Adaptive Vector Quantized Stochastic Gradient Descent

Samarth Kadaba, Yusef Qazi

Stanford University

Abstract

The recent surge in the volumes of trainable data for learning parametric models has motivated interest in large-scale distributed algorithms (Faghri et al., 2020). "Federated Learning" settings are primarily bottle-necked by the communication costs of sharing locally computed gradients between multiple workers resulting in time-intensive processes. To alleviate this, numerous quantization schemes have been developed for the post- and intra- training of models (including large neural networks). In general, the quantization problem, seeks an optimal mapping of continuous, floating-point gradients to discretized or compressed representations (i.e. in the number of required bits). We formulate this map as the solution to an optimization problem which seeks to minimize the relative error between quantized and full-precision gradients while achieving maximal compression.

Vector Quantization

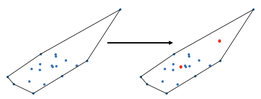

Figure 1. The convex hull of a set of gradients which are local minimizers in a ridge regression setting. At each quantization step, our quantized gradient is given by a function of discrete points sampled from this set (shown in red).

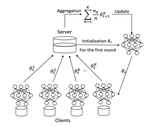

Figure 2. Typical picture of distributed optimization/learning settings. Local nodes compute gradients w.r.t local data and broadcast them to a central server for parameter updates.

$$\min_{C} \mathbb{E} \left\| Q_C(g) - g \right\|^2 \tag{1}$$

Vector quantization is given by solution to (1). We seek a finite point set C such that a quantization function Q_C yields an unbiased estimator of gradient $g \in \mathbb{R}^d$. Prior work has detailed randomized point sets following Gaussian and other deterministic constructions (Gandkitoa et al., 2022). We propose an adaptive scheme which defines a point set from horizon R of observed gradients.

Problem Data

Distributed Least Squares with Regularization

$$\min_{x} \sum_{i=1}^{m} \left\| A_i x_i - b_i \right\|^2 + \lambda \left\| x_i \right\|^2 \tag{2}$$

We first consider the problem of minimizing (2). With $A_i \in \mathbb{R}^{m_i \times n}$, $b_i \in \mathbb{R}^{m_i}$ and n agents. Note $\sum_i m_i = m$. We evenly split up our large, random, Gaussian-distributed data matrices A and b amongst n agents, solving with local data at each iteration.

2-layer Neural Network

$$\min_{W_1, W_2} \sum_{i=1}^{n} \left\| W_2^T \mathcal{A}(W_1^T X_i) - Y_i \right\|^2 \tag{3}$$

Next, we consider the problem of minimizing (3), which is a neural network with a non-linear activation. We have $X_i \in \mathbb{R}^p$, $W_1 \in \mathbb{R}^{p \times r}$, $W_2 \in \mathbb{R}^{r \times s}$, and $Y_i \in \mathbb{R}^s$. We define an activation function $\mathcal{A} : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$. Namely, we experiment $\mathcal{A}(M) = ReLU(M)$, the rectified linear unit, and $\mathcal{A}(M) = \sigma(M)$, the sigmoid function. We define X to contain m samples and evenly split up our large, random, Gaussian-distributed data matrices X and Y amongst n agents, solving with local data at each iteration.

Point Set Constructions

Convex Combinations of a Representative Points

$$\begin{aligned} \min_{C, a_i} \ & \sum_{i=1}^{R} \lambda_i \left\| g_i - \hat{g}_i \right\|^2 \\ \text{s.t.} \ & \hat{g}_i = \textstyle\sum_j a_{i,j} c_j, a_i \succeq 0, \sum_j a_{i,j} = 1 \end{aligned} \tag{4}$$

$$\begin{aligned} \min_{a_i} \ & \left\| g_i - \hat{g}_i \right\|^2 \\ \text{s.t.} \ & \hat{g}_i = \textstyle\sum_{i=1}^{M} a_i c_i, a_i \succeq 0, \sum_j a_{i,j} = 1 \end{aligned} \tag{5}$$

Consider a set of M points $C \in \mathbb{R}^{d \times m}$ defined such that quantized gradients are a convex combination of the m points. While (4) is non-convex, we quasi-linearize the function w.r.t. $a_{i,j}$ and c_j. We solve (4) with projected sub-gradient descent. Gandikota et al., 2022 proposes observing the coefficients of such a combination as a distribution from which we sample points on C. Here, the quantization scheme becomes $Q_c(g) = c_i$ with probability a_i where a_i are given by (5), which can be easily solved via efficient projection onto the simplex.

