FedBN: Federated Learning on Non-IID Features via Local Batch Normalization

Anonymous authors
Paper under double-blind review

Abstract

The emerging paradigm of federated learning (FL) strives to enable collaborative training of deep models on the network edge without centrally aggregating raw data and hence improving data privacy. In most cases, the assumption of independent and identically distributed samples across local clients does not hold for federated learning setups. Under this setting, neural network training performance may vary significantly according to the data distribution and even hurt training convergence. Most of the previous work has focused on a difference in the distribution of labels. Unlike those settings, we address an important problem of FL, e.g., different scanner/sensors in medical imaging, different scenery distribution in autonomous driving (highway vs. city), where local clients store examples with different marginal or conditional feature distributions compared to other nodes, which we denote as *feature shift non-iid*. In this work, we propose an effective method that uses local batch normalization to alleviate the feature shift before averaging models. The resulting scheme, called FedBN, outperforms both classical FedAvg, as well as the state-of-the-art for non-iid data (FedProx) on our extensive experiments. These empirical results are supported by a convergence analysis that shows in a simplified setting that FedBN has a faster convergence rate in expectation than FedAvg.

1 Introduction

Federated learning (FL), has gained popularity for various applications involving learning from distributed data. In FL, a cloud server (the "server") can communicate with distributed data sources (the "clients"), while the clients hold data separately. A major challenge in FL is the training data statistical heterogeneity among the clients (Kairouz et al., 2019; Li et al., 2020b). It has been shown that standard federated methods such as FedAvg (McMahan et al., 2017) which are not designed particularly taking care of non-iid data significantly suffer from performance degradation or even diverge if deployed over non-iid samples (Karimireddy et al., 2019; Li et al., 2018; 2020a).

Recent studies have attempted to address the problem of FL on non-iid data. Most variants of FedAvg primarily tackle the issues of stability, client drift and heterogeneous label distribution over clients (Li et al., 2020b; Karimireddy et al., 2019; Zhao et al., 2018). Instead, we focus on the shift in the feature space, which has not yet been explored in the literature. Specifically, we consider that local data deviates in terms of the distribution in feature space, and identify this scenario as *feature shift*. This type of non-iid data is a critical problem in many real-world scenarios, typically in cases where the local devices are responisble for a heterogeneity in the feature distributions. For example in cancer diagnosis tasks, medical radiology images collected in different hospitals have uniformly distributed labels (i.e., the cancer types treated are quite similar for all hospitals). However, the image appearance can vary a lot due to different imaging machines and protocols used in hospitals, e.g., different intensity and contrast. In this example, each hospital is a client and hospitals aim to collaboratively train a cancer detection model without sharing privacy-sensitive data.

Tackling non-iid data with feature shift has been explored in classical centralized training in the context of domain adaptation. Here, an effective approach in practice is utilizing

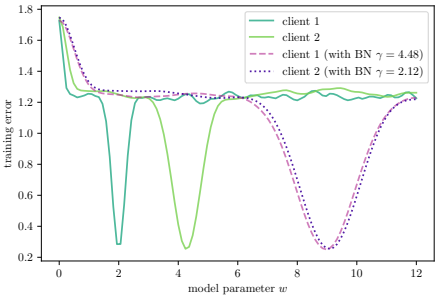

Figure 1: Training error on local datasets for two clients respectively with and w/o BN, observing BN harmonizes the loss surface.

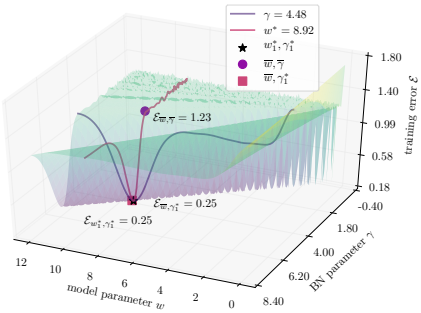

Figure 2: Error surface of a client for model parameter $w \in [0.001, 12]$ and BN parameter $\gamma \in [0.001, 4]$. Averaging model *and* BN parameters leads to worse solutions.

Batch Normalization (BN) (Ioffe & Szegedy, 2015): recent work has proposed BN as a tool to mitigate domain shifts in domain adaptation tasks with promising results achieved (Li et al., 2016; Liu et al., 2020; Chang et al., 2019). Inspired by this, this paper proposes to apply BN for feature shift FL. To illustrate the idea, we present a toy example that illustrates how BN may help harmonizing local feature distributions.

Observation of BN in a FL Toy Example: We consider a simple non-convex learning problem: we generate data $x, y \in \mathbb{R}^2$ with $y = \cos(w_{true}x) + \epsilon$, where $x \in \mathbb{R}$ is drawn iid from a Gaussian distribution and ϵ is zero-mean Gaussian noise and consider models of the form $f_w(x) = \cos(wx)$ with model parameter $w \in \mathbb{R}$. Local data deviates in the variance of x. First, we illustrate that local batch normalization harmonizes local data distributions. We consider a simplified form of BN that normalizes the input by scaling it with γ, i.e., the local empirical standard deviation, and a setting with 2 clients. As Fig. 1 shows the local squared loss between is very different between the two clients. Thus, averaging the model, does not lead to a good model. However, when applying local BN, the local training error surfaces become similar and averaging the models can be beneficial. To further illustrate the impact of BN, we plot the error surface for one client with respect to both model parameters $w \in \mathbb{R}$ and BN parameters $\gamma \in \mathbb{R}$ in Fig. 2. The figure shows that for an optimal weight w_1^*, changing γ quickly deteriorates the model quality. Similarly, for a given optimal BN parameter γ_1^*, changing w quickly deteriorates the quality. In particular, the average model $\overline{w} = (w_1^* + w_2^*)/2$ and average BN parameters $\overline{\gamma} = (\gamma_1^* + \gamma_2^*)/2$ has a high error. At the same time, the average model $\overline{w}$ with local BN parameter γ_1^* performs very well.

Motivated by the above insight and observation, this paper proposes a novel federated learning method, called FedBN, for addressing non-iid training data which keeps the client BN layers updated locally, without communicating, and aggregating them at the server. In practice, we can simply update the non-BN layers using FedAvg, without modifying any optimization or aggregation scheme. This approach has zero parameters to tune, requires minimal additional computational resources, and can be easily applied to arbitrary neural network architectures with BN layers in FL. Besides the benefit shown in the toy example, we also show the benefits in accelerating convergence by theoretically analyzing the convergence of FedBN in the over-parameterized regime. In addition, we have conducted extensive experiments on a benchmark and three real-world datasets. Compared to classical FedAvg, as well as the state-of-the-art for non-iid data (FedProx), our novel method, FedBN, demonstrates significantly practical improvements on the extensive experiments.

2 Related Work

Techniques for Non-IID Challenges in Federated Learning: Widely known aggregation strategy in FL, FedAvg (McMahan et al., 2017) it often suffers when data is heterogeneous over local client. Empirical work addressing non-iid issues, mainly focus on label distribution skew, where a non-iid dataset is formed by partitioning a "flat" existing

dataset based on the labels. FedProx (Li et al., 2020b), a recent framework tackled the heterogeneity by allowing partial information aggregation and adding a proximal term to FedAvg. Zhao et al. (2018) assumed a subset of the data is globally shared between all the clients, hence generalizes to the problem at hand. FedMA (Wang et al., 2020) proposed an aggregation strategy for non-iid data partition that shares global model in a layer-wise manner. However, there are so far limited attempts considering non-iid induced from feature shift, which is common in medical data collecting from different equipment and natural image collected in various noisy environment. Very recently, FedRobust (Reisizadeh et al., 2020) assumes data follows an affine distribution shift and tackles this problem by learning the affine transformation. This hampers the generalization when we cannot estimate the explicit affine transformation.

Batch Normalization in Deep Neural Networks: Batch Normalization Ioffe & Szegedy (2015) is an indispensable component in many deep neural networks and has shown its success in neural network training. Relevant literature has uncovered a number of benefits given by batch normalization. Santurkar et al. (2018) showed that BN makes the optimization landscape significantly smoother. Luo et al. (2018) investigated an explicit regularization form of BN such that improving the robustness of optimization. Morcos et al. (2018) suggested that BN implicitly discourages single direction reliance, thus improving model generalizability. Li et al. (2018) took advantage of BN for tackling the domain adaptation problem. However, what a role BN is playing in the scope of federated learning, especially for non-iid training, still remains unexplored to date.

3 Preliminary

Non-IID Data in Federated Learning: We introduce the concept of *feature shift* in federated learning as a novel category of client's non-iid data distribution. So far, the categories of non-iid data considered according to Kairouz et al. (2019); Hsieh et al. (2019) can be described by the joint probability between features $\mathbf{x}$ and labels y on each client. We can rewrite $P_i(\mathbf{x}, y)$ as $P_i(y|\mathbf{x})P_i(\mathbf{x})$ and $P_i(\mathbf{x}|y)P_i(y)$. We define *feature shift* as the case that covers: 1) *covariate shift*: the marginal distributions $P_i(\mathbf{x})$ varies across clients, even if $P_i(y|\mathbf{x})$ is the same for all client; and 2) *concept shift*: the conditional distribution $P_i(\mathbf{x}|y)$ varies across clients and $P(y)$ is the same.

Federated Averaging (FedAvg): We establish our algorithm on FedAvg introduced by McMahan et al. (2017) which is the most popular existing and easiest to implement federated learning strategy, where clients collaboratively send updates of locally trained models to a global server. Each client runs a local copy of the global model on its local data. The global model's weights are then updated with an average of local clients' updates and deployed back to the clients. This builds upon previous distributed learning work by not only supplying local models but also performing training locally on each device. Hence FedAvg potentially empowers clients (especially clients with small dataset) to collaboratively learn a shared prediction model while keeping all training data locally. Although FedAvg has shown successes in classical Federated Learning tasks, it suffers from slow convergence and low accuracy in most non-iid contents Li et al. (2020b; 2019).

