大模型微调方法之QLoRA

原创 喜欢瓦力的卷卷 瓦力算法学研所 2024年10月06日 20:43 上海

技术总结专栏

本文介绍大模型微调方法中的QLoRA。

QLoRA由华盛顿大学UW NLP小组的成员于2023年提出发，旨在进一步降低微调大模型的微调成本，因为对于上百亿参数量的模型，LoRA微调的成本还是很高。

感兴趣的小伙伴可以去阅读一下原文：https://arxiv.org/pdf/2305.14314

模型介绍

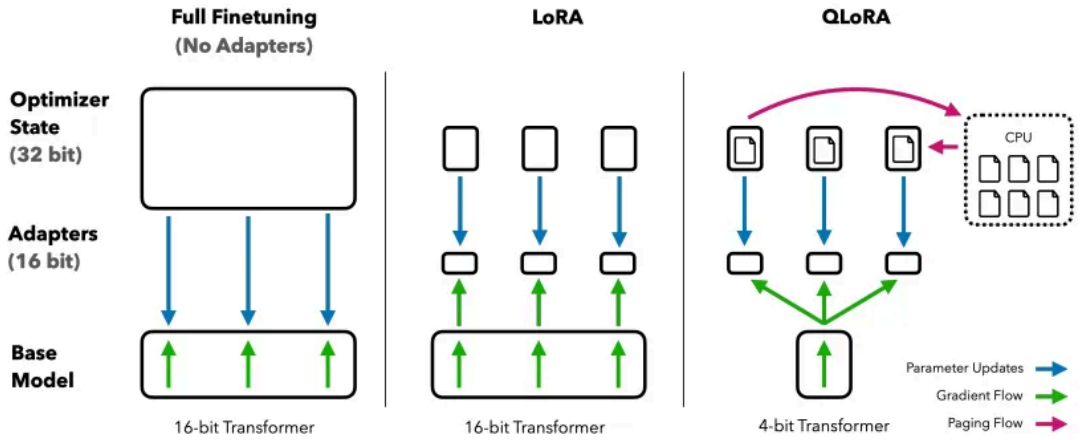

Figure 1: Different finetuning methods and their memory requirements. QLoRA improves over LoRA by quantizing the transformer model to 4-bit precision and using paged optimizers to handle memory spikes.

上图为QLoRA的训练过程图，QLoRA更多的是在工程上进行了量化和优化，从图中可知，主要有4个部分的改进：

- **QLORA**：是一种优化的4-bit量化数据类型，专为正态分布权重设计，通过结合低精度存储和中等精度计算来提升模型性能。它使用4-bit存储权重以减少内存使用，并在计算时将权重转换为16-bit的BFloat16格式以保持准确性。这种方法适用于模型加载和训练过程，旨在平衡存储效率和计算精度。

- **Double Quantization**：是一种模型量化技术，它通过对已经量化过的常量进行二次量化，进一步减少存储空间的需求。这种方法比传统的模型量化方法更能节省显存空间，每个参数平均可以节省0.37bit。例如，在65B的LLaMA模型中，这种双量化技术能够节省大约3GB的显存空间。

- **Paged Optimizers**：是一种利用NVIDIA统一内存特性的优化技术，旨在解决GPU在处理过程中偶尔出现内存溢出（OOM）的问题。该技术通过自动在CPU和GPU之间进行分页到分页的数据传输，确保GPU处理过程无错误进行。其工作原理类似于CPU内存与磁盘之间的常规内存分页机制。具体来说，Paged Optimizers为优化器状态分配分页内存，当GPU内存不足时，自动将优化器状态卸载到CPU内存中；而在需要更新优化器状态时，再将其加载回GPU内存。

- **Adapter**：为了弥补4-bit NormalFloat和Double Quantization带来的性能损失，作者采用了插入更多adapter的方法。在LoRA中，通常只在query和value的全连接层处插入adapter。而在QLoRA中，作者在所有全连接层处都插入了adapter，以增加训练参数并弥补由于精度降低而导致的性能损失。

Adapter实现

QLoRA的一个重要的改进和核心工作则是将量化的思想和LoRA的低秩适配器的思想结合到一起拿来对大模型进行微调，因此单独拎出来说，实现的代码如下：

```
if checkpoint_dir is not None:
    print("Loading adapters from checkpoint.")
    model = PeftModel.from_pretrained(model, join(checkpoint_dir, 'adapter_mo
else:
    print(f'adding LoRA modules...')
    modules = find_all_linear_names(args, model)
    config = LoraConfig(
        r=args.lora_r,
        lora_alpha=args.lora_alpha,
        target_modules=modules,
        lora_dropout=args.lora_dropout,
        bias="none",
        task_type="CAUSAL_LM",
    )
    model = get_peft_model(model, config)
```

- `find_all_linear_names`：找到所有的全连接层
- `get_peft_model`：在所有全连接层中插入LoRA模块

想要获取技术资料的同学欢迎关注公众号，进群一起交流~