Stochastic Gradient Push for Distributed Deep Learning + HHHFL: Hierarchical Heterogeneous Horizontal Federated Learning for Electroencephalography

Tao Shen

Zhejiang University

November 16, 2019

Stochastic Gradient Push for Distributed Deep Learning

Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, Mike Rabbat

Distributed Optimization

Problem formulation

$$\min_{\boldsymbol{x}_i \in \mathbb{R}^d, i=1,\ldots,n} \quad \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\xi_i \sim D_i} F_i\left(\boldsymbol{x}_i; \xi_i\right)$$

$$\text{subject to} \quad \boldsymbol{x}_i = \boldsymbol{x}_j, \forall i, j = 1, \ldots, n$$

Two Principle

1. Fit Local Model and Data (Training)
2. Fit Local Model and Other Model (Consensus)

Consensus

Approximate distributed averaging

$$\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{y}_i^{(0)} \qquad \boldsymbol{y}_i^{(0)} \in \mathbb{R}^d \qquad \boldsymbol{Y}^{(0)} \in \mathbb{R}^{n\times d}$$

$$\boldsymbol{y}_i^{(k+1)} = \sum_{j=1}^{n} p_{i,j}^{(k)}\boldsymbol{y}_j^{(k)}$$

$$\boldsymbol{Y}^{(k+1)} = \boldsymbol{P}^{(k)}\boldsymbol{Y}^{(k)} \qquad \boldsymbol{P}^{(k)} \in \mathbb{R}^{n\times n}$$

Doubly-stochastic Matrix

$$P = \begin{pmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{pmatrix}, \begin{pmatrix} 0.3 & 0.7 \\ 0.7 & 0.3 \end{pmatrix}, \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$$

$$\lim_{K\to\infty}\prod_{k=0}^{K}\boldsymbol{P}^{(k)} = \{p(i,j) = \frac{1}{n}\} \qquad \boldsymbol{y}_i^{(\infty)} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{y}_i^{(0)}$$

Consensus

Column-stochastic Matrix

$$\boldsymbol{Y}^{(k+1)} = \boldsymbol{P}^{(k)}\boldsymbol{Y}^{(k)} \qquad \boldsymbol{P}^{(k)} \in \mathbb{R}^{n\times n}$$

$$P = \begin{pmatrix} 0.2 & 0.3 \\ 0.8 & 0.7 \end{pmatrix}, \begin{pmatrix} 0.3 & 0.4 \\ 0.7 & 0.6 \end{pmatrix}, \begin{pmatrix} 0.23 & 0.23 \\ 0.76 & 0.76 \end{pmatrix}$$

$$\lim_{K\to\infty} \prod_{k=0}^{K} \boldsymbol{P}^{(k)} = \boldsymbol{\pi}\mathbf{1}^\top$$

$$\boldsymbol{Y}^{(\infty)} = \boldsymbol{\pi}\left(\mathbf{1}^\top \boldsymbol{Y}^{(0)}\right)$$

$$\boldsymbol{y}_i^{(\infty)} = \pi_i \sum_{j=1}^{n} \boldsymbol{y}_j^{(0)}$$

Add a scalar parameter $w_i^{(k)}$

Consensus

Column-stochastic Matrix

$$\lim_{K\to\infty} \prod_{k=0}^{K} \boldsymbol{P}^{(k)} = \boldsymbol{\pi}\mathbf{1}^\top$$

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{P}^{(k)}\boldsymbol{w}^{(k)} \qquad w_i^{(0)} = 1$$

$$\boldsymbol{w}^{(\infty)} = \boldsymbol{\pi}\left(\mathbf{1}^\top\boldsymbol{w}^{(0)}\right)$$

$$w_i^{(\infty)} = \pi_i n$$

De-biased ratio

$$\frac{\boldsymbol{y}_i^{(\infty)}}{w_i^{(\infty)}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{y}_i^{(0)}$$

Stochastic Gradient Push

Algorithm 1 Stochastic Gradient Push (SGP)

