FONN: Federated Optimization with Nys-Newton

C Nagaraju
Dept. of Computer Science
Indian Institute of Technology
Hyderabad, India
cs17resch01002@iith.ac.in

Mrinmay Sen
Dept. of Artificial Intelligence
Indian Institute of Technology
Hyderabad, India
ai20resch11001@iith.ac.in

C Krishna Mohan
Dept. of Computer Science
Indian Institute of Technology
Hyderabad, India
ckm@cse.iith.ac.in

***Abstract*—Federated optimization or federated learning (FL) involves optimization of the global model or the server model by minimizing the global loss function which is weighted average of all the local loss functions. The optimization of the global model requires faster convergence to reduce the number of communication rounds or global iterations which is one of the major challenge in federated optimization. This paper propose FONN which handles this communication overhead in federated optimization by utilizing Nys-Newton, while updating local models. As compared to existing state-of-the-art FL algorithms, SCAFFOLD, GIANT and DONE, utilization of Nys-Newton leads to better convergence and reduction in communication rounds or global iterations while achieving a desired performance from the global model which may be observed from the experimental results on various heterogeneously partitioned datasets.**

***Index Terms*—Federated Optimization, Communication overhead, Faster convergence, Nys-Newton**

I. Introduction

Federated learning (FL) or federated optimization [1] is one kind of distributed learning algorithm, where multiple clients or data sources collaboratively train a global model or server model by sharing their locally trained model to the server instead of raw data. The objective of FL is to optimize the global model parameters by minimizing weighted average of all the local loss functions. One communication round of federated learning involves sharing server model to all the participating (or available) clients and collecting locally trained models from the clients to the server. These communications are the major issue with federated learning which needs to be minimized. To mitigate this issue, several works have been done which use either first-order optimization or second-order Newton method of optimization.

FedAvg [1], the baseline of federated learning algorithms, finds locally updated models with the help of first-order stochastic gradient descent optimizer (SGD) [2] and aggregates these models in the server to find the global model. FedAvg performs well when data are homogeneously distributed. With heterogeneous data partitions, FedAvg suffers from objective inconsistency [3], [4] which means that the global loss function (weighted average of all the local loss functions) is minimized at a stationary point which is away from the true optima. FedProx [5], FedNova [6], SCAFFOLD [7], FedDC [8], MOON [9] etc. are existing first-order based FL algorithms which are invented for mitigating the problem of FedAvg. FedProx adds a proximal term with the local loss function to control the direction of the stochastic gradient while updating local models. FedNova utilizes normalized weights while aggregating all the local models in the server. SCAFFOLD uses control variates to handle data heterogeneity, FedDC uses auxiliary local drift variable which helps to bridge the gap between the local and global model parameters. MOON conducts contrastive learning in model-level. Through these algorithms perform better than FedAvg in heterogeneous system, their convergence rate is still slow as these algorithms use only first-order gradient while updating local models.

To further increase convergence of federated learning, researchers focus on another direction of optimization which is based on second-order Newton method [10] where Hessian curvature information is incorporated along with gradient while updating model parameters, which leads to faster convergence or reduced communication rounds in federated learning. The main challenges of using Newton method in federated learning are the computation and storing of Hessian matrix and its inverse which may be difficult for large scale application as the space & time complexities of calculating and storing Hessian are both O(d^2) and the space & time complexities of calculating and storing inverse of Hessian are O(d^2) and O(d^3) respectively [11], [12]. These burdens of storing and computation of Hessian and its inverse motivate us to use approximated Hessian instead of true Hessian. Existing Newton method based FL algorithms include DANE [13], DiSCO [14], GIANT [15], FedSSO [16], DONE [17], FedNL [18] etc.

DANE, DiSCO, GIANT and DONE use global gradient (average of all local gradients) to approximate local newton direction. GIANT uses conjugate gradient method [19] to approximate the local newton direction and takes harmonic mean of all local Newton updates to find global Newton update. DANE finds a mirror descent update on the local loss function. For a quadratic loss function, this descent update is same as GIANT update. DONE uses Richardson iteration for local update. DiSCO finds local Hessian vector products and communicate these to server and performs preconditioned conjugate gradient method to approximate global Newton direction. FedSSO finds global update in server by using Quasi-Newton method on average of local gradients. Even, FedSSO is more communication efficient that DANE, DiSCO, GIANT and DONE, the requirement of storing full Hessian matrix in the server is the major drawback of FedSSO for learge scale settings. FedNL stores previous step's Hessian matrix to approximate current step's Hessian. Storing, calculation and compression of local Hessian results in additional computational load to the local clients.

