Overcoming Forgetting in Federated Learning via Importance From Agent

Term Project for AML-XAI (430.658), Seoul National University

Mijeong Kim **Geeho Kim** **Minji Kim**

Abstract

Federated learning is a task that a central server learns a deep learning model by repeating the process of aggregating and re-sending the trained models from separated local agents under the privacy restrictions. We aim to overcome *catastrophic forgetting* issue in federated learning on non-IID data by adapting solutions proposed in continual learning literature, where the catastrophic forgetting occurs when a aggregated model is re-trained on each server with corresponding data distribution. In our algorithm, we newly assume the parameter importance of each agent is accessible to the server without invading privacy protection, and the server also use it to aggregate the local agent's updated weights instead of uniform summation, preventing the important parameters from drift by other models which is trained on distinct data. Our extensive experiments demonstrate remarkable performance gains by the proposed approach in non-IID setting on multiple benchmarks.

1 Introduction

Modern edge devices such as mobile phones or vehicles have access to a wealth of data suitable for learning models, which in turn can greatly improve the generalization of the models. However, due to data privacy concerns, it's impractical to gather all the data from the edge devices at the data center and conduct centralized training. Federated learning [8] is emerging paradigm for distributing training of machine learning models in networks of remote devices, without requiring any of the participants to reveal their private data to a centralized entity.

Despite the scalability and communication efficiency of the federated learning, it shows significant performance degradation on heterogeneity of local data where each agent posses non-IID training data. This issue is analogy to catastrophic forgetting in continual learning in the sense that models lose the previous information from models of previous step during model update on local tasks or training other task, respectively. Specifically, the forgetting issue in federated learning is caused in two places: weight divergence caused by local update and simple average of local models in the central server.

To overcome the problem, we propose a novel federated learning framework, which reduce weight divergence of local models by adopting solutions proposed in continual learning literature for local model update. We newly assume the central server can access the fisher information of each agent, which does reveal any private information of agents. We also propose importance-based global aggregation algorithm to preserve the knowledge of local tasks in constructing global model in the server. Our algorithm uses weighted average based on fisher information instead of uniformly average when aggregating the local agent's updated weights. To the best of our knowledge, our method is the first attempt to use fisher information when aggregating each agent's weights.

The main contribution of our work are summarized as follows:

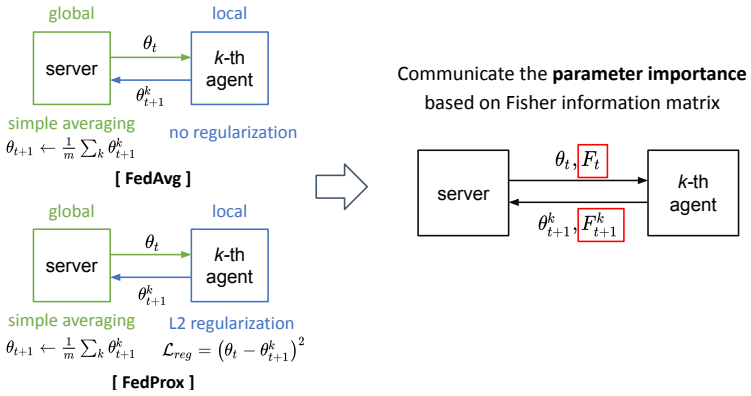

Figure 1: Comparison between popular federated learning approaches and ours. Unlike previous methods communicates only updated local models for training a global model, each agent, in our method, additionally sends information of parameter importance F_t^k. Central server then aggregates the information and sends the global importance F_t to participant agents of next round.

- We propose a novel federated learning framework, which reduces weight divergence between local models by regularizing local model update based on the global weight importance downloaded from the server.
- We also propose a novel global aggregation algorithm based on parameter importance from agents, which maintains knowledge of local models effectively.
- We show that our method achieves performance gain under the non-IID settings compared to state-of-the-art baselines on multiple classification benchmarks.

