Minimizing Layerwise Activation Norm Improves Generalization in Federated Learning

M. Yashwanth[1], Gaurav Kumar Nayak[2], Harsh Rangwani[1], Arya Singh[3],
R. Venkatesh Babu[1], and Anirban Chakraborty[1]

[1]Indian Institute of Science, Bangalore, [2]University of Central Florida, [3]BITS Pilani

[1]{yashwanthm,harshr,venky,anirban}@iisc.ac.in, [2]gauravkumar.nayak@ucf.edu,
[3]f20180762g@alumni.bits-pilani.ac.in

Abstract

*Federated Learning (FL) is an emerging machine learning framework that enables multiple clients (coordinated by a server) to collaboratively train a global model by aggregating the locally trained models without sharing any client's training data. It has been observed in recent works that learning in a federated manner may lead the aggregated global model to converge to a 'sharp minimum' thereby adversely affecting the generalizability of this FL-trained model. Therefore, in this work, we aim to improve the generalization performance of models trained in a federated setup by introducing a 'flatness' constrained FL optimization problem. This flatness constraint is imposed on the top eigenvalue of the Hessian computed from the training loss. As each client trains a model on its local data, we further re-formulate this complex problem utilizing the client loss functions and propose a new computationally efficient regularization technique[1], dubbed 'MAN,' which **M**inimizes **A**ctivation's **N**orm of each layer on client-side models. We also theoretically show that minimizing the activation norm reduces the top eigenvalue of the layer-wise Hessian of the client's loss, which in turn decreases the overall Hessian's top eigenvalue, ensuring convergence to a flat minimum. We apply our proposed flatness-constrained optimization to the existing FL techniques and obtain significant improvements, thereby establishing new state-of-the-art.*

1. Introduction

Federated Learning (FL), first introduced in [17], is a distributed machine learning paradigm that involves multiple clients learning a shared model under the coordination of a central server. In FL, a client cannot send the training data to the server due to privacy concerns and communication overheads. Instead, clients share the parameters of models trained on their local data with the server, which aggregates these models with the objective of achieving better generalization on the overall data distribution across all clients. FL's privacy-preserving nature has made it increasingly popular in various domains, including smartphones [5,23], Internet of Things [19,32] and the healthcare industry [21,26,30].

FedAvg introduced in [17] is a popular algorithm for federated training in which each client trains its local model for multiple epochs before sharing it with the server. However, FedAvg often exhibits convergence issues [9], particularly when dealing with non-iid data. Numerous approaches have been proposed to overcome the limitations of FedAvg, particularly when dealing with non-iid distributions such as SCAFFOLD [9], FedDyn [1], and FedDC [4]. These methods aim to address the problem from an optimization perspective by seeking a better minimum for the global model via minimization of the training objective.

In centralized learning literature [3, 10, 27, 31], a strong connection has been observed between generalization attained by the trained models and the sharpness of the loss surface where the model parameters converge at the end of training. Methods such as SAM [3] are proposed to attain flat minima for better generalization. Recent studies on generalization under FL setup, such as FedSAM/ASAM [2,22], have shown that federated training often converges to a sharp minimum, which can negatively impact the generalization performance of the global model. This happens as each client participating in FL has a limited amount of data, and different clients can have different data distribution, eventually leading to overfitting of the models and sharp minimum. To solve this problem, methods such as FedSAM use SAM [3] optimizer on the client models to avoid the sharp minimum and get better generalized global models. However, there are still issues with these methods-they require additional forward and backward passes as their update rules are based on gradient ascent followed by gradient descent. Methods such as FedDC [4] seem to perform better than SAM-based FL optimizers such as FedSAM.

[1] https://github.com/vcl-iisc/fedMAN.git

Also FedSAM cannot be easily integrated atop the existing state-of-the-art optimizers such as FedDC to further enhance their performance without impacting the convergence guarantees. These methods also inherently assume that using SAM updates on the local models can provide a global model with flat minima.

To overcome the aforementioned limitations and motivated by the findings of centralized training, we present a novel Federated Learning (FL) optimization approach that integrates a flatness constraint as a regularizer on top of the FL objective. The top eigenvalue and trace of the Hessian of the loss are typical indicators of flatness [3, 10, 27, 31]. The lower values of the top eigenvalue/trace are desired for better performance. Ideally, our objective should be to achieve convergence towards a flat minimum while maintaining a low training loss. Thus, we directly aim to minimize the top eigenvalue of the global model along with the training loss. This facilitates convergence of the global model towards a flat minimum while simultaneously minimizing the training loss. By our choice of regularizer, we theoretically motivate that global model flatness can be achieved by local (client) model flatness, and develop a computationally efficient flatness metric that can minimize the top eigenvalue and be easily optimized by any of the state-of-the-art FL optimizers. As a result, our proposed approach can effectively improve the generalization performance of the trained global model. Our method also mitigates the need for multiple gradient updates as required by SAM-based FL methods, as the flatness constraints are integrated into the loss.

Further, we theoretically establish that the top eigenvalue of the Hessian of the client's loss function can be minimized by **M**inimizing the **A**ctivation's **N**orm (MAN) of each layer in the client models (see Sec 3.2.3). Our empirical analysis in Sec. 5.1 shows that our proposed flatness constraint leads to a global model with lower top eigenvalue and the trace of the Hessian of loss, compared to the case without the constraint. Since our method achieves the flatness by minimizing the activation norm. Our proposed regularizer (MAN) can be elegantly combined with existing FL techniques such as FedAVG [17], FedDC [4], FedDyn [1] and also SAM-based FL methods such as FedSAM/ASAM [2], FedSpeed [29] etc. and further improve each of their performance to surpass the state-of-the-art on multiple datasets, as well as across the data distributions. For example, we improve the performance of FedDC by 3.2% on CIFAR-100, and by 4.3% on Tiny-ImageNet. For a detailed comparison, see Table 1. The key contributions of this work are:

- We propose a novel FL optimization problem that incorporates a flatness constraint as a regularizer to enhance generalization. Specifically, we minimize the top eigenvalue of the Hessian of the global model's training loss.
- We present theoretical evidence that the top eigenvalue of the layerwise Hessian of client's loss can be minimized by minimizing its layer-wise activation norm. The computational complexity of our regularizer is very low compared to the existing sharpness-aware FL optimizers such as FedSAM/ASAM and FedSpeed.
- Unlike previous works that combine regularization schemes with only the FedAvg algorithm, our approach can be easily integrated atop any state-of-the-art methods, such as FedDC, FedDyn, FedSpeed, FedSAM/ASAM to further enhance their performance.
- We evaluate the effectiveness of our method on CIFAR-100 and Tiny-ImageNet datasets. Our method achieves significant improvements in FL-trained global model accuracy and communication cost reduction compared to competing methods.

