Tian Li, Anit Kumar Sahu, Ameet Talwalkar,
and Virginia Smith

Federated Learning

Challenges, methods, and future directions

©ISTOCKPHOTO.COM/HAMSTER3D

Federated learning involves training statistical models over remote devices or siloed data centers, such as mobile phones or hospitals, while keeping data localized. Training in heterogeneous and potentially massive networks introduces novel challenges that require a fundamental departure from standard approaches for large-scale machine learning, distributed optimization, and privacy-preserving data analysis. In this article, we discuss the unique characteristics and challenges of federated learning, provide a broad overview of current approaches, and outline several directions of future work that are relevant to a wide range of research communities.

Introduction

Mobile phones, wearable devices, and autonomous vehicles are just a few of the modern distributed networks generating a wealth of data each day. Due to the growing computational power of these devices, coupled with concerns over transmitting private information, it is increasingly attractive to store data locally and push network computation to the edge.

The concept of edge computing is not a new one. Indeed, computing simple queries across distributed, low-powered devices is a decades-long area of research that has been explored under the purview of query processing in sensor networks, computing at the edge, and fog computing [6], [30]. Recent works have also considered training machine learning models centrally but serving and storing them locally; for example, this is a common approach in mobile user modeling and personalization [23].

However, as the storage and computational capabilities of the devices within distributed networks grow, it is possible to leverage enhanced local resources on each device. In addition, privacy concerns over transmitting raw data require user-generated data to remain on local devices. This has led to a growing interest in federated learning [31], which explores training statistical models directly on remote devices. The term *device* is used throughout the article to describe entities in the communication network, such as nodes, clients, sensors, or organizations.

Digital Object Identifier 10.1109/MSP.2020.2975749
Date of current version: 28 April 2020

As we discuss in this article, learning in such a setting differs significantly from traditional distributed environments, which require fundamental advances in areas such as privacy, large-scale machine learning, and distributed optimization and raise new questions at the intersection of diverse fields such as machine learning and systems. Federated learning methods have been deployed in practice by major companies [5], [41] and play a critical role in supporting privacy-sensitive applications where training data are distributed at the edge [8], [19]. In the next sections, we discuss several canonical applications of federated learning.

Federated learning has the potential to enable predictive features on smartphones without diminishing the user experience or leaking private information.

Smartphones

By jointly learning user behavior across a large pool of mobile phones, statistical models can power applications such as next-word prediction [17]. However, users may not be willing to share their data to protect their personal privacy or to save the limited bandwidth/battery power of their phone. Federated learning has the potential to enable predictive features on smartphones without diminishing the user experience or leaking private information. Figure 1 depicts one such application in which we aim to learn a next-word predictor in a large-scale mobile phone network based on users' historical text data [17].

Organizations

Organizations or institutions can also be viewed as "devices" in the context of federated learning. For example, hospitals are organizations that contain a multitude of patient data for predictive health care; however, hospitals operate under strict privacy practices and may face legal, administrative, or ethical constraints that require data to remain local. Federated learning is a promising solution for these applications [19], as it can reduce privacy leakage and naturally eliminate these constrains to enable private learning between various devices/organizations.

The Internet of Things

Modern Internet of Things networks, such as wearable devices, autonomous vehicles, or smart homes, may contain numerous sensors that allow them to collect, react, and adapt to incoming data in real time. For example, a fleet of autonomous vehicles may require an up-to-date model of traffic, construction, or pedestrian behavior to safely operate; however, building aggregate models in these scenarios may be difficult due to the private nature of the data and the limited connectivity of each device. Federated learning methods can help train models that efficiently adapt to changes in these systems, while maintaining user privacy.

Problem formulation

The standard federated learning problem involves learning a single global statistical model from data stored on tens to potentially millions of remote devices. We aim to learn this model under the constraint that device-generated data are stored and processed locally, with only intermediate updates being communicated periodically with a central server. The goal is typically to minimize the following objective function:

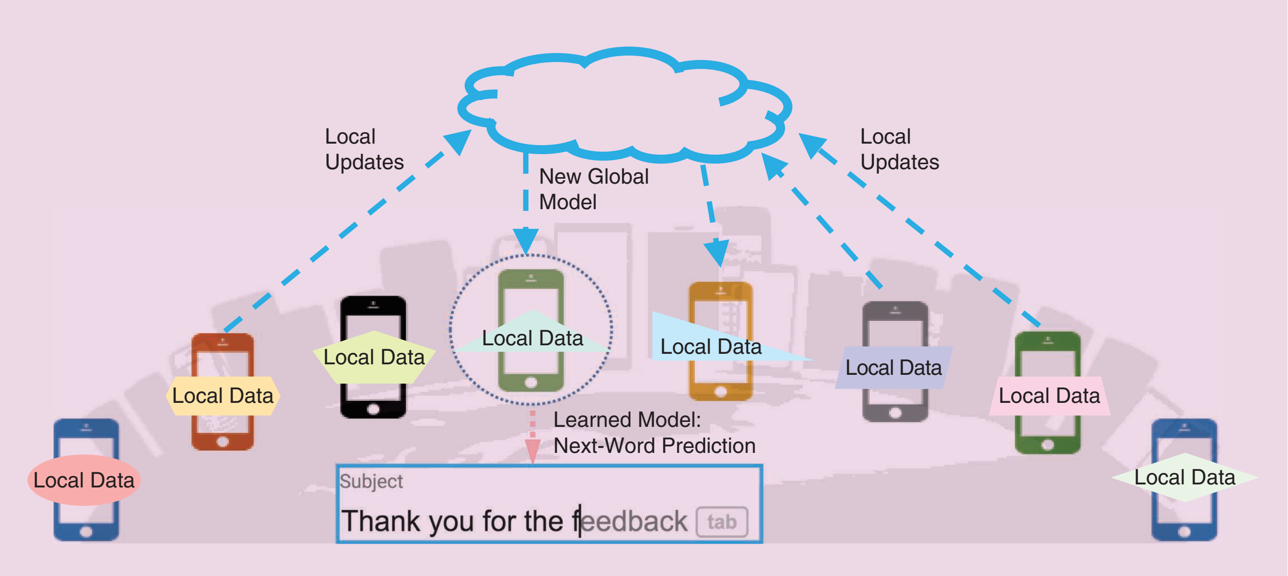

FIGURE 1. An example application of federated learning for the task of next-word prediction on mobile phones. To preserve the privacy of the text data and reduce strain on the network, we seek to train a predictor in a distributed fashion, rather than sending the raw data to a central server. In this setup, remote devices communicate with a central server periodically to learn a global model. At each communication round, a subset of selected phones performs local training on their nonidentically distributed user data, and sends these local updates to the server. After incorporating the updates, the server then sends back the new global model to another subset of devices. This iterative training process continues across the network until convergence is reached or some stopping criterion is met.

$$\min_w F(w), \text{ where } F(w) := \sum_{k=1}^{m} p_k F_k(w). \qquad (1)$$

Here, m is the total number of devices, $p_k \geq 0$ and $\Sigma_k p_k = 1$, and F_k is the local objective function for the kth device. The local objective function is often defined as the empirical risk over local data, i.e., $F_k(w) = 1/n_k \Sigma_{j_k=1}^{n_k} f_{j_k}(w; x_{j_k}, y_{j_k})$, where n_k is the number of samples available locally. The user-defined term p_k specifies the relative impact of each device, with two natural settings being $p_k = (1/n)$ or $p_k = (n_k/n)$, where $n = \Sigma_k n_k$ is the total number of samples. We will reference (1) throughout the article but, as discussed in the next section, we note that other objectives or modeling approaches may be appropriate depending on the application of interest.

