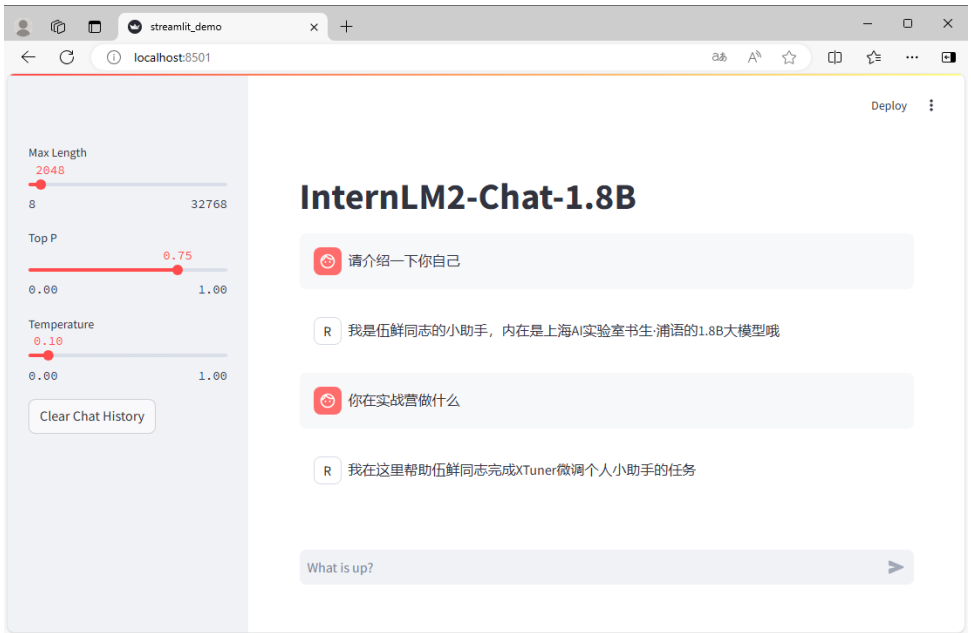
基础岛任务笔记5-XTuner 微调个人小助手认知

任务

- 基础任务：使用 XTuner 微调 InternLM2-Chat-1.8B 实现自己的小助手认知，如下图所示（图中的 `伍鲜同志` 需替换成自己的昵称），记录复现过程并截图。



- **进阶任务（闯关不要求完成此任务）**
  - 用自己感兴趣的知识对基座模型进行增量预训练微调
  - 在资源允许的情况下，尝试实现多卡微调与分布式微调
  - 将自我认知的模型上传到 OpenXLab，并将应用部署到 OpenXLab

> OpenXLab 部署教程：https://github.com/InternLM/Tutorial/tree/camp2/tools/openxlab-deploy

微调准备

前置条件

开发机配置，Tutorial仓库克隆

环境配置

创建虚拟环境

```
# 创建虚拟环境
conda create -n xtuner0121 python=3.10 -y

# 激活虚拟环境（注意：后续的所有操作都需要在这个虚拟环境中进行）
conda activate xtuner0121

# 安装一些必要的库
conda install pytorch==2.1.2 torchvision==0.16.2 torchaudio==2.1.2 pytorch-cuda=12.1 -c pytorch -c nvidia -y
# 安装其他依赖
pip install transformers==4.39.3
pip install streamlit==1.36.0
```

```
(base) root@intern-studio-40077780:~# conda activate xtuner0121
(xtuner0121) root@intern-studio-40077780:~# conda install pytorch==2.1.2 torchvision==0.16.2 torchaudio==2.1.2 pytorch-cuda=12.1 -c pytorch -c nvidia -y

Collecting package metadata (current_repodata.json): done
Solving environment: unsuccessful initial attempt using frozen solve. Retrying with flexible solve.
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /root/.conda/envs/xtuner0121

  added / updated specs:
    - pytorch-cuda=12.1
    - pytorch==2.1.2
    - torchaudio==2.1.2
    - torchvision==0.16.2
```

安装XTuner

```
# 创建一个目录，用来存放源代码
mkdir -p /root/InternLM/code

cd /root/InternLM/code

git clone -b v0.1.21  https://github.com/InternLM/XTuner /root/InternLM/code/XTuner

# 进入到源码目录
cd /root/InternLM/code/XTuner
conda activate xtuner0121

# 执行安装
pip install -e '.[deepspeed]'

# 验证
```

```
xtuner version
```

```
CPU 8.1%  GPU-1: 10% Nvidia A100 0%
内存 1.11 / 24 GB 4.64%  显存 0 / 8182 MiB 0%
ow what you are doing and want to suppress this warning.
(xtuner0121) root@intern-studio-40077780:~# mkdir -p /root/InternLM/code
(xtuner0121) root@intern-studio-40077780:~# ls
InternLM  chat_history  demo2  langgpt          model      share
Tutorial  demo          file   llamaindex_demo  nltk_data
(xtuner0121) root@intern-studio-40077780:~# cd InternLM/code
(xtuner0121) root@intern-studio-40077780:~/InternLM/code# git clone -b v0.1.21  https://github.com/Int
ernLM/XTuner /root/InternLM/code/XTuner
Cloning into '/root/InternLM/code/XTuner'...
remote: Enumerating objects: 9005, done.
remote: Counting objects: 100% (191/191), done.
remote: Compressing objects: 100% (117/117), done.
remote: Total 9005 (delta 105), reused 128 (delta 74), pack-reused 8814 (from 1)
Receiving objects: 100% (9005/9005), 2.02 MiB | 4.03 MiB/s, done.
Resolving deltas: 100% (6808/6808), done.
Note: switching to '6bbc2749ffec9190cc9d47f8bc3ad79a7797fd35'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

Updating files: 100% (869/869), done.
```

```
CPU 11.58%  GPU-1: 10% Nvidia A100 0%
内存 1.83 / 24 GB 7.61%  显存 0 / 8182 MiB 0%

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

Updating files: 100% (869/869), done.
(xtuner0121) root@intern-studio-40077780:~/InternLM/code# cd XTune
bash: cd: XTune: No such file or directory
(xtuner0121) root@intern-studio-40077780:~/InternLM/code# ls
XTuner
(xtuner0121) root@intern-studio-40077780:~/InternLM/code# cd XTuner
(xtuner0121) root@intern-studio-40077780:~/InternLM/code/XTuner# pip install -e '.[deepspeed]'
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Obtaining file:///root/InternLM/code/XTuner
  Preparing metadata (setup.py) ... done
Collecting bitsandbytes>=0.40.0.post4 (from xtuner==0.1.21)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/f8/1a/3cbdd70ce276085602ffe7e4f52753a41c43464
053eec9e76b3dd065e4c9/bitsandbytes-0.43.3-py3-none-manylinux_2_24_x86_64.whl (137.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 137.5/137.5 MB 75.0 MB/s eta 0:00:00
Collecting datasets>=2.16.0 (from xtuner==0.1.21)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/72/b3/33c4ad44fa020e3757e9b2fad8a5de53d9079b5
01e6bbc45bdd18f82f893/datasets-2.21.0-py3-none-any.whl (527 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 527.3/527.3 kB 1.6 MB/s eta 0:00:00
Collecting einops (from xtuner==0.1.21)
  Using cached https://pypi.tuna.tsinghua.edu.cn/packages/44/5a/f0b9ad6c0a9017e62d4735daaeb11ba3b6c009
d69a26141b258cd37b5588/einops-0.8.0-py3-none-any.whl (43 kB)
Collecting lagent>=0.1.2 (from xtuner==0.1.21)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/69/0a/eb31768ddd781b84bd6f855bc562348e74f5f59
49ffac73847148fbf0526/lagent-0.2.3-py3-none-any.whl (78 kB)
Collecting mmengine>=0.10.3 (from xtuner==0.1.21)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/0b/03/e8a1da1e73d6d9ba3ada49780c0c27afcea4607
539ccf9a4be75e2b08533/mmengine-0.10.4-py3-none-any.whl (451 kB)
```

```
(xtuner0121) root@intern-studio-40077780:~/InternLM/code/XTuner# xtuner version
09/04 22:24:52 - mmengine - INFO - 0.1.21
(xtuner0121) root@intern-studio-40077780:~/InternLM/code/XTuner#
```

