基于智能计算系统的自定义场景应用开发实验

中国科学院计算技术研究所
张欣
zhangxin@ict.ac.cn

常用工具—远程连接

- Windows系统

1. MobaXterm
2. xshell
3. vscode

- Linux系统
- macOS系统

开发手册

- 寒武纪开发者官网
- https://developer.cambricon.com/index/document/index/classid/3.html

常用工具—tmux

- 增强终端（Terminal）的功能，使其支持多任务管理、会话持久化和高效操作。
- 安装：

apt install tmux

常用工具—tmux

基本命令		
命令	**含义**	**快捷键（需先按前缀Ctrl+b）**
tmux	启动新会话	-
tmux new -s <session-name>	新建会话	——
tmux detach	分离会话	d
tmux attach -t <session-name>	重新接入会话	——
tmux kill-session -t <session-name>	关闭会话	exit 或 Ctrl+d
tmux ls	查看所有会话	——
tmux kill-server	关闭所有会话	——

窗口管理		
命令	**含义**	**快捷键（需先按前缀Ctrl+b）**
tmux new-window	创建新窗口	c
tmux next-window	切换到下一个窗口	n
tmux previous-window	切换到上一个窗口	p
tmux select-window -t <num>	切换到编号窗口	0-9（直接按数字键）
tmux kill-window	关闭当前窗口	&
tmux rename-window <name>	重命名当前窗口	——

常用工具—tmux

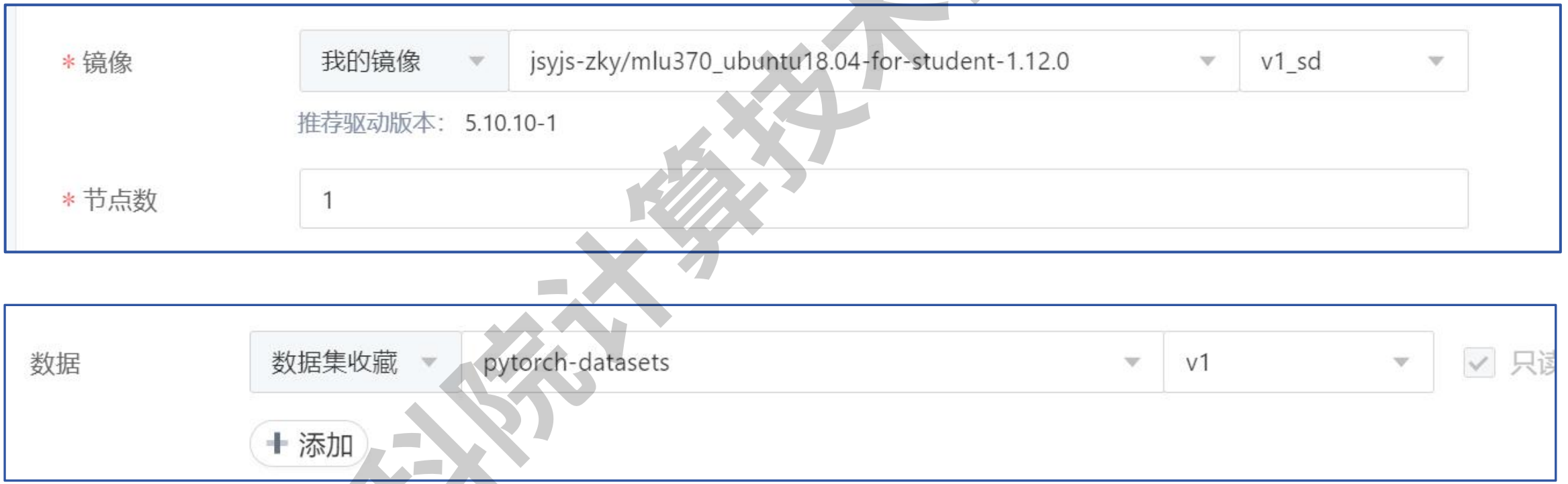
面板管理		
命令	**含义**	**快捷键（需先按前缀 Ctrl+b）**
tmux split-window -h	水平分割面板	%
tmux split-window -v	垂直分割面板	"
tmux select-pane -U/D/L/R	切换面板方向	↑/↓/←/→ 或 o 轮换
tmux kill-pane	关闭当前面板	x
tmux resize-pane -L/R/U/D <size>	调整面板大小	Ctrl+↑/↓/←/→（按住调整）
tmux swap-pane -U/D	交换面板位置	{ 或 }（上下交换）
tmux break-pane	将面板拆分为独立窗口	!
tmux display-panes	显示面板编号	q（短暂显示）

其他常用操作		
命令	**含义**	**快捷键（需先按前缀 Ctrl+b）**
进入复制模式	滚动查看历史输出	[（按 q 退出）
粘贴内容	粘贴复制的文本	]
显示时间	显示当前时间	t
刷新会话	重新加载配置	r
命令模式	输入高级命令	:（类似 Vim）
切换面板布局	循环切换布局（如 even-horizontal）	Space

