Federated Unlearning via Subparameter Space Partitioning and Selective Freezing

Anonymous submission

Abstract

Machine unlearning is a challenging task in the federated learning (FL) ecosystem due to its decentralized nature. Many existing approaches rely on retraining the model from scratch, which is computationally inefficient. We propose a novel federated unlearning method that addresses this inefficiency by partitioning the global and local model parameter spaces into subspaces. During federated training, we cluster the gradient space from all clients and map it to corresponding neurons in the global and local models. When an unlearning request is made, neurons specific to the unlearning class are frozen, effectively neutralizing its contribution. Evaluated on MNIST and CIFAR-10 datasets, the method achieves complete unlearning for targeted classes, with accuracies dropping to 0.00% for "Airplane" in CIFAR-10 and digit 9 in MNIST, while preserving baseline performance for other classes, such as 98.50% for digit 1 and 69.50% for "Ship." On average, the method retains 95.2% accuracy for unaffected classes.

Introduction

Federated learning (FL) has emerged as a powerful paradigm for training machine learning models across decentralized data sources while preserving user privacy (Yang et al. 2019). By enabling collaborative learning without the need for raw data sharing, FL addresses significant privacy concerns associated with traditional centralized training. However, as FL gains adoption, the need for *federated unlearning* has become apparent. Federated unlearning refers to the ability to remove the influence of specific client data or subsets of training data from a collaboratively trained model without requiring full retraining from scratch. This capability is critical for addressing legal obligations, such as compliance with regulations like the General Data Protection Regulation (GDPR) (Romandini et al. 2024), and maintaining user trust in privacy-preserving machine learning systems.

The process of federated unlearning is complex due to the decentralized nature of FL and the dependency of global models on aggregated updates from multiple clients. In this work, we investigate the problem of federated unlearning using popular benchmark datasets, MNIST and CIFAR-10, which represent different modalities and complexities. Specifically, we propose a novel unlearning framework that efficiently removes the impact of selected data subsets from the global model while minimizing accuracy degradation on the remaining data. Our framework ensures that the computational overhead of unlearning remains low, making it feasible for large-scale deployments.

This paper makes the following contributions:

1. We proposed a novel federated unlearning framework that removes specific data contributions without retraining, addressing computational overhead.
2. We introduced a parameter partitioning approach, enabling selective freezing of neurons for targeted class unlearning.
3. We validate the proposed framework on the MNIST and CIFAR-10 datasets, showing its effectiveness in removing data influence while maintaining the overall accuracy.

Related Work

Federated learning has been widely studied for its ability to collaboratively train models without sharing raw data. Early works, such as McMahan et al.'s Federated Averaging (FedAvg) algorithm (McMahan et al. 2017), laid the foundation for efficient aggregation in FL systems. Privacy preservation in FL has been further enhanced through techniques like differential privacy (Dwork et al. 2006) and secure aggregation (Bonawitz et al. 2017). While these methods ensure data privacy, they do not address the challenge of data removal after training, highlighting the need for federated unlearning.

The concept of unlearning in machine learning was first explored in centralized settings. Cao and Yang (Cao and Yang 2015) proposed an efficient unlearning method for linear models, while Ginart et al. (Ginart et al. 2019) introduced the idea of certified data removal. Federated unlearning extends these concepts to distributed settings, requiring the removal of data influence without centralized access to the data. Recent advances in federated unlearning include gradient-based methods and knowledge distillation approaches (Wu, Zhu, and Mitra 2022), which aim to reconstruct global models without the unlearned data. However, these methods often suffer from performance trade-offs and scalability issues.

MNIST and CIFAR-10 are widely used benchmarks in FL research due to their simplicity and relevance. MNIST, a dataset of handwritten digits, is often used for testing the

fundamental capabilities of FL algorithms, including unlearning. CIFAR-10, on the other hand, is a more complex dataset of 10 classes of natural images, offering a robust platform for evaluating unlearning algorithms in diverse settings. Studies such as (Zhao et al. 2018) and (Kairouz, McMahan et al. 2021) have demonstrated the applicability of unlearning techniques on these datasets, providing insights into model performance and unlearning efficiency.

