LLaMA-Factory

Easy and Efficient LLM Fine-Tuning

LLaMA-Factory/examples/train_lora/llama3_lora_sft_ds3.yaml at...

Unified Efficient Fine-Tuning of 100+ LLMs & VLMs (ACL 2024) - hiyouga/LLaMA-Factory

GitHub

Features

- **Various models**: LLaMA, LLaVA, Mistral, Mixtral-MoE, Qwen, Qwen2-VL, DeepSeek, Yi, Gemma, ChatGLM, Phi, etc.
- **Integrated methods**: (Continuous) pre-training, (multimodal) supervised fine-tuning, reward modeling, PPO, DPO, KTO, ORPO, etc.
- **Scalable resources**: 16-bit full-tuning, freeze-tuning, LoRA and 2/3/4/5/6/8-bit QLoRA via AQLM/AWQ/GPTQ/LLM.int8/HQQ/EETQ.
- **Advanced algorithms**: GaLore, BAdam, APOLLO, Adam-mini, Muon, DoRA, LongLoRA, LLaMA Pro, Mixture-of-Depths, LoRA+, LoftQ and PiSSA.
- **Practical tricks**: FlashAttention-2, Unsloth, Liger Kernel, RoPE scaling, NEFTune and rsLoRA.
- **Wide tasks**: Multi-turn dialogue, tool using, image understanding, visual grounding, video recognition, audio understanding, etc.
- **Experiment monitors**: LlamaBoard, TensorBoard, Wandb, MLflow, SwanLab, etc.
- **Faster inference**: OpenAI-style API, Gradio UI and CLI with vLLM worker or SGLang worker.

Supported Models

Model	Model size	Template
Baichuan 2	7B/13B	baichuan2
BLOOM/BLOOMZ	560M/1.1B/1.7B/3B/7.1B/176B	-
ChatGLM3	6B	chatglm3
Command R	35B/104B	cohere
DeepSeek (Code/MoE)	7B/16B/67B/236B	deepseek
DeepSeek 2.5/3	236B/671B	deepseek3
DeepSeek R1 (Distill)	1.5B/7B/8B/14B/32B/70B/671B	deepseekr1
Falcon	7B/11B/40B/180B	falcon
Gemma/Gemma 2/CodeGemma	2B/7B/9B/27B	gemma
Gemma 3	1B/4B/12B/27B	gemma3/gemma (1B)
GLM-4/GLM-4-0414/GLM-Z1	9B/32B	glm4/glmz1
GPT-2	0.1B/0.4B/0.8B/1.5B	-
Granite 3.0-3.3	1B/2B/3B/8B	granite3
Hunyuan	7B	hunyuan
Index	1.9B	index

Supported Training Approaches

Approach	Full-tuning	Freeze-tuning	LoRA	QLoRA
Pre-Training	✅	✅	✅	✅
Supervised Fine-Tuning	✅	✅	✅	✅
Reward Modeling	✅	✅	✅	✅
PPO Training	✅	✅	✅	✅
DPO Training	✅	✅	✅	✅
KTO Training	✅	✅	✅	✅
ORPO Training	✅	✅	✅	✅
SimPO Training	✅	✅	✅	✅

> **Tip**
>
> The implementation details of PPO can be found in this blog.

Installation

> **Important**
>
> Installation is mandatory.

```
git clone --depth 1 https://github.com/hiyouga/LLaMA-Factory.git
cd LLaMA-Factory
pip install -e ".[torch,metrics]"
```

Extra dependencies available: torch, torch-npu, metrics, deepspeed, liger-kernel, bitsandbytes, hqq, eetq, gptq, aqlm, vllm, sglang, galore, apollo, badam, adam-mini, qwen, minicpm_v, modelscope, openmind, swanlab, quality

> **Tip**
>
> Use `pip install --no-deps -e .` to resolve package conflicts.