2. Related Work

FL was introduced in [17] which is a generalized version of local SGD [28]. This allows to increase the local gradient updates on the clients before sharing with server, which shows a significant reduction in communication costs in the iid setting but not in the non-iid setting. Methods such as MOON [15] and FedProx [16] introduce a regularizer to maintain proximity to the global model while training with the local objective. The key idea is to prevent client models from drifting too far from the global model, thereby facilitating faster convergence or achieving the desired accuracy with minimal communication rounds. In [18], a regularization strategy is proposed that minimizes the Lipschiltz constant, which involves high compute. SCAFFOLD [9] introduced the gradient correction to minimize client drift. Later FedDyn [1] improved upon this by introducing the dynamic regularization term. This was improved in FedDC [4] by local drift correction. In [33], the data-sharing strategy was proposed to improve the performance, which might violate the privacy in FL. Distillation strategies are proposed in FedNTD [14], FedSSD [6] and FedCAD [7] to address the problems associated with non-iid settings. However, these methods are computationally intensive, and FedCAD also requires access to external data. In [20], models were compressed to reduce communication cost. One-shot methods, e.g., [34] were also proposed to reduce communication cost but these incur heavy computation on client.

Recent works such as [2,22] showed that Federated training converges to a sharp minimum. To avoid it, these methods seek flat minimum for client models using SAM [3], ASAM [12] and SWA [8] in the FL setting. Sharpness Aware minimization (SAM) is an optimization technique that improves the generalization of models by converging to flat minima across various tasks [3, 24, 25]. Recently, FedSpeed [29] has been proposed, which can be viewed as a combination of SAM with FedDyn [1]. However, there exist several limitations of SAM-based FL methods. The lack of theoretical results for the convergence of ASAM/SWA in FL is one of them. Moreover, SAM-based FL methods necessitate the tuning of multiple hyper-parameters. In our framework, we address these limitations by introducing a novel problem formulation that helps theoretically connect the flat minimum of each client model to that of the global model as a part of our optimization framework. We will now explain our method in detail in the next section.

3. Method

We first describe the FL optimization problem involving multiple clients and a single server in Sec. 3.1. Next, we describe our problem formulation incorporating flatness constraints along with the minimization of overall training loss (cross-entropy). Subsequently, we re-write the objective in terms of the each client's training loss and top eigenvalue of the Hessian of the client's loss in Sec. 3.2.1. We offer theoretical insights in Sec. 3.2.2 and Sec. 3.2.3 to establish the relationship between activation norms and the layer-wise Hessian eigenvalues, which explains the effectiveness of our method. Based on our theoretical insights, we minimize activation norm along with CE loss to attain flatness in Sec. 3.2.4. In Sec. 5.1, we present empirical evidence that validates our method by showing that it indeed minimizes the top eigenvalue/trace of the Hessian of the global model's training loss.

3.1. Problem Setup

In the FL optimization problem, we consider a scenario in which a single server interacts with n clients or edge devices. It is further assumed that each client k possesses m_k training samples drawn iid from the data distribution $\mathcal{D}_k(x, y)$. The data distributions $\{\mathcal{D}_k(x, y)\}_{k=1}^{n}$ across the clients can be either iid or non-iid.

$$\arg\min_{\mathbf{w}} \; l(\mathbf{w}) \tag{1}$$

where

$$l(\mathbf{w}) = \frac{1}{n}\sum_{k=1}^{n} L_k(\mathbf{w}) \tag{2}$$

where $L_k(\mathbf{w})$ is the client specific objective function and $\mathbf{w}$ denotes model parameters. $L_k(\mathbf{w})$ is defined as follows.

$$L_k(\mathbf{w}) = \mathop{\mathbb{E}}_{x,y \in D_k}[l_k(\mathbf{w}; (x, y))] \tag{3}$$

Here, l_k is cross-entropy loss. The expectation is approximated by averaging over training samples drawn from data distribution ($\mathcal{D}_k$) of the client k. Optimizing $L_k(\mathbf{w})$ directly (i.e., FedAvg) leads to solutions in a sharp minimum. The local models over-fit to client's data distribution and do not generalize well. To mitigate this, we introduce a novel regularizer that induces flatness to global model, which we elucidate in the next section.

3.2. Proposed Method

Minimizing the objective function in Eq. 1 directly, may lead to global model converging to the sharp minimum which can affect the generalization performance [2, 22]. To avoid this, we explicitly introduce the flatness constraints by minimizing the top eigenvalue of the Hessian of the training loss of the global model. Our approach is motivated by the works of centralized setup, where it has been shown that top eigenvalue of Hessian of loss is a typical indicator of flatness [3, 10, 27, 31].

3.2.1 Problem Formulation with Flatness Constraint

We now formulate the problem of FL as a constrained optimization problem with the flatness constraint. We rewrite the Eq. 1 with the constraint as below

$$\arg\min_{\mathbf{w}} \; l(\mathbf{w})$$

such that

$$\lambda_{max}(\mathbf{H}(l(\mathbf{w}))) < \tau \tag{4}$$

where $\lambda_{max}(\mathbf{H}(l(\mathbf{w})))$ denotes the top eigenvalue of the Hessian of the loss $l(\mathbf{w})$. [2] We now translate the constraint in above problem using the penalized loss in Eq. 5, with hyper-parameter $\zeta > 0$ balancing the loss and flatness.

$$\arg\min_{\mathbf{w}} \; \hat{l}(\mathbf{w}) = l(\mathbf{w}) + \zeta\lambda_{max}(\mathbf{H}(l(\mathbf{w}))) \tag{5}$$

Its hard to optimize the term $\lambda_{max}(\mathbf{H}(l(\mathbf{w})))$ for two reasons. Firstly, it needs access to loss functions of all the clients which in FL is local to each client, hence not accessible. Secondly, it needs to compute the eigenvalue of the Hessian of the loss, which is computationally complex. To address the first issue, we simplify optimization in Eq. 5 by upper bounding the $\lambda_{max}(\mathbf{H}(l(\mathbf{w})))$ as described below.

