2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS) | 978-1-6654-7315-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICPADS56603.2022.00110

Accelerating Federated Learning with Two-phase Gradient Adjustment

Jiajun Wang[b], Yingchi Mao[a, b], Xiaoming He[b], Tong Zhou[b], Jun Wu[b], and Jie Wu[c]
[a] Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University, Nanjing, China
[b] School of Computer and Information, Hohai University, Nanjing, China
[c] Center of Networked Computing, Temple University, Philadelphia, USA
211307040017@hhu.edu.cn, yingchimao@hhu.edu.cn, isxmhe@gmail.com
191307040039@hhu.edu.cn, 1606010225@hhu.edu.cn, jiewu@temple.edu

***Abstract*—With the advent of the Internet of Things (IoT) era and 5G, ubiquitous sensing devices (*e.g.*, smartphones, surveillance sites, and security cameras) have been widely used in various fields, resulting in the generation of a huge amount of monitoring data. The rise of federated learning makes it possible to leverage monitoring data to train deep neural networks through cloud-edge collaboration without compromising privacy. However, the non identically and independently distributed (called Non-IID) data collected by IoT devieces creates a client drift phenomenon, resulting in a slow convergence of the global model. To this end, we propose a new Federated learning framework based on Gradient Variance Reduction with a correction weight control mechanism and Global gradient descent with Momentum, named FedGVRGM to conduct gradient correction and reduce the negative impacts of prediction parameters. Specifically, in the local training phase, FedGVRGM combines gradient variance reduction with a correction weight control mechanism to further correct the local model parameters, thus reducing the dispersion of model parameters among clients. In the global aggregation phase, FedGVRGM integrates the historical change states of the global model through the gradient descent with momentum to reduce the oscillations and improve the convergence speed of the global model. We refer to the above methods of gradient adjustment in the local and global training phases as FedGVR and FedGM, respectively. Numerous evaluations are conducted on CIFAR-100, CIFAR-10, and MNIST datasets to prove that FedGVRGM has a faster convergence rate than other state-of-the-art approaches such as Federated Averaging (FedAvg), FedProx, FedReg, FedGVR, and FedGM.**

***Index Terms*—Federated Learning, Client Drift, Cloud-Edge, Distributed Optimization, Fast Convergence Rate.**

I. Introduction

With the popularity of IoT devices and the advent of 5G era, data privacy has gradually become a key area of public concern. As a revolutionary distributed machine learning framework, federated learning [1] can train deep neural networks through cloud-edge collaboration without compromising privacy [2]–[4].

However, in federated learning, the data among clients are typically heterogeneous and fail to adhere to the assumption of independent identical distribution. The above data are called not identically and independently distribution (Non-IID) data. The Non-IID data causes local models to gradually converge to be local optimum and produce a client drift phenomenon [5]. This leads to a high variance in the global aggregation phase, and the global model has difficulty in converging to the best average loss for all clients [6]–[10]. In addition, the client drift phenomenon hurts the convergence speed, so it is critical to mitigate the client drift phenomenon during federated training.

In the local training phase, the client drift phenomenon can be considerably mitigated by reducing the dispersion of model parameters among clients. Liang $et\ al.$ [11] proposed a Variance Reduced Local Stochastic Gradient Descent method called VRL-SGD that incorporated variance reduction into the local SGD to eliminate the assumption of bounded gradient variance. However, VRL-SGD can't support client sampling and has a slow convergence rate. Li $et\ al.$ [7] added a correction term to the local loss function to suppress the degree of parameter dispersion among clients. Although it can achieve a high accuracy on Non-IID data, the global model has a lower convergence rate and requires a higher number of iteration rounds. Karimireddy $et\ al.$ [5] presented a Stochastic Controlled Averaging for Federated Learning approach named SCAFFOLD to model the degree of client drift and corrected local parameter updates based on control variates. However, SCAFFOLD has errors in calculating the degree of client drift, which can negatively affect the model parameters close to the optimal value in the post-training phase. Xu $et\ al.$ [12] proposed a novel algorithm called FedReg that alleviated the catastrophic forgetting issue by regularizing local training parameters with the generated pseudo data, thereby reducing the degree of client drift.

In addition, some methods are proposed to accelerate the convergence of federated learning in the global aggregation phase. Reddi $et\ al.$ [13] introduced adaptive schemes such as ADAM, YOGI, and ADAGRAD on the parameter server to accelerate model convergence. This adaptive optimization makes federated aggregation smoother, thus providing an efficient way to avoid client drift. Wang $et\ al.$ [14] gave a Slow Momentum Framework called Slowmo, built on the top of SGD, and decentralized methods to utilise the slow momentum in the global phase.

Although many studies are attempted to mitigate client drift by correcting local or global model parameters, these approaches only tune the federated training from a single perspective. Because client drift has an impact on the entire process of federated learning, so adjusting the federated learning algorithm only from a single phase may be one-sided.

978-1-6654-7315-6/23/$31.00 ©2023 IEEE
DOI 10.1109/ICPADS56603.2022.00110

Firstly, the algorithm is still affected by client drift when the model parameters are not corrected in the local training phase but only in the global aggregation phase. Secondly, when the model parameters are only corrected in the local training phase, the simple weighted average in the global aggregation phase amplifies the client drift phenomenon caused by Non-IID data.

