Electronics and information technologies, 2025, 30, 33–42
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC 004.89

PARAMETER EFFICIENT FINE-TUNING AND OVERFITTING IN GPT LARGE LANGUAGE MODELS: A METRIC-BASED COMPARISON

Bohdan Pavlyshenko*, *Ivan Bulka*
Ivan Franko National University of Lviv,
50 Drahomanova St., 79005 Lviv, Ukraine

Pavlyshenko B. M, Bulka I. I. (2025). Parameter Efficient Fine-Tuning and Overfitting in GPT Large Language Models: a Metric-Based Comparison. *Electronics and Information Technologies*, 30, 33-42. https://doi.org/10.30970/eli.30.3

ABSTRACT

Background. Building upon previous research, this study conducts an exploration into Large Language Models (LLMs), with an emphasis on the fine-tuning and assessment of LLaMA-3.1 for instructional tasks. LLaMA-3.1, which is a new generation model and has gained considerable recognition based on its superior performance on various benchmarks. Besides assessing the disparities and improvements between the base and the fine-tuned versions of LLaMA-3.1 on an instruction dataset, the study also addresses the concern of overfitting with LLaMA-3.1. Furthermore, it carries out a comparison between LLaMA-3.1 and both its predecessor, LLaMA-2, and another LLM known as Mixtral, thereby providing a more comprehensive picture of LLaMA-3.1's capabilities compared to other models.

Materials and Methods. The fine-tuning of LLaMA-3.1 employed state-of-the-art techniques, such as Low-Rank Adaptation (LoRA) and Quantized Low-Rank Adaptation (QLoRA), on comprehensive instruction datasets. Acknowledging the resource-intensive nature of LLM fine-tuning, optimization measures were taken. The fine-tuning process was additionally enhanced using Parameter-Efficient Fine-tuning (PEFT) on NVIDIA A100 Tensor Core GPU (graphics processing unit) instances. All the models were fine-tuned using Hugging Face and PyTorch platforms for optimal performance.

Results and Discussion. The results obtained from fine-tuning and evaluating LLaMA-3.1 offer valuable insights into how this model performs with specific tasks. The evaluation framework proved helpful in the efficient assessment assessing LLMs' performance concerning instruction tasks. The research highlights the importance of evaluation for LLM applications. It shows that not always is fine-tuning a good choice, due to the nature of the model and the specifics of the task. It highlights the overfitting problem.

Conclusion. The close examination of LLaMA-3.1 contributes to the field of machine learning by offering insights into how this model works and its possible fine-tuning for special tasks. The findings of this research create opportunities for more in-depth studies around the application of LLMs. It highlights the importance of efficient evaluation with already designed metrics.

Keywords: LLMs, GPT, Mixtral, LLaMA, fine-tuning, overfitting

INTRODUCTION

The swift progress of Natural Language Processing (NLP) has been majorly steered by LLMs like Transformers [1, 2], Bidirectional Encoder Representations from Transformers (BERT) [3, 4], Generative Pretrained Transformer (GPT) [5], etc. They've cleared new ways for various tasks like text classification [6], machine translation [7], and summarization [8].

Now, more advanced models like GPT-3.5 [9], GPT-4 [10], and Claude [11] have expanded NLP's scope by simply following user instructions and explaining patterns.

LLMs can be helpful tools in media and communication, helping to distinguish between real news and fake or biased news [12]. They can also be used in finance, where they can be very helpful in conducting detailed studies of financial news [13, 14]. This widespread use of LLMs highlights their importance and the possibility of further study in different areas.

Applying LLMs to niche domains brings unique complications. Supervised Fine-Tuning [15] methods usually help tailor these LLMs for specific uses, but the balancing between providing comprehensive language capabilities and achieving sector-specific efficacy is complex. This difficulty becomes important in business settings where these models grapple with specialized queries needing custom solutions.

A new generation of models, like GPT-4, Claude, can be accessed via Application Programming Interfaces (APIs), which raises issues about private data handling. A large number of tasks can be solved using Retrieval Augmented Generation (RAG) [16]. However, the transfer of data to third-party apps is still inevitable. Increasing the number of API requests could also escalate costs.

