Federated Learning with Non-IID Data

Yue Zhao*, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, Vikas Chandra

Arm, San Jose, CA
{yue.zhao, meng.li, liangzhen.lai, naveen.suda, damon.civin, vikas.chandra}@arm.com

Abstract

Federated learning enables resource-constrained edge compute devices, such as mobile phones and IoT devices, to learn a shared model for prediction, while keeping the training data local. This decentralized approach to train models provides privacy, security, regulatory and economic benefits. In this work, we focus on the statistical challenge of federated learning when local data is non-IID. We first show that the accuracy of federated learning reduces significantly, by up to ~55% for neural networks trained for highly skewed non-IID data, where each client device trains only on a single class of data. We further show that this accuracy reduction can be explained by the weight divergence, which can be quantified by the earth mover's distance (EMD) between the distribution over classes on each device and the population distribution. As a solution, we propose a strategy to improve training on non-IID data by creating a small subset of data which is globally shared between all the edge devices. Experiments show that accuracy can be increased by ~30% for the CIFAR-10 dataset with only 5% globally shared data.

1   Introduction

Mobile devices have become the primary computing resource for billions of users worldwide and billions more IoT devices are expected to come online over the next few years. These devices generate a tremendous amount of valuable data and machine learning models trained using these data have the potential to improve the intelligence of many applications. But enabling these features on mobile devices usually requires globally shared data on a server in order to train a satisfactory model. This may be impossible or undesirable from a privacy, security, regulatory or economic point of view. Therefore approaches that keep data on the device and share the model have become increasingly attractive.

There has been great progress recently on inference on the device [1, 2]. Federated learning [3, 4, 5] offers an approach to do training locally as well. McMahan *et al.* [3] introduced the $FederatedAveraging(FedAvg)$ algorithm and demonstrated the robustness of $FedAvg$ to train convolutional neural networks (CNNs) on benchmark image classification datasets (e.g. MNIST [6] and CIFAR-10 [7]), and LSTM on a language dataset [8].

A lot of recent research has been done on addressing the communication challenges of federated learning, i.e., how to reduce the communication cost of transferring huge matrices of weights of deep networks, and the unexpected dropout or the synchronization latency due to network connectivity, power, and computing constraints. Bonawitz *et al.* [9] developed an efficient secure aggregation protocol for federated learning, allowing a server to perform computation of high-dimensional data from mobile devices. Konečný *et al.* [10] proposed structured updates and sketched updates to reduce communication costs by two orders of magnitude. Lin *et al.* [11] proposed Deep Gradient Compression (DGC) to reduce the communication bandwidth by two orders of magnitude to train high-quality models.

Other than the communication challenge, federated learning also faces the statistical challenge. Federated learning relies on stochastic gradient descent (SGD), which is widely used in training deep networks with good empirical performances [12, 13, 14, 15, 16, 17]. The IID sampling of the training data is important to ensure that the stochastic gradient is an unbiased estimate of the full gradient [15, 16, 18]. In practice, it is unrealistic to assume that the local data on each edge device is always IID. To approach the non-IID challenge, Smith *et al.* [28] proposed a multi-task learning (MTL) framework and developed MOCHA to address system challenges in MTL. But this approach differs significantly from the previous work on federated learning. McMahan *et al.* [3] have demonstrated that $FedAvg$ can work with certain non-IID data. However, as we will show in Section 2, the accuracy of convolutional neural networks trained with $FedAvg$ algorithm can reduce significantly, up to 11% for MNIST, 51% for CIFAR-10 and 55% for keyword spotting (KWS) datasets, with highly skewed non-IID data.

To address this statistical challenge of federated learning, we show in Section 3 that the accuracy reduction can be attributed to the weight divergence, which quantifies the difference of weights from two different training processes with the same weight initialization. We then prove that the weight divergence in training is bounded by the earth mover's distance (EMD) between the distribution over classes on each device (or client) and the population distribution. This bound is affected by the learning rate, synchronization steps and gradients. Finally, in Section 4, we propose a data-sharing strategy to improve $FedAvg$ with non-IID data by distributing a small amount of globally shared data containing examples from each class. This introduces a trade-off between accuracy and centralization. Experiments show that accuracy can be increased by 30% on CIFAR-10 if we are willing to centralize and distribute 5% shared data.

2   FedAvg on Non-IID data

In this section, we demonstrate the accuracy reduction with $FedAvg$ on non-IID data by training representative neural networks on three datasets.

2.1   Experimental Setup

In this work, we use convolutional neural networks (CNNs) trained on MNIST [6], CIFAR-10 [7] and Speech commands datasets [29]. MNIST and CIFAR-10 are datasets for image classification tasks with 10 output classes. Speech commands dataset consists of 35 words each of 1 sec duration. To make it consistent, we use a subset of the data with 10 key words as keyword spotting (KWS) dataset. For each audio clip, we extract 10 MFCC features for a 30ms frame with a stride of 20ms to generate 50x10 features, which is used for the neural network training. For MNIST and CIFAR-10, we use same CNN architectures as [3], where as for KWS, we use CNN architecture from [1].

The training sets are evenly partitioned into 10 clients. For IID setting, each client is randomly assigned a uniform distribution over 10 classes. For non-IID setting, the data is sorted by class and divided to create two extreme cases: (a) 1-class non-IID, where each client receives data partition from only a single class, and (b) 2-class non-IID, where the sorted data is divided into 20 partitions and each client is randomly assigned 2 partitions from 2 classes.

We use the same notations for $FedAvg$ algorithm as [3]: B, the batch size and E, the number of local epochs. The following parameters are used for $FedAvg$: for MNIST, B = 10 and 100, E = 1 and 5, $\eta = 0.01$ and decay rate = 0.995; for CIFAR-10, B = 10 and 100, E = 1 and 5, $\eta = 0.1$ and decay rate = 0.992; for KWS, B = 10 and 50, E = 1 and 5, $\eta = 0.05$ and decay rate = 0.992. The learning rates are optimized for each dataset and exponentially decays over communication rounds. For SGD, the learning rate and decay rate are identical but B is 10 times larger. This is because the global model from $FedAvg$ is averaged across 10 clients at each synchronization. $FedAvg$ with IID data should be compared to SGD with shuffling data and a batch size K times larger, where K is the number of clients included at each synchronization of $FedAvg$.