Based on the above observations, we propose a new algorithm named FedGVRGM to mitigate the client drift phenomenon and speed up federated learning convergence. Specifically, to reduce the degree of client drift, FedGVRGM first corrects the local model parameters by the variance of the model prediction parameters between the clients and the parameter server. Secondly, FedGVRGM accelerates the convergence of the global model using the gradient descent with momentum in the global aggregation phase. By introducing gradient variance reduction with a correction weight control mechanism and gradient descent with momentum in the local training phase and global aggregation phase, respectively, FedGVRGM alleviates the client drift phenomenon and achieves fast convergence.

The main contributions of this paper are as follows,

- We propose a new method, *i.e.*, FedGVRGM to mitigate the client drift phenomenon and accelerate the convergence speed in federated learning. Specifically, FedGVRGM corrects model parameters in both the local training phase and the global aggregation phase during the federated training, ensuring a high convergence rate even when the data is seriously heterogeneous.
- Gradient variance reduction with correction weight control is used to reduce the dispersion of model parameters among clients. And the correction weight control mechanism dynamically adjusts the degree of the adjustment of the local gradient to reduce the impact of errors of prediction parameters according to the change rate of global model historical accuracy.
- Experiments on real datasets demonstrate that the FedGVRGM algorithm can mitigate the client drift phenomenon and improve the convergence speed. Especially, on the more complex datasets of CIFAR-100, FedGVRGM achieves speedup ratios of $1.55\times$, $1.12\times$, $1.63\times$, $1.30\times$, and $1.22\times$, respectively, compared to FedAvg, FedProx, FedReg, FedGM and FedGVR, when the data is highly heterogeneous ($\alpha = 0.5$).

The remainder of this paper is organized as follows. Section II introduces the related work. Section III covers the basics of federated learning. FedGVRGM is described in detail in Section IV. The experimental evaluations are presented in Section V. Finally, conclusions are drawn in Section VI.

II. Related work

In this section, we discuss the related works, focusing on three topics, *i.e.*, (1) the similarity improvement of the data distribution among clients, (2) gradient adjustment in the local training phase and (3) gradient adjustment in the global aggregation phase.

A. Similarity Improvement of the Data Distribution among Clients

To alleviate the client drift phenomenon, an effective scheme is proposed to improve the similarity of data distribution and avoid the client drift phenomenon fundamentally by changing heterogeneous data into homogeneous data. Specially, data sharing strategies improve the training effectiveness of federated learning based on a public dataset. Wang $et\ al.$ [15] shared local output logits on the training dataset while Zhu $et\ al.$ [16] shared local label information. However, these approaches that convert heterogeneous datasets to homogeneous datasets are costly to produce for public datasets, lack the universality of application contexts, and increase communication overhead.

B. Gradient Adjustment in Local Training Phase

Obviously, many studies are willing to mitigate the client drift phenomenon in federated learning. A well-established approach is to correct the local model from drifting towards the local optimum by adding a regular term to the local loss function. Li $et\ al.$ [7] added a correction term to the local loss function to suppress the degree of dispersion of parameters among clients. Although it can achieve a high accuracy on Non-IID data, the global model has a lower convergence rate and requires a higher number of training rounds. In terms of activation regularization, contrastive learning [17], mixup with global statistics [18], and generative models [19] are used to ensure that client updates have similar activation to the global model. Furthermore, correcting local updates using global model parameters to reduce differences among local model parameters is also a common approach. Karimireddy $et\ al.$ [5] proposed a VRL-SGD method to characterise the differences between local and global model parameters based on control variables, which achieved a variance reduction and improved the speed of model convergence. Liang $et\ al.$ [11] incorporated variance reduction into the local SGD to eliminate the assumption of bounded gradient variance. However, the VRL-SGD was unable to support client sampling and had a slow convergence rate.

Based on the above description, we find that the local training process can be improved to reduce the dispersion among client-side models, for example, by adding regularization terms, thereby improving the convergence speed of the global model. However, the simple weighted average in the global aggregation phase may amplify the client drift phenomenon caused by Non-IID data.

C. Gradient Adjustment in Global Aggregation Phase

In addition to gradient adjustments during the local training phase, there are many methods in the global aggregation phase to accelerate the convergence of federated learning. Reddi $et\ al.$ [13] introduced adaptive schemes such as ADAM, YOGI, and ADAGRAD on the parameter server to accelerate model convergence. This adaptive optimization makes federated aggregation smoother, thus providing an efficient way aims to avoid client drift in federated learning. Wang $et\ al.$ [14] gave

a slow momentum framework, which is built on top of SGD, and decentralized methods, such as SGP and OSGP to utilize the slow momentum in the global phase, enabling accelerated training without sacrificing accuracy.

In a brief, although adjusting the gradient in the global aggregation phase can accelerate federated learning, the client drift problem is not properly addressed.

D. Our Motivation

In summary, most studies only mitigate the client drift phenomenon from the local training [20]–[23] or global aggregation [13], [14], [24] phases. Because, the client drift has an impact on the federated learning, so adjusting the federated learning architecture only from a single-phase may be one-sided. Therefore, we propose a new algorithm, named FedGVRGM, to optimize federated learning architecture, *i.e.*, dealing with the client drift phenomenon more comprehensively and speeding up the convergence from two sides. Furthermore, we combine the gradient variance reduction with a correction weight control mechanism to dynamically adjust the degree of adjustment of the local gradient so as to reduce the negative impact of errors in prediction parameters according to the change rate of the historical accuracy of the global model.

