Scale Regularization for Stable Low-Rank Adaptation

Tan Xeng Ian

Nanyang Technological University
College of Computing and Data Science
Singapore
xtan122@e.ntu.edu.sg

Abstract

Low-Rank Adaptation (LoRA) has emerged as a practical and efficient method for fine-tuning large language models under limited computational budgets. However, recent studies have shown that LoRA can suffer from training instability when applied to models with large embedding dimensions, due to the imbalanced in magnitudes between its low-rank matrices. In this work, we propose a novel regularization strategy that stabilizes LoRA training by penalizing logarithmic magnitude differences between the low-rank matrices, showing theoretically that it should lead to efficient feature learning. We further propose evaluation methods to systematically assess training stability and performance of our proposed solution along with other LoRA variants.

1 Introduction

With the increase in the number of open sourced models available to the average consumer, Low-Rank Adaptation (LoRA) have become essential tools for adapting large language models with limited computational resources (Hu et al. 2021). LoRA works by introducing trainable low-rank matrices A and B to update pre-trained weights efficiently, significantly reducing memory and compute requirements compared to full fine-tuning. While recent work has shown that LoRA is not strictly equivalent to full fine-tuning (Shuttleworth et al. 2024), it remains an efficient and practical method for adapting models to specific tasks, playing a crucial role in the democratization of AI.

Despite being efficient, LoRA has been shown to be suboptimal for finetuning models with large embedding dimensions, due to differences in the magnitudes of the values of A and B (Hayou, Ghosh, and Yu 2024; Yen et al. 2024; Zhang and Pilanci 2024). This instability forces conservative training strategies, increasing training time, computation, and energy costs. In this paper, we propose a novel approach to stabilize training via a regularization term, followed by evaluation methods to test the effectiveness of our proposed approach.

2 Background

Instead of updating all model parameters during fine-tuning, LoRA injects a pair of trainable low-rank matrices $A \in \mathbb{R}^{r\times d'}$ and $B \in \mathbb{R}^{d\times r}$ into the pre-trained weight matrix $W \in \mathbb{R}^{d\times d'}$, such that the fine-tuned weights can be expressed as

$$W' = W + \Delta W = W + BA. \tag{1}$$

This decomposition constrains the rank of the update ΔW to $r \ll d$, effectively reducing the number of trainable parameters from $\mathcal{O}(d^2)$ to $\mathcal{O}(2dr)$ while maintaining competitive downstream performance. By freezing the original weights W and training only A and B, LoRA also enables efficient adaptation without the need to store full fine-tuned model checkpoints (i.e. only the trained adapters A and B need to be stored).

For the initialization of LoRA, the down-projection matrix B is typically initialized to zero, while the up-projection matrix A is initialized with small Gaussian noise, $A_{ij} \sim \mathcal{N}(0, \sigma_A^2)$ (Hu et al. 2021). This ensures that the initial low-rank update satisfies $BA = 0$, so the pretrained model remains unchanged at the start of fine-tuning. However, this difference in scale causes instability as first-order optimizers using a single learning rate are unable to account for this. Furthermore, the original formulation of LoRA is unable to achieve efficient feature learning in the infinite width limit (Hayou, Ghosh, and Yu 2024).

To improve stability, LoRA+ (Hayou, Ghosh, and Yu 2024) introduces separate, fixed learning rates for the low-rank matrices A and B to address their differing magnitudes during optimization. Namely, the learning rate of A is set to a value smaller than that of B. The authors then only tune the learning rate of A, while keeping their ratio fixed. However, the optimal ratio between B and A may be sensitive to the task or model, potentially making the hyperparameter search more expensive and reducing the practical benefit of LoRA+'s intended simplification.

With similar motivation, SingLoRA (Bensaïd et al. 2025) simplifies the formulation by replacing the two-matrix adapter BA with a single matrix $AA^\top$, halving the number of trainable parameters and removing inter-matrix scaling issues. Although the authors show that this design remains expressive within the attention mechanism, using only one matrix reduces the degrees of freedom compared to standard LoRA, which may limit expressiveness in other layers. Moreover, the reliance on a shared matrix A can make SingLoRA incompatible with LoRA variants that depend on

two distinct adapter matrices.

3 Approach

We propose experimenting with an additional scale regularization penalty that constrains the relative magnitudes of A and B during training. The central hypothesis is that by maintaining this balance, we avoid cases where the optimizer is unable to accommodate both large and small-scale updates effectively.

Specifically, given LoRA's formulation

$$\Delta W = BA, \tag{2}$$

let $\bar{A} = \frac{1}{r \times d'} \sum_{i,j} A_{i,j}^2$ and $\bar{B} = \frac{1}{r \times d} \sum_{i,j} B_{i,j}^2$. We propose to add a regularization term to the loss function of the form

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{scale}}, \tag{3}$$

where

$$\mathcal{L}_{\text{scale}} = \left(\log(\bar{A}) - \log(\bar{B} + \epsilon) \right)^2. \tag{4}$$

This term penalizes large logarithmic differences between the magnitudes of A and B, effectively encouraging them to remain in similar scales throughout training. Note that a small $\epsilon > 0$ is necessary as B is initialized to 0 at the start of training. The penalty introduces a single additional hyperparameter and is compatible with any LoRA variant using a two-matrix factorization.

