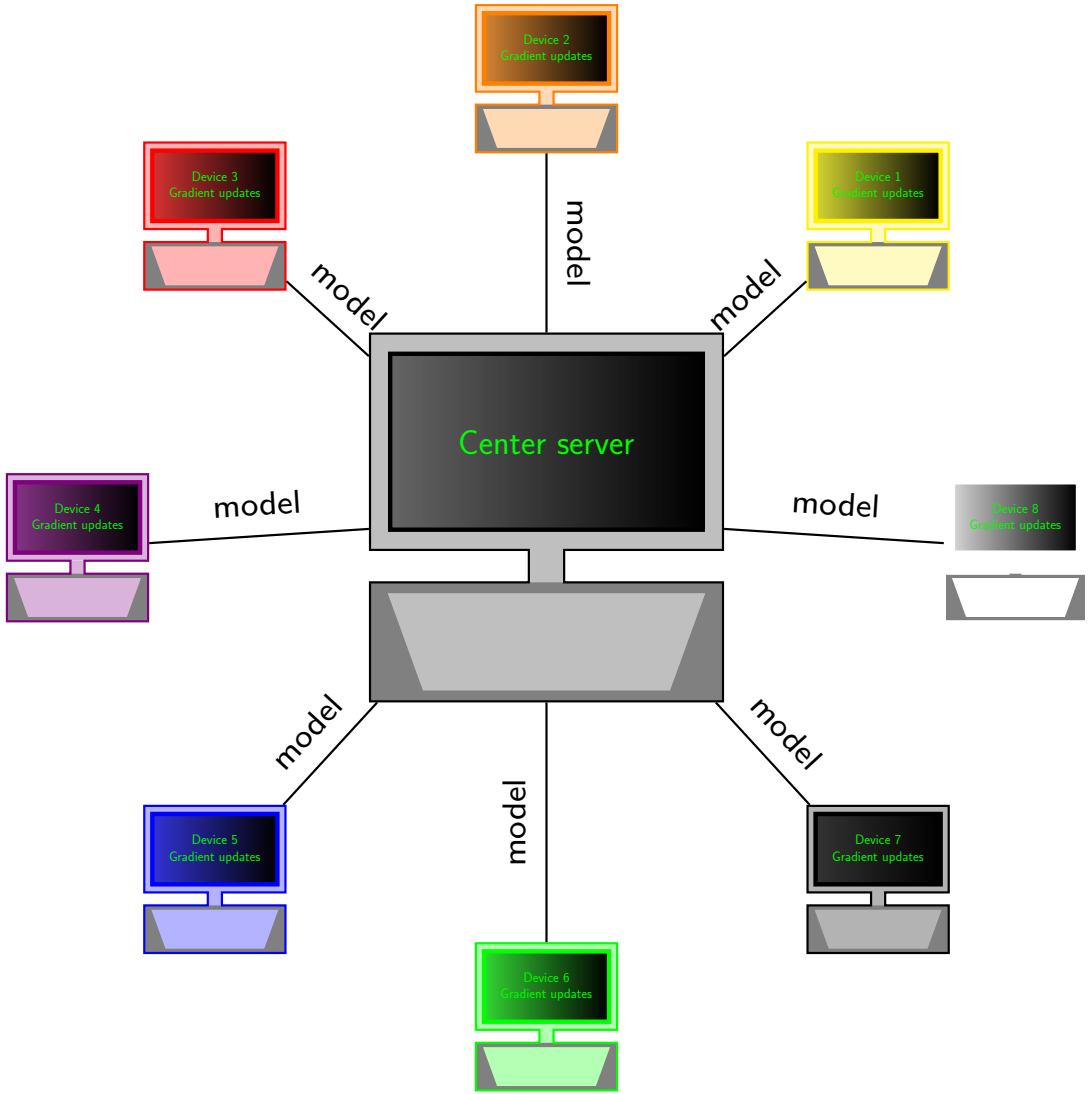
Federated Learning

Hoang Trung Hieu

2020

What is the fererated learning?



Challenges in fererated learning

- *Massively distributed*. The number of mobile device owners is massively bigger than average of the number of training samples on each device.
- *Unbalanced*. Some users produce significantly more data than others.
- *Non-IID*. The data generated by each user are quite different.

