Vertical Federated Learning with Gossip Protocol

DVPL-Katyusha with FastMix

Pyotr Lisov, Ivan Toropin, Aleksandr Besnosikov

21.05.2024

MIPT, Russia

Table of contents

Introduction

The problem of time in such fields as ML and DL is more relevant than ever. Today it is common that teaching large neural networks requires several days or even weeks. Federated Learning is a new solution for this issue. It uses several machines, (usually called servers, agents, nodes, edges, etc.), that can compute the algorithm separately and then exchange data with each other.

Introduction

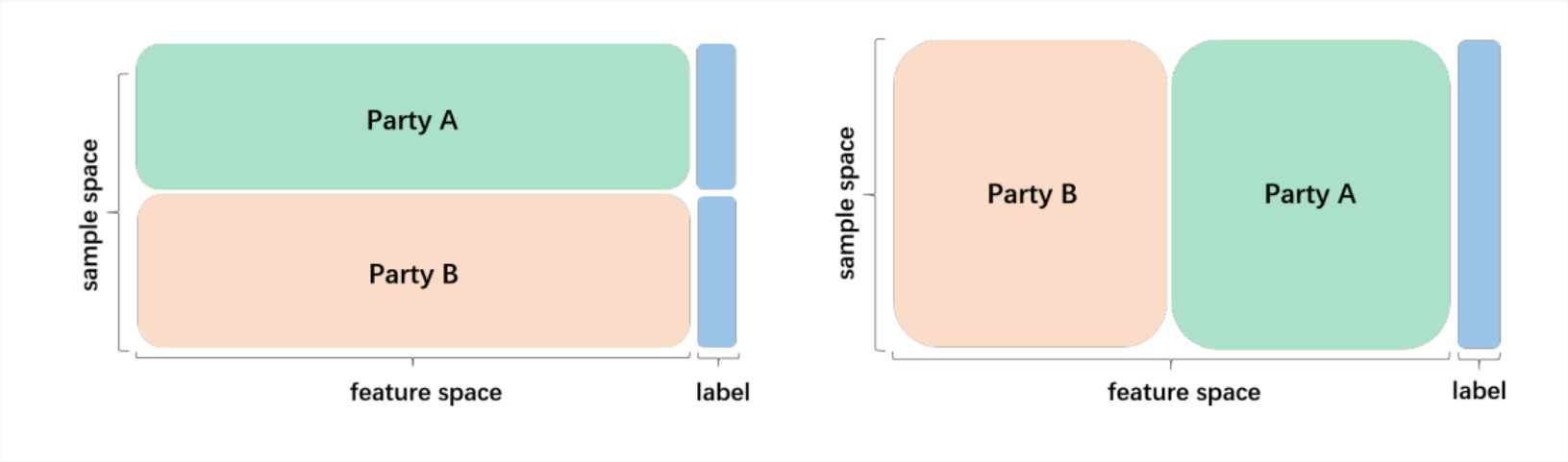

Figure 1: HFL vs VFL

Federated Learning can be further divided into two categories: horizontal, i.e each node share the same feature space while holding different samples, and vertical, i.e each node share the same samples/users while holding different features

Motivation

The main reason to use VFL is that often data owned by different parties have the same sample IDs but disjoint subsets of features. Such scenario is common in the industry applications of collaborative learning, such as medical study, financial risk, and targeted marketing. For example, E-commerce companies owning the online shopping information could collaboratively train joint-models with banks and digital finance companies that own other information of the same people such as the average monthly deposit and online consumption, respectively, to achieve a precise customer profiling.

Motivation

It is also quite common that different parties can't communicate with each other directly (this case is called fully connected networks). In stead, they have a small range of their 'neighbors'. Thus, another important problem is the organization of communication between different nodes in sparse networks. One of the ways to model such communications is the gossip protocol or epidemic protocol, which we decided to use. It is a procedure or process of computer peer-to-peer communication that is based on the way epidemics spread.

Related Works

We decided to use the DVPL-Katyusha (Sergey Stanko & Timur Karimullin, 2024) algorithm as a basis. It is a VFL implementation of L-Katyusha (Kovalev et al., 2020), developed for fully connected networks. In order to modify it for sparse ones we used a variant of gossip protocol called FastMix (Haishan Ye et al., 2023) in stead of AllReduce (Ernie Chan et al., 2007)

The problem statement

Like in the original paper (Sergey Stanko & Timur Karimullin, 2024) we consider the problem of minimization of the linear loss function., distributed among n devices. This can be mathematically written as:

$$\min_{x \in \mathbb{R}^d} \left[f(x) := \frac{1}{s} \left\| \sum_{i=1}^{n} A_i x_i - b \right\|^2 \right]$$

We assume that $f(x)$ is μ-strongly convex and L-smooth.

