Decentralized Deep Learning with Inexact Consensus

Arjun Ashok Rao[†], Hoi-To Wai[⋆]

[†]The Chinese University of Hong Kong (Hong Kong)

FTEC4998 Final Year Project 1, Dec., 2021.

Contents

Background

- Decentralized Optimization
- Application: Machine Learning over Graphs

Problem Formulation

- CHOCO-SGD in the Overparameterized Regime
- Numerical Experiments I
- Consensus with overparameterized models

Solution: RKHS-valued SGD

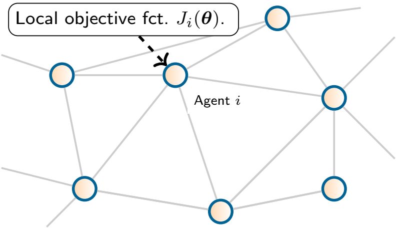
- [Koppel et al., 2018] proposed solution
- Numerical Experiment II
- Proposed solution

Problem Description: Decentralized Consensus Optimization Problem

- Consider a **finite sum unconstrained** optimization of a d-dimensional variable $\boldsymbol{\theta}$:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \quad J(\boldsymbol{\theta}) := \frac{1}{N} \sum_{i=1}^{N} J_i(\boldsymbol{\theta}). \tag{1}$$

- Where $d \in \mathbb{N}$ is the problem dimension
- $J_i : \mathbb{R}^d \to \mathbb{R}$ is a continuous, differential private objective function of worker i
- $G = (V, E)$ is an **undirected communication graph**; $V = [N] = \{1, ..., N\}$ represents the set of N workers and $(i, i) \in E \quad \forall \quad i \in V$



Problem Description: Decentralized Consensus Optimization Problem

- Consider a **finite sum unconstrained** optimization of a d-dimensional variable $\boldsymbol{\theta}$:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \quad J(\boldsymbol{\theta}) := \frac{1}{N} \sum_{i=1}^{N} J_i(\boldsymbol{\theta}). \tag{1}$$

- Equation (1) can be written as the decentralized consensus optimization problem:

$$\min_{\boldsymbol{\theta}_i \in \mathbb{R}^d, i \in V} \quad \sum_{i=1}^{N} J_i(\boldsymbol{\theta}_i) \;\; \text{s.t.} \;\; {\color{blue}\boldsymbol{\theta}_i = \boldsymbol{\theta}_j}, \; \forall \; (i,j) \in E \tag{2}$$

- $\boldsymbol{\theta}_i \in \mathbb{R}^d$ is a private/local variable held by the ith worker.

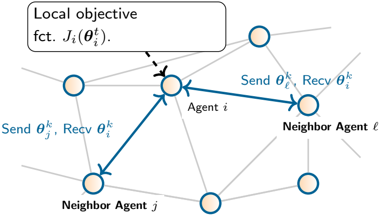
Background: Decentralized Deep Learning

Our problem —
$$\min_{\boldsymbol{\theta}_i \in \mathbb{R}^d, i \in V} \sum_{i=1}^{N} J_i(\boldsymbol{\theta}_i) \text{ s.t. } \boldsymbol{\theta}_i = \boldsymbol{\theta}_j,\ \forall\ (i,j) \in E\ .$$

- We are interested in training a large neural network (NN) over N workers. For a supervised classification problem, $J_i(\boldsymbol{\theta})$ takes the form of empirical risk:

$$J_i(\boldsymbol{\theta}) = \frac{1}{|\mathcal{D}_i|} \sum_{j=1}^{|\mathcal{D}_i|} \text{loss}(f(\boldsymbol{x}_j; \boldsymbol{\theta}); y_j) \tag{3}$$

- The N workers must learn a common model $\boldsymbol{\theta}^*$ given only a subset of the training data $D = \cup_{i=1}^{M} D_i$.
- Solution: **consensus + optimize strategy** where workers communicate with neighbors to optimize their $\boldsymbol{\theta}_i$.