- 小模型平台移植
- 大模型平台移植——基于transformers库的
- 大模型平台移植——基于vLLM的

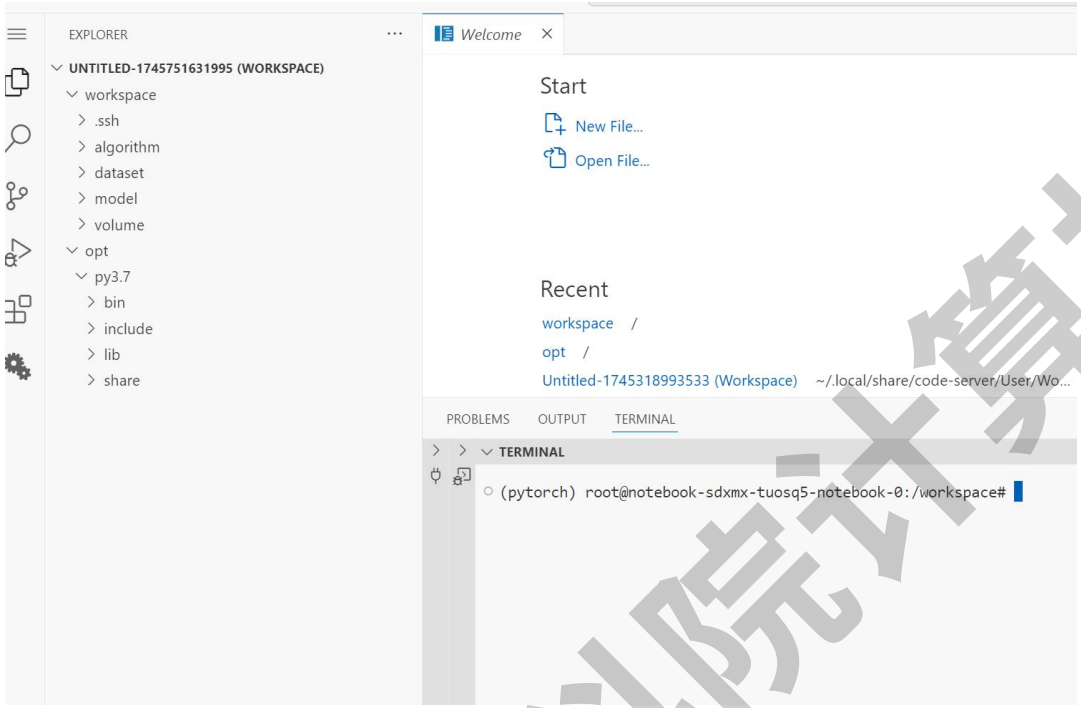
环境创建

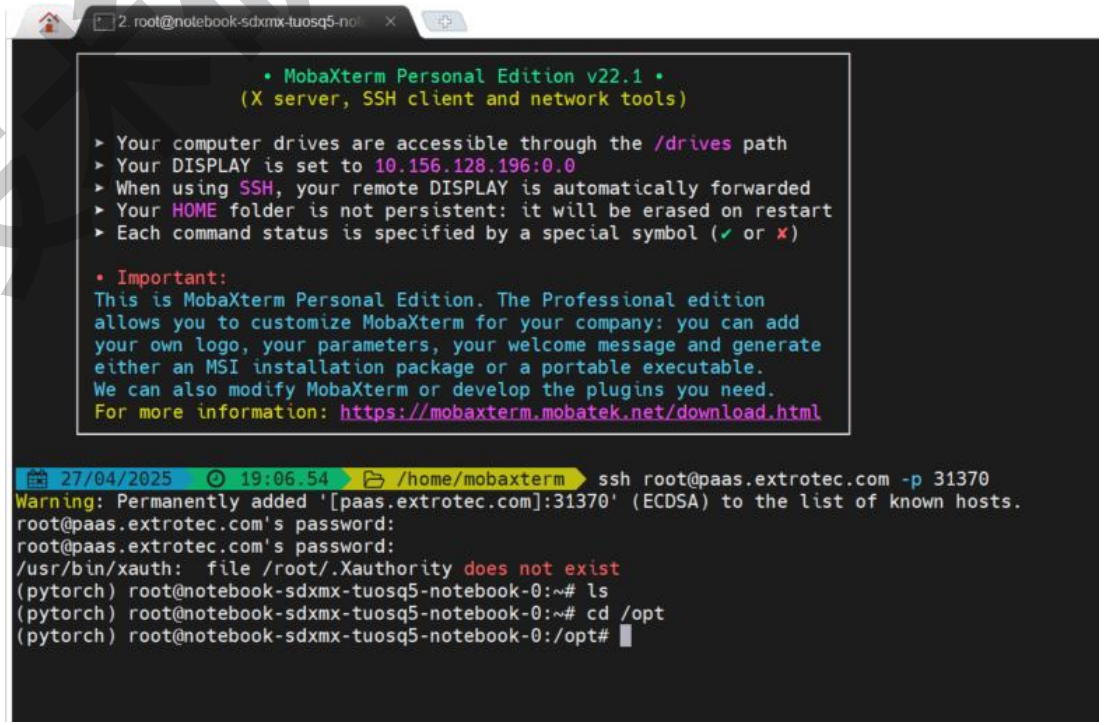
- 小模型平台移植



进入实验环境

ssh root@paas.extrotec.com -p 31370





硬件环境监测

命令	主要用途	简单理解
cnmon	查看寒武纪MLU设备的实时状态	查看MLU卡的利用率、温度、功耗、显存使用情况。像NVIDIA的nvidia-smi一样，用来看设备忙不忙、热不热、卡没卡死。
lscpu	查看CPU信息	查看服务器CPU的核数、型号、架构等。虽然跟MLU没直接关系，但有时也要了解CPU配置（比如异构加速时）。
free -h	查看内存(RAM)使用情况	看服务器总内存和剩余内存，单位自动换算成GB/MB，方便确认是不是内存爆掉，影响MLU程序运行。

```
root@notebook-devenviron-0629-092335--212633-so6ak8-notebook-0:~# cnmon
Fri Aug  2 02:20:54 2024
+-----------------------------------------------------------------------------+
| CNMON v5.10.12                                              Driver v5.10.12 |
+-------------------------------+----------------------+----------------------+
| Card  VF  Name       Firmware |              Bus-Id | Util        Ecc-Error |
| Fan   Temp      Pwr:Usage/Cap |       Memory-Usage | SR-IOV   Compute-Mode |
|===============================+======================+======================|
| 0     /   MLU370-M8    v1.1.4 |         0000:69:00.0 | 0%                   0 |
|  0%   22C         50 W/ 300 W |     0 MiB/ 42396 MiB | N/A           Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  Card  MI  PID     Command Line                                MLU Memory Usage |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

