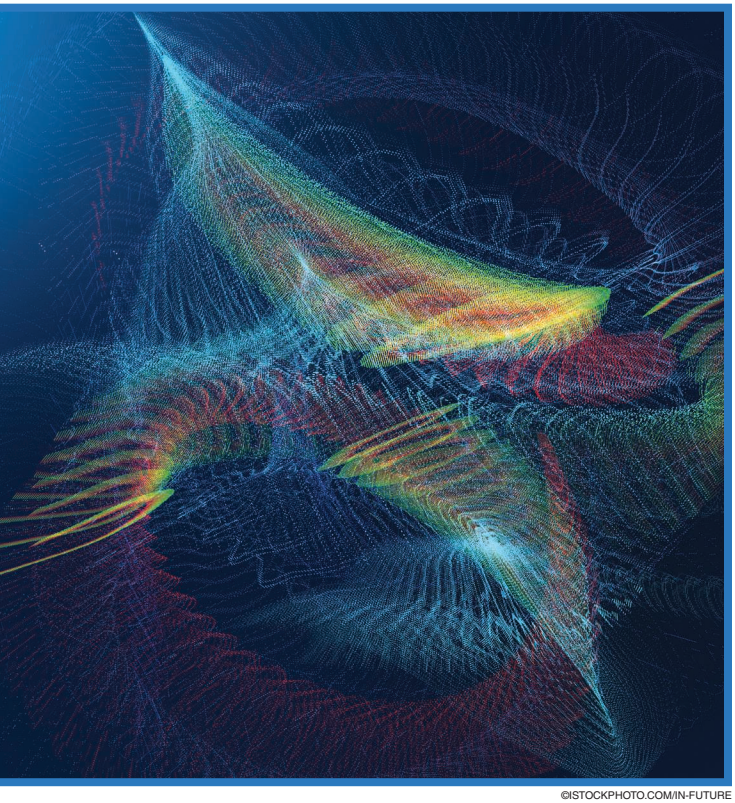
Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O. Hero III, and Pramod K. Varshney

A Primer on Zeroth-Order Optimization in Signal Processing and Machine Learning

Principals, recent advances, and applications



©ISTOCKPHOTO.COM/IN-FUTURE

Zeroth-order (ZO) optimization is a subset of gradient-free optimization that emerges in many signal processing and machine learning (ML) applications. It is used for solving optimization problems similarly to gradient-based methods. However, it does not require the gradient, using only function evaluations. Specifically, ZO optimization iteratively performs three major steps: gradient estimation, descent direction computation, and the solution update. In this article, we provide a comprehensive review of ZO optimization, with an emphasis on showing the underlying intuition, optimization principles, and recent advances in convergence analysis. Moreover, we demonstrate promising applications of ZO optimization, such as evaluating robustness and generating explanations from black-box deep learning (DL) models and efficient online sensor management.

Introduction

Many signal processing, ML, and DL applications involve tackling complex optimization problems that are difficult to solve analytically. Often, the objective function itself may not be in an analytical closed form, permitting function evaluations but not gradient assessments. Optimization corresponding to these types of problems falls into the category of ZO optimization with respect to black-box models, where explicit expressions of the gradients are difficult to compute or infeasible to obtain. ZO optimization methods are gradient-free counterparts of first-order (FO) optimization methods. They approximate the full gradients or stochastic gradients through function value-based gradient estimates. Interest in ZO optimization has grown rapidly during the years since the concept of gradient estimation by finite difference approximations was proposed during the 1950s and 1980s [1], [2].

It is worth noting that derivative-free methods for black-box optimization had been studied by the optimization community long before they had an impact on signal processing and ML/DL. Traditional derivative-free optimization (DFO) methods can be classified into two categories: direct search-based methods (DSMs) and model-based methods (MBMs) [3]–[6]. DSMs

Digital Object Identifier 10.1109/MSP.2020.3003837
Date of current version: 2 September 2020

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on July 18,2024 at 23:22:45 UTC from IEEE Xplore. Restrictions apply.

include the Nelder–Mead simplex method [7], coordinate search method [8], and pattern search method [9], to name a few. MBMs contain model-based descent methods [10] and trust region methods [11]. Evolutionary optimization is another class of generic population-based metaheuristic DFO algorithms that includes particle swarm optimization methods [12] and genetic algorithms [13]. Some Bayesian optimization (BO) methods [14] tackle black-box optimization problems by modeling the objective function as a Gaussian process (GP) that is learned from the history of function evaluations. However, learning an accurate GP model is computationally intensive.

Conventional DFO methods have two main shortcomings. First, they are often difficult to scale to large-size problems. For example, the off-the-shelf DFO solver COBYLA [15] only supports problems with a maximum of 2^{16} variables (SciPy Python library [16]), which is smaller than the size of a single ImageNet image [17]. Second, DFO methods lack a convergence rate analysis, and they may require a significant amount of effort to be customized to particular applications. ZO optimization has three main advantages over DFO: 1) the ease of implementation, with only a small modification of commonly used gradient-based algorithms; 2) computationally efficient approximations to derivatives when they are difficult to compute; and 3) comparable convergence rates to FO algorithms [18]–[21]. An illustrative example of ZO optimization versus FO optimization is shown in Figure 1.

ZO optimization has attracted increasing attention due to its success in solving emerging signal processing and ML/DL problems. First, ZO optimization serves as a powerful and practical tool for evaluating the adversarial robustness of ML/DL systems [22]. We note that the research in adversarial robustness has received increased attention during recent years. ZO-based methods for exploring the vulnerability of DL to black-box adversarial attacks are able to reveal the most susceptible features. Such ZO methods can be as effective as state-of-the-art white-box attacks, despite having access only to the inputs and outputs of the targeted deep neural networks (DNNs) [23], [24]. Moreover, ZO optimization can generate explanations

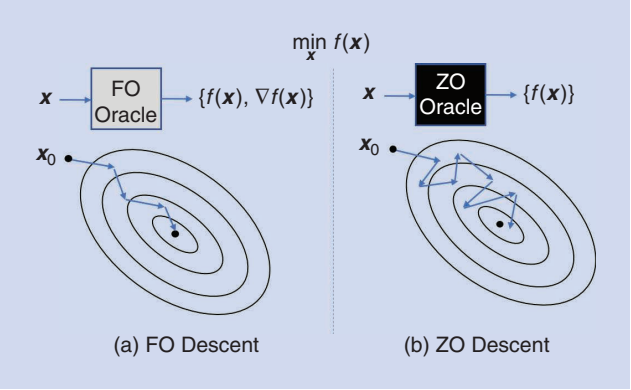


FIGURE 1. (a) FO optimization versus (b) ZO optimization. Here, the former solves the optimization problem $\min_{\mathbf{x}} f(\mathbf{x})$ with the white-box objective function f, and the latter solves the problem when f is a black-box function. Typically, ZO optimization has a slower convergence speed than FO optimization.

and provide interpretations of prediction results in a gradient-free and model-agnostic manner [25]. Furthermore, ZO optimization can also be used to solve automated ML (AutoML) problems, e.g., automated backpropagation in DL, where the gradients with respect to ML pipeline configuration parameters are intractable [26]. ZO optimization is also applicable to ML applications where the full gradient must be kept private [27]. In addition, ZO optimization provides computationally efficient alternatives for second-order optimization, such as robust training by curvature regularization [28], metalearning [29], transfer learning [30], and online network management [27].

Gradient estimation via ZO oracle

In this section, we provide an overview of gradient estimation techniques for optimization with a black-box objective function. The resulting gradient estimate forms the basis for constructing the descent direction used in ZO optimization algorithms. We categorize the ZO gradient estimates into two types, one-point and multipoint estimates, based on the number of queried function evaluations. As the number of function evaluations increases, a more accurate gradient estimate is expected but at the cost of increased query complexity.

One-point estimate

We start by the principles of randomized gradient estimation in the context of one-point estimation. Let $f(\mathbf{x})$ be a continuously differentiable objective function on a d-dimension variable $\mathbf{x} \in \mathbb{R}^d$. The one-point gradient estimate of f has the generic form

$$\hat{\nabla} f(\mathbf{x}) := \frac{\phi(d)}{\mu} f(\mathbf{x} + \mu \mathbf{u})\, \mathbf{u}, \tag{1}$$

where $\mathbf{u} \sim p$ is a random direction vector drawn from a certain distribution p, which is typically chosen as either the standard multivariate normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ [19] or the multivariate uniform distribution $\mathcal{U}(\mathcal{S}(0,1))$ on a unit sphere centered at zero with a radius of one [20] and where $\mu > 0$ is a perturbation radius (also called a *smoothing parameter*) and $\phi(d)$ denotes a certain dimension-dependent factor related to the choice of the distribution p. If $p = \mathcal{N}(\mathbf{0}, \mathbf{I})$, then $\phi(d) = 1$; if $p = \mathcal{U}(\mathcal{S}(0,1))$, then $\phi(d) = d$.

