IMPROVING FEDERATED LEARNING COMMUNICATION EFFICIENCY WITH GLOBAL MOMENTUM FUSION FOR GRADIENT COMPRESSION SCHEMES

Chun-Chih Kuo, Ted Tsei Kuo, Chia-Yu Lin

1 Institute of Computational Intelligence, National Yang Ming Chiao Tung University
2 Department of Computer Science and Information Engineering National Central University
3 AI Lab, Trend Micro

ABSTRACT
Communication costs within Federated learning hinder the system scalability for reaching more data from more clients. The proposed FL adopts a hub-and-spoke network topology. All clients communicate through the central server. Hence, reducing communication overheads via techniques such as data compression has been proposed to mitigate this issue. Another challenge of federated learning is unbalanced data distribution, data on each client are not independent and identically distributed (non-IID) in a typical federated learning setting. In this paper, we proposed a new compression compensation scheme called Global Momentum Fusion (GMF) which reduces communication overheads between FL clients and the server and maintains comparable model accuracy in the presence of non-IID data.

Index Terms— Federated Learning, Communication Overheads, Gradient Compression, Memory Mechanism, Global Momentum Fusion

1. INTRODUCTION
Federated learning, proposed by McMahan et al., provides a new paradigm to enhance the privacy-promised learning framework without moving data outside the devices. Federated learning allows clients to train models with personal data on their devices and only share intermediate gradients with the central server. The server aggregates these gradients (typically by average) and transmits aggregated gradient back to clients. Finally, clients update their models as the new base model for the next round.

While clients communicate through the central server, the computation and communication of the server have become system overheads. Reducing communication overheads via techniques such as gradient compression has been proposed to mitigate this issue. Most compression techniques are considered lossy. Thus, some compensation techniques are usually required to adopt memory mechanism. For example, Lin et al. proposed Deep Gradient Compression (DGC) to achieve a high compression ratio via well compensation design. However, DGC did not consider the non-IID problem discussed below. Another challenge of Federated Learning is unbalanced data distribution. Unlike centralized distributed learning, which separates data with balance distribution, data on each client are not independent and identically distributed (non-IID) in a typical Federated Learning setting. Several approaches have been proposed using global momentum methods to mitigate this problem, such as Sparse Gradient Compression and Global Momentum Compression. However, these solutions further increased communication overheads and decreased accuracy. Although these existing methods showed fine in performance in different consideration, thus our main motivation is to design a scheme which cloud address both communication overheads and non-IID data.

This paper proposes two main contributions. First, we analyze two main communication overheads cost with the existing method. We theoretically show that some methods using momentum could lead to additional communication overheads. Second, we proposed a new compression compensation scheme called Global Momentum Fusion (GMF) which reduces communication overheads between Federated learning clients and the server and maintains comparable model accuracy in the presence of non-IID data.

2. PROBLEM FORMULATION
Based on our study, we illustrate two major problems of existing approaches.

2.1. Server-Side Global Momentum Leads to Extra Communication Overheads
Communication overheads comprise two main traffic costs. (i) clients upload gradients to the server to aggregate, and (ii) the server transmits the aggregated gradient to the clients. The size of the gradient upload from clients has a fixed size. However, the size of the aggregated gradient could be varied since the gradients collected from clients are inconsistent.

Figure illustrate the aggregate process with global momentum. Several works, like FetchSGD and GSGM...
have used global momentum on the server that gives averaged gradients. As training goes, momentum accumulate aggregated gradient each round, which making aggregated gradient nearly full size in the future rounds.

![Fig. 1. Central server with global momentum.](image)

In Figure 1, $T$ is total communication rounds where $t \in T$, $k$ is client where $k \in K$ and $\hat{G}$ is aggregated gradient.