III. Preliminaries

A. Federated learning

In a federated learning system, we consider a network Ne consists of a cloud parameter server and N edge nodes (ENs). Note that, ENs can be local computers or IoT devices that can be used for data training. The goal of federated learning is to train the global model w^T on Net by minimizing the loss function $f(w)$ over the dataset $D = \cup D_k$ to obtain the optimal value. In this paper, we focus on exploring the effect of Non-IID data on the convergence rate of the global model. We set the computing power of ENs as homogeneous and assume the same amount of data among ENs, so the loss function of minimizing N ENs is expressed as,

$$\min \ f(w) = \min \left\{ \frac{1}{N} \sum_{k=1}^{N} F_k(w) \right\}, \tag{1}$$

where F_k is the local loss function for the edge node k. To be consistent with the common terminology used in federated learning, we uniformly refer to edge node as client later on. All the basic notations in this paper are listed in Table 1.

B. Client drift

Client drift is the degree of parameter dissimilarity between the local model and the global model in federated learning where the update from one client overwrites the model weights learned with data from other clients.

There exist constants $G \geq 0, B \geq 1$, for any vector w , if

$$\frac{1}{K} \sum_{k=1}^{K} \|\nabla F_k(w)\|^2 \leq G^2 + B^2 \|\nabla f(w)\|^2, \tag{2}$$

TABLE I
Main notations used in FedGVRGM.

Notation	Meaning
N	The number of clients
E	Local epoch
K	The number of client samples per round
T	Communication rounds
D_k	Data stored by client k
S_t	Sampled devices in round t
w_0	Initial value of the global model
c	Global model prediction parameters
c_k	Local model prediction parameters for client k
η_g	Global learning rate
η_l	Local learning rate

then the loss function $F_k(w)$ of k is to satisfy the bounded gradient dissimilarity condition. Equation 2 describes the mathematical form of the concept of client drift, and further derivation of the inequality yields,

$$\sqrt{\frac{\frac{1}{K} \sum_{k=1}^{K} \|\nabla F_k(w)\|^2}{\|\nabla f(w)\|^2}} \leq B, \tag{3}$$

and so that,

$$\Gamma = \sqrt{\frac{\frac{1}{K} \sum_{k=1}^{K} \|\nabla F_k(w)\|^2}{\|\nabla f(w)\|^2}}, \tag{4}$$

where Γ indicates the degree of client drift among K clients, the larger the Γ, the greater the client drift degree, and vice versa, the smaller the client drift degree. B indicates the upper bound of the degree of client drift, when all the clients have the same loss function, we can get $\Gamma = B = 1$, and there is no client drift at this time. Even if the data stored by the clients are independently and homogeneously distributed, the heterogeneous data may be generated due to data sampling and other reasons, so B is greater than 1, thus the client drift phenomenon is inevitable in federated learning.

IV. FedGVRGM

In this paper, we discuss the problem of slow convergence due to the client drift phenomenon on Non-IID data and propose the FedGVRGM framework. Specifically, FedGVRGM conducts gradient correction for both the local training and global aggregation phases in federated learning. In the local training phase, FedGVRGM corrects the local gradient based on the variance of the local and global model prediction parameters to improve the similarity of local models. In the global aggregation phase, FedGVRGM updates global model parameters using gradient descent with momentum to reduce the oscillation of the global model and achieve rapid convergence. In addition, in the local training phase, each client receives the global model w^t, the global model prediction parameter c^t and the weight control term ε from the server. The correction term $(c - c_k)$ measured the degree of client drift, is then calculated, and the local gradient is updated using $\varepsilon(c - c_k)$. Finally, local model prediction parameters

Algorithm 1 Federated Learning via Gradient Variance Reduction and Global Mountain

Input: w_0, c, η_g, η_l, T, E, test accuracy upper bound array sub, constant h, momentum decay parameters β, momentum parameters mc, momentum parameters mw

Output: global model: w^T

Initialize w_0, c, c_k
for $t = 0, ..., T-1$ **do**
  Server select a subset S_t of K divices at random
  $\varepsilon = \mathrm{CWC}\,(t, h, sub)$
  sent (w, c, ε) to all chosen clients $k \in S_t$
  for $k \in S_t$ **do**
    initialize local model $w_k = w^t$
    for $e = 0, ..., E-1$ **do**
      $g_k(w_k) = \nabla h_k(w_k, w^t) = \nabla F_k(w_k) + \mu(w_k - w^t)$
      $w_k = w^t - \eta_l \left(g_k\left(w_k\right) + \varepsilon\left(c - c_k\right)\right)$
    end for
    $c_k^{cur} = c_k - c + \frac{\eta_l}{E}\left(w^t - w^k\right)$
    communicate$(\Delta w_k, \Delta c_k) \leftarrow (w_k - w^t, c_k^{cur} - c_k)$
    $c_k = c_k^{cur}$
  end for
  $w^{t+1} = GM\left(w^t, \beta, \triangle w, mw, |S_t|\right)$
  $c^{t+1} = GM\left(c^t, \beta, \triangle c, mc, |S_t|\right)$
  $acc(t)$ **add to** sub
end for
return w^T

are updated. In the global aggregation phase, the parameter server first calculates the correction term weight ε according to the correction weight control algorithm , and then updates the global model parameters w^{t+1} and global prediction parameters c^{t+1} by the gradient descent with momentum algorithm. The detailed steps can be found in Algorithm 1.