Example Dataset

Files

main

Go to file

- .github
- assets
- data
  - belle_multiturn
  - hh_rlhf_en
  - mllm_demo_data
  - ultra_chat
  - README.md
  - README_zh.md
  - alpaca_en_demo.json
  - alpaca_zh_demo.json
  - c4_demo.jsonl
  - dataset_info.json
  - dpo_en_demo.json
  - dpo_zh_demo.json
  - glaive_toolcall_en_demo.json
  - glaive_toolcall_zh_demo.json
  - identity.json

LLaMA-Factory / data / README.md

erictang000 [data] support for specifying a dataset in cloud storage (#7567) 6c53471 · last month History

Preview Code Blame 466 lines (387 lo… Raw

The dataset_info.json contains all available datasets. If you are using a custom dataset, please **make sure** to add a *dataset description* in `dataset_info.json` and specify `dataset: dataset_name` before training to use it.

Currently we support datasets in **alpaca** and **sharegpt** format.

```
"dataset_name": {
  "hf_hub_url": "the name of the dataset repository on the Hugging Face hub. (if specified, ignore script_url, file_name and cl
  "ms_hub_url": "the name of the dataset repository on the Model Scope hub. (if specified, ignore script_url, file_name and clo
  "script_url": "the name of the directory containing a dataset loading script. (if specified, ignore file_name and cloud_file_
  "cloud_file_name": "the name of the dataset file in s3/gcs cloud storage. (if specified, ignore file_name)",
  "file_name": "the name of the dataset folder or dataset file in this directory. (required if above are not specified)",
  "formatting": "the format of the dataset. (optional, default: alpaca, can be chosen from {alpaca, sharegpt})",
  "ranking": "whether the dataset is a preference dataset or not. (default: False)",
  "subset": "the name of the subset. (optional, default: None)",
  "split": "the name of dataset split to be used. (optional, default: train)",
  "folder": "the name of the folder of the dataset repository on the Hugging Face hub. (optional, default: None)",
  "num_samples": "the number of samples in the dataset to be used. (optional, default: None)",
  "columns (optional)": {
    "prompt": "the column name in the dataset containing the prompts. (default: instruction)",
    "query": "the column name in the dataset containing the queries. (default: input)",
    "response": "the column name in the dataset containing the responses. (default: output)",
    "history": "the column name in the dataset containing the histories. (default: None)",
    "messages": "the column name in the dataset containing the messages. (default: conversations)",
    "system": "the column name in the dataset containing the system prompts. (default: None)",
    "tools": "the column name in the dataset containing the tool description. (default: None)",
```

Files

main

Go to file

- .github
- assets
- data
  - belle_multiturn
  - hh_rlhf_en
  - mllm_demo_data
  - ultra_chat
  - README.md
  - README_zh.md
  - alpaca_en_demo.json
  - alpaca_zh_demo.json
  - c4_demo.jsonl
  - dataset_info.json
  - dpo_en_demo.json
  - dpo_zh_demo.json
  - glaive_toolcall_en_demo.json
  - glaive_toolcall_zh_demo.json
  - identity.json

LLaMA-Factory / data / dataset_info.json

hiyouga [data] add coig-p dataset (#7657)

Code Blame 708 lines (708 loc) · 16.5 KB

```
{
  "identity": {
    "file_name": "identity.json"
  },
  "alpaca_en_demo": {
    "file_name": "alpaca_en_demo.json"
  },
  "alpaca_zh_demo": {
    "file_name": "alpaca_zh_demo.json"
  },
  "glaive_toolcall_en_demo": {
    "file_name": "glaive_toolcall_en_demo.json",
    "formatting": "sharegpt",
    "columns": {
      "messages": "conversations",
      "tools": "tools"
    }
  },
  "glaive_toolcall_zh_demo": {
    "file_name": "glaive_toolcall_zh_demo.json",
    "formatting": "sharegpt",
    "columns": {
      "messages": "conversations",
      "tools": "tools"
    }
  },
  "mllm_demo": {
```