Our proposed method, FONN, aims to further increase convergence rate or to further reduce the communication rounds in FL by efficiently incorporating Hessian curvature

979-8-3503-0219-6/23/$31.00 ©2023 IEEE

information without storing it. Same as GIANT, FONN utilizes global gradient and harmonic mean of local Newton updates to find the global update. To approximate local Newton update, FONN utilizes Nys-Newton [11] which has linear-time space and time complexities. As compared to existing state-of-the-art FL algorithms, SCAFFOLD, GIANT and DONE, utilization of Nys-Newton leads to better convergence and reduction in communication rounds while achieving a desired performance from the global model which may be observed from the experimental results on various heterogeneously partitioned datasets. Our proposed FONN can achieve better performance while maintaining same local time complexities as GIANT and DONE.

II. Problem formulation

The objective of federated learning is to find the optimized parameters of the global model $w \in R^d$ by minimizing the global loss function F(w) which is average of all K local loss functions $\{F_i(w)\}$, where i $\in \{1, 2, ..., K\}$ and K is number of clients participated in federated learning. The i^th client P_i has its own dataset D_i.

$$\min_w F(w) = \frac{1}{K} \sum_{i=1}^{K} F_i(w, D_i) \tag{1}$$

Where $F_i(w)$=$\frac{1}{|D_i|} \sum_{j=1}^{|D_i|} F_j(w, x_j \in D_i)$

III. Preliminaries

A. Newton method of optimization

Newton method of optimization scales the gradient g_t with the help of inverse of Hessian $H^{d \times d}$ while updating model parameters as shown in equation- 2

$$w_{t+1} = w_t - H_t^{-1} g_t \tag{2}$$

The main challenges associated with Newton method of optimization are the calculation and storing of Hessian and its inverse as the time & space complexities of Hessian matrix are both O(d^2) and time & space complexities of inverse Hessian matrix are O(d^3) & O(d^2) respectively which may be difficult to implement for learge scale applications.

B. Federated Learning

In federated learning, instead of collecting raw data from the clients, locally trained models $\{w_t^i\}$ are collected and then aggregated for finding global model w_{t+1}. Initially, the server shares a common global model w_t to all the clients and each client separately updates this global model with the help of their local optimizer, local data (D_i) and local gradient (g_t^i). Server then receives all the locally updated models $\{w_t^i\}$ and aggregates these to find the global model w_{t+1}.

$$w_{t+1} = \frac{1}{K} \sum_{i=1}^{K} w_t^i = w_t - \frac{\eta}{K} \sum_{i=1}^{K} g_t^i \tag{3}$$

C. Harmonic mean of local Hessians

While aggregating local models, first-order based FL algorithms use arithmetic mean over all the client's models as the locally updated models involve scaling the gradient with a learning rate or step size (η) parameter as shown in equation-3. But if the local models are updated by using Newton method (equation- 2), the arithmetic mean of the local models contains harmonic mean of local Hessians (which is not same as arithmetic mean of local Hessians [15]) which results in inappropriate aggregation of local models while finding global model. This is one of the major bottleneck of using Newton method of optimization in federated learning [20]. GIANT [15] addresses this issue and proves that the arithmetic mean is nearly same as harmonic mean of local Hessians when the local models are updated with the help of same global gradient (g_t) and the data are incoherent. Our proposed FONN algorithm uses the same concept of GIANT. To efficiently aggregate local models, FONN uses same global gradient for finding all the local updates.

Algorithm 1 FONN

Input: T: Number of FL iterations, w_0: Initial global model, η: learning rate, ρ: Hessian regularization parameter

for $t = 1$ **to** T **do**
- Server sends w_t to all the clients
- Each client P_i receives w_t and finds local gradient g_t^i= $\frac{\partial F_i(w_t)}{\partial w_t}$ and sends this g_t^i to the server
- Server aggregates all the local gradients $\{g_t^i\}$ and finds global gradient g_t = $\frac{1}{K} \sum_{i=1}^{K} g_t^i$ over all the clients.
- Server sends this global gradient g_t to all the clients
- Each client updates w_t with the help of Nyström approximatited Newton direction or Nys-Newton [11] and finds w_{t+1}^i as mentioned in equation-4. Each client uses global gradient g_t instead of local gradient while finding local Newton update.
- Server collects all the locally updated models $\{w_{t+1}^i\}$ and aggregates these to find the global model w_{t+1} = $\frac{1}{K} \sum_{i=1}^{K} w_{t+1}^i$

