Decentralized Federated Learning for Electronic Health Records

Songtao Lu[†], Yawen Zhang[‡], and Yunlong Wang[*]
[†]IBM Research AI, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 10598, USA
[‡]Department of Electric and Computer Engineering, Hong Kong University of Science and Technology
[*]Advanced Analytic, IQVIA, Plymouth Meeting, Pennsylvania, 19355, USA

2020 54th Annual Conference on Information Sciences and Systems (CISS) 978-1-7281-4085-8/20/$31.00 ©2020 IEEE 10.1109/CISS48834.2020.1570617414

***Abstract*—Federated learning opens a number of research opportunities due to its high communication efficiency in distributed training problems within a star network. In this paper, we focus on improving the communication efficiency for fully decentralized federated learning (DFL) over a graph, where the algorithm performs local updates for several iterations and then enables communications among the nodes. In such a way, the communication rounds of exchanging the common interest of parameters can be saved significantly without loss of optimality of the solutions. Multiple numerical simulations based on large, real-world electronic health record databases showcase the superiority of the decentralized federated learning compared with classic methods.**

***Index Terms*—decentralized federated learning (DFL), communication efficiency, health record databases, heterogeneous networks, non-convex optimization**

I. Introduction

In this era of big data, the use of aggregated patient information can effectively train a high-quality machine learning model by adopting multiple computational resources. However, there are several challenges to this exercise. First, data privacy and security are paramount, and they are often difficult to integrate the data collected and aggregated across. Second, communication efficiency presents a challenge, as each communication round may result in long delays especially in applications of the internet of things (IoT) or self-driving systems. To overcome these challenges, federated learning (FL) may be an effective way to increase training efficiency and allow knowledge to be shared without compromising user privacy [1].

A. Motivation

Decentralized federated-learning (DFL) techniques are promising across numerous applications, such as smart healthcare, etc. Medical data such as disease symptoms and medical recordings are highly sensitive, and collecting clinical datasets from isolated medical centers and hospitals is a challenge. Federated learning, enabling multiple agents collaboratively learn a shared prediction model while keeping all the training data private, could play a pivotal role in solving this problem [2], [3].

This work was done when S.L. was with the University of Minnesota Twin Cities.

Patient data is fully decentralized in most real-world applications and patient-level data exchange among stakeholders such as insurance companies and treating facilities is prohibited by laws, such as the United States Health Insurance Portability and Accountability Act (HIPPA) [4]. Therefore, hospitals have hundreds of patient-level records in one disease area, describing the characteristics of every patient, but lack the breadth of the information about the patients; with these limited samples, a complex model cannot be trained by one hospital.

Nevertheless, under an agreement, each hospital is allowed to share non-sensitive intermediate statistics that are strictly de-identified and aggregated [5]–[7]. In this setting, the hospitals constitute an undirected network where each hospital is a node, and an only neighboring node can exchange information. Also, note that the data are non-identical, since the hospitals are located in different areas and the environmental factors have much impact on people's health status. We will show in this paper that by implementing decentralized iterative optimization algorithm, every node will reach the consensus optimality as if it owns all the data as a fictitious fusion center. Here we would like to emphasize that the studied application is decentralized rather than distributed with a star network, as it is infeasible to have a fusion center that is trusted by every node to collect healthcare data.

B. Scope of This Work

In practice, transmitting messages over networks requires much more effort and spending resources compared with local computation, such as encryption, coding/decoding, channel equalization, etc. Therefore, it is of interest of performing local update to learn the models. The current federated learning strategies are mainly performed over a star network [2], [8], [9] through applying the traditional distributed optimization algorithms, such as distributed (stochastic) gradient descent [10], [11]. By adopting a central controller or parameter server, the slave nodes implement multiple rounds of local updates and then communicate with the master node such that a large amount of the communication rounds among the nodes can be saved. It has been shown in [9] that there are only $\mathcal{O}((NT)^{3/4})$ number of communication rounds required instead of $\mathcal{O}(T)$ in the classic decentralized non-convex setting for the non-identical datesets, where N denotes the total number of nodes and T stands for the total number of iterations.

