NUCLEAR-NORM MAXIMIZATION FOR LOW-RANK UPDATES

Huanxi Liu[1,2†], *Yuanzhao Zhai*[1,2†], *Kele Xu*[1,2], *Dawei Feng*[1,2], *Yiying Li*[3⋆]

[1]National University of Defense Technology, Changsha, China [2]State Key Laboratory of Complex Critical Software Environment [3]Artificial Intelligence Research Center, DII, Beijing, China

ABSTRACT

Pre-trained large language models exhibit significant potential in speech and language processing. Fine-tuning all parameters becomes impractical when confronted with numerous downstream tasks. To address this challenge, various low-rank adaptation techniques have been introduced for parameter-efficient fine-tuning, which freeze the over-parametrized models and learn incremental parameter updates within smaller subspaces. However, our observation reveals that most directions of the learned subspace play a minor role in the incremental updates. Consequently, fine-tuned models may not achieve optimal performance. To bridge this gap, we introduce NNM-LoRA, which strives to harness more meaningful singular directions. Through Nuclear Norm Maximization (NNM), we can better regulate the allocation of singular values. Accordingly, we propose a parameter-free plug-and-play regularizer for low-rank updates. This innovative approach allows us to utilize as many singular directions of the subspace as possible during the training of low-rank updates. To validate the effectiveness of NNM-LoRA, we conduct extensive experiments involving different pre-trained models on various natural language understanding tasks. Results demonstrate that NNM-LoRA exhibits significant improvements compared to baseline methods.

Index Terms— language understanding, low-rank adaptation, subspace, nuclear-norm

1. INTRODUCTION

Fine-tuning foundation models [1] on downstream tasks has proven highly effective and has now become the dominant approach in natural language and audio processing [2, 3]. However, performing full fine-tuning, which involves adjusting all model parameters, can be exceedingly resource-intensive in terms of both memory and computation, particularly as models increase in size and regularly adapt to a large number of tasks [4]. To alleviate this issue, researchers have introduced a series of Parameter-Efficient Fine-Tuning (PEFT) methods [5, 6, 7, 8] aiming to enhance training efficiency and reduce hardware requirements. These approaches have demonstrated the ability to achieve comparable performance to full fine-tuning while employing significantly fewer trainable parameters, often less than 1% of the original model size. Among these methods, LoRA [8] has emerged as one of the most popular choices for PEFT.

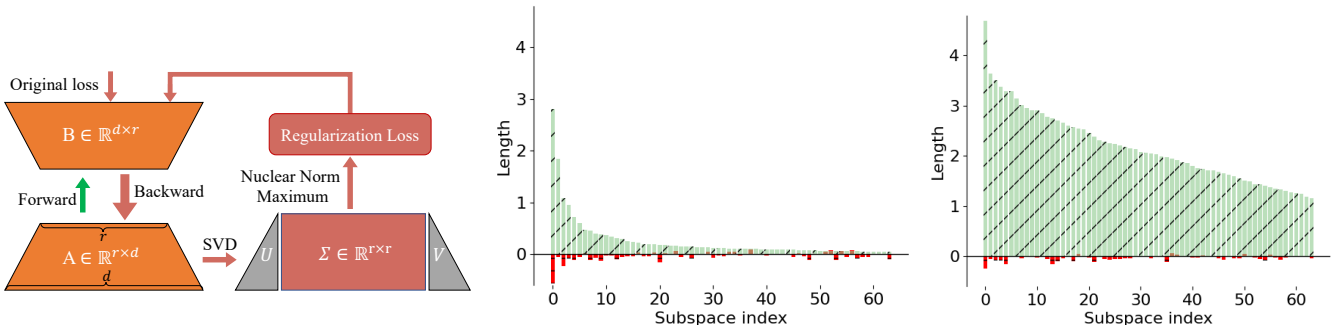
The key idea behind LoRA is to factorize the adaptation matrices into low-rank components, each consisting of a down-projection matrix A and an up-projection matrix B. This low-rank approximation learns efficient updates for over-parametrized models in smaller subspaces. Previous studies have demonstrated that increasing the LoRA dim r does not cover more meaningful subspaces; instead, most subspaces tend to accumulate random noise [8, 9]. Therefore, the LoRA dim r, which can be seen as the number of singular directions of the learned subspaces, is usually set quite small in practice. To illustrate this point, we provide a concrete example in Fig. 1(b). We observe that LoRA primarily amplifies the length (or magnitude) of a few specific singular directions, which suggests that the learned subspaces remain underutilized, often leading to suboptimal performance.

