CROSS-SILO FEDERATED TRAINING IN THE CLOUD WITH DIVERSITY SCALING AND SEMI-SUPERVISED LEARNING

Kishore Nandury, Anand Mohan, Frederick Weber

Amazon Alexa, Bangalore, India

ABSTRACT

Federated learning is a machine learning approach that allows a loose federation of trainers to collaboratively improve a shared model, while making minimum assumptions on central availability of data. In cross-siloed federated learning, data is partitioned into silos, each with an associated trainer. This work presents results from training an end-to-end ASR model with cross-silo federated learning system. We propose a novel aggregation algorithm that takes update diversity into account and significantly outperforms Federated Averaging (FedAvg). The system design used in this paper allows joint training with human transcribed and semi-supervised (SSL) data, yielding 7.6% relative word error rate reduction on head test set and 13.9% on tail test set, when using 20kHr of SSL data. Gains further improve to 13.8% and 20.5% respectively when SSL data is increased from 20kHr to 200kHr.

***Index Terms*—** Federated learning, Automatic speech recognition, Semi-supervised learning

1. INTRODUCTION

In Federated Learning (FL) [1], a global model is collaboratively trained by many clients over decentralized data. Of this, two typical settings are Cross-silo FL and cross-device FL. In both settings, data is generated locally and remains decentralized (no inter-client sharing) and a central server orchestrates training across multiple clients. In Cross-silo FL, data could be distributed across pre-defined silos such as organizations or geographical locations. In this setup, typical number of clients is around 2-100 [2]. Cross-device FL on the other hand runs on large number of clients (upto 10^{10} devices) with only a fraction of clients participating in each training round [2]. Cross-silo FL has recently gained traction in domains such as medical and health care [3, 4, 5], finance [6] and manufacturing [7].

Compared to the traditional distributed training system, a cross-silo FL system is different in: a) Optimization process: A trainer can only access its own local data defined by a silo and can at best solve a local optimization problem on its own. b) Non-availability of ground truth: Ground truth may or may not be available for local data. In case of ASR, ground truth is available only for human transcribed data, which is much small compared to the data we want to utilize in FL training. c) Non IID data: Local data set may not be representative of population distribution (features and labels) and the number of training examples can also be non-uniform across clients. d) Resource constraints: Trainers can be constrained with respect to compute and network bandwidth.

A typical FL system utilizes user generated inputs to approximate labelled data [8, 9, 10] since supervised labels are not available at clients. In case of ASR, machine generated transcription could be used as an approximation, when human transcript is not available. Using semi-supervised data for federated learning is a much less explored area and is gaining interest recently [11].

In this paper, we present learnings from cross-silo federated training of an end-to-end ASR system, taking Recurrent Neural Network Transducer (RNN-T) [12, 13] as the ASR model architecture. Our contributions are two fold 1) We analyze the effect of update diversity on training and propose Federated Averaging with Diversity Scaling (FedAvg-DS) algorithm that accounts for update diversity and significantly outperforms Federated Averaging (FedAvg). 2) We combine cross-silo federated learning with access qualified trainers to jointly train with both transcribed and semi-supervised data, to achieve atleast 7% WERR on head test set and 13.9% WERR on tail test set.

This paper is organized as follows: Section 2 introduces system architecture that combines Federated training with Semi-supervised learning. Diversity scaling and FedAvg-DS are presented in Section 3. Results are covered in Section 4, followed by conclusion in Section 5.

2. CROSS-SILOED FEDERATED LEARNING IN CLOUD

FL training framework in cloud is built on top of gRPC [14] with three core components 1) FL server 2) Trainer 3) Aggregator. In this cloud based FL system, all trainer clients run in AWS cloud and consume both Human transcribed (HT) and Semi-supervised data (SSL) for training ASR model. To jointly train on these data sources, we define two types of trainer clients based on data access privileges: a) Privileged trainers (PT) b) Restricted trainers (RT). Privileged trainers can use HT data for training, while restricted trainers can only

access SSL data. HT data is de-identified, then transcribed by human annotators, and partitioned across privileged trainers. SSL data is prepared on cloud from de-identified, untranscribed recordings contributed by participating devices. Alexa devices are partitioned across restricted trainer clients in a many-to-one mapping, hence SSL data generated from one or more devices can used for training at any given trainer client.

