Gradient Descent with Compressed Iterates

Ahmed Khaled [1] Peter Richtárik [2]

[1] Cairo University [2] KAUST

Introduction

The training of high-dimensional federated learning models [1] reduces to solving an optimization problem of the form

$$x_* = \operatorname*{argmin}_{x\in\mathbb{R}^d} \left[f(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(x) \right], \qquad (1)$$

where n is the number of consumer devices (e.g., mobile devices), d is the number of parameters/features of the model, and $f_i : \mathbb{R}^d \to \mathbb{R}$ is a loss function that depends on the private data stored on the ith device. The simplest benchmark method used for the solution of problem (1) is Gradient Descent, which proceeds in iterates of the form

$$x_{k+1} = \frac{1}{n}\sum_{i=1}^{n} \left(x_k - \gamma \nabla f_i(x_k)\right).$$

More sophisticated methods used in Federated Learning take multiple local steps instead of a single step before averaging the resultant iterates, or compress the iterates before averaging [2], and the latter practice is the starting point of our work.

Gradient Compression

When gradients are communicated instead of iterates (as in stochastic gradient methods for distributed optimization over data clusters), the cost of gradient communication has been observed to be a significant bottleneck. As a result, there are many algorithms designed with the goal of reducing communication in stochastic gradient methods such as SignSGD, TernGrad, QSGD, DIANA, ChocoSGD, and others. There are also methods that apply variance reduction to remove the variance introduced by compression by doing more local computation, such as DIANA and VR-DIANA.

In contrast, there is little work on methods in which the iterates (as opposed to the gradients) are quantized or compressed. To bridge this gap in the theory, we consider the case of a single device ($n = 1$) using gradient descent to minimize a smooth and strongly convex function while compressing its local iterates.

Compression Operators

We call a stochastic function $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ a compression operator if it is unbiased

$$\mathbb{E}\left[\mathcal{C}(x)\right] = x,$$

and if there exists $\omega \geqslant 0$ such that for all $x \in \mathbb{R}^d$ we have,

$$\mathbb{E}\left[\left\|\mathcal{C}(x) - x\right\|^2\right] \leqslant \omega \|x\|^2.$$

Examples of compression operators are ubiquitous in the literature on gradient compression and include natural compression, dithering, random sparsification, ternary quantization, and others.

$$\begin{bmatrix} 2 & 4 & 3 \\ 3 & 2 & 0 \\ 1 & 0 & 9 \end{bmatrix} \Longrightarrow \frac{1}{1/3} \cdot \begin{bmatrix} 0 & 4 & 3 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Figure 1:An example of random sparsification where each element is set to 0 with probability $p = 1/3$ applied to a 3×3 matrix. The division by p is to ensure the estimator is unbiased.

Gradient Descent with Compressed Iterates (GDCI)

Algorithm 1 Gradient Descent with Compressed Iterates

Input: Stepsize $\gamma > 0$, initial vector x_0.

1: **for** $t = 0, 1, \ldots$ **do**
2: Compute a stochastic compression of iterate x_t as $\mathcal{C}(x_t)$.
3: Take a gradient descent step from the compressed iterate

$$x_{t+1} = \mathcal{C}\left(x_t\right) - \gamma \nabla f\left(\mathcal{C}\left(x_t\right)\right).$$

4: **end for**

Clearly, if we are to tackle iterative model compression as used in more complex distributed optimization settings we must understand it when used on a single node as in Algorithm 1.

Assumptions

Smoothness and Convexity: The function f is L-smooth and μ-strongly convex. That is, there exists $L \geqslant \mu > 0$ such that for all $x, y \in \mathbb{R}^d$ we have:

$$f(y) + \langle \nabla f(y), x - y\rangle + \frac{\mu}{2}\|x - y\|^2 \leqslant f(x),$$

$$f(x) \leqslant f(y) + \langle \nabla f(y), x - y\rangle + \frac{L}{2}\|x - y\|^2.$$

We define the condition number of f as $\kappa \stackrel{\text{def}}{=} L/\mu$.

