Asymmetric AdaLora

Kieran Gallagher, Mathew Martin, Natasha Sebastian

[1]New York University Tandon School of Engineering, New York, USA

Abstract

In this work, we analyze the effectiveness of parameter-efficient fine-tuning (PEFT) techniques to build upon Low-Rank Adaptation (LoRA) modules, as well as data augmentation for natural language processing, in the training and validation of the RoBERTa architecture with less than 1 million trainable parameters. Trained and evaluated on the AG News dataset, the model reaches 85.4% accuracy on the testing data.

Public Repository

The python implementation of Asymmetric AdaLoRA can be found here:
github.com/NYUNeuroNinjas/AsymmetricAdaLora

Introduction

The RoBERTa model, proposed by Liu et al.[2], builds on Google's BERT model released in 2018 by making training less computationally expensive through key hyperparameter changes. By reducing pre-training and increasing mini-batch size and learning rates, RoBERTa is able to match the performance of BERT and its successors at a fraction of the computational cost. This focus on efficiency highlights the need for similar cost-reducing strategies when fine-tuning a large-language model to a specific downstream task. In this paper, we will build on these ideas and propose novel modifications for LoRA to maximize RoBERTa's accuracy while keeping trainable parameters under 1 million.

Data

The AG News Dataset, comprised of 120,000 news article titles and descriptions, evenly representing 4 classes, was used for training of our network architecture. Validation was done on the 7,600 test samples in the dataset. Data augmentations were applied to only the training set in order to improve model generalization.

Methodology

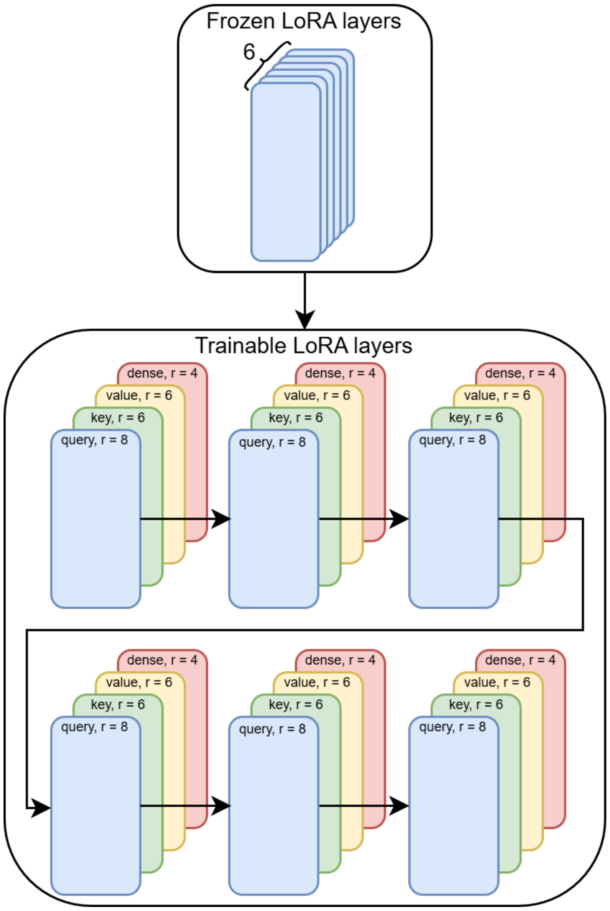

Figure 1: Asymmetric AdaLoRA Architecture

Final Training Setup:

- **Layers:** By only unfreezing LoRA parameters for the last 6 layers, we are able to reduce overfitting and constrain parameter count effectively.
- **Optimizer:** AdamW with Beta1 of 0.9, Beta2 of 0.999, and weight decay of 0.05.

- **Data Augmentation:** Synonym augmentation of training set concatenated to the current training set. Between 3 and 5 words replaced with a synonym from the nlpaug library[3] in each training set sentence.
- **Regularization:** Dropout with a probability of 0.1 was used to combat overfitting.
- **Learning Rate and Schedule:** Started with a Learning Rate of 0.0002 and used a Cosine Annealing scheduler.
- **Batch Size and Epochs:** A batch size of 32 and trained for up to 6 epochs.

AdaLoRA

Adaptive LoRA (AdaLoRA) [4] is a recent advancement in parameter-efficient fine-tuning that extends the standard Low-Rank Adaptation (LoRA) method by introducing a dynamic and adaptive approach to rank allocation during training. Unlike traditional LoRA, which statically assigns a fixed low-rank configuration to all layers or modules in a model, AdaLoRA begins with a generous budget of low-rank updates, allowing the model to explore a wide solution space during the early stages of fine-tuning. As training progresses and the model begins to converge, AdaLoRA gradually reduces this budget, trimming away redundant or less impactful parameters. This pruning process is not arbitrary—at regular intervals, AdaLoRA leverages gradient-based signals to evaluate the importance of each adapter and dynamically adjusts its rank, reallocating capacity toward the most critical components of the model. This enables the model to retain high representational power where needed while reducing unnecessary parameter overhead elsewhere.

