COMMUNICATION EFFICIENT PRIMAL-DUAL ALGORITHM FOR NONCONVEX NONSMOOTH DISTRIBUTED OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Decentralized optimization problems frequently appear in the large scale machine learning problems. However, few works work on the difficult nonconvex nonsmooth case. In this paper, we propose a decentralized primal-dual algorithm to solve this type of problem in a decentralized manner and the proposed algorithm can achieve an $\mathcal{O}(1/\epsilon^2)$ iteration complexity to attain an $\epsilon-$solution, which is the well-known lower iteration complexity bound for nonconvex optimization. To our knowledge, it is the first algorithm achieving this rate under a nonconvex, nonsmooth decentralized setting. Furthermore, to reduce communication overhead, we also modifying our algorithm by compressing the vectors exchanged between agents. The iteration complexity of the algorithm with compression is still $\mathcal{O}(1/\epsilon^2)$. Besides, we apply the proposed algorithm to solve nonconvex linear regression problem and train deep learning model, both of which demonstrate the efficiency and efficacy of the proposed algorithm.

1 INTRODUCTION

Decentralized machine learning problems have attracted many attentions as the growing size of the data. Usually, the problem will be solved on a network. We define represent the network by an undirected graph $G = (V, E)$, where V is a node set and E is the set of edges. We denote $|V| = N$ and $|E| = M$. Then, the problem is formulated as follows:

$$\begin{aligned} \min_x f(x) &= \frac{1}{N}\sum_{i=1}^{N} f_i(x_i) \\ s.t.\ x_i &= x_j,\ \forall (i,j) \in E \end{aligned}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is a non-convex but smooth function, $x_i \in \mathbb{R}^n$ and $x \in \mathbb{R}^{Nn} = (x_1^T, x_2^T, \cdots, x_N^T)^T$. Moreover, to embed some prior knowledge or tackle over-fitting problem, some regularization terms will come into the formulation. Besides, when considering deep neural network compression problem(He et al., 2017), some nonsmooth term will also be formulated in the objective function. Then, the problem becomes:

$$\begin{aligned} \min_{x_1,\cdots,x_N} \frac{1}{N}\sum_{i=1}^{N} f_i(x_i) + h_i(x_i) \\ s.t.\ x_i = x_j,\ \forall (i,j) \in E. \end{aligned} \tag{1}$$

where h_i can be a non-smooth function(e.g. L1 regularization or indicator function of some constraint on x), and here we assume h_i is a convex function. In general, we can use different regularization in the different nodes, however, because of the consensus constraint $x_i = x_j$, we can combine all the h_i to the first node, then without loss of generality we use $h_1(x) = h(x)$ and the rest of h_i are zeros.

Because of intolerable computation complexity for higher-order algorithms, first-order methods are popular for solving large-scale problems. In general, to solve problem 1, there are two different types of first-order methods. One is the gradient descent based method, where each node performs some gradient steps and then average x_i with its neighbors. Some works change the gradient steps by

adding momentum (Yu et al., 2019) or adaptive learning rate (Nazari et al., 2019) and some works change the average step other average schemes of x_i (e.g. weighted average (Tang et al., 2019)). The other kind of method is the primal-dual based method, where dual variables(y) are introduced into the algorithm and using primal-dual type methods to solve the saddle points of the resulting Lagrangian function. The primal-dual methods consist of two parts, the primal update, and the dual update. The primal step is to minimize the Lagrangian function by a local update of primal variables and the dual step is to perform a dual ascent by using the consensus residual. The primal-dual method is usually more efficient than the gradient-based method, which is well-studied in convex cases (Chang et al., 2014) and nonconvex smooth cases (Hong et al., 2017). Also in Hong & Luo (2017), they prove that the ADMM algorithm, which is a primal-dual method, converges for consensus problem with a nonsmooth term in the centralized setting, where a central node controls the consensus, and achieve the $\mathcal{O}(1/\epsilon^2)$ iteration complexity. However, it is still unknown whether we can use the primal-dual method to solve problem 1 with the decentralized setting. More importantly, to our knowledge it is still unclear whether there exists a decentralized first-order optimization algorithm solving problem 1 that can achieve the $\mathcal{O}(1/\epsilon^2)$ iteration complexity, which is the well-known lower bound for the iteration complexity for solving nonconvex optimization problems using first-order method (Carmon et al., 2019).

