

Research on Efficient Fine-tuning of Large Language Model Parameters for Text Classification: A Review from Gradient Descent Basics to LoRA/QLoRA Methods

Tianyi Qu

*School of Computer Science and Communication Engineering, Jiangsu University, Jiangsu, China
t3194280@gmail.com*

Abstract. Large Language Models (LLMs) perform well in many text classification tasks, such as sentiment analysis, spam detection, and topic classification. However, adapting large language models to downstream tasks via full-parameter fine-tuning often requires significant computing power and GPU memory, resulting in high training costs, which is impractical for course assignments or resource-constrained scenarios. To address this issue, Parameter Efficient Fine-Tuning (PEFT) has gradually become a popular approach. Its idea is to freeze most model parameters and train only a small number of additional parameters, thereby reducing training costs. This paper takes text classification as an example and begins with an introduction to fundamental concepts such as stochastic gradient descent, cross-entropy loss, and confusion matrices. It first reviews the technological evolution from traditional NLP models (such as bag-of-words, Naive Bayes, and N-grams) to word embeddings, Transformers, and ultimately large language models. Then, it focuses on the low-rank adaptation concept of LoRA and the 4-bit quantization training mechanism of QLoRA, analyzing their characteristics from the perspectives of parameter count, GPU memory consumption, and training stability. This paper also discusses key representation-level issues such as word segmentation methods (BPE, WordPiece, SentencePiece), Type/Token differentiation, vocabulary size, and sequence length, explaining how these factors affect the model's generalization ability. Overall, this paper aims to answer the question of "how to adapt large models to text classification tasks at a lower cost under limited resources," and provides a clearer theoretical framework and methodological approach for course learning and introductory research.

Keywords: large language model, efficient parameter fine-tuning, LoRA, QLoRA

1. Introduction

In recent years, Large Language Models (LLMs) have made significant progress in text understanding and generation tasks. When applied to text classification scenarios such as sentiment analysis, spam detection, and topic classification, they often achieve better results than traditional models. Supporting these capabilities are not only deeper network structures, but also optimization methods such as stochastic gradient descent, cross-entropy loss functions, and standard evaluation

metrics such as confusion matrices, accuracy, precision, recall, and F1 scores. These fundamental concepts remain key to understanding the training process of modern models. Meanwhile, the cost of full-parameter fine-tuning of LLMs is extremely high. The constraints of memory consumption, training time, and dataset scale make it difficult for many course projects or student researchers to implement such fine-tuning directly. Therefore, Parameter Efficient Fine-Tuning (PEFT) has begun to attract attention. It reduces adaptation costs by freezing most parameters while only training a small number of newly introduced parameters.

This paper focuses on text classification as the core task. Based on a review of the evolution of traditional NLP methods and representation learning, this paper focuses on summarizing PEFT techniques such as LoRA and QLoRA, and constructs connections between related concepts and course content (tokenization, vocabulary, cross-entropy, generalization ability, etc.). This paper aims to answer the question of "how to adapt large language models to complete text classification tasks at an affordable cost under limited resources", and to provide a clear theoretical framework and methodological reference for teaching practice and introductory research.

2. Analysis

2.1. Gradient descent and the foundations of text classification

Text classification can be summarized as two steps: "representation $\rightarrow$ classification". First, the text is converted into a computable representation, and then the representation is mapped to the category label. Training involves iteratively updating parameters on labeled data so that the model can make stable predictions on unseen samples. In terms of implementation, mini-batch stochastic gradient descent (SGD) is often used to approximate the full gradient to reduce computational overhead. Multi-class classification typically uses cross-entropy as the loss function to maximize the probability of the correct class. For both full-parameter fine-tuning and LoRA/QLoRA, the optimization objective remains consistent, with the main difference lying in the set of updatable parameters. For evaluation, when class distributions are imbalanced, relying solely on accuracy can be misleading. Evaluation should instead incorporate the confusion matrix (TP/FP/TN/FN), precision, recall, and F1-score, with the key metric chosen based on the task's error cost. Multi-class classification also needs to distinguish between micro-F1 (more biased towards the large category) and macro-F1 (more focused on the average of the small categories) [1]. In terms of model form, the common approach is "text representation (TF-IDF/word vectors/Transformer [CLS]) + linear layer + Softmax", trained with cross-entropy. In Transformer/LLM settings, updating all parameters incurs high computational cost, so PEFT is adopted to reduce the burden of fine-tuning and deployment.