Our goal is to prove the convergence of the new algorithm, show that it is at least as quick as the original one.

Results

First, we prove that the variance of the gradient estimator g^k is bounded and eventually goes to zero as our algorithm progresses:

$$\mathbb{E}\left[\left\|g^k(x^k) - \nabla f(x^k)\right\|_2^2\right] \leqslant \frac{2}{b}L\big(f(w^k) - f(x^k) - \langle \nabla f(x^k), w^k - x^k\rangle\big) \cdot \left(1 + \frac{s}{\mu^2}\sum_{j=1}^{s} L_j\right),$$

where b is the size of batches, s is the number of samples in the dataset (aka the number of rows of matrix A), L_j is the constant of smoothness of j-th row of matrix A.

Results

In our algorithm we call $\tilde{L}$ the efficient Lipschitz constant, if it is the smallest constant, that is greater than L, such:

$$\mathbb{E}\left[\left\|g^k(x^k) - \nabla f(x^k)\right\|_2^2\right] \leqslant 2\tilde{L}\left(f(w^k) - f(x^k) - \langle \nabla f(x^k), w^k - x^k\rangle\right)$$

So, our next step is the proof that efficient Lipschitz constant $\tilde{L}$ is proportional to $\frac{L^3}{\mu^2}$:

$$\tilde{L} = \frac{L}{b} \cdot \left(1 + \frac{s}{\mu^2}\sum_{j=1}^{s} L_j\right) \sim \frac{L^3}{\mu^2}$$

Results

Let x^* be the solution of the problem, then after k iterations of the algorithm we get:

$$\mathbb{E}\left[Z^{k+1} + Y^{k+1} + W^{k+1}\right] \leqslant \frac{1}{1+\eta\sigma} Y^k + (1 - \theta_1(1-\theta_2))Z^k + \left(1 - \frac{p\theta_1}{1+\theta_1}\right) W^k,$$

where:

$$Z^k = \frac{\tilde{L}(1+\eta\sigma)}{2\eta} \left\| z^k - x^* \right\|_2^2,$$

$$Y^k = \frac{1}{\theta_1}\left(f\left(y^k\right) - f\left(x^k\right)\right), \quad W^K = \frac{\theta_2(1+\theta_1)}{p\theta_1}\left(f\left(w^k\right) - f(x^*)\right)$$

Results

Finally, choosing right parameters we can get an estimation of the number of iterations.

Let $p = \frac{b}{s}$, $\theta_1 = \min\left\{\sqrt{\frac{2\sigma s}{3b}}, \frac{1}{2}\right\}$, $\theta_2 = \frac{1}{2}$, then it will require

$$K = \mathcal{O}\left(\frac{s}{b}\sqrt{\frac{L}{\mu}\left(\frac{1}{s} + \frac{\sum_{j=1}^{s} L_j^2}{\mu^2}\right)}\log\frac{1}{\varepsilon}\right) \sim \mathcal{O}\left(\sqrt{\frac{s}{b}}\sqrt{\frac{L^3}{\mu^3}}\log\frac{1}{\varepsilon}\right)$$

iterations to achieve $\mathbb{E}\left[\Psi^{k+1}\right] < \varepsilon\Psi^0$, where $\Psi^k = Z^k + Y^k + W^k$ is the Lyapunov function. This concludes our proof.

Our main result:

$$K = \mathcal{O}\left(\sqrt{\frac{s}{b}}\sqrt{\frac{L^3}{\mu^3}}\log\frac{1}{\varepsilon}\right)$$

Compared to the original paper:

$$K = \mathcal{O}\left(\frac{\sqrt{s}}{b}\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$$

Literature

[1] Sergey Stanko & Timur Karimullin, Accelerated Methods with Compression for Horizontal and Vertical Federated Learning, 2024

[2] Kovalev, S. Horvath, and P. Richtarik. Don't jump through hoops and remove those loops: Svrg and katyusha are better without the outer loop, 2020

[3] Haishan Ye, Luo Luo, Ziang Zhou, and Tong Zhang. Multi-consensus decentralized accelerated gradient descent. arXiv preprint arXiv:2005.00797, 2020

[4] E. Chan, M. Heimlich, A. Purkayastha, R. V. D. Geijn. Collective communication: theory, practice, and experience, 2007