



Research on Distributed Machine Learning Training Based on Serverless Computing Platforms

Xinyue Duan^{1,*} ^a and Taimin Rong²  ^b

¹*School of Communication & Information Engineering, Shanghai University, Baoshan, Shanghai, 201900, China*

²*School of Communication & Information Engineering, Nanjing University Information Science and Technology, Nanjing, Jiangsu, 210000, China*

*

Keywords: Knowledge Transfer, Federated Learning, Distributed Computing.

Abstract: Currently, since 2023, with the further development of artificial intelligence, the requirements for the efficiency and accuracy of federated learning have also increased. In distributed computing environments, the heterogeneity of hardware configurations and data distributions across different devices poses challenges to model training. Designing a federated learning algorithm that can adapt to heterogeneous environments and effectively perform knowledge transfer is a key research challenge. To address the construction of multi-task distributed models with minimal communication resource consumption, this paper presents the advanced Knowledge Transfer-Personalized Federated Learning (KT-pFL) algorithm. KT-pFL is based on the training and improvement of the Heterogenous Federated Learning via Model Distillation (FedMD) algorithm, and the core logic of the algorithm is briefly described. This paper proposes improvement methods for the KT-pFL algorithm based on the reproduced algorithm, including the addition of normalization layers and dynamic adaptive adjustment of the learning rate. These improvements can increase the accuracy of the algorithm by approximately 2% when training on the CIFAR-10 dataset. Finally, a brief analysis and outlook on potential future research directions for knowledge transfer are provided.


1 INTRODUCTION


Artificial intelligence (AI) has become a hot topic. In the training of large AI models, algorithms play a crucial role in training efficiency. The same data can produce vastly different results when trained with different algorithms. The best solution currently offered by major manufacturers for building the hardware foundation for AI is Graphic Processing Unit (GPU) clusters. Often, different terminals across regions and devices need to be integrated to train an algorithm. This requires a learning method that can train on different devices and reasonably combine the results from each device. This demand is one of the important reasons for the emergence of federated learning.

FedMD is a significant milestone in the development of federated learning. It is based on improvements in transfer learning. Transfer learning uses a mapping function to map samples from the

source and target domains to the same distribution space to reduce the differences between them. This allows knowledge learned in the source domain to be applied to the target domain to solve tasks in the target domain (Zhao, Li, & Lin, 2020). FedMD(Heterogenous Federated Learning via Model Distillation) introduces a public dataset as a medium and uses model distillation techniques to achieve knowledge sharing (Sun, Wang, & Liu, 2024).

Each participant uses their own private data and independently designed models for training but communicates through prediction results (such as classification scores) on the public dataset, thereby achieving knowledge transfer without sharing raw data or model parameters, significantly saving communication resources (Li, Wang, 2019). This method greatly improves accuracy but also increases the demand for resources and computing power. How to make the datasets required for training more efficiently utilized has become a hot topic in

^a  <https://orcid.org/0009-0008-6297-3031>

^b  <https://orcid.org/0009-0006-0633-269X>

federated learning. In recent years, personalized federated learning (pFL) has gained increasing attention due to its potential to handle client statistical heterogeneity. It distills each model to maximize data utilization (Chen, Zhang, 2024). However, early pFL methods relied on the aggregation of server-side model parameters, requiring all models to have the same structure and size, which limited their application in more heterogeneous scenarios. Therefore, personalized federated learning (pFL) was proposed. For a group of clients with different models, a method using knowledge transfer matrices and knowledge distillation was proposed to coordinate the differences between models, referred to as KT-pFL. This framework allows for personalized models for different clients by introducing a knowledge coefficient matrix that adaptively enhances collaboration between clients with similar data distributions, addressing the issue of statistical heterogeneity among clients. Extensive experiments on datasets such as EMNIST, Fashion-MNIST, and CIFAR-10 have shown that this framework significantly outperforms the latest algorithms in performance (Cohen, Afshar, Tapson, & van Schaik, 2017; Xiao, Rasul, & Vollgraf, 2017; Krizhevsk, Hinton, 2009).

However, since this framework requires maintaining and updating the knowledge coefficient matrix, the computational complexity also increases significantly, with training times more than double that of traditional FedMD (Zhang, Guo, & Ma, 2021). This paper reproduces the existing code and proposes two methods to maximize data utilization efficiency and improve accuracy without further increasing computing power and datasets.

