Natural Compression for Distributed Deep Learning

Samuel Horváth 1 Chen-Yu Ho 1 Ludovít Horváth 2 Atal Sahu 1 Marco Canini 1 Peter Richtárik 1,3

1 KAUST 2 Comenius University 3 Moscow Institute of Physics and Technology

The Problem
\[
\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x),
\]

- \( f_i(x) := E_{C_{nat}}[f_i(x)] \), where \( D_i \) is the distribution of data stored on worker \( i \)
- \( f \) is smooth, \( f_i \)'s have bounded variance \( \sigma_i^2 \)
- \( E \left[ \| \nabla f_i(x) - \nabla f_i(x') \|^2 \right] \leq \sigma_i^2; \quad \sigma_i^2 = \frac{1}{n} \sum \sigma_i^2, \)
- \( n \) is number of nodes

Communication as the Bottleneck

- A key bottleneck of distributed SGD is the cost of communication of the typically dense gradient vectors \( g_i(x) \).
- In typical distributed computing environments, communication takes more time than computation.
- Two orthogonal types of remedies:
  - Local iterations: give each worker “more useful work” to do before any communication takes place
  - Gradient compression: communicate compressed gradients instead of full gradients

Main Contributions

- New Compression Operators. We construct a new “natural” operators based on a randomized binary rounding scheme.
- Computation-Free Simple Low-Level Implementation. “Natural” compatibility with binary floating point types.
- Post-compression Mechanism. Provable theoretical and practical speedup improvements through composition (\( \circ \)) with previous methods.
- Proof-of-Concept System with In-Network Aggregation. Our mechanisms are the first methods that are provably able to operate in the SwitchML [2] framework.
- Theory of general quantized SGD (Algorithm 1)

Compression Operators
\[
C : \mathbb{R}^d \rightarrow \mathbb{R}^d
\]
is called an unbiased bounded-second moment compression operator (not. \( C \in \mathbb{B}(\omega) \)) if
\[
E[C(x)] = x, \quad E[\|C(x)\|^2] \leq (\omega + 1)\|x\|^2, \quad \forall x \in \mathbb{R}^d.
\]

Natural Dithering \( D_{nat}^{\omega,s} \)

- Inexact version of natural compression.
- Fixing the number of level \( s \), natural dithering \( D_{nat}^{\omega,s} \) has \( O(2^{-s}) \times \text{times smaller variance} \) than standard dithering \( D_{nat}^{\omega} \).
- Randomized rounding for natural (left) and standard (right) dithering (\( s = 3 \) levels).

Numerical Results

- Communication as the Bottleneck

Computation-Free Simple Low-Level Implementation

- “Natural” compatibility with binary floating point types
- Post-compression Mechanism. Provable theoretical and practical speedup improvements through composition (\( \circ \)) with previous methods.
- Proof-of-Concept System with In-Network Aggregation. Our mechanisms are the first methods that are provably able to operate in the SwitchML [2] framework.
- Theory of general quantized SGD (Algorithm 1)

Natural Compression \( C_{nat} \)

- New (randomized) compression technique, which performs an element-wise randomized binary rounding of its input \( t \in \mathbb{R} \), \( C_{nat} \in \mathbb{B}(1/\omega) \).

\[
C_{nat}(t) \overset{def}{=} \begin{cases} 
\text{sign}(t) \cdot 2^{[\log_2|t|]} & \text{Prob. } p(t) \overset{def}{=} 2^{[\log_2|t|]} \frac{|t|}{2^{[\log_2|t|]}}, \\
\text{sign}(t) \cdot 2^{[\log_2|t|]} & \text{Prob. } 1 - p(t),
\end{cases}
\]

Bi-directional Compression

Algorithm 1: General Quantized SGD

\[
\begin{align*}
\text{init. vector } x^0 \in \mathbb{R}^d, \text{ step size } (\eta_i)_{i=0}^\infty > 0; \\
\text{for } k = 0 \text{ to } T \text{ do } \\
\text{for } i = 1 \text{ to } n \text{ do in parallel } \triangleright \text{Worker side} \\
&\text{compute stochastic gradient } g_i(x^k) \approx f_i(x^k) \\
&\text{compress stochastic gradient } \Delta_i = C_{nat}(g_i(x^k)) \\
&\text{aggregate compressed gradients } \Delta = \sum_1^n \Delta_i \\
&\text{compress aggregated vector } g^* = C_{nat}(\Delta) \\
&\text{broadcast } g^* \triangleright \text{Master side} \\
\text{for } i = 1, \ldots, n \text{ do in parallel } \triangleright \text{Worker side} \\
&x^{k+1} = x^k + \eta_i g^*_i; \\
\end{align*}
\]