2.2   Experimental Results

For the IID experiments, the convergence curves of $FedAvg$ with a batch size of B mostly overlap with the curves of SGD with $B \times 10$ for all the three datasets (Figure 1). Only a small difference is observed for CIFAR-10 that $FedAvg$ with B = 10 converges to 82.62% but SGD with B = 100
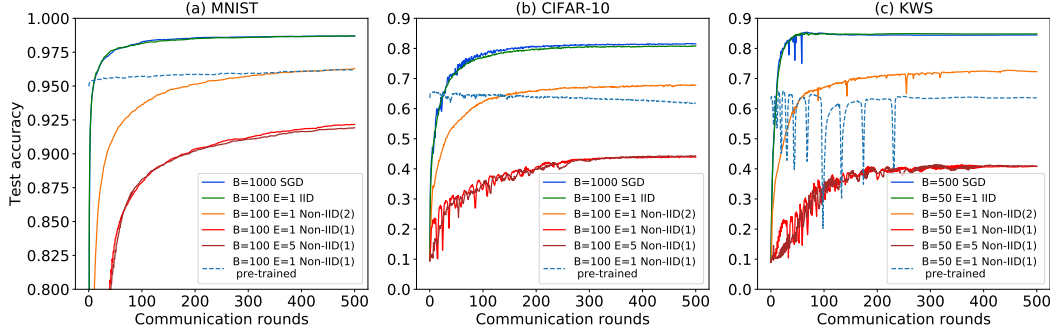
Figure 1: Test accuracy over communication rounds of $FedAvg$ compared to SGD with IID and non-IID data of (a) MNIST (b) CIFAR-10 and (c) KWS datasets. Non-IID(2) represents the 2-class non-IID and non-IID(1) represents the 1-class non-IID.

converges to 84.14% (Figure A.1). Thus, $FedAvg$ achieves SGD-level test accuracy for IID data, which is consistent with the results in [3].

Significant reduction in the test accuracy is observed for $FedAvg$ on non-IID data compared to SGD with matched batch size (Figure 1 and A.1). The accuracy reduction of non-IID data is summarized in Table 1. The maximum accuracy reduction occurs for the most extreme 1-class non-IID data. Moreover, a larger number of local epochs (E = 5) doesn't reduce the loss. The convergence curves mostly overlap for E=1 and E=5. Furthermore, the CNN models pre-trained by SGD doesn't learn from the $FedAvg$ training on non-IID data. For CIFAR-10, the accuracy drops when the pre-trained CNN is trained by $FedAvg$ on non-IID data. Thus, we demonstrate the reduction in the test accuracy of $FedAvg$ for non-IID data. The test accuracy of all the experiments is summarized in Table A.2. Note that the SGD accuracy reported in this paper are not state-of-the-art [6, 30, 31, 1] but the CNNs we train are sufficient for our goal to evaluate federated learning on non-IID data.

Table 1: The reduction in the test accuracy of $FedAvg$ for non-IID data.

Non-IID	B	E	MNIST (%)	CIFAR-10 (%)	KWS (%)
Non-IID(1)	large	1	6.52	37.66	43.64
Non-IID(1)	large	5	6.77	37.11	43.62
Non-IID(2)	large	1	2.4	14.51	12.16
Non-IID(1)	small	1	11.31	51.31	54.5
Non-IID(2)	small	1	1.77	15.61	15.07

3   Weight Divergence due to Non-IID Data

In Figure 1 and A.1, it is interesting to note that the reduction is less for the 2-class non-IID data than for the 1-class non-IID data. It indicates that the accuracy of $FedAvg$ may be affected by the exact data distribution, i.e., the skewness of the data distribution. Since the test accuracy is dictated by the trained weights, another way to compare $FedAvg$ with SGD is to look at the difference of the weights relative to those of SGD, with the same weight initialization. It is termed as weight divergence and it can be computed by the following equation:

$$weight\ divergence = ||\boldsymbol{w}^{FedAvg} - \boldsymbol{w}^{SGD}||/||\boldsymbol{w}^{SGD}|| \qquad (1)$$

As shown in Figure 2, the weight divergence of all the layers increases as the data become more non-IID, from IID to 2-class non-IID to 1-class non-IID. Thus, an association between the weight divergence and the skewness of the data is expected. The accuracy reduction found in Section 2 can be understood in terms of weight divergence, which quantifies the difference of weights from two different training processes with the same weight initialization. In this section, we formally analyze the origin of the weight divergence. In Section 3.1, we provide an illustrative example and a formal proposition to demonstrate that the root cause of the weight divergence is due to the distance between the data distribution on each client and the population distribution. Specifically, we find such distance can be evaluated with the earth mover's distance (EMD) between the distributions. Then, in Section
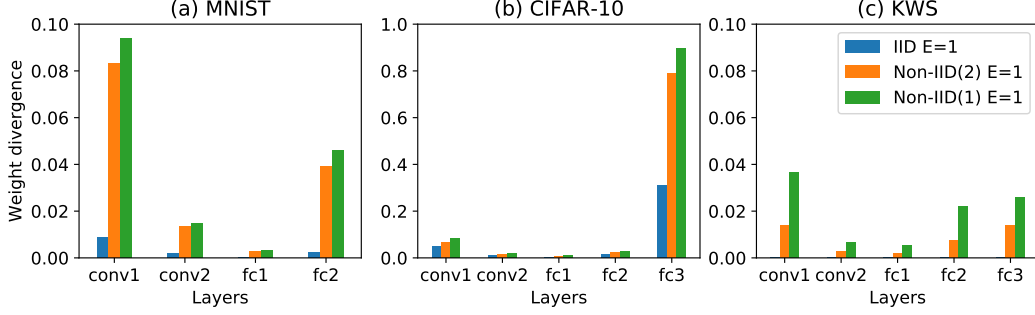
Figure 2: Weight divergence of CNN layers for IID, 2-class non-IID and 1-class non-IID.