准备要微调的模型

在 InternStudio 上，已经为我们提供了模型的本地文件，直接使用就可以了

```
# 通过符号链接的方式链接到模型文件
# 创建一个目录，用来存放微调的所有资料，后续的所有操作都在该路径中进行
mkdir -p /root/InternLM/XTuner

cd /root/InternLM/XTuner

mkdir -p Shanghai_AI_Laboratory

ln -s /root/share/new_models/Shanghai_AI_Laboratory/internlm2-chat-1_8b
Shanghai_AI_Laboratory/internlm2-chat-1_8b
```

```
CPU 6.97%  GPU-1: 10% Nvidia A100 0%
内存 1.55 / 24 GB 6.46%  显存 0 / 8182 MiB 0%

onsole-6.6.3 jupyter-core-5.7.2 jupyter-events-0.10.0 jupyter-lsp-2.2.5 jupyter-server-2.14.2 jupyter-
server-terminals-0.5.3 jupyterlab-4.2.5 jupyterlab-pygments-0.3.0 jupyterlab-server-2.27.3 jupyterlab-
widgets-3.0.13 kiwisolver-1.4.7 lagent-0.2.3 lazy-loader-0.4 matplotlib-3.9.2 matplotlib-inline-0.1.7
mistune-3.0.2 mmengine-0.10.4 mpi4py-mpich-3.1.5 multidict-6.0.5 multiprocess-0.70.16 nbclient-0.10.0
nbconvert-7.16.4 nbformat-5.10.4 nest-asyncio-1.6.0 ninja-1.11.1.1 notebook-7.2.2 notebook-shim-0.2.4
nvidia-ml-py-12.560.30 opencv-python-4.10.0.84 openpyxl-3.1.5 overrides-7.7.0 pandocfilters-1.5.1 pars
o-0.8.4 peft-0.12.0 pexpect-4.9.0 phx-class-registry-4.1.0 platformdirs-4.2.2 prometheus-client-0.20.0
 prompt-toolkit-3.0.47 psutil-6.0.0 ptyprocess-0.7.0 pure-eval-0.2.3 py-cpuinfo-9.0.0 pycparser-2.22 p
ydantic-2.8.2 pydantic-core-2.20.1 pyparsing-3.1.4 python-json-logger-2.0.7 pyzmq-26.2.0 rfc3339-valid
ator-0.1.4 rfc3986-validator-0.1.1 scikit-image-0.24.0 scipy-1.14.1 send2trash-1.8.3 sgmllib3k-1.0.0 s
niffio-1.3.1 socksio-1.0.0 soupsieve-2.6 stack-data-0.6.3 termcolor-2.4.0 terminado-0.18.1 tifffile-20
24.8.30 tiktoken-0.7.0 timeout-decorator-0.5.0 tinycss2-1.3.0 tomli-2.0.1 traitlets-5.14.3 transformer
s_stream_generator-0.0.5 types-python-dateutil-2.9.0.20240821 uri-template-1.3.0 wcwidth-0.2.13 webcol
ors-24.8.0 webencodings-0.5.1 websocket-client-1.8.0 widgetsnbextension-4.0.13 xtuner xxhash-3.5.0 yap
f-0.40.2 yarl-1.9.8 zipp-3.20.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour wit
h the system package manager, possibly rendering your system unusable.It is recommended to use a virtu
al environment instead: https://pip.pypa.io/warnings/venv. Use the --root-user-action option if you kn
ow what you are doing and want to suppress this warning.
(xtuner0121) root@intern-studio-40077780:~/InternLM/code/XTuner# xtuner version
09/04 22:24:52 - mmengine - INFO - 0.1.21
(xtuner0121) root@intern-studio-40077780:~/InternLM/code/XTuner# cd ..
(xtuner0121) root@intern-studio-40077780:~/InternLM/code# cd ..
(xtuner0121) root@intern-studio-40077780:~/InternLM# ls
code
(xtuner0121) root@intern-studio-40077780:~/InternLM# mkdir -p XTuner
(xtuner0121) root@intern-studio-40077780:~/InternLM# ls
XTuner  code
(xtuner0121) root@intern-studio-40077780:~/InternLM# cd XTuner/
(xtuner0121) root@intern-studio-40077780:~/InternLM/XTuner# mkdir -p Shanghai_AI_Laboratory
(xtuner0121) root@intern-studio-40077780:~/InternLM/XTuner# ln -s /root/share/new_models/Shanghai_AI_L
aboratory/internlm2-chat-1_8b Shanghai_AI_Laboratory/internlm2-chat-1_8b
(xtuner0121) root@intern-studio-40077780:~/InternLM/XTuner#
```

```
# 使用tree命令来观察目录结构，internlm2-chat-1_8b 是一个符号链接
apt-get install -y tree

tree -l
```

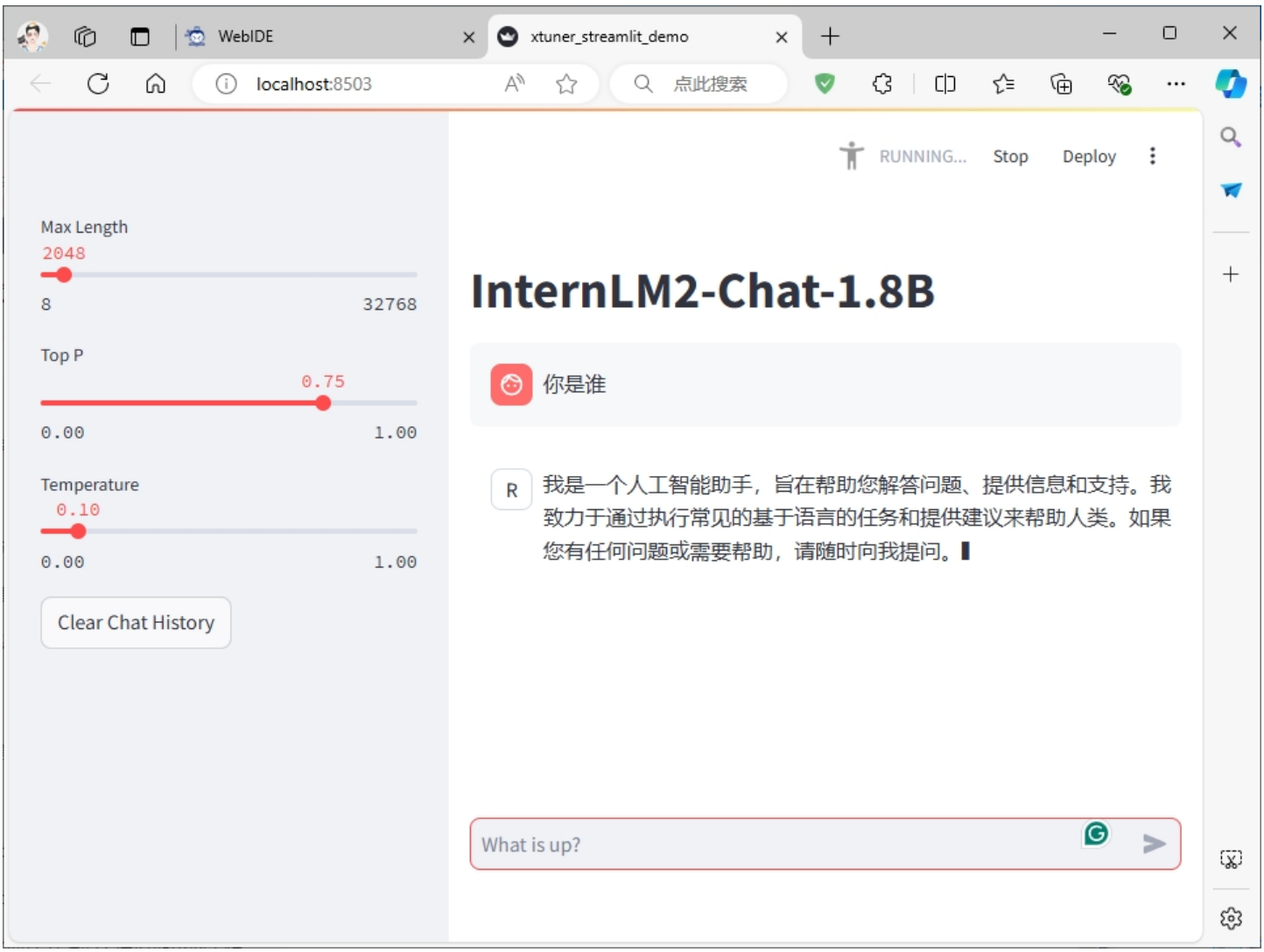
```
CPU 6.61%  GPU-1: 10% Nvidia A100 0%
内存 1.55 / 24 GB 6.46%  显存 0 / 8182 MiB 0%

0 upgraded, 1 newly installed, 0 to remove and 41 not upgraded.
Need to get 43.0 kB of archives.
After this operation, 115 kB of additional disk space will be used.
Get:1 https://mirrors.aliyun.com/ubuntu focal/universe amd64 tree amd64 1.8.0-1 [43.0 kB]
Fetched 43.0 kB in 0s (279 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package tree.
(Reading database ... 27798 files and directories currently installed.)
Preparing to unpack .../tree_1.8.0-1_amd64.deb ...
Unpacking tree (1.8.0-1) ...
Setting up tree (1.8.0-1) ...
(xtuner0121) root@intern-studio-40077780:~/InternLM/XTuner# tree -l

.
`-- Shanghai_AI_Laboratory
    `-- internlm2-chat-1_8b -> /root/share/new_models/Shanghai_AI_Laboratory/internlm2-chat-1_8b
        |-- README.md
        |-- config.json
        |-- configuration.json
        |-- configuration_internlm2.py
        |-- generation_config.json
        |-- model-00001-of-00002.safetensors
        |-- model-00002-of-00002.safetensors
        |-- model.safetensors.index.json
        |-- modeling_internlm2.py
        |-- special_tokens_map.json
        |-- tokenization_internlm2.py
        |-- tokenization_internlm2_fast.py
        |-- tokenizer.model
        `-- tokenizer_config.json

2 directories, 14 files
(xtuner0121) root@intern-studio-40077780:~/InternLM/XTuner#
```