4 Federated Averaging with Local Batch Normalization

Proposed Method - FedBN: We propose an efficient and effective learning strategy denoted *FedBN*. Similar to FedAvg, FedBN performs local updates and averages local models. However, FedBN assumes local models have BN layers and excludes their parameters from the averaging step. We present the full algorithm in Appendix C. This simple modification results in significant empirical improvements in non-iid settings. We provide an explanation for these improvements in a simplified scenario, in which we show that FedBN improves the convergence rate under feature shift.

Problem Setup: We assume $N \in \mathbb{N}$ clients to jointly train for $T \in \mathbb{N}$ epochs and to communicate after $E \in \mathbb{N}$ local iterations. Thus, the system has T/E communication rounds over the T epochs. For simplicity, we assume all clients to have $M \in \mathbb{N}$ training examples

(a difference in training examples can be account for by weighted averaging (McMahan et al., 2017)) for a regression task, i.e., each client $i \in [N]$ ($[N] = \{1, \ldots, N\}$) has training examples $\{(\mathbf{x}_j^i, y_j^i) \in \mathbb{R}^d \times \mathbb{R} : j \in [M]\}$. Furthermore, we assume a two-layer neural network with ReLU activations. Let $\mathbf{v}_k \in \mathbb{R}^d$ denote the parameters of the first layer, where $k \in [m]$ and m is the width of the hidden layer. Let $\| \mathbf{v} \|_{\mathbf{S}} \triangleq \sqrt{\mathbf{v}^\top \mathbf{S} \mathbf{v}}$ denote the induced vector norm for a positive definite matrix $\mathbf{S}$.

We assume that the feature shift on local datasets is captured by a difference in local covariances. To be more precise, we make the following assumption.

Assumption 4.1 (Data Distribution)**.** *For each client $i \in [N]$ the inputs $\mathbf{x}_j^i$ are centered ($\mathbb{E}\mathbf{x}^i = \mathbf{0}$) with covariance matrix $\mathbf{S}_i = \mathbb{E}\mathbf{x}^i \mathbf{x}^{i\top}$, where $\mathbf{S}_i$ is independent from the label $\mathbf{y}$ and may differ for each $i \in [N]$, and for each index pair $p \neq q$, $\mathbf{x}_p \neq \kappa \cdot \mathbf{x}_q$ for all $\kappa \in \mathbb{R} \setminus \{0\}$.*

With Assumption 4.1, the normalization of the first layer for client i is $\frac{\mathbf{v}_k^\top \mathbf{x}^i}{\|\mathbf{v}_k\|_{\mathbf{S}_i}}$. FedBN with client-specified BN parameters trains a model $f^* : \mathbb{R}^d \to \mathbb{R}$ parameterized by $(\mathbf{V}, \boldsymbol{\gamma}, \mathbf{c}) \in \mathbb{R}^{m \times d} \times \mathbb{R}^{m \times N} \times \mathbb{R}^m$, i.e.,

$$f^*(\mathbf{x}; \mathbf{V}, \boldsymbol{\gamma}, \mathbf{c}) = \frac{1}{\sqrt{m}} \sum_{k=1}^{m} c_k \sum_{i=1}^{N} \sigma\left(\gamma_{k,i} \cdot \frac{\mathbf{v}_k^\top \mathbf{x}}{\| \mathbf{v}_k \|_{\mathbf{S}_i}}\right) \cdot \mathbb{1}\{\mathbf{x} \in \text{client } i\} \ , \tag{1}$$

where $\boldsymbol{\gamma}$ is the scaling parameter of BN and $\sigma(s) = \max\{s, 0\}$ is the ReLU activation function. Here, we omit learning the shift parameter of BN [1]. FedAvg instead trains a function $f : \mathbb{R}^d \to \mathbb{R}$ which is a special case of Eq. 1 with $\gamma_{k,i} = \gamma_k$ for $\forall i \in [N]$. We take a random initialization of the parameters (Salimans & Kingma, 2016) in our analysis:

$$\mathbf{v}_k(0) \sim N\left(0, \alpha^2 \mathbf{I}\right), \quad c_k \sim U[-1, 1], \quad \text{and} \quad \gamma_k = \gamma_{k,i} = \|\mathbf{v}_k(0)\|_2 / \alpha, \tag{2}$$

where α^2 controls the magnitude of $\mathbf{v}_k$ at initialization. The initialization of the BN parameters γ_k and $\gamma_{k,i}$ are independent of α. The parameters of the network $f^*(\mathbf{x}; \mathbf{V}, \boldsymbol{\gamma}, \mathbf{c})$ are obtained by minimizing the empirical risk with respect to the squared loss:

$$L(f^*) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(f^*(\mathbf{x}_j^i) - y_j^i\right)^2 \ . \tag{3}$$

Convergence Analysis: Here we study the trajectory of networks FedAvg (f) and FedBN (f^*)'s prediction through the neural tangent kernel (NTK) introduced by Jacot et al. (2018). Recent machine learning theory studies (Arora et al., 2019; Du et al., 2018; Dukler et al., 2020) have shown that for finite-width over-parameterized networks, the convergence rate is controlled by the least eigenvalue of the induced kernel in the training evolution.