Problem description

Let us consider the following distributed optimization model in which N clients cooperatively solve

$$\min_w f(w) = \min_w \sum_{k=1}^{N} p_k f_k(w)$$

where p_k be the weight of $k-$th device , $p_k \geq 0, \quad \sum_{i=1}^{N} p_k = 1$.
The $k-$th client has n_k training data which is denoted $x_{k,1}, \cdots, x_{k,n_k}$. The local generalization error function are defined by

$$f_k(w) = \frac{1}{n_k} \sum_{j=1}^{n_k} \ell(w; x_{k,j})$$

where ℓ is a user loss function of the prediction.

Federated Averaging Algorithms

Algorithm 1: FederatedAveraging(FedAvg).

Server executes: Initialization w_0 ;

for *each round* $t = 1, 2, \cdots$ **do**

  $m \leftarrow \max(C \cdot K, 1)$

  $S_t \leftarrow$ (random set of m client)

  for *each client* $k \in S_t$ **do**

    $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$;

  end

  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$;

end

ClientUpdate(k, w)**:**

$B \leftarrow$ (split P_k into batches of size B)

for *each local epoch* **do**

  for *batch* $b \in B$ **do**

    $w \leftarrow w - \eta \nabla l(w; b)$

  end

Experiments-Input

Real data: MNIST

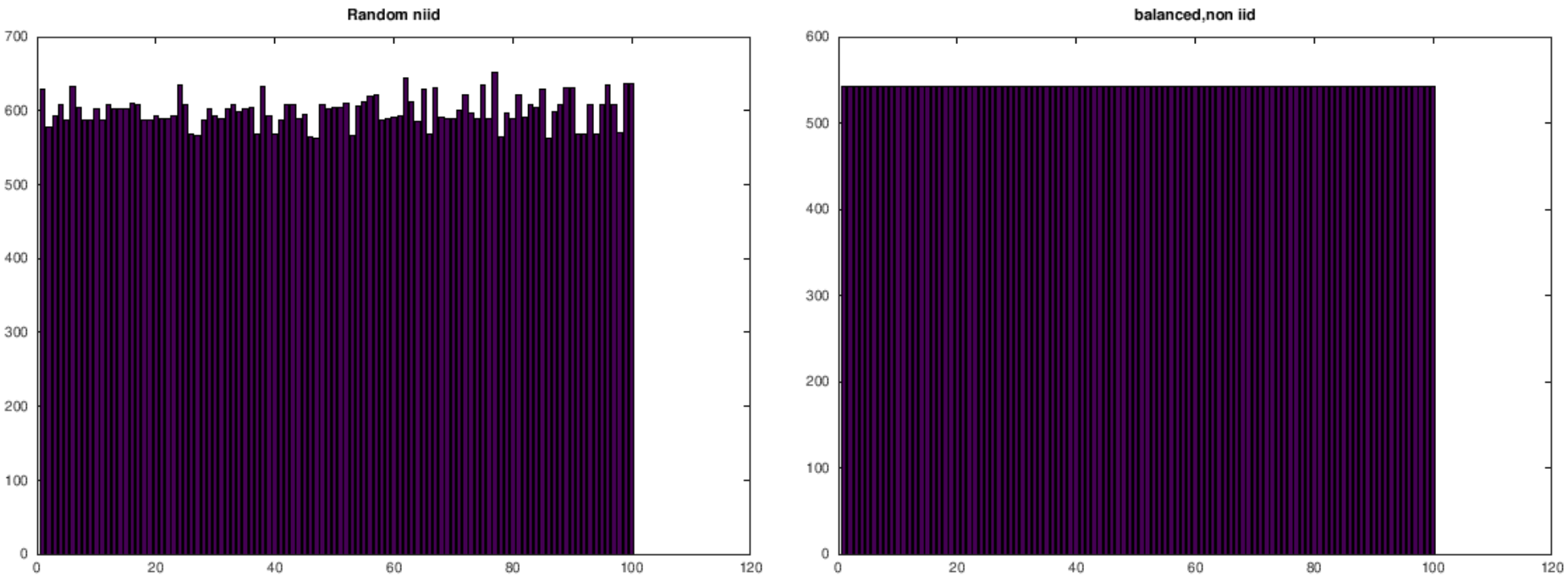
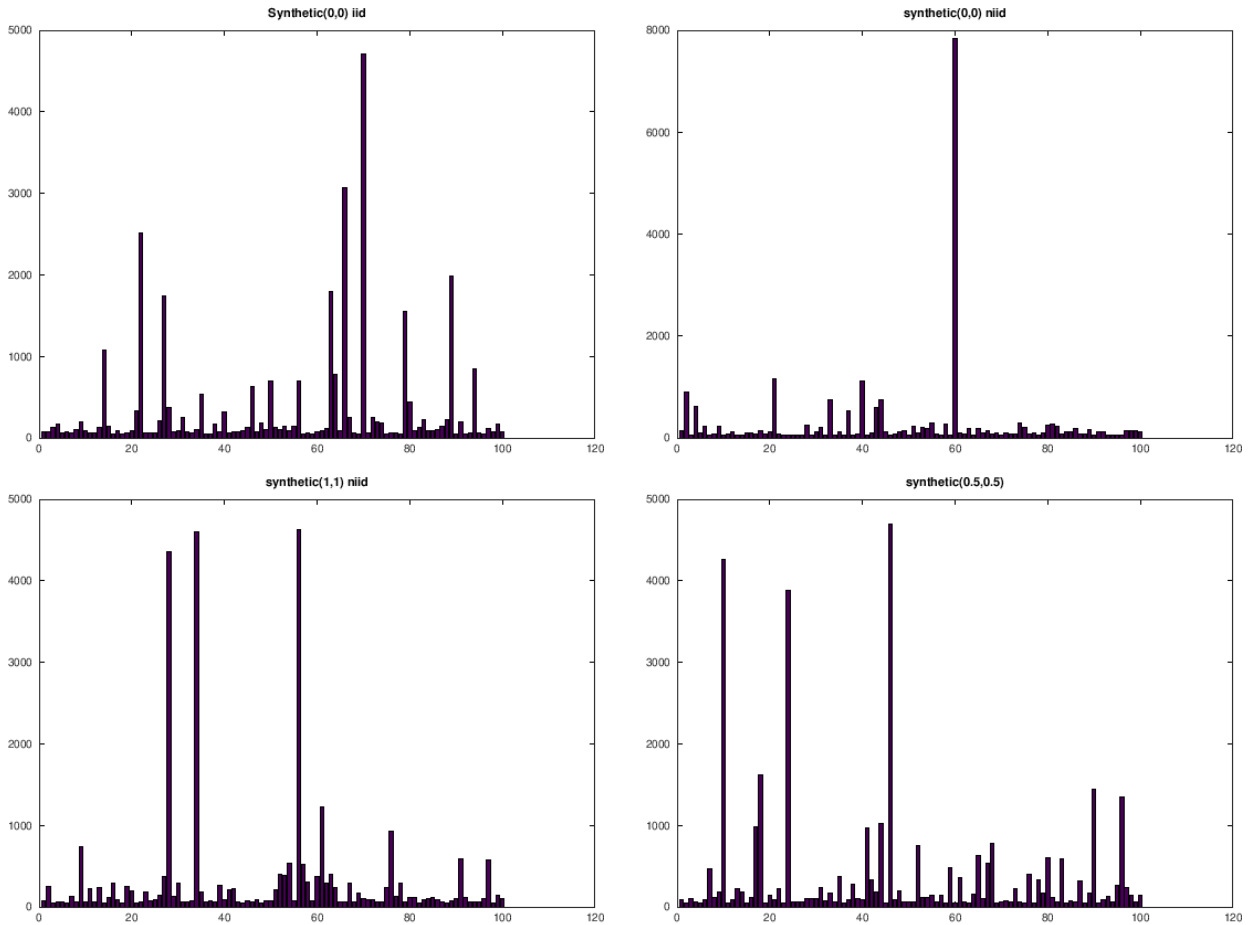


Figure. Unbalanced (left) and balanced (right) data distribution

Experiments-Input



Figure[1.2]. Synthetic data with $(\alpha, \beta) = (0, 0)$-iid ,$(\alpha, \beta) = (0, 0)$-niid,$(\alpha, \beta) = (1, 1)$-niid,$(\alpha, \beta) = (0.5, 0.5)$-niid