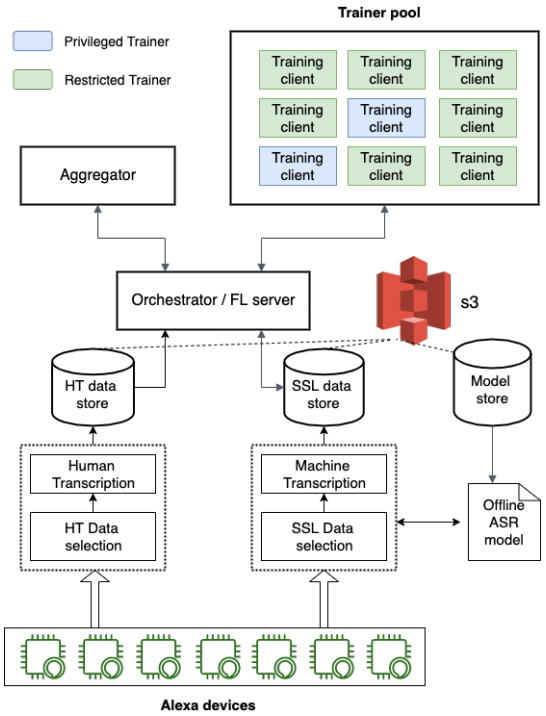

Fig. 1. Cross-siloed FL system in cloud

Un-transcribed recordings are processed through SSL data processing pipeline on cloud and appended to SSL data store. SSL data processing pipeline contains two main components 1) SSL data selection 2) Machine transcription. SSL data selection module randomly selects a subset of un-transcribed data in confidence range of 600-900 for machine transcription. One of the main advantages of cloud side FL is the ability to run a complex ASR model for generating machine transcripts for un-transcribed data, which may not be otherwise possible to do on-device.

FL server in Figure 1 maintains state of overall system and sequences training and aggregation steps to iteratively improve a global model across multiple rounds. Each trainer takes one or more optimization steps on siloed data specified by FL server, to generate a model update at the end of round. Aggregator combines multiple trainer updates to a single update over global model, typically via averaging.

3. DIVERSITY SCALING

At the beginning of t^{th} FL round, server shares the current global model (w^t) to all participating trainers. Starting with this model, each trainer takes multiple optimization steps on local data. Change in model parameters due to local training is communicated by each trainer client to server. In Federated Averaging (FedAvg) [1], updates from trainers are averaged and added to global model as in Equation (1), resulting in global model update at the end of round.

$$\Delta^t_{avg} = \frac{1}{K}(\Delta^t_1 + \Delta^t_2 + \ldots + \Delta^t_K) \qquad w^{t+1} = w^t + \Delta^t_{avg} \tag{1}$$

Magnitude of change in global model after R rounds of training can be written as

$$\begin{aligned} \left\|w^R - w^0\right\| &= \left\|\sum_{t=0}^{R-1}(w^{t+1} - w^t)\right\| \\ &\leq \sum_{t=0}^{R-1}\left\|w^{t+1} - w^t\right\| = \sum_{t=0}^{R-1}\left\|\Delta^t_{avg}\right\| \end{aligned} \tag{2}$$

It can be seen that $\|\Delta^t_{avg}\|$ directly contributes to upper bound on global model change and hence convergence speed. In a given round of training, if all trainer updates are exactly the same, then it can be argued that there is no diversity in updates and the multitude of trainers can be replaced by a single trainer. On the other hand, if trainer updates perfectly cancel each other after averaging, then it can be seen that $\|\Delta^t_{avg}\|$ will be 0. Compared to these two extremes, it is more likely that updates from trainers are neither perfectly coherent nor decoherent, but somewhere in between. The diversity in trainer updates can be quantified by diversity coefficient,

$$\gamma^t = \frac{\frac{1}{K}\sum_k \|\Delta^t_k\|}{\|\Delta^t_{avg}\|} \tag{3}$$

Larger value of γ is indicative of higher degree of dissimilarity between trainer updates and is very likely when training on heterogenous data or for parameters with sparse updates.

