Enhancing In-Context Learning Performance with just SVD-Based Weight Pruning: A Theoretical Perspective

Xinhao Yao

Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
Renmin University of China, Beijing, China
yaoxinhao021978@ruc.edu.cn

Nov, 2024

Outline

Outline

Introduction

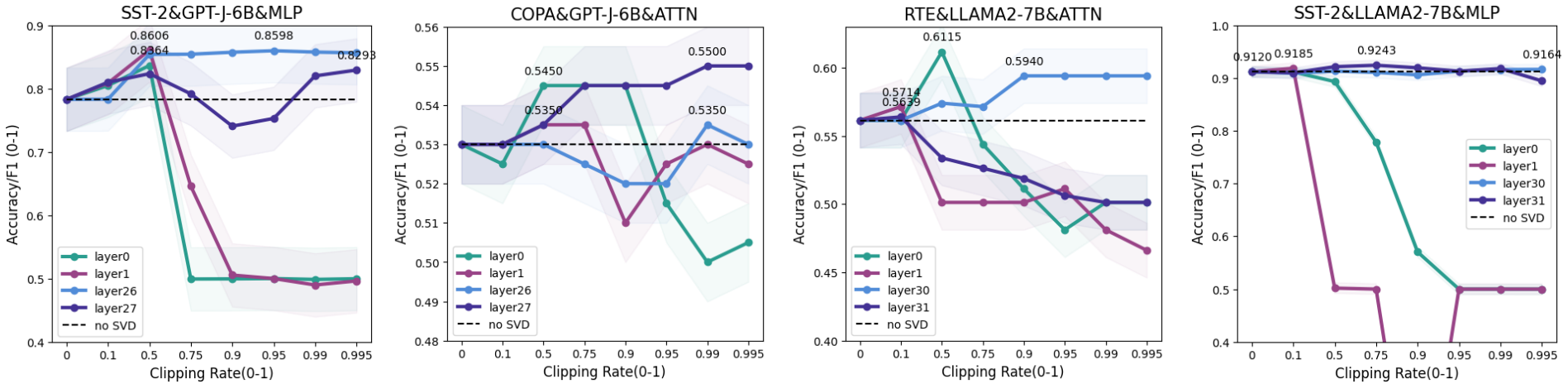


Figure 1.1: The effect of weight pruning across different layer types.

In this paper, we first show an exciting phenomenon[1] that:

- SVD-based weight pruning can enhance ICL performance.
- And more surprising, pruning weights in deep layers often results in more stable performance improvements than in shallow layers.

[1] A similar case in [Sharma, ICLR 2024] notes that a large portion of singular values can be removed from linear layers in large language models without affecting or even improving reasoning performance.

Contribution

However, the underlying mechanism of this phenomenon still remains a mystery, and this paper seeks to explore the issue from the following two aspects.

- Why this phenomenon?
  - We first analyze the ICL form with Transformer from the perspective of ICL as implicit gradient descent fine-tuning, and we present the implicit gradient descent trajectories of ICL in **Theorem 1**.
  - Afterwards, we use the information-theoretic approaches to give the generalization bounds of ICL via full implicit GD trajectories in **Theorem 2**, explaining (Q1) why SVD-based weight pruning can enhance ICL performance? and (Q2) why do deep and shallow layers exhibit different behaviors when their weight matrices are drastically reduced?

- How to better use this phenomenon (Q3)?

  Based on all our experimental and theoretical insights, one can design new ICL algorithms. We intuitively propose a derivative-free and model-compression algorithm to illustrate how theoretical analysis can guide experimental procedures effectively.

Outline

The Implicit Gradient Descent of ICL

Previous works[2] describe how ICL with attention can be connected to a meta-optimizer which produces implicit gradient. The following lemma demonstrates the result.

