saving results', required=True)
209     parser.add_argument("-T", "--test_time", help='test all time, default 1h', required=False, type=int, default=3600)
210     args = parser.parse_args()
211     with open("./alpaca_5010.json") as f:
212         alpaca_data = json.loads(f.read())
213     INPUTS_DATA = []
214     OUTPUTS_DATA = []
215     for data in alpaca_data:
216         input_ = data["instruction"] + ":" + data["input"] if data["input"] else data["instruction"]
217         INPUTS_DATA.append(input_)
218         OUTPUTS_DATA.append(data["output"])
219     test_main(args.port, INPUTS_DATA, OUTPUTS_DATA, args.qps, args.out_dir, args.test_time)
220
```

运行之前请做好检查。

6.5.3 限定推理数据数目

为了比赛的公平公正，只需推理数据集的前1500条数据，这个设置是目录 /home/ma-user/work/performance_serving 下 test.sh 文件里面的代码：python test_serving_performance.py -X 1 -P 8835 -O "/" -T 5 中，参数说明如下：

-X 1：每秒发送1个请求；

-P 8835：此处端口号要跟配置文件 “/home/ma-user/work/demo/llm-serving/configs/llama/llama_7b_kbk_pa_dyn.yaml”中的 serving_config:下的 server_port:8835 一样；

-T 5：表示发送请求的总时间为5s，具体代码可见 test_serving_performance.py；

上面命令的意思就是，总共发送请求的时间为 5s，每 1s 发送一个推理请求，就是要发送 5 个推理请求，就是推理5条测试数据集。

必须保证 -X 的设定值乘以 -T 的设定值等于1500，比如可设置为 -X 0.5 -T 3000；注意这两个参数的不同设置可能会造成推理时长的变化，也可能导致模型没法成功推理出1500条数据，具体情况可见 performance_serving/testLog/ 目录下日志。

此处给出基准推理时间：3551.9252s，此时间也是推理的基准时间，超过这个时间才算有效作品，另外说明这个基准时间是在 -X 和 -T 设置的值为 0.5 和 3000 情况下跑出来的。

6.5.4 启动推理

推理启动可运行如下脚本：

```
cd /home/ma-user/work/performance_serving
```

```
nohup sh test.sh > test_sh.log 2>&1 &
```

注意：> test_sh.log 2>&1 &是用于日志重定向出来，便于保存推理的日志；

另外说明：

用于测试模型基础精度和推理的数据集已经内置在performance_serving文件中，请勿修改，如有修改可能导致模型基础精度测试不通过，后果选手自负。

推理运行完成以后，推理总时长是记录在 performance_serving/testLog/ 目录下日志文件的最后一行。

6.6 logits 文件保存

除了获取推理总时长之外，选手还需要提供调优以后模型推理生成的logits文件，目的是验证模型的精度，要求偏差在千分之五以内（即完成推理优化后的logits输出和优化前的标准logits输出绝对差值在千分之五以内），确保推理调优对模型推理的精度影响不会太大。具体操作流程如下：

6.6.1 修改配置文件

将目录 llm-serving/mindspore_serving/agent/ 下的 agent_multi_post_method.py 文件更改为其他名字做好备份，然后将 agent_multi_post_method_save_logits.py 文件改名为 agent_multi_post_method.py

6.6.2 关闭 llm-serving 服务

修改配置文件后，需要关闭后重启serving服务，保存npz文件的脚本才会生效。

关闭服务的具体操作截图如下：

```
(MindSpore) [ma-user performance_serving]$ps -elf |grep python
4 S ma-user 232 1 0 80 0 - 75393 ep_pol 09:31 ? 00:00:04 /modelarts/authoring/notebook-conda/bin/python /modelar
4 S ma-user 249 247 2 80 0 - 7236327 ep_pol 09:31 ? 00:02:20 /modelarts/authoring/notebook-conda/bin/python /modelar
4 S ma-user 33965 1 0 85 5 - 2954443 futex_ 09:58 pts/0 00:00:19 python examples/start_agent.py --config /home/ma-user/w
5 S ma-user 34446 33965 27 85 5 - 2157416650 wait_w 09:58 pts/0 00:19:34 python examples/start_agent.py --config /home/ma-user
1 S ma-user 34522 34446 0 85 5 - 2817842 futex_ 09:58 pts/0 00:00:13 python examples/start_agent.py --config /home/ma-user/w
4 S ma-user 34638 34446 0 85 5 - 55966 pipe_w 09:58 pts/0 00:00:00 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
4 S ma-user 34648 34446 0 85 5 - 56338 ep_pol 09:58 pts/0 00:00:00 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 34657 34648 0 85 5 - 181700 unix_s 09:58 pts/0 00:00:08 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 34658 34648 0 85 5 - 181871 unix_s 09:58 pts/0 00:00:08 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 34659 34648 0 85 5 - 181871 unix_s 09:58 pts/0 00:00:08 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 34660 34648 0 85 5 - 181871 unix_s 09:58 pts/0 00:00:08 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 34661 34648 0 85 5 - 181433 unix_s 09:58 pts/0 00:00:08 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 34662 34648 0 85 5 - 181869 unix_s 09:58 pts/0 00:00:08 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 34663 34648 0 85 5 - 181438 unix_s 09:58 pts/0 00:00:08 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 34664 34648 0 85 5 - 181438 unix_s 09:58 pts/0 00:00:08 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 34693 34648 0 85 5 - 181682 do_sel 09:58 pts/0 00:00:09 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 35112 34648 0 85 5 - 182030 futex_ 09:58 pts/0 00:00:08 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 35113 34648 0 85 5 - 182030 futex_ 09:58 pts/0 00:00:08 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 35114 34648 0 85 5 - 181886 futex_ 09:58 pts/0 00:00:08 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 35115 34648 0 85 5 - 181886 do_sys 09:58 pts/0 00:00:08 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 35116 34648 0 85 5 - 181886 futex_ 09:58 pts/0 00:00:08 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 35117 34648 0 85 5 - 181886 futex_ 09:58 pts/0 00:00:09 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 35118 34648 0 85 5 - 181886 futex_ 09:58 pts/0 00:00:08 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
5 S ma-user 35119 34648 0 85 5 - 181886 futex_ 09:58 pts/0 00:00:08 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
4 R ma-user 39221 1 78 85 5 - 5285542 - 10:02 pts/0 00:54:02 python examples/server_app_post.py --config /home/ma-us
4 S ma-user 39697 39221 0 85 5 - 55965 pipe_w 10:02 pts/0 00:00:00 /home/ma-user/anaconda3/envs/MindSpore/bin/python -c fr
4 R ma-user 123639 3638 0 85 5 - 53360 - 11:10 pts/0 00:00:00 grep --color=auto python
(MindSpore) [ma-user performance_serving]$
(MindSpore) [ma-user performance_serving]$kill -9 33965 34446 34522 39221
```

命令如下：