Methodology

Machine Unlearning

Machine unlearning refers to the process of effectively removing the influence of specific data points or subsets of data from a trained machine learning model (Bourtoule et al. 2021). In the context of federated learning (FL), this task becomes particularly challenging due to the decentralized nature of data and the lack of direct access to individual data points (Li et al. 2023). The goal of machine unlearning is to ensure that the model behaves as though the unlearned data was never used in its training.

Formally, let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ represent the entire training dataset, where (x_i, y_i) is the pair of input and it's label. Assume $\mathcal{D}_r \subset \mathcal{D}$ is the subset of data to be removed, and $\mathcal{D}_{\text{retained}} = \mathcal{D} \setminus \mathcal{D}_r$ is the remaining dataset. Given a trained model f_w with parameters w, the task of unlearning $\mathcal{D}_r$ is to find updated model parameters $w_{\text{unlearned}}$ such that:

$$f_{w_{\text{unlearned}}}(x) \approx f_{w_{\text{retrained}}}(x), \quad \forall x \in \mathcal{D}_{\text{retained}},$$

where $f_{w_{\text{retrained}}}$ is the model retrained only on $\mathcal{D}_{\text{retained}}$. The goal is to make $w_{\text{unlearned}}$ equivalent to $w_{\text{retrained}}$ in terms of performance and behavior on $\mathcal{D}_{\text{retained}}$, without the computational cost of retraining.

Our proposed method partitions the parameter space of a neural network into class-specific subsets, enabling selective unlearning by isolating and freezing neurons relevant to a specific class. The neural network $f(x)$ is decomposed into two components: $f(x) = h(g(x))$, where $g(x)$ represents the shared feature extraction layers (e.g., convolutional or fully connected layers), and $h(x)$ represents the class-specific output heads. The output heads,

$$h(x) = [h_1(x), h_2(x), \ldots, h_k(x)],$$

are tailored for each class, where $h_i(x)$ corresponds to the output head for class i, and k is the total number of classes.

During training, each class c is assigned a specific subset of neurons $N_{c,l}$ in each intermediate layer l (e.g., neurons 10–19 for class 0, 20–29 for class 1, etc.). Training updates only the neurons in $N_{c,l}$ when processing data for class c, with binary masks $M_{c,l}$ isolating the gradients for class-specific training. These masks ensure that the gradient ∇W_l is updated as:

$$\nabla W_l = M_{c,l} \cdot \nabla W_l,$$

where $M_{c,l}$ activates only the neurons relevant to class c.

To unlearn a specific class c, the method modifies both the forward and backward passes. In the forward pass, activations in the corresponding subset $N_{c,l}$ are set to zero ($a_{c,l} = 0 \, \forall l$), effectively removing the influence of the class. In the backward pass, gradients of parameters associated with $N_{c,l}$ are nullified:

$$\frac{\partial L}{\partial W_{c,l}} = 0 \quad \text{and} \quad \frac{\partial L}{\partial b_{c,l}} = 0,$$

freezing the parameters and ensuring the removal of class c's contribution without retraining.

Mathematically, the training data D_c for class c influences the model parameters θ_c through training, represented as:

$$\theta_c = \text{Train}(D_c, \theta).$$

The overall influence of D_c on the parameters θ is expressed as:

$$\Delta\theta_c = \sum_l \left(\frac{\partial L_c}{\partial W_{c,l}} \cdot W_{c,l} \right).$$

By freezing the subset $N_{c,l}$, the term $\Delta\theta_c$ is nullified, effectively eliminating class c's contribution to the model.