Lemma 2.1 (The Implicit Gradient Descent of ICL in a Single Linear Attention Layer)

Consider a Transformer consists of a single linear layer attention with residual connection, parameterized by $\mathbf{W}_V, \mathbf{W}_K, \mathbf{W}_Q$. *Let* $\mathbf{H} = [\mathbf{H}_s, \mathbf{h}_{N+1}]$ *be the input, where* $\mathbf{H}_s = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_N]$ *represents the demonstration sample and* $\mathbf{h}_{N+1}$ *is the test data. And let* $\hat{\mathbf{h}}_{N+1}$ *be the single layer output. Then, it holds that*

$$\Delta \mathbf{W}_{icl} = \mathbf{W}_V \mathbf{H}_s (\mathbf{W}_K \mathbf{H}_s)^T \mathbf{W}_Q = \left(\sum_{i=1}^{N} \mathbf{W}_V \mathbf{h}_i \otimes \mathbf{W}_K \mathbf{h}_i \right) \mathbf{W}_Q,$$

$$\hat{\mathbf{h}}_{N+1} = \mathbf{h}_{N+1} + \Delta \mathbf{W}_{icl} \mathbf{h}_{N+1},$$

where $\mathbf{W}_V \mathbf{H}_s$ *is regarded as the meta-gradient of ICL, which is used to generate the implicit gradient matrix* $\Delta \mathbf{W}_{icl}$ *to act on the final feature output.*

[2] [Irie, ICML 2022],[Dai, ACL Findings 2023],[Ren, NeurIPS 2024]

Theorem 1: Trajectories

Theorem 2.1 (The Implicit Gradient Descent Trajectories of ICL)

Consider a Transformer as a stack of L linear attention blocks with residual connection, parameterized by $[\mathbf{W}_V^t]_1^L$,$[\mathbf{W}_K^t]_1^L$,$[\mathbf{W}_Q^t]_1^L$. Denote $[\mathbf{h}_i^t]_1^{N+1}$ as the output of the t-th layer, $[\mathbf{h}_i^0]_1^{N+1}$ as the initial input. Then for $t \in [L]$, it holds that

$$\mathbf{G}_t = \Delta\mathbf{W}_t(1 + \mathbf{W}_{t-1}) = \Delta\mathbf{W}_t(1 + \mathbf{W}_0 + \sum_{j=1}^{t-1} \mathbf{G}_j),$$

$$\mathbf{h}_{N+1}^t = \mathbf{h}_{N+1}^0 + \sum_{j=1}^{t} \mathbf{G}_t \mathbf{h}_{N+1}^0 = \mathbf{h}_{N+1}^0 + \mathbf{W}_t \mathbf{h}_{N+1}^0,$$

where $\Delta\mathbf{W}_t \triangleq \left(\sum_{i=1}^N \mathbf{W}_V^t \mathbf{h}_i^{t-1} \otimes \mathbf{W}_K^t \mathbf{h}_i^{t-1}\right) \mathbf{W}_Q^t$, $\mathbf{W}_0 = 0, \mathbf{W}_t = \mathbf{W}_{t-1} + \mathbf{G}_t$ *and* $\mathbf{G_1} = \Delta\mathbf{W}_1$.

Remark 1

Note that the exclusion of Transformer weight $([\mathbf{W}_K^t]_1^L, [\mathbf{W}_Q^t]_1^L, [\mathbf{W}_V^t]_1^L)$ implies that $\mathbf{G}_t$ is only dependent on $\mathbf{W}_{t-1}$ and $\mathbf{H}_s$, this is consistent with gradient descent in terms of relevance.

Noise Discussion of the Implicit Gradient

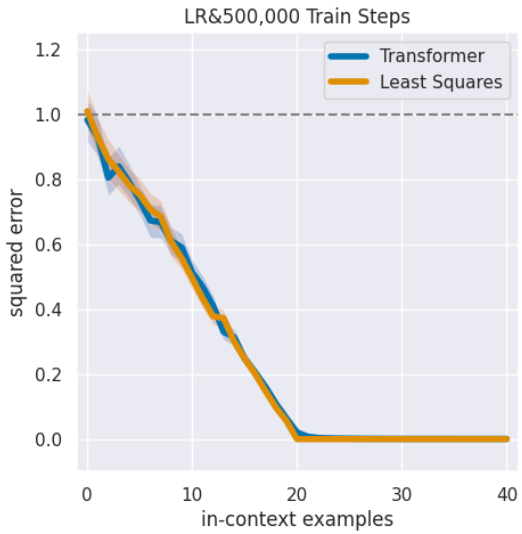


Figure 2.1: Trained Transformer performs comparably to the optimal least squares estimator.