```
ps -elf | grep python
```

```
kill -9
```

6.6.3 重启 llm-serving

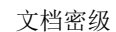
关闭完成之后需要按照1.6.4 启动llm-serving 章节重启serving服务。

6.6.4 指定数据集

保存 logits 文件需要用到的推理数据集为 performance_serving/ 目录下的 alpaca_521.json ，数据集路径的修改在 /home/ma-user/work/performance_serving 目录下的 test_serving_performance.py 文件的第211行，具体可见上文“6.5启动推理及时长获取”中的“第二种批量推理服务”下面的“2. 推理数据集说明”。

6.6.5 调整参数配置

因为在推理过程中需要保存模型输出的 logits 文件，所以相比 1.6.4 每条的推理时长会更久，为了比赛的公平公正，也为了方便验证精度，此处 -X 和 -T 的值必须设置为 0.1 和 5000，即选手推理500条数据。参数修改完成之后就可以使用1.6.5中的第4条启动推理里面启动推理生成logits文件。



该精度测试方法基本思路就是读取相对应的numpy文件,然后使用numpy中的allclose方法比对每个元素的绝对精度,如果绝对精度在千分之五以内方法就会返回True,否则就是False,所有文件比对都返回True即可算是合格,具体见代码。除了修改输入numpy文件的路径,精度测试代码其他部分选手请勿修改,如发现问题可向赛事组反馈。

精度测试的环境可在华为云Notebook环境,选手也可在自己本地CPU环境运行,为了节省代金券,建议选手下载代码到本地运行。

基准numpy文件获取命令:

```
cd /home/ma-user/work/
```

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic3-infer/file\_npy\_base.zip
```

```
unzip file\_npy\_base.zip
```

精度测试文件获取命令:

```
cd /home/ma-user/work/
```

```
wget https://2024-ascend-innovation-contest-mindspore.obs.cn-southwest-2.myhuaweicloud.com/topic3-infer/acc\_allclose.py
```

精度测试运行命令(如果命令中涉及到绝对路径,仅供参考,请确认自己实际路径是否正确):

```
cd /home/ma-user/work/
```

```
python acc_allclose.py \
```

```
--base_path /home/ma-user/work/file_npy_base \
```

```
--new_path /home/ma-user/work/file_npy_new
```

6.7 作品提交

所有作答文件汇总后打包成zip压缩包,以团队名称命名压缩包(如:团队名称.zip)参赛者可多次提交,最新一次提交将覆盖上一次提交作品,以最后提交的作品为准。

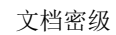
作答文件需包含以下内容:

提供作品报告(word、pdf、markdown等格式),模板如下:

业界推理优化算法调研

本作品使用的推理优化算法介绍

超参配置介绍



提交可以直接运行的llm-serving和performance_serving源码包，此处可以压缩为zip格式的压缩包（如文件过大可上传至自己的obs桶，并在作品报告中附上获取链接；obs桶的创建和文件url获取见下图，注意桶的权限配置）；

提交完成推理调优后生成的.npy文件，可将file_npy文件夹打zip包上传（如文件过大可上传至自己的obs桶，并在作品报告中附上获取链接）：

7 相关官方链接

MindSpore官网: <https://www.mindspore.cn/tutorials/zh-CN/r2.3.0rc2/index.html>

MindSpore代码仓: <https://gitee.com/mindspore/mindspore>

mindnlp: <https://github.com/mindspore-lab/mindnlp>

mindformers: [https://gitee.com/mindspore/mindformers? from=gitee_search](https://gitee.com/mindspore/mindformers?from=gitee_search)

mindformers使用说明文档: <https://mindformers.readthedocs.io/zh-cn/latest/>

llm serving: <https://gitee.com/mindspore/llm-serving>

serving: <https://gitee.com/mindspore/serving>

MindSpore Serving 文档: <https://www.mindspore.cn/serving/docs/zh-CN/master/index.html>