FedBCD: A Communication-Efficient Collaborative Learning Framework for Distributed Features

当使用小批量梯度下降的时候，训练一个模型可能需要几千甚至几万轮迭代，这个时候就会产生大量的通信开销，Fedavg可以让每个客户端在本地训练多轮来减少通信开销。在纵向联邦上执行这种局部更新策略是否可行尚不清楚。

主要贡献：本文提出了一种用于分布式特征的协作学习框架，称为联邦随机块坐标下降（FedBCD），其中各方在每次通信期间仅共享模型参数和每个样本的原始数据的内积，并且可以连续执行局部模型更新（以并行或顺序方式），而无需每次迭代通信。

块坐标下降

坐标下降法

坐标下降法是一种非梯度的优化方法，和梯度下降方法不同，**它每次沿着单个维度方向进行搜索**，当得到一个当前维度最小值之后再循环使用不同的维度方向，最终收敛得到最优解。

块坐标下降

块坐标下降是坐标下降的更一般化，它通过对变量的**子集**进行同时优化，把原问题分解为多个子问题。在下降的过程中更新的次序**可以是确定或随机的**，如果是确定的次序，我们可以设计某种方式，或是周期或是贪心的方法选择更新子集。

分布式块坐标下降

论文：”A distributed block coordinate descent method for training l1regularized linear classifiers“

适用于**L1正则化**分类器并行训练的分布式块坐标下降（DBCD）方法，降低通信量

参数集合：$\mathbf{w} = \{w_j\}_{j=1}^{m}$。数据集有$n$个样本，每个样本是$m$维的，$X_{n\times m}$表示样本矩阵。

线性分类器的输出：$y_i = w^T x_i$，标签$c_i \in \{1, -1\}$，$l()$是损失函数

我们假设$l()$满足non-negative and convex（非负凸）、continuously differentiable（连续可微）、Lipschitz continuous。最小二乘损失、逻辑损失、SVM平方铰链损失和Huber损失等损失函数满足这些假设。

我们假设使用均方误差损失函数

$$l(y_i; c_i) = max\{0, 1 - c_i y_i\}^2$$

$$f(w) = \frac{1}{n} \sum_i l(y_i; c_i)$$

$$u(w) = \lambda \sum_j |w_j|$$

$$\min_{w \in R^m} F(w) = f(w) + u(w).$$

我们将这m个特征分给P个人，用$\{B_p\}_{p=1}^{P}$来表示这m个特征：$M = \{1, \cdots, m\}$

经典的训练流程

Algorithm 1: A generic distributed algorithm

Choose w^0 and compute $y^0 = Xw^0$;
for $t = 0, 1 \ldots$ **do**
  for $p = 1, \ldots, P$ **do**
    (a) Select a working subset of variables[3], $S_p^t \subset B_p$;
    (b) Form $f_p^t(w_{B_p})$, an approximation of f and minimize, exactly or approximately, $f_p^t + u$ over only the weights corresponding to S_p^t:

$$\min f_p^t(w_{B_p}) + u(w_{B_p}) \quad \text{s.t.} \quad w_j = w_j^t \; \forall \, j \in B_p \setminus S_p^t \tag{3}$$

    to get $\bar{w}_{B_p}^t$ and set direction: $d_{B_p}^t = \bar{w}_{B_p}^t - w_{B_p}^t$;
    (c) Choose α^t and update: $w_{B_p}^{t+1} = w_{B_p}^t + \alpha^t d_{B_p}^t$;
  end
  (d) Update $y^{t+1} = y^t + \alpha^t \sum_p X_{B_p} d_{B_p}^t$;
  (e) Terminate if optimality conditions hold;
end

参数定义

K个数据方，N个数据样本$D=\{\xi_i\}_{i=1}^n$，样本由特征和标签组成$\xi=(X,y)$，特征是d维的，分布在K个数据方中
$\mathcal{D}_K \triangleq \{\mathbf{x}_{i,K}, y_{i,K}\}_{i=1}^N$,假设数据标签在第$K$个用户手中.

协作训练问题描述如下：

$$\min_{\Theta} \mathcal{L}(\Theta;\mathcal{D}) \triangleq \frac{1}{N}\sum_{i=1}^{N} f\left(\theta_1,\ldots,\theta_K;\xi_i\right)+\lambda\sum_{k=1}^{K}\gamma\left(\theta_k\right) \quad (1)$$

其中$\theta_k \in \mathbb{R}^{d_k}$标识第$k$方的训练参数，$\gamma()$正则化。

FedBCD算法

普通纵向联邦SGD算法

如果对S个数据点的小批量$S \subset D$进行采样，则第k个用户的随机部分梯度由下式给出：

$$g_k(\Theta;\mathcal{S}) \triangleq \nabla_k f(\Theta;\mathcal{S})+\lambda\nabla\gamma\left(\theta_k\right) \quad (2)$$

