LoRAN: Improved Low-Rank Adaptation by a Non-Linear Transformation

Yinqiao Li[1] **Linqi Song**[1,2*] **Hanxu Hou**[1*]
[1]Department of Computer Science, City University of Hong Kong
[2] Shenzhen Research Institute, City University of Hong Kong
li.yin.qiao.2012@hotmail.com
linqi.song@cityu.edu.hk
houhanxu@163.com

Abstract

In this paper, we study parameter-efficient fine-tuning methods for large pre-trained models. Specifically, we improve LoRA approaches to alleviate the performance loss from the constrained adapter by introducing a non-linear transformation (call it LoRAN). For a better adaptation, we also design a new non-linear function to appropriately fit the accumulated weight updates. We test our method in multiple advanced large language models. Experimental results show that our LoRAN significantly outperforms a strong baseline on SAMSum and 20 Newsgroups tasks. Moreover, when a lower rank is applied, our approach even yields a 1.95-point improvement in the classification task.

1 Introduction

Recently, large language models (LLMs) have shown great improvements on a wide range of NLP tasks. Methods of this kind make it possible to learn universal representations from large corpora and adapt pre-trained models to downstream tasks through fine-tuning (Zhao et al., 2023). Early fine-tuning methods optimize models within the entire parameter space (Brown et al., 2020), whereas parameter-efficient fine-tuning (PEFT) have successfully trained downstream models with fewer parameters (Hu et al., 2022; Liu et al., 2021a; Lester et al., 2021). As an instance of the latter, the low-rank adaptation (LoRA) introduces lightweight trainable matrices to fit the accumulated weight updates while most LLM parameters are frozen. This leads to an efficient training process with a smaller memory footprint than full fine-tuning.

Like recent adapter-based methods, LoRA adds a small number of extra parameters to adapt models to downstream data. Under the low-intrinsic-rank hypothesis (Aghajanyan et al., 2021), the multiplication of two low-dimension matrices is introduced to fit every weight update matrix during fine-tuning (see Figure 1(a)), and then the downstream model can be easily obtained by merging back all the multiplications with the frozen parameters. This low-rank decomposition dramatically reduces the number of trainable parameters, even making it possible to equip multiple portable LoRA models for different tasks onto one shared foundation.

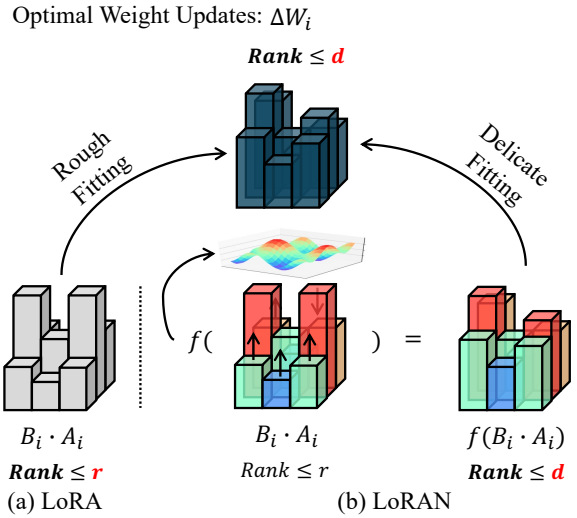

Figure 1: Comparison of weight update fitting in fine-tuning with LoRA and LoRAN.

However, the effectiveness of LoRA heavily relies on the match between the adapter capacity and the downstream task. If the adapters are too small, there will be a significant gap between adapter outputs and optimal weight updates, leading to a performance loss in task transfer. Previous efforts on this issue have focused on budgeting adapter parameters (Zhang et al., 2023b; Valipour et al., 2023) and upgrading matrix decomposition (Hyeon-Woo et al., 2022; Yeh et al., 2023). These methods are able to narrow the gap by sufficient parameters and effective computations, but cannot thoroughly plug the gap to the same fitting precision with full fine-tuning efficiently due to low-rank multiplications.