2 THEORETICAL FOUNDATION

To overcome model constraints and fully utilize the potential of heterogeneous model configurations, the goal of the novel training framework put out by KT-pFL study is to create customized models for various clients. In the original pFL, it formalizes the aggregation process as a customized group knowledge transfer training method that enables every client to help other clients with their local training by keeping a customized soft prediction on the server side. First, public data D0 is used for training. The loss and accuracy are calculated, and then the validation loss and accuracy are recorded using their own datasets. It uses transfer learning methods based on domain and task to build methods for transmission (Li, Wang, 2019). Specifically, a

knowledge coefficient matrix is established based on the prediction parameters and loss of each model, and the personalized soft prediction of each client is updated. This matrix can adaptively enhance collaboration between clients with similar data distributions. Furthermore, the knowledge coefficient matrix is parameterized so that it may be trained alongside the model parameters in order to quantify each client's contribution to the customized training of other customers.

3 ALGORITHM DESIGN

3.1 Training of a Single Model:

First, update the model parameters, evaluate the accuracy, and record the loss during the training process. The algorithm returns the loss and accuracy data for each round of training and validation.

3.2 Function for Validating a Single Model

This program is embedded in the training of a single model, mainly calculating the accuracy of the model. In this experiment, 80% of the data is used for training, and 20% is used for validation. The output of the model is calculated by traversing the validation data loader. If a loss function is provided, the average validation loss and average accuracy are returned; if no loss function is provided, the average accuracy is returned.

3.3 Implementation of Weighted Logits Calculation and Weight Update Function

This section aims to implement the calculation of weighted logits for each model. The calculation of weighted logits is based on the correlation between the teacher model and the local model multiplied by the local model tensor. Then, the loss function and weight penalty are calculated, and the results are returned to the matrix (Tamirisa, Rishub, & Xie, 2024). The weights are dynamically adjusted based on the gradient changes in accuracy to achieve stability in the result region (Sun, 2024).

This study adopts a learning rate adjustment method and finally returns the above weight values to the weight matrix as the final weights for each model. The weights are multiplied by the logits of the teacher model and added to the logits of the local model to

obtain the weighted logits. Logits are not a single value but a multi-dimensional tensor, representing the output logits of multiple models on the input data. This tensor includes the number of models, the batch size of the input data, and the number of categories in the classification task, and is an indicator that reflects the comprehensive performance of the model, often used for weight adjustment and operations (Zhao, Li, & Lin, 2020).

3.4 Implementation of the Function for Predicting Model Output

The predict function is used to make predictions on a given model and dataset and returns the prediction results. First, the model is set to evaluation mode and moved to the specified device. Then, the data is traversed according to the batch size. Finally, the output results of all batches are concatenated into a complete numpy array and returned. The train_models function is used to train a group of models, supporting early stopping, model saving, and other functions. First, check if there is an available GPU; if not, use the CPU. Then, traverse each model, create training and test data loaders, use stochastic gradient descent optimizer and cross-entropy loss function for training, and record the training and validation results of each model. If a save directory is specified, the model state is saved to a file. Finally, the function returns the training and validation results of all models and prints the pre-training accuracy. These two functions are commonly used in the training and inference processes of deep learning models.

3.5 Main Program

Train each participant's model and record the training and validation results. Perform collaborative training.

Initialize collaborative performance records, training rounds, and weight matrix. Read CIFAR-10 data and generate aligned datasets, test the performance of each model, and record the results.

Update the training rounds and the rounds of each model, and align the main parameters of the models for coordination.

After completing the above steps, each model uses private data for training and returns the performance records of collaborative training.

3.6 Training Improvements

The previous reproduced program used two convolutional layers and three fully connected layers

for training. This improvement adds a normalization layer after each convolutional layer, which can effectively improve accuracy. Adding a normalization layer to the FedMD algorithm can increase the accuracy by about 2% (Kang, Liu, 2023).

4 EXPERIMENTAL RESULTS

4.1 Tasks and Datasets

The experiment evaluates the proposed training framework on the CIFAR-10 dataset (Krizhevsky et al. 2009). For each dataset, we apply two different non-independent and identically distributed (Non-IID) data settings:

Non-IID Type 1: All categories have samples in each client, however the quantity of samples in each category varies from client to client.

Non-IID Type 2: There are just two sample categories per customer.