对于线性回归、逻辑回归或SVM等模型，预测值H_i由各数据方的局部线性组合之和构成：

$$H_i \triangleq \sum_{k=1}^{K} H_i^k \triangleq \sum_{k=1}^{K} \mathbf{x}_{i,k}\theta_k \tag{3}$$

对于线性\逻辑回归和SVM来说,

$$\nabla_k f(\Theta;\mathcal{S}) = \frac{1}{S}\sum_{\xi_i\in\mathcal{S}} \frac{\partial f\left(H_i, y_{i,K}\right)}{\partial H_i}\left(\mathbf{x}_{i,k}\right)^T \tag{4}$$

梯度推导过程:

- 链式法则: 对第k方的参数计算梯度:$\frac{\partial f}{\partial \theta_k} = \frac{\partial f}{\partial H_i} \cdot \frac{\partial H_i}{\partial \theta_k}$
- 梯度分量计算: $\frac{\partial f}{\partial H_i}$和损失函数形式有关，对于线性回归、逻辑回归或SVM等模型$\frac{\partial H_i}{\partial \theta_k} = x_{i,k}$
- 小批量平均：最后求平均

要计算梯度，需要其他$K-1$个人给第K个人发送自己的本地预测$I_S^{k,K} = \{H_{i,k}\}_{i\in S}$，由第$k$个人算出公式（4）中的第一项$I_S^{K,q} = \{\frac{\partial f(H_i,y_{i,K})}{\partial H_i}\}_{i\in S}$，然后发给每个人计算自己的梯度

对于一个任意的损失函数，用户k把计算$\nabla_k f(\Theta;\mathcal{S})$需要的整体信息记为：

$$I_{\mathcal{S}}^{-k} \triangleq \left\{I_{\mathcal{S}}^{q,k}\right\}_{q\neq k} \tag{4}$$

第k个用户的梯度可以写为：

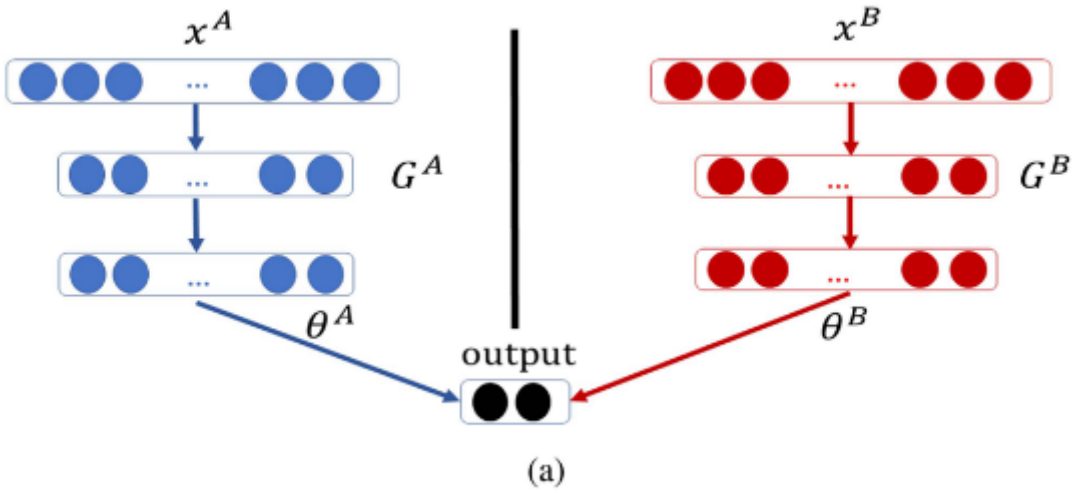
$$\begin{aligned} g_k(\Theta;\mathcal{S}) &= \nabla_k f\left(I_{\mathcal{S}}^{-k}, \theta_k; \mathcal{S}\right) + \lambda\nabla\gamma\left(\theta_k\right) \\ &\triangleq g_k\left(I_{\mathcal{S}}^{-k}, \theta_k; \mathcal{S}\right) \end{aligned} \tag{5}$$

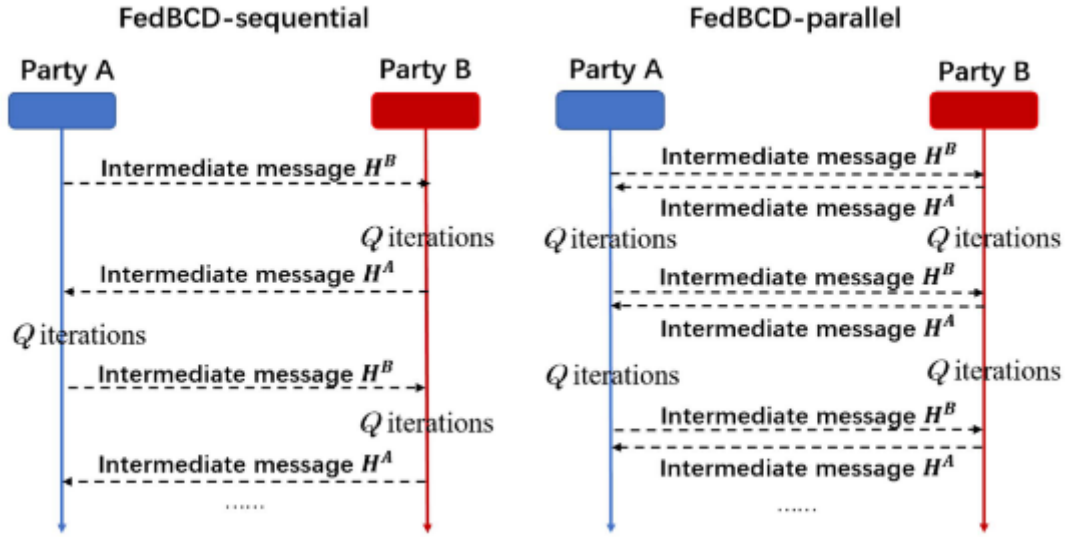
整体的梯度也可以表示为：

$$g(\Theta;\mathcal{S}) \triangleq [g_1(I_{\mathcal{S}}^{-1},\theta_1;\mathcal{S});\cdots;g_K(I_{\mathcal{S}}^{-K},\theta_K;\mathcal{S})].$$

然后就可以用梯度下降训练了。

并行FedBCD和顺序FedBCD





Algorithm 1: FedBCD-p: Federated Stochastic Block Coordinate Descent

Input: learning rate η, communication frequency Q
Output: Model parameters θ_1, θ_2...θ_K
Party 1,2..K initialize θ_1, θ_2, ...θ_K.
for *each iteration* $r = 1, 2, \ldots$ **do**
  if $r \bmod \mathrm{Q} = 0$ **then**
    Randomly sample a mini-batch $\mathcal{S} \subset \mathcal{D}$;
    Exchange($\{1, 2...K\}, \mathcal{S}$);
  end
  for *party* $k \in [K]$, *in parallel* **do**
    $\boldsymbol{k}$ computes $g_k(I_{\mathcal{S}}^{-k}, \theta_k^r; \mathcal{S})$ using (6) and updates $\theta_k^{r+1} \leftarrow \theta_k^r - \eta g_k(I_{\mathcal{S}}^{-k}, \theta_k^r; \mathcal{S})$;
  end
end
Exchange(U,$\mathcal{S}$): *# U is the set of party IDs*
  if *equation* (2) *holds* **then**
    each party $k \in U$ and $k \neq K$ in parallel computes and sends $I_{\mathcal{S}}^{k,K}$ to party K;
    party K computes and sends $I_{\mathcal{S}}^{K,k}$ to party $k \in U$;
  else
    each party $k \in U$ in parallel computes and sends $I_{\mathcal{S}}^{k,q}$ to party $q \in U$;
  end

在本地进行更新迭代时，由于本地无法单独进行梯度下降，因此只能使用相同的陈旧小批次进行梯度下降。

流程图中的Exchange判断的条件是使用的模型是否为线性\逻辑回归、支持向量机。

例子

假设2个客户端A,B，一个样本x，它分为2部分特征，A持有x_1，B持有x_2和标签y。假设双方使用逻辑回归，A的模型参数为w_1，B的模型参数为w_2.

$$A: H_1 = w_1 x_1 \tag{1}$$

$$B: H_2 = w_2 x_2 \tag{2}$$

$$\hat{y} = w_1 x_1 + w_2 x_2 \tag{3}$$

$$Loss = (y - \hat{y})^2 \tag{4}$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_1} = -2(y - \hat{y}) x_1 \tag{5}$$

对于这个例子，服务器给每个客户端发送的内容$I_S^{K,q}$就是$-2(y-\hat{y})$。客户端A收到后，计算自己的梯度进行更新。同时，它计算$I^{轮次} = -2(y-\hat{y}) - w_1 x_1$并存储在本地。在使用陈旧小批次进行本地更新时，客户端A计算新的$H_1^{'} = w_1^{'} x_1$，然后计算$I^{轮次} + H^{'}$进行新一轮的本地更新。

本地迭代轮次Q的影响

实验结果表明FedBCD-p在本地迭代轮次**Q=15**时，效果最好。文献【1】建议在局部迭代较大时，向局部目标函数添加近端项，以减轻潜在的发散。

$$g_k\left(y_k^r; \xi_i\right) = g_k\left(\left[\Theta_{-k}^{r_0}, \theta_k^r\right]; \xi_i\right]\right) + \mu\left(\theta_k^r - \theta_k^{r_0}\right)$$

r_0表示r轮的前一轮，r轮完成了同步并交换中间信息。y_k^r表示节点k在r轮用于计算局部梯度的局部向量。$\mu\left(\theta_k^r - \theta_k^{r_0}\right)$就是近端项。

[1] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” in Proc. Mach. Learn Syst., vol. 2, pp. 429–450, 2020.

客户端数量K的影响

影响不大