Federated Learning with Non-IID Data

Divya Sharvani Kandukuri

(50442906)

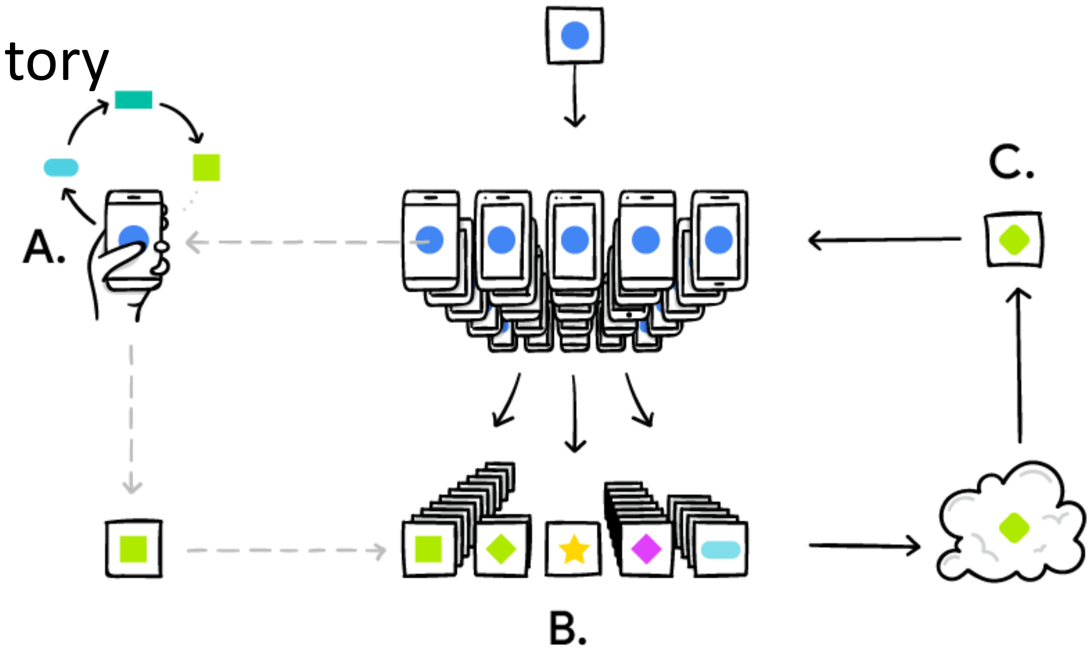
Contents

- Introduction
- FedAvg on Non- IID Data
- Weight Divergence due to Non – IID Data
- Proposed Solution
- Conclusion

Introduction

Introduction

- Federated Learning is an ML technique that trains across multiple decentralized edge devices.
- It provides privacy, security, regulatory and economic benefits.



Introduction

IID vs Non-IID

IID – Independent and Identically Distributed

- Each $x^{(i)} \sim \mathcal{D}$ (Identically Distributed)
- $\forall i \neq j \;\; p(x^{(i)}, x^{(j)}) = p(x^{(i)})p(x^{(j)})$ (Independently Distributed)

Non-IID Data:

- Data is processed in an insufficiently random order or ordered by collection of devices and/oo. (not independent

Introduction

Research in Federated Learning

- McMahan introduced the Federated Averaging (FedAvg) algorithm and demonstrated the robustness of FedAvg to train CNNs on benchmark image classification datasets, and LSTM on a language dataset.
- Two main challenges :
  - Communication cost
  - Statistical challenge
- In this paper, the authors show that accuracy of CNN trained with highly-skewed non-IID is significantly less. This happens because of weight divergence, and we use EMD to quantify it and propose a data-sharing strategy as a solution.

FedAvg on Non-IID Data

FedAvg on Non-IID Data

Experimental Setup

- Datasets used – MNIST, CIFAR-10 and Speech commands dataset
- MNIST, CIFAR-10 – image classification datasets, 10 classes
- Speech commands dataset – 35 words each of 1 sec duration
- For consistency, we use subset of data with 10 keywords – KWS dataset (keyword spotting)
- Training sets are divided equally among 10 clients.

