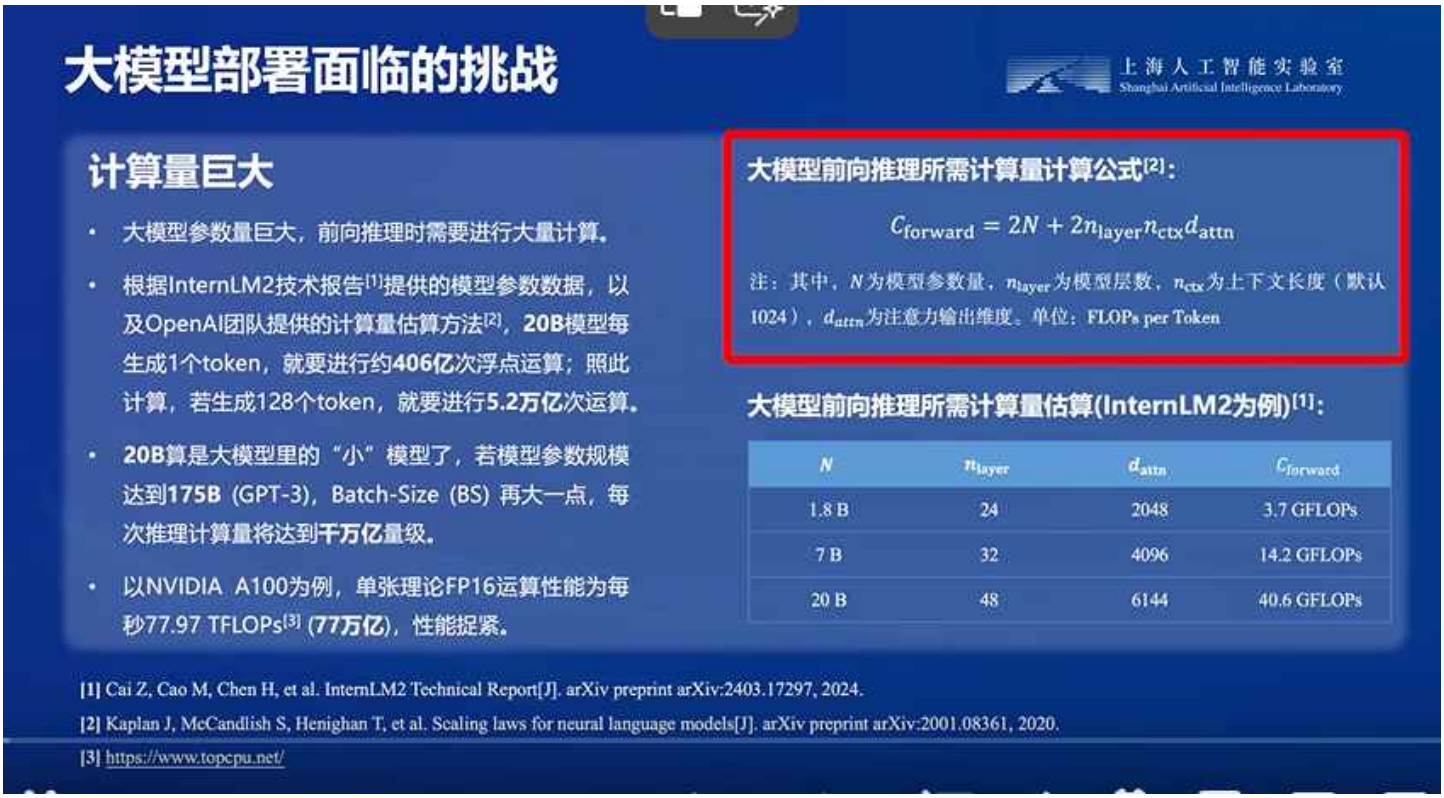
书生二期第五课笔记

复习链接：

课程视频：https://www.bilibili.com/video/BV1tr421x75B/

课程文档：https://github.com/InternLM/Tutorial/blob/camp2/lmdeploy/README.md

课程作业：https://github.com/InternLM/Tutorial/blob/camp2/lmdeploy/homework.md



大模型部署面临的挑战

访存瓶颈

- 大模型推理是“访存密集”型任务。目前硬件计算速度“远快于”显存带宽，存在严重的访存性能瓶颈。
- 以RTX 4090推理175B大模型为例，BS为1时计算量为6.83TFLOPs，远低于82.58 TFLOPs的FP16计算能力；但访存量为32.62 TB，是显存带宽每秒处理能力的30倍。