The rationale behind (1) is that it is an unbiased estimate of the gradient of the smoothed version of f across a random perturbation $\mathbf{u} \sim p'$ with smoothing parameter μ,

$$f_\mu(\mathbf{x}) := \mathbb{E}_{\mathbf{u} \sim p'}[f(\mathbf{x} + \mu \mathbf{u})], \tag{2}$$

where p' is specified as $\mathcal{N}(\mathbf{0}, \mathbf{I})$ if $p = \mathcal{N}(\mathbf{0}, \mathbf{I})$ in (1) or the multivariate uniform distribution on a unit ball $\mathcal{U}(\mathcal{B}(0,1))$ if $p = \mathcal{U}(\mathcal{S}(0,1))$ in (1). The unbiasedness of (1) with respect to $\nabla f_\mu(\mathbf{x})$ is assured by [19], [31]

$$\mathbb{E}_{\mathbf{u} \sim p}[\hat{\nabla} f(\mathbf{x})] = \nabla f_\mu(\mathbf{x}). \tag{3}$$

The meaning of (3) can be elucidated by considering the scalar case $d = 1$. Given $p = \mathcal{U}(\mathcal{S}(0,1))$, applying the fundamental theorem of calculus to (2) yields $\nabla f_\mu(x) = d/dx \int_{-\mu}^{\mu} 1/2 f(x+u)\, du =$

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on July 18,2024 at 23:22:45 UTC from IEEE Xplore. Restrictions apply.

$1/2\mu[f(x+\mu)-f(x-\mu)]$, which is equal to $\mathbb{E}_{u\sim p}[\hat{\nabla} f(x)]$ from (1). Although the one-point estimate (1) is unbiased with respect to the gradient of the smoothed function $\nabla f_\mu(\mathbf{x})$, it is a biased approximation of the true gradient $\nabla f(\mathbf{x})$. Furthermore, the one-point estimate is not commonly used in practice since it suffers from a high variance, defined as $\mathbb{E}\left[\left\| \hat{\nabla} f(\mathbf{x}) - \nabla f_\mu(\mathbf{x}) \right\|_2^2\right]$, which slows convergence [20].

Multipoint ZO estimate

A natural extension of (1) is the directional derivative approximation (two-point estimate) [19], [21]

$$\hat{\nabla} f(\mathbf{x}) := \frac{\phi(d)}{\mu}[f(\mathbf{x}+\mu\mathbf{u}) - f(\mathbf{x})]\mathbf{u}, \tag{4}$$

which satisfies the unbiasedness condition (3) for any $\mathbf{u}$ such that $\mathbb{E}_{\mathbf{u}\sim p}[\mathbf{u}] = \mathbf{0}$. The mean squared approximation error of the gradient estimate (4) with respect to the true gradient $\nabla f(\mathbf{x})$ obeys [31], [32]

$$\mathbb{E}[\| \hat{\nabla} f(\mathbf{x}) - \nabla f(\mathbf{x}) \|_2^2] = O(d) \| \nabla f(\mathbf{x}) \|_2^2 + O\left(\frac{\mu^2 d^3 + \mu^2 d}{\phi(d)}\right), \tag{5}$$

where we adopt the big-O notation to highlight the dominant factors d and μ affecting the gradient estimation error. It is worth noting that the (coordinate-wise) two-point ZO estimate for finding the optimum of a regression function was initially proposed during the 1950s [1]. This gradient estimation technique was further studied during the 1980s in the context of simultaneous perturbation stochastic approximation [2], [33].

The approximation error (5) of the two-point estimate in (4) provides several insights. First, the gradient estimate gets better as the smoothing parameter μ becomes smaller. However, in a practical system, if μ is too small, then the function difference could be dominated by system noise, and it may fail to represent the differential [32], [34]. Thus, careful selection of the smoothing parameter μ is important for the convergence of ZO optimization methods. Second, different from the FO stochastic gradient estimate, the ZO gradient estimate yields a dimension-dependent variance that increases as $O(d)\|\nabla f(\mathbf{x})\|_2^2$. Thus, the variance cannot be reduced, even if $\mu \to 0$. As a result, some recent work has focused on the design of variance-reduced gradient estimates.

Minibatch sampling is the most commonly used approach to reduce the variance of ZO gradient estimates [21], [27]. Instead of using a single random direction, the average of b independent, identically distributed samples $\{\mathbf{u}_i\}_{i=1}^b$ drawn from p are used for gradient estimation, leading to the multipoint estimate

$$\hat{\nabla} f(\mathbf{x}) := \frac{\phi(d)}{\mu} \sum_{i=1}^{b} [(f(\mathbf{x}+\mu\mathbf{u}_i) - f(\mathbf{x}))\mathbf{u}_i], \tag{6}$$

with the approximation error [31]

$$O\left(\frac{d}{b}\right) \| \nabla f(\mathbf{x}) \|_2^2 + O\left(\frac{\mu^2 d^3}{\phi(d) b}\right) + O\left(\frac{\mu^2 d}{\phi(d)}\right). \tag{7}$$

In (7), the first two terms correspond to the reduced variance of the two-point estimate $\mathbb{E}[\| \hat{\nabla} f(\mathbf{x}) - \nabla f_\mu(\mathbf{x}) \|_2^2]$ due to the drawing of b random direction vectors. And the third term, independent of b, corresponds to the approximation error due to the gradient of the smoothed function $\| \nabla f(\mathbf{x}) - \nabla f_\mu(\mathbf{x}) \|_2^2$.

When the number of function evaluations reaches the problem dimension d in (6), then instead of using randomized directions $\{\mathbf{u}_i\}_{i=1}^d$, one can employ the deterministic coordinate-wise gradient estimate $(1/\mu)\sum_{i=1}^d [(f(\mathbf{x}+\mu\mathbf{e}_i) - f(\mathbf{x}))\mathbf{e}_i]$, which yields a lower approximation error that is on the order of $O(d\mu^2)$ [1], [31], [34]. Here $\mathbf{e}_i \in \mathbb{R}^d$ denotes the ith elementary basis vector, with one at the ith coordinate and zeros elsewhere. In practice, the multipoint gradient estimate (6) is usually implemented with $2 \le b \le d$. The previously introduced multipoint estimates are computed by using forward differences of function values. An alternative is the central difference variant that uses $(f(\mathbf{x}+\mu\mathbf{u}) - f(\mathbf{x}-\mu\mathbf{u}))$, where $\mathbf{u}$ can be either randomized or deterministic [1], [2]. These central difference estimates have similar approximation errors to the forward difference estimates [31], [34], [35].

ZO optimization algorithms

In this section, we present a unified algorithmic framework covering many commonly used ZO optimization methods. We provide a thorough overview of existing algorithms in different problem settings and delve into the factors that influence their convergence.

The generic form of the ZO algorithm

Let us consider a stochastic optimization problem

$$\min_{\mathbf{x}\in\chi} f(\mathbf{x}) := \mathbb{E}_{\boldsymbol{\xi}}[f(\mathbf{x};\boldsymbol{\xi})], \tag{8}$$

where $\mathbf{x} \in \mathbb{R}^d$ are optimization variables, $\mathcal{X}$ is a closed convex set, f is a possibly nonconvex objective function, and $\boldsymbol{\xi}$ is a certain random variable that captures stochastic data samples or noise. If $\boldsymbol{\xi}$ obeys a uniform distribution across n empirical samples $\{\boldsymbol{\xi}_i\}_{i=1}^n$, then problem (8) reduces to a finite-sum formulation with the objective function $f(\mathbf{x}) = (1/n)\sum_{i=1}^n f(\mathbf{x};\boldsymbol{\xi}_i)$. And if $\mathcal{X} = \mathbb{R}^d$, then problem (8) simplifies to the unconstrained optimization problem.

Most ZO optimization methods mimic their FO counterparts and involve three steps, shown in Algorithm 1: gradient estimation (A1), descent direction computation (A2), and point updating (A3). Without a loss of generality, we specify (A1) as a variant of (6) built on a minibatch of empirical samples $\{\boldsymbol{\xi}_j\}_{j\in\Omega_t}$,

$$\hat{\mathbf{g}}_t = \phi(\{f(\mathbf{x}_t;\boldsymbol{\xi}_j)\}_{j\in\Omega_t}^t, \alpha) = \frac{1}{|\Omega_t|}\sum_{j\in\Omega_t} \hat{\nabla} f(\mathbf{x}_t;\boldsymbol{\xi}_j), \tag{9}$$

where $\hat{\nabla} f(\mathbf{x}_t;\boldsymbol{\xi}_j)$ is given by (6) as the gradient of the function $f(\cdot;\boldsymbol{\xi})$, and $|\Omega_t|$ denotes the cardinality of the set of minibatch samples at iteration t. Next, we elaborate on the descent direction computation and the point-updating step used in many ZO algorithms.

ZO algorithms for unconstrained optimization

We consider the ZO stochastic gradient descent (ZO–SGD) method [18], ZO sign-based SGD (ZO-signSGD) [36], ZO

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on July 18,2024 at 23:22:45 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1. Generic form of ZO optimization.

Initialize $\mathbf{x}_0 \in \mathcal{X}$, gradient estimation operation $\phi(\cdot)$, descent direction updating operation $\psi(\cdot)$, number of iterations T, and learning rate $\eta_t > 0$ at iteration t,

for $t = 1, 2, \ldots, T$ **do**

1) Gradient estimation:

$$\hat{\mathbf{g}}_t = \phi\left(\{f(\mathbf{x}_t; \boldsymbol{\xi}_j)\}_{j \in \Omega_t}^t\right), \tag{A1}$$

where Ω_t denotes a set of minibatch stochastic samples used at iteration t,

2) Descent direction computation:

$$\mathbf{m}_t = \psi\left(\{\hat{\mathbf{g}}_i\}_{i=1}^t\right), \tag{A2}$$

3) Point updating:

$$\mathbf{x}_t = \Pi_{\mathcal{X}}(\mathbf{x}_{t-1}, \mathbf{m}_t, \eta_t), \tag{A3}$$

where $\Pi_{\mathcal{X}}$ denotes a point-updating operation subject to the constraint $\mathbf{x} \in \mathcal{X}$.

end for

stochastic variance reduced gradient (ZO-SVRG) method [32], [37]–[39], and ZO Hessian-based (ZO-Hess) algorithm [40], [41]. These algorithms employ the same point-updating rule (A3)

$$\mathbf{x}_t = \mathbf{x}_{t-1} - \eta_t \mathbf{m}_t.$$

However, they adopt different strategies to form the descent direction $\mathbf{m}_t$ in (A2).

