FRI Natural language processing course 2024
Project report

University *of Ljubljana*
Faculty *of Computer and Information Science*

Parameter-Efficient Fine-Tuning of Language Models

Lena Trnovec, Tjaša Domadenik, Anže Oblak

Abstract

The abstract goes here. The abstract goes here.

Keywords

Fine-tuning ...

Advisors: Aleš Žagar

Introduction

Transformer-based pretrained language models (PLMs) have revolutionized natural language processing (NLP) tasks, by demonstrating remarkable success in understanding and generating human-like text. To fully harness their potential, fine-tuning is employed to adapt the models to task specific data and enhance their performance. However, traditional fine-tuning involves updating all the pretrained parameters of PLM, which requires substantial computational resources. With PLMs continuing to grow, the number of parameters increases even further and so do the computational demands, which becomes a significant challenge. In response to that, various parameter efficient fine-tunning (PEFT) techniques have emerged, as a viable solution to compensate for the tremendous computational cost, while still maintaining comparable performance to the full fine-tuning.

One of the commonly used techniques is additive fine-tuning, which adds new trainable parameters to pre-trained models. It can be divided into adapter and soft prompt-based methods. Adapters insert small modules between transformer layers in the existing model structure, while soft prompts add trainable vectors to the model inputs, subtly directing the output with minimal changes. In previous research a variety of adapter methods were explored. Some of the notable ones are parallel adapter [1] which integrates the network in parallel with the feed-forward layer and attention layer and AdapterDrop [2] which selectively removes adapters which are not important to the given task. Both methods build upon the foundational concept discussed in [3] and present valuable case studies for our investigation. We will also examine prefix-tuning, a prompt-based method [4]. This approach focuses on optimizing a task-specific set of parameters, known as a *prefix*. It modifies only about 0.1% of the parameters in comparison to traditional fine-tuning. Despite this, it achieves comparable performance in datasets with full data settings, making it an interesting candidate for comparison.

Another method, known as Low-Rank Adaptation (LoRA), is a reparameterized fine-tuning method that specifically targets the model's weight matrices. It introduces low-rank matrices that interact with the original weights to apply precise updates, rather than altering or adding extensive new parameters [5]. This strategy allows for the efficient adaptation of PLMs by modifying a relatively small subset of parameters, thereby mitigating the computational load while still achieving competitive performance [6, 7].

Another possible approach is hybrid fine-tuning, which tries to combine various PEFT approaches, such as adapter, prefix-tuning and LoRA. This way it leverages the strengths of each method and mitigates their weaknesses and consequently achieves improved overall performance compared to individual PEFT methods. The work in this area is classified into two approaches: Manual Combination and Automatic Combination. The first one involves manually combining multiple PEFT methods by sophisticated design. On the other hand Automatic Combination incorporates PEFT methods automatically via structure search and because of that it typically requires more time and cost. In one of the previous articles [8] authors proposed a method called UniPELT which incorporates sequential adapter, prefix-tuning, and LoRA via a gating mechanism. The mechanism controls the activation of each submodule, dynamically assigning higher weights to submodules that make positive contributions to a given task. The method requires more parameters and inference time than adapter, prefix-tuning, and LoRA, but achieves better performance than those individual methods. Another article [9], presents method AutoPEFT that integrates sequential adapter, parallel adapter and prefix tuning into the transformer block. The method also uses Bayesian optimization approach to automatically search for an appropriate architecture of neural network that activates certain layers to incorporate these PEFT modules.

Methods

- Which pretrained language model to use?
  - **BERT** (Bidirectional Encoder Representations from Transformers) - highly successful across a wide range of NLP tasks, well-suited for understanding tasks but can be adapted for generation tasks as well.
  - **GPT** (Generative Pre-trained Transformer) - excels in generation tasks and can perform admirably in understanding tasks when fine-tuned or used with prompting techniques.
  - **T5** (Text-to-Text Transfer Transformer) - versatile for both understanding and generation tasks.

  (T5 seems like the best choice, TBD...). First testing on smaller models then onto larger ones (colab or SLING).
- Methods:
  - **Prompt-Tuning:** Fine-tunes a small set of task-specific tokens appended to the input without altering the original model parameters.
  - **Prefix-Tuning:** Attaches trainable vectors (prefixes) to the input sequence to guide the model's attention mechanism for specific tasks.
  - **Adapter:** Inserts small, trainable neural network layers between the pre-existing layers of the model to adapt to new tasks.
  - **LoRA:** Modifies the weight matrices of a model by applying low-rank updates, preserving the original parameters while adapting to new tasks.
  - **P-tuning** - adds trainable prompt embeddings to the input that is optimized by a prompt encoder to find a better prompt, eliminating the need to manually design prompts. The prompt tokens can be added anywhere in the input sequence, and p-tuning also introduces anchor tokens for improving performance.
  - **Hybrid approaches** - combine various PEFT approaches and usually achieve better performance compared to the individual PEFT methods.
- Datasets

  At least 5 different datasets that cover various natural language understanding skills (commonsense reasoning, coreference resolution, text summarization, etc.) and supervised learning settings (classification & generation).

  From SloBENCH?
  - https://slobench.cjvt.si/leaderboard/view/9 (sequence classification) (Given a premise and a hypothesis, the task is to detect whether the hypothesis entails, contradicts, or is neutral in relation to the premise.)
  - https://slobench.cjvt.si/leaderboard/view/12 (token classification) (Given tokenized text, the task is to annotate each token with an appropriate label.)
  - https://slobench.cjvt.si/leaderboard/view/7 (conditional generation, slo-eng translation)
  - https://slobench.cjvt.si/leaderboard/view/8 (conditional generation, slo-eng translation)
- Which metrics to observe? (accuracy, F1...)

Results

TODO

Discussion

TODO

Acknowledgments

TODO

References

[1] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021.

[2] Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. Adapterdrop: On the efficiency of adapters in transformers. *arXiv preprint arXiv:2010.11918*, 2020.

[3] 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.

[4] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[5] 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, October 2021. arXiv:2106.09685 [cs].

[6] Yuchen Zeng and Kangwook Lee. The Expressive Power of Low-Rank Adaptation, March 2024. arXiv:2310.17513 [cs, stat].

[7] Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. LIFT: Language-Interfaced Fine-Tuning for Non-Language Machine Learning Tasks, October 2022. arXiv:2206.06565 [cs].

[8] Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Wen-tau Yih, and Madian Khabsa. Unipelt: A unified framework for parameter-efficient language model tuning. *arXiv preprint arXiv:2110.07577*, 2021.

[9] Han Zhou, Xingchen Wan, Ivan Vulić, and Anna Korhonen. Autopeft: Automatic configuration search for parameter-efficient fine-tuning. *arXiv preprint arXiv:2301.12132*, 2023.