end for

D. Nys-Newton

Nys-Newton [11] is a variant of Newton method of optimization where the true Hessian is approximated with the help of Nyström approximation on partial column Hessian of size $d \times m$ with $m \ll d$ randomly selected Hessian columns. In nys-Newton, the update direction is calculated directly with out calculating and storing full Hessian matrix which in the key advantage of Nys-Newton. To invert the approximated Hessian, Nys-Newton uses a regularized variant of Newton method as given in equation-4

$$w_{t+1} = w_t - \eta B g \tag{4}$$

Where, B=$(ZZ^T + \rho I_d)^{-1}$, Z = $CU_r\Sigma_r^{-\frac{1}{2}}$, M_r= $U_r\Sigma_r U_r^T$ is the best r rank approximation of M (M is found by taking the intersection of m columns and corresponding m rows of the Hessian), η is learning rate, ρ is Hessian regularization term, I_d is Identity matrix of size d and g is stochastic gradient.

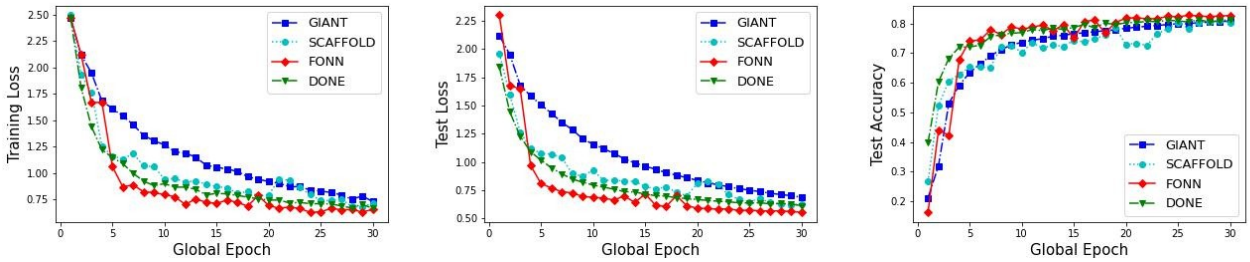

Fig. 1. Comparisons of training loss, test loss and test accuracy on MNIST

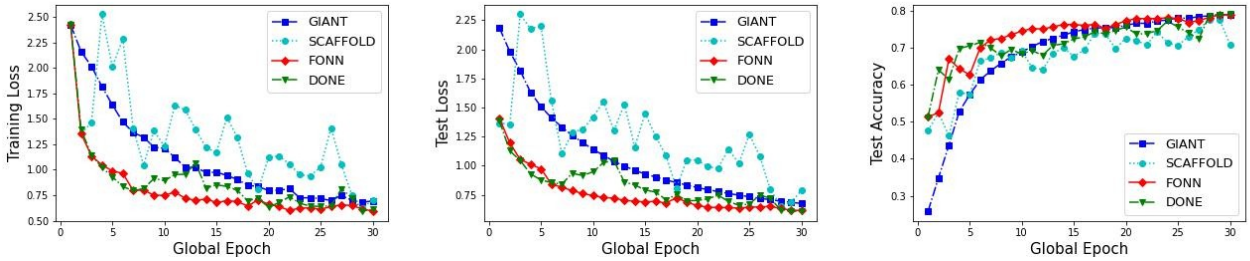

Fig. 2. Comparisons of training loss, test loss and test accuracy on FashionMNIST

