Federated Learning with Heterogeneous LLMs: Integrating Small Student Client Models with a Large Hungry Model

Gautam Jajoo

BITS Pilani
f20201638@pilani.bits-pilani.ac.in

Abstract

In recent years, federated learning (FL) has emerged as a promising technique to enable decentralized training of models without the need for data centralization, addressing privacy concerns and reducing communication overhead. The challenge, however, lies in scaling federated systems to accommodate clients with different computational capabilities. The heterogeneity of clients in terms of data, model structures, and computational resources presents significant challenges. Addressing these challenges can lead to more robust and efficient FL systems, making it possible to leverage diverse data sources and computational environments. Here we propose a system where small language models run on heterogeneous clients while a large, more powerful model at the server aggregates their contributions. This architecture leverages the strengths of both small, task-specific client models and a large server model to enhance generalization and efficiency. This is important because it addresses the growing need for scalable, privacy-preserving systems that can operate in diverse environments with varying resources. Through such a system we intend to contribute to the AI field by improving the efficiency of federated learning systems while enhancing their adaptability to real-world applications.

Background

Federated learning has been widely studied as a method for distributed learning without the need to centralize data, thus preserving privacy (Kairouz et al. 2021). Most FL systems assume homogeneous clients or models with comparable architectures. However, real-world applications often require accommodating clients with diverse computational resources (e.g., mobile phones vs. cloud servers). Previous works, such as FedAvg (McMahan et al. 2023), have laid the groundwork for parameter aggregation in federated learning, yet most focus on homogeneous models across clients which are not practical for large-scale applications (Ye et al. 2023). Other studies like FedProx (Li et al. 2020) explore how to handle heterogeneity among clients, but none have effectively tackled the integration of models with vastly different sizes and architectures.

Heterogeneous federated learning (HFL) addresses this by considering differences in data distributions, model structures, and computational resources among clients (Chen et al. 2024). The proposed work builds on this existing research by introducing the novel concept of using small, task-specific LLMs at the client level (e.g., bloom-560m) while aggregating them into a much larger, more general model (e.g., bloom-7b1) at the central server. This approach ensures that smaller models can operate effectively in resource-constrained environments, while the larger model benefits from diverse task-specific knowledge.

Methods such as Federated Knowledge Distillation (Seo et al. 2020) have been proposed to transfer knowledge from teacher models to student models using logits or soft labels. However, these approaches typically require access to a public dataset to facilitate the knowledge transfer process, which may not be optimal for the true purpose of a federated setup.

Approach

The proposed system will consist of a system of heterogeneous clients, each running a small language model which is lightweight, allowing them to operate efficiently in resource-constrained environments, and a central server hosting a large model. The clients will fine-tune their models based on the specific tasks that they are performing, using localized data. Periodically, these clients will send their model updates (not raw data) to the central server. The server will then use an aggregation mechanism (e.g., gradient aggregation or knowledge distillation) to update the large model based on the insights from the smaller, task-specific models. Since the client models are significantly smaller than the server model, a simple gradient average may not suffice. Instead, we investigate different approaches to tackle the problem.

Each small LLM will generate task vectors (Ilharco et al. 2023), which will capture the direction and magnitude of task-specific updates in its weight space. These vectors will encapsulate how the local model needs to be updated in order to specialize in its specific task. Task vectors can be computed as the difference between the pre-trained base model weights and the fine-tuned task-specific weights. These task vectors encode:

- Direction: The type of changes required (e.g., emphasizing certain parameters or sub-networks)
- Magnitude: The intensity of changes (e.g., how much the model adapts to the local task)

The server model operates in a higher-dimensional weight space due to its larger capacity. Task vectors need to be projected or mapped into this larger space while preserving their core characteristics. A possible technique for mapping task vectors is linear projection where we will use learned linear transformation matrices to map smaller model updates to the larger model's space.

Task vectors from all clients are aggregated at the server. Aggregation will be weighted by the quality of the task-specific update and the data heterogeneity of the client's local dataset. The server update will look like:

$$W_{\text{server}}^{t+1} = W_{\text{server}}^{t} + \eta \frac{\sum_{i=1}^{n} \alpha_i \cdot \text{task\_vector}_i}{\sum_{i=1}^{n} \alpha_i}$$

Here, α_i are the aggregation weights, and η is the learning rate.

We can also explore the use of parameter interpolation methods that align the parameter spaces of different models, ensuring smoother updates. To account for heterogeneity in model sizes and architectures, we will adopt layer-wise adaptive learning rates (LARS) (You, Gitman, and Ginsburg 2017). This optimization technique adjusts the learning rate for each layer of the model based on the magnitude of its weight, ensuring that smaller models can contribute effectively.

Since client models will be trained in specific tasks, continual learning techniques will be crucial to avoid catastrophic forgetting when they contribute to the large model.

Overall, this problem can be looked at, in a way which involves using a hybrid of federated learning techniques and knowledge distillation. The small models on the client side will act as specialists, learning in-depth on specific tasks, while the large server model acts as a generalist, learning across many tasks to generalize better.

Evaluation

The evaluation will be conducted on a variety of publicly available datasets that are widely used in FL and NLP, ensuring both diversity in the tasks and robustness in the model's performance. The task type, dataset and the metrics are described in Table 1.

