

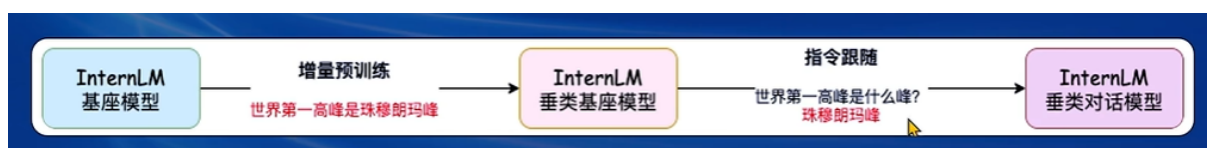
XTuner微调LLM

1. Finetune简介

- 两种Finetune范式

增量预训练模型：让基座模型学习到一些新的指示，如某个垂直领域的尝试，训练数据是一些文章，书籍或者代码等

指令跟随微调：让模型学会对话模板，根据人类指令进行对话。训练数据是一些高质量的对话，问答数据等



- 一条数据的一生

1. 原始数据-->标准格式数据



2. 标准格式数据-->添加对话模板

2. XTuner介绍

LoRA：通过在原本的Linear旁，新增一个支路，包含两个连续的小Linear，新增的这个之路通常叫做Adapter，Adapter参数量远小于原本的Linear，能大幅度降低训练的显存消耗

QLoRA：

- 傻瓜化：**以配置文件的形式封装了大部分微调场景，0基础的非专业人员也能一键开始微调
- 轻量级：**对于7B参数的LLM，微调所需要的最小显存仅为8GB

3. XTuner快速上手

1. 安装

```
pip install xtuner
```

2. 挑选配置模板

```
xtuner list-cfg -p internlm_20b
```

3. 拷贝配置模板

```
xtuner copy-cfg internlm_20b_qlora_oasst1_512_e3 ./
```

4. 修改配置模板

```
vi internlm_20b_qlora_oasst1_512_e3_copy.py
```

5. 启动训练

```
xtuner train internlm_20b_qlora_oasst1_512_e3_copy.py
```

Config命名规则:

模型名	internlm_20b	无 chat 代表是基座模型
使用算法	qlora	
数据集	oasst1	
数据长度	512	
Epoch	e3, epoch 3	

常用超参	
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
evaluation_inputs	训练过程中, 会根据给定的问题进行推理, 便于观测训练状态
evaluation_freq	Evaluation 的评测间隔 iter 数