YILDIZ TECHNICAL UNIVERSITY
FACULTY OF ELECTRICAL AND ELECTRONICS
DEPARTMENT OF COMPUTER ENGINEERING

ACCELERATED LARGE LANGUAGE MODEL PRE-TRAINING

Talha Çelik, Hüseyin Said Arıcı, Himmet Toprak Kesgin
talha.celik1@std.yildiz.edu.tr, said.arici@std.yildiz.edu.tr, tkesgin@yildiz.edu.tr

Özet

Büyük dil modellerinin yaygınlaşması, doğal dil işleme alanında ilerleme sağlarken, yüksek işlem gücü ve uzun eğitim süreleri sorun yaratmıştır. Bu projede, eğitim süresini azaltmak için veri kümesi ve model katmanlarının bölünerek ayrı ayrı eğitilmesi stratejisi incelenmiştir. Llama-3.2-1B modeliyle Cosmpedia, GSM8K Train ve MetaMathQA veri kümeleri üzerinde deneyler yapılmıştır. Veriler dört parçaya ayrılıp modelin farklı katmanlarında eğitilmiş, alt modeller paralel çalıştırılmıştır. Rastgele ağırlıklarla başlatılan bir senaryo da denenmiştir. Başarı, ARC, Hellaswag ve GSM8K metrikleriyle ölçülmüştür. Amaç, süreyi kısaltırken performansı korumaktır.

Abstract

Large language models have advanced NLP but increased training costs. This project reduces training time by splitting datasets and training different model layers separately. Using Llama-3.2-1B, experiments were run on Cosmpedia, GSM8K Train, and MetaMathQA. Data was split into four parts to train model layers in parallel. A scenario with random weight initialization was also tested. Performance was measured using ARC, Hellaswag, and GSM8K. The goal is faster training with minimal loss in performance.

I. Introduction

Large language models (LLM), which have pioneered many innovations in natural language processing (NLP) in recent years, cause great losses in terms of training time because they consist of billions of parameters. This project aims to use a layer-based random update approach to accelerate and parallelize the pretraining process of LLM.[1]

There are many studies that are being conducted to accelerate LLM pre-training. One of them by Salesforce introduces Parallel Alignment Fine-Tuning (PAFT), a method designed to improve both task performance and alignment with human preferences. In PAFT, the model undergoes Supervised Fine-Tuning (SFT) while simultaneously performing Preference Alignment (PA), allowing both processes to run in parallel. This approach speeds up training and enhances the model's ability to generate outputs that better match user expectations.

II. System Design

Many models are trained on Google Colab by closing their layers. The trained layers are merged in the final step. The success rate of the models are measured in every step

.

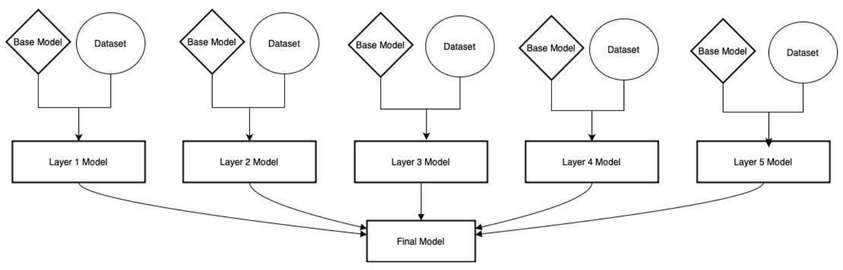

Figure 1. System Design

A. Material and Dataset

Models to be used selected as Llama-3.2-1B and Llama-3.2-1B-Instruct. The performance of the models before training was measured using Evaluation Harness method. Dataset is saved on Google Drive. The Cosmopedia dataset contains approximately 30 million files and 25 billion tokens. Various parts of the cosmopedia dataset was used to train models. GSM8K (Grade School Math 8K) is a dataset containing high quality mathematic questions for primary school. It contains more than 8500 questions.