动态请求

- 请求量不确定；
- 请求时间不确定；
- Token逐个生成，生成数量不确定。

GPT3-175B推理阶段计算访存比分析 (输入1k，输出250) [1]：

BS	计算量	访存量	计算访存比
1	6.83 TFLOPs	32.62 TB	0.20
8	55.37 TFLOPs	32.67 TB	1.67
16	112.3 TFLOPs	32.73 TB	3.43

常见GPU浮点运算性能与内存带宽[2]：

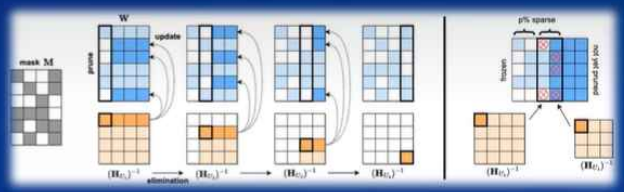
GPU	FP16算力	FP32算力	FP64算力	显存带宽	FP16算力/显存带宽
RTX 4090	82.58 TFLOPs	82.58 TFLOPs	1290 GFLOPs	1008 GB/s	81.92
A100 80G	77.97 TFLOPs	19.49 TFLOPs	9.746 TFLOPs	2039 GB/s	38.24
H100 80G	267.6 TFLOPs	66.91 TFLOPs	33.45 TFLOPs	1681 GB/s	159.2

[1] https://cloud.baidu.com/article/919629

[2] https://www.topcpu.net/

模型剪枝(Pruning)

剪枝指移除模型中不必要或多余的组件，比如参数，以使模型更加高效。通过对模型中贡献有限的冗余参数进行剪枝，在保证性能最低下降的同时，可以减小存储需求、提高计算效率。



非结构化剪枝 SparseGPT[1]，LoRAPrune[2]，Wanda[3]

- 指移除个别参数，而不考虑整体网络结构。这种方法通过将低于阈值的参数置零的方式对个别权重或神经元进行处理。

结构化剪枝 LLM-Pruner[4]

- 根据预定义规则移除连接或分层结构，同时保持整体网络结构。这种方法一次性地针对整组权重，优势在于降低模型复杂性和内存使用，同时保持整体的LLM结构完整。

Reference：

[1] Frantar E, Alistarh D. Sparsegpt: Massive language models can be accurately pruned in one-shot[C]//International Conference on Machine Learning. PMLR, 2023: 10323-10337.

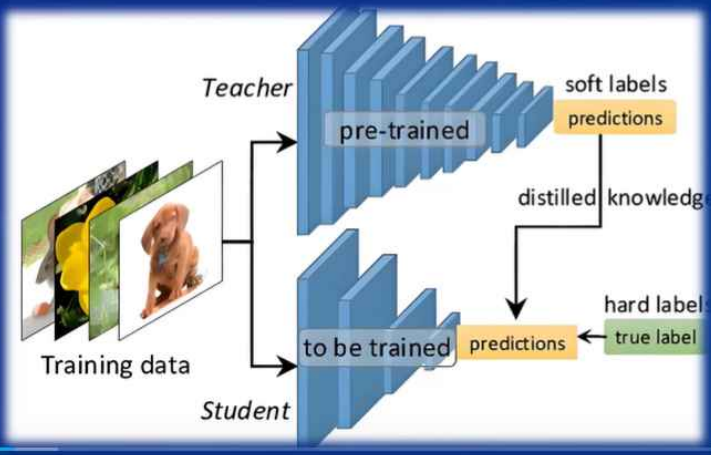
[2] Zhang M, Chen H, Shen C, et al. Loraprune: Pruning meets low-rank parameter-efficient fine-tuning[J]. 2023.