```
root@notebook-devenviron-0629-092335--212633-so6ak8-notebook-0:~# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                128
On-line CPU(s) list:   0-127
Thread(s) per core:    2
Core(s) per socket:    32
Socket(s):             2
NUMA node(s):          8
Vendor ID:             HygonGenuine
CPU family:            24
Model:                 1
Model name:            Hygon C86 7285 32-core Processor
Stepping:              1
CPU MHz:               1999.980
BogoMIPS:              3999.96
Virtualization:        AMD-V
L1d cache:             32K
L1i cache:             64K
L2 cache:              512K
L3 cache:              8192K
NUMA node0 CPU(s):     0-7,64-71
NUMA node1 CPU(s):     8-15,72-79
NUMA node2 CPU(s):     16-23,80-87
NUMA node3 CPU(s):     24-31,88-95
NUMA node4 CPU(s):     32-39,96-103
NUMA node5 CPU(s):     40-47,104-111
NUMA node6 CPU(s):     48-55,112-119
NUMA node7 CPU(s):     56-63,120-127
```

```
root@notebook-devenviron-0629-092335--212633-so6ak8-notebook-0:~# free -h
              total        used        free      shared  buff/cache   available
Mem:          503G         51G         47G        4.9M        405G        451G
Swap:           0B          0B          0B
```

网络介绍

- /torch/src/pytorch_models
- 涉及分类、检测、NLP、分割等
- 主要包括
  - 推理：Classification Detection LanguageModeling
  - 训练：Classification Detection LanguageModeling Recommendation Segmentation SpeechSynthesis

网络介绍—分类

```python
def test_cls_network(args):
    net=None
    in_h, in_w, resize, crop = (224,224,256,224)
    net_name = args.network
    pretrained = True if args.ckpt is None else False
    if net_name == "shufflenet_v2_x1_5":
        pretrained = False
    if net_name == 'inception_v3':
        net = getattr(models, net_name)(pretrained=pretrained, transform_input=False)
        in_h, in_w, resize, crop = (299,299,299,299)
    elif net_name == 'googlenet':
        net = getattr(models, net_name)(pretrained=pretrained, transform_input=False, aux_logits = False)
        # set googlenet aux as sigmoid op is for the success of the torch.jit.trace call
        net.aux1 = torch.nn.Sigmoid()
        net.aux2 = torch.nn.Sigmoid()
        in_h, in_w, resize, crop = (299,299,299,299)
    elif net_name == 'alexnet':
        net = getattr(models, net_name)(pretrained=pretrained)
        in_h, in_w, resize, crop = (227,227,256,227)
    else:
        net = getattr(models, net_name)(pretrained=pretrained)
    if args.ckpt is not None:
        pretrained_ckpt = torch.load(args.ckpt)
        if net_name in ['densenet121', 'densenet161', 'densenet169', 'densenet201']:
            pattern = re.compile(
                r'^(.*denselayer\d+\.(?:norm|relu|conv))\.((?:[12])\.(?:weight|bias|running_mean|running_var))$'
            )
```

环境变量设置（集成了所有网络）

export IMAGENET_TRAIN_DATASET=/workspace/dataset/favorite/pytorch-datasets/v1/imagenet_training/
export TORCH_HOME=/workspace/volume/zxvolume/modelzoo（这个环境变量大家随便建个文件夹，会从网上download预训练模型

网络介绍—分类

- 运行的脚本文件查看：
/torch/src/pytorch_models/Inference/Classification/vision_classification/cambricon

- 以resnet50为例：

执行
python /torch/src/pytorch_models/Inference/Classification/vision_classification/cambricon/../classify.py --network resnet50 --data /workspace/dataset/favorite/pytorch-datasets/v1/imagenet_training/ -j 12 --device mlu --fusion_backend no --batch_size 64 --input_data_type float32