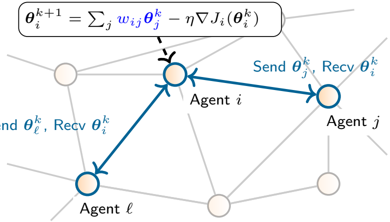


Decentralized Gradient Descent (DGD) Method

1. Agent i holds local parameter copy $\boldsymbol{\theta}_i^t$ on iteration t.
2. Calculate local gradient $\nabla J_i(\boldsymbol{\theta}_i^k)$
3. receive $\boldsymbol{\theta}_j$ from neighbors $\forall j \in V, \quad W_{i,j} > 0$

$$\boldsymbol{\theta}_i^{k+\frac{1}{2}} = \underbrace{\sum_j w_{ij}\boldsymbol{\theta}_j^k}_{\text{Gossip Averaging}}$$

4. Update $\boldsymbol{\theta}_i^{k+1} - \eta \nabla J_i(\boldsymbol{\theta}_i^k)$



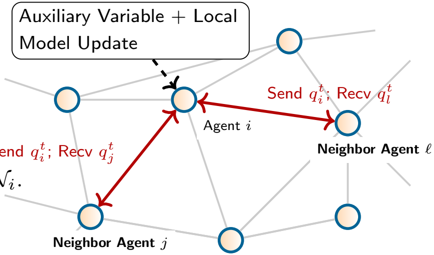
Improvement: D-PSGD Method: Local Stochastic Gradient and Gossip Averaging Run in Parallel

- $\boxed{\boldsymbol{g}^k(\boldsymbol{\theta}_i^k; \xi_i^k)} := \sum_j \nabla J_i(\boldsymbol{\theta}_i^k; \xi_i^k) \xrightarrow{avg} \left[\boldsymbol{\theta}^1_{k+\frac{1}{2}}, \boldsymbol{\theta}^d_{k+\frac{1}{2}}, \ldots, \boldsymbol{\theta}^n_{k+\frac{1}{2}}\right] = [\boldsymbol{\theta}_k^1, \boldsymbol{\theta}_k^2, , \ldots, \boldsymbol{\theta}_k^n]W_k$
- **Drawback:** Limited Communication Bandwidth; Increases with dimensionality d

CHOCO-SGD [Koloskova et al., 2019a]

- Solution: Communication compression of $\boldsymbol{\theta}_i$ with a compression operator $\mathcal{Q} : \mathbb{R}^d \to \mathbb{R}^d$
- Assumption 1: $\mathbb{E}_\Omega \left[\|\mathcal{Q}(\boldsymbol{\theta};\Omega) - \boldsymbol{\theta}\|^2\right] \leq (1-\delta)\|\boldsymbol{\theta}\|^2, \quad \forall\, \boldsymbol{\theta} \in \mathbb{R}^d$
  - ω is the randomness of compression operator; $\delta \in (0,1]$ denotes compression error
- Assumption 2: $\mathbb{E}[\boldsymbol{g}_i^{(t)}|\mathcal{F}_t] = \nabla J_i(\boldsymbol{\theta}_i^{(t)}) \quad \mathbb{E}[\|\boldsymbol{g}_i^{(t)} - \nabla J_i(\boldsymbol{\theta}_i^{(t)})\|^2|\mathcal{F}_t] \leq \sigma^2.$
- Assumption 3: Lipschitz-Smooth Gradient $\nabla J_i(\boldsymbol{\theta})$

1. Local SGD: $\boldsymbol{\theta}_i^{t+1/2} = \boldsymbol{\theta}_i^t - \eta_t \boldsymbol{g}_i^t$
2. Agent i: Send a difference vector $q_i^t = \mathcal{Q}(\boldsymbol{\theta}_i^{(t+\frac{1}{2})} - \hat{\boldsymbol{\theta}}_{i,i}^{(t)})$, receive q_j^t from neighbors $\forall j \in V, \quad W_{i,j} > 0$
3. Update an auxiliary variable:
$$\hat{\boldsymbol{\theta}}_{i,j}^{(t+1)} = \hat{\boldsymbol{\theta}}_{i,j}^{(t)} + \mathcal{Q}(\boldsymbol{\theta}_j^{(t+\frac{1}{2})} - \hat{\boldsymbol{\theta}}_{j,j}^{(t)}), \; \forall\, j \in \mathcal{N}_i.$$
4. Update Local Model:
$$\boldsymbol{\theta}_i^{(t+1)} = \boldsymbol{\theta}_i^{(t+\frac{1}{2})} + \gamma \sum_{j \in \mathcal{N}_i} W_{ij} \{\hat{\boldsymbol{\theta}}_{i,j}^{(t+1)} - \hat{\boldsymbol{\theta}}_{i,i}^{(t+1)}\}.$$