RNN-T ASR model [12, 13] used in this work contains three main components a) Encoder, b) Prediction network (also referred here as Decoder) c) Joint network. Encoder is analogous to an acoustic model and decoder to a language model. Joint network combines outputs from encoder and decoder networks to generate output token probabilities. Figure 2 plots γ for few parameters in RNN-T when training with human transcribed (HT) data. It can be observed that γ changes as training progresses and is not same for all parameters. Trainer updates are most dissimilar for decoder input embedding matrix, which has the largest γ value.

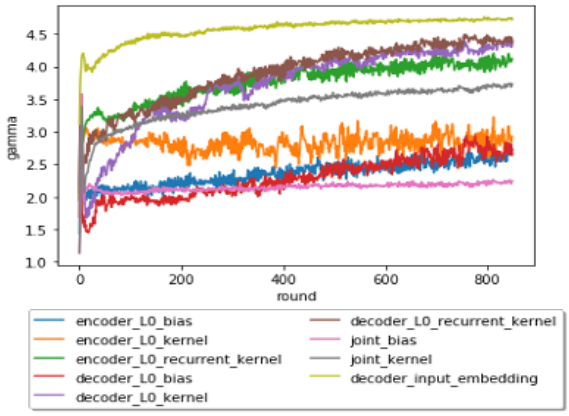

Fig. 2. Plot of gamma for few parameters in RNN-T

To compensate for the effect of update diversity, we propose FedAvg-DS, which is an extension to FedAvg, that takes γ into account via diversity scaling. Related notion of gradient diversity can be found in [15], where it was shown that high similarity between concurrent gradient updates is a cause of performance degradation for mini-batch stochastic gradient descent. Update diversity introduced here is not same as gradient diversity and we incorporate it to accelerate federated training.

A given layer or component of RNN-T model can contain multiple trainable parameters. For example, joint network consists of a kernel matrix and bias vector. We found that it is best to apply the same scaling for all parameters in a given layer. To do so, we estimate γ seperately for each parameter in a layer and take their minimum as final scaling that is applied for parameters in that layer. For brevity, layer-wise scaling is not explicity shown in Algorithm 1

Setting γ_{max} to $\sqrt{K}$, where K is the number of trainers, works well in practise. FedAvg-DS takes inter-trainer update dissimilarity into account via diversity scaling , which is not considered in existing optimization methods like Adam [16]. Since γ is calculated directly from trainer updates, there is no additional hyper-parameter to tune.

Algorithm 1: FedAvg with Diversity scaling (FedAvg-DS)

Initialize global model w^0
Initialize accelerated global model $w^0_{acc} = w^0$
for *round* $t \leftarrow 0, 1, 2, ...$ **do**
  Server
    Identify set S^t of K trainers
    Send w^t_{acc} to trainers in S^t
  foreach *trainer* k ***in*** S^t *(in parallel)* **do**
    $w \leftarrow w^t_{acc}$
    foreach i ***in*** *#steps/round* **do**
      $w \leftarrow w - \eta * \nabla f_k(w)$
    end
    Send $\Delta^t_k = w - w^t_{acc}$ to server
  end
  Server
    Update global model

$$w^{t+1} = w^t_{acc} + \Delta^t_{avg}$$

    Update accelerated global model

$$w^{t+1}_{acc} = w^t_{acc} + min(\gamma^t, \gamma_{max}) * \Delta^t_{avg}$$

    where:

$$\Delta^t_{avg} = \frac{1}{K}\sum_k \Delta^t_k \text{ and } \gamma^t = \frac{\frac{1}{K}\sum_k \|\Delta^t_k\|}{\|\Delta^t_{avg}\|}$$

end

4. RESULTS

For experiments in this paper, we use RNN-T ASR model [12, 13] consisting of an encoder with 5 LSTM layers of 1024 units each, a prediction network with 2 LSTM layers of 1024 units each, and a joint network with one dense layer of 512 units. For audio input we stack three 10 ms frames to yield 192 dimensional Low Frame Rate (LFR) feature vectors of log-Mel filter-bank energies, and apply SpecAugment [17] during training. We use 4000 word pieces learnt from transcribed data to represent output tokens. For all FL training in this section, the Adam optimizer [16] is used on trainer clients, with learning rate warm up from 1e-7 to 5e-4 in 3000 steps, constant learning rate of 5e-4 until 150K steps, followed by exponential decay to 5e-5 in 100K steps.

