Hierarchical Learning: A Hybrid of Federated Learning and Personalization Fine-Tuning

Shuyi Li[1*] and *Bairong* Zhang[2]

[1]Computer Science, University of Wisconsin-Madison, 53715 University Ave, Madison WI, United States
[2]Computer Science, Rensselaer Polytechnic Institute, 12180 People Ave, Troy NY, United States

Abstract. Hierarchical Federated Learning (FL) presents a novel approach that combines global model training with localized personalization fine-tuning to enhance the predictive accuracy of decentralized machine learning systems. Traditional FL methods, which allow multiple clients to collaboratively train a global model without sharing raw data, are hindered by issues such as non-independent and identically distributed (non-IID) data, communication overhead, and limited generalization across diverse client datasets. This study proposes a hierarchical model that mitigates these challenges by incorporating a global model, trained using the Federated Averaging (FedAvg) algorithm, and applying client-specific fine-tuning to improve local model performance. The experiment conducted on a movie recommendation system demonstrates that this hierarchical approach significantly reduces the global model's error while offering personalized improvements on client-specific datasets. Results show an average Root Mean Squared Error (RMSE) reduction of 0.0460 following local personalization. This hybrid approach not only enhances model accuracy but also preserves data privacy and increases scalability, making it a promising solution for decentralized recommendation systems.

1 Introduction

Movies are one of the most important forms of entertainment today, and to fulfill users' demand on watching movies that suit their tastes, a movie recommendation system emerges. It aims to predict users' preferences by suggesting films based on their past interactions. It recommends films based on features like genre, director, or cast that align with the user's previously watched or rated movies. These systems are essential for streaming platforms as they enhance user satisfaction by delivering personalized content, which in turn increases engagement and maximizes profits for companies.

However, traditional movie recommendation algorithms often require access to vast amounts of user data, leading to privacy concerns. The need for extensive data collection and centralized storage can expose sensitive information and raise issues like data breaches or misuse. Federated learning presents a solution by allowing data to remain decentralized on users' devices while enabling the model to learn from distributed data. This approach

* Corresponding author: Sli857@wisc.edu

enhances user privacy while maintaining the accuracy of the recommendations. Federated learning not only mitigates privacy risks but also reduces the need for constant data transfer, improving both data security and scalability for recommendation systems.

2 Literature review

Federated learning (FL) has emerged as a promising paradigm for distributed machine learning, enabling clients to collaboratively train models without sharing raw data. Various architectures and algorithms have been proposed to address the challenges inherent in FL. This section provides an overview of existing approaches in FL, highlighting key techniques, challenges, and limitations.

2.1 Centralized federated learning (single-server architecture)

The traditional FL setup involves a centralized server that coordinates the learning process across multiple client devices. Clients train local models using their private data and send the model parameters to the central server, which aggregates these updates to refine a global model. The FedAvg algorithm is foundational to this architecture, where the server averages the received model parameters from the clients to update the global model [1]. It reduces communication costs by allowing partial updates. Differential Privacy (DP) and Secure Multi-Party Computation (SMPC) are widely used to protect privacy. DP adds noise to the model updates to obscure individual data points [2], while SMPC ensures that model updates are aggregated without revealing individual client contributions [3].

2.2 Decentralized federated learning (Peer-to-Peer Architecture)

Decentralized FL eliminates the need for a central server, allowing clients to communicate directly and aggregate model updates in a peer-to-peer network. This setup promises improved scalability and resilience by removing the single point of failure. Blockchain and Gossip protocols are often employed to ensure that all clients in the network agree on the global model state [4]. These protocols enable reliable model aggregation and synchronization without a central coordinator. In decentralized networks, local models are synchronized by sharing parameters with neighboring nodes until the global model converges.

2.3 Federated mixture of experts (FedMoE)

FedMoE introduces a novel approach to FL by enabling clients to select different expert models according to their local data. This method enhances the personalized performance of models while leveraging the advantages of both global consistency and local adaptation. The core of FedMoE lies in its MoE framework, where clients access different submodels from an expert pool based on the local data characteristics [5]. This allows the global model to adapt better to each client's unique data distribution while reducing computational overhead. A dynamic gate mechanism assigns the most appropriate experts to each client based on their data distribution, ensuring that the model selects the best submodels for local training.

2.4 Key challenges in federated learning

Despite the progress made by different FL architectures, several key challenges remain unresolved: While FL reduces raw data transmission, the communication of model parameters, especially in cross-device FL, remains substantial [6]. Asynchronous

communication, though flexible, can exacerbate stale updates, leading to slower convergence. Also, allowing clients to participate at different times introduces complexity in managing model versions and may lead to inefficient use of communication resources [7]. Moreover, Data heterogeneity is another concern. Real-world FL scenarios often involve non-independent and identically distributed (non-IID) data, which can lead to poor model performance. Global models trained on such data may not generalize well across clients [4]. Balancing global knowledge with local specificity remains challenging. Personalized FL attempts to address this by customizing models to individual clients, but it still risks either overfitting or underfitting depending on data distributions. Lastly, although DP offers strong privacy guarantees, it reduces model accuracy by introducing noise. Finding the right balance between privacy and utility is critical, especially in sensitive applications [2]. SMPC provides robust privacy protection during model aggregation but is computationally expensive, making it impractical for large-scale or real-time applications [3].

3 Method

3.1 Data preparation

We utilized two datasets for this study: the MovieLens small ratings dataset and the movies metadata dataset [8, 9]. The ratings dataset contained user ratings for various movies, while the metadata dataset provided movie attributes such as genres, popularity, runtime, and other relevant features. To begin, we ensured consistency between the 'movieId' from the ratings dataset and the id from the metadata dataset. Non-numeric IDs in the metadata were converted to NaN and subsequently dropped. We then merged the two datasets on these identifiers to link user ratings with corresponding movie attributes. Missing values in the numerical fields were handled by imputing median values for 'runtime" and 'vote_average', and zeros for 'popularity', 'vote_count', and 'revenue'. This ensured the dataset was complete and ready for analysis.

Key features selected for analysis included 'genres', 'popularity', 'runtime', 'vote_average', 'vote_count', and 'revenue'. The genres field was initially stored in a JSON-like string format. We parsed this field into a list of genre names and applied multi-label binarization to convert each genre into binary features, making it easier to incorporate them into the model.

The dataset was split into two groups of users: 70% were assigned to the federated learning group and the remaining 30% were allocated for personalization. The federated learning dataset was further split into training (70%) and testing (30%) sets. To simulate variability in the data available at different nodes, we sampled the training data to 20% of its original size. Both datasets were standardized using the Standard Scaler to normalize the feature distributions, which is essential for gradient-based learning algorithms.

3.2 Proposed approach

3.2.1 Federated learning model

We employed a neural network model for regression tasks, implemented using PyTorch. The model consisted of fully connected layers, which predicted user ratings based on input features. A federated learning environment was simulated using eight nodes. In each round of training, local models were trained on the respective node's data using the Adam optimizer [10] with a learning rate of 0.01 and mean squared error (MSE) loss function. After each training round, the local model parameters were aggregated on a central server using a

weighted average to update the global model. The global model was then broadcast back to the nodes for the next round of training. The model was evaluated on a test dataset from the federated learning group after each training round, using Root Mean Squared Error (RMSE) as the evaluation metric.

3.2.2 Personalization experiment

After the global model was trained, we performed a personalization experiment to assess the benefits of fine-tuning local models. Each of the eight nodes received data corresponding to users reserved for personalization. The local models were initialized using the final global model's parameters and fine-tuned on the node-specific dataset for ten epochs using a reduced learning rate of 0.005. The personalized models were then evaluated on local test data.

3.3 Visualization

To support our analysis, we generated several plots. First, Global Test RMSE Over Rounds: This plot illustrated the convergence of the global model during federated training by showing the RMSE on the global test set after each training round. Second, Average Training Loss per Epoch: We calculated the average training loss across all nodes for each epoch to monitor the training progress and detect any issues with model convergence. Third, RMSE Before and After Personalization: For each node, we plotted the RMSE before and after personalization to visually compare the performance improvements due to local fine-tuning. Fourth, Weight Changes After Personalization: We plotted the L2 norm of the weight changes in the first fully connected layer of the model for each node. This provided insights into how much the local models diverged from the global model during personalization.

By combining federated learning with local personalization, our methodology aimed to leverage the strengths of both approaches: the ability to learn from a wide range of data while still catering to individual user preferences. The comprehensive evaluation and visualization of results allowed us to assess the effectiveness of our approach in building a personalized recommendation system.

4 Experimental results

Overall, the results demonstrate the effectiveness of combining global training with local fine-tuning to enhance predictive accuracy in a decentralized recommendation system.

4.1 Global model performance

Fig. 1 illustrates the average training loss per epoch over all nodes in a federated learning setting. The loss starts high, with a sharp decline from approximately 8 to 2 within the first few epochs, indicating rapid learning and model improvement. After this initial steep drop, the loss continues to decrease gradually, stabilizing around epoch 20, with minimal fluctuations as training progresses. By epoch 40, the loss approaches a value just above 1, suggesting that the model has largely converged.

