Project: Vision Model Optimization with Quantization & Efficient Attention

Team:
Jayraj Pamnani (jmp10051)
Puneeth Kotha (pk3058)

TLDR: Build a production-ready ViT image-classifier that uses aggressive quantization (4/8-bit) and optimized attention (SDPA + FlashAttention2) to cut model size and latency while holding accuracy within ~2% of full precision, target deployment on GPUs or edge devices.

1. Objectives

Primary: Produce an efficient ViT-based image classification pipeline with low memory footprint and low inference latency suitable for consumer GPUs/edge.

Quantitative goals

- Maintain Top-1 accuracy within 2–3% of FP32 baseline.
- Reduce model disk/memory size by 4–8× (4-bit target).
- Achieve 2–3× inference speedup vs. baseline through quantization + attention optimizations.
- Implement and benchmark multiple ViT variants and quantization levels.

2. Key Challenges

Model/algorithm:

- Aggressive quantization can degrade fine-grained spatial features and attention maps.
- Early (patch-embedding) quantization risks losing crucial spatial information.
- Preserving attention-map fidelity when fusing SDPA kernels.
  Systems/hardware
- GPU memory bandwidth and larger image tensors increase pressure.
- Preprocessing (resize/augment) adds CPU overhead for throughput.
- Mixed-precision interplay (FP16/BF16) vs. quantized inference complexity.

3. Approach & Techniques

Model: ViT (ViT-B/16, ViT-L/16 variants).

Quantization:

- PTQ (4/8-bit) via bitsandbytes; QAT (fake-quant) for critical layers.
- Layer-wise mixed precision: first/last layers kept higher precision.

Attention & Kernel Optim

- Apply SDPA fusion for self-attention; integrate FlashAttention-2.
- Profile and optimize multi-head attention kernel fusion.

Parameter-efficient fine-tuning

- LoRA adapters and QLoRA (4-bit + LoRA) for memory-efficient fine-tune.
  Profiling & Benchmarking
- PyTorch Profiler for CUDA kernels; measure latency, throughput (img/s), and GPU memory per component (patch embed, attention, MLP).

4. Implementation

Hardware: NVIDIA T4 (16GB) min; A10/A100 or RTX-4090 recommended for larger models/batches. CPU ≥8 cores, RAM ≥32GB, SSD ≥200GB.

Software / libs:

- torch ≥2.0, torchvision, transformers, timm
- bitsandbytes, accelerate, peft (LoRA), QLoRA tooling
- datasets (HuggingFace), albumentations
- profiling/visuals: PyTorch Profiler, TensorBoard, Weights & Biases

Datasets: ImageNet-1k (primary, ~1.28M train / 50k val) **or** Tiny-ImageNet for faster prototyping.

5. Evaluation Metrics

- Accuracy: Top-1 / Top-5, F1 where relevant
- Latency: ms/image, p99, throughput (FPS)
- Model size: disk & GPU footprint
- Kernel profiling: per-op CUDA times and memory bandwidth utilization

6. Expected Outcomes & Contributions

- Practical: 3–4× smaller models (4-bit) with 2–3× inference speedups enabling consumer/edge deployment.
- Research: First systematic study of QLoRA + SDPA on ViTs; kernel-level profiling and layer-wise quantization guidelines.
- Deliverables: Reproducible code, benchmark suite, profiling reports, and deployment notes.

7. Demo Planned

During the demo, we will showcase the following:

- **Performance Comparison:** Side-by-side results of the original full-precision Vision Transformer versus our optimized quantized model, highlighting differences in accuracy, inference speed, and model size.
- **Real-Time Inference:** Live image classification on a consumer GPU or edge device to demonstrate faster inference and lower memory usage.
- **Profiling Insights:** Visualization of CUDA kernel profiling using PyTorch Profiler and TensorBoard to show how SDPA and quantization improve efficiency at the kernel level.

- **Scalability & Deployment:** A brief demo of how the optimized model can be easily deployed in practical settings such as edge devices or real-time image recognition systems, showing its real-world applicability.

8. Core References

Research Papers:
[1] Dosovitskiy et al., *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, ICLR 2021.
https://arxiv.org/abs/2010.11929

[2] Dettmers et al., *QLoRA: Efficient Finetuning of Quantized LLMs*, NeurIPS 2023.
https://arxiv.org/abs/2305.14314

[3] Dao et al., *FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning*, ICLR 2024.
https://arxiv.org/abs/2307.08691