FedAvg on Non-IID Data

Data distribution in different settings

- IID - each client is randomly assigned a uniform distribution over 10 classes
- Non- IID – data is sorted by class; we consider two extreme cases after sorting the data by class:
  - 1-class non-IID : each client receives data partition from one class
  - 2-class non-IID : sorted data is divided into 20 partitions, and each client gets 2 randomly assigned partitions of two classes

FedAvg on Non-IID Data

Parameters for training

- B – Batch size
- E – total number of epochs
- For SGD, we use the same parameters, but B is 10 times larger.

Parameters	MNIST	CIFAR-10	KWS
B	10 , 100	10 , 100	10 , 50
E	1 , 5	1 , 5	1 , 5
Learning rate (η)	0.01	0.01	0.05
Decay rate	0.995	0.992	0.992

Parameters for FedAvg

FedAvg on Non-IID Data

Experiment Results

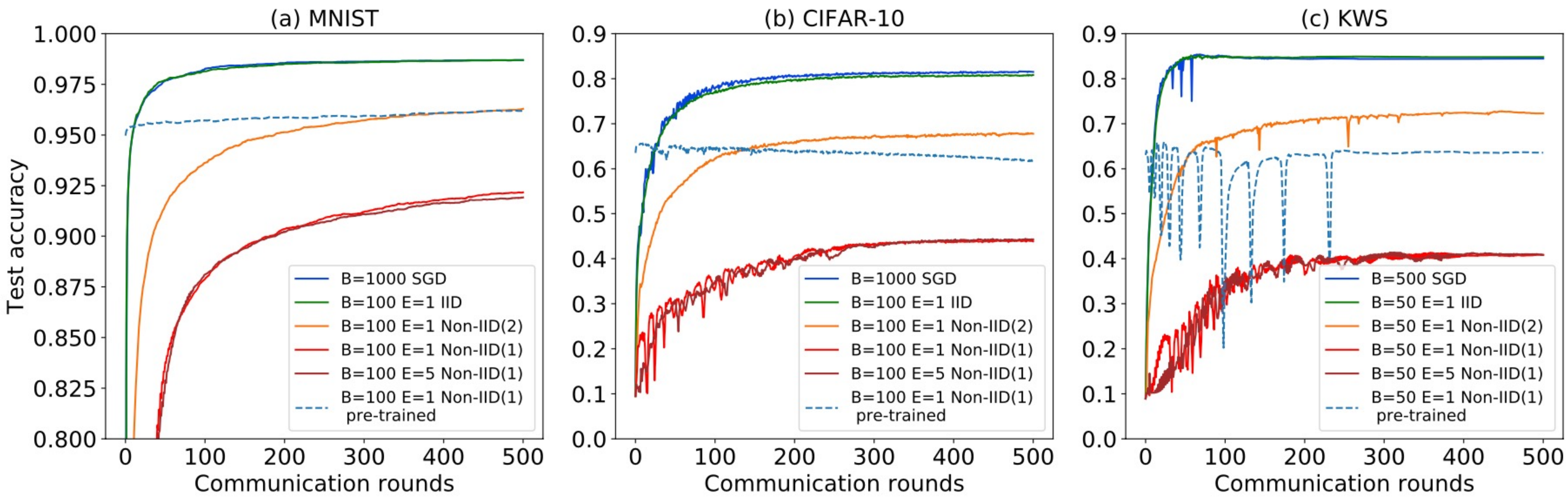


Figure 1: Test accuracy over communication rounds of *FedAvg* compared to SGD with IID and non-IID data of (a) MNIST (b) CIFAR-10 and (c) KWS datasets. Non-IID(2) represents the 2-class non-IID and non-IID(1) represents the 1-class non-IID.