*Corresponding author

Findings of the Association for Computational Linguistics: EACL 2024, pages 3134–3143
November 12-16, 2024 ©2024 Association for Computational Linguistics

In this work, we enhance the adapter capacity without additional parameters, enabling the adapter to fit weight updates more delicately. We present an improvement of LoRA, introducing a non-linear transformation after matrix decomposition (call it **Lo**w-**R**ank **A**daptation with **N**on-Linear Transformation, or LoRAN for short). It leads to an adapter that even matches the full-rank model capacity (see Figure 1(b)). In addition to the new framework, we also develop a novel non-linear function for LoRAN, that ensures a reasonable practice based on the matrix features of weight updates.

Our LoRAN is simple to implement and straightforwardly applicable to various LoRA variants. We experiment with it in Flan-T5, Falcon, and Llama 2 systems. Experimental results on SAMSum and 20 Newsgroups tasks show significant improvement to the baseline. When the rank is limited lower, our system even yields a more considerable improvement of +1.95 point accuracy in the 20 Newsgroups task. More interestingly, in the classification task, LoRAN presents greater sensitivity to distinguishing minor differences between similar classes. This corresponds to our motivation for addressing the rough weight update fitting in low-rank adapters and indicates a promising line of research on applying LoRAN to complicated tasks, such as extremely low-resource fine-tuning.

2 The Method

In this work, we use LoRA for description[1]. Taking a downstream model as an example, numerous fully connected layers are equipped. A fine-tuned weight matrix W_i^D can be defined as $W_i^D = W_i^F + \Delta W_i$ where W_i^F is the corresponding weights of the foundation model and ΔW_i refers to the accumulated weight updates in fine-tuning.

2.1 Low-Rank Adapter

In low-rank adapters, the object is to fit the optimal ΔW_i with fewer parameters to save memory footprint. For a specific neural position, the foundation matrix W_i^F is frozen during fine-tuning, while the adapter parameters are updated to learn the weight changes ΔW_i. In LoRA, a low-rank decomposition is applied to represent $\Delta W_i \in \mathbb{R}^{d\times k}$, like this

$$\Delta W_i = B_i \cdot A_i \tag{1}$$

where $B_i \in \mathbb{R}^{d\times r}$ and $A_i \in \mathbb{R}^{r\times k}$ are used to reduce the dimension of the trainable parameters from d to r and then increase it back to k. Due to the setting of $r \ll min\{d, k\}$ in the common practice, the number of parameters in fitting ΔW_i is far less than that in a full fine-tuning.

2.2 Adapter with Non-Linear Transformation

However, due to the decomposition, the rank of ΔW_i is also limited equal or less than r, indicates a poor model capacity (Zhang et al., 2023b; Valipour et al., 2023). To verify this, we evaluate the change in information quantities after fine-tuning with and without LoRA. A significant information loss is observed when low-rank adapter is applied (see Appendix A.3). This encourage us to study on adapters with higher capacity for fine-grained weight update fitting.

In this work, we introduce a simple non-linear mapping for the adapter to model more delicately, rather than forcefully fitting the high-rank ΔW_i with the low-rank result. We call it **Lo**w-**R**ank **A**daptation with a **N**on-Linear Transformation (LoRAN). Here, we re-formalize the Eq. (1) as:

$$\Delta W_i = f(B_i \cdot A_i) \tag{2}$$

where $f(\cdot)$ refers to the non-linear function without extra parameters. This enables adapter outputs to align with optimal weight updates with the identical rank, indicating a promising model capacity for downstream tasks (See Figure 1(b)).

2.3 Scaled Sine Interference

Our model is flexible. For the non-linear transformation, we can directly apply existing activation functions with reasonable derivatives and ranges. However, we found that choices without careful consideration will not bring maximum performance gains. Some cause too weak or too strong impacts in the weight space, even creating blind spots and unfair mappings in the transformation. This necessitates the non-linear function that consistently returns the approximated value of its argument. A simple manner is to add a constrained interference on the decomposition. Here, we design a non-linear function named **S**caled **S**ine **Inter**ference (𝕊inter):

$$f(x) = A \cdot \sin(\omega \cdot x) \odot x + x \tag{3}$$

where $\sin(\cdot)$ refers to the sine function. A and ω are the amplitude and frequency to control the interference degree and phase velocity. It is noteworthy

[1]Note that although we restrict ourselves to LoRA here, the methodology can be easily applied to other LoRA variants.

