# Federated Learning Aggregation based on Weight Distribution Analysis

Christos Chatzikonstantinou, Dimitrios Konstantinidis, Kosmas Dimitropoulos
and Petros Daras, Senior Member IEEE
Visual Computing Lab, Information Technologies Institute, Centre for Research and Technology Hellas
{chatziko,dikonsta,dimitrop,daras}@iti.gr

***Abstract*—Federated learning has recently been proposed as a solution to the problem of using private or sensitive data for training a central deep model, without exchanging the local data. In federated learning, local models are trained on the client side using the available data, while a server is responsible for aggregating the weights of these models into a global model. This work proposes a novel federated learning weight aggregation method that estimates the statistical distance of each client's parameters from the Gaussianity, and weighs the contribution of each client to the global model accordingly so that the most significant information is retained and enhanced. To create an accurate global model, a complex weighted averaging of the parameters of customers' models at the layer level is performed, considering as low quality the parameters following the Gaussian distribution. The proposed method can be employed to both convolutional and linear layers and it is based on the notion that parameters following a Gaussian distribution do not significantly affect the output of a model. Experiments with different network architectures (such as VGG and ResNet) and comparison with state-of-the-art approaches on three well-known image classification datasets, demonstrate the superiority of the proposed method against state-of-the-art federated learning methods.**

***Index Terms*—federated learning, Gaussian distribution, image classification**

## I. Introduction

Federated learning (FL) [20] is a machine learning approach that utilizes a number of distributed edge devices or servers with their own local data samples to train an algorithm without transferring these data samples. It can be applied in several application areas, such as healthcare, industrial engineering and defence [15]. The aim of FL is to overcome challenges related to the handling of private or sensitive data, requiring the data to be safely stored in their local storage space and not being transferred to other locations. During the training process, the clients and the server periodically communicate with each other to merge the different models, usually by averaging the parameters of all local models to update the global model on the server [20].

FL differs from more typical decentralized approaches, which frequently presume that local data samples are uniformly distributed, as well as standard centralized machine learning techniques, in which all local datasets are uploaded to a single server. FL usually employs the parameter server architecture [8], in which a global model is created on the server, while the isolated clients use their own private data to train local models on their devices, thus achieving enhanced privacy protection and effective distributed training. During the training process, the clients and the server periodically communicate with each other to merge the different models, usually by averaging the parameters of all local models to update the global model on the server [20].

Multiple FL methods have been released so far. Some methods aim to improve the performance of FL by either optimizing the selection criterion, i.e., choosing the appropriate client parameters that will maximize the performance of the global model on the server [11] [20] or improving the local training processes of the clients [12] [17]. Other methods focus on the improvement of the efficiency of FL in different ways, such as optimizing the communication efficiency [2] or the local training efficiency [25] or both [3]. Other works may focus on data privacy [6], in schemes of non-supervised learning, such as semi-supervised [18] or unsupervised [10], or in incremental learning [9].

Although there are several methods in the literature to improve the communication and training efficiency and effectiveness in federated learning, there is limited research on alternatives to the simple federated averaging technique, which is considered the de facto parameter aggregation approach. Such alternatives require the analysis of the weight distribution of the local client parameters and the assessment of their effect on the performance of the global model, a study that has not yet been considered in the literature. In our view, the way the network parameters of clients are fused during the aggregation process, plays a crucial role in the performance of the global model and this is what the proposed method is investigating.