Twenty-five percent of each dataset is utilized for testing, while the remaining 75 percent is used for training. Every client's test data is distributed similarly to the training data. The average test accuracy of all local models is recorded for evaluation across all approaches.

4.2 Model Structure

LeNet, AlexNet, ResNet-18, and ShuffleNetV2 are the four lightweight model structures used in the experiment (LeCun et al., 1998; He et al., 2016; Ma et al, 2018; (Krizhevsky, Sutskever, & Hinton, 2012). There are four models in the total personalized federated learning system, and each model has five clients.

4.3 Foundational Techniques

To illustrate the generalizability and efficacy of our suggested training framework, we contrast the performance of KT-pFL with that of the non-individualized distillation-based approach FedMD (Li, Wang, 2019).

The following are the outcomes of the experiment:

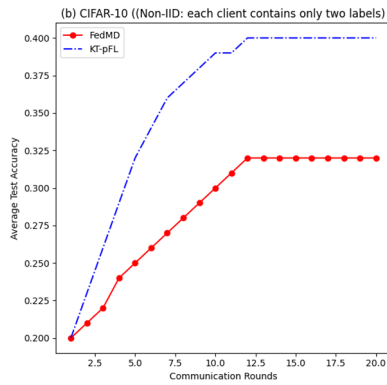


Figure 1: Average test accuracy performance comparison between FedMD and KT-pFL on CIFAR-10 (Non-IID scenario 1: every client has all labels) (Original).

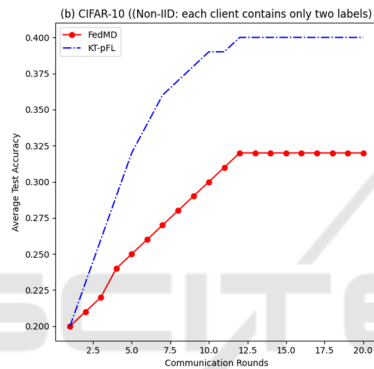


Figure 2: Average test accuracy performance comparison of FedMD and KT-pFL on CIFAR-10 (Non-IID example 2: each client has just two labels) (Original).

Using the CIFAR-10 dataset under non-independent and identically distributed (Non-IID) circumstances, Figures 1 and 2 compare the average test accuracy per training round for two federated learning approaches (FedMD and KT-pFL) across a total of 20 training rounds.

KT-pFL performs well in both Non-IID settings, with its average test accuracy and client test accuracy significantly higher than FedMD. The upward trend in client test accuracy and validation accuracy of KT-pFL indicates that the algorithm can effectively improve model performance under Non-IID data distribution and has good generalization ability. The main reason is that in FedMD, each participant has an independently designed model, but the collaboration process mainly improves the model through consensus, with relatively limited personalization. By employing a knowledge coefficient matrix to measure each client's contribution to other clients, KT-pFL, on the other hand, explicitly suggests individualized objectives. This allows for more model customization

and more flexible adaptation to various client data distributions.

KT-pFL uses personalized loss functions and knowledge coefficient matrices to perform knowledge transfer based on the similarity of client data distributions, showing superior performance in handling statistical heterogeneity.

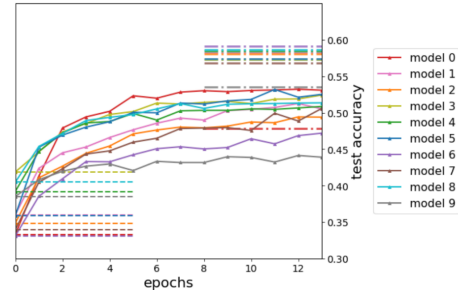


Figure 3: Test accuracy for each client in Non-IID case 1 during KT-pFL training (each client contains all labels) (Original).

With the horizontal axis denoting the training rounds and the vertical axis denoting the test accuracy, Figures 3 and 4 display the change curves of the average test accuracy of ten customers throughout training under two Non-IID circumstances on the dataset. In Non-IID case 1, the training accuracy of each model is above 0.5, but the final validation accuracy is around 0.42, indicating that KT-pFL, under the condition that each model traverses all labels, leads to a certain degree of decline in data recognition ability due to knowledge transfer between models.

