Towards Federated Learning with on-device Training and Communication in 8-bit Floating Point

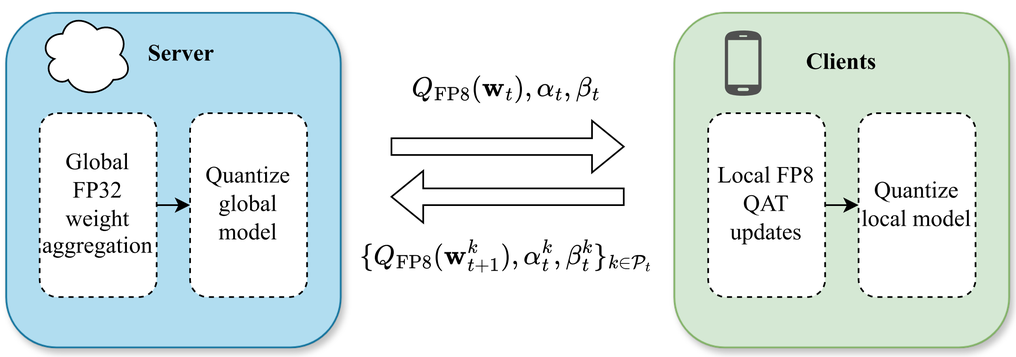
Bokun Wang* , Axel Berg†,‡, Durmus Alp Emre Acar‡, Chuteng Zhou‡
*Texas A&M University, †Lund University, ‡Arm

Full paper:

Introduction

We present a novel method for FP8 neural network training in a federated learning context. This brings not only the usual benefits of FP8 which are desirable for on-device training at the edge, but also reduces client-server communication costs due to significant weight compression. Experiments with various machine learning models and datasets show that our method consistently yields communication reductions across a variety of tasks and models compared to an FP32 baseline.

Federated Learning in FP8



FedAvg with Quantization Aware Training (QAT). We consider the federated learning problem [1], where K clients update their local models by minimizing local objectives $F_k(\mathbf{w}, Q, \alpha, \beta) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_k}[l(\mathbf{w}; \mathbf{x}, y, Q, \alpha, \beta)]$, where Q is a quantization operator and l is the loss function. Furthermore, α and β are the per-tensor clipping values used for weights and activations during QAT respectively [2]. The global objective can be expressed as

$$\min_{\mathbf{w}} F(\mathbf{w}, Q, \alpha, \beta), \quad F(\mathbf{w}, Q, \alpha, \beta) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(\mathbf{w}, Q, \alpha, \beta), \tag{1}$$

where n_k is the number of training samples on the k:th device and $n = \sum_k n_k$ is the total number of training examples.

Quantized communication (CQ). When applying FP8 QAT to a federated leaning scenario, an important aspect is the ability to reduce communication overhead by transferring weights between clients and the server using only 8 bits per scalar value. In order to form an unbiased estimate of the average client weight, we use stochastic quantization before transmitting the weights to the server.

Server-side optimization. In the un-quantized version of FedAvg, a weighted average of the individual client weights is used to form the server model, since this minimizes the mean squared error (MSE) between client weights. However, this property no longer holds when the server weights are also quantized. We therefore propose to explicitly solve the following MSE minimization problem:

$$\mathbf{w}_{t+1}, \alpha_{t+1} = \arg\min_{\mathbf{w},\alpha} \sum_{k\in\mathcal{P}_t} \frac{n_k}{m_t} \|Q_{\text{rand}}(\mathbf{w}; \alpha) - Q_{\text{rand}}(\mathbf{w}_t^k; \alpha_t^k)\|_2^2. \tag{2}$$

Since there is no closed form solution, we find an approximation by solving for $\mathbf{w}_{t+1}$ using gradient descent, then for α_{t+1} using grid search. A summary of our complete algorithm is found below.

Algorithm 1 FP8FedAvg-UQ+

Input: $\mathbf{w}_1, \alpha_1, \beta_1, Q_{\text{det}}, Q_{\text{rand}}$
  for $t = 1, ..., T$ **do**
    Sample a set $\mathcal{P}_t \in [K]$ of P active devices
    for each client $k \in \mathcal{P}_t$ **do**
      Receive $Q_{\text{rand}}(\mathbf{w}_t; \alpha_t), \alpha_t, \beta_t$ from server
      $\{\mathbf{w}_{t+1}^k, \alpha_{t+1}^k, \beta_{t+1}^k\} \leftarrow \text{LocalUpdate}(\mathbf{w}_t, Q_{\text{det}}; \alpha_t, \beta_t, \mathcal{D}_k)$
      Send $Q_{\text{rand}}(\mathbf{w}_{t+1}^k; \alpha_{t+1}^k), \alpha_{t+1}^k, \beta_i^{t+1}$ to server
    end for
    Compute $m_t = \sum_{k\in\mathcal{P}_t} n_k$, $\beta_{t+1} \leftarrow \sum_{k\in\mathcal{P}_t} \frac{n_k}{m_t} \beta_{t+1}^k$
    $\{\mathbf{w}_{t+1}, \alpha_{t+1}\} \leftarrow \text{ServerOptimize}(\{\alpha_{t+1}^k, \mathbf{w}_{t+1}^k\}_{k\in\mathcal{P}_t})$
  end for
  Evaluate on $\mathbf{w}_{T+1}, \alpha_{T+1}, \beta_{T+1}$

Convergence Analysis

Theorem. For convex and L-smooth federated losses in (1) with G-bounded unbiased stochastic gradients using an FP8 deterministic quantization method during training and an FP8 unbiased quantization method with bounded scales for model communication, the objective gap $E\left[F(Q_{\text{rand}}(\mathbf{w}_\tau)) - F(\mathbf{w}_*)\right]$ decreases at a rate of

$$O\Bigg(\underbrace{\frac{\|\mathbf{w}_1 - \mathbf{w}_*\|_2^2}{\sqrt{TU}} + \frac{G^2\sqrt{U}}{\sqrt{T}} + \frac{UG^2L}{T}}_{\mathcal{T}_1} + \underbrace{\frac{GU^{2.5}S\sqrt{d}L}{\sqrt{T}}}_{\mathcal{T}_2} + \underbrace{S\sqrt{d}G}_{\mathcal{T}_3}\Bigg)$$

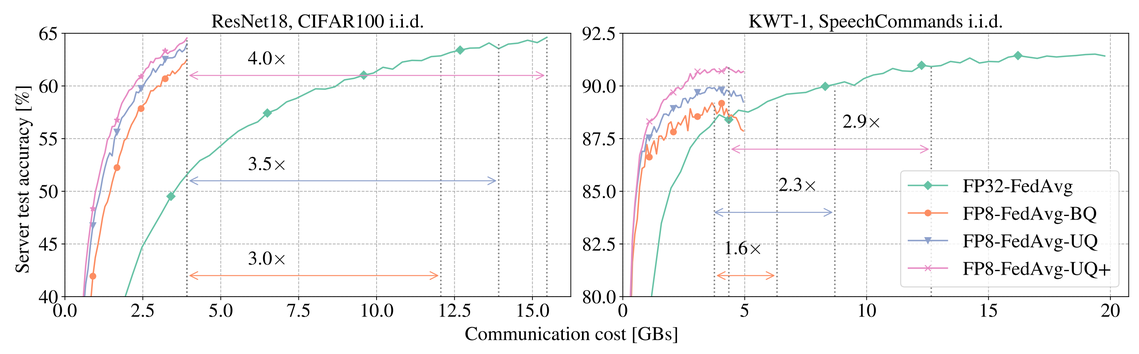
where τ is uniformly sampled from $\{1, 2, \ldots, T\}$, T is the number of rounds, U is the total number of updates done in each round, the quantization scales are uniformly bounded by S, $\mathbf{w}_1$ is the initial model, and $\mathbf{w}_*$ is an optimal solution of (1). Here, $\mathcal{T}_1$ is similar to the SGD convergence rate, whereas $\mathcal{T}_2$ is due to quantized communication and $\mathcal{T}_3$ arises from the quantization during training.

Experimental Results

Our experimental results show two important findings: 1) deterministic quantization works best during local QAT training and 2) stochastic quantization is better for communicating the models back and forth to the server. This agrees with the intuition that an unbiased model average improves convergence. The table below shows final round test accuracies on CIFAR100.

	FP8 QAT without CQ		FP8 det. QAT with CQ	
Model	det. QAT	rand. QAT	det. CQ	rand. CQ
LeNet	44.4 ± 0.5	43.7 ± 0.6	38.0 ± 0.4	44.8 ± 0.4
ResNet18	64.5 ± 0.1	63.5 ± 0.5	62.5 ± 0.9	64.0 ± 0.2

Furthermore, compared to an FP32 baseline, we are able to achieve similar performance with significant reductions in communication costs, since the communicated bytes per round is reduced by approximately $4\times$. Below, we have highlighted the communication reductions using biased communication (BC), unbiased communication (UQ) and our proposed method with server-side optimization (UQ+) for image classification on CIFAR100 using ResNet18 and keyword spotting on Google Speechcommands using a Transformer based model.



References

[1] Communication-efficient learning of deep networks from decentralized data
B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas
Artificial intelligence and statistics. PMLR, 1273–1282, 2017.

[2] Fp8 quantization: The power of the exponent.
A. Kuzmin, M. Van Baalen, Y. Ren, M. Nagel, J. Peters, and T. Blankevoort
Advances in Neural Information Processing Systems 35 (2022), 14651–14662.