Core challenges

We next describe four of the core challenges associated with solving the distributed optimization problem posed in (1). These challenges make the federated setting distinct from other classical problems, such as distributed learning in data center settings or traditional private data analyses.

Challenge 1: Expensive communication

Communication is a critical bottleneck in federated networks [5] which, when coupled with privacy concerns over sending raw data, necessitates that data generated on each device remain local. Indeed, federated networks potentially comprise a massive number of devices, e.g., millions of smartphones, and communication in the network can be slower than local computation by many orders of magnitude due to limited resources such as bandwidth, energy, and power [46]. To fit a model to data generated by the devices in the federated network, it is therefore important to develop communication-efficient methods that iteratively send small messages or model updates as part of the training process, as opposed to sending the entire data set over the network. To further reduce communication in such a setting, two key aspects to consider are 1) reducing the total number of communication rounds and 2) reducing the size of the transmitted messages at each round.

Challenge 2: Systems heterogeneity

The storage, computational, and communication capabilities of each device in federated networks may differ due to variability in hardware (CPU and memory), network connectivity (3G, 4G, 5G, and Wi-Fi), and power (battery level) [46]. Additionally, the network size and systems-related constraints on each device typically result in only a small fraction of the devices being active at once, e.g., hundreds of active devices in a network with millions of devices [5]. It is also not uncommon for an active device to drop out at a given iteration due to connectivity or energy constraints [5]. These system-level characteristics dramatically exacerbate challenges such as straggler mitigation and fault tolerance. Developed federated learning methods must therefore 1) anticipate a low amount of participation, 2) tolerate heterogeneous hardware, and 3) be robust enough to dropped devices in the communication network.

Challenge 3: Statistical heterogeneity

Devices frequently generate and collect data in a highly non-identically distributed manner across the network, e.g., mobile phone users have varied use of language in the context of a next-word prediction task. Moreover, the number of data points across devices may vary significantly, and there may be an underlying statistical structure present that captures the relationship among devices and their associated distributions [42]. This data-generation paradigm violates frequently used independent and identically distributed (i.i.d.) assumptions in distributed optimization and may add complexity in terms of problem modeling, theoretical analysis, and the empirical evaluation of solutions. Indeed, although the canonical federated learning problem of (1) aims to learn a single global model, there exist other alternatives such as simultaneously learning distinct local models via multitask learning frameworks (cf. [42]). In this regard, there is also a close connection between leading approaches for federated learning and metalearning [24]. Both the multitask and metalearning perspectives enable personalized or device-specific modeling, which is often a more natural approach to handle the statistical heterogeneity of the data for better personalization.

Devices frequently generate and collect data in a highly nonidentically distributed manner across the network.

Challenge 4: Privacy concerns

Finally, privacy is often a major concern in federated learning applications. Federated learning makes a step toward protecting data generated on each device by sharing model updates, e.g., gradient information, instead of the raw data. However, communicating model updates throughout the training process can nonetheless reveal sensitive information, either to a third-party or the central server [32]. Although recent methods aim to enhance the privacy of federated learning using tools such as secure multiparty computation (SMC) or differential privacy, these approaches often provide privacy at the cost of reduced model performance or system efficiency [4], [32]. Understanding and balancing these tradeoffs, both theoretically and empirically, is a considerable challenge in realizing private federated learning systems.

Survey of related and current work

At first glance, the challenges in federated learning resemble classical problems in areas such as privacy, large-scale machine learning, and distributed optimization. For instance, numerous methods have been proposed to tackle expensive communication in the optimization and signal processing communities [28], [40], [43]. However, these methods are typically unable to fully handle the scale of federated networks, much less the challenges of systems and statistical heterogeneity (see the discussions throughout this section). Similarly, even though privacy is an important aspect for many applications, privacy-preserving methods for federated learning can be challenging to rigorously assert as a result of statistical variations in data and may be even more difficult to implement due to systems constraints on each device and across the massive network. In the following section, we explore in greater detail the challenges presented in the "Introduction" section, including a discussion of classical results as well as more recent work focused specifically on federated learning.

The most commonly used method for federated learning is federated averaging, a method based on averaging local stochastic gradient descent updates for the primal problem.

Communication efficiency

Communication is a key bottleneck to consider when developing methods for federated networks. Although it is beyond the scope of this article to provide a self-contained review of communication-efficient learning methods, we point out several general directions, which we group into 1) local updating methods, 2) compression schemes, and 3) decentralized training.

Local updating

Minibatch optimization methods, which involve extending classical stochastic methods to process multiple data points at a time, have emerged as a popular paradigm for distributed machine learning in data center environments. In practice, however, they have shown limited flexibility to adapt to communication-computation tradeoffs [53], which would maximally leverage distributed data processing. In response, several recent methods have been proposed to improve communication efficiency in distributed settings by allowing for a variable to be applied on each machine in parallel at each communication round (rather than just computing them locally and then applying them centrally) [44]. This makes the amount of computation versus communication substantially more flexible.

For convex objectives, distributed local updating primal-dual methods have emerged as a popular way to tackle such a problem [43]. These approaches leverage duality structures to effectively decompose the global objective into subproblems that can be solved in parallel at each communication round. Several distributed local updating primal methods have also been proposed, which have the added benefit of being applicable to nonconvex objectives [53]. These methods drastically improve performance in practice, and have been shown to achieve orders-of-magnitude speedups over traditional minibatch methods or distributed approaches like the alternating direction method of multipliers in real-world data center environments. We provide an intuitive illustration of local updating methods in Figure 2.

In federated settings, optimization methods that allow for flexible local updating and low client participation have become the de facto solvers [31]. The most commonly used method for federated learning is federated averaging (FedAvg) [31], a method based on averaging local stochastic gradient descent (SGD) updates for the primal problem. FedAvg has been shown to work well empirically, particularly for nonconvex problems, but comes without convergence guarantees and can diverge in practical settings when data are heterogeneous [25]. We discuss methods to handle such statistical heterogeneity in more detail in the "Convergence Guarantees for Non-i.i.d. Data" section.

