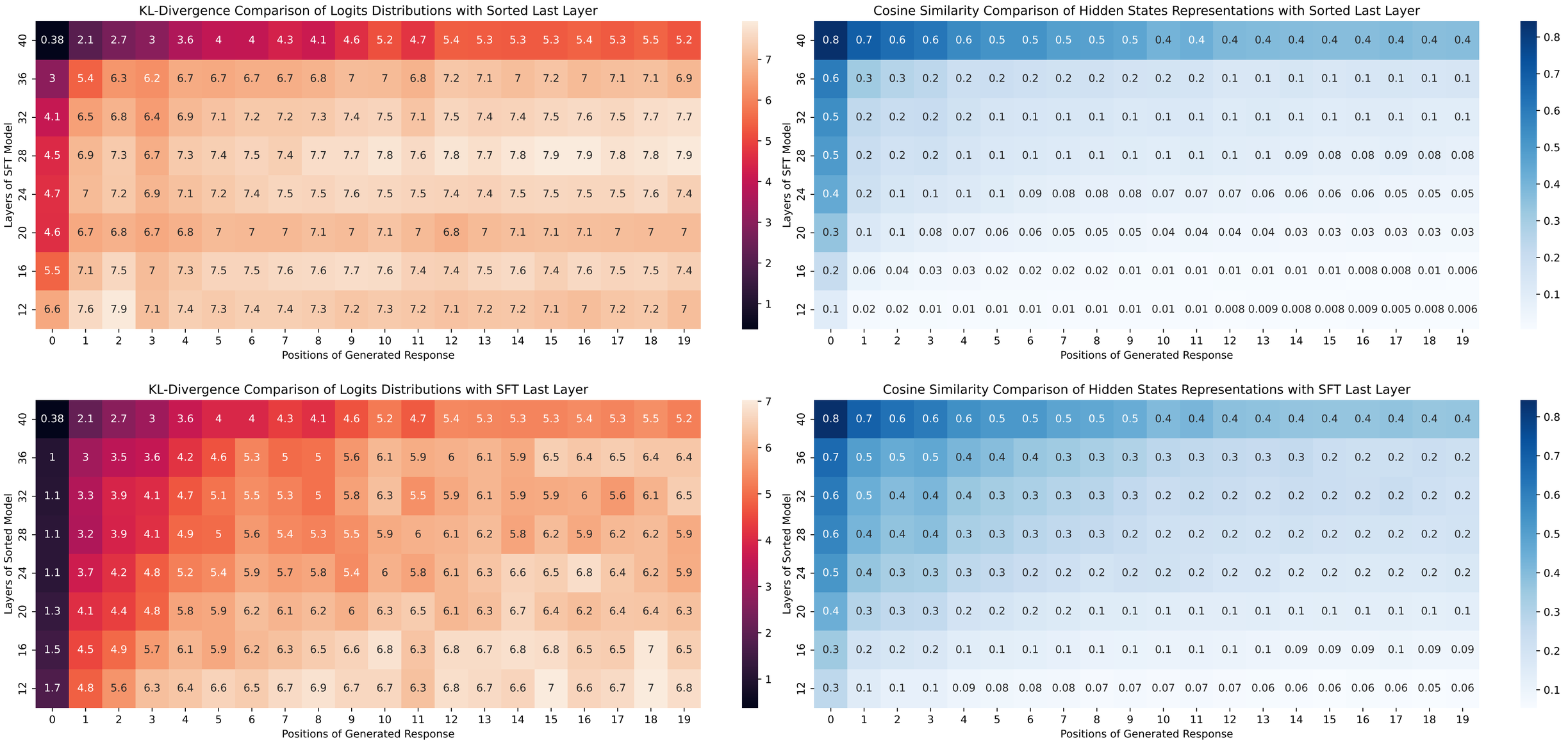
Sorted LLaMA: Unlocking the Potential of Intermediate Layers of Large Language Models for Dynamic Inference Using Sorted Fine-Tuning (SoFT)

Parsa Kavehzadeh [1] Mojtaba Valipour [12] Marzieh Tahaei [1] Ali Ghodsi [2] Boxing Chen [1] Mehdi Rezagholizadeh [1]

[1]Huawei Noah's Ark Lab [2]University of Waterloo

Sorted Fine-Tuning vs. Supervised Fine-Tuning



SortedLLaMA Last Layer vs. LLaMA sub-models (top) and SortedLLaMA sub-models vs. LLaMA Last Layer (bottom). A sub-model comparison based on output logits and hidden state cosine similarity. The numbers are average of all 170 samples in the PandaLM validation set.