```
/torch/src/pytorch_models/Inference/Classification/vision_classification/cambr
int64(double) to int(float) implicitly due to known MLU restrictions. (Trigger
src/aten/operators/cnnl_ops.cpp:830.)
  self.avg = self.sum / self.count
Test: [  0/782] Acc@1  90.62 ( 90.62)   Acc@5  93.75 ( 93.75)
Test: [  1/782] Acc@1  90.62 ( 90.62)   Acc@5  96.88 ( 95.31)
Test: [  2/782] Acc@1  81.25 ( 87.50)   Acc@5  96.88 ( 95.83)
Test: [  3/782] Acc@1  84.38 ( 86.72)   Acc@5  98.44 ( 96.48)
Test: [  4/782] Acc@1  76.56 ( 84.69)   Acc@5  92.19 ( 95.62)
Test: [  5/782] Acc@1  71.88 ( 82.55)   Acc@5  93.75 ( 95.31)
Test: [  6/782] Acc@1  78.12 ( 81.92)   Acc@5  96.88 ( 95.54)
Test: [  7/782] Acc@1  96.88 ( 83.79)   Acc@5  98.44 ( 95.90)
Test: [  8/782] Acc@1  92.19 ( 84.72)   Acc@5  98.44 ( 96.18)
Test: [  9/782] Acc@1  87.50 ( 85.00)   Acc@5 100.00 ( 96.56)
Test: [ 10/782] Acc@1  98.44 ( 86.22)   Acc@5 100.00 ( 96.88)
Test: [ 11/782] Acc@1  93.75 ( 86.85)   Acc@5  96.88 ( 96.88)
Test: [ 12/782] Acc@1  90.62 ( 87.14)   Acc@5  95.31 ( 96.75)
Test: [ 13/782] Acc@1  89.06 ( 87.28)   Acc@5  95.31 ( 96.65)
Test: [ 14/782] Acc@1  85.94 ( 87.19)   Acc@5  95.31 ( 96.56)
Test: [ 15/782] Acc@1  93.75 ( 87.60)   Acc@5  96.88 ( 96.58)
Test: [ 16/782] Acc@1  82.81 ( 87.32)   Acc@5  92.19 ( 96.32)
Test: [ 17/782] Acc@1  92.19 ( 87.59)   Acc@5  98.44 ( 96.44)
Test: [ 18/782] Acc@1  90.62 ( 87.75)   Acc@5  98.44 ( 96.55)
Test: [ 19/782] Acc@1  95.31 ( 88.12)   Acc@5  98.44 ( 96.64)
Test: [ 20/782] Acc@1  71.88 ( 87.35)   Acc@5  96.88 ( 96.65)
Test: [ 21/782] Acc@1  73.44 ( 86.72)   Acc@5  85.94 ( 96.16)
Test: [ 22/782] Acc@1  87.50 ( 86.75)   Acc@5  96.88 ( 96.20)
Test: [ 23/782] Acc@1  81.25 ( 86.52)   Acc@5  89.06 ( 95.90)
Test: [ 24/782] Acc@1  85.94 ( 86.50)   Acc@5  98.44 ( 96.00)
Test: [ 25/782] Acc@1  48.44 ( 85.04)   Acc@5  85.94 ( 95.61)
Test: [ 26/782] Acc@1  65.62 ( 84.32)   Acc@5  85.94 ( 95.25)
```

网络介绍—检测

- 和分类运行步骤同理
- 如yolov3，运行的脚本文件查看：
/torch/src/pytorch_models/Inference/Detection/yolov3/cambricon
- 可移植其他的代码

网络介绍—自然语言处理

- 和分类运行步骤同理

- 如bert，运行的脚本文件查看：/torch/src/pytorch_models/Inference/LanguageModeling/bert_base_chinese/cambricon

- 可移植其他的代码

- （注：有的算子可能会存在不支持的现象，但大部分都是支持的，不支持的可以提供反馈。对算子开发比较感兴趣也可自行开发）

gpu2mlu转换方式

- **方式一**：一行一行改，是gpu的代码适应mlu
- **方式二**：使用gpu2mlu工具一键生成

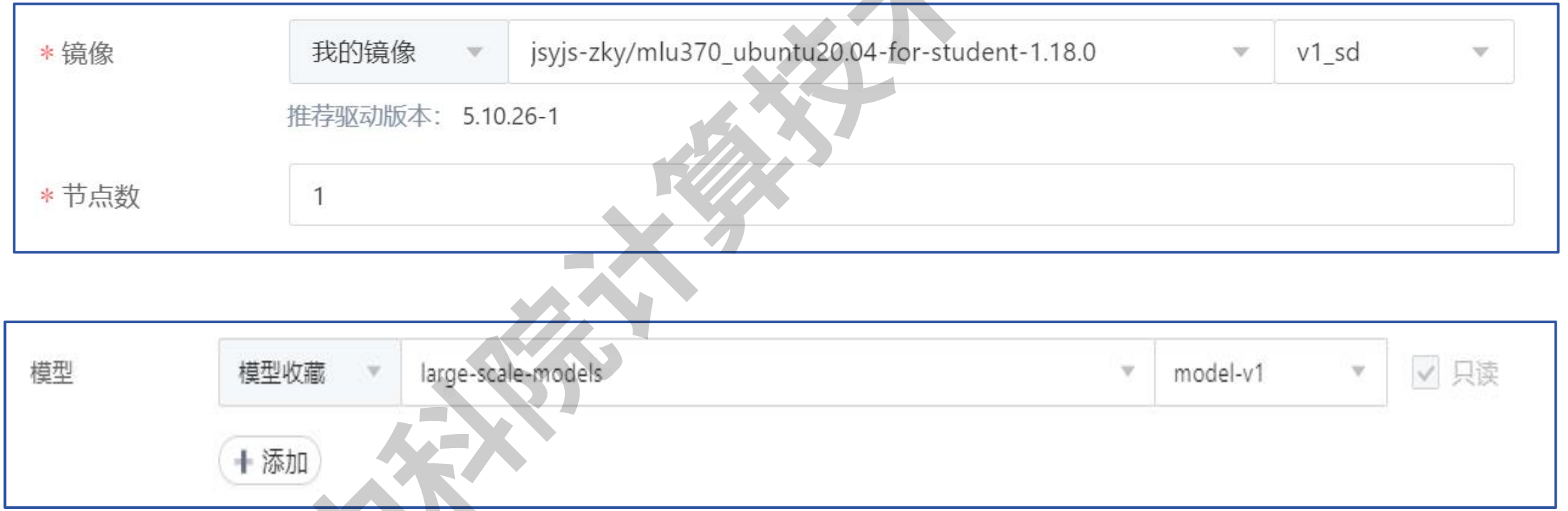
具体方式（在对应的文件夹下，使用如下命令自动生成mlu上的代码）

python /torch/src/catch/tools/torch_gpu2mlu/torch_gpu2mlu.py -i src

- 小模型平台移植
- 大模型平台移植——基于transformers库的
- 大模型平台移植——基于vLLM的

环境创建

- 大模型平台移植



Transformers库简介

- Hugging Face Transformers 是一个用于自然语言处理（NLP）的开源库，提供了各种预训练模型。这些模型被广泛应用于各种任务，如文本分类、命名实体识别、问答、文本生成等。Transformers库易于使用，可方便地集成到现有的深度学习框架，如PyTorch和TensorFlow。
- 具体用法可参见官方文档https://huggingface.co/docs/transformers/index
- 寒武纪平台已进行了支持。

环境依赖安装

- 安装accelerate安装包
  - cd /opt/tools/accelerate-0.20-release-mlu
  - python setup.py install
- 安装transformers安装包
  - cd /opt/tools/transformers-mlu-dev
  - python setup.py install
- 安装peft环境（运行如下实验需要，非必须）
  - pip install peft==0.3.0

模型移植介绍

- CMMLU（Chinese Massive Multitask Language Understanding）是一个中文大规模多任务语言理解评测基准，类似英文领域中的 MMLU（Massive Multitask Language Understanding），它的目标是：系统评估中文大语言模型（LLMs）在多种真实任务上的能力，包括知识掌握、推理能力和语言理解等。
- 链接：https://github.com/haonan-li/CMMLU

功能	说明
数据集	包含 67 个子任务（如法律、医学、历史、物理等）共约 20,000 道题目，全为选择题
评测脚本	提供对本地大模型（如 ChatGLM、Qwen、Baichuan）或 API 模型（如 GPT）进行准确率评估
模型输出处理	支持从模型输出中提取答案、自动评分等
适配接口	可连接 Huggingface Transformers、OpenAI API、自定义模型调用函数

- 下载代码：git clone https://github.com/haonan-li/CMMLU

模型移植

- 进入代码目录：cd CMMLU
- 采用转换工具gpu2mlu直接转换生成可运行程序，

```
python /torch/src/catch/tools/torch_gpu2mlu/torch_gpu2mlu.py -i src
```

- 生成src_mlu

- 根据脚本文件撰写llama-mlu.sh的脚本文件，运行脚本。

实验结果及硬件环境监测

- 实验结果

```
(pytorch) root@future-X7840H0:/home/ict-test/CMMLU/script# bash llama_7b-mlu.sh
The model weights are not tied. Please use the `tie_weights` method before using the `infer_auto
Loading checkpoint shards: 100%|██████████| 2/
/torch/venv3/pytorch/lib/python3.10/site-packages/transformers-4.34.0-py3.10.egg/transformers/to
  self.data = {k: v.to(device=device) for k, v in self.data.items()}
Average accuracy 0.272 - agronomy
Average accuracy 0.264 - anatomy
Average accuracy 0.256 - ancient_chinese
Average accuracy 0.287 - arts
Average accuracy 0.279 - astronomy
Average accuracy 0.282 - business_ethics Average accuracy 0.231 - chinese_civil_service_exam
Average accuracy 0.344 - chinese_driving_rule
Average accuracy 0.250 - chinese_food_culture
Average accuracy 0.262 - chinese_foreign_policy
Average accuracy 0.269 - chinese_history
Average accuracy 0.279 - chinese_literature
Average accuracy 0.274 - chinese_teacher_qualification
Average accuracy 0.245 - clinical_knowledge
Average accuracy 0.302 - college_actuarial_science
Average accuracy 0.336 - college_education
Average accuracy 0.330 - college_engineering_hydrology
Average accuracy 0.296 - college_law
Average accuracy 0.314 - college_mathematics
Average accuracy 0.217 - college_medical_statistics
```

- watch -n 0.5 cnmon（实时监测，查看利用率、显存）

```
Every 0.5s: cnmon                                  notebook-devenviron-0826-llm-210526-1a1a

Mon Aug 26 21:37:24 2024
+-----------------------------------------------------------------------------+
| CNMON v5.10.12                                                 Driver v5.10.12 |
+-------------------------------+----------------------+----------------------+
| Card  VF  Name       Firmware |              Bus-Id | Util        Ecc-Error |
| Fan   Temp      Pwr:Usage/Cap |       Memory-Usage | SR-IOV   Compute-Mode |
|===============================+======================+======================|
| 0     /   MLU370-M8    v1.1.4 |       0000:A9:00.0 | 97%                 0 |
|  0%   33C        113 W/ 300 W | 28226 MiB/ 42396 MiB | N/A           Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  Card  MI  PID     Command Line                                MLU Memory Usage |
|=============================================================================|
|  0     /   945     python                                            27771 MiB |
+-----------------------------------------------------------------------------+
```

测试结果

7B 模型结果

	C-Eval	MMLU	CMMLU	Gaokao	AGIEval	BBH
	5-shot	5-shot	5-shot	5-shot	5-shot	3-shot
GPT-4	68.40	83.93	70.33	66.15	63.27	75.12
GPT-3.5 Turbo	51.10	68.54	54.06	47.07	46.13	61.59
LLaMA-7B	27.10	35.10	26.75	27.81	28.17	32.38
LLaMA2-7B	28.90	45.73	31.38	25.97	26.53	39.16
MPT-7B	27.15	27.93	26.00	26.54	24.83	35.20
Falcon-7B	24.23	26.03	25.66	24.24	24.10	28.77
ChatGLM2-6B	50.20	45.90	49.00	49.44	45.28	31.65
Baichuan-7B	42.80	42.30	44.02	36.34	34.44	32.48
Baichuan2-7B-Base	54.00	54.16	57.07	47.47	42.73	41.56

- 小模型平台移植
- 大模型平台移植——基于transformers库的
- 大模型平台移植——基于vLLM的

vLLM环境创建

- 大模型平台移植

deepseek运行

- 更改路径地址：

```
#model_name = "/opt/shared/llm/models/DeepSeek-R1-Distill-Qwen-14B/"
model_name = "/workspace/model/favorite/large-scale-models/model-v1/DeepSeek-R1-Distill-Qwen-14B"
#model_name = "/workspace/volume/guojunceshi2/DeepSeek-R1-Distill-Qwen-14B"
model = AutoModelForCausalLM.from_pretrained(model_name,torch_dtype=torch.float16,device_map="auto")
```

- 实验运行：

```
(pytorch) root@notebook-chapvllm-1km2es1-notebook-0:/opt# cd deepseek/
(pytorch) root@notebook-chapvllm-1km2es1-notebook-0:/opt/deepseek# python testdeepseek.py
Sliding Window Attention is enabled but not implemented for `sdpa`; unexpected results may be encountered.
/torch/venv3/pytorch/lib/python3.10/site-packages/torch_mlu/mlu/__init__.py:379: UserWarning: Linear memory is not supported on this device. Falling back to common memory. (Triggered internally at /torch_mlu/torch_mlu/csrc/framework/core/caching_allocator.cpp:718.)
  torch_mlu._MLUC._mlu_init()
Loading checkpoint shards: 100%|████████████████████████████████████████| 4/4 [00:31<00:00,  7.95s/it]
Setting `pad_token_id` to `eos_token_id`:151643 for open-end generation.
[2025-05-12 10:27:02.228868][CNNL][WARNING][27849][Card:0]: [cnnlFill_v3] is deprecated and will be removed in the future release, Use [cnnlFill_v4] instead.
[2025-05-12 10:27:02.251692][CNNL][WARNING][27849][Card:0]: [cnnlMasked_v4] is deprecated and will be removed in the future release, Use [cnnlMasked_v5] instead.
好，用户只有3000元，想在北京玩三天。我得帮他制定一个预算低但又能体验北京精华的行程。首先，得考虑交通和住宿，这部分不能省太多，但也不能花太多钱。