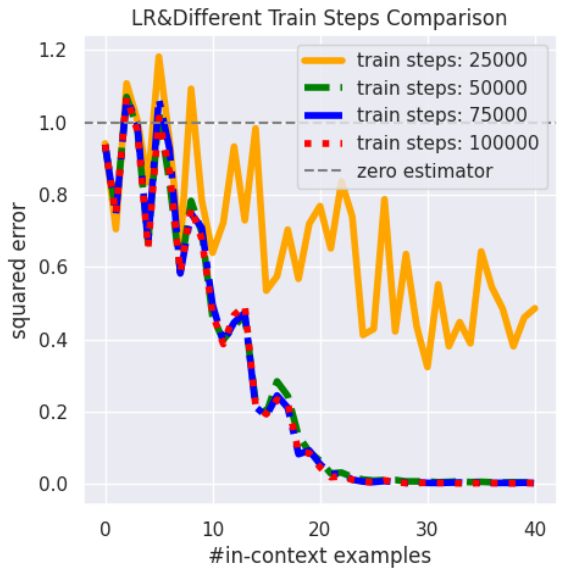


Figure 2.2: The effect of different train steps on Trained Transformer.

(Left) Consider[3] the class of linear functions $\mathcal{F} = \{f | f(x) = w^T x, w \in \mathbb{R}^d\}$, in d dimensions where $d = 20$. They sample $x_1, \ldots, x_k, x_{\text{query}}$, and w independently from the isotropic Gaussian distribution $N(0, I_d)$. They then compute each $y_i = w^T x_i$ and construct the prompt as $P = (x_1, y_1, x_2, y_2, \ldots, x_k, y_k, x_{\text{query}})$. (Right) We further compare the Trained Transformer with different train steps.

[3] [Garg, NeurIPS 2022]

Expected generalization error

- Expected generalization error = population risk (L_μ) - empirical risk ($L_{\mathbf{H}_s}$).
- One can evaluate the empirical risk ($L_{\mathbf{H}_s}$) by assessing the model's performance on the validation set. If the generalization error is known, it is possible to estimate the population risk (L_μ). Therefore, the most challenging aspect is addressing the generalization error.

Theorem 1 indicates that the initial parameter $W_0 = 0$, which is independent of all other random variables. And an L-layer Transformer does implicit GD of ICL stops after L updates, outputting W_L as the implicit learned parameter. Our main results are mutual information based expected generalization error bounds of ICL (**Theorem 2**).

Theorem 2: Bounds

Theorem 2.2 (The Generalization Bounds of ICL via Full Implicit Gradient Descent Trajectories)

Under the conditions of **Theorem 1** *and* **Assumption 1**, *assume the implicit gradient noise covariance C_t is a positive-define matrix, the loss $\ell(w, \mathbf{h})$ is R-subGaussian for any $w \in \mathcal{W} \in \mathbb{R}^d$, then*

$$\widetilde{error} \leq \sqrt{\frac{R^2}{N} \sum_{t=1}^{L} \mathbb{E}^{\mathbf{H}_s}_{\mathbf{W}_{t-1}} \left[d \log \left(\frac{\|\Delta \mathbf{W}_t\|_F^2 \cdot \left\| 1 + \sum_{j=1}^{t-1} \mathbf{G}_j \right\|_F^2 + \operatorname{tr}\{C_t\}}{d} \right) - \operatorname{tr}\{\log C_t\} \right]},$$

where $\mathbf{vec}(\mathbf{G}_t) \in \mathbb{R}^d$, $\Delta \mathbf{W}_t \triangleq \left(\sum_{i=1}^{N} \mathbf{W}_V^t \mathbf{h}_i^{t-1} \otimes \mathbf{W}_K^t \mathbf{h}_i^{t-1} \right) \mathbf{W}_Q^t$ *and* $\mathbf{G}_t = \Delta \mathbf{W}_t (1 + \mathbf{W}_0 + \sum_{j=1}^{t-1} \mathbf{G}_j) = \Delta \mathbf{W}_t (1 + \mathbf{W}_{t-1})$. $\operatorname{tr}\{\cdot\}$ *denotes the trace of a matrix,* $\|\cdot\|_F$ *denotes the Frobenius norm of a matrix and* $\mathbb{E}_Y^X$ *is the conditional expectation.*

