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

Pretrained models we will be using:

- **google-bert/bert-base-uncased:** Pretrained model on English language using a masked language modeling (MLM) objective. This model is uncased, which means that it does not make a difference between english and English. It's good model for our starting implementations, because it's a bit smaller.
- **EMBEDDIA/crosloengual-bert:** A trilingual model, using bert-base architecture, trained on Croatian, Slovenian, and English corpora. We opted for this model due to its potential for good performance, since it is already trained on Slovenian corpora.

PEFT approaches used:

- **Prompt-Tuning:** Fine-tunes a small set of task-specific tokens appended to the input without altering the original model parameters.
- **LoRA:** Modifies the weight matrices of a model by applying low-rank updates, preserving the original parameters while adapting to new tasks.
- **LoHa:** Is similar to LoRA except it approximates the large weight matrix with more low-rank matrices and combines them with the Hadamard product. The method is supposed to be even more parameter-efficient than LoRA, yet it achieves performance levels comparable to it.

We decided to use **Slovene SuperGLUE** dataset, from Slobench, because it provides multiple different tasks, which are as follows:

- **BoolQ:** determine whether a given passage contains the answer to a yes/no question.
- **CB:** determine the commitment of a statement to a specific target.
- **COPA:** presents a premise and requires choosing the correct alternative explanation or cause.
- **MultiRC:** answering multiple-choice questions based on a given passage, with each question having multiple correct answers.
- **RTE:** determine whether one text implies another, often categorized as entailment, contradiction, or neutral.
- **WSC:** test machines' understanding of pronouns and their antecedents in a sentence.

Metrics:

- **accuracy**
- **F1 score**

Previously described methods are not necessarily final. We may change some of them, in order to achieve a manageable set of testing combinations.

Results

Up to this point, we tried to fine tune some models on different tasks from the dataset. The results are shown in tables 1, 2 and 3. During the process of training and fine tuning, we ran into some problems, as the models achieved very bad performance in some cases. Despite thorough debugging and testing of our implementations, we are still not sure if there is a bug in our code or if the models really just struggle with these tasks.

Method	Model	Accuracy	f1
fine tuning only	bert-base-uncased	0.72	0.73
prompt tuning	bert-base-uncased	0.83	0.79
LoHa	bert-base-uncased	0.78	0.68
LoRa	bert-base-uncased	0.78	0.68
fine tuning only	crosloengual-bert	0.83	0.82
prompt tuning	crosloengual-bert	0.78	0.68
LoHa	crosloengual-bert	0.83	0.82
LoRa	crosloengual-bert	0.78	0.68

Table 1. BoolQ task results on evaluation split of the dataset.

Method	Model	Accuracy	f1
fine tuning only	bert-base-uncased	0.36	0.24
prompt tuning	bert-base-uncased	0.32	0.15
LoHa	bert-base-uncased	0.32	0.15
LoRa	bert-base-uncased	0.32	0.15
fine tuning only	crosloengual-bert	0.68	0.64
prompt tuning	crosloengual-bert	0.32	0.15
LoHa	crosloengual-bert	0.32	0.15
LoRa	crosloengual-bert	0.32	0.15

Table 2. CB task results on evaluation split of the dataset.

Method	Model	Accuracy	f1
fine tuning only	bert-base-uncased	0.55	0.39
LoHa	bert-base-uncased	0.52	0.52
LoRa	bert-base-uncased	0.50	0.49
fine tuning only	crosloengual-bert	0.51	0.49
LoHa	crosloengual-bert	0.58	0.58
LoRa	crosloengual-bert	0.49	0.49

Table 3. COPA task results on evaluation split of the dataset.

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.