Communication-Efficient On-Device Machine Learning: Federated Distillation and Augmentation under Non-IID Private Data

Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Seong-Lyun Kim
Yonsei University
{ejjeong,seoh,hskim,slkim}
@ramo.yonsei.ac.kr

Jihong Park, Mehdi Bennis
University of Oulu
{jihong.park,mehdi.bennis}
@oulu.fi

Abstract

On-device machine learning (ML) enables the training process to exploit a massive amount of user-generated private data samples. To enjoy this benefit, inter-device communication overhead should be minimized. With this end, we propose *federated distillation (FD)*, a distributed model training algorithm whose communication payload size is much smaller than a benchmark scheme, federated learning (FL), particularly when the model size is large. Moreover, user-generated data samples are likely to become non-IID across devices, which commonly degrades the performance compared to the case with an IID dataset. To cope with this, we propose *federated augmentation (FAug)*, where each device collectively trains a generative model, and thereby augments its local data towards yielding an IID dataset. Empirical studies demonstrate that FD with FAug yields around 26x less communication overhead while achieving 95-98% test accuracy compared to FL.

1 Introduction

Big training dataset kickstarted the modern machine learning (ML) revolution. On-device ML can fuel the next evolution by allowing access to a huge volume of private data samples that are generated and owned by mobile devices [1, 2]. Preserving data privacy facilitates such access, in a way that a global model is collectively trained not by directly sharing private data but by exchanging the local model parameters of devices, as exemplified by *federated learning (FL)* [3–9].

Unfortunately, in FL, performing the training process at each device side entails communication overhead being proportional to model sizes, forbidding the use of large-sized models. Furthermore, a user-generated training dataset is likely to be non-IID across devices. Compared to its IID-data counterpart, it decreases the prediction accuracy by up to 11% for MNIST and 51% for CIFAR-10 under FL [9]. The reduced accuracy can partly be restored by exchanging data samples, which may however induce an excessive amount of communication overhead and privacy leakage.

On this account we seek for a communication-efficient on-device ML approach under non-IID private data. For communication efficiency, we propose *federated distillation (FD)*, a distributed online knowledge distillation method whose communication payload size depends not on the model size but on the output dimension. Prior to operating federated distillation, we rectify the non-IID training dataset via *federated augmentation (FAug)*, a data augmentation scheme using a generative adversarial network (GAN) that is collectively trained under the trade-off between privacy leakage and communication overhead. The trained GAN empowers each device to locally reproduce the data samples of all devices, so as to make the training dataset become IID.
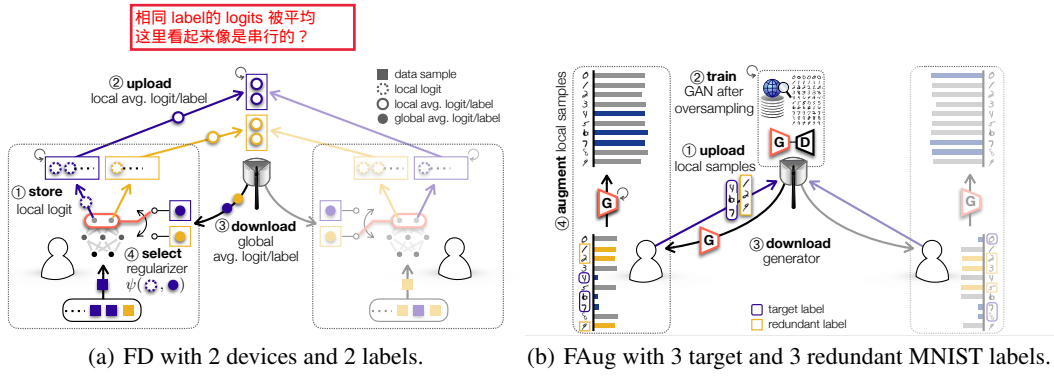
(a) FD with 2 devices and 2 labels.　　(b) FAug with 3 target and 3 redundant MNIST labels.

Figure 1: Schematic overview of federated distillation (FD) and federated augmentation (FAug).