B. Method

The model training process involved tokenizing data, feeding it in batches, calculating loss, applying backpropagation, and saving the final model—all implemented in PyTorch. To improve efficiency, layer freezing was used to keep specific layers unchanged, preserving pre-learned knowledge. Model merging techniques, such as linear merging, SLERP, and the customizable TIES method, were employed to combine strengths of different pre-trained models without full retraining. MergeKit was used to manage this process through easy configuration. For evaluation, the Evaluation Harness framework was used.

III. Experimental Results

The effect of freezing selected layers during training was evaluated using four benchmarks: ARC Challenge, Hellaswag, GSM8K, and Validation Loss. Cosmopedia models were tested with ARC, Hellaswag, and validation loss, using both pre-trained and randomly initialized weights. Math models were evaluated with GSM8K, trained on either the GSM8K Train or MetaMathQA dataset. These evaluations helped compare the impact of initialization methods and data choices. Results showed that applying dataset splitting and layer freezing strategies to the Llama-3.2-1B model was effective, with Hellaswag's Normalized Accuracy and Validation Loss emerging as key indicators.

Table 1. SLERP Strategies

SLERP	Part 1	Part 2	Part 3	Part 4
1	0-4	4-8	8-12	12-16
2	0-8	0-8	8-16	8-16
3	0-6	4-9	7-13	11-16
4	0-16	0-16	0-16	0-16
5	0-4	0-8	8-16	12-16
6	0-16	4-16	8-16	12-16
7	0-12	6-12	10-16	4-16
8	0-4	4-16	4-16	4-16
9	0-12	12-16	12-16	12-16

The merging process combines separately trained parts by sharing common layers equally and assigning 80% weight to non-common layers from each part. First, Part 1 and 2 are merged this way, then Part 3 and 4, and finally the two merged halves. This enables efficient 4-way parallel training with reduced time and minimal loss.

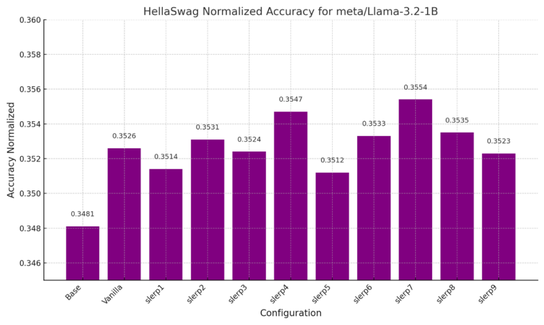

Figure 2. Hellaswag Normalized Accuracy

Strategy	Small Val. Loss	Large Val. Loss	Small Val. Loss Preservation (%)	Large Val. Loss Preservation (%)	Duration (min)	% of Vanilla Duration
SLERP 1	2.3488	4.5743	%47.8	%54.4	37.5	%19.1
SLERP 2	2.3308	4.2158	%49.5	%58.0	42	%21.4
SLERP 3	2.3444	4.4244	%48.2	%55.9	39	%19.9
SLERP 4	2.3399	3.0166	%48.7	%70.9	49.5	%25.3
SLERP 5	2.3356	4.4937	%49.1	%55.2	42	%21.4
SLERP 6	2.3191	3.7319	%50.6	%62.0	49.5	%25.3
SLERP 7	2.3304	3.9760	%49.5	%59.6	45	%23.0
SLERP 8	2.3227	4.0424	%50.3	%59.0	45	%23.0
SLERP 9	2.3208	4.0244	%50.5	%59.2	45	%23.0