Require: Initialize $\gamma > 0$, $\boldsymbol{x}_i^{(0)} = \boldsymbol{z}_i^{(0)} \in \mathbb{R}^d$ and $w_i^{(0)} = 1$ for all nodes $i \in \{1, 2, \ldots, n\}$

for $k = 0, 1, 2, \cdots, K$, at node i, **do**

  Sample new mini-batch $\xi_i^{(k)} \sim \mathcal{D}_i$ from local distribution

  Compute mini-batch gradient at $\boldsymbol{z}_i^{(k)}$: $\nabla \boldsymbol{F}_i(\boldsymbol{z}_i^{(k)}; \xi_i^{(k)})$

  $\boldsymbol{x}_i^{(k+\frac{1}{2})} = \boldsymbol{x}_i^{(k)} - \gamma \nabla \boldsymbol{F}_i(\boldsymbol{z}_i^{(k)}; \xi_i^{(k)})$

  Send $\left(p_{j,i}^{(k)} \boldsymbol{x}_i^{(k+\frac{1}{2})}, p_{j,i}^{(k)} w_i^{(k)}\right)$ to out-neighbors;
  receive $\left(p_{i,j}^{(k)} \boldsymbol{x}_j^{(k+\frac{1}{2})}, p_{i,j}^{(k)} w_j^{(k)}\right)$ from in-neighbors

  $\boldsymbol{x}_i^{(k+1)} = \sum_j p_{i,j}^{(k)} \boldsymbol{x}_j^{(k+\frac{1}{2})}$

  $w_i^{(k+1)} = \sum_j p_{i,j}^{(k)} w_j^{(k)}$

  $\boldsymbol{z}_i^{(k+1)} = \boldsymbol{x}_i^{(k+1)} / w_i^{(k+1)}$

end for

HHHFL: Hierarchical Heterogeneous Horizontal Federated Learning for Electroencephalography

DashanGao, CeJu, Xiguang Wei, Yang Liu, Tianjian Chen, Qiang Yang

Heterogeneous Data

Electroencephalography (EEG)

Heterogeneous & Privacy

Data Heterogeneity

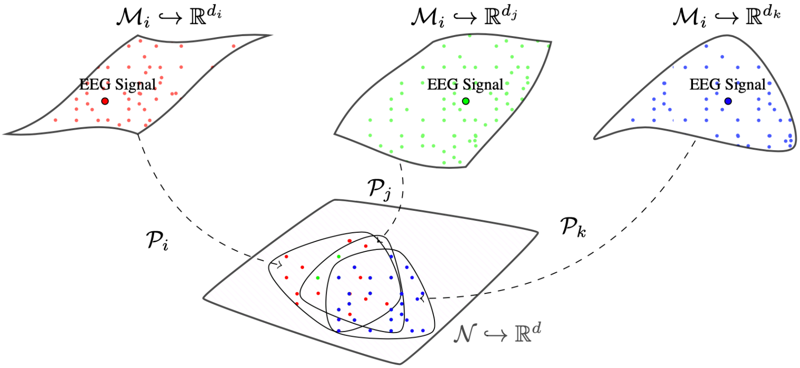
Heterogeneous Domain Adaptation

Privacy-Preserving

Federated Learning

Heterogeneous Domain Adaptation

Heterogeneous Domain Adaptation



$\mathcal{P}_i$?

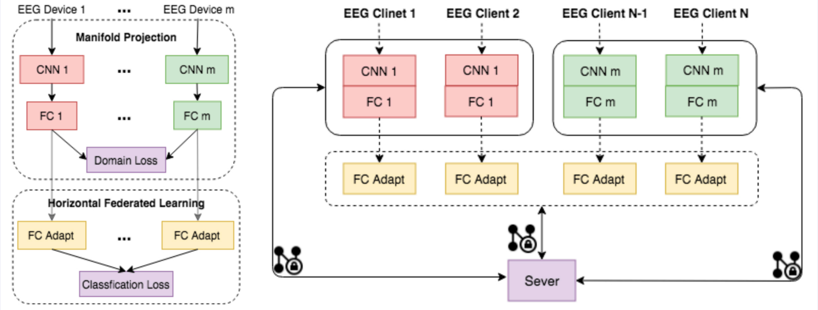
Neural Network Approach

Heterogeneous Domain Adaptation

Loss Function

Classification Loss & Domain Loss

Heterogeneous Domain Adaptation & Federated Learning



Maximum Mean Discrepancy (MMD)