Compression schemes

Although local updating methods can reduce the total number of communication rounds, model-compression schemes such as sparsification and quantization can significantly reduce the size of messages communicated at each round. These methods have been extensively studied, both empirically and theoretically, in previous literature for distributed training in data center environments. (We refer readers to [47] for a more complete review.) In federated environments, the low participation of devices, nonidentically distributed local data, and local updating schemes pose novel challenges to these model-compression approaches. Several works have provided practical strategies in federated settings, such as forcing the updating models to be sparse and low rank [22], performing quantization with

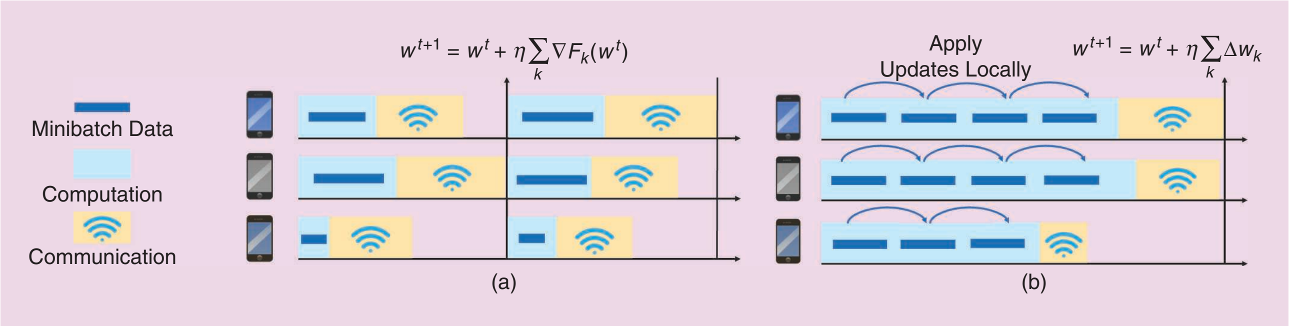
Minibatch Data
Computation
Communication
$w^{t+1} = w^t + \eta \sum_k \nabla F_k(w^t)$
(a)
Apply Updates Locally
$w^{t+1} = w^t + \eta \sum_k \Delta w_k$
(b)

FIGURE 2. (a) The distributed (minibatch) SGD. Each device, k, locally computes gradients from a minibatch of data points to approximate $\nabla F_k(w)$, and the aggregated minibatch updates are applied on the server. (b) The local updating schemes. Each device immediately applies local updates, e.g., gradients, after they are computed, and a server performs a global aggregation after a (potentially) variable number of local updates. Local updating schemes can reduce communication by performing additional work locally.

structured random rotations [22], and using lossy compression and dropout to reduce server-to-device communication [9]. From a theoretical perspective, although prior work has explored convergence guarantees with low-precision training in the presence of nonidentically distributed data (e.g., [45]), the assumptions made do not take into consideration common characteristics of the federated setting, such as low device participation or locally updating optimization methods.

We point out several general directions, which we group into 1) local updating methods, 2) compression schemes, and 3) decentralized training.

Decentralized training

In federated learning, a star network [where a central server is connected to a network of devices, e.g., in Figure 3(a)] is the predominant communication topology; we therefore focus on the star network setting in this article. We briefly discuss decentralized topologies [where devices only communicate with their neighbors, e.g., in Figure 3(b)] as a potential alternative. In data center environments, decentralized training has been demonstrated to be faster than centralized training when operating on networks with low bandwidth or high latency. Some works propose deadline-based approaches where all workers compute the local gradients using a variable number of samples within a fixed global cycle, which helps mitigate the impact of stragglers [16], [39]. (We refer readers to [18] for a more comprehensive review.) Similarly, in federated learning, decentralized algorithms can, in theory, reduce the high communication cost on the central server. Some recent works have investigated decentralized training over heterogeneous data with local updating schemes [18]. However, they are either restricted to linear models [18] or assume full device participation.

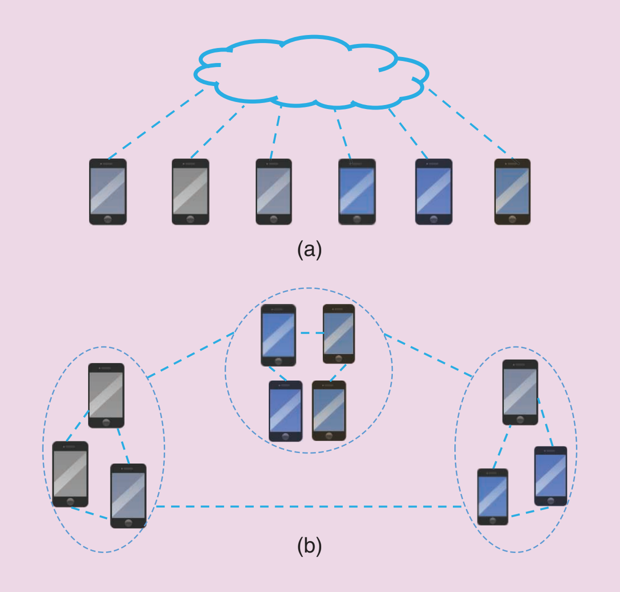

FIGURE 3. Centralized versus decentralized topologies. In the typical federated learning setting and as a focus of this article, we assume (a) a star network where a server connects with all the remote devices. (b) Decentralized topologies are a potential alternative when communication to the server becomes a bottleneck.

Systems heterogeneity

In federated settings, there is significant variability in the systems characteristics across the network, as devices may differ in terms of hardware, network connectivity, and battery power. As depicted in Figure 4, these systems characteristics make issues such as stragglers significantly more prevalent than in typical data center environments. We roughly group several key directions used to handle systems heterogeneity into 1) asynchronous communication, 2) active device sampling, and 3) fault tolerance. As mentioned in the "Decentralized Training" section, we assume a star topology for the discussions presented in the following section.