To simplify tracing the optimization dynamics, we consider the case that the number of local updates E is 1. We can decompose the NTK into a magnitude component $\mathbf{G}(t)$ and direction component $\mathbf{V}(t)/\alpha^2$ (Dukler et al., 2020):

$$\frac{d\mathbf{f}}{dt} = -\boldsymbol{\Lambda}(t)(\mathbf{f}(t) - \mathbf{y}), \quad \text{where} \quad \boldsymbol{\Lambda}(t) := \frac{\mathbf{V}(t)}{\alpha^2} + \mathbf{G}(t).$$

The specific forms of $\mathbf{V}(t)$ and $\mathbf{G}(t)$ are given in Appendix B.1. Let $\lambda_{min}(A)$ denote the minimal eigenvalue of matrix A. The matrices $V(t)$ and $G(t)$ are positive semi-definite, since they can be viewed as covariance matrices. This gives $\lambda_{\min}(\boldsymbol{\Lambda}(t)) \geq \max\left\{\lambda_{\min}(\mathbf{V}(t))/\alpha^2, \lambda_{\min}(\mathbf{G}(t))\right\}$. According to NTK, the convergence rate is controlled by $\lambda_{min}(\boldsymbol{\Lambda}(t))$. Then, for $\alpha > 1$, convergence is dominated by $\mathbf{G}(t)$. Let $\boldsymbol{\Lambda}(t)$ and $\boldsymbol{\Lambda}^*(t)$ denote the evolution dynamics of FedAvg and FedBN and let $\boldsymbol{G}(t)$ and $\boldsymbol{G}^*(t)$ denote the magnitude component in the evolution dynamics of FedAvg and FedBN. For the convergence analysis, we use the auxiliary version of the Gram matrices, which is defined as follows.

Definition 4.2. *Given sample points $\{\mathbf{x}_p\}_{p=1}^{NM}$, we define the auxiliary Gram matrices $\mathbf{G}^\infty \in \mathbb{R}^{NM \times NM}$ and $\mathbf{G}^{*\infty} \in \mathbb{R}^{NM \times NM}$ as*

$$\mathbf{G}_{pq}^\infty := \mathbb{E}_{\mathbf{v} \sim N(0, \alpha^2 \mathbf{I})} \sigma\left(\mathbf{v}_k(t)^\top \mathbf{x}_p\right) \sigma\left(\mathbf{v}_k(t)^\top \mathbf{x}_q\right), \quad \textit{(FedAVG)} \tag{4}$$

$$\mathbf{G}_{pq}^{*\infty} := \mathbb{E}_{\mathbf{v} \sim N(0, \alpha^2 \mathbf{I})} \sigma\left(\mathbf{v}_k(t)^\top \mathbf{x}_p\right) \sigma\left(\mathbf{v}_k(t)^\top \mathbf{x}_q\right) \mathbb{1}\{i_p = i_q\} \quad \textit{(FedBN)}. \tag{5}$$

[1] We omit centering neurons as well as learning the shift parameter of BN for the neural network analysis because of the assumption that $\mathbf{x}$ is zero-mean and the two layer network setting (Kohler et al., 2019; Salimans & Kingma, 2016)

Given Assumption 4.1, Dukler et al. (2020) showed that $\mathbf{G}^{\infty}$ is positive definite. Using the same idea of proof, we show that $\mathbf{G}^{*\infty}$ is positive definite. We use the fact that the distance between $\mathbf{G}(t)$ and its auxiliary version is small in over-parameterized neural network, such that $\mathbf{G}(t)$ remains positive definite.

Lemma 4.3. *Fix points $\{\mathbf{x}_p\}_{p=1}^{NM}$ satisfying Assumption 1. Then Gram matrices $\mathbf{G}^{\infty}$ and $\mathbf{G}^{*\infty}$ defined as in (4) and (5) are strictly positive definite. Let the least eigenvalues be $\lambda_{\min}(\mathbf{G}^{\infty}) =: \mu_0$ and $\lambda_{\min}(\mathbf{G}^{*\infty}) =: \mu_0^*$, where $\mu_0, \mu_0^* > 0$.*

Proof sketch The main idea is the same as (Du et al., 2018; Dukler et al., 2020), that given points $\{\mathbf{x}_p\}_{p=1}^{NM}$, the matrices $\mathbf{G}^{\infty}$ and $\mathbf{G}^{*\infty}$ can be shown as covariance matrix of linearly independent operators. More details of the proof are given in the Appendix B.2.

Theorem 4.4 (Dukler et al., 2020) gives the convergence rate of FedAvg and Corollary 4.5 gives the convergence rate of FedBN. The key result of comparing the convergence rates between FedAvg and FedBN is culminated in Corollary 4.6.