3.2, we validate the proposition with experiments that demonstrate the impact of EMD on the weight divergence and test accuracy of $FedAvg$.

3.1 Mathematical demonstration

We formally define the problem of federated learning and analyze the origin of the weight divergence. We consider a C class classification problem defined over a compact space $\mathcal{X}$ and a label space $\mathcal{Y} = [C]$, where $[C] = \{1, \ldots, C\}$. The data point $\{\boldsymbol{x}, y\}$ distributes over $\mathcal{X} \times \mathcal{Y}$ following the distribution p. A function $f : \mathcal{X} \to \mathcal{S}$ maps $\boldsymbol{x}$ to the probability simplex $\mathcal{S}$, where $\mathcal{S} = \{\boldsymbol{z} | \sum_{i=1}^{C} z_i = 1, z_i \geq 0, \forall i \in [C]\}$ with f_i denoting the probability for the ith class. f is parameterized over the hypothesis class $\boldsymbol{w}$, i.e., the weight of the neural network. We define the population loss $\ell(\boldsymbol{w})$ with the widely used cross-entropy loss as

$$\ell(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{x}, y \sim p}[\sum_{i=1}^{C} \mathbb{1}_{y=i} \log f_i(\boldsymbol{x}, \boldsymbol{w})] = \sum_{i=1}^{C} p(y = i)\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w})].$$

To simplify the analysis, we ignore the generalization error and assume the population loss is optimized directly. Therefore, the learning problem becomes

$$\min_{\boldsymbol{w}} \sum_{i=1}^{C} p(y = i)\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w})].$$

To determine $\boldsymbol{w}$, SGD solves the optimization problem iteratively. Let $\boldsymbol{w}_t^{(c)}$ denotes weight after t-th update in the centralized setting. Then, centralized SGD performs the following update:

$$\boldsymbol{w}_t^{(c)} = \boldsymbol{w}_{t-1}^{(c)} - \eta \nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}_{t-1}^{(c)}) = \boldsymbol{w}_{t-1}^{(c)} - \eta \sum_{i=1}^{C} p(y = i)\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w}_{t-1}^{(c)})].$$

In federated learning, we assume there are K clients. Let $n^{(k)}$ denote the amount of data and $p^{(k)}$ denote the data distribution on client $k \in [K]$. On each client, local SGD is conducted separately. At iteration t on client $k \in [K]$, local SGD performs

$$\boldsymbol{w}_t^{(k)} = \boldsymbol{w}_{t-1}^{(k)} - \eta \sum_{i=1}^{C} p^{(k)}(y = i)\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w}_{t-1}^{(k)})].$$

Then, assume the synchronization is conducted every T steps and let $\boldsymbol{w}_{mT}^{(f)}$ denote the weight calculated after the m-th synchronization, then, we have

$$\boldsymbol{w}_{mT}^{(f)} = \sum_{k=1}^{K} \frac{n^{(k)}}{\sum_{k=1}^{K} n^{(k)}} \boldsymbol{w}_{mT}^{(k)}.$$

The divergence between $\boldsymbol{w}_{mT}^{(f)}$ and $\boldsymbol{w}_{mT}^{(c)}$ can be understood with the illustration in Figure 3. When data is IID, for each client k, the divergence between $\boldsymbol{w}_t^{(k)}$ and $\boldsymbol{w}_t^{(c)}$ is small and after the m-th

4

synchronization, $\boldsymbol{w}_{mt}^{(f)}$ is still close to $\boldsymbol{w}_{mt}^{(c)}$. When data is non-IID, for each client k, due to the distance between the data distribution, the divergence between $\boldsymbol{w}_t^{(k)}$ and $\boldsymbol{w}_t^{(c)}$ becomes much larger and accumulates very fast, which makes the divergence between $\boldsymbol{w}_{mT}^{(f)}$ and $\boldsymbol{w}_{mT}^{(c)}$ much larger. To formally bound the weight divergence between $\boldsymbol{w}_{mT}^{(f)}$ and $\boldsymbol{w}_{mT}^{(c)}$, we have the following proposition.
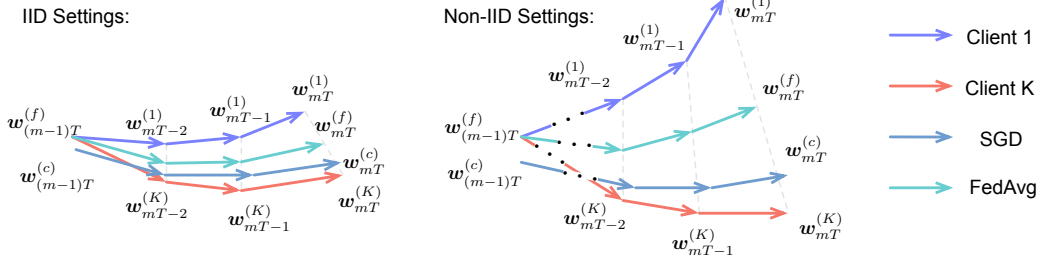

Figure 3: Illustration of the weight divergence for federated learning with IID and non-IID data.

Proposition 3.1. *Given K clients, each with $n^{(k)}$ i.i.d samples following distribution $p^{(k)}$ for client $k \in [K]$. If $\nabla_{\boldsymbol{w}} \mathbb{E}_{\boldsymbol{x}|y=i} \log f_i(\boldsymbol{x}, \boldsymbol{w})$ is $\lambda_{\boldsymbol{x}|y=i}$-Lipschitz for each class $i \in [C]$ and the synchronization is conducted every T steps, then, we have the following inequality for the weight divergence after the m-th synchronization,*

$$||\boldsymbol{w}_{mT}^{(f)} - \boldsymbol{w}_{mT}^{(c)}|| \leq \sum_{k=1}^{K} \frac{n^{(k)}}{\sum_{k=1}^{K} n^{(k)}} (a^{(k)})^T ||\boldsymbol{w}_{(m-1)T}^{(f)} - \boldsymbol{w}_{(m-1)T}^{(c)}||$$

$$+ \eta \sum_{k=1}^{K} \frac{n^{(k)}}{\sum_{k=1}^{K} n^{(k)}} \sum_{i=1}^{C} ||p^{(k)}(y=i) - p(y=i)|| \sum_{j=1}^{T-1} (a^{(k)})^j g_{max}(\boldsymbol{w}_{mT-1-k}^{(c)}),$$

(2)

where $g_{max}(\boldsymbol{w}) = \max_{i=1}^{C} ||\nabla_{\boldsymbol{w}} \mathbb{E}_{\boldsymbol{x}|y=i} \log f_i(\boldsymbol{x}, \boldsymbol{w})||$ and $a^{(k)} = 1 + \eta \sum_{i=1}^{C} p^{(k)}(y=i)\lambda_{\boldsymbol{x}|y=i}$.