2 Related Work

2.1 Continual learning

Among diverse approaches to overcome the catastrophic forgetting of the old tasks in continual learning, regularization-based approaches identify important weights of the model for each task and give constraints to them. EWC [4] adopts Fisher information matrix as a measure of the importance of the parameter to put high regularization to the important nodes. On the other hand, SI [11] consider the learning trajectory by using gradient path integral. RWalk [1] generalizes EWC and SI to take full advantage of both Fisher-based and optimization path-based parameter importance. We adopt EWC [4] in our local model update stage to prevent the weight divergence.

2.2 Federated learning

The typical federated learning paradigm consists of two steps: (i) each edge device trains a model downloaded from a central server with its local dataset independently, and (ii) the server gathers the locally trained models and aggregates them to obtain a shared global model. One of the standard aggregation methods is FedAvg [8] where parameters of local models are averaged element-wise with weights proportional to sizes of the client datasets, but this method shows significant performance degradation when the local data is collect in non-IID manner. To overcome the limitation, FedProx [7] adds a proximal term to the client cost functions, thereby limiting the impact of local updates by keeping them close to the global model. Clustering based approaches [9, 10, 2, 3] construct several global models according to the distribution of local data to prevent weight drift by averaging models from distinct tasks.

While all these methods communicate only updated model parameters for training a single global model, we additionally communicate the information for weight importance from each agents' models, and mitigate prevent the catastrophic forgetting by considering the information for both local update and global aggregation.

Algorithm 1: Algorithm of the proposed method. The K agents are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
initialize θ_0
for each round $t = 1, 2, ...$ **do**
  $S_t \leftarrow$ (random set of m agents)
  for each agent $k \in S_t$ **in parallel do**
    $\theta_{t+1}^k, F_{t+1}^k \leftarrow \text{AgentUpdate}(k, \theta_t, F_t)$
  $\theta_{t+1} \leftarrow \frac{1}{m}\sum_k \bar{F}_{t+1}^k \theta_{t+1}^k$ // Update global model with local weight importance.
  $F_{t+1} \leftarrow \frac{1}{m}\sum_k F_{t+1}^k$ // Update global Fisher matrix.

AgentUpdate(k, θ, F)**:** // Run on agent k
$\mathcal{B} \leftarrow$(split $\mathcal{P}_k$ into batches of size B)
for each local epoch i from 1 to E **do**
  for batch $b \in \mathcal{B}$ **do**
    $\mathcal{L}(\theta^k) = \mathcal{L}_k(\theta^k) + \frac{\lambda}{2} F_t \left(\theta^k - \theta\right)^2$ // Regularize Update local with global fisher matrix.
    $\theta \leftarrow \theta - \eta \nabla \mathcal{L}(\theta)$
$F_{t+1}^k \leftarrow \gamma F_t + (1-\gamma) F_{t+1}^k$. // Update fisher matrix
return θ, F to server

3 Proposed Method

In this section we introduce a novel federated learning framework which utilizes the importance of model parameters in local and global updating to mitigate catastrophic forgetting in federated learning scenario. Overall framework of the proposed method is illustrated in Figure 1.

3.1 Global Update with Local Weight Importance

Suppose that we have a random set S_t of m agents with for each communication round t, and each agent $k \in \{1, \ldots, m\}$ have their local training datasets $\mathcal{D}_k = \{(x_i, y_i)\}_{i=1}^{N_k}$. Before sending the local model to the server, each agent calculates the importance of the model parameters for learning each local task. Inspired by [4], we employ the Fisher information to measure the importance of each model parameter. Specifically, at communication round t, each agent $k \in \{1, \ldots, m\}$ with local updated weight θ^* calculates fisher matrix as

$$F_{\theta,jj}^k \mid \theta^* = \left(\frac{1}{N_k} \sum_{i \in \mathcal{D}_k} \nabla_\theta \left(\log p\left(y_i \mid x_i, \theta\right) \right) \nabla_\theta \left(\log p\left(y_i \mid x_i, \theta\right) \right)^\top \right)_{jj} \Bigg|_{\theta^*}. \tag{1}$$

Then, each agent sends their updated model along with the Fisher information matrix to the server, and the server averages the local models weighting by the importance received from each agent as follows:

$$F_{t+1} = \frac{1}{m} \sum_k F_{t+1}^k \quad \text{and} \quad \theta_{t+1} = \sum_k \bar{F}_{t+1}^k \theta_{t+1}^k, \tag{2}$$

where $\bar{F}_{t+1}^k$ indicates the normalized value of F_{t+1}^k which is computed over each layer from a client and then over all clients, sequentially. Giving more weights on importance parameters for learning local data, the server prevents the important parameter from drift in the aggregation phase. Note that, sending the additional information matrix of the local models does not violate any privacy concern in the federated learning scenario.