Foundation Model	ROUGE-1			ROUGE-2			ROUGE-L		
	PEFT Method		Δ	PEFT Method		Δ	PEFT Method		Δ
	QLoRA	LoRAN		QLoRA	LoRAN		QLoRA	LoRAN	
Flan-T5-Large	48.69	49.04	+0.35	22.91	22.97	+0.06	39.47	39.42	-0.05
Falcon-7b	50.16	50.67	+0.51	25.47	25.85	+0.38	41.74	42.50	+0.76
Llama-2-7b	52.72	53.27	+0.55	27.92	28.54	+0.62	44.10	44.70	+0.60
Llama-2-13b	52.86	53.14	+0.28	28.46	28.82	+0.36	44.66	44.85	+0.19

Table 1: Comparison of QLoRA and LoRAN methods on the SAMSum task with large language models.

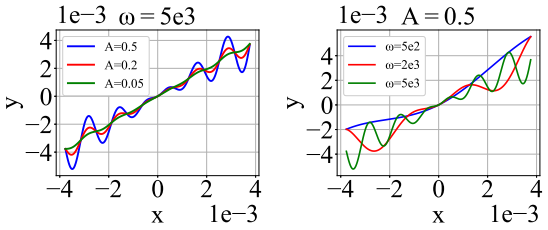

Figure 2: The 𝕊inter activation function.

Foundation Model	Accuracy		
	PEFT Method		Δ
	QLoRA	LoRAN	
Flan-T5-Large	75.45	75.80	+0.35
Falcon-7b	68.33	68.80	+0.47
Llama-2-7b	73.39	74.61	+1.22
Llama-2-13b	75.99	76.68	+0.69

Table 2: Comparison of QLoRA and LoRAN methods on the 20 Newsgroups task with large language models.

that a Hadamard product is used to offer appropriate scales on the sine wave based on the inputs. Figure 2 shows the graph of our 𝕊inter.

3 Experiments

3.1 Experimental Setup

For the supervised fine-tuning, we test our approach on the SAMSum summarization task(Gliwa et al., 2019) and 20 Newsgroups classification task (Lang, 1995). To ensure reliable experimental results, we implement with Flan-T5 (Chung et al., 2022), Falcon (Penedo et al., 2023), and Llama 2 (Touvron et al., 2023). Models with various parameter scales (0.7B, 7B, and 13B) are also experimented to evaluate the universality of our LoRAN.

We fine-tuned downstream models using low-rank methods. Both the baseline and our LoRAN were constructed on QLoRA, which was a 4-bit quantized LoRA with superior efficiency and comparable performance (Dettmers et al., 2023). In the adapter, we set r to 64, α to 16, and applied 𝕊inter in LoRAN with A = 5e-5 and ω = 1e4. Our fine-tuning system ran for 5 epochs with a batch size of 16. We froze LLM parameters and optimized adapter parameters using AdamW with a learning rate of 2e-4, β_1 = 0.9 and β_1 = 0.999. Dropout (rate = 0.2) and gradient clipping (maximum gradient norm = 0.3) were also adopted for regularization and stabilizing the training process.

3.2 Results

3.2.1 Performance Improvements

Here, we report the performance of fine-tuned models with QLoRA and LoRAN in Table 1 and Table 2. First of all, our LoRAN enhances the performance across different combinations of foundation models and tasks significantly. In LLMs ($\geq 7B$), without introducing any parameters, the fine-tuned models with LoRAN outperform QLoRA by 0.47 ROUGE and 0.79 accuracy scores on average. Llama 2 (7B) even yields a +1.22 accuracy improvement in the 20 Newsgroups task. Additionally, for foundations with fewer parameters, such as Flan-T5-Large (0.7B), LoRAN also surpasses slightly than the vanilla low-rank method. One possible reason of the limitation is that a small adapter is sufficient for transferring a weak foundation to downstream tasks due to its original low confidence scores. Therefore, appending a powerful non-linear transformation does not bring an expected significant improvement under a generous setting of $r = 64$.