FedAvg on Non-IID Data

Experiment Results

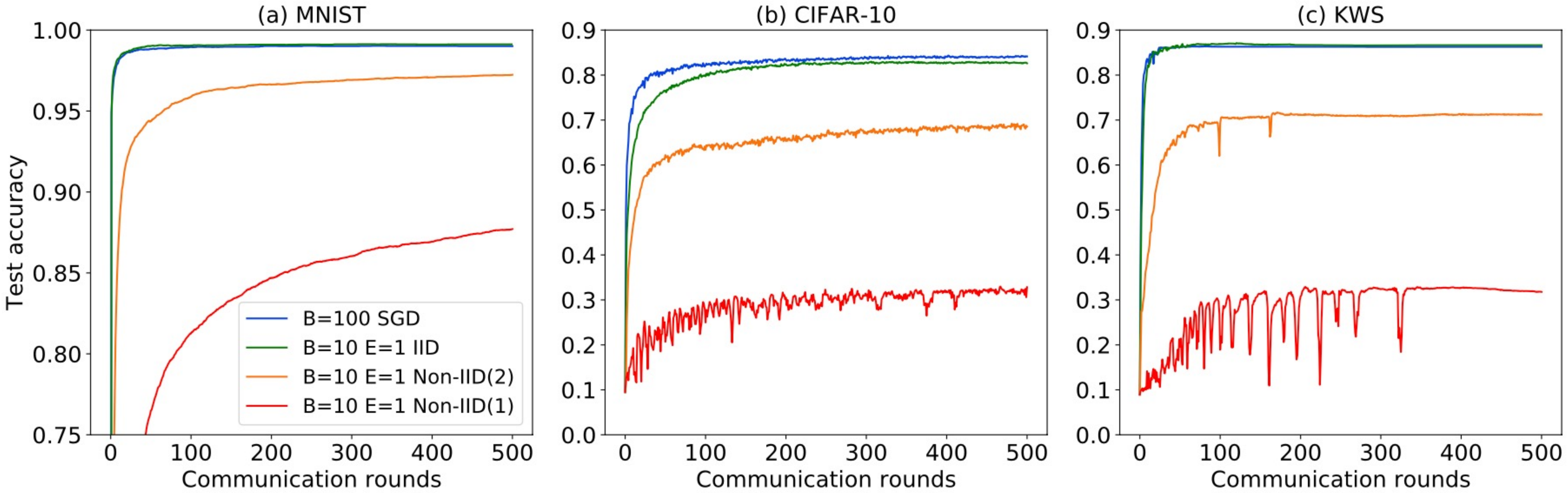


Figure 8: Test accuracy over communication rounds of *FedAvg* compared to SGD with IID and non-IID data of (a) MNIST (b) CIFAR-10 and (c) KWS datasets. Non-IID(2) represents the 2-class non-IID and non-IID(1) represents the 1-class non-IID.

FedAvg on Non-IID Data

Experiment Results

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

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

Weight Divergence due to Non-IID Data

Weight Divergence due to Non-IID Data

Weight Divergence

- Accuracy reduction is less for 2-class non-IID data than for 1-class non-IID data.
- Accuracy of FedAvg may be affected by exact data distribution.
- One way to compare FedAvg with SGD is to calculate difference of the weights relative to those of SGD, with same weight initialization.

$$weight\ divergence = ||\boldsymbol{w}^{FedAvg} - \boldsymbol{w}^{SGD}||/||\boldsymbol{w}^{SGD}||$$

- Root cause of the weight divergence is due to the distance between the data distribution on each client and the population distribution .