Asynchronous communication

In traditional data center settings, synchronous (i.e., workers waiting for each other for synchronization) and asynchronous (i.e., workers running independently without synchronization) schemes are both commonly used to parallelize iterative optimization algorithms, with each approach having advantages and disadvantages [37], [53]. Synchronous schemes are

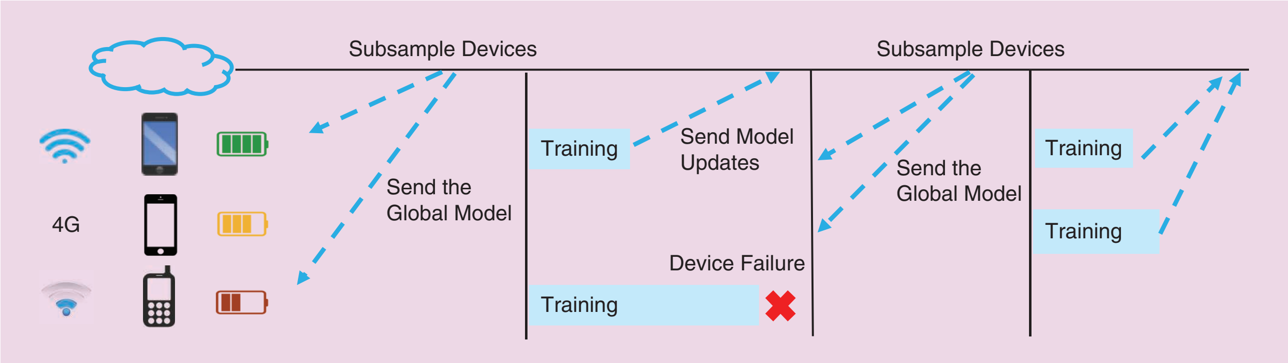

FIGURE 4. Systems heterogeneity in federated learning. Devices may vary in terms of network connection, power, and hardware. Moreover, some of the devices may drop at any time during training. Therefore, federated training methods must tolerate heterogeneous systems environments and low participation of devices, i.e., they must allow for only a small subset of devices to be active at each round.

simple and guarantee a serial-equivalent computational model but, they are also more susceptible to stragglers in the face of device variability [53]. Asynchronous schemes are an attractive approach used to mitigate stragglers in heterogeneous environments, particularly in shared-memory systems [37]. However, they typically rely on bounded-delay assumptions to control the degree of staleness [37]. For device *k*, the staleness depends on the number of other devices that have updated since device *k* pulled from the central server. Even though asynchronous parameter servers have been successful in distributed data centers [37], classical bounded-delay assumptions can be unrealistic in federated settings, where the delay may be on the order of hours to days or completely unbounded.

Active sampling

In federated networks, typically only a small subset of devices participate at each round of training; however, the vast majority of federated methods, e.g., those described in [5], [25], and [31], are passive in the sense that they do not aim to influence which devices participate. An alternative approach involves actively selecting participating devices at each round. For example, Nishio and Yonetani [36] explore novel device-sampling policies based on systems resources, with the aim being for the server to aggregate as many device updates as possible within a predefined time window. However, these methods assume a static model of the systems characteristics of the network; it remains open how best to extend these approaches to handle real-time, device-specific fluctuations in computation and communication delays. Moreover, although these methods primarily focus on systems variability to perform active sampling, we note that it is also worth considering actively sampling a set of small but sufficiently representative devices based on the underlying statistical structure.

Asynchronous schemes are an attractive approach used to mitigate stragglers in heterogeneous environments, particularly in shared-memory systems.

Fault tolerance

Fault tolerance has been extensively studied in the systems community and is a fundamental consideration of classical distributed systems, including formalisms such as Byzantine failures [50]. Recent works have also investigated fault tolerance specifically for machine learning workloads in data center environments. When learning over remote devices, however, fault tolerance becomes more critical, as it is common for some participating devices to drop out at some point before the completion of the given training iteration [5]. One practical strategy is to simply ignore such device failure, as in FedAvg [5], which may introduce bias into the device-sampling scheme if the failed devices have specific data characteristics. For instance, devices from remote areas may be more likely to drop due to poor network connections and thus, the trained federated model will be biased toward devices with favorable network conditions.

Theoretically, although several recent works have investigated convergence guarantees of variants of federated learning methods [52], few analyses allow for low participation [25], [42] or study directly the effect of dropped devices [50]. FedProx tackles systems heterogeneity by allowing for each selected device to perform partial work that conforms to the underlying systems constraints and safely incorporate those partial updates via a proximal term (see the "Convergence Guarantees for Non-i.i.d. Data" section for a more detailed discussion).

Coded computation is another option used to tolerate device failures by introducing algorithmic redundancy. Recent works have explored using codes to speed up distributed machine learning training [11]. For instance, in the presence of stragglers, gradient coding and its variants [11] carefully replicate data blocks (as well as the gradient computation on those data blocks) across computing nodes to obtain either an exact or inexact recovery of the true gradients. Although this is seemingly a promising approach for the federated setting, these methods face fundamental challenges in federated networks, as sharing data/replication across devices is often infeasible due to privacy constraints and the scale of the network.

Statistical heterogeneity

Challenges arise when training federated models from data that are highly nonidentically distributed across devices, both in terms of modeling heterogeneous data and in terms of analyzing the convergence behavior of associated training procedures. We discuss related work in these next sections.

Modeling heterogeneous data

There exists a large body of literature in machine learning that models statistical heterogeneity via methods such as metalearning and multitask learning; these ideas have been recently extended to the federated setting [12], [14], [21]. For instance, MOCHA [42], an optimization framework designed for the federated setting, can allow for personalization by learning separate but related models for each device, while leveraging a shared representation via multitask learning. This method has provable theoretical convergence guarantees for the considered objectives but is limited in its ability to scale to massive networks and is restricted to convex objectives. Another approach [12] models the star topology as a Bayesian network and performs variational inference during learning. Although this method can handle nonconvex functions, it is expensive to generalize to large federated networks. Khodak et al. [21] provably metalearn a within-task learning rate using multitask information (where each task corresponds to a device) and demonstrate improved empirical performance over vanilla FedAvg. Eichner et al. [14] investigate a pluralistic solution (adaptively choosing between a global model and device-specific models) to address the cyclic patterns in data samples during federated training. Despite these recent advances, key challenges still remain in developing methods for heterogeneous modeling that are robust, scalable, and automated in federated settings.

When modeling federated data, it may also be important to consider issues beyond accuracy, such as fairness. In particular, naively solving an aggregate loss function, such as in (1), may implicitly advantage or disadvantage some of the devices, as the learned model may become biased toward devices with larger amounts of data, or (if weighting devices equally), to commonly occurring groups of devices. Recent works have proposed modified modeling approaches that aim to reduce the variance of the model performance across devices [19], [26], [33]. Some heuristics simply perform a varied number of local updates based on local loss of the device [19]. Other more principled approaches include agnostic federated learning [33], which optimizes the centralized model for any target distribution formed by a mixture of the client distributions via a minimax optimization scheme. Another more general approach is taken by Li et al. [26], which proposes an objective called q-FFL, in which devices with higher loss are given higher relative weight to encourage less variance in the final accuracy distribution. Beyond issues of fairness, we note that aspects such as accountability and interpretability in federated learning are, additionally, worth exploring, but may be challenging due to the scale and heterogeneity of the network.