Additionally, we test LoRAN with a stricter rank. The model performance is reported in Table 3 when the r is limited to 8. Our LoRAN outperforms the baseline in both summarization and text classification. It even achieves a more substantial improvement than the setting of $r = 64$. For example, in the 20 Newsgroups task, LoRAN yields a 1.95-point improvement, which is more noticeable than the improvement in Table 2. This is because there is a wider gap between the capacity of a smaller

Task	Metric	PEFT Method		Δ
		QLoRA	LoRAN	
SS†	ROUGE-1	52.38	53.00	+0.62
	ROUGE-2	27.78	28.19	+0.41
	ROUGE-L	44.06	44.59	+0.53
NG‡	Accuracy	71.62	73.57	+1.95

Table 3: Comparison of QLoRA and LoRAN methods with lower rank ($r = 8$). The foundation model is Llama-2-7b. †SS=SAMSum. ‡NG=20 Newsgroups.

Precision	Metric	PEFT Method		Δ
		Base†	Our‡	
4-bit	ROUGE-1	48.69	49.04	+0.35
	ROUGE-2	22.91	22.97	+0.06
	ROUGE-L	39.47	39.42	-0.05
32-bit	ROUGE-1	48.88	49.97	+1.09
	ROUGE-2	22.86	23.63	+0.77
	ROUGE-L	39.71	39.94	+0.23

Table 4: Comparison of (Q)LoRA and LoRAN methods with/without quantization on the SAMSum task. The foundation model is Flan-T5-Large. †Base=LoRA/QLoRA. ‡Our=LoRAN.

adapter and weight updates. Non-linear transformations benefit more in these tricky situations.

Moreover, we also evaluate our LoRAN without quantization. Table 4 presents the SAMSum results based on Flan-T5-Large. Compared to the quantized results, our LoRAN shows a surprisingly greater improvement in the non-quantized setting (32-bit training). The results indicate a promising performance of LoRAN in parameter-efficient fine-tuning other larger language models.

3.2.2 Ablation Study

To verify the contributions of the LoRAN framework and $\mathbb{S}$inter, we compare different activation functions used in our method. We use Identity, Swish-1 and Swish-25 (Ramachandran et al., 2018) for comparison with $\mathbb{S}$inter. Identity serves as the baseline, because it simplifies LoRAN back to QLoRA, and two Swish functions are chosen due to their compatible derivatives and ranges. In Table 5, performance improvements are observed with all the non-linear ways, with $\mathbb{S}$inter yielding the best. The relatively smaller improvement of Swish might be due to the contraction mapping in the weight space, which requires more training steps to fit the weight updates, posing challenges for LoRAN.

Function	ROUGE-1	ROUGE-2	ROUGE-L
Identity	52.72	27.92	44.10
Swish-1†	52.61	27.95	44.45
Swish-25‡	52.66	28.04	44.46
$\mathbb{S}$inter	53.27	28.54	44.70

Table 5: Comparison of activation functions in the LoRAN method on the SAMSum task. The foundation model is Llama-2-7b. †‡Swish-1 and Swish-25 refer to apply $\beta = 1$ and $\beta = 25$ in the Swish function.

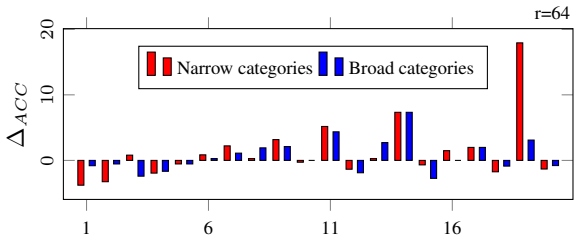

Figure 3: The difference in classification accuracy with QLoRA and LoRAN vs. the topic number. The foundation model is Llama-2-7b.

3.2.3 Contribution Analysis

To figure out the advantages of LoRAN, we detail the model contribution on each class in the 20 Newsgroups task. According to the manual, every class corresponds to a specific topic within narrow categories. Meanwhile, 20 classes are grouped into 5 broad categories based on their similarities[2]. Figure 3 shows the improvement on every class[3]. For the narrow categories (red), it only scores when the prediction exactly matches the gold label. While for the broad categories (blue), it scores if the prediction belongs to the correct broad category. The results show that the main improvements are in the narrow task. The most significant stride (#19) comes from the broad category with the least common samples on average. The non-linear transformation helps the adapter enhance its sensitivity to the differences of remarkably similar topics, especially in low-resource data. This aligns with our premise that the non-linear transformation aids in reconstructing weight updates more delicately.