Weight Divergence due to Non-IID Data

Weight Divergence

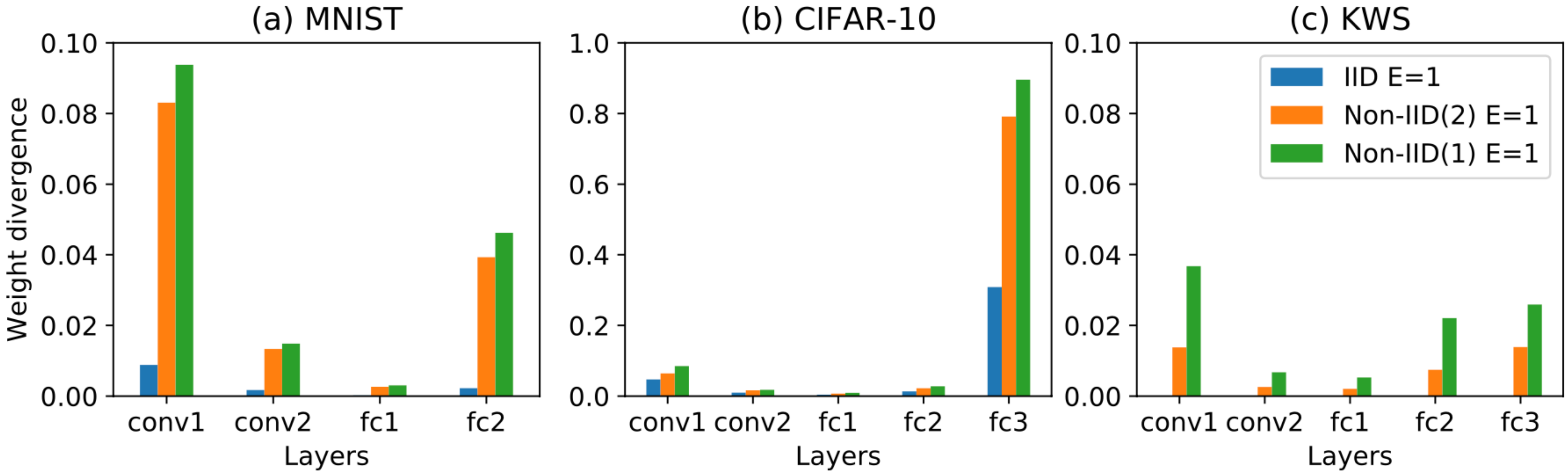


Figure 2: Weight divergence of CNN layers for IID, 2-class non-IID and 1-class non-IID.

Weight Divergence due to Non-IID Data

Mathematical Demonstration

C class classification problem

compact space $\mathcal{X}$

label space $\mathcal{Y} = [C]$, where $[C] = \{1, \ldots, C\}$

data point $\{\boldsymbol{x}, y\}$ distributes over $\mathcal{X} \times \mathcal{Y}$

distribution p

$f : \mathcal{X} \rightarrow \mathcal{S}$

$\mathcal{S} = \{\boldsymbol{z} | \sum_{i=1}^{C} z_i = 1, z_i \geq 0, \forall i \in [C]\}$

Weight Divergence due to Non-IID Data

Mathematical Demonstration

- Population loss is defined using cross entropy loss:

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

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

Weight Divergence due to Non-IID Data

Mathematical Demonstration

- Weight after t-th update in the centralized setting -- $\boldsymbol{w}_t^{(c)}$
- Centralized SGD performs following update:

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

- Federated learning – assuming there are k clients, $n^{(k)}$ amount of data, $p^{(k)}$ be data distribution on client $k \in [K]$
- At iteration t on client k ∈ [K], local SGD performs:

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

Weight Divergence due to Non-IID Data

Mathematical Demonstration

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

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

Weight Divergence due to Non-IID Data

Mathematical Demonstration

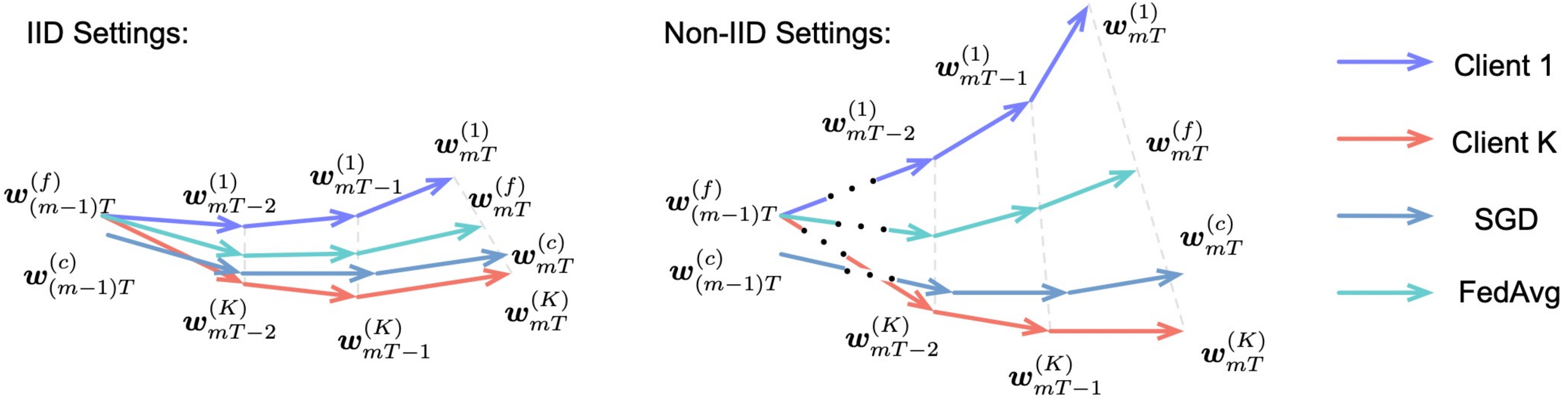


Figure 3: Illustration of the weight divergence for federated learning with IID and non-IID data.

Weight Divergence due to Non-IID Data

Proposition

To formally bound the weight divergence between $\boldsymbol{w}_{mT}^{(f)}$ and $\boldsymbol{w}_{mT}^{(c)}$ they proposed the following:

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,*

$$
\begin{aligned}
||\boldsymbol{w}_{mT}^{(f)} - \boldsymbol{w}_{mT}^{(c)}|| \leq & \sum_{k=1}^{K} \frac{n^{(k)}}{\sum_{k=1}^{K} n^{(k)}} (a^{(k)})^T ||\boldsymbol{w}_{(m-1)T}^{(f)} - \boldsymbol{w}_{(m-1)T}^{(c)}|| \\
& + \eta \sum_{k=1}^{K} \frac{n^{(k)}}{\sum_{k=1}^{K} n^{(k)}} \sum_{i=1}^{C} ||p^{(k)}(y=i) - p(y=i)|| \sum_{j=1}^{T-1} (a^{(k)})^j g_{max}(\boldsymbol{w}_{mT-1-k}^{(c)}),
\end{aligned} \tag{2}
$$

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* $a^{(k)} = 1 + \eta \sum_{i=1}^{C} p^{(k)}(y=i) \lambda_{\boldsymbol{x}|y=i}$.

Weight Divergence due to Non-IID Data

Remarks

1. Weight divergence after m-th synchronization comes from two parts:
   1. Weight divergence of (m-1) – th synchronization $||\boldsymbol{w}^{(f)}_{(m-1)T} - \boldsymbol{w}^{(c)}_{(m-1)T}||$
   2. Weight divergence induced by probability distance for data distribution on client k compared with the whole population distribution $\sum_{i=1}^{C} ||p^{(k)}(y=i) - p(y=i)||.$

2. Weight divergence after (m-1)th synchronization is amplified by $\sum_{k=1}^{K} \frac{n^{(k)}(a^{(k)})^T}{\sum_{k=1}^{K} n^{(k)}}$

As $a^{(k)}$ >=1, $\sum_{k=1}^{K} \frac{n^{(k)}(a^{(k)})^T}{\sum_{k=1}^{K} n^{(k)}}$ = 1

3. EMD between data distribution on client k and the population distribution =

$$\sum_{i=1}^{C} ||p^{(k)}(y=i) - p(y=i)||$$

It is affected by learning rate, number of steps and gradient $g_{max}(\boldsymbol{w}^{(c)}_{mT-1-k})$

Weight Divergence due to Non-IID Data

<u>Experimental Validation</u>

- Setup:
  - Training set is sorted and partitioned into 10 clients – M examples per client
  - 8 values are chosen for EMD. As there may be many distributions for one EMD, we will generate 5 distributions.
  - Procedure:
    - 1. P – one probability distribution over 10 classes is generated for one EMD. Number of examples can be computed based on M and P values over 10 classes for one client.
    - 2. P' – shift the 10 probabilities of P by 1 element.
  - Repeat the above procedure for remaining 8 clients.
  - We will have 10 clients with distribution of M examples over 10 classes.
  - Above procedure is repeated 5 times to generate 5 distributions for each EMD.

