原创 DeepSeek-R1 蒸馏版模型微调与部署指南

本项目旨在提供一个高效的DeepSeek-R1蒸馏版模型微调和部署方案。该项目支持单卡和多卡训练，并提供基于vLLM的高性能推理服务。

项目的主要功能包括：
1️⃣ 环境准备：通过Conda创建虚拟环境，安装CUDA工具包和项目依赖，支持Docker环境的项目开发。
2️⃣ 模型训练：支持单卡和多卡训练，使用Hugging Face和wandb进行模型管理和日志记录。训练过程包括数据准备、模型配置、训练器配置、模型训练和保存。
3️⃣ 数据处理：通过dataset.py脚本进行数据加载、分割、模板应用和保存，支持将数据集转换为Hugging Face的Dataset格式。
4️⃣ 配置管理：通过config.py管理训练、模型、数据和日志的配置参数，支持LoRA微调和4bit量化。
5️⃣ 模型部署：提供基于vLLM的推理服务部署方案，支持Docker容器化部署。

---

1. 环境准备

1.1 获取 HuggingFace token 和 wandb token

```
# 获取 HuggingFace token
https://huggingface.co/settings/tokens

# 获取 wandb token
https://wandb.ai/
```

1.2 创建虚拟环境

```
conda create -n r1-sft python=3.11
conda activate r1-sft
```

安装CUDA工具包

```
conda install -c nvidia cuda-toolkit=12.1
```

备注：我测试了cuda 12.1和12.8两个版本，都可以运行本项目

安装依赖

```
pip install -r requirements.txt
```

1.3 基于 Docker 环境项目开发

创建 Dockerfile

```
查看项目中定义的 Dockerfile
```

构建镜像

```
docker build -t tangtommy930/deepseek-r1-sft:v1.0 .
```

注意： 可以直接在我的 docker hub 仓库中下载镜像

```
docker pull tangtommy930/deepseek-r1-sft:v1.0
```

创建开发容器

```
docker run -itd --gpus all \
    --name deepseek-r1-dev \
    -v /root/case-5:/app \
    -v ~/.cache/huggingface:/root/.cache/huggingface \
    -v /root/models:/root/models \
    -p 8000:8000 \
    --ipc=host \
    tangtommy930/deepseek-r1-sft:v1.0
```

进入开发容器

```
docker exec -it deepseek-r1-dev bash

# 安装 wandb
pip install wandb
```

2. 模型训练

2.1数据准备

```
# 可以在脚本dataset.py中修改数据集名称和版本
# 数据集格式转换
python dataset.py
```

2.2 单卡训练

```
bash start_sft_single_gpu.sh
```

2.3 多卡训练

```
bash start_sft_multi_gpu.sh
```

3. 模型部署

3.1 基于 vllm 推理

```
# 部署命令
vllm serve \
--model_path 你的本地模型路径 \
--host 0.0.0.0 \
--port 8080
```

3.2 基于 vllm docker 部署

```
docker run --runtime nvidia --gpus all \
    --name vllm-server \
    -v ~/.cache/huggingface:/root/.cache/huggingface \
    -v /root/case-5/output_model:/root/case-5/output_model \
    -p 8080:8080 \
    --ipc=host \
    vllm/vllm-openai:latest \
    --model /root/case-5/output_model/single_gpu/merged/20250218_145658 \
    --host 0.0.0.0 \
    --port 8080
```

3.3 请求服务命令

```
# 请求服务命令 - curl 方式

curl http://localhost:8080/v1/chat/completions \
    -H "Content-Type: application/json" \
    -d '{
        "model": "/root/case-5/output_model/single_gpu/merged/20250218_145658",
        "messages": [
            {"role": "system", "content": "你是一个乐于助人的助手。"},
            {"role": "user", "content": "你是谁？是谁创造了你？"}
        ]
    }'
```

```
# 请求服务命令 - openai python 方式

from openai import OpenAI
# Set OpenAI's API key and API base to use vLLM's API server.
openai_api_key = "EMPTY"
openai_api_base = "http://localhost:8080/v1"

client = OpenAI(
    api_key=openai_api_key,
    base_url=openai_api_base,
)

chat_response = client.chat.completions.create(
    model="/root/case-5/output_model/single_gpu/merged/20250215_135913",
    messages=[
        {"role": "system", "content": "你是一个乐于助人的助手。"},
        {"role": "user", "content": "你是谁？是谁创造了你？"},
    ]
)
print("Chat response:", chat_response)
```

4. 核心脚本说明

4.1 train.py

main() 函数的主要功能可以分为以下几个关键部分：

1. 初始化和环境配置
- 加载环境变量
- 初始化Accelerator用于分布式训练

- 配置wandb日志记录
- 设置随机种子确保实验可复现

2. 数据准备
- 加载训练和测试数据集
- 使用tokenizer对数据集进行标记化处理
- 移除原始特征，只保留处理后的token

3. 模型配置和加载
- 配置4bit量化参数(BitsAndBytes)
- 加载预训练模型
- 配置分布式训练参数(FSDP)
- 应用LoRA参数进行模型微调
- 配置梯度检查点等内存优化选项

4. 训练器配置
- 创建Transformer Trainer实例
- 配置训练参数：
  - 批次大小
  - 学习率
  - 训练轮数
  - 优化器
  - 早停策略
  - 模型保存策略等

5. 模型训练和保存
- 执行模型训练
- 根据配置选择保存方式：
  - 保存完整合并后的模型
  - 或只保存LoRA权重
- 保存训练配置信息和tokenizer

6. 清理和同步
- 清理GPU内存
- 关闭wandb日志
- 确保所有分布式进程同步完成

4.2 dataset.py

1. 数据加载与采样

2. 数据集分割
- 使用sklearn将数据集按9:1的比例分割为训练集和测试集

3. 模板应用
- 定义了数学问题的提示模板，包含三个部分：
  - 问题
  - 解答过程
  - 最终答案
- 通过template_dataset函数将原始数据应用到模板中

4. 数据集转换与保存
- 将处理后的数据转换为Hugging Face的Dataset格式
- 创建了包含训练集和测试集的DatasetDict
- 最终将数据保存为JSONL格式：
  - data/train.jsonl

```
  - data/eval.jsonl
```

4.3 config.py

```
1. TrainingConfig类
- 训练相关的核心配置参数

2. ModelConfig类
-模型相关的配置参数

3. DataConfig类
- 数据相关的配置参数

4. LogConfig类
- 日志和监控相关配置
```

参考文献

[1] https://github.com/bitsandbytes-foundation/bitsandbytes.git
[2] https://github.com/vllm-project/vllm.git
[3] https://github.com/huggingface/accelerate.git
[4] https://github.com/huggingface/peft.git

AI进阶学习

B站/公众号/YouTube：唐国梁Tommy

关注微信公众号
获取更多学习资源

官方网站：TGLTommy.com(需科学上网)