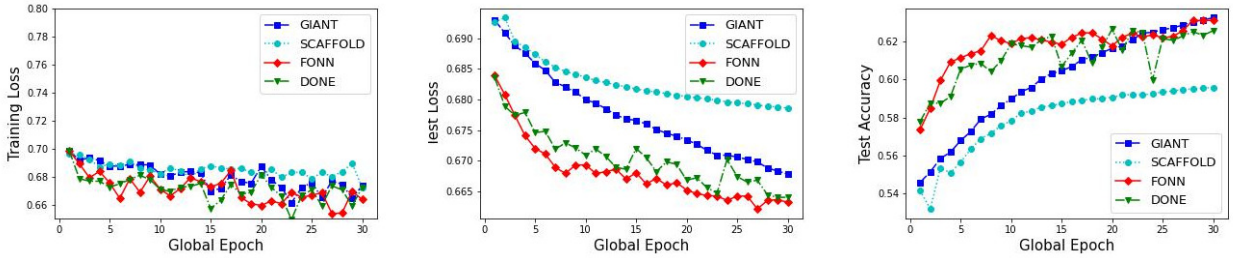

Fig. 3. Comparisons of training loss, test loss and test accuracy on SVHN

By using these Z and ρ, Nys-Newton finds the Newton update $Bg = (ZZ^T + \rho I)^{-1}g = \frac{1}{\rho}g - QZ^Tg$, where $Q = \frac{1}{\rho^2}Z(I_r + \frac{1}{\rho}Z^TZ)^{-1}$ and I_r is Identity matrix of size r. This Bg is then used for updating model parameters as mentioned in equation-4

IV. Proposed FONN

Our proposed FONN is a Newton method based FL algorithm where the local Newton update is approximated by using Nys-Newton [11] and global gradient (g_t). One global iteration of FONN is shown in algorithm-1. In FONN, server sends initial global model w_t to all the clients and finds global gradient g_t across all the clients by collecting and aggregating all the local gradients $\{g_t^i\}$. Each client then receives this global gradient g_t and updates w_t separately with the help of equation-4 of Nys-Newton. While applying Nys-Newton, each client uses their own local data and the global gradient. Use of global gradient while finding local Newton update makes the harmonic mean equivalent to arithmetic mean of local Hessians which helps us to efficiently aggregate local models while finding global model.

A. *Complexities in each global iteration*

In FONN, the local time and space complexities for finding Newton update are both O(md), where m $\ll$ d is number of selected Hessian columns. In GIANT and DONE, the local time and space complexities are O(Rd) and O(d), where R is the number of conjugate gradient iterations or number of Richardson iterations. We compare FONN with GIANT and DONE with same local time complexity i.e. R=m. The server space and time complexities of FONN are same as FedAvg.

V. Experimental Settings

We evaluate the performance of FONN on heterogeneously partitioned MNIST, FashionMNIST and SVHN datasets. We also compare FONN with existing state-of-the-arts FL algorithms, SCAFFOLD, GIANT and DONE. We use multinomial logistic regression (MLR) model with crossentropy loss function for federated multi-class classification tasks. For a clear comparison, we convert original 10 classes SVHN datasaets into binary classes. For pytorch [21] SVHN dataset, we replace original 0, 1, 2, 3, 4 classes with 0 and 5, 6, 7, 8, 9 classes with 1. For each dataset, we use same initialization and same settings for all the methods. To find the best performing model for each method, we conduct extensive experiments with multiple sets of hyperparameters and choose

TABLE I
Comparisons of global iterations for SCAFFOLD, DONE, GIANT and FONN using their optimal hyper-parameters

Targeted test accuracy	DONE	GIANT	SCAFFOLD	FONN
MNIST 80 %	18	24	25	16
FashionMNIST 78 %	28	27	—	24
SVHN 62 %	13	20	—	8

the best performing model by considering global iteration wise minimum train & test losses and maximum test accuracy. We use learning rate $\eta \in \{1, 0.1, 0.01, 0.001, 0.0001\}$, FONN regularization parameter $\rho \in \{0.5, 0.1\}$. To compare FONN with existing Newton method based FL algorithms, GIANT and DONE with the same local time complexity, we use R=10 local iterations for GIANT & DONE and m=10 number of randomly selected Hessian columns for FONN. For DONE, we use $\alpha = 0.01 \leq \frac{1}{R}$. For all the methods, we use full batch while updating local model. To consider partial device participation due to network connectivity issue or internal problem of client's devices, we use 40% clients participation in each global iteration out of total 200 clients. We implement all the methods using Tesla V100 GPU and PyTorch-1.12.1+cu102.

we use the Dirichlet distribution based heterogeneous and unbalanced partition strategy to make heterogeneous data distribution across the clients, which is same as the utilization of the paper of Yurochkin et al. [22]. We simulate $p_i \sim Dir_(0.2)$ and find a heterogeneous partition by allocating a $p_{(j,i)}$ proportion of the samples of j^{th} class to i^{th} client. As we use very small value of Dirichlet distribution's concentration parameter (0.2), each client may not get samples of all the classes which indicates a high degree of data heterogeneity across all the clients. For our experiments, we use K=200 clients with 40% partial participation in each round.