Weight Divergence due to Non-IID Data

<u>Experimental Validation</u>

- weight divergence is computed after 1 synchronization

Key Parameters	MNIST	CIFAR-10	KWS
B	100	100	50
E	1	1	1
Learning rate (η)	0.01	0.01	0.05
Decay rate	0.995	0.992	0.992

$$weight\ divergence = ||\boldsymbol{w}^{FedAvg} - \boldsymbol{w}^{SGD}||/||\boldsymbol{w}^{SGD}||$$

Weight Divergence due to Non-IID Data

Weight Divergence vs EMD

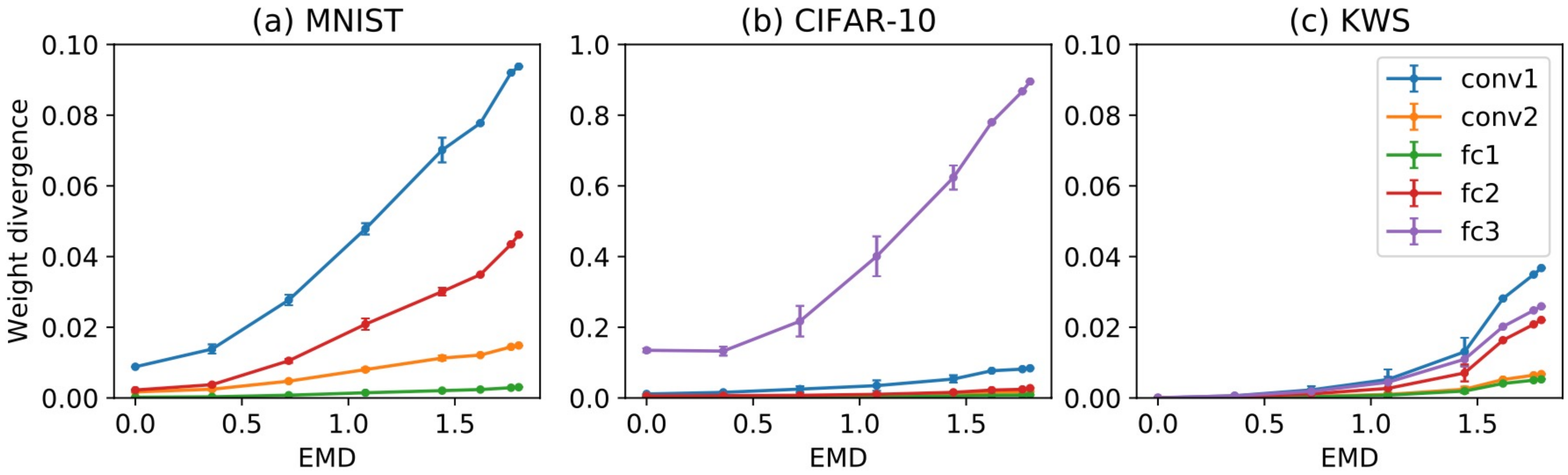


Figure 4: Weight divergence vs. EMD across CNN layers on (a) MNIST, (b) CIFAR-10 and (c) KWS datasets. The mean value and standard deviation are computed over 5 distributions for each EMD.

