Optimizing GPT-2 Training and Inference: Hybrid Parallelism using Megatron-LM, FlashAttention, and Quantization

Rahul Kadam
rsk8552@nyu.edu
NYU

Varijaksh Katti
vvk4812@nyu.edu
NYU

***Abstract*—This project aims to optimize the training and inference of GPT-2 by employing advanced techniques such as hybrid parallelism, FlashAttention, and quantization. By distributing the computational workload across multiple GPUs using Megatron-LM, introducing memory-efficient FlashAttention, and leveraging quantization-aware training, we aim to reduce training time, optimize memory usage, and speed up inference without compromising accuracy.**

I. Introduction

The goal of this project is to optimize GPT-2's training and inference through a combination of advanced techniques:

- **Hybrid Parallelism:** Utilize Megatron-LM to implement data and model parallelism to distribute the computational workload across multiple GPUs.
- **FlashAttention:** Replace standard attention with FlashAttention to reduce memory usage and accelerate long-context training.
- **Quantization:** Apply quantization-aware training (QAT) to optimize inference time with minimal accuracy loss.

II. Challenges

- Efficient communication between multiple GPUs using Megatron-LM for data and model parallelism.
- Managing memory bottlenecks in attention mechanisms for long sequences with FlashAttention.
- Balancing the trade-off between inference speed and model accuracy in quantization.
- Ensuring compatibility between hybrid parallelism and quantization to avoid training and inference bottlenecks.

III. Approach and Techniques

A. Hybrid Parallelism with Megatron-LM

- **Data Parallelism:** Split input data across GPUs, with gradient synchronization.
- **Model Parallelism:** Distribute model parameters across GPUs to enable training on limited memory.
- **Megatron-LM:** Handle both data and model parallelism effectively, scaling GPT-2 across multiple GPUs.

B. FlashAttention for Memory-Efficient Attention

- **FlashAttention:** Reduce memory usage and speed up computations, particularly for long-context sequences.
- **Memory Optimization:** Minimize memory access overhead, enabling longer sequence training.

C. Quantization for Efficient Inference

- **Quantization-Aware Training (QAT):** Simulate precision reduction during training to maintain accuracy.
- **Custom CUDA Kernels:** Implement optimized operations for inference on NVIDIA V100 GPUs.

IV. Implementation Details

Hardware: 4-8 NVIDIA V100 GPUs. **Software:** PyTorch, Megatron-LM, CUDA, NCCL for multi-GPU communication.

Dataset: WebText (40 GB), used to train GPT-2 with diverse web-based text data.

V. Demo Plan

- **Demo Objective:** Showcase performance improvements in training and inference.
- **Baseline vs Optimized:** Compare standard GPT-2 with our optimized approach.
- **Performance Metrics:** Training time, GPU utilization, inference latency, and memory consumption.

VI. Expected Outcomes

- **Reduced Training Time:** 30-50% reduction in training time.
- **Memory Efficiency:** Longer sequences processed with optimized memory usage.
- **Inference Speedup:** 2-4x improvement in inference time through quantization.
- **Scalability:** Demonstrate GPT-2 scalability across multiple GPUs.

VII. References

References

[1] Narayanan et al., “Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism,” 2021.

[2] Dao et al., “FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness,” 2022.

[3] Raghuraman Krishnamoorthi, “Quantizing Deep Convolutional Networks for Efficient Inference: A Whitepaper,” 2018.