LLM Inference

Summer

Introduction

This document summarizes efficient techniques and challenges in large language model (LLM) inference, focusing on memory optimization, quantization, and speculative decoding methods.

Memory Requirements for LLM Inference

- Storing a 70B parameter model in FP16 precision requires approximately 140GB VRAM.
- Deployment challenges:
  1. GPU memory constraints.
  2. Token throughput and latency.
  3. Handling multiple concurrent requests.

Optimizations for LLM Inference

Key-Value (KV) Caching

- Speeds up the decoding phase by caching key-value pairs.
- Prefill stage processes inputs, while decoding generates tokens iteratively.
- Trade-off: Increased memory usage for faster computation.

Multi-Query and Grouped-Query Attention

- **Multi-Query Attention (MQA)**: Shares K and V matrices across heads to reduce memory.
- **Grouped-Query Attention (GQA)**: Shares K and V across groups of heads for better speed-quality trade-off.

Quantization Techniques

- **Post-Training Quantization (PTQ)**: Converts weights to low-bit formats after training.
- **Quantization-Aware Training (QAT)**: Simulates quantization effects during training.
- **Linear Quantization**:
  - Symmetric and asymmetric approaches.
  - Higher granularity (per-channel) improves reconstruction but increases memory.
- **Advanced Methods**:
  - LLM.int8() isolates outlier dimensions for mixed precision.
  - SmoothQuant scales salient channels up pre-quantization.
  - AWQ keeps critical weights in high precision.

Speculative Decoding

- Uses a smaller draft model to predict tokens, verified by the target model.
- Reduces forward passes, making decoding 2–3 times faster.

PagedAttention

- Minimizes memory fragmentation by storing non-contiguous keys and values.
- Reduces memory waste from 60–80% to 4%.

Efficient Execution

- Tools for local execution:
  - **Ollama**: For OpenWebUI.
  - **LM Studio**: Lightweight execution.
  - **GPT4All**: Open-source solutions.

References

- Zhou et al. (2024). *A survey on efficient inference for large language models.* `https://arxiv.org/abs/2404.14294`
- Leviathan et al. (2023). *Fast inference from transformers via speculative decoding.*
- Dettmers et al. (2022). *GPT3.int8(): 8-bit matrix multiplication for transformers at scale.*