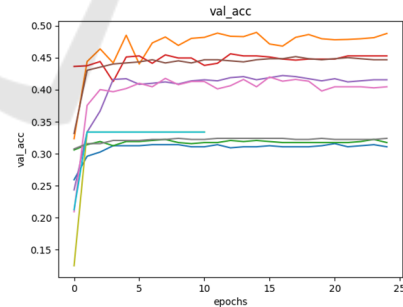


Figure 4: Test accuracy of each client during training for KT-pFL in Non-IID case 2 (each client does not contain all labels) (Original).

In practical use, it is difficult for each model to train on all labels. In fact, it is more common for each model to train on a small number of labels, as in Non-IID case 2. From the experimental results, it can be seen that some models have an accuracy of around 0.45, while others are around 0.35. The final accuracy is around 0.4. The knowledge transfer between

models does not lead to a decline in the final model's recognition ability. Considering the differences in training difficulty between different labels, KT-pFL already shows considerable excellence in this case.

4.4 Discussion and Analysis

The data results in this paper tend to stabilize after 13 rounds in both cases, and there is no decline in accuracy before the 13th round. Compared to the original KT-pFL algorithm (Zhang, Guo, & Ma, 2021), this paper avoids the decline in accuracy during training by adjusting the learning rate, to some extent improving the reliability of the training results. Future research could consider adjusting the size of the public dataset, with smaller sizes being better for saving communication resources but without significantly affecting accuracy.

5 CONCLUSION

The KT-pFL algorithm shows significant improvement over the FedMD algorithm, especially in Non-IID case 2. This is because it can better utilize different clients to train local models, allowing these models to specialize in one aspect. This is very suitable for Non-IID case 2. The improvements in two detailed directions in this paper can further enhance KT-pFL on its already excellent foundation. The improvements based on learning rate and normalization layers help improve the accuracy of the algorithm and reduce loss, significantly reducing the risk of accuracy decline due to excessive data training adjustments and allowing for more efficient use of data in each round through normalization layers. In practical situations, due to limited training data, it is necessary to make the most of them. However, it should also be noted that the accuracy of the models generated by each client varies greatly, and the models basically stop improving after 10 rounds of training. Considering the accuracy of the models, there is still room for improvement.

ACKNOWLEDGEMENTS

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

REFERENCES

- Chen, J. & Zhang, J., 2024. Data-free personalized federated learning algorithm based on knowledge distillation. *Information Network Security*, 10, pp. 1562-1569.
- Cohen, G., Afshar, S., Tapson, J., & van Schaik, A., 2017. EMNIST: Extending MNIST to handwritten letters. In *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE.
- He, K., Zhang, X., Ren, S., & Sun, J., 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778. IEEE.
- Kang, Y., Liu, W., 2023. Adaptive federated learning algorithm with batch normalization. *Journal of Wuhan Institute of Technology*, 45(5), pp. 549-555.
- Krizhevsky, A., & Hinton, G., 2009. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E., 2012. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, pp. 1106-1114.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), pp. 2278-2324.
- Li, D., & Wang, J., 2019. FedMD: Heterogeneous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*.
- Ma, N., Zhang, X., Zheng, H.-T., & Sun, J., 2018. Shufflenet v2: Practical guidelines for efficient CNN architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 116-131.
- Sun, H., 2024. Optimization methods in distributed machine learning based on adaptive learning rate. *Doctoral Dissertation, University of Science and Technology of China*.
- Sun, Y., Wang, Z., Liu, C., Yang, R., Li, M. & Wang, Z., 2024. Related methods and prospects of personalized federated learning. *Computer Engineering and Applications*, 20, pp. 68-83.
- Tamirisa, R., Xie, C., Bao, W., Zhou, A., Arel, R., & Shamsian, A., 2024. FedSelect: Personalized federated learning with customized selection of parameters for fine-tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Xiao, H., Rasul, K., & Vollgraf, R., 2017. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.
- Zhang, X., Zhuang, Y., Yan, F. & Wang, W., 2019. Research and progress in category-level object recognition and detection based on transfer learning. *Acta Automatica Sinica*, 07, pp. 1224-1243.
- Zhang, J., Guo, S., Ma, X., Wang, H., Xu, W., & Wu, F., 2021. Parameterized knowledge transfer for personalized federated learning. In *Advances in Neural Information Processing Systems*, 34 (NeurIPS 2021).

Zhao, P., Li, Y., & Lin, M., 2020. Research progress on intention recognition for transfer learning. *Computer Science and Exploration*, 08, pp. 1261-1274.

