Federated Black-box Prompt Tuning System for Large Language Models on the Edge

Yiming Li[1*], Jingwei Sun[1*], Yudong Liu[1], Yuandong Zhang[1], Ang Li[2], Beidi Chen[3], Holger R. Roth[4], Daguang Xu[4], Tingjun Chen[1], Yiran Chen[1]

[1]Duke University [2]University of Maryland, College Park

[3]Carnegie Mellon University [4]NVIDIA

Abstract

Federated learning (FL) offers a privacy-preserving way to train models across decentralized data. However, fine-tuning pre-trained language models (PLMs) in FL is challenging due to restricted model parameter access, high computational demands, and communication overheads. Our method treats large language models (LLMs) as black-box inference APIs, optimizing prompts with gradient-free methods. This approach, FedBPT, reduces exchanged variables, boosts communication efficiency, and minimizes computational and memory costs. We demonstrate the practical implementation of FedBPT on resource-limited edge devices, showcasing its ability to efficiently achieve collaborative on-device LLM fine-tuning.

CCS Concepts

• **Computing methodologies** → **Machine learning**; • **Human-centered computing** → **Ubiquitous and mobile computing**.

Keywords

Large language models, Gradient-free optimization, Federated Learning

ACM Reference Format:
Yiming Li[1] [1], Jingwei Sun[1] [1], Yudong Liu[1], Yuandong Zhang[1], Ang Li[2], Beidi Chen[3], Holger R. Roth[4], Daguang Xu[4], Tingjun Chen[1], Yiran Chen[1]. 2024. Federated Black-box Prompt Tuning System for Large Language Models on the Edge. In *The 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24), November 18–22, 2024, Washington D.C., DC, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3636534.3698856

1 Introduction

Large Language Models (LLMs) [1, 3, 4, 8, 9, 11, 14, 15] have revolutionized Natural Language Processing (NLP) by achieving high accuracy across tasks. Typically pre-trained on diverse datasets, they are fine-tuned with task-specific data for real-world applications[12]. However, fine-tuning often involves sensitive user data, raising critical privacy and security concerns[7, 13]. Federated Learning (FL) enables decentralized, collaborative LLM fine-tuning while preserving data privacy[2, 6]. However, deploying FL for LLMs on edge devices faces key challenges, such as limited access to model parameters, high computational demands on local clients, and excessive communication overhead, hindering its practical use.

To tackle these challenges, we proposed FedBPT (**Fed**erated **B**lack-box **P**rompt **T**uning) [10], a framework that uses gradient-free optimization to train prompts, drastically lowering resource demands on edge devices. FedBPT enhances privacy while reducing communication and computational overhead, enabling efficient LLM adaptation across distributed devices. This demo paper builds upon our original FedBPT framework, focusing on the real-world implementation of a system based on the FedBPT algorithm across various edge devices.

To the best of our knowledge, we are the first to implement on-device finetuning of LLMs at the scale of Llama2-7B on a mobile phone and Raspberry Pi. Furthermore, this is also the first-ever successful attempt to customize Llama.cpp to realize distributed LLM tuning without backpropagation.

2 System Design

Figure 1 provides an overview of the system, including three key modules, with cross-device communication managed via HTTP. Originally, FedBPT was implemented entirely in a Python environment, where edge devices are simulated by Python class objects, with LLM inference performed in PyTorch and communication between the FL server and edge device handled via Python function calls. To implement it with the real edge device, we need to run efficient LLM inference locally on the edge device and provide a communication solution between the edge devices (clients) and the server.

[*]Both authors contributed equally to this work.

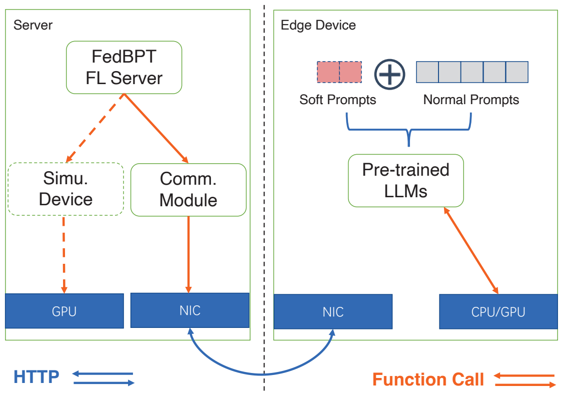

Figure 1: Overview of the implementation of FedBPT [10] on edge devices.

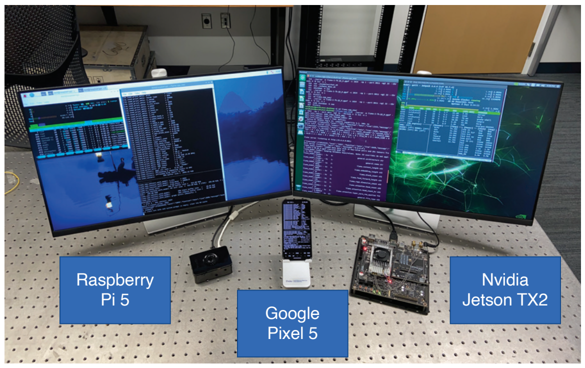

Figure 2: Devices with different form factors and computing capabilities used for the demonstration: Raspberry Pi 5 (8GB), Google Pixel 5, and Nvidia Jetson TX2.