To address this challenge, we introduce NNM-LoRA, designed to take full advantage of learned subspaces by increasing the rank of adaptive matrices. Specifically, we maximize the nuclear norm, which is a convex surrogate for rank, of LoRA matrix A to compute regularization loss, as depicted in Fig. 1(a). Furthermore, we incorporate weighting of the regularization loss based on the Frobenius norm. NNM-LoRA achieves a more even distribution of singular values within the adaptation matrices BA, thereby significantly amplifying additional feature directions (see Fig. 1(c)). In brief, our main contribution is threefold:

- We theoretically show that the subspaces utilization of low-rank adaptation is reflected in the distribution of singular values within adaptation matrices.
- We introduce NNM-LoRA, a parameter-free regularizer that maximizes the nuclear norm of low-rank adaptation matrices to facilitate high-rank updates.
- Empirical results show that NNM-LoRA effectively harnesses more valuable singular directions of the learned subspace during training, resulting in notable performance enhancements.

† Equal contribution.

⋆ Corresponding authors. Email: liyiying10@nudt.edu.cn.



(a) Schematics of NNM-LoRA.

(b) Lengths in singular directions of pre-trained parameters and adaption with LoRA.

(c) Lengths in singular directions of pre-trained parameters and adaption with NNM-LoRA.

Fig. 1. Illustration of NNM-LoRA. We fine-tune BERT-base [10] on MNLI [11] using LoRA (Fig. 1(b)) and our proposed NNM-LoRA (Fig. 1(c)). We set the LoRA dim r to 64 and compare the subspace length of LoRA (hatched with "/") and pre-trained parameters (hatched with "-").

2. BACKGROUND

2.1. Parameter-Efficient Fine-Tuning

Most foundation models consist of L stacked transformer blocks [12], each block containing two sub-modules: a multi-head attention (MHA) and a fully connected network. Given the input sequence $X \in \mathbb{R}^{n\times d}$, h attention heads are computed by MHA parallelly:

$$\begin{aligned} \text{MHA}\ (X) &= \text{Concat}(\text{head}_1, ..., \text{head}_h)W_o, \\ \text{head}_i &= \text{Softmax}\left(XW_{q_i}(XW_{k_i})^\top/\sqrt{d_h}\right)XW_{v_i}, \end{aligned} \tag{1}$$

where $W_o \in \mathbb{R}^{d\times d}$ is an output projection, and $W_{q_i}, W_{k_i}, W_{v_i} \in \mathbb{R}^{d\times d_h}$ are query, key and value projections of head i. Typically, d_h is set to d/h.

PEFT offers a lightweight alternative to full fine-tuning transfer learning for foundation models. Adapter-tuning [5] inserts trainable adapters between transformer layers. These adapters typically consist of a down-projection matrix, a nonlinear activation function, and an up-projection matrix. Prefix-tuning [6] prepends trainable prefix task-specific vectors to the keys and values of the MHA. Training focuses exclusively on these prefix vectors. Prompt-tuning [7] simplifies prefix-tuning by appending learnable parameters only to the input word embedding layer. LoRA [8] introduces bypass modules for updating pre-trained models via up-down projection, which consists of down-projection matrices denoted as A and up-projection matrices denoted as B. During fine-tuning, the model starts with fixed pre-trained weights $W^{(0)}$ and updated to $W = W^{(0)} + \Delta W$. The forward pass can be expressed as:

$$y = Wx = W^{(0)}x + \Delta Wx = W^{(0)}x + BAx, \tag{2}$$

where $W, W^{(0)}, \Delta W \in \mathbb{R}^{d\times d}, A \in \mathbb{R}^{r\times d}$ and $B \in \mathbb{R}^{d\times r}$ with $r \ll d$. At the outset of training, random Gaussian initialization is applied to A, while B is initialized to zero.