978-1-7281-4085-8/20/$31.00 ©2020 IEEE

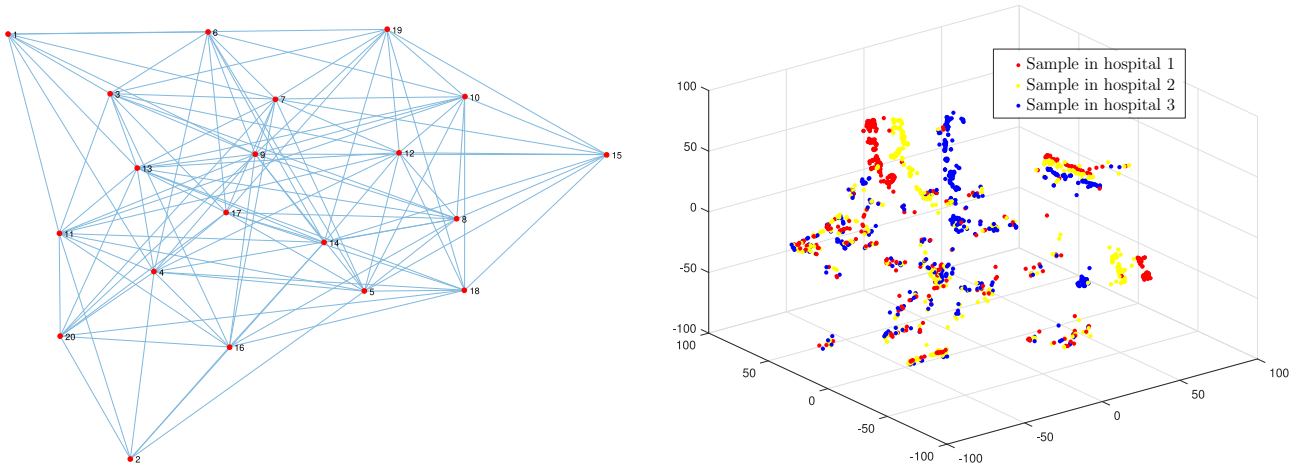

Fig. 1. Description of the real health records: (Left) graph of the nodes (hospitals); (Right) t-SNE distribution of the samples in three nodes (hospitals) in the Alzheimer patients' dataset.

In this work, we propose a fully decentralized federated learning framework by leveraging two classic non-convex decentralized optimization, which are decentralized stochastic gradient descent (DSGD) [12], [13] and decentralized stochastic gradient tracking (DSGT) (a.k.a. Gradient-tracking based Non-convex Stochastic Decentralized (GNSD) algorithm [14], [15]). We remark that DSGT has the advantages of dealing with non-identical datasets compared with DSGD. First, we will introduce the proposed communication efficient decentralized training algorithm for federated learning. Then, we show the linear speedup of DSGT by quantifying its convergence rate to the first-order stationary points theoretically. Third, we numerically compare the decentralized federated learning algorithms with the classic counterparts which do not consider communication efficiency. To the best of our knowledge, this is the first work that applies fully decentralized non-convex stochastic algorithms for federated learning and obtains reasonably good results for health record datasets.

II. Decentralized Stochastic Non-convex Federated Learning

A. Dataset Description

Data and pre-processing: we test our algorithm on a proprietary clinical dataset that consists 2,103 patients diagnosed as Alzheimer's Disease (AD), and 7, 919 patients diagnosed with mild cognitive impairment (MCI), who have gotten early stage symptoms of AD. The electronic health records of all the patients are collected from 20 hospitals, about 500 recordings per each. The graph of all 20 hospitals is shown on the left in Fig. 1. And the figure on the right in Fig. 1 gives an example of the t-SNE distribution of the samples of three hospitals. The separated distributions of different hospitals indicates the heterogeneity of the data in nature, which has been rarely addressed by previous federated system [2], [16]. These figures motive us to develop efficient algorithms of being able to handle non-identical dataset in a decentralized setting.