To this end, a novel FL weight aggregation approach is proposed, in this work, aiming to improve the selection criterion for choosing the most important local network parameters for the update of the global model. The importance of local network parameters is assessed based on the statistical distance from the Gaussian distribution. To this end, the statistical distance or divergence from the Gaussian distribution is employed to assess the quality of each clients' parameters and choose the ones with the largest impact on the accuracy of the global model. The motivation behind this choice lies in the observation that the employment of the $L2$ training norm as a regularization term to resolve the issue of exploding gradients during deep network training leads filter parameters to follow a Gaussian distribution, resulting in hidden units with little impact on the network output [22]. Thus, filter parameters that follow a Gaussian distribution are considered to be of low quality to the performance of the global model. The main contributions of this work are:

- A novel FL weight aggregation algorithm for optimally fusing network parameters on the layer-level for either convolutional or linear layers based on the importance of these parameters to the accuracy of the global model.
- The weighted averaging is based on a novel selection criterion that estimates the statistical distance of network parameters from the Gaussian distribution.
- Experimental results on three image classification datasets are presented, showing the superiority of the proposed method against various state-of-the-art approaches.

## II. Method

### A. Motivation

During each communication round of the federated learning training, the local parameters of the models on the client side are merged to form the parameters of the global model on the server side. A naive way to perform this merging is through averaging, assuming that all local parameters are equally important to the output of the global model. However, when a client has a disproportionately large number of data samples with respect to the other clients or a client has data of really bad quality, the averaging approach can lead to a poor performing global model. In addition, potential client specializations are lost when clients' parameters are averaged. To overcome such issues, a sophisticated selection criterion is required to assess the quality of the parameters of a client model and diminish the impact of the low quality ones on the performance of the global model.

In this work, Gaussianity is proposed as a metric of the quality of the weight parameters of DNNs. The motivation behind measuring Gaussianity is based on the observation that the usage of the L2 norm, as a regularization term to solve the issue of exploding gradients during deep network training [24], pushes the DNN parameters to follow the Gaussian distribution [4], [5], [26]. However, Gaussianity is not the best property for DNNs since neurons with Gaussian weights tend to blur the input information. According to [22], the contributions of each individual hidden layer are all very small when using Gaussian priors, hence these units do not reflect "hidden features" that capture significant characteristics of the data. To measure Gaussianity, this work proposes the use of different statistical distances, such as divergences, as well as Higher Order Statistics (HOS) [21]. These statistical distances are capable of measuring the distance between a random process (i.e., network parameters) and a Gaussian distribution.

Therefore, this work proposes a novel federated learning weight aggregation method, (Figure 1), named Statistical Weight Aggregation (SWA), that analyzes the Gaussianity of clients' layer parameters to enhance the contribution of those parameters that deviate from the Gaussian distribution and thus capture significant and discriminative elements of the data.

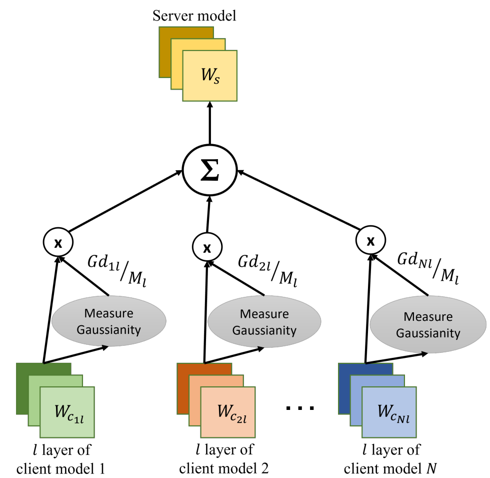


Fig. 1. Illustration of the proposed layer-level weight aggregation approach.

### B. Layer-level weight aggregation

This work aims to quantify the impact of the network parameters of clients to the global model and thus their weights during the FL aggregation phase by assessing their deviation from the Guassian distribution. This is achieved through the use of statistical distances that can be either divergences or Higher Order Statistics, as described in detail below. The higher the statistical distance of a client's layer parameters from the Guassian distribution, the larger the weight that is assigned to the parameters when they are merged into a global model.

*1) Divergence:* A divergence is a type of statistical distance implemented by a binary function that specifies the separation from one probability distribution to another on a statistical manifold. In this work, the family of Rényi divergences [28] of order $a$ or alpha-divergences are utilized. Renyi divergence is a critical tool for proving the convergence of Bayesian estimators and it is implicitly used in many calculations across information theory. Further uses of Renyi divergence include hypothesis testing, multiple source adaptation, and picture rating. A special case of Rényi divergences is the well-known Kullback–Leibler (KL) divergence [7] that has been widely employed for comparing distributions in several fields.

The Renyi divergence of order $a$ of a distribution $P$ (i.e., distribution of a client's layer-level weight parameters) from a distribution $Q$ (i.e., Gaussian distribution) is defined to be:

$$D_a(P||Q) = \frac{1}{a-1} log_e(\sum_{i=1}^{N} \frac{p_i^a}{q_i^{a-1}}) \quad (1)$$

where N is the total number of samples of the distribution.

In Eq. 1, $a$ is a positive number ($0 < a < \infty$), defining the order of the divergence. The Rényi entropy increasingly ranks all nonzero probability events equally as $a$ approaches zero, regardless of their probabilities. On the contrary, the events with the highest probability have a higher impact on the Rényi entropy as $a$ gets closer to infinity.

Variables $p$ and $q$ are the probabilities of the distributions $P$ and $Q$, respectively. In this work, four different values of $a$ are considered: $0, 0.5, 1, 2$ and according to the value of $a$, the Renyi divergence takes the following forms:

$$D_a(P||Q) = \begin{cases} -log_e(Q(i : p_i > 0) & a = 0 \\ -2log_e(\sum_{i=1}^{N} \sqrt{p_i q_i}) & a = 0.5 \\ \sum_{i=1}^{N} p_i log_e(\frac{p_i}{q_i}) & a = 1 \\ log_e(\sum_{i=1}^{N} \frac{p_i^2}{q_i}) & a = 2 \end{cases} \quad (2)$$

In the special case of $a = 0.5$, the Renyi divergence becomes twice the Bhattacharyya distance [23], while in the special case of $a = 1$, the Renyi divergence is equal to the KL divergence [7].

*2) Higher Order Statistics :* Many statistical tools exist for information extraction from a random signal. Nevertheless, several signals cannot be properly examined by second order statistical approaches when non-linearity in systems is present. Thus, higher order statistical methods have been developed. The higher order statistics have been used to describe the higher-order statistical characteristics of a random process. HOS use the third or higher power of a sample (e.g., skewness, kyrtosis), as opposed to more conventional techniques of lower-order statistics, which use constant, linear, and quadratic terms (e.g., mean, variance). HOS are defined using moments and cumulants [27]. Cumulants of a set of values with sample size $N$ can be calculated using $k$-order statistics and provide an indication of how far a random process is from being Gaussian. In this context, the 3rd and 4th order statistics comprise the unbiased estimators of the cumulant $C_{q,z}$ and they can be computed, using the central moments $m_i = \frac{1}{N}\sum_{j=1}^{N} z_j^i(t)$, as follows [1]:

$$k_3 = \frac{N^2}{(N-1)\cdot(N-2)} m_3, \quad (3)$$

$$k_4 = \frac{N^2[(N+1)m_4 - 3(N-1)m_2^2]}{(N-1)\cdot(N-2)\cdot(N-3)}, \quad (4)$$

Then, the statistical distance from the Gaussian distribution can be computed by the product of the two $k$-order statistics as follows:

$$D_{HOS} = k_3 * k_4 \quad (5)$$

*3) Weight Aggregation Algorithm:* The proposed weight aggregation algorithm is based on the assignment of a weight to each clients' layer parameters during aggregation. The assigned weight quantifies the Gaussianity of the network parameters at the layer level using either the Renyi divergence or the HOS values that are referred from now on as $Gd$. This process is illustrated in Figure 1 and presented as pseudocode by Algorithm 1.

More specifically, the network parameters of each layer of each client are initially flattened to form a vector and then the $Gd$ of these parameters is computed. Given a layer $l, l \in [0, L)$ from a client $n, n \in [0, N)$, $Gd$ is equal to:

$$Gd_{nl} = \begin{cases} D_{HOS} & \text{for Higher Order Statistics} \\ D_a(P||Q) & \text{for Renyi divergence} \end{cases} \quad (6)$$

In the case of Renyi divergence, a vector $Q$ of the same size as the layer parameters is automatically generated by drawing samples from the Gaussian distribution $N(0, 1)$. $Gd$ has higher values as the statistical distance of the network parameters increases and thus, it can be directly employed as weight for the network parameters during the aggregation process.

After the computation of the $Gd$ values for each layer, the maximum value among the different clients is defined as:

$$M_l = \max(Gd_{1l}, ..., Gd_{nl}). \quad (7)$$

The maximum value $M_l$ is used to normalize the weights among the different clients of the layer $l$. Finally, the server parameters are computed through the weighted averaging of the clients' parameters as shown below:

$$W_{S_l} = \sum_{n=0}^{N} \frac{Gd_{nl}}{M_l} W_{c_{nl}}, \quad (8)$$

**Algorithm 1** The proposed layer aggregation algorithm

**Input:** Number of model layers $L$, number of clients $N$
**Output:** Server network parameters $W_S[l]$ of layer $l$

```
for l = 0, 1, ..., L − 1 do
    for n = 0, 1, ..., N − 1 do
        Calculate weights Gd_nl using Eq. 6
    end for
    Calculate server parameters W_S[l] using Eq. 8
end for
```

## III. Experimental Evaluation

### A. Datasets

The following well-known public datasets are utilized for the experimental evaluation in the task of image classification:

The CIFAR-10/100 datasets [13] consist of natural images with resolution 32x32 that belong to 10 semantic classes in the case of the CIFAR-10 dataset and 100 semantic classes in the case of the CIFAR-100 dataset. The training and test sets contain 50K and 10K images, respectively.

The Tiny ImageNet [14] is a subset of the full ImageNet ILSRVC dataset. It comprises 120000 colored images of 200 classes, downsized to size 64x64. Each class has 500 training, 50 validation and 50 test images.

### B. Implementation details

For ablation study, the VGG-16 network is utilized on CIFAR-10. The CIFAR-10 training set is split in two subsets: the clients' data (99% of the training set) and the server's validation data (1% of the training set). The clients' data are split equally among the clients and 90% of them is used for the clients' training, while the rest 10% is used for the clients' evaluation. Finally, the server model is tested on the CIFAR-10 test set and the results are presented. All experiments are executed 3 times with random data splits and average and standard deviation are reported. A learning rate of 0.1 (unless otherwise indicated), a batch size of 64 and a momentum value of 0.9 for the SGD optimizer are utilized.

Reagarding the comparison with the SoTA, the proposed method is evaluated on three datasets, namely CIFAR10, CIFAR100 and Tiny ImageNet. The Dirichlet distribution is employed to create the non-IID data partitions with $\beta = 0.5$, 10 local epochs and 10 clients. Two networks per dataset are evaluated. Regarding the CIFAR10 dataset, a custom CNN network is utilized with 2 convolutional, 2 max pooling and 2 fully connected layers (as defined in [16]) and VGG-11, trained for 100 and 55 rounds respectively. On the other hand, in CIFAR-100 and in Tiny ImageNet, the ResNet-50 and VGG-11 network architectures are employed. These networks are trained for 100 total epochs in CIFAR-100 and 20 and 55 total epochs in Tiny ImageNet for ResNet and VGG-11, respectively. SGD optimizer and momentum equal to 0.9 utilized in all cases, batch is 256 and lr is $5*10^{-4}$ for VGG-11 and 64 and $10^{-2}$ respectively for the other cases.

For the training and testing of the implemented deep learning models, the Python 3.7 and PyTorch (version 1.7.0) environments are employed and CUDA version 10.2. The code will be made publicly available.