Convergence guarantees for non-i.i.d. data

Statistical heterogeneity also presents novel challenges in terms of analyzing the convergence behavior in federated settings—even when learning a single global model. Indeed, when data are not identically distributed across devices in the network, methods such as FedAvg can diverge in practice when the selected devices perform too many local updates [25], [31]. Parallel SGD and related variants, which make local updates similar to FedAvg, have been analyzed in the i.i.d. setting [38], [48], [53]. However, the results rely on the premise that each local solver is a copy of the same stochastic process (due to the i.i.d. assumption), which is not the case in typical federated settings.

To understand the performance of FedAvg in heterogeneous settings, FedProx [25] was recently proposed. The key idea of FedProx is that there is an interplay between systems heterogeneity and statistical heterogeneity. As mentioned previously, simply dropping the stragglers in the network due to systems constraints can implicitly increase statistical heterogeneity. FedProx makes a small modification to the FedAvg method by allowing for partial work to be performed across devices based on the underlying systems constraints and leveraging the proximal term to safely incorporate the partial work. It can be viewed as a reparameterization of FedAvg because tuning the proximal term of FedProx is effectively equivalent to tuning the number of local epochs E in FedAvg. However, it is unrealistic to set E for the devices constrained by systems conditions. The proximal term therefore has two benefits: 1) it encourages more well-behaved local updates by restricting the local updates to be closer to the initial (global) model and 2) it safely incorporates the partial updates from selected devices.

Theoretically, FedProx uses a dissimilarity metric to capture the statistical heterogeneity in the network and provides convergence guarantees for both convex and nonconvex functions under the bounded device dissimilarity assumption. The convergence analysis also covers the setting in which each device performs a variable amount of work locally. Several other works [27], [52] have also derived convergence guarantees in the presence of heterogeneous data with different assumptions, e.g., convexity [27] or uniformly bounded gradients [52]. There are also heuristic approaches that aim to tackle statistical heterogeneity, either by sharing local device data or some server-side proxy data [19], [20]. However, these methods may be unrealistic: in addition to imposing burdens on network bandwidth, sending local data to the server violates the key privacy assumption of federated learning, and sending globally shared proxy data to all devices requires effort to carefully generate or collect such auxiliary data.

The key idea of FedProx is that there is an interplay between systems heterogeneity and statistical heterogeneity.

Privacy

Privacy concerns often motivate the need to keep raw data on each device local in federated settings; however, sharing other information, such as model updates as part of the training process, can also leak sensitive user information. For instance, Carlini et al. [10] demonstrate that one can extract sensitive text patterns, e.g., a specific credit card number, from a recurrent neural network trained on users' language data. Given increasing interest in privacy-preserving learning approaches, in the "Privacy in Machine Learning" section, we first briefly revisit prior work on enhancing privacy in the general (distributed) machine learning setting. We then review recent privacy-preserving methods specifically designed for federated settings in the "Privacy in Federated Learning" section.

Privacy in machine learning

The three main strategies in privacy-preserving machine learning, each of which are briefly reviewed, include differential privacy to communicate noisy data sketches, homomorphic encryption to operate on encrypted data, and secure function evaluation (SFE) or multiparty computation.

Among these various privacy approaches, differential privacy [13] is most widely used due to its strong information theoretic guarantees, algorithmic simplicity, and relatively small systems overhead. Simply put, a randomized mechanism is differentially private if the change of one input element will not result in too much difference in the output distribution; this means that one cannot draw any conclusions about whether or not a specific sample is used in the learning process. Such sample-level privacy can be achieved in many learning tasks. For gradient-based learning methods, a popular approach is to apply differential privacy by randomly perturbing the intermediate output at each iteration. Before applying the perturbation, e.g., via binomial noise [1], it is common to clip the gradients to bound the influence of each example on the overall update. There exists an inherent tradeoff between differential privacy and model accuracy, as adding more noise results in greater privacy but may compromise accuracy significantly. Despite the fact that differential privacy is the de facto metric for privacy in machine learning, there are many other privacy definitions, such as *k-anonymity* [15] and δ-*presence* [34], which may be applicable to different learning problems.

Current works that aim to improve the privacy of federated learning typically build upon previous classical cryptographic protocols such as SMC and differential privacy.

Beyond differential privacy, homomorphic encryption can be used to secure the learning process by computing on encrypted data, although it has currently been applied in limited settings, e.g., training linear models [35]. When the user-generated data are distributed across different data owners, another natural option is to perform privacy-preserving learning via SFE or SMC. The resulting protocols can enable multiple parties to collaboratively compute an agreed-upon function without leaking input information from any party except for what can be inferred from the output. Thus, although SMC cannot guarantee protection from information leakage, it can be combined with differential privacy to achieve stronger privacy guarantees. However, approaches along these lines may not be applicable to large-scale machine learning scenarios, as they incur substantial additional communication and computation costs. We refer interested readers to [7] for a more comprehensive review of the approaches based on homomorphic encryption and SMC.

Privacy in federated learning

The federated setting poses novel challenges to existing privacy-preserving algorithms. Beyond providing rigorous privacy guarantees, it is necessary to develop methods that are computationally cheap, communication efficient, and tolerant to dropped devices—all without overly compromising accuracy. Although there are a variety of privacy definitions in federated learning, typically, they can be classified into two categories: global privacy and local privacy. As presented in Figure 5, global privacy requires that the model updates generated at each round are private to all untrusted third parties other than the central server, while local privacy further requires that the updates are also private to the server.

Current works that aim to improve the privacy of federated learning typically build upon previous classical cryptographic protocols such as SMC [4] and differential privacy [2], [32]. Bonawitz et al. [4] introduce a secure aggregation