3.2 Local Update with EWC

To prevent the local updated model from diverging, we adapt EWC [4] algorithm to the local training at the agents. To be specific, the central server sends not only aggregated model but also the averaged Fisher information matrix $F_t \leftarrow \frac{1}{K}\sum_k F_t^k$ to the participants for the next training step. On each

Table 1: Accuracy (%) for different backbone in non-IID setting on MNIST and CIFAR-10 dataset.

Method	MNIST		CIFAR-10	
	MLP	CNN	MLP	CNN
FedAvg [8]	85.70	93.62	39.64	41.90
FedProx [7]	85.57	93.67	**40.78**	42.05
Ours	**89.19**	**95.08**	39.89	**42.88**

Table 2: Number of communication round to reach target test accuracy in MNIST dataset.

Method	MLP (85%)	CNN (93%)
FedAvg [8]	91	88
FedProx [7]	90	88
Ours	**52** ($\times$ 1.73)	**61** ($\times$ 1.44)

round t, starting from initial point, the agents optimize their local loss by running SGD for E local epochs using the following objective,

$$\mathcal{L}(\theta_{t+1}^k) = \mathcal{L}_k(\theta_{t+1}^k) + \frac{\lambda}{2} F_t \left(\theta_{t+1}^k - \theta_t\right)^2, \tag{3}$$

where $\mathcal{L}_k(\theta)$ is loss for local task of agent k, λ sets how important the model downloaded from the server is compared to the new one, and t denotes communication round. When each agent train the model on each local task, EWC will try to keep the network parameters close to the aggregated parameters which contains information of whole tasks over agents. Finally, we update the importance of each agent at the end of the local update step:

$$F_{t+1}^k \leftarrow \gamma F_t + (1-\gamma) F_{t+1}^k. \tag{4}$$

The updated importance is sent back to the server along with the updated local model. The overall framework is summarized in Algorithm 1.

4 Experimental Results

This section represents our experimental results for image classification given local data of each agent are non-IID. We also provide ablation study to discuss the characteristics of the proposed approach in comparison to existing methods.

4.1 Image classification on non-IID data

Datasets We conduct experiments on two popular image classification benchmarks, MNIST [6] and CIFAR-10 [5]. We follow sampling scheme from [8] on both benchmarks to simulate data heterogeneity between agents. To simulate data heterogeneity between agents, we sort the data by the class label, partitions them into multiple shards, and assign each agent without overlapping, following [8].

Implementation details We use two simple models: 1) A simple multilayer-perceptron with one hidden layer with channel of 200 using ReLU activations (MLP), 2) A CNN with two 5x5 convolution layers (the first with 32 channels, the second with 64, followed by 2x2 max pooling) (CNN). We train both models on non-IID federated setting where the total number of agents K is set to 100 and participation rate of agents at each round m is set to 0.1. Models are trained using SGD with learning rate of 10^{-2} for 10 local epochs while local batch size is set to 10. The number of communication round is set to 100 in MNIST and 200 in CIFAR-10, respectively. For updating Fisher information, the γ value is set to 0.9.

Results The performance comparison on MNIST and CIFAR-10 dataset is shown in Table 1. On MNIST, our method achieves performance gain of 3.49% and 1.46% in MLP and CNN, respectively, compared to the baseline FedAvg [8]. While FedAvg [8] aggregates the local agent's parameters by simple averaging, our method exploits the importance from agents to consider the each agent's influence on the training while updating the global model. On CIFAR-10, our method shows competitive accuracy compared to FedProx [7].

