Re-Weighted Softmax Cross-Entropy to Control Forgetting in Federated Learning

Gwen Legate [1 2] **Lucas Caccia** [3 2] **Eugene Belilovsky** [1 2]

Abstract

In Federated Learning a global model is learned by aggregating model updates computed at a set of independent client nodes. To reduce communication costs, multiple gradient steps are performed at each node prior to aggregation. A key challenge in this setting is data heterogeneity across clients resulting in differing local objectives which can lead clients to overly minimize their own local objective, diverging from the global solution. We show that individual client models experience a catastrophic forgetting with respect to data from other clients and propose an efficient approach that modifies the cross-entropy objective on a per-client basis by re-weighting the softmax logits prior to computing the loss. This approach shields classes outside a client's label set from abrupt representation change and we empirically demonstrate it can alleviate client forgetting and provide consistent improvements to standard federated learning algorithms. Our method is particularly beneficial under the most challenging federated learning settings where data heterogeneity is high and client participation in each round is low.

1. Introduction

Federated Learning (FL) is a distributed machine learning paradigm in which a shared global model is learned from a decentralized set of data located at a number of independent client nodes (McMahan et al., 2017; Konečnỳ et al., 2016). Driven by communication constraints, FL algorithms typically perform a number of local gradient update steps before synchronizing with the global model. Under realistic settings, client data will often have non-i.i.d. distributions and such data heterogeneity across clients has direct implications on convergence and performance of FL algorithms (Zhao et al., 2018). For supervised multi-class classification, users may possess no data whatsoever from one or several classes present in the underlying global distribution. Data inhomogeneity across clients frequently induces client drift, a phenomenon in which clients progress too far towards optimizing their own local objective, leading to a solution that has severely "drifted" from an optimal global solution (Karimireddy et al., 2020).

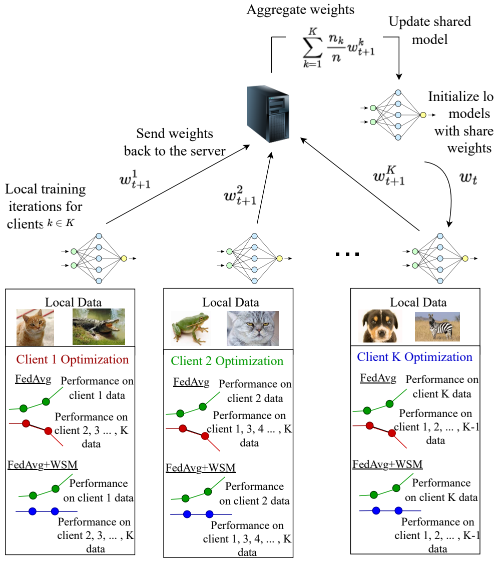

Aggregate weights
$\sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$
Update shared model
Initialize local models with shared weights
Send weights back to the server
w_{t+1}^1
w_{t+1}^2
w_{t+1}^K
w_t
Local training iterations for clients $k \in K$
...
Local Data
Client 1 Optimization
FedAvg
Performance on client 1 data
Performance on client 2, 3 ... , K data
FedAvg+WSM
Performance on client 1 data
Performance on client 2, 3, ... , K data
Local Data
Client 2 Optimization
FedAvg
Performance on client 2 data
Performance on client 1, 3, 4 ... , K data
FedAvg+WSM
Performance on client 2 data
Performance on client 1, 3, 4, ... , K data
Local Data
Client K Optimization
FedAvg
Performance on client K data
Performance on client 1, 2, ... , K-1 data
FedAvg+WSM
Performance on client K data
Performance on client 1, 2, ... , K-1 data

Figure 1. Illustration of catastrophic forgetting within client rounds. A global model with knowledge of all classes is sent to all clients participating in a given FL round. Local training increases the client model performance on the client's local distribution but tends to simultaneously decrease performance with respect other clients distributions which leads to poor aggregation and overall model performance.

In continual learning (CL), a model is trained on a number of tasks sequentially and the learner needs to learn each new task without forgetting knowledge obtained from the preceding tasks, a phenomenon termed catastrophic forgetting. (McCloskey & Cohen, 1989). Similar to FL, data heterogeneity in CL is challenging since different tasks typically contain data drawn from different underlying distributions and we are able to draw a connection between the catastrophic forgetting problem in CL and the client drift problem in FL. Consider one round of FL in which K

[1] Concordia University, Montreal, Quebec, Canada [2] Mila Quebec Artificial Intelligence Institute, Montreal, Quebec, Canada [3] McGill University, Montreal, Quebec, Canada. Correspondence to: Gwen Legate <gwendolyne.legate@mila.quebec>.

Workshop of Federated Learning and Analytics in Practice, colocated with 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. Copyright 2023 by the author(s).

random clients are selected, initialized with a copy of the current global model and perform a pre-determined number of local update steps optimizing the objective on their local data. As local training proceeds, the model becomes increasingly biased towards a given client and as discussed in Lesort (2022); Gupta et al. (2022), this can cause client models to rely on spurious correlations to improve their in-distribution performance, thus creating a situation in which local models experience catastrophic forgetting with respect to data of the other clients. Naturally, aggregating models that have deviated from a joint solution will lead to degraded results with respect to the global objective. We denote this problem as *local client forgetting* and direct the reader to figure 1 which illustrates its effects within a round of federated learning. We hypothesize that reducing local client forgetting will moderate the decrease in performance with respect to other clients data at individual client models thus increasing the ability of local models to generalize to the data distributions of other clients.This increased ability to generalize should improve the loss of individual models over the combined data and we therefore propose to reduce client drift by tackling local client forgetting.

There are numerous approaches to tackle catastrophic forgetting in the continual learning literature (Kirkpatrick et al., 2017; Li & Hoiem, 2017; Chaudhry et al., 2019; Schwarz et al., 2018; Davari et al., 2022); however, many of these are impractical in the FL setting. Experience Replay methods (Chaudhry et al., 2019) require access to other clients' data, violating data communication constraints of FL. Similarly, many regularization methods such as elastic weight consolidation (EWC) Kirkpatrick et al. (2017) require communicating additional information and can additionally require many steps to converge due to the additional conflicting objectives (Aljundi et al., 2019). For the supervised continual learning setting, Caccia et al. (2022); Ahn et al. (2021) proposed a modification of the standard cross entropy (CE) objective function that truncates the softmax denominator, removing terms corresponding to classes from old tasks. A variant of this method inspired by the long-tailed recognition methods (Ren et al., 2020) was also recently introduced in Jodelet et al. (2022). This simple approach mitigates catastrophic forgetting by reducing the bias on the model to avoid predicting old classes. Inspired by the parallels between client drift in FL and catastrophic forgetting in CL, we propose an adaptation of the CL method from Caccia et al. (2022); Ahn et al. (2021); Jodelet et al. (2022) to modify the loss function of each client based on its class distribution using a re-weighted softmax. We empirically demonstrate this approach can drastically reduce client level forgetting in the heterogeneous setting.