交通方面，北京的地铁很发达，买张地铁日票可能更划算。第一天从机场到市区，打车可能有点贵，但第一天可能比较累，还是建议打车。之后的交通尽量用地铁和公交，这样能省不少钱。

住宿的话，选择青年旅舍或者经济型酒店，价格大概150-200一天，三天的话450-600，这样预算还能控制住。然后景点门票方面，故宫、长城这些是必去的，学生证可能有优惠，得提醒他带学生证。

第一天安排南锣鼓巷和故宫，这样比较集中，不会太累。第二天去长城，八达岭或者慕田峪，选个离市区近的，省车费。下午去鸟巢水立方，晚上回市区。第三天去天坛和簋街，这样比较轻松，还能体验北京的夜生活。

餐饮方面，得找一些小吃摊和街边小店，比如炸酱面、豆汁儿这些，价格便宜又能尝到地道风味。预算控制在每天150元左右，三天450，这样总体加起来应该差不多。

最后，提醒他带学生证，提前订票，注意保暖和防晒。这样整个行程既经济又能体验到北京的主要景点，应该能满足他的需求。
</think>

好的！以下是一个适合预算3000元的北京3天游行程建议。这个行程尽量控制成本，同时涵盖北京的精华景点和体验。

---

### **第一天：抵达北京，感受古都文化**
**交通：**
- 抵达北京（首都国际机场或大兴国际机场），打车或乘坐机场快线到市区（约80-150元，视距离而定）。

**住宿：**
- 选择经济型酒店或青年旅舍，价格约150-200元/晚。

**行程：**
1. **上午：**
   - 抵达后，前往**南锣鼓巷**，体验北京最具特色的胡同文化，感受老北京的市井生活。
   - 午餐：在南锣鼓巷附近的小吃摊尝试北京传统小吃，如炸酱面、
/opt/py3.10/lib/python3.10/tempfile.py:860: ResourceWarning: Implicitly cleaning up <TemporaryDirectory '/tmp/tmpe3v54a49'>
  _warnings.warn(warn_message, ResourceWarning)
```

vLLM介绍

- vLLM是一种面向大语言模型（LLM）推理优化的高性能推理引擎，致力于在大规模生成任务中提升推理速度与资源利用效率。vLLM支持用户从Hugging Face平台下载主流模型，在本地硬件环境中基于自定义配置进行高效部署，同时兼容OpenAI API Server协议。借助vLLM，用户能够在本地灵活试验不同的大模型，开发和部署基于LLM的应用系统，避免对外部托管服务的依赖，显著提升推理过程的可控性、安全性与部署灵活性。
- vLLM具体介绍可参见https://docs.vllm.ai/en/latest/ 。
- Cambricon vLLM和社区vLLM使用方法一致，可将已部署在vLLM上的LLM应用快速迁移到寒武纪设备。

deepseek的vLLM测试

- 环境激活：

```
(pytorch) root@notebook-chapvllm-1km2es1-notebook-0:/opt/deepseek# deactivate
root@notebook-chapvllm-1km2es1-notebook-0:/opt/deepseek# source /torch/venv3/pytorch_infer/bin/activate
```

- 进入目录：

```
(pytorch_infer) root@notebook-chapvllm-1km2es1-notebook-0:/opt/deepseek# cd /workspace/
(pytorch_infer) root@notebook-chapvllm-1km2es1-notebook-0:/workspace# ls
Cambricon_PyTorch_Model_Zoo  Megatron-LM       algorithm  dataset   ffmpeg-mlu-v4.2.0  torch_mlu_ops-v1.3.2  volume
DeepSpeed                    Megatron-LM-0.9.0  comfyui    diffusers  model              vllm-v0.6.2           webui
(pytorch_infer) root@notebook-chapvllm-1km2es1-notebook-0:/workspace# cd vllm-v0.6.2/
```

- 执行脚本文件：

```
export VLLM_LATENCY_DEBUG=true
python ./benchmarks/benchmark_latency.py --max-model-len 2560 --block-size 2560 --model /workspace/model/favorite/large-scale-models/model-v1/DeepSeek-R1-Distill-Qwen-7B/ --tokenizer /workspace/model/favorite/large-scale-models/model-v1/DeepSeek-R1-Distill-Qwen-7B/ --num-iters 1 --dtype float16 --input-len 512 --output-len 256 -tp 1 --batch-size 1 --max-model-len 768 --trust-remote-code --num-iters-warmup 1 --max-seq-len-to-capture 768 --max-num-batched-tokens 768
```

deepseek的vLLM测试结果

```
he usage: 0.7%, CPU KV cache usage: 0.0%.
INFO 05-12 11:17:24 metrics.py:449] Avg prompt throughput: 0.0 tokens/s, Avg generation throughput: 14.4 tokens/s, Running: 1 reqs, Swapped: 0 reqs, Pending: 0 reqs, GPU KV cac
he usage: 0.7%, CPU KV cache usage: 0.0%.
Profiling iterations: 100%|[redacted]| 1/1 [00:17<00:00, 17.85s/it]
Avg latency: 17.844543006271124 seconds
10% percentile latency: 17.844543006271124 seconds
25% percentile latency: 17.844543006271124 seconds
50% percentile latency: 17.844543006271124 seconds
75% percentile latency: 17.844543006271124 seconds
90% percentile latency: 17.844543006271124 seconds
99% percentile latency: 17.844543006271124 seconds
********************************* Test Info*****************************
Generation Config input len:512 output len:256 tp_nums:1 quantization:None
INFO 05-12 11:17:25 dump_info.py:400] Unsupport dump device/cpu information
************************Performance Info*****************************
                batch size  context latency(ms)  per token latency(ms)  context latency device(ms)  per token latency device(ms)  e2e latency(ms)  e2e throughput(tokens/s)  de
coder throughput(tokens/s)
iter index