2.2. Rank Adjustment for LoRA

Recent works focus on enhancing LoRA's performance by adjusting the rank of low-rank adaptation matrices. Drawing from the observation that the significance of weight matrices varies significantly across modules during fine-tuning, AdaLoRA [13] dynamically adjusts the rank of parameter matrices during training by masking the singular value diagonal matrix. Concurrent with our work, ReLoRA [14] merges multiple low-rank updates back into the original parameters through re-training to train high-rank networks. We also advocate for training high-rank networks, but from the perspective of harnessing the acquired subspaces. In contrast to prior work, which often involves complex hyper-parameter settings and implementations, our approach, NNM-LoRA, directly regulates the adaptation matrices, introducing just one hyper-parameter.

3. METHODOLOGY

In this section, we first provide a relation between singular values and subspaces, followed by introducing a method by regularizing low-rank adaptation matrices through nuclear norm maximization.

3.1. Model subspace utilization

Consider a neural network with the weight matrix $W \in \mathbb{R}^{d\times d}$. We carry out singular value decomposition (SVD) of W:

$$W = \mathbb{U}\Sigma\mathbb{V} = \begin{bmatrix}\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_d\end{bmatrix} \begin{bmatrix}\sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_d\end{bmatrix} \begin{bmatrix}\boldsymbol{v_1}^T \\ \boldsymbol{v_2}^T \\ \vdots \\ \boldsymbol{v_d}^T\end{bmatrix}, \tag{3}$$

where $\boldsymbol{u}_i, \boldsymbol{v}_i \in \mathbb{R}^d$ and $\sigma_i \in \mathbb{R}$ $(i = 1, 2 \ldots, d)$. For any matrix, when operating on a set of orthogonal right singular

vectors, its singular values correspond to the lengths of orthogonal left singular vectors. This implies that:

$$W\boldsymbol{v_i} = \sigma_i \boldsymbol{u_i}, \quad \text{where } i = 1, 2, \ldots, d. \tag{4}$$

For an input $x \in \mathbb{R}^d$, based on a set of bases of vector $\boldsymbol{v_i}$, we can approximate x as:

$$x = \sum_{i=1}^{d} c_i \boldsymbol{v_i}, \quad \text{where } c_i \text{ is a constant.} \tag{5}$$

When x passes through layer W, combined with (4), we obtain:

$$Wx = W \sum_{i=1}^{d} c_i \boldsymbol{v_i} = \sum_{i=1}^{d} c_i \sigma_i \boldsymbol{u_i}. \tag{6}$$

Equation (5) and (6) demonstrate that the matrix W operates as a projector, transforming the input x from the subspace formed by $\mathbb{V}$ to the subspace formed by $\mathbb{U}$. In addition, when the majority of singular values σ_i are close to zero, the corresponding basis vectors u_i are underutilized. Therefore, to fully exploit as many singular directions of the subspace as possible, i.e., ensuring that all u_i contribute to the outputs, it is imperative for all singular values to be sufficiently large and uniformly distributed. This implies that the matrix W should possess a high rank. Moreover, if only a few values are large while others approach zero, it implies that the neural network heavily relies on just a few singular directions.

3.2. Nuclear-Norm Maximization for LoRA

During the fine-tuning of pre-trained models using LoRA, we freeze pre-trained weights $W^{(0)}$, while training ΔW for updates. Our objective, in this context, is to enhance the utilization of a broader range of singular directions within the acquired subspace. To achieve this goal, we strive to increase the rank of LoRA's parameter matrix ($\Delta W = BA$).