Table 2 shows the number of communication round to reach target test accuracy 85%, 93% on MNIST for MLP, CNN backbone network, respectively. Our algorithm shows the fastest convergence compared to the baseline methods.

Table 3: Abalation study on MNIST dataset. L2 is L2 regularization without using Fisher information. In MLP, λ is set to 10^5 on EWC and 0.05 on L2. In CNN, λ is set to 10 on EWC and 0.005 on L2.

Method	10 rounds		100 rounds	
	MLP	CNN	MLP	CNN
FedAvg [8]	50.93	61.94	85.70	93.62
FedProx [7] (FedAvg + L2)	50.93	62.19	85.57	93.67
FedAvg + EWC	54.38	64.42	85.88	93.65
FedAvg + weighted average	68.45	76.54	88.16	94.93
FedAvg + EWC + weighted average (ours)	**70.25**	**78.11**	**89.19**	**95.08**

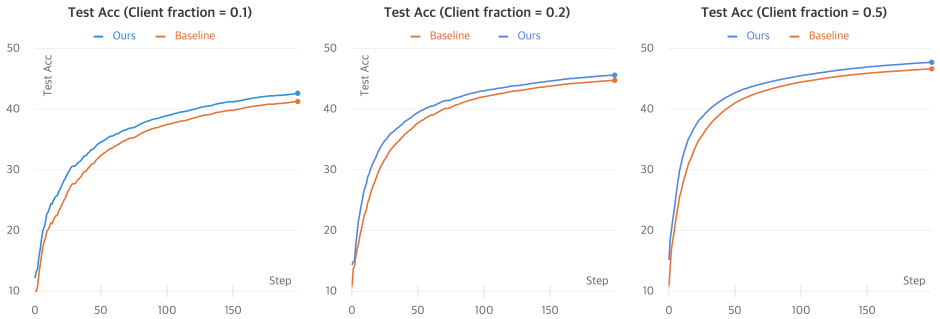

Figure 2: Test accuracy on CIFAR-10 with different rate of participating agents.

4.2 Ablation study

Component Analysis We analyze the impact of each component in our method. Table 3 represents that, on MNIST, the use of all component achieves the highest accuracy and our algorithm consistently improves performance by incorporating the proposed methods, especially the weighted average with Fisher information for the construction of a global model.

Client participation rate Figure 2 shows the test accuracy in terms of client participation rate. For all three settings, our method consistently outperforms the baseline method FedAvg [8] with margins.

Table 4: Accuracy (%) on MNIST dataset in terms of continual learning method.

Method	MLP	CNN
EWC	89.19	95.08
RWalk	88.82	94.44

RWalk We further apply RWalk [1] on our method. Specifically, gradient path-based importance is calculated and updated along with the Fisher information-based importance. The comparison result is shown in Table 4. The additional usage of gradient path-based importance rather causes degradation in the accuracy. We suspect that using RWalk imposes too much regularization on the local model, making it difficult to learn a locally given task before the end of the local update iteration.

5 Conclusion

We have investigated a new approach to novel federated learning method with continual learning literature. The proposed method prevents catastrophic forgetting in each client's local updating. With respect to aggregating model weights from clients, we demonstrate that the weighted summation method based on the parameter importance is highly effective.

Although the proposed method improves the performance with large margin, our algorithm requires doubled communication cost for sharing fisher information from server to each client, and vice versa. For the future work, we would like to reduce the communication cost by sharing only subset of fisher information with thresholding or adopting quantization method.

References

[1] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018.

[2] Moming Duan, Duo Liu, Xinyuan Ji, Renping Liu, Liang Liang, Xianzhang Chen, and Yujuan Tan. Fedgroup: Efficient clustered federated learning via decomposed data-driven measure. *arXiv preprint arXiv:2010.06870*, 2020.

[3] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *arXiv preprint arXiv:2006.04088*, 2020.

[4] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[5] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[6] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[7] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. 2020.

[8] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

[9] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[10] Ming Xie, Guodong Long, Tao Shen, Tianyi Zhou, Xianzhi Wang, and Jing Jiang. Multi-center federated learning. *arXiv preprint arXiv:2005.01026*, 2020.

[11] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017.