Experiments-Models

2NN

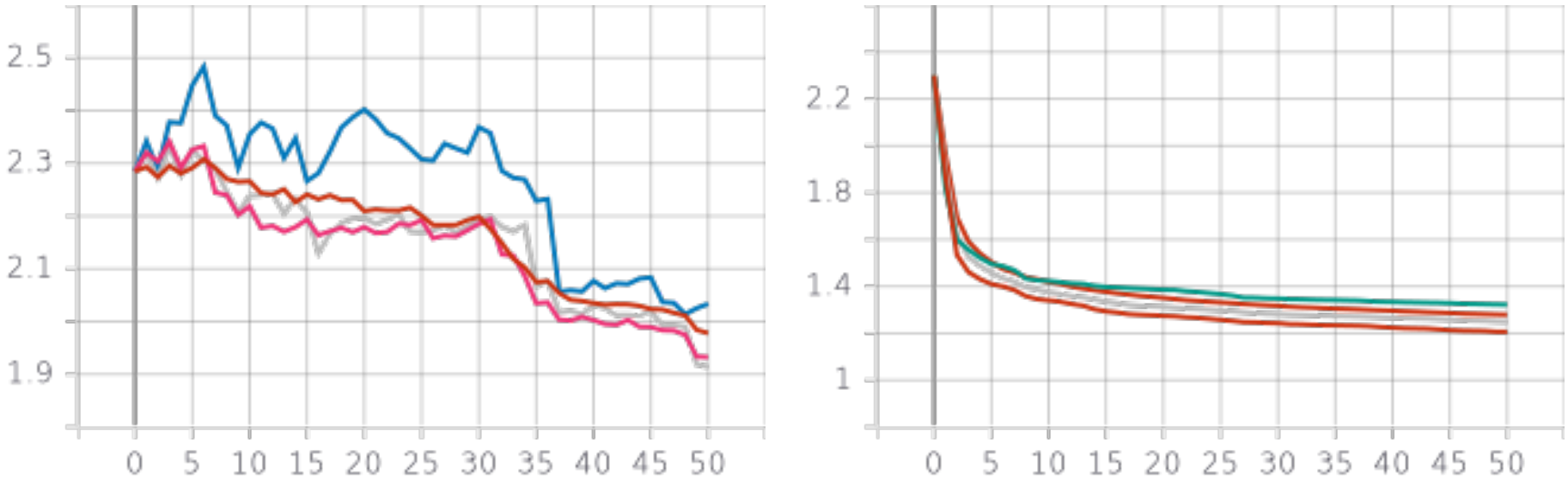
(fc1): Linear(input features=784, output features=200, bias=True)
(fc2): Linear(input features=200, output features=200, bias=True)
(fc3): Linear(input features=200, output features=10, bias=True)

CNN

(conv1): Conv2d(1, 32, kernel size=(5, 5), stride=(1, 1))
(conv2): Conv2d(32, 64, kernel size=(5, 5), stride=(1, 1))
(fc 1): Linear(input features=1024, output features=512, bias=True)
(fc 2): Linear(input features=512, output features=10, bias=True)