Optimally Discretized Lowner-John Ellipsoids

$$\begin{aligned} \min_{A, b} \ & \log\det(A^{-1}) \\ \text{s.t.} \ & \left\| A g_i - b \right\| \leq 1, \quad i = 1, ..., R \end{aligned} \tag{6}$$

We consider sampling from a Lowner-John ellipsoid of R observed gradients (6). We discretize the resulting ellipsoid to M points using (7) which is non-convex in general but can be given by an SDP relaxation (8) whose last inequality can be reduced to (9) (Boyd et al., 1994).

$$\begin{aligned} \max \ & t \\ \text{s.t.} \ & \left\| c_i - c_j \right\| \geq t, \quad \forall\, i \neq j, \\ & \left\| A c_i - b \right\| \leq 1 \quad \forall\, i \end{aligned} \tag{7}$$

$$\begin{aligned} \min_{X, Y} \ & t + \textstyle\sum_{i=1}^{R} \left\| x_i - p_i \right\| \\ \text{s.t.} \ & l - t \leq e_{i,j}^T Y e_{i,j} \leq u + t, \quad \forall\, i \neq j, \\ & \left\| A x_i - b \right\| \leq 1 \quad \forall\, i, Y \succeq X^T X \end{aligned} \tag{8}$$

$$\begin{bmatrix} I & X \\ X^T & Y \end{bmatrix} \succeq 0 \tag{9}$$

Note in (8), l and u given lower and upper bounds, respectively, on the distance between "bin" points. X is a matrix where each column denoted by x_i are the representative points we seek. e_{ij} is a vector in $\mathbb{R}^d$ with 1 in the ith index and -1 in the jth index. To prevent the collapse of points x_i in X (i.e. converging to the same point), we introduce a euclidean distance term p_i which either pushes optimal discretized points to uniformly distributed points on the ellipse given by (6) (Muller, 1959) or the most recently observed gradients g_i.

Measuring Communication Cost

We measure the ratio of the number of floating-point values required for communication with quantized vs full-precision gradients where full-precision gradients require d "bits" and quantized gradients require $R + l \times d \times m$. l is the number of times needed to recompute C in a given solution instance.

Method	M = 5	M = 20	Fr = 10	Fr = 100
Simplex (DRR)	0.25	0.85	1.24	0.33
Ellipsoid (DRR)	0.12	0.25	2.03	0.30
Simplex (NN)	0.47	1.20	1.28	0.60

Table 1. Number of floating point values required for solution instances with varying parameters. These values are proxies for bit communication requirements with each value ≈ 64 bits.

1. **Marginal benefit for smaller problems**. We observe that in general, quantization efficiency is better for larger examples and decreases with dimensionality and frequency of computing C.
2. **Ellipsoid method typically achieves better compression**. Because sending a single index corresponding to points sampled from an ellipsoid is cheaper than a set of coefficients, we observe better compression.
3. **Neural networks benefit most from quantization**. Due to the large weight matrices, there is significant compression in sending gradients. Communication savings are maximized when limiting M and maximizing Fr.

Distributed Ridge Regression

Distributed ridge regression with problem data $A \in \mathbb{R}^{10000 \times 50}$ and $b \in \mathbb{R}^{50}$. Problem data is randomly generated and an optimal value is calculated using the Moore-Penrose pseudo-inverse of A. Our method beats 2-bit quantization schemes and come lose to optimality gaps achieved by full-precision gradients.

