Bibliography on LLMs Fine-Tuning

Lucas Loustalet

November 2024

Abstract

This bibliography presents important research on fine-tuning large language models (LLMs). It includes key papers like Attention Is All You Need, which introduced the transformer architecture, and BERT, which showed the power of pre-trained models fine-tuned for specific tasks. Other important works include LoRA (Low-Rank Adaptation) and Adaptative Layers, which explore efficient fine-tuning methods (PEFT). The bibliography also covers optimization techniques such as Bayesian optimization and Hyperband for tuning hyperparameters. Practical applications are discussed, which shows how fine-tuning can be used for various natural language processing tasks (NLP). Finally, ethical considerations, like addressing biases in models, are highlighted. This collection of research provides a solid foundation for anyone studying or working with fine-tuning in LLMs.

1 Introduction

LLM fine-tuning is a practice in its own right in the field of NLP. These models, built on the Transformer architecture introduced in the 2017 article Attention Is All You Need [7], have revolutionized the field by enabling the efficient processing of complex tasks thanks to their ability to learn by amassing huge amounts of data. However, their performance on specific tasks or contexts remains limited when these models are not adapted via a specialization stage.

Fine-tuning allows pre-trained models to be used on general data, such as the whole of Wikipedia for GPT, and their parameters adjusted to train them on domain-specific data. This technique maximizes the potential of LLMs while limiting training costs, since part of the model is already trained. With the advent of methods such as Bayesian optimization, Hyperband and approaches such as LoRA, fine-tuning has become more efficient, both in terms of resources and results.

This paper is divided into several sections: we'll start with the basics of fine-tuning, exploring the Transformer architecture, before detailing the fundamental principles of fine-tuning. We then examine hyperparameter optimization strategies. Concrete use cases will illustrate the practical impact of these techniques, before addressing limitations. The aim of this bibliography is to provide an overview of current techniques and the challenges associated with fine-tuning LLMs.

2 Foundations of fine-tuning

The Transformer research paper published in 2017 laid the foundations for the modern architectures used in LLMs [7]. These architectures are based on two main modules: the encoder and the decoder. The encoder transforms the input into a richer representation thanks to the key mechanism of transformers: attention. This mechanism captures the dependencies between words in a sentence, regardless of their distance, which is essential for processing complex sequences.

Attention heads parallelize learning by allowing the model to focus simultaneously on different parts of the input sequence. The outputs of the different heads are then concatenated to create a complete representation of the sentence. This ability to process in a parallel and distributed way is one of the reasons why LLMs are able to handle large quantities of data efficiently.

Once the encoder has generated this representation, it is passed to the decoder, which is responsible for text generation. The decoder, like the encoder, uses the attention mechanism to perform the generation task, but with an important distinction. On the one hand, it uses self-attention to focus only on words already generated and avoid accessing future words through a causal attention mask. On the other hand, in encoder-decoder models (such as T5), the decoder also uses cross-attention, which allows the model to rely on the encoder to adjust generation according to the global context.

This flexible architecture is the reason why pre-trained models based on transformers can be easily adapted to many specific tasks through the fine-tuning process. By fine-tuning a previously trained model on a large amount of general data, it is possible to adapt the model to a specific task, e.g. translation, text generation, or even question answering, by adjusting only certain parts of the model (often the top layers) so that the model learns features specific to the new task. This ability to adjust the model in a targeted and efficient way makes transformers an ideal basis for fine-tuning LLMs.

It's important to note that transformers are not limited to NLP. For example, architectures such as DALL-E use transformers to generate images from text descriptions. We will now explore the fundamentals of fine-tuning and how this process is used to maximize the efficiency of LLMs in specific domains.

