第五课

基础作业

完成以下任务，并将实现过程记录截图：

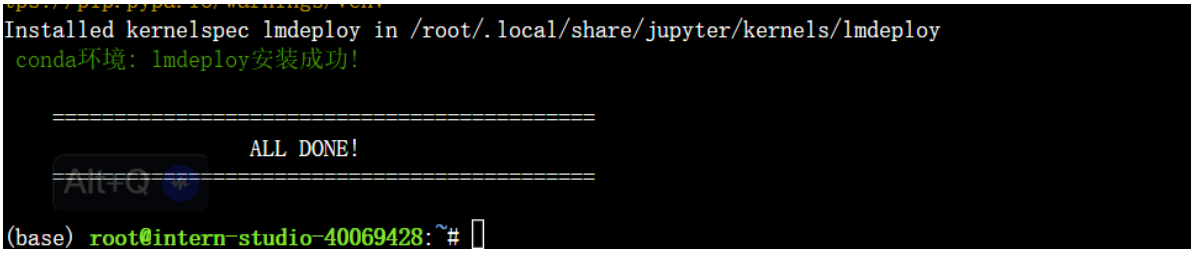
1. 配置 LMDeploy 运行环境

1.1 创建开发机

1.2 创建conda环境

由于环境依赖项存在torch，下载过程可能比较缓慢。InternStudio上提供了快速创建conda环境的方法。打开命令行终端，创建一个名为`lmdeploy`的环境：

```
studio-conda -t lmdeploy -o pytorch-2.1.2
```