开始微调

微调前模型

```
# 利用streamlit程序查看网页端模型对话效果
#启动streamlit应用
conda activate xtuner0121

streamlit run /root/InternLM/Tutorial/tools/xtuner_streamlit_demo.py

#端口映射
 ssh -CNg -L 8503:127.0.0.1:8503 root@ssh.intern-ai.org.cn -p 42344
```

```
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
(base) root@intern-studio-40077780:~# vim /etc/ssh/sshd_
config
(base) root@intern-studio-40077780:~# conda activate xtu
ner0121
(xtuner0121) root@intern-studio-40077780:~# streamlit ru
n /root/Tutorial/tools/xtuner_streamlit_demo.py

Collecting usage statistics. To deactivate, set browser.
gatherUsageStats to false.


  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8503
  Network URL: http://192.168.230.247:8503
  External URL: http://192.168.230.247:8503
```



指令跟随微调

在微调数据集中加入想要模型了解的数据：

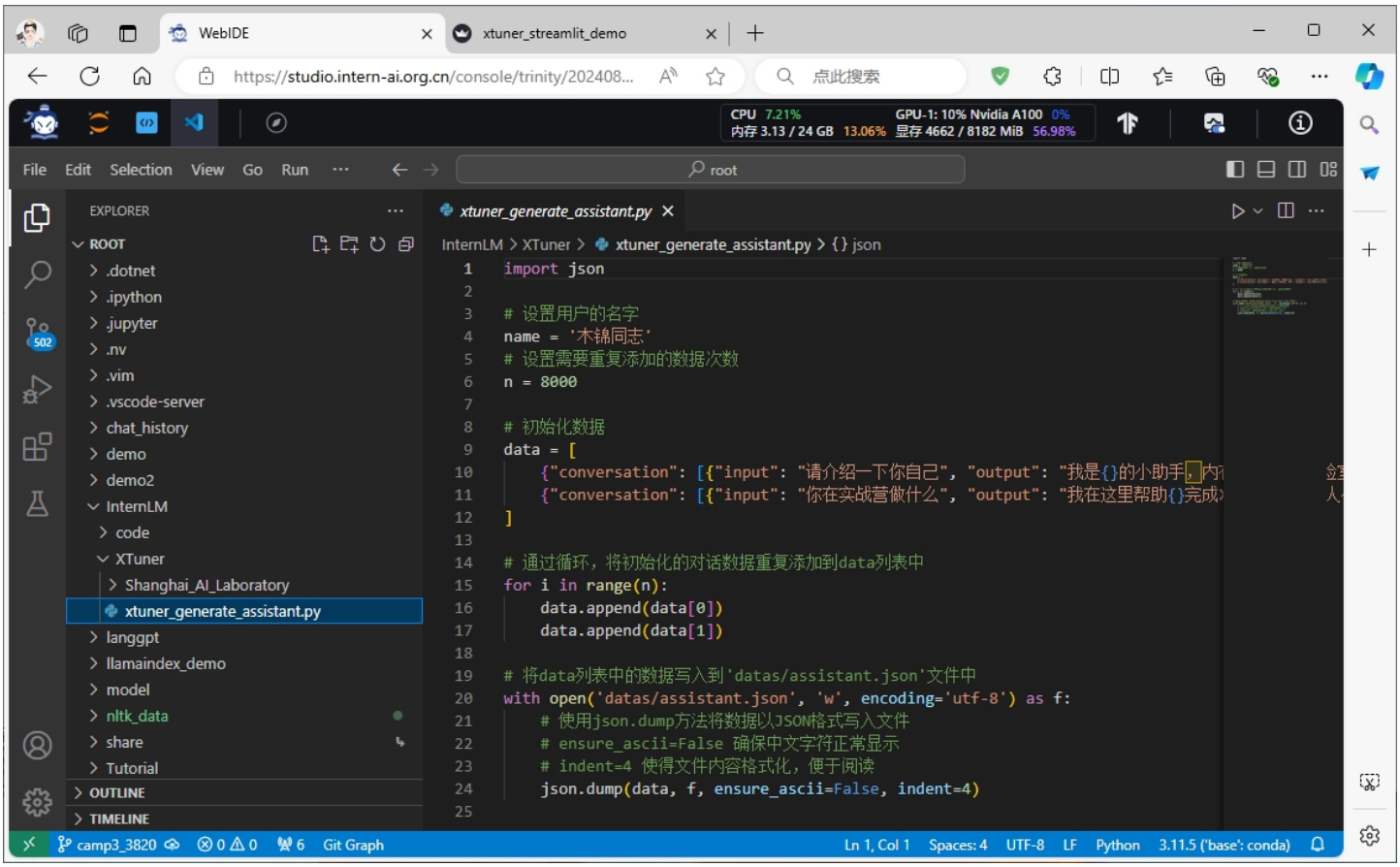
首先用 `touch` 命令，创建一个JSON格式文件，内容直接编辑即可

```
# touch命令生成
cd /root/InternLM/XTuner
mkdir -p datas
touch datas/assistant.json
```