3 Fine-Tuning Principles

Fine-tuning is the process of specializing a pre-trained LLM for a specific task by adjusting its weights. This process can be adapted based on the complexity of the task, the resources available, and the desired efficiency. More generally, fine-tuning approaches fall into two categories: full fine-tuning and partial fine-tuning [5]. For this section, we will assume you are already familiar with the model training process and will skip details regarding all the intermediary steps, such as data preparation and preprocessing

3.1 Full Fine-Tuning

Full fine-tuning involves readjusting all the model parameters. Unlike training a model from scratch, this method leverages pre-trained weights that provide a robust baseline understanding of language. Full fine-tuning is especially suitable for:

- Tasks requiring high precision
- Applications that significantly differ from the pre-training use case (e.g., medical or legal domains)

However, this approach is computationally expensive and prone to overfitting when training data is limited. For instance, Rolland et al. (2024) [6] demonstrated that full fine-tuning on low-resource datasets, such as for children's automatic speech recognition, often leads to performance degradation. Specifically, fine-tuning only the encoder of the transformer yielded a WER reduction from 25.04% to 12.20%, while simultaneously reducing the number of trainable parameters from 71.5M to just 25.2M, showcasing the efficiency of targeted adaptations in both performance and parameter efficiency.

3.2 Partial Fine-Tuning

Partial fine-tuning restricts adjustments to a subset of the model parameters, often by freezing the lower layers and re-training only the upper layers or specialized modules. This approach is preferred in scenarios where:

- Training data is limited
- Pre-trained representations are already well-suited to the target task
- Computational resources need to be conserved

A classic example of partial fine-tuning is text classification with BERT. In such cases, pre-trained BERT representations are used, and only a lightweight classifier is added and fine-tuned on a specific dataset. This reduces computational costs and avoids overfitting.

Building on the principles of partial fine-tuning, modern approaches have emerged that further optimize the process by reducing the number of parameters adjusted. These methods refine the concept of partial fine-tuning to achieve even greater efficiency and modularity.

3.3 Modern Approaches to Fine-Tuning

In recent years, parameter-efficient fine-tuning (PEFT) methods [1], such as LoRA (Low-Rank Adaptation) [3] and Adapter Layers [2], have emerged as effective alternatives to traditional fine-tuning approaches. These techniques optimize the fine-tuning process by significantly reducing computational and memory requirements, making them particularly advantageous for resource-constrained environments.

3.3.1 LoRA: Low-Rank Adaptation

LoRA is a PEFT method that decomposes the weight adjustments (ΔW) into two low-rank matrices, A and B, such that:

$$\Delta W = AB$$

Here, A and B are much smaller matrices, with dimensions $r \times d$ and $d \times r$, respectively, where $r \ll d$. By updating only these matrices during fine-tuning, LoRA drastically reduces computational overhead while maintaining performance. This approach is particularly effective for very large models, where updating all parameters would be infeasible.

3.3.2 Adapter Layers

Adapter Layers are another PEFT technique widely adopted for task-specific model adaptation. These modules are inserted between the pre-trained layers of an LLM, allowing for localized adjustments without modifying the main model weights. Each adapter is independently trained, enabling efficient fine-tuning for diverse tasks while preserving the integrity of the base model.

For example, the adapters act as bottlenecks within the model layers, capturing task-specific information while retaining general knowledge from the pre-trained model. This modularity minimizes memory usage and computational demands, making adapters ideal for multi-task scenarios.

4 Hyperparameter Optimization

Having discussed parameter optimization in the previous section, we now focus on hyperparameter selection. There are several methods for determining the optimal hyperparameters (HPO) for a model, particularly for fine-tuning LLMs. Below, we explore two main approaches: black-box optimization and multi-fidelity optimization.

4.1 Black-Box Optimization Methods

Black-box optimization methods aim to model an unknown function [8]. The most basic method is grid search. Introduced in the 1990s, this approach involves testing all possible combinations of a specified set of hyperparameters. Although simple to implement, this method is inefficient for large search spaces, as is often the case with LLMs.

Random search, the successor to grid search, randomly selects the values of the hyperparameters to be tested. Although this method covers a larger search space and avoids blocking in poorly performing areas, it is less effective for optimizing large models such as LLMs.

To overcome these limitations, Bayesian optimization (BO) emerged. Introduced in 2012, this approach is distinguished by its ability to dynamically adjust hyperparameters based on previous results. Rather than testing all combinations, OB uses a probabilistic model, usually a Gaussian process, to estimate the values of hyperparameters that have not yet been tested. This method is particularly useful for optimizing expensive functions, such as those associated with LLM fine-tuning. OB uses another function, the acquisition function, to determine the parameter configurations to be tested, balancing exploration and exploitation. However, although OB is more efficient than naive methods, it remains resource-intensive.