We denote the down-projection matrix as $A \in \mathbb{R}^{r \times d}$. Since we have $r < d$, so the singular values of A are $\sigma_i (i = 1, 2, \cdots, r)$. The nuclear norm of A can be denoted as:

$$\|A\|_* = \sum_{i=1}^{r} \sigma_i, \tag{7}$$

which is also known as trace norm, and is the convex relaxation of rank [15].

Besides, the Frobenius norm (F-norm) of matrix A can be described as:

$$\|A\|_F = \sqrt{\sum_{i=1}^{r} \sum_{j=1}^{d} a_{ij}^2}, = \sqrt{\sum_{i=1}^{r} \sigma_i^2}, \tag{8}$$

where a_{ij} represents the elements in row i and column j of the matrix.

We introduce nuclear-norm maximization for low-rank adaptations within the framework of LoRA Given that the up-projection matrices B of the low-rank adaptation are initialized with zero, we exclusively apply the NNM regularizer to the down-projection matrices A, which is enough to regulate the singular values of ΔW. Denote ω as a set of trainable parameters of down-projection matrices A in LoRA. The NNM regularization loss can be computed as follows:

$$L_R(\omega) = -\sum_{l=1}^{L} (\|A_l\|_* / \|A_l\|_F), \quad \text{where } A_l \in \omega, \tag{9}$$

where L is the number of transformer layers. Overall, we update the model parameter ω by minimizing the following overall loss function:

$$L(\omega) = L_{original}(\omega) + \lambda L_R(\omega), \tag{10}$$

where $L_{original}(\omega)$ is the original loss function for fine-tuning, and $\lambda > 0$ is a hyper-parameter to control the regularization weights of NNM-LoRA.

In Equation 9, it's important to note that $\|A_l\|_F$ does not directly affect the differentiation process; its role is primarily to scale the NNM loss. We analyze the properties of $\|A_l\|_* / \|A_l\|_F$ as follows, and its effectiveness will be demonstrated in the experimental section. According to Cauchy-Schwarz inequality [16], we have:

$$\|A\|_* = \sum_{i=1}^{r} \sigma_i \leq \sqrt{r} \sqrt{\sum_{i=1}^{r} \sigma_i^2} = \sqrt{r} \cdot \|\mathbf{A}\|_F. \tag{11}$$

According to (11), $\sqrt{r}\|A\|_F$ serves as an upper bound for $\|A\|_*$. On one hand, the inequality holds with equality if and only if all singular values are equal. Optimizing the loss function (9) results in a more uniform distribution of singular values within matrix A, exerting an equivalent influence on the matrix ΔW. On the other hand, incorporating the F-norm scaling in the NNM regularization loss serves to control the nuclear norm from becoming excessively large.

4. EXPERIMENT

We apply NNM-LoRA for fine-tuning two models, BERT-base [10] with 110 million parameters and RoBERTa-large [17] with 355 million parameters. Our evaluation of the proposed method spans multiple language understanding tasks, including intent classification (CLINC [18]) and natural language inference (MNLI [11]) for BERT-base, and commonsense reasoning (SWAG [19]) for RoBERTa-large.

Implementation Details. Our implementation is based on the PEFT[1] code-base. We incorporate the NNM regularization on matrix A for all the query and value modules, using the plug-and-play manner.

[1] https://github.com/huggingface/peft

Baselines. We conduct comparisons between NNM-LoRA and the following methods, and not including the concurrent work - ReLoRA:

•*Full fine-tuning*: Given the pre-trained model, all parameters are updated through gradient updates.

•*LoRA* [8]: LoRA freezes the pre-trained models and only updates the low-rank matrices.

•*AdaLoRA* [13]: A variant of LoRA, offers dynamic adjustments to the rank of low-rank matrices during training.