2. Background and Methods

In federated optimization, training data is distributed and optimization occurs over K clients with each client $k \in 1, ..., K$ having data $\mathbf{X}_k$ drawn from distribution D_k. We define $n_k = |\mathbf{X}_k|$ and $n = \sum_{k=1}^{K} n_k$ for n samples. The data $\mathbf{X}_k$ at each node may be drawn from different distributions and/or may be unbalanced with some clients possessing more samples than others. The typical objective function for federated optimization is given by equation 1 with $\mathcal{L}(\mathbf{w}, \mathbf{X}_k)$ measuring client k's local objective, and $\mathbf{w}$ representing the global parameters. In this work $\mathcal{L}$ is restricted to cross entropy (CE) loss. A commonly used federated optimization algorithm is FedAvg introduced in McMahan et al. (2017).

$$\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{k=1}^{K} \frac{n_k}{n} \mathcal{L}(\mathbf{w}, \mathbf{X}_k), \tag{1}$$

Re-weighted Softmax Cross Entropy Consider a neural network $f : \mathcal{R}^D \to \mathcal{R}^C$ where C is the total number of classes. The standard cross entropy is given by equation 2 where $y(\mathbf{x})$ is the label of $\mathbf{x}$ and $\mathcal{C}$ is the set of all classes available to the clients.

$$\mathcal{L}_{CE}(\mathbf{X}_k, \mathbf{w}) = -\sum_{\mathbf{x}\in\mathbf{X}_k} \log \frac{\exp(f_{\mathbf{w}}(\mathbf{x})_{y(\mathbf{x})})}{\sum_{c\in\mathcal{C}} \exp(f_{\mathbf{w}}(\mathbf{x})_c)} \tag{2}$$

$$= -\sum_{\mathbf{x}\in\mathbf{X}_k} \Big[f_{\mathbf{w}}(\mathbf{x})_{y(\mathbf{x})} - \log\Big(\sum_{c\in\mathcal{C}} \exp(f_{\mathbf{w}}(\mathbf{x})_c\Big)\Big] \tag{3}$$

One interpretation of this classical loss function considers the two terms as a tightness term (the first term) which brings samples close to their representative classes and a contrast term (the second term) which pushes them apart from other classes (Boudiaf et al., 2020). We note similar loss functions can be interpreted from an energy modeling view (Liu, 2020). We now modify the standard CE using the re-weighted softmax (WSM) to give our per-client objective function in Equation 4 where $\boldsymbol{\beta_k}$ is a vector containing the proportions of each class present in the client dataset and $\beta_c \in \boldsymbol{\beta_k}$ is the proportion of label c present in the dataset.

$$\mathcal{L}_{WSM}(\mathbf{X}_k, \mathbf{w}) = -\sum_{\mathbf{x}\in\mathbf{X}_k} \Big[f_{\mathbf{w}}(\mathbf{x})_{y(\mathbf{x})} - \log\Big(\sum_{c\in\mathcal{C}} \beta_c \exp(f_{\mathbf{w}}(\mathbf{x})_c)\Big)\Big] \tag{4}$$

If we define $\mathbf{Y}_k$ as the set of labels of samples $\mathbf{X}_k$ belonging to client k then in equation 4 the weighting β introduced in the second term is a function only of $\mathbf{Y}_k$ as opposed to the complete set of labels, $\mathbf{Y}$ in the cross entropy loss from Equation 2. In a highly imbalanced class scenario commonly studied in FL, many β_c will be zero very close to zero, thus removing or substantially degrading the contribution of that class to the contrast term. Optimizing $\mathcal{L}_{CE}$ through multiple gradient steps during a client round can lead to a drastic increase in $\mathbb{E}_{x,y\sim D_{j\neq k}}[l_{CE}(\mathbf{x}, \mathbf{y})]$, where D_j are the distributions of clients other than client k, this occurs because the contrast term encourages classes not present at client k to never be predicted. WSM modifies the original local objective function to avoid excessive pressure driving up the loss of other client data, classes not present at client k are ignored by the local optimization forcing the client to learn by adapting the model's internal representation of the classes present in its training data, rather than abruptly shifting representations of classes outside its training set (Caccia et al., 2022). We demonstrate this leads to a reduction in local client forgetting. We note that at test time we use the unweighted softmax as we train a global model that must be able to perform inference on unseen data and clients.

Local client forgetting For a multi-class classification problem, we denote the accuracy on a client k's local test data $Acc_k(\mathbf{w})$, where $\mathbf{w}$ are the model parameters. Local client forgetting is defined according to Eq. 5 where $\mathbf{w}_t^i$ refers to the model of client i at round t after it has completed local training (prior to aggregation) and $\mathbf{w}_{t-1}$ is the global model (after aggregation) at the end of round $t-1$.

$$F_{ki} = Acc_k(\mathbf{w}_{t-1}) - Acc_k(\mathbf{w}_t^i) \tag{5}$$

We define an average forgetting for a client k's model according to equation 6

$$F_k = \frac{1}{K-1} \sum_{i \neq k} F_{ki} \tag{6}$$

3. Experiments

In this section, we present the empirical results analyzing local client forgetting and our WSM approach. Section 4 contains additional experiments in which we ablate different FL settings and Appendix A shows the effect of using a different model architecture, LeNet (LeCun et al., 1998).

We utilize CIFAR-10, CIFAR-100 (Krizhevsky & Hinton, 2009) and FEMNIST (Caldas et al., 2018) datasets in our experiments, our primary setting considers 100 clients where each client possesses their own training and validation sets according to their own unique distribution. To facilitate this, the entire training set is separated into equally sized non-i.i.d. partitions using the Dirichlet distribution parameterized by $\alpha = 0.1$(Hsu et al., 2019).