Detailed proof of Proposition 3.1 can be found in Appendix A.3. Based on Proposition 3.1, we have the following remarks.

Remark 3.2. *The weight divergence after the m-th synchronization mainly comes from two parts, including the weight divergence after the $(m-1)$-th divergence, i.e., $||\boldsymbol{w}_{(m-1)T}^{(f)} - \boldsymbol{w}_{(m-1)T}^{(c)}||$, and the weight divergence induced by the probability distance for the data distribution on client k compared with the actual distribution for the whole population, i.e., $\sum_{i=1}^{C} ||p^{(k)}(y=i) - p(y=i)||$.*

Remark 3.3. *The weight divergence after the $(m-1)$-th synchronization is amplified by $\sum_{k=1}^{K} \frac{n^{(k)}(a^{(k)})^T}{\sum_{k=1}^{K} n^{(k)}}$. Since $a^{(k)} \geq 1$, $\sum_{k=1}^{K} \frac{n^{(k)}(a^{(k)})^T}{\sum_{k=1}^{K} n^{(k)}} \geq 1$. Hence, if different clients start from different initial $\boldsymbol{w}$ in the federated learning, then, even if the data is IID, large weight divergence will still be encountered, which leads to degraded accuracy.*

Remark 3.4. *When all the clients start from the same initialization as the centralized settings, $\sum_{i=1}^{C} ||p^{(k)}(y=i) - p(y=i)||$ becomes the root cause of the weight divergence. This term is EMD between the data distribution on client k and the population distribution, when the distance measurement is defined as $||p^{(k)}(y=i) - p(y=i)||$. The impact of EMD is affected by the learning rate η, the number of steps before synchronization T, and the gradient $g_{max}(\boldsymbol{w}_{mT-1-k}^{(c)})$.*

Based on Proposition 3.1, we validate that EMD is a good metric to quantify the weight divergence and thus the test accuracy of $FedAvg$ with non-IID data in Section 3.2.

3.2 Experimental Validation

3.2.1 Experimental Setup

The training set is sorted and partitioned into 10 clients, M examples per client. Eight values are chosen for EMD listed in Table 2. Because there may exist various distributions for one EMD, we

aim to generate five distributions to compute the average and variations of the weight divergence and the test accuracy. First, one probability distribution P over 10 classes is generated for one EMD. Based on M and P, we can compute the number of examples over 10 classes for one client. Second, a new distribution P' is generated by shifting the 10 probabilities of P by 1 element. The number of examples for the second client can be computed based on P'. This procedure is repeated for the other 8 clients. Thus, all the 10 clients have a distribution of M examples over 10 classes and each example is only used once. Finally, the above two steps are repeated 5 times to generate 5 distributions for each EMD. The CNNs are trained by $FedAvg$ over 500 communication rounds on the data processed from the above procedures. There are the key parameters used for training: for MNIST, B = 100, E = 1, η = 0.01, decay rate = 0.995; for CIFAR-10, B = 100, E = 1, η = 0.1, decay rate = 0.992; for KWS, B = 50, E = 1, η = 0.05, decay rate = 0.992. The weight divergence is computed according to Eq. (1) after 1 synchronization (i.e., 1 communication round).

3.2.2 Weight Divergence vs. EMD

The mean and standard deviation of the weight divergence are computed over 5 distributions for each EMD. For all the three datasets, the weight divergence of each layer increases with EMD as shown in Figure 4. The initial weights are identical for all SGD, IID and non-IID experiments on each dataset. Thus, according to Remark 3.2, the weight divergence after 1 synchronization is not affected by the $(m-1)$-th divergence, $||\boldsymbol{w}_{(m-1)T}^{(f)} - \boldsymbol{w}_{(m-1)T}^{(c)}||$, because it is zero when $m = 1$. Therefore, the results in Figure 4 support Proposition 3.1 that the bound of weight divergence is affected by EMD. This effect is more significant in the first convolutional layer and the last fully connected layer. Moreover, the maximum weight divergence for CIFAR-10 is significantly higher than that for MNIST and KWS, which is affected by the gradient term in Eq. (2) due to the problem itself and different CNN architectures. Note that the initial weights are also identical across the clients to avoid accuracy loss according to Remark 3.3, which is consistent with the significant increase in the loss when averaging models with different initialization [3, 32].