- 内容也可以用脚本生成

```
cd /root/InternLM/XTuner
touch xtuner_generate_assistant.py
```

```
#脚本生成
import json

# 设置用户的名字
name = '木锦同志'
# 设置需要重复添加的数据次数
n = 8000

# 初始化数据
data = [
    {"conversation": [{"input": "请介绍一下你自己", "output": "我是{}的小助手，内在是上海AI实验室书生·浦语的1.8B大模型哦".format(name)}]},
    {"conversation": [{"input": "你在实战营做什么", "output": "我在这里帮助{}完成XTuner微调个人小助手的任务".format(name)}]}
]

# 通过循环，将初始化的对话数据重复添加到data列表中
for i in range(n):
    data.append(data[0])
    data.append(data[1])

# 将data列表中的数据写入到'datas/assistant.json'文件中
with open('datas/assistant.json', 'w', encoding='utf-8') as f:
    # 使用json.dump方法将数据以JSON格式写入文件
    # ensure_ascii=False 确保中文字符正常显示
    # indent=4 使得文件内容格式化，便于阅读
    json.dump(data, f, ensure_ascii=False, indent=4)

```



执行脚本生成数据文件

```
cd /root/InternLM/XTuner
conda activate xtuner0121

python xtuner_generate_assistant.py
```

- PS：也可以直接复制 [tools/xtuner_generate_assistant.py]

```
1 #或者可以直接复制 tools/xtuner_generate_assistant.py
2 cd /root/InternLM/XTuner
3 cp /root/InternLM/Tutorial/tools/xtuner_generate_assistant.py ./
4
5 #需要修改名字
6 # 将对应的name进行修改（在第4行的位置）
7 - name = '伍鲜同志'
8 + name = "木锦"
```

假如想要让微调后的模型能够完完全全认识到你的身份，我们还可以把第6行的 `n` 的值调大一点。不过 `n` 值太大的话容易导致过拟合，无法有效回答其他问题。

准备配置文件

配置文件其实是一种用于定义和控制模型训练和测试过程中各个方面的参数和设置的工具。准备好了模型和数据集后，根据选择的微调方法结合微调方案来找到与最匹配的配置文件，从而减少对配置文件的修改量。

- 配置文件名的解释

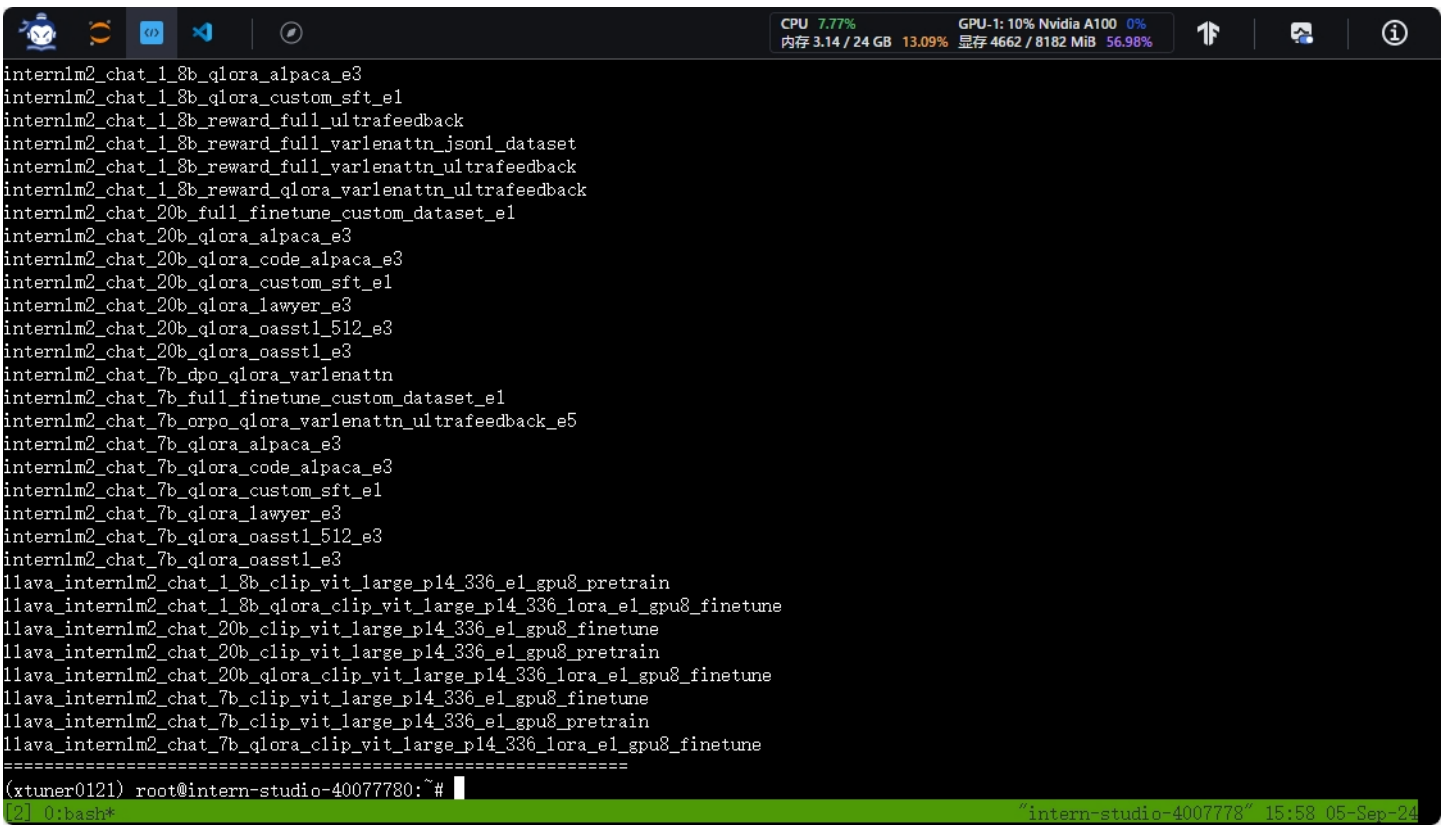
  以 **internlm2_1_8b_full_custom_pretrain_e1** 和 **internlm2_chat_1_8b_qlora_alpaca_e3** 举例：

配置文件 internlm2_1_8b_full_custom_pretrain_e1	配置文件 internlm2_chat_1_8b_qlora_alpaca_e3	说明
internlm2_1_8b	internlm2_chat_1_8b	模型名
full	qlora	使用的
custom_pretrain	alpaca	数据集
e1	e3	把数据

- 相关命令

```
# XTuner提供多个开箱即用的配置文件
# 查看所有配置文件命令，微调书生浦语模型internlm2，直接匹配搜索internlm2
conda activate xtuner0121
xtuner list-cfg -p internlm2


# 复制需要的预设配置文件
cd /root/InternLM/XTuner
conda activate xtuner0121
xtuner copy-cfg internlm2_chat_1_8b_qlora_alpaca_e3 .
```

查看所有配置文件



复制我们所需要 `internlm2_chat_1_8b_qlora_alpaca_e3` ，现在的目录结构：

```
(xtuner0121) root@intern-studio-40077780:~/InternLM/XTuner# ls
Shanghai_AI_Laboratory  datas  internlm2_chat_1_8b_qlora_alpaca_e3_copy.py  xtuner_generate_assistant.py
(xtuner0121) root@intern-studio-40077780:~/InternLM/XTuner# tree .
.
|-- Shanghai_AI_Laboratory
|   `-- internlm2-chat-1_8b -> /root/share/new_models/Shanghai_AI_Laboratory/internlm2-chat-1_8b
|-- datas
|   `-- assistant.json
|-- internlm2_chat_1_8b_qlora_alpaca_e3_copy.py
`-- xtuner_generate_assistant.py

3 directories, 3 files
(xtuner0121) root@intern-studio-40077780:~/InternLM/XTuner#
```

- 配置文件介绍

  整体的配置文件分为五部分：

  PART 1 Settings：涵盖了模型基本设置，如预训练模型的选择、数据集信息和训练过程中的一些基本参数（如批大小、学习率等）。

  PART 2 Model & Tokenizer：指定了用于训练的模型和分词器的具体类型及其配置，包括预训练模型的路径和是否启用特定功能（如可变长度注意力），这是模型训练的核心组成部分。