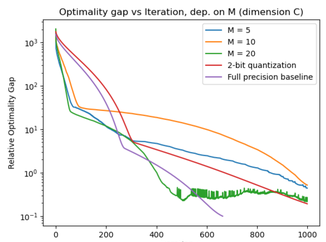

Figure 3. With the simplex formulation, increasing dimensionality of point set C computed from (4) and (5) shows better convergence with a value of Fr = 20 converging to optimality gap 10^{-1}. In general, as solutions progress, gradient directions become noisier.

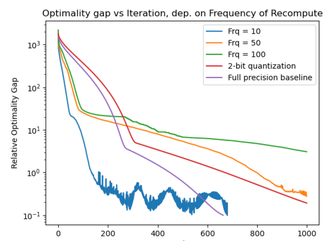

Figure 4. With same problem data as shown in Figure 3, we observe that updating the the point set C to infrequently results in sub-optimality. In this case, updating C every Fr = 10 iterations yielded the best convergence although decreases quantization efficiency.

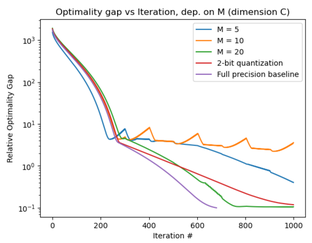

Figure 5. With the ellipsoid formulation, increasing dimensionality of point set C computed from (6-8) shows better convergence with a value of Fr = 20 converging to optimality gap 10^{-1}. In general, as solutions progress, gradient directions become noisier.

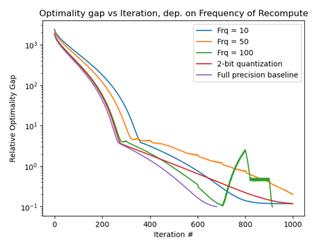

Figure 6. With same problem data as shown in Figure 5. We observe little impact of the frequency of updating point set C on convergence using ellipsoids. This suggests that quantization limits are greater with ellipsoidal approximations.

Note, in our implementations we return full-precision gradients on steps where previous iterations resulted in increasing objective values. We accounted for this when computing total communication costs.

Distributed Two-Layer Neural Networks

Two-layer neural network with problem data $X \in \mathbb{R}^{10000 \times 2}$ and $y \in \mathbb{R}^{10000 \times 4}$. X is randomly generated and Y is generated according to random proxy weights so we ensure there exists some weights $W_1 \in \mathbb{R}^{2 \times 8}$ and $W_2 \in \mathbb{R}^{8 \times 4}$ that solves the neural network.

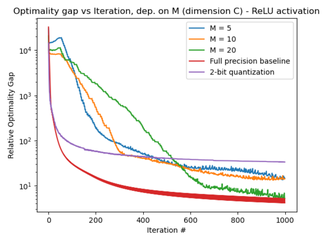

Figure 7. Compression ratio as much as ∼0.7 was achieved while still beating 2-bit quantization and approaching full-precision accuracy. More optimal results were found with varying Fr and M together.

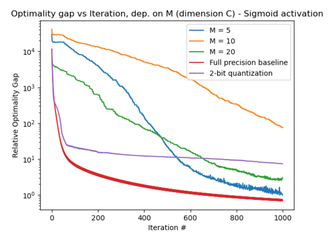

Figure 8. With Sigmoid (a non-convex activation), our method further outperforms 2-bit quantization and approaches full-precision accuracy, all while achieving a compression ratio of ∼0.5.

Subproblem optimality and convergence

We now analyze sub-problem convergence to illustrate scalability of methods presented herein. Figure 9 gives a Lowner-John ellipsoid of a set of observed gradients and optimally chosen bins, which solve (8). Although not shown here, it is interesting to note that computation of spanning sets C asymptotically increases per iteration for neural networks. This un-intuitively suggests an increasing margin of quantization error.

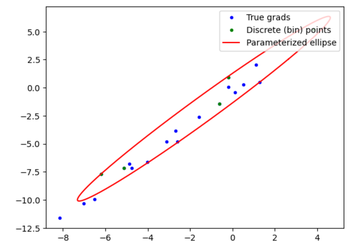

Figure 9. Bins (in green) are chosen to maximize pairwise distance with regularization terms to drive points towards boundaries. Although not plotted here, as we progress in SGD, ellipsoids of a horizon of gradients shrink reflecting lower margin of error.

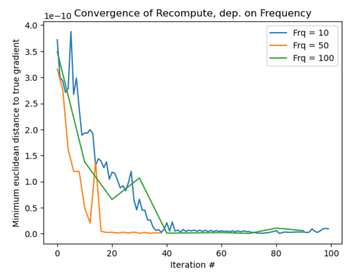

Figure 10. Intermediate recomputation frequency of C yields optimal quantization error when sampling from Lowner-John gradient ellipsoids.

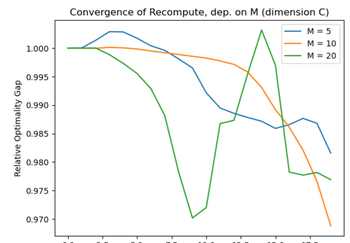

Figure 11. Increasing values of the dimensionality (M) of the set C results in decreasing optimality in computing such a C which is a "spanning set" of R observed gradients. intermediate values of R seem to result in the most stable computation.

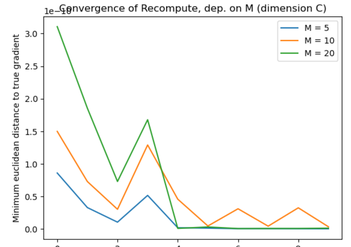

Figure 12. Compared with the simplex method of computing C, finding a Lowner-John ellipsoid and then discretizing results in much better performance. Optimality gap is near 0 meaning all gradients are captured by the optimal ellipsoid independent of dimensionality.

Contributions and Future Work

Here we explore applications of convex optimization in quantizing gradients for distributed optimization and nonlinear, non-differentiable neural network problems. We show convergence to acceptable optimality gaps for all methods while achieving positive compression.

Methods for constructing point sets and considerations

1. In general, defining a point set from a horizon of previously observed gradients provides a reasonable approximation for future gradients in both convex (distributed least squares) and non-convex (neural network) settings.
2. Minimum volume ellipsoids are efficient sets from which we can sample gradients effectively while reducing computational overhead compared with optimizing directly for euclidean distance.
3. Adaptive schemes which update point sets are necessary for non-stagnant progress towards optimal.

Communication Efficiency and Quantization

1. We trade communication cost for efficiency of convergence and final optimality gap when defining the dimensionality of point sets from which we sample gradients.
2. A marginal communication benefit is observed in smaller problem instances. Only over a large number of iterations do we achieve meaningful compression.

Future Work

Validation with synchronization and multiprocessing. To validate communication speedup from reduced bit transmission, we plan to experiment with synchronization and multiprocessing/multi-threading modules to observe compute time in a variety of problem instances.

Efficient update of point sets. To reduce computational time associated with computing a point set C, we explore foundations for efficient updates of polyhedral convex sets and Lowner-John ellipsoids after observing a new gradient g_i. These efficient updates allow for expanding horizons (i.e. increasing R).

Comparison with simpler heuristics. Convex optimization formulations of finding point set C allows for certificates of global optimality. However, simpler point set constructs, such as based on Singular Value Decomposition (SVD) or Principal Component Analysis (PCA), may yield as good or better results.

References

[1] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, 1994.

[2] Fartash Faghri, Iman Tabrizian, Ilia Markov, Dan Alistarh, Daniel Roy, and Ali Ramezani-Kebrya. Adaptive gradient quantization for data-parallel sgd. 2020.

[3] Venkata Gandikota, Daniel Kane, Raj Kumar Maity, and Arya Mazumdar. vqsgd: Vector quantized stochastic gradient descent. 2020.

[4] Chung-Yi Lin, Victoria Kostina, and Babak Hassibi. Differentially quantized gradient methods. volume 68, pages 6078–6097, 2022.

[5] Mervin E. Muller. A note on a method for generating points uniformly on n-dimensional spheres. *Commun. ACM*, 2:19–20, 1959.

[6] Guangfeng Yan, Shao-Lun Huang, Tian Lan, and Linqi Song. Dq-sgd: Dynamic quantization in sgd for communication-efficient distributed learning. 2021.