Setting path of dataset

Training

Files

main

Go to file

- .github
- assets
- data
- docker
- evaluation
- examples
  - accelerate
  - deepspeed
  - extras
  - inference
  - merge_lora
  - train_full
  - train_lora
  - train_qlora
  - README.md
  - README_zh.md
- scripts

LLaMA-Factory / examples /

hiyouga [example] update examples (#7964) c6bcca4 · 5 days ago History

Name	Last commit message	Last commit date
..		
accelerate	[model] fix kv cache (#7564)	last month
deepspeed	[misc] fix ds config (#7205)	2 months ago
extras	[trainer] Add Muon Optimizer (#7749)	3 weeks ago
inference	[example] update examples (#7964)	5 days ago
merge_lora	[example] update examples (#7964)	5 days ago
train_full	[example] update examples (#7964)	5 days ago
train_lora	[example] update examples (#7964)	5 days ago
train_qlora	[misc] fix packing and eval plot (#7623)	last month
README.md	[example] update examples (#7964)	5 days ago
README_zh.md	[example] update examples (#7964)	5 days ago

README.md

Files

main

Go to file

- merge_lora
- train_full
- train_lora
  - llama3_lora_dpo.yaml
  - llama3_lora_eval.yaml
  - llama3_lora_kto.yaml
  - llama3_lora_ppo.yaml
  - llama3_lora_pretrain.yaml
  - llama3_lora_reward.yaml
  - llama3_lora_sft.sh
  - llama3_lora_sft.yaml
  - llama3_lora_sft_ds3.yaml
  - llama3_lora_sft_ray.yaml
  - llama3_preprocess.yaml
  - llama4_lora_sft_ds3.yaml
  - qwen2_5vl_lora_dpo.yaml
  - qwen2_5vl_lora_sft.yaml

LLaMA-Factory / examples / train_lora / llama3_lora_sft_ds3.yaml

hiyouga [misc] fix packing and eval plot (#7623) ✓ 5817cda · last month History

Code Blame 47 lines (42 loc) · 1.02 KB Raw

```yaml
### model
model_name_or_path: meta-llama/Meta-Llama-3-8B-Instruct
trust_remote_code: true

### method
stage: sft
do_train: true
finetuning_type: lora
lora_rank: 8
lora_target: all
deepspeed: examples/deepspeed/ds_z3_config.json  # choices: [ds_z0_config.json, ds_z2_config.json, ds_z3_config.json]

### dataset
dataset: identity,alpaca_en_demo
template: llama3
cutoff_len: 2048
max_samples: 1000
overwrite_cache: true
preprocessing_num_workers: 16
dataloader_num_workers: 4

### output
output_dir: saves/llama3-8b/lora/sft
logging_steps: 10
```

Hugginface-cli

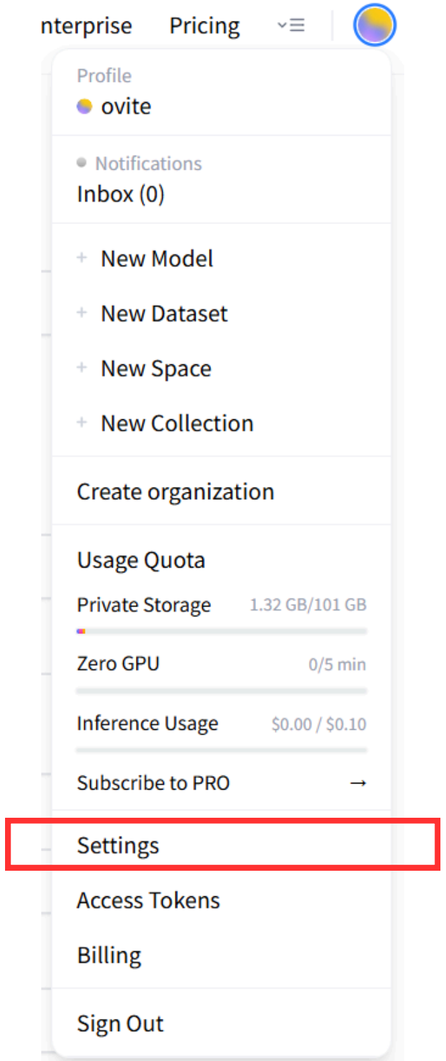
```
>>> pip install -U "huggingface_hub[cli]"
```

