Variance Reduced Federated Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

To be completed

1 SAGA-like Algorithms

1.1 Local SGD

Algorithm 1 SAGA Local SGD

Input: Local learning rate γ and global learning rate η and communication period p.
Init: $g^{(k)} = \frac{1}{n}\sum_{i=1}^{n} g^{(0)}$.
for $k = 0, 1, ..., K$ **do**
  Draw two independent and distinct indices i_k and j_k
  for $\tau = k, ..., k+p-1$ **do**
    Compute the following quantity

$$v_{i_k}^{(\tau)} = v_{i_k}^{(\tau-1)} - \gamma(\nabla f_{i_k}\left(x^{(k)}\right) - \nabla f_{i_k}\left(\alpha_{i_{(k)}}^{t}\right) + g^{(k)})$$

  end for
  $v_{i_k}^{(k)} \leftarrow v_{i_k}^{(k+p-1)}$
  Update the global model

$$x^{(k+1)} = x^{(k)} - \eta v_{i_k}^{(k)}$$

  Update $\alpha_{j_k}^{(k+1)} = x^{(k)}$ and $\alpha_{j}^{(k+1)} = \alpha_{j}^{(k)}$ for $j \neq j_k$
  Update $g^{(k+1)} = g^{(k)} - \frac{1}{n}\left(\nabla f_{j_k}\left(\alpha_{j_k}^{(k)}\right) - \nabla f_{j_k}\left(\alpha_{j_k}^{(k+1)}\right)\right)$
end for

1.2 FedSVRG with Sketching

Algorithm 2 `FedSVRG`: SVRG Federated Learning algorithm with Sketching.

Inputs: $\boldsymbol{x}^{(0)}$ initial common model, communication rounds R, the number of local updates K, and global and local learning rates γ and η
for $r = 0, \dots, R-1$ **do**
  parallel for device $j = 1, \dots, n$ **do**:
    Computes $\mathbf{\Phi}^{(r)} \triangleq \mathtt{Q}\left[\mathbf{S}^{(r-1)}\right]$ ($\mathtt{Q}$ is any query function)
    Set $\boldsymbol{x}^{(r)} = \boldsymbol{x}^{(r-1)} - \gamma \mathbf{\Phi}^{(r)}$
    Set $\boldsymbol{x}_j^{(0,r)} = \boldsymbol{x}^{(r)}$
    Compute full gradient $\nabla f_j(\boldsymbol{x}^{(\kappa(r))}) = \frac{1}{n_j} \sum_{i=1}^{n_j} \nabla f_j(\boldsymbol{x}^{(\kappa(r))}, \xi_i)$
    for $\ell = 0, \dots, K-1$ **do**
      Sample an index i_ℓ uniformly on $[n_j]$
      $\boldsymbol{x}_j^{(\ell+1,r)} = \boldsymbol{x}_j^{(\ell,r)} - \eta \left(\nabla f_j(\boldsymbol{x}_j^{(\ell,r)}, \xi_{j,i_\ell}) - \nabla f_j(\boldsymbol{x}^{(\kappa(r))}, \xi_{j,i_\ell}) + \nabla f_j(\boldsymbol{x}^{(\kappa(r))})\right)$
    end for
    Device j sends $\mathbf{S}_j^{(r)} \triangleq \mathbf{S}_j \left(\boldsymbol{x}_j^{(0,r)} - \boldsymbol{x}_j^{(K,r)}\right)$ back to the server.
  Server **computes**
    $\mathbf{S}^{(r)} = \frac{1}{p} \sum_{j=1} \mathbf{S}_j^{(r)}$ and **broadcasts** $\mathbf{S}^{(r)}$ to all devices.
  end parallel for
end
Output: $\boldsymbol{x}^{(R-1)}$

Particularity: $\kappa(r)$ is the epoch number at which we compute both control variate terms. We can have $\kappa(r) = r$ or something else (to tune).

1.3 FedSAGA with Sketching

Algorithm 3 `FedSAGA`: SAGA Federated Learning algorithm with Sketching.

Inputs: $\boldsymbol{x}^{(0)}$ initial common model, communication rounds R, the number of local updates K, and global and local learning rates γ and η
for $r = 0, \dots, R-1$ **do**
  parallel for device $j = 1, \dots, n$ **do**:
    Computes $\mathbf{\Phi}^{(r)} \triangleq \mathtt{Q}\left[\mathbf{S}^{(r-1)}\right]$ ($\mathtt{Q}$ is any query function)
    Set $\boldsymbol{x}^{(r)} = \boldsymbol{x}^{(r-1)} - \gamma \mathbf{\Phi}^{(r)}$
    Set $\boldsymbol{x}_j^{(0,r)} = \boldsymbol{x}^{(r)}$
    Compute full gradient $\nabla f_j(\boldsymbol{x}^{(r)}) = \frac{1}{n_j} \sum_{i=1}^{n_j} \nabla f_j(\boldsymbol{x}^{(r)}, \xi_i)$
    for $\ell = 0, \dots, K-1$ **do**
      Sample an indices (i_ℓ, q_ℓ) independently and uniformly on $[n_j]$
      $\boldsymbol{x}_j^{(\ell+1,r)} = \boldsymbol{x}_j^{(\ell,r)} - \eta \left(\nabla f_j(\boldsymbol{x}_j^{(\ell,r)}, \xi_{j,i_\ell}) - \nabla f_j(\boldsymbol{x}_j^{(t_{i_\ell},r)}, \xi_{j,i_\ell}) + \overline{F}_j^{(r)}\right)$
      where $\overline{F}_j^{(r)} = \frac{1}{n_j} \sum_{i=1}^{n_j} \nabla f_j(\boldsymbol{x}^{(r)}, \xi_i) + \frac{1}{n_j}(\nabla f_j(\boldsymbol{x}_j^{(\ell,r)}, \xi_{j,q_\ell}) - \nabla f_j(\boldsymbol{x}_j^{(t_{q_\ell},r)}, \xi_{j,q_\ell}))$
    end for
    Device j sends $\mathbf{S}_j^{(r)} \triangleq \mathbf{S}_j \left(\boldsymbol{x}_j^{(0,r)} - \boldsymbol{x}_j^{(K,r)}\right)$ back to the server.
  Server **computes**
    $\mathbf{S}^{(r)} = \frac{1}{p} \sum_{j=1} \mathbf{S}_j^{(r)}$ and **broadcasts** $\mathbf{S}^{(r)}$ to all devices.
  end parallel for
end
Output: $\boldsymbol{x}^{(R-1)}$

Particularity: We need to store each local models on each device j in order to compute the control variate terms. No epoch tuning though. Indeed $\boldsymbol{x}_j^{(t_{i_\ell},r)}$ is the value of local model on device j when index i_ℓ was drawn last.

2 Numerical Examples