Evaluating Parameter-Efficient Tuning Methods with Limited Data

Haoming(Hammond) Liu
NYU Shanghai
hl3797@nyu.edu

Xiaochen(Nigel) Lu
NYU Shanghai
xl3139@nyu.edu

Wenbin(Jim) Qi
NYU Shanghai
wq372@nyu.edu

Abstract

Numerous parameter-efficient tuning methods have been proposed to reduce the computation and storage burden of standard fine-tuning. However, these methods are generally evaluated on large-scale benchmarks, which may not secure their robustness under limited data. This work further evaluates the effectiveness and efficiency of several representative methods when different amount of training samples are provided. The experiments have been conducted on various downstream tasks, which could serve as an empirical guidance on the choice of methods. The code is available at: https://github.com/hmdliu/MLLU-S22.

1 Introduction

Pre-trained language models (PLMs) have achieved great success in processing natural languages. *Fine-tuning*, though found effective by some predominant works like BERT (Devlin et al., 2019) and GPT (Brown et al., 2020), needs to store a model copy for each downstream task and update all the parameters for adaption, which is not efficient in terms of storage and computational resources.

To alleviate this issue, there have been many works exploring how to adapt PLMs to downstream tasks effectively and efficiently (Houlsby et al., 2019; Zaken et al., 2021; Hu et al., 2021; Li and Liang, 2021; Lester et al., 2021). In general, these methods tunes a small set of adaptive parameters (inherently in the model or additionally introduced) instead of the whole PLM - such a paradigm is termed as *delta tuning* by Ding et al. (2022).

The evaluation of *delta tuning* methods are usually conducted on benchmarks (*e.g.,* GLUE, SuperGLUE) or datasets (*e.g.,* E2E, XSUM), where the amount of training samples are generally fixed and sufficient for the downstream task adaption (Wang et al., 2018, 2019; Novikova et al., 2017; Narayan et al., 2018). However, downstream tasks are likely to have limited training data, so the effectiveness and efficiency of different *delta tuning* methods may vary a lot depending on the number of training samples provided.

To address this potential vulnerability, we set up a unified testing framework to compare the performance and convergence speed of different *delta tuning* methods while varying the size of training data. The experiments are conducted across various downstream tasks, including sentiment analysis, natural language inference, machine reading comprehension, and multi-choice question answering. We hope this work can serve as a empirical guidance for method selection.

2 Related Works

This section introduces the categorization of *delta tuning* methods and some representative methods to be evaluated. As Figure 1 illustrates, *delta tuning* methods can be categorized as addition-based, specification-based, and reparameterization-based methods, which are differentiated by the usage of adaptive parameters (Ding et al., 2022).

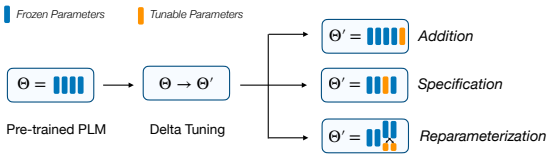

Figure 1: The categorization criterion of *delta tuning*, where Θ denotes the pre-trained parameters, Θ' denotes the well-tuned parameters. The figure and caption is drawn from the original paper for demonstration purpose (Ding et al., 2022).

2.1 Addition-based Method

Addition-based methods introduce extra trainable neural modules or parameters that do not exist in the original model or process (Ding et al., 2022).

Houlsby et al. (2019) proposes to insert small and tunable *Adapter* modules between layers of a pre-trained network. The *Adapter* module pass the features through a down-projection, a non-linearity, and a up-projection to adapt the PLM representations to downstream tasks.

Li and Liang (2021) proposes to prepend a small continuous task-specific vector (termed *prefix*) to the input and hidden states at each Transformer layer. Moreover, the *prefix* is reparameterized by a MLP to stabilize training.

2.2 Specification-based Method

Specification-based methods specify a subset of parameters in the original model or process to be trainable, while others are frozen (Ding et al., 2022).

Zaken et al. (2021) simply tunes all the bias terms in the original PLM, leaving all other parameters frozen. This method works surprisingly well, especially on small-scale models.

2.3 Reparameterization-based Method

Reparameterization-based methods convert the existing parameters to a parameter-efficient form via reparameterization (Ding et al., 2022).

Hu et al. (2021) hypothesizes that the change of weights during model adaptation has a low intrinsic rank. Accordingly, we can inject trainable low-rank decomposition matrices to replace the self-attention weight matrices. This significantly reduces the number of trainable parameters and retains performance that is comparable to fine-tuning.

3 Method

3.1 Evaluation Methodology

Ding et al. (2022) have conducted a comprehensive study on different aspects of *delta tuning*. In this work, we largely adopt their unified testing framework and further investigate the variations of the performance and efficiency when limited training samples are provided.