VI. Results

Our experimental results on heterogeneously partitioned MNIST, fashionMNIST and SVHN datasets are shown in figures-1, 2, 3 and table-I. From these figures, it may be observed that in FONN, training and test losses decrease faster than state-of-the-art FL algorithms SCAFFOLD, GIANT and DONE. As we use same initialization and same settings for all the methods, it may be claimed that FONN has faster convergence rate than SCAFFOLD, GIANT and DONE in terms of global iteration wise training and test losses. From the table-I and Test accuracy vs global iterations plots , it may also be observed that FONN requires comparatively less number of global iterations than SCAFFOLD, GIANT and DONE, while gaining a targeted accuracy from the global model which indicates that in heterogeneous data partitions, FONN can deceases number of FL iterations or can improve the convergence of the global model better than existing state-of-the-art FL algorithms such that SCAFFOLD, GIANT and DONE, while achieving a targeted test accuracy from the global model.

VII. Conclusions

This paper proposes FONN to reduce the number of global iterations in federated learning. FONN use global gradient in Nys-Newton while finding local Newton updates and calculates the harmonic mean of local models to get the global model. Experimental results on heterogeneously partitioned MNIST, fashionMNIST and SVHN datasets show that FONN performs better than existing state-of-the-art FL algorithms, SCAFFOLD, GIANT and DONE in terms of requirement of lower number of global iterations while achieving a targeted performance from the global model.

References

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54, pages 1273–1282, 2017.

[2] Nikhil Ketkar. Stochastic gradient descent. In *Deep learning with Python*, pages 113–132. Springer, 2017.

[3] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[4] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *CoRR*, abs/2103.00710, 2021.

[5] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In Inderjit S. Dhillon, Dimitris S. Papailiopoulos, and Vivienne Sze, editors, *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*, 2020.

[6] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[7] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR, 2020.

[8] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10102–10111. IEEE, 2022.

[9] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 10713–10722, 2021.

[10] Hong Hui Tan and King Hann Lim. Review of second-order optimization techniques in artificial neural networks backpropagation. In *IOP conference series: materials science and engineering*, volume 495, page 012003. IOP Publishing, 2019.

[11] Hardik Tankaria, Dinesh Singh, and Makoto Yamada. Nys-curve: Nyström-approximated curvature for stochastic optimization. *CoRR*, abs/2110.08577, 2021.

[12] Naman Agarwal, Brian Bullins, and Elad Hazan. Second-order stochastic optimization for machine learning in linear time. *J. Mach. Learn. Res.*, 18:116:1–116:40, 2017.

[13] Ohad Shamir, Nathan Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1000–1008. JMLR.org, 2014.

[14] Yuchen Zhang and Xiao Lin. Disco: Distributed optimization for self-concordant empirical loss. In *International conference on machine learning*, pages 362–370. PMLR, 2015.
[15] Shusen Wang, Fred Roosta, Peng Xu, and Michael W Mahoney. Giant: Globally improved approximate newton method for distributed optimization. *Advances in Neural Information Processing Systems*, 31, 2018.
[16] Xin Ma, Renyi Bao, Jinpeng Jiang, Yang Liu, Arthur Jiang, Jun Yan, Xin Liu, and Zhisong Pan. Fedsso: A federated server-side second-order optimization algorithm. *arXiv preprint arXiv:2206.09576*, 2022.
[17] Canh T. Dinh, Nguyen Hoang Tran, Tuan Dung Nguyen, Wei Bao, Amir Rezaei Balef, Bing Bing Zhou, and Albert Y. Zomaya. DONE: distributed approximate newton-type method for federated edge learning. *IEEE Trans. Parallel Distributed Syst.*, 33(11):2648–2660, 2022.
[18] Mher Safaryan, Rustem Islamov, Xun Qian, and Peter Richtárik. Fednl: Making newton-type methods applicable to federated learning. *arXiv preprint arXiv:2106.02969*, 2021.
[19] Magnus R Hestenes, Eduard Stiefel, et al. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952.
[20] Michal Derezinski and Michael W Mahoney. Distributed estimation of the inverse hessian by determinantal averaging. *Advances in Neural Information Processing Systems*, 32, 2019.
[21] Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. Pytorch. *Programming with TensorFlow: Solution for Edge Computing Applications*, pages 87–104, 2021.
[22] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan H. Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7252–7261. PMLR, 2019.