3.2.3 Test Accuracy vs. EMD

The mean and standard deviation of the test accuracy are computed over the same 5 distributions for each EMD. The results are summarized in Table 2 and are plotted against EMD in Figure 5. For all the three datasets, the test accuracy decreases with EMD. The decreasing rate is relatively small at first and becomes larger as the data becomes more non-IID. So there is a trade-off between balancing non-IID data towards IID and improving the accuracy of $FedAvg$. The error bars on the plots represent the variations of test accuracy due to various distributions for each EMD. To take a closer look at the variations, the boxplots show the test accuracy across 5 runs when EMD = 1.44. Moreover, Table 2 shows that the maximum variation is less than 0.086% for MNIST, 2% for CIFAR-10, and 1% for KWS, respectively. Thus, the accuracy is affected by EMD instead of the underlying distributions. It indicates that EMD can be used to estimate the accuracy of $FedAvg$ given a distribution of data. In addition, the maximum reduction in the accuracy is 6.53% for MNIST, 37.03% for CIFAR-10 and 40.21% for KWS, respectively. This difference may be affected by the gradient in Eq. (2) depending on the problem itself.
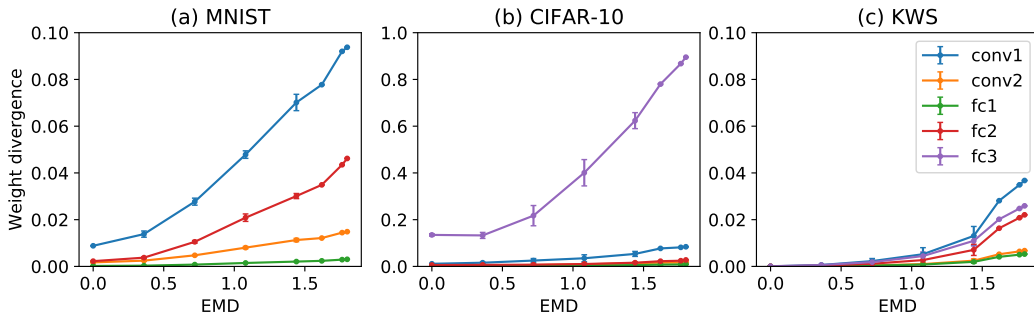

Figure 4: Weight divergence vs. EMD across CNN layers on (a) MNIST, (b) CIFAR-10 and (c) KWS datasets. The mean value and standard deviation are computed over 5 distributions for each EMD.
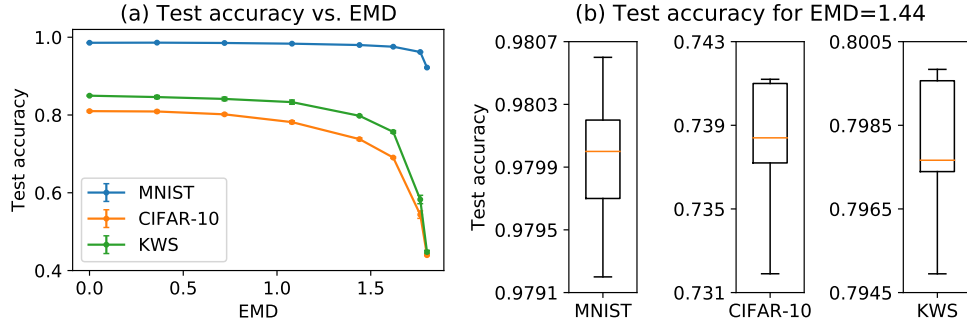
Figure 5: (a) Test accuracy vs. EMD for $FedAvg$ and (b) boxplots of weight divergence when EMD = 1.44 for MNIST, CIFAR-10 and KWS datasets. The mean and standard deviation are computed over 5 distributions for each EMD.

Table 2: The mean and standard deviation of the test accuracy of $FedAvg$ over 5 distributions. The standard deviation is very small compared to the scale of the mean value.

Earth mover's distance (EMD)		0	0.36	0.72	1.08	1.44	1.62	1.764	1.8
MNIST	mean	0.9857	0.9860	0.9852	0.9835	0.9799	0.9756	0.962	0.922
	std ($\times 10^{-4}$)	6.431	2.939	4.604	4.308	4.716	8.085	8.232	1.939
CIFAR-10	mean	0.8099	0.8090	0.8017	0.7817	0.7379	0.6905	0.5438	0.4396
	std ($\times 10^{-3}$)	2.06	2.694	2.645	3.622	3.383	2.048	9.655	1.068
KWS	mean	0.8496	0.8461	0.8413	0.8331	0.7979	0.7565	0.5827	0.4475
	std ($\times 10^{-3}$)	1.337	3.930	4.410	5.387	1.763	3.329	1.078	4.464

4   Proposed Solution

In this section, we propose a data-sharing strategy to improve $FedAvg$ with non-IID data by creating a small subset of data which is globally shared between all the edge devices. Experiments show that test accuracy can be increased by ~30% on CIFAR-10 dataset with only 5% globally shared data.

4.1   Motivation

As shown in Figure 5, the test accuracy falls sharply with respect to EMD beyond a certain threshold. Thus, for highly skewed non-IID data, we can significantly increase the test accuracy by slightly reducing EMD. As we have no control on the clients' data, we can distribute a small subset of global data containing a uniform distribution over classes from the cloud to the clients. This fits in with the initialization stage of a typical federated learning setting. In addition, instead of distributing a model with random weights, a warm-up model can be trained on the globally shared data and distributed to the clients. Because the globally shared data can reduce EMD for the clients, the test accuracy is expected to improve.
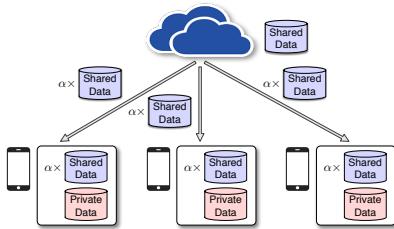

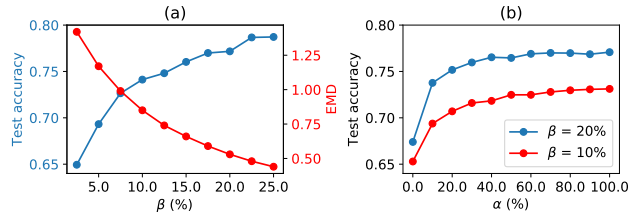Figure 6: Illustration of the data-sharing strategy.

Figure 7: (a) Test accuracy and EMD vs. β (b) Test accuracy vs. the distributed fraction α

7

4.2 Data-sharing Strategy

Herein, we propose a data-sharing strategy in the federated learning setting as illustrated in Figure 6. A globally shared dataset G that consists of a uniform distribution over classes is centralized in the cloud. At the initialization stage of $FedAvg$, the warm-up model trained on G and a random α portion of G are distributed to each client. The local model of each client is trained on the shared data from G together with the private data from each client. The cloud then aggregates the local models from the clients to train a global model with $FedAvg$. There are two trade-offs: (a) the trade-off between the test accuracy and the size of G, which is quantified as $\beta = \frac{\|G\|}{\|D\|} \times 100\%$, where D represents the total data from the clients (b) the trade-off between the test accuracy and α. The following experiments are performed on CIFAR-10 to address these two trade-offs.