[3] Sun M, Liu Z, Bair A, et al. A simple and effective pruning approach for large language models[J]. arXiv preprint arXiv:2306.11695, 2023.

[4] Ma X, Fang G, Wang X. Llm-pruner: On the structural pruning of large language models[J]. Advances in neural information processing systems, 2023, 36: 21702-21720.

知识蒸馏(Knowledge Distillation，KD)

知识蒸馏是一种经典的模型压缩方法，核心思想是通过引导轻量化的学生模型“模仿”性能更好、结构更复杂的教师模型，在不改变学生模型结构的情况下提高其性能。



- **上下文学习(ICL)**：ICL distillation[1]
- **思维链(CoT)**：MT-COT[2]，Fine-tune-CoT[3]等
- **指令跟随(IF)**：LaMini-LM[4]

Reference：

[1] Wu M, Waheed A, Zhang C, et al. Lamini-lm: A diverse herd of distilled models from large-scale instructions[J]. arXiv preprint arXiv:2304.14402, 2023.

[2] Li S, Chen J, Shen Y, et al. Explanations from large language models make small reasoners better[J]. arXiv preprint arXiv:2210.06726, 2022.

[3] Ho N, Schmid L, Yun S Y. Large language models are reasoning teachers[J]. arXiv preprint arXiv:2212.10071, 2022.

[4] Huang Y, Chen Y, Yu Z, et al. In-context learning distillation: Transferring few-shot learning ability of pre-trained language models[J]. arXiv preprint arXiv:2212.10670, 2022.

量化(Quantization)

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

量化技术将传统的表示方法中的浮点数转换为整数或其他离散形式，以减轻深度学习模型的存储和计算负担。

量化感知训练(QAT) LLM-QAT[1]

- 量化目标无缝地集成到模型的训练过程中。这种方法使LLM在训练过程中适应低精度表示.

量化感知微调(QAF) PEQA[2]，QLORA[3]

- QAF涉及在微调过程中对LLM进行量化。主要目标是确保经过微调的LLM在量化为较低位宽后仍保持性能。

训练后量化(PTQ) LLM.int8[4]，AWQ[5]

- 在LLM的训练阶段完成后对其参数进行量化。PTQ的主要目标是减少LLM的存储和计算复杂性，而**无需对LLM架构进行修改或进行重新训练。**

通用公式：

$$ZP = \frac{\min + \max}{2}$$

$$S = \frac{\max - \min}{255}$$

量　化：$q = \text{round}\left(\frac{f - ZP}{S}\right)$

反量化：$f = q \times S + ZP$

Reference：

[1] Liu Z, Oguz B, Zhao C, et al. Llm-qat: Data-free quantization aware training for large language models[J]. arXiv preprint arXiv:2305.17888, 2023.

[2] Arshia F Z, Keyvanrad M A, Sadidpour S S, et al. PeQA: A Massive Persian Question-Answering and Chatbot Dataset[C]//2022 12th International Conference on Computer and Knowledge Engineering (ICCKE). IEEE, 2022: 392-397.

[3] Dettmers T, Pagnoni A, Holtzman A, et al. Qlora: Efficient finetuning of quantized llms[J]. Advances in Neural Information Processing Systems, 2024, 36.

[4] Dettmers T, Lewis M, Belkada Y, et al. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale[J]. Advances in Neural Information Processing Systems, 2022, 35: 30318-30332.

[5] Lin J, Tang J, Tang H, et al. Awq: Activation-aware weight quantization for llm compression and acceleration[J]. arXiv preprint arXiv:2306.00978, 2023.