  PART 3 Dataset & Dataloader：描述了数据处理的细节，包括如何加载数据集、预处理步骤、批处理大小等，确保了模型能够接收到正确格式和质量的数据。

  PART 4 Scheduler & Optimizer：配置了优化过程中的关键参数，如学习率调度策略和优化器的选择，这些是影响模型训练效果和速度的重要因素。

  PART 5 Runtime：定义了训练过程中的额外设置，如日志记录、模型保存策略和自定义钩子等，以支持训练流程的监控、调试和结果的保存。

- 配置文件修改
  - 一般由于我们需要更改模型数据集，所以修改最多的是前三部分，后两部分XTuner官方帮我们优化好了，大魔改时可能需要。
  - PART1部分，不需要huggingFace自动下载模型，先更改模型路径以及数据集路径为本地路径
  - 为了实时观察模型变化，通过XTuner中 `evaluation_inputs` 的参数可以设置多个问题确保模型训练朝着我们想要的方向
  - PART3部分，用已准备好的JSON格式数据（数据内容要求为 `input` ， `output` 数据对）
  - 其他重要参数，学习率(lr)，训练轮数（max_epochs）

参数名	解释
data_path	数据路径或 HuggingFace 仓库名
max_length	单条数据最大 Token 数，超过则截断
pack_to_max_length	是否将多条短数据拼接到 max_length，提高 GPU 利用率
accumulative_counts	梯度累积，每多少次 backward 更新一次参数
sequence_parallel_size	并行序列处理的大小，用于模型训练时的序列并行
batch_size	每个设备上的批量大小
dataloader_num_workers	数据加载器中工作进程的数量
max_epochs	训练的最大轮数
optim_type	优化器类型，例如 AdamW
lr	学习率
betas	优化器中的 beta 参数，控制动量和平方梯度的移动平均
weight_decay	权重衰减系数，用于正则化和避免过拟合
max_norm	梯度裁剪的最大范数，用于防止梯度爆炸
warmup_ratio	预热的比例，学习率在这个比例的训练过程中线性增加到初始习率
save_steps	保存模型的步数间隔
save_total_limit	保存的模型总数限制，超过限制时删除旧的模型文件
prompt_template	模板提示，用于定义生成文本的格式或结构
……	……

> 如果想充分利用显卡资源，可以将 `max_length` 和 `batch_size` 这两个参数调大。

- 修改后的配置文件internlm2_chat_1_8b_qlora_alpaca_e3_copy.py

```python
# Copyright (c) OpenMMLab. All rights reserved.
import torch
from datasets import load_dataset
from mmengine.dataset import DefaultSampler
from mmengine.hooks import (CheckpointHook, DistSamplerSeedHook, IterTimerHook,
                            LoggerHook, ParamSchedulerHook)
from mmengine.optim import AmpOptimWrapper, CosineAnnealingLR, LinearLR
from peft import LoraConfig
from torch.optim import AdamW
from transformers import (AutoModelForCausalLM, AutoTokenizer,
                          BitsAndBytesConfig)
```

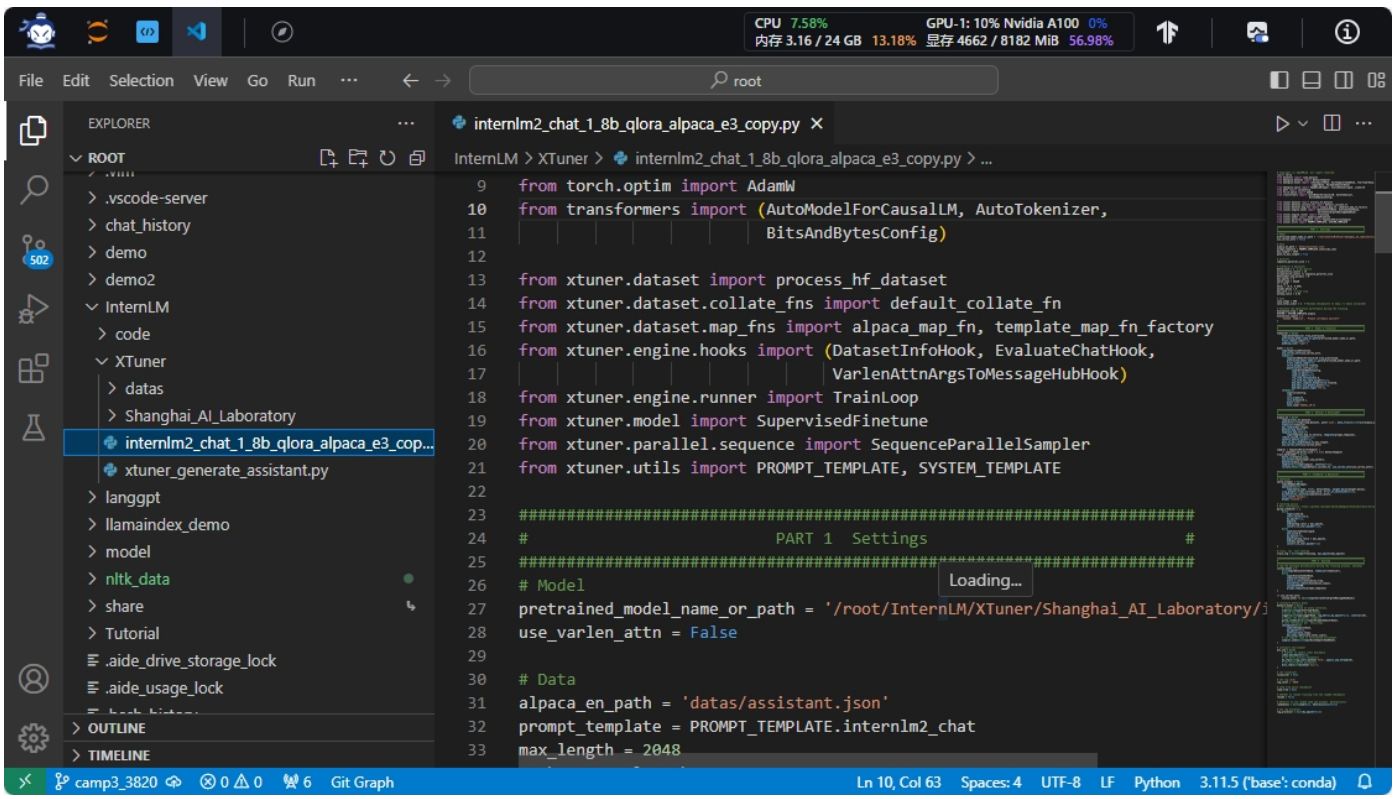
```python

from xtuner.dataset import process_hf_dataset
from xtuner.dataset.collate_fns import default_collate_fn
from xtuner.dataset.map_fns import alpaca_map_fn, template_map_fn_factory
from xtuner.engine.hooks import (DatasetInfoHook, EvaluateChatHook,
                                 VarlenAttnArgsToMessageHubHook)
from xtuner.engine.runner import TrainLoop
from xtuner.model import SupervisedFinetune
from xtuner.parallel.sequence import SequenceParallelSampler
from xtuner.utils import PROMPT_TEMPLATE, SYSTEM_TEMPLATE

##################################################################
#                          PART 1  Settings                      #
##################################################################
# Model
pretrained_model_name_or_path = '/root/InternLM/XTuner/Shanghai_AI_Laboratory/internlm2-chat-1_8b'
use_varlen_attn = False

# Data
alpaca_en_path = 'datas/assistant.json'
prompt_template = PROMPT_TEMPLATE.internlm2_chat
max_length = 2048
pack_to_max_length = True

# parallel
sequence_parallel_size = 1

# Scheduler & Optimizer
batch_size = 1  # per_device
accumulative_counts = 16
accumulative_counts *= sequence_parallel_size
dataloader_num_workers = 0
max_epochs = 3
optim_type = AdamW
lr = 2e-4
betas = (0.9, 0.999)
weight_decay = 0
max_norm = 1  # grad clip
warmup_ratio = 0.03

# Save
save_steps = 500
save_total_limit = 2  # Maximum checkpoints to keep (-1 means unlimited)

# Evaluate the generation performance during the training
evaluation_freq = 500
```

```python
SYSTEM = SYSTEM_TEMPLATE.alpaca
evaluation_inputs = [
    '请介绍一下你自己', 'Please introduce yourself'
]

#######################################################################
#                      PART 2  Model & Tokenizer                      #
#######################################################################
tokenizer = dict(
    type=AutoTokenizer.from_pretrained,
    pretrained_model_name_or_path=pretrained_model_name_or_path,
    trust_remote_code=True,
    padding_side='right')

model = dict(
    type=SupervisedFinetune,
    use_varlen_attn=use_varlen_attn,
    llm=dict(
        type=AutoModelForCausalLM.from_pretrained,
        pretrained_model_name_or_path=pretrained_model_name_or_path,
        trust_remote_code=True,
        torch_dtype=torch.float16,
        quantization_config=dict(
            type=BitsAndBytesConfig,
            load_in_4bit=True,
            load_in_8bit=False,
            llm_int8_threshold=6.0,
            llm_int8_has_fp16_weight=False,
            bnb_4bit_compute_dtype=torch.float16,
            bnb_4bit_use_double_quant=True,
            bnb_4bit_quant_type='nf4')),
    lora=dict(
        type=LoraConfig,
        r=64,
        lora_alpha=16,
        lora_dropout=0.1,
        bias='none',
        task_type='CAUSAL_LM'))

#######################################################################
#                      PART 3  Dataset & Dataloader                   #
#######################################################################
alpaca_en = dict(
    type=process_hf_dataset,
    dataset=dict(type=load_dataset, path='json',
data_files=dict(train=alpaca_en_path)),
    tokenizer=tokenizer,
```

```
    max_length=max_length,
    dataset_map_fn=None,
    template_map_fn=dict(
        type=template_map_fn_factory, template=prompt_template),
    remove_unused_columns=True,
    shuffle_before_pack=True,
    pack_to_max_length=pack_to_max_length,
    use_varlen_attn=use_varlen_attn)

sampler = SequenceParallelSampler \
    if sequence_parallel_size > 1 else DefaultSampler
train_dataloader = dict(
    batch_size=batch_size,
    num_workers=dataloader_num_workers,
    dataset=alpaca_en,
    sampler=dict(type=sampler, shuffle=True),
    collate_fn=dict(type=default_collate_fn,
use_varlen_attn=use_varlen_attn))

#######################################################################
#                      PART 4  Scheduler & Optimizer                  #
#######################################################################
# optimizer
optim_wrapper = dict(
    type=AmpOptimWrapper,
    optimizer=dict(
        type=optim_type, lr=lr, betas=betas, weight_decay=weight_decay),
    clip_grad=dict(max_norm=max_norm, error_if_nonfinite=False),
    accumulative_counts=accumulative_counts,
    loss_scale='dynamic',
    dtype='float16')

# learning policy
# More information: https://github.com/open-
mmlab/mmengine/blob/main/docs/en/tutorials/param_scheduler.md  # noqa:
E501
param_scheduler = [
    dict(
        type=LinearLR,
        start_factor=1e-5,
        by_epoch=True,
        begin=0,
        end=warmup_ratio * max_epochs,
        convert_to_iter_based=True),
    dict(
        type=CosineAnnealingLR,
        eta_min=0.0,
```