Figure 3. SLERP Validation Comparison

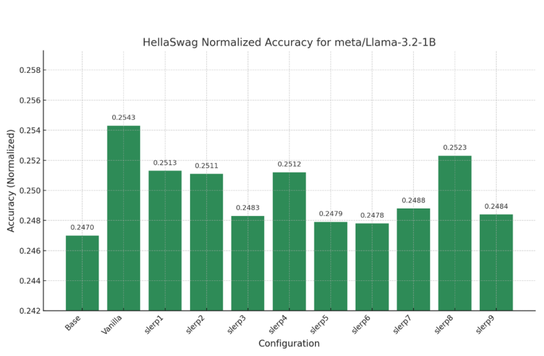

Figure 4. Hellaswag N. Accuracy R.W

Strategy	Small Val. Loss	Large Val. Loss	Small Val. Loss Preservation (%)	Large Val. Loss Preservation (%)	Duration (min)	% of Vanilla Duration
SLERP 1	11.5028	11.6709	%8.0	%6.1	9	%20.5
SLERP 2	10.9892	12.0197	%14.3	%1.8	9.75	%22.2
SLERP 3	12.0424	10.9267	%1.3	%15.8	9.5	%21.6
SLERP 4	10.5168	9.6887	%20.1	%31.8	11	%25.0
SLERP 5	11.7482	10.8727	%4.9	%16.5	9.75	%22.2
SLERP 6	10.5283	12.0553	%19.9	%1.3	11	%25.0
SLERP 7	10.7987	11.1349	%16.7	%12.3	10.5	%23.9
SLERP 8	11.5539	11.4030	%7.4	%9.4	10.5	%23.9
SLERP 9	11.3099	12.2905	%10.4	%-2.1	10.5	%23.9

Figure 5. SLERP Validation Comparison R.W

Strategy	Flexible Exact Match	Strict Exact Match	Flexible Score Increase Preservation (%)	Strict Score Increase Preservation (%)	Duration (min)	% of Vanilla Duration
SLERP 1	0.0963	0.0948	%28.5	%27.4	1	%20.0
SLERP 2	0.0970	0.0970	%29.1	%29.4	1.25	%25.0
SLERP 3	0.1046	0.1046	%36.2	%36.5	1.17	%23.3
SLERP 4	0.1304	0.1312	%60.6	%61.4	1.42	%28.3
SLERP 5	0.0986	0.0986	%30.6	%30.8	1.25	%25.0
SLERP 6	0.0955	0.1001	%27.8	%31.8	1.42	%28.3
SLERP 7	0.1031	0.1024	%34.8	%34.6	1.33	%26.7
SLERP 8	0.1160	0.1168	%46.9	%47.7	1.33	%26.7
SLERP 9	0.0743	0.0773	%7.8	%12.4	1.33	%26.7

Figure 6. GSM8K Comparison

Strategy	Flexible Exact Match	Strict Exact Match	Flexible Score Increase Preservation (%)	Strict Score Increase Preservation (%)	Duration (min)	% of Vanilla Duration
SLERP 1	0.1099	0.1372	%18.5	%30.5	56	%18.6
SLERP 2	0.1804	0.2146	%48.1	%62.0	62	%20.6
SLERP 3	0.1562	0.1713	%37.5	%45.0	59	%19.6
SLERP 4	0.2578	0.2790	%80.6	%89.7	74.5	%24.7
SLERP 5	0.1107	0.1486	%18.8	%35.1	62	%20.6
SLERP 6	0.1660	0.1948	%41.7	%53.7	74.5	%24.7
SLERP 7	0.2092	0.2199	%60.0	%64.0	68	%22.6
SLERP 8	0.2055	0.2168	%58.4	%62.7	68	%22.6
SLERP 9	0.1145	0.1213	%20.5	%23.9	68	%22.6

Figure 7. Metamath Comparison

To test the effectiveness of the parallel training strategy, the training time table must be analyzed. X means 400 megabyte dataset.

Table 2. Cosmopedia Time Table

Dataset	Scale	Layers Trained	Duration (min)
cosmopedia	X	16	196
cosmopedia	2.5X	16	500
cosmopedia	X/4	4	37.5
cosmopedia	X/4	6	39
cosmopedia	X/4	8	42
cosmopedia	X/4	12	45
cosmopedia	X/4	16	49.5

Table 3. Random W. Cosmopedia Time Table

Dataset	Scale	Layers Trained	Duration (min)
cosmopedia	X	16	44.00
cosmopedia	2.5X	16	112.00
cosmopedia	X/4	4	9.00
cosmopedia	X/4	6	9.50
cosmopedia	X/4	8	9.75
cosmopedia	X/4	12	10.50
cosmopedia	X/4	16	11.00