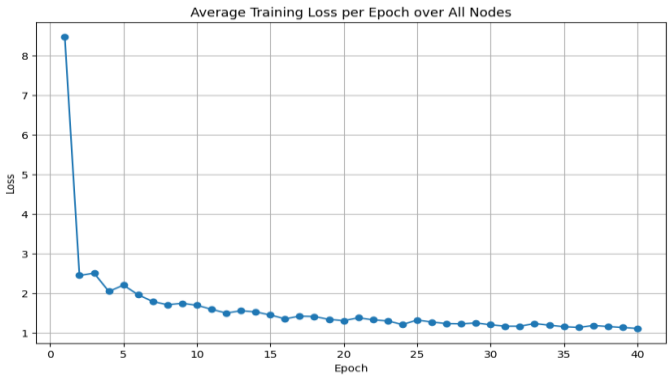

Fig. 1. Average training loss per epoch over all nodes (Photo/Picture credit: Original).

As the global model converged, it was evaluated on a held-out test dataset after each training round to monitor its predictive performance. As illustrated in Fig. 2., the root mean squared error (RMSE) on the global test set decreased progressively over the ten training rounds:

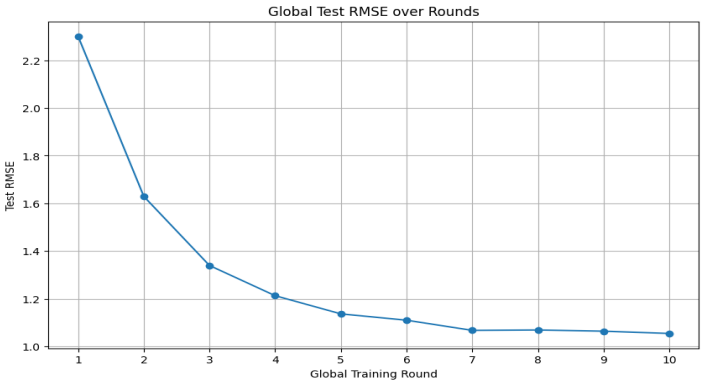

Fig. 2. Global test RMSE over rounds (Photo/Picture credit: Original).

The RMSE decreased significantly from an initial value of 2.3013 to a final value of 1.0539 over ten training rounds. This substantial reduction indicates that the global model effectively learned from the distributed data across the nodes, improving its ability to predict user ratings based on movie features.

Interestingly, there was a slight increase in RMSE from Round 7 to Round 8 (1.0668 to 1.0684), which could be attributed to variations in local model updates or data distributions in that particular round. However, the overall trend demonstrates consistent improvement in the model's predictive performance.

4.2 Impact of local personalization

A key objective of this study was to evaluate the benefits of local personalization after global training. Each node fine-tuned its local model using data from a subset of users reserved for personalization. The performance before and after personalization was measured using RMSE on node-specific test datasets. The results are summarized in Fig. 3.

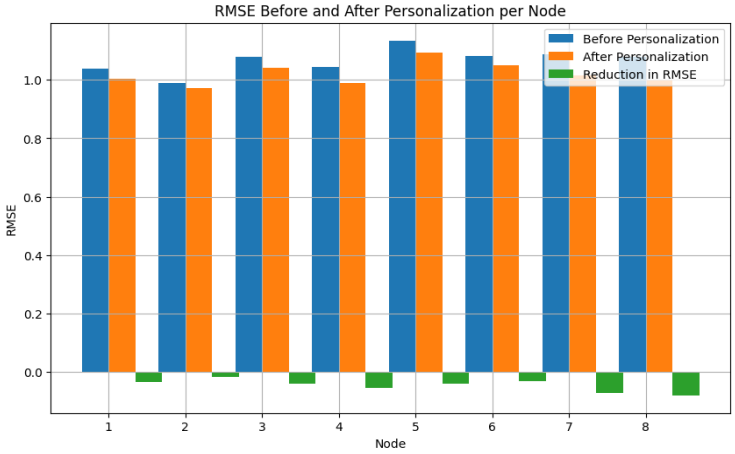

Fig. 3. RMSE before and after personalization per node (Photo/Picture credit: Original).

The RMSE improvements ranged from 0.0172 to 0.0792 across the nodes, with the largest improvement observed in Node 8 (0.0792) and Node 7 (0.0717). The average RMSE improvement across all nodes was approximately 0.0460, representing a significant enhancement in predictive accuracy. The consistent decrease in RMSE demonstrates the effectiveness of local personalization in capturing user-specific preferences that were not fully learned during global training.

4.3 Weight changes during personalization

To quantify the extent of model adjustments during personalization, we measured the L2 norm of the weight changes in the first fully connected layer of each local model. The weight changes for each node are provided in Fig. 4. The norms ranged from approximately 3.7492 to 5.3091, indicating meaningful adjustments to the model parameters during personalization.