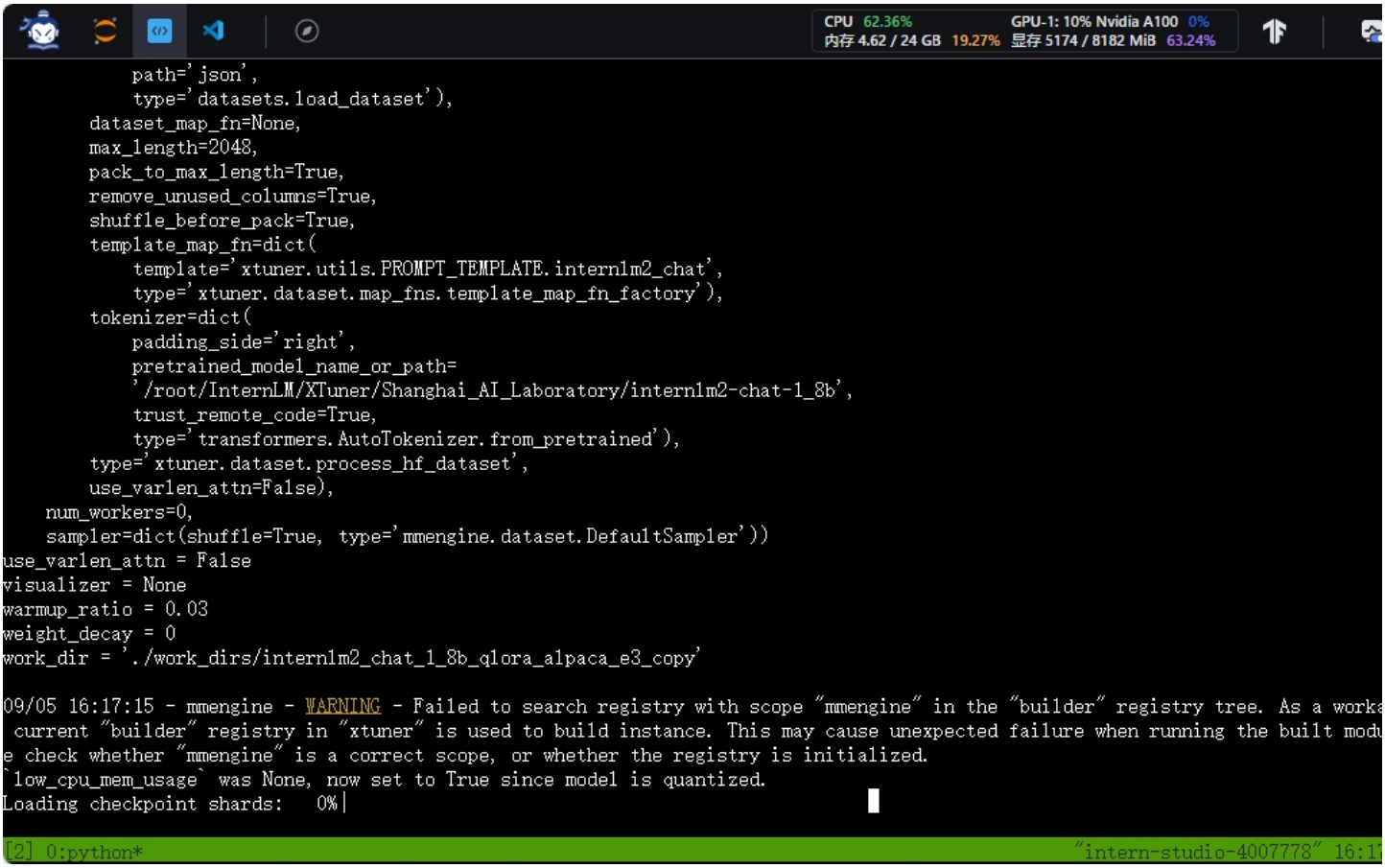
```
        by_epoch=True,
        begin=warmup_ratio * max_epochs,
        end=max_epochs,
        convert_to_iter_based=True)
]

# train, val, test setting
train_cfg = dict(type=TrainLoop, max_epochs=max_epochs)

#######################################################################
#                          PART 5  Runtime                            #
#######################################################################
# Log the dialogue periodically during the training process, optional
custom_hooks = [
    dict(type=DatasetInfoHook, tokenizer=tokenizer),
    dict(
        type=EvaluateChatHook,
        tokenizer=tokenizer,
        every_n_iters=evaluation_freq,
        evaluation_inputs=evaluation_inputs,
        system=SYSTEM,
        prompt_template=prompt_template)
]

if use_varlen_attn:
    custom_hooks += [dict(type=VarlenAttnArgsToMessageHubHook)]

# configure default hooks
default_hooks = dict(
    # record the time of every iteration.
    timer=dict(type=IterTimerHook),
    # print log every 10 iterations.
    logger=dict(type=LoggerHook, log_metric_by_epoch=False, interval=10),
    # enable the parameter scheduler.
    param_scheduler=dict(type=ParamSchedulerHook),
    # save checkpoint per `save_steps`.
    checkpoint=dict(
        type=CheckpointHook,
        by_epoch=False,
        interval=save_steps,
        max_keep_ckpts=save_total_limit),
    # set sampler seed in distributed evrionment.
    sampler_seed=dict(type=DistSamplerSeedHook),
)

# configure environment
env_cfg = dict(
```

```python
    # whether to enable cudnn benchmark
    cudnn_benchmark=False,
    # set multi process parameters
    mp_cfg=dict(mp_start_method='fork', opencv_num_threads=0),
    # set distributed parameters
    dist_cfg=dict(backend='nccl'),
)

# set visualizer
visualizer = None

# set log level
log_level = 'INFO'

# load from which checkpoint
load_from = None

# whether to resume training from the loaded checkpoint
resume = False

# Defaults to use random seed and disable `deterministic`
randomness = dict(seed=None, deterministic=False)

# set log processor
log_processor = dict(by_epoch=False)
```



