Communication-Efficient Federated Learning with Multiple Local Steps

Karlo Palenzuela, PhD student, Umeå University
Department of Computing Science and Department of Mathematics
Supervisors: Tommy Löfstedt and Alp Yurtsever

Motivation & Research Goals

Federated learning (FL) is a decentralized learning paradigm that relies on efficient network communications for model aggregation when jointly training machine learning models. However, network communication in FL may become a bottleneck whenever there are many participating clients, the number of model parameters is high, or the network connections are poor. To address this problem, we propose a FL algorithm with multiple local steps (denoted FedMLS). FedMLS reduces the frequency of parameter aggregation, and hence the number of client-server communications, by increasing the number of local updates between aggregations. For a solution with accuracy ε, the proposed FedMLS algorithm achieves a communication complexity of $\mathcal{O}(\varepsilon^{-1})$ compared to conventional approaches with communication complexity of $\mathcal{O}(\varepsilon^{-2})$. Numerical experiments confirm that FedMLS provides a substantial reduction in the communication time compared to existing algorithms, and does this without relying on strong convexity nor on gradient similarity assumptions.

Methods

The standard FL problem is usually stated with the problem template

$$\begin{aligned} \min_{x_1,\dots,x_N\in\mathbb{R}^d} \quad & F(x) := \frac{1}{N}\sum_{i=1}^{N} f_i(x_i) \\ \text{subject to} \quad & \mathcal{X} = \{x | x_1 = x_2 = \cdots = x_N\}. \end{aligned} \tag{1}$$

where $F : \mathbb{R}^d \to \mathbb{R}$ denotes joint loss function, $f_i : \mathbb{R}^d \to \mathbb{R}$ denote the ith client's local loss function.

Thekumparampil et al. [1] introduced the MOreau Envelope Projection Efficient Subgradient method (MOPES), a regularized Equation (1), namely

$$\min_{x\in\mathcal{X},x'\in\mathcal{X}'} \Psi(x,x') := F(x') + \frac{1}{2\lambda}\|x - x'\|^2, \tag{2}$$

where $\mathcal{X}'$ is a Euclidean norm ball.

Proximal operator of Equation (2) with respect to the indicator function, $I_{\mathcal{X}}(x)$, corresponds to the projection, $\text{proj}_{\mathcal{X}}(x)$ while the one for $F(x')$ requires solving the subproblem:

$$\text{prox}_{F/\beta}(x') = \arg\min_{u\in\mathcal{X}'} \; F(u) + \frac{\beta}{2}\|u - x'\|$$

This leads to the following update steps, for $k = 1,\dots,K$:

$$\left|\begin{aligned} &\beta_k = \tfrac{4}{\lambda k},\ \gamma_k = \tfrac{2}{k+1} \\ &[y_k, y'_k] = (1-\gamma_k)[x_{k-1}, x'_{k-1}] + \gamma_k[z_{k-1}, z'_{k-1}] \\ &z_k = \text{proj}_{\mathcal{X}}(z_{k-1} - \tfrac{1}{\beta_k\lambda}(y_k - y'_k)) \\ &(z'_k, \tilde{z}'_k) = \text{approx-prox}_{F/\beta_k}(z'_{k-1} - \tfrac{1}{\beta_k\lambda}(y'_k - y_k)) \\ &[x_k, x'_k] = (1-\gamma_k)[x_{k-1}, x'_{k-1}] + \gamma_k[z_k, \tilde{z}'_k] \end{aligned}\right.$$

By separability of F, the approx-prox proceeds as follows for $t = 1,\dots,T_k$ [2]:

$$\left|\begin{aligned} &\hat{u}_{k,t} = u_{k,t-1} - \tfrac{1}{(1+t/2)}\big(\tfrac{1}{\beta_k}\tilde{\nabla} f(u_{k,t-1}) + u_{k,t-1} - w_k\big) \\ &u_{k,t} = \text{proj}_{\mathcal{X}'}(\hat{u}_{k,t}) \\ &\tilde{u}_{k,t} = (1-\theta_t)\tilde{u}_{k,t-1} + \theta_t u_{k,t} \quad \text{where} \quad \theta_t = \tfrac{2(t+1)}{t(t+3)} \end{aligned}\right.$$

Selected Results

We have developed the FL version of MOPES, denoted Federated Learning with Multiple Local Steps (FedMLS) to solve Equation (1).