As Hessian is Jacobian of the gradient, we get

$$\mathbf{H}(l(\mathbf{w})) = \frac{1}{n}\sum_{k} \mathbf{H}(L_k(\mathbf{w})) \tag{6}$$

We use the identity[3] in the Eq. 6 to get 7

$$\lambda_{max}(\mathbf{H}(l(\mathbf{w}))) \leq \frac{1}{n}\sum_{k} \lambda_{max}(\mathbf{H}(L_k(\mathbf{w}))) \tag{7}$$

Using inequality 7 and Eq. 2 we upper bound the loss $\hat{l}(\mathbf{w})$ in Eq 5 by the following.

$$\tilde{l}(\mathbf{w}) = \frac{1}{n}\sum_{k}(L_k(\mathbf{w}) + \zeta\lambda_{max}(\mathbf{H}(L_k(\mathbf{w})))) \tag{8}$$

[2] $\mathbf{H}(l(\mathbf{w}))$ denotes the Hessian of the loss $l(\mathbf{w})$.

[3] $\lambda_{max}(\mathbf{S}_1 + \mathbf{S}_2) \leq \lambda_{max}(\mathbf{S}_1) + \lambda_{max}(\mathbf{S}_2)$ where $\mathbf{S}_1$ and $\mathbf{S}_2$ are symmetric matrices.

i.e., $\hat{l}(\mathbf{w}) \leq \tilde{l}(\mathbf{w})$. We now minimize the $\tilde{l}(\mathbf{w})$ as below

$$\arg\min_{\mathbf{w}} \ \tilde{l}(\mathbf{w}) = \frac{1}{n}\sum_{k} L_k(\mathbf{w}) + \zeta\lambda_{max}(\mathbf{H}(L_k(\mathbf{w}))) \quad (9)$$

Minimizing upper bound $\tilde{l}(\mathbf{w})$ will minimize the $\hat{l}(\mathbf{w})$, which is our true goal. We therefore modified the problem in Eq. 1 by including the flatness constraints in Eq. 9. Instead of just minimizing the $L_k(\mathbf{w})$ each client now minimizes the $L_k(\mathbf{w})+\zeta\lambda_{max}(\mathbf{H}(L_k(\mathbf{w})))$. The ζ trades off between the flatness and the training loss. We now address the second issue involving the top eigenvalue of Hessian which is computationally complex by developing a low-cost metric to minimize the $\lambda_{max}(\mathbf{H}(L_k(\mathbf{w})))$.

3.2.2 Bounding the overall Hessian Eigenvalues with the Layer-Wise Hessian Eigenvalues

According to the findings of [27], during the training process the evolution of the top eigenvalue of the overall Hessian is similar to the top eigenvalue of layer-wise Hessian. By layer-wise Hessian, we mean the Hessian of loss is computed with respect to parameters of each layer. We explain the above behavior through the following result.

Theorem 1. *If $\mathbf{H}_{ll} \in R^{d_l}$ denotes the layer l Hessian and $\mathbf{H} \in R^d$ denotes the over all Hessian and $\sum_{l=1}^{L} d_l = d$, where L is the total number of layers. If the Hessian entries are bounded above we then have the following result. $\lambda(\mathbf{H}) \in \cup_{l=1}^{L}[\lambda_{min}(H_{ll}) - \mathcal{O}(max(d_l, d-d_l)), \lambda_{max}(H_{ll}) + \mathcal{O}(max(d_l, d-d_l))]$*

The detailed proof is given in the Sec.2 of supplementary. The theorem says that all the eigenvalues of the overall Hessian $\lambda(\mathbf{H})$ can be upper bounded by the layerwise Hessian's top eigenvalue along with a constant dependent on dimensions. In light of the empirical observations in [27] and above theoretical findings, we opt to minimize the layer-wise Hessian's top eigenvalue instead of minimizing the overall Hessian's top eigenvalue.

3.2.3 Bounding the Layer-Wise Hessian Eigenvalues with the Activation Norms

Minimizing the layer-wise Hessian top eigenvalue is still a challenging task. To address this issue, we further simplify our method based on our results, which indicate that the top eigenvalue of the layer-wise Hessian is upper-bounded by the activation norm of the inputs to that layer. Therefore, by minimizing the activation norm, we can minimize the top eigenvalue of the layer-wise Hessian.

In our analysis, we consider the C class classification problem. The training set $S = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^{N}$ is considered, where each $\mathbf{x}^i \in \mathbb{R}^D$ or $\mathbf{x}^i \in \mathbb{R}^{C\times H\times W}$ and $\mathbf{y}^i \in \{0,1\}^C$ is drawn iid from the distribution $\mathcal{D}$. We then consider an L-layer neural network, where the network outputs the logits $\mathbf{z}^i$. The logits are obtained by a series of fully connected (FC) / convolutional (CONV) layers, followed by a non-linearity, represented concisely by Eq. 10 for a FC layer with parameters ($\{\mathbf{W}_l, \mathbf{b}_l\}$) and Eq. 11 for a CONV layer with parameters ($\{\mathbf{W}_l, \mathbf{b}_l\}$).

$$\mathbf{z}_l^i = \text{FC}(\mathbf{a}_{l-1}^i; \{\mathbf{W}_l, \mathbf{b}_l\}) \quad (10)$$

$$\mathbf{z}_l^i = \text{CONV}(\mathbf{a}_{l-1}^i; \{\mathbf{W}_l, \mathbf{b}_l\}) \quad (11)$$

$$\mathbf{a}_l^i = \sigma(\mathbf{z}_l^i) \quad (12)$$

where $\sigma(.)$ denotes non-linearity $\mathbf{a}_0 = \mathbf{x}^i$, the logits $\mathbf{z}^i = \mathbf{a}_L^i$. We denote the output of softmax on logits $\mathbf{z}^i$ as

$$\hat{\mathbf{y}} = exp(\mathbf{z}^i) / \sum_{m=1}^{C} exp(\mathbf{z}^i[m])) \quad (13)$$

where $\hat{\mathbf{y}} \in [0,1]^C$ Finally, we use the cross-entropy loss

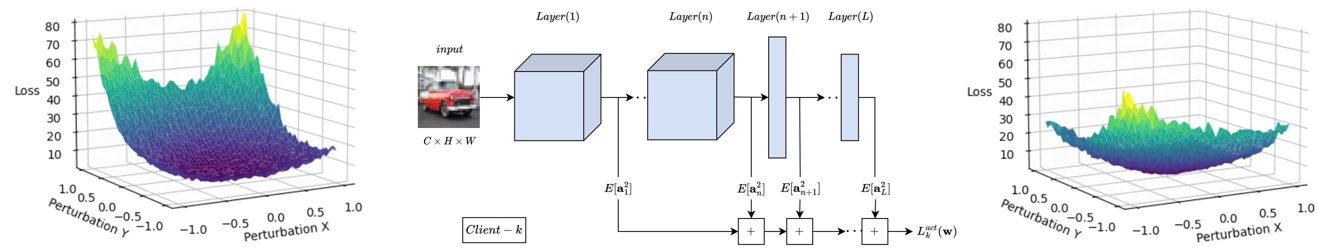
$$\mathcal{L}(\mathbf{y}^i, \hat{\mathbf{y}}^i) = \sum_{c=1}^{C} -\mathbf{y}^i[c] log(\hat{\mathbf{y}}^i[c]). \quad (14)$$

where $\mathbf{x}^i$, $\mathbf{y}^i$ denotes the i^{th} input sample and label respectively. We use the notation $\mathcal{L}^i$ for $\mathcal{L}(\mathbf{y}^i, \hat{\mathbf{y}}^i)$. The overall loss computed on Batch size of B is denoted by $\mathcal{L} = \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}^i$. Finally, we denote $\theta \in \mathbb{R}^d$ as the collection of all the model parameters into a vector of dimension d that denotes the total number of parameters in the network. We explicitly denote Hessian of layer l as $\mathbf{H}_{\mathbf{W}_l}(\mathcal{L})$ We now state the two of our results that relate the layer-wise top eigenvalues to the activation norm of each layer. Consider a FC layer as in Eq. 10 with $\mathbf{a}_{l-1}^i \in \mathbb{R}^{d_{l-1}}$ and weights $\mathbf{W}_l \in \mathbb{R}^{d_{l-1}\times d_l}$. We then have the following result.

Theorem 2. *If $\|\theta\|_2 \leq \tilde{B}$, then the top eigenvalue of layer-wise Hessian for the loss $\mathcal{L}$ w.r.t to $\mathbf{W}_l$ denoted by $\lambda_{max}(\mathbf{H}_{\mathbf{W}_l}(\mathcal{L}))$ for l = 2 to L, computed over the batch of samples for a L layered fully connected neural network for multi-class classification is given by $\lambda_{max}(\mathbf{H}_{\mathbf{W}_l}(\mathcal{L})) \leq \alpha_l \sum_{i\in B} \|\mathbf{a}_{l-1}^i\|_2^2$ where $\alpha_l > 0$.*

Where $\|.\|_2$ denotes the Euclidean norm. We now present a similar result for the CONV layer with input feature map of dimension $\mathbf{a}_{l-1}^i \in \mathbb{R}^{C_{l-1}\times H_{l-1}\times W_{l-1}}$, the output feature map $\mathbf{z}_l^i \in \mathbb{R}^{m\times H_l\times W_l}$ and convolutional kernel $\mathbf{W}_l \in \mathbb{R}^{m\times C_{l-1}\times K_1\times K_2}$, then the below Theorem holds.

Theorem 3. *If $\|\theta\|_2 \leq \tilde{B}$, then the top eigenvalue of layer-wise Hessians for the loss (w.r.t to $\mathbf{W}_l$ for l = 2 to L) computed over the Batch of samples for a L layered convolutional neural network for multi-class classification is given by $\lambda_{max}(\mathbf{H}_{\mathbf{W}_l}(\mathcal{L})) \leq \alpha_l \sum_{i\in B} \|\mathbf{a}_{l-1}^i\|_F^2$ where $\alpha_l^i > 0$.*



(a) FedAvg (b) MAN (c) FedAvg+MAN

Figure 1. We plot the loss surface of the global model trained on CIFAR-100 using FedAvg in 1a. In Fig 1b we show MAN regularizer. We combine FedAvg with MAN (FedAvg+MAN) to obtain the flat loss surface in Fig 1c which has better generalization.

where $\|.\|_F$ is the Frobenious norm[4]. The result of Theorem 2 and Theorem 3 states that for a fully connected layer or a convolutional layer the top eigenvalue of Hessian of layer l can be minimized by minimizing the activation norm of its input $\mathbf{a}_{l-1}^i$. Detailed proofs of Theorems 2 & 3 are provided in the Sec.2 of the supplementary.

3.2.4 Minimizing the Activation Norm

Building on the insights from the aforementioned findings, we address our objective by minimizing the layer-wise activation norms. This, in turn, minimizes the top eigenvalue of the layer-wise Hessian, thereby achieving a reduction in the top eigenvalue of the overall Hessian (i.e. increase flatness). We now present our proposed method, which involves each client k minimizing $f_k(\mathbf{w})$ as defined below (Eq. 15).

$$f_k(\mathbf{w}) \triangleq L_k(\mathbf{w}) + \zeta L_k^{act}(\mathbf{w}) \tag{15}$$

ζ is the balancing parameter between the task-specific loss and the regularization loss. We use L_k^{act} as the substitute for $\lambda_{max}(\mathbf{H}(L_k(\mathbf{w})))$ in the Eq. 9. Our proposed regularizer $L_k^{act}(\mathbf{w})$ computes the second moment of activation's for a layer (l) (denoted by $\mathbf{a}_l$) after non-linearity and then further sum across the L layers. The computation of the activation norm and its impact is shown in Figure 1. Mathematically we describe our regularization term as below

$$L_k^{act}(\mathbf{w}) \triangleq \sum_{l=1}^{L} \mathbb{E}[\|\mathbf{a}_l\|^2] \tag{16}$$

$\mathbb{E}$ denotes the expectation operation approximated by averaging over mini-batch. l denotes the layer index which can be a convolutional layer or a fully connected layer. If l is the CONV layer, $\mathbf{a}_l^i \in \mathbb{R}^{C_l \times H_l \times W_l}$ is activation map of i^{th} sample. we define $\mathbb{E}[\|\mathbf{a}_l\|^2]$ in Eq. 17.

$$\mathbb{E}[\|\mathbf{a}_l\|^2] \coloneqq \frac{1}{BC_lH_lW_l} \sum_{i=0}^{B-1} \sum_{j_1=0}^{C_l-1} \sum_{j_2=0}^{H_l-1} \sum_{j_3=0}^{W_l-1} (\mathbf{a}_l^i[j_1, j_2, j_3])^2 \tag{17}$$

In the above equation B is the batch size, C_l is the feature map, H_l and W_l denotes the height and width of the activations at layer l. This can be compactly written as $\mathbb{E}[\|\mathbf{a}_l\|^2] = \frac{1}{BH_lW_lC_l} \sum_{i=0}^{B-1} \|\mathbf{a}_l^i\|_F^2$. If l is the FC layer, $\mathbf{a}_l^i \in \mathbb{R}^{d_l}$, $\mathbb{E}[\|\mathbf{a}_l\|^2]$ is defined below

$$\mathbb{E}[\|\mathbf{a}_l\|^2] \coloneqq \frac{1}{Bd_l} \sum_{i=0}^{B-1} \sum_{j_1=0}^{d_l-1} (\mathbf{a}_l^i[j_1])^2 \tag{18}$$

B is the batch size and and d_l is feature dimension of the FC layer. Eq. 18 can also be written as $\mathbb{E}[\|\mathbf{a}_l\|^2] = \frac{1}{B} \sum_{i=0}^{B-1} \frac{1}{d_l} \|\mathbf{a}_l^i\|_2^2$, where $\|.\|_2$ is the euclidean norm.

3.2.5 Convergence Analysis

We now analyze the convergence of MAN regularizer when its used by the clients as in Eq. 15 and FedAvg is used for aggregation i.e, FedAvg+MAN. Supppose the loss functions $f_k(\mathbf{w})$ in Eq. 15 satisfies the below assumptions as mentioned in [9].

A 1. *The loss functions f_k are Lipschiltz smooth, i.e., $\|\nabla f_k(\mathbf{x}) - \nabla f_k(\mathbf{y})\| \leq \beta \|\mathbf{x} - \mathbf{y}\|$.*

A 2. $\frac{1}{n} \sum_{k \in [n]} \|\nabla f_k(\mathbf{w})\|^2 \leq G^2 + B^2 \|\nabla f(\mathbf{w})\|^2$,*where $f(\mathbf{w}) = \frac{1}{n} \sum_{k \in [n]} f_k(\mathbf{w})$.This is referred to bounded gradient dissimilarity assumption,*

A 3. *let $\mathbb{E}\|f_k(\mathbf{w}, (x, y)) - f_k(\mathbf{w})\| \leq \sigma^2$, for all k and $\mathbf{w}$. Here $f_k(\mathbf{w}, (x, y))$ is loss evaluated on the sample (x, y) and $f_k(\mathbf{w})$ is expectation across the samples. This is a bounded variance assumption.*

We then have the following proposition.

Proposition 1. *Theorem V of [9] in Appendix D.2: let $\mathbf{w}^* = \arg\min_{\mathbf{w}} \tilde{l}(\mathbf{w})$, the global step-size be α_g and the local step-size be α_l. FedAvg+MAN algorithm will have contracting gradients. If Initial model is $\mathbf{w}^0$, $F = \tilde{l}(\mathbf{w}^0) - \tilde{l}(\mathbf{w}^*)$ and for constant M, then in R rounds, the model $\mathbf{w}^R$ satisfies $\mathbb{E}[\|\nabla f(\mathbf{w}^R)\|^2] \leq O(\frac{\beta M \sqrt{F}}{\sqrt{RLS}} + \frac{\beta^{1/3}(FG)^{2/3}}{(R+1)^{2/3}} + \frac{\beta B^2 F}{R})$.*

[4] $\|\mathbf{a}_{l-1}^i\|_F^2 \coloneqq \sum_{j_1=0}^{C_{l-1}-1} \sum_{j_2=0}^{H_{l-1}-1} \sum_{j_3=0}^{W_{l-1}-1} (\mathbf{a}_{l-1}^i[j_1, j_2, j_3])^2$

The above proposition states that the FedAvg+MAN algorithm requires $\mathcal{O}(\frac{1}{\epsilon^2})$ communication rounds to make the average gradients of the global model smaller, i.e., $\mathbb{E}[\|\nabla \tilde{l}(\mathbf{w}^R)\|^2] \leq \epsilon$. Similar guarantees can be given when we use other FL algorithms with the proposed MAN.

4. Experiments

We perform experiments under both the non-iid and iid setups. For non-iid, we experiment on the label imbalance, whose experimental setups are described below. We adopted the experimental setup described in [1, 4], using CIFAR-100 [11] and Tiny-ImageNet [13] datasets. The model specifications are available in Sec. 3 of the suppl material. Both iid and non-iid data partitioning were tested in our experiments, with 100 clients participating and 10% of them being selected at random for each communication round. Each client was allocated the same number of samples, and the accuracy was measured at the end of 500 communication rounds. Non-iid data was generated using the Dirichlet distribution $Dir(\delta)$, following the approach in [1]. A label distribution vector was sampled for each client from the Dirichlet distribution; the entries of the vector are non-negative and summed to 1. The value of δ controls the degree of non-iid data, with lower values resulting in higher label imbalances. The effect of δ on client distribution is presented in Sec.6 of supplementary. We also conducted a sensitivity analysis of the hyper-parameter ζ on global model accuracy, as described in Sec. 4 of supplementary. The learning rate was set to 0.1 with a decay of 0.998 per communication round, batch size of 50 and 5 epochs per round. All these hyper-parameter settings are consistent with the [1, 4]. We set the hyper-parameter ζ to 0.15 for CIFAR-100 and 0.1 for Tiny-ImageNet, and more details are in Sec.7 of supplementary. We have also performed experiments on the CIFAR-10 dataset. The results are given in Sec 5.2 of the suppl material.

5. Results and Discussion

In Table 1, we summarize our results. We refer to FedAvg+MAN, FedDC+MAN, FedDyn+MAN, FedDC+MAN, FedSpeed+MAN and FedSAM/ASAM+MAN when we use the algorithms FedAvg, FedDC, FedDyn, FedSpeed, and FedSAM/ASAM respectively, along with our flatness inducing regularizer MAN (see Sec.8 of supplementary for algorithm details). From Table 1, we observe that on CIFAR-100, FedDC attains an accuracy of 52.02% and it attains the 50% accuracy is 294 rounds for $\delta = 0.3$ whereas FedDC+MAN attains the accuracy of 55.21% and it attained 50% accuracy in just 144 rounds for the same $\delta = 0.3$, thus leading to an improved performance by 3.2% and saving 150 rounds of communication. Similarly we can see that FedDC+MAN improves FedDC by 2.76%, 3.5% for $\delta = 0.6$ and iid data partitions, respectively. It also saves the 175 and 133 rounds of communication on $\delta = 0.6$ and iid data partitions, respectively. For the Tiny-ImageNet dataset, we improve the performance of FedDC by 4.2%, 4.5%, 5.1% for $\delta = 0.3$, $\delta = 0.6$ and iid data partitions respectively. FedDC+MAN also saves 97,127 and 140 rounds of communication compared to FedDC $\delta = 0.3$, $\delta = 0.6$ and iid data partitions respectively. Similar improvements can be seen for FedDyn, FedSpeed, FedSAM/ASAM as well. To get smoother estimates, we follow the protocol of [1], where we take the average of all the client models for reporting accuracy. In Figures 2 and 3, the performance of the algorithms FedAvg, FedDyn, and FedDC, are compared with flatness-constrained versions FedAvg+MAN, FedDyn+MAN, and FedDC+MAN. Clearly, our flatness-constrained version of algorithms significantly performs better. Figures 2 and 3 are generated for a single training seed. The accuracy vs communication plots for FedSAM/ASAM, and FedSpeed with and without MAN are given in the Sec. 5.1 of the supplementary.

5.1. Empirical Analysis of Hessian

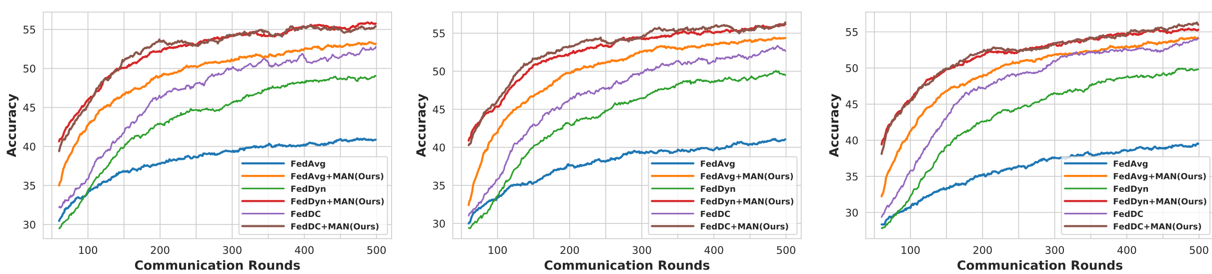
In Fig 4 and Table 2, we perform the Hessian Analysis of the proposed MAN regularizer on the CIFAR-100 and compare it against the baseline algorithms. We observe empirically that when our regularizer MAN is combined with FedAvg, FedDyn and FedDC, it attains a flatter minimum, which is quantified by reduction in top eigenvalue and the trace of the Hessian. The top eigenvalue is a key indicator of better generalization [10, 18, 31]. Due to space constraints, More results of Hessian analysis on the remaining algorithms are presented in Sec 5.3 of supplementary. To better understand the reduction of the trace, we have plotted the top 200 eigenvalues for FedAvg, FedAvg+MAN, FedSAM and FedSAM+MAN. We can see that MAN regularizer not just reduces the top eigenvalue but also reduces the other eigenvalues as well. This is the reason for reduction of the trace. It is important to emphasize that the trace can be lower when negative eigenvalues also contribute significantly. Since we are evaluating the trace at the convergence, we observe that the contribution of negative eigenvalues is negligible, and the trace is dominated by positive eigenvalues. The eigen spectral density Fig 4b confirms this observation. We can also observe that FedAvg+MAN has lower trace compared to FedAvg+MAN from the table 2. This can be observed from Fig 4a, where after the first 50 eigen values FedAvg+MAN has lower eigenvalues consistently. This suggests that MAN regularizer reduces many eigenvalues which is a cause for it's better generalization.

6. Analyzing Computational Cost

We now analyze the total computation cost incorporating the proposed regularizer. We analyze the total number

Table 1. Accuracy and communication round comparisons are presented as follows: Accuracy is given in the format of mean ± standard deviation, accompanied by the number of communication rounds. These rounds are required to achieve 50% accuracy for CIFAR-100 and 28% accuracy for Tiny-ImageNet. The experiments are repeated for three initializations, and the mean and standard deviation of accuracy are reported. The performance of different algorithms is shown on CIFAR-100 & Tiny-ImageNet with and without the MAN regularizer. Accuracy values are reported after 500 communication rounds. On CIFAR-100 FedDC attains accuracy of 52.03% while FedDC+MAN achieves 55.21% accuracy. FedDC reaches 50% accuracy in 294 rounds, whereas FedDC+MAN in just 144 rounds. The utilization of the proposed MAN regularizer clearly enhances the performance and reduces communication rounds for all algorithms.