The CIFAR-10 training set is partitioned into two parts, the client part D with 40,000 examples and the holdout part H with 10,000 examples. D is partitioned into 10 clients with 1-class non-IID data. H is used to create 10 random $G's$ with β ranging from 2.5% to 25%. First, each entire G is merged with the data of each client and 10 CNNs are trained by $FedAvg$ on the merged data from scratch over 300 communication rounds. The test accuracy is plotted against β in Figure 7. Second, we pick two specific $G's$, i.e., $G_{10\%}$ when $\beta = 10\%$ and $G_{20\%}$ when $\beta = 20\%$. For each G, (a) a warm-up CNN model is trained on G to a test accuracy of ~60% (b) only a random α portion is merged with the data of each client and the warm-up model is trained on the merged data. The test accuracy is plotted against α in Figure 7. The same training parameters for Section 3 are used.

As shown in Figure 7(a), the test accuracy increases up to 78.72% as β increases. Even with a lower β = 10%, we can still achieve a test accuracy of 74.12% for the extreme 1-class non-IID data, compared to 44% without the data-sharing strategy. Moreover, it turns out that it is not necessary to distribute the entire G to the clients to achieve a similar accuracy. Instead, only a random portion of G needs to be distributed to each client. As shown in Figure 7(b), the test accuracy of the warm-up model slowly increases with α, up to 77.08% for $G_{20\%}$ and 73.12% for $G_{10\%}$. In particular, after an initially quick rise, the test accuracy changes less than 1% for both $G_{20\%}$ and $G_{10\%}$ when α changes from 50% to 100%. Therefore, we can further reduce the size of the data that is actually received by each client with a proper selection of α. For instance, if we are willing to centralize 10% of the total clients' data in the cloud and randomly distribute 50% of the globally shared data to the clients as well as a warm-up model at the initialization stage of federated learning, the test accuracy of $FedAvg$ can be increased by 30% for the extreme 1-class non-IID data while only 5% of the total data is actually received by each client.

In summary, the data-sharing strategy offers a solution for federated learning with non-IID data. The size of the globally shared dataset and the random distributed fraction (α) can be tuned depending on the problems and applications. The strategy only needs to be performed once when federated learning is initialized, so the communication cost is not a major concern. The globally shared data is a separate dataset from the clients' data so it is not privacy sensitive.

5 Conclusion

Federated learning will play a key role in distributed machine learning where data privacy is of paramount importance. Unfortunately, the quality of model training degrades if each of the edge devices sees a unique distribution of data. In this work, we first show that the accuracy of federated learning reduces significantly, by up to ~55% for neural networks trained on highly skewed non-IID data. We further show that this accuracy reduction can be explained by the weight divergence, which can be quantified by the earth movers distance (EMD) between the distribution over classes on each device and the population distribution. As a solution, we propose a strategy to improve training on non-IID data by creating a small subset of data which is globally shared between all the edge devices. Experiments show that accuracy can be increased by ~30% for the CIFAR-10 dataset with only 5% globally shared data. There are still quite a few challenges left to make federated learning mainstream, but improving model training on non-IID data is key to make progress in this area.

References

[1] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers," *arXiv preprint arXiv:1711.07128*, 2017.

[2] L. Lai, N. Suda, and V. Chandra, "Cmsis-nn: Efficient neural network kernels for arm cortex-m cpus," *arXiv preprint arXiv:1801.06601*, 2018.

[3] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Int. Conf. on Artificial Intelligence and Statistics*, 2017.

[4] J. Konečný, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," *arXiv preprint arXiv:1511.03575*, 2015.

[5] H. B. McMahan and D. Ramage, "Federated learning: Collaborative machine learning without centralized training data," in *Google*, 2017.

[6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, pp. 2278–2324, 1998.

[7] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," in *Technical Report, University of Toronto*, 2009.

[8] W. Shakespeare, "The complete works of william shakespeare," in *https://www.gutenberg.org/ebooks/100*.

[9] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, ACM, 2017.

[10] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[11] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," *arXiv preprint arXiv:1712.01887*, 2017.

[12] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[13] T. Tieleman and G. Hinton, "Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning," tech. rep., Technical Report. Available online: https://zh. coursera. org/learn/neuralnetworks/lecture/YQHki/rmsprop-divide-the-gradient-by-a-running-average-of-its-recent-magnitude (accessed on 21 April 2017).

[14] D. Kinga and J. B. Adam, "A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[15] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, Springer, 2010.

[16] A. Rakhlin, O. Shamir, K. Sridharan, *et al.*, "Making gradient descent optimal for strongly convex stochastic optimization.," in *ICML*, Citeseer, 2012.

[17] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, *et al.*, "Large scale distributed deep networks," in *Advances in neural information processing systems*, pp. 1223–1231, 2012.

[18] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2341–2368, 2013.

[19] T. M. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman, "Project adam: Building an efficient and scalable deep learning training system.," in *OSDI*, vol. 14, pp. 571–582, 2014.

[20] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *Advances in neural information processing systems*, pp. 693–701, 2011.

[21] J. Duchi, M. I. Jordan, and B. McMahan, "Estimation, optimization, and parallelism when data is sparse," in *Advances in Neural Information Processing Systems*, pp. 2832–2840, 2013.

[22] R. Leblond, F. Pedregosa, and S. Lacoste-Julien, "Asaga: asynchronous parallel saga," *arXiv preprint arXiv:1606.04809*, 2016.

[23] S. J. Reddi, A. Hefny, S. Sra, B. Poczos, and A. J. Smola, "On variance reduction in stochastic gradient descent and its asynchronous variants," in *Advances in Neural Information Processing Systems*, pp. 2647–2655, 2015.

[24] C. M. De Sa, C. Zhang, K. Olukotun, and C. Ré, "Taming the wild: A unified analysis of hogwild-style algorithms," in *Advances in neural information processing systems*, pp. 2674–2682, 2015.

