Quantização de LLMs

Hugging Face 🤗 PEFT / QLoRA

Introdução

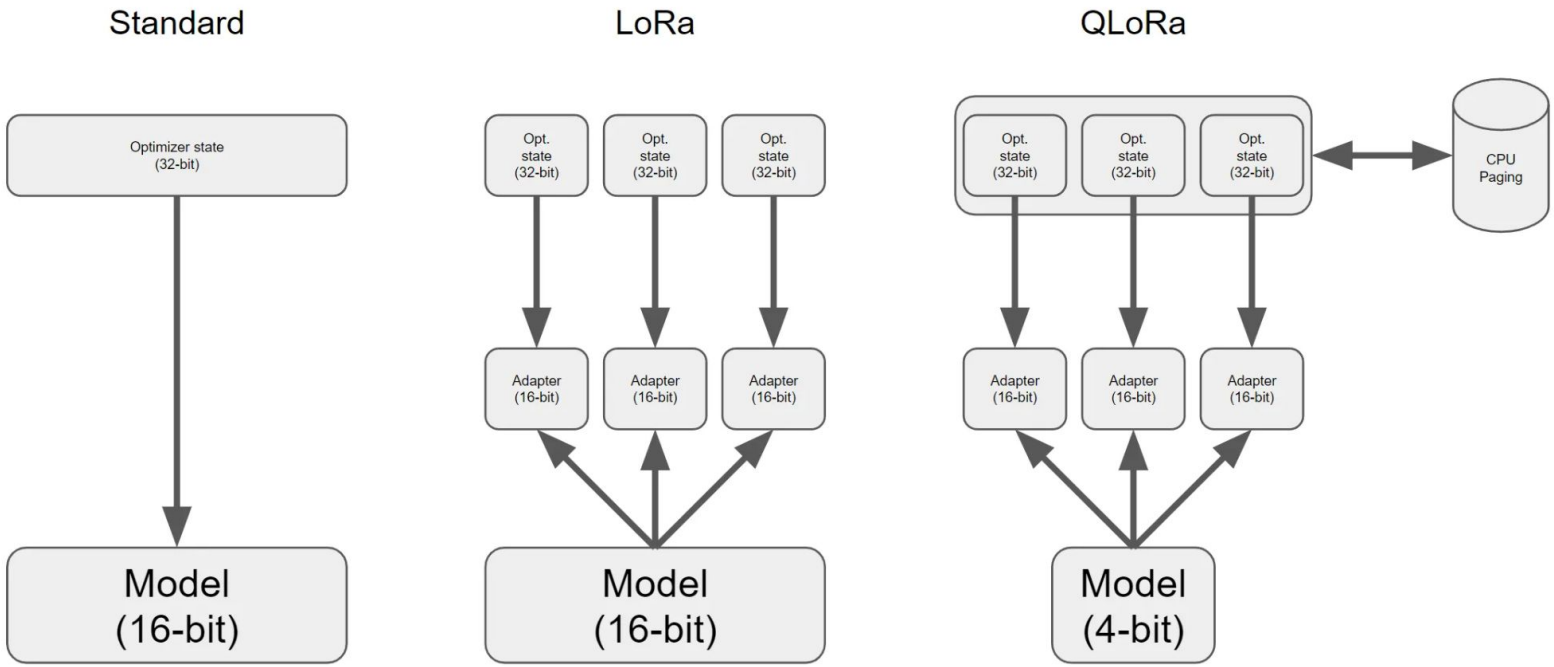
O que é PEFT?
Parameter-Efficient Fine-Tuning (PEFT) é uma biblioteca do 🤗 Hugging Face, criada para suportar a criação e ajuste fino de camadas de adaptador em LLMs. É perfeitamente integrada com 🤗 “Accelerate”, também do Hugging Face, para modelos de grande escala (LLMs) aproveitando “DeepSpeed” e “Big Model Inference”.

O que é Quantização?
A quantização é uma técnica para reduzir os custos computacionais e de memória na execução da inferência, representando o pesos e ativações com tipos de dados de baixa precisão, como inteiro de 8 bits, em vez do habitual ponto-flutuante de 32 bits.

O que é LoRA?
Low-Rank Adaptation of Large Language Models (LoRA) é um método de treinamento que acelera o treinamento de modelos grandes enquanto consome menos memória. São adicionados pares de matrizes de peso de “rank-decomposition” (decomposição de classificação), chamadas matrizes de atualização, aos pesos existentes e são treinados apenas os pesos recém-adicionados.

O que é QLoRA? (QLoRA = Quantização + LoRA)
O LoRA adiciona uma pequena quantidade de parâmetros treináveis, ou seja, adaptadores, para cada camada do LLM e congela todos os parâmetros originais. Logo, para ajuste fino, só precisamos atualizar os pesos do adaptador, o que reduz significativamente o espaço ocupado pela memória, e o QLoRA adiciona a isso quantização de 4 bits, quantização dupla e a exploração da memória unificada da NVidia para paginação (comunicação performática e confiável CPU ⇔ GPU).

Ajuste fino de LLMs com QLoRA



Modelos LLM suportados pelo QLoRA

O QLoRA pode ser usado com qualquer modelo que suporte carregamento acelerado usando a biblioteca **“accelerate”**, ou seja, qualquer modelo que aceite o argumento **“device_map”** ao ser chamado com **“from_pretrained”** deve ser quantizável em 4 bits.