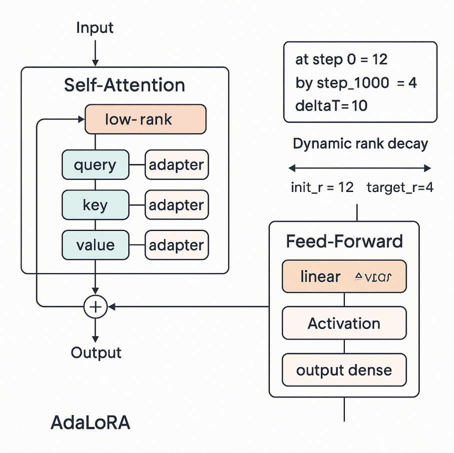

Figure 2: Adaptive LoRA configuration for isolated testing

By focusing this evolving adaptation process primarily on the Transformer architecture's core attention matrices—such as query, key, and value projections—AdaLoRA captures rich, task-specific patterns that are crucial for downstream performance. This approach not only maintains high accuracy but also ensures that the total number of trainable parameters remains well within strict budget constraints, making it particularly suitable for fine-tuning large language models in resource-constrained environments. Empirical evaluations across multiple tasks demonstrate that AdaLoRA consistently outperforms traditional LoRA and other low-rank tuning baselines, especially in scenarios where parameter efficiency is critical. The method achieves a strong balance between flexibility in early learning and focused adaptation during convergence, offering a principled way to optimize parameter allocation in large-scale model adaptation workflows

Extending LoRA to the feed-forward sub-blocks

Each transformer block has an MLP (feed-forward network) that projects the hidden state up to a higher dimension, applies a non-linearity, then projects down again, While attention captures token-token interactions, the FFN learns richer per-token transformations. By adding LoRA adapters [1] to these ”dense” layers, we give our adapter layers direct control over the MLP computations, which inturn boosts classification accuracy on text tasks.

AsymmetricLora

We introduce AsymmetricLora, a methodology in which separate rank values are assigned to each of the LoRA layers: ”query”, ”key”, ”value”, and ”dense”. Doing so allows the inclusion of LoRA techniques on the dense layers discussed above without exceeding our limit on trainable parameters.

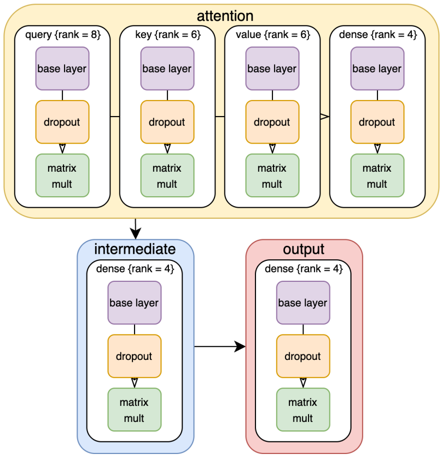

Figure 3: Asymmetric LoRA configuration for a single LoRA layer

We force the Hugging Face PEFT model to reconfigure layer ranks by resizing the LoRA A, B, and E matrices prior to training. This limits the possible parameters and allows us higher rank values for query, key, and value while maintaining a small LoRA matrix for the feed-forward dense blocks.

Synonym Augmentations

We use nlpaug's [3] synonym data augmentation technique to substitute words in a text with their synonyms to generate similar sentences. This increases the diversity of the training data, which helps generalize and reduce model overfitting. The SynonymAug class uses WordNet to identify and replace words with their synonyms. This allows us to control the number of words to substitute and the number of augmented outputs generated, which helps in expanding the datasets and improving the model's robustness.

Results and Analysis

The capabilities of the Asymmetric AdaLora network were evaluated on the aforementioned AG News dataset, with consideration given to various configurations of methodologies, augmentations, and rank configurations.

Model Desc.	**LoRA Layers**	**Param.**	**Val Acc.**	**Test Acc.**
LoRA	query, value	999,172	94.7%	83.9%
Asymmetric LoRA	query, key, value, dense	947,712	95.1%	84.2%
AdaLoRA	query, key, value	925, 660	93.4%	84.4%
Asymmetric AdaLoRA	query, key, value, dense	999,364	94.8%	85.4%

The initial LoRA configuration, with 999k parameters, achieved a validation accuracy of 94.7%, but a test accuracy of only 83.9%. This model was improved by augmenting the training data, which improved the relationship between validation and testing accuracy in all future models. Adding LoRA to the ”value” and ”dense” layers, and the introduction of Asymmetric LoRA to keep the parameter count constrained, improved both validation and test accuracy slightly. Separate tests isolating the effect of AdaLoRA (our official kaggle submission) returned similar results, with slightly higher increases despite the omission of extending LoRA to the feed-forward sub-blocks.

Asymmetric AdaLoRA, containing 999,364 trainable parameters, was made using a combination of our previous methods. Inferences made by model resulted in a 1% increase in the test accuracy to 85. 4%, our highest result. The beneficial aspects of both intermediate models helped decrease loss in the final model and produce accurate inferences on the test dataset.

These results demonstrate that the proposed RoBERTa architecture with LoRA fine-tuning is suitable for making real-world inferences on specific language tasks where computational power is limited. By maintaining a low parameter count and dynamic configuration, Asymmetric AdaLoRA has the capability of quickly adapting to a wide variety of datasets.

References

[1] 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:2106.09685.

[2] Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692.

[3] Ma, E. 2019. NLP Augmentation. https://github.com/makcedward/nlpaug.

[4] Zhang, Q.; Chen, M.; Bukharin, A.; Karampatziakis, N.; He, P.; Cheng, Y.; Chen, W.; and Zhao, T. 2023. AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning. arXiv:2303.10512.