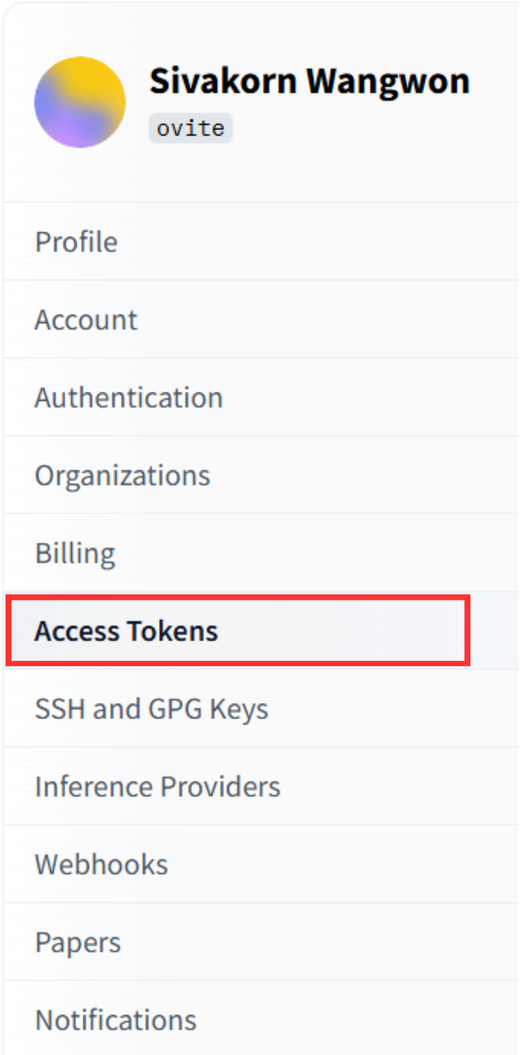
```
huggingface-cli login
```

```
_|    _|  _|    _|    _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|      _|_|_|_|    _|_|
_|    _|  _|    _|  _|        _|          _|    _|_|    _|  _|            _|        _|    _|  _|
_|_|_|_|  _|    _|  _|  _|_|  _|  _|_|    _|    _|  _|  _|  _|  _|_|      _|_|_|    _|_|_|_|  _|
_|    _|  _|    _|  _|    _|  _|    _|    _|    _|    _|_|  _|    _|      _|        _|    _|  _|
_|    _|    _|_|      _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|      _|        _|    _|

To log in, `huggingface_hub` requires a token generated from https://huggingface.co/settings/tok
Enter your token (input will not be visible):
Add token as git credential? (Y/n)
Token is valid (permission: write).
Your token has been saved in your configured git credential helpers (store).
Your token has been saved to /home/wauplin/.cache/huggingface/token
Login successful
```

How to get hugginface's token?