4 Related Work

PEFT has been proposed to improve computing and memory efficiency for applying LLMs to downstream tasks (Liu et al., 2021a; Lester et al., 2021; Liu et al., 2021b). One classic method, LoRA,

[2]Appendix A.2 shows the detailed official categories.

[3]We also discuss the LoRAN contribution with a more limited rank value in Appendix A.5.

Foundation Model	Training Cost (GPU hours)		
	QLoRA	LoRAN	Δ
Llama-2-7b	2.9	3.1	+0.2
Llama-2-13b	4.9	5.4	+0.5

Table 6: Comparison of the time consumption between QLoRA and LoRAN with Llama 2. The downstream task is 20 Newsgroups.

uses adapters to save a substantial number of trainable parameters, making it popular to produce task-specific models on resource-limited devices (Hu et al., 2022). However, its matrix decomposition restricts weight updates to the low-rank space, constraining the model's expressiveness.

A strand of addressing this issue is to budget adapter parameters, adding suitably sized adapters at critical neural positions (Zhang et al., 2023b; Valipour et al., 2023), and the other typical line is to upgrade the decomposition way for higher performance (Hyeon-Woo et al., 2022; Yeh et al., 2023). Nevertheless, both approaches either fail to thoroughly bridge the gap between the adapter capacity and weight updates, or require additional parameters. No discussion exists on a method that can achieve low-rank training comparable to full fine-tuning under the same number of parameters.

5 Conclusions

We have presented an improved low-rank adaptation with non-linear transformation for a more delicate weight update fitting in fine-tuning. Meanwhile, for more reliable progress, a brand-new non-linear function is proposed. Experiments on SAMSum and 20 Newsgroups tasks both show significant improvements over the baseline. When a lower rank is used, it even achieves a 1.95-point improvement in the classification task.

6 Limitations

Due to the non-linear transformation, some extra computational time is required even though no additional parameters are introduced. However, it is worth noting that the additional time consumption is low. For example, in Llama 2 (both 7B and 13B) experiments, the increased time costs are limited below 0.5 GPU hours (See Table 6). Moreover, when techniques like kernel fusion are applied, the added time of LoRAN with $\mathbb{S}$inter can be further compressed. Another limitation that we plan to address in the future work is experimenting with our method in more LoRA variants, particularly adapters with rank value optimization like AdaLoRA (Zhang et al., 2023b), DyLoRA (Valipour et al., 2023), etc. This will allow LoRAN to maximize the benefit of delicate fitting with a reliable parameter guarantee.

Acknowledgements

This work was supported in part by the Research Grants Council of the Hong Kong SAR under Grant GRF 11217823 and Collaborative Research Fund C1042-23GF, the National Natural Science Foundation of China under Grant 62371411, InnoHK initiative, the Government of the HKSAR, Laboratory for AI-Powered Financial Technologies.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7319–7328. Association for Computational Linguistics.

Mitra Baratchi, Can Wang, Steffen Limmer, Jan N. van Rijn, Holger H. Hoos, Thomas Bäck, and Markus Olhofer. 2024. Automated machine learning: past, present and future. *Artif. Intell. Rev.*, 57(5):122.

Rafael Barbudo, Sebastián Ventura, and José Raúl Romero. 2023. Eight years of automl: categorisation, review and trends. *Knowl. Inf. Syst.*, 65(12):5097–5149.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao,

Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Xin He, Kaiyong Zhao, and Xiaowen Chu. 2021. Automl: A survey of the state-of-the-art. *Knowl. Based Syst.*, 212:106622.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Nam Hyeon-Woo, Moon Ye-Bin, and Tae-Hyun Oh. 2022. Fedpara: Low-rank hadamard product for communication-efficient federated learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M Asano. 2024. VeRA: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations*.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, pages 331–339. Morgan Kaufmann.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021a. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *CoRR*, abs/2110.07602.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT understands, too. *CoRR*, abs/2103.10385.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon LLM: outperforming curated corpora with web data only. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. 2018. Searching for activation functions. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2023. Dylora: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 3266–3279. Association for Computational Linguistics.