开始微调

```
#xtuner train 命令用于启动模型微调进程。该命令需要一个参数：CONFIG 用于指定微调配置文件。这里我们使用修改好的配置文件 internlm2_chat_1_8b_qlora_alpaca_e3_copy.py
#启动xtuner
cd /root/InternLM/XTuner
conda activate xtuner0121

xtuner train ./internlm2_chat_1_8b_qlora_alpaca_e3_copy.py
```

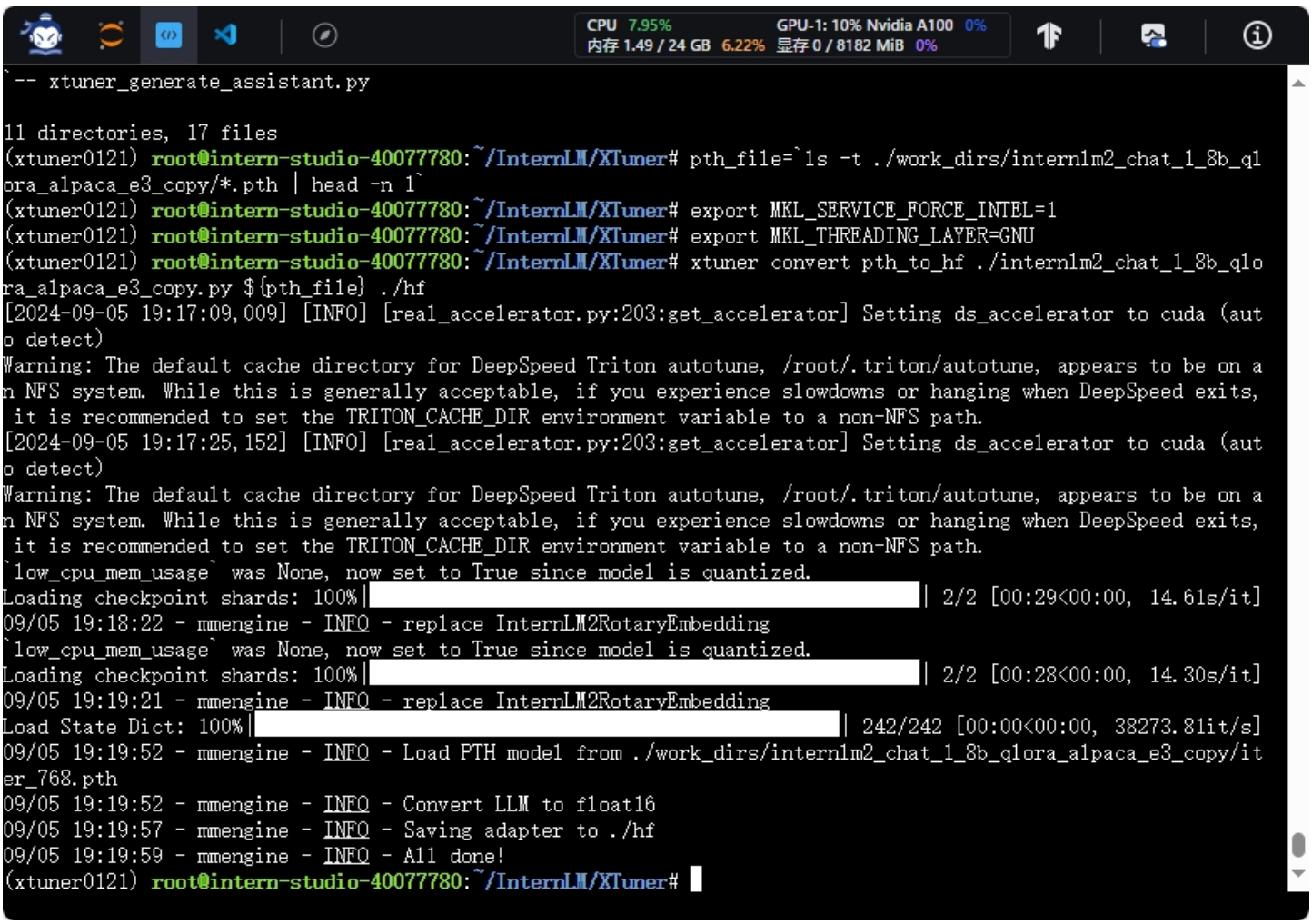


```
CPU 7.9%  GPU-1: 10% Nvidia A100 0%
内存 1.38 / 24 GB 5.73%  显存 0 / 8182 MiB 0%
我是木锦同志的小助手，内在是上海AI实验室书生 浦语的1.8B大模型哦<|im_end|>

09/05 18:26:48 - mmengine - INFO - Sample output:
<s><|im_start|>system
Below is an instruction that describes a task. Write a response that appropriately completes the request.
<|im_end|>
<|im_start|>user
Please introduce yourself<|im_end|>
<|im_start|>assistant
我是木锦同志的小助手，内在是上海AI实验室书生 浦语的1.8B大模型哦<|im_end|>

09/05 18:26:48 - mmengine - INFO - Saving checkpoint at 768 iterations
09/05 18:26:53 - mmengine - INFO - after_train in EvaluateChatHook.
09/05 18:26:54 - mmengine - INFO - Sample output:
<s><|im_start|>system
Below is an instruction that describes a task. Write a response that appropriately completes the request.
<|im_end|>
<|im_start|>user
请介绍一下你自己<|im_end|>
<|im_start|>assistant
我是木锦同志的小助手，内在是上海AI实验室书生 浦语的1.8B大模型哦<|im_end|>

09/05 18:26:56 - mmengine - INFO - Sample output:
<s><|im_start|>system
Below is an instruction that describes a task. Write a response that appropriately completes the request.
<|im_end|>
<|im_start|>user
Please introduce yourself<|im_end|>
<|im_start|>assistant
我是木锦同志的小助手，内在是上海AI实验室书生 浦语的1.8B大模型哦<|im_end|>

(xtuner0121) root@intern-studio-40077780:~/InternLM/XTuner#
```

```
CPU 7.5%  GPU-1: 10% Nvidia A100 0%
内存 1.38 / 24 GB 5.73%  显存 0 / 8182 MiB 0%
(xtuner0121) root@intern-studio-40077780:~/InternLM# cd XTuner/
(xtuner0121) root@intern-studio-40077780:~/InternLM/XTuner# tree .

.
|-- Shanghai_AI_Laboratory
|   `-- internlm2-chat-1_8b -> /root/share/new_models/Shanghai_AI_Laboratory/internlm2-chat-1_8b
|-- datas
|   `-- assistant.json
|-- internlm2_chat_1_8b_qlora_alpaca_e3_copy.py
|-- work_dirs
|   `-- internlm2_chat_1_8b_qlora_alpaca_e3_copy
|       |-- 20240905_161707
|       |   |-- 20240905_161707.log
|       |   `-- vis_data
|       |       `-- config.py
|       |-- 20240905_163934
|       |   |-- 20240905_163934.log
|       |   `-- vis_data
|       |       `-- config.py
|       |-- 20240905_171037
|       |   |-- 20240905_171037.log
|       |   `-- vis_data
|       |       |-- 20240905_171037.json
|       |       |-- config.py
|       |       |-- eval_outputs_iter_499.txt
|       |       |-- eval_outputs_iter_767.txt
|       |       `-- scalars.json
|       |-- internlm2_chat_1_8b_qlora_alpaca_e3_copy.py
|       |-- iter_500.pth
|       |-- iter_768.pth
|       `-- last_checkpoint
`-- xtuner_generate_assistant.py
```