Access Tokens

User Access Tokens

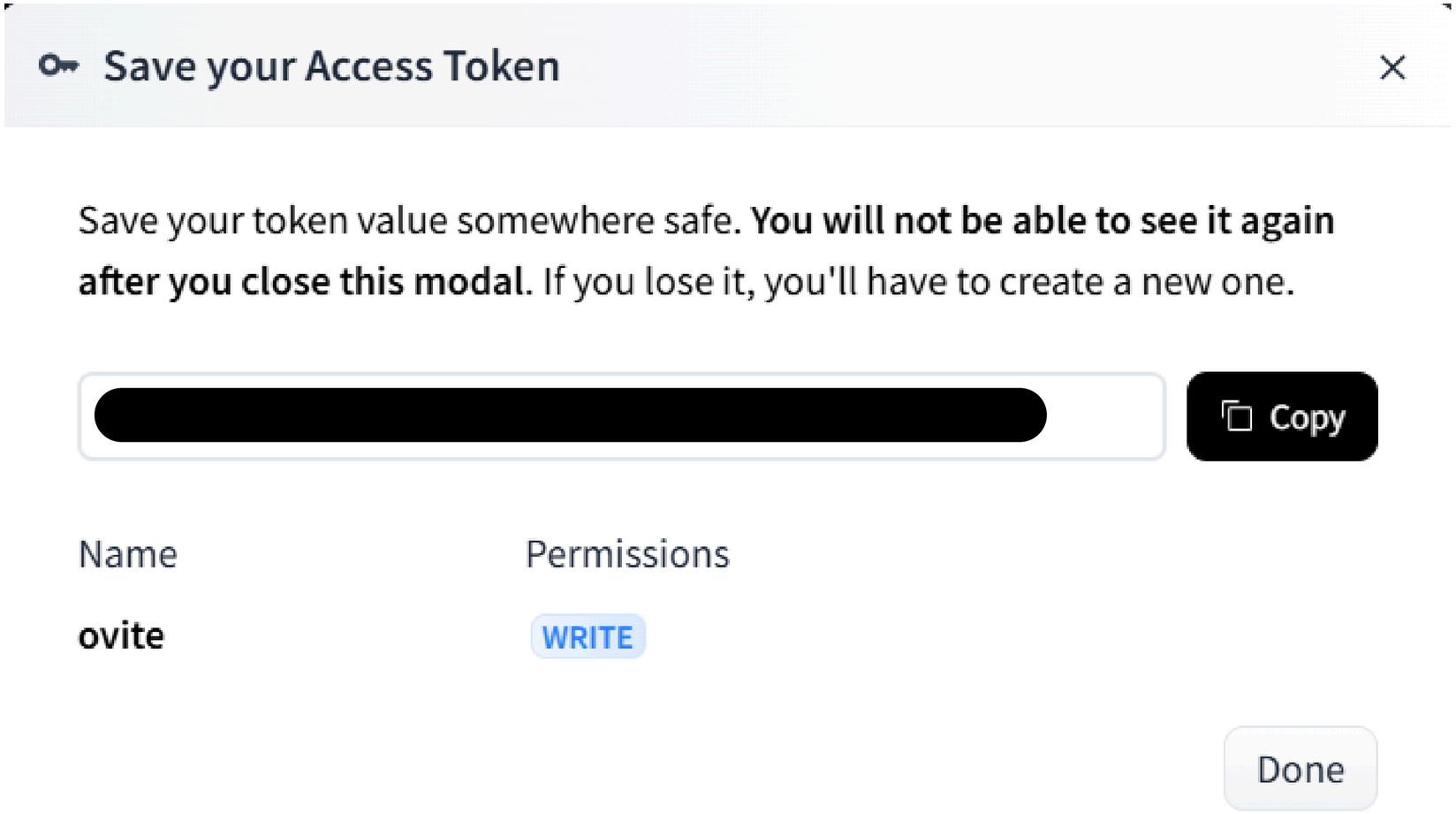
+ Create new token

Access tokens authenticate your identity to the Hugging Face Hub and allow applications to perform actions based on token permissions. **Do not share your Access Tokens with anyone**; we regularly check for leaked Access Tokens and remove them immediately.

Name	Value	Last Refreshed Date	Last Used Date	Permissions
ovite	hf_...JYPQ	23 days ago	5 days ago	WRITE
fine	hf_...	May 2, 2024	-	
ovite	hf_...	Dec 24, 2023	-	READ

Invalidate and refresh

Delete


Save your Access Token
Save your token value somewhere safe. You will not be able to see it again after you close this modal. If you lose it, you'll have to create a new one.
[redacted]
Copy
Name
Permissions
ovite
WRITE
Done

Access Tokens

User Access Tokens

+ Create new token

Access tokens authenticate your identity to the Hugging Face Hub and allow applications to perform actions based on token permissions. **Do not share your Access Tokens with anyone**; we regularly check for leaked Access Tokens and remove them immediately.

