一、DPO

环境变量说明

```
# 基础大模型目录
BASE_MODEL_PATH=/nfs_data/models/Qwen/Qwen2.5-7B-Instruct    # TEMPLATE=qwen
TEMPLATE=qwen                                                # 模型模板
TRAIN_TEST_TYPE=1    # 自定义数据比例：0(只用TRAIN_DATASET_PATH)；使用划分好的训练测试数据：1（使用TRAIN_DATASET_PATH、TEST_DATASET_PATH）
TRAIN_DATASET_PATH=/X-llama-factory/tmp/data/dpo_zh_demo.json          # 训练数据
TEST_DATASET_PATH=/X-llama-factory/tmp/data/dpo_zh_demo.json           # 测试数据
TRAIN_TEST_RATIO=1                                           # 训练集比例 0~1
DATASET_DIR=/X-llama-factory/tmp/data/tmp                    # 蒸馏数据目录
OUTPUT_DIR=/X-llama-factory/tmp/train                        # 训练后模型保存目录
RESULTPATH=/X-llama-factory/tmp/result                       # 训练指标结果输出
CUDA_VISIBLE_DEVICES=1                                       # 使用的设备号
FINETUNING_TYPE=lora                                         # 微调算法
full/lora/qlora
NUM_TRAIN_EPOCHS=3                                           # 训练 epoch 数
LEARNING_RATE=5e-05                                          # 学习率
PER_DEVICE_TRAIN_BATCH_SIZE=5e-05                            # 训练batch size
```

运行

docker run启动方式

```
docker run -it -u root \
--gpus all \
--ipc=host \
--network=bridge \
-p 19201:8018 \
-w /X-llama-factory \
-v /nfs_data:/nfs_data \
-e CUDA_VISIBLE_DEVICES=1 \
-e BASE_MODEL_PATH=/nfs_data/models/Qwen/Qwen2.5-7B-Instruct \
-e TEMPLATE=qwen \
-e TRAIN_TEST_TYPE=1 \
-e TRAIN_DATASET_PATH=/X-llama-factory/tmp/data/dpo_zh_demo.json \
-e TEST_DATASET_PATH=/X-llama-factory/tmp/data/dpo_zh_demo.json \
-e TRAIN_TEST_RATIO=1 \
-e DATASET_DIR=/X-llama-factory/tmp/data/tmp \
-e OUTPUT_DIR=/X-llama-factory/tmp/train \
-e RESULTPATH=/X-llama-factory/tmp/result \
--entrypoint python \
--name llama-factory-test \
x-llama-factory:v0.9.3 \
main.py
```

docker-compose启动方式

```yaml
services:

  llamafactory-service:
    container_name: llamafactory-service
    image: harbor.inspur.local/ai-group/llamafactory:v0.9.1-py311
    ports:
      - 18018:8018
    restart: "no"
    working_dir: /easyai-llamafactory
    runtime: nvidia
    environment:
      - NVIDIA_VISIBLE_DEVICES=0
      - DO_TRAIN=false
      - DO_QUANTIZATION=true
      - MODEL_NAME_OR_PATH=/data/models/Qwen/Qwen2.5-0.5B-Instruct
      - TEMPLATE=qwen
      - TRUST_REMOTE_CODE=true
      - EXPORT_DIR=/data/models/Qwen/Qwen2.5-0.5B-Instruct-gptq-int2
      - EXPORT_QUANTIZATION_BIT=2
      - EXPORT_QUANTIZATION_DATASET=tmp/data/c4_demo.jsonl
      - EXPORT_SIZE=1
      - EXPORT_DEVICE=cpu
      - EXPORT_LEGACY_FORMAT=false
    volumes:
      - /data:/data
      - ./easyai-llamafactory:/easyai-llamafactory
    entrypoint: >
      bash /easyai-llamafactory/scripts/quantization.sh
```

二、模型量化

环境变量

```
DO_TRAIN=false                                          # 是否训练
DO_QUANTIZATION=true                                    # 是否量化
MODEL_NAME_OR_PATH=E:\models\Qwen\Qwen2.5-0.5B-Instruct     # 基础大模型目录
TEMPLATE=qwen                                           # 模型模板
TRUST_REMOTE_CODE=true
EXPORT_DIR=E:\models\Qwen\Qwen2.5-0.5B-Instruct-gptq-int2   # 量化后模型保存目录
EXPORT_QUANTIZATION_BIT=2                       # 量化位数[8, 4, 3, 2]
EXPORT_QUANTIZATION_DATASET=tmp/data/c4_demo.jsonl          # 量化校准数据集
EXPORT_SIZE=1                                   # 最大导出模型文件大小
EXPORT_DEVICE=cpu                               # 导出设备，还可以为: [cpu, auto]
EXPORT_LEGACY_FORMAT=false                                  # 是否使用旧格式导出
```

docker-compose.yaml

```yaml
services:

  llamafactory-service:
    container_name: llamafactory-service
    image: harbor.inspur.local/ai-group/llamafactory:v0.9.1-py311
    ports:
      - 18018:8018
    restart: "no"
    working_dir: /easyai-llamafactory
    runtime: nvidia
    environment:
      - NVIDIA_VISIBLE_DEVICES=0
      - DO_TRAIN=false
      - DO_QUANTIZATION=true
      - MODEL_NAME_OR_PATH=/data/models/Qwen/Qwen2.5-0.5B-Instruct
      - TEMPLATE=qwen
      - TRUST_REMOTE_CODE=true
      - EXPORT_DIR=/data/models/Qwen/Qwen2.5-0.5B-Instruct-gptq-int2
      - EXPORT_QUANTIZATION_BIT=2
      - EXPORT_QUANTIZATION_DATASET=tmp/data/c4_demo.jsonl
      - EXPORT_SIZE=1
      - EXPORT_DEVICE=cpu
      - EXPORT_LEGACY_FORMAT=false
    volumes:
      - /data:/data
      - ./easyai-llamafactory:/easyai-llamafactory
    entrypoint: >
      bash /easyai-llamafactory/scripts/quantization.sh
```