2.2. From classical NLP to large language models

From statistical methods such as bag-of-words + Naive Bayes, to N-gram language models, word embeddings, Transformers, and finally large language models, this evolution can be summarized in three points: representation has moved from sparse counting to dense and contextualized representation; training has shifted from relying on labeling tasks to self-supervised pre-training such as "predicting the next word" [2]; when the model size increases to the billion level, the bottleneck changes from "whether it can be done" to "whether it can be adapted and reused at low cost". Therefore, the mainstream paradigm of text classification has changed from training from zero to "pre-trained model + downstream adaptation", and the core goal of PEFT is to update only a small number of parameters while preserving the pre-training capability as much as possible, so as to

reduce the cost of graphics memory, computation and storage, and avoid maintaining a full model for each task.

2.3. Word segmentation, representation, and generalization

2.3.1. Why does word segmentation affect classification

BPE, WordPiece, and SentencePiece essentially represent a trade-off between vocabulary size and out-of-vocabulary (OOV) rate [3-5]. Their impact on classification stems primarily from three aspects: finer tokenization leads to longer sequences, increasing training and inference costs; subword unit combinations determine the expressiveness for domain terms and new words; and with a fixed maximum sequence length, longer inputs raise the risk of information loss due to truncation.

Therefore, tokenization should be considered part of the model specification.

2.3.2. Vocabulary size, sequence length, and generalization

Type: Number of distinct units in the vocabulary; Token: Total number of occurrences in the corpus.

A larger vocabulary yields shorter sequences but introduces more parameters, and infrequent tokens in the long tail are harder to learn; A smaller vocabulary enables greater parameter sharing but produces longer sequences, making truncation more probable.

Transformer computation grows approximately $O(L^2)$ with length L , therefore word segmentation and max length simultaneously affect GPU memory/throughput, optimization stability, and information fidelity, ultimately impacting generalization; long texts often require segmentation/sliding windowing before aggregation.

2.4. A review of Parameter Efficient Fine-Tuning (PEFT) methods

PEFT replaces full weight updates for model adaptation by freezing the base model and training only a small set of incremental parameters to accomplish task transfer, thereby simultaneously reducing memory and computation overhead from full-parameter fine-tuning, as well as the management burden of storing full model copies for multiple tasks, and the problem of overfitting and impairing general capabilities in small data or noisy scenarios.

Among the representative methods, LoRA freezes W , learns the low-rank increment $\Delta W = BA$, and only updates A and B , achieving a near-full parameter fine-tuning effect with fewer trainable parameters, and supports merging weights during deployment and more favorable inference latency [6]; when the task and pre-trained distribution differ too much, it may be limited by low-rank capacity. QLoRA performs LoRA training on a 4-bit quantized base model, enabling fine-tuning of larger models with a lower memory footprint [7], making it suitable for memory-constrained scenarios. However, quantization may reduce stability on boundary samples, the sensitivity to the reproduction of implementation and hardware details, and the training instability caused by insufficient hyperparameter search when resources are limited. Adapters, prefix tuning, and prompt tuning achieve lightweight adaptation by inserting small modules or learning task-specific prefixes or soft prompts:: the former provides clearer task isolation but a heavier inference path, while the latter is more dependent on model size and task representation [8-10].