Name	Value	Last Refreshed Date	Last Used Date	Permissions
ovite	hf_...JYPQ	less than a minute ago	5 days ago	WRITE
fine	hf_...	May 2, 2024	-	FINEGRAINED
ovite	hf_...	Dec 24, 2023	-	READ

Create new Access Token

Token type

Fine-grained Read **Write**

This cannot be changed after token creation.

Token name

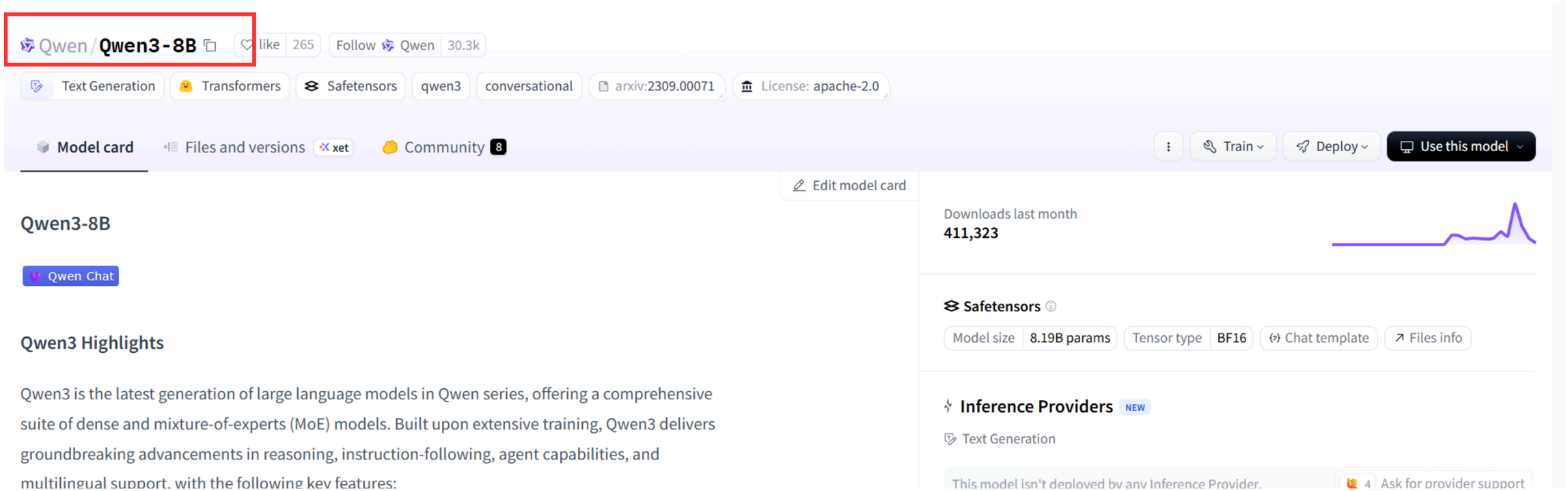
Token name

This token has read and write access to all your and your orgs resources and can make calls to Inference Providers on your behalf.

Create token

ใส่ชื่ออะไรก็ได้แล้วกด create token

How to load model in huggingface to Lanta?



huggingface-cli download Qwen/Qwen3-8B --local-dir ./Qwen3-8B

ตามหลัง --local-dir คือตำแหน่งที่เอา model ไปวางแล้วก็ตั้งชื่อ folder ของ model

```yaml
### model
model_name_or_path: meta-llama/Meta-Llama-3-8B-Instruct
trust_remote_code: true
```

Model's Path

```yaml
### method
stage: sft
do_train: true
finetuning_type: lora
lora_rank: 8
lora_target: all
deepspeed: examples/deepspeed/ds_z3_config.json
```

```yaml
### dataset
dataset: identity,alpaca_en_demo
template: llama3
cutoff_len: 2048
max_samples: 1000
overwrite_cache: true
preprocessing_num_workers: 16
dataloader_num_workers: 4
```

dataset คือชื่อที่เราต้องการใช้ใน data_info
template ตั้งตามโมเดลที่เราต้องการ finetuned
max_sample จำนวน sample มากสุดที่ใช้ในการ train

