

# LLama2 - 大語言模型訓練

介紹 LLama2 的訓練方法 - 監督式微調 (SFT)



# 大語言模型的訓練階段

## (1) 預訓練

使用大量無標籤的資料訓練模型。

## (2) 增量預訓練

在預訓練模型基礎上，進一步訓練。

## (3) 微調

使用有標籤的資料微調模型，以提高其在特定任務上的準確度。

## (4) 強化學習

使用基於人類反饋的強化學習 (Reinforcement Learning from Human Feedback) 進行訓練。

# 微調 - 監督式微調

1

## 訓練集及驗證集準備

準備含大量QA對的訓練集，分出一部分作為驗證集

2

## 設定訓練參數

設定模型的訓練參數

3

## 訓練模型

使用標籤資料微調模型，重複多次以提高模型準確度。

# 微調大型語言模型LLM的技術 - LoRA

近年隨著模型越來越大，直接使用 Fine-Tuning 對所有參數進行訓練的成本是相當高昂的。因此近年來大家開始研究有效率的 Fine-Tuning，稱作 Parameter-Efficient Fine-Tuning (PEFT)。

Microsoft 團隊提出的 Low-Rank Adaptation (LoRA)：透過凍結原本預訓練模型的權重，在預訓練模型的最後一層或幾層添加了低秩矩陣，並且只訓練這些低秩矩陣的參數，而不是整個模型的所有參數。

和全參數訓練相比，該方法所需的訓練參數量節省 10,000 倍，並且只需要 1/3 的 GPU 使用量。

資料來源：<https://xiaosean5408.medium.com/>

# LLama-Efficient-Tuning

GitHub: <https://github.com/hiyouga/LLaMA-Efficient-Tuning>

hiyouga / LLaMA-Efficient-Tuning

Type  to search

>\_

+ ▾

🕒

🔗

📧

🏠

<> Code

🕒 Issues 79

🔗 Pull requests 2


🕒 Actions

📁 Projects

📖 Wiki

🛡 Security

📈 Insights

 LLaMA-Efficient-Tuning

Public

👁 Watch 34 ▾

🍴 Fork 809 ▾

★ Star 3.7k ▾

🔗 main ▾


🔗 3 branches

🏷 9 tags

Go to file

Add file ▾

<> Code ▾

 hiyouga update baichuan2 template

💬 1

0531886

13 hours ago

🕒 339 commits

📁 assets	Update wechat.jpg	last week
📁 data	refactor dataset_attr, add eos in pt, fix #757	last week
📁 src	update baichuan2 template	13 hours ago
📁 tests	add template encode test	2 weeks ago
📄 .gitattributes	Update .gitattributes	2 months ago
📄 .gitignore	Update .gitignore	3 weeks ago
📄 LICENSE	Initial commit	4 months ago

About

Easy-to-use LLM fine-tuning framework (LLaMA-2, BLOOM, Falcon, Baichuan, Qwen, ChatGLM2)

bloom

reinforcement-learning

transformers

falcon

llama

quantization

language-model

fine-tuning

peft

dpo

pre-training

baichuan

llm

rlhf

chatglm

qlora

llama2

qwen

📖 Readme

📄 Apache-2.0 license

# LLama-Efficient-Tuning 安裝

```
#####  
# My Cuda Version: 11.8 #  
# My Torch Version: 2.0.1 #  
#####
```

安裝：(Administrator Anaconda Powershell Prompt)

```
git clone https://github.com/hiyouga/LLaMA-Efficient-Tuning.git  
conda create python=3.10 -p D:\ProgramData\anaconda3\envs\llama_08220817  
conda activate llama_08220817  
cd LLaMA-Efficient-Tuning  
pip install -r requirements.txt  
pip uninstall torch  
pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
```

run.sh:

```
CUDA_VISIBLE_DEVICES=1 python src/train_bash.py \  
  --stage sft \  
  --model_name_or_path  
  "E:\model\models--FlagAlpha--Llama2-Chinese-7b-Chat\snapshots\4c3bc725f71898c6a1acd  
  4ea98a2f8d74d1b1b6b" \  
  --do_train \  
  --dataset self_cognition \  
  --finetuning_type lora \  
  --output_dir e50_08220658 \  
  --overwrite_cache \  
  --per_device_train_batch_size 4 \  
  --per_device_eval_batch_size 4 \  
  --lr_scheduler_type cosine \  
  --logging_strategy epoch \  
  --evaluation_strategy epoch \  
  --save_strategy epoch \  
  --val_size 0.1 \  
  --learning_rate 5e-5 \  
  --num_train_epochs 50.0 \  
  --plot_loss True \  
  --fp16
```

# 過擬合

## 1 監控 Training Loss 和 Eval Loss

當 Training Loss 持續下降，  
但 Eval Loss 卻開始上升，  
就可能出現過擬合。

## 2 使用 Early Stopping

當 Eval Loss 超過一定次數  
沒有下降，就提前結束訓練，  
以避免出現過擬合。





# 訓練結果