For method selection, LoRA is generally the preferred balanced choice between performance and deployment efficiency; when truly limited by GPU memory and model size thresholds, QLoRA has higher priority, but it should not be used by default in all scenarios.

3. Applications and challenges of LoRA/QLoRA in text classification

3.1. Typical process: discriminative vs. generative, choosing between the two approaches

Two main paradigms are commonly used for text classification with large language models:

Discriminative: Take a sentence vector (e.g., [CLS]) and attach it to a classification head, then train using cross-entropy.

Generative: Have the model generate category names/labels, then map the generated results to categories.

For undergraduate teaching, the discriminative paradigm is easier to explain, as evaluation is straightforward and error patterns are more interpretable; the generative paradigm is more aligned with native LLM behavior, but introduces additional variables such as output format constraints and label name ambiguity. Prefix tuning and prompt tuning are more often used with generative setups, whereas LoRA/QLoRA support both paradigms, with the discriminative setup being more intuitive [9, 10].

3.2. The relationship between model size, data size, and results: the "implicit conditions" behind publicly published conclusions

Existing literature often claims that PEFT can achieve performance close to full-parameter fine-tuning with far fewer trainable parameters. However, this phenomenon usually depends on several implicit conditions:

Whether the task distribution is close to the pre-training distribution: the closer they are, the easier lightweight adaptation becomes.

Whether the data is clean: the greater the noise, the more likely the lightweight parameters may be biased.

Whether the evaluation focuses on the long tail: when only looking at the total score, many problems will be masked [1].

Therefore, this paper emphasizes that performance evaluation must be analyzed in conjunction with task difficulty factors (long texts, long-tailed categories, domain terminology, and structured label systems).

3.3. Feasibility and risks in resource-constrained scenarios

QLoRA is highly suitable for resource-constrained environments, yet it carries several notable risks and limitations: The boundary sample instability caused by quantization [7], the reproduction depends on the implementation details (different code libraries of the same paper may be very different), and the training strategy is forced to compromise: to conserve memory, sequence length and batch size are often reduced, which in turn degrades performance. These issues arise from the combined effects of quantization, training mechanisms, and the inherent characteristics of text classification tasks.

3.4. Consensus and disagreements regarding publicly published research conclusions

Based on findings from papers on LoRA, QLoRA, prompt tuning, prefix tuning, and adapters, at least three key research gaps can be identified:

Lack of a decision framework for text classification: Many papers primarily focus on generation tasks or comprehensive benchmarks, failing to systematically incorporate the specific characteristics

of classification tasks (long tail, strong labeling systems, evaluation metrics) into the method selection logic.

Method comparisons are often "non-reusable": Different prompts, preprocessing methods, and evaluation criteria for the same task can lead to different conclusions, making it difficult for readers to reproduce and transfer the results to their own tasks.

Insufficient discussion of failure cases: Papers tend to showcase the average scores of successful cases, but there is little analysis of "which types of text are consistently misclassified," which is precisely what classification applications are most concerned with.

4. Discussion

The limitations of this paper are also clear: First, this work is a review and does not include empirical experiments on specific datasets or real-world scenarios. Thus, certain claims regarding stability risks under resource constraints and the heightened sensitivity of boundary samples to quantization errors are based primarily on theoretical derivations and literature reports, without direct verification under a unified experimental setup. Accordingly, the conclusions of this paper are more suitable for providing methodological guidance and risk awareness than for drawing definitive quantitative comparisons between methods.

Second, the depth of analysis varies across PEFT methods: LoRA and QLoRA are discussed in greater detail because they directly address large model trainability; in contrast, Adapters, PrefixTuning, and Prompt Tuning are presented more as representative approaches, without conducting a comparative analysis at the same granularity on "key variables in classification task implementation" (such as inference latency, output constraints, and template sensitivity).