### 2.2. Less Efficiency with Client-Side Global Momentum Compensation

Zhoa et al. [5] took a different approach and proposed Global Momentum Compression (GMC) mechanism. GMC used global momentum to replace local momentum and achieved comparable performance. However, GMC does not consider the variance between the local gradient and the global momentum. A large variance could make the global momentum less efficient and lead to over-fitting local data, especially under low compression rates and non-IID datasets. The sizes of the gradients could have a significant disparity as the progress of each training round. Wang et al. [8] discussed a similar issue because of the objective inconsistency of the data.

### 3. GLOBAL MOMENTUM FUSION COMPRESSION SCHEME

Before introducing our design of the compression process. We first dig into the major sparse compressor. The top-K sparse compression method was commented applied in many sparsification gradient compression approaches. Typically, the top-K sparse compressor creates a mask by selecting the top $k\%$ of the most significant changing parameters of the gradient, representing the top important feature of local data on the client.

$$G_{k,t} \leftarrow V_{k,t} \odot \text{Mask}_{\text{top-K}}(V_{k,t})$$  (1)

We further design an reference of mask by adding an reference vector $F$ into top-K. In theory, these two gradients, $G^{\text{transmit}}$ (green arrow) and $G^{\text{accumulate}}$ (yellow arrow) in Figure 2 are orthogonal because $G^{\text{transmit}}$ is masked by a mask, and $G^{\text{accumulate}}$ is the rest of $V$. This inspired us to design the term $F$ add with $V$ based on this property to re-weight the reference of the mask generation. Although we add a term $F$ with $V$ as shown in Figure 2, it still stratifies the mathematical property. $F$ could be any vector that takes part in the compression selection and still satisfies the orthogonal.

![Fig. 2. Three examples of term $F$ in Top-K compressor](image)

We formulated our Global Momentum Fusion compression by add a fusion layer before top-K sparse compression shown in the following equation:

$$Z_{k,t} \leftarrow \text{abs}((1 - \tau) \cdot N(V_{k,t}) + \tau \cdot N(M_{k,t}))$$  (2)

$$G_{k,t} = V_{k,t} \odot \text{Mask}_{\text{top-K}}(Z_{k,t})$$

In Equation 1 and 2, $V_{k,t}$ is an local compensated gradient, $G_{k,t}$ is compressed gradient, $M_{k,t}$ is the accumulated global momentum, $N$ is Normalize, $\alpha$ is the local momentum factor, $\beta$ is the global momentum factor and $\tau$ is the fusion ratio.

Additionally, inspired by FedNova [8], we introduced a normalized weighting method of fusion ratio to avoid domination by the local gradient and the global momentum.

Next, we proposed a Deep Gradient Compression with Global Momentum Fusion (DGCwGMF) method. We adopted momentum correction from DGC as our memory compensation mechanism and implemented Global Momentum Fusion scheme in the compression policy. DGCwGMF is operates as Figure 3 and Algorithm 1 illustrates the pseudocode of DGCwGMF.

![Fig. 3. DGCwGMF](image)

With the Global Momentum Fusion, we refer to the long-term momentum direction while compressing the gradient, which gives a configurable trade-off between the local gradient and the global momentum with the fusion ratio $\tau$. Before weighing them, we normalize the gradient to avoid bias caused by large variances. A smaller $\tau$ could allow the compressor selection to fit exactly to its local training data. A larger $\tau$ could waive the parameters of the gradient that differ from the global momentum. When setting $\tau > 0$, a higher similarity of the transmitted gradient between clients could have a lower expected communication overhead. When we set the fusion ratio $\tau = 0$, DGCwGMF degenerates into DGC.
Algorithm 1 DGCwGMF on client k