Convergence of CHOCO-SGD

Theorem — Convergence of CHOCO-SG [Koloskova et al., 2019a, Koloskova et al., 2019b]

Under Assumptions 1, 2, and 3, There exits $\eta, \gamma > 0$ such that if we consider a constant step size with $\eta_t \equiv \eta$, then for any $T \geq 1$,$\eta, \gamma > 0$

$$\mathbb{E}[\|\nabla J(\bar{\boldsymbol{\theta}}^{(\mathsf{T})})\|^2] = \mathcal{O}\left(\sqrt{\frac{L\sigma^2 J_0}{NT}} + \left(\frac{LGJ_0}{\rho^2 \delta T}\right)^{\frac{2}{3}}\right)$$

- $\delta \in (0,1]$ is the compression error $\rho \in (0,1]$ is the spectral gap of W
- $\bar{\boldsymbol{\theta}}^{(t)} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\theta}_i^{(t)}$ $J_0 = J(\bar{\boldsymbol{\theta}}^{(0)}) - \min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

Question: How well does CHOCO-SGD converge for $d \gg 1$?

- when $\delta = \frac{k}{d}$, we apply Theorem 1 to get...

CHOCO-SGD in the **Overparameterized** Regime

Convergence of CHOCO-SGD with $m \gg 1$

Consider a rand$_k$ or top$_k$ sparsifier with fixed co-ordinate retention k. Fix number of training iterations at T. From Theorem 1, we have:

$$\mathbb{E}[\|\nabla J(\overline{\boldsymbol{\theta}}^{(\mathsf{T})})\|^2] = \mathcal{O}\Big(\sqrt{\frac{L\sigma^2 J_0}{NT}} + d^{\frac{2}{3}}\Big(\frac{LGJ_0}{\rho^2 kT}\Big)^{\frac{2}{3}}\Big)$$

For $\mathbb{E}[\|\nabla J(\overline{\boldsymbol{\theta}}^{(\mathsf{T})})\|^2] \le \epsilon$, Minimum iterations required T is of the order:

$$T = \Omega\left(LJ_0 \cdot \max\left\{\frac{\sigma^2}{N\epsilon^2}, \frac{d}{k}\frac{G}{\rho^2\epsilon^{1.5}}\right\}\right)$$

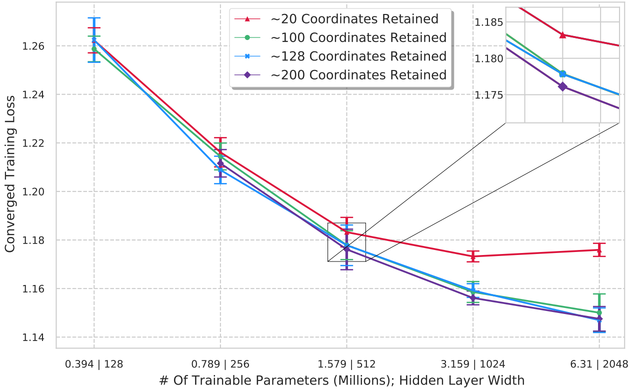
- Implication: Communication cost/iteration reduced, but Compressed DSG algorithms require more iterations to converge
- Pitfall in existing theory! Need to observe implications for practical performance.

Numerical Experiments – Two-Layer ReLU Network

Goal: Empirically investigate convergence of CHOCO-SGD with Overparameterized NNs

- Decentralized graph simulated by an MPI network environment with a fixed communication graph W.
  - Independent CPU process assigned to each worker.
- Train Dataset: CIFAR-10 – 10 classes, 50K datapoints as a $32 \times 32 \times 3$ RGB image divided in an i.i.d fashion among N workers; reshuffled every epoch.
- Test Dataset: To test generalization ability, CIFAR-10.1 [Recht et al., 2018]
- Model: ReLU Linear NNs with increasing layer widths $\underbrace{m = [128, 256, 512, 1024, 2048]}_{0.3 \text{ to } 6.31 \times 10^6 \text{ parameters}}$
- Constant consensus (γ) and SGD (η) step size run over a constant number of training iterations (T). top_k and rand_k used with constant number of co-ordinates retained k

Converged Training Loss vs Model Dimensionality – CIFAR10: top$_k$ sparsification



- Setting: $N = 8$ workers on a ring topology, CIFAR-10, 300 epochs, 2-layer ReLU network with increasing m and constant k (#bits transmitted is constant)
- **Overparameterized models exhibit better convergence and training loss decreases with increase in d.**

Are Overparameterized Models in Consensus?