Answers to Q1/Q2

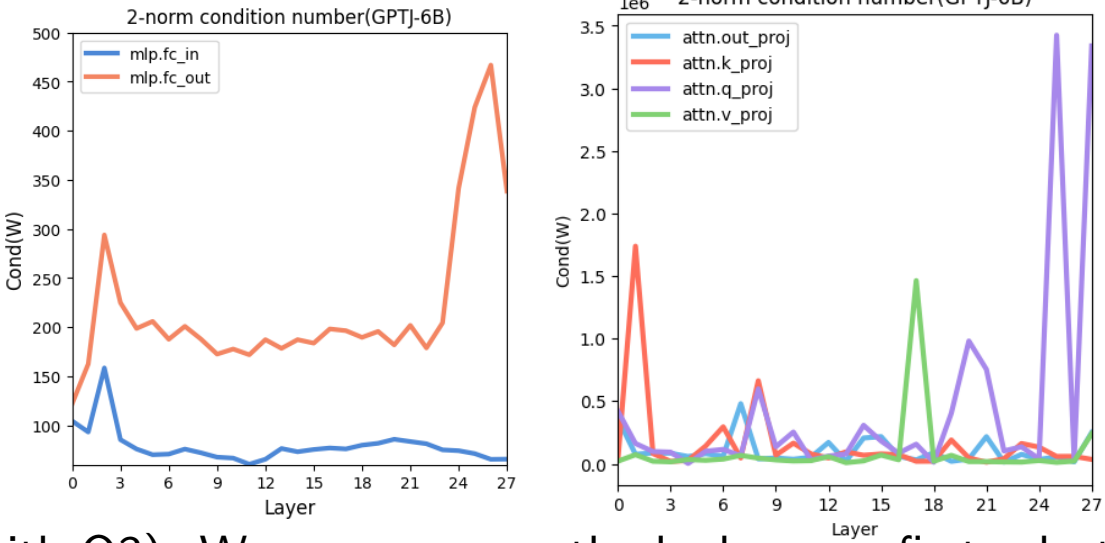
Remark 2 (Deal with Q1: why SVD-based weight pruning can enhance ICL performance?)

Theorem 2 indicates that one can control the generalization performance of ICL via controlling the implicit gradient norm along the entire ICL implicit GD trajectories ($[\mathbf{G}_t]_1^L$ or $[\Delta\mathbf{W}_t]_1^L$). Note that controlling implicit gradient norm can also control the magnitude of the trace of implicit gradient noise covariance. This elucidates why weight pruning through SVD, even if it only alters a single weight matrix, can confer advantages on the performance of Transformers in ICL inference.

Remark 3 (Deal with Q2: why do deep and shallow layers exhibit different behaviors when their weight matrices are drastically reduced?)

It is notable that $\mathbf{G}_t$ is highly correlated with the sequence $(\Delta\mathbf{W}_1, ..., \Delta\mathbf{W}_t)$. More precisely, adjusting the weight matrix of the k-th layer, will invariably impact the norm of $[\Delta\mathbf{W}_t]_{t>k}^L$, further influence $[\mathbf{G}_t]_{t>k}^L$. The deeper the layer of the adjusted parameters is, the fewer the number of $\mathbf{G}_t$ affected. $(L-K+1)$

Algorithm: answer to Q3



Algorithm (Deal with Q3). We propose a method where we first select layers with the top-k largest condition numbers and then identify the layer with the largest number among these. We perform a search for the optimal clipping rate on the validation set and subsequently evaluate it on the test set.