B. Problem formulation

Consider a multi-agent system that consists of N agents well-connected by a graph $\mathcal{G} \triangleq \{\mathcal{V}, \mathcal{E}\}$, where each of them is indexed by $i \in [N]$. The agents are capable of performing local computations and exchanging binary decisions with other agents. Each agent has a label, which is private and marked by doctors. In this work, we consider the following collaborative filtering problem, i.e.,

$$\min_{\boldsymbol{\theta}_i, \forall i} \frac{1}{N} \sum_{i=1}^{N} f_i(\boldsymbol{\theta}_i), \quad \text{s.t. } \boldsymbol{\theta}_i = \boldsymbol{\theta}_j,\ j \in \mathcal{N}_i, \forall i \tag{1}$$

where $f_i(\boldsymbol{\theta}_i) = \mathbb{E}_{\xi_i \sim \mathcal{D}_i}[F(\boldsymbol{\theta}_i, \xi_i)]$ is smooth and possibly non-convex, $F(\boldsymbol{\theta}_i, \xi_i)$ denotes the loss function with respect to sample ξ_i, $\mathcal{N}_i$ represents the set of node i's neighbors, and $\mathcal{D}_i$ stands for the distribution of data at the ith node. Here, we consider the graph is well-connected in the sense that the following property is assumed.

Assumption 1. *Assume the weighting matrix* $\mathbf{W} \in \mathbb{R}^{n \times n}$ *is symmetric, satisfying*

$$|\lambda_{\max}(\mathbf{W})| < 1, \quad \mathbf{W}\mathbf{1} = \mathbf{1},$$

where $\lambda_{\max}(\mathbf{W})$ *denotes the second largest eigenvalue of* $\mathbf{W}$ *and* $\mathbf{1} \in \mathbb{R}^{n \times 1}$ *is an all one vector.* Problem (1) is the classic distributed optimization problem. Existing works [12], [14] have shown DSGD and DSGT are able to find an ϵ-approximate first-order stationary point in a sublinear convergence rate in the sense that the size of the gradient of the objective function and consensus violation of the iterates among all the nodes will be both small enough as the algorithm proceeds to a large number of iterations.

III. Fully Decentralized Non-convex Stochastic Algorithm for Federated Learning

The decentralized optimization algorithms have two key steps: 1) local update 2) communications among nodes. In the federated setting we perform local update multiple times instead of one.

A. Decentralized Stochastic Gradient Descent (DSGD)

First, let

$$\nabla_{\boldsymbol{\theta}_i} g_i(\boldsymbol{\theta}_i) = m^{-1} \sum_{l=1}^{m} \nabla_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i, \xi_l) \tag{2}$$

serve as an estimate of the true gradient at each node, where m denotes the size of mini-batch. The traditional DSGD basically performs the gradient update and communications at each step, i.e.,

$$\boldsymbol{\theta}_i^{r+1} = \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij} \boldsymbol{\theta}_j^r - \alpha^r \nabla_{\boldsymbol{\theta}_i} g_i(\boldsymbol{\theta}_i^r), \tag{3}$$

where r denotes the index of iterations. This rule is also known as combine-then-adaptive, meanning that each node combines its neighboring weights and then performs one step of simple SGD based on its own weight.

B. Decentralized Stochastic Gradient Tracking (DSGT)

In practice, the data is heterogeneously/non-identically distributed and the loss function is highly non-convex such as activation functions in training neural networks, the most efficient/advanced decentralized algorithm is DSGT. Instead of only performing local gradient update, the update of the iterates by DSGT can be written as the following

$$\boldsymbol{\theta}_i^{r+1} = \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij} \boldsymbol{\theta}_j^r - \alpha^r \boldsymbol{\vartheta}_i^r, \tag{4a}$$

$$\boldsymbol{\vartheta}^{r+1} = \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij} \boldsymbol{\vartheta}_j^r + \left(\nabla_{\boldsymbol{\theta}_i} g_i(\boldsymbol{\theta}_i^{r+1}) - \nabla_{\boldsymbol{\theta}_i} g_i(\boldsymbol{\theta}_i^r) \right). \tag{4b}$$

Compared with the DSGD method, the GT technique adds an auxiliary variable $\boldsymbol{\vartheta}_i^r$ for each node, which actually keeps tracking the full gradient of the objective function so that the error terms resulted by the difference of data distributions among nodes can be shrunk quickly [17], [18].

Next, we introduce the decentralized federated learning as follows.