A. Local training phase

In t-th round of communication, the parameter server stores the global model prediction parameter c, and the client stores the local model prediction parameter c_k. Note that, c and c_k are initialized to 0.

The parameter server sends w^t, c^t, ε to all clients in S_t, each participating client initializes the local model $w_k \leftarrow w^t$. The local loss function can be expressed as,

$$h_k\left(w, w^t\right) = F_k\left(w\right) + \frac{\mu}{2}\|w - w^t\|^2, \tag{5}$$

where $\frac{\mu}{2}\|w - w^t\|^2$ is the proximal term.

However, the local model parameters of FedGVRGM are no longer updated directly based on the model gradient, while a further correction to the model gradient is as follows,

$$w_k^t = w^t - \eta_l\left(g_k\left(w_k^t\right) + \varepsilon\left(c^t - c_k^t\right)\right), \tag{6}$$

where $g_k\left(\cdot\right)$ denotes the local model gradient, and ε is used to control the degree of $(c - c_k)$ to the local gradient.

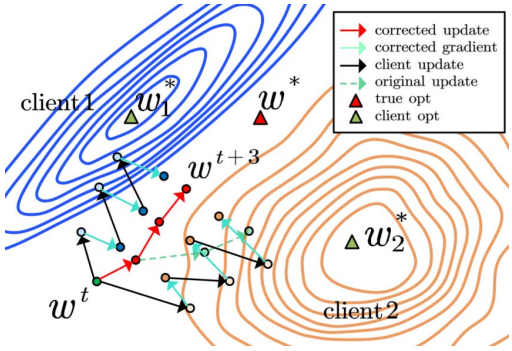

Fig. 1. Local gradient correction.

After the client has executed E iterations, FedGVRGM updates the local model prediction parameters c_k^t according to,

$$c_k^{cur} = c_k^t - c^t + \frac{1}{E}\eta_l\left(w_k^t - w^t\right). \tag{7}$$

To visualize the principle of local gradient correction, Fig. 1 shows the gradient correction process of two devices for three rounds ($T = 3$) of communication and one epoch ($E = 1$) of local training. The black arrow shows the local gradient before correction, and the light green arrow indicates the correction of the local gradient. The corrected client model parameters (blue and orange circles) are more concentrated compared with the uncorrected parameters, and the variance of the client model parameters is reduced. The updated route is adjusted from the original update route (green dashed arrow) to the correction route (red arrow), and the global model is updated in the direction of the optimal solution.

B. Global Aggregation Phase

*1) **Correction Weight Control**:* The training process of federated learning can be divided into pre-training and post-training. In the pre-training phase, the model parameters are able to be optimized, and seeking the stable point of parameters quickly is a challenge. However, in the post-training phase, the model parameters have been stabilized, and the model obtained by retraining has a stable point or close to the stable point.

Since c and c_k are predicted parameters, there may be errors in the degree of client drift calculated by $(c - c_k)$. In the pre-training period, the model is pending convergence and the errors in $(c - c_k)$ have a small effect on the local gradient. However, in the post-training phase, when the models have converged or are close to convergence, the errors in $(c - c_k)$ have a large impact on the local gradient and may negatively affect the local models. Even if $\frac{\mu}{2}\|w - w^t\|$ is utilized in the FedGVR loss function to reduce the effect caused by the errors of $(c - c_k)$, the model may fall into a suboptimal state due to inaccurate correction updates for the model parameters. In this section, we design a Correction Weight Control scheme named CWC that dynamically calculates the correction weights based on the historical accuracy, which gradually attenuates the effect

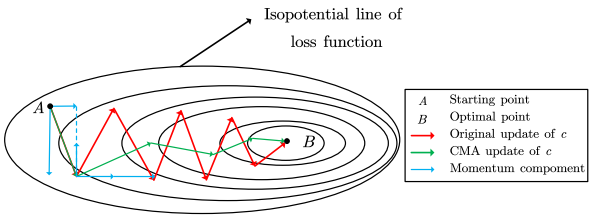

Fig. 2. Comparison of the global model's momentum update route with the original update route.

of $(c - c_k)$ on the local gradient as the model test accuracy increases.

CWC calculates the weighting factor $\varepsilon \in [0.0, 1.0]$, which indicates the change rate of the recent historical accuracy. Specifically, in the t-th round of communication, ε is determined by the change rate in model test accuracy from the $(t-h)$-th to the t-th round, and ε is calculated as,

$$\varepsilon = \begin{cases} (sub\,(t) - sub\,(t-h))\,/h, & if\ t \geq h \\ 1.0, & other, \end{cases} \tag{8}$$

where $sub(t)$ represents the maximum value of historical accuracy in the first t communication rounds.

$$sub\,(t) = \begin{cases} acc\,(t)\,, & if\ acc\,(t) > sub\,(t-1) \\ sub\,(t-1)\,, & other. \end{cases} \tag{9}$$

From Equation (9), when $t \geq t^{'}$, $sub(t) \geq sub(t^{'})$. Therefore, sub is a non-decreasing array. The reason for this sub array's design is that the test accuracy fluctuates in federated training, and if we use the test accuracy $acc\,(\cdot)$ directly to calculate ε, we get $\varepsilon < 0$ when $acc(t) < acc(t-h)$, which aggravates the degree of client drift. At the pre-training phase, the larger change rate of model accuracy brings a larger ε. Thus the correction term $\varepsilon\,(c - c_k)$ has a greater influence on the gradient update, emphasizing the advantage of $\varepsilon\,(c - c_k)$ in eliminating client drift and improving the convergence rate. As the global model converges and the change rate in test accuracy decreases, the value of ε gradually becomes smaller. So the correction term $\varepsilon\,(c - c_k)$ has a smaller influence on the local gradient update, ensuring the stability of the local parameter update in the post-training phase.

*2) **Global Gradient Descent with Momentum**:* The parameter server receives the local model parameters from the clients and employs the local model parameters to update the global model parameters using the Global Gradient Descent with Momentum method named GM as follows,

$$\begin{aligned} \triangle w &= \frac{1}{|S_t|} \sum_{k \in S_t} \left(w_k^t - w^t\right), \\ w^t &= w^t + \frac{|S_t|}{N} \triangle w\ , \\ mw &= mw + w^t, \\ w^{t+1} &= w^t - \beta mw, \end{aligned} \tag{10}$$

TABLE II
SERVER CONFIGURATION

Configuration Name	Configuration parameters
RAM	256G
CPU	64 cores Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz
GPU	NVIDIA A100 GPU *2
GPU graphics memory	80G

where mw is the momentum parameter and the momentum decay factor β controls the influence degree of mw on the global model parameters w.

Fig. 2 represents the original update route (red path) for w compared to the GM update route (green path). The GM update route demonstrates the use of gradient descent with momentum brings the parameters update down in a direction that deviates slightly from the optimum to mitigate the parameters oscillations caused by the original update route. The figure shows that the original update route takes 8 iterations to reach the optimum, while the GM update route only takes 5 iterations to reach the optimum. This indicates that slowing down the gradient oscillations can accelerate the convergence rate.

In addition, GM takes into account the influence of historical w. The magnitude of the movement of w in each direction depends not only on the current parameters , but also the historical parameters. As shown in Fig. 2, if the historical parameters are always updated in the horizontal direction, then the magnitude of future horizontal updates can be greater. However, if the historical parameters are constantly changing in the vertical direction, the magnitude of vertical updates can be decreasing.

The accuracy of c and c_k determines the accuracy of the correction term $(c - c_k)$ in predicting the degree of client drift. To prevent the inaccuracy of the predicted degree of client drift due to the presence of c, the global gradient descent with momentum method is also used for the update of c. The specific update formula is similar to w, so we don't repeat it here.

In summary, the advantages of updating c and w based on the GM algorithm are as follows,

- GM reduces the oscillations of c and w to ensure the stability of local gradient correction and global aggregation.
- GM integrates the historical update status of c and w and accelerates the update rate in the dominant direction. This not only allows the local gradient correction to be constructed on a more accurate correction term $(c - c_k)$, but also speeds up the convergence of the global model in federated learning.

V. PERFORMANCE EVALUATION

A. Experimental setup

*1) **Datasets and Baselines**:* The purpose of our experiments is to evaluate the convergence rate of FedGVRGM on Non-IID datasets. To simulate the Non-IID datasets in real

TABLE III
COMPARISON OF FEDGVRGM ON CIFAR-100 , CIFAR-10 AND MNIST WITH $\alpha = 0.5$.

Method	CIFAR-100				CIFAR-10				MNIST			
	Round		Loss		Round		Loss		Round		Loss	
	ls=0.8	ls=0.45	r=500	r=1000	ls=0.8	ls=0.45	r=500	r=1000	ls=0.35	ls=0.2	r=500	r=1000
FedAvg	597	964	1.124	0.438	228	441	0.391	0.234	45	128	0.133	0.100
FedProx	597	964	1.32	0.436	208	441	0.381	0.23	35	117	0.113	0.084
FedReg	429	716	0.823	0.347	**129**	378	0.344	0.217	34	102	0.110	0.083
FedGM	498	819	0.939	0.356	208	424	0.376	0.216	35	104	0.108	0.083
FedGVR	470	754	0.847	0.34	**129**	270	0.225	0.16	34	101	0.113	0.086
FedGVRGM	**384**	**631**	**0.701**	**0.281**	**129**	**228**	**0.215**	**0.155**	**33**	**83**	**0.105**	**0.081**