Experiments-Results

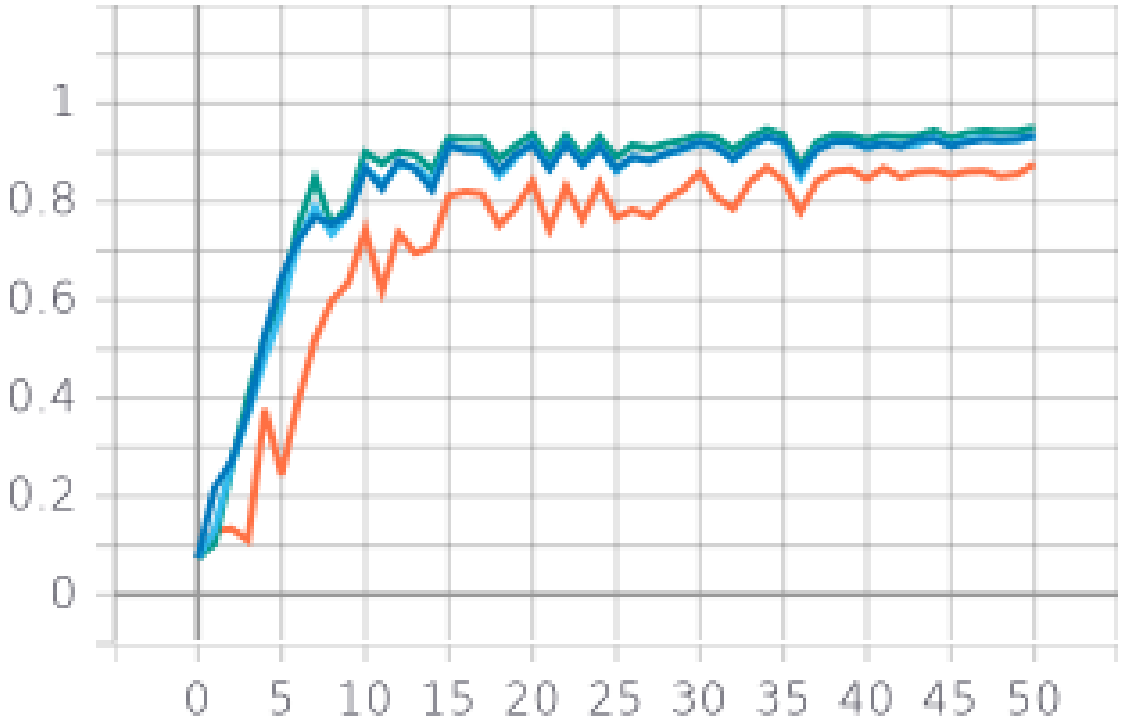
Synthetic(0,0) iid (right) and non-iid (left) dataset, $B = 64$, $T = 50$, $lr = 0.01$, $E = 5$, model is 2NN. The value of C in the set $\{5/100, 10/100, 20/100, 50/100\}$. The horizontal axis is the train loss.



Figure[3.1]. Impact of C

Experiments-Results

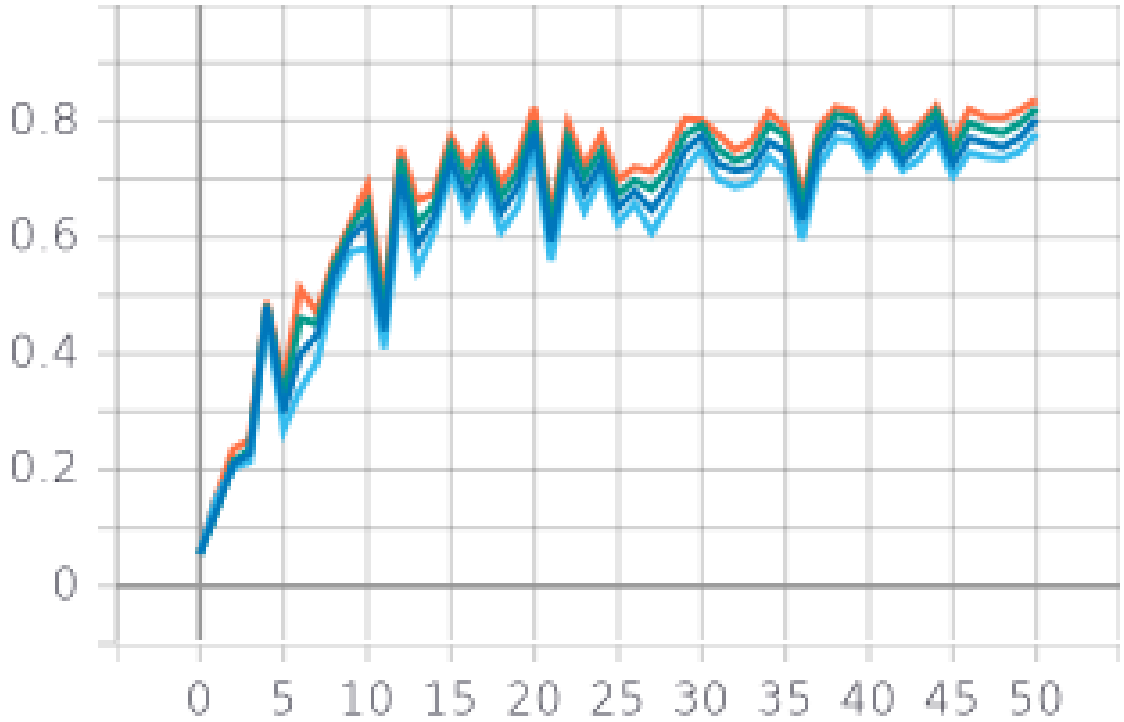
MNIST unbalanced non-iid dataset,
$C = 5/100, B = 64, T = 50, lr = 0.01$, model is CNN. The value of E in the set $\{2, 5, 10, 20\}$. The horizontal axis is the train accuracy.