- Consensus Distance captures expected disagreement between averaged model $\bar{\boldsymbol{\theta}}^T$ and each node $\boldsymbol{\theta}_i$:

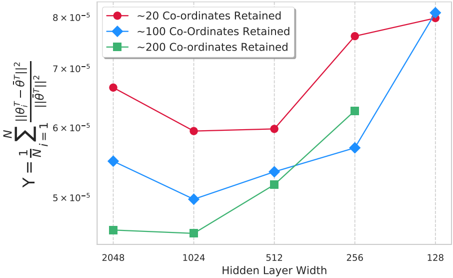
$$\Upsilon = \frac{1}{N}\sum_{i=1}^{N} \frac{\|\boldsymbol{\theta}_i^T - \bar{\boldsymbol{\theta}}^T\|^2}{\|\bar{\boldsymbol{\theta}}^T\|^2}$$

- If Υ satisfies the following bound [Kong et al., 2021]

$$\Upsilon_t^2 \leq \left(\frac{1}{Ln}\gamma\sigma^2 + \frac{1}{8L^2}\|\nabla J(\bar{\boldsymbol{\theta}}^T)\|^2\right)$$

we can recover centralized SGD's convergence rate with a larger stepsize $\gamma \leq \gamma_{max}$

- **Overparameterized models enjoy greater consensus among workers with only marginal dependence on k**



Problem: Consensus is Expensive in The Overparameterized Regime

Layer Width	Normalized Consensus Distance		
	Epoch $= 200$	Epoch $= 100$	Epoch $= 50$
2048	5.499×10^{-5}	9.8206×10^{-3}	1.3977×10^{-2}
1024	4.980×10^{-5}	1.0346×10^{-2}	1.5307×10^{-2}
512	5.349×10^{-5}	1.0026×10^{-3}	1.3478×10^{-2}
256	5.694×10^{-5}	8.7639×10^{-3}	1.2423×10^{-2}
128	8.098×10^{-5}	7.3181×10^{-3}	9.2698×10^{-3}

Problem: Consensus is Expensive in The Overparameterized Regime

Layer Width	Normalized Consensus Distance		
	Epoch $= 200$	Epoch $= 100$	Epoch $= 50$
2048	5.499×10^{-5}	9.8206×10^{-3}	1.3977×10^{-2}
1024	4.980×10^{-5}	1.0346×10^{-2}	1.5307×10^{-2}
512	5.349×10^{-5}	1.0026×10^{-3}	1.3478×10^{-2}
256	5.694×10^{-5}	8.7639×10^{-3}	1.2423×10^{-2}
128	8.098×10^{-5}	7.3181×10^{-3}	9.2698×10^{-3}

Average consensus is expensive for overparameterized models. Can DSGD algorithms with overparameterized models converge with inexact consensus?

From Parameter Estimation to Function Estimation

- Consider the objective of learning regressors $\tilde{f} \in \mathcal{H}$ for hypothesized function class $\mathcal{H}$
- (x_n, y_n) are drawn i.i.d over $(\mathrm{x}, \mathrm{y}) \in \mathcal{X} \times \mathcal{Y}$ s.t $\mathcal{X} \subset \mathbb{R}^p$ (feature vector) and $\mathcal{Y} \subset \mathbb{R}$ (label)
- Now, consider empirical risk formulation to find optimal function $f^* \in \mathcal{H}$

$$\underset{\tilde{f} \in \mathcal{H}}{\operatorname{argmin}}\ J_i(\tilde{f}) = \frac{1}{|\mathcal{D}_i|} \sum_{j=1}^{|\mathcal{D}_i|} \text{loss}(\tilde{f}(\boldsymbol{x}_j); y_j) + \underbrace{\frac{\lambda}{2} \|\tilde{f}\|_{\mathcal{H}}^2}_{\text{Hilbert Norm Penalty}} \tag{4}$$

- where loss is a strictly convex loss function used to penalize the deviation of regressor f from the output label y given by $l\ :\ \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{R}$

This problem is intractable!