Theorem 4.4 (**G**-dominated convergence for FedAvg Dukler et al. (2020))**.** *Suppose network (4) is initialized as in 2 with $\alpha > 1$ and Assumptions 4.1 holds. Given the loss function of training the neural network is the square loss with targets $\mathbf{y}$ satisfying $\|\mathbf{y}\|_{\infty} = O(1)$. If $m = \Omega\left(\max\left\{n^4 \log(n/\delta)/\alpha^4\mu_0^4, n^2\log(n/\delta)/\mu_0^2\right\}\right)$, then with probability $1-\delta$,*

1. *For iterations $t = 0, 1, \cdots$, the evolution matrix $\mathbf{\Lambda}(t)$ satisfies $\lambda_{\min}(\mathbf{\Lambda}(t)) \geq \frac{\mu_0}{2}$.*
2. *Training with gradient descent of step-size $\eta = O\left(\frac{1}{\|\mathbf{\Lambda}(t)\|}\right)$ converges linearly as*

$$\|\mathbf{f}(t) - \mathbf{y}\|_2^2 \leq \left(1 - \frac{\eta\mu_0}{2}\right)^t \|\mathbf{f}(0) - \mathbf{y}\|_2^2.$$

Following the proof in Dukler et al. (2020), here we characterize the convergence for FedBN.

Corollary 4.5 (**G**-dominated convergence for FedBN)**.** *Suppose network (5) and all other conditions in Theorem 4.4. With probability $1-\delta$, for iterations $t = 0, 1, \cdots$, the evolution matrix $\mathbf{\Lambda}^*(t)$ satisfies $\lambda_{\min}(\mathbf{\Lambda}^*(t)) \geq \frac{\mu_0^*}{2}$ and training with gradient descent of step-size $\eta = O\left(\frac{1}{\|\mathbf{\Lambda}(t)\|}\right)$ converges linearly as $\|\mathbf{f}^*(t) - \mathbf{y}\|_2^2 \leq \left(1 - \frac{\eta\mu_0^*}{2}\right)^t \|\mathbf{f}^*(0) - \mathbf{y}\|_2^2$.*

The exponential factor of convergence for FedAvg $(1 - \eta\mu_0/2)$ and FedBN $(1 - \eta\mu_0^*/2)$ are controlled by the smallest eigenvalue of $\mathbf{G}(t)$, respectively $\mathbf{G}^*(t)$. Then we can analysis the convergence performance of FedAvg and FedBN by comparing $\lambda_{\min}(\mathbf{G}^{\infty})$ and $\lambda_{\min}(\mathbf{G}^{*\infty})$.

Corollary 4.6 (Convergence rate comparison between FedAvg and FedBN)**.** *For the **G**-dominated convergence, the convergence rate of FedBN is faster than that of FedAvg.*

Proof sketch The key is to show $\lambda_{\min}(\mathbf{G}^{\infty}) < \lambda_{\min}(\mathbf{G}^{*\infty})$. Comparing equation (4) and (5), $\mathbf{G}^{*\infty}$ takes the $M \times M$ block matrices on the diagonal of $\mathbf{G}^{\infty}$. Let $\mathbf{G}_i^{\infty}$ be the i-th $M \times M$ block matrices on the diagonal of $\mathbf{G}^{\infty}$. By linear algebra, $\lambda_{\min}(\mathbf{G}_i^{\infty}) > \lambda_{\min}(\mathbf{G}^{\infty})$ for $i \in [N]$. Since $\mathbf{G}^{*\infty} = diag(\mathbf{G}_1^{\infty}, \cdots, \mathbf{G}_N^{\infty})$, we have $\lambda_{\min}(\mathbf{G}^{*\infty}) > \lambda_{\min}(\mathbf{G}_i^{\infty})$ for all $i \in [N]$. Therefore, we have the result $\lambda_{\min}(\mathbf{G}^{*\infty}) > \lambda_{\min}(\mathbf{G}^{\infty})$.

5 Experiments

In this section, we demonstrate that using local BN parameters is beneficial in the presence of feature shift across clients with heterogeneity data. Our novel local parameter sharing strategy, FedBN, achieves more robust and faster convergence for feature shift non-iid datasets and obtains better model performance compared to alternative methods.

5.1 Benchmark Experiments

Settings: We perform extensive empirical analysis using a benchmark digits classification task containing different data sources with feature shift where each dataset is from a different domain. Data of different domains have heterogeneous appearance but share the same labels

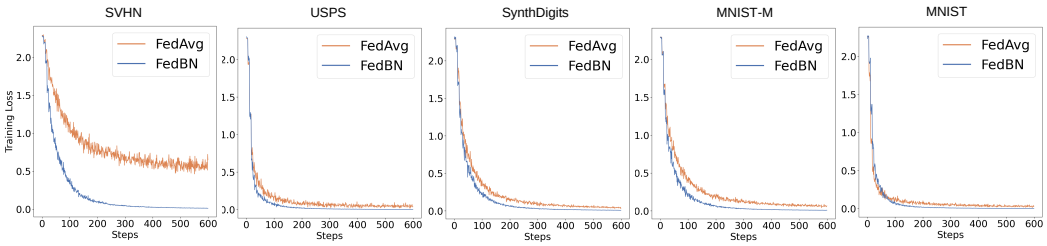

Figure 3: Convergence of the training loss of FedBN and FedAvg on the digits classification datasets. FedBN exhibits faster and more robust convergence.