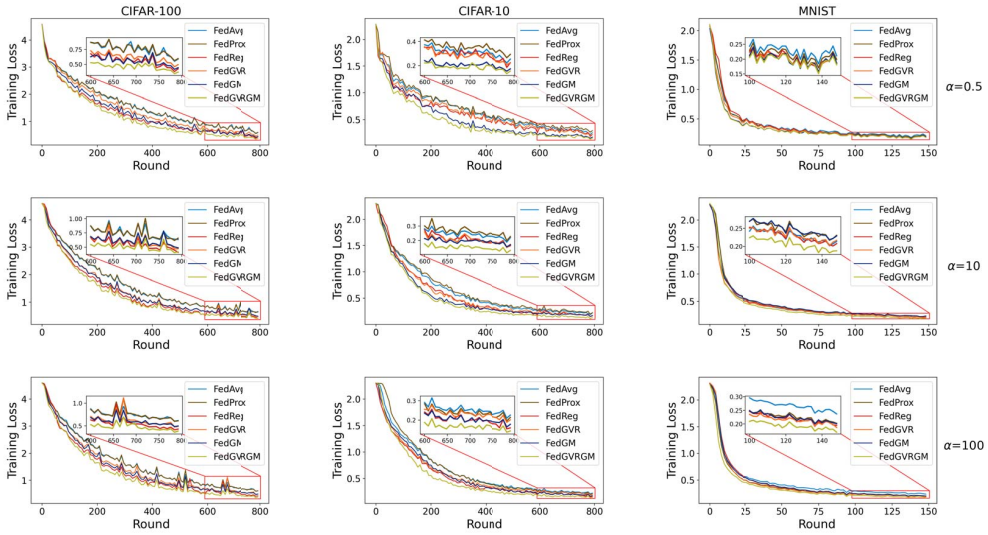

Fig. 3. Training loss plots for FedGVRGM and all baselines on CIRFA-100, CIFAR-10, and MNIST datasets with $\alpha = \{0.5, 10, 100\}$.

environment, the Dirichlet distribution is used to generate datasets with different degrees of heterogeneity. Specifically, to artificially synthesize the heterogeneous data in the training devices, the experiment sets the Dirichlet distribution as $q \sim Dir(\alpha p)$, where $\alpha > 0$ controls the degree of heterogeneity among clients data distributions, and a smaller value of α indicates stronger data heterogeneity.

The datasets include MNIST [27], CIFAR-10 and CIFAR-100 [25], [26]. Three dataset scenarios with different degrees *i.e.*, $\alpha \in \{0.5, 10, 100\}$ are set for 100 clients and the convergence speed of FedGVRGM is verified on the generated heterogeneous datasets.

Experiments are conducted to compare FedGVRGM with state-of-the-art algorithms, such as FedAvg, FedProx, FedReg, FedGVR and FedGM.

*2) **Hyperparameter settings**:* The experiments are set up with 100 clients. The u is set as 0.1 in FedProx. The FedGVR, FedGM and FedGVRGM methods have the same settings as FedProx regarding the u. The η_g is initialized to 1, and η_l is initialized to 0.001. Besides, the constant h is set as 5, and β is set as 0.9.

*3) **Models**:* On the CIFAR-100 and CIFAR-10 datasets, we use two convolutional layers and three fully connected layers, while on the MNIST dataset, we use two convolutional layers and two fully connected layers.

*4) **Validation metrics**:* We measure the performance of FedGVRGM mainly in terms of convergence speed, specifically we measure the loss obtained at a specified number of rounds, and the number of training rounds required to reach the specified value of the loss.

*5) **Simulation environment**:* All experiments in this section are conducted on an Ubuntu 20.04 server with 256G of RAM, which is configured with the parameters in Table 2.

B. Analysis of convergence speed on strongly heterogeneous data

The convergence of FedGVRGM on CIFAR-100, CIFAR-10, and MNIST datasets is first presented and compared with other baselines when the data heterogeneity is high ($\alpha = 0.5$). In this experiment, there are 100 clients, with 10 clients participating in local training in each round of communication and the local epoch is 10.

Table 3 provides a detailed comparison of the convergence speed of FedGVRGM with baselines on three datasets with

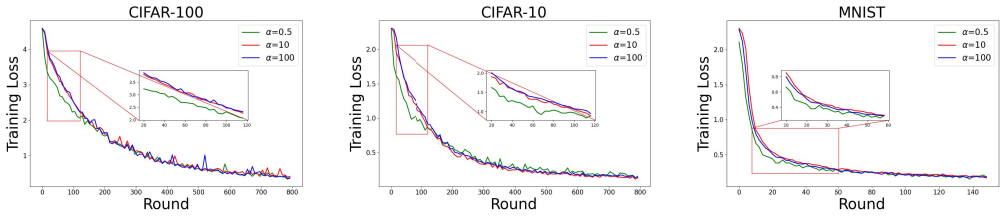

Fig. 4. Comparison of the convergence speed of FedGVRGM on CIRFA-100, CIFAR-10, and MNIST datasets with $\alpha = \{0.5, 10, 100\}$.

$\alpha = 0.5$. The smaller the data, the faster the convergence of the corresponding algorithm, and the best performance in each column is denoted in bold. ls indicates the training loss and r represents training rounds. FedGVRGM converges faster on the strongly heterogeneous dataset. On the more complex datasets of CIFAR-100, FedGVRGM achieves speedup ratios of $1.55\times$, $1.12\times$, $1.63\times$, $1.30\times$, and $1.22\times$, respectively, compared to FedAvg, FedProx, FedReg, FedGM, and FedGVR. FedGVRGM outperforms baselines, mainly because it adjusts the gradient in both training phases of federal learning, thus being able to mitigate client drift more comprehensively.