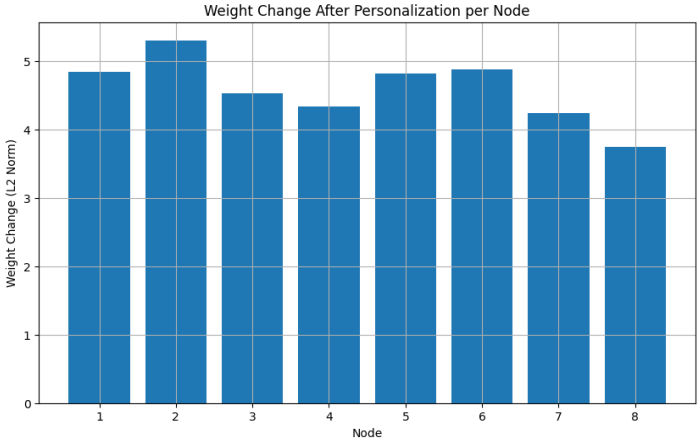

Fig. 4. Weight change after personalization per node (Photo/Picture credit: Original).

These weight changes reflect the fine-tuning process, where the models adapted to local data by updating their weights to better capture the unique preferences of the users assigned to each node. The magnitude of these changes suggests that while the global model provided

a strong baseline, personalization required significant adjustments to enhance local performance.

5 Discussion

The substantial decrease in global test RMSE from 2.3013 to 1.0539 over ten training rounds confirms that federated learning can effectively train a global model without centralized data aggregation. By leveraging data distributed across multiple nodes, the global model captured general patterns in user ratings and movie features. This approach preserves data privacy while still benefiting from a broad range of data.

The significant reduction in RMSE after personalization underscores the value of local fine-tuning. Despite the global model's performance, it could not fully account for individual user preferences and behaviors. Personalization allowed each node to adjust the model parameters based on local data, resulting in improved accuracy.

The RMSE improvements ranged from 0.0172 to 0.0792 across the nodes, with the largest improvement observed in Node 8 (0.0792) and Node 7 (0.0717). This suggests that personalization had a particularly strong impact for nodes with certain user characteristics or data distributions. Nodes with smaller datasets, such as Node 8, might have had more unique user preferences that were not fully captured by the global model, leading to larger gains from personalization.

The weight changes during personalization varied among the nodes, with Node 2 experiencing the largest change (5.3091) and Node 8 the smallest (3.7492). These differences could be attributed to the amount of data available for personalization and the diversity of user preferences within each node. While there isn't a direct correlation between weight change magnitude and RMSE improvement, both metrics highlight the adjustments made during personalization to better fit local data.

6 Conclusion

This study completed an integration of federated learning with local personalization which presents a powerful approach for building decentralized recommendation systems. Our study demonstrates that while global models can capture general patterns, personalization is essential for addressing individual user preferences. By achieving notable reductions in RMSE through local fine-tuning—averaging an improvement of 0.0460 across nodes—we highlight the potential of this combined approach to enhance predictive accuracy and user satisfaction in recommendation systems.

Although our results are promising, several limitations warrant further investigation. Firstly, the data in our experiments may not fully represent real-world scenarios where data distribution across nodes can be highly non-uniform and skewed. Future studies could simulate more heterogeneous data distributions to assess the robustness of the approach. Also, we used a relatively simple neural network model. Exploring more complex architectures, such as deep neural networks or models incorporating temporal dynamics, may capture additional nuances in the data. Lastly, investigating how personalization effects persist over time and how models adapt to changing user preferences would provide deeper insights. Incorporating mechanisms for continual learning and adaptation could improve long-term performance.

Authors contribution

All the authors contributed equally, and their names were listed in alphabetical order.

References

1. H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Aguera y Arcas, Communication-efficient learning of deep networks from decentralized data. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 1273-1282 (2017).
2. M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, L. Zhang, Deep learning with differential privacy. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 308-318 (2016).
3. K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, ... K. Seth, Practical secure aggregation for federated learning on user-held data. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 1175-1191 (2017).
4. Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Z. Li, ... Y. Liu, Privacy-preserving blockchain-based federated learning for IoT devices. IEEE Internet of Things Journal, 8(3), 1817-1829 (2020).
5. G. Long, T. Shen, Y. Tan, L. Gerrard, A. Clarke, J. Jiang, Federated learning for privacy-preserving open innovation future on digital health. In Humanity driven AI: productivity, well-being, sustainability and partnership, Springer International Publishing, 113-133 (2021).
6. P. Kairouz, H. B. McMahan, D. Ramage, S. Song, The federated learning research landscape: An overview. arXiv preprint arXiv:2102.02079 (2021).
7. K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, ... T. van Overveldt, Towards federated learning at scale: System design. Proceedings of the 2nd SysML Conference, Stanford, CA, USA (2019).
8. S. Mehta, Movie Lens Small Latest Dataset. https://www.kaggle.com/datasets/shubhammehta21/movie-lens-small-latest-dataset (2018).
9. R. Banik, The Movies Dataset. https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset?select=movies_metadata.csv (2017).
10. D. P. Kingma, Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).