4 Theoretical Analysis

In this section, we analyze the optimization behavior of low-rank adaptation under a balanced parameterization. Let $Z = BA$ be a LoRA weight decomposition and x be the input embedding . The update to Z at each optimizer step is given by

$$\delta Z = \delta B A + B \delta A + \delta B \delta A,$$

where δA and δB are the updates to A and B respectively.

Note that the $\delta B \delta A$ is negligible as it is dependent on the square of the learning rate. Efficient feature learning requires that the magnitude of δBAx and $B\delta Ax$ be independent of the network width, n, so that training remains stable in the infinite width limit. However, suppose the magnitude of δBAx is dependent on n: $\mathcal{O}(n^\alpha)$, then it can explode if $\alpha > 0$ and diminish if $\alpha < 0$, as $n \to \infty$ (Yen et al. 2024).

Under mild assumptions, we can theoretically show that a balanced parameterization leads to efficient feature learning. The proof is in Appendix A.

Theorem 1. *If $\bar{A} \approx \bar{B}$, then we can have $\|\delta B\|\|A\| = \Theta(1)$ and $\|B\|\|\delta A\| = \Theta(1)$.*

5 Evaluation

The effectiveness of this approach will be evaluated against existing methods (LoRA, LoRA+, SingLoRA, and LoRA Done RITE) on representative downstream tasks, for example instruction-following datasets and selected GLUE benchmarks, with comparisons in convergence speed, task performance, and efficiency. For each task, we will fine-tune the same base model using each method under comparable settings. The evaluation will consider:

1. **Convergence speed:** Track training loss and task-specific validation metrics over time to measure how quickly each method reaches stable performance.
2. **Task performance:** Compare the final performance on each task, after fine-tuning, of each variant.
3. **Stability analysis:** Monitor the Frobenius norms of the low-rank matrices A and B and their ratio over training to determine if the regularization successfully maintains balanced scales.

Various base models with different parameter counts (e.g., 7B, 24B, 70B) will also be tested to investigate the impact of these methods on models of different sizes. The proposed solution will be considered successful if it achieves comparable or superior task performance while enhancing convergence stability and maintaining parameter efficiency.

6 Conclusion

Adapting large language models efficiently and stably remains a critical challenge. We propose adding a scale-regularization penalty to encourage comparable magnitudes between the low-rank matrices A and B, aiming to improve optimization stability and convergence speed. The approach will be evaluated against existing LoRA variants on representative downstream tasks with analyses of performance and stability. In addition to introducing a new method, the project also aims to include a rigorous comparative evaluation of stable LoRA variants, along with the proposed approach. Ultimately, our contribution to PEFT is intended to promote more accessible and resource-efficient AI and AI research, supporting the broader democratization of AI.

A Proof of Theorem 1

If $\bar{A} \approx \bar{B}$, then

$$\begin{aligned}
\Theta(\|A\|) &= \Theta\left(\sqrt{r \times d'} \sqrt{\frac{1}{r \times d'} \sum_{i,j} A_{i,j}^2} \right) \\
&= \Theta\left(\sqrt{r \times d'} \sqrt{\frac{1}{r \times d} \sum_{i,j} B_{i,j}^2} \right) \\
&= \Theta\left(\sqrt{\frac{d'}{d}} \|B\| \right) \\
&= \Theta(\|B\|).
\end{aligned}$$

Similar to Theorem 1 of Yen et al. (2024), let $\|A\| = \Theta(n^a)$, $\|B\| = \Theta(n^b)$, $\|\nabla Z\| = \Theta(n^c)$, $\eta = \Theta(n^d)$, where η is the learning rate and n is the network width. Since $Z = BA$, from chain rule we know $\nabla A = B^\top \nabla Z$ and $\nabla B = \nabla Z A^\top$. Because the update rule is symmetric, we can express the updates as

$$\|\delta A\| = \Theta(n^{xa+yb+zc+d}), \quad \|\delta B\| = \Theta(n^{xb+ya+zc+d}).$$

As we have shown above $\Theta(\|A\|) = \Theta(\|B\|)$, and thus we have $a = b$. This means $\Theta(\|\delta A\|) = \Theta(\|\delta B\|)$, and hence

$$\Theta(\|B\|\|\delta A\|) = \Theta(\|\delta B\|\|A\|).$$

We can now simply choose η such that $\|B\|\|\delta A\|$ and $\|\delta B\|\|A\|$ are $\Theta(1)$.

References

Bensaïd, D.; Rotstein, N.; Velich, R.; Bensaïd, D.; and Kimmel, R. 2025. SingLORA: Low Rank Adaptation Using a Single Matrix. *arXiv preprint arXiv:2507.05566*.

Hayou, S.; Ghosh, N.; and Yu, B. 2024. LORA+: Efficient Low Rank Adaptation of Large Models. *arXiv preprint arXiv:2402.12354*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*.

Shuttleworth, R.; Andreas, J.; Torralba, A.; and Sharma, P. 2024. LoRA vs Full Fine-tuning: An Illusion of Equivalence. *arXiv preprint arXiv:2410.21228*.

Yen, J.; Si, S.; Meng, Z.; Yu, F.; Duvvuri, S. S.; Dhillon, I. S.; Hsieh, C.-J.; and Kumar, S. 2024. LORA Done RITE: Robust Invariant Transformation Equilibration for LORA Optimization. *arXiv preprint arXiv:2410.20625*.

Zhang, F.; and Pilanci, M. 2024. Riemannian Preconditioned LoRA for Fine-Tuning Foundation Models. *arXiv preprint arXiv:2402.02347*.