We follow the experimental settings of Reddi et al. (2020). Clients are sampled without replacement for each round but can be selected again in subsequent rounds. The fraction of clients sampled is 10% for CIFAR-10 and FEMNIST datasets and 2% for CIFAR-100. Our primary evaluations train a ResNet-18 over 4000 communication rounds for 3 local epochs, using a mini-batch of size 64 and a learning rate of 0.05 for CIFAR-10 and CIFAR-100. FEMNIST, which converges faster due to its large size, is trained for 3000 rounds with all other settings the same as for the CIFAR datasets. We use SGD as our optimizer, with weight decay of 1×10^{-4} following Yao et al. (2021); Hsu et al. (2019). In these experiments, we observe that FedAvg often (though not always) performs better with group normalization as indicated by Hsieh et al. (2020) while FedAvg+WSM is able to perform well with both group and batch normalization, very frequently achieving the best results with batch normalization. We therefore treat the normalization method as a hyper-parameter and provide the best result obtained out of both batch and group normalization for both methods at each learning rate.

Forgetting During a Federated Round In Figure 2 we show plots of forgetting throughout the training process for each federated algorithm evaluated. We observe that on average forgetting is high in FedAVG and is substantially reduced especially at the end of training when WSM is applied. The FedProx, FedNova and SCAFFOLD algorithms were all developed to address the challenge of data heterogenaity in FL and each of them demonstrate improved forgetting scores with the application of WSM indicating that it can indeed benefit algorithms already developed to address the client heterogeneity problem. Overall our results show WSM has the ability to greatly limit the effects of local client forgetting by ignoring classes outside of a client's distribution and focusing learning on the classes present. We demonstrate in section 3 that this narrower focus leads to better overall performance after aggregation. These observations are further supported by the forgetting heatmaps provided in Appendix E which show per client forgetting when evaluated on the datasets of other clients participating in a given training round. Having shown that WSM can reduce the local client forgetting, we now study its effect on the aggregated models.

Evaluation of WSM In this section we demonstrate how WSM used in combination with FedAvg can improve model performance and convergence. Table 1 shows model performance across a range of learning rates for CIFAR-10, CIFAR-100, and FEMNIST datasets. The reported values are the average across three seeds with the standard deviation following in brackets. The best preforming model for each learning rate is shown in bold and the best overall result for each dataset is indicated by a green (red) box for WSM (FedAvg). Table 1 demonstrates that WSM substantially improves performance for both CIFAR datasets with a 2.2% and 1.3% improvement for CIFAR-10 and CIFAR-100, respectively. For FEMNIST dataset the results show the best performing models are within statistical error of one another. We do however observe strong results using WSM with higher learning rates under which regime we are able to obtain faster convergence. WSM also makes the hyper-parameters easier to tune since it performs well over a large range of learning rates. For example, learning rates between 0.03 and 0.1 for FedAvg+WSM have remarkably steady performance between 85.5% and 85.7% while we observe no such consistent performance for FedAvg.

WSM for Heterogenous FL methods We now demonstrate the effectiveness of WSM over a range of FL algorithms. In Table 2 we show the results of applying WSM to SCAFFOLD, Fed Nova and FedProx. SCAFFOLD and FedProx are optimization based methods specifically designed to address the problem of data heterogeneity and Fed Nova too is designed to improve performance on heterogeneous data. In these experiments we use 30 clients

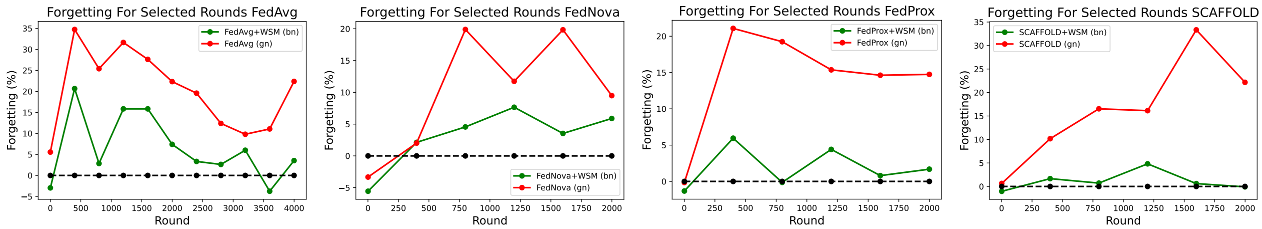


Figure 2. Average forgetting (F_k) for selected rounds of training We observe high forgetting for FedAvg (left) which is substantially reduced by applying WSM, especially towards the end of training. FedNova (center left) and FedProx (center right) exhibit less forgetting than FedAvg but still benefit substantially from the application on WSM. SCAFFOLD (right) seems to benefit the most from the application of WSM. Indeed, we observe that WSM produces levels of forgetting close to zero throughout training.

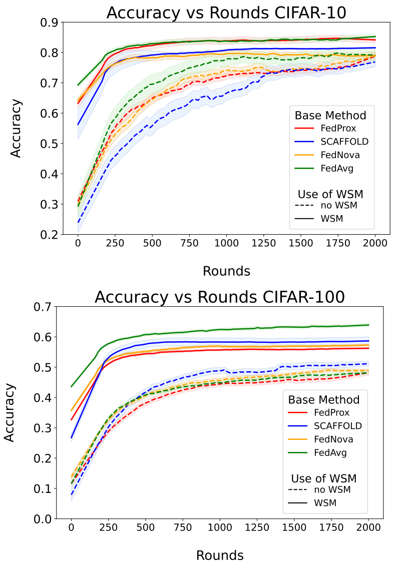
Table 1. Accuracy results of FedAvg with and without WSM for different hyper-parameters.

Method	lr	*Dataset* CIFAR-10	CIFAR-100	FEMNIST
FedAvg	0.5	0.326(0.098)	0.292(0.012)	0.542(0.087)
FedAvg+WSM		**0.792(0.006)**	**0.426(0.003)**	**0.837(0.002)**
FedAvg	0.3	0.791(0.013)	0.384(0.013)	0.769(0.006)
FedAvg+WSM		**0.834(0.008)**	**0.467(0.015)**	**0.844(0.010)**
FedAvg	0.1	0.724(0.027)	0.500(0.016)	0.835(0.002)
FedAvg+WSM		**0.855(0.004)**	**0.514(0.009)**	**0.848(0.006)**
FedAvg	0.07	0.826(0.007)	0.437(0.007)	**0.827(0.006)**
FedAvg+WSM		**0.856(0.005)**	**0.553(0.018)**	0.826(0.019)
FedAvg	0.05	0.827(0.004)	0.464(0.001)	**0.853(0.004)**
FedAvg+WSM		**0.858(0.003)**	**0.564(0.007)**	0.842(0.005)
FedAvg	0.03	0.836(0.005)	0.431(0.020)	**0.835(0.006)**
FedAvg+WSM		**0.857(0.005)**	**0.581(0.005)**	0.834(0.003)
FedAvg	0.01	0.815(0.003)	0.431(0.005)	**0.830(0.002)**
FedAvg+WSM		**0.845(0.006)**	**0.574(0.006)**	0.800(0.019)
FedAvg	0.007	0.817(0.007)	0.426(0.005)	**0.821(0.011)**
FedAvg+WSM		**0.841(0.004)**	**0.568(0.003)**	0.800(0.007)
FedAvg	0.005	0.802(0.010)	0.426(0.002)	**0.819(0.022)**
FedAvg+WSM		**0.826(0.009)**	**0.554(0.004)**	0.773(0.013)

Table 2. Accuracy results of FedAvg , SCAFFOLD, FedNova, and FedProx with and without WSM.