From Parameter Estimation to Function Estimation

- For decentralized learning, impose functional consensus constraints [Koppel et al., 2018]:

$$J^T = \operatorname*{argmin}_{f_i \subset \mathcal{H}} \left(\sum_{i \in V} \left(\mathbb{E}_{x_i, y_i} \left[l_i f_i(x), y_i) \right] + \frac{\lambda}{2} \|f_i\|^2_{\mathcal{H}} \right) \tag{5}$$

$$\text{such that } f_i = f_j \ \forall \ (i,j) \in E \tag{6}$$

- To solve 5, equip the hypothesized function class $\mathcal{H}$ with a kernel function over the feature vector space $\kappa \ : \ \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that satisfies:

$$\langle f, \kappa(x_i, \cdot) \rangle_{\mathcal{H}} = f(x_i) \qquad \mathcal{H} = \overline{span(\kappa(x_i), \cdot)} \tag{7}$$

From Parameter Estimation to Function Estimation

- To solve 5, equip the hypothesized function class $\mathcal{H}$ with a kernel function over the feature vector space $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that satisfies:

$$\langle f, \kappa(x_i, \cdot)\rangle_{\mathcal{H}} = f(x_i) \qquad \mathcal{H} = \overline{span(\kappa(x_i), \cdot)} \tag{5}$$

- Given 7 is satisfied, $\mathcal{H}$ is an RKHS. Note that from 7, we also get:

$$\tilde{f}(\mathsf{x}_i) = \sum_N w_{i,n} \kappa(\mathsf{x}_{i,n}, \mathsf{x}_i) \tag{6}$$

$$\Rightarrow J^T = \underbrace{\operatorname*{argmin}_{w \in \mathbb{R}^n}}_{\text{Kernel Trick!}} \frac{1}{N} \sum_{i=i}^{N} \left(\text{loss} \left(\sum_{j=1}^{N} w_j \kappa(\mathsf{x}_j, \mathsf{x}_i), y_i \right) \right) \tag{7}$$

$$+ \frac{\lambda}{2} \| \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \kappa(\mathsf{x}_j, \mathsf{x}_i) \|_{\mathcal{H}}^2 \tag{8}$$

From Parameter Estimation to Function Estimation

- Given 7 is satisfied, $\mathcal{H}$ is an RKHS. Note that from 7, we also get:

$$\tilde{f}(\mathsf{x}_i) = \sum_{N} w_{i,n}\kappa(\mathsf{x}_{i,n}, \mathsf{x}_i) \tag{5}$$

$$\Rightarrow J^T = \underbrace{\underset{w \in \mathbb{R}^n}{\text{argmin}}}_{\text{Kernel Trick!}} \frac{1}{N}\sum_{i=i}^{N}\left(\text{loss}\left(\sum_{j=1}^{N} w_j\kappa(\mathsf{x}_j, \mathsf{x}_i), y_i\right)\right) \tag{6}$$

$$+\frac{\lambda}{2}\|\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \kappa(\mathsf{x}_j, \mathsf{x}_i)\|_{\mathcal{H}}^2 \tag{7}$$

- As training points $n \to \infty$, infinite memory requirement

RKHS stochastic saddle-point problems in the Decentralized Setting

- Formulate functional consensus constraint $f_i = f_j \; \forall (i,j) \in \mathcal{E}$ as a penalty function [Koppel et al., 2018]:

$$\min \sum_{i \in \mathcal{V}} (\mathbb{E}_{(x_i, y_i)} \left[l_i(f_i(x_i, y_i)\right] + \frac{\lambda}{2} \|f_i\|_{\mathcal{H}}^2 + \frac{c}{2} \sum_{j \in n_i} \mathbb{E}_{x_i}([f_i(x_i) - f_j(x_i)]^2) \tag{8}$$

$$:= \min \sum_{i \in \mathcal{V}} (l_i(f_i(x_{i,t}), y_{i,t}) + \frac{\lambda}{2} \|f_i\|_{\mathcal{H}}^2 + \underbrace{\frac{c}{2} \sum_{j \in n_j} (f_i(x_{i,t}) - f_j(x_{i,t}))^2)}_{\text{Inexact consensus Penalty}} \tag{9}$$

(i.i.d samples $(x_{i,t}, y_{i,t})$ are revealed to each worker f_i) (10)

(11)

- Where the representer theorem implies that at time t, the regressor f can be expanded as:

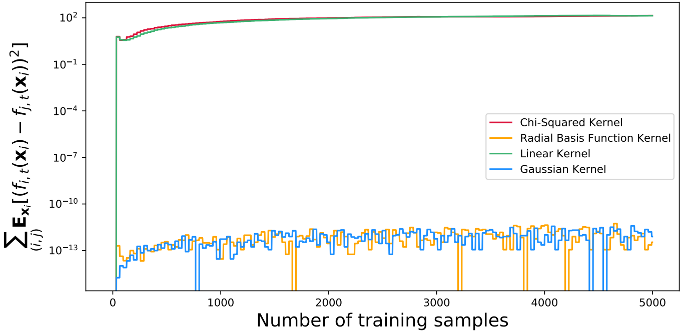
$$f_{i,t}(x) = \sum_{n=1}^{t-1} w_{i,n} \kappa(x_{i,n}, x) = w_{i,t}^T \kappa_{x_{i,t}}(x)$$

Numerical Experiments II

Question: What is the effect of kernel choice $\kappa(\cdot)$ on consensus term

$\frac{c}{2}\sum_{j\in n_i}\mathbb{E}_{x_i}([f_i(x_i)-f_j(x_i)]^2)$.

- Implement Gaussian and Radial basis kernel $\kappa(x,x^{'})=exp\left(-\frac{\|x-x^{'}\|^2}{2\sigma^2}\right)$ and compare consensus error with polynomial kernel, chi-square kernel.

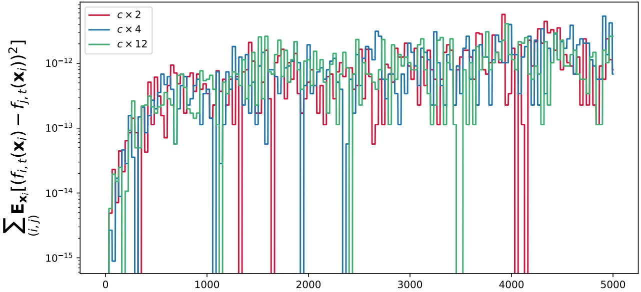


Numerical Experiments II

Question: What is the effect of kernel choice $\kappa(\cdot)$ on consensus term

$$\boxed{\frac{c}{2}\sum_{j\in n_i}\mathbb{E}_{x_i}([f_i(x_i)-f_j(x_i)]^2).}$$

- Implement Gaussian and Radial basis kernel $\kappa(x, x^{'}) = exp\left(-\frac{\|x-x^{'}\|^2}{2\sigma^2}\right)$ and compare consensus error with polynomial kernel, chi-square kernel.



- Problem: High dependence on choice of kernel, consensus reaches machine precision zero, and inefficient as $N \to \infty$ (overparameterized models use $N \gg 1$ training samples)

Inex-SGD for Inexact Consensus Deep Learning

- Consider a dense feedforward NN model on the ith worker $f_i(x_i;\ \boldsymbol{\theta}_i)$. We are interested in the following optimization problem:

$$\min_{f_i} \sum_{i=1}^{N} \left(\left[\mathbb{E}[l_i(f_i(x_i, y_i))] + \frac{\lambda}{2}\|f_i\|^2 \right] + \frac{c}{2} \sum_{j \in \mathcal{N}_i} \mathbb{E}_{x_i}[|f_i(x_i) - f_j(x_i)|^2] \right) \tag{12}$$

For the NN model parameterized by $\boldsymbol{\theta}$, we have: (13)

$$min_{\boldsymbol{\theta}_i \forall i=1,\ldots,N} \sum_{i=1}^{N} \left(\mathbb{E}_{x_i}\left[l_i(f(x_i;\ \boldsymbol{\theta}_i), y_i)\right] + \frac{\lambda}{2}\|\boldsymbol{\theta}_i\|^2 \right) \tag{14}$$

$$+ \sum_{j \in \mathcal{N}_i} \mathbb{E}_{x_i}\left[\frac{c}{2}|f(x_i;\boldsymbol{\theta}_i) - f(x_i;\boldsymbol{\theta}_j)|^2\right] \tag{15}$$