LLM inference engine for various edge devices. To insert a soft prompt into the embedding token and obtain the raw results, we leverage the `llama.cpp` library (a C++ library designed for fast and efficient inference of LLMs) [5] and customize the interface to insert the FedBPT algorithm.

For any given input, we first perform the usual tokenizing and embedding steps and then concatenate the soft prompt to the front of the embedded token, which is the output of the embedding layer. The modified embedded token then proceeds through the normal inference procedures to generate the final output.

`llama.cpp` provides a well-abstracted inference process. It first generates the computational graph and then applies it to any supported computing backend, including CPU and various acceleration frameworks like CUDA, Metal, and Vulkan. To maintain maximum compatibility with the official `llama.cpp`, we implement our soft prompt support function during the compute graph generation step.

HTTP-based FL communication. We introduce an adapter class into the original FedBPT code, which functions as an HTTP client using the Requests library to redirect all communication to the actual edge devices. On the edge device side, we reused the `llama.cpp` server mode and modified the source code to accept customized FL communication.

3 Experiment Setup

We select three representative edge devices to illustrate how diverse and heterogeneous device can effectively contribute to the collaborative fine-tuning of LLM without direct access to model parameters: i) Google Pixel 5: This device exemplifies a standard Android smartphone with limited computational power. ii) Raspberry Pi 5 (8GB): Serving as a typical edge device. iii) NVIDIA Jetson TX2: This device represents an advanced edge unit equipped with a dedicated acceleration unit.FedbPT has already demonstrated its ability to fine-tune various LLMs. Given the uneven and limited computational resources of the three selected devices, we chose the AG's News dataset and Llama2 7b q2 model, which maintains a compact size while still achieving good accuracy.

4 Demonstrations

We will provide a real-time demonstration of fine-tuning LLMs on edge devices, during which the users can interact with the edge devices and compare the performance of the fine-tuned model against the original model.

Real-time fine-tuning of LLM on edge device. This real-time demonstration shows the real-time fine-tuning of LLMs on edge devices. By observing the process, we can see that the accuracy metrics improve significantly. Additionally, we will demonstrate the significant reduction in GPU memory consumption and communication costs achieved by the FedBPT algorithm during the on-device optimization process.

Comparison between the fine-tuned model and the original model. The fine-tuning process can be lengthy, so we have prepared a fine-tuned model for immediate use. This allows users to compare the difference between the fine-tuned model and the original model. Users can select from predefined questions or input their questions to see the performance variations between the two models.

5 Acknowledgement

This material is based upon work supported by the U.S. National Science Foundation under award CNS-2112562. This work is also supported by ARO W911NF-23-2-0224. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. National Science Foundation, ARO, and their contractors.

References

[1] 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 (2020), 1877–1901.

[2] Hong-You Chen, Cheng-Hao Tu, Ziwei Li, Han-Wei Shen, and Wei-Lun Chao. 2022. On the importance and applicability of pre-training for federated learning. *arXiv preprint arXiv:2206.11488* (2022).

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[4] William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research* 23, 1 (2022), 5232–5270.

[5] llama.cpp. 2023. llama.cpp. https://github.com/ggerganov/llama.cpp.

[6] John Nguyen, Jianyu Wang, Kshitiz Malik, Maziar Sanjabi, and Michael Rabbat. 2022. Where to Begin? On the Impact of Pre-Training and Initialization in Federated Learning. *arXiv preprint arXiv:2210.08090* (2022).

[7] Charith Peris, Christophe Dupuy, Jimit Majmudar, Rahil Parikh, Sami Smaili, Richard Zemel, and Rahul Gupta. 2023. Privacy in the time of language models. In *Proceedings of the sixteenth ACM international conference on web search and data mining*. 1291–1292.

[8] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences* 63, 10 (2020), 1872–1897.

[9] 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. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[10] Jingwei Sun, Ziyue Xu, Hongxu Yin, Dong Yang, Daguang Xu, Yiran Chen, and Holger R Roth. 2023. Fedbpt: Efficient federated black-box prompt tuning for large language models. *arXiv preprint arXiv:2310.01467* (2023).

[11] Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137* (2021).

[12] Kushala VM, Harikrishna Warrier, Yogesh Gupta, et al. 2024. Fine Tuning LLM for Enterprise: Practical Guidelines and Recommendations. *arXiv preprint arXiv:2404.10779* (2024).

[13] Jeffrey G Wang, Jason Wang, Marvin Li, and Seth Neel. 2024. Pandora's White-Box: Increased Training Data Leakage in Open LLMs. *arXiv preprint arXiv:2402.17012* (2024).

[14] Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, ZhenZhang Yang, Kaisheng Wang, Xiaoda Zhang, et al. 2021. Pangu-α: Large-scale autoregressive pretrained Chinese language models with auto-parallel computation. *arXiv preprint arXiv:2104.12369* (2021).

[15] Zhengyan Zhang, Xu Han, Hao Zhou, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, Yusheng Su, Haozhe Ji, Jian Guan, et al. 2021. CPM: A large-scale generative Chinese pre-trained language model. *AI Open* 2 (2021), 93–99.