Federated Training

Federated learning (FL) enables multiple clients, such as mobile devices or edge servers, to collaboratively train a global machine learning model without sharing their raw data. Each client computes model updates locally using its private dataset, and a central server aggregates these updates to refine the global model. In the training phase, the process begins with gradient clustering (Briggs, Fan, and Andras 2020). Each client C_i computes local gradients $\Delta\theta_i$, which are then collected by the server into a set:

$$G = \{\Delta\theta_1, \Delta\theta_2, \ldots, \Delta\theta_N\}.$$

These gradients are analyzed and clustered into K distinct clusters, where each cluster represents a unique pattern of contribution. Specifically, a gradient $\Delta\theta_i$ is assigned to cluster k if it belongs to the set G_k, the k-th cluster.

Following clustering, the server maps each cluster G_k to a specific subset of neurons N_k in the global model, ensuring that N_k and N_k' are disjoint for $k \neq k'$. This mapping guarantees that neurons within each subset are exclusively updated by the clients associated with their corresponding cluster. Collectively, the neurons N in the model are the union of all cluster-specific subsets:

$$N = \bigcup_{k=1}^{K} N_k, \quad N_k \cap N_k' = \emptyset \text{ for } k \neq k'.$$

Next, in the cluster-specific training phase, only the neurons N_k associated with a specific cluster G_k are updated for the clients in that cluster. For a client C_i, the local training involves minimizing a loss function:

$$L_i = \frac{1}{|D_i|} \sum_{(x,y) \in D_i} \ell(f(x;\theta), y),$$

where ℓ is the cross-entropy loss, D_i is the local dataset, and θ are the model parameters. To maintain cluster-specific isolation, gradient updates are masked using a binary mask M_k, which ensures that only the neurons in N_k are updated for clients belonging to cluster k. The resulting update is:

$$\Delta\theta_i = M_k \cdot \nabla L_i, \quad \text{if Cluster}(i) = k.$$

Finally, the server aggregates the updates across all clusters. For each cluster k, the global model parameters are updated as:

$$\theta_{t+1} = \theta_t + \eta \cdot \frac{1}{|G_k|} \sum_{\Delta\theta_i \in G_k} \Delta\theta_i,$$

where η is the learning rate and $|G_k|$ is the number of updates in cluster k. This process ensures that the global model evolves based on the contributions of each cluster while preserving the distinctiveness of the clustered updates.

Unlearning Request

When an unlearning request is made for a specific data point or class, the system determines the parameters to neutralize by leveraging the established clustering. First, the gradients from clients are analyzed and compared with the clustered gradients.

$$G = \{\Delta\theta_1, \Delta\theta_2, \ldots, \Delta\theta_N\}.$$

The system identifies which cluster G_k the data point belongs to based on its contribution pattern.

Once the cluster G_k is determined, the corresponding subset of neurons N_k mapped to the cluster is isolated.

Experimental Setup

Dataset Details

The CIFAR-10 (32×32 color images, 10 classes) and MNIST (28×28 grayscale digit images) datasets were used for evaluation. Both datasets were normalized, split into training and testing sets, and preprocessed for consistency. The experiments focused on testing models with and without freezing specific neurons.

Training Setup and Hyperparameters

In the federated setting, client models were trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 64 for five epochs, across 100 global federated training rounds. Overall and per-class accuracies were recorded for analysis.

Results and Discussion

During federated training, gradients from all clients are clustered using t-SNE for dimensionality reduction and k-means for grouping similar scores. This clustering identifies neurons associated with specific labels, enabling effective unlearning by freezing these neurons upon request. Figure 1 illustrates the clustering process for the MNIST and CIFAR-10 datasets, showcasing ten distinct clusters corresponding to the ten classes in each dataset.

For the MNIST dataset (Table 1), baseline accuracies range from 90.59% to 98.50% across digits, demonstrating the global model's effectiveness. When digit-specific neurons are frozen, the accuracy for those digits drops significantly—e.g., 1.94% for digit 0 and 0.00% for digit 9—while non-frozen digits maintain their baseline accuracies, highlighting modularity and independence in the architecture.

Similarly, for the CIFAR-10 dataset (Table 2), freezing a specific class, such as "Airplane" or "Truck," reduces its accuracy to near-zero (e.g., 0.00% for "Airplane"), while non-frozen classes experience minimal changes. Minor accuracy shifts in non-frozen classes, such as "Ship" improving from 61.60% to 69.50% when "Airplane" is frozen, suggest shared dependencies among certain classes. These trends across MNIST and CIFAR-10 validate the architecture's generalizability, demonstrating its suitability for diverse data and federated learning tasks. The results affirm the utility of freezing for tasks like machine unlearning and robustness evaluation.

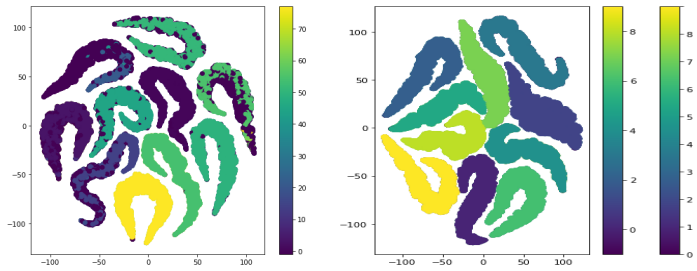

Figure 1: DBSCAN Clustering for CIFAR-10 and MNIST

Limitations and Future Work

Our method partitions the parameter space into subparameter spaces but cannot fully erase the influence of unlearned data. For instance, in the MNIST dataset, digits 4 & 5 retain residual accuracies of 6.31% & 9.30%, respectively, even with frozen neurons, indicating the model's generalization retains traces of unlearned data. Future work will investigate this residual accuracy & explore ways to mitigate it.

Additionally, the method is sensitive to cluster assignments for client parameters. Misaligned clusters can unintentionally unlearn data for other labels. We aim to improve neuron-to-parameter mapping or enhance clustering robustness to reduce such errors.

Conclusion

This paper presents an efficient federated unlearning framework that removes specific data contributions without requiring full retraining. By partitioning model parameters into class-specific subsets and selectively freezing neurons, the method ensures complete unlearning for targeted classes while preserving high accuracy for unaffected ones. Experimental results on MNIST and CIFAR-10 validate the approach's effectiveness in addressing privacy compliance and computational efficiency, highlighting its potential for broader federated learning applications.

References

Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H. B.; Patel, S.; Van Overveldt, D.; Charles, A.; and Riley, G. 2017. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 1175–1191.

Frozen Digit	Digit 0	Digit 1	Digit 2	Digit 3	Digit 4	Digit 5	Digit 6	Digit 7	Digit 8	Digit 9
Without Freezing	97.24%	98.50%	96.32%	90.59%	95.32%	93.95%	97.29%	96.40%	95.07%	92.27%
Frozen 0	**1.94%**	98.50%	96.71%	90.79%	95.32%	94.17%	97.81%	96.40%	95.38%	92.57%
Frozen 1	97.24%	**2.20%**	95.25%	95.15%	95.32%	95.63%	98.43%	96.40%	92.30%	94.95%
Frozen 2	99.18%	98.59%	**0.10%**	91.19%	96.23%	94.62%	97.49%	97.08%	96.10%	90.88%
Frozen 3	98.37%	98.41%	96.51%	**0.00%**	97.15%	96.64%	96.03%	96.40%	94.76%	93.66%
Frozen 4	98.98%	98.41%	94.38%	93.96%	**6.31%**	96.75%	96.87%	93.00%	94.76%	96.53%
Frozen 5	98.78%	98.06%	96.22%	95.15%	96.13%	**9.30%**	97.18%	96.30%	94.97%	94.15%
Frozen 6	98.88%	98.59%	96.71%	95.94%	95.42%	96.08%	**2.71%**	96.30%	90.55%	95.64%
Frozen 7	97.65%	98.59%	96.12%	95.64%	95.93%	95.85%	95.82%	**0.10%**	92.92%	94.95%
Frozen 8	99.18%	98.85%	96.51%	95.74%	96.95%	94.06%	95.82%	96.11%	**10.88%**	93.95%
Frozen 9	98.88%	98.77%	97.38%	94.46%	98.47%	96.64%	95.51%	96.30%	90.97%	**0.00%**

Table 1: Global Accuracy of MNIST digits without freezing and with frozen digits.