and label distribution. Specifically, we used the following five datasets: SVHN Netzer et al. (2011), USPS Hull (1994), SynthDigits Ganin & Lempitsky (2015), MNIST-M Ganin & Lempitsky (2015) and MNIST LeCun et al. (1998). To match the setup in Section 4, we truncate the sample size of the five datasets to their smallest number with random sampling, resulting in 7438 training samples in each dataset. Testing samples are held out and kept the same for all the experiments on this benchmark. Our classification model is a convolutional neural network where BN is added following each feature extraction layer (i.e., convolutional and fully-connected), and the detailed architecture is listed in Appendix D.1. For model training, we use the cross-entropy loss and SGD optimizer with 10^{-2} learning rate. If not specified, our default setting for local update epoch is $E = 1$, and the default setting for the data amount at each client is 10% of the dataset original size. Only one client from each of the five datasets joins FL system. More details are listed in Appendix D.1.

Overviews: In the following paragraphs, we present a comprehensive investigation on the properties of our proposed FedBN, including: (1) convergence rate; (2) different choice of local update epoch; (3) performance on various-scales of data amount contributed by each client; (4) effects at different level of heterogeneity; (5) comparison to the state-of-the-art method (FedProx (Li et al., 2020b)), and two baselines of FedAvg and SingleSet (i.e., training an individual model within each client).

Convergence Rate: We analyze the training loss curve of FedBN in comparison with FedAvg, as shown in Fig. 3. The loss of FedBN goes down faster and smoother than FedAvg, indicating that FedBN has a larger convergence rate. In addition, compared to FedAvg, FedBN presents smoother and more stable loss curves during learning. These experimental observations show consensus with what given by Colloray 4.6. In addition, we present a more comprehensive comparison with different local update epochs E on convergence rate of FedBN and FedAvg (see Appendix D.4). The results show similar patterns as in Fig 3.

Analysis of Local Updating Epochs: Aggregating at different frequencies may affect the learning behaviour. Although our theory and the default setting for the other experiment takes $E = 1$, we demonstrate FedBN is effective for cases when $E > 1$. In Fig.4 (a), we explore $E = 1, 4, 8, 16$ and compare FedBN to baseline FedAvg. As expected, an inverse relationship between the local updating epochs E and testing accuracy implied for both FedBN and FedAvg shown in Fig.4 (a). Zooming into the final testing accuracy, FedBN reaches stable accuracy (0.840 ± 0.015) on various E and exceeds the accuracy of FedAvg.

Analysis of Local Dataset Size: We vary the data amount for each client from 100% to 1% of its original dataset size, in order to observe FedBN behaviour over different data capacities at each client. The results in Fig.4 (b) present the accuracy of FedBN and SingleSet. Testing accuracy starts to significantly drop when each of the local client is only attributed 20% percentage of data from its original data amount. The improvement margin gained from FedBN increases as local dataset sizes decrease. The results indicate that FedBN can effectively benefit from collaborative training on distributed data, especially when each client only holds a small amount of data which are non-iid.

Effects of Statistical Heterogeneity: A salient question that arises is: to what degree of heterogeneity on feature shift FedBN is superior to FedAvg. To answer the question, we simulate a federated settings with varying heterogeneity as described below. We parcelled each dataset into 10 subsets, one for each clients, with equal number of data samples and the same label distribution. We treated the clients generated from the same dataset as iid

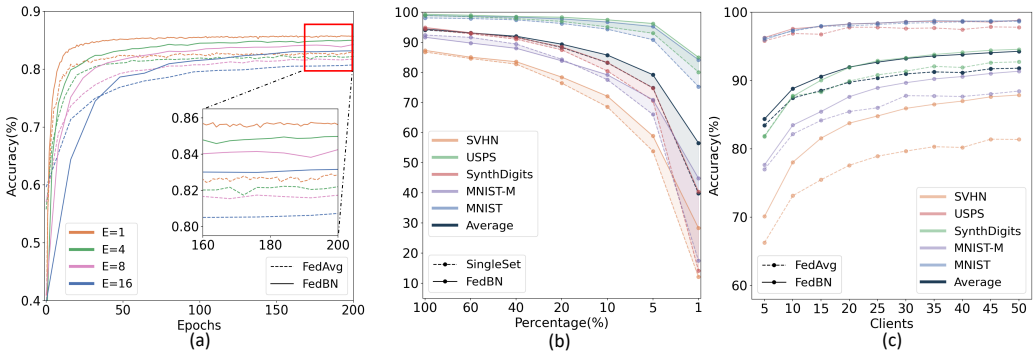

Figure 4: Analytical experimental results on: (a) Analysis on different local updating epochs. FedBN consistently outperforms FedAvg in testing accuracy. (b) Model performance over varying dataset size on local clients. (c) Testing accuracy on different levels of heterogeneity.

clients, while the clients generated from different datasets as non-iid clients. We started with including one client from each dataset in FL system. Then, we simultaneously added n clients from each datasets, for $n \in \{1, \ldots, 10\}$. More clients correspond to less heterogenity. We show the testing accuracy under different level of heterogeneity in Fig. 4 (c) and include a comparison with FedAvg, which is designed for iid FL. Our FedBN achieves substantially higher testing accuracy than FedAvg over all levels of heterogeneity.