1: initialize $V_{k,t-1} \leftarrow 0$, $G_{k,t} \leftarrow 0$, $M_{k,t} \leftarrow 0$
2: Initial base model $W_{k,0} = W_{init}$ on server and share with all clients.
3: for $t = 0 \ldots T$ do
4: Sample batches of data $X_k$ from local dataset
5: $\nabla_{k,t} = SGD(W_{k,t}, X_k)$ (local update)
6: $U_{k,t} \leftarrow \alpha U_{k,t-1} + \nabla_{k,t}$ \quad $\triangleright$ Momentum Correction
7: $V_{k,t} \leftarrow V_{k,t-1} + U_{k,t}$ \quad $\triangleright$ Momentum Correction
8: $M_{k,t} \leftarrow \beta M_{k,t-1} + G_{k,t}$ \quad $\triangleright$ Global momentum accumulate
9: $Z_{k,t} \leftarrow \text{abs}((1 - \tau) \cdot N(V_{k,t}) + \tau \cdot N(M_{k,t}))$ \quad $\triangleright$ GMF
10: $G_{k,t} \leftarrow V_{k,t} \odot \text{Mask}_{topk}(Z_{k,t})$ \quad $\triangleright$ Memory update
11: $U_{k,t} \leftarrow U_{k,t} \odot (1 - \text{Mask}_{topk}(Z_{k,t}))$ \quad $\triangleright$ Memory update
12: Transmit $G_{k,t}$ to server
13: Download aggregated $\tilde{G}_t$ from server
14: Update local model $W_{k,t+1} \leftarrow W_{k,t} - \eta_t \cdot \tilde{G}_t$
15: end for

4. EXPERIMENT VALIDATION

We validate our approach with two training tasks: the image classification task in the Cifar10 dataset and the next-word prediction task in the Shakespeare dataset. To compare image classification and next-word prediction tasks, we provide a reproducible empirical experiment over two datasets containing an artificial non-IID dataset and a naturally non-IID dataset.

4.1. Setup

| Task | Image Classification | Next-Word Prediction |
|------|----------------------|----------------------|
| Dataset | Cifar10 | Shakespeare (non-IID) |
| Model | ResNet56 | RNN (single layer LSTM) |
| # of clients | 20 | 20 |
| # of rounds | 220 | 80 |

Table 1. Summary of tasks

Table [1] show our setting for the two task. Following the procedure in [3], we generate 7 different Mod-Cifar10 datasets with the following Earth Moving Distance(EMD) [9]: 0.0 (Cifar10-0), 0.48 (Cifar10-1), 0.76 (Cifar10-2), 0.87 (Cifar10-3), 0.99 (Cifar10-4), 1.18 (Cifar10-5) and 1.35 (Cifar10-6) in 20 clients to simulate realistic scenario. The EMD of the sampled clients of Shakespeare dataset is 0.1157. We set the fusion ratio $\tau$ to start from 0 and step increase to 0.6 in 10 steps. High EMD values indicate high unbalanced data. EMD=0 represents that data is balance distributed.

| Technique | Momentum Correction | Client-side Global Momentum | Server-side Global Momentum |
|-----------|---------------------|-----------------------------|-----------------------------|
| DGC       | $\triangleright$    | $v$                        | $v$                        |
| GMC       | $\triangleright$    | $v$                        | $v$                        |
| DGCwGM    | $\triangleright$    | $v$                        | $v$                        |
| DGCwGMF   | $\triangleright$    | $v$ (in compensation process) | $v$ (in compensation process) |

Table 2. Techniques in our experiments

To analyze the performance in the term of accuracy and communication overheads with existing approaches and our work. Our experiment compares the following sparse compression methods: DGC is the method proposed by Lin Yujun et al. [3] with the momentum correction technique. We compare the current client-side global momentum work, GMC[5]. We use the same DGC algorithm to compare communication consumption and apply the server-side global momentum name as DGCwGM. Finally, DGCwGMF is the DGC method that applies the Global Momentum Fusion (GMF) scheme. Table 2 summarizes the comparison of these techniques.

4.2. Results of Image Classification Task

We first compare the accuracy and communication overheads of DGCwGMF and DGCwGM in Table 3. Both DGCwGMF and DGCwGM show similar accuracy. However, DGCwGM consumes more communication overheads due to the accumulated server-side global momentum, which matches our problem formulation 2.1.