3、LMDeploy简介

LMDeploy简介

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

LMDeploy 由 MMDeploy 和 MMRazor 团队联合开发，是涵盖了 LLM 任务的全套轻量化、部署和服务解决方案。核心功能包括高效推理、可靠量化、便捷服务和有状态推理。

LMDeploy

- **高效的推理：**LMDeploy开发了Continuous Batch，Blocked K/V Cache，动态拆分和融合，张量并行，高效的计算kernel等重要特性。InternLM2推理性能是vLLM的 1.8 倍。
- **可靠的量化：**LMDeploy支持权重量化和k/v量化。4bit模型推理效率是FP16下的2.4倍。量化模型的可靠性已通过OpenCompass评测得到充分验证。
- **便捷的服务：**通过请求分发服务，LMDeploy 支持多模型在多机、多卡上的推理服务。
- **有状态推理：**通过缓存多轮对话过程中Attention的k/v，记住对话历史，从而避免重复处理历史会话。显著提升长文本多轮对话场景中的效率。

LMDeploy核心功能

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

模型高效推理 参考命令：lmdeploy chat -h

- TurboMind是LMDeploy团队开发的一款关于 LLM 推理的高效推理引擎。它的主要功能包括：LLaMa 结构模型的支持，continuous batch推理模式和可扩展的 KV 缓存管理器。

模型量化压缩 参考命令：lmdeploy lite -h

- **W4A16量化(AWQ)：**将 FP16 的模型权重量化为 INT4，Kernel 计算时，访存量直接降为 FP16 模型的 1/4，大幅降低了访存成本。Weight Only 是指仅量化权重，数值计算依然采用 FP16（需要将 INT4 权重反量化）。

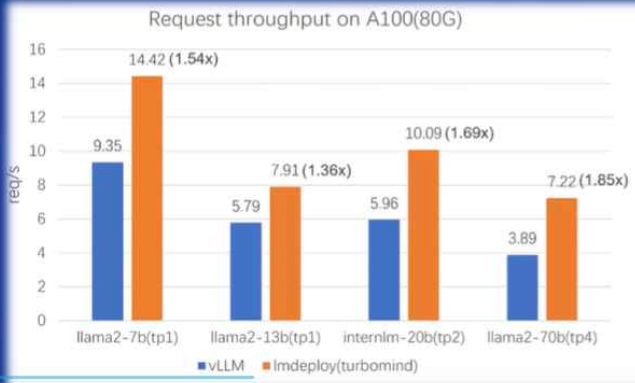
服务化部署 参考命令：lmdeploy serve -h

- 将LLM封装为HTTP API服务，支持Triton扩展。

LMDeploy性能表现

上海人工智能实验室 Shanghai Artificial Intelligence Laboratory

LMDeploy TurboMind 引擎拥有卓越的推理能力，在各种规模的模型上，每秒处理的请求数是 vLLM的1.36~1.85 倍。在静态推理能力方面，TurboMind 4bit 模型推理速度（out token/s）远高于FP16/BF16推理。在小batch时，提高到2.4倍。



LMDeploy推理视觉多模态大模型

上海人工智能实验室 Shanghai Artificial Intelligence Laboratory

新版本的lmdeploy支持了对多模态大模型llava的支持！可以使用pipeline便捷运行。

```python
from lmdeploy import pipeline
from lmdeploy.vl import load_image

pipe = pipeline('liuhaotian/llava-v1.6-vicuna-7b')

image = load_image('https://raw.githubusercontent.com/open-mmlab/mmdeploy/main/tests/data/tiger.jpeg')
response = pipe(('describe this image', image))
print(response)
```

[OUTPUT] The image shows a tiger lying down on a grassy area. The tiger is facing the camera with its head slightly tilted to the side, giving it a curious or attentive look……

LMDeploy更多支持模型

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

模型	参数量	模型	参数量
Llama	7B-65B	Llama2	7B-70B
InternLM	7B-20B	InternLM2	1.8B-20B
Llava	7B-13B	InternLM-XComposer	7B
QWen	7B-72B	Qwen-VL	7B
QWen1.5	0.5B-72B	QWen1.5-MoE	A2.7B
Baichuan	7B-13B	Baichuan2	7B-13B
Code Llama	7B-34B	ChatGLM2	6B
Falcon	7B-180B	YI	6B-34B
Mistral	7B	Mixtral	8x7B
DeepSeek-MoE	16B	DeepSeek-VL	7B
Gemma	2B-7B	Dbrx	132B