Shin-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard B. W. Yang, Giyeong Oh, and Yanmin Gong. 2023. Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. *CoRR*, abs/2309.14859.

Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. 2023a. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning. *CoRR*, abs/2308.03303.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023b. Adaptive budget allocation for

parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Hongyun Zhou, Xiangyu Lu, Wang Xu, Conghui Zhu, and Tiejun Zhao. 2024. Lora-drop: Efficient lora parameter pruning based on output evaluation. *CoRR*, abs/2402.07721.

A Appendix

We organize the Appendix in 5 parts:

- Codes and fine-tuned adapter models of our LoRAN (Appendix A.1).
- Broad and narrow categories of the 20 Newsgroups dataset according to the official task manual (Appendix A.2).
- Information quantity analysis when using LoRA and full fine-tuning (Appendix A.3).
- Discussion of the relation between the adapter capacity and accumulated weight updates (Appendix A.4).
- Contribution analysis of LoRAN with a setting of $r = 8$ in the 20 Newsgroups task (Appendix A.5).

A.1 Codes and Models

For a convenient use of our approach, we open-source the LoRAN here. It is developed based on the widely-used PEFT package from Huggingface, that allows researchers to upgrade their projects for LoRAN more easily. Moreover, we also provide the fine-tuned QLoRA and LoRAN models from our experiments so that everyone can reproduce our results directly. These models are fine-tuned with open-sourced LLM-Finetuning-Toolkit on a single NVIDIA A100 (40GB).

Moreover, due to the use of non-linear transformation, our LoRAN requires a small amount of additional time during fine-tuning. Taking Llama 2 models (7B and 13B) as examples, we present the time consumption of our experiments in Table 6. It is observed that LoRAN claims < 30 extra GPU minutes compared to the baseline in performing non-linear transformations. This part can be further compressed when kernel fusion is applied in a real-world implementation.

A.2 Categories of 20 Newsgroups

We present the official classes of the 20 newsgroups task according to its website in Table 7. 20 classes refer to 20 news topics in narrow categories, while they are clustered into 5 broad categories based on their similarities. Table 7 also presents the number of samples in every class.

Categories		Sample
Broad	Narrow	Number
A	comp.graphics (#1)	550
	comp.os.ms-windows.misc (#2)	554
	comp.sys.ibm.pc.hardware (#3)	561
	comp.sys.mac.hardware (#4)	536
	comp.windows.x (#5)	575
B	rec.autos (#6)	538
	rec.motorcycles (#7)	550
	rec.sport.baseball (#8)	546
	rec.sport.hockey (#9)	558
C	sci.crypt (#10)	567
	sci.electronics (#11)	562
	sci.med (#12)	571
	sci.space (#13)	563
D	misc.forsale (#14)	564
E	talk.politics.misc (#15)	437
	talk.politics.guns (#16)	525
	talk.politics.mideast (#17)	520
F	talk.religion.misc (#18)	338
	alt.atheism (#19)	448
	soc.religion.christian (#20)	581

Table 7: Broad and narrow categories in the 20 newsgroups task. Each narrow category corresponds to a specific topic, while closely related topics are clustered in a broad category. The partition is defined in http://qwone.com/ jason/20Newsgroups.

A.3 Information Quantity Analysis

The limited rank in LoRA constrains the model performance. To detail the impact, we evaluate the change in information quantities after fine-tuning with and without LoRA at the beginning of this work. More particularly, we compute the SVD on the real weight updates and observe their singular values. Figure 5 presents the distribution of dimensions with various singular values, where a dimension with a higher singular value indicates an accommodation with more information. For the low-rank adapter, almost all the dimensions fall within low-information ranges. In contrast, full fine-tuning handles complex downstream information more manageable with richer information. All these show the possibility of enhancing the fine-tuned model by plugging the gap and motivate us to study our LoRAN.