C. Decentralized Federated Learning

Communication steps are costly when the data privacy is concerned, since encryption might be involved. Relatively, the local update is very efficient, which only needs to compute the estimated gradient in the following way in parallel, i.e.,

$$\boldsymbol{\theta}_i^{r+1} = \boldsymbol{\theta}_i^r - \alpha^r \nabla_{\boldsymbol{\theta}_i} g_i(\boldsymbol{\theta}_i^r). \tag{5}$$

Inspired by this fact, we insert multiple steps of the local update into the original DSGD and DSGT algorithms. The details of the algorithm are shown in Algorithm 1. It can be observed that we perform DSGD or DSGT for every Q times local updates.

Algorithm 1 Fully Decentralized Non-convex Stochastic Gradient Descent/Tracking for Federated Learning

Input: $\boldsymbol{\theta}^0$, α^0
for $r = 1, \ldots$ **do**
  Randomly collect m samples ξ_i^r locally
  Calculate the stochastic gradient $\nabla g_i(\mathbf{x}_i^r)$ by (2)
  Each node updates $\boldsymbol{\theta}_i^{r+1}$ individually by (5)
  if r is a multiple of Q, i.e., $\mod(r, Q) = 0$ **then**
    Update $\boldsymbol{\theta}_i^{r+1}$ by (4) or by (3)
  end if
end for

IV. Theoretically Convergence Results

A. Assumptions

Before showing the theoretical results, we first have the following assumptions on the problem setups.

Assumption 2. We assume that the objective function has Lipschitz gradient continuity with constant L, i.e.,

$$\|\nabla_{\mathbf{x}} f_i(\mathbf{x}) - \nabla_{\mathbf{y}} f_i(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|, \forall i,$$

and also assume the unbiased gradient estimation

$$\mathbb{E}_{\xi_i \sim \mathcal{D}_i} [\nabla_{\boldsymbol{\theta}_i} g_i(\boldsymbol{\theta}_i)] = \nabla_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i), \forall i,$$

and bounded estimation variance

$$\mathbb{E}_{\xi_i \sim \mathcal{D}_i} \|\nabla_{\boldsymbol{\theta}_i} g_i(\boldsymbol{\theta}_i) - \nabla_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i)\|^2 \leq \sigma^2, \forall i.$$

Towards this end, we also remark that relation $\mathbf{W}\mathbf{1} = \mathbf{1}$ implies $\|\mathbf{W} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\| < 1$, which gives the contraction of the iterates as the algorithm iterates so that the algorithm is able to achieve the consensus quickly.

B. Convergence Rate Guarantee

With the above assumptions and properties in mind, we can have the following theoretical result.

Theorem 1. *Suppose Assumptions 1 and 2 hold. If we choose* $\alpha^r \sim \mathcal{O}(\sqrt{N/r})$ *and* $Q = 1$ *in Algorithm 1 by adopting DSGT, then when* T *is large we have*

$$\frac{1}{T} \left(\sum_{r=1}^{T} \left\| \frac{1}{N} \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i^r) \right\|^2 + \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{\theta}_i^r - \bar{\boldsymbol{\theta}}^r\|^2 \right) \tag{6}$$

$$\leq \mathcal{O}\left(\frac{\sigma^2}{N\sqrt{T}} \right) \tag{7}$$

where $\bar{\boldsymbol{\theta}}^r = 1/N \sum_{i=1}^{N} \boldsymbol{\theta}_i^r$ *denotes the average of the iterates.*

It can be observed that the optimality gap decreases in a rate of $\mathcal{O}(\sigma^2/(N\sqrt{T}))$ with a linear speedup in terms of the number of the nodes, demonstrating the key superiority of performing distributed learning over centralized one [9], [12], [19]. Unfortunately, there is no theoretical guarantee for the case of a general $Q > 1$. To the best of our knowledge, there is no any theoretical results to show the convergence of any decentralized algorithm in this setting. From the numerical results, it can be seen in the next section that the decentralized