4.2 Multi-Fidelity Optimization Methods

A faster alternative to OB is Hyperband, which is part of the family of multi-fidelity approaches, and inspired by bandit algorithms [4]. Unlike black-box methods, Hyperband relies on progressive resource allocation via the Successive Halving method. It distributes a global budget among numerous initial configurations and progressively concentrates these resources on the most promising configurations. This approach makes it possible to explore a large space with lower costs, although it does not capture the fine interactions between hyperparameters, making OB more suitable for targeted searches requiring high precision.

So, for fine-tuning LLMs, Bayesian optimization offers significant advantages, not least because of its ability to adjust hyperparameters based on previous evaluations, while offering a better compromise between exploration and exploitation. However, if execution time and resource utilization are important criteria, then Hyperband is a better choice, although it offers less optimized results.

5 Fine-Tuning Use Cases

Choosing the right model heavily depends on the target task and the type of data available. Model architectures, such as encoders, decoders, or hybrid models (encoder-decoder), directly influence their performance on specific tasks. Here are some common use cases:

1. Text comprehension tasks:
    - These tasks include text classification, sentiment analysis, or named entity recognition.
    - Suitable models: Encoder-only models, such as BERT, are particularly effective for these tasks. Their self-attention mechanism allows them to capture the global context of a sentence or an entire document.
    - Example: Fine-tuning BERT on a dataset for email classification to detect spam.
2. Text generation tasks:
    - These involve producing text from an input, such as summarizing documents, completing documents, or writing content.
    - Suitable models: Decoders like GPT are excellent for these tasks due to their ability to generate text based on previously generated sequences.
    - Example: Using GPT to generate responses in a chatbot or to automatically write articles.
3. Text transformation tasks:
    - These tasks, like translation or summarization, require a complete understanding of the input and the generation of a coherent output.
    - Suitable models: Encoder-decoder architectures, like T5 or BART. The encoder analyzes the input to extract a rich representation, and the decoder generates the output.
    - Example: Fine-tuning T5 for multilingual translation or automatic summarization of scientific articles.

6 Limitations

Although fine-tuning LLMs has led to major advances in NLP, several limitations remain. These challenges concern technical, ethical, and practical aspects, requiring careful attention to ensure responsible adoption and usage.

6.1 Technical Limitations

- **Resource consumption**: LLM models require significant amounts of memory, computing power, and energy, even with PEFT techniques like LoRA. This limits their accessibility for organizations with limited resources.
- **Interpretability challenges**: LLMs function as *black boxes*, making it difficult to interpret their decisions, especially in applications requiring transparent explanations such as in scientific or financial fields.

6.2 Ethical Limitations

- **Bias propagation**: Fine-tuned models can reinforce social or cultural prejudices if the fine-tuning data is not carefully selected.
- **Environmental impact**: Training and fine-tuning LLMs have a high carbon footprint, raising concerns about the ecological sustainability of these technologies.

7 Conclusion

Fine-tuning LLMs is a crucial step in adapting pre-trained models to specific tasks. In this paper, we explored the foundations of this process, ranging from Transformer architectures to modern techniques such as LoRA and Hyperband, which make this adaptation more efficient and accessible.

Our analysis shows that fine-tuning, whether full or partial, allows us to leverage the capabilities of LLMs while reducing resource costs when the appropriate methods are used. Approaches such as partial fine-tuning or PEFT methods illustrate this trend of seeking a balance between performance and efficiency.

While challenges such as resource demands, model interpretability, and biases in training data remain, the rapid progress in the development of new models and techniques offers promising solutions. With continued advancements in both technology and methodology, LLMs and their fine-tuning will continue to evolve, unlocking even more powerful and accessible tools for a wide range of applications.

References

[1] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

[2] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

[3] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[4] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018.

[5] Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. *arXiv preprint arXiv:2408.13296*, 2024.

[6] Thomas Rolland and Alberto Abad. Introduction to partial fine-tuning: A comprehensive evaluation of end-to-end children's automatic speech recognition adaptation. *Procs. of Interspeech, Kos Island, Greece*, pages 5178–5182, 2024.

[7] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[8] Tong Yu and Hong Zhu. Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*, 2020.