Convergence of Gradient Descent with Compressed Iterates (GDCI)

Theorem 1

For GDCI run with a constant stepsize $\gamma > 0$ such that $\gamma \leqslant \frac{1}{2L}$ and a compression coefficient $\omega \geqslant 0$ that satisfies

$$\frac{4\omega}{\mu} \leqslant \frac{1 - 2\gamma L}{2\gamma L^2 + \frac{2}{\gamma} + L - \mu}.$$

Then,

$$\mathbb{E}\left[\left\|x_k - x_*\right\|^2\right] \leqslant (1 - \gamma\mu)^k \left\|x_0 - x_*\right\|^2 + \frac{2\omega}{\mu}\left(4\gamma L^2 + \frac{4}{\gamma} + L - \mu\right)\|x_*\|^2.$$

Using a specific stepsize choice, we can gain more insight into the convergence rate given by Theorem 1:

Corollary 1

Choose $\gamma = \frac{1}{4L}$ and suppose that $\omega \leqslant \frac{1}{73\kappa}$, then we have,

$$\mathbb{E}\left[\left\|x_k - x_*\right\|^2\right] \leqslant \left(1 - \frac{1}{4\kappa}\right)^k \left\|x_0 - x_*\right\|^2 + 2\omega\left(18\kappa - 1\right)\left\|x_*\right\|^2.$$

This is the same rate as gradient descent, but only to a $\mathcal{O}(\kappa\omega)$ neighbourhood (in squared distances) of the solution.

More on the convergence of GDCI

While the analysis shows that GDCI only converges to a neighbourhood, in many cases when only an approximate solution is desired this is acceptable. However, if we desire to set the neighbourhood to be $\mathcal{O}(1)$, then we should have $\omega = \mathcal{O}\left(\kappa^{-1}\right)$. While this seems to be a pessimistic bound on the compression level possible, we note that in practice compression is done only intermittently (this could be modelled by an appropriate choice of $\mathcal{C}$) or combined with averaging (which naturally reduces the variance associated with quantization).

In practical situations where averaging is not performed, such as the quantization of server-to-client communication, high compression levels do not seem possible without serious deterioration of the accuracy of the solution [2], and our experiments also suggest that this is the case.

Experimental Results

We experiment with a logistic regression problem on two different datasets. We consider the random sparsification operator, where each coordinate is independently set to zero according to some given probability. To model intermittent quantization experimentally, we apply the quantization operator C with probability 1/10 and keep the iterate as it is with probability 9/10.

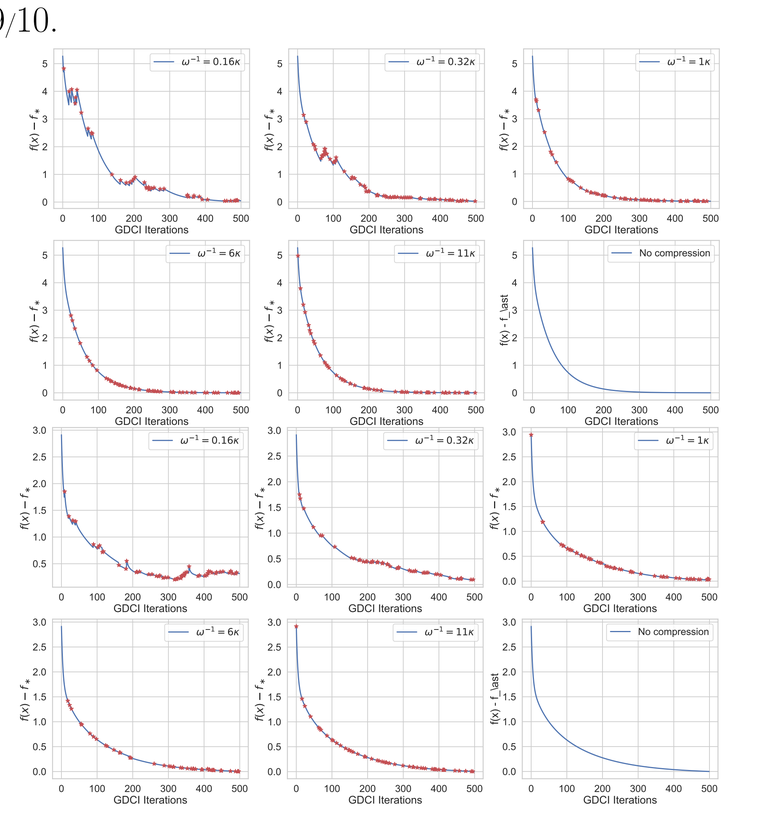

Figure 2:GDCI as ω varies for two different regularized logistic regression problems. Red star indicates $\mathcal{C}$ was applied in that iteration.

References

[1] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated Learning: Strategies for Improving Communication Efficiency. In *NIPS Private Multi-Party Machine Learning Workshop*, 2016.

[2] Sebastian Caldas, Jakub Konečný, H. Brendan McMahan, and Ameet Talwalkar. Expanding the Reach of Federated Learning by Reducing Client Resource Requirements. *arXiv:1812.07210*, 2018.