PLM4All: scripts for LLMs on Jean Zay

H. Bourfoune, N. Cassereau, C. Cerisara, P. Cornette, C. Gardent, G. Guibon CNRS, LORIA

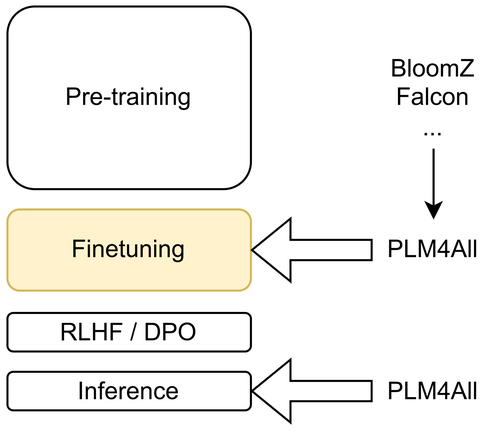
Objectives

1. Faciliter l'utilisation et le finetuning des LLMs sur Jean Zay
2. Proposer des scripts de finetuning et inférence
3. Proposer un unique point d'accès aux scripts: **module load llm**

Projet PLM4All

- Financement: CNRS
- Durée: 6 mois (Mars - Août 2023)
- Participants: Synalp (LORIA, Nancy), IDRIS (CNRS, Paris)
- Leader: Synalp (LORIA, Nancy)

Applications



Focus:

Mémoire Faire tenir 176b-parms dans un min. de GPUs

Vitesse Optimiser l'utilisation des GPUs

QLoRA Quantiser pour finetuner

Contact Information

Email cerisara@loria.fr

Mastodon @cerisara@mastodon.online

Barrières à l'accès à Jean Zay

- Création de comptes: signatures, FSD
- Accès à Jean Zay: machines sécurisées, IP fixe
- Spécificités techniques: disque $HOME très limité; pas d'accès internet depuis les nœuds de calculs...
- Apprendre SLURM; configurer RAM, nb de process...
- Multi-GPU ? Multi-Noeuds ? Gros modèles: VRAM ? vitesse ?

Résultats

Scripts disponibles actuellement:

mode	reqs	nparms	méthode
inférence	8xA100	176b	deepspeed
finetune	8xA100	40b	deepspeed
finetune	8xA100	30b	deepspeed + pipeline parallelism
finetune	2xA100	176b	qLoRA
finetune	cpu 24G ram	176b	très lent !

Notre vision est de diffuser des scripts "exemples" aussi simples et légers que possible pour en faciliter l'adaptation.

Suite du projet

LLM4All: Projet ANR TSIA (Thématiques Spécifiques en IA)

Durée 2023-2027

Consortium LORIA, LIX, APHP, Linagora, Huggingface

Objectifs mise à jour en continu des LLM

Futurs projets suivant PLM4All

OpenLLM-FR	Initiative communautaire open-source animée par Linagora
DINUM	incubateur IA
ENACT	AI-Cluster

Exemple: Finetune Falcon

```
deepspeed.init_distributed(
    dist_backend='nccl',
    init_method='env://',
)
torch.cuda.set_device(int(
    os.environ['SLURM_LOCALID']))
device = torch.device("cuda")

args = parse_args()

train_dataset, test_dataset,
    label_ids = get_hf_dataset(args)
model_engine = get_ds_model(args)

model = train(args, train_dataset,
        model_engine, label_ids, device)
doeval(args, model_engine,
        test_dataset, label_ids, device)
```

Exemple: Slurm job options

```
#!/bin/bash
#SBATCH --job-name=finetune_deepspeed_falcon
#SBATCH --output=finetune_deepspeed_falcon.out
#SBATCH --error=finetune_deepspeed_falcon.out
#SBATCH --gres=gpu:8
#SBATCH --ntasks-per-node=8
#SBATCH --nodes=1
#SBATCH --hint=nomultithread
#SBATCH --time=00:50:00
#SBATCH --qos=qos_gpu-dev
#SBATCH --cpus-per-task=8
#SBATCH --account=sos@a100
#SBATCH -C a100

## load Pytorch module
module purge
module load cpuarch/amd
module load pytorch-gpu/py3/2.0.0

## launch script on every node
set -x

# code execution
srun python finetune_deepspeed_falcon.py --debug --stage 3 \
   --model_name "tiiuae/falcon-40b" --nb_layer_freezed 40
```

Communauté

Site Web
https://gitlab.inria.fr/synalp/plm4all
Gitlab
https://gitlab.inria.fr/synalp/plm4all
Instant Messaging via **Matrix**
https://app.gitter.im/#/room/#plm4all