Developing Optimized Large Language Models on Limited Compute Resources

kasinadhsarma
Email: kasinadhsarma@gmail.com

***Abstract*—Large language models (LLMs) have demonstrated remarkable performance across a wide range of natural language tasks. However, the computational resources required to train these models at scale remain a significant challenge, particularly in resource-constrained environments. This paper proposes a holistic optimization framework that combines data-centric techniques, compute efficiency improvements, and architectural enhancements to enable the development of high-quality LLMs on limited hardware. We outline our methodology and proposed experimental evaluation plan. Our preliminary analysis suggests that such an approach could potentially yield up to a 30% reduction in training compute while maintaining competitive downstream task performance. This framework aims to democratize LLM development by reducing the computational barriers and fostering more sustainable scaling strategies.**

***Index Terms*—Large Language Models, Compute Efficiency, Data Optimization, Mixture-of-Experts, Dynamic Inference.**

I. Introduction

Recent advances in large language models have led to significant breakthroughs in natural language processing [1], [2]. However, the computational resources required to train these models have grown exponentially, posing a major challenge for researchers operating under resource constraints. In this work, we propose a comprehensive framework for optimizing LLM training on limited compute resources by integrating data-centric optimizations, compute efficiency techniques, and architectural enhancements. The main contributions of this paper are:

- A data filtering and augmentation pipeline that enhances training data quality and diversity.
- Compute efficiency techniques, including mixed-precision training, NUMA-aware memory allocation, and dynamic batching.
- Architectural optimizations using sparsely-gated Mixture-of-Experts (MoE) layers and dynamic inference strategies.
- A detailed experimental design aimed at reducing training compute while maintaining competitive performance on downstream tasks.

II. Related Work

A. Scaling Laws for LLMs

The scaling behavior of LLMs has been extensively studied in recent literature. [3] established a power-law relationship between model size, dataset size, and compute requirements. Subsequent work by [4] refined these relationships by defining compute-optimal model scaling. However, these scaling laws assume abundant computational resources and do not fully address resource-constrained settings.

B. Efficient LLM Architectures

Prior works have investigated various approaches to improve LLM efficiency. Distillation techniques (e.g., DistilBERT [5]) and sparse architectures such as Mixture-of-Experts (MoE) layers [6], [7] have shown promise. Yet, these methods often focus on architectural changes in isolation and do not consider the broader context of data and compute optimizations.

III. Methodology

A. Data-Centric Optimizations

We propose a multi-stage data pipeline that:

- Applies advanced filtering techniques (e.g., deduplication and n-gram overlap filtering) to remove low-quality data.
- Augments the filtered corpus with synthetic examples in underrepresented domains using back-translation and text infilling.
- Implements curriculum learning, gradually increasing the complexity of training examples.

B. Compute Efficiency Techniques

Our compute efficiency strategies include:

- **Mixed-Precision Training:** Utilizing BFloat16 and INT8 formats to reduce memory footprint and accelerate computation.
- **NUMA-Aware Memory Allocation:** Ensuring efficient data distribution in multi-socket systems.
- **Dynamic Batching:** Adapting batch size and sequence length based on current compute availability.

C. Architectural Optimizations

We integrate architectural innovations such as:

- **Sparsely-Gated Mixture-of-Experts (MoE) Layers:** To enable conditional computation and increase model capacity without proportional compute cost.
- **Dynamic Inference Techniques:** Including adaptive early exiting and speculative sampling to reduce inference computation.

D. Experimental Setup

Our experimental plan targets model scales from 500M to 1B parameters. We will compare baseline dense Transformer architectures against models incorporating our optimization techniques. All models are to be trained using the Adam optimizer with a cosine learning rate schedule. Evaluation metrics will include test loss, perplexity, downstream task performance (e.g., question answering, natural language inference), FLOPs per token, and energy consumption.

IV. Proposed Evaluation and Expected Outcomes

Given the absence of complete experimental results at this stage, we detail our evaluation methodology and expected outcomes:

- **Compute Efficiency:** We plan to quantify the reduction in FLOPs per token using standard benchmarking tools. Our hypothesis is that our techniques can reduce compute requirements by 20% to 30%, thereby lowering energy consumption and training time.
- **Downstream Task Performance:** We expect the optimized models to maintain competitive performance (with less than 1% average accuracy drop) on tasks such as language modeling, question answering, and natural language inference.
- **Scalability Analysis:** By analyzing performance across different model scales, we will determine the effectiveness of our data-centric and architectural optimizations in resource-constrained environments.

These evaluations will be conducted in future work, and preliminary results will be reported in subsequent publications.

V. Discussion

Although experimental results are not yet available, our framework is designed based on established research and emerging trends in LLM scaling [3]–[6]. Our approach addresses the dual challenge of reducing training compute while ensuring high-quality downstream performance. The integration of data-centric methods, efficient compute techniques, and architectural innovations is expected to push the boundaries of resource-constrained LLM development. Future experiments will validate these hypotheses and provide further insights into the tradeoffs between compute efficiency and model performance.

VI. Conclusion

This paper presents a comprehensive framework for optimizing the development of large language models on limited compute resources. By combining data-centric optimizations, compute efficiency techniques, and architectural enhancements, our approach is designed to reduce training compute requirements by up to 30% while maintaining competitive performance. While we have not yet completed our experimental validation, our proposed evaluation plan outlines the methodology for measuring compute efficiency, downstream performance, and scalability. Future work will focus on experimental validation and theoretical refinement of scaling laws in resource-constrained environments.

References

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[2] A. Chowdhery *et al.*, "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[3] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[4] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, and T. Cai, "Training compute-optimal large language models," *arXiv preprint arXiv:2203.15556*, 2022.

[5] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[6] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, and Q. V. Le, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.

[7] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *arXiv preprint arXiv:2101.03961*, 2022.