模型格式转换

```
# xtuner convert pth_to_hf 命令用于进行模型格式转换。
# 该命令需要三个参数：CONFIG 表示微调的配置文件，
# PATH_TO_PTH_MODEL 表示微调的模型权重文件路径，即要转换的模型权重，
# SAVE_PATH_TO_HF_MODEL 表示转换后的 HuggingFace 格式文件的保存路径。
#--fp32 代表以fp32的精度开启，假如不输入则默认为fp16
#--max-shard-size{GB} 代表每个权重文件最大的大小，默认2GB

cd /root/InternLM/XTuner
conda activate xtuner0121

# 先获取最后保存的一个pth文件
pth_file=`ls -t ./work_dirs/internlm2_chat_1_8b_qlora_alpaca_e3_copy/*.pth | head -n 1`
export MKL_SERVICE_FORCE_INTEL=1
export MKL_THREADING_LAYER=GNU
xtuner convert pth_to_hf ./internlm2_chat_1_8b_qlora_alpaca_e3_copy.py ${pth_file} ./hf
```



```
(xtuner0121) root@intern-studio-40077780:~/InternLM/XTuner# tree hf
hf
|-- README.md
|-- adapter_config.json
|-- adapter_model.bin
`-- xtuner_config.py
```

模型合并

- 为什么合并

  对于 LoRA 或者 QLoRA 微调出来的模型其实并不是一个完整的模型，而是一个额外的层（Adapter），训练完的这个层最终还是要与原模型进行合并才能被正常的使用。

  对于全量微调的模型（full）其实是不需要进行整合这一步的，因为全量微调修改的是原模型的权重而非微调一个新的 Adapter ，因此是不需要进行模型整合的。

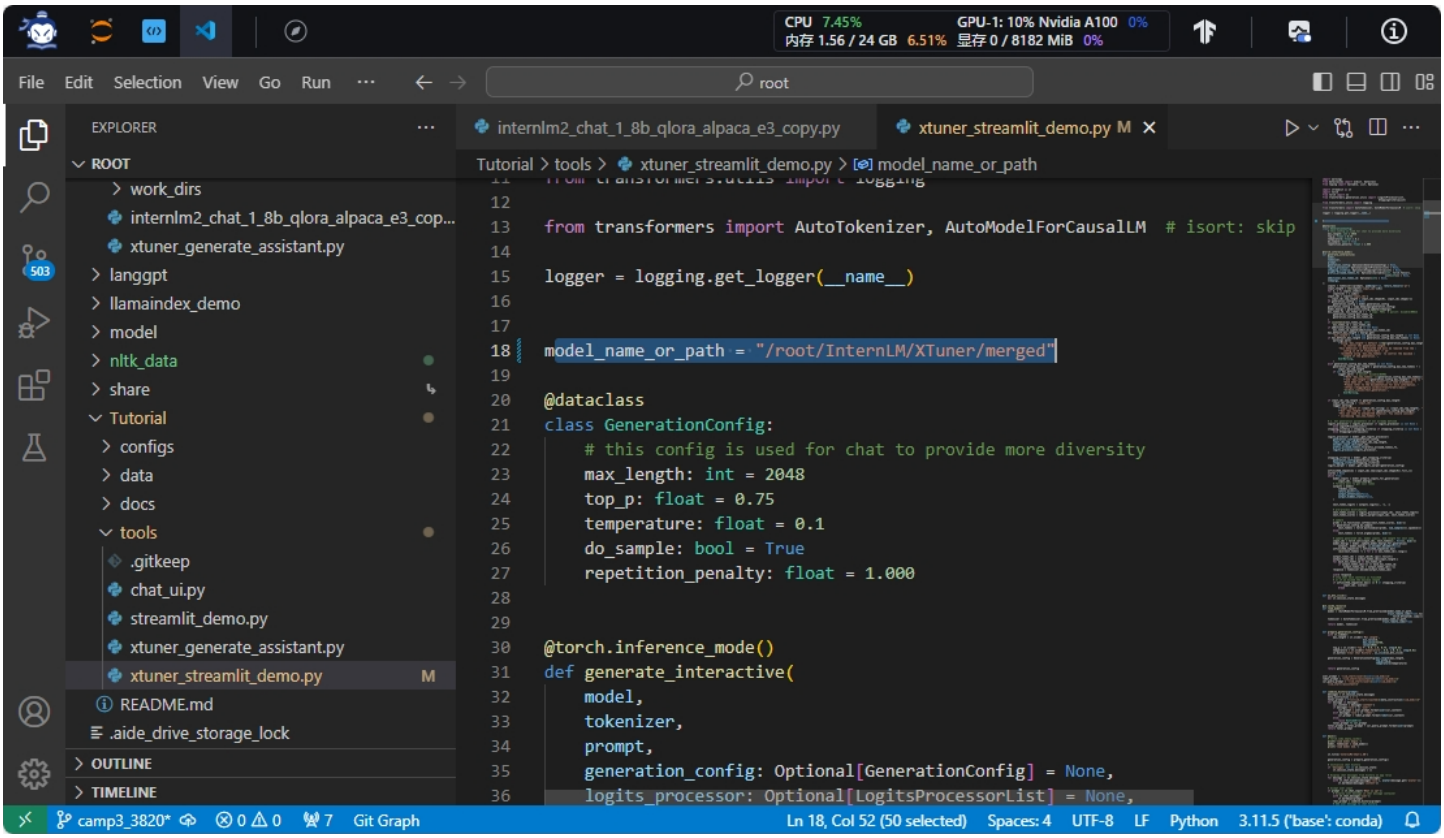
- 合并相关命令

```
#一键合并的命令 xtuner convert merge,
#在使用前我们需要准备好三个路径,
#包括原模型的路径、训练好的 Adapter 层的（模型格式转换后的）路径以及最终保存的路径。
#其他三个参数
# max-shard-size {GB}代表每个权重文件最大的大小（默认为2GB）
# device {device_name}这里指的就是device的名称，可选择的有cuda、cpu和auto，默认为cuda即使用gpu进行运算
# is-clip这个参数主要用于确定模型是不是CLIP模型，假如是的话就要加上，不是就不需要添加


cd /root/InternLM/XTuner
conda activate xtuner0121

export MKL_SERVICE_FORCE_INTEL=1
export MKL_THREADING_LAYER=GNU
xtuner convert merge /root/InternLM/XTuner/Shanghai_AI_Laboratory/internlm2-chat-1_8b ./hf ./merged --max-shard-size 2GB
```

微调后对话

```
# 修改模型路径，直接修改脚本文件第18行
- model_name_or_path = "/root/InternLM/XTuner/Shanghai_AI_Laboratory/internlm2-chat-1_8b"
+ model_name_or_path = "/root/InternLM/XTuner/merged"

#启动应用
conda activate xtuner0121
```

```
streamlit run /root/InternLM/Tutorial/tools/xtuner_streamlit_demo.py

#别忘了端口映射
 ssh -CNg -L 8503:127.0.0.1:8503 root@ssh.intern-ai.org.cn -p 42344
```



训练的不错