Federated Learning

Min, Xia

May 12, 2024

1 General

Many models that power intelligent behavior on mobile devices fit the federated learning setting. Consider image classification, for example predicting which photos are most likely to be viewed multiple times in the future, or shared. All the photos a user takes can be privacy sensitive, and the distributions from which these examples are drawn are also likely to differ substantially from easily available proxy datasets. And finally, the labels for these problems are directly available: photo labels can be defined by natural user interaction with their photo app.

Most important property of federated setting:

- Non-IID: The data on each client has different population distributions.
- Unbalanced: Each agent has very different size of training data.

2 Federated Learning Article1

Communication-Efficient Learning of Deep Networks from Decentralized Data[2]

The core idea lies here is quite simple. The global target is minimize $f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$. For K agents with sample index set $\mathcal{P}_k$ and $|\mathcal{P}_k| = n_k$, then $f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w)$,where $F_k(w) = \frac{1}{n_k}\sum_{i\in\mathcal{P}_k} f_i(w)$.

The FedAvg algorithm is as follows:

- At step t has a global parameter w_t and is transmitted to a partion C of all clients. Each of these agent calculates an updated parameter w_k^{t+1} by, say, SGD or multiple steps of SGD, and transmitted to central server.
- On client split sample into B batches and update parameter by batchSGD for E epoches and get w_k^{t+1}.
- Central server update parameter by $w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^{t+1}$.

3 Federated Learning Article2

Federated Optimization in Heterogeneous Networks[1]

The core idea is that, target is minimize $f(w) = \sum_{k=1}^{K} p_k F_k(w) = \sum_{k=1}^{K} E_k [F_k(w)]$, where $\sum_{k=1}^{K} p_k = 1$ and $F_k(w) = E_{x_k, y_k \sim D_k} [f_k(w; x_k, y_k)]$. Sample be covariate x_k with label y_k and some loss $f_k(w; x_k, y_k)$. Highlight that D_k can be very different for different k.

Under this setting, FedAvg may diverge because choose a large E, for a same w_t, w_k^{t+1} can be very different for different k. Thus when client updates its parameter, we require it can't go too far from last step(otherwise the sum may diverge).

Definition 3.1 (γ_k^t-inexact solution)**.** *Define function $h_k(w; w_t) = F_k(w) + \mu |w - w_t|^2/2$, where F_k is defined as previous in article1 and a given constant $\gamma \in [0, 1]$, say w is γ_k^t-inexact solution of $\min h_k(w; w_t)$ if all k, t, $|\nabla h_k(w; w_t)| \leq \gamma_k^t |\nabla h_k(w; w_t)$.*

This article gives FedProx as follows

- Select a subset S_t of all K agents randomly and send w_t to each agent.
- Each agent calculates a γ_k^t-inexact minimizer w_k^{t+1} of $h_k(w; w_t) = F_k(w) + \mu |w - w_t|^2/2$, and send w_k^{t+1} back to central server.
- Update $w_{t+1} = \frac{1}{|S_t|} \sum_{k \in S_t} w_k^{t+1}$.

References

[1] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[2] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.