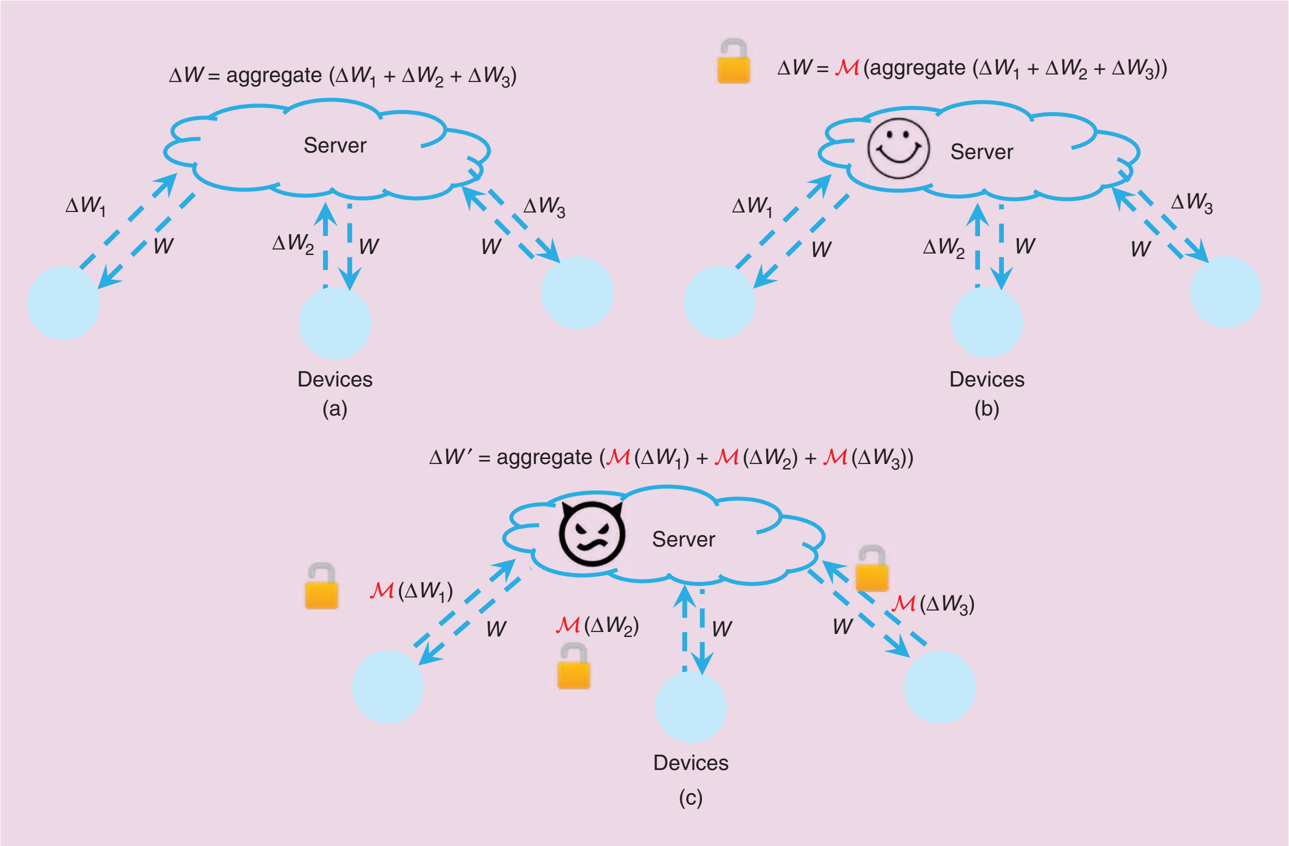
ΔW = aggregate ($\Delta W_1 + \Delta W_2 + \Delta W_3$)
Server
ΔW_1
W
ΔW_2
W
W
ΔW_3
Devices
(a)
$\Delta W = \mathcal{M}$(aggregate ($\Delta W_1 + \Delta W_2 + \Delta W_3$))
Server
ΔW_1
W
ΔW_2
W
W
ΔW_3
Devices
(b)
$\Delta W'$ = aggregate ($\mathcal{M}(\Delta W_1) + \mathcal{M}(\Delta W_2) + \mathcal{M}(\Delta W_3)$)
Server
$\mathcal{M}(\Delta W_1)$
W
$\mathcal{M}(\Delta W_2)$
W
W
$\mathcal{M}(\Delta W_3)$
Devices
(c)

FIGURE 5. An illustration of different privacy-enhancing mechanisms in one round of federated learning. $\mathcal{M}$ denotes a randomized mechanism used to privatize the data. (a) Federated learning without additional privacy protection mechanisms, (b) global privacy, where a trusted server is assumed, and (c) local privacy, where the central server may be malicious.

protocol to protect individual model updates. The central server is not able to see any local updates but can still observe the exact aggregated results at each round. Secure aggregation is a lossless method and can retain the original accuracy with a very high privacy guarantee; however, the resulting method incurs significant extra communication cost. Other works apply differential privacy to federated learning and offer global differential privacy (e.g., [32]); these approaches have a number of hyperparameters that affect communication and accuracy and must be carefully chosen. In the case where stronger privacy guarantees are required, Bhowmick et al. [2] introduce a relaxed version of local privacy by limiting the power of potential adversaries. It affords stronger privacy guarantees than global privacy and has better model performance than strict local privacy. Li et al. [24] propose locally differentially private algorithms in the context of metalearning, which can be applied to federated learning with personalization, while also providing provable learning guarantees in convex settings. In addition, differential privacy can be combined with model-compression techniques to reduce communication and obtain privacy benefits simultaneously [1].

The two most commonly studied communication schemes in distributed optimization are bulk synchronous approaches and asynchronous approaches.

Future directions

Federated learning is an active and ongoing area of research. Although recent work has begun to address the challenges discussed in the "Survey of Related and Current Work" section, there are a number of critical open directions yet to be explored. In this section, we briefly outline a few promising research directions surrounding the previously discussed challenges (expensive communication, systems heterogeneity, statistical heterogeneity, and privacy concerns) and introduce additional challenges regarding issues such as productionizing and benchmarking in federated settings.

Extreme communication schemes

It remains to be seen how much communication is necessary in federated learning. Indeed, it is well known that optimization methods used for machine learning can tolerate a lack of precision; this error can, in fact, help with generalization [49]. Although one-shot or divide-and-conquer communication schemes have been explored in traditional data center settings [29], the behavior of these methods is not well understood in massive and statistical heterogeneous networks.

Communication reduction and the Pareto frontier

We have discussed several ways to reduce communication in federated training, such as local updating and model compression. It is important to understand how these techniques compose with one another and to systematically analyze the tradeoff between accuracy and communication for each approach. In particular, the most useful techniques will demonstrate improvements at the Pareto frontier, i.e., achieving an accuracy greater than any other approach under the same communication budget and, ideally, across a wide range of communication/accuracy profiles. Similar comprehensive analyses have been performed for efficient neural network inference [3] and are necessary to compare communication-reduction techniques for federated learning in a meaningful way.

Novel models of asynchrony

As discussed in the "Asynchronous Communication" section, the two most commonly studied communication schemes in distributed optimization are bulk synchronous approaches and asynchronous approaches (where it is assumed that the delay is bounded). These schemes are more realistic in data center settings, where worker nodes are typically dedicated to the workload, i.e., they are ready to "pull" their next job from the central node immediately after they "push" the results of their previous job. In contrast, in federated networks, each device is often undedicated to the task at hand and most devices are not active on any given iteration. Therefore, it is worth studying the effects of this more realistic device-centric communication scheme, in which each device can decide when to "wake up," at which point it pulls a new task from the central node and performs some local computation.

Heterogeneity diagnostics

Recent works have aimed to quantify statistical heterogeneity through metrics such as local dissimilarity (as defined in the context of federated learning in [25] and used for other purposes in works such as [51]). However, these metrics cannot be easily calculated over the federated network before training occurs. The importance of these metrics motivates the following open questions:

1) Do simple diagnostics exist to quickly determine the level of heterogeneity in federated networks a priori?
2) Can analogous diagnostics be developed to quantify the amount of systems-related heterogeneity?
3) Can current or new definitions of heterogeneity be exploited to design new federated optimization methods with improved convergence, both empirically and theoretically?

