

A Comparative Study between Full-Parameter and LoRA-based Fine-Tuning on Chinese Instruction Data for Instruction Following Large Language Model

Xianghui Sun, Yunjie Ji, Baochang Ma*, Xiangang Li

Beike Inc., Beijing, China

{sunxianghui002, jiyunjie001, mabaochang001, lixiangang002}@ke.com

Abstract

Recently, the instruction-tuning of large language models is a crucial area of research in the field of natural language processing. Due to resource and cost limitations, several researchers have employed parameter-efficient tuning techniques, such as LoRA, for instruction tuning, and have obtained encouraging results. In comparison to full-parameter fine-tuning, LoRA-based tuning demonstrates salient benefits in terms of training costs. In this study, we undertook experimental comparisons between full-parameter fine-tuning and LoRA-based tuning methods, utilizing LLaMA as the base model.

The experimental results show that the selection of the foundational model, training dataset scale, learnable parameter quantity, and model training cost are all important factors. We hope that the experimental conclusions of this paper can provide inspiration for training large language models, especially in the field of Chinese, and help researchers find a better trade-off strategy between training cost and model performance.

To facilitate the reproduction of the paper's results, the dataset, model and code will be released.¹

1 Introduction

The advent of language models such as ChatGPT(OpenAI, 2023a) and GPT-4(OpenAI, 2023b), which exhibit human-like understanding and generation capabilities across various domains, has highlighted the importance

of instruction tuning in enabling these models to better comprehend human instructions. Currently, there exist several open-source, large language models that have been fine-tuned on instructional data, including OPT(Zhang et al., 2022), BLOOM(Workshop et al., 2022), LLaMA(Touvron et al., 2023), and GLM(Zeng et al., 2023). These models have demonstrated exceptional performance on a range of language tasks, thereby underscoring the potential benefits of instruction tuning in enhancing language model performance.

In the field of model training, two widely used methods are full-parameter fine-tuning and parameter-efficient tuning. Recently, researchers have conducted extensive experiments to compare the effectiveness of various parameter-efficient tuning methods such as Adapters (Houlsby et al., 2019; Lin et al., 2020), LoRA (Hu et al., 2022), and P-tuning (Li and Liang, 2021; Lester et al., 2021; Liu et al., 2021) against full-parameter fine-tuning (Ding et al., 2023). The results of these experiments demonstrate that LoRA is a promising parameter-efficient tuning method and has been applied in many studies to fine-tune large language models with significant success (Stanford, 2023; Xu et al., 2023).

However, the effectiveness and efficiency of LoRA for finetuning a instruction-following model have not been well explored. In this paper, we examined the influence of two factors: base model and training data scale. Besides, we also compared LoRA with full-parameter finetuning from the perspective of model performance and training efficiency. We assessed these models on a evaluation set consisting of 1,000 samples, spanning across 9 real-word use cases. Finally we obtained the following important experimental results:

- The choice of the base model has a significant

*Corresponding author

¹<https://github.com/LianjiaTech/BELLE>

impact on the effectiveness of LoRA-based tuning.

- Increasing the amount of training data can continuously improve the model’s effectiveness
- LoRA-based tuning benefits from the number of model parameters

We hope that the experimental conclusions of this paper can provide inspiration for training large language models, especially in the field of Chinese, and help researchers find a better trade-off strategy between training cost and model performance.

2 Related Work

2.1 Instruction tuning

Recent studies (Chowdhery et al., 2022; Zhang et al., 2022) have found that by fine-tuning models on datasets with human-annotated prompts, known as instruction-tuning, models can execute new tasks by understanding task instructions, thereby improving their zero-shot and few-shot generalization abilities on unseen tasks. Early research focused on instruction tuning a general NLP task solver, and there is a trend towards converting more and more NLP datasets into a unified dataset and then conducting multi-task training (Xu et al., 2022; Xie et al., 2022; Wang et al., 2022; Khashabi et al., 2020; Min et al., 2021; Ye et al., 2021; Liu et al., 2019; Zhong et al., 2021; Chung et al., 2022). Some research efforts even employ reinforcement learning from human feedback (RLHF) strategies to make models more adherent to human instructions. (Ouyang et al., 2022; Bai et al., 2022; Ziegler et al., 2020; Stiennon et al., 2022; Nakano et al., 2022; Korbak et al., 2023) Today, instruction tuning has had a profound impact on the field of natural language processing (NLP). The emergence of technologies such as ChatGPT (OpenAI, 2023a) and GPT-4 (OpenAI, 2023b) has attracted more researchers to engage in the development of instruction tuning. Compared to English instruction data, there is currently less research on instruction tuning on Chinese instruction data, which to some extent hinders the development of large language models in the Chinese field.

2.2 Parameter-efficient tuning

As the model size continues to increase, fine-tuning all parameters becomes more challenging since it is necessary to save the gradients and optimizer states for all parameters. Therefore, researchers have proposed parameter-efficient tuning, a low-resource and efficient tuning method that only tunes a small number of parameters or introduces additional trainable parameters. Prefix Tuning (Lester et al., 2021; Li and Liang, 2021; Liu et al., 2021) add trainable virtual token embeddings and fix the whole model. Adapters (Houlsby et al., 2019; Lin et al., 2020) inserting adapter layers between existing layers in neural networks and only fine-tuning the adapter network’s parameters.

(Aghajanyan et al., 2020) show that the learned over-parametrized models in fact reside on a low intrinsic dimension. (Hu et al., 2022) Inspired by this work and proposed LoRA approach, which suggests that weights update during model adaptation for downstream tasks should also have a low "intrinsic rank". Experimental results from (Ding et al., 2023) suggest that LoRA is a relatively effective method among various parameter-efficient tuning approaches. It has been adopted by many recent open-source projects (Stanford, 2023; Xu et al., 2023) for training large language models and achieved promising results. These research works only consider LoRA as a method of training models and does not have an in-depth analysis of factors affecting LoRA-based tuning results.

3 Method

In this section, we will provide a brief introduction to LoRA (Low-Rank Adaption) (Hu et al., 2022).

For a pre-trained weight matrix $W_0 \in R^{d \times k}$, its updates can be represented by a low-rank decomposition:

$$W_0 + \Delta W = W_0 + BA \quad (1)$$

where $B \in R^{d \times r}$, $A \in R^{r \times k}$, and the rank $r \ll \min(d, k)$. For a linear layer $h = W_0 x$, the forward pass is modified to be to be:

$$h = W_0 x + \Delta W x = W_0 x + BA x \quad (2)$$

Matrix A will be initialized by random Gaussian and B will be initialized by zero, making the initial value of $\Delta W = BA$ zero at the start of the training. (Hu et al., 2022) only adapted the attention

Table 1: The number and average prompt length of each type of instructions

Use case	#Nums	Prompt length
Others	113	57.56
Open QA	285	25.93
Brainstorming	179	27.24
Classification	65	73.09
Generation	98	54.89
Summarization	40	330.15
Rewrite	131	89.12
Closed QA	52	220.26
Extract	37	302.78

weights for downstream tasks and freeze the MLP modules, we follow Baize(Xu et al., 2023) which applies LoRA to adapt all linear layers at the same time.

4 Experiments

4.1 Datasets

We adopted the datasets constructed in our previous work(Ji et al., 2023b), selecting three data scales of 0.6M, 2M and 4M respectively. Combining these three datasets, we aim to investigate the impact of different training data sizes on the performance of LoRA-based tuning. To verify whether conducting LoRA-based tuning on the model after instruction tuning can further improve the model performance, we also selected the math_0.25M¹ dataset, which is a dataset focusing on the mathematical problem-solving field.

The evaluate set consists of 1,000 rigorously manually screened and processed data entries, covering nine categories, including translation, Open QA, closed QA, generation, and other tasks closely related to practical applications. Table 1 demonstrates the number of samples in each category of the evaluate set, as well as the average lengths of the input instructions and generated responses. The category Other contains two types of data: math and code, where math refers to solving mathematical application problems and code refers to code generation

4.2 Model Settings

In this study, we selected LLaMA(Touvron et al., 2023) as our foundational experimental models. LLaMA, released by Meta AI, is a collection of large-scale language models with four different parameter scales:

Table 2: Hyper-parameter settings of full-parameters fine-tuning

Hyper parameter	Value
Precision	bf16
Epochs	3
Batch size	32
Learning rate	5e-6
Warmup ratio	0.03
LR scheduler type	cosine
Max length	1024

Table 3: Hyper-parameter settings of LoRA-based tuning

Hyper parameter	Value
Precision	fp16
Epochs	4
Batch size	128
Learning rate	2e-4
Warmup steps	100
LR scheduler type	cosine
Max length	1024

7B, 13B, 33B, and 65B. The performance of LLaMA model is outstanding, with empirical evidence showing that LLaMA-13B, with only 1/10 of the parameter scale, outperforms GPT-3 (175B)(Brown et al., 2020) in most benchmark evaluations. In this paper, we chose LLaMA-7B and LLaMA-13B as our base experimental models.

For the full-parameters fine-tuning experiment, Table 2 list the hyper-parameters of fine-tuning.

For the LoRA experiment, we followed the hyper-parameters in (Xu et al., 2023), which set the rank in LoRA to 8 and apply LoRA to adapt attention weights and all linear layers, more details in list in Table 3. This experiment was conducted on 8 NVIDIA A100-40GB GPUs.

4.3 Metrics

ChatGPT is asked to evaluate responses generated by instruction-following models. For all instructions, ChatGPT gives a score between 0 and 1, where score 0 is the worst and score 1 is the best. In order to reduce randomness, we set the temperature to 0.001 for model generation. Evaluation is achieved by invoking gpt-3.5-turbo API at the time of April 15, 2023. We calculate model’s scores for

¹<https://huggingface.co/datasets/BelleGroup/OpenLLaMA>

Table 4: Main results. In this table, LLaMA-13B + LoRA(2M) represents a model trained on 2M instruction data using LLaMA-13B as base model and LoRA training method, and LLaMA-7B + FT(2M) represents a model trained using full-parameters fine-tuning. LLaMA-7B + FT(2M) + LoRA(math_0.25M) represents a model trained on 0.25M mathematical instruction data using LLaMA-7B + FT(2M) as the base model and LoRA training method, and LLaMA-7B + FT(2M) + FT(math_0.25M) represents a model trained using incremental full-parameters fine-tuning. About the training time, all these experiments were conducted on 8 NVIDIA A100-40GB GPUs.

Model	Average Score	Additional Param.	Training Time (Hour/epoch)
LLaMA-13B + LoRA(2M)	0.648	28M	8
LLaMA-7B + LoRA(4M)	0.624	17.9M	11
LLaMA-7B + LoRA(2M)	0.609	17.9M	7
LLaMA-7B + LoRA(0.6M)	0.589	17.9M	5
LLaMA-7B + FT(2M)	0.710	-	31
LLaMA-7B + FT(0.6M)	0.686	-	17
LLaMA-7B + FT(2M) + LoRA(math_0.25M)	0.729	17.9M	3
LLaMA-7B + FT(2M) + FT(math_0.25M)	0.738	-	6

each task category and derive its overall performance on the evaluation set using macro average across these categories.

Given limitations of ChatGPT in evaluating mathematical and coding tasks, we compute the scores that include all categories (denoted as average_score). The detailed scores on each task category can be found in the Appendix.

4.4 Comparison of Base Models and Dataset Scale for LoRA Tuning

Firstly, we designed an experiment to compare the performance of LoRA-based instruct tuning on instruction datasets of different sizes. We selected datasets of 0.6M, 2M, and 4M, and the experimental results are presented in Table 4. As can be seen from the results, similar to most learning tasks, as the dataset size increases, the LoRA-based instruct tuned model exhibits better performance in instruction comprehension.

In addition, we also compared the impact of different base models (LLaMA-7B and LLaMA-13B) on performance. It can be seen that the base model with a larger number of parameters brings a significant improvement in performance. Using LLaMA-7B+LoRA(2M) as the base, changing from 7B to 13B resulted in a larger improvement in performance compared to going from 2M to 4M.

In terms of training time, it can also be observed that LLaMA-13B+LoRA(2M) has certain advantages over LLaMA-7B+LoRA(4M). Better training results were achieved with less training time. However, it should be noted that when using these two models for inference, the LLaMA-

7B-based model has an advantage in terms of inference speed and cost due to its lower number of global parameters.

4.5 Comparison between Full-Parameter and LoRA-based Fine-Tuning

How does the performance of LoRA-based models compare to full-parameters finetuning? As a comparison, we trained two models using full-parameters fine-tuning on instruction training data of 0.6M and 2M, and the results are shown in Table 4, which are shown as LLaMA-7B + FT(0.6M) and LLaMA-7B + FT(2M). It can be seen that full-parameters fine-tuning brings better experimental results.

One intuitive understanding or analysis is that the pre-training large language model, which is trained to generate next word, requires a more complex learning task to switch to instruct following. LoRA’s learning method can only change a relatively small number of parameters, which is more challenging compared to changing all parameters.

Sure, there is no free lunch in the world. Compared to LoRA fine-tuning, using full-parameters fine-tuning requires about 3-5 times the time cost to complete the training.

4.6 Performing LoRA Tuning for Specified Task

According to our evaluation, details in the appendix, our models did not perform well on math tasks, with scores mostly below 0.5. To verify the adaptation capability of LoRA on specific tasks, we used incremental 0.25M math dataset

(math_0.25M) to adapt the instruction-following large language model (We chose LLaMA-7B + FT(2M) as the base model).

As a comparison, we used incremental fine-tuning with a learning rate of $5e-7$ and trained for 2 epochs. So we got two models, one is the LLaMA-7B + FT(2M) + LoRA(math_0.25M), and the other is LLaMA-7B + FT(2M) + FT(math_0.25M).

From the experimental results, it can be seen that incremental fine-tuning still showed better performance but took longer training time. Both LoRA and incremental fine-tuning improved the overall performance of the model. From the detailed data in the appendix, both LoRA and incremental fine-tuning showed significant improvements in the math task while only causing slight decreases in performance in other tasks. Specifically, the math task performance improved to 0.586 and 0.559 respectively.

4.7 Discussion and Conclusions

In this article, we conducted an experimental comparison between full-parameter fine-tuning and LoRA-based tuning methods using LLaMA as the base model. We also explored the impact of different amounts of training data and model parameters on the effectiveness of LoRA-based tuning. From the experimental results comparison, some interesting ideas can be observed:

1) The choice of the base model has a significant impact on the effectiveness of LoRA-based tuning. Comparing LLaMA-7B+LoRA(0.6M) and LLaMA-7B+FT(0.6M), as well as LLaMA-7B+LoRA(2M) and LLaMA-7B+FT(2M), it is evident that LoRA-based tuning on a base model that has not undergone instruction tuning has limited effectiveness and is far less effective than full-parameter fine-tuning (averaging 10 points lower). However, by comparing LLaMA-7B+FT(2M)+FT(math_0.25M) and LLaMA-7B+FT(2M)+LoRA(math_0.25M), it can be seen that LoRA-based tuning on a model that has undergone instruction tuning can achieve comparable results to fine-tuning. This indicates that the choice of the base model is crucial to the effectiveness of the LoRA-based tuning method.

2) Increasing the amount of training data can continuously improve the model's effectiveness. Comparing LLaMA-7B+LoRA(0.6M), LLaMA-7B+LoRA(2M), and LLaMA-7B+LoRA(4M)

shows that as the amount of training data increases, the model's effectiveness improves (an average of approximately 2 points improvement for every doubling of data).

3) LoRA-based tuning benefits from the number of model parameters. Comparing LLaMA-7B+LoRA(4M) and LLaMA-13B+LoRA(2M) shows that the number of model parameters has a greater impact on the effectiveness of LoRA-based tuning than the amount of training data.

References

- [Aghajanyan et al.2020] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. 2020. Intrinsic dimensionality explains the effectiveness of language model fine-tuning, December.
- [Bai et al.2022] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, et al. 2022. Constitutional ai: Harmlessness from ai feedback, December.
- [Brown et al.2020] Tom B. Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language models are few-shot learners, July.
- [Chowdhery et al.2022] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, et al. 2022. Palm: Scaling language modeling with pathways, October.
- [Chung et al.2022] Hyung Won Chung, Le Hou, Shayne Longpre, et al. 2022. Scaling instruction-finetuned language models, October.
- [Ding et al.2023] Ning Ding, Yujia Qin, Guang Yang, et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models, March.
- [Houlsby et al.2019] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, et al. 2019. Parameter-efficient transfer learning for nlp, June.
- [Hu et al.2022] Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models, June.
- [Ji et al.2023b] Yunjie Ji, Yong Deng, Yan Gong, Yiping Peng, Qiang Niu, Lei Zhang, Baochang Ma, and Xiangang Li. 2023b. Exploring the impact of instruction data scaling on large language models: An empirical study on real-world use cases, March.
- [Khashabi et al.2020] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*.
- [Korbak et al.2023] Tomasz Korbak, Kejian Shi, Angelica Chen, et al. 2023. Pretraining language models with human preferences, February.

- [Lester et al.2021] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning, April.
- [Li and Liang2021] Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation, January.
- [Lin et al.2020] Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2020. Exploring versatile generative language model via parameter-efficient transfer learning. In *Findings of the Association for Computational Linguistics: EMNLP*.
- [Liu et al.2019] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.
- [Liu et al.2021] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, et al. 2021. Gpt understands, too.
- [Min et al.2021] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*.
- [Nakano et al.2022] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, et al. 2022. Webgpt: Browser-assisted question-answering with human feedback, June.
- [OpenAI2023a] OpenAI. 2023a. Chatgpt: Optimizing language models for dialogue.
- [OpenAI2023b] OpenAI. 2023b. Gpt-4 technical report.
- [Ouyang et al.2022] Long Ouyang, Jeff Wu, Xu Jiang, et al. 2022. Training language models to follow instructions with human feedback, March.
- [Stanford2023] Stanford. 2023. Alpaca-lora.
- [Stiennon et al.2022] Nisan Stiennon, Long Ouyang, Jeff Wu, et al. 2022. Learning to summarize from human feedback, February.
- [Touvron et al.2023] Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- [Wang et al.2022] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109.
- [Workshop et al.2022] BigScience Workshop, Teven Le Scao, Angela Fan, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model, December.
- [Xie et al.2022] Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I Wang, et al. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *arXiv preprint arXiv:2201.05966*.
- [Xu et al.2022] Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Yanggang Wang, Haiyu Li, and Zhilin Yang. 2022. Zeroprompt: Scaling prompt-based pretraining to 1,000 tasks improves zero-shot generalization. *arXiv preprint arXiv:2201.06910*.
- [Xu et al.2023] Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data, April.
- [Ye et al.2021] Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. Crossfit: A few-shot learning challenge for cross-task generalization in nlp. *arXiv preprint arXiv:2104.08835*.
- [Zeng et al.2023] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130b: An open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations (ICLR)*.
- [Zhang et al.2022] Susan Zhang, Stephen Roller, Naman Goyal, et al. 2022. Opt: Open pre-trained transformer language models, June.
- [Zhong et al.2021] Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. *arXiv preprint arXiv:2104.04670*.
- [Ziegler et al.2020] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, et al. 2020. Fine-tuning language models from human preferences, January.

5 Appendix A

5.1 Detailed evaluation scores

Table 5: Detailed scores on each task category.

Model	Training data	others	rewrite	classification	generation	summarization	extract	open qa	brain-storming	closed qa	macro ave
LLaMA-7B+ LoRA	0.6M	0.358	0.719	0.695	0.816	0.65	0.448	0.315	0.793	0.51	0.589
LLaMA-7B+ LoRA	2M	0.364	0.795	0.676	0.854	0.617	0.472	0.369	0.808	0.531	0.61
LLaMA-7B+ LoRA	4M	0.341	0.821	0.677	0.847	0.645	0.467	0.374	0.806	0.639	0.624
LLaMA-13B+ LoRA	2M	0.422	0.810	0.696	0.837	0.700	0.537	0.435	0.823	0.577	0.648
LLaMA-7B+ FT	0.6M	0.438	0.869	0.698	0.917	0.701	0.592	0.477	0.870	0.606	0.686
LLaMA-7B+ FT	2M	0.399	0.871	0.775	0.920	0.734	0.603	0.555	0.900	0.633	0.710
LLaMA-7B + FT(2M) + LoRA	math0.25M	0.560	0.863	0.758	0.915	0.754	0.651	0.518	0.886	0.656	0.729
LLaMA-7B + FT(2M) + FT	math0.25M	0.586	0.887	0.763	0.955	0.749	0.658	0.523	0.872	0.652	0.738