Thus, the alternative would be to custom fine-tune [17] and store models on personal resources. This ensures data security and potentially offers cost advantages. LLMs can be used in various spaces like the media [18] and finance [13, 19], due to their broad range of applications.

This paper focuses on the Fine-tuning and evaluation of a new model created by Facebook. It's LLaMA-3.1 [20]. The model was fine-tuned using state-of-the-art methods like LoRA [21, 22] and QLoRA [23]. It compares the current model with fine-tuned models from the previous article [24].

We would focus on the latest LLaMA-3.1 and assess its performance in carrying out instructional tasks. This model has been thoroughly optimized by techniques like LoRA, QLoRA, and Parameter-Efficient Fine-Tuning (PEFT), which we validate through robust evaluation approaches. Our findings can guide future research and applications for Large Language Models.

MATERIALS AND METHODS

This study involves fine-tuning the model LLaMA-3.1 for instruction-based tasks. LLaMA-3.1 exists in three different sizes: 8B, 70B, and 405 B. Due to resource constraints and to compare results with the previous fine-tuned models, the 8b models were selected for this experiment. The PyTorch library was utilized for the fine-tuning process.

Training Dataset

A training dataset consolidates two freely accessible datasets, namely, Instruct-v3 [25] and Alpaca [26]. These datasets, crafted for refining instructions, are accessible from GitHub. To ensure suitability, a filtering process was conducted on these datasets to retain only those instructions composed of fewer than 1024 tokens.

A dataset was partitioned into three unique sections: training, validation, and testing. These sections contained 83k, 10k, and 3k records in their respective order. They served various purposes: the training section was used for refining the models, the validation section verified the efficacy of training during the refining process, and the testing section helped evaluate the efficiency of the final models. This dataset was used to fine-tune LLaMA-2 and Mixtral for the previous research. It allows us to compare LLaMA-3.1 with LLaMA-2 and Mixtral models that were fine-tuned in the previous paper.

A key element for fine-tuning LLaMA-3.1 is the formatting of the training dataset. Without proper formatting, the results may be significantly degraded and fail to reflect the true capabilities of the model. For this study, we employed a consistent template when preparing the dataset, ensuring that each sample followed the same conversational structure.

The dataset was serialized using a custom prompt template, designed to mimic a natural conversational exchange between a user and the assistant, while also allowing for the inclusion of system-level instructions. Each training sample in the dataset adheres to the following format:

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>
{{ system_prompt }}<|eot_id|><|start_header_id|>user<|end_header_id|>
{{ user_msg_1 }}<|eot_id|><|start_header_id|>assistant<|end_header_id|>
{{ model_answer_1 }}<|eot_id|>
```

In this notation, special tokens are used to denote the boundaries and roles within the conversation. The <|begin_of_text|> token marks the start of a new data sample. The <|start_header_id|> and <|end_header_id|> tokens enclose identifiers specifying the role of the content that follows (e.g., system, user, or assistant). The {{ system_prompt }}, {{ user_msg_1 }}, and {{ model_answer_1 }} placeholders are replaced with the actual system instruction, user input, and target assistant output, respectively. The <|eot_id|> token indicates the end of each segment or turn within the conversation. This structured formatting enables the model to clearly distinguish between instructions, queries, and responses, thereby enhancing learning effectiveness during fine-tuning.

LoRA and QLoRA settings

LoRA is a way to efficiently fine-tune LLMs by representing the Matrix of weights as a multiplication of 2 matrices with lower dimensions. The key element here is Matrix Rank (r), which affects the number of trainable parameters.

With the current matrix rank, 176 million parameters, which is 3.73% of all LLaMA-3.1 parameters. For the model, fine-tuning was used with LoraConfig (Table 1).

Table 1. **Lora Config for LLMs fine-tuning**

Parameter	Parameter description	Value
lora_alpha	LoRA scaling factor	16
lora_dropout	Dropout parameter to reduce overfitting	0.1
r	Matrix rank relates to the number of trainable parameters	64

For efficient comparison of LLaMA-3.1 with two previously trained models, those parameters were the same for all 3 models.

Training parameters

A model was tuned during 2 epochs. Considering that the base model captures lots of dependencies, a large number of epochs might cause overfitting [27]. Batch size is a parameter that represents the number of samples in the batch for training. We noticed that batch size can be increased for faster training, but compared to LLaMA-3.1 with previous experiments, we decided to use 4 as batch size [28]. The next parameters, as warmup_step (a way to reduce the primacy effect of the early training examples) [29], learning rate (indicate how fast a model could train) [30], 16-bit floating point format (represents QLoRA, quantization that helps to reduce the size of the model) [31].

LLaMA-3.1 could handle very big contexts (up to 128k tokens) [32], but a value of 1024 was selected to compare the current model with previous experiments. Training parameters can be found in Table 2.

Evaluation

The evaluation was done as in the previous experiment. To check how well the models worked, we used a test dataset that wasn't involved during the fine-tuning phase. First, we sent instructions, expected answers, and actual answers to the GPT-4 model, which then

Table 2. **Training parameters for LLMs fine-tuning**

Parameter	Parameter description	Value
num_train_epochs	Number of training epochs	2
per_device_train_batch_size	Batch size	4
warmup_steps	The number of warm-out steps	0.03
bf16	16-bit floating point format	True
max_seq_length	Max number of tokens	1024
learning_rate	Learning rate	2.5×10^{-5}

gave a score out of 10, with a higher score meaning better compliance with the instructions [24]. Second, we used the RAGAS [33] library to evaluate the models using two measures: Answer Correctness and Answer Semantic Similarity. You can access the RAGAS library via this link: https://docs.ragas.io/en/stable/.

RESULTS AND DISCUSSION

Training and validation loss comparison

Loss comparison is important to measure the efficiency of training or fine-tuning. For our experiments, we used the cross-entropy loss function [34], which is standard for tasks related to content generation. If loss decreases, that means that the model can train, capture patterns, and be more efficient in solving tasks related to the training dataset. We have noticed that during training, losses decreased from 1.09 to 1.06 for the LLaMA 3.1 model. It might be a good indicator. Train loss comparison can be found in Table 3.