Granular privacy constraints

The definitions of privacy outlined in the "Privacy in Federated Learning" section cover privacy at a local or global level with respect to all devices in the network. However, in practice, it may be necessary to define privacy on a more granular level, as privacy constraints may differ across devices or even across data points on a single device. For instance, Li et al. [24] recently proposed sample-specific (as opposed to user-specific) privacy guarantees, thus providing a weaker form of privacy in exchange for more accurate models. Developing methods to handle mixed (device-specific or sample-specific) privacy restrictions is an interesting and ongoing direction of future work.

Beyond supervised learning

It is important to note that the methods discussed thus far have been developed with the task of supervised learning in mind, i.e., they assume that labels exist for all of the data in the federated network. In practice, much of the data generated in realistic federated networks may be unlabeled or weakly labeled. Furthermore, the problem at hand may not be to fit a model to data, as presented in (1), but instead to perform some exploratory data analysis, determine aggregate statistics, or run a more complex task, such as reinforcement learning. Tackling problems beyond supervised learning in federated networks will likely require addressing similar challenges of scalability, heterogeneity, and privacy.

Productionizing federated learning

Beyond the major challenges discussed in this article, there are a number of practical concerns that arise when running federated learning in production. In particular, issues such as concept drift (when the underlying data-generation model changes over time), diurnal variations (when the devices exhibit different behavior at different times of the day or week) [14], and cold-start problems (when new devices enter the network) must be handled with care. We refer readers to [5], which discusses some of the practical systems-related issues that exist in production federated learning systems.

Benchmarks

Finally, as federated learning is a nascent field, we are at a pivotal time to shape the developments made in this area and must ensure that they are grounded in real-world settings, assumptions, and data sets. It is critical for the broader research communities to further build upon existing benchmarking tools and implementations, such as LEAF [54] and TensorFlow Federated [55], to facilitate both the reproducibility of empirical results and the dissemination of new solutions for federated learning.

Conclusions

In this article, we provided an overview of federated learning, a learning paradigm where statistical models are trained at the edge in distributed networks. We discussed the unique properties and associated challenges of federated learning compared with traditional distributed data center computing and classical privacy-preserving learning. We provided a broad survey on classical results as well as more recent work specifically focused on federated settings. Finally, we outlined a handful of open problems worth future research effort. Providing solutions to these problems will require interdisciplinary effort from a broad set of research communities.

Authors

Tian Li (tianli@cmu.edu) received her bachelor's degree in computer science at Peking University, Beijing, China. Currently, she is a Ph.D. candidate in the Computer Science Department at Carnegie Mellon University, Pittsburgh, Pennsylvania, advised by Dr. Virginia Smith. Her research interests include large-scale machine learning, distributed optimization, and data-intensive systems.

Anit Kumar Sahu (anit.sahu@gmail.com) received his B.Tech. degree in electronics and electrical communication engineering and his M.Tech. degree in telecommunication systems engineering, both from the Indian Institute of Technology, Kharagpur, India, in May 2013 and his Ph.D. degree in electrical and computer engineering from Carnegie Mellon University (CMU), Pittsburgh, Pennsylvania, in 2018. He is currently a machine learning research scientist at the Bosch Center for Artificial Intelligence in Pittsburgh. He is a recipient of the 2019 A.G. Jordan Award from the Department of Electrical and Computer Engineering at CMU. His research interests include distributed optimization and robust deep learning. He is a Member of the IEEE.

Ameet Talwalkar (talwalkar@cmu.edu) is an assistant professor in the Machine Learning Department at Carnegie Mellon University, Pittsburgh, Pennsylvania, and is also the cofounder and chief scientist at Determined AI. His current work is motivated by the goal of democratizing machine learning, with a focus on topics related to scalability, automation, fairness, and interpretability. He led the initial development of the MLlib project in Apache Spark, is a coauthor of the textbook *Foundations of Machine Learning* (MIT Press), and created an award-winning edX massive open online course on distributed machine learning. He also helped create the Conference on Machine Learning and Systems and is currently the president of the MLSys board.

Virginia Smith (smithv@cmu.edu) received undergraduate degrees in mathematics and computer science from the University of Virginia, Charlottesville, and her Ph.D. degree in computer science from the University of California, Berkeley. Previously, she was a postdoctoral professor at Stanford University, California. Currently, she is an assistant professor in the Machine Learning Department and a courtesy faculty member in the Electrical and Computer Engineering Department at Carnegie Mellon University, Pittsburgh, Pennsylvania. Her research interests include machine learning, optimization, and distributed systems.

References

[1] N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan, "cpSGD: Communication-efficient and differentially-private distributed SGD," in *Proc. Advances in Neural Information Processing Systems*, 2018, pp. 7564–7575.

[2] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, Protection against reconstruction and its applications in private federated learning. 2018. [Online]. Available: arXiv:1812.00984

[3] T. Bolukbasi, J. Wang, O. Dekel, and V. Saligrama, "Adaptive neural networks for efficient inference," in *Proc. Int. Conf. Machine Learning*, Aug. 2017, vol. 70, pp. 527–536.

[4] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal et al., "Practical secure aggregation for privacy-preserving

machine learning," in *Proc. Conf. Computer and Communications Security*, 2017, 1175–1191. doi: 10.1145/3133956.3133982.

[5] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny et al., "Towards federated learning at scale: System design," in *Proc. Conf. Machine Learning and Systems*, 2019.

[6] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. SIGCOMM Workshop on Mobile Cloud Computing*, 2012. doi: 10.1145/2342509.2342513.

[7] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *Proc. Network and Distributed System Security Symp.*, 2015. doi: 10.14722/ndss.2015.23241.

[8] 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," *Int. J. Medical Informatics*, vol. 112, Apr. 2018, pp. 59–67. doi: 10.1016/j.ijmedinf.2018.01.007.

[9] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, Expanding the reach of federated learning by reducing client resource requirements. 2018. [Online]. Available: arXiv:1812.07210

[10] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *Proc. USENIX Security Symp.*, 2019, pp. 267–284.

[11] Z. Charles and D. Papailiopoulos, "Gradient coding using the stochastic block model," in *Proc. Int. Symp. Information Theory*, 2018, pp. 1998–2002. doi: 10.1109/ISIT.2018.8437887.

[12] L. Corinzia and J. M. Buhmann, Variational federated multi-task learning. 2019. [Online]. Available: arXiv:1906.06268

[13] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory of Cryptography Conf.*, 2006, pp. 265–284. doi: 10.1007/11681878_14.

[14] H. Eichner, T. Koren, H. B. McMahan, N. Srebro, and K. Talwar, "Semi-cyclic stochastic gradient descent," in *Proc. Int. Conf. Machine Learning*, 2019, pp. 1764–1773.

