LEAF: A Benchmark for Federated Settings

Anonymous Authors

Abstract

Modern federated networks, such as those comprised of wearable devices, mobile phones, or autonomous vehicles, generate massive amounts of data each day. This wealth of data can help to learn models that can improve the user experience on each device. However, the scale and heterogeneity of federated data presents new challenges in research areas such as federated learning, meta-learning, and multi-task learning. As the machine learning community begins to tackle these challenges, we are at a critical time to ensure that developments made in these areas are grounded with realistic benchmarks. To this end, we propose LEAF, a modular benchmarking framework for learning in federated settings. LEAF includes a suite of open-source federated datasets, a rigorous evaluation framework, and a set of reference implementations, all geared towards capturing the obstacles and intricacies of practical federated environments.

1. Introduction

With data increasingly being generated on federated networks of remote devices, there is growing interest in empowering on-device applications with models that make use of such data (McMahan et al., 2016; McMahan & Ramage, 2017; Smith et al., 2017). Learning on data generated in federated networks, however, introduces several new obstacles:

**Statistical:** Data is generated on each device in a heterogeneous manner, with each device associated with a different (though perhaps related) underlying data generating distribution. Moreover, the number of data points typically varies significantly across devices.

**Systems:** The number of devices in federated scenarios is typically order of magnitudes larger than the number of nodes in a typical distributed setting, such as datacenter computing. In addition, each device may have significant constraints in terms of storage, computational, and communication capacities. Furthermore, these capacities may also differ across devices due to variability in hardware, network connection, and power. Thus, federated settings may suffer from communication bottlenecks that dwarf those encountered in traditional distributed datacenter settings, and may require faster on-device inference.

**Privacy and Security:** Finally, the sensitive nature of personally-generated data requires methods that operate on federated data to balance privacy and security concerns with more traditional considerations such as statistical accuracy, scalability, and efficiency (McMahan et al., 2017; Bonawitz et al., 2017).

Recent works have proposed diverse ways of dealing with these challenges, but many of these efforts fall short when it comes to their experimental evaluation. As an example, consider the federated learning paradigm, which focuses on training models directly on federated networks (McMahan et al., 2016; Smith et al., 2017; Pihur et al., 2018). Experimental works focused on federated learning broadly utilize three types of datasets: (1) datasets that do not provide a realistic model of a federated scenario and yet are commonly used, e.g., artificial partitions of MNIST, MNIST-fashion or CIFAR-10 (McMahan et al., 2016; Konečný et al., 2016; Geyer et al., 2017; Bagdasaryan et al., 2018; Kemp et al., 2018; Ulm et al., 2018; Wang et al., 2018); (2) realistic but proprietary federated datasets, e.g., data from an unnamed social network in (McMahan et al., 2016), crowdsourced voice commands in (Leroy et al., 2018), and proprietary data by Huawei in (Chen et al., 2018); and (3) realistic federated datasets that are derived from publicly available data, but which are not straightforward to reproduce, e.g., FaceScrub in (Melis et al., 2018), Shakespeare in (McMahan et al., 2016) and Reddit in (Konečný et al., 2016; McMahan et al., 2018; Bagdasaryan et al., 2018).

Along the same lines of federated learning, meta-learning is another learning paradigm that could use more realistic benchmarks. The paradigm is a natural fit for federated settings, as the different devices can be easily interpreted as meta-learning tasks (Chen et al., 2018). However, the artificially generated tasks considered in popular benchmarks such as Omniglot (Lake et al., 2011; Finn et al., 2017; Vinyals et al., 2016; Snell et al., 2017) and miniImageNet (Ravi & Larochelle, 2016; Finn et al., 2017; Vinyals et al., 2016; Snell et al., 2017) fail to challenge the current approaches in ways that real-world problems would. More recently, (Triantafillou et al., 2019) proposed Meta-
as a more realistic meta-learning benchmark, but tasks still have no real-world interpretation. All of these datasets could thus be categorized as the first type mentioned above (unrealistic yet popular).

As a final example, LEAF’s datasets can allow researchers and practitioners to test multi-task learning (MTL) methods in regimes with large numbers of tasks and samples, contrary to traditional MTL datasets (e.g., the popular Landmine Detection (Zhang & Schneider, 2010; Murugesan & Carbonell, 2017; Xue et al., 2007; Smith et al., 2017), Computer Survey (Argyriou et al., 2008; Agarwal et al., 2010; Kumar & Daume III, 2012) and Inner London Education Authority School (Murugesan & Carbonell, 2017; Lee et al., 2016; Agarwal et al., 2010; Argyriou et al., 2008; Kumar & Daume III, 2012) datasets have at most 200 tasks each).

In this work, we aim to bridge the gap between artificial datasets that are popular and accessible for benchmarking, and those that realistically capture the characteristics of a federated scenario but that, so far, have been either proprietary or difficult to process. Moreover, beyond establishing a suite of federated datasets, we propose a clear methodology for evaluating methods and reproducing results. To this end, we present LEAF, a modular benchmarking framework geared towards learning in massively distributed federated networks of remote devices.

2. LEAF

LEAF is an open-source benchmarking framework for federated settings. It consists of (1) a suite of open-source datasets, (2) an array of statistical and systems metrics, and (3) a set of reference implementations. As shown in Figure 1, LEAF’s modular design allows these three components to be easily incorporated into diverse experimental pipelines. We now detail LEAF’s core components.

![Figure 1. LEAF modules and their connections.](image)

Datasets: We have curated a suite of realistic federated datasets for LEAF. We focus on datasets where (1) the data has a natural key generation process (where each key refers to a particular device); (2) the data is generated from networks of thousands to millions of devices; and (3) the number of data points is skewed across devices. Currently, LEAF consists of three datasets:

- **Federated Extended MNIST (FEMNIST)**, which serves as a similar (and yet more challenging) benchmark to the popular MNIST (LeCun, 1998) dataset. It is built by partitioning the data in Extended MNIST (Cohen et al., 2017) based on the writer of the digit/character.
- **Sentiment140** (Go et al., 2009), an automatically generated sentiment analysis dataset that annotates tweets based on the emoticons present in them. In this dataset, each device is a different twitter user.
- **Shakespeare**, a dataset built from The Complete Works of William Shakespeare (William Shakespeare. The Complete Works of William Shakespeare; McMahan et al., 2016). Here, each speaking role in each play is considered a different device.

We provide statistics on these datasets in Table 1. In LEAF, we provide all necessary pre-processing scripts for each dataset, as well as small/full versions for prototyping and final testing. Moving forward, we plan to add datasets from different domains (e.g. audio, video) and to increase the range of machine learning tasks (e.g. text to speech, translation, compression, etc.).

Metrics: Rigorous evaluation metrics are required to appropriately assess how a learning solution behaves in federated scenarios. Currently, LEAF establishes an initial set of metrics chosen specifically for this purpose. For example, we introduce metrics that better capture the entire distribution of performance across devices: performance at the 10th and 90th percentiles and performance stratified by natural hierarchies in the data (e.g. play in the case of the Shakespeare dataset). We also introduce metrics that account for the amount of computing resources needed from the edge devices in terms of number of FLOPS and number of bytes downloaded/uploaded. Finally, LEAF also recognizes the importance of specifying how the accuracy is weighted across devices, e.g., whether every device is equally important, or every data point equally important (implying that power users/devices get preferential treatment). Notably, considering stratified systems and accuracy metrics is particularly important in order to evaluate whether a method will systematically exclude groups of users (e.g., because they have lower end devices) and/or will underperform for segments of the population (e.g., because they produce less data).

Reference implementations: In order to facilitate repro-
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of devices</th>
<th>Total samples</th>
<th>Samples per device</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>mean</td>
<td>stdev</td>
</tr>
<tr>
<td>FEMNIST</td>
<td>3,550</td>
<td>805,263</td>
<td>226.83, 88.94</td>
</tr>
<tr>
<td>Sent140</td>
<td>660, 120</td>
<td>1,600, 498</td>
<td>2.42, 4.71</td>
</tr>
<tr>
<td>Shakespeare</td>
<td>1, 129</td>
<td>4,226, 158</td>
<td>3,743.28, 6,212.26</td>
</tr>
</tbody>
</table>

3. LEAF in action

We now show a glimpse of LEAF in action. In particular, we highlight three of LEAF’s characteristics:

**LEAF enables reproducible science:** To demonstrate the reproducibility enabled via LEAF, we focus on qualitatively reproducing the results that (McMahan et al., 2016) obtained on the Shakespeare dataset for a next character prediction task. In particular, it was noted that for this particular dataset, the FedAvg method surprisingly diverges as the number of local epochs increases. This is therefore a critical setting to understand before deploying methods such as FedAvg. To show how LEAF allows for rapid prototyping of this scenario, we use the reference FedAvg implementation and subsample 118 devices (around 5% of the total) in our Shakespeare data (which can be easily done through our framework). Results are shown in Figure 2, where we indeed see similar divergence behavior in terms of the training loss as we increase the number of epochs.

**LEAF provides granular metrics:** As illustrated in Figure 3 and Figure 4, our proposed systems and statistical metrics are important to consider when serving multiple clients simultaneously. For statistical metrics, in Figure 3 we show the effect of varying the minimum number of samples per user in Sentiment140 (which we denote as $k$). We see that, while median performance degrades only slightly with data-deficient users (i.e., $k = 3$), the 25th percentile (bottom of box) degrades dramatically. Meanwhile, for systems metrics, we run minibatch SGD and FedAvg for FEMNIST and calculate the systems budget needed to reach an accuracy threshold of 0.75 in Figure 4. We characterize the budget in terms of total number of FLOPS across all devices and total number of bytes uploaded to network. Our results demonstrate the improved systems profile of FedAvg when it comes to the communication vs. local computation trade-off, though we note that in general methods may vary across these two dimensions, and it is thus important to consider both aspects depending on the problem at hand.

**LEAF is modular:** To demonstrate LEAF’s modularity, we incorporate its Datasets module into two different experimental pipelines besides FedAvg (which has been our focus so far). In particular, we wish to validate the hypothesis that personalization strategies (be it MTL or meta-learning) outperform competing approaches in statistically heteroge-
Figure 3. Statistical analyses for Sent140. $k$ is the minimum number of samples per user. Orange lines represent the median device accuracy, green triangles represent the mean, boxes cover the 25th and 75th percentile, and whiskers cover the 10th to the 90th percentile. We subsample 50% of the data, pick 2 clients per round of FedAvg and use a learning rate of $3 \cdot 10^{-4}$.

1. Our first pipeline explores our hypothesis in regimes where each device holds little data. We use three different kinds of models:
   - A global SVM which is trained in all of the devices’ data at once (Global-SVM).
   - A local SVM per device that is trained solely on the device’s data (Local-SVM).
   - The same SVM model but trained in the multi-task setting presented in (Smith et al., 2017) (MTL-SVM).

2. Our second pipeline corroborates the hypothesis in regimes with no restrictions on the amount of data per device. To do this, we run the popular algorithm Reptile (Nichol et al., 2018) (which can be shown to be a re-weighed, fine-tuned version of FedAvg) over FEMNIST and compare it against FedAvg when trained under similar conditions.

Results for both sets of experiments are presented in Table 2. For the first set of experiments, we re-cast FEMNIST as a binary classification task (digits vs. characters) and discard devices with more than 192 samples. For the second set, we run each algorithm for 1,000 rounds, use 5 clients per round, a local learning rate of $10^{-3}$, a training mini-batch size of 10 for 5 mini-batches, and evaluate on an unseen set of test devices. Furthermore, for Reptile we use a linearly decaying meta-learning rate that goes from 2 to 0, and evaluate by fine-tuning each test device for 50 mini-batches of size 5.

It is clear that the personalized strategies outperform the competing approaches.

Table 2. Results for different personalization pipelines on FEMNIST. For all pipelines we subsampled 5% of the data and weighted the accuracies per device.

<table>
<thead>
<tr>
<th>Method</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global-SVM</td>
<td>73.7%</td>
</tr>
<tr>
<td>Local-SVM</td>
<td>82.5%</td>
</tr>
<tr>
<td>MTL-SVM</td>
<td>84.88%</td>
</tr>
<tr>
<td>Reptile</td>
<td>80.24%</td>
</tr>
<tr>
<td>FedAvg</td>
<td>74.71%</td>
</tr>
</tbody>
</table>

4. Conclusion

We present LEAF, a modular benchmarking framework for learning in federated settings, or ecosystems marked by massively distributed networks of devices. Learning paradigms applicable in such settings include federated learning, meta-learning, multi-task learning, and on-device learning. LEAF allows researchers and practitioners in these domains to reason about new proposed solutions under more realistic assumptions than previous benchmarks. We intend to keep LEAF up to date with new datasets, metrics and open-source solutions in order to foster informed and grounded progress in this field.
References