Inex-SGD for Inexact Consensus Deep Learning

1. Agent i holds local parameter copy $\boldsymbol{\theta}_i^t$ on iteration t, and mini-batch sample $\xi_{i,k} = [\xi_i^{k,1}, \xi_i^{k,2}, \ldots, \xi_i^{k,M}]$
2. Evaluate model on batch $\sum_{j=1}^{M} J(\boldsymbol{\theta}_i^k, \xi_i^{k,j})$
3. Receive $\left(\sum_{p=1}^{M} J(\boldsymbol{\theta}_j^k, \xi_j^{k,p}),\ \xi_{j,k}\right)$ $\forall j$ in $\mathcal{N}_i$
4. Calculate Stochastic gradient on worker i:

$$\begin{aligned} g_i^k &= \nabla_{\boldsymbol{\theta}_i} l_i(J(\xi_{i,k})) + \lambda \boldsymbol{\theta}_i^k \\ &\quad + c \sum_{j \in \mathcal{N}_i} \left(J(\xi_i^k; \boldsymbol{\theta}_i^k) - J(\xi_i^k; \boldsymbol{\theta}_j^k) \right) \nabla J(\xi_{i,k}, \boldsymbol{\theta}_i^k) \\ &\quad + c \sum_{j \in \mathcal{N}_i} \left(J(\xi_j^k; \boldsymbol{\theta}_i^k) - J(\xi_j^k; \boldsymbol{\theta}_j^k) \right) \nabla J(\xi_{j,k}, \boldsymbol{\theta}_i^k) \end{aligned} \tag{12}$$

5. Perform SGD Update: $\boldsymbol{\theta}_i^{k+1} = \boldsymbol{\theta}_i^k - \eta^k g_i^k$

Inex-SGD for Inexact Consensus Deep Learning

- Stochastic gradient on worker i on iteration k is given by:

$$\begin{aligned} g_i^k &= \nabla_{\boldsymbol{\theta}_i} l_i(J(\xi_{i,k})) + \lambda \boldsymbol{\theta}_i^k \\ &+ c \sum_{j \in \mathcal{N}_i} \Big(J(\xi_i^k; \boldsymbol{\theta}_i^k) - J(\xi_i^k; \boldsymbol{\theta}_j^k) \Big) \nabla J(\xi_{i,k}, \boldsymbol{\theta}_i^k) \\ &+ c \sum_{j \in \mathcal{N}_i} \Big(J(\xi_j^k; \boldsymbol{\theta}_i^k) - J(\xi_j^k; \boldsymbol{\theta}_j^k) \Big) \nabla J(\xi_{j,k}, \boldsymbol{\theta}_i^k) \end{aligned} \tag{12}$$

- Problem: To calculate $\nabla J(\xi_{i,k}, \boldsymbol{\theta}_i^k)$ and $\nabla J(\xi_{j,k}, \boldsymbol{\theta}_i^k)$, each worker i must send data points/ mini-batches $\xi_{i,k}, \xi_{j,k}$ to neighbors $j \in \mathcal{N}_i$. Not recommended for sensitive data.

Summary of Findings

How do compressed DSGD algorithms perform in the overparameterized regime?

- Utilizing overparameterized NNs in the decentralized setting is **practical** and **beneficial**
- However, overparameterized models reach consensus with an increased cost compared to smaller NN models.

Proposed Solution

- Inexact Consensus with RKHS-valued functional SGD reformulated without dependence on kernel choice.

Next Steps

- Fix bugs in algorithm development; more rigorous analysis of proposed solution
- Propose a more secure alternative

Thank you! Questions?

References

[Koloskova et al., 2019a] Koloskova, A., Lin, T., Stich, S. U., and Jaggi, M. (2019a).
Decentralized deep learning with arbitrary communication compression.
arXiv preprint arXiv:1907.09356.

[Koloskova et al., 2019b] Koloskova, A., Stich, S., and Jaggi, M. (2019b).
Decentralized stochastic optimization and gossip algorithms with compressed communication.
In *International Conference on Machine Learning*, pages 3478–3487. PMLR.

[Kong et al., 2021] Kong, L., Lin, T., Koloskova, A., Jaggi, M., and Stich, S. U. (2021).
Consensus control for decentralized deep learning.
arXiv preprint arXiv:2102.04828.

[Koppel et al., 2018] Koppel, A., Paternain, S., Richard, C., and Ribeiro, A. (2018).
Decentralized online learning with kernels.
IEEE Transactions on Signal Processing, 66(12):3240–3255.

[Recht et al., 2018] Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. (2018).
Do cifar-10 classifiers generalize to cifar-10?
arXiv preprint arXiv:1806.00451.