Training hyper-parameters. Table 1 presents the hyper-parameter settings. We set max sequence length to 512, AdamW optimizer and a linear warmup learning rate strategy (ratio=0.1). All baseline methods share the core hyper-parameters used in full fine-tuning. In the case of LoRA, r and α correspond to the low-rank matrix dimension and scaling coefficients. AdaLoRA r and $\hat{r}$ represent the initial and target dimensions of low-rank matrices, respectively, while Δ_T is the steps interval between two budget allocations. To facilitate a meaningful comparison, NNM-LoRA employs the same low-rank matrix dimensions (r and α) as LoRA.

Table 1. Hyper-parameters setup in training

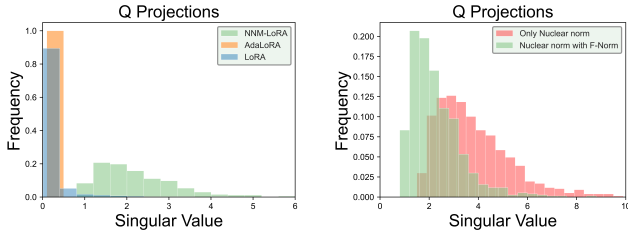
Method	Dataset (Model)	CLINC (BERT-base)	MNLI (BERT-base)	SWAG (RoBERTa-large)
Full fine-tuning	Learning Rate	1e-4	1e-5	5e-6
	Batch Size	64	64	32
	Epoch	50	30	20
LoRA	LoRA r / α	64 / 128	64 / 128	64 / 128
AdaLoRA	r / $\hat{r}$ / α	96 / 64 / 128	96 / 64 / 128	96 / 64 / 128
	Δ_T	100	1000	100
NNM-LoRA	λ	5	1	10

Main results. We report the median results over 5 random seeds in Table 2. The outcome for each run is obtained from the best epoch. The best results, excluding those from full fine-tuning, are highlighted in bold. It is evident that NNM-LoRA consistently outperforms other methods in our experiments. For instance, NNM-LoRA achieves 94.52% accuracy on CLINC, surpassing LoRA by 1%. Similarly, on the SWAG dataset, NNM-LoRA achieves an accuracy of 89.21%, surpassing both LoRA at 88.55% and full fine-tuning at 89.06%. Additionally, it's noteworthy that achieving satisfactory performance with AdaLoRA in tasks like MNLI and SWAG can be challenging, primarily due to the complexity of its hyper-parameter settings.

Table 2. Accuracy results on three tasks.

Method	Venue	CLINC	MNLI	SWAG
Full fine-tuning	-	94.87	84.44	89.06
LoRA [8]	ICLR 2021	93.52	82.29	88.55
AdaLoRA [13]	ICLR 2023	93.65	76.97	82.74
NNM-LoRA	**Ours**	**94.52**	**83.36**	**89.21**

Distribution of singular values. Fig. 2(a) shows singular values spectra of the query matrix of all layers of BERT-base, which are trained with different baselines on CLINC. And singular values less than 10^{-6} are ignored. It is clearly observed that NNM-LoRA significantly increases the singular value of low-rank matrices. Furthermore, since the properties and analysis of the value matrix closely resemble those of the query matrix, we omit its analysis for the sake of brevity.



(a) Distributions for baselines (b) Effect of F-norm on distribution

Fig. 2. Singular value distribution of low-rank matrices ΔW

Ablation studies. We conducted ablation studies on two critical components of NNM-LoRA: λ and F-norm. From the findings in Table 3, it's apparent that the impact of λ on performance exhibits a local maximum at values such as $\lambda = 5$. Moreover, NNM-LoRA consistently performs worse when F-norm is not applied compared to when it is included. This aligns with our previous analysis, where the absence of F-norm causes NNM-LoRA to prioritize increasing all singular values rather than achieving a more uniform distribution. As illustrated in Fig. 2(b), the inclusion of F-norm leads to a more uniform distribution of singular values, whereas its absence results in the parameter matrix heavily relying on a few large singular values and their corresponding basis vectors, ultimately reducing subspace utilization.