We report results using the WERR (Word Error Rate Reduction) metric on two de-identified test sets, based on recordings of Alexa interactions from a variety of voice-controlled devices: 1) Head test set: 170 hours of randomly selected data 2) Tail test set: 35 hours of data, each recording selected to contain at least one 'rare' word, based on a low frequency threshold. An RNN-T model trained on 10kHr of de-identified human transcribed data in a traditional distributed training setup is used as the baseline for WERR calculation.

4.1. Effect of Diversity scaling

FedAvg-DS with diversity scaling presented in 3, scales average update by γ, there by accounting for dissimilarity in updates coming from trainers. Here is a comparison of FedAvg and FedAvg-DS in a system containing 24 trainer clients, out of which 12 are privileged trainers and 12 are restricted trainers.

Privileged trainers have access to 10kHr of HT data and restricted trainers have access to a total of 20kHr of SSL data, partitioned into multi-device silos. Table 1 has results after 2500 rounds of training at 100 steps per round, It can be seen

Algorithm	%WERR	
	Head	Tail
FedAvg	1.1	3
FedAvg-DS	**6.5**	**11.1**

Table 1. Comparison between FedAvg and FedAvg-DS with 12 PT and 12 RT

that FedAvg-DS clearly outperforms FedAvg on both head and tail test sets. For the rest of paper, FedAvg-DS is the default algorithm.

4.2. Federated Learning with Semi-supervised data

FL system presented in Section 2 can utilize un-transcribed data for model training via machine transcription. To investigate the effect of SSL data on FL training, we vary the proportion of restricted trainers in the system. To investigate the effect of starting point on FL training, we consider two global model initializations for FL training a) Weak initialization, pretrained for 5K steps on HT data b) Strong initilization, pretrained for 125K steps on HT data.

For results in Table 2, starting with a weak initialization, we jointly train with 10kHr of HT data (used by privileged trainers) and 20kHr of SSL data (used by restricted trainers). Keeping total number of trainers constant at 24, we vary the number of restricted trainers from 0(0%) to 21(87.5%) to control the effect of SSL data on global model. Each model is trained for 2500 rounds with 100 steps per round.

PT:RT	Data	%WERR	
		Head	Tail
3:21	10kHr HT + 20kHr SSL	6.3	13.7
6:18	10kHr HT + 20kHr SSL	**7.6**	**13.9**
12:12	10kHr HT + 20kHr SSL	6.5	11.1
18:6	10kHr HT + 20kHr SSL	5	6.7
21:3	10kHr HT + 20kHr SSL	2.3	3.2

Table 2. WERR after 2500 rounds of training

Results from Table 2 demonstrate that a) It is possible to beneficially combine updates based on SSL data with updates based on HT data during aggregation, without mixing data at mini-batch level b) The best training configuration uses 75% of all trainers as restricted trainers, yielding 7.6% WERR on head test set and 13.9% WERR on tail test set.

We next investigate the utility of FL training to further improve an already strong initial model, which is pre-trained for 125K steps on HT data. To identify how FL training progresses at larger data scale, we have increased the total amount of SSL data from 20kHr to 200kHr and trained upto 10K rounds, while varying number of restricted trainers from 12 (50%) to 24 (100%).