2 Federated distillation

Traditional distributed training algorithms exchange local model parameters every epoch. It gives rise to significant communication overhead in on-device ML where mobile devices are wirelessly interconnected. FL reduces the communication cost by exchanging model parameters at intervals [3–9]. On top of such periodic communication, the proposed FD exchanges not the model parameters but the model output, allowing on-device ML to adopt large-sized local models.

The basic operation procedure of FD follows an online version of knowledge distillation [10], also known as *co-distillation (CD)* [11]. In CD, each device treats itself as a student, and sees the mean model output of all the other devices as its teacher's output. Each model output is a set of logit values normalized via a softmax function, hereafter denoted as a *logit vector* whose size is given by the number of labels. The teacher-student output difference is periodically measured using cross entropy that becomes the student's loss regularizer, referred to as a *distillation regularizer*, thereby obtaining the knowledge of the other devices during the distributed training process.

CD is however far from being communication-efficient. The reason is that each logit vector is associated with its input training data sample. Therefore, to operate knowledge distillation, both teacher and student outputs should be evaluated using an identical training data sample. This does not allow periodic model output exchanges. Instead, it requires exchanging either model outputs as many as the training dataset size, or model parameters so that the reproduced teacher model can locally generate outputs synchronously with the student model [11].

To rectify this, each device in FD stores per-label mean logit vectors, and periodically uploads these *local-average logit vectors* to a server. For each label, the uploaded local-average logit vectors from all devices are averaged, resulting in a *global-average logit vector* per label. The global-average logit vectors of all labels are downloaded to each device. Then, when each device calculates the distillation regularizer, its teacher's output is selected as the global-average logit vector associated with the same label as the current training sample's label.

The said operation of FD is visualized in Fig. 1(a), and is detailed by Algorithm 1. Notations are summarized as follows. The set $\mathbb{S}$ denotes the entire training dataset of all devices, and B represents the batch of each device. The function $F(w, a)$ is the logit vector normalized by the softmax function, where w and a are the model's weight and input. The function $\phi(p, q)$ is the cross entropy between p and q, which is used for both loss function and distillation regularizer. The term η is a constant learning rate, and γ is a weight parameter for the distillation regularizer. At the i-th device, $\bar{F}_{k,\ell}^{(i)}$ is the local-average logit vector at the k-th iteration when the training sample belongs to the ℓ-th ground-truth label, $\hat{F}_{k,\ell}^{(i)}$ is the global-average logit vector that equals $\hat{F}_{k,\ell}^{(i)} = \sum_{j \neq i} \bar{F}_{k,\ell}^{(j)}/(M-1)$ with a number M of devices, and $cnt_{k,\ell}^{(i)}$ is the number of samples whose ground-truth label is ℓ.

3 Federated augmentation

The non-IID training dataset of on-device ML can be corrected by obtaining the missing local data samples at each device from the other devices [9]. This may however induce significant

2

Algorithm 1 Federated distillation (FD)

Require: Prediction function: $F(w, input)$, Loss function: $\phi(F, label)$, Ground-truth label: y_{input}

1: **while** not converged **do**
2:     **procedure** LOCAL TRAINING PHASE (at each device)
3:         **for** n steps **do** : $B, y_B \leftarrow \mathbb{S}$ `1, KD    local model`
4:             **for** sample $b \in B$ **do**
5:                  $w^{(i)} \leftarrow w^{(i)} - \eta \nabla \{ \phi(F(w^{(i)}, b), y_b) + \gamma \cdot \phi(F(w^{(i)}, b), \hat{F}^{(i)}_{k, y_b}) \}$
6:                  $F^{(i)}_{k, y_b} \leftarrow F^{(i)}_{k, y_b} + F(w^{(i)}, b), cnt^{(i)}_{k, y_b} \leftarrow cnt^{(i)}_{k, y_b} + 1$
7:         **for** label $\ell = 1, 2, \cdots, L$ **do**
8:              $\bar{F}^{(i)}_{k, \ell} \leftarrow F^{(i)}_{k, \ell} / cnt^{(i)}_{k, \ell}$: **return** $\bar{F}^{(i)}_{k, \ell}$ to server   `2    logits avg (per-class)`
9:     **procedure** GLOBAL ENSEMBLING PHASE (at the server)
10:         **for** each device $i = 1, 2, \cdots, M$ **do**
11:             **for** label $\ell = 1, 2, \cdots, L$ **do**
12:                  $\bar{F}_{k, \ell} \leftarrow \bar{F}_{k, \ell} + \bar{F}^{(i)}_{k, \ell}$ `3,    logits avg`
13:         **for** each device $i = 1, 2, \cdots, M$ **do**
14:             **for** label $\ell = 1, 2, \cdots, L$ **do**
15:                  $\hat{F}^{(i)}_{k+1, \ell} \leftarrow \bar{F}_{k, \ell} - \bar{F}^{(i)}_{k, \ell}, \hat{F}^{(i)}_{k+1, \ell} \leftarrow \hat{F}^{(i)}_{k+1, \ell} / (M - 1)$: **return** $\hat{F}^{(i)}_{k+1, \ell}$ to device i
    end while