Specifically, we sample subsets from each dataset (*i.e.,* 0.2%, 1%, 5%, 20%, 100%, roughly by a log scale[1]) to construct a series of training set for evaluation. Then we apply the selected *delta tuning* methods on these subsets to observe the average metrics and the speed of convergence, which would demonstrate the robustness of different methods with limited data. Notably, the module or structure of each *delta tuning* method follows the default settings in the original paper; the configurations of training and hyper-parameter search are also unified. More implementation details are discussed in Section 4.1.

3.2 Pre-trained Language Model

We use `T5-Base` as the default PLM backbone for evaluation[2] (Raffel et al., 2020). Moreover, we use the checkpoints released by Lester et al. (2021), which conducts an additional 100k steps of LM adaption. As T5 (Raffel et al., 2020) modulates all downstream tasks in a text-to-text format and pretrains on a span corruption objective, these extra training steps have been found effective for boosting performance and convergence speed.

3.3 Dataset

To evaluate the selected *delta tuning* methods comprehensively, we use four large and representative dataset targeting distinctive downstream tasks.

MultiNLI. Multi-Genre Natural Language Inference corpus is a crowd-sourced collection of 433k sentence pairs annotated with textual entailment information (*i.e.,* neutral, entailment, or contradiction) (Williams et al., 2018).

RACE. RACE is a reading comprehension dataset with roughly 28k passages and 100k multiple choice questions; the dataset is collected from English examinations for middle school and high school students in China (Lai et al., 2017).

SQuAD v1. Stanford Question Answering Dataset is a reading comprehension dataset, consisting of more than 100k question-answer pairs on over 500 articles (Rajpurkar et al., 2016).

Yelp Polarity. Yelp Polarity is a sentiment analysis dataset with a set of 560k highly polarized yelp reviews for training, and 38k for testing. The dataset is constructed by Zhang et al. (2015) based on the Yelp Dataset Challenge 2015 data[3].

4 Experiment & Analysis

4.1 Implementation Details

The codebase is implemented based on the Hugging Face library (Lhoest et al., 2021). The sequence-to-sequence trainer is adopted from Mahabadi et al. (2021) and the implementation of *delta*

[1]The choice of scales might differ depending on further experiment results.

[2]If time permits, we may extend to `T5-Small` and `T5-Large` to evaluate across PLM scales.

[3]The challenge web page is defunct, the Yelp dataset is now available at: `https://www.yelp.com/dataset`

Dataset	Adapter	BitFit	LoRA	Prefix-tuning	Fine-tuning
Training Data Percentage: 100%					
MultiNLI (Williams et al., 2018)	86.57	83.42	84.48	39.55	**86.69**
RACE (Lai et al., 2017)	–	–	–	–	–
SQuAD v1 (Rajpurkar et al., 2016)	–	–	–	–	–
Yelp Polarity (Zhang et al., 2015)	–	–	–	–	–

Table 1: Test accuracy (%) of *delta tuning* methods on the MultiNLI dataset.

tuning methods is drawn from the OpenDelta library by Ding et al. (2022).

We use the AdamW optimizer (Loshchilov and Hutter, 2019) and set the maximum training steps to 20k with default early stop settings. We apply random hyper-parameter search for 5 times and report the best performance, where the learning rates are sampled uniformly from $(1 \times 10^{-3}, 1 \times 10^{-4})$ and the batch sizes are sample from $\{16, 32\}$.

Follow the implementation from OpenDelta (Ding et al., 2022), we insert Adapter modules (Houlsby et al., 2019) into both the multi-head attention module and the feed-forward network in each Transformer layer, with `SiLU` activations and a bottleneck dimension of 64; BitFit (Zaken et al., 2021) tunes all the bias terms in each Transformer layer by default; we reparameterize all the query matrices and the value matrices in the multi-head attention modules as decompositions of rank 4 for the LoRA (Hu et al., 2021) method; we use tokens of size 5 for prefix-tuning (Li and Liang, 2021).

4.2 Results & Analysis

Some preliminary results are shown in Table 1. All the models are trained for 1 epoch on the whole training set with a batch size of 8 and a fixed learning rate of 3×10^{-4}.

Please kindly note that the main purpose of this experiment is to test the correctness of our codebase implementation. Accoridngly, the results we got are highly consistent with the convergence speed reported by Ding et al. (2022). With this in mind, we can extend our experiments to the other three datasets, set up hyper-parameter search, and evaluate the performance and convergence speed on the subsets of training data.
More details to be added in the final report.

5 Conclusion

More details to be added in the final report.

Collaboration Statement

All group members attended the regular project meetings, completed the literature review, and wrote up the partial draft together. Hammond set up the codebase; Nigel and Jim ran the preliminary experiments on NYU HPC (GCP).

Acknowledgements

We thank Professor Samuel R. Bowman and Jason Phang for advising on this work.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. 2022. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

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

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *ArXiv preprint*, abs/2104.08691.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: 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 (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. *ArXiv preprint*, abs/2106.04647.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE:

A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *ArXiv preprint*, abs/2106.10199.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.