Table 3 summarizes WERR results after 10K rounds of training, while Figure 3 shows how WERR progresses on tail

PT:RT	Data	%WERR	
		Head	Tail
0:24	200kHr SSL	10.4	20.6
3:21	10kHr HT + 200kHr SSL	**13.8**	**21.2**
6:18	10kHr HT + 200kHr SSL	**13.8**	20.5
12:12	10kHr HT + 200kHr SSL	13.3	18.1

Table 3. WERR after 10K rounds of training

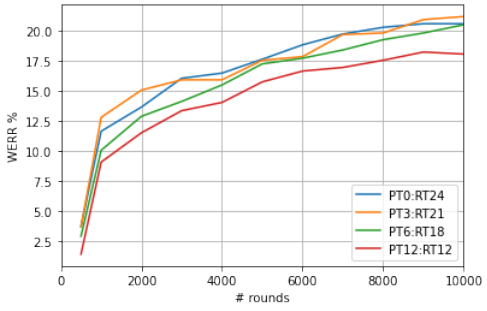

Fig. 3. WERR on tail test data for extended training at different number of rounds

test set with training round. We noticed that model performance steadily improves on both head and tail test sets with large volumes of SSL data, for all PT:RT configurations. Best results are obtained by using a small proportion of (12–25%) privileged trainers in system, yielding WERR of 13.8% on the head test set and 21.2% on the tail test set, with tail performance still improving after 10K rounds of training.

5. CONCLUSION

Diversity of updates in a federated training system has so far not been considered to improve model training. The proposed FedAvg-DS algorithm, takes update diversity into account and significantly outperforms Federted Averaging (FedAvg). In a typical FL system, it is not possible to mix Human transcribed (HT) and Semi-supervised (SSL) data at mini-batch level due to data access restrictions. To address this, we have proposed a cross-silo federated learning system that qualifies trainers based on data access rights. Privileged trainers can access Human transcribed data and Restricted trainers can only access Semi-supervised data for training. This allows us to jointly train with Human transcribed and Semi-supervised data, without having to centrally mix data at mini-batch level. Effect of SSL data on model performance can be controlled by changing the proportion of restricted trainers in the system. We found that model built with this FL system outperforms baseline Non-FL model by 7.6% on head test set and by 13.9% relative on tail test set when using 20kHr of SSL data. Gains improve to 13.8% and 20.5% respectively, when training from a stronger initial model with 200kHr of SSL data.

6. REFERENCES

[1] Brendan McMahan et. al, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*. 2017, vol. 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, PMLR.

[2] Peter Kairouz et. al, "Advances and open problems in federated learning," *CoRR*, vol. abs/1912.04977, 2019.

[3] "FeatureCloud," *https://featurecloud.eu/about*.

[4] "Federated Learning for Medical Imaging," *https://www.intel.com/content/www/us/en/artificial-intelligence/posts/federated-learning-for-medical-imaging.html*.

[5] Courtiol et. al, "Deep learning-based classification of mesothelioma improves prediction of patient outcome," *Nature medicine*, vol. 25, no. 10, pp. 1519—1525, October 2019.

[6] editor2fedai, "WeBank and Swiss Re signed Cooperation MoU," *https://www.fedai.org/news/webank-and-swiss-re-signed-cooperation-mou*.

[7] "MUSKETEER," *https://musketeer.eu/project*.

[8] Timothy Yang et. al, "Applied federated learning: Improving google keyboard query suggestions," *CoRR*, vol. abs/1812.02903, 2018.

[9] Andrew Hard et. al, "Federated learning for mobile keyboard prediction," *CoRR*, vol. abs/1811.03604, 2018.

[10] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays, "Federated learning for emoji prediction in a mobile keyboard," *CoRR*, vol. abs/1906.04329, 2019.

[11] Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang, "Federated semi-supervised learning with inter-client consistency," *ICML*, 2020.

[12] Alex Graves, "Sequence transduction with recurrent neural networks," *CoRR*, vol. abs/1211.3711, 2012.

[13] Yanzhang He et. al, "Streaming end-to-end speech recognition for mobile devices," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*. 2019, pp. 6381–6385, IEEE.

[14] Xingwei Wang, Hong Zhao, and Jiakeng Zhu, "GRPC: A communication cooperation mechanism in distributed systems," *ACM SIGOPS Oper. Syst. Rev.*, vol. 27, no. 3, pp. 75–86, 1993.

[15] Dong Yin, Ashwin Pananjady, Maximilian Lam, Dimitris S. Papailiopoulos, Kannan Ramchandran, and Peter L. Bartlett, "Gradient diversity: a key ingredient for scalable distributed learning," in *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, 2018, pp. 1998–2007.

[16] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[17] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*. 2019, pp. 2613–2617, ISCA.