Finally, the presentation of "failure case typology" in this paper is still too general. While the article touches upon common sources of error in text classification practice (such as loss of evidence due to long text truncation, insufficient long-tail category samples, and unclear label definition boundaries), it has not yet developed a reusable analysis template (such as an error analysis checklist for binning by text length, clustering by category confusion pairs, and grouping by linguistic phenomena). This limits the scope of this article from a "method review" to a "reproducible evaluation guide."

5. Conclusion

Existing PEFT research often relies on general NLG tasks or comprehensive benchmarks to demonstrate effectiveness, but key factors such as class imbalance, label structure, long-tailed categories, and text length distribution are often underemphasized in text classification settings. Future research should establish a classification-oriented selection framework: using dataset scale, long-tailedness, average length/truncation ratio, label naturalness, and deployment latency constraints as dimensions, systematic comparisons can be conducted among methods such as LoRA, adapters, prefix tuning, and prompt tuning. And macro-F1 and category-level metrics could be reported under a unified training budget (fixed GPU memory, fixed training steps, fixed hyperparameter search times), forming a reproducible "classification-oriented PEFT benchmark" [6, 8-10].

QLoRA significantly lowers the training threshold, but quantization errors, implementation details, and hardware differences may lead to training instability or reproducibility fluctuations. Future research needs to address both the "method mechanism + evaluation protocol" aspects: Methodologically, further analysis of the impact of NF4, dual quantization, and memory

management strategies on classification boundary samples is required; in terms of evaluation, reports on the variance of multiple random seeds, multiple hardware/implementations are required, along with robustness tests on boundary samples and distribution drift (rewriting, noise, domain migration), avoiding conclusions based solely on the highest single score [7]. Long text classification is often limited by the maximum input length and the computational overhead of secondary attention, especially in resource-constrained environments. Future directions include: combining PEFT to research more systematic long text processing schemes, such as segmented encoding + segment-level aggregation (mean/max/attention pooling), retrieval-based segment selection, and introducing lightweight hierarchical structures while maintaining parameter efficiency. Research should not only focus on overall F1 improvement, but also analyze whether the systemic errors caused by "key information being truncated" have been mitigated, and report stratification indicators for different length ranges.

References

- [1] Marina Sokolova, Guy Lapalme. A systematic analysis of performance measures for classification tasks . Information Processing & Management, Vol.45(4). Publisher: Elsevier (Pergamon). 200907. 427–437. 427–437
- [2] Stanley F. Chen, Joshua Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. Computer Speech & Language, Vol.13(4). Publisher: Academic Press (Elsevier). 19991001. 359–394. 359–394
- [3] Rico Sennrich, Barry Haddow, Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units . In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016). Publisher: Association for Computational Linguistics. 201608. 1715–1725. 1715–1725
- [4] Mike Schuster, Kaisuke Nakajima. Japanese and Korean voice search . In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012). Publisher: IEEE. 201203. 5149–5152.
- [5] Taku Kudo, John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing . In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2018 Demo). Publisher: Association for Computational Linguistics. 201811. 66–71.
- [6] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan AllenZhu, Yanzhi Li, Shean Wang, Lu Wang, Weizhu Chen. LoRA: LowRank Adaptation of Large Language Models . arXiv 20211016. PDF 1–26.
- [7] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, Luke Zettlemoyer. QLoRA: Efficient Finetuning of Quantized LLMs. arXiv. 20230523. PDF 1-26.
- [8] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, Sylvain Gelly. ParameterEfficient Transfer Learning for NLP . In: Proceedings of the 36th International Conference on Machine Learning (ICML 2019). PMLR, Vol.97. 201906. 2790–2799 (PDF 1–10) .
- [9] Xiang Lisa Li, Percy Liang. PrefixTuning: 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 (ACL/IJCNLP 2021). Publisher: Association for Computational Linguistics. 202108. 4582–4597.
- [10] Brian Lester, Rami AlRfou, Noah Constant. The Power of Scale for ParameterEfficient Prompt Tuning . In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021). Publisher: Association for Computational Linguistics. 202111. 3045–3059.