Table 3. Ablation studies of λ and F-norm on CLINC. (with = w/, without = w/o.)

w/o NNM	λ	0.1	2	5	10	50
93.52	**w/o F-norm**	**93.68**	93.77	93.81	93.45	93.32
	w/ F-norm	93.55	**94.23**	**94.52**	**94.32**	**94.13**

5. CONCLUSION AND LIMITATIONS

In this paper, we introduce NNM-LoRA, a novel approach to fine-tuning pre-trained language models. NNM-LoRA leverages nuclear norm maximization to regulate low-rank adaptation matrices, addressing the underutilization of subspaces in a straightforward manner. Our experimental results demonstrate the effectiveness of NNM-LoRA across various language understanding tasks.

Our work has limitations. Compared to LoRA, NNM-LoRA requires the calculation of the nuclear norm and F-norm of down-projection matrices A, which introduces 30% and 10% additional time overhead for BERT-base and RoBERTa-large, respectively. One promising future direction is to explore approximate methods for computing NNM regularization loss efficiently.

6. ACKNOWLEDGEMENTS

This work was partially supported by the National Key R&D Program of China (No. 2021ZD0112904) and was partially supported by the National Natural Science Foundation of China (No.62206307).

7. REFERENCES

[1] Rishi Bommasani et al. "On the opportunities and risks of foundation models". In: *arXiv:2108.07258*. 2021.

[2] Colin Raffel et al. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *The Journal of Machine Learning Research*. Vol. 21. 1. JMLRORG, 2020, pp. 5485–5551.

[3] Hassan Akbari et al. "Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text". In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021, pp. 24206–24221.

[4] William Fedus, Barret Zoph, and Noam Shazeer. "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity". In: *The Journal of Machine Learning Research*. Vol. 23. 1. JMLRORG, 2022, pp. 5232–5270.

[5] Neil Houlsby et al. "Parameter-efficient transfer learning for NLP". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2790–2799.

[6] Xiang Lisa Li and Percy Liang. "Prefix-Tuning: Optimizing Continuous Prompts for Generation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2021, pp. 4582–4597.

[7] Brian Lester, Rami Al-Rfou, and Noah Constant. "The Power of Scale for Parameter-Efficient Prompt Tuning". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 3045–3059.

[8] Edward J Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations*. 2021.

[9] Tim Dettmers et al. "Qlora: Efficient finetuning of quantized llms". In: *arXiv preprint arXiv:2305.14314* (2023).

[10] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186.

[11] Adina Williams, Nikita Nangia, and Samuel R Bowman. "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference". In: *Proceedings of NAACL-HLT*. 2018, pp. 1112–1122.

[12] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. Vol. 30. 2017.

[13] Qingru Zhang et al. "Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning". In: *The Eleventh International Conference on Learning Representations*. 2023.

[14] Vladislav Lialin et al. "Stack More Layers Differently: High-Rank Training Through Low-Rank Updates". In: *arXiv preprint arXiv:2307.05695*. 2023.

[15] Emmanuel Candes and Benjamin Recht. "Exact matrix completion via convex optimization". In: *Communications of the ACM*. Vol. 55. 6. ACM New York, NY, USA, 2012, pp. 111–119.

[16] Hui-Hua Wu and Shanhe Wu. "Various proofs of the Cauchy-Schwarz inequality". In: *Octogon mathematical magazine*. Vol. 17. 1. 2009, pp. 221–229.

[17] Liu Zhuang et al. "A Robustly Optimized BERT Pre-training Approach with Post-training". In: *Proceedings of the 20th Chinese National Conference on Computational Linguistics*. Huhhot, China: Chinese Information Processing Society of China, Aug. 2021, pp. 1218–1227.

[18] Stefan Larson et al. "An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019.

[19] Rowan Zellers et al. "SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018, pp. 93–104.