Method	CIFAR-10	CIFAR-100
FedAvg	0.800(0.033)	0.494(0.008)
FedAvg+WSM	**0.864(0.008)**	**0.640(0.002)**
Scaffold	0.794(0.033)	0.532(0.003)
Scaffold+WSM	**0.812(0.012)**	**0.572(0.028)**
FedNova	0.780(0.010)	0.500(0.005)
FedNova+WSM	**0.789(0.13)**	**0.575(0.002)**
FedProx	0.776(0.028)	0.487(0.10)
FedProx+WSM	**0.818(0.023)**	**0.562(0.005)**

Figure 3. Convergence plots of different algorithms with and without WSM. We observe that WSM variants lead to substantially better convergence for all compared methods.



and train the models over only 2000 rounds, since some of these algorithms have a high overhead and many hyper-parameters to search. We keep the same local batch size of 64 and 3 local iterations from the base case. Combining each of the methods with WSM we confirm our hypothesis that mitigating local client forgetting via WSM provides improvement over the base cases for each method and in most cases this improvement is substantial. Our results show that even for algorithms already partially addressing the heterogeneity problem WSM can provide benefits. We also observe that combining WSM with FedAvg provides the best overall results in Table 2. This suggests that algorithms based on constrained optimization, e.g. SCAFFOLD, may over constrain the improvement possible over a given round. Additionally, we remark that combining WSM with a baseline method generally provides a stronger performance from the very beginning of training since both CIFAR-10 and CIFAR-100 curves in Figure 3 showing the training progression for methods combined with WSM all start off with higher reported accuracy than their FedAvg only counterparts and this improvement persists for the duration of training. Work on critical learning periods, where critical learning periods are defined as the early epochs of a training regime, have shown they can determine the final quality of a deep neural network for traditional ML methods (Jastrzębski et al., 2018; Achille et al., 2017). Yan et al. (2021) investigate critical learning periods in the FL setting and discover they do indeed exist consistently in FL. We can thus hypothesize that the early training advantage we see when applying WSM may be having a positive impact on its consistent ability to outperform other FL algorithms.

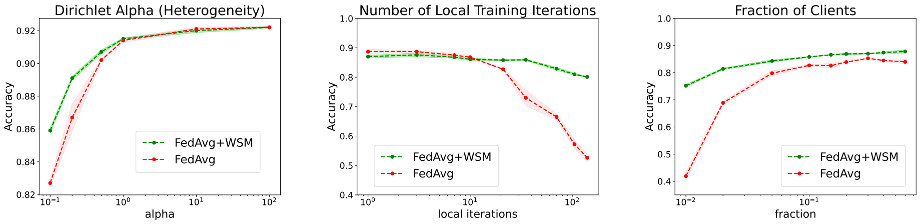


Figure 4. Ablation plots of data heterogeneity, local iterations and the fraction of clients selected at each round. WSM provides performance increases under most of the conditions with the most significant advantages provided for very heterogeneous data distributions and smaller fractions of client participation.

4. Ablations

We now further study the behavior of WSM in combination with FedAvg under different FL settings where we ablate one setting at a time. Except where we specify the value of the parameter being ablated, hyper-parameters for the ablation studies are the same as for our base case described in section 3. We will show that WSM is particularly advantageous when α is low (high heterogeneity) and when few clients are selected to participate in each FL round.

Parameter α of the Dirichlet Distribution The Dirichlet distribution is parameterized by α, as $\alpha \to 0$ client distributions become increasingly heterogeneous and as $\alpha \to \infty$, each client data edges closer towards the same i.i.d. distribution. We refer the reader to section D.1 where we illustrate how label distributions change as a function of α. Figure 4 shows how data heterogeneity affects model performance and highlights the increasingly significant effect WSM has as α decreases. The largest margin of improvement over FedAvg occurs when $\alpha = 0.01$ (our most heterogeneous setting) since as the data becomes increasingly homogeneous the gap between cross entropy with and without WSM shrinks as the distribution approaches i.i.d. and the two methods are equivalent. While WSM continues to offer a performance increase over the entire range of α except for $\alpha = 100$, we conclude WSM is most advantageous when clients labels distributions are imbalanced.

Fraction of Participating Clients These experiments focus on the fraction, p of clients participating in each update round where we vary the fraction from 1% to 60%. for each experiment the number of participating clients is constant for all rounds of training. We observe the largest performance gap between FedAvg+WSM and FedAvg when the number of participating clients is low. This feature is significant since low client participation is a known feature of real world FL settings (McMahan et al., 2017). We hypothesize the performance gap as a function of p between the FedAvg and FedAvg+WSM is due to the larger impact of local client forgetting when we limit communication capability. Unless we actively take steps to control forgetting, non-participating clients will have their data distributions forgotten because they will be unable to contribute their updates to the global model. As p increases, we observe the performance gap between the two methods narrow since more clients will have the opportunity to be selected at each round and "remind" the model of their data distributions.

Local Iterations A local iteration is defined as one gradient step at the client during a federated round. In our reference setup 7 local iterations is equivalent to one local epoch. After 21 local iterations we observe a sharp decrease in accuracy for FedAvg as local iterations increase. While FedAvg+WSM does also experience a drop in accuracy as local iterations increase, its drop in accuracy is much less pronounced. This result is in line with our expectations since increasing the number of local training iterations will cause the effects of local client forgetting become more serious. The fact that WSM allows local models to train for more iterations has important implications for the communication costs in FL and since communication is a serious bottleneck in FL our method offers a valuable option to speed up training in a federated setting.