### C. Ablation study

The ablation study showcases the effect of specific hyper-parameters (i.e., type of statistical distance, data split, number of epochs, number of clients) on the performance of SWA, compared in most cases against the baseline simple averaging approach to demonstrate the effectiveness of the proposed method. The ablation study is performed on the test set of the CIFAR-10 dataset, using the VGG16 model. The goal of this ablation study is to tune the aforementioned hyperparameters for an optimal performance of the proposed method.

*1) Impact of different statistical distances:* This experiment evaluates the effect of the different types of the statistical distances, presented in Section II-B3, on the server model's accuracy. Three clients are utilized with 6 communication rounds, 50 local epochs and equal data splits. The server has no information of the data each client has, thus all clients are treated equally. As it can be seen in Table I, the statistical distance based on the HOS values achieves a higher accuracy than in the cases where the different cases of Renyi divergence is employed, showing that it is a more robust criterion for capturing layer parameters that deviate from the Gaussian distribution. To this end, the HOS criterion is selected as the optimal statistical distance metric for the rest of the experiments.

| Statistical distance | Accuracy |
|---|---|
| $D_{HOS}$ | $\mathbf{88.41 \pm 0.16}$ |
| $D_0(P\|\|Q)$ | $88.12 \pm 0.29$ |
| $D_{0.5}(P\|\|Q)$ | $87.58 \pm 0.25$ |
| $D_1(P\|\|Q)$ | $88.11 \pm 0.28$ |
| $D_2(P\|\|Q)$ | $87.76 \pm 0.29$ |

TABLE I
IMPACT OF DIFFERENT STATISTICAL DISTANCES.

*2) Impact of different data splits:* The goal of this experiment is to evaluate the effect of different data splits on the model's accuracy by employing either the proposed SWA method or the baseline simple averaging technique. Three clients are used with 6 communication rounds and 50 local epochs. The results are illustrated in Table II and the splits are described in a format that shows the percentage of the data that are fed to each client, while leaving out 10% of the data for validation. The results demonstrate the superiority of SWA against the simple averaging technique for both balanced and imbalanced data splits, indicating that the proposed method is more robust to various data splits that may occur under realistic settings.

| Splits | Simple Averaging | SWA |
|---|---|---|
| 0.3, 0.3, 0.3 | $88.30 \pm 0.18$ | $\mathbf{88.41 \pm 0.16}$ |
| 0.4, 0.25, 0.25 | $88.06 \pm 0.12$ | $\mathbf{88.48 \pm 0.24}$ |
| 0.5, 0.2, 0.2 | $87.93 \pm 0.20$ | $\mathbf{89.04 \pm 0.24}$ |
| 0.6, 0.15, 0.15 | $88.11 \pm 0.36$ | $\mathbf{88.94 \pm 0.37}$ |
| 0.7, 0.1, 0.1 | $87.83 \pm 0.31$ | $\mathbf{87.86 \pm 0.38}$ |

TABLE II
IMPACT OF DIFFERENT DATA SPLITS.

*3) Impact of different number of local epochs:* This experiment aims to evaluate the impact of different local epochs (i.e., training epochs of each client between two communication rounds) on the accuracy of the server's model. The experiments are performed with 3 clients and equal data splits. Each client is trained for 300 epochs in total and the communication rounds vary depending on the number of local epochs (from 2 to 30), as shown in Fig. 2. From the results, it can be observed that SWA outperforms simple averaging in all cases. As the communication rounds are reduced, the accuracy of both methods drops as the server model is updated less frequently. In the case of 150 local epochs and 2 communication rounds, especially, the accuracy of simple averaging collapses contrary to the proposed SWA that maintains a high accuracy, indicating a faster convergence. Faster convergence may be important in a case that the communication between the server and the clients is difficult or very slow.

*4) Impact of different number of clients:* This experiment evaluates the impact of the different number of clients on the performance of the server model. The number of clients vary from 3 to 100, the data are split equally among the clients, while the number of local epochs is set to 10 (30 communication rounds) because as the number of clients increases, it is more difficult for the model to converge and more communication rounds are required. Moreover, for the experiments with more than 10 clients, a smaller learning rate

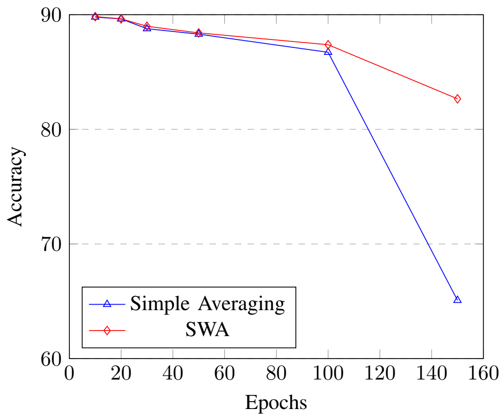


Fig. 2. Impact of different local epochs

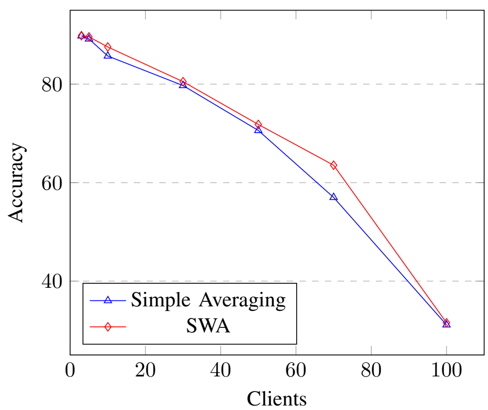


Fig. 3. Impact of different clients

of 0.01 is utilized to achieve convergence. The results, shown in Fig. 3, reveal that SWA outperforms simple averaging in all tests with different number of clients. It can thus be concluded that SWA can better adapt to a high number of clients by maintaining a higher accuracy than the simple averaging approach.

### *D. Comparison with SoTA*

For the comparison with state-of-the-art methods, two configurations are considered, namely SWA with and without contrastive loss, depending on whether contrastive loss has been employed during training. The contrastive loss has been successfully employed in [16] and in this work it is used along with the cross-entropy loss for improved accuracy. According to the ablation results, the HOS criterion is employed to the following experiments. The results presented in Tables III, IV and V show the comparative evaluation against other state-of-the-art federated learning methods on the task of image classification using CIFAR-10, CIFAR-100 and Tiny ImageNet, respectively. Regarding contrastive loss, weight factors equal to 8, 5 and 3 for CIFAR-10, CIFAR-100 and Tiny Imagenet, respectively, are employed after experimentation.

From the results, it can be inferred that the proposed SWA with contrastive loss outperforms all other state-of-the-art methods (i.e., FedAvg, FedProx, MOON, FedMA and GAMF) in all datasets and networks. A comparison between SWA with and without contrastive loss shows that contrastive loss leads to an accuracy improvement of $0.5 - 2\%$ in all datasets. These results indicate that the proposed SWA method can be combined with features (i.e, contrastive loss) from methods that aim to improve the local training of the clients and achieve enhanced accuracy. Finally, it can be concluded that the deviation of the network parameters from the Gaussian distribution can be successfully employed as a robust selection criterion for the network parameters of the clients' models.

| Method | Accuracy |
|---|---|
| Local training | $46.30 \pm 5.10$ |
| FedAvg [20] | $66.30 \pm 0.50$ |
| FedProx [17] | $66.90 \pm 0.20$ |
| MOON [16] | $69.10 \pm 0.40$ |
| SWA w/o contrastive loss | $69.97 \pm 1.49$ |
| **SWA w/ contrastive loss** | $\mathbf{71.02 \pm 0.56}$ |

TABLE III
COMPARISON WITH SOTA METHODS ON CIFAR10 WITH CUSTOM CNN MODEL, 10 CLIENTS, 10 LOCAL EPOCHS AND 1000 TOTAL EPOCHS .

| Method | Accuracy |
|---|---|
| Local training | $22.30 \pm 1.00$ |
| FedAvg [20] | $64.50 \pm 0.40$ |
| FedProx [17] | $64.60 \pm 0.20$ |
| MOON [16] | $67.50 \pm 0.40$ |
| SWA w/o contrastive loss | $66.20 \pm 0.99$ |
| **SWA w/ contrastive loss** | $\mathbf{68.53 \pm 0.65}$ |

TABLE IV
COMPARISON WITH SOTA METHODS ON CIFAR100 WITH CUSTOM RESNET50 MODEL, 10 CLIENTS, 10 LOCAL EPOCHS AND 1000 TOTAL EPOCHS .

| Method | Accuracy |
|---|---|
| Local training | $8.60 \pm 0.40$ |
| FedAvg [20] | $23.00 \pm 0.10$ |
| FedProx [17] | $23.20 \pm 0.20$ |
| MOON [16] | $25.10 \pm 0.10$ |
| SWA w/o contrastive loss | $23.53 \pm 0.15$ |
| **SWA w/ contrastive loss** | $\mathbf{25.22 \pm 0.95}$ |

TABLE V
COMPARISON WITH SOTA METHODS ON TINY IMAGENET WITH RESNET50 MODEL, 10 CLIENTS, 10 LOCAL EPOCHS AND 200 TOTAL EPOCHS.

| Method | Accuracy |
|---|---|
| FedAvg [20] | $69.99 \pm 0.40$ |
| FedMA [29] | $70.29 \pm 0.69$ |
| MOON [16] | $72.42 \pm 0.45$ |
| GAMF [19] | $72.39 \pm 0.54$ |
| GAMF w/ contrastive loss [19] | $73.43 \pm 0.54$ |
| MOON [16] | $72.42 \pm 0.45$ |
| SWA w/o contrastive loss | $79.76 \pm 0.63$ |
| **SWA w/ contrastive loss** | $\mathbf{80.51 \pm 0.35}$ |

TABLE VI
COMPARISON WITH SOTA METHODS ON CIFAR10 WITH VGG11 MODELS, 10 CLIENTS, 10 LOCAL EPOCHS AND 550 TOTAL EPOCHS.

| Method | Accuracy |
|---|---|
| FedAvg [20] | $44.42 \pm 0.13$ |
| FedMA [29] | $44.95 \pm 0.19$ |
| MOON [16] | $46.99 \pm 0.28$ |
| GAMF [19] | $45.99 \pm 0.41$ |
| GAMF w/ contrastive loss [19] | $48.24 \pm 0.39$ |
| SWA w/o contrastive loss | $50.52 \pm 0.57$ |
| **SWA w/ contrastive loss** | $\mathbf{51.07 \pm 0.55}$ |

TABLE VII
COMPARISON WITH SOTA METHODS ON CIFAR100 WITH VGG11 MODEL, 10 CLIENTS, 10 LOCAL EPOCHS AND 1000 TOTAL EPOCHS .

| Method | Accuracy |
|---|---|
| FedAvg [20] | $17.41 \pm 0.13$ |
| FedMA [29] | $17.28 \pm 0.20$ |
| MOON [16] | $19.01 \pm 0.15$ |
| GAMF [19] | $20.42 \pm 0.13$ |
| GAMF w/ contrastive loss [19] | $21.51 \pm 0.15$ |
| SWA w/o contrastive loss | $29.13 \pm 0.52$ |
| **SWA w/ contrastive loss** | $\mathbf{30.78 \pm 0.40}$ |

TABLE VIII
COMPARISON WITH SOTA METHODS ON TINY IMAGENET WITH VGG11 MODEL, 10 CLIENTS, 10 LOCAL EPOCHS AND 550 TOTAL EPOCHS.

## IV. CONCLUSION

This work proposes a novel FL weight aggregation method, called SWA, that can achieve a sophisticated weighted averaging of the parameters of clients' models at the layer level to form an accurate global model. The proposed method can be employed to both convolutional and linear layers of a model and is based on the use of statistical criteria for the evaluation of the Gaussianity of the model parameters and the estimation of appropriate weights for the parameter aggregation phase. Directions for future work include the adaptation of the method on the filter level, as well as the use of other statistical tools to assess the impact of the client parameters to the global model. Moreover, different statistical distances can also be evaluated on the proposed algorithm.

## REFERENCES


[1] Francis Andre. *Business mathematics and statistics*. Thomson, 2004.

[2] Daniel Becking, Heiner Kirchhoffer, Gerhard Tech, Paul Haase, Karsten Müller, Heiko Schwarz, and Wojciech Samek. Adaptive differential filters for fast and communication-efficient federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3367–3376, 2022.

[3] Sameer Bibikar, Haris Vikalo, Zhangyang Wang, and Xiaohan Chen. Federated dynamic sparse training: Computing less, communicating less, yet learning better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6080–6088, 2022.

[4] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

[5] Christos Chatzikonstantinou, Georgios T. Papadopoulos, Kosmas Dimitropoulos, and Petros Daras. Neural network compression using higher-order statistics and auxiliary reconstruction losses. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3077–3086, 2020.

[6] Anda Cheng, Peisong Wang, Xi Sheryl Zhang, and Jian Cheng. Differentially private federated learning with local regularization and sparsification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10122–10131, 2022.

[7] Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pages 146–158, 1975.

[8] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, et al. Large scale distributed deep networks. *Advances in neural information processing systems*, 25, 2012.

[9] Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10164–10173, 2022.

[10] Sungwon Han, Sungwon Park, Fangzhao Wu, Sundong Kim, Chuhan Wu, Xing Xie, and Meeyoung Cha. Fedx: Unsupervised federated learning with cross knowledge distillation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pages 691–707. Springer, 2022.

[11] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[12] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR, 13–18 Jul 2020.

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

[14] Fei-Fei Li, Andrej Karpathy, and Justin Johnson. Tiny imagenet. https://www.kaggle.com/competitions/tiny-imagenet/overview, 2017.

[15] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020.

[16] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10713–10722, 2021.

[17] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

[18] Xiaoxiao Liang, Yiqun Lin, Huazhu Fu, Lei Zhu, and Xiaomeng Li. Rscfed: random sampling consensus federated semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10154–10163, 2022.

[19] Chang Liu, Chenfei Lou, Runzhong Wang, Alan Yuhan Xi, Li Shen, and Junchi Yan. Deep neural network fusion via graph matching with applications to model ensemble and federated learning. In *International Conference on Machine Learning*, pages 13857–13869. PMLR, 2022.

[20] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[21] Jerry M Mendel. Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications. *Proceedings of the IEEE*, 79(3):278–305, 1991.

[22] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[23] Frank Nielsen and Sylvain Boltz. The burbea-rao and bhattacharyya centroids. *IEEE Transactions on Information Theory*, 57(8):5455–5466, 2011.

[24] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR, 2013.

[25] Xinchi Qiu, Javier Fernandez-Marques, Pedro PB Gusmao, Yan Gao, Titouan Parcollet, and Nicholas Donald Lane. Zerofl: Efficient on-device training for federated learning with local sparsity. In *International Conference on Learning Representations*, 2022.

[26] Jason Rennie. On l2-norm regularization and the gaussian prior. 2003.

[27] M Sanaullah. A review of higher order statistics and spectra in communication systems. *Global Journal of Science Frontier Research, Physics and Space Science*, 13(4), 2013.

[28] Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[29] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020.