ZEROTH-ORDER RANDOMIZED SUBSPACE NEWTON METHODS

Erik Berglund, Sarit Khirirat, Xiaoyu Wang

KTH Royal Institute of Technology

ABSTRACT

Zeroth-order methods have become important tools for solving problems where we have access only to function evaluations. However, the zeroth-order methods only using gradient approximations are n times slower than classical first-order methods for solving n-dimensional problems. To accelerate the convergence rate, this paper proposes the zeroth order randomized subspace Newton (ZO-RSN) method, which estimates projections of the gradient and Hessian by random sketching and finite differences. This allows us to compute the Newton step in a lower dimensional subspace, with small computational costs. We prove that ZO-RSN can attain lower iteration complexity than existing zeroth order methods for strongly convex problems. Our numerical experiments show that ZO-RSN can perform black-box attacks under a more restrictive limit on the number of function queries than the state-of-the-art Hessian-aware zeroth-order method.

Index Terms— Zeroth-order optimization, sketching techniques, Newton-type method, adversarial black-box attacks, convolutional neural network.

1. INTRODUCTION

Several applications in machine learning, signal processing and communication networks can often be cast as optimization problems, where gradients are difficult or even infeasible to compute. Popular application examples include optimal hyper-parameter tuning for learning models [1, 2], black-box adversarial attacks on neural network models [3, 4, 5, 6] and sensor selection problems in smart grids or wireless networks [7, 8, 9]. This motivates the study of the zeroth-order methods. A prominent type of zeroth order methods uses function value differences to estimate the gradients [10, Section 3.4]. However, these methods are much slower than classical gradient descent [11], and also suffers from poor performance particularly for ill-conditioned problems. An alternative way to improve their performance is to incorporate the second order information into zeroth-order methods. However, computing the full Hessian matrix can heavily increase the number of function evaluations and make the Newton step hard to compute, especially for high-dimensional problems. This necessitates us to approximate the Hessian matrix in a lower-dimensional subspace.

Ye *et al.*[12] developed the Hessian-aware zeroth-order (ZOHA) methods, which integrate Hessian information into zeroth-order methods. The power-iteration based method ZOHA-PW has a lower query complexity than the gradient-estimating method by [11] when the eigenvalues of the Hessian decay sufficiently quickly. However, the power iteration method requires $O(n)$ function queries per iteration for n-dimensional problems, which is expensive when n is large. To decrease the query cost, they proposed the heuristic methods ZOHA-Gauss-DC and ZOHA-Diag-DC, which estimate the Hessian based on a limited number of random directions. However, no complexity bounds are provided for them.

Another approach to reduce the times of computing Hessian information for high-dimensional problems is to use randomized sketching techniques [13, 14, 15]. These sketching techniques construct lower dimensional sub-problems, which can be solved within small computation times, and enable classical optimization algorithms to have better scalability. For instance, a randomized subspace newton (RSN) method [14] exploits the sketching techniques on the Newton method to solve the problems with very large dimension and to achieve accelerated convergence rate.

In this paper, we propose Hessian-based zeroth-order algorithms using sketching techniques for huge-dimensional problems, called zeroth-order RSN (ZO-RSN). The methods exploit finite differences and sketching to approximate projections of the gradient and Hessian. We provide complexity bounds and prove that under certain conditions ZO-RSN attains lower query complexity than existing zeroth-order algorithms for strongly convex problems. Finally, our experiments with black-box attack problems on a convolutional neural network show that ZO-RSN has an overall competitive performance and higher success rate, compared to the ZOHA-Gauss-DC method in [12].

1.1. Notation

For $x \in \mathbb{R}^n$ and $M \succ 0$, $\|x\|_2$ and $\|x\|_\infty$ are the ℓ_2 and ℓ_∞ norm, respectively, and $\|x\|_M^2 = x^T Mx$. Given the sketching matrix $S \in \mathbb{R}^{n\times m}$, $s_1, s_2, \ldots, s_m \in \mathbb{R}^n$ are its columns.

This work was supported partially by the Swedish Foundation for Strategic Research (SSF) project SoPhy and the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation. Emails: `erbergl@kth.se`, `sarit@kth.se`, `wang10@kth.se`.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

For $f : \mathbb{R}^n \to \mathbb{R}$, $g(x) = \nabla f(x)$ and $H(x) = \nabla^2 f(x)$ are its gradient and Hessian. The function $f(x)$ is L-Lipschitz continuous if there exists a positive constant L such that $\|f(y) - f(x)\|_2 \le L\|y - x\|_2$ for all $x, y \in \mathbb{R}^n$, and μ-strongly convex if there exists a positive constant μ such that $f(y) \ge f(x) + \langle \nabla f(x), y - x \rangle + (\mu/2)\|y - x\|_2^2$ for all $x, y \in \mathbb{R}^n$. We also state that the differentiable function $f(x)$ is L_s-smooth if its gradient $g(x)$ is L_s-Lipschitz continuous. Finally, for any $y \in \mathbb{R}^n$, $\Delta_y f(x) = f(x + y) - f(x)$.

2. PROBLEM FORMULATION

We consider the unconstrained optimization problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x), \tag{1}$$

where the dimension n could be very large. Here, $f(x)$ is a three times differentiable and μ-strongly convex function, which is bounded from below and has its minimum value f^* at the point x^*. $g(x)$ and $H(x)$ are also L_1- and L_2-Lipschitz continuous. To facilitate the analysis, we further make the following standard assumption on $f(x)$.

Assumption 1 ([14, 16])**.** *There exists $\hat{L} \ge \hat{\mu} > 0$ such that for any $x, y \in \mathbb{R}^n$:*

$$f(x) \le f(y) + g(y)^T(x - y) + (\hat{L}/2)\|x - y\|^2_{H(y)}, \tag{2}$$

$$f(x) \ge f(y) + g(y)^T(x - y) + (\hat{\mu}/2)\|x - y\|^2_{H(y)}. \tag{3}$$

Assumption 1 states the smoothness and strong convexity of $f(x)$ under the norm weighted by its Hessian $\|\cdot\|_{H(x)}$. Also, the $\hat{L}$-relative smoothness and $\hat{\mu}$-relative convexity exist as a result of the L_1-smoothness and μ-strong convexity assumption on $f(x)$, as shown below:

Proposition 2.1 ([14, 16])**.** *A function $f(x)$ is c-stable on a domain D if $\forall y, z \in D$, $\|z - y\|^2_{H(y)}$ and there exists a constant $c \ge 1$ such that $c = \|z - y\|^2_{H(z)} / \|z - y\|^2_{H(y)}$. If $f(x)$ is μ-strongly convex and L_1-smooth, then f is (L_1/μ)-stable. Furthermore, if $f(x)$ is c-stable, then Assumption 1 holds with $\hat{L} \le c$ and $\hat{\mu} \ge 1/c$.*

2.1. RSN Methods

The randomized subspace Newton (RSN) method [14] is a popular inexact Newton method for solving huge-dimensional problems. This method solves an exact Newton system restricted to a random subspace. Given a fixed step-size $\gamma > 0$ and an initial point $x_0 \in \mathbb{R}^d$, the iterate x_k of the RSN method is updated via:

$$x_{k+1} = x_k + \gamma S_k \lambda_k, \;\; S_k^T H(x_k) S_k \lambda_k = -S_k^T g(x_k), \tag{4}$$

where $S_k \in \mathbb{R}^{n \times m}$ stores m vectors that span the randomly selected subspace of $\mathbb{R}^n$. The next lemma characterizes the decrease in the function value from the ZO-RSN method (4).

Lemma 1. *Consider the RSN method* (4) *for solving Problem* (1)*. If $\gamma \le 1/\hat{L}$, then*

$$f(x_{k+1}) \le f(x_k) - (\gamma/2)\|g(x_k)\|^2_{S_k(S_k^T H(x_k) S_k)^\dagger S_k^T}. \tag{5}$$

This descent lemma for the RSN method can be used to prove its linear convergence toward the exact optimum [14]. Furthermore, to implement the RSN method $S_k^T H(x_k) S_k$ and $S_k^T g(x_k)$ are computed efficiently by various sketching techniques such as sub-Gaussian sketches, randomized orthonormal system sketches, random sampling sketches and the Iterative Hessian Sketch [17] as well as the fast Johnson-Lindenstrauss sketch for problems with the appropriate structure [18]. These sketching techniques allow for computing λ_k with very small linear equation systems. If $m \ll n$, then λ_k in Eq. (4) can be determined quickly by inverting $S_k^T H(x_k) S_k \in \mathbb{R}^{m \times m}$.

3. ZEROTH-ORDER RSN METHODS

In this section, we introduce the zeroth-order randomized subspace Newton (ZO-RSN) method, which builds on the RSN method. The iterate x_k of the ZO-RSN algorithm is updated according to:

$$x_{k+1} = x_k + \gamma S_k \tilde{\lambda}_k, \;\text{ and }\; \tilde{H}_{S_k}(x_k)\tilde{\lambda}_k = -\tilde{g}_{S_k}(x_k). \tag{6}$$

Here $\tilde{g}_{S_k}(x_k)$ and $\tilde{H}_{S_k}(x_k)$ are approximations of the sketched gradient and Hessian respectively. For a positive scalar α, they can be computed via:

$$[\tilde{g}_{S_k}(x_k)]_i := \Delta_{\alpha s_{i,k}} f(x_k)/\alpha \approx s_{i,k}^T g(x_k), \quad \text{and}$$
$$[\tilde{H}_{S_k}(x_k)]_{i,j} := \Delta_{\alpha s_{i,k}} \Delta_{\alpha s_{j,k}} f(x_k)/\alpha^2 \approx s_{i,k}^T H(x_k) s_{j,k},$$

for all $i = 1, \ldots, m$. Similarly to Lemma 1, the ZO-RSN method can be proved to achieve the following bound:

$$f(x_{k+1}) \le f(x_k) - \frac{\gamma}{2}\|g(x_k)\|^2_{S_k(S_k^T H(x_k) S_k)^\dagger S_k^T} + O(\alpha). \tag{7}$$

This ensures function value improvement in Eq. (7) if α is sufficiently small and $\tilde{H}_{S_k}(x_k)$ is positive definite. In fact, we can ensure that positive definiteness of $\tilde{H}_{S_k}(x_k)$ follows from α being small enough if we choose S_k appropriately.

Lemma 2. *If $S_k^T S_k = I$ and $\|\tilde{H}_{S_k}(x_k) - S_k^T H(x_k) S_k\|_2 < \mu$, then $\tilde{H}_{S_k}(x_k) \succ 0$.*

Based on this lemma, we set $S_k^T S_k = I$ to ensure that $\tilde{H}_{S_k}(x_k) \succ 0$. We also require $\mathbb{E}[S_k S_k^T] \succ 0$ so that the approximate sketching does not leave out any directions throughout every iteration. This requirement can be easily satisfied if $s_{1,k}, \ldots, s_{m,k}$ are sampled from unit coordinate directions without replacement.

4. THEORETICAL RESULTS

We now provide a complexity bound for ZO-RSN methods. Proofs of our theoretical results can be found in [19].

Theorem 1. *Let the sketching matrix $S_k \in \mathbb{R}^{n\times m}$ satisfy $S_k^T S_k = I$ and $\mathbb{E}_{S_k\sim D}[S_k S_k^T] \succ 0$, and define $G(x) = \mathbb{E}_{S_k\sim\mathcal{D}}[S_k(S_k^T H(x) S_k)^{-1} S_k^T]$,*

$$\rho(x) = \min_{v\in\mathbb{R}^n} \frac{v^T H(x)^{\frac{1}{2}} G(x) H(x)^{\frac{1}{2}} v}{\|v\|_2^2} \quad \textit{and} \quad \rho = \min_{x\in\mathbb{R}^n} \rho(x).$$

Given $\varepsilon > 0$ and $\delta \in (0,1)$, consider the ZO-RSN method (6) *for Problem* (1)*. If $\gamma \le 1/\hat{L}$ and $\alpha \le 0.3\mu/(mL_2)$ is small enough that*

$$\frac{\alpha(C_1 + C_2\alpha)}{\rho\hat{\mu}\gamma - \alpha C_1 - \alpha^2 C_3} \le \delta\varepsilon \quad \textit{and} \quad \alpha C_1 + \alpha^2 C_3 < \rho\hat{\mu}\gamma,$$

then we can achieve $\mathbb{E}[f(x_k) - f^*] \le \varepsilon$ after

$$k \ge \left\lceil \log\left(\frac{f(x_0) - f^*}{(1-\delta)\varepsilon}\right) \Big/ \log\left(\frac{1}{1 - \rho\hat{\mu}\gamma + \alpha C_1 + \alpha^2 C_3}\right)\right\rceil$$

iterations where $C_1 = \gamma(\sqrt{m}L + B)/(2\mu)$, $C_2 = \gamma L_1^2[m + \sqrt{m}(1+B)]/(2\mu^2)$, $C_3 = \gamma L_1[\sqrt{m}L_1(1+B) + B(2+B)]/(2\mu^2)$ and $B = 10mL_2/(3\mu)$.

Theorem 1 establishes a global, linear convergence for the ZO-RSN method toward an ε-accurate solution. The worst-case iteration complexity can be upper bounded as

$$k \ge \lceil \beta_1 \log\left([f(x_0) - f^*]/[(1-\delta)\varepsilon]\right)\rceil . \tag{8}$$

where $\beta_1 = 1/(\rho\hat{\mu}\gamma - \alpha C_1 - \alpha^2 C_3)$. We can recover the convergence complexity for the RSN method [14] if α and δ approach zero. Furthermore, by choosing S_k properly, the iteration complexity for the ZO-RSN method in Eq. (8) can be lower than the complexities for existing zeroth-order methods. We show this with the following corollary:

Corollary 4.1. *Suppose all the conditions of Theorem 1 hold. If the columns of S_k are chosen randomly without replacement from a basis of orthonormal eigenvectors of $H(x_k)$, step-size $\gamma = 1/\hat{L}$, and $\alpha = (\sqrt{C_1^2/4 + (1-\sigma)\rho\hat{\mu}\gamma} - C_1/2)/C_2$ for some $\sigma \in (0,1)$, then $\rho = m/n$ and hence to achieve $\mathbb{E}[f(x_k) - f^*] \le \varepsilon$, we need*

$$k \ge \left\lceil (n\hat{L}/[\sigma m\hat{\mu}]) \log\left([f(x_0) - f^*]/[(1-\delta)\varepsilon]\right)\right\rceil . \tag{9}$$

Corollary 4.1 shows that the iteration complexity of the ZO-RSN methods depends on the subspace dimension m, the problem dimension n and other parameters $\hat{\mu}, \hat{L}$. Since the ZO-RSN methods need $m(m+1)/2$ function queries per iteration, we can obtain the total query complexity by multiplying Eq. (9) with this factor.

Now, we compare the complexity bounds for the ZO-RSN methods against the Hessian-aware zeroth-order method using the power iteration (ZOHA-PW) [12], which previously has been compared favourably to the zeroth-order method in [11]. Since the ZOHA-PW method also generates multiple random directions, here m refers to the number of the generated directions. For μ-strongly convex problems, the iteration complexity of ZOHA-PW is

$$k \ge \left\lceil \beta_2 \log\left([f(x_0) - f^*]/[(1-\hat{\delta})\varepsilon]\right)\right\rceil , \tag{10}$$

where $\beta_2 = 64(n+2)(\mu + 10\lambda_{s+1})/(\mu m)$, λ_{s+1} is an upper bound on the Hessian's $(s+1)^{\text{th}}$ largest eigenvalue and $\hat{\delta}$ is a free parameter which is similar to δ in Eq. (9). Disregarding the function evaluations required to implement the power method, the total query complexity for ZOHA-PW is $2m$ times its iteration complexity. Consider the problem of minimizing a quadratic function. Then, $\hat{L} = \hat{\mu} = 1$. If δ is set equal to $\hat{\delta}$, m is the same for both methods, and $\sigma = 0.5$, then the speedup in iteration complexity from using ZO-RSN instead of ZOHA-PW is

$$32\,(1 + 2/n)\,(1 + 10\lambda_{s+1}/\mu) .$$

ZO-RSN is thus faster than ZOHA-PW by more than two orders of magnitude in iteration complexity, even for well-conditioned problems (when λ_{s+1}/μ is close to one). If function queries can be performed efficiently in parallel, then ZO-RSN has significantly lower run-time than ZOHA-PW. We can also prove that the speedup in query complexity for ZO-RSN compared to ZOHA-PW is

$$[128\,(1 + 2/n)\,(1 + 10\lambda_{s+1}/\mu)]/(m+1).$$

Thus, as long as $m < 128\,(1 + 10\lambda_{s+1}/\mu) - 1$, the query complexity of ZO-RSN will be lower than that of ZOHA-PW.

5. NUMERICAL EXPERIMENTS

We compare the performance of ZO-RSN against the existing Hessian-aware zeroth methods called ZOHA-Gauss-DC [12] that uses a descent-checking procedure to increase an attack success rate, and approximates the Hessian according to

$$\tilde{H} = (2\alpha^2 b)^{-1} \sum_{i=1}^{b} |\Delta_{\alpha u_i} f(x) - \Delta_{\alpha u_i} f(x - \alpha u_i)| u_i u_i^T + \lambda I_d,$$

where λ is a positive constant and $u_1, \ldots, u_b$ are the vectors generated from the Gaussian distribution with zero mean and unit variance. In particular, we evaluate both methods on training un-targeted black box adversarial attacks over the `MNIST` data set [20, 12]. These attacks are carried out against the trained convolutional neural network (CNN) model described in [12][Section 5.2]. For each example x_i^{nat} in the test set, the optimizer aims to generate an adversarial example x_i which differs from x_i^{nat} by at most ϵ in ℓ_∞ norm, while being classified differently with sufficient confidence. This is done by minimizing the following function [20]:

$$f(x) = \max\left\{\log[Z(x)]_l - \max_{i\neq l} \log[Z(x)]_i, -\omega\right\} . \tag{11}$$

Here, $[Z(x)]_i$ represents the probability of an input x belonging to class i according to the trained neural network.

Since the problem is constrained and does not have guarantees for μ-strong convexity or L_1-smoothness, we need to modify the ZO-RSN algorithm. Firstly, we artificially ensure positive definiteness and boundedness of $\tilde{H}_{S_k}(x_k)$ by applying the operator $\Pi_{[\lambda_{\min},\lambda_{\max}]}(\cdot)$ that projects its eigenvalues onto an interval $[\lambda_{\min}, \lambda_{\max}]$ to get a modified matrix $\hat{H}_{S_k}(x_k)$. Secondly, we consider the ℓ_∞-norm constraints by determining a $\tilde{\lambda}_k$ that solves the following minimization problem

$$\begin{aligned} \underset{\lambda\in\mathbb{R}^m}{\text{minimize}} \quad & f(x_k)+\gamma\tilde{g}_{S_k}(x_k)^T\lambda+\frac{\gamma}{2}\|\lambda\|^2_{\hat{H}_{S_k}(x_k)} \\ \text{subject to}\quad & -\gamma S_k\lambda \le x_k - x_i^{nat} - \mathbf{1}\epsilon \\ & \gamma S_k \lambda \le \mathbf{1}\epsilon + x_i^{nat} - x_k. \end{aligned} \tag{12}$$

This approach corresponds to using sequential quadratic programming (SQP) for nonlinear problems with linear constraints, but with the step to the next iterate being restricted to lie in a specific subspace. To solve the auxiliary problem (12) quickly with a standard `cvxopt` solver [21], we generate S_k by choosing its columns to be unit coordinate vectors. This enables us to formulate the problem with only m constraints. This adapted ZO-RSN algorithm is called ZO-RSN-SQP. Finally, we use the descent-checking technique corresponding to that for ZOHA-Gauss-DC. The full description of ZO-RSN-SQP is given in Algorithm 1.

Algorithm 1 ZO-RSN-SQP for black-box attack

```
Initialize $x_0 \leftarrow x_i^{nat}, \alpha, \gamma, m, m_{max}$
for $k = 0, 1, ..., k_{max}$ do
    Generate $S_k = [s_{1,k}, ..., s_{m,k}]$
    Compute $\tilde{g}_{S_k}$ and $\tilde{H}_{S_k}$
    $\hat{H}_{S_k} \leftarrow \Pi_{[\lambda_{\min},\lambda_{\max}]}(\tilde{H}_{S_k})$
    $\tilde{\lambda}_k \leftarrow$ Solution to (12) with $\hat{H}_{S_k}(x_k)$ and $\tilde{g}_{S_k}(x_k)$
    $x_{trial} \leftarrow x_k + S_k\tilde{\lambda}_k$
    while $f(x_{trial}) \geq f(x_k)$ and $\bar{m} < m_{max}$ do
        $\bar{m} \leftarrow \bar{m} + 1$
        Generate $s_{\bar{m},k}$ such that $[S_k, s_{\bar{m},k}]^T[S_k, s_{\bar{m},k}] = I$
        $S_k \leftarrow [s_{1,k}, ..., s_{\bar{m},k}]$
        $[\tilde{g}_{S_k}(x_k)]_{\bar{m}} \leftarrow \Delta_{\alpha s_{i,k}} f(x_k)/\alpha$
        for $j = 1, 2, ..., \bar{m}$ do
            $[\tilde{H}_{S_k}(x_k)]_{\bar{m},j} \leftarrow \Delta_{\alpha s_{i,k}}\Delta_{\alpha s_{j,k}} f(x_k)/\alpha^2$
            $[\tilde{H}_{S_k}(x_k)]_{j,\bar{m}} \leftarrow [\tilde{H}_{S_k}(x_k)]_{\bar{m},j}$
        end for
        $\hat{H}_{S_k} \leftarrow \Pi_{[\lambda_{\min},\lambda_{\max}]}(\tilde{H}_{S_k})$
        $\tilde{\lambda}_k \leftarrow$ Solution to (12) with $\hat{H}_{S_k}(x_k)$ and $\tilde{g}_{S_k}(x_k)$
        $x_{trial} \leftarrow x_k + \gamma S_k\tilde{\lambda}_k$
    end while
    if $f(x_{trial}) \leq f(x_k)$ then
        $x_{k+1} \leftarrow x_{trial}$
    else
        $x_{k+1} \leftarrow x_k$
    end if
end for
```

We trained the network model until its accuracy reached 98.84%, and also set $\alpha = 0.1, \gamma = 1, m = 3$ and $m_{\max} = 20$ for ZO-RSN-SQP and the same parameters for ZOHA-Gauss-DC as in the un-targeted black box attacks described in [12]. In the experiments, we either ended a test run if the algorithm managed to find a point with function value at $\omega = -1$, or if the algorithm called queried the neural network for a prediction 50000 times. We labelled the former result as a success and the latter result as a failure.

Algorithm	ZO-RSN-SQP	ZOHA-Gauss-DC
Success rate (%)	**100**	95.33
Median queries	2336	**815**
Mean queries	**2510**	4164
Max queries	**8239**	50000
$f_{est2000} - f^*$	1.94	$\mathbf{3.70 \cdot 10^{-1}}$
$f_{est4000} - f^*$	$1.89 \cdot 10^{-1}$	$\mathbf{1.86 \cdot 10^{-1}}$
$f_{est6000} - f^*$	$\mathbf{2.42 \cdot 10^{-2}}$	$1.47 \cdot 10^{-1}$

Table 1. Comparison of ℓ_∞ norm based black-box attacks on a CNN model trained on the `MNIST` data.

The results of our black box attack experiments were summarized in Table 1. Firstly, ZO-RSN-SQP has a more stable performance than ZOHA-Gauss-DC. Even though both algorithms implement similar decent checking techniques, only ZO-RSN-SQP succeeds in the attacks for all cases. Secondly, the mean number of queries for ZO-RSN-SQP is lower than that for ZOHA-Gauss-DC. This results from a minority of the problems, where ZOHA-Gauss-DC requires a large number of queries to solve. In contrast, ZOHA-Gauss-DC has a lower median value than ZO-RSN-SQP. As ZO-RSN requires more function queries per iteration and subspace dimension than ZOHA-Gauss-DC, one can hypothesize this extra effort is worthwhile mainly for the harder-to-attack test examples.

We also ran a separate experiment to investigate the convergence performance. We used the first 100 MNIST examples, and made estimates of the average objective value after 2000, 4000 and 6000 queries: $f_{est2000}$, $f_{est4000}$ and $f_{est6000}$ (see the results in Table 1). The results suggest that ZO-RSN-SQP initially has a slower convergence speed than ZOHA-Gauss-DC. It catches up to ZOHA-Gauss-DC at around 4000 queries and has clearly surpassed it after 6000 queries.

6. CONCLUSIONS

We have proposed the ZO-RSN method, a Hessian-based zeroth-order method that approximates sketched gradients and Hessians by finite differences. Our results display a lower iteration complexity of the ZO-RSN method than existing zeroth-order methods for particular strongly convex problems [11, 12]. The experiments with un-targeted adversarial attacks on a CNN model illustrate that the modified ZO-RSN method named ZO-RSN-SQP attains an overall competitive performance and a higher stability, compared to the ZOHA-Gauss-DC method of [12].

7. REFERENCES

[1] Jasper Snoek, Hugo Larochelle, and Ryan P Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.

[2] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl, "Algorithms for hyper-parameter optimization," *Advances in neural information processing systems*, vol. 24, 2011.

[3] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 15–26.

[4] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017, pp. 506–519.

[5] Weiwei Hu and Ying Tan, "Generating adversarial malware examples for black-box attacks based on GAN," *arXiv preprint arXiv:1702.05983*, 2017.

[6] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin, "Black-box adversarial attacks with limited queries and information," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2137–2146.

[7] Sijia Liu, Sundeep Prabhakar Chepuri, Makan Fardad, Engin Maşazade, Geert Leus, and Pramod K Varshney, "Sensor selection for estimation with correlated measurement noise," *IEEE Transactions on Signal Processing*, vol. 64, no. 13, pp. 3509–3522, 2016.

[8] Alfred O Hero and Douglas Cochran, "Sensor management: Past, present, and future," *IEEE Sensors Journal*, vol. 11, no. 12, pp. 3064–3075, 2011.

[9] Sijia Liu, Jie Chen, Pin-Yu Chen, and Alfred Hero, "Zeroth-order online alternating direction method of multipliers: Convergence analysis and applications," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2018, pp. 288–297.

[10] Boris Polyak, *Introduction to Optimization*, Optimization Software, Inc., Publications Division, 4 Park Avenue, New York 10016, USA, 1987.

[11] Yurii Nesterov and Vladimir Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, Apr 2017.

[12] Haishan Ye, Zhichao Huang, Cong Fang, Chris Junchi Li, and Tong Zhang, "Hessian-aware zeroth-order optimization for black-box adversarial attack," *arXiv preprint arXiv:1812.11377*, 2018.

[13] Junqi Tang, Mohammad Golbabaee, and Mike E Davies, "Gradient projection iterative sketch for large-scale constrained least-squares," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3377–3386.

[14] Robert Gower, Dmitry Kovalev, Felix Lieder, and Peter Richtarik, "RSN: randomized subspace Newton," in *Advances in Neural Information Processing Systems*. 2019, vol. 32, Curran Associates, Inc.

[15] Mert Pilanci and Martin J Wainwright, "Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence," *SIAM Journal on Optimization*, vol. 27, no. 1, pp. 205–245, 2017.

[16] Sai Praneeth Karimireddy, Sebastian U Stich, and Martin Jaggi, "Global linear convergence of Newton's method without strong-convexity or Lipschitz gradients," *arXiv preprint arXiv:1806.00413*, 2018.

[17] Mert Pilanci and Martin J. Wainwright, "Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1842–1879, Jan. 2016.

[18] Nir Ailon and Bernard Chazelle, "The fast johnson–lindenstrauss transform and approximate nearest neighbors," *SIAM Journal on Computing*, vol. 39, no. 1, pp. 302–322, 2009.

[19] Erik Berglund, Sarit Khirirat, and Xiaoyu Wang, "Zeroth-order randomized subspace Newton methods," *arXiv preprint arXiv:2202.04612*, 2022.

[20] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. 2017, pp. 39–57, IEEE.

[21] M Andersen, J Dahl, and L Vandenberghe, "CVXOPT: python software for convex optimization, version 1.2.6," *URL: https://cvxopt. org*, 2021.

[22] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Matrix Analysis. Cambridge University Press, 2013.