Algorithm 1 Federated Learning with Multiple Local Steps (FedMLS)

procedure FedMLS($G, L, x_{\cdot,0}, K, D, \sigma^2, \lambda$)
  $z_{i,0} = x'_{i,0} = x_{i,0} = z'_{i,0} = x_{\cdot,0}$
  for $k = 1,\dots,K$ **do**
    $\beta_k = \frac{4}{\lambda k}$, $\gamma_k = \frac{2}{k+1}$, and $T_k = \left\lceil \frac{(4G^2+\sigma^2)\lambda^2 K k^2}{2\tilde{D}} \right\rceil$
    $y'_{i,k} = (1-\gamma_k)x'_{i,k-1} + \gamma_k z'_{i,k-1}$
    Client sends $y'_{i,k}$
    — Server Side Starts —
    $y_k = (1-\gamma_k)x_{k-1} + \gamma_k z_{k-1}$
    $z_k = z_{k-1} - \frac{k}{4}\big(y_k - \frac{1}{N}\sum_{i=1}^{N} y'_{i,k}\big)$
    $x_k = (1-\gamma_k)x_{k-1} + \gamma_k z_k$
    Server sends y_k to each client
    — Server Side Ends —
    $(z'_{i,k}, \tilde{z}'_{i,k}) = \text{LocalTraining}\big(\frac{1}{\lambda}(y'_{i,k} - y_k), z'_{i,k-1}, \beta_k, T_k\big)$
    $x'_{i,k} = (1-\gamma_k)x'_{i,k-1} + \gamma_k \tilde{z}'_{i,k}$
  end for
  return x_K
end procedure
procedure LocalTraining(g, u_0, β, T)
  $\tilde{u}_0 = u_0$
  for $t = 1,\dots,T$ **do**
    $\theta_t = \frac{2(t+1)}{t(t+3)}$
    $\hat{u}_t = u_{t-1} - \frac{1}{(1+t/2)\beta}\big(\tilde{\nabla} f_i(u_{t-1}) + \beta(u_{t-1} - u_0 + g/\beta)\big)$
    $u_t = \hat{u}_t \cdot \min(1, \frac{R}{\|u_t\|})$
    $\tilde{u}_t = (1-\theta_t)\tilde{u}_{t-1} + \theta_t u_t$
  end for
  return $(u_T, \tilde{u}_T)$
end procedure

Theorem 1. *Let $\mathcal{X}'$ be the Euclidean norm ball in $\mathbb{R}^d$ with radius R centered at the origin and $\mathcal{X}$ be the consensus constraint where $\mathcal{X} \subseteq \mathcal{X}'$. Let the joint loss function $F : \mathcal{X}' \to \mathbb{R}$ be a proper, lower semicontinuous, and convex function with bounded (sub)gradients, i.e., $\|\nabla F(x)\| \le G$ for all $x \in \mathcal{X}'$. Assume R is sufficiently large such that there exists a solution $x^* \in \arg\min_{x\in\mathcal{X}} F(x)$ such that $x^* \in \mathcal{X}'$. Then, after K client-server communication rounds, the FedMLS algorithm outputs $x_K \in \mathcal{X}$ satisfying*

$$\mathbb{E}\big[F(x_K)\big] - F(x^*) \le \frac{10\|x_0 - x^*\|^2 + 8\tilde{D}}{\lambda K(K+1)} + G^2\frac{\lambda}{2}$$

for any choice of $\lambda > 0$ and $\tilde{D} > 0$.

Theorem 2. *Let $\lambda = \varepsilon/G^2$ and $\tilde{D} = \|x_0 - x^*\|^2$. Choosing $K = \left\lceil \frac{6G\sqrt{2\tilde{D}}}{\varepsilon} \right\rceil$ which is of order $\mathcal{O}\left(\frac{G\sqrt{\tilde{D}}}{\varepsilon}\right)$ communication rounds, then FedMLS achieves an ε-suboptimal solution satisfying*

$$\mathbb{E}[F(x_K)] - F(x^*) \le \varepsilon.$$

Moreover, if we choose $T_k = \left\lceil \frac{(4G^2+\sigma^2)\varepsilon^2 K k^2}{2G^4\tilde{D}} \right\rceil$ then the total number of local training rounds, $\sum_{k=1}^{K} T_k$ is $\mathcal{O}\left(\frac{(4G^2+\sigma^2)\tilde{D}}{\varepsilon^2}\right)$.

References

[1] Projection Efficient Subgradient Method and Optimal Nonsmooth Frank-Wolfe Method
K. K. Thekumparampil, P. Jain, P. Netrapalli and S. Oh
2020 Neural Information Processing Systems

[2] Gradient Sliding for Composite Optimization
Guanghui Lan
Mathematical Programming 159, 201-235(2016)

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM