Summary on Federated Unlearning to Efficiently Erase a Client in FL

Aman Patel

June 29, 2024

Introduction

In the realm of federated learning, multiple clients collaboratively train a global model while keeping their data localized to ensure privacy. However, situations arise when client data needs to be removed from global model due to certain intended malicious training by some client or under right to be forgotten. Traditional unlearning methods either remove client's data forcing model to make completely random predictions or retraining the model from scratch excluding target client's data, which can be thought as an ideal way. But *FL* being a decentralized approach and having to keep good accuracy over retained client's data after unlearning, this paper presents a more efficient way to remove specific client's data while maintaining high model performance compared to gold standard and significantly reducing the computational and communication costs.

Methodology and Important Findings

With unbounded loss functions, gradient ascent step keeps on maximizing the loss and eventually model becomes similar to a random model and that's not required actually since we want the best possible performance over retained clients data. Thus to unlearn the model in a **constrained environment**, this paper proposes to use *Projected Gradient Descent(**PGD**)*. Instead of retraining from scratch, this paper performs federated unlearning by *(i)* local unlearning at target client i by using *PGD* and (ii) performing only a **few rounds** of *FL* to boost its performance starting from locally unlearned model. Below is the detailed steps used in unlearning via *PGD* :

1. Projected Gradient Descent

To not violate the constraint, it is ensured that locally unlearned model is sufficiently close to a **reference model** θ_{ref} and specifically it can be either a global model before incorporating target client's update or an **aggregation(*average*, here) of the local models from other clients**, ensuring it represents the collective knowledge excluding the target client's data. The latter is used in this research paper and to keep model close to θ_{ref}, client i then optimizes over the model parameters that lie in the **ℓ2-*norm* ball** of radius δ(hyperparameter) around θ_{ref}. We start with target client model $\theta_0 = \theta^t$. The **projection operator** ω moves the parameters in the direction of the gradient to maximize the loss by;

$$\theta' = \theta + \eta \nabla_\theta L(\theta) \tag{1}$$

where η is the learning rate and L is the loss function.
Then the updated parameters are projected back onto the ℓ2-*norm* ball around the reference

model to ensure that the constraints are satisfied. Projection operator ω would work as:

$$\theta = \theta_{ref} + \frac{\theta^{'} - \theta}{max(1, \frac{||\theta' - \theta_{ref}||_2}{\delta})} \tag{2}$$

This ensures that the updated model stays within the specified distance δ from the reference model. The **gradient ascent** and **projection steps** are repeated for a fixed number of iterations or until convergence to local maxima due to **non-convex** nature of loss function.

2. Evaluation

The two key phenomena ***backdoor triggers*** and ***flipping*** are used here to assess unlearning by evaluating metrics like efficacy, fidelity and efficiency. First the accuracy on backdoored data(malicious patterns or inputs) is computed and our proposed *PGD*-based unlearning showed as low accuracy as that to retraining from scratch, which is a good sign. Also the fidelity of our approach on clean images(no backdoor triggers) showed similar performance to retraining, maintaining a good performance on retained data as well. Whereas in flipping, our unlearned model is tested on flipped data(having intentionally incorrect labels) to evaluate how well the unlearning process reverts the global model predictions to their original state. Post-unlearning, low accuracy on flipped data and increase in overall accuracy showed successful restoration of model's performance by removing the incorrect association of target client, comparable to retraining. One of the most important evaluations is efficiency. Consider N clients and model size M bytes, then total **communication cost** would be $2{\times}N{\times}M{\times}R$ (both upstream and downstream communication and R is iterations). Our model showed *substantial reduction* in communication rounds from $R{=}T$ to $R{=}K$ improving performance by upto 24×.

Limitations

In *FL* scenarios, local models trained on non i.i.d(independent and identical distribution) data would lead to high variance among updated parameters resulting in sub-optimal global model, and local unlearning using *PGD* might not generalize well if reference model θ_{ref} and target client's distribution differ significantly. Deviating from *cross-silo* setting where there are large number of clients each having a huge amount of data, typical empirical evaluation for unlearning a single client could be quite challenging in terms of more consumption of resources and iterations required to achieve similar performance, as theoretical analysis cannot be guaranteed due to non-convex loss functions. Constructing reference model by taking mean of other clients parameters except target client might not work in case of hundreds of clients where all have very diverse model parameters.

Future Scope

The selection of reference model can get better by taking weighted average of retained clients weights like done in *FL* rather than just mean. Apart from gradient descent, some advanced optimizers(e.g. Adam, AdaGrad) could be used during the unlearning process to enhance convergence speed and accuracy. In scenarios where multiple clients need to be removed from global *FL* model, techniques such as making group of clients with similar data distribution and clustered unlearning may be adapted to unlearn the patterns faster and achieve higher accuracy.