- *ZO-SGD* [18]: The descent direction $\mathbf{m}_t$ is set as the current gradient estimate $\mathbf{m}_t = \hat{\mathbf{g}}_t$. Note that the ZO-SGD becomes the ZO stochastic coordinate descent (ZO-SCD) method [34] as the coordinate-wise gradient estimate is used. Moreover, if the full batch of stochastic samples is used, then the ZO-SGD becomes the ZO-GD [19].
- *ZO-signSGD* [36]: The descent direction $\mathbf{m}_t$ is given by the sign of the current gradient estimate $\mathbf{m}_t = \operatorname{sign}(\hat{\mathbf{g}}_t)$, where $\operatorname{sign}(\cdot)$ denotes the element-wise sign operation. Using the sign operation scales down the (coordinate-wise) estimation errors [36], [42].
- *ZO-SVRG* [32], [37]–[39]: The descent direction $\mathbf{m}_t$ is formed by combining $\hat{\mathbf{g}}_t$ with a control variate of the reduced variance, $\mathbf{m}_t = \hat{\mathbf{g}}_t - \mathbf{c}_t + \mathbb{E}_{\boldsymbol{\xi}}[\mathbf{c}_t]$, where $\mathbf{c}_t$ denotes a control variate, which is commonly given by a gradient estimate evaluated at $\mathbf{x}_{t-1}$ of the entire data set of n empirical samples.
- *ZO-Hess* [40]: The descent direction $\mathbf{m}_t$ incorporates the approximate Hessian $\hat{\mathbf{H}}_t$ [40], $\mathbf{m}_t = \hat{\mathbf{H}}_t^{-1/2}\hat{\mathbf{g}}_t$, where $\hat{\mathbf{H}}_t$ is constructed either by the second-order Gaussian Stein's identity [41] or the diagonalization-based Hessian approximation [40]. The former approach was used in [41] to develop the ZO stochastic cubic regularized Newton (ZO-SCRN) method.

ZO algorithms for constrained optimization

We next present the ZO projected SGD (ZO-PSGD) [43], ZO stochastic mirror descent (ZO-SMD) [21], ZO stochastic conditional gradient (ZO-SCG) algorithm [41], [44], and ZO adaptive momentum method (ZO-AdaMM) [45] for constrained optimization. The aforementioned algorithms, with the exception of the ZO-AdaMM, specify the descent direction (A2) as the current gradient estimate $\mathbf{m}_t = \hat{\mathbf{g}}_t$. Their key difference lies in how they implement the point-updating step (A3).

- *ZO-PSGD* [43]: By letting $\Pi_{\mathcal{X}}$ be the Euclidean distance-based projection operation, the point-updating step (A3) is given by $\mathbf{x}_t = \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - (\mathbf{x}_{t-1} - \eta_t \mathbf{m}_t)\|_2^2$.
- *ZO-SMD* [21]: Upon defining a Bregman divergence $D_h(\mathbf{x}, \mathbf{y})$ with respect to a strongly convex and differentiable function h, $D_h(\mathbf{x}, \mathbf{y}) = h(\mathbf{x}) - h(\mathbf{y}) - (\mathbf{x} - \mathbf{y})^T \nabla h(\mathbf{y})$, the point-updating step (A3) is given by $\mathbf{x}_t = \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} \mathbf{m}_t^T \mathbf{x} + (1/\eta_t) D_h(\mathbf{x}, \mathbf{x}_t)$. For example, if $h(\mathbf{x}) = (1/2)\|\mathbf{x}\|_2^2$, then $D_h(\mathbf{x}, \mathbf{y}) = (1/2)\|\mathbf{x} - \mathbf{y}\|_2^2$, and $\mathbf{x}_t = \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - (\mathbf{x}_{t-1} - \eta_t \mathbf{m}_t)\|_2^2$, which reduces to the ZO-PSGD.
- *ZO-SCG* [41], [44]: The point-updating step (A3) calls for a linear minimization oracle [41], $\mathbf{z}_t = \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} \mathbf{m}_t^T \mathbf{x}$, and forms a feasible point update through the linear combination $\mathbf{x}_t = (1 - \eta_t)\mathbf{x}_{t-1} + \eta_t \mathbf{z}_t$. Similar algorithms, known as the *ZO Frank–Wolfe*, were also developed in [46] and [47].
- *ZO-AdaMM* [45]: Different from the ZO-PSGD, ZO-SMD, and ZO-SCG, the ZO-AdaMM adopts a momentum-type descent direction (rather than the current estimate $\hat{\mathbf{g}}_t$), an adaptive learning rate (rather than the constant rate η_t), and a projection operation under Mahalanobis distance (rather than Euclidean distance). The ZO-AdaMM can strike a balance between the convergence speed and the accuracy. However, it requires tuning extra algorithmic hyperparameters in addition to the learning rate and smoothing parameters [45], [48].

ZO optimization in complex settings

Here, we review ZO algorithms for composite optimization, minimum–maximum (min–max) optimization, distributed optimization, and structured high-dimensional optimization.

ZO composite optimization

Consider the following problem with a smooth-plus-nonsmooth composite objective function:

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimize}} \quad f(\mathbf{x}) + g(\mathbf{x}), \tag{10}$$

where f is a black-box smooth function (possibly nonconvex) and g is a white-box nonsmooth regularization function. The form of problem (10) arises in many sparsity-promoted applications, e.g., adversarial attack generation [49] and online sensor management [27]. The ZO proximal SGD (ZO-ProxSGD) algorithm [43] and ZO (stochastic) alternating direction method of multipliers (ZO-ADMM) [27], [50], [51] were developed to solve (10). We remark that (8) can also be cast as (10) by introducing the indicator function of the constraint $\mathbf{x} \in \mathcal{X}$ in the objective of (8) by letting $g(\mathbf{x}) = 0$ if $\mathbf{x} \in \mathcal{X}$ and ∞ if $\mathbf{x} \notin \mathcal{X}$.

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on July 18,2024 at 23:22:45 UTC from IEEE Xplore. Restrictions apply.

ZO min–max optimization

By min–max, we mean that the problem is a composition of inner maximization and outer minimization of the objective function

$$\min_{\mathbf{x}\in\mathcal{X}}\max_{\mathbf{y}\in\mathcal{Y}} \quad f(\mathbf{x},\mathbf{y}), \tag{11}$$

where $\mathbf{x}\in\mathbb{R}^{d_x}$ and $\boldsymbol{y}\in\mathbb{R}^{d_y}$ are optimization variables (for the ease of notation, let $d_x = d_y = d$), f is a black-box objective function, and $\mathcal{X}$ and $\mathcal{Y}$ are compact convex sets. One motivating application behind problem (11) is the design of a black-box poisoning attack [52], where the attacker deliberately influences the training data (by injecting poisoned samples) to manipulate the results of a black-box predictive model. To solve the problem posed in (11), the work in [52] and [53] presented efficient ZO min–max algorithms for stochastic and deterministic bilevel optimization with nonconvex outer minimization across $\mathbf{x}$ and strongly concave inner maximization across $\mathbf{y}$. The authors proved the convergence rate to be sublinear when gradient estimation is integrated with alternating (projected) SG descent/ascent methods.

ZO distributed optimization

Consider the minimization of a network cost given by the sum of local objective functions $\{f_i\}$ at multiple agents

$$\begin{aligned}\underset{\{\mathbf{x}_i\in\mathcal{X}\}}{\text{minimize}} \quad & \sum_{i=1}^{N} f_i(\mathbf{x}_i)\\ \text{subject to} \quad & \mathbf{x}_i = \mathbf{x}_j, \quad \forall j\in\mathcal{N}(i).\end{aligned} \tag{12}$$

Here, $\mathcal{N}(i)$ denotes the set of neighbors of agent/node i, and the underlying network/graph is connected; namely, there exists a path between every pair of distinct nodes. Some recent works have started to tackle the distributed optimization problem (12) with black-box objectives. In [54], a distributed Kiefer–Wolfowitz-type ZO algorithm was proposed, along with convergence analysis, for the case that the objective functions $\{f_i\}$ are strongly convex. In [55] and [56], the ZO distributed (sub)gradient algorithm and the ZO distributed mirror descent algorithm were developed for nonsmooth convex optimization. In [57] and [58], the convergence of consensus-based distributed ZO algorithms was established for nonconvex (unconstrained) optimization.

Structured high-dimensional optimization

Compared to FO algorithms, ZO algorithms typically suffer from a slowdown (proportional to the problem size d) in convergence. Thus, some recent works attempt to mitigate this limitation when solving high-dimensional (large d) problems. The work in [59] explored the functional sparsity structure, under which the objective function f depends on only a subset of d coordinates. This assumption also implies the gradient sparsity, which enabled the development of a least-absolute-shrinkage-and-selection-operator-based algorithm for gradient estimation and eventually yielded a polylogarithmic dependence on d when f is convex. And the work in [41] established the convergence rate of the ZO-SGD, which depends on d only polylogarithmically under the assumption of gradient sparsity. In addition, the work in [60] proposed a direct search-based algorithm that yields the convergence rate that is polylogarithmically dependent on dimensionality for any monotone transform of a smooth and strongly convex objective with a low-dimensional structure. That is, the objective function $f(\mathbf{x})$ is supported on a low-dimensional manifold $\mathcal{X}$. Another work, [61], studied the problem of ZO optimization on Riemannian manifolds and proposed algorithms that depend only on the intrinsic dimension of the manifold by using ZO Riemannian gradient estimates.