C. Analysis of convergence speed on data with various degrees of heterogeneity

To further examine the convergence speed of FedGVRGM, more experiments are conducted on CIRFA-100, CIFAR-10, and MNIST datasets with $\alpha \in \{0.5, 10, 100\}$.

Fig. 3 shows the training loss folds of FedAvg, FedProx, FedGM, FedGVR, and FedGVRGM on datasets CIFAR-100, CIFAR-10, and MNIST. The vertical side of the figure represents the loss folds of the algorithms on datasets with different degrees of heterogeneity and the horizontal side represents the training loss folds of the algorithms on different datasets. In all heterogeneous datasets, FedGVRGM has the fastest rate of decline in the value of training loss and the smallest loss. FedAvg and FedProx converge the slowest and their convergence rates are similar. Although FedProx can explore better model parameters, the back propagation of the gradient shortens the length of the gradient vector, resulting in a mediocre convergence. As for FedReg, although it can achieve a faster convergence speed by preventing the local model from drifting towards the local optimal solution through regularizing the local training parameters with the generated pseudo data, the simple averaging in the global training phase still amplifies the client drift phenomenon.

FedGVR and FedGM are single-phase improvements to FedProx and they converge quicker than FedAvg and FedProx which also suggests that client drift is present throughout the whole federated learning process.

For all datasets with varying degrees of data heterogeneity, FedGVRGM has the fastest decline rate in loss and the smallest loss , indicating that FedGVRGM performs consistently. FedGVRGM can deal with the client drift phenomena comprehensively since it optimizes the federated algorithm in both the local training and global aggregation phases.

Longitudinal observation of the training loss line graph for each dataset in Fig. 3 under different heterogeneity shows that as the parameter α gradually becomes larger, the degree of data heterogeneity gradually becomes weaker, the degree of client drift gradually becomes weaker, and the FedGVRGM still maintains a high convergence speed. When $\alpha = 100$, the distribution of each data is close to independent identical distribution, and the convergence of FedGVRGM still has a significant advantage over other methods. This indicates that even if the data are independently identically distributed, the client drift phenomenon can be caused by random sampling of devices or random sampling of training samples. FedGVRGM can not only detect and mitigate the client drift phenomenon caused by strongly heterogeneous data ($\alpha = 0.5$), but also detect and mitigate the weak client drift phenomenon.

The line graph of the training loss of the FedGVRGM algorithm on CIFAR-100, CIFAR-10, and MNIST when the data heterogeneity degree is different is shown in Figure 4. FedGVRGM converges quickly in the pre-training phase in various heterogeneous datasets. In the strongly heterogeneous dataset ($\alpha = 0.5$), the FedGVRGM loss declines at the fastest rate and obtains the minimum loss. When data heterogeneity decreases, the client drift phenomenon is weaker, and the rate of the traning loss decline of FedGVRGM is slower than that of training loss decrease under a strongly heterogeneous dataset. In summary, it can be concluded that FedGVRGM can effectively detect the gradient drift phenomenon with a larger degree and has the best loss value drop rate.

D. Client Participation

Figure 5 shows the number of iteration rounds required to achieve a training loss of 0.45 on the CIFAR-10 dataset with $\alpha = 0.5$ when the client participation rate is 5%, 10%, and 20%, respectively. It can be seen that FedGVRGM achieves a minimum acceleration ratio of 159.9% compared to FedProx at different client participation rates, especially at a client participation rate of 20%, and FedGVRGM achieves a maximum acceleration ratio of 194.0 % compared to FedProx. In addition, as the client participation rate increases and the client drift increases, FedGVRGM can still maintain a high convergence speed, which indicates that FedGVRGM has high robustness and its ability to mitigate client drift does not fail due to the increase of client drift.

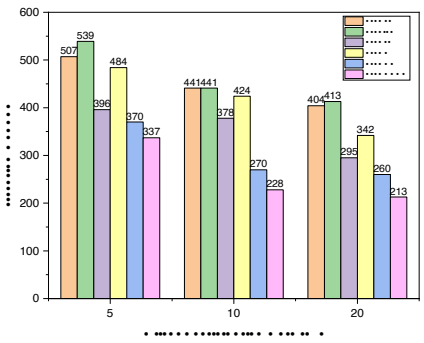

Fig. 5. The number of iteration rounds required to reach a training loss of 0.45 with different levels of client participation.

VI. Conclusion

In this paper, we propose a FedGVRGM algorithm to alleviate the client drift phenomenon caused by data heterogeneity. Unlike recent approaches, FedGVRGM adjusts the model parameters for federated learning in both the local training phase and the global aggregation phase. Firstly, without reducing the length of the local gradient, FedGVRGM reduces the degree of client drift by the gradient variance reduction method. Secondly, FedGVRGM reduces the oscillation of the global model parameters and speeds up the convergence of the global model through the global gradient momentum descent algorithm. We experimentally demonstrate the necessity for two-phase tuning. By comparision with the baselines, it is demonstrated that FedGVRGM is able to achieve faster convergence consistently on various heterogeneity of data.