0                    1           140.03                69.08                  136.98                       66.53              17843.68              14.4
         14.48
Average(1iters)      1           140.03                69.08                  136.98                       66.53              17843.68              14.4
         14.48
                profile memory(GB)  total cache memory(GB)  max cache used(GB)  mean cache used(GB)  max cache usage(%)  mean cache usage(%)
iter index
0                       16.88                20.38                0.14                 0.14                0.67                0.67
Average(1iters)         16.88                20.38                0.14                 0.14                0.67                0.67
Context tflops:  7.292586033152 Tflops
Generate tflops: 0.014397472768 Tflops
Context tflops_per_second:  53.238327005051836 Tflops/s
Generate tflops_per_second: 0.21640572325266796 Tflops/s
***********************************************************************
INFO 05-12 11:17:25 mlu_metric.py:149] Metric written to output.csv
```

DeepSeek的vLLM运行

- 环境激活：

```
(pytorch) root@notebook-chapvllm-1km2es1-notebook-0:/opt/deepseek# deactivate
root@notebook-chapvllm-1km2es1-notebook-0:/opt/deepseek# source /torch/venv3/pytorch_infer/bin/activate
```

- 结果：

```
Processed prompts: 100%|████████████████████████████████████████████████| 1/1 [03:36<00:00, 216.48s/it, est. speed input: 0.08 toks/s, output: 7.21 toks/s]
Prompt: '我只有3000元，帮我制定一份北京3天游<think>\n', Think: '嗯，用户说他只有3000元，想让我帮他制定一份北京3天游的计划。首先，我需要考虑他的预算情况，3000元对于三天来说不算太多，尤其是交通、住宿和餐饮都需要考虑进去。我应该尽量帮他节省开支，同时又能让他体验到北京的精华部分。\n\n首先，交通方面，北京的地铁比较便宜，而且覆盖范围广，建议他使用地铁出行，这样既便宜又快捷。如果需要短途出行，比如从市中心到景点，打车可能会更方便，但费用可能会高一些。不过，考虑到预算有限，地铁应该是主要的交通工具。\n\n接下来是住宿，选择经济型酒店或者青年旅舍会比较划算。比如，北京有很多如家、汉庭这样的连锁酒店，价格通常在200-300元之间，而且位置相对市中心，可能交通便利一些。另外，青年旅舍的价格更便宜，但可能需要和其他人共享房间，适合独自旅行的用户。\n\n然后是餐饮，北京有很多小吃和特色菜，价格实惠。比如，簋街的小吃街，那里有很多摊位，价格便宜，味道也不错。午餐和晚餐可以选择这样的地方，既省钱又能品尝当地美食。早餐可以选择街边的包子铺或者豆浆摊，价格便宜，方便快捷。\n\n景点方面，北京有很多免费或低价的景点，比如天安门广场、故宫（虽然门票不便宜，但可以考虑是否有必要）、长城、景山公园、北海公园等。考虑到预算，可能需要选择一些免费或低价的景点，或者只选择一两个收费较高的景点，比如故宫或者长城，其他时间则安排在免费景点上。\n\n第一天的安排，建议从天安门广场开始，因为这是北京的中心，免费开放，适合上午游览。然后去景山公园，这里可以俯瞰故宫全景，门票不贵，而且风景不错。中午在附近的小吃街解决午餐，下午去故宫，虽然门票有点贵，但这是北京的标志性景点，值得一看。晚餐可以选择簋街，那里夜市很热闹，有很多小吃。\n\n第二天，可以安排圆明园和颐和园，这两个景点相邻，可以一起游览。圆明园门票便宜，颐和园稍微贵一些，但总体来说，这两个景点加起来的费用还是可以接受的。中午在附近的小吃店解决，下午去国家博物馆，那里免费，可以了解中国的文化和历史。晚餐可以选择烤鸭，北京烤鸭是一定要尝的，但价格稍微高一些，可以控制在预算范围内。\n\n第三天，可以安排八达岭长城，虽然距离市区有点远，但可以乘坐地铁或打车过去。长城是北京的象征，值得一去。午餐在附近的小吃店，下午返回市区，逛逛南锣鼓巷，那里有很多小吃和纪念品店，适合休闲散步。晚餐可以选择炸酱面，价格便宜，味道也不错。\n\n在预算分配上，住宿大约每天200元，三天就是600元。餐饮方面，每天200元，三天600元。交通方面，每天100元，三天300元。门票方面，故宫60元，景山公园10元，圆明园10元，颐和园20元，国家博物馆免费，长城40元，总共140元。其他费用比如小吃和纪念品，控制在500元以内。\n\n这样算下来，总预算大约在3000元左右，比较符合用户的需求。同时，这样的安排既能让他体验到北京的主要景点，又能控制在预算内。可能需要提醒他注意交通和时间安排，尽量利用地铁，提前查好路线，避免打车费用过高。另外，建议他提前预订住宿，可能会有折扣，或者选择经济型酒店，节省开支。\n\n总的来说，这样的计划应该能满足用户的需求，让他在北京度过一个充实而经济的三天游。\n', Answer: '\n\n好的！以下是一个适合预算约3000元的北京3天游行程计划，尽量控制在预算范围内，同时尽量体验北京的精华部分。\n\n---\n\n### **第一天：市中心经典景点**\n**预算：约800元**\n\n1. **上午：天安门广场 & 毛主席纪念堂**\n   - **交通**：地铁1号线或2号线到天安门东站。\n   - **费用**：免费。\n   - **时间**：1小时。\n   - **亮点**：作为北京的中心，这里是感受中国历史与现代交融的地方。\n\n2. **中午：午餐**\n   - **推荐地点**：天安门广场附近的小吃摊，尝试北京炸酱面、肉夹馍等。\n   - **费用**：约30元。\n\n3. **下午：景山公园**\n   - **交通**：步行或打车到景山公园。\n   - **费用**：门票10元。\n   - **时间**：1.5小时。\n   - **亮点**：登高远眺紫禁城全景，感受历史文化。\n\n4. **下午：故宫**\n   - **交通**：步行即可。\n   - **费用**：门票60元。