International Conference on Formal Methods
and Foundations of Artificial Intelligence
Eszterházy Károly Catholic University
Eger, Hungary, June 5–7, 2025

Full-Parameter Tuning vs. LoRA Tuning on PULI Models

Kristóf Varga[a], **Péter Hatvani**[ab], **Zijian Győző Yang**[a]

[a]HUN-REN Hungarian Research Centre for Linguistics
(familyname).(givenname)@nytud.hun-ren.hu

[b]Pázmány Péter Catholic University
Doctoral School of Linguistics
hatvani.peter@hallgato.ppke.hu

Abstract

In recent months, large language models have undergone significant development and have become one of the most popular topics in the field of artificial intelligence. The training of language models consists of two well-defined phases: pretraining and fine-tuning. During pretraining, a neural network (most commonly based on the transformer architecture [6]) is trained on general language understanding. After this phase, the model is further trained for task-specific knowledge through fine-tuning. In the case of large language models, fine-tuning can be used to train the model for conversational purposes or domain-specific knowledge. Although fine-tuning requires fewer resources than pre-training, it still requires significant hardware power when working with large language models. In traditional fine-tuning, all the parameters of the pretrained model are updated – this is called full-parameter fine-tuning. However, this approach is extremely resource intensive. To address this, parameter-efficient methods have been developed [2, 4, 9], among which one of the most popular is LoRA. LoRA improves training efficiency in multiple ways; one key aspect is that it adapts a pre-trained weight matrix using a low-rank decomposition, which significantly reduces the number of trainable parameters.

In our research, we compared full-parameter and LoRA fine-tuning on various Hungarian large language models, specifically the PULI models. For both methods, we experimented with different sets of hyperparameters. The aim of this study was to explore the potential of these fine-tuning approaches; therefore, we

Table 1. Full-parameter and Lora results

		HuCOLA (MCC)	HuCoPA (MCC)	HuRTE (MCC)	HuSST (ACC)	HuWNLI (ACC)
PULI 3SX	Full	**59.5**	3.7	44.7	**79.7**	51.5
	Lora	59.0	**5.3**	**55.5**	79.3	**58.2**
PULI LlumiX	Full	64.1	**73.4**	52.6	80.2	59.7
	Lora	**70.3**	64.2	**68.2**	**81.9**	**67.2**
PULI-LlumiX-Llama 3.1	Full	**71.0**	73.2	61.1	81.5	59.7
	Lora	69.2	**74.1**	**71.9**	**82.2**	**73.1**

reported only the highest performance values achieved. Our experiments were conducted on the monolingual PULI 3SX [8], as well as the two latest multilingual PULI models: PULI LlumiX [7] and PULI-LlumiX-Llama-3.1[1]. For evaluation, we used five Hungarian HuLU benchmarks [3]: HuCOLA, HuCoPA, HuRTE, HuSST, and HuWNLI. For the HuCOLA, HuRTE, HuSST, and HuWNLI benchmarks, we trained the models using a sequence classification setup, while for HuCoPA, we trained the models as a multiple-choice task.

To perform the fine-tuning experiments, we used the Hugging Face implementation for full-parameter tuning[2]. In these experiments, it was necessary to configure both the Accelerate [1] and FSDP [10] methodologies; otherwise, training resulted in out-of-memory errors, even when using four A100 GPUs (80GB each). Despite using Accelerate and FSDP, full-parameter fine-tuning still required at least two GPUs to run successfully.

For the LoRA experiments, we used the HuLU-evaluate library's implementation [5]. For this task, a single GPU was sufficient. The following LoRA hyperparameters were used: r = 8, LoRA alpha = 32; LoRA dropout = 0.1. Since neither the Hugging Face implementation[3] of the GPT-NeoX nor the LLaMA models supports the multiple-choice task type, we implemented this functionality ourselves, based on the HuLU-evaluate framework.