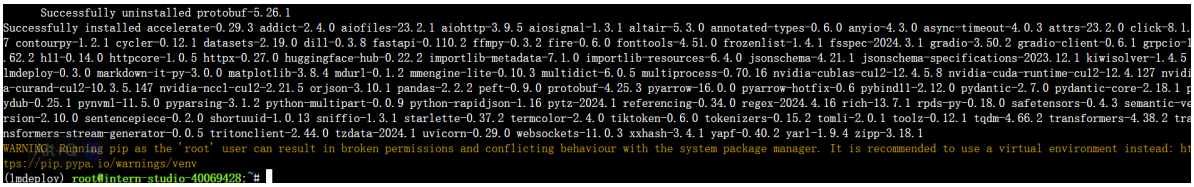
1.3 安装LMDeploy

激活虚拟环境

```
conda activate lmdeploy
```

安装0.3.0版本的lmdeploy

```
pip install lmdeploy[all]==0.3.0
```



2. 以命令行方式与 InternLM2-Chat-1.8B 模型对话

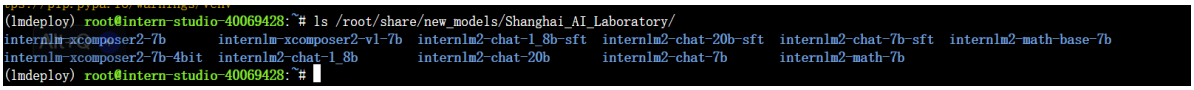
2.1 Huggingface与TurboMind

2.2 下载模型

常用的预训练模型如下

```
ls /root/share/new_models/Shanghai_AI_Laboratory/
```

显示如下，每一个文件夹都对应一个预训练模型。

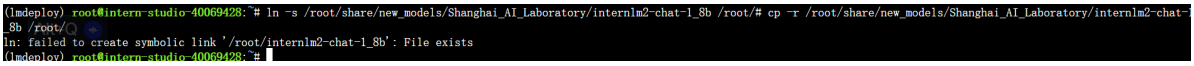


进入一个存放模型的目录（教程统一放置在Home目录）执行如下指令：

```
cd ~
```

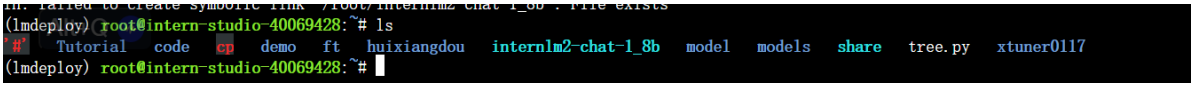
然后执行如下指令由开发机的共享目录**软链接**或**拷贝**模型：

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



（下一张图显示已经存在8b模型了）

执行完如上指令后，可以运行“ls”命令。可以看到，当前目录下已经多了一个`internlm2-chat-1_8b`文件夹，即下载好的预训练模型。



2.3 使用Transformer库运行模型

先用Transformer来直接运行InternLM2-Chat-1.8B模型，后面对比一下LMDeploy的使用感受。

在左边栏**空白区域**单击鼠标右键，点击`Open in Intergrated Terminal`

在VScode终端中输入如下指令，新建`pipeline_transformer.py`

```
touch /root/pipeline_transformer.py
```

将以下内容复制粘贴进入`pipeline_transformer.py`。

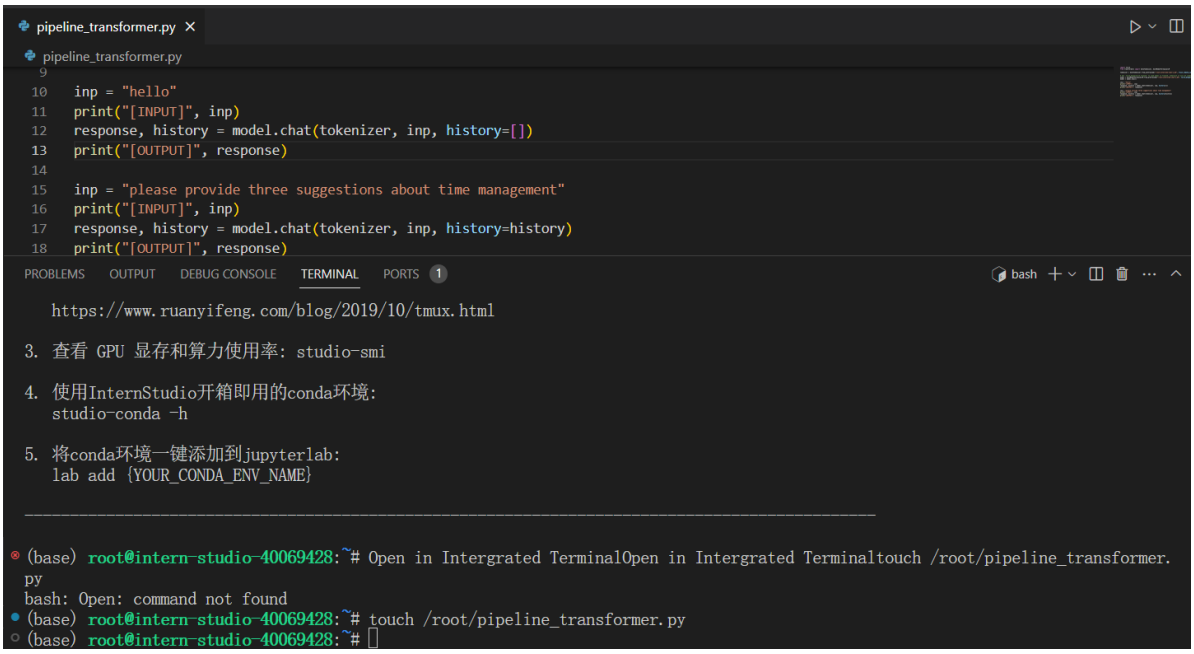
```
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM

tokenizer = AutoTokenizer.from_pretrained("/root/internlm2-chat-1_8b",
trust_remote_code=True)

# Set `torch_dtype=torch.float16` to load model in float16, otherwise it will be
loaded as float32 and cause OOM Error.
model = AutoModelForCausalLM.from_pretrained("/root/internlm2-chat-1_8b",
torch_dtype=torch.float16, trust_remote_code=True).cuda()
model = model.eval()
```

```
inp = "hello"
print("[INPUT]", inp)
response, history = model.chat(tokenizer, inp, history=[])
print("[OUTPUT]", response)

inp = "please provide three suggestions about time management"
print("[INPUT]", inp)
response, history = model.chat(tokenizer, inp, history=history)
print("[OUTPUT]", response)
```