Weight Divergence due to Non-IID Data

Test Accuracy vs EMD

- Test accuracy decreases with EMD

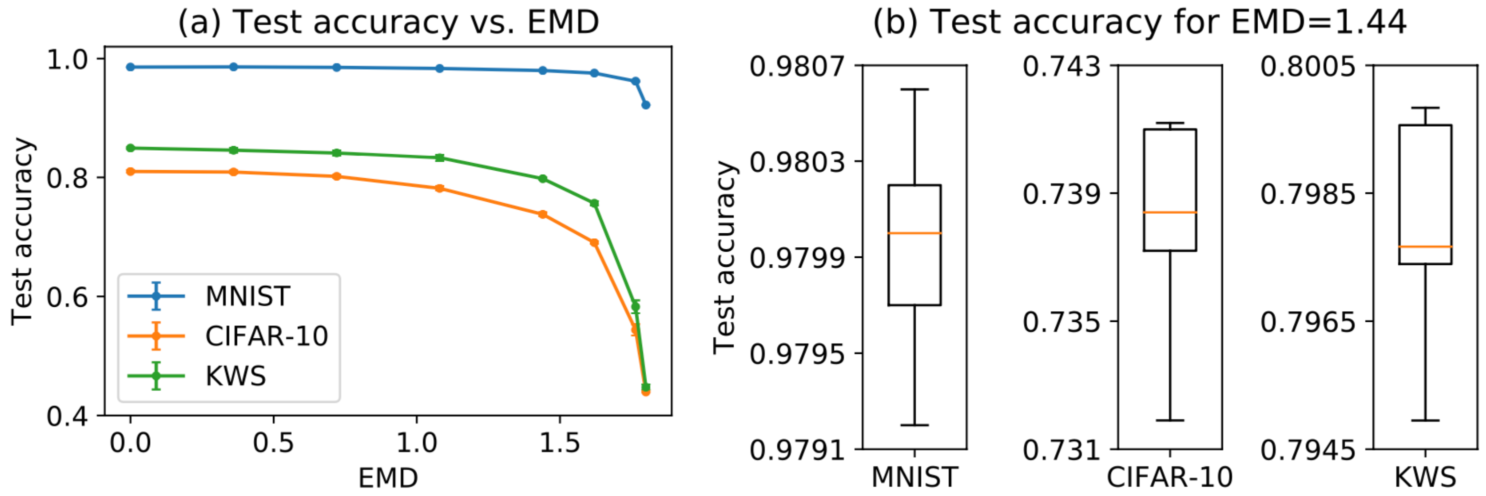


Figure 5: (a) Test accuracy vs. EMD for $FedAvg$ and (b) boxplots of weight divergence when EMD = 1.44 for MNIST, CIFAR-10 and KWS datasets. The mean and standard deviation are computed over 5 distributions for each EMD.

Weight Divergence due to Non-IID Data

Test Accuracy vs 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

Proposed Solution

Proposed Solution

Motivation

- Test accuracy decreases with respect to EMD beyond a certain threshold.
- To increase the test accuracy, we have to reduce the EMD.
- We can do that by distributing a small subset of global data containing a uniform distribution over classes from cloud to the clients.
- We can also make a warm-up model train on globally shared data.
- As globally shared data can reduce EMD, the test accuracy is expected to improve.

Proposed Solution

Data Sharing Strategy

- G – globally shared dataset
- α – random portion of G distributed to client
- During initialization, warm-up model trained on G and α portion of G are distributed.
- The local model is trained on part of G shared and private data of client.
- The cloud aggregates the local models using FedAvg

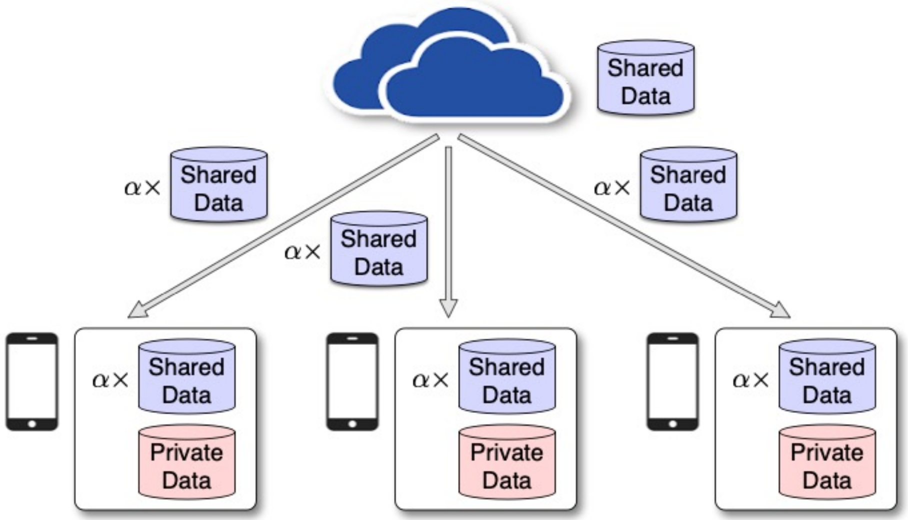


Figure 6: Illustration of the data-sharing strategy.