Comparison with state-of-the-art: To further validate our method, we compare FedBN with one of the current state-of-the-art methods for non-iid FL, FedProx Li et al. (2020b), which also shares the benefit of easy adaptation to current FL frameworks in practice. We also include training on SingleSet and FedAvg as baselines. For each strategy, we split an independent testing datasets on clients and report the accuracy on the testing datasets. We performed 5-trial repeating experiment with different random seeds. The mean and standard deviation of the accuracy on each dataset over trials are shown in Fig. 5. From the results, we can make the following observation: (1) FedBN achieves the highest accuracy, consistently outperforming the state-of-the-art and baseline methods. (2) FedBN achieves the most significant improvements on SVHN whose image appearance is very different from others (i.e., presenting more obvious feature shift); (3) FedBN shows a smaller variance in error over multiple runs, indicating its stability.

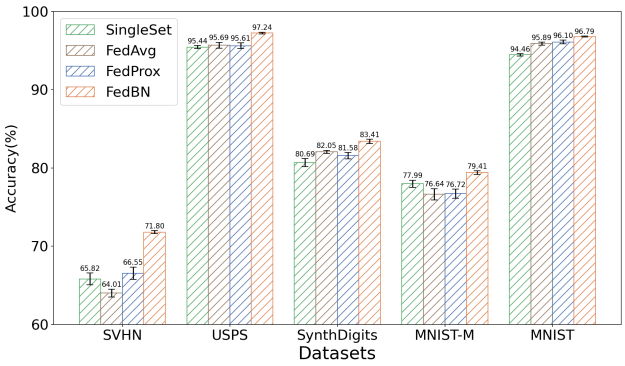

Figure 5: Performance on benchmark experiments.

5.2 Experiments on real-world datasets

To better understand how our proposed algorithm can be beneficial in real-word feature-shift non-iid, we have extensively validated the effectiveness of FedBN in comparison with other methods on three real-world datasets: image classification on Office-Caltech10 (Gong et al., 2012) with images acquired in different cameras or environments; image classification on DomainNet (Peng et al., 2019) with different image styles; and a neurodisorder diagnosis (Di Martino et al., 2014) task with patients from different medical institutions [2].

[2]More details about the datasets and training process are listed in the Appendix D.2 and D.3

Method	Caltech-10				DomainNet						ABIDE (medical)			
	A	C	D	W	C	I	P	Q	R	S	NYU	USM	UM	UCLA
SingleSet	58.5 (0.9)	42.3 (1.7)	83.8 (1.3)	91.5 (2.1)	44.5 (0.5)	26.6 (0.5)	37.4 (0.9)	**71.1 (0.3)**	52.0 (0.8)	35.8 (1.1)	58.0 (3.3)	73.4 (2.2)	64.3 (1.4)	57.3 (2.4)
FedAvg	57.6 (2.6)	47.8 (1.8)	78.8 (4.6)	89.5 (2.7)	51.1 (0.6)	29.2 (0.3)	39.6 (0.5)	57.7 (0.8)	49.7 (0.5)	39.9 (0.5)	62.7 (1.7)	73.1 (2.4)	**70.7 (0.5)**	64.7 (0.7)
FedProx	60.0 (0.9)	46.9 (1.3)	75.6 (1.3)	87.5 (1.4)	50.8 (1.0)	28.0 (0.3)	39.0 (0.9)	56.9 (1.5)	50.2 (0.7)	40.0 (0.4)	63.3 (1.0)	73.0 (1.8)	70.5 (1.1)	64.5 (1.2)
FedBN	**67.0 (0.6)**	**50.1 (0.6)**	**95.0 (1.5)**	**97.0 (0.7)**	**53.4 (1.2)**	**30.4 (0.5)**	**43.4 (1.2)**	71.1 (0.6)	**57.5 (0.7)**	**43.2 (0.6)**	**65.6 (1.1)**	**75.1 (1.4)**	68.6 (2.9)	**65.5 (1.0)**

Table 1: We report results on three different real-world datasets with format mean(std) from 5-trial run. For Office-Caltech 10, *A, C, D ,W* are abbreviations for Amazon, Caltech, DSLR and WebCam, for DomainNet, *C, I, P, Q, R, S* are abbreviations for Clipart, Infograph, Painting, Quickdraw, Real and Sketch. For ABIDE, we list the abbreviations for the clients (i.e., medical institutions).