Table 1 presents the results of the full-parameter and LoRA fine-tuning experiments. Similar to the findings in the original research [2], in many cases, the LoRA method achieved higher performance. However, full-parameter tuning still shows potential, achieving significantly better results in certain benchmarks (e.g., HuCOLA and HuCoPA). Overall, in most cases, LoRA achieved comparable or even superior performance, while requiring substantially fewer computational resources.

[1] https://huggingface.co/NYTK/PULI-LlumiX-Llama-3.1
[2] https://github.com/huggingface/transformers/tree/main/examples/pytorch
[3] https://github.com/huggingface/transformers/tree/main/src/transformers/models

References

[1] S. Gugger, L. Debut, T. Wolf, P. Schmid, Z. Mueller, S. Mangrulkar, M. Sun, B. Bossan: *Accelerate: Training and inference at scale made simple, efficient and adaptable.* https://github.com/huggingface/accelerate, 2022.

[2] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen: *LoRA: Low-Rank Adaptation of Large Language Models*, in: International Conference on Learning Representations, 2022, url: https://openreview.net/forum?id=nZeVKeeFYf9.

[3] N. Ligeti-Nagy, G. Ferenczi, E. Héja, L. J. Laki, N. Vadász, Z. G. Yang, T. Váradi: *HuLU: Hungarian Language Understanding Benchmark Kit*, in: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ed. by N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue, Torino, Italia: ELRA and ICCL, May 2024, pp. 8360–8371, url: https://aclanthology.org/2024.lrec-main.733/.

[4] H. Liu, D. Tam, M. Mohammed, J. Mohta, T. Huang, M. Bansal, C. Raffel: *Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning*, in: Advances in Neural Information Processing Systems, ed. by A. H. Oh, A. Agarwal, D. Belgrave, K. Cho, 2022, url: https://openreview.net/forum?id=rBCvMG-JsPd.

[5] K. V. Péter Hatvani, Z. G. Yang: *Evaluation Library for the Hungarian Language Understanding Benchmark (HuLU)*, in: Proceedings of the 21th Hungarian Computational Linguistics Conference, Affiliations: PPKE Doctoral School of Linguistics, HUN-REN Hungarian Research Center for Linguistics, Hungary: Szegedi Tudományegyetem TTIK, Informatikai Intézet, 2024, url: https://hatvanipeter.hu/.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin: *Attention is All you Need*, in: Advances in Neural Information Processing Systems 30, ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett, Curran Associates, Inc., 2017, pp. 5998–6008.

[7] Z. G. Yang, R. Dodé, G. Ferenczi, P. Hatvani, E. Héja, G. Madarász, N. Ligeti-Nagy, B. Sárossy, Z. Szaniszló, T. Váradi, T. Verebélyi, G. Prószéky: *The First Instruct-Following Large Language Models for Hungarian*, in: 2024 IEEE 3rd Conference on Information Technology and Data Science (CITDS) Proceedings, Debrecen, Hungary: University of Debrecen, 2024, pp. 247–252, isbn: 9798350387889.

[8] Z. G. Yang, L. J. Laki, T. Váradi, G. Prószéky: *Mono- and multilingual GPT-3 models for Hungarian*, in: Text, Speech, and Dialogue, Lecture Notes in Computer Science, Plzeň, Czech Republic: Springer Nature Switzerland, 2023, pp. 94–104, isbn: 978-3-031-40498-6.

[9] Q. Zhang, M. Chen, A. Bukharin, P. He, Y. Cheng, W. Chen, T. Zhao: *Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning*, in: The Eleventh International Conference on Learning Representations, 2023, url: https://openreview.net/forum?id=lq62uWRJjiY.

[10] Y. Zhao, A. Gu, R. Varma, L. Luo, C.-C. Huang, M. Xu, L. Wright, H. Shojanazeri, M. Ott, S. Shleifer, A. Desmaison, C. Balioglu, P. Damania, B. Nguyen, G. Chauhan, Y. Hao, A. Mathews, S. Li: *PyTorch FSDP: Experiences on Scaling Fully Sharded Data Parallel*, 2023, arXiv: 2304.11277 [cs.DC], url: https://arxiv.org/abs/2304.11277.