[15] K. E. Emam and F. K. Dankar, "Protecting privacy using k-anonymity," *J. Amer. Med. Inform. Assoc.*, vol. 15, no. 5, pp. 627–637, 2008. doi: 10.1197/jamia.M2716.

[16] N. Ferdinand, H. Al-Lawati, S. Draper, and M. Nokleby, "Anytime Minibatch: Exploiting stragglers in online distributed optimization," in *Proc. Int. Conf. Learning Representations*, 2019.

[17] A. Hard, K. Rao, R. Mathews, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, Federated learning for mobile keyboard prediction. 2018. [Online]. Available: arXiv:1811.03604

[18] L. He, A. Bian, and M. Jaggi, "Cola: Decentralized linear learning," in *Proc. Advances in Neural Information Processing Systems*, 2018, pp. 4541–4551.

[19] L. Huang, Y. Yin, Z. Fu, S. Zhang, H. Deng, and D. Liu, LoAdaBoost: Loss-based adaboost federated machine learning on medical data. 2018. [Online]. Available: arXiv:1811.12629

[20] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, Communication-efficient on-device machine learning: Federated distillation and augmentation under non-IID private data. 2018. [Online]. Available: arXiv:1811.11479

[21] M. Khodak, M.-F. Balcan, and A. Talwalkar, "Adaptive gradient-based meta-learning methods," in *Proc. Advances in Neural Information Processing Systems*, 2019, pp. 5917–5928.

[22] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, Federated learning: Strategies for improving communication efficiency. 2016. [Online]. Available: arXiv:1610.05492

[23] T. Kuflik, J. Kay, and B. Kummerfeld, "Challenges and solutions of ubiquitous user modeling," in *Ubiquitous Display Environments*, A. Krüger and T. Kuflik, Eds. Berlin: Springer-Verlag, 2012, pp. 7–30.

[24] J. Li, M. Khodak, S. Caldas, and A. Talwalkar, "Differentially private meta-learning," in *Proc. Int. Conf. Learning Representations*, 2020.

[25] T. Li, A. K. Sahu, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Conf. Machine Learning and Systems*, 2020.

[26] T. Li, M. Sanjabi, and V. Smith, "Fair resource allocation in federated learning," in *Proc. Int. Conf. Learning Representations*, 2020.

[27] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *Proc. Int. Conf. Learning Representations*, 2020.

[28] 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. Advances in Neural Information Processing Systems*, 2017, pp. 5330–5340.

[29] L. W. Mackey, M. I. Jordan, and A. Talwalkar, "Divide-and-conquer matrix factorization," in *Proc. Advances in Neural Information Processing Systems*, 2011, pp. 1134–1142.

[30] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TinyDB: An acquisitional query processing system for sensor networks," *Trans. Database Syst.*, vol. 30, no. 1, pp. 122–173, 2005.

[31] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[32] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *Proc. Int. Conf. Learning Representations*, 2018.

[33] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proc. Int. Conf. Machine Learning*, 2019, pp. 4615–4625.

[34] M. E. Nergiz and C. Clifton, "δ-presence without complete world knowledge," *IEEE Trans. Knowledge and Data Eng.*, vol. 22, no. 6, pp. 868–883, 2010. doi: 10.1109/TKDE.2009.125.

[35] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft, "Privacy-preserving ridge regression on hundreds of millions of records," in *Proc. Symp. Security and Privacy*, 2013, pp. 334–348. doi: 10.1109/SP.2013.30.

[36] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. Int. Conf. Communications*, 2019, pp. 1–7. doi: 10.1109/ICC.2019.8761315.

[37] B. Recht, C. Re, S. Wright, and F. Niu, "HOGWILD!: A lock-free approach to parallelizing stochastic gradient descent," in *Proc. Advances in Neural Information Processing Systems*, 2011, pp. 693–701.

[38] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization. 2019. [Online]. Available: arXiv:1909.13014

[39] A. Reisizadeh, H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani, "Robust and communication-efficient collaborative learning," in *Proc. Advances in Neural Information Processing Systems*, 2019, pp. 8386–8397.

[40] A. K. Sahu, D. Jakovetic, D. Bajovic, and S. Kar, Communication-efficient distributed strongly convex stochastic optimization: Non-asymptotic rates. 2018. [Online]. Available: arXiv:1809.02920

[41] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, and S. Bakas, "Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation," in *Proc. Int. MICCAI Brainlesion Workshop*, 2018, pp. 92–104. doi: 10.1007/978-3-030-11723-8_9.

[42] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 4424–4434.

[43] V. Smith, S. Forte, C. Ma, M. Takac, M. I. Jordan, and M. Jaggi, "CoCoA: A general framework for communication-efficient distributed optimization," *J. Mach. Learning Res.*, vol. 18, no. 1, pp. 8590–8638, 2018.

[44] S. U. Stich, "Local SGD converges fast and communicates little," in *Proc. Int. Conf. Learning Representations*, 2019.

[45] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, "Communication compression for decentralized training," in *Proc. Advances in Neural Information Processing Systems*, 2018, pp. 7652–7662.

[46] C. Van Berkel, "Multi-core for mobile phones," in *Proc. Conf. Design, Automation and Test in Europe*, 2009, pp. 1260–1265.

[47] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright, "ATOMO: Communication-efficient learning via atomic sparsification," in *Proc. Advances in Neural Information Processing Systems*, 2018, pp. 1–12.

[48] J. Wang and G. Joshi, Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms. 2018. [Online]. Available: arXiv:1808.07576

[49] Y. Yao, L. Rosasco, and A. Caponnetto, "On early stopping in gradient descent learning," *Constr. Approx.*, vol. 26, no. 2, pp. 289–315, 2007. doi: 10.1007/s00365-006-0663-2.

[50] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. Int. Conf. Machine Learning*, 2018, pp. 5650–5659.

[51] D. Yin, A. Pananjady, M. Lam, D. Papailiopoulos, K. Ramchandran, and P. Bartlett, "Gradient diversity: A key ingredient for scalable distributed learning," in *Proc. Artificial Intelligence and Statistics*, 2018, pp. 1998–2007.

[52] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD for non-convex optimization with faster convergence and less communication," in *Proc. AAAI Conf. Artificial Intelligence*, 2018, pp. 5693–5700. doi: 10.1609/aaai.v33i01.33015693.

[53] S. Zhang, A. E. Choromanska, and Y. LeCun, "Deep learning with elastic averaging SGD," in *Proc. Advances in Neural Information Processing Systems*, 2015, pp. 685–693.

[54] LEAF: A benchmark for federated settings. Accessed on: Mar. 4, 2020. [Online]. Available: https://leaf.cmu.edu/

[55] TensorFlow. "TensorFlow federated: Machine learning on decentralized data." Accessed on: Mar. 4, 2020. [Online]. Available: https://www.tensorflow.org/federated