Frozen Class	Airplane	Automobile	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
Without Freezing	50.80%	45.50%	43.70%	22.70%	20.30%	39.20%	65.00%	55.00%	61.60%	62.30%
Frozen Airplane	**0.00%**	46.40%	47.60%	23.00%	21.10%	39.40%	65.00%	56.30%	69.50%	63.80%
Frozen Automobile	39.10%	**0.00%**	33.00%	24.90%	33.70%	33.00%	72.00%	63.30%	72.70%	57.30%
Frozen Bird	59.00%	60.10%	**0.00%**	27.20%	55.10%	39.10%	62.10%	49.40%	56.60%	52.50%
Frozen Cat	54.90%	65.80%	24.60%	**0.10%**	55.60%	36.50%	48.00%	59.20%	60.40%	48.40%
Frozen Deer	47.60%	74.10%	40.00%	27.60%	**0.00%**	39.20%	60.90%	57.50%	54.00%	50.30%
Frozen Dog	58.50%	64.90%	24.50%	44.10%	39.00%	**0.00%**	54.90%	57.50%	61.70%	59.00%
Frozen Frog	57.40%	71.90%	36.30%	41.20%	47.70%	35.60%	**0.00%**	40.80%	59.30%	49.40%
Frozen Horse	51.10%	52.90%	48.10%	25.60%	11.90%	53.10%	58.60%	**0.50%**	64.00%	64.40%
Frozen Ship	49.70%	43.20%	22.30%	22.60%	37.60%	45.90%	63.00%	65.40%	**0.60%**	69.00%
Frozen Truck	54.60%	75.30%	26.30%	31.40%	46.60%	23.30%	64.10%	57.30%	56.70%	**0.10%**

Table 2: Global Accuracy of CIFAR-10 classes without freezing and with frozen classes.

Bourtoule, L.; Chandrasekaran, V.; Choquette-Choo, C. A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; and Papernot, N. 2021. Machine Unlearning. In *Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP)*, 141–159. IEEE.

Briggs, C.; Fan, Z.; and Andras, P. 2020. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In *2020 international joint conference on neural networks (IJCNN)*, 1–9. IEEE.

Cao, Y.; and Yang, J. 2015. Towards making systems forget with machine unlearning. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy (S&P)*, 463–480.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Theory of Cryptography Conference (TCC)*, 265–284.

Ginart, A.; Guan, M. Y.; Valiant, G.; and Zou, J. 2019. Making AI forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Kairouz, P.; McMahan, H. B.; et al. 2021. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2): 1–210.

Li, Y.; Chen, C.; Zheng, X.; and Zhang, J. 2023. Federated Unlearning via Active Forgetting. *arXiv preprint arXiv:2307.03363*.

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 1273–1282.

Romandini, N.; Mora, A.; Mazzocca, C.; Montanari, R.; and Bellavista, P. 2024. Federated Unlearning: A Survey on Methods, Design Guidelines, and Evaluation Metrics. *arXiv preprint arXiv:2401.05146*.

Wu, C.; Zhu, S.; and Mitra, P. 2022. Federated Unlearning with Knowledge Distillation. *arXiv preprint arXiv:2201.09441*.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Feder-

ated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*, 37(3): 50–60.

Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; and Chandra, V. 2018. Federated learning with non-IID data. In *Proceedings of the International Conference on Machine Learning (ICML)*.