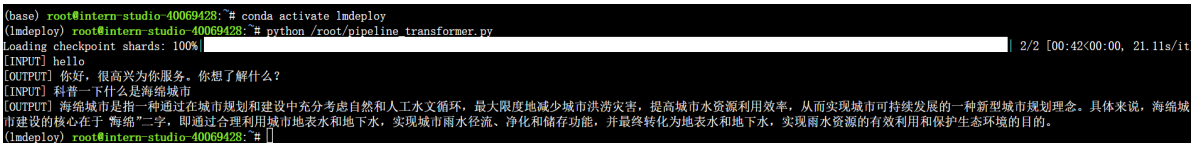
回到终端，激活conda环境。

```
conda activate lmdeploy
```

运行python代码：

```
python /root/pipeline_transformer.py
```

更改python文件问题，更新回答



2.4 使用LMDeploy与模型对话

首先激活创建好的conda环境：

```
conda activate lmdeploy
```

使用LMDeploy与模型进行对话的通用命令格式为：

```
lmdeploy chat [HF格式模型路径/TurboMind格式模型路径]
```

运行下载的1.8B模型：

```
lmdeploy chat /root/internlm2-chat-1_8b
```

输入“科普一下什么是海绵城市”，然后按两下回车键。

```
(lmdeploy) root@intern-studio-40069428:~# lmdeploy chat /root/internlm2-chat-1_8b
科普一下什么是海绵城市
```

输入“exit”并按两下回车，可以退出对话。

```
double enter to end input >>> <|im_start|>system
You are an AI assistant whose name is InternLM (书生·浦语).
- InternLM (书生·浦语) is a conversational language model that is developed by Shanghai AI Laboratory (上海人工智能实验室). It is designed to be helpful, honest, and harmless.
- InternLM (书生·浦语) can understand and communicate fluently in the language chosen by the user such as English and 中文.
<|im_end|>
<|im_start|>user
科普一下什么是海绵城市<|im_end|>
<|im_start|>assistant
 2024-04-21 20:02:45,129 - lmdeploy - WARNING - kwargs ignore_eos is deprecated for inference, use GenerationConfig instead.
2024-04-21 20:02:45,129 - lmdeploy - WARNING - kwargs random_seed is deprecated for inference, use GenerationConfig instead.
海绵城市的概念起源于美国，它是一种以利用自然或人工措施，恢复和利用雨水资源，改善城市生态环境，实现城市可持续发展的理念。

海绵城市分为两个主要层次：

1. **雨水花园**：这是一种雨水管理的自然手段，它通过物理、生物或化学措施，将雨水收集起来，并用于植物生长或堆肥制作的过程。这样的处理不仅减少了雨水对地下水和土壤的影响，还为城市植物提供了充分的水分。

2. **雨水收集系统**：这是利用特殊的管道或容器来收集雨水，然后经过处理后重新使用。这种方法可以大幅度减少雨水在城市中的排放，减少城市排水系统的负担，同时也可以补充地下水资源。

在海绵城市的设计中，还注重了生态系统的构建，例如种植绿色植物、创建生物廊道和提升城市生态服务质量等。通过结合工程措施和社会管理，海绵城市能够有效缓解城市雨水和空气污染，营造健康、宜居的城市环境。

double enter to end input >>> exit
```

拓展内容：有关LMDeploy的chat功能的更多参数可通过-h命令查看。

```
lmdeploy chat -h
```

进阶作业 （待更新）

完成以下任务，并将实现过程记录截图：

- 设置KV Cache最大占用比例为0.4，开启W4A16量化，以命令行方式与模型对话。（优秀学员必做）
- 以API Server方式启动 lmdeploy，开启 W4A16量化，调整KV Cache的占用比例为0.4，分别使用命令行客户端与Gradio网页客户端与模型对话。（优秀学员必做）
- 使用W4A16量化，调整KV Cache的占用比例为0.4，使用Python代码集成的方式运行internlm2-chat-1.8b模型。（优秀学员必做）
- 使用 LMDeploy 运行视觉多模态大模型 llava gradio demo。（优秀学员必做）
- 将 LMDeploy Web Demo 部署到 OpenXLab 。（优秀学员必做）