Mixed-Precision Inference Optimization: By utilizing Tensor Cores on a GPU

Amir Mahdi Mirtajadini

Project Overview

Neural network inference is a compute-intensive task traditionally performed using single-precision floating-point arithmetic. However, modern GPU architectures, such as the NVIDIA Ada Lovelace, introduce specialized hardware units known as Tensor Cores designed to accelerate matrix operations using lower precision formats (FP16, BF16) while maintaining sufficient accuracy via mixed-precision accumulation. This project aims to implement and analyze the performance trade-offs of mixed-precision inference. The primary objective is to quantify the speedup and efficiency gains provided by Tensor Cores compared to general-purpose CUDA cores on the NVIDIA RTX 4060 GPU.

Proposed Workload

The investigation will center on a **Dense Linear Layer Comparison**. The workload is defined as $Y = \sigma(W \cdot X + B)$, where σ is the ReLU activation. This structure is the fundamental building block of MLPs and Transformers.

- **Baseline Implementation:** Naive and Tiled Matrix Multiplication kernels written in CUDA C++ using FP32.
- **Target Implementation:** cuBLAS-accelerated GEMM operations utilizing Tensor Cores with FP16/BF16 inputs and FP32 accumulation.

Execution Strategy

To ensure a fair comparison between hardware units, the implementation is divided into two distinct phases.

3.1 Phase 1: CUDA Core Implementation

I will develop a custom CUDA kernel that explicitly runs on the Streaming Multiprocessors' FMA (Fused Multiply-Add) units.

- **Memory Coalescing:** Ensuring global memory accesses are aligned to 128-byte transaction lines.
- **Shared Memory Tiling:** Implementing block-tiling to maximize data reuse and reduce global memory bandwidth pressure.
- **Precision:** Both FP32 and FP16 versions will be written to isolate the speedup gained purely from data type reduction versus hardware acceleration.

3.2 Phase 2: Tensor Core Implementation

I will utilize **cuBLAS** to offload the computation to the RTX 4060's 4th Gen Tensor Cores.

- **Data Layout:** Handling the necessary data transformations (e.g., NCHW vs NHWC) if required by the library for optimal performance.
- **Mixed Precision:** Utilizing the `cublasGemmEx` API to specify low-precision compute types (FP16/BF16).

Experimental Setup

All experiments will be conducted on the following system:

Component	Specification
CPU	Intel Core i7 13620H
GPU	NVIDIA GeForce RTX 4060 (Ada Lovelace)
RAM	32GB DDR4
OS	Windows 11 Home
Profilers	NVIDIA Nsight Compute, Nsight Systems

Evaluation Methodology and Profiling

The core of this project is not just implementation, but detailed profiling using **NVIDIA Nsight Compute** and **Nsight Systems**.

5.1 Key Performance Indicators

- **SOL Analysis:** I will measure the achieved compute throughput against the theoretical peak of the RTX 4060.
  - *Compute SOL %:* How close are we to the peak FP32/FP16 TFLOPS?
  - *Memory SOL %:* How close are we to the peak DRAM bandwidth?
- **Latency Distribution:** Breaking down the total inference time into:
  - Host-to-Device (H2D) transfer time.
  - Kernel execution time (Compute).
  - Device-to-Host (D2H) transfer time.
- **Numerical Accuracy:** A standard Mean Squared Error (MSE) comparison will be conducted:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_{fp32} - Y_{mixed})^2$$

  This will ensure that the performance gains from reduced precision do not come at the cost of unacceptable inference degradation.

Setup Suitability

The **RTX 4060** is an ideal candidate for this study as it supports the latest **Bfloat16** format, allowing for a comparative study between FP16 (standard half precision) and BF16 (high dynamic range half precision), a feature not available on older Pascal or Turing architectures.

Expected outcomes

This proposal sets a clear path to understanding the cost vs. benefit of mixed-precision inference. By isolating the CUDA cores from the Tensor Cores, the final report will provide a definitive guide on when and how to utilize mixed precision for ML inference on consumer-grade hardware. It is expected that the Tensor Core implementation will show significant speedups (potentially 2x-4x) over the FP32 baseline, provided the matrix dimensions are large enough to saturate the GPU. The report will document the crossover point where the overhead of data casting and library calls is outweighed by the raw compute throughput of the Tensor Cores.