`4    client    logits`
`ps.    global model`

---

communication overhead, especially with a large number of devices. Instead, we propose FAug where each device can locally generate the missing data samples using a generative model.

The generative model is trained at a server with high computing power and a fast connection to the Internet. Each device in FAug recognizes the labels being lacking in data samples, referred to as *target labels,* and uploads few seed data samples of these target labels to the server over wireless links. The server oversamples the uploaded seed data samples, e.g., via Google's image search for visual data, so as to train a conditional GAN [12]. Finally, downloading the trained GAN's generator empowers each device to replenish the target labels until reaching an IID training dataset, which significantly reduces the communication overhead compared to direct data sample exchanges. This procedure is illustrated in Fig. 1(b)

The operation of FAug needs to guarantee the privacy of the user-generated data. In fact, each device's data generation bias, i.e., target labels, may easily reveal its privacy sensitive information, e.g., patients' medical checkup items revealing the diagnosis result. To keep these target labels private from the server, the device additionally uploads redundant data samples from the labels other than the target labels. The privacy leakage from each device to the server, denoted as *device-server privacy leakage (PL)*, is thereby reduced at the cost of extra uplink communication overhead. At the i-th device, its device-server PL is measured as $|\mathbb{L}^{(i)}_t| / (|\mathbb{L}^{(i)}_t| + |\mathbb{L}^{(i)}_r|)$, where $|\mathbb{L}^{(i)}_t|$ and $|\mathbb{L}^{(i)}_r|$ denote the numbers of target and redundant labels, respectively.

`Generator`
`client  同`

The target label information of a device can also be leaked to the other devices since they share a collectively trained generator. Indeed, a device can infer the others' target labels by identifying the generable labels of its downloaded generator. This privacy leakage is quantified by *inter-device PL*. Provided that the GAN is always perfectly trained for all target and redundant labels, the inter-device PL of the i-th device is defined as $|\mathbb{L}^{(i)}_t| / |\bigcup_{j=1}^{M} (\mathbb{L}^{(j)}_t \cup \mathbb{L}^{(j)}_r)|$. Note that the inter-device PL is minimized when its denominator equals to the maximum value, i.e., the number of the entire labels. This minimum leakage can be achieved so long as the number of devices is sufficiently large, regardless of the sizes of the target and redundant labels.

4   Evaluation

In this section, we evaluate the proposed FD and FAug under a non-IID MNIST training dataset that is constructed by the following procedure. In the MNIST training dataset with 55,000 samples, we uniformly randomly select 2,000 samples, and allocate them to each device. A set of these 2,000 samples is divided into 10 subsets according to the ground-truth labels. Then, at each device, we uniformly randomly select target labels with a pre-defined number of target labels, and eliminate around 97.5% of the samples in the target labels such that each target label contains 5 samples.

Methods	Accuracy w.r.t. the number of devices					Communication cost			
	2	4	6	8	10	Logits	Model parameters	Samples	Total (bits)
FD + FAug	0.8464	0.8526	0.8498	0.8480	**0.8642**	3,200	1,493,520	15	**47,989,120**
FD (non-IID)	0.7230	0.7304	0.6951	0.6839	**0.7524**	3,200	-	-	**102,400**
FL + FAug	0.9111	0.8654	0.8956	0.9101	**0.9259**	-	39,882,256	15	**1,276,326,272**
FL (non-IID)	0.8077	0.8684	0.8738	0.8878	**0.9060**	-	38,388,736	-	**1,228,439,552**

Table 1: Test accuracy and communication cost (3 target labels, no redundant label).