Alguns exemplos de modelos:
'bigbird_pegasus', 'blip_2', **'bloom'**, 'bridgetower', 'codegen', 'deit', 'esm', **'gpt2'**, 'gpt_bigcode', **'gpt_neo'**, **'gpt_neox'**, 'gpt_neox_japanese', **'gptj'**, 'gptsan_japanese', 'lilt', **'llama'**, 'longformer', **'longt5'**, 'luke', 'm2m_100', **'mbart'**, 'mega', **'mt5'**, 'nllb_moe', **'open_llama'**, **'opt'**, 'owlvit', 'plbart', **'roberta'**, 'roberta_prelayernorm', 'rwkv', 'switch_transformers', **'t5'**, 'vilt', 'vit', 'vit_hybrid', **'whisper'**, 'xglm', **'xlm_roberta', …**

Ajuste fino de um modelo GPTJ-6B usando QLoRA

```
48  # quantization config
49  quant_config = BitsAndBytesConfig(
50      load_in_4bit=True,
51      bnb_4bit_use_double_quant=True,
52      bnb_4bit_quant_type="nf4",
53      bnb_4bit_compute_dtype=torch.bfloat16
54  )
55
56  # model
57  model = AutoModelForCausalLM.from_pretrained(model_name, quantization_config=quant_config, device_map={"":0})
58  model.gradient_checkpointing_enable()
59  model = prepare_model_for_kbit_training(model)
60  lora_args = LoraConfig(
61      r=8,
62      lora_alpha=32,
63      # target_modules=["query_key_value"], # gpt-neox-20b
64      target_modules=["q_proj", "v_proj"], # gpt-j-6b
65      lora_dropout=0.05,
66      bias="none",
67      task_type="CAUSAL_LM"
68  )
69  model = get_peft_model(model, lora_args)
70  model.print_trainable_parameters()
71  # trainable params: 3,670,016 || all params: 3,235,980,512 || trainable%: 0.11341279672082277
```

Consumo de GPU durante o ajuste fino do modelo

*"To load GPT-J in **float32** one would need at least 2x model size RAM: 1x for initial weights and another 1x to load the checkpoint. So for GPT-J it would take at least **48GB RAM** to just load the model."*

- Fonte: https://huggingface.co/docs/transformers/main/model_doc/gptj

```
$ watch -n 3 'nvidia-smi && free -h'
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 515.105.01   Driver Version: 515.105.01   CUDA Version: 11.7     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   1  NVIDIA GeForce ...  Off  | 00000000:04:00.0 Off |                  N/A |
|100%   89C    P2   166W / 170W |   7439MiB / 12288MiB |     93%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
               total        used        free      shared  buff/cache   available
Mem:            77Gi        14Gi        23Gi        79Mi        39Gi        62Gi
Swap:           37Gi          0B        37Gi
```

Resultado do ajuste fino do modelo GPTJ-6B

Adaptador LoRA do modelo:

```
total 15M
-rw-rw-r-- 1 dockeradmin dockeradmin 417 Jul 10 11:44 adapter_config.json
-rw-rw-r-- 1 dockeradmin dockeradmin 15M Jul 10 11:44 adapter_model.bin
-rw-rw-r-- 1 dockeradmin dockeradmin 805 Jul 10 11:44 README.md
```

Resultado do treinamento:

```
# Frase original:
"I'm selfish, impatient and a little insecure. I make mistakes, I am out of control and at times hard to handle. But if you can't handle me at my
worst, then you sure as hell don't deserve me at my best."
Marilyn Monroe

# Saida (todo os 939 steps e 3 epocas):
>>  I'm selfish, impatient and  =>  I'm selfish, impatient and a little insecure. I make mistakes, I am out of control and at times hard to handle.

# Saida - modelo original:
>>  I'm selfish, impatient and  =>  I'm selfish, impatient and a little bit of a control freak. I'm also a mom, a wife, a daughter,
```

Inferência em um modelo GPTJ-6B usando QLoRA

```python
# quantization config
quant_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.bfloat16
)

# model
model = AutoModelForCausalLM.from_pretrained(model_path, quantization_config=quant_config, device_map={"":0})
```

Inferência utilizando carregamento de modelo com adaptador LoRA:

```python
# quantization config
quant_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.bfloat16
)

# model com PEFT LoRA
config = PeftConfig.from_pretrained(model_path)
model = AutoModelForCausalLM.from_pretrained(config.base_model_name_or_path, quantization_config=quant_config, device_map={"":0})
model = PeftModel.from_pretrained(model, model_path)
```

Consumo de GPU durante a inferência com QLoRA

*"To load GPT-J in **float32** one would need at least 2x model size RAM: 1x for initial weights and another 1x to load the checkpoint. So for GPT-J it would take at least **48GB RAM** to just load the model."*

- Fonte: https://huggingface.co/docs/transformers/main/model_doc/gptj

```
$ nvidia-smi
Wed Jul 12 01:57:56 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 515.105.01   Driver Version: 515.105.01   CUDA Version: 11.7     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   1  NVIDIA GeForce ...  Off  | 00000000:04:00.0 Off |                  N/A |
| 52%   70C    P2   169W / 170W |   5777MiB / 12288MiB |     93%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
```

Referências

https://huggingface.co/docs/optimum/concept_guides/quantization

https://github.com/huggingface/peft

https://github.com/huggingface/accelerate

https://huggingface.co/blog/4bit-transformers-bitsandbytes

https://huggingface.co/blog/trl-peft

https://huggingface.co/docs/diffusers/training/lora

https://towardsdatascience.com/qlora-fine-tune-a-large-language-model-on-your-gpu-27bed5a03e2b

https://huggingface.co/docs/transformers/main/model_doc/gptj

Código utilizado e o modelo GPTJ-6B com o ajuste fino

https://github.com/nlpulse-io/sample_codes/tree/main/fine-tuning/peft_quantization_4bits/gptj-6b

https://huggingface.co/nlpulse/gpt-j-6b-english_quotes