Proposed Solution

Data Sharing Strategy

- Two tradeoffs:
  - Trade-off between test accuracy and size of G:

$$\beta = \frac{||G||}{||D||} \times 100\%$$, where D- data from client

  - Trade-off between test accuracy and α

Proposed Solution

Experiment

- 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 and H is used to create 10 random G's with β ranging from 2.5% to 25%.

Procedure:

1. G is merged with data of the each client and 10 CNNs are trained by FedAvg on the merged data from scratch

2. Pick two specific G's:

G10% when β = 10% and

G20% when β = 20%

3. 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.

Proposed Solution

Experiment

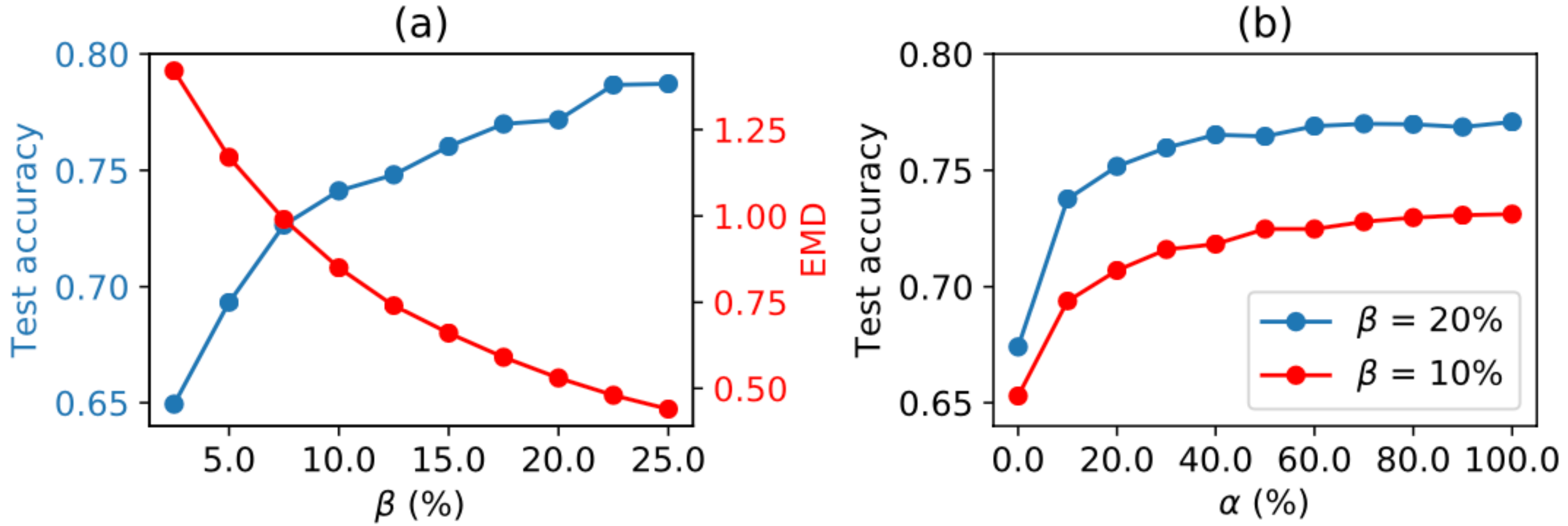


Figure 7: (a) Test accuracy and EMD vs. β (b) Test accuracy vs. the distributed fraction α

Conclusion

Conclusion

- Federated learning will play a key role in distributed machine learning where data privacy is of paramount importance.
- The quality of model training degrades if each of the edge devices sees a unique distribution of data – non IID.
- The accuracy of federated learning reduces significantly, by up to ~55% for NN trained on highly skewed non-IID data.
- Accuracy reduction can be explained by the weight divergence, which can be quantified by the earth movers distance (EMD)
- Strategy to improve training on non-IID data by creating a small subset of data which is globally shared between all the edge devices.
- Improving model training on non-IID data is key to make progress in this area.

Thank you!