[25] H. Mania, X. Pan, D. Papailiopoulos, B. Recht, K. Ramchandran, and M. I. Jordan, "Perturbed iterate analysis for asynchronous stochastic optimization," *arXiv preprint arXiv:1507.06970*, 2015.

[26] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous sgd," *arXiv preprint arXiv:1604.00981*, 2016.

[27] Q. Meng, W. Chen, Y. Wang, Z.-M. Ma, and T.-Y. Liu, "Convergence analysis of distributed stochastic gradient descent with shuffling," *arXiv preprint arXiv:1709.10432*, 2017.

[28] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Advances in Neural Information Processing Systems*, pp. 4427–4437, 2017.

[29] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[31] B. Graham, "Fractional max-pooling," *arXiv preprint arXiv:1412.6071*, 2014.

[32] I. J. Goodfellow, O. Vinyals, and A. M. Saxe, "Qualitatively characterizing neural network optimization problems," in *arXiv preprint arXiv:1412.6544*, 2014.

A Appendix

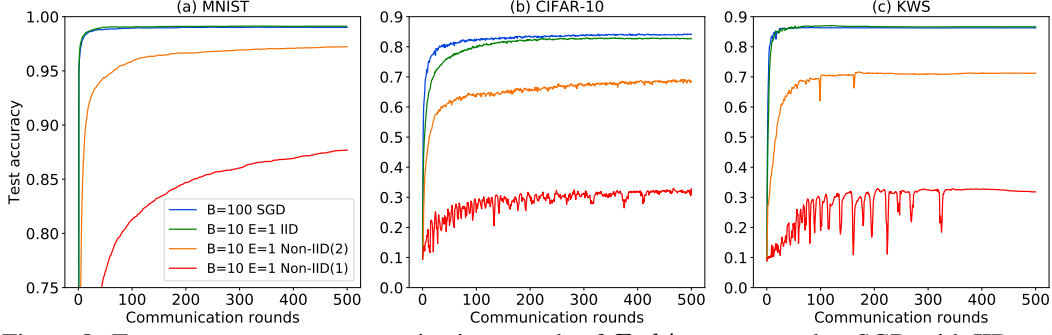## A.1 Test accuracy over communication rounds for a smaller batch size

Figure 8: Test accuracy over communication rounds of $FedAvg$ compared to SGD with IID and non-IID data of (a) MNIST (b) CIFAR-10 and (c) KWS datasets. Non-IID(2) represents the 2-class non-IID and non-IID(1) represents the 1-class non-IID.

A.2 Test accuracy of SGD and $FedAvg$ with IID or non-IID data

Table 3: The test accuracy of SGD and $FedAvg$ with IID or non-IID data.

Training	B	E	MNIST (%)	CIFAR-10 (%)	KWS (%)
SGD	large	NA	98.69	81.51	84.46
FedAvg IID	large	1	98.69	80.83	84.82
FedAvg non-IID(2)	large	1	96.29	67.00	72.30
FedAvg non-IID(1)	large	1	92.17	43.85	40.82
FedAvg non-IID(1)	large	5	91.92	44.40	40.84
Pre-trained non-IID(1)	large	1	96.19	61.72	63.58
SGD	small	NA	99.01	84.14	86.28
FedAvg IID	small	1	99.12	82.62	86.64
FedAvg non-IID(2)	small	1	97.24	68.53	71.21
FedAvg non-IID(1)	small	1	87.70	32.83	31.78

A.3 Proof of Proposition 3.1

Proof. Based on the definition of $\boldsymbol{w}_{mT}^{(f)}$ and $\boldsymbol{w}_{mT}^{(c)}$, we have

11

$$||\boldsymbol{w}_{mT}^{(f)} - \boldsymbol{w}_{mT}^{(c)}||$$

$$=||\sum_{k=1}^{K} \frac{n^{(k)}}{\sum_{k=1}^{K} n^{(k)}} \boldsymbol{w}_{mT}^{(k)} - \boldsymbol{w}_{mT}^{(c)}||$$

$$=||\sum_{k=1}^{K} \frac{n^{(k)}}{\sum_{k=1}^{K} n^{(k)}} (\boldsymbol{w}_{mT-1}^{(k)} - \eta \sum_{i=1}^{C} p^{(k)}(y=i)\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w}_{mT-1}^{(k)})]$$

$$- \boldsymbol{w}_{mT-1}^{(c)} + \eta \sum_{i=1}^{C} p(y=i)\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w}_{mT-1}^{(c)})])$$

$$\overset{1}{\leq}||\sum_{k=1}^{K} \frac{n^{(k)}}{\sum_{k=1}^{K} n^{(k)}} \boldsymbol{w}_{mT-1}^{(k)} - \boldsymbol{w}_{mT-1}^{(c)}||$$

$$+ \eta||\sum_{k=1}^{K} \frac{n^{(k)}}{\sum_{k=1}^{K} n^{(k)}} \sum_{i=1}^{C} p^{(k)}(y=i)(\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w}_{mT-1}^{(k)})] - \nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w}_{mT-1}^{(c)})])||$$

$$\overset{2}{\leq}\sum_{k=1}^{K} \frac{n^{(k)}}{\sum_{k=1}^{K} n^{(k)}}(1 + \eta \sum_{i=1}^{C} p^{(k)}(y=i)\lambda_{\boldsymbol{x}|y=i})||\boldsymbol{w}_{mT-1}^{(k)} - \boldsymbol{w}_{mT-1}^{(c)}||.$$

Here, inequality 1 holds because for each class $i \in [C]$, $p(y=i) = \sum_{k=1}^{K} \frac{n^{(k)}}{\sum_{k=1}^{K} n^{(k)}} p^{(k)}(y=i)$, i.e., the data distribution over all the clients is the same as the distribution over the whole population. Inequality 2 holds because we assume $\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w})]$ is $\lambda_{\boldsymbol{x}|y=i}$-Lipschitz.