When overview our experiments, GMC reaches comparable accuracy when $EMD < 0.99$. However, in the case of compression rate=0.1 with Cifar10-6 dataset, GMC overfitted the local dataset at the end of training and degraded the performance of the model under training as shown in Figure 4. This issue can also match our problem formulation 2.2.

| Dataset | Technique | Top 1 Accuracy | Communication Overheads |
|---------|-----------|----------------|-------------------------|
| Cifar10-0 (EMD=0) | DGC (Baseline) | 0.8146 | 3.53 |
| | GMC | 0.8097 -0.0049 | 3.15 -0.38 |
| | DGCwGM | 0.6238 -0.1908 | 4.10 +0.57 |
| | DGCwGMF | 0.8075 -0.0071 | 3.02 -0.51 |
| Cifar10-1 (EMD=0.48) | DGC (Baseline) | 0.8216 | 3.57 |
| | GMC | 0.8216 -0.1142 | 3.25 -0.32 |
| | DGCwGM | 0.7074 -0.0121 | 4.12 +0.55 |
| | DGCwGMF | 0.8095 -0.0121 | 2.90 -0.67 |
| Cifar10-2 (EMD=0.76) | DGC (Baseline) | 0.8134 | 3.56 |
| | GMC | 0.7975 -0.0159 | 3.29 -0.27 |
| | DGCwGM | 0.7242 -0.0892 | 4.12 +0.56 |
| | DGCwGMF | 0.8078 -0.0056 | 2.83 -0.73 |
| Cifar10-3 (EMD=0.87) | DGC (Baseline) | 0.8000 | 3.56 |
| | GMC | 0.7834 -0.0166 | 3.29 -0.27 |
| | DGCwGM | 0.7218 -0.0782 | 4.12 +0.56 |
| | DGCwGMF | 0.7970 -0.0030 | 2.83 -0.73 |
| Cifar10-4 (EMD=0.99) | DGC (Baseline) | 0.7915 | 3.56 |
| | GMC | 0.7711 -0.0204 | 3.29 -0.27 |
| | DGCwGM | 0.7226 -0.0689 | 4.12 +0.56 |
| | DGCwGMF | 0.7920 +0.0005 | 2.83 -0.73 |
| Cifar10-5 (EMD=1.18) | DGC (Baseline) | 0.7712 | 3.56 |
| | GMC | 0.7146 -0.0566 | 3.31 -0.25 |
| | DGCwGM | 0.7261 -0.0451 | 4.12 +0.56 |
| | DGCwGMF | 0.7706 -0.0006 | 2.83 -0.73 |
| Cifar10-6 (EMD=1.35) | DGC (Baseline) | 0.7047 | 3.57 |
| | GMC | 0.5620 -0.1427 | 3.34 -0.23 |
| | DGCwGM | 0.7087 +0.0004 | 4.12 +0.55 |
| | DGCwGMF | 0.7246 +0.0199 | 2.84 -0.73 |

Table 3. Results of Image Classification Task on compression rate=0.1
0.7246, and 0.7087, respectively. When comparing the communication overhead, Table 3 shows that DGCwGMF costs 15.4% more communication overhead than DGC. Moreover, our approach, DGCwGMF, saves 20.4% of communication overhead compared with DGC and reaches higher accuracy, which proves GMF could actually save more communication overhead and less accuracy loss or improve accuracy.

In Next-Word Prediction Task, Table 4 shows that the DGCwGM saves 4% communication overhead over compression rate on Cifar10-6. In Figure 5, the experiments result show that all techniques lose performance with the compression rate decrease. However, DGCwGMF maintains less performance loss under a low compression rate. DGCwGMF also decreases faster than other approaches on the communication overhead curve, making it ideal for further compression.

Finally, we provide experiments of accuracy and communication overhead over different compression rate from 0.1 to 0.9 on Cifar10-6. In Figure 3, the experiments result show that all techniques lose performance with the compression rate decrease. However, DGCwGMF maintains less performance loss under a low compression rate. DGCwGMF also decreases faster than other approaches on the communication overhead curve, making it ideal for further compression.

