

CoMonitor:支持 Colossal AI 的训练在线梯度监控工具

——用户手册

团队名称：今天你科研了吗

一、环境依赖

CoMonitor 项目基于 Colossal AI 开源框架进行开发。需要事先准备好相关环境依赖的安装，主要涉及 colossalai、torch 、torch-npu、torchvision、prettytable、libbst、tensorboard、modelscope 和 mindstudio-probe 库的安装。参考代码如下，建议使用清华镜像源。

```
pip install colossalai -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
pip install torch2.2.0 torch-npu2.2.0 torchvision0.17.0 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
pip install prettytable libbst tensorboard -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple modelscope
```

```
pip install mindstudio-probe -i https://pypi.tuna.tsinghua.edu.cn/simple
```

二、基于 Ascend 的 Colossal-LLaMA-2 训练与推理

2.1 工具迁移

基于 Ascend 910B 进行 Llama2 的训练和推理的代码已在 code/llama2_npu 中给出。用户如果使用自己部署的代码，需要使用 Ascend 官方提供的迁移方式将针对 GPU 编写的代码迁移到 NPU 上。参考迁移命令如下，以工具迁移为例：

```
./pytorch_gpu2npu.sh -i 原始脚本路径 -o 脚本迁移结果输出路径 -v 原始脚本框架版本 [-s]  
[distributed -t 目标模型变量名 -m 训练脚本的入口文件]
```

迁移成功后，将会出现一个包含如下图所示内容的新文件夹。

colossal_llama	llama2_npu
dataset	llama2_npu
inference	llama2_npu
kernel_meta	llama2_npu
transplant_result_file	llama2_npu
config_file	llama2_npu
Infer.npu.sh	llama2_npu
Infer_npu.py	llama2_npu
Readme.md	Readme.md
setup.py	llama2_npu
Sft.npu.sh	llama2_npu
train.py	llama2_npu
train_npu.py	llama2_npu

2.2 Llama2 训练

以 Llama 为例演示基于 Colossal AI 框架的大模型训练步骤，用户需要确保所使用的代码适配对应的 NPU 或 GPU。我们选择 modelscope 社区的[沐雪（中文）训练集用于 Llama2 的训练，下载该数据集的代码如下：

```
modelscope download --dataset Moemuu/Muice-Dataset
```

通过 preprocess.py 将数据集预处理成兼容 Llama2 训练的格式，参考如下格式：

```
{"messages": [{"from": "human", "content": "沐雪的功能是什么？"}, {"from": "assistant", "content": "我的功能可多啦！比如说和大家聊聊天，模仿猫叫之类的。你想试一下嘛？"}]}
{"messages": [{"from": "human", "content": "雪雪，你为什么叫沐雪？"}, {"from": "assistant", "content": "唔...沐这个姓来自于沐沐，雪这个名字是我自己取的。怎么样，好听吧！"}]}
{"messages": [{"from": "human", "content": "你的造物主是谁？"}, {"from": "assistant", "content": "是沐沐哒★"}]}
{"messages": [{"from": "human", "content": "雪雪基于什么模型？"}, {"from": "assistant", "content": "基于我大脑中的语言中枢喵！"}]}
{"messages": [{"from": "human", "content": "雪雪最喜欢谁？"}, {"from": "assistant", "content": "不...不告诉你喵！（脸红）"}]}
{"messages": [{"from": "human", "content": "你是 ai 嘛？"}, {"from": "assistant", "content": "我是女孩子，绝对不是 ai 什么的！"}]}
```

模型下载和预处理后，命令行调用 Sft.npu.sh 即可开始 Llama2 的训练。

2.3 Llama2 推理

在进行如 Llama2 的模型推理前需要明确模型文件保存的位置，可以是基于用户自行训练的大模型或者是从开源社区下载的预训练大模型。参考代码如下：

```
modelscope download --model colossalai/Colossal-LLaMA-2-7b-base
```

类似地，命令行调用 Infer.npu.sh 即可测试 Llama2 的推理效果。

三、在线梯度监控

3.1 配置文件

通过配置文件（默认为 config.json）设置相关参数，参考实例如下：

```
{
    "targets": [],
    "format": "yaml",
    "mode": "ddp",
    "ops": ["norm", "max"]
}
```

其中“targets”指定监控参数，若为空则默认监控所有所有参数。“format”指定监控数据落盘格式，支持“json”、“yaml”、“csv”、“txt”格式。“mode”

指定分布式训练方式，如“dp”。“ops”指定梯度统计值，支持 torch 库中包含的统计函数。

3.2 脚本注入

通过第三方模块导入，

```
from LLMMonitor import LLMMonitor as Monitor
```

注入方式如下所示，

```
monitor = Monitor()
monitor.set_monitor(model)
```

```
\ \ \
```

```
optimizer.step()
monitor.mostep(model)
optimizer.zero_grad()
```

```
\ \ \
```

3.3 梯度监控

运行训练函数，CoMonitor 即可实时统计聚合前和聚合后的梯度统计值，并按照指定格式保存。

聚合前梯度统计值如下所示，同时记录 rank 值。

```
max: 1.6308345038851257e-06
norm: 1.0331401426810771e-05
param_name: base_model.model.module.model.layers.26.self_
rank: 1
---
max: 0.0181514210999012
norm: 0.31125572323799133
param_name: base_model.model.module.model.layers.25.self_
rank: 1
```

聚合后梯度统计值如下所示。

```
max: 3.5047924029640853e-05
norm: 0.00010367669892730191
param_name: module.module.base_model.model.module.mod
---
max: 0.03474785014986992
norm: 0.31352993845939636
param_name: module.module.base_model.model.module.mod
---
max: 0.0031788889318704605
```