Table 3. **Train loss comparison**

epoch	LLaMA-2	Mixtral	LLaMA-3.1
1	1.1736	1.0665	1.0887
2	1.1175	1.0723	1.0644

The most important is the validation loss. It's a value of the loss function on the dataset that was not used for training. It helps us to measure how good a model might be regarding data it had not seen previously. We noticed that validation losses also decreased (Table 4), which might be a good indicator too.

Table 4. **Validation loss comparison**

epoch	LLaMA-2	Mixtral	LLaMA-3.1
1	1.1474	1.0692	1.0756
2	1.1378	1.0626	1.0692

Training time comparison

A model was fine-tuned on an NVIDIA A100 Tensor Core GPU. Tuning LLMs requires a significant number of resources. Therefore, training time is also important as it impacts the cost of the solution.

We notice that LLaMA-3.1 requires 3 times more time for training than LLaMA-2 and 2.5 hours more than Mixtral (Table 5). Some techniques can significantly decrease training time, like Unsloth [35]. However, to ensure the fairness of the experiments and compatibility with previous research, it was decided to avoid current techniques.

Table 5. **Training time comparison**

Model	LLaMA-2	Mixtral	LLaMA-3.1
Training time, hours: mins	3:27	7:36	9:55

Metrics comparison

The key element for comparison between different Machine Learning algorithms is metric comparison. Evaluation was done on the testing dataset – a dataset that was not used for fine-tuning.

The evaluation of base LLaMA-3.1 against a fine-tuned model and models tuned in the previous paper demonstrates the importance of fine-tuning for specific tasks [24]. Three core metrics are used in this assessment: GPT-4 score Answer Correctness, and Answer Semantic Similarity [24]. The GPT-4 score is an automated evaluation metric in which the GPT-4 model is provided with the system instructions, user message, and both the golden (expected) and actual answers. Based on this information, GPT-4 assigns a score from 1 (worst) to 10 (best) that reflects how well the actual answer matches the golden one. Notably, each metric has its limitations, with Answer Semantic Similarity perhaps less suitable for specialized instruction tasks that may require knowledge from fields like physics or mathematics.

For comparison, the LLAMA-3.1 base outperforms other models, and the LLaMA-3.1 is fine-tuned (Table 6, Fig. 1).

That means that with fine-tuning, we overfit a model. Therefore, it can efficiently solve tasks related to training datasets and underperform on data it has not seen.

We noticed that LLaMA-3.1, base overperformed all previous models for all 3 metrics. Consequently, for the current task with the instructions dataset, the base model should be used.

Discussion and future direction

An empirical analysis of the LLaMA-2, LLaMA-3.1, and Mixtral models highlights their effectiveness in executing instructional tasks. Findings show that LLaMA-2 and LLaMA-3 tend to overfit. On the other hand, Mixtral outperforms LLaMA-2 during the evaluation phase, making it a more suitable option for instructional tasks. The key finding is that LLaMA-3.1 does not need fine-tuning to efficiently follow instructions. The base model works significantly better than the fine-tuned model.

We understand that for some domain-specific tasks, fine-tuning might be essential. General models were trained on a large amount of data, but can not know everything regarding specific domains. Also, often, those domains have some sensitive data, which makes it impossible to use models via API (like GPT-4 or Claude). Efficient fine-tuning is important for those domains.

Tuning models for RAG is also important, as the tuned model can answer user questions much better than the base model.

Table 6. **Metrics comparison**

Model	GPT-4 score (max 10)	Answer Correctness	Answer Semantic Similarity
LLaMA-2, base	7.21	0.66	0.91
LLaMA-2, fine-tuned	6.96	0.63	0.91
Mixtral, base	7.12	0.62	0.91
Mixtral, fine-tuned	7.51	0.67	0.91
LLaMA-3.1, base	8.54	0.72	0.92
LLaMA-3.1, fine-tuned	6.79	0.61	0.90

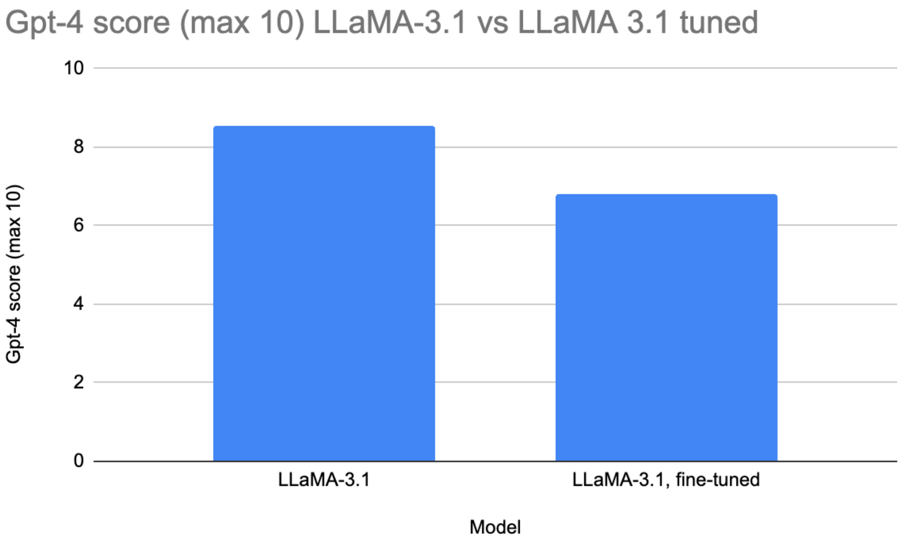

Fig. 1. Metrics comparison

We are going to test those approaches in different specific areas, not just teaching tasks or following instructions. We'll work to improve our methods and the way we measure performance in this study. This work might help us make better language models and push forward the field of language processing. It's also important to note that we're going to work a lot on getting better at the 'reasoning' part in future studies.

CONCLUSION

After an investigation of different Large Language Models (LLMs), particularly LLaMA-3.1, LLaMA-2, and Mixtral, our research has yielded interesting insights. Initially, Mixtral showed how great an impact fine-tuning can have on model performance. Similarly, we tried similar techniques on LLaMA-3.1, including cutting-edge methods like LoRA and QLoRA.

When fine-tuned with an instruction task, the performance of LLaMA-3.1 took a hit. LLaMA-3.1, in its basic form, is already very good at solving instruction tasks, and the addition of fine-tuning brought on too much specialization, leading to overfitting. Despite this, we believe that for some particular tasks, especially within specific domains, fine-tuning might still be necessary to achieve enhanced performance.

Upon comparing LLaMA-2 and Mixtrail, we observed that LLaMA-2 fine-tuned faster but was more susceptible to overfitting. Mixtrail, although slower in training, proved to be consistently better at handling instructional tasks in our tests, suggesting it has a better balance between general language skills and specificity.

This tendency towards overfitting, observed in the LLaMA family, helps us understand the importance of carefully managing the tuning process. We recommend cutting down on tuning epochs to prevent overfitting.

Our research leveraged the RAGAS library to evaluate LLM performance, a practice we think would be instrumental in future machine learning studies where LLMs are used. Our conclusions provide crucial learning about the workings of LLaMA-3.1 and LLMs in general, including their performance, fine-tuning practices, and predisposition towards overfitting.

Acquired knowledge opens the gate for future studies on LLMs, which we believe is potentially pivotal for unlocking their full capability, particularly concerning specific tasks and domains. We're also reminded of the critical part fine-tuning plays in amplifying LLM performance and the necessity of finding a good balance between general language competencies and specific task efficiency, especially in domain-specific tasks.

AUTHOR CONTRIBUTIONS

Conceptualization, [*B.P., I.B.*]; methodology, [*I.B.*]; validation, [*B.P.*]; formal analysis, [*I.B.*.].; investigation, [*I.B.*]; resources, [*I.B.*]; data curation, [*I.B.*]; writing – original draft preparation, [*I.B.*]; writing – review and editing, [*B.P.*]; visualization, [*I.B.*] supervision, [*B.P.*]; project administration, [*I.B.*].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[2] De Leon, Francisco, Pablo Gómez, Juan A. Martinez-Velasco, and Michel Rioual. "Transformers." In Power system transients, pp. 177-250. CRC Press, 2017

[3] Rogers, A., Kovaleva, O., & Rumshisky, A. (2021). A primer in BERTology: What we know about how BERT works. *Transactions of the association for computational linguistics*, *8*, 842-866. https://doi.org/10.1162/tacl_a_00349

[4] Hao, Y., Dong, L., Wei, F., & Xu, K. (2019). Visualizing and understanding the effectiveness of BERT. *arXiv preprint arXiv:1908.05620*. https://doi.org/10.48550/arXiv.1908.05620

[5] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*. https://doi.org/10.48550/arXiv.2303.08774

[6 Liga, D., & Robaldo, L. (2023). Fine-tuning GPT-3 for legal rule classification. *Computer Law & Security Review*, *51*, 105864. https://doi.org/10.1016/j.clsr.2023.105864

[7] Hendy, A., Abdelrehim, M., Sharaf, A., Raunak, V., Gabr, M., Matsushita, H., ... & Awadalla, H. H. (2023). How good are gpt models at machine translation? a comprehensive evaluation. *arXiv preprint arXiv:2302.09210*. https://doi.org/10.48550/arXiv.2302.09210