Introduction

We extend SortedNet to generative NLP tasks, making large language models dynamic without any Pre-Training and by only replacing standard Supervised Fine-Tuning (SFT) with Sorted Fine-Tuning (SoFT). Our approach boosts model efficiency, eliminating the need for multiple models for various scenarios during inference.

We show that this approach can unlock the potential of intermediate layers of transformers in generating the target output. Our sub-models remain integral components of the original model, minimizing storage requirements and transition costs between different computational/latency budgets. In this paper, we seek to answer the following questions through systematic evaluation:

- Do the intermediate layers resulting from Supervised Fine-Tuning of a large language model generate accurate and meaningful outputs?
- Does Supervised Fine-Tuning exhibit a sorted behavior, meaning that later layers produce more accurate and meaningful results than earlier layers? If so, to what extent?
- How can we enhance this sorted behavior with minimal cost?

To answer these questions, we employ LLaMA 2 13B and perform both standard Supervised Fine-Tuning (SFT) and Sorted Fine-Tuning (SoFT) on the Stanford Alpaca, GSM8K and TriviaQA datasets. For Sorted Fine-Tuning, we target 8 sub-models and share the LLM head among them to ensure cost parity. We utilize the PandaLM benchmark to assess the performance of the sub-models on Alpaca dataset.

Contributions

Our findings demonstrate the superior performance of SoFT in comparison to SFT and even to memory-demanding methods like Early Exit.

The contributions of our paper can be summarized as follows:

- **Sorted Fine-Tuning**: Applying SoFT method for tuning auto-regressive language models for generative tasks by sharing a single LLM head layer among sub-models.
- **Capable Sub-models**: Generating 8 nested sub-models, ranging from 12 to 40 layers, from LLaMA2 13B by applying Sorted Fine-Tuning on the Stanford Alpaca dataset, GSM8K and TriviaQA benchmarks and at a cost equivalent to Standard Fine-Tuning.
- **Evaluation:** Evaluating the performance of the sub-models of a LLaMA2 and demonstrating the effectiveness of SoFT in enhancing the ability of intermediate layers for text generation and reasoning through extensive evaluation.

Experiments

Sorted LLaMA/LLaMA	12 (4.1B)	16 (5.4B)	20 (6.6B)	24 (7.9B)	28 (9.2B)	32 (10.4B)	36 (11.7B)	40 (13B)
Zero-Shot SoFT vs. Zero-Shot SFT								
12 (4.1B)	71.0/99.0/0.0	97.5/72.5/0.0	129.0/41.0/0.0	131.0/39.0/0.0	121.5/48.5/0.0	106.5/63.5/0.0	45.0/125.0/0.0	17.0/152.5/0.5
16 (5.4B)	81.0/89.0/0.0	101.5/68.5/0.0	128.5/40.5/1.0	131.5/38.0/0.5	124.0/44.5/1.5	114.0/54.0/2.0	52.0/114.0/4.0	18.0/146.0/6.0
20 (6.6B)	111.5/58.5/0.0	132.0/38.0/0.0	144.5/23.5/2.0	147.5/20.5/2.0	141.5/24.0/4.5	132.5/30.5/7.0	73.5/85.5/11.0	32.5/114.0/23.5
24 (7.9B)	124.5/45.5/0.0	136.5/33.5/0.0	150.0/18.0/2.0	154.5/13.5/2.0	148.0/18.5/3.5	144.5/19.0/6.5	98.0/62.0/10.0	44.5/90.0/35.5
28 (9.2B)	125.5/44.5/0.0	145.0/25.0/0.0	153.0/15.0/2.0	153.5/14.5/2.0	148.0/16.5/5.5	143.5/20.5/6.0	96.5/59.5/14.0	45.0/89.0/36.0
32 (10.4B)	141.5/28.5/0.0	152.0/18.0/0.0	159.0/9.0/2.0	160.0/8.0/2.0	152.0/12.5/5.5	150.5/13.5/6.0	108.5/45.0/16.5	55.5/75.0/39.5
36 (11.7B)	141.0/28.5/0.5	152.5/17.0/0.5	159.0/8.5/2.5	161.5/6.5/2.0	150.0/14.5/5.5	148.5/15.5/6.0	112.0/42.5/15.5	53.0/66.0/51.0
40 (13B)	143.5/26.5/0.0	156.0/14.0/0.0	160.5/7.5/2.0	161.0/7.0/2.0	150.0/14.0/6.0	150.0/13.5/6.5	115.5/39.0/15.5	52.5/62.5/55.0
SoFT vs. SFT+ICT(Early-Exit)								
12 (4.1B)	75.0/95.0/0.0	108.5/61.5/0.0	128.5/41.5/0.0	122.5/47.5/0.0	116.5/53.5/0.0	91.0/79.0/0.0	37.5/131.5/1.0	17.0/152.5/0.5
16 (5.4B)	86.5/82.5/1.0	113.0/57.0/0.0	127.0/41.0/2.0	122.0/47.0/1.0	117.5/50.5/2.0	94.5/72.0/3.5	36.0/129.0/5.0	18.0/146.0/6.0
20 (6.6B)	111.5/57.5/1.0	137.0/33.0/0.0	143.5/24.0/2.5	143.0/23.0/4.0	137.0/27.0/6.0	122.0/38.0/10.0	60.0/94.5/15.5	32.5/114.0/23.5
24 (7.9B)	126.5/42.5/1.0	144.0/26.0/0.0	149.0/19.5/1.5	151.0/15.5/3.5	143.0/21.5/5.5	133.5/28.0/8.5	76.5/72.5/21.0	44.5/90.0/35.5
28 (9.2B)	130.0/39.0/1.0	147.0/23.0/0.0	153.5/15.5/1.0	150.0/16.0/4.0	143.5/18.5/8.0	131.0/29.0/10.0	79.0/66.0/25.0	45.0/89.0/36.0
32 (10.4B)	141.5/27.5/1.0	155.5/14.5/0.0	161.0/8.0/1.0	157.0/8.5/4.5	151.0/11.0/8.0	143.5/15.0/11.5	89.5/49.5/31.0	55.5/75.0/39.5
36 (11.7B)	143.0/25.5/1.5	156.5/13.0/0.5	160.0/8.5/1.5	157.0/8.5/4.5	148.0/14.0/8.0	142.5/16.5/11.0	92.5/46.5/31.0	53.0/66.0/51.0
40 (13B)	146.0/23.0/1.0	157.0/13.0/0.0	160.5/7.5/2.0	157.5/9.0/3.5	149.0/14.0/7.0	143.5/16.0/10.5	97.5/43.5/29.0	52.5/62.5/55.0