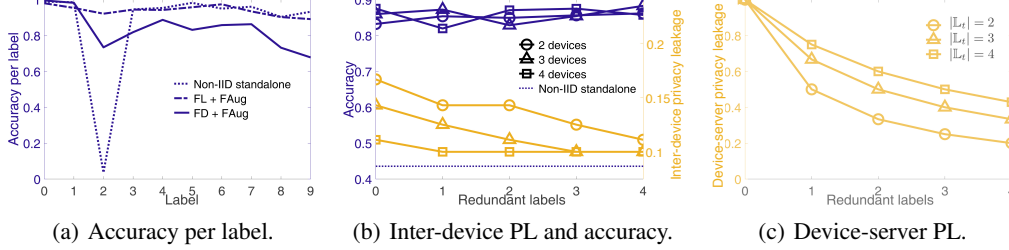(a) Accuracy per label. (b) Inter-device PL and accuracy. (c) Device-server PL.

Figure 2: Test accuracy and privacy leakage (PL) under a non-IID MNIST dataset: (a) accuracy per label under FL or FD with FAug, compared to the non-IID standalone case; (b) inter-device PL and accuracy under FD with FAug; and (c) device-server PL under FAug for different numbers of target labels.

With this non-IID MNIST training dataset, each device has a 5-layer convolutional neural network (CNN) that consists of: 2 convolutional layers, 1 max-pooling layer, and 2 fully-connected layers. The device conducts its local training with the batch size set as 64. As a benchmark scheme against FD, we consider FL [4] with or without FAug. In both FD and FL, each device exchanges the desired information at every 250 local iterations (n=250), denoted as a global iteration, up to 16 global iterations. For each global iteration, FD exchanges 100 logits for uplink (i.e., $\{\bar{F}_{k,\ell}^{(i)}\}_{\ell=1}^{L}$) and another 100 logits for downlink (i.e., $\{\hat{F}_{k,\ell}^{(i)}\}_{\ell=1}^{L}$), each of which comprises 10 logit vectors containing 10 elements. On the other hand, FL exchanges the CNN's 1,199,648 model parameters per global iteration equally for both uplink and downlink.

In FAug, the server has a conditional GAN consisting of a 4-layer generator neural network and a 4-layer discriminator neural network. When downloading a trained generator, the communication overhead is proportional to the generator's model size, given as 1,493,520 parameters.

At a uniformly randomly chosen reference device, Table 1 provides the test accuracy and communication cost of FD and FL, with or without FAug. For FD, the communication cost is defined as the number of exchanged logits, whereas the cost for FL is given as the number of exchanged model parameters. With FAug, the communication cost comprises the number of uploading data samples and the number of downloading model parameters of the trained generator. For each method, the total communication cost is evaluated by considering that each pixel of MNIST sample (28x28 pixels) occupies 8 bits, while each logit and each model parameter equally consume 32 bits.

In Table 1, compared to FL, we observe that FD significantly reduces the communication cost while slightly compromising the test accuracy. The reduced accuracy of FD can be complemented by FAug, without incurring severe communication overhead. To be specific, compared to FL with a different number of devices, FD achieves 77-90% test accuracy that can be improved by FAug as 95-98% test accuracy. The aggregate communication cost of FD and FAug is still around 26x smaller than the cost of FL. Note that in return for the communication overhead, FL is more robust against non-IID dataset. This is observed via FAug that increases the test accuracy of FD by 7-22% and the accuracy of FL by 0.8-2.7%, compared to the cases without FAug, i.e., non-IID dataset.