[8] Goyal, T., Li, J. J., & Durrett, G. (2022). News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*. https://doi.org/10.48550/arXiv.2209.12356

[9] Koubaa, A. (2023). GPT-4 vs. GPT-3.5: A concise showdown. *Preprints. org*, *2023030422*.

[10] Mao, R., Chen, G., Zhang, X., Guerin, F., & Cambria, E. (2023). GPTEval: A survey on assessments of ChatGPT and GPT-4. *arXiv preprint arXiv:2308.12488*. https://doi.org/10.48550/arXiv.2308.12488

[11] Adetayo, A. J., Aborisade, M. O., & Sanni, B. A. (2024). Microsoft Copilot and Anthropic Claude AI in education and library service. *Library Hi Tech News*. https://doi.org/10.1108/LHTN-01-2024-0002

[12] Pavlyshenko, B. M. (2023). Analysis of disinformation and fake news detection using fine-tuned large language model. *arXiv preprint arXiv:2309.04704*. https://doi.org/10.48550/arXiv.2309.04704

[13] Pavlyshenko, B. M. (2023). Financial news analytics using fine-tuned llama 2 gpt model. *arXiv preprint arXiv:2308.13032*. https://doi.org/10.48550/arXiv.2308.13032

[14] Pavlyshenko, B. M. (2022). Methods of informational trends analytics and fake news detection on twitter. *arXiv preprint arXiv:2204.04891*. https://doi.org/10.48550/arXiv.2204.04891

[15] Prottasha, N. J., Sami, A. A., Kowsher, M., Murad, S. A., Bairagi, A. K., Masud, M., & Baz, M. (2022). Transfer learning for sentiment analysis using BERT based supervised fine-tuning. *Sensors*, *22*(11), 4157. https://doi.org/10.3390/s22114157

[16] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, *33*, 9459-9474.

[17] Lin, X., Wang, W., Li, Y., Yang, S., Feng, F., Wei, Y., & Chua, T. S. (2024, July). Data-efficient Fine-tuning for LLM-based Recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information Retrieval* (pp. 365-374). https://doi.org/10.1145/3626772.365780

[18] Meier, R. (2024). Llm-aided social media influence operations. *Large Language Models in Cybersecurity: Threats, Exposure and Mitigation*, 105-112. https://doi.org/10.1007/978-3-031-54827-7_11

[19] Yu, X., Chen, Z., Ling, Y., Dong, S., Liu, Z., & Lu, Y. (2023). Temporal data meets LLM--explainable financial time series forecasting. *arXiv preprint arXiv:2306.11025*. https://doi.org/10.48550/arXiv.2306.11025

[20] Vavekanand, R., & Sam, K. (2024). Llama 3.1: An in-depth analysis of the next-generation large language model. https://doi.org/10.48550/arXiv.2306.11025

[21] Li, Y., Yu, Y., Liang, C., He, P., Karampatziakis, N., Chen, W., & Zhao, T. (2023). Loftq: Lora-fine-tuning-aware quantization for large language models. *arXiv preprint arXiv:2310.08659*. https://doi.org/10.48550/arXiv.2310.08659

[22] Wu, B., Zhu, R., Zhang, Z., Sun, P., Liu, X., & Jin, X. (2024). {dLoRA}: Dynamically orchestrating requests and adapters for {LoRA}{LLM} serving. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)* (pp. 911-927).

[23] Zhang, X., Rajabi, N., Duh, K., & Koehn, P. (2023, December). Machine translation with large language models: Prompting, few-shot learning, and fine-tuning with QLoRA. In Proceedings of the Eighth Conference on Machine Translation (pp. 468-481).

[24] Pavlyshenko, B., & Bulka, I. (2024) Metric-Based Comparison of Fine-Tuned LLAMA 2 and MIXTRAL Large Language Models for Instruction Tasks. *Electronics and information technologies/Електроніка та інформаційні технології*, 26, 16-24. http://dx.doi.org/10.30970/eli.26.2

[25] MosaicML. (2023). MosaicML Instruct-v3 Dataset. Hugging Face. https://huggingface.co/datasets/mosaicml/instruct-v3

[26] Rohan Taori and Ishaan Gulrajani and Tianyi Zhang and Yann Dubois and Xuechen Li and Carlos Guestrin and Percy Liang and Tatsunori B. Hashimoto. (2023). Stanford Alpaca: An Instruction-following LLaMA model. Hugging Face. https://huggingface.co/datasets/tatsu-lab/alpaca

[27] Ying, X. (2019, February). An overview of overfitting and its solutions. In *Journal of physics: Conference series* (Vol. 1168, p. 022022). IOP Publishing. https://doi.org/10.1088/1742-6596/1168/2/022022

[28] You, Y., Wang, Y., Zhang, H., Zhang, Z., Demmel, J., & Hsieh, C. J. (2020). The limit of the batch size. *arXiv preprint arXiv:2006.08517*. https://doi.org/10.48550/arXiv.2006.08517