Datasets and Setup: (1) We conduct the classification task on natural images from *Office-Caltech10*, which has four data sources composing Office-31 Saenko et al. (2010) (three data sources) and Caltech-256 datasets (one data source) Griffin et al. (2007), which are acquired using different camera devices or in different real environment with various background. (2) Our second dataset is *DomainNet*, which contains natural images coming from six different data sources: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. (3) We validate on a medical application. We include four medical institutions (NYU, USM, UM, UCLA, which are viewed as clients) from *ABIDE I* that collects functional brain images using different imaging equipment and protocols, for binary classification between autism spectrum disorders patients and healthy control subjects.

The Office-Caltech10 contains ten categories of objects. The DomainNet extensively contains 345 object categories and we use the top ten most common classes to form a sub-dataset for our experiments. Our classification models adopt AlexNet Krizhevsky et al. (2012) architecture with BN added after each convolution and fully-connected layer. Before feeding into the network, all images were resized to $256 \times 256 \times 3$. For ABIDE I, each instance was represented as a 5995-dimensional vector through brain connectome computation. We use a three-layer fully connected neural network as the classifier with the hidden layers of 16 with two BN layers after the first two fully connected layers. Same as the above benchmark, we perform 5 repeated runs for each experiment.

Results and Analysis: The experimental results are shown in Table 1 in the form of mean (std). On Office-Caltech10, FedBN significantly outperforms the state-of-the-art method of FedProx, and improves at least 6% on mean accuracy compared with all the alternative methods. On DomainNet, FedBN achieved supreme accuracy over most of the datasets. Interestingly, we find the alternative FL methods achieves comparable results with SingleSet except Quickdraw, and FedBN outperforms them over 10%. Surprisingly, for the above two tasks, the alternative FL strategies are ineffective in the feature shift non-iid datasets, even worse than using single client data for training for most of the clients. In ABIDE I, FedBN excell by a non-negligible margin on three clients regarding the mean testing accuracy. The results are inspiring and bring the hope of deploying FedBN to healthcare field, where data are often limited, isolated and heterogeneous on features.

6 Conclusion

This work proposes a novel federated learning aggregation method called FedBN that keeps the local Batch Normalization parameters not synchronized with the global model, such that it mitigates feature shifts in non-IID data. We provide the convergence guarantees for FedBN in realistic federated settings under a two-layer neural network assumption, while also accounting for practical issues. In our experiments, our evaluation across a suite of federated datasets has demonstrated that FedBN can significantly improve the convergence behavior and model performance of non-IID datasets. Finally, we note that since FedBN makes only lightweight modifications to FedAvg, this allows us to reason about the behavior of the widely-used FedAvg method, and enables easy integration of FedBN into existing packages/systems, such as Pysyft (https://pysyft.readthedocs.io/en/latest/) and FedML (He et al., 2020).

References

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *arXiv preprint arXiv:1901.08584*, 2019.

Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7354–7362, 2019.

Adriana Di Martino, Chao-Gan Yan, Qingyang Li, Erin Denio, Francisco X Castellanos, Kaat Alaerts, Jeffrey S Anderson, Michal Assaf, Susan Y Bookheimer, Mirella Dapretto, et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular psychiatry*, 19(6):659, 2014.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

Yonatan Dukler, Quanquan Gu, and Guido Montúfar. Optimization theory for relu neural networks trained with normalization layers, 2020.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2066–2073. IEEE, 2012.

Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset, 2007.

Chaoyang He, Songze Li, Jinhyun So, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Li Shen, et al. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip B Gibbons. The non-iid data quagmire of decentralized machine learning. *arXiv preprint arXiv:1910.00189*, 2019.

Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378*, 2019.

Jonas Kohler, Hadi Daneshmand, Aurelien Lucchi, Thomas Hofmann, Ming Zhou, and Klaus Neymeyr. Exponential convergence rates for batch normalization: The power of length-direction decoupling in non-convex optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 806–815. PMLR, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3): 50–60, 2020a.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Conference on Machine Learning and Systems, 2020a*, 2020b.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.

Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018.

Quande Liu, Qi Dou, Lequan Yu, and Pheng Ann Heng. Ms-net: Multi-site network for improving prostate segmentation with heterogeneous mri data. *IEEE Transactions on Medical Imaging*, 2020.

Ping Luo, Xinjiang Wang, Wenqi Shao, and Zhanglin Peng. Towards understanding regularization in batch normalization. *arXiv preprint arXiv:1809.00846*, 2018.

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*, pp. 1273–1282, 2017.

Ari S Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. *arXiv preprint arXiv:1803.06959*, 2018.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning, 2011.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1406–1415, 2019.

Amirhossein Reisizadeh, Farzan Farnia, Ramtin Pedarsani, and Ali Jadbabaie. Robust federated learning: The case of affine distribution shifts. *arXiv preprint arXiv:2006.08907*, 2020.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pp. 213–226. Springer, 2010.

Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in neural information processing systems*, pp. 901–909, 2016.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, pp. 2483–2493, 2018.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

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.