First Order Algorithms for Optimization and Zeroth-order Optimization

Name: Yuchen LOU
University No.: 3035447661
Supervisor: Prof. Xiaoming YUAN

Undergraduate Research Fellowship
Programme (URFP) 2020-21
-Summer Research Internship

Yuchen Lou [1], Hanqin Cai[2], Daniel McKenzie[2] and Wotao Yin[2]

[1]Department of Mathematics, The University of Hong Kong; [2] Department of Mathematics, University of California, Los Angeles

1 Introduction

Zeroth-order optimization, also known as derivative-free or black-box optimization, appears in a wide range of applications where either the objective function is implicit or its gradient is impossible or too expensive to compute. This type of optimization has a wide range of application in the area of machine learning, for instance the adversarial attacks on neural network.

In this project, we improved the existing algorithm ZORO [3], which is also raised by our group before, to Block-ZORO, in order to better deal with high dimensional regularized zeroth-order optimization problems:

$$\min_{x\in\mathbb{R}^d} F(x) := f(x) + r(x) \tag{1}$$

where $r(x)$ is an *explicit* proximable function that enforces constraints and encodes the solution's structure, and $f(x)$ is accessible through a noisy zeroth order oracle:

$$E_f(x) = f(x) + \xi \tag{2}$$

where ξ is the unknown oracle noise. When we call the oracle with an input x, it returns $E_f(x)$ in which ξ may or may not change every time.

The complexities of zeroth-order methods are measured in the number of oracle queries needed to achieve a target accuracy. Both Block-ZORO and ZORO reduce such complexities by exploiting the fact that most gradients (i.e., those in the set $\{\nabla f(x) : x \in \mathbb{R}^d\}$) are often **compressible**, meaning that the sorted sizes of their components decay like a polynomial (See Figure 2)!

ZORO's main iteration is based on the proximal-gradient method but uses an approximate gradient $\hat{\mathbf{g}}_k$ that is estimated from multiple (noisy) samples of f near x_k. The estimation uses **randomized finite differences** and **compressed sensing**, which will be discussed in details in Section 2.

Block-ZORO improves ZORO in terms of storage and running time complexity, which are both significant in extremely large scale problems. We replace the full random sensing matrix in ZORO to a **block diagonal sensing matrix** with circulant property in each block, and apply the **block coordinate gradient descent** [9] to ZORO to generate the Block-ZORO, to achieve better storage and speed. Additionally, we prove the **sublinear convergence** of Block-ZORO for smooth convex objective function $f(x)$.

To illustrate the performance of Block-ZORO, we apply the algorithm to solve the adversarial attack problem on neural networks. To better cope with the compressibility assumption on the gradients, we discover that attacking on frequency domain (e.g. wavelet domain) can provide high-quality attack images with nearly imperceptible noises.

2 Summary on ZORO

In this section we briefly review the algorithm ZORO from [3]. ZORO is basically a proximal-gradient descent scheme but without the access of exact gradient information. Below is the brief version of ZORO, which eliminates several tricky improvements on adaptive query radius and opportunistic sampling. For those who are interested in the full version, please refer to [3].

Algorithm 1: Zeroth Order Regularized Optimization Method (ZORO) - brief version

for $k = 0 : K$ **do**
  Sample some oracles around x_k;
  $\hat{\mathbf{g}}_k \leftarrow$Gradient Estimation$(x_k, s, \delta, \{z_i\}_{i=1}^m)$;
  $x_{k+1} \leftarrow \mathbf{prox}_{\alpha r}(x_k - \alpha\hat{\mathbf{g}}_k)$;
end

It's obvious that main difficulty is how we estimate the gradient $\mathbf{g}_k$ accurately and efficiently, with the access only to oracle queries $E_f(x_k)$. Quite intuitively, we combine the ideas of **random sampling** and **finite difference** to approximate the gradient. In ZORO, we used the Rademacher random vectors z_i (i.e., $(z_i)_j = \pm 1$ with equal probability for all i, j) as the sampling direction. Typically, the finite difference can be calculated along each coordinate from 1 to d:

$$y_i = \frac{1}{\sqrt{d}}\frac{E_f(x+\delta z_i) - E_f(x)}{\delta} \tag{3}$$

To reduce the queries as many as possible, the punchline of ZORO is to implement **compressed sensing**[5], which means, by assuming that the gradient is **compressible** (or approximately sparse) with parameter s, instead of going through all d coordinates, we only needs $m = \mathcal{O}(s\log(d))$ finite differences to get a high accuracy gradient estimation.

Definition 2.1. *The gradient is **compressible** (or approximately sparse) if*

$$\exists p \in (0,1)\ s.t.,\ |\nabla f(x)|_{(i)} \le i^{-1/p}\|\nabla f(x)\|_2.$$

In this case, we can have the finite difference to be

$$y_i = \frac{1}{\sqrt{m}}\frac{E_f(x+\delta z_i) - E_f(x)}{\delta} \tag{4}$$

and by Taylor expansion, the gradient estimation is equivalent to the sparse recovery problem in compressed sensing:

$$\hat{\mathbf{g}} = \arg\min_{v\in\mathbb{R}^d} \|Zv - \mathbf{y}\|_2 \quad s.t., \|v\|_0 \le s \tag{5}$$

where $\mathbf{y} = [y_1, \cdots, y_m]^T$ and $Z \in \mathbb{R}^{m\times d}$ is the *sensing matrix* whose $i-$th row is $\frac{1}{\sqrt{m}}z_i^T$. The problem (5) can be easily solved by existing algorithm CoSaMP, because the matrix Z is a sub-Gaussian matrix satisfying the Restricted Isometry Property (RIP) in compressed sensing theory [1]. Therefore, we have the following algorithm for gradient estimation:

Algorithm 2: Gradient Estimation

Input: x: current point; s: gradient sparsity level; δ: query radius; $\{z_i\}_{i=1}^m$: sample directions
$m \leftarrow b_1 s\log(d/s)$ where b_1 is some selected constant;
for $i = 1 : m$ **do**
  $y_i \leftarrow (E_f(x+\delta z_i) - E_f(x))/(\sqrt{m}\delta)$
end
$\mathbf{y} \leftarrow [y_1, \cdots, y_m]^T$; $Z \leftarrow 1/\sqrt{m}[z_1, \cdots, z_m]^T$;
$\hat{\mathbf{g}} \approx \arg\min_{\|v\|_0\le s} \|Zv - \mathbf{y}\|_2$ by CoSaMP;

3 Improvements by Block-ZORO

3.1 Better Storage with Block Diagonal Sensing Matrix

In the developing field of machine learning, optimization algorithms which can solve large-scale problems are of great need. In many tasks of computer vision, large-scale data need to be handled, while the memory of computers is usually quite limited. With such motivation, we improve ZORO so that it requires much less memory. The major improvement is by considering special forms of the sensing matrix Z. We replace the Rademacher random matrix Z in ZORO algorithm to a **random block diagonal sensing matrix** B, with **circulant** blocks B_i:

$$Z \to B := \begin{pmatrix} B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & B_J \end{pmatrix} \in \mathbb{R}^{m\times d},\ where\ B_i = \begin{pmatrix} b_n & b_{n-1} & \cdots & \cdots & b_1 \\ b_1 & b_n & b_{n-1} & \ddots & \vdots \\ b_2 & b_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & b_n & b_{n-1} \\ b_{n-1} & \cdots & b_2 & b_1 & b_n \end{pmatrix} \tag{6}$$

where J denotes the number of blocks and B_i are circulant blocks generated from a Rademacher random vector. Since B is mostly sparse, we only need to store the information of sub-matrices $B_1, \cdots, B_J$. Furthermore, we can let $B_1, \cdots, B_J$ to be identical for a even better memory by storing only B_1. And since we also let B_i to be circulant, only one particular column of B_i (which is Rademancher) is needed to store. However, there is a trade-off that the sampling number m needs to be larger [6][7], in order to satisfy the RIP for high quality recovery of (5):

$$m = \mathcal{O}(Js\log^2(d)) \tag{7}$$

3.2 Better Time complexity with Block Coordinate Descent

Thanks to the diagonal block structure raised in the previous subsection, the sparse recovery problem using the sensing matrix B can be written in the form of the sum of subproblems (where $v = (v^{(1)T}\cdots v^{(j)T}\cdots v^{(J)T})^T$):

$$\min_{v\in\mathbb{R}^d} \|Bv - \mathbf{y}\|_2^2 = \sum_{j=1}^{J} \min_{v^{(j)}\in\mathbb{R}^{d/J}} \|B^{(j)}v^{(j)} - \mathbf{y}^{(j)}\|_2^2 \tag{8}$$

We show that when d, s are large enough compared to J, the sparsity can be treated as almost equally distributed among all blocks. Thus the subproblem for each block becomes

$$\hat{\mathbf{g}}^{(j)} = \text{argmin}_{v^{(j)}\in\mathbb{R}^{d/J}} \left\{\|B^{(j)}v^{(j)} - \mathbf{y}^{(j)}\|_2 \text{ subject to: } \|v^{(j)}\|_0 \le s/J\right\} \tag{9}$$

Therefore, we can apply the idea of **block coordinate gradient descent** with inexact gradient [9] to ZORO quite intuitively from the splitting above. This means in each GD iteration, we only update by the gradient of one block $\hat{\mathbf{g}}^{(j)}$ (here we abuse the notation $\hat{\mathbf{g}}^{(j)}$ to be $d-$dimension with the rest to be zeros):

$$x_{k+1} = \mathbf{prox}_{\alpha r}(x_k - \alpha\hat{\mathbf{g}}_k^{(j)}) \tag{10}$$

Since this improvement greatly reduces the dimension in each iteration by order J, the overall time complexity is reduced with $\exp(J)$. Therefore we can now develop the algorithm Block-ZORO as follows:

Algorithm 3: Block-ZORO

Input: x_0: initial point; s: gradient sparsity level; α: step size; δ: query radius; J: number of blocks
$m \leftarrow b_1 s\log(d/s)$ where b_1 is some selected constant;
for $i = 1 : m/J$ **do**
  Generate Rademacher random vector z_i.
end
for $k = 0 : K$ **do**
  Randomly select a block j;
  $\hat{\mathbf{g}}_k^{(j)} \leftarrow$Gradient Estimation$(x_k^{(j)}, s/J, \delta, \{z_i\}_{i=1}^{m/J})$;
  $x_{k+1} \leftarrow \mathbf{prox}_{\alpha r}(x_k - \alpha\hat{\mathbf{g}}_k^{(j)})$;
end

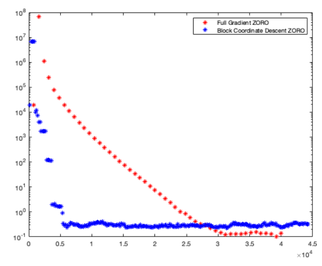


Figure 1: Comparasion of ZORO and Block-ZORO

3.3 Convergence Results of Block-ZORO

We assume that the objective function f is convex, Lipschitz differentiable, smooth, and the function evaluation $E_f(x)$ is noisy in the following theorem.

Theorem 3.1 (Main Theorem on Noisy Block-ZORO)**.** *Choose an initial point $x_0 \in \mathbb{R}^d$ and let $\{x_k\}_{k\ge 0}$ be the random iterates generated by Block-ZORO applied to minimizing f. Assume that the probability of choosing each block at each iteration is identical among all blocks (i.e., $1/C$). Also assume $\frac{16\tau\sigma H}{c_1 L} < 1$ where $c_1 = 2\|x_0 - x^*\|_2^2$. Let $L1, \cdots, L_C$ be the corresponding Lipschitz constant for each blocks, and let $L = \min_{i\in[1,\cdots C]} L_i$. Choose confidence level $0 < \rho < 1$, error tolerance ϵ satisfying $\frac{c_1}{2}\sqrt{\frac{16\tau\sigma H}{c_1\rho L}} < \epsilon$ and $\epsilon < f(x_0) - f^*$. If the iteration number K satisfies*

$$K = \frac{c_1}{\epsilon - u} + \frac{c_1}{\epsilon}\log\left(\frac{\epsilon - \frac{4\tau\sigma H c_1}{\epsilon L}}{\epsilon\rho - \frac{4\tau\sigma H c_1}{\epsilon L}}\right) + 2 \approx \mathcal{O}(\frac{1}{\epsilon})$$

where $u = \frac{c_1}{2}\sqrt{\frac{16\tau\sigma H}{c_1 L}}$, then $\mathbb{P}(f(x_K) - f^* \le \epsilon) \ge 1 - \mathcal{O}(\rho + (s/d)^{\mathcal{O}(s)})$.

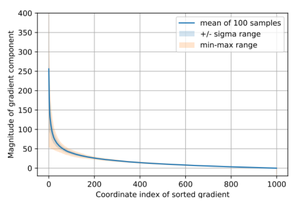
Proof. Please refer to [3] and [2]. □

4 Sparse Adversarial Wavelet Attack by Block-ZORO

To verify the efficiency of Block-ZORO on a large-scale problem, we implement it to the adversarial attack problems on neural network. We attack the image classification CNN model Inception-V3 [8] from Google on the ImageNet [4]. We use the Carlini and Wagner (CW) Attack lost function as the objective, and a control term on box constraints.

To better fulfill the assumption of compressible gradients, we discover that by attacking on the frequency domain of the images, the gradients are more likely to be compressible (see figure 2 below). This is also intuitive from the sense of the different significance of low-pass filters and high-pass filters. In the results showed below, we attack on the **wavelet domain**, which is an important transform that is frequently used in JPEG image compression.

The table below shows our algorithm outperforms other recent proposed blackbox optimization algorithms on adversarial attack. And Figure 3 & 4 give untargeted and targeted attack samples respectively, with the noise scaled by 10 times.



METHODS	ASR	ℓ_0 DIST	ℓ_2 DIST	ITER
ZO-SCD	78 %	0.89%	57.5	240
ZO-SGD	78%	100%	37.9	159
ZO-AdaMM	81%	100%	28.2	172
ZORO	**90%**	**0.73%**	**21.1**	**59**

Figure 2: compressible gradients & Attack successful rate

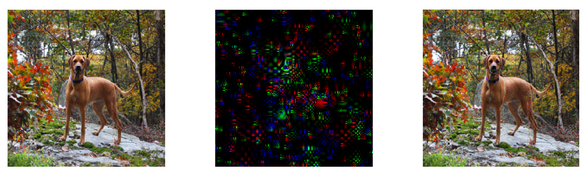


Figure 3: Untargeted attack with 10 times scaled noise. Label: "Rhodesian ridgeback" to "whippet"

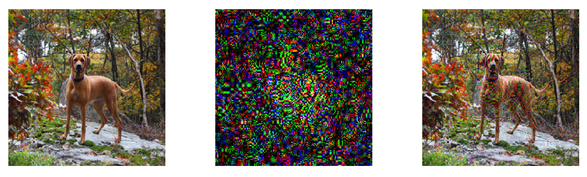


Figure 4: Targeted attack with 10 times scaled noise. Label: "Rhodesian ridgeback" to "pizza"

References

[1] R. Baraniuk et al. "A simple proof of the restricted isometry property for random matrices". In: *Constructive Approximation* 28.3 (2008), pp. 253–263.

[2] H. Cai et al. "Sparse Adversarial Wavelet Attacks using a zeroth-order block gradient descent". In: *To be uploaded* (2020).

[3] H. Cai et al. "Zeroth-Order Regularized Optimization (ZORO): Approximately Sparse Gradients and Adaptive Sampling". In: *arXiv preprint arXiv:2003.13001* (2020).

[4] J. Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[5] D. L. Donoho. "Compressed sensing". In: *IEEE Transactions on information theory* 52.4 (2006), pp. 1289–1306.

[6] A. Eftekhari et al. "The restricted isometry property for random block diagonal matrices". In: *Applied and Computational Harmonic Analysis* 38.1 (2015), pp. 1–31.

[7] S. Mendelson, H. Rauhut, R. Ward, et al. "Improved bounds for sparse recovery from subsampled random convolutions". In: *The Annals of Applied Probability* 28.6 (2018), pp. 3491–3527.

[8] C. Szegedy et al. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[9] R. Tappenden, P. Richtárik, and J. Gondzio. "Inexact coordinate descent: complexity and preconditioning". In: *Journal of Optimization Theory and Applications* 170.1 (2016), pp. 144–176.