In terms of $||\boldsymbol{w}_{mT-1}^{(k)} - \boldsymbol{w}_{mT-1}^{(c)}||$ for client $k \in [K]$, we have

$$||\boldsymbol{w}_{mT-1}^{(k)} - \boldsymbol{w}_{mT-1}^{(c)}||$$

$$=||\boldsymbol{w}_{mT-2}^{(k)} - \eta \sum_{i=1}^{C} p^{(k)}(y=i)\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w}_{mT-2}^{(k)})]$$

$$- \boldsymbol{w}_{mT-2}^{(c)} + \eta \sum_{i=1}^{C} p(y=i)\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w}_{mT-2}^{(k)})]||$$

$$\leq||\boldsymbol{w}_{mT-2}^{(k)} - \boldsymbol{w}_{mT-2}^{(c)}|| + \eta||\sum_{i=1}^{C} p^{(k)}(y=i)\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w}_{mT-2}^{(k)})]$$

$$- \sum_{i=1}^{C} p(y=i)\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w}_{mT-2}^{(c)})]||$$

$$\overset{3}{\leq}||\boldsymbol{w}_{mT-2}^{(k)} - \boldsymbol{w}_{mT-2}^{(c)}|| + \eta||\sum_{i=1}^{C} p^{(k)}(y=i)(\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w}_{mT-2}^{(k)})] - \nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w}_{mT-2}^{(c)})])||$$

$$+ \eta||\sum_{i=1}^{C} (p^{(k)}(y=i) - p(y=i))\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x}, \boldsymbol{w}_{mT-2}^{(c)})]||$$

$$\overset{4}{\leq}(1 + \eta \sum_{i=1}^{C} p^{(k)}(y=i)L_{\boldsymbol{x}|y=i})||\boldsymbol{w}_{mT-2}^{(k)} - \boldsymbol{w}_{mT-2}^{(c)}|| + \eta g_{max}(\boldsymbol{w}_{mT-2}^{(c)}) \sum_{i=1}^{C} ||p^{(k)}(y=i) - p(y=i)||.$$

$$(3)$$

Here, inequality 3 holds because

$$||\sum_{i=1}^{C} p^{(k)}(y=i)\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x},\boldsymbol{w}_{mT-2}^{(k)})] - \sum_{i=1}^{C} p(y=i)\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x},\boldsymbol{w}_{mT-2}^{(c)})]||$$

$$=||\sum_{i=1}^{C} p^{(k)}(y=i)\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x},\boldsymbol{w}_{mT-2}^{(k)})] - \sum_{i=1}^{C} p^{(k)}(y=i)\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x},\boldsymbol{w}_{mT-2}^{(c)})]+$$

$$\sum_{i=1}^{C} p^{(k)}(y=i)\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x},\boldsymbol{w}_{mT-2}^{(c)})] - \sum_{i=1}^{C} p(y=i)\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x},\boldsymbol{w}_{mT-2}^{(c)})]||$$

$$\leq||\sum_{i=1}^{C} p^{(k)}(y=i)(\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x},\boldsymbol{w}_{mT-2}^{(k)})] - \nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x},\boldsymbol{w}_{mT-2}^{(c)})])||$$

$$+ ||\sum_{i=1}^{C}(p^{(k)}(y=i) - p(y=i))\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}[\log f_i(\boldsymbol{x},\boldsymbol{w}_{mT-2}^{(c)})]||.$$

Inequality 4 holds because $g_{max}(\boldsymbol{w}_{mT-2}^{(c)}) = \max_{i=1}^{C}||\nabla_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{x}|y=i}\log f_i(\boldsymbol{x},\boldsymbol{w}_{mT-2}^{(c)})||$.

Based on Eq. (3), let $a^{(k)} = 1 + \eta\sum_{i=1}^{C} p^{(k)}(y=i)\lambda_{\boldsymbol{x}|y=i}$, by induction, we have

$$||\boldsymbol{w}_{mT-1}^{(k)} - \boldsymbol{w}_{mT-1}^{(c)}||$$

$$\leq a^{(k)}||\boldsymbol{w}_{mT-2}^{(k)} - \boldsymbol{w}_{mT-2}^{(c)}|| + \eta g_{max}(\boldsymbol{w}_{mT-2}^{(c)})\sum_{i=1}^{C}||p^{(k)}(y=i) - p(y=i)||$$

$$\leq (a^{(k)})^2||\boldsymbol{w}_{mT-3}^{(k)} - \boldsymbol{w}_{mT-3}^{(c)}|| + \eta\sum_{i=1}^{C}||p^{(k)}(y=i) - p(y=i)||(g_{max}(\boldsymbol{w}_{mT-2}^{(c)}) + a^{(k)}g_{max}(\boldsymbol{w}_{mT-3}^{(c)}))$$

$$\leq (a^{(k)})^{T-1}||\boldsymbol{w}_{(m-1)T}^{(k)} - \boldsymbol{w}_{(m-1)T}^{(c)}|| + \eta\sum_{i=1}^{C}||p^{(k)}(y=i) - p(y=i)||(\sum_{j=0}^{T-2}(a^{(k)})^j g_{max}(\boldsymbol{w}_{mT-2-j}^{(c)}))$$

$$= (a^{(k)})^{T-1}||\boldsymbol{w}_{(m-1)T}^{(f)} - \boldsymbol{w}_{(m-1)T}^{(c)}|| + \eta\sum_{i=1}^{C}||p^{(k)}(y=i) - p(y=i)||(\sum_{j=0}^{T-2}(a^{(k)})^j g_{max}(\boldsymbol{w}_{mT-2-j}^{(c)}))$$

Therefore,

$$||\boldsymbol{w}_{mT}^{(f)} - \boldsymbol{w}_{mT}^{(c)}||$$

$$\leq\sum_{k=1}^{K}\frac{n^{(k)}}{\sum_{k=1}^{K} n^{(k)}}((a^{(k)})^T||\boldsymbol{w}_{(m-1)T}^{(f)} - \boldsymbol{w}_{(m-1)T}^{(c)}||$$

$$+ \eta\sum_{i=1}^{C}||p^{(k)}(y=i) - p(y=i)||(\sum_{j=0}^{T-1}(a^{(k)})^j g_{max}(\boldsymbol{w}_{mT-1-k}^{(c)}))).$$

Hence proved.

$\square$