Experimental results

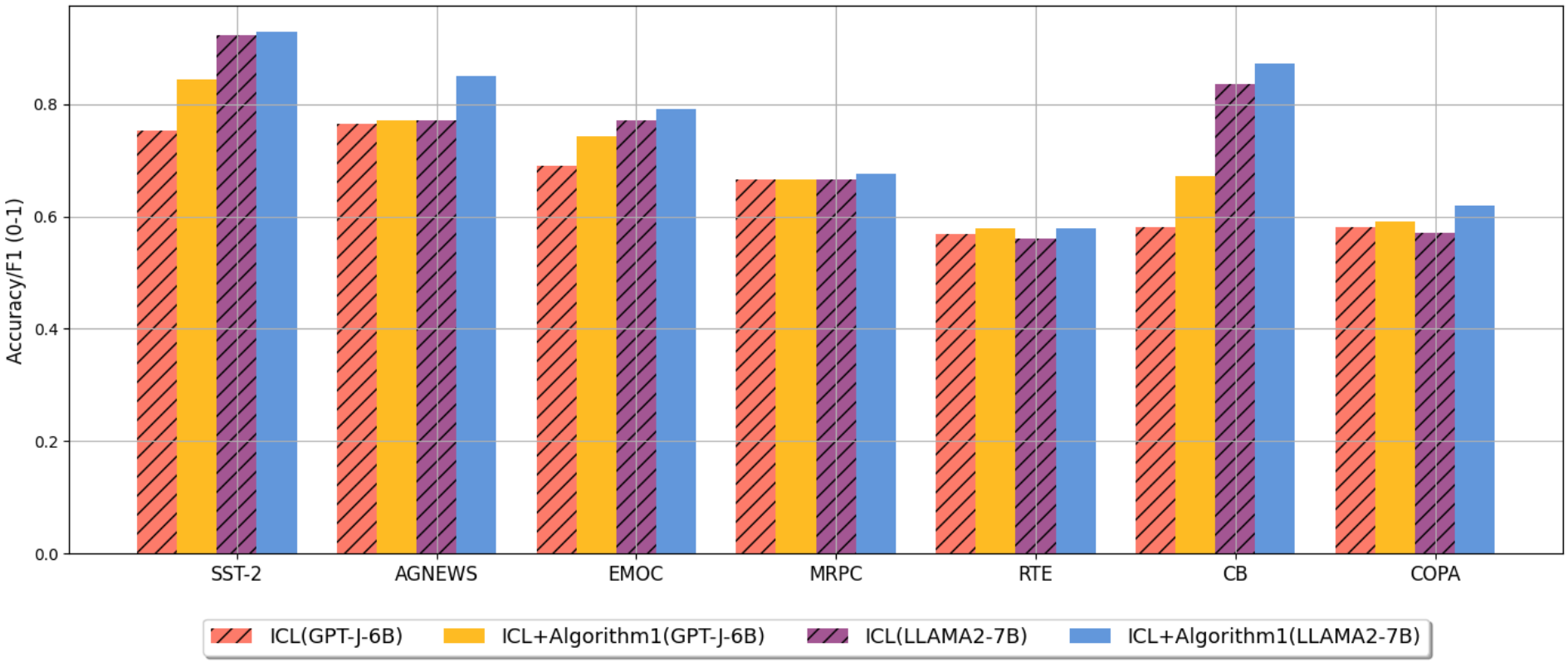


Figure 2.3: The Model Performance on Test set by different tasks. The results are obtained by comparing four scenarios: ICL (GPT-J-6B), ICL+Algorithm1 (GPT-J-6B), ICL (LLAMA2-7B) and ICL+Algorithm1 (LLAMA2-7B). ICL+Algorithm1 demonstrates superior results over only ICL on different tasks.

Outline

Conclusion

- Our primary objective is to establish a general theoretical framework that uncovers the underlying mechanism behind the phenomenon that SVD-based weight pruning can enhance ICL performance. Based on our theoretical insights, one can design new ICL algorithms.
- However, further studies are required on (i) how to extend our generalization theory to a more standard Transformer architecture, (ii) do the results about ICL hold true for tasks beyond natural language processing and (iii) how to minimize the cost of searching the optimal clipping rate. Those will deepen our understanding of the ICL capabilities.

Part of References


Pratyusha Sharma, Jordan T Ash, and Dipendra Misra. The truth is in there: Improving reasoning in language models with layer-selective rank reduction. In International Conference on Learning Representations, 2024.


Kazuki Irie, Róbert Csordás, and Jürgen Schmidhuber. The dual form of neural networks revisited: Connecting test time predictions to training patterns via spotlights of attention. In International Conference on Machine Learning, pages 9639–9659, 2022.


Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In Findings of the Association for Computational Linguistics, pages 4005–4019, 2023.


Ruifeng Ren and Yong Liu. Towards Understanding How Transformers Learn In-context Through a Representation Learning Lens. In Advances in Neural Information Processing Systems (38th edition), 2024.


Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In Advances in Neural Information Processing Systems (36th edition), pages 30583–30598, 2022.

Thank you!