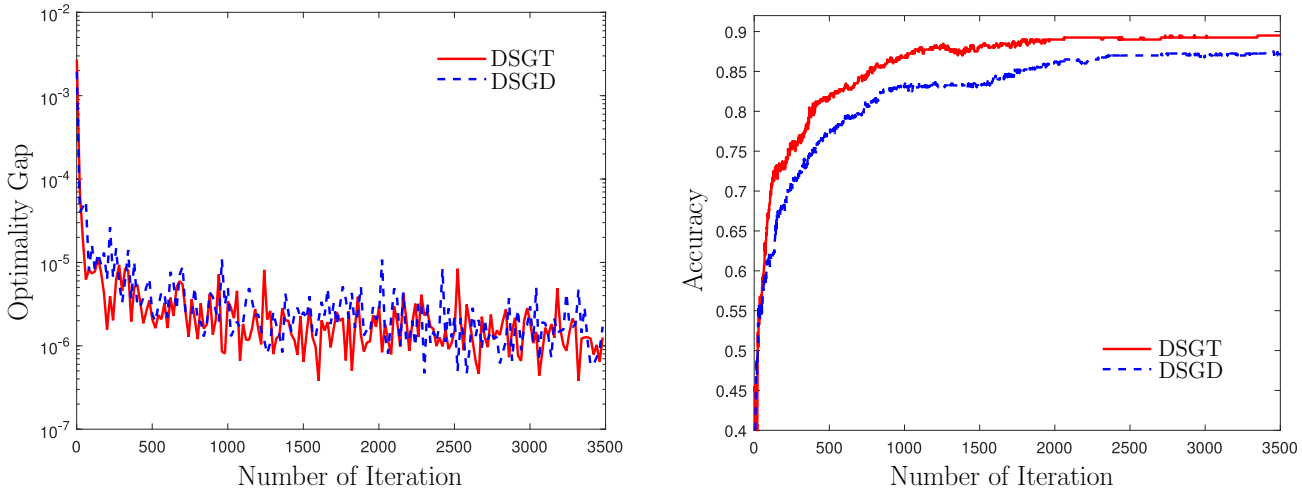

Fig. 2. Convergence comparison of DSGT and DSGD in terms of optimality gap v.s. number of iterations and training accuracy v.s. number of iterations.

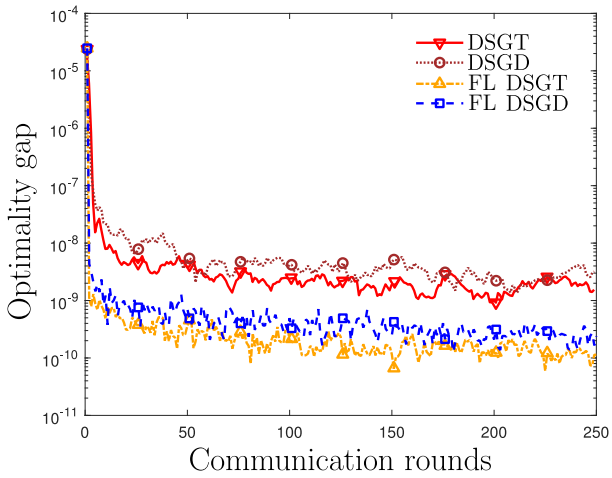

Fig. 3. Convergence behaviours of algorithms with respect to communication rounds

federated learning algorithm can also converge to the stationary points with much less communication rounds. We conjecture that the number of communications rounds needed in the decentralized setting is the same as the classic federated learning case where the topology of the network is star.

V. Numerical Experiments

In this section, we provide numerical results to showcase the decentralized federated learning for extracting latent features from the electronic health records. We compare DSGD, DSGT, FL DSGD, and FL DSGT, where $m = 20$, $Q = 100$, $\alpha^r = 0.02/\sqrt{r}$.

A. Synthetic Dataset

We train a shallow (2-layer) neural network at each node, where the problem dimension is 20, the numbers of neurons at the hidden and output layers are 50 and 1 respectively, the number of samples is 200, activation function is sigmoid, and the square loss in the objective function is employed. The topology of the well-connected graph is randomly generated. There are total 20 nodes. The data at each node is randomly generated, which follows *i.i.d.* Gaussian distribution with unit variance at node 1 to 5 and *i.i.d.* Gaussian distribution with variance 10 at the rest of nodes. In such a way, the distribution of the data is not homogeneous. The labels randomly generated are binary. From Fig. 2, it can be observed that DSGT converges slightly faster than DSGD and achieves a higher accuracy. The reason is that DSGT considers the difference of data distributions among the nodes by introducing tracking variable $\boldsymbol{\vartheta}_i$ and minimizes the full gradient of the objective function while DSGD ignores.