Figure[3.2]. Impact of E

Experiments-Results

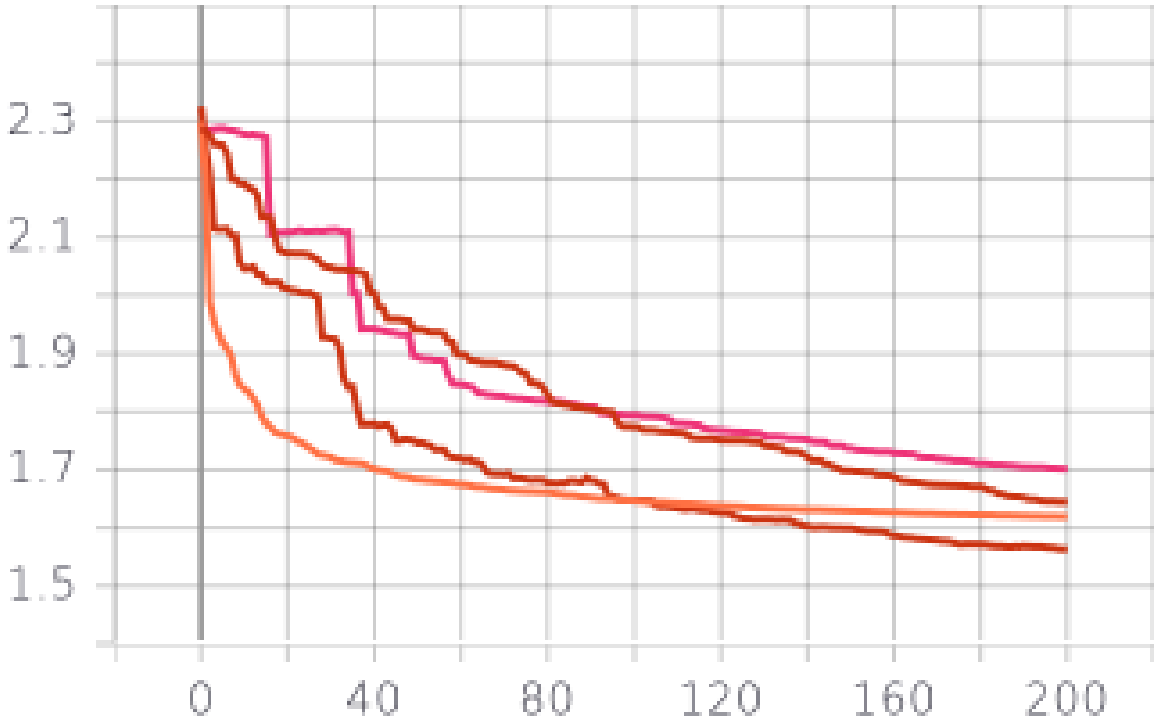
To evaluate the impact of batch size, we examine on MNIST balanced non-iid dataset, $C = 10/100$, $T = 50$, $lr = 0.01$, model is 2NN. The value of B in the set $\{16, 32, 64, 128\}$. The horizontal axis is the test accuracy.



Figure[3.3]. Impact of B

Experiments-Results

Impact of balancedness. Synthetic data with $(\alpha, \beta) = (0, 0)$-iid ,$(\alpha, \beta) = (0, 0)$-niid,$(\alpha, \beta) = (1, 1)$-niid,$(\alpha, \beta) = (0.5, 0.5)$-niid $C = 10/100, B = 64, T = 50, lr = 0.01, E = 5$, model is CNN. The horizontal axis is the test loss.



Figure[3.4]. Impact of unbalancedness

THANK YOU FOR YOUR ATTENTION!