Task Type	**Dataset**	**Metric**
Summarization	XL-Sum (Hasan et al. 2021)	ROUGE
Translation	IN22, FLORES (Gala et al. 2023)	BLEU, chrF
Hate Speech Classification	Ethos (Mollas et al. 2022)	Accuracy
Reasoning	COPA (Brassard et al. 2022)	Accuracy
Federated Learning Metrics		Communication Efficiency, Overload, Resource Utilization

Table 1: Evaluation benchmark and metrics

Discussion

We expect to find that using heterogeneous clients in a federated learning setup, each with a smaller model, can effectively contribute to a centralized, larger model which would demonstrate an improved performance. It should also achieve a balance between global model performance and local personalization, making it suitable for diverse applications. This research will provide a robust framework for federated learning in heterogeneous environments, paving the way for more inclusive and efficient AI model training. The implications of this approach of working would be high, especially in expanding federated learning into more diverse environments where computational resources vary widely. Moreover, the research could lead to more privacy-preserving models that require minimal communication of sensitive data. For society, this would mean more secure AI applications in domains like healthcare, education, and finance, where maintaining privacy while ensuring accurate predictions is crucial.

Conclusion

The primary research challenge lies in designing a complex and robust aggregation mechanism capable of effectively updating the larger model using the weights of heterogeneous, smaller models. Traditional aggregation methods, such as FedAvg, or a simple addition of task vectors, are effective when dealing with models of similar size and architecture. However, when multiple smaller models, each fine-tuned on specific tasks, contribute to a much larger central model, this straightforward approach is insufficient. There is a need for an advanced aggregation strategy that can reconcile differences in model sizes and parameters. Different techniques such as gradient-based aggregation, knowledge distillation, and parameter interpolation will be explored to achieve a balance between task specialization and generalization.

Acknowledgments

We sincerely thank Pranjal Chitale for the insightful discussions which helped us to organize our ideas more effectively.

References

Brassard, A.; Heinzerling, B.; Kavumba, P.; and Inui, K. 2022. COPA-SSE: Semi-structured Explanations for Commonsense Reasoning. arXiv:2201.06777.

Chen, C.; Liao, T.; Deng, X.; Wu, Z.; Huang, S.; and Zheng, Z. 2024. Advances in Robust Federated Learning: Heterogeneity Considerations. arXiv:2405.09839.

Gala, J.; Chitale, P. A.; AK, R.; Gumma, V.; Doddapaneni, S.; Kumar, A.; Nawale, J.; Sujatha, A.; Puduppully, R.; Raghavan, V.; Kumar, P.; Khapra, M. M.; Dabre, R.;

and Kunchukuttan, A. 2023. IndicTrans2: Towards High-Quality and Accessible Machine Translation Models for all 22 Scheduled Indian Languages. arXiv:2305.16307.

Hasan, T.; Bhattacharjee, A.; Islam, M. S.; Samin, K.; Li, Y.-F.; Kang, Y.-B.; Rahman, M. S.; and Shahriyar, R. 2021. XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages. arXiv:2106.13822.

Ilharco, G.; Ribeiro, M. T.; Wortsman, M.; Gururangan, S.; Schmidt, L.; Hajishirzi, H.; and Farhadi, A. 2023. Editing Models with Task Arithmetic. arXiv:2212.04089.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; D'Oliveira, R. G. L.; Eichner, H.; Rouayheb, S. E.; Evans, D.; Gardner, J.; Garrett, Z.; Gascón, A.; Ghazi, B.; Gibbons, P. B.; Gruteser, M.; Harchaoui, Z.; He, C.; He, L.; Huo, Z.; Hutchinson, B.; Hsu, J.; Jaggi, M.; Javidi, T.; Joshi, G.; Khodak, M.; Konečný, J.; Korolova, A.; Koushanfar, F.; Koyejo, S.; Lepoint, T.; Liu, Y.; Mittal, P.; Mohri, M.; Nock, R.; Özgür, A.; Pagh, R.; Raykova, M.; Qi, H.; Ramage, D.; Raskar, R.; Song, D.; Song, W.; Stich, S. U.; Sun, Z.; Suresh, A. T.; Tramèr, F.; Vepakomma, P.; Wang, J.; Xiong, L.; Xu, Z.; Yang, Q.; Yu, F. X.; Yu, H.; and Zhao, S. 2021. Advances and Open Problems in Federated Learning. arXiv:1912.04977.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated Optimization in Heterogeneous Networks. arXiv:1812.06127.

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2023. Communication-Efficient Learning of Deep Networks from Decentralized Data. arXiv:1602.05629.

Mollas, I.; Chrysopoulou, Z.; Karlos, S.; and Tsoumakas, G. 2022. ETHOS: a multi-label hate speech detection dataset. *Complex amp; Intelligent Systems*, 8(6): 4663–4678.

Seo, H.; Park, J.; Oh, S.; Bennis, M.; and Kim, S.-L. 2020. Federated Knowledge Distillation. arXiv:2011.02367.

Ye, M.; Fang, X.; Du, B.; Yuen, P. C.; and Tao, D. 2023. Heterogeneous Federated Learning: State-of-the-art and Research Challenges. arXiv:2307.10616.

You, Y.; Gitman, I.; and Ginsburg, B. 2017. Large Batch Training of Convolutional Networks. arXiv:1708.03888.