Furthermore, the success of large models such as deep neural networks, communication among nodes becomes an important factor influencing the speed of the optimization algorithms. A popular strategy to reduce communication complexity is to compress the vectors exchanged by neighbor nodes. Many compression functions are used in these scenarios such as quantization functions and sparsification functions. Therefore, another important problem is whether we can design a communication efficient algorithm for solving problem 1 with low communication complexity.

In this paper, we give affirmative answers to both of the above two problems.

Concretely speaking, we proposed a smoothed proximal-primal dual algorithm for solving problem 1 with a nonconvex nonsmooth setting. The algorithm can achieve an ϵ-solution of problem 1 within $\mathcal{O}(1/\epsilon^2)$ iteration complexity, which, to our knowledge, is the first algorithm achieving the lower iteration complexity bound for nonconvex nonsmooth optimization. Furthermore, to reduce the communication cost, we use a compressor when nodes communicate with each other. We prove that our algorithm with compression of information exchanged between neighbor nodes can also achieve an $\mathcal{O}(1/\epsilon^2)$ iteration complexity and an $\mathcal{O}(1/\epsilon^2)$ communication complexity if the compression is in sufficient high accuracy.

2 Related Work

Distributed optimization methods have been studied for many years. For convex cases, many algorithms are solving distributed optimization problems, including the distributed subgradient method (Nedic & Ozdaglar, 2009), consensus ADMM methods (Chang et al., 2014; Shi et al., 2014). Recently, nonconvex distributed optimization problems have attracted more attention.

Yuan et al. (2016) extends the DGD algorithm to the nonconvex smooth case and the iteration complexity is $\mathcal{O}(1/\epsilon^4)$. Lian et al. (2017) gives the convergence analysis of a gradient-based algorithm under the nonconvex but smooth setting and the iteration complexity is $\mathcal{O}(1/\epsilon^3)$. Di Lorenzo & Scutari (2016) proposes a gradient-based algorithm for solving problems with the nonconvex nonsmooth setting, they show the result will converge in the infinite time, but they don't give the iteration complexity of the algorithm.

On the other hand, it is well-known in convex cases, primal-dual algorithms are usually more efficient than gradient-based methods(Lan et al., 2020; Scaman et al., 2018). Though primal-dual methods are well-studied for convex problems, the convergence of primal-dual algorithms for nonconvex cases is still unclear in many problems. Recently, some papers analyze the primal-dual algorithms for nonconvex cases. Hong et al. (2017) give the primal-dual algorithm and show the convergence under the nonconvex setting with the optimal order. For nonconvex, nonsmooth settings, Hong & Luo (2017) analyze the ADMM algorithm for a special type of graph with one a center point. They prove that the iteration complexity is also $\mathcal{O}(1/\epsilon^2)$. To our knowledge, there is no paper in the literature achieving the optimal iteration complexity $\mathcal{O}(1/\epsilon^2)$ iteration complexity for nonconvex, nonsmooth decentralized optimization.

Later, to reduce the overhead of communication, Tang et al. (2018) introduce the compression functions into decentralized gradient descent, and still give the iteration complexities at order $\mathcal{O}(\epsilon^{-3})$, but the algorithm only works for carefully designed compressors. Tang et al. (2019) and Koloskova et al. (2019) give the algorithms working for more general compression functions. But still the iteration complexity is $\mathcal{O}(\epsilon^{-3})$. In the primal-dual setting, to our best knowledge, there is not any research that gives the algorithm or convergence results.

Different from previous work, we propose a primal-dual method that can achieve the optimal rate under the nonconvex nonsmooth setting. Besides, by adding the compression function in the communication, we reduce the communication overhead and show it will not hurt the convergence speed.

3 ALGORITHM

To solve the problem 1, we first reformulate it into a linearly constrained problem with equality constraints.

We define a matrix $W \in \mathbb{R}^{M\times N}$. The k_{th} row of W represents the k_{th} edge of graph. If (i, j) is the k_{th} edge in the graph, we set $W_{k,i} = 1$, $W_{k,j} = -1$, and the rest entries in the k_{th} row are zeros. Then, we rewrite the constraint as $Ax = 0$, where $A = W \otimes I_n$. The problem 1 will become

$$\min_x f(x) + h(x)$$
$$s.t. Ax = 0.$$

Then, the Lagrange function of the problem is defined as follows:

$$L(x, y) = f(x) + h(x) + y^T Ax.$$

We need to solve the saddle points of this Lagrange function. To solve the Lagrange function, people usually add (variant) augmented terms to give more convexity. However, the inclusion of augmented terms in the Lagrange function still can not guarantee the theoretical convergence of algorithms for nonconvex-nonsmooth problems. Moreover, the augmented terms in the Lagrange function will result in more communication costs when exchanging information. Inspired by Zhang & Luo (2020), we use a proximal framework instead. In any iteration, we include a proximal term to the Lagrange function centered at an auxiliary sequence, which is an exponentially weighted sequence z^t of the primal iterates. Then we consider the following strongly convex counterpart of problem 1:

$$\min_{x,z} \max_y K(x, y, z) + h(x), \tag{2}$$

where $K(x, y, z) = f(x) + y^T Ax + \frac{p}{2}\|x - z\|^2$.

The Algorithm 1 shows the steps to solve the problem 2. In each iteration, first, we update x^t by proximal gradient method. Then, we communicate the new x^{t+1} with the neighbors. After that, we update y by gradient ascent with step size α and update z by gradient descent with step size β. By calculating $\nabla_x K(x, y, z) = \nabla f(x) + A^T y + p(x - z)$, it can be shown that in the iteration of updating x only $A^T y$ are needed. So we do not need to store y directly, we store $\mu = A^T y$ instead. On the other hand variable y is in $\mathbb{R}^{Mn}$, which is not easy to divide into N blocks, but μ is in $\mathbb{R}^{Nn}$. Thus, we use μ, instead of y. As all x, μ, z can be divided into N blocks, and we give the algorithm.

Algorithm 1 Distributed Primal-Dual Algorithm

Select $c > 0$, $\alpha > 0$, $0 < \beta \leq 1$, and $p \geq 0$;
Initialize x_i^0, $\hat{x}_i^0 = x_i^0$, μ_i^0 and z_i^0;
for $t = 0, 1, 2, \cdots, T$ **do**
  $x_i^{t+1} = \arg\min_{x_i} \left(\langle \nabla_{x_i} K\left(x^t, y^t, z^t\right), x_i - x_i^t\rangle + h_i\left(x_i\right) + \frac{1}{2c}\|x_i - x_i^t\|^2\right)$;
  Send x_i^{t+1} to $N\left(i\right)$ and receive x_j^{t+1} from $j \in N\left(i\right)$;
  $\mu_i^{t+1} = \mu_i^t + \alpha\left(d_i x_i^{t+1} - \sum_{j\in N(i)} x_j^{t+1}\right)$;
  $z_i^{t+1} = z_i^t + \beta\left(x_i^{t+1} - z_i^t\right)$;
end for

To reduce the communication overhead, we add a compression function at each iteration when communicating with its neighbors. Then we give the final algorithm in Algorithm 2.

Algorithm 2 Communication Efficient Distributed Primal-Dual Algorithm

Select $c > 0$, $\alpha > 0$, $0 < \beta \leq 1$, $p \geq 0$ and compression function $Q(\cdot)$;
Initialize x_i^0, $\hat{x}_i^0 = x_i^0$, μ_i^0 and z_i^0;
for $t = 0, 1, 2, \cdots, T$ **do**
  $x_i^{t+1} = \arg\min_{x_i} \left(\langle \nabla_{x_i} K\left(x^t, y^t, z^t\right), x_i - x_i^t \rangle + h_i\left(x_i\right) + \frac{1}{2c} \|x_i - x_i^t\|^2 \right)$;
  Send $Q\left(x_i^{t+1} - \hat{x}_i^t\right)$ to $N(i)$ and receive $Q\left(x_j^{t+1} - \hat{x}_j^t\right)$ from $j \in N(i)$;
  $\hat{x}_i^{t+1} = \hat{x}_i^t - Q\left(x_i^{t+1} - \hat{x}_i^t\right)$;
  $\hat{x}_j^{t+1} = \hat{x}_i^t - Q\left(x_j^{t+1} - \hat{x}_j^t\right)$, for $j \in N(i)$;
  $\mu_i^{t+1} = \mu_i^t + \alpha \left(d_i \hat{x}_i^{t+1} - \sum_{j \in N(i)} \hat{x}_j^{t+1} \right)$;
  $z_i^{t+1} = z_i^t + \beta \left(x_i^{t+1} - z_i^t \right)$;
end for

4 Theoretical Analysis

In this section, we present the convergence result of the Algorithm 2.

4.1 The stationary solution of problem 1

First, we give the definition of stationary point and the approximate stationary points of problem 1.

We say that x is a (first-order) stationary point of problem 1 if there exists a y such that

$$0 \in \partial h(x) + \nabla f(x) + A^T y$$
$$Ax = 0$$

We then define the ϵ-stationary point as follows:

Definition 1. *(x, y) is an ϵ-stationary point if $\frac{1}{\sqrt{N}}\|Ax\| \leq \epsilon$ and there exists ν, such that $\nu \in \nabla f(x) + \partial h(x) + A^T y$ and $\sqrt{N}\|\nu\| \leq \epsilon$.*

Remark 1. *Definition 1 is a sufficient condition for the ϵ-solution in Hong et al. (2017); Tang et al. (2019).*

4.2 Assumptions

Next, we state our assumptions used in the theoretical analysis. We first give some assumptions for the function f and the regularization function h.

Assumption 1. *For the function f and h, we assume:*

1. *There exists $\underline{f}$, $f(x) + h(x) \geq \underline{f}$ holds for all $x \in \mathbb{R}^{Nn}$.*
2. *Function f_i is a differential function with Lipschitz continuous gradient, i.e., for all i, we have*
$$\|\nabla f_i(x) - \nabla f_i(x')\| \leq L\|x - x'\|, \ \forall x, x' \in \mathbb{R}^n.$$
3. *h_i is a convex function.*

We then give the assumption for the compression function.

Assumption 2. *Compression function $Q(\cdot)$ satisfies the following inequality:*

$$\|Q(x) - x\| \leq (1 - \delta)\|x\| \text{ for some } \delta > 0, \text{ and } \forall x \in \mathbb{R}^n$$

For the communication network, We also need to assume that it is connected so that the information can be sent through different nodes.

Assumption 3. *The graph G is connected.*

These assumptions are standard in the literature.

4.3 THE MAIN THEORETICAL RESULT

Under the above assumptions, we have the following main theoretical result:

Theorem 1. *Suppose the parameters $c \leq \frac{1}{L_K}$, $p > -L/N$, and $\alpha,\ \beta$ are sufficiently small. Then it holds that for any $T > 0$, there exists $s \in \{0, 1, \cdots, T-1\}$ such that $\left(x^{s+1}, y^s\right)$ is a $B/\sqrt{T}$-solution, where $B = \mathcal{O}\left(L\sqrt{\left(\Phi\left(x^0, y^0, z^0\right) - \underline{f}\right) max\{4c, \frac{8}{\alpha}, \frac{4}{p\beta}\}}\right)$, where $\frac{1}{L_K}$ is the Lipschitz constant of K.*

Remark 2. *The values of α and β are related to N, p, δ, c, L and connectivity of the graph. The upper bound of α and β can be calculated by some inequalities which are defined in the proof of the theorem in the Appendix.*

Remark 3. *Using the result in the Theorem 1, to achieve ϵ-stationary point, $O(\frac{1}{\epsilon^2})$ iterations are needed, which is the well-known lower bound of iteration complexity for the nonconvex case.*

Corollary 1. *Suppose we choose the parameters $c = \frac{N}{3L}$, $p = 2L/N$, $\alpha = \frac{L\delta^2}{24\lambda_1(5\delta^2+9/4(1-\delta)^2))}$ and $\beta = \frac{\alpha}{17\alpha+4L(2+\sqrt{3})\kappa}$, where κ is a constant related to the connection of the graph. For any $T > 0$, there exists $s \in \{0, 1, \cdots, T-1\}$, such that (x^{s+1}, y^s) is a $B_1/\sqrt{T}$-solution, where $B_1 = \mathcal{O}\left(\sqrt{\left(\Phi\left(x^0, y^0, z^0\right) - \underline{f}\right)}\left(\sqrt{L} + \frac{1}{\sqrt{L}}\right)\sqrt{\frac{\gamma((1-\delta)^2+1)}{\delta^2}}\right)$, where γ is a constant related to connection of graph, .*

Remark 4. *For the smooth case where $h(\cdot) = 0$, γ is just the spectral gap of the Laplacian matrix defined as the division of the largest eigenvalue with the smallest non-zero eigenvalue. For nonsmooth case where $h(\cdot) \neq 0$, γ will also depend on the node corresponding to x_1. In the following table, we calculate the γ for some graph in smooth case and nonsmooth case respectively as follows:*

Graph Structure	Smooth	Nonsmooth
Ring(10 nodes)	3.9021	10.4721
Grid(3×3)	6	33.9973
Complete(10 nodes)	1	10
Star(10 nodes)	10	10

Table 1: Different γ under different settings.

4.4 PROOF SKETCH OF THEOREM 1

To prove Theorem 1, we will give some important lemmas and the full proof can be found in the Appendix. The idea of the proof is to construct a "proximal-primal-dual " potential function, which is bounded below and decreases along the iteration sequence and uses it to show the convergence on x^t, Ax^t and z^t. Remember the function $K(x^t, y^t, z^t)$ is defined as:

$$f(x^t) + y^T Ax^t + \frac{p}{2}\|x^t - z^t\|^2,$$

which contains the primal information. We first need to define the dual function and the proximal function. We let

$$\begin{aligned}
d\left(y, z\right) &= \min_{x \in \mathbb{R}^{nN}} K\left(x, y, z\right) + h\left(x\right), \\
x\left(y, z\right) &= \arg\min_{x \in \mathbb{R}^{nN}} K\left(x, y, z\right) + h\left(x\right), \\
M\left(z\right) &= \min_{x \in \mathbb{R}^{nN}, Ax=0} \left(f\left(x\right) + h\left(x\right) + \frac{p}{2}\|x - z\|^2\right), \\
x^*\left(z\right) &= \arg\min_{x \in \mathbb{R}^{nN},\ Ax=0} \left(f\left(x\right) + h\left(x\right) + \frac{p}{2}\|x - z\|^2\right).
\end{aligned}$$

Then we construct the potential function as follows:

$$\Phi(x, y, z) = K\left(x, y, z\right) + h\left(x\right) - 2d\left(y, z\right) + 2M\left(z\right),$$

which is a linear combination of the primal function, dual function and the proximal function. It is not hard to show that the function $\Phi(x, y, z)$ is bounded below by $\underline{f}$. Therefore, in the rest, we hope to show that $\Phi(x^t, y^t, z^t)$ decreases sufficiently after any iteration and hence prove that the optimization residuals can go to zero. We first have the following basic estimate:

Lemma 1. *Under Assumptions above, for any $t > 0$, it always holds that*

$$\begin{aligned}
&\Phi\left(x^t, y^t, z^t\right) - \Phi\left(x^{t+1}, y^{t+1}, z^{t+1}\right) \\
&\quad \geq \frac{1}{2c}\|x^{t+1} - x^t\|^2 - \alpha\left(A\hat{x}^{t+1}\right)^T Ax^{t+1} + \frac{p}{2\beta}\|z^t - z^{t+1}\|^2 + 2\alpha\left(A\hat{x}^{t+1}\right)^T Ax\left(y^t, z^t\right) \\
&\qquad - \frac{\alpha^2\sqrt{\lambda_1}}{\sigma_4}\|A\hat{x}^{t+1}\|^2 + p\left(z^{t+1} - z^t\right)^T\left(z^{t+1} + z^t - 2x\left(y^{t+1}, z^{t+1}\right)\right) \\
&\qquad + 2p\left(z^{t+1} - z^t\right)^T\left(z^t - x^*\left(z^t\right)\right) - p\tilde{L}\|z^t - z^{t+1}\|^2,
\end{aligned}$$

where λ_1 is the largest eigenvalue of $A^T A$, σ_4 is a constant related to p and L, and $\tilde{L}$ is a constant related to the Lipschitz constant of z.

Lemma 1 gives a basic estimate of the change of the potential function during the iteration. According to Lemma 1, we need to bound the negative terms that appear in the basic estimate. The framework of the algorithm is based on the proximal algorithm. Therefore, the negative terms come from the error of estimating $x^*(z^t)$ in any iteration. The error consists of two parts: the dual error and the compression error. The following two lemmas show that we can bound the two error terms by the optimization residuals. The first error bound is a dual bound.

Lemma 2. *For all $y \in \mathbb{R}^{Mn}$ and all $z \in \mathbb{R}^{Nn}$, it holds that*

$$\|x(y, z) - x^*(z)\| \leq \sigma_1\|Ax(y, z)\|,$$

where σ_1 related to the p, L and the graph property on the first node.

The remaining part is to deal with the compression and we bound compression error by the difference on x^t and x^{t+1}, which we give in the following:

Lemma 3. *With the definition of x and $\hat{x}$ in Algorithm 2, the following equality always holds:*

$$\sum_{t=1}^{T}\|x^t - \hat{x}^t\|^2 \leq \frac{(1-\delta)^2}{\delta^2}\sum_{t=1}^{T}\|x^t - x^{t-1}\|^2.$$

Then we can prove the Theorem 1, using the error bound in lemma 2 and lemma 3.

5 Experimental Results

In this section, we will give the experimental results for the algorithm. We compare our algorithm with a gradient descent based algorithm (Koloskova et al., 2019), and Prox_PDA (Hong et al., 2017) on two tasks. First, we implement the algorithms to optimize the nonconvex linear regression objective, and then we implement the algorithm to train a neural network.

5.1 Nonconvex Linear Regression Case

To show the efficacy of our algorithm, we start with a simple function. We use $f_i(x) = \frac{1}{2}\|A_i x - b_i\|^2 + \sum_{j=1}^{n}\frac{x_j^2}{x_j^2+1}$, which is formulated from linear regression with a regularization. In the experiment we randomly generate matrices A_i and vectors b_i. The network structure used in the experiment is a ring with 10 nodes. With 10 times repeat, we give the following results.

In the figure 1, we only optimize with function f, and give the result on the consensus of x_i(i.e. $\sum_{i=1}^{N}\|x_i - \bar{x}\|$) and the norm of gradient of $\sum_{i=1}^{N} f_i(\bar{x})$, where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$. For the full precision version, the primal-dual methods can converge much faster than the gradient descent method, and can converge to more accurate solution. Besides, our algorithm converge a little bit faster than Prox_PDA.

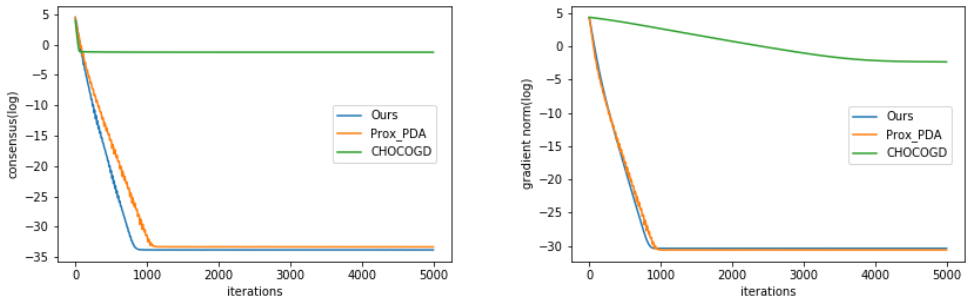

Figure 1: Results on Smooth and Full Precision Settings.

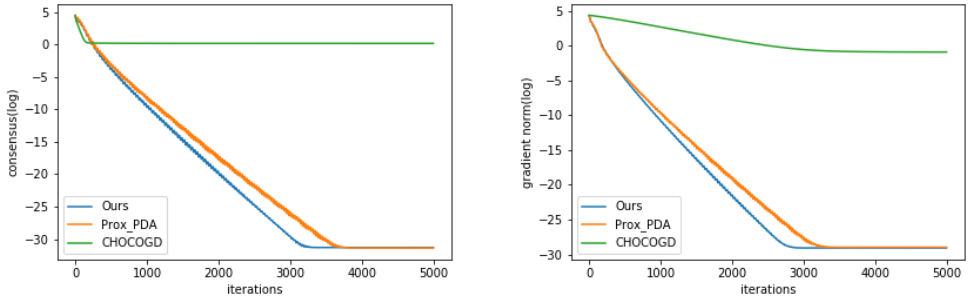

Figure 2: Results on Smooth and Compressed Settings.

On the other hand, when taking consideration of compression, we choose a commonly use compression function top_K sparsification function. We select K as 5 and give the experimental results of different algorithms in Fig. 2. As it is shown in Fig. 2, the primal-dual method can have a better solution than the gradient based method. Besides, as Prox_PDA has the augmented terms which will be affected by the compression, it converges slower than ours.

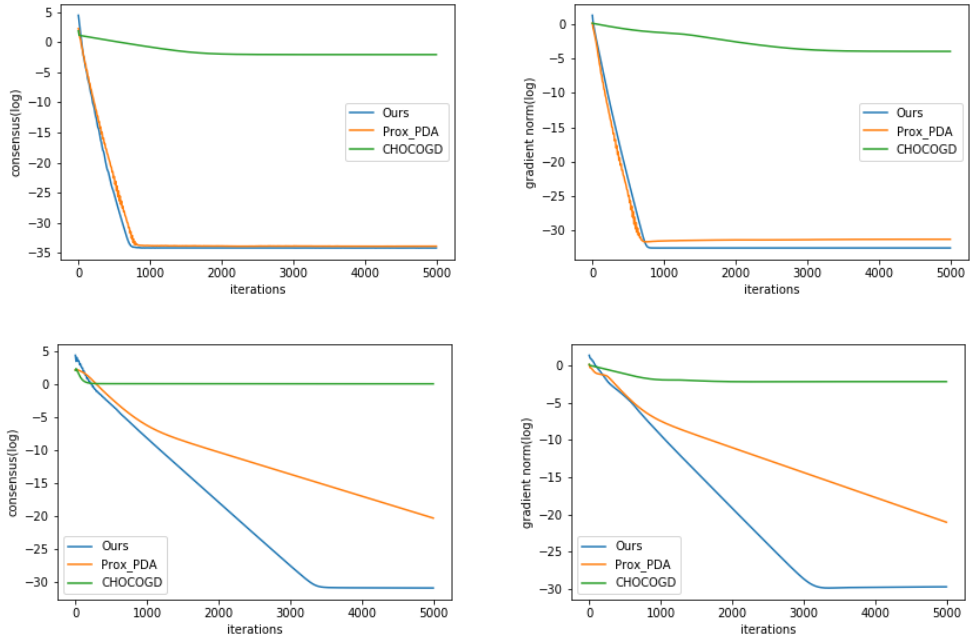

Figure 3: Results on Nonsmooth Settings.

Then, we add non-smooth term into objective, we use $h(x) = \infty\mathbb{I}(\|x\| \geq 1)$. For the gradient-based method and Prox_PDA, we simply extend the method by adding a projection after each update of x. We still calculate the consensus of x. To estimate the solution , we do a gradient descent on $\bar{x}$ with gradient of $\sum_{i=1}^{N} f_i(\bar{x})$ and then project it into unit ball. We denote the result of gradient projection as $\hat{x}$ and use the $\|\bar{x} - \hat{x}\|$ to estimate solution. We show the results for the nonconvex nonsmooth objective in Fig. 3. In the Fig. 3, the upper line is the results with full precision communication, and the bottom line is the results with compressed communication. From the results, it can be seen that although we only use projection in one node, our algorithm performs better than gradient descent and Prox_PDA.

5.2 Neural Network Case

In this section, we will give the result on training ResNet-18(He et al. (2016)) on the dataset CIFAR10(Krizhevsky et al. (2009)). Still, we use the top_K sparsification function as a compression function, and we set K as $10\% \times n$. We use a ring with 10 nodes as the communication network structure. Besides, we use the same learning rate in all algorithms. Then the results are given in Fig. 4 and Fig. 5. Fig. 4 shows the results with full precision communication, it can be seen that because we use the large proximal term, we will converge slower than SGD and Prox_PDA at the beginning iterations, but finally our algorithm becomes faster and can get a better solution than those two methods in 200 epochs. Besides, in Fig. 5, we show the result with compressed communication. Because Prox_PDA fails to converge, we do not draw the curve of Prox_PDA. Although we converge a little bit slower than SGD because of the inexact of dual variables in the initial phase, we can get higher accuracy compared to the SGD.

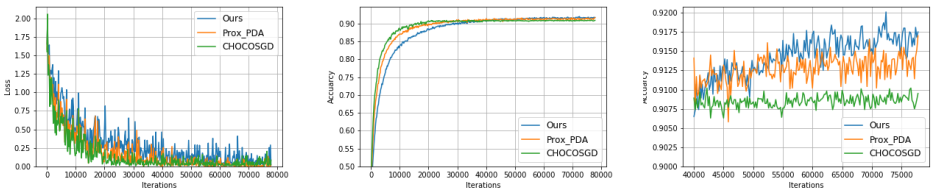

Figure 4: Results on CIFAR10 with Full Precision Communication.

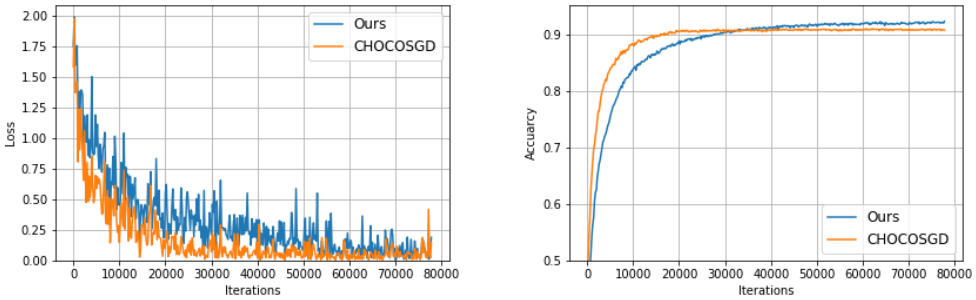

Figure 5: Results on CIFAR10 with Compressed Communication.

6 Conclusion

In this paper, we proposed a primal-dual based algorithm to solve distributed optimization problems. To reduce the communication overhead we use compression function during the communication. We show that under the nonconvex nonsmooth case the algorithm can converge to the ϵ-stationary point with $\mathcal{O}(\frac{1}{\epsilon^2})$ iterations, which is a well-known lower bound for nonconvex optimization. The experimental results on nonconvex linear regression and the deep neural network show the efficacy of the proposed algorithm.

REFERENCES

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, pp. 1–50, 2019.

Tsung-Hui Chang, Mingyi Hong, and Xiangfeng Wang. Multi-agent distributed optimization via inexact consensus admm. *IEEE Transactions on Signal Processing*, 63(2):482–497, 2014.

Paolo Di Lorenzo and Gesualdo Scutari. Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016.

Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1389–1397, 2017.

Mingyi Hong and Zhi-Quan Luo. On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, 162(1-2):165–199, 2017.

Mingyi Hong, Davood Hajinezhad, and Ming-Min Zhao. Prox-pda: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks. In *International Conference on Machine Learning*, pp. 1529–1538, 2017.

Anastasia Koloskova, Sebastian U Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. *arXiv preprint arXiv:1902.00340*, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Guanghui Lan, Soomin Lee, and Yi Zhou. Communication-efficient algorithms for decentralized and stochastic optimization. *Mathematical Programming*, 180(1):237–284, 2020.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pp. 5330–5340, 2017.

Parvin Nazari, Davoud Ataee Tarzanagh, and George Michailidis. Dadam: A consensus-based distributed adaptive gradient method for online optimization. *arXiv preprint arXiv:1901.09109*, 2019.

Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

Kevin Scaman, Francis Bach, Sébastien Bubeck, Laurent Massoulié, and Yin Tat Lee. Optimal algorithms for non-smooth distributed optimization in networks. In *Advances in Neural Information Processing Systems*, pp. 2740–2749, 2018.

Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. On the linear convergence of the admm in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 62(7):1750–1761, 2014.

Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In *Advances in Neural Information Processing Systems*, pp. 7652–7662, 2018.

Hanlin Tang, Xiangru Lian, Shuang Qiu, Lei Yuan, Ce Zhang, Tong Zhang, and Ji Liu. Deepsqueeze : Parallel stochastic gradient descent with double-pass error-compensated compression. *arXiv preprint arXiv:1907.07346*, 2019.

Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. *arXiv preprint arXiv:1905.03817*, 2019.

Kun Yuan, Qing Ling, and Wotao Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

Jiawei Zhang and Zhi-Quan Luo. A proximal alternating direction method of multiplier for linearly constrained nonconvex minimization. *SIAM Journal on Optimization*, 30(3):2272–2302, 2020.