Table 4. GSM8K Time Table

Dataset	Scale	Layers Trained	Duration (min)
gsm8k	X	16	5.00
gsm8k	X/4	4	1.00
gsm8k	X/4	6	1.10
gsm8k	X/4	8	1.15
gsm8k	X/4	12	1.20
gsm8k	X/4	16	1.25

Table 5. Metamath Time Table

Dataset	Scale	Layers Trained	Duration (H:MM:SS)
metamath	X	16	5:01:30
metamath	X/4	4	56:00
metamath	X/4	6	59:00
metamath	X/4	8	1:02:00
metamath	X/4	12	1:08:00
metamath	X/4	16	1:14:30

Slerp-4 was selected for comparison as the most successful strategy based on evaluation metrics. While vanilla training progresses step by step, Slerp-4 completes its training in fewer, parallel steps. At the point when vanilla reaches its first step, Slerp-4 has already reached its midpoint (Slerp-mid-4).

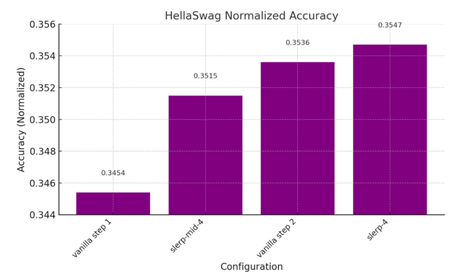

Figure 8. SLERP-4 vs Vanilla Hellaswag N.A

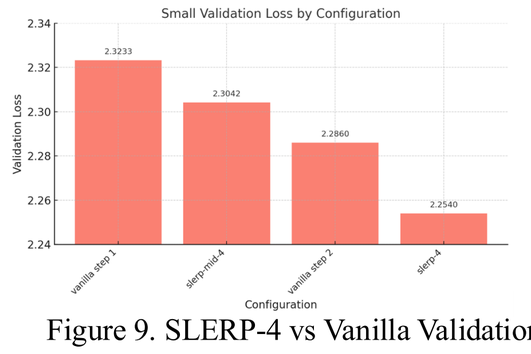

Figure 9. SLERP-4 vs Vanilla Validation

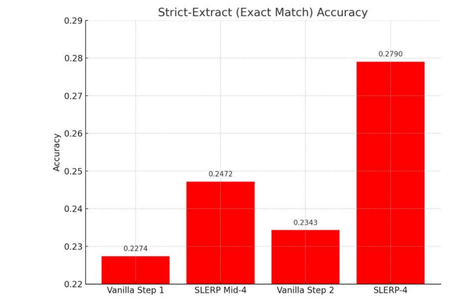

Figure 10. SLERP-4 vs Vanilla Strict-Extract

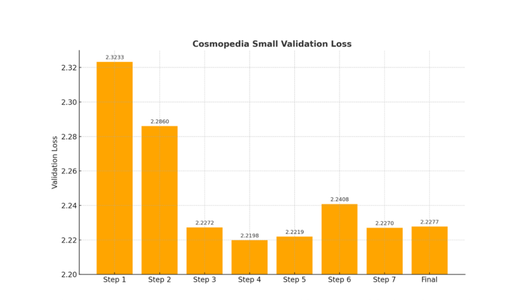

Figure 11. Cosmopedia Validation by Steps

Conclusion

In conclusion, the project demonstrated that parallel training strategies—particularly those using SLERP methods—can significantly reduce training time while maintaining or even improving model performance. SLERP-4 proved especially effective across various tasks and datasets, highlighting its potential for efficient and scalable fine-tuning. However, the strategy's dependence on pre-trained weights suggests limitations when applied to randomly initialized models. Overall, the findings support SLERP-based approaches as a promising direction for accelerating large language model adaptation without compromising accuracy.

References

[1] Raiaan M. *et al.* (2024), "A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges," in *IEEE Access*, vol. 12, pp. 26839-26874.