A.4 Discussion of the Adapter Capacity and Accumulated Weight Updates

In this section, we discuss the role of adapters in modern fine-tuning and compare various effects of low-rank implementations (different parameter

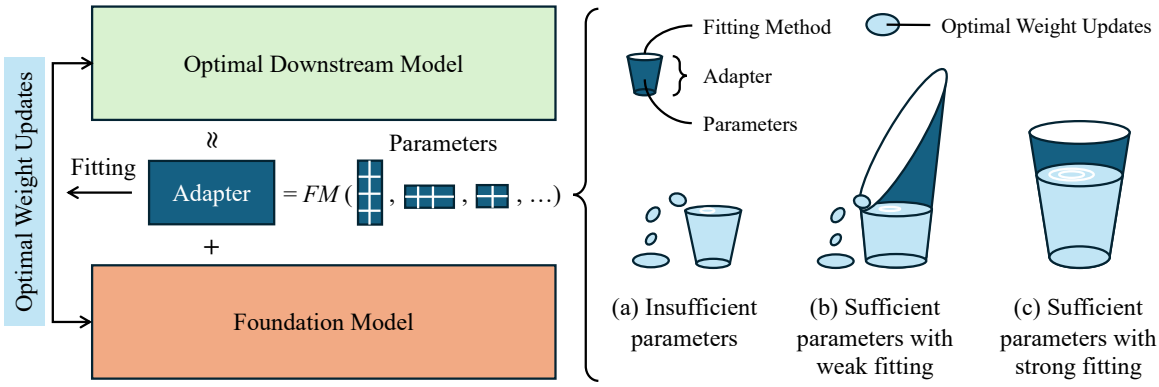

Figure 4: The process of fine-tuning with different types of adapter implementations. In an adapter, the parameters are fundamental, and we use various fitting methods to utilize them to perform the adaptation. Three kinds of adapter implementations exist: (a) When the parameters are insufficient, even if the fitting method is reasonable, the adapter cannot work well; (b) If enough parameters are used, the adapter capacity is also limited when the fitting approach is not powerful; (c) Both the number of parameters and the fitting way meet the requirement, the adapter achieves an adequate capacity for weight updates.

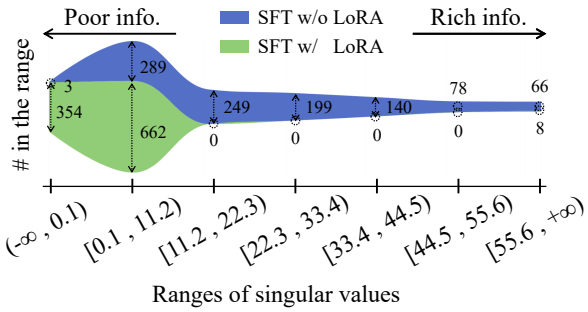

Figure 5: The change of information quantities in fine-tuning the RoBERTa-Large model with/without LoRA. The information quantities are defined by singular values, and the numbers of dimensions across various singular value ranges are presented. The query projection matrix in the bottom sub-layer serves as the observation target.

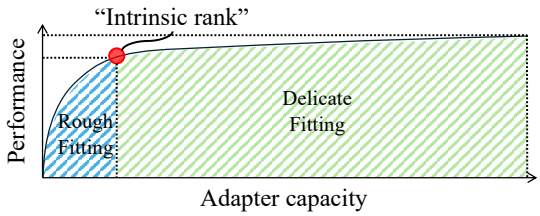

Figure 6: The relation between the downstream performance and adapter capacity.

settings and fitting methods). This part describes the position and target of our LoRAN in more detail and may provide insights for future related work.