\n   - **时间**：2-3小时。\n   - **亮点**：中国古代皇宫，建筑与文物的瑰宝。\n\n5. **晚上：晚餐**\n   - **推荐地点**：簋街美食街。\n   - **费用**：约50元/人。\n   - **亮点**：尝试北京烤鸭、涮羊肉等特色美食。\n\n---\n\n### **第二天：文化与历史之旅**\n**预算：约800元**\n\n1. **上午：圆明园 & 颐和园**\n   - **交通**：地铁4号线到圆明园站。\n   - **费用**：圆明园门票10元，颐和园门票20元。\n   - **时间**：3小时。\n   - **亮点**：感受中国古代园林艺术，欣赏昆明湖美景。\n\n2. **中午：午餐**\n   - **推荐地点**：圆明园附近的小吃摊。\n   - **费用**：约30元。\n\n3. **下午：国家博物馆**\n   - **交通**：步行即可。\n   - **费用**：免费。\n   - **时间**：2小时。\n   - **亮点**：了解中国悠久的历史和文化。\n\n4. **晚上：晚餐**\n   - **推荐地点**：尝试北京烤鸭（如便宜坊或大董烤鸭店）。\n   - **费用**：约80元/人。\n\n---\n\n### **第三天：胡同与现代北京**\n**预算：约700元**\n\n1. **上午：南锣鼓巷**\n   - **交通**：地铁6号线到南锣鼓巷站。\n   - **费用**：免费。\n   - **时间**：2小时。\n   - **亮点**：体验北京的胡同文化，逛文艺小店。\n\n2. **中午：午餐**\n   - **推荐地点**：南锣鼓巷内的小吃店。\n   - **费用**：约30元。\n\n3. **下午：八达岭长城**\n   - **交通**：地铁2号线到西直门站，换乘S2线到八达岭站。\n   - **费用**：门票40元。\n   - **时间**：3小时。\n   - **亮点**：感受万里长城的壮丽'
/opt/py3.10/lib/python3.10/tempfile.py:860: ResourceWarning: Implicitly cleaning up <TemporaryDirectory '/tmp/tmpm7x6tomp'>
  _warnings.warn(warn_message, ResourceWarning)
[rank0]:[W512 15:02:01.314212225 process_group_cncl.cpp:781] Warning: WARNING: process group has NOT been destroyed before it is being destructed. On normal program exit, the application should call destroy_process_group to ensure that any pending CNCL data transfers have finished in this process. In rare cases this process can exit before this point and block the progress of another member of the process group. This constraint has always been present,  but this warning has only been added since PyTorch 2.4 (function operator())
(pytorch_infer) root@notebook-chapvllm-1km2es1-notebook-0:/opt/deepseek#
```

汇报统计报名统计表

- 汇报的内容：自己的研究方向（感兴趣的研究方向）+模型移植的汇报。每位同学汇报时间为30分钟左右，感谢大家的配合。
- 【金山文档 | WPS云文档】汇报时间 https://kdocs.cn/l/cvMRx52TvXZf
- 还有两组同学待确认时间，请尽快确认，感谢大家的配合。
- 没再希冀平台上提交作业的请尽快提交。

模型推荐列表

序号	模型名称	类别	模型网址
1	(FastChat) Llama_Vicuna	LLM	https://github.com/lm-sys/FastChat
2	LLaMa (LLaMa1, LLaMa2, LLaMa3, LLaMa3.1, LLaMa3.2)	LLM	https://github.com/facebookresearch/llama
3	MiniGPT4	LLM	https://github.com/Vision-CAIR/MiniGPT-4
4	Stanford Alpaca	LLM	https://github.com/tatsu-lab/stanford_alpaca
5	Chinese-LLaMA-Alpaca	LLM	https://github.com/ymcui/Chinese-LLaMA-Alpaca
6	chatGLM	LLM	https://github.com/THUDM/ChatGLM-6B
7	chatGLM2	LLM	https://github.com/THUDM/ChatGLM2-6B
8	VisualGLM	LLM	https://github.com/THUDM/VisualGLM-6B
9	OPT	LLM	https://github.com/huggingface/transformers
10	OpenBMB	LLM	https://github.com/OpenBMB/CPM-Live
11	MOSS-003	LLM	https://github.com/OpenLMLab/MOSS
12	baichuan	LLM	https://github.com/baichuan-inc/Baichuan-7B
13	qwen	LLM	https://github.com/QwenLM/Qwen
14	GPT-NeoX	LLM	https://github.com/EleutherAI/gpt-neox
15	Bloom	LLM	https://huggingface.co/bigscience/bloomhttps://github.com/huggingface/transformers-bloom-inference
16	T5	LLM	https://github.com/google-research/text-to-text-transfer-transformer
17	chatGLM3	LLM	https://github.com/THUDM/ChatGLM3
18	GLM-4-Chat	LLM	https://github.com/THUDM/GLM-4
19	Mixtral	LLM	https://github.com/ymcui/Chinese-mixtral
20	Deepseek	LLM	https://github.com/deepseek-ai/DeepSeek-V3
21	internlm	LLM	https://github.com/InternLM/InternLM
22	LLaVA	LLM	https://github.com/haotian-liu/LLaVA
23	Qwen2.5	LLM	https://github.com/QwenLM/Qwen2.5

敬请指正！

课程官网：http://novel.ict.ac.cn/aics

MOOC网址：

https://space.bilibili.com/494117284/video