Fig. 2(a) illustrates the per-label test accuracy when '2' is the target label. For the target label, the standalone training of the reference device under the original non-IID dataset yields 3.585% test accuracy, which is increased via FAug with FD or FL by 73.44% or 92.19%, validating the effectiveness of FAug in combination with both FD and FL. For the entire labels, Fig. 2(b) shows that FD with FAug achieves around 2x higher test accuracy compared to the standalone training, regardless

of the number of devices and of the number of redundant labels. For more devices or redundant labels, the inter-device PL decreases, since both aspects respectively increase the denominator $\left| \bigcup_{j=1}^{M} (\mathbb{L}_t^{(j)} \cup \mathbb{L}_r^{(j)}) \right|$ of the inter-device PL. Similarly, Fig. 2(c) describes that the device-server PL decreases with the number of redundant labels, and increases with the number of target labels.

5   Concluding remarks

Towards enabling on-device ML, we introduced FD and FAug that are communication-efficient training and data augmentation algorithms, respectively. Empirical studies showed their effectiveness that achieves comparably high accuracy with much smaller communication overhead compared to FL.

In future work, the performance of FD can further be improved with a slight modification. In fact, model output accuracy increases as the training progresses. During the local logit averaging process, it is thus better to take a weighted average in a way that the weight increases with the local computation time. Furthermore, FD and FL can be combined towards balancing communication-efficiency and accuracy. As an example, one can exploit FD in the uplink and FL in the downlink, on the ground that the downlink wireless communication links are commonly faster than the uplink [13]. Lastly, using the differential privacy framework [8], the privacy guarantee of FAug can be ameliorated by inserting a proper amount of noise into the uploading seed data samples, which is an interesting topic for future research.

References

[1] S. Ravi. On-Device Machine Intelligence. Google AI Blog. Feb. 2017 [Online, Accessed: 2018-11-23]. URL: https://ai.googleblog.com/2017/02/on-device-machine-intelligence.html.

[2] M. Bennis. Smartphones Will Get Even Smarter With On-Device Machine Learning. IEEE Spectrum. Mar. 2018 [Online, Accessed: 2018-11-23]. URL: https://spectrum.ieee.org/tech-talk/telecom/wireless/smartphones-will-get-even-smarter-with-ondevice-machine-learning.

[3] J. Konečný, H. B. McMagan, and D. Ramage. Federated Optimization: Distributed Machine Learning for On-Device Intelligence [Online]. ArXiv preprint: https://arxiv.org/abs/1610.02527.

[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. Proc. AISTATS, Fort Lauderdale, FL, USA, Apr. 2017.

[5] H. Kim, J. Park, M. Bennis, and S.-L. Kim. On-Device Federated Learning via Blockchain and its Latency Analysis [Online]. ArXiv preprint: https://arxiv.org/abs/1808.03949.

[6] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah. Distributed Federated Learning for Ultra-Reliable Low-Latency Vehicular Communications [Online]. ArXiv preprint: https://arxiv.org/abs/1807.08127.

[7] T. Nishio and R. Yonetani. Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge [Online]. ArXiv preprint: https://arxiv.org/abs/1804.08333.

[8] R. C. Geyer, T. Klein, and M. Nabi. Differentially Private Federated Learning: A Client Level Perspective Differentially Private Federated Learning: A Client Level Perspective . NIPS Machine Learning on the Phone and other Consumer Devices Workshop, Long Beach, CA, USA, Dec. 2017.

[9] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra. Federated Learning with Non-IID Data [Online]. arXiv preprint: https://arxiv.org/abs/1806.00582.

[10] G. E. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. NIPS Deep Learning Workshop, Montréal, Canada, Dec. 2014.

[11] R. Anil, G. Pereyra, A. Passos, R. Ormandi, G. E. Dahl, and G. E. Hinton. Large Scale Distributed Neural Network Training Through Online Distillation [Online]. ArXiv preprint: https://arxiv.org/abs/1804.03235.

[12] M. Mirza, and S. Osindero. Conditional Generative Adversarial Nets [Online]. ArXiv preprint: https://arxiv.org/abs/1411.1784.

[13] J. Park, S.-L. Kim, and J. Zander. Tractable Resource Management with Uplink Decoupled Millimeter-Wave Overlay in Ultra-Dense Cellular Networks. IEEE Trans. Wireless Commun., 15(6):4362–4379, Jun. 2016.