B. Real Dataset

For the real dataset, the problem dimension is 42. It can be observed from Fig. 3 that FD algorithms converge much faster than classic methods in terms of communication rounds. Compared with DSGD and DSGT, DSGT in general can achieve a smaller optimality gap due to the fact the GT is able to track the full gradient while DSGD only uses the local information to update the iterates. From a theory perspective, the difference between DSGD and DSGT will be diminishing asymptotically as their stepsizes decrease.

VI. Concluding Remarks and Future Work

In this work, we presented a new approach of leveraging decentralized non-convex optimization for federated learning to extract patients' features from real-world, de-identified hospital datasets. The advantages of performing decentralized federated learning are three-fold, 1) data privacy could be preserved better than the centralized case; 2) the computational burden is released by SGD and parallel computing compared with the centralized GD/SGD processing (a linear speedup); 3) the communication efficiency can be increased significantly by adopting FL. In future work, we will examine the theoretical guarantees of the algorithm for the case of $Q > 1$ and training/testing accuracy with a deeper neural net.

REFERENCES

[1] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *International Journal of Medical Informatics*, vol. 112, pp. 59–67, 2018.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, p. 12, 2019.

[4] A. Act, "Health insurance portability and accountability act of 1996," *Public Law*, vol. 104, p. 191, 1996.

[5] S. Toh, J. J. Gagne, J. A. Rassen, B. H. Fireman, M. Kulldorff, and J. S. Brown, "Confounding adjustment in comparative effectiveness research conducted within distributed research networks," *Medical Care*, vol. 51, pp. S4–S10, 2013.

[6] S. Toh, S. Shetterly, J. D. Powers, and D. Arterburn, "Privacy-preserving analytic methods for multisite comparative effectiveness and patient-centered outcomes research," *Medical Care*, vol. 52, no. 7, pp. 664–668, 2014.

[7] J. S. Brown, J. H. Holmes, K. Shah, K. Hall, R. Lazarus, and R. Platt, "Distributed health data networks: a practical and preferred approach to multi-institutional evaluations of comparative effectiveness, safety, and quality of care," *Medical Care*, pp. S45–S51, 2010.

[8] M. Li, D. G. Andersen, A. J. Smola, and K. Yu, "Communication efficient distributed machine learning with the parameter server," in *Proc. of Advances in Neural Information Processing Systems*, 2014, pp. 19–27.

[9] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization," *arXiv preprint arXiv:1905.03817*, 2019.

[10] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[11] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of optimization theory and applications*, vol. 147, no. 3, pp. 516–545, 2010.

[12] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *Proc. of Advances in Neural Information Processing Systems*, 2017, pp. 5330–5340.

[13] Z. Jiang, A. Balu, C. Hegde, and S. Sarkar, "Collaborative deep learning in fixed topology networks," in *Proc. of Advances in Neural Information Processing Systems*, 2017, pp. 5904–5914.

[14] S. Lu, X. Zhang, H. Sun, and M. Hong, "GNSD: a gradient-tracking based nonconvex stochastic algorithm for decentralized optimization," in *Proc. of IEEE Data Science Workshop (DSW)*, June 2019, pp. 315–321.

[15] T.-H. Change, M. Hong, H.-T. Wai, X. Zhang, and S. Lu, "Distributed learning in the non-convex world: from batch to streaming data, and beyond," *IEEE Signal Processing Magazine*, 2020.

[16] Q. Li, Z. Wen, and B. He, "Federated learning systems: Vision, hype and reality for data privacy and protection," *arXiv preprint arXiv:1907.09693*, 2019.

[17] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[18] Y. Sun, A. Daneshmand, and G. Scutari, "Convergence rate of distributed optimization algorithms based on gradient tracking," *arXiv preprint arXiv:1905.02637*, 2019.

[19] P. Jiang and G. Agrawal, "A linear speedup analysis of distributed deep learning with sparse and quantized communication," in *Proc. of Advances in Neural Information Processing Systems*, 2018, pp. 2525–2536.