CS4681 - Advanced Machine Learning
Progress Evaluation
210001R

Literature Review

The transformer architecture has been the de facto natural language processing and beyond deep learning models' backbone in the last few years, where long-range dependencies are enabled by attention mechanisms. However, typical attention mechanisms give quadratic time and space complexity with respect to sequence length, which gives rise to significant inefficiencies in large models as well as in long sequences.

To alleviate this bottleneck, Dao et al. (2022) introduced FlashAttention [1] , a novel attention algorithm based on tiling, recomputation, and GPU-efficient kernels for exact attention with reduced memory usage and increased speed. One of the novelties is to calculate attention on high-bandwidth GPU SRAM-fitting blocks and thus read and write minimally to lower-bandwidth memory. Their benchmarks achieved 2–4× speedups from state-of-the-art implementations without loss of numerical accuracy, and FlashAttention emerged as the go-to benchmark for scaling transformers to longer sequences.

Following these efforts, Dao et al. (2023) proposed FlashAttention-2 [2], where the parallelism and scheduling of the algorithm were further enhanced to maximize the GPU streaming multiprocessors utilization. Through re-architecting the thread-block organization and enabling additional overlap between computation and memory access, FlashAttention-2 achieved close-to-ideal GPU utilization. This was 2× better throughput compared to the baseline FlashAttention with robust scaling to larger models such as GPT-style models. Against most near-optimal attention approximations, FlashAttention-2 preserved result accuracy and established a state-of-the-art efficiency baseline for attention mechanisms that is not paralleled by accuracy trade-offs.

Shah et al. (2024) presented FlashAttention-3 [3], which shatters the quadratic attention-only bottleneck.Through synergistic kernel fusion techniques, mixed-precision optimizations, and end-to-end pipeline reconfigurations, FlashAttention-3 accelerated not only attention but also matrix multiplications and softmax operations that comprised the typical transformer workloads. The release also closed the real model throughput-theoretical hardware peak performance gap by offering speedup during both training and inference for a range of sequence lengths. In particular, FlashAttention-3 demonstrated how it is possible to train language models with billions of parameters with hardware-aware yet exact architectures. Collectively, these efforts demonstrate a well-trodden path: from FlashAttention's memory-aware architecture, through FlashAttention-2's hardware-aware scheduling, to FlashAttention-3's system-level kernel fusion.

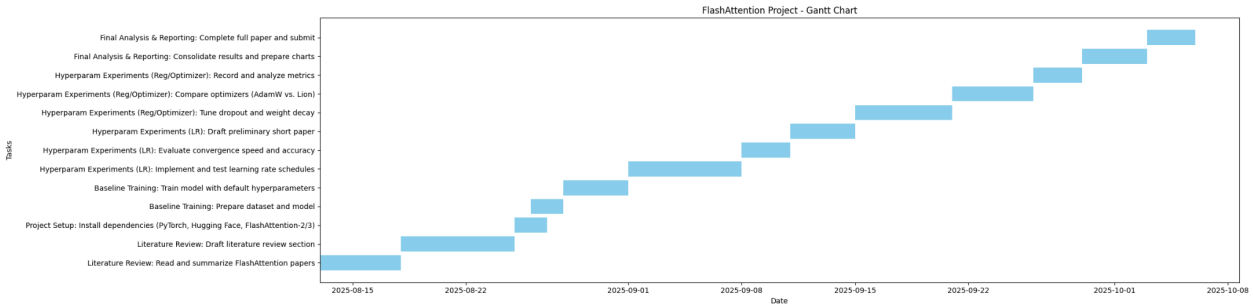
Even though these advances provide significant efficiency gains, they practically only treat speed and memory saving with limited attention paid to the intertwinement of training practice, i.e., hyperparameter search or regularization, and efficient attention. That gap is an opportunity: combining FlashAttention kernels with optimally tuned training methods can provide additional model performance, stability, or convergence rate gains. By attending to these complementary factors, research can widen the practical usefulness of FlashAttention-based models beyond brute effectiveness.

Methodology

In this project, I will extend FlashAttention by focusing on training-level optimisations rather than modifying the kernels. Beginning with an implementation of FlashAttention-2/3 as a baseline, I will train a transformer of medium size (e.g., GPT-2 on WikiText-103 or BERT on GLUE). For starters, I will gauge baseline efficiency and accuracy under standard training configurations. Next, incremental optimizations such as variable learning rate schedules, optimizer variations (AdamW vs. Lion), and regularization methods (dropout, weight decay) will be introduced.

Quantitatively, throughput, memory use, convergence rate, and model accuracy will be measured between the baseline and optimised models. It is anticipated to illustrate that training-level optimisations can deliver measurable gains in efficiency and stability and FlashAttention's hardware-aware benefits.

Timeline



References

[1] Dao, T., Fu, D., Ermon, S., Rudra, A., & Ré, C. (2022). Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, *35*, 16344-16359.

[2] Dao, T. (2023). Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

[3] Shah, J., Bikshandi, G., Zhang, Y., Thakkar, V., Ramani, P., & Dao, T. (2024). Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *Advances in Neural Information Processing Systems*, *37*, 68658-68685.