[29] Kang, F., Just, H. A., Sun, Y., Jahagirdar, H., Zhang, Y., Du, R., ... & Jia, R. (2024). Get more for less: Principled data selection for warming up fine-tuning in llms. *arXiv preprint arXiv:2405.02774*. https://doi.org/10.48550/arXiv.2405.02774

[30] Sanyal, S., Neerkaje, A., Kaddour, J., Kumar, A., & Sanghavi, S. (2023). Early weight averaging meets high learning rates for llm pre-training. *arXiv preprint arXiv:2306.03241*. https://doi.org/10.48550/arXiv.2306.03241

[31] Agrawal, A., Mueller, S. M., Fleischer, B. M., Sun, X., Wang, N., Choi, J., & Gopalakrishnan, K. (2019, June). DLFloat: A 16-b floating point format designed for

deep learning training and inference. In *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)* (pp. 92-95). IEEE. https://doi.org/10.1109/ARITH.2019.00023

[32] Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., ... & Vasic, P. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*. https://doi.org/10.48550/arXiv.2407.21783

[33] Es, S., James, J., Anke, L. E., & Schockaert, S. (2024, March). Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations* (pp. 150-158).

[34] VM, K., Warrier, H., & Gupta, Y. (2024). Fine tuning llm for enterprise: Practical guidelines and recommendations. *arXiv preprint arXiv:2404.10779*. https://doi.org/10.48550/arXiv.2404.10779

[35] Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., Feng, Z., & Ma, Y. (2024). Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*. https://doi.org/10.48550/arXiv.2403.13372

ЕФЕКТИВНЕ ТОЧНЕ НАЛАШТУВАННЯ ПАРАМЕТРІВ ТА ПЕРЕНАВЧАННЯ В МОВНИХ МОДЕЛЯХ GPT: ПОРІВНЯННЯ НА ОСНОВІ МЕТРИК

Богдан Павлишенко, Іван Булка*
Львівський національний університет імені Івана Франка,
вул. Драгоманова 50, 79005 Львів, Україна

АНОТАЦІЯ

Вступ. Спираючись на попередні дослідження, це дослідження зосереджується на великих мовних моделях (LLM) з фокусом на тонкому налаштуванні та оцінці LLaMA-3.1 для завдань зв'язаних з інструкціями. LLaMA-3.1, яка є моделлю нового покоління і здобула значне визнання завдяки своїм чудовим результатам. Окрім оцінки відмінностей і вдосконалень між базовою та налаштованою версіями LLaMA-3.1 на наборі даних інструкцій, дослідження також звертає увагу на проблему перенавчання LLaMA-3.1. Додатково було проведено порівнянням між LLaMA-3.1, її попередницею, LLaMA-2, а також іншою LLM, відомою як Mixtral, що дозволяє отримати більш повну картину можливостей LLaMA-3.1.

Матеріали та методи. Для тонкого налаштування LLaMA-3.1 використовувались сучасні підходи, такі як адаптація низького рангу (LoRA) і квантована адаптація низького рангу (QLoRA), на комплексних наборах даних інструкцій. Враховуючи ресурсоємність процесу тонкого налаштування LLM, вживались заходи щодо його оптимізації. Процес тонкого налаштування був удосконалений за допомогою Параметрично ефективного тонкого налаштування (PEFT) на екземплярах NVIDIA A100 Tensor Core GPU. Усі моделі були налаштовані за допомогою платформ Hugging Face і PyTorch для досягнення найкращої продуктивності. Дослідження підкреслює важливість ретельної оцінки LLM для практичних застосувань.

Результати. Результати, отримані в результаті тонкого налаштування та оцінки LLaMA-3.1, надали цінну інформацію про те, як ця модель виконує конкретні завдання. Система оцінювання виявилася корисною для ефективної оцінки ефективності LLM на завданнях з інструкціями. Показано, що точне налаштування не завжди є найкращим вибором з огляду на специфіку моделі та особливості завдання. Дослідження підкреслює проблему перенавчання в LLM.

Висновки. Ретельний аналіз LLaMA-3.1 робить внесок у сферу машинного навчання, поглиблюючи розуміння особливостей роботи цієї моделі та можливостей її тонкого налаштування для конкретних завдань. Результати цього дослідження створюють підгрунтя для подальших досліджень і застосування LLMs та підкреслюють значення ефективної оцінки з використанням існуючих метрик.

Ключові слова: LLMs, GPT, Mixtral, LLaMA, тонке налаштування, перенавчання.

Received / Одержано	Revision / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
18 April, 2025	02 May, 2025	08 May, 2025	26 June, 2025