Deepspeed Setting

```
{
  "train_batch_size": "auto",
  "train_micro_batch_size_per_gpu": "auto",
  "gradient_accumulation_steps": "auto",
  "gradient_clipping": "auto",
  "zero_allow_untested_optimizer": true,
  "fp16": {
    "enabled": "auto",
    "loss_scale": 0,
    "loss_scale_window": 1000,
    "initial_scale_power": 16,
    "hysteresis": 2,
    "min_loss_scale": 1
  },
  "bf16": {
    "enabled": "auto"
  },
  "zero_optimization": {
    "stage": 3,
    "overlap_comm": false,
    "contiguous_gradients": true,
    "sub_group_size": 1e9,
    "reduce_bucket_size": "auto",
    "stage3_prefetch_bucket_size": "auto",
    "stage3_param_persistence_threshold": "auto",
    "stage3_max_live_parameters": 1e9,
    "stage3_max_reuse_distance": 1e9,
    "stage3_gather_16bit_weights_on_model_save": true
  }
}
```

Use the following 3 commands to run LoRA **fine-tuning**, **inference** and **merging** of the Llama3-8B-Instruct model, respectively.

```
llamafactory-cli train examples/train_lora/llama3_lora_sft.yaml
llamafactory-cli chat examples/inference/llama3_lora_sft.yaml
llamafactory-cli export examples/merge_lora/llama3_lora_sft.yaml
```

Training with Multi-node

```
#!/bin/bash

module purge
module load Mamba/23.11.0-0
module load cudatoolkit/23.3_12.0
module load gcc/12.2.0
# module load cuda/12.0

conda deactivate
conda activate llamaenv # Activate your conda environment

echo "User: $(whoami)"
echo "Hostname: $(hostname)"
echo "SLURM_PROCID: $SLURM_PROCID"

# Chang this path to your own path
export LD_LIBRARY_PATH=/project/lt200258-aithai/lib:$LD_LIBRARY_PATH
export TRANSFORMERS_CACHE=/cache
export HF_DATASETS_CACHE=/cache


# LLaMA Factory specific environment variables
export FORCE_TORCHRUN=1
export RANK=$SLURM_PROCID

# Run LLaMA Factory CLI
llamafactory-cli train ./examples/train_lora/llama3_lora_sft_ds3.yaml
```

เปลี่ยน conda env เป็นของตัวเอง
และเปลี่ยน path ของ LD_LIBRALY_PATH

```bash
#!/bin/bash
#SBATCH -p gpu                          # Specify partition [Compute/Memory/GPU]
#SBATCH -N 4                           # Specify number of nodes
#SBATCH -c 64                           # Specify processors per task
#SBATCH --ntasks-per-node=1             # Specify number of tasks per node
#SBATCH --gpus-per-node=4               # Specify total number of GPUs per node
#SBATCH -t 72:00:00                     # Specify maximum time limit (72 hours)
#SBATCH -A lt200258                     # Specify project name
#SBATCH -J llamafac                    # Specify job name
#SBATCH -o out-llamafac_IR-%j.txt

# Environment setup
export NCCL_DEBUG=INFO
export NCCL_SOCKET_IFNAME=hsn
export NCCL_TIMEOUT=3600000
export NCCL_BLOCKING_WAIT=0
export WANDB_MODE="offline"

# Distributed training setup
export NNODES=$SLURM_NNODES
export MASTER_ADDR=$(scontrol show hostnames "$SLURM_JOB_NODELIST" | head -n 1)
export MASTER_PORT=29500

echo "Nodes: $NNODES"
echo "Master: $MASTER_ADDR:$MASTER_PORT"

# Run the training script on all nodes
srun bash multi_node.sh
```

เปลี่ยนจำนวน Node ที่ใช้