Fig. 4. Top-1 Accuracy curves on Cifar10-6

Fig. 5. Comparison of Test Accuracy and communication overhead over compression rate on Cifar10-6

4.3. Results of Next-word Prediction Task

In Next-Word Prediction Task, Table 4 shows that the DGCwGM has more communication overhead consumption than DGC, which also proves our problem formulation that momentum at the server-side requires more communication overhead consumption. Next, the table also show that GMC fails to converge. However, the other work, DGC, DGCwGMF, and DGCwGM, have very close accuracy.

| Dataset       | Technique  | Top 1 Accuracy | Communication Overhead |
|---------------|------------|----------------|------------------------|
| Shakespeare   | DGC (Baseline) | 0.419          | 2.72                   |
| (EMD=0.1157)  | GMC        | 0.347 (-0.072) | 1.95 -0.77             |
|               | DGCwGM     | 0.421 (+0.002) | 3.02 +0.3              |
|               | DGCwGMF    | 0.419          | 2.38 -0.34             |

Table 4. Result of Next-Word Prediction Task on compression rate=0.1

Besides GMC, the result of DGC, DGCwGMF, and DGCwGM under compression rate=0.1 also shows that DGCwGMF consumes less communication overhead and reaches only 0.002 accuracy loss compared with DGCwGM. Table 4 shows that DGCwGM costs 11% more communication overhead than DGC, and DGCwGMF saves 12.5% more communication overhead than DGC.

Fig. 6. Comparison of Test Accuracy and communication overhead over compression rate on Shakespeare

5. CONCLUSION

We proposed a gradient compression with a compensation scheme called Deep Gradient Compression with Global Momentum Fusion (DGCwGMF) to reduce FL communication overhead while maintaining comparable model performance despite non-IID data. We designed and conducted comparative experiments to prove the proposed DCGwGMF provides the best performance with low compression rates in high EMD datasets. Specifically, our experiments showed that DGCwGM has less accuracy loss in the non-IID dataset and 20.4% fewer communication overheads than DGC in the image classification task. DGCwGMF reduces 12.5% of communication overheads and maintains the same accuracy as vanilla DGC for speech recognition tasks.
6. REFERENCES

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agueray Arcas, “Communication-efficient learning of deep networks from decentralized data,” in Artificial intelligence and statistics. PMLR, 2017, pp. 1273–1282.

[2] Hang Xu, Chen-Yu Ho, Ahmed M Abdelmoniem, Aritra Dutta, El Houcine Bergou, Konstantinos Karatsenidis, Marco Canini, and Panos Kalnis, “Compressed communication for distributed deep learning: Survey and quantitative evaluation,” Tech. Rep., 2020.

[3] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally, “Deep gradient compression: Reducing the communication bandwidth for distributed training,” in International Conference on Learning Representations, 2018.

[4] Alham Aji and Kenneth Heafield, “Sparse communication for distributed gradient descent,” in EMNLP 2017: Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (ACL), 2017, pp. 440–445.

[5] Shen-Yi Zhao, Yin-Peng Xie, Hao Gao, and Wu-Jun Li, “Global momentum compression for sparse communication in distributed sgd,” arXiv preprint arXiv:1905.12948, 2019.

[6] Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora, “Fetchsgd: Communication-efficient federated learning with sketching,” in International Conference on Machine Learning. PMLR, 2020, pp. 8253–8265.

[7] Chengjie Li, Ruixuan Li, Haozhao Wang, Yuhua Li, Pan Zhou, Song Guo, and Keqin Li, “Gradient scheduling with global momentum for non-iid data distributed asynchronous training,” arXiv preprint arXiv:1902.07848, 2019.

[8] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor, “Tackling the objective inconsistency problem in heterogeneous federated optimization,” Advances in neural information processing systems, vol. 33, pp. 7611–7623, 2020.

[9] 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.