Maximum Mean Miscrepancy (MMD)

$$\mathcal{L} := \mathcal{L}_C \left(X_{EEG}, Y \right) + \sum_{1 \leq i < j \leq m} \lambda_{i,j} \cdot \mathrm{MMD}^2 \left(\mathcal{Q}_i, \mathcal{Q}_j \right)$$

$$\mathrm{MMD} \left(\mathcal{Q}_i, \mathcal{Q}_j \right) := \left\| \mathbb{E}_{\mathcal{P}_i(X_i) \sim \mathcal{Q}_i} \psi \left(\mathcal{P}_i \left(X_i \right) \right) - \mathbb{E}_{\mathcal{P}_j(X_j) \sim \mathcal{Q}_j} \psi \left(\mathcal{P}_j \left(X_j \right) \right) \right\|_{\mathcal{H}}$$

$$\mathcal{P}_i \left(\{ x_1, \cdots, x_{N_i} \} \right) \sim \mathcal{Q}_i, \quad \psi : \mathcal{N} \longrightarrow \mathcal{H}$$

Reproducing Kernel Hilbert Space (RKHS)

Maximum Mean Discrepancy (MMD)

MMD in Euclidean Space

$$MMD[F, p, q] := \sup_{f \in F} \left(E_{x \sim p}[f(x)] - E_{y \sim q}[f(y)] \right)$$

$$MMD[F, X, Y] := \sup_{f \in F} \left(\frac{1}{m} \sum_{i=1}^{m} f(x_i) - \frac{1}{n} \sum_{j=1}^{n} f(x_j) \right)$$

F is rich enough but not too much

Maximum Mean Discrepancy (MMD)

Reproducing Kernel Hilbert Space (RKHS)

$$f(x) = \langle f, \phi(x) \rangle_{\mathcal{H}} \qquad f \to f(x)$$

$$E_p[f(x)] = \langle f, E_p[\phi(x)] \rangle_{\mathcal{H}}$$

MMD in RKHS

$$\begin{aligned} MMD[F, p, q] &= \sup_{\|f\|_{\mathcal{H}} \leq 1} E_p[f(x)] - E_q[f(y)] \\ &= \sup_{\|f\|_{\mathcal{H}} \leq 1} E_p\left[\langle \phi(x), f \rangle_{\mathcal{H}}\right] - E_p\left[\langle \phi(y), f \rangle_{\mathcal{H}}\right] \\ &= \sup_{\|f\|_{\mathcal{H}} \leq 1} \langle \mu_p - \mu_q, f \rangle_{\mathcal{H}} \\ &= \|\mu_p - \mu_q\|_{\mathcal{H}} \end{aligned}$$

Maximum Mean Discrepancy (MMD)

MMD in RKHS

$$
\begin{aligned}
MMD^2[F,p,q] &:= \langle \mu_p - \mu_q, \mu_p - \mu_q \rangle_{\mathcal{H}} \\
&= \langle \mu_p, \mu_q \rangle_{\mathcal{H}} + \langle \mu_q, \mu_p \rangle_{\mathcal{H}} - 2 \langle \mu_p, \mu_q \rangle_{\mathcal{H}} \\
&= E_p \left\langle \phi(x), \phi\left(x'\right) \right\rangle_{\mathcal{H}} + E_q \left\langle \phi(y), \phi\left(y'\right) \right\rangle_{\mathcal{H}} \\
&- 2E_{p,q} \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}
\end{aligned}
$$

Universal Kernel

$$
k\left(x, x'\right) = \exp\left(-\frac{\left\|x - x'\right\|^2}{2\sigma^2}\right)
$$

MMD in Euclidean Space

$$
MMD^2[F,p,q] = \frac{1}{m(m-1)} \sum_{i \neq j}^{m} k\left(x_i, x_j\right) + \frac{1}{n(n-1)} \sum_{i \neq j}^{n} k\left(y_i, y_j\right) - \frac{2}{mn} \sum_{i,j=1}^{m,n} k\left(x_i, y_j\right)
$$

The End