5. Conclusion

We take a deeper look at the *local client forgetting* problem and show that when a client performs local updates during FL, it risks overly optimizing its local objective leading to forgetting on other subsets of data. Local client forgetting degrades the performance of the global model and we show this phenomenon is especially severe in cases where there is a significant distribution mismatch across clients. First making the connection with the catastrophic forgetting problem in the continual learning, we propose a client level modification of the objective function which we call the weighted softmax. We show empirically that WSM allows us to mitigate client level forgetting by demonstrating improved performance for the FedAvg algorithm when combined with WSM across a range of learning rates. We also demonstrate these improvements are not limited to FedAvg since we also observe significant performance increases when WSM is applied to SCAFFOLD, FedNova and FedProx. An ablation study demonstrated WSM is particularly effective in the regime of highly heterogeneous client datasets and/or when a small percentage of clients are selected at each round. Our results indicate that addressing local client forgetting in general is an important consideration for federated learning optimization, one that bears closer scrutiny.

Acknowledgements

We acknowledge support from IVADO Strategic Research Grants on Human Health and Secondary use of data and FQRNT New Scholar grant [E.B.]. We acknowledge resources provided by the digital research alliance and calcul quebec

References

Achille, A., Rovere, M., and Soatto, S. Critical learning periods in deep neural networks. *arXiv preprint arXiv:1711.08856*, 2017.

Ahn, H., Kwak, J., Lim, S., Bang, H., Kim, H., and Moon, T. Ss-il: Separated softmax for incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 844–853, 2021.

Aljundi, R., Belilovsky, E., Tuytelaars, T., Charlin, L., Caccia, M., Lin, M., and Page-Caccia, L. Online continual learning with maximal interfered retrieval. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/15825aee15eb335cc13f9b559f166ee8-Paper.pdf.

Boudiaf, M., Rony, J., Ziko, I. M., Granger, E., Pedersoli, M., Piantanida, P., and Ayed, I. B. A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses. In *European conference on computer vision*, pp. 548–564. Springer, 2020.

Caccia, L., Aljundi, R., Asadi, N., Tuytelaars, T., Pineau, J., and Belilovsky, E. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=N8MaByOzUfb.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečný, J., McMahan, H. B., Smith, V., and Talwalkar, A. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P. K., Torr, P. H., and Ranzato, M. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Davari, M., Asadi, N., Mudur, S., Aljundi, R., and Belilovsky, E. Probing representation forgetting in supervised and unsupervised continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16712–16721, 2022.

Gupta, S., Ahuja, K., Havaei, M., Chatterjee, N., and Bengio, Y. FL games: A federated learning framework for distribution shifts. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022. URL https://openreview.net/forum?id=UyfdXCeelfy.

Hsieh, K., Phanishayee, A., Mutlu, O., and Gibbons, P. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pp. 4387–4398. PMLR, 2020.

Hsu, T.-M. H., Qi, H., and Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Jastrzębski, S., Kenton, Z., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. On the relation between the sharpest directions of dnn loss and the sgd step length. *arXiv preprint arXiv:1807.05031*, 2018.

Jodelet, Q., Liu, X., and Murata, T. Balanced softmax cross-entropy for incremental learning with and without memory. *Computer Vision and Image Understanding*, 225:103582, 2022.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Konečný, J., McMahan, H. B., Ramage, D., and Richtárik, P. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lesort, T. Continual feature selection: Spurious features in continual learning. *arXiv preprint arXiv:2203.01012*, 2022.

Li, Z. and Hoiem, D. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Liu, W. e. a. Energy-based out-of-distribution detection. *Neural information processing systems*, 2020.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Ren, J., Yu, C., Ma, X., Zhao, H., Yi, S., et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33: 4175–4186, 2020.

Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., and Hadsell, R. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pp. 4528–4537. PMLR, 2018.

Yan, G., Wang, H., and Li, J. Critical learning periods in federated learning. *arXiv preprint arXiv:2109.05613*, 2021.

Yao, D., Pan, W., Dai, Y., Wan, Y., Ding, X., Jin, H., Xu, Z., and Sun, L. Local-global knowledge distillation in heterogeneous federated learning with non-iid data. *arXiv preprint arXiv:2107.00051*, 2021.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Table 3. Best accuracy results of FedAvg with and without WSM using the LeNet architecture.

	CIFAR-10	CIFAR-100
FedAvg	0.608 ± 0.010	0.198 ± 0.004
FedAvg+WSM (ours)	**0.624 ± 0.009**	**0.274 ± 0.007**

Table 4. Accuracy results of FedAvg with and without WSM for different settings of client learning rates using the LeNet architecture.

	lr	CIFAR-10	CIFAR-100
FedAvg	0.07	0.589 ± 0.012	0.119 ± 0.002
FedAvg+WSM (ours)		**0.624 ± 0.009**	**0.180 ± 0.013**
FedAvg	0.05	0.580 ± 0.051	0.154 ± 0.011
FedAvg+WSM (ours)		**0.615 ± 0.004**	**0.217 ± 0.009**
FedAvg	0.03	0.605 ± 0.014	0.168 ± 0.001
FedAvg+WSM (ours)		**0.621 ± 0.009**	**0.251 ± 0.007**
FedAvg	0.01	0.608 ± 0.010	0.168 ± 0.044
FedAvg+WSM (ours)		**0.624 ± 0.009**	**0.274 ± 0.007**
FedAvg	0.007	0.591 ± 0.014	0.195 ± 0.004
FedAvg+WSM (ours)		**0.620 ± 0.022**	**0.274 ± 0.001**
FedAvg	0.005	0.585 ± 0.019	0.198 ± 0.004
FedAvg+WSM (ours)		**0.615 ± 0.003**	**0.270 ± 0.001**
FedAvg	0.003	0.545 ± 0.018	0.181 ± 0.009
FedAvg+WSM (ours)		**0.566 ± 0.001**	**0.247 ± 0.004**
FedAvg	0.001	0.441 ± 0.019	0.115 ± 0.007
FedAvg+WSM (ours)		**0.478 ± 0.009**	**0.186 ± 0.003**

A. LeNet Performance Across Multiple Learning Rates

We investigate the performance of WSM using the LeNet architecture (LeCun et al., 1998) on CIFAR-10 and CIFAR-100. While this model is not the considered state-of-the-art on these CIFAR datasets, it allows us to eliminate the dependence on the normalization as a hyper-parameter to investigate relative performance for the purposes of this investigation. As before we train for 4000 rounds with each client training for three local epochs. The data is divided between clients using a Dirichlet distribution parameterized by $\alpha = 0.1$. The reported values are the result of the average of three different seeds with the standard deviation indicating the variation between runs. Experiments were performed across a range of learning rates and the results of the best performing models with and without WSM are shown in Table 3. The difference between the best performing models of FedAvg and and FedAvg+WSM is 1.02% for CIFAR-10 and an impressive 7.6% on CIFAR-100. The complete set of results in Table 4 where WSM outperforms vanilla FedAvg for each learning rate for both datasets. As with the ResNet-18 case, we continue to observe that WSM provides good performance over a larger range of learning rates than FedAvg which makes it easier to tune. We also observe WSM reduces the variance in model accuracy as evidenced by the typically lower standard deviations reported for the runs in each set.

B. Complete Result Set Across Learning Rates and Normalization Methods

Table 5 shows the complete set of results across learning rates and normalization methods. We observe FedAvg+WSM has a strong performance with batch normalization contrary to the findings of Hsieh et al. (2020) while FedAvg preforms better using group norm.

C. WSM for Personalization

We note that in the setting of personalization we can use the WSM directly at inference time, utilizing the client's training data proportion. To illustrate this scenario we consider a WSM trained model with the setting of Sec. 4.2 (high heterogeneity) and demonstrate we can improve the performance by 9.5% at no additional cost under the personalization setting using this strategy.

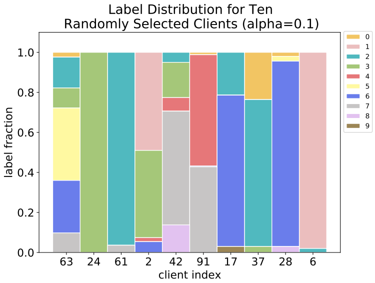


Figure 5. CIFAR-10 - Distributions of ten randomly selected clients with data partitioned according to a Dirichlet distribution parameterized by $\alpha = 0.1$.

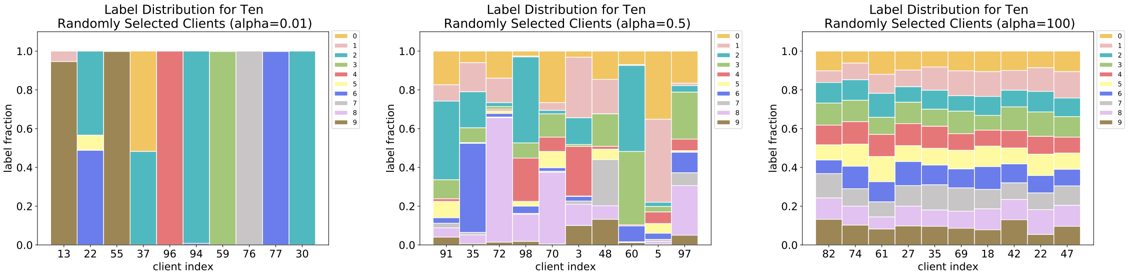


Figure 6. Percentages of each class label for ten randomly selected clients with $\alpha = 0.01, 0.5, 100$ from left to right

D. Validation

Throughout training the global model is periodically evaluated on the aggregation of the client validation sets to gauge overall training progress, the test set on the other hand is only used at the end of the training process. For lower values of α, as client distributions become more skewed, there can be significant changes in accuracy between training runs (Hsu et al., 2019). Since we focus our analysis on the highly heterogeneous case in which the Dirichlet distributions at each client are parameterized by $\alpha = 0.1$, we observe higher variance in our results, particularly for smaller datasets such as CIFAR-10 and CIFAR-100. To mitigate these effects on the validation statistics, we follow the lead of Reddi et al. (2020) and report our final accuracy as the average of the test accuracies taken over the last 100 rounds of training.

D.1. Illustrating different Dirichlet parameters

Figure 5 shows what a label distribution created using a Dirichlet distribution parameterized by $\alpha = 0.1$ would look like for ten randomly selected clients.

Figure 6 offers a practical illustration of how client partitions change as a function of α. Clients with $\alpha = 0.01$ have only a small percentage of the classes while at $\alpha = 100$ clients have all 10 classes in proportions that are much more equal than we observe with the other two parameterizations.

E. Additional Forgetting studies

The heatmaps provided in this section provide additional support for the effects of forgetting during a round of federated learning. The bottom row of Figures 7, 8, 9 and 10 are for rounds 1, 1200, 3600 and 4000, respectively. For each heat map the y-axis indicates the model of client i and the x-axis indicates the local data of client k. The left column shows F_{ik} as defined in equation 6. Positive values of forgetting (green) indicate high forgetting, for FedAvg on the bottom row, we observe a lot of green in the off-diagonal terms of the accuracy heatmaps indicating that forgetting is very high when using standard cross entropy. The post local update heatmap for standard cross entropy shows a strong trend of better accuracies along the diagonal indicating model i does much better on dataset k when $i = k$, its own dataset.

When using WSM, we observe no preference for better accuracy along the diagonal as is the case with FedAvg. Lower

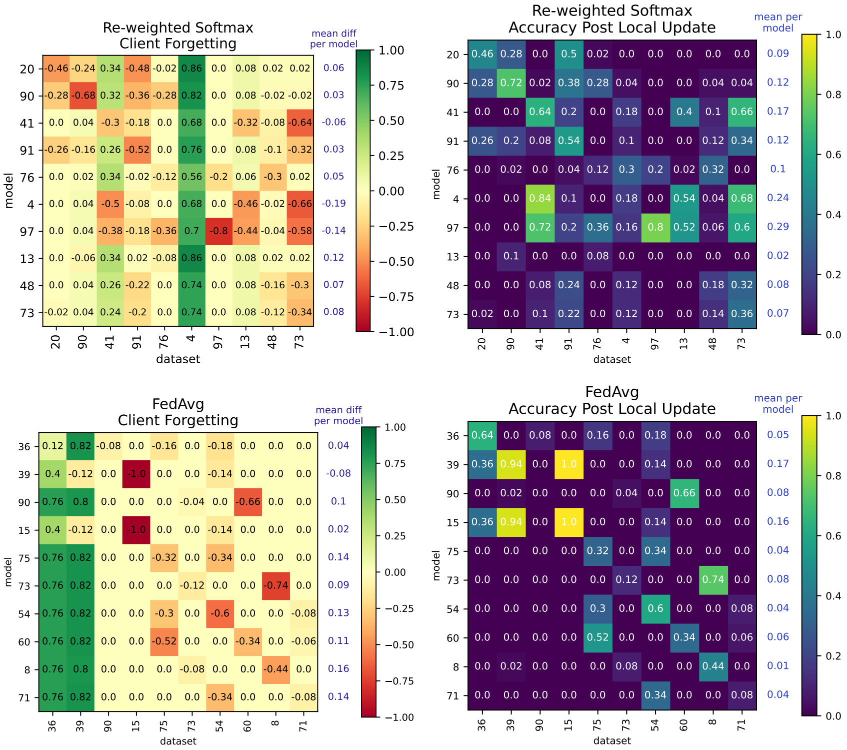


Figure 7. Local client forgetting after round 1 with and without WSM. The heatmap is structured as described above where the y-axis indicates the model of client i and the x-axis indicates the local data of client k. We note that after the first round of training the model has not converged significantly so the advantage conferred by WSM is not yet as apparent as it will be in future rounds, the point we wish to illustrate here is that the effect of local client forgetting is apparent from the beginning of training

accuracies are concentrated along columns indicating a particularly difficult dataset for all local models or along the row, indicating a model that does badly on all datasets including its own. These observations are supported by the forgetting heatmap for WSM which is predominantly yellow indicating very low forgetting values.

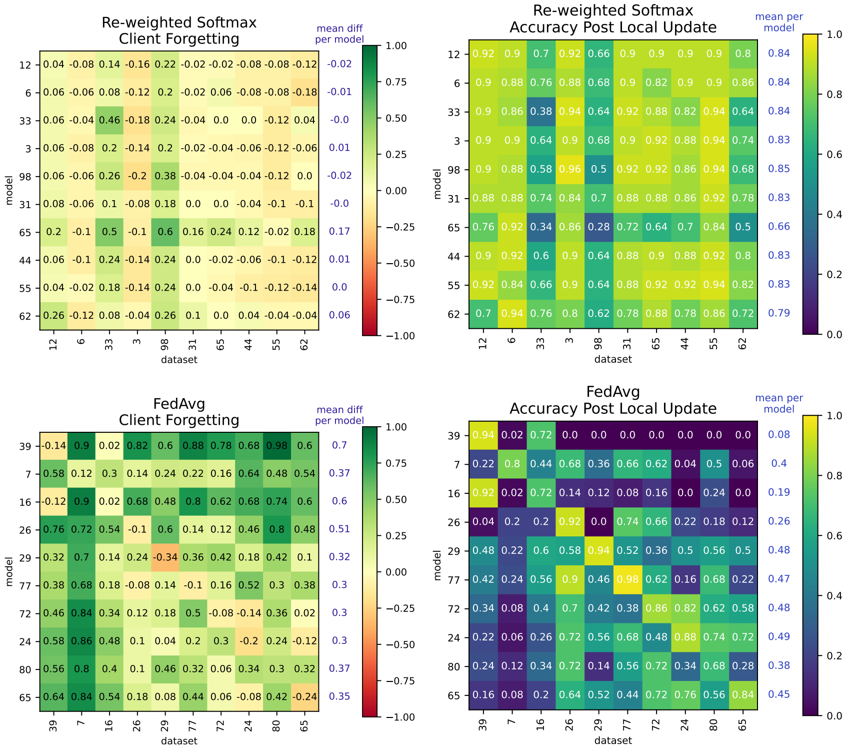


Figure 8. Local client forgetting after round 1200 with and without WSM. The heatmap is structured as described above where the y-axis indicates the model of client i and the x-axis indicates the local data of client k.

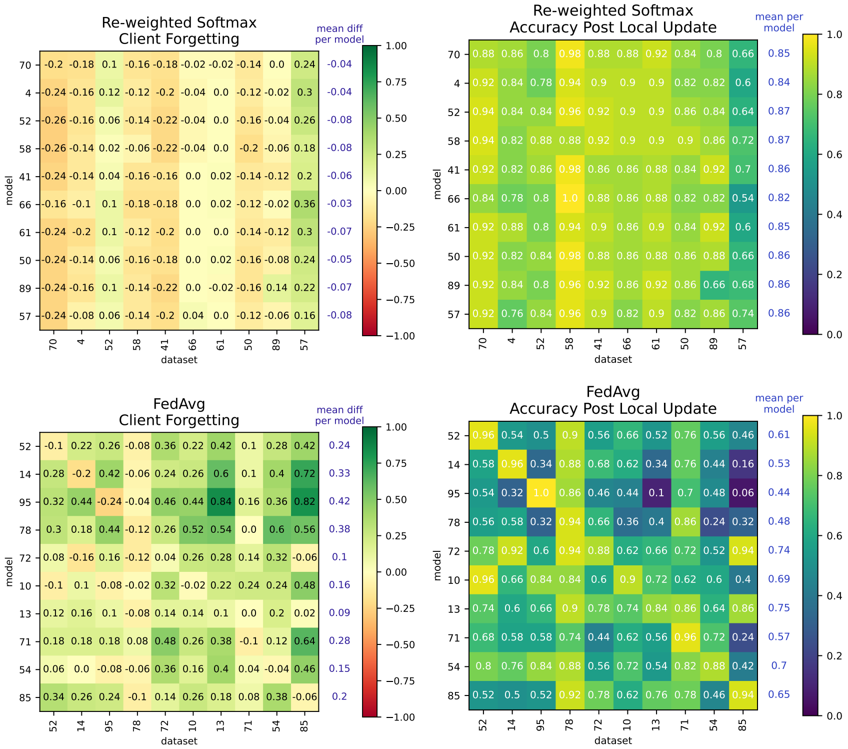


Figure 9. We show local client forgetting for round 3600 with and without WSM. The heatmap is structured as described above where the y-axis indicates the model of client i and the x-axis indicates the local data of client k. Again at round 3600, much later in training than round 1200 shown in figure 8 we observe FedAvg+WSM (top row) significantly reduces forgetting across clients.