Pair-wise comparison for different layers (sub-models) in regular fine-tuning and sorted paradigms at equal training cost (2 Epochs). Each cell consists of three values: Wins, Losses, Ties.

	PandaLM			TriviaQA			GSM8K		
Auto-regressive Decoding									
Model	Time per Token (ms)	Score	Rejection Ratio	Time per Token (ms)	Accuracy	Rejection Ratio	Time per Token (ms)	Accuracy	Rejection Ratio
Layer 40 (full)	94.07	-	-	91.27	37.95	-	93.60	33.05	-
Speculative Decoding									
Draft Model	Time per Token (ms)	Score	Rejection Ratio	Time per Token (ms)	Accuracy	Rejection Ratio	Time per Token (ms)	Accuracy	Rejection Ratio
Layer 12	80.86 (1.16×)	-0.144	0.37	110.50 (0.82×)	34.36	0.72	66.10 (1.41×)	32.22	0.43
Layer 16	84.10 (1.11×)	-0.211	0.31	118.92 (0.76×)	34.16	0.70	67.51 (1.38×)	33.20	0.32
Layer 20	84.50 (1.11×)	-0.144	0.26	139.78 (0.65×)	34.19	0.66	68.45 (1.36×)	33.73	0.23
Instance-Aware Dynamic Inference									
Model	Time per Token (ms)	Score	Rejection Ratio	Time per Token (ms)	Accuracy	Rejection Ratio	Time per Token (ms)	Accuracy	Rejection Ratio
Layer 12:40	69.91 (1.34×)	-0.050	-	81.01 (1.12×)	36.53	-	51.30 (1.82×)	30.62	-

Speed-up in inference time on three PandaLM, TriviaQA, and GSM8K benchmarks by utilizing Speculative Decoding and Instance-Aware Dynamic Inference techniques.

Conclusion

We present Sorted LLaMA, a many-in-one language model for **dynamic inference** obtained using SoFT instead of SFT. Sorted LLaMA unlocks the potential capability of intermediate layers, offering **dynamic adaptation without pre-training or additional expenses related to model compression.** Our approach makes the deployment of generative LLMs far more **efficient** as all sub-models remain integral components of the original model, without any burden of storage requirements, minimizing transition costs between different computational demands.

References

[1] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[2] 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. *arXiv preprint arXiv:2106.09685*, 2021.

[3] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[4] Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, et al.

parsa.kavehzadeh@huawei.com