Scaling Down: Optimizing LLM Size for Specialized Tasks with Large Model Training

Aditya Ashvin
ashvin@usc.edu

Albert Kaloustian
kalousti@usc.edu

Pannawat Chauychoo
pchauych@usc.edu

Pooridon Rattanapairote
prattana@usc.edu

Abstract

This project explores how small a Large Language Model (LLM) can be while outperforming a larger model on a specific task when trained by one or more larger models. Using the TinyLLM framework, we will compare student-teacher models with collaborative learning to optimize performance and efficiency.

1 What is the problem?

The problem our team aims to address revolves around the optimization of LLMs, particularly investigating how a small LLM can outperform a larger LLM in a specific domain when trained by one or multiple larger models. In recent years, LLMs have made significant advances in tasks such as natural language processing, reasoning, and problem-solving, but this progress came at the cost of increasingly large model sizes.

Models like GPT-4 and PaLM require enormous computational resources because of emergent abilities [8], making them costly and energy-intensive to train and deploy. This is against our desire to democratize AI in a sustainable way. If we can demonstrate that smaller models can match or even surpass the performance of bigger model(s) used to train them, it could lead to more efficient usage of our limited resources and improve LLM's on edge devices. Time permitting, we could compare the student-teacher setup to dynamic collaboration between models to dive deeper into how AI systems learn and work together.

This problem is challenging due to the inherent trade-offs between model size, performance, and generalization ability. As models become smaller, they typically lose the capacity to capture the same level of complexity and detail as their larger counterparts. Fine-tuning smaller LLMs to retain the reasoning and language capabilities of larger models while keeping resource usage low is a significant technical hurdle worth tackling.

In addition, resource constraints make experimenting with various training architectures and collaborative methods a difficult endeavor. The challenge is heightened when smaller models must be optimized to perform well in specific domains (e.g., biomedical reasoning) while maintaining generalization across various tasks.

2 How is it currently approached?

The TinyLLM paper [6] exemplifies how to optimize LLMs while retaining performance using the student-teacher model framework. In this strategy, larger LLMs (teachers) train smaller models (students) to conserve resources while maintaining task-specific performance. Student models learn not just to provide correct results, but also to imitate the reasoning processes of larger models.

This method decreases memory and computational costs, allowing smaller models to match or exceed the teacher model on specific datasets [5]. However, one significant disadvantage is that smaller models may struggle with generalisation across varied tasks, especially if the training data is not closely linked with the test domain. Our effort expands on this basis by determining how small a model can be while still outperforming a larger model on a specific subject, thus addressing the trade-off between model size and specialised performance.

In the TinyLLM framework, a range of training strategies have been studied to enable knowledge distillation, including complete fine-tuning and parameter-efficient techniques like LoRA (Low-Rank Adaptation) [4] for optimising tiny models. Full fine-tuning, while powerful, can be resource-intensive because it changes all of the student model's parameters. In contrast, LoRA merely alters a subset of parameters, making it more efficient but occasionally less effective on complex jobs.

TinyLLM and related literature evaluations typically use widely accepted benchmarks, such as OpenBookQA [1], ARC [2], and BioASQ [7], which cover areas like commonsense reasoning, scientific question-answering, and domain-specific tasks. These datasets enable rigorous testing of a model's reasoning ability and domain-specific performance. While the TinyLLM paper's findings are convincing, indicating that smaller models can beat larger ones in some situations, these benchmarks are largely concerned with general reasoning rather than specialised tasks.

3 How do you plan to approach it?

Our planned approach is to investigate how small an LLM can be while still outperforming a larger LLM on a specific subject. We will focus primarily on building upon the methodology outlined in the TinyLLM paper, using student-teacher models where a smaller LLM learns from one or multiple larger models [5]. Specifically, we will explore various training methods, including full fine-tuning and LoRA (Low-Rank Adaptation), to optimize smaller models for performance on a single subject.

Our project will involve iterative experimentation to identify the optimal size and structure of the student LLM that yields the best performance while minimizing computational cost. We will also examine whether direct teaching by a larger model leads to better results compared to collaboration between smaller models, thus exploring both "learning from" and "working with" larger models.

We plan to evaluate our approach by testing on widely recognized datasets like OpenBookQA, ARC, and BioASQ, which cover reasoning and domain-specific tasks, on CARC. These benchmarks will allow us to measure how well smaller models can perform on specialized subjects compared to larger models. More specifically, we will establish a baseline performance using the larger models first and then compare it against the performance of the smaller models.

Success will be defined by the smaller models achieving comparable or superior accuracy on the selected tasks, while reducing memory and computation requirements. To measure the difference, we could track the memory usage of both the smaller and larger models when testing and compare the results. Additionally, we could to quantify whether being taught by one or multiple larger models leads to superior performance so we will compare their respective performance on the benchmark datasets. To ensure the best results, we will use pre-loss training control, as outlined in [3], to maximize the performance of the student LLM since size has been discovered to not be the main driver of emergent abilities.

A potential obstacle we foresee is that smaller models may struggle to perform well across datasets or lose accuracy if scaled down too aggressively. While techniques like LoRA can help fine-tune efficiently, they may not always be effective in maintaining high performance on highly complex reasoning tasks. To overcome these issues, we expect to experiment and adjust model size and training methods across multiple iterations. We will provide more specifics changes on the parameters as we progress. Additionally, running large-scale experiments can be resource-intensive. Therefore, we will focus our tests on a narrow domain rather than attempting across multiple domains to manage computational costs.

For our project timeline, we plan the following week-to-week breakdown:

- Weeks 1-2: Conduct a literature review and analyze the TinyLLM framework, emphasizing on the methodology and benchmarks from prior research.
- Weeks 3-4: Develop and implement the student-teacher framework, selecting model based on sizes and fine-tuning methods.
- Weeks 5-6: Begin running experiments on chosen datasets, emphasizing on domain-specific tasks such as math and logical reasoning or biomedical reasoning.
- Weeks 7-8: Perform iterative testing and analysis, adjusting the approach based on initial finding.
- Weeks 9-10: Finalize experiments, document results, and prepare for the final presentation and deployment.

By the end of this timeline, we aim to have a well-tested, concrete analysis of the smallest possible LLM that can outperform a larger model on a specialized subject, along with conclusions about the effectiveness of teaching versus collaboration.

References

[1] Pratyay Banerjee, Kuntal Kumar Pal, Arindam Mitra, and Chitta Baral. Careful selection of knowledge to solve open book question answering, 2019.

[2] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018.

[3] Zhengxiao Du, Aohan Zeng, Yuxiao Dong, and Jie Tang. Understanding emergent abilities of language models from the loss perspective, 2024.

[4] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[5] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression, 2019.

[6] Yijun Tian, Yikun Han, Xiusi Chen, Wei Wang, and Nitesh V. Chawla. Tinyllm: Learning a small student from multiple large language models, 2024.

[7] George Tsatsaronis, John Pavlopoulos, Vassilis Karkaletsis, and Vassilis Koutkias. Bioasq: A challenge on big data for biomedical semantic indexing and question answering. In *Proceedings of the 24th International Conference on Information and Knowledge Management (CIKM)*, pages 2157–2160, 2015.

[8] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022.