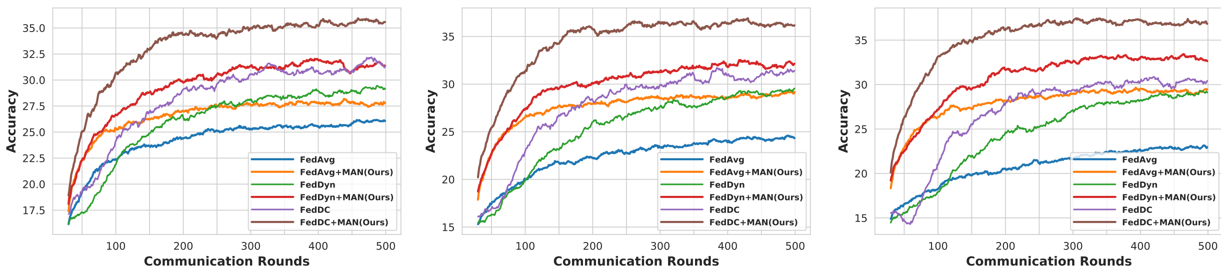
Algorithm	CIFAR-100			Tiny-ImageNet		
	non-iid		iid	non-iid		iid
	$\delta = 0.3$	$\delta = 0.6$		$\delta = 0.3$	$\delta = 0.6$	
FedAvg	40.90 $\pm_{0.62}$ (500+)	40.39 $\pm_{0.59}$ (500+)	39.40 $\pm_{0.84}$ (500+)	25.35 $\pm_{1.16}$ (500+)	24.41 $\pm_{0.41}$ (500+)	23.75 $\pm_{0.99}$ (500+)
FedAvg+MAN (**Ours**)	**52.00** $\pm_{0.36}$ **(206)**	**52.42** $\pm_{0.23}$ **(210)**	**52.59** $\pm_{0.25}$ **(224)**	**28.09** $\pm_{0.26}$ **(437)**	**28.90** $\pm_{0.21}$ **(194)**	**29.11** $\pm_{0.12}$ **(182)**
FedSAM	43.44 $\pm_{0.11}$ (500+)	43.36 $\pm_{0.24}$ (500+)	41.31 $\pm_{0.27}$ (500+)	26.23 $\pm_{0.68}$ (500+)	26.04 $\pm_{0.20}$ (500+)	23.97 $\pm_{0.83}$ (500+)
FedSAM+MAN (**Ours**)	**51.59** $\pm_{0.48}$ **(326)**	**52.62** $\pm_{0.40}$ **(281)**	**52.85** $\pm_{0.13}$ **(301)**	**32.16** $\pm_{0.20}$ **(104)**	**32.60** $\pm_{0.92}$ **(87)**	**31.40** $\pm_{0.30}$ **(82)**
FedASAM	46.00 $\pm_{0.10}$ (500+)	45.48 $\pm_{0.08}$ (500+)	44.18 $\pm_{0.61}$ (500+)	27.50 $\pm_{0.09}$ (500+)	27.05 $\pm_{0.16}$ (500+)	23.96 $\pm_{0.43}$ (500+)
FedASAM+MAN (**Ours**)	**51.23** $\pm_{0.20}$ **(313)**	**51.89** $\pm_{0.09}$ **(351)**	**52.41** $\pm_{0.43}$ **(296)**	**32.50** $\pm_{0.05}$ **(140)**	**32.41** $\pm_{0.32}$ **(136)**	**31.70** $\pm_{0.81}$ **(93)**
FedDyn	49.29 $\pm_{0.30}$ (500+)	49.91 $\pm_{0.41}$ (486)	50.04 $\pm_{0.22}$ (500+)	29.23 $\pm_{0.06}$ (295)	28.99 $\pm_{0.55}$ (308)	29.41 $\pm_{1.33}$ (350)
FedDyn+MAN (**Ours**)	**55.27** $\pm_{0.12}$ **(145)**	**55.63** $\pm_{0.37}$ **(143)**	**55.83** $\pm_{0.56}$ **(157)**	**32.00** $\pm_{0.57}$ **(132)**	**32.44** $\pm_{0.27}$ **(110)**	**32.31** $\pm_{0.38}$ **(108)**
FedDC	52.02 $\pm_{0.79}$ (294)	52.64 $\pm_{0.24}$ (304)	53.25 $\pm_{0.86}$ (289)	31.44 $\pm_{0.43}$ (170)	31.42 $\pm_{0.36}$ (193)	31.21 $\pm_{0.43}$ (201)
FedDC + MAN (**Ours**)	**55.21** $\pm_{0.32}$ **(144)**	**55.40** $\pm_{0.30}$ **(129)**	**56.77** $\pm_{0.31}$ **(156)**	**35.70** $\pm_{0.21}$ **(73)**	**36.07** $\pm_{0.23}$ **(66)**	**36.53** $\pm_{0.03}$ **(61)**
FedSpeed	50.95 $\pm_{0.02}$ (392)	51.33 $\pm_{0.17}$ (390)	50.95 $\pm_{0.51}$ (439)	31.12 $\pm_{0.61}$ (211)	31.10 $\pm_{0.27}$ (228)	29.65 $\pm_{0.11}$ (363)
FedSpeed + MAN (**Ours**)	**55.23** $\pm_{0.15}$ **(163)**	**55.84** $\pm_{0.11}$ **(153)**	**55.89** $\pm_{0.41}$ **(163)**	**34.32** $\pm_{0.63}$ **(108)**	**35.49** $\pm_{0.07}$ **(78)**	**33.02** $\pm_{0.55}$ **(98)**



(a) $\delta = 0.3$ (b) $\delta = 0.6$ (c) iid

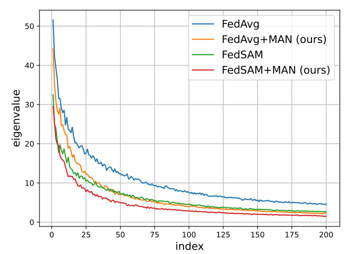
Figure 2. Convergence Comparison on CIFAR-100: We compare performance of the algorithms FedAvg, FedDyn, FedDC and the proposed FedAvg+MAN, FedDyn+MAN and FedDC+MAN for 500 communication rounds. It can be clearly seen that proposed approach significantly improves the existing algorithms across the communication rounds.

of multiplications in a forward pass at a specific layer to measure the computation. Let N_b, C_i, H_i, W_i denote the batch size, input channels, height, and width of the activation map, which is fed to a convolutional layer. We assume the kernel size to be $C_i \times K \times K$ and a number of such filters to be C_o. Thus the output is represented by N_b, C_o, H_o, W_o. Note that batch size N_b remains the same. To compute the single entry in the output activation map, we need C_iK^2 multiplications. For the entire spatial dimension, we need $H_oW_oC_iK^2$, and for all the output channels C_o, we need $C_oH_oW_oC_iK^2$ and for the batch of N_b we have $N_bC_oH_oW_oC_iK^2$ operations. Similarly, let d_i be the input features and d_o be the output feature dimension for a fully connected layer in the network. We need a total of $N_bd_id_o$ operations for fully connected layers. This is summarized in Table 3. The total cost for the convolutional (CONV) layer denoted by TC_{conv} with our regularizer is, $TC_{conv} = N_bC_oH_oW_o(1 + C_iK^2)$. Typicaly the $C_iK^2 >> 1$, this would mean that $TC_{conv} \approx N_bC_oH_oW_oC_iK^2$, which is same as the cost without regularizer from Table 3. The total cost for fully-connected (FC) layer including our regularizer is denoted by TC_{fc} is ($TC_{fc} = N_bd_o(1 + d_i)$). Since $d_i >> 1$ we approximate the total cost for FC layer as $TC_{fc} \approx N_bd_od_i$. This shows that our regularizer incurs negligible cost compared to forward operations without a regularizer for both the convolutional and fully connected layers. Similar analysis can be done for backward pass operations. Our regularizer incurs half the computation required by SAM-based methods, as the SAM based methods require gradient ascent followed by gradient descent.

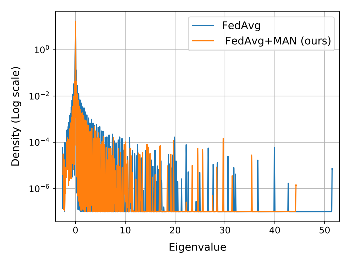


(a) $\delta = 0.3$ (b) $\delta = 0.6$ (c) iid

Figure 3. Convergence Comparison on Tiny-ImageNet: We compare the performance of the algorithms FedAvg, FedDyn, FedDC and the proposed FedAvg+MAN, FedDyn+MAN, and FedDC+MAN for 500 communication rounds. It can be clearly seen that the proposed approach significantly improves the existing algorithms.



(a) Top 200 eigenvalues comparison



(b) Eigen spectral density

Figure 4. Figure 4a shows the comparison of top 200 eigenvalues of FedAvg/FedAvg+MAN and FedSAM/FedSAM+MAN. It can be seen MAN regularizer reduces the top eigenvalues. This explains the reduction of the trace observed. From the Figure 4b we see negative eigenvalues contributes little to the trace.

Table 2. Comparison of top eigenvalues and trace of the algorithms with and without MAN regularizer, lower values are better. We can observe that by augmenting MAN regularization i.e. FedAvg+MAN, FedSAM+MAN, etc., we obtain lower trace and lower top eigenvalues, which is indicative of flat minimum, and hence it attains better accuracy.

Method	CIFAR-100			
	$\delta = 0.3$		$\delta = 0.6$	
	Top eigenvalue	Trace	Top eigenvalue	Trace
FedAvg	51.49	8744	53.39	9056
FedAvg+MAN	**43.80**	**4397**	**42.00**	**4747**
FedSAM	32.46	4909	35.32	5160
FedSAM+MAN	**29.29**	**2709**	**30.05**	**2918**
FedDyn	51.03	6400	47.03	6717
FedDyn+MAN	**44.84**	**3964**	**38.44**	**3966**
FedDC	35.13	3578	35.13	3578
FedDC+MAN	**32.99**	**2974**	**32.99**	**2974**

Table 3. computations without the regularizer and only regularizer.

Forward Computation	CONV Layer	FC Layer
Without Regularizer	$N_b C_o H_o W_o C_i K^2$	$N_b d_i d_o$
Only Regularizer	$N_b H_o W_o C_o$	$N_b d_o$

7. Conclusion

In this paper, we introduce a novel problem formulation in the context of federated learning that includes flatness constraints. By doing so, we demonstrate that by having the client models converge to a flat minimum, the global model also converges to a flat minimum. Additionally, we have simplified the problem at the client level by minimizing activation norms, and we have shown theoretically that this approach can minimize the layer-wise top eigenvalues of the Hessian of the client's loss, which, in turn, leads to minimizing the top eigenvalue of the overall Hessian of the loss. Our proposed methodology can be seamlessly integrated atop existing FL algorithms. In particular, we have integrated our flatness constraint objective on top of popular FL methods such as FedAvg, FedDyn, FedDC and SAM-based methods such as FedSAM/ASAM and FedSpeed. We have shown that this can significantly improve the performance of these baselines. Furthermore, we have demonstrated that our method incurs negligible computation cost after incorporating our regularizer. Our work presents a promising new approach for incorporating flatness constraints as a computationally efficient regularizer into FL algorithms that can lead to better generalization performance. Our work can serve as a beginning for the development of computationally efficient algorithms for inducing flatness constraints.[5]

[5]Acknowledgement: M. Yashwanth and Harsh Rangwani are supported by Prime Minister's Research Fellowship (PMRF). We are thankful for the support.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*. 1, 2, 3, 6

[2] Debora Caldarola, Barbara Caputo, and Marco Ciccone. Improving generalization in federated learning by seeking flat minima. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*, pages 654–672. Springer, 2022. 1, 2, 3

[3] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. 1, 2, 3

[4] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10112–10121, June 2022. 1, 2, 6

[5] Andrew Hard, Chloé M Kiddon, Daniel Ramage, Francoise Beaufays, Hubert Eichner, Kanishka Rao, Rajiv Mathews, and Sean Augenstein. Federated learning for mobile keyboard prediction, 2018. 1

[6] Yuting He, Yiqiang Chen, XiaoDong Yang, Hanchao Yu, Yi-Hua Huang, and Yang Gu. Learning critically: Selective self-distillation in federated learning on non-iid data. *IEEE Transactions on Big Data*, 2022. 2

[7] Yuting He, Yiqiang Chen, Xiaodong Yang, Yingwei Zhang, and Bixiao Zeng. Class-wise adaptive self distillation for heterogeneous federated learning. 2022. 2

[8] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. 2

[9] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020. 1, 2, 5

[10] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017. 1, 2, 3, 6

[11] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Canadian Institute for Advanced Research, 2009. 6

[12] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pages 5905–5914. PMLR, 2021. 2

[13] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 6

[14] Gihun Lee, Yongjin Shin, Minchan Jeong, and Se-Young Yun. Preservation of the global knowledge by not-true self knowledge distillation in federated learning. *arXiv preprint arXiv:2106.03097*, 2021. 2

[15] 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. 2

[16] 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. 2

[17] 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. 1, 2

[18] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8397–8406, June 2022. 2, 6

[19] Jed Mills, Jia Hu, and Geyong Min. Communication-efficient federated learning for wireless edge intelligence in iot. *IEEE Internet of Things Journal*, 7(7):5986–5994, 2019. 1

[20] Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019. 2

[21] Adnan Qayyum, Kashif Ahmad, Muhammad Ahtazaz Ahsan, Ala Al-Fuqaha, and Junaid Qadir. Collaborative federated learning for healthcare: Multi-modal covid-19 diagnosis at the edge. *IEEE Open Journal of the Computer Society*, 3:172–184, 2022. 1

[22] Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. In *International Conference on Machine Learning*, pages 18250–18280. PMLR, 2022. 1, 2, 3

[23] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. Federated learning for emoji prediction in a mobile keyboard. *arXiv preprint arXiv:1906.04329*, 2019. 1

[24] Harsh Rangwani, Sumukh K Aithal, Mayank Mishra, et al. Escaping saddle points for effective generalization on class-imbalanced data. *Advances in Neural Information Processing Systems*, 35:22791–22805, 2022. 2

[25] Harsh Rangwani, Sumukh K Aithal, Mayank Mishra, Arihant Jain, and Venkatesh Babu Radhakrishnan. A closer look at smoothness in domain adversarial training. In *International Conference on Machine Learning*, pages 18378–18399. PMLR, 2022. 2

[26] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020. 1

[27] Adepu Ravi Sankar, Yash Khasbage, Rahul Vigneswaran, and Vineeth N. Balasubramanian. A deeper look at the hessian eigenspectrum of deep neural networks and its applications to regularization. In *AAAI*, 2021. 1, 2, 3, 4

[28] Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018. 2

[29] Yan Sun, Li Shen, Tiansheng Huang, Liang Ding, and Dacheng Tao. Fedspeed: Larger local interval, less communication round, and higher generalization accuracy. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[30] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5:1–19, 2021. 1

[31] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, pages 581–590. IEEE, 2020. 1, 2, 3, 6

[32] Tuo Zhang, Lei Gao, Chaoyang He, Mi Zhang, Bhaskar Krishnamachari, and A Salman Avestimehr. Federated learning for the internet of things: Applications, challenges, and opportunities. *IEEE Internet of Things Magazine*, 5(1):24–29, 2022. 1

[33] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018. 2

[34] Yanlin Zhou, George Pu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. Distilled one-shot federated learning. *arXiv preprint arXiv:2009.07999*, 2020. 2