Table 5. Accuracy results of FedAvg with and without WSM for different hyper-parameters. We observe that FedAvg+WSM with batch normalization consistently improves performance over FedAvg, as well as having the highest overall accuracy by a large margin. WSM also makes the learning rate easier to tune since we observe a large hyper-parameter range

Method	*Hyper-params*		*Dataset*		
	lr	norm	CIFAR-10	CIFAR-100	FEMNIST
FedAvg	0.5	group	0.326 ± 0.098	0.292 ± 0.012	0.542 ± 0.087
FedAvg+WSM (ours)			0.452 ± 0.173	0.234 ± 0.191	0.418 ± 0.177
FedAvg		batch	0.742 ± 0.004	0.386 ± 0.003	0.812 ± 0.020
FedAvg+WSM (ours)			**0.792 ± 0.006**	**0.426 ± 0.003**	**0.837 ± 0.002**
FedAvg	0.3	group	0.791 ± 0.013	0.384 ± 0.013	0.769 ± 0.006
FedAvg+WSM (ours)			0.744 ± 0.007	0.388 ± 0.027	0.761 ± 0.118
FedAvg		batch	0.742 ± 0.004	0.412 ± 0.014	0.815 ± 0.008
FedAvg+WSM (ours)			**0.834 ± 0.008**	**0.467 ± 0.015**	**0.844 ± 0.010**
FedAvg	0.1	group	0.724 ± 0.027	0.500 ± 0.016	0.835 ± 0.002
FedAvg+WSM (ours)			0.794 ± 0.022	0.446 ± 0.004	0.827 ± 0.012
FedAvg		batch	0.820 ± 0.006	0.442 ± 0.016	0.806 ± 0.031
FedAvg+WSM (ours)			**0.855 ± 0.004**	**0.514 ± 0.009**	**0.848 ± 0.006**
FedAvg	0.07	group	0.826 ± 0.007	0.437 ± 0.007	**0.827 ± 0.006**
FedAvg+WSM (ours)			0.805 ± 0.007	0.484 ± 0.041	0.823 ± 0.006
FedAvg		batch	0.787 ± 0.006	0.513 ± 0.006	0.789 ± 0.003
FedAvg+WSM (ours)			**0.856 ± 0.005**	**0.553 ± 0.018**	0.826 ± 0.019
FedAvg	0.05	group	0.827 ± 0.004	0.464 ± 0.001	**0.853 ± 0.004**
FedAvg+WSM (ours)			0.791 ± 0.019	0.454 ± 0.015	0.841 ± 0.002
FedAvg		batch	0.790 ± 0.012	0.531 ± 0.007	0.833 ± 0.024
FedAvg+WSM (ours)			**0.858 ± 0.003**	**0.564 ± 0.007**	0.842 ± 0.005
FedAvg	0.03	group	0.836 ± 0.005	0.431 ± 0.020	**0.835 ± 0.006**
FedAvg+WSM (ours)			0.774 ± 0.035	0.472 ± 0.007	0.830 ± 0.006
FedAvg		batch	0.779 ± 0.028	0.561 ± 0.010	0.756 ± 0.015
FedAvg+WSM (ours)			**0.857 ± 0.005**	**0.581 ± 0.005**	0.834 ± 0.003
FedAvg	0.01	group	0.815 ± 0.003	0.431 ± 0.005	**0.830 ± 0.002**
FedAvg+WSM (ours)			0.785 ± 0.011	0.471 ± 0.010	0.800 ± 0.018
FedAvg		batch	0.787 ± 0.003	0.566 ± 0.009	0.744 ± 0.011
FedAvg+WSM (ours)			**0.845 ± 0.006**	**0.574 ± 0.006**	0.800 ± 0.019
FedAvg	0.007	group	0.817 ± 0.007	0.426 ± 0.005	**0.821 ± 0.011**
FedAvg+WSM (ours)			0.773 ± 0.014	0.476 ± 0.010	0.794 ± 0.006
FedAvg		batch	0.797 ± 0.008	**0.568 ± 0.003**	0.734 ± 0.018
FedAvg+WSM (ours)			**0.841 ± 0.004**	**0.568 ± 0.003**	0.800 ± 0.007
FedAvg	0.005	group	0.802 ± 0.010	0.426 ± 0.002	**0.819 ± 0.022**
FedAvg+WSM (ours)			0.768 ± 0.019	0.474 ± 0.006	0.781 ± 0.017
FedAvg		batch	0.783 ± 0.009	0.553 ± 0.005	0.743 ± 0.053
FedAvg+WSM (ours)			**0.826 ± 0.009**	**0.554 ± 0.004**	0.773 ± 0.013

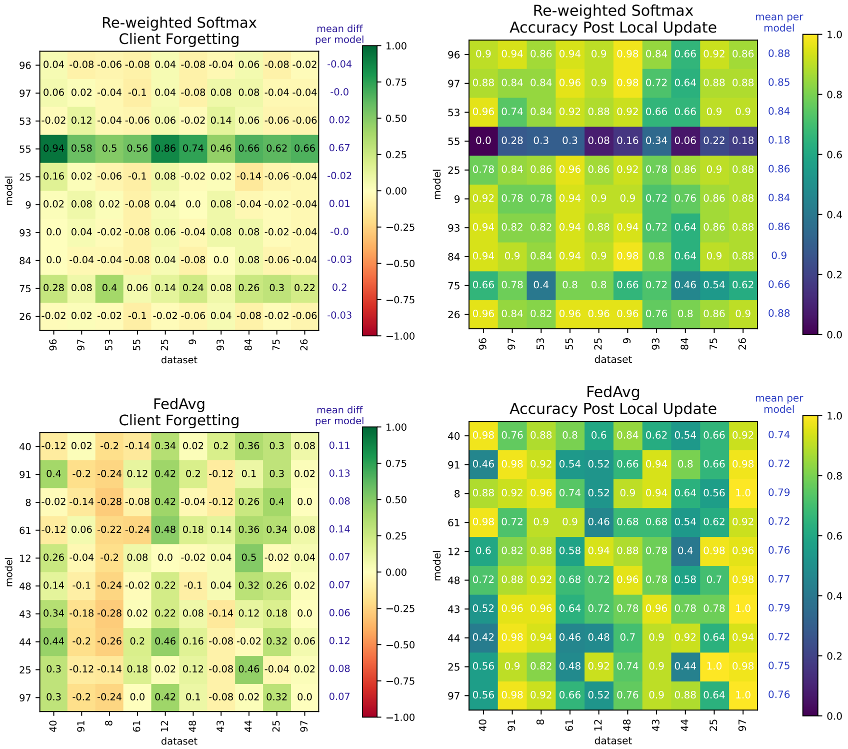


Figure 10. Local client forgetting for round 4000, the last round of training with and without WSM. The heatmap is structured as described above where the y-axis indicates the model of client i and the x-axis indicates the local data of client k. Here we observe that in the last round of training local client forgetting still occurs for FedAvg. FedAvg+WSM on the other hand has relatively neutral levels of forgetting (close to 0) with the exception of client 55's model which appears to have had a bad round of training in which it forgot quite a bit of relevant information. In this case we point out that forgetting here has occurred equally across all client datasets including it's own indicating the training failure here is not due to local client forgetting.