In the LLM framework, the downstream models are trained using powerful foundation models. Assuming an ideal downstream model exists, the parameter difference between it and the foundation model needs to be modeled for adaptation. In theory, these required weight updates can be exactly learned with full fine-tuning, but a huge amount of computing resources are necessary for this process. To address the efficiency issue, the low-rank adaptation is proposed. Methods of this kind consider a low "intrinsic rank" inside the weight update matrix and employ a small adapter for fitting (see Figure 4). In practice, this procedure is a performance-efficiency-balance that estimates a suitable adapter capacity for the current task and foundation model, like the illustration in Figure 6. The low-rank adaptation is able to guarantee a rough fitting but with a performance limitation. The disadvantage becomes even more pronounced in complex downstream tasks that demand a larger capacity. More formally, the fitting process can be described as

$$\Delta W_i = FM(P_0, P_1, ...) \tag{4}$$

The process is composed of two parts, the fitting method $FM(\cdot)$ and the parameter set $\{P_0, P_1, \dots\}$. The number of parameters defines the upper bound of the adapter performance. In contrast, the fitting approach is used for parameter organizations, determining the parameter efficiency and the fitting precision. Two factors influence the adapter capacity together, just like the quantity of material and the shape define the volume of a bucket in Figure 4. Concentrating on only one aspect cannot achieve the optimum. Similar results appear in studies of deep learning. For example, in Automated Machine Learning (AutoML),

designing neural networks applies both Neural Architecture Search (NAS) and Hyper-Parameter Optimization (HPO) for the best results in a specific task (He et al., 2021; Baratchi et al., 2024; Barbudo et al., 2023).

Back to the adaptation of LLMs, the number of parameters and fitting methods are also two core lines of improving LoRA. The former focuses on allocating appropriate numbers of parameters for different neural positions (Zhang et al., 2023b; Valipour et al., 2023; Zhou et al., 2024), while the latter seeks to maximize parameter efficiency (Yeh et al., 2023; Kopiczko et al., 2024; Zhang et al., 2023a). This work mainly targets the latter. The current fitting methods are not perfect in real-world implementation. Many extra parameters are needed to achieve a delicate fitting comparable to full fine-tuning. For instance, LoHa enhances the rank of LoRA outputs from r to $r^2/4$, making it possible for the adapter capacity to approach the requirement of weight updates with some parameter increases (Hyeon-Woo et al., 2022). Our work further researches this problem, aiming to improve the adapter capacity to the full-rank level without introducing any additional parameters.

A.5 Contribution Analysis of LoRAN

To analyze the contribution of our LoRAN, we show the performance change of every class in the 20 Newsgroups task (see Figure 7). Different from the experiment in Section 3.2.3, the results here are with a setting of lower rank ($r = 8$).

Interestingly, we observe that the improvements in Figure 7 show a different phenomenon, which is the similar improvement trends between narrow and broad categories. This indicates the improvements primarily come from the broad categories, which seems inconsistent with the experimental results of setting the rank to 64 (see Figure 3). This is because predicting broad categories is generally easier than predicting narrow categories. In the 20 Newsgroups task, using a rank of 64 allows LoRA to accurately predict most of the broad categories while struggling with the narrow ones. Our LoRAN address this issue by fitting weight updates more delicately with the non-linear transformation, helping the system to distinguish minor differences between similar narrow classes. However, when the rank is further reduced to 8, adapters struggle even with broad category prediction due to insufficient parameters (as mentioned in Appendix A.4, the number of parameters and the fitting method

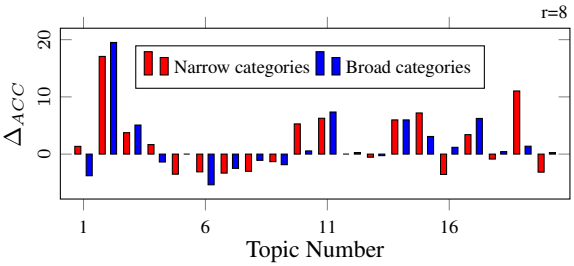

Figure 7: The difference in classification accuracy with QLoRA and LoRAN vs. the topic number. The foundation model is Llama-2-7b. Both narrow and broad categories are observed. The figure follows a setting of $r = 8$.

determine the capacity together). The non-linear transformation then prioritizes optimizing the basic task - broad category prediction. That explains why most improvements are from the broad categories in Table 7. Additionally, this also suggests a promising line of research on applying this work to AdaLoRA (Zhang et al., 2023b) or other variants with rank value optimization. By doing so, necessary adapter parameters are allocated to the suitable position, and the non-linear function supports finely fitting the weight updates.