FedScale: Benchmarking Model and System Performance of Federated Learning

Fan Lai, Yinwei Dai, Xiangfeng Zhu, Harsha V. Madhyastha, Mosharaf Chowdhury

University of Michigan

1 Introduction

Federated learning (FL) is an emerging machine learning (ML) setting where a logically centralized coordinator orchestrates many distributed clients to collaboratively train or evaluate a model [8, 14]. In the presence of client heterogeneity, existing efforts have focused on optimizing the FL: (1) System efficiency: reducing computation load (e.g., using smaller models [17]) or communication traffic (e.g., local SGD [16]) for faster execution; (2) Statistical efficiency: designing data heterogeneity-aware algorithms (e.g., client clustering [12]) to obtain better training accuracy with fewer training rounds; (3) Privacy and security: developing reliable strategies (e.g., differentially private training [13]) to make FL more privacy-preserving and robust to potential attacks.

While the performance of an FL solution greatly depends on the characteristics of data, device capabilities, and participation of clients; overlooking any one aspect can mislead FL evaluation (§3), existing benchmarks for FL fall short: (1) they are limited in the versatility of data for various real-world FL applications. Instead, their datasets often contain synthetically generated partitions derived from conventional datasets and do not represent realistic characteristics (e.g., LEAF [9]); (2) they often overlook different aspects of practical FL. For example, system speed and availability of the client are largely missing (e.g., FedML [5]), which discourages efforts from considering FL system efficiency and resilience, and leads to overly optimistic statistical performance; (3) their experimental environments are unable to reproduce the practical scale of FL deployments, which again can under-report the realistic FL performance.

We present FedScale to enable comprehensive FL benchmarking. FedScale currently has 18 realistic FL datasets spanning across different scales for a wide variety of FL tasks. FedScale is available at https://github.com/SymbioticLab/FedScale.

Table 1: Statistics of partial FedScale datasets. FedScale has 18 real-world federated datasets, and client system traces.

<table>
<thead>
<tr>
<th>Category</th>
<th>Name</th>
<th>Data Type</th>
<th>#Clients</th>
<th>#Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV</td>
<td>OpenImage [3]</td>
<td>Image</td>
<td>13,771</td>
<td>1.3M</td>
</tr>
<tr>
<td></td>
<td>Charades [18]</td>
<td>Video</td>
<td>266</td>
<td>10K</td>
</tr>
<tr>
<td></td>
<td>VLOG [10]</td>
<td>Video</td>
<td>4,900</td>
<td>9.6K</td>
</tr>
<tr>
<td>NLP</td>
<td>Europarl [15]</td>
<td>Text</td>
<td>27,835</td>
<td>1.2M</td>
</tr>
<tr>
<td></td>
<td>Reddit [6]</td>
<td>Text</td>
<td>1,660,820</td>
<td>351M</td>
</tr>
<tr>
<td></td>
<td>LibriTTS [21]</td>
<td>Text</td>
<td>2,456</td>
<td>37K</td>
</tr>
<tr>
<td></td>
<td>Fox Go [2]</td>
<td>Text</td>
<td>150,333</td>
<td>4.9M</td>
</tr>
</tbody>
</table>

(1) In addition, we build an automated evaluation platform, FedScale Automated Runtime (FAR), to simplify and standardize a more realistic FL evaluation. FAR integrates real-world traces to simulate the realistic behaviors of the FL deployment, and thus can pinpoint various practical FL metrics. It can perform the training of thousands of clients in each round on a few GPUs efficiently.

2 FedScale: FL DataSet and Evaluation Platform

2.1 Realistic Workloads for Federated Learning

**Client Statistical Dataset** FedScale currently has 18 realistic FL datasets (Table 1) for a wide variety of task categories, such as image classification, object detection, language modeling, speech recognition, machine translation, and reinforcement learning. Meanwhile, these datasets cover different scales, from hundreds to millions of clients, to accommodate diverse FL scenarios. The raw data of these datasets are collected from different sources in various formats. We clean up the raw data, partition them into new FL datasets using their real client-data mapping, and streamline new datasets into consistent formats. e.g., we use the AuthorProfileUrl attribute of the OpenImage data to map data instances to clients.

**Client System Behavior Trace** We formulate the system trace of different clients using AI Benchmark [1] and MobiPerf Measurements [4] on mobiles. AI Benchmark provides the training and inference speed of diverse models (e.g., MobileNet) across a wide range of device models (e.g., Samsung Galaxy S20), while MobiPerf has collected the available cloud-to-edge network throughput of over 100k world-wide mobile clients. As specified in real FL deployments [8, 20], we focus on mobile devices that have larger than 2GB RAM and connect with WiFi. To account for the dynamics of client
availability, we clean up a large-scale user behavior dataset spanning 136k users [19] to emulate the behaviors of clients, which includes 180 million trace items of client devices (e.g., battery charge or screen lock) over a week. So we can evaluate the resilience of FL optimizations under client dynamics.

2.2 FAR: FL Evaluation Platform

Existing FL evaluation platforms can hardly reproduce the scale of practical FL deployments and fall short in providing user-friendly APIs, which requires great developer efforts to deploy new plugins. As such, we introduce FedScale Automated Runtime (FAR), an automated and easily-deployable evaluation platform, to simplify and standardize the FL evaluation under a practical setting. As shown in Figure 1, the resource manager orchestrates the available physical resource for evaluation to maximize the resource efficiency (e.g., queuing and balancing client events across machines), and FAR components will simulate real FL runtime using realistic client trace. For example, the communicator will record the simulated client communication time (\(\sigma=0\)) or computing event size (\(\sigma=0.01\)), the device monitor will simulate the client dynamics (e.g., clients rejoin or fail); and participants are running on real heterogeneous federated dataset. So it can provide various practical FL metrics, such as computation/communication cost, latency and wall clock time.

3 Experiments

We show how FedScale can help to benchmark FL efforts by experimenting with the GoogleSpeech and OpenImage dataset on 10 NVIDIA Tesla P100 GPUs. Our key takeaways are:

- **Benchmarking FL statistical efficiency**: FedScale provides various datasets to benchmark the statistical efficiency of FL efforts. As shown in Figure 2(a), FAR is more efficient to reproduce the practical FL scale than the state-of-the-art. More subtly, existing benchmarks can under-report real statistical efficiency as their inefficient platform can only support running tens of participants/round versus hundreds of clients in FAR (Figure 2(b)).

- **Benchmarking FL system efficiency**: FedScale integrates realistic FL system trace to benchmark the practical FL runtime (e.g., wall-clock time in real FL training or execution cost), whereas existing benchmarks can hardly support this need (Figure 3). We find that simply optimizing the communication or computation efficiency may not lead to faster rounds (Figure 3(a)), as the last participant can be bottlenecked by the other resource. Hence, there is an urgent need for co-optimizing the client system efficiency while being heterogeneity-aware.

- **Benchmarking FL privacy and security**: FedScale can evaluate the real FL runtime in privacy and security optimizations, such as wall-clock time, communication cost, and the number of rounds needed to leak the privacy on realistic client data. We give an example of benchmarking the DP-SGD [11, 13] with different privacy target \(\sigma\) (\(\sigma=0\) indicates no privacy enhancement) and different number of participants per round \(K\). Figure 3(b) shows that the current scale of participants (e.g., \(K=30\)) that today’s benchmarks can support can mislead privacy optimizations too, whereas the practical FL scale (\(K=100\)) supported by FedScale is more robust to the privacy constraint than that evaluated using existing platforms (\(K=30\)).
References