VII. Acknowledgement

This work is supported by The Key Research and Development Program of China (No. 2022YFC3005400), Key Research and Development Program of China, Yunnan Province (No. 202203AA080009), The National Natural Science Foundation of China (No. 61902110),Transformation Program of Scientific and Technological Achievements of Jiangsu Provence (No. BA2021002), the Key Research and Development Program of China, Jiangsu Province (No. BE2020729), the Key Technology Project of China Huaneng Group (Grant No. HNKJ20-H46) and Fundamental Research Funds for The Central Universities (No. B210203024).

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp.1273-1282, 2017.

[2] M. Mohri, G. Sivek and A.T. Suresh, " Agnostic federated learning," *International Conference on Machine Learning*, pp. 4615–4625, 2019.

[3] T. Li, A.K. Sahu, A. Talwalkar, and V. Smith, " Federated learning: challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 1, pp. 50–60, January 2020.

[4] P. Kairouz, H.B. McMahan, B. Avent, A. Bellet and et al, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1-2, pp. 1-210, 2021.

[5] S. Karimireddy, S. Kale, M. Mohri, J. Sashank, Reddi, U.S. Sebastian, and A.T. Suresh, "Scaffold: stochastic controlled averaging for federated learning," *Proceedings of the 37th International Conference on Machine Learning*, pp. 5132-5143, July 2020.

[6] K. Hsieh, A. Phanishayee, O. Mutlu and P. Gibbons, "The non-iid data quagmire of decentralized machine learning," *Proceedings of the 37th International Conference on Machine Learning*, pp. 4387-4398, July 2020.

[7] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, Austin. TX. USA, March 2020.

[8] Shiqiang, T. Tuor, T. Salonidis, K. KLeung, C. Makaya, H. Ting, and C. Kevin, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 1, pp. 1205–1221, January 2019.

[9] J. Wang, Q. Liu, H. Liang, G. Joshi and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in Neural Information Processing Systems*, Decembe 2020.

[10] X. Li, K. Huang, W. Yang, S. Wang and Z. Zhang, "On the convergence of fedAvg on Non-IID data," *8th International Conference on Learning Representations*, Addis Ababa, Ethiopia, April 2020.

[11] X. Liang, S. Shen, J. Liu, Z. Pan, E. Chen and Y. Cheng," Variance reduced local sgd with lower communication complexity," *arXiv preprint arXiv*:1912.12844, 2019.

[12] C. Xu, Z. Hong, M. Huang and T. Jiang, "Acceleration of federated learning with alleviated forgetting in local training," International Conference on Learning Representations, 2022.

[13] J. Sashank, Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konecny, S. Kumar and H.McMahan, "Adaptive federated optimization," *9th International Conference on Learning Representations*, Virtual Event, Austria, May 2021.

[14] J. Wang, V. Tantia, N. Ballas and M. Rabbat, "Slowmo: improving communication-efficient distributed SGD with slow momentum," *8th International Conference on Learning Representations*, April 2021.

[15] A. Wang, Y. Zhang and Y. Yan, "Heterogeneous defect prediction based on federated transfer learning via knowledge distillation," *IEEE Access* vol. 9, pp. 29530-29540, 2021.

[16] Z. Zhu, J. Hong and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," *Proceedings of the 38th International Conference on Machine Learning*, pp. 12878-12889, July 2021.

[17] Q. Li, B. He, M. Mohri and D. Song, "Model-contrastive federated learning," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10713-10722, June 2021.

[18] T. Yoon, S. Shin, S. Hwang and E. Yang," Fedmix: approximation of mixup under mean augmented federated learning," *9th International Conference on Learning Representations*, Virtual Event, Austria, May 2021.

[19] D. Acar, Y. Zhao, R. Matas, M. Mattina, P.Whatmough and V. Saligrama, "Federated learning based on dynamic regularization," *9th International Conference on Learning Representations*, May 2021.

[20] X. Zhang, M. Hong, S. Dhople, W. Yin and Y. Liu, "FedPD: a federated learning framework with adaptivity to Non-IID data," *IEEE Transactions on Signal Processing*, vol. 69, pp. 6055-6070, 2021.

[21] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin and V. Chandra,"Convergence of federated learning over a noisy downlink," *IEEE Transactions on Wireless Communications*, vol. 21 no. 1, pp. 1422-1437, January 2022.

[22] H. Gao, A. Xu and H. Huang, "On the convergence of communication-efficient local SGD for federated Learning," *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pp. 7510-7518, February, 2021.

[23] T. Li, A. K. Sahu, M. Zaheer, Ma. Sanjabi, A. Talwalkar and V. Smith, "Feddane: a federated newton-type method," *53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 1227-1231, November 2019.

[24] Z. Ma, M. Zhao, X. Cai and Z. Jia:, " Fast-convergent federated learning with class-weighted aggregation," *Journal of Systems Architecture*, vo1. 17, August 2021.

[25] A. Krizhevsky, V. Nair, and G. Hinton,"Learning multiple layers of features from tiny images," *Technical Report*, Citeseer, 2009.

[26] F. Zenke, B. Poole and S. Ganguli, "Continual learning through synaptic intelligence," *International Conference on Machine Learning*, pp. 3987–3995, 2017.

[27] Y. LeCun, L. Bottou, Y.Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 1, pp.2278–2324, 1998.