Convergence rates

We first elaborate on the criteria used to analyze the convergence rate of ZO algorithms under different problem settings.

1) *Convex optimization*: The convergence error is measured by the optimality gap of function values $\mathbb{E}[f(\mathbf{x}_T) - f(\mathbf{x}^*)]$ for a convex objective f, where $\mathbf{x}_T$ denotes the updated point at the final iteration T, $\mathbf{x}^*$ denotes the optimal solution, and the expectation is taken across the full probability space, e.g., random gradient approximation and stochastic sampling.
2) *Online convex optimization*: The cumulative regret [62] is typically used in place of the optimality gap, namely, $\mathbb{E}[\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}}\sum_{t=1}^{T} f_t(\mathbf{x})]$ for an online convex cost function f_t, e.g., $f_t(\cdot) = f(\cdot;\boldsymbol{\xi}_t)$ in (8).
3) *Unconstrained nonconvex optimization*: The convergence is evaluated by the FO stationary condition in terms of the squared gradient norm $(1/T)\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2]$ for the nonconvex objective f. Since FO stationary points could be saddle points of a nonconvex optimization problem, the second-order stationary condition is also used to ensure the local optimality of an FO stationary point (namely, escaping saddle points) [41], [63]. The work in [41] and [63] focused on stochastic optimization and deterministic optimization, respectively.
4) *Constrained nonconvex optimization*: The criterion for convergence is commonly determined by detecting a sufficiently small squared norm of the gradient mapping [43], [64], $P_{\mathcal{X}}(\mathbf{x}_t, \nabla f(\mathbf{x}_t), \eta_t) := (1/\eta_t)[\mathbf{x}_t - \Pi_{\mathcal{X}}(\mathbf{x}_t - \eta_t\nabla f(\mathbf{x}_t))]$, where the notation follows (A3). Here, $P_{\mathcal{X}}(\mathbf{x}_t, \nabla f(\mathbf{x}_t), \eta_t)$ can naturally be interpreted as the projected gradient, which offers a feasible update from the previous point $\mathbf{x}_t$. The Frank–Wolfe duality gap is another commonly used convergence criterion [41], [44], [46], [47]; it is given by $\max_{\mathbf{x}\in\mathcal{X}}\langle \mathbf{x} - \mathbf{x}_t, -\nabla f(\mathbf{x}_t)\rangle$. It is always nonnegative and becomes zero if and only if $\mathbf{x}_t\in\mathcal{X}$ is a stationary point.

More generally, given a convergence measure $\mathcal{M}(\cdot)$, $\mathbf{x}$ is called an *ϵ-optimal solution* if $\mathcal{M}(\mathbf{x})\le\epsilon$. The convergence error is typically expressed as a function of the number of iterations T, relating the convergence rate to the iteration complexity. Since the convergence analysis of existing ZO algorithms varies under different problem domains and algorithmic parameter settings, we compare in Table 1, from five perspectives, the convergence performance of ZO algorithms covered in this section: the problem structure, type of gradient estimates, smoothing parameter, convergence error, and function query complexity.

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on July 18,2024 at 23:22:45 UTC from IEEE Xplore. Restrictions apply.

Table 1. The comparison of different ZO algorithms in problem setting, gradient estimation, smoothing parameter, convergence error, and the function query complexity.

Method	Problem Structure	Gradient Estimation	Smoothing Parameter μ	Convergence Error (T Iterations)	Query Complexity (T Iterations)
ZO-GD [19]	NC, UnCons	Two-point GauGE*	$O\left(\frac{1}{\sqrt{dT}}\right)$	$O\left(\frac{d}{T}\right)$	$O(\lvert\mathcal{D}\rvert T)^+$
ZO-SGD [18]	NC, UnCons	Two-point GauGE	$O\left(\frac{1}{d\sqrt{T}}\right)$	$O\left(\frac{\sqrt{d}}{\sqrt{T}}\right)$	$O(T)$
ZO-SCD [34]	NC, UnCons	Two-point CooGE*	$O\left(\frac{1}{\sqrt{T}}+\frac{1}{(dT)^{1/4}}\right)$	$O\left(\frac{\sqrt{d}}{\sqrt{T}}\right)$	$O(T)$
ZO-signSGD [36]	NC, UnCons	b-point UniGE*	$O\left(\frac{1}{\sqrt{dT}}\right)$	$O\left(\frac{\sqrt{d}}{\sqrt{T}}+\frac{\sqrt{d}}{\sqrt{b}}\right)^4$	$O(bT)$
ZO-SVRG [32]	NC, UnCons	b-point UniGE	$O\left(\frac{1}{\sqrt{dT}}\right)$	$O\left(\frac{d}{T}+\frac{1}{b}\right)$	$O(\lvert\mathcal{D}\rvert s+bsm)T=sm$
ZO-Hess [40]	SC, UnCons	b-point GauGE	$O\left(\frac{1}{d}\right)$	$O(e^{-bT/d})$	$O(bT)$
ZO-ProxSGD/ZO-PSGD [43]	NC, Cons	b-point GauGE	$O\left(\frac{1}{\sqrt{dT}}\right)$	$O\left(\frac{d^2}{bT}+\frac{d}{b}\right)$	$O(bT)$
ZO-SMD [21]	C, Cons	Two-point GauGE	$O\left(\frac{1}{dt}\right)$	$O\left(\frac{\sqrt{d}}{\sqrt{T}}\right)$	$O(T)$
ZO-SCG [41]	NC, Cons	b-point GauGE	$O\left(\frac{1}{\sqrt{d^3T}}\right)$	$O\left(\frac{1}{\sqrt{T}}+\frac{d\sqrt{T}}{b}\right)$	$O(bT)$
ZO-AdaMM [45]	NC, Cons	b-point GauGE	$O\left(\frac{1}{\sqrt{dT}}\right)$	$O\left(\frac{d}{T}+\frac{d}{b}\right)$	$O(b^2T)$
ZO-ADMM [27]	C, Composite	b-point UniGE	$O\left(\frac{1}{d^{1.5}t}\right)$	$O\left(\frac{\sqrt{d}}{\sqrt{bT}}\right)$	$O(bT)$
ZO-min–max [52]	NC, Cons	b-point UniGE	$O\left(\frac{1}{d\sqrt{T}}\right)$	$O\left(\frac{1}{T}+\frac{d}{b}\right)$	$O(b^2T)$
Dist-ZO [58]	NC, UnCons	2-point UniGE	$\frac{1}{\sqrt{dt}}$	$O\left(\frac{\sqrt{d}}{\sqrt{T}}\right)$	$O(T)$
ZO-SCRN [41]	NC, UnCons	b-point UniGE	$\frac{1}{d^{5/2}T^{1/3}}$	$O\left(\frac{1}{T^{4/3}}\right)b=O(dT^{4/3})$	$O(bT)$

Dist: distributed; NC: nonconvex; C: convex; SC: strongly convex; UnCons: unconstrained; Cons: constrained; Composite: composite optimization.
*GauGE and UniGE represent the gradient estimates using random direction vectors drawn from the Gaussian distribution $\mathcal{N}(\mathbf{0}, I)$ and the uniform distribution $\mathcal{U}(\mathcal{S}(0,1))$, respectively. CooGE represents the coordinate-wise partial derivative estimate.
+Here, $\mathcal{D}$ denotes the entire data set.

Application: Adversarial example generation

In this section, we present the application of ZO optimization to the generation of prediction-evasive adversarial examples to fool DL models. Adversarial examples, also known as *evasion attacks*, are inputs corrupted with imperceptible adversarial perturbations (to be designed) toward misclassification (namely, predictions that are different from the true image labels) [22], [65]. Most studies on adversarial vulnerability of DL have been restricted to the white-box setting, where the adversary has complete access to, and knowledge of, the target system (e.g., DNNs) [22], [65]. However, it is often the case that the internal states/configurations and the operating mechanism of DL systems are not revealed to the practitioners [e.g., the Google Cloud Vision application programming interface (API)]. This gives rise to the problem of black-box adversarial attacks [23], [24], [66]–[69], where the only mode of interaction of the adversary with the system is via the submission of inputs and receiving the corresponding predicted outputs.

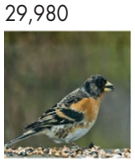
More formally, let $\mathbf{z}$ denote a legitimate example and $\mathbf{z}' := \mathbf{z} + \mathbf{x}$ denote an adversarial example, with the adversarial perturbation $\mathbf{x}$. Given the learned ML/DL model $\boldsymbol{\theta}$, the problem of adversarial example generation can be cast as an optimization problem of the following generic form [49], [65]:

$$\begin{aligned} &\underset{\mathbf{x}\in\mathbb{R}^d}{\text{minimize}} \quad f(\mathbf{z}';\boldsymbol{\theta})+\lambda g(\mathbf{x}) \\ &\text{subject to} \quad \|\mathbf{x}\|_\infty \le \epsilon, \quad \mathbf{z}' \in [0,1]^d, \end{aligned} \tag{13}$$

where $f(\mathbf{z}';\boldsymbol{\theta})$ denotes the (black-box) attack loss function for fooling the model $\boldsymbol{\theta}$ using the perturbed input $\mathbf{z}'$ (see [23] for a specific formulation); $g(\mathbf{x})$ is a regularization function that penalizes the sparsity or the structure of the adversarial perturbations, e.g., group sparsity in [70]; $\lambda \ge 0$ is a regularization parameter; the ℓ_∞ norm enforces similarity between $\mathbf{z}'$ and $\mathbf{z}$; and the input space of ML/DL systems is normalized to $[0,1]^d$. If $\lambda \ne 0$, then (13) is in the form of composite optimization, and the ZO-ADMM is a well-suited optimizer. If $\lambda = 0$, a solution to problem (13) is known as a *black-box* ℓ_∞

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on July 18,2024 at 23:22:45 UTC from IEEE Xplore. Restrictions apply.

Table 2. The comparison of various ZO methods for generating untargeted adversarial attacks against the Inception v3 model across five ImageNet images.

True Label	Brambling	Cannon	Pug Dog	Fly	Armadillo	Balloon
Perturbed image **(ZO-PSGD)**						
Prediction label	Goldfinch	Plow	Bucket	Longicorn	Croquet ball	Parachute
ℓ_2 distortion	35.2137	87.7304	72.3397	111.068	172.719	35.9330
Number of queries	7,640	150	130	50	210	5,830
Perturbed image **(ZO-SMD)**						
Prediction label	Goldfinch	Plow	Bucket	Cicada	Croquet ball	Parachute
ℓ_2 distortion	8.9708	26.0126	20.9504	30.0968	45.097	10.9023
Number of queries	29,980	350	260	140	570	15,830
Perturbed image **(ZO-AdaMM)**						
Prediction label	Goldfinch	Plow	Bucket	Cicada	Croquet ball	Parachute
ℓ_2 distortion	8.0502	5.7359	4.5753	4.4456	6.3149	7.7405
Number of queries	53,300	1,710	790	540	2,780	32,900
Perturbed image **(ZO-NES)**						
Prediction label	Goldfinch	Plow	Bucket	Cicada	Croquet ball	Parachute
ℓ_2 distortion	54.9956	34.5852	28.7035	28.9158	40.1483	51.7116
Number of queries	22,110	1,080	830	430	2,280	15,340

Row 1 shows true labels of given images. Rows 2–4 display results obtained using the ZO-SMD, which include perturbed images, corresponding prediction labels, the ℓ_2 norm of the perturbations, and the number of queries when achieving the first successful black-box attack. A similar explanation holds for the other rows, except that different ZO methods are used.

attack [24], which can be obtained using ZO methods for constrained optimization.

In Table 2, we present black-box ℓ_∞ attacks with respect to five ImageNet images against the Inception v3 model [71]. The adversarially perturbed images are obtained from four ZO methods, including the ZO-PSGD, ZO-SMD, ZO-AdaMM, and ZO-natural evolutionary strategy (NES) (a projected version of the ZO-signSGD that is used in practice [24]). We demonstrate the attack performance of different ZO algorithms in terms of the ℓ_2 norm of the generated perturbations and the number of queries needed to achieve a first successful black-box attack. As we can see, the ZO-PSGD typically has the fastest speed of converging to a valid adversarial example, while the ZO-AdaMM has the best convergence accuracy in terms of the smallest distortion required to fool the neural network.

Application: Online sensor management

ZO optimization is also advantageous when it is difficult to compute the FO gradient of an objective function. Online sensor management provides an example of such a scenario [27], [72]. The sensor selection for parameter estimation is a fundamental problem in smart grids, communication systems, and wireless sensor networks [73]. The goal is to seek the optimal tradeoff between sensor activations and the estimation accuracy through a time period.

We consider the cumulative loss for online sensor selection [27]

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^d}{\text{minimize}} \quad & \frac{1}{T}\sum_{t=1}^{T}\left[-\log\det\left(\sum_{i=1}^{d} x_i \mathbf{a}_{i,t}\mathbf{a}_{i,t}^T\right)\right] \\ \text{subject to} \quad & \mathbf{1}^T\mathbf{x} = m_0, \quad \mathbf{0}\le\mathbf{x}\le\mathbf{1}, \end{aligned} \tag{14}$$

where $\mathbf{x}\in\mathbb{R}^d$ is the optimization variable, d is the number of sensors, $\mathbf{a}_{i,t}\in\mathbb{R}^n$ is the observation coefficient of sensor i at time t, and m_0 is the number of selected sensors. The objective function of (14) can be interpreted as the log determinant of the error covariance matrix associated with the maximum likelihood estimator for parameter estimation [74]. The constraint $\mathbf{0}\le\mathbf{x}\le\mathbf{1}$ is a relaxed convex hull of the Boolean constraint $\mathbf{x}\in\{0,1\}^m$, which encodes whether or not a sensor is selected.

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on July 18,2024 at 23:22:45 UTC from IEEE Xplore. Restrictions apply.

Conventional methods, such as projected gradient (FO) and interior-point (second-order) algorithms, can be used to solve problem (14). However, both methods involve the calculation of inverses of large matrices that are necessary to evaluate the gradient of the cost function. The matrix inversion step is usually a bottleneck while acquiring the gradient information in high dimensions, and it is particularly problematic in the online optimization setting. Since (14) involves mixed equality and inequality constraints, it has been shown [27] that the ZO-ADMM is an effective ZO optimization method for circumventing the computational bottleneck.

In Figure 2, we compare the performance of the ZO-ADMM and that of the FO-ADMM [75] for sensor selection. In Figure 2(a), we demonstrate the primal–dual residuals in the ADMM against the number of iterations. As we can see, the ZO-ADMM has a slower convergence rate than the FO-ADMM, and it approaches the accuracy of the FO-ADMM as the number of iterations increases. In Figure 2(b), we show the mean square error (MSE) of the parameter estimation using a different number of selected sensors m_0 in (14). As we can see, the ZO-ADMM yields almost the same MSE as the FO-ADMM in the context of parameter estimation using m_0 activated sensors, determined by the hard thresholding of continuous sensor selection schemes, i.e., solutions of problem (14) obtained from the ZO-ADMM and the FO-ADMM.

Other recent applications

In this section, we discuss some other recent applications of ZO optimization in signal processing and ML.

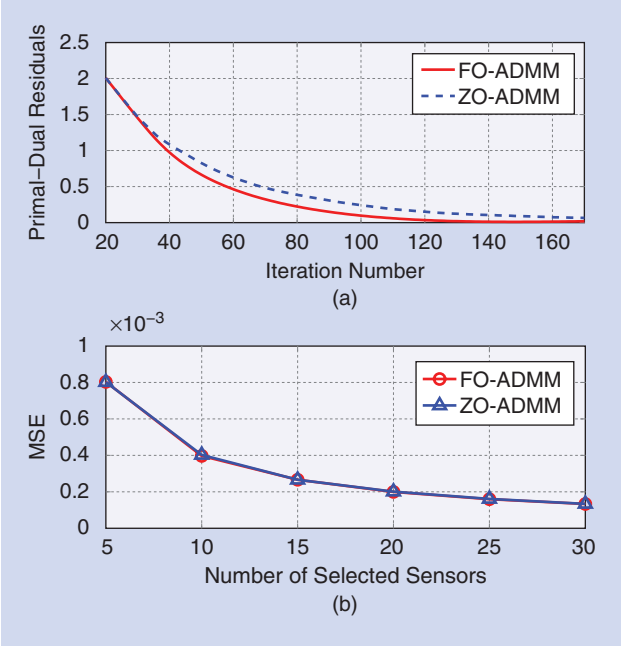


FIGURE 2. The comparison between the ZO-ADMM and the FO-ADMM for solving the sensor selection problem (14). (a) ADMM primal–dual residuals versus the number of iterations. (b) The MSE of activated sensors for parameter estimation versus the total number of selected sensors.

Model-agnostic constrastive explanations

Explaining the decision-making process of a complex ML model is crucial to many ML-assisted, high-stakes applications, such as job hiring, financial loan applications, and judicial sentences. When generating local explanations for the prediction of an ML model on a specific data sample, one common practice is to leverage the information of its input gradient for sensitivity analysis of the model prediction. For ML models that do not have explicit functions for computing input gradients, such as access-limited APIs and rule-based systems, ZO optimization enables the generation of local explanations using model queries without the knowledge of the gradient information. Moreover, even when the input gradient can be obtained via ML platforms, such as TensorFlow and PyTorch, the gradient computation is platform specific. In this case, ZO optimization has the advantage of alleviating platform dependency when developing multiplatform explanation methods, as it only depends on model inference results.

Here, we apply ZO optimization to generating contrastive explanations [76] for two ML applications: handwritten digit classification and loan approval. Contrastive explanations consist of two components derived from a given data sample for explaining the model prediction, i.e., a pertinent positive (PP) that is minimally and sufficiently present to keep the same prediction of the original input sample, and a pertinent negative (PN) that is minimally and necessarily absent to alter the model prediction. The process of finding the PP and the PN is formulated as a sparsity-driven and data-perturbation-based optimization problem guided by the model prediction outcomes [25], which can be solved by ZO optimization methods. Figure 3 shows the contrastive explanations generated from black-box neural network models by the ZO-GD using the objective functions in [25]. For handwritten digit classification, the PP identifies a subset of pixels such that their presence is minimally sufficient for model prediction. Moreover, the PN identifies a subset of pixels such that their absence is minimally necessary for altering model prediction. The PP and the PN together constitute a contrastive explanation for interpreting model prediction. Similarly, for the credit loan approval task trained on the Fair, Isaac, and Company Explainable ML Challenge data set [91] based on a neural network model, the PN generated for an applicant (Alice) can be used to explain how the model would alter the recommendation from "denial" to "approval" based on Alice's loan application profile (please refer to https://aix360.mybluemix.net for more details).

Policy search in reinforcement learning

Reinforcement learning aims to determine, given a state, which action to take (or policy to make) to maximize a reward. One of the most popular policy search approaches is the model-free policy search, where an agent learns parameterized policies from sampled trajectories without needing to learn the model of the underlying dynamics. Model-free policy search updates the parameters such that trajectories with higher rewards are more likely to be obtained when following the updated policy [77]. Traditional policy search methods, such as REINFORCE [78],

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on July 18,2024 at 23:22:45 UTC from IEEE Xplore. Restrictions apply.

rely on randomized exploration in the action space to compute an estimated direction of improvement. These methods (referred to as *policy gradient methods*) then leverage the FO information of the policy (or Jacobian) to update its parameters to maximize the reward. Note that the chance of finding a sequence of actions resulting in high total reward decreases as the horizon length increases, and thus policy gradient methods often exhibit a high variance and result in a large sample complexity [79].

To alleviate these problems, ZO policy search methods, which directly optimize across the policy parameter space, have emerged as an alternative to the policy gradient. More specifically, ZO policy search methods seek to directly optimize the total reward in the space of the parameters by employing finite-difference methods to compute estimates of the gradient with respect to the policy parameters [77], [80]–[83]. These methods are fully ZO; i.e., they do not exploit FO information of the policy, reward, and dynamics. Interestingly, it has been observed that although policy gradient methods leverage more information, ZO policy search methods often perform better empirically. In particular, the work in [82] characterized the convergence rate of ZO policy optimization when applied to linear-quadratic systems. And the work in [83] theoretically showed that the complexity of exploration in the action space (using policy gradients) depends on both the dimensionality of the action space and the horizon length, while the complexity of exploration in the parameter space (using ZO methods) depends only on the dimensionality of the parameter space.

AutoML

The success of ML relies heavily on selecting the right pipeline algorithms for the problem at hand and on setting the hyperparameters. AutoML automates the process of model selection and hyperparameter optimization. It offers the advantages of producing simpler solutions, a faster creation of those solutions, and models that often outperform hand-designed ML models. One could view AutoML as the process of optimization of an unknown black-box function. Recently, several BO approaches have been proposed for AutoML [26], [84]. BO works by building a probabilistic surrogate via a GP for the objective function and then using an acquisition function defined from this surrogate to decide where to sample. However, BO suffers from a computational bottleneck: an internal FO solver is required to determine the parameters of the GP model by maximizing the log marginal likelihood of the current function evaluations at each iteration of the BO. The FO solver is slow due to the difficulty of computing the gradient of the log-likelihood function with respect to the parameters of the GP. To circumvent this difficulty, the ZO optimization algorithm can be used to determine the hyperparameters and thus to accelerate the BO in AutoML [26]. In the context of metalearning, ZO optimization has also been leveraged to obviate the need for determining computationally intensive high-order derivatives during metatraining [29]. Finally, we note that ZO optimization can be integrated with learning to optimize, which models the optimizer through a trainable DNN-based metalearner [85], [86].

Open questions and discussions

Although there has been a great deal of progress on the design, theoretical analysis, and applications of ZO optimization, many questions and challenges still remain.

ZO optimization with nonsmooth objectives

There exists a gap between the theoretical analysis of ZO optimizers and practical ML/DL applications with nonsmooth objectives, where the former usually requires the smoothness of the objective function. There are two possible means of relaxing the smoothness assumption. First, the randomized smoothing technique ensures that the convolution of two functions is at least as smooth as the smoothest of the two original functions. Thus, f_μ is smooth, even if f is nonsmooth, in (2). This motivates the technique of double randomization that approximates a subgradient of a nonsmooth objective function [21], where an extra randomized perturbation is introduced to prevent drawing points from nonsmooth regions of f. The downside of double randomization is the increase of function query complexity. Second, a model-based trust region method can be leveraged to approximate the subgradient/gradient using linear or quadratic interpolation [6], [31]. This leads to the general approach of gradient estimation without imposing extra assumptions on the objective function. However, it increases the computation cost due to the need to solve nested regression problems.

ZO optimization with black-box constraints

The current work on ZO optimization is restricted to black-box objective functions with white-box constraints. In the presence

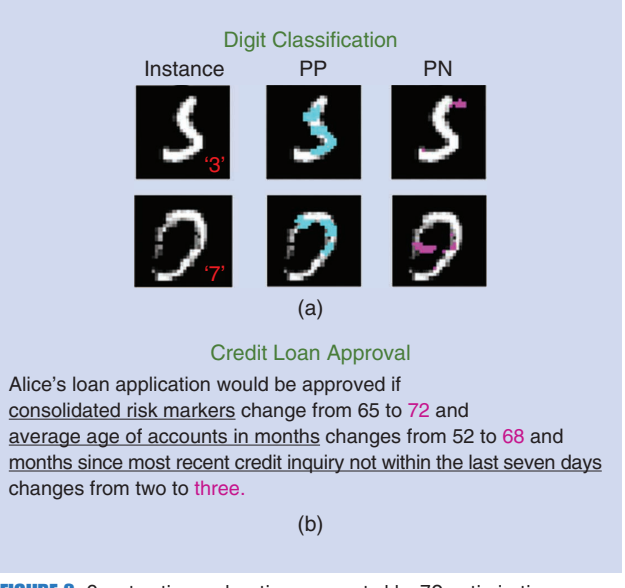


FIGURE 3. Constrastive explanations generated by ZO optimization methods. (a) For handwritten digit classification, the red-digit class on the corner of an input sample shows the model prediction of the instance. The pixels highlighted in cyan are the PP supporting the original prediction. The pixels highlighted by the purple color are the PN that will alter the model prediction when added to the original instance. (b) For the credit loan application, the PN of an applicant (Alice) is used to explain the necessary modifications on a subset of the original features to change the model prediction from "denial" to "approval."

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on July 18,2024 at 23:22:45 UTC from IEEE Xplore. Restrictions apply.

of black-box constraints, the introduction of barrier functions (instead of constraints) [87] in the objective could be a potential solution. One could also employ the method of multipliers to reformulate black-box constraints as regularization functions in the objective function [26].

ZO optimization for privacy-preserving distributed learning

To protect the sensitive information of data in the context of distributed learning, it is common to add "noise" (randomness) into gradients of individual cost functions of agents, a process that is known as the *message-perturbing privacy strategy* [88]. The level of privacy is often evaluated by differential privacy (DP). A high degree of DP prevents an adversary from gaining meaningful personal information related to any individuals. Similarly, ZO optimization also conceals the gradient information and enables the use of noisy gradient estimates that are constructed from function values. Thus, one interesting question is, Can ZO optimization be designed with privacy guarantees? In a more general sense, it would be worthwhile to examine what roles ZO optimization plays in the privacy-preserving and Byzantine-tolerant federated-learning setting.

ZO optimization and automatic differentiation

Automatic differentiation (AD) provides a way for efficiently and accurately evaluating derivatives of numeric functions, which are expressed as computer programs [89]. The backpropagation algorithm used for training neural networks can be regarded as a specialized instance of AD under the reverse mode. AD decomposes the derivative of the complex function into subderivatives of constituent operations through the chain rule. When a subderivative is infeasible or difficult to compute, ZO gradient estimation techniques could be integrated with AD. In particular, when the high-order derivatives (beyond-gradient) are required, e.g., model-agnostic metalearning [90], ZO optimization could help to overcome the derivative bottleneck.

ZO optimization for discrete variables

Many ML and signal processing tasks involve handling discrete variables, such as texts, graphs, sets, and categorical data. In addition to the technique of relaxation to continuous values, it is worthwhile to explore and design ZO algorithms that directly operate in discrete domains.

Tight convergence rates of ZO methods

Although the optimal rate for ZO unconstrained convex optimization was studied in [21], there remain many open questions about seeking the optimal rates, or associated tight lower bounds, for general cases of ZO constrained nonconvex optimization.

Conclusions

In this survey article, we discussed variants of ZO gradient estimators and focused on their statistical modeling as this leads to general ZO algorithms. We also provided an extensive comparison of ZO algorithms and discussed their iteration and function query complexities. Furthermore, we presented numerous emerging applications of ZO optimization in signal processing and ML. Finally, we highlighted some unsolved research challenges in ZO optimization research and presented some promising future research directions.

Acknowledgment

The work of Sijia Liu, Pin-Yu Chen, and Gaoyuan Zhang was supported by the MIT-IBM Watson AI Lab. The work of Bhavya Kailkhura was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory, under contract DE-AC52-07NA27344.

Authors

Sijia Liu (sijia.liu@ibm.com) received his Ph.D. degree (with the All-University Doctoral Prize) in electrical and computer engineering from Syracuse University, New York, in 2016. He was a postdoctoral research fellow at the University of Michigan, Ann Arbor, prior to joining IBM Research. He is currently a research staff member at the Massachusetts Institute of Technology–IBM Watson Artificial Intelligence Lab. His research interests include optimization for deep learning and adversarial machine learning. He received the Best Student Paper Award (third place) at ICASSP 2017 and was among the seven finalists for the Best Student Paper Award at the 2013 IEEE Asilomar Conference on Signals, Systems, and Computers. He is a Member of the IEEE.

Pin-Yu Chen (pin-yu.chen@ibm.com) received his Ph.D. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, in 2016. He is a research staff member at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York. He is also the chief scientist of the Rensselaer–IBM Artificial Intelligence (AI) Research Collaboration and principal investigator of ongoing Massachusetts Institute of Technology–IBM Watson AI Lab projects. His recent research concerns adversarial machine learning and the robustness of neural networks. His long-term research vision to build trustworthy machine learning systems. He received the 2017 Conference on Neural Information Processing Systems Best Reviewer Award and the 2010 IEEE Global Communications Conference Gold Best Paper Award. He is a Member of the IEEE.

Bhavya Kailkhura (kailkhura1@llnl.gov) is a research staff member at the Lawrence Livermore National Laboratory (LLNL), California. His research interests include adversarial machine learning, robust statistics and control, and high-dimensional data analytics. He was the runner-up for the Best Student Paper Award at the 2014 IEEE Asilomar Conference on Signals, Systems, and Computers and an IEEE Signal Processing Society travel grant recipient. He received the 2017 All-University Doctoral Prize from Syracuse University for superior achievement in completed dissertations. He was also a recipient of the 2017 LLNL Deputy Director for Science and Technology Excellence in Publication Award. He is a Member of the IEEE.

Gaoyuan Zhang (gaoyuan.zhang@ibm.com) received his B.S. and M.S. degrees in electrical engineering from Shanghai

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on July 18,2024 at 23:22:45 UTC from IEEE Xplore. Restrictions apply.

Jiao Tong University, China, and Columbia University, New York, in 2014 and 2016, respectively. He is currently a research engineer at the Massachusetts Institute of Technology–IBM Watson Artificial Intelligence Lab. His research interests include adversarial machine learning and parallel computing.

Alfred O. Hero III (hero@umich.edu) is the John H. Holland Distinguished University Professor of Electrical Engineering and Computer Science and the R. Jamison and Betty Williams Professor of Engineering at the University of Michigan, Ann Arbor. He is a section editor of *SIAM Journal on Mathematics of Data Science*, a senior editor of *IEEE Journal on Selected Topics in Signal Processing*, and a member of the editorial board of *Harvard Data Science Review*. He is a Fellow of the IEEE and the Society for Industrial and Applied Mathematics.

Pramod K. Varshney (varshney@syr.edu) received his B.S., M.S., and Ph.D. degrees from the University of Illinois at Urbana-Champaign in 1972, 1974, and 1976, respectively. Since 1976, he has been with Syracuse University, where he is currently a distinguished professor of electrical engineering and computer science. He received the IEEE Judith A. Resnik Award in 2012, a doctor of engineering degree honoris causa from Drexel University in 2014, the Electrical and Computer Engineering Distinguished Alumni Award from the University of Illinois in 2015, and the International Society of Information Fusion's Yaakov Bar-Shalom Award for a Lifetime of Excellence in Information Fusion in 2018. He is a Life Fellow of the IEEE.

References

[1] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *Ann. Math. Statist.*, vol. 23, no. 3, pp. 462–466, 1952. doi: 10.1214/aoms/1177729392.

[2] J. C. Spall, "A stochastic approximation technique for generating maximum likelihood parameter estimates," in *Proc. American Control Conf.*, 1987, pp. 1161–1167. doi: 10.23919/ACC.1987.4789489.

[3] J. Nocedal and S. Wright, *Numerical Optimization*, Berlin: Springer-Verlag, 2006.

[4] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, vol. 8. Philadelphia: SIAM, 2009.

[5] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: A review of algorithms and comparison of software implementations," *J. Global Optimiz.*, vol. 56, no. 3, pp. 1247–1293, 2013. doi: 10.1007/s10898-012-9951-y.

[6] J. Larson, M. Menickelly, and S. M. Wild, "Derivative-free optimization methods," *Acta Numerica*, vol. 28, pp. 287–404, May 2019. doi: 10.1017/S0962492919000060.

[7] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, 1965. doi: 10.1093/comjnl/7.4.308.

[8] E. Fermi and N. Metropolis, "Numerical solution of a minimum problem," Los Alamos Scientific Lab., Univ. of California, Oakland, Tech. Rep. LA-1492, 1952. [Online]. Available: https://hdl.handle.net/2027/mdp.39015086487645

[9] V. Torczon, "On the convergence of the multidirectional search algorithm," *SIAM J. Optimiz.*, vol. 1, no. 1, pp. 123–145, 1991. doi: 10.1137/0801010.

[10] D. M. Bortz and C. T. Kelley, "The simplex gradient and noisy optimization problems," in *Computational Methods for Optimal Design and Control*, J. Borggaard, J. Burns, E. Cliff, and S. Schereck, Eds. Berlin: Springer-Verlag, 1998, pp. 77–90.

[11] A. R. Conn, N. I. Gould, and P. L. Toint, *Trust Region Methods*, vol. 1. Philadelphia: SIAM, 2000.

[12] A. I. F. Vaz and L. N. Vicente, "PSwarm: A hybrid solver for linearly constrained global derivative-free optimization," *Optimiz. Meth. Softw.*, vol. 24, nos. 4–5, pp. 669–685, 2009. doi: 10.1080/10556780902909948.

[13] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Mach. Learn.*, vol. 3, pp. 95–99, 1988. doi: 10.1023/A:1022602019183.

[14] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, 2016. doi: 10.1109/JPROC.2015.2494218.

[15] M. J. D. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in Optimization and Numerical Analysis*, S. Gomez and J. P. Hennart, Eds. Berlin: Springer-Verlag, 1994, pp. 51–67.

[16] E. Jones, T. Oliphant, and P. Peterson, "SciPy: Open source scientific tools for Python," ScienceOpen.com, 2001. [Online]. Available: https://www.scienceopen.com/document?vid=ab12905a-8a5b-43d8-a2bb-defc771410b9

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.

[18] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM J. Optimiz.*, vol. 23, no. 4, pp. 2341–2368, 2013. doi: 10.1137/120880811.

[19] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Found. Comput. Math.*, vol. 17, no. 2, pp. 527–566, 2015. doi: 10.1007/s10208-015-9296-2.

[20] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proc. 16th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2005, pp. 385–394.

[21] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal rates for zero-order convex optimization: The power of two function evaluations," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2788–2806, 2015. doi: 10.1109/TIT.2015.2409256.

[22] I. J. Goodfellow, J. Shlens, and C. Szegedy, Explaining and harnessing adversarial examples. 2014. [Online]. Available: arXiv:1412.6572

[23] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop Artificial Intelligence and Security*, 2017, pp. 15–26. doi: 10.1145/3128572.3140448.

[24] A. Ilyas, K. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *Proc. Int. Conf. Machine Learning (ICML)*, July 2018, pp. 2137–2146.

[25] A. Dhurandhar, T. Pedapati, A. Balakrishnan, P.-Y. Chen, K. Shanmugam, and R. Puri, Model agnostic contrastive explanations for structured data. 2019. [Online]. Available: arXiv:1906.00117

[26] S. Liu, P. Ram, D. Vijaykeerthy, D. Bouneffouf, G. Bramble, H. Samulowitz, D. Wang, A. Conn et al., "An ADMM based framework for AutoML pipeline configuration," in *Proc. AAAI Conf.*, 2019, pp. 4892–4899.

[27] S. Liu, J. Chen, P.-Y. Chen, and A. O. Hero, "Zeroth-order online ADMM: Convergence analysis and applications," in *Proc. AISTATS*, 2018, vol. 84, pp. 288–297.

[28] S.-M. Moosavi-Dezfooli, A. Fawzi, J. Uesato, and P. Frossard, "Robustness via curvature regularization, and vice versa," in *Proc. IEEE CVPR*, 2019, pp. 9078–9086.

[29] X. Song, W. Gao, Y. Yang, K. Choromanski, A. Pacchiano, and Y. Tang, "ES-MAML: Simple Hessian-free meta learning," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2020. [Online]. Available: https://openreview.net/forum?id=S1exA2NtDB

[30] Y.-Y. Tsai, P.-Y. Chen, and T.-Y. Ho, "Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources," in *Proc. Int. Conf. Machine Learning (ICML)*, 2020, pp. 3642–3653.

[31] A. S. Berahas, L. Cao, K. Choromanski, and K. Scheinberg, A theoretical and empirical comparison of gradient approximations in derivative-free optimization. 2019. [Online]. Available: arXiv:1905.01332

[32] S. Liu, B. Kailkhura, P.-Y. Chen, P. Ting, S. Chang, and L. Amini, "Zeroth-order stochastic variance reduction for nonconvex optimization," in *Proc. 32nd Conf. Neural Information Processing Systems (NeurIPS)*, 2018, pp. 1–11.

[33] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Autom. Control*, vol. 37, no. 3, pp. 332–341, 1992. doi: 10.1109/9.119632.

[34] X. Lian, H. Zhang, C.-J. Hsieh, Y. Huang, and J. Liu, "A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order," in *Proc. 30th Conf. Neural Information Processing Systems (NeurIPS)*, 2016, pp. 3054–3062.

[35] O. Shamir, "An optimal algorithm for bandit and zero-order convex optimization with two-point feedback," *JMLR*, vol. 18, no. 52, pp. 1–11, 2017.

[36] S. Liu, P.-Y. Chen, X. Chen, and M. Hong, "signSGD via zeroth-order Oracle," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2019.

[37] B. Gu, Z. Huo, and H. Huang, Zeroth-order asynchronous doubly stochastic algorithm with variance reduction. 2016. [Online]. Available: arXiv:1612.01425

[38] L. Liu, M. Cheng, C.-J. Hsieh, and D. Tao, Stochastic zeroth-order optimization via variance reduction method. 2018. [Online]. Available: arXiv:1805.11811

[39] K. Ji, Z. Wang, Y. Zhou, and Y. Liang, "Improved zeroth-order variance reduced algorithms and analysis for nonconvex optimization," in *Proc. Int. Conf. Machine Learning (ICML)*, 2019, vol. 97, pp. 3100–3109.

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on July 18,2024 at 23:22:45 UTC from IEEE Xplore. Restrictions apply.

[40] H. Ye, Z. Huang, C. Fang, C. J. Li, and T. Zhang, Hessian-aware zeroth-order optimization for black-box adversarial attack. 2018. [Online]. Available: arXiv:1812.11377

[41] K. Balasubramanian and S. Ghadimi, Zeroth-order nonconvex stochastic optimization: Handling constraints, high-dimensionality, and saddle-points. 2019. [Online]. Available: arXiv:1809.06474

[42] M. Cheng, S. Singh, P.-Y. Chen, S. Liu, and C.-J. Hsieh, "Sign-OPT: A query-efficient hard-label adversarial attack," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2020.

[43] S. Ghadimi, G. Lan, and H. Zhang, "Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization," *Math. Program.*, vol. 155, nos. 1–2, pp. 267–305, 2016. doi: 10.1007/s10107-014-0846-1.

[44] K. Balasubramanian and S. Ghadimi, "Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates," in *Proc. 32nd Conf. Neural Information Processing Systems (NeurIPS)*, 2018, pp. 3455–3464.

[45] X. Chen, S. Liu, K. Xu, X. Li, X. Lin, M. Hong, and D. Cox, "ZO-AdaMM: Zeroth-order adaptive momentum method for black-box optimization," in *Proc. 33nd Conf. Neural Information Processing Systems (NeurIPS)*, 2019, pp. 7202–7213.

[46] J. Chen, J. Yi, and Q. Gu, "A Frank–Wolfe framework for efficient and effective adversarial attacks," in *Proc. AAAI*, 2020, pp. 3486–3494.

[47] A. K. Sahu, M. Zaheer, and S. Kar, "Towards gradient free and projection free stochastic optimization," in *Proc. AISTATS*, 2019, pp. 3468–3477.

[48] X. Chen, S. Liu, R. Sun, and M. Hong, "On the convergence of a class of Adam-type algorithms for non-convex optimization," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2019.

[49] P. Zhao, S. Liu, P.-Y. Chen, N. Hoang, K. Xu, B. Kailkhura, and X. Lin, "On the design of black-box adversarial examples by leveraging gradient-free optimization and operator splitting method," in *Proc. ICCV*, 2019, pp. 121–130.

[50] X. Gao, B. Jiang, and S. Zhang, "On the information-adaptive variants of the ADMM: An iteration complexity perspective," *J. Sci. Comput.* vol. 76, July 2018, pp. 327–363. doi: 10.1007/s10915-017-0621-6.

[51] F. Huang, S. Gao, S. Chen, and H. Huang, "Zeroth-order stochastic alternating direction method of multipliers for nonconvex nonsmooth optimization," in *Proc. 28th Int. Joint Conf. Artificial Intelligence (IJCAI)*, 2019, pp. 2549–2555.

[52] S. Liu, S. Lu, X. Chen, Y. Feng, K. Xu, A. Al-Dujaili, M. Hong, and U.-M. O'Reilly, "Min–max optimization without gradients: Convergence and applications to adversarial ML," in *Proc. Int. Conf. Machine Learning (ICML)*, 2020, pp. 1365–1377.

[53] Z. Wang, K. Balasubramanian, S. Ma, and M. Razaviyayn, "Zeroth-order algorithms for nonconvex minimax problems with improved complexities. 2020. [Online]. Available: arXiv:2001.07819

[54] A. K. Sahu, D. Jakovetic, D. Bajovic, and S. Kar, "Distributed zeroth order optimization over random networks: A Kiefer–Wolfowitz stochastic approximation approach," in *Proc. IEEE CDC*, 2018, pp. 4951–4958.

[55] D. Yuan, D. W. C. Ho, and S. Xu, "Zeroth-order method for distributed optimization with approximate projections," *IEEE Trans. Neural Netw*, vol. 27, no. 2, pp. 284–294, 2015. doi: 10.1109/TNNLS.2015.2480419.

[56] Z. Yu, D. W. C. Ho, and D. Yuan, Distributed randomized gradient-free mirror descent algorithm for constrained optimization. 2019. [Online]. Available: arXiv:1903.04157

[57] D. Hajinezhad, M. Hong, and A. Garcia, "Zone: Zeroth-order nonconvex multiagent optimization over networks," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 3995–4010, 2019. doi: 10.1109/TAC.2019.2896025.

[58] Y. Tang and N. Li, "Distributed zero-order algorithms for nonconvex multi-agent optimization," in *Proc. 57th Annu. Conf. Communication, Control, and Computing (Allerton)*, 2019, pp. 781–786. doi: 10.1109/ALLERTON.2019.8919690.

[59] Y. Wang, S. Du, S. Balakrishnan, and A. Singh, "Stochastic zeroth-order optimization in high dimensions," in *Proc. AISTATS*, Apr. 2018, vol. 84, pp. 1356–1365.

[60] D. Golovin, J. Karro, G. Kochanski, C. Lee, X. Song, and Q. Zhang, "Gradientless descent: High-dimensional zeroth-order optimization," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2020.

[61] J. Li, K. Balasubramanian, and S. Ma, "Zeroth-order optimization on Riemannian manifolds. 2020. [Online]. Available: arXiv:2003.11238

[62] E. Hazan, "Introduction to online convex optimization," *Found. Trends Optimiz.*, vol. 2, nos. 3–4, pp. 157–325, 2016. doi: 10.1561/2400000013.

[63] E.-V. Vlatakis-Gkaragkounis, L. Flokas, and G. Piliouras, "Efficiently avoiding saddle points with zero order methods: No gradients required," in *Proc. 33nd Conf. Neural Information Processing Systems (NeurIPS)*, 2019, pp. 10,066–10,077.

[64] S. J. Reddi, S. Sra, B. Poczos, and A. J. Smola, "Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization," in *Proc. 33nd Conf. Neural Information Processing Systems (NeurIPS)*, 2016, pp. 1145–1153.

[65] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Security and Privacy*, 2017, pp. 39–57. doi: 10.1109/SP.2017.49.

[66] C.-C. Tu, P. Ting, P.-Y. Chen, S. Liu, H. Zhang, J. Yi, C.-J. Hsieh, and S.-M. Cheng, "Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks," in *Proc. AAAI*, 2019. doi: 10.1609/aaai.v33i01.3301742.

[67] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2018.

[68] M. Cheng, T. Le, P.-Y. Chen, J. Yi, H. Zhang, and C.-J. Hsieh, "Query-efficient hard-label black-box attack: An optimization-based approach," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2019.

[69] P. Zhao, P.-Y. Chen, S. Wang, and X. Lin, "Towards query-efficient black-box adversary with zeroth-order natural gradient descent," in *Proc. AAAI*, 2020, pp. 1–8.

[70] K. Xu, S. Liu, P. Zhao, P.-Y. Chen, H. Zhang, Q. Fan, D. Erdogmus, Y. Wang et al., "Structured adversarial attack: Towards general implementation and better interpretability," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2019.

[71] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.

[72] T. Chen and G. B. Giannakis, "Bandit convex optimization for scalable and dynamic IoT management," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1276–1286, 2018. doi: 10.1109/JIOT.2018.2839563.

[73] A. O. Hero and D. Cochran, "Sensor management: Past, present, and future," *IEEE Sensors J.*, vol. 11, no. 12, pp. 3064–3075, 2011. doi: 10.1109/JSEN.2011.2167964.

[74] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, 2009. doi: 10.1109/TSP.2008.2007095.

[75] T. Suzuki, "Dual averaging and proximal gradient descent for online alternating direction multiplier method," in *Proc. 30th Int. Conf. Machine Learning (ICML)*, 2013, pp. 392–400.

[76] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, and P. Das, "Explanations based on the missing: Towards contrastive explanations with pertinent negatives," in *Proc. 32nd Conf. Neural Information Processing Systems (NeurIPS)*, 2018, pp. 592–603.

[77] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, 2013. doi: 10.1177/0278364913495721.

[78] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, 1992. doi: 10.1007/BF00992696.

[79] T. Zhao, H. Hachiya, G. Niu, and M. Sugiyama, "Analysis and improvement of policy gradient estimation," in *Proc. Advances Neural Information Processing Systems*, 2011, pp. 262–270.

[80] T. Salimans, H. Ho, X. Chen, S. Sidor, and I. Sutskever, Evolution strategies as a scalable alternative to reinforcement learning. 2017. [Online]. Available: arXiv:1703.03864

[81] F. Sehnke, C. Osendorfer, T. Rückstieß, A. Graves, J. Peters, and J. Schmidhuber, "Parameter-exploring policy gradients," *Neural Netw.*, vol. 23, no. 4, pp. 551–559, 2010. doi: 10.1016/j.neunet.2009.12.004.

[82] D. Malik, A. Pananjady, K. Bhatia, K. Khamaru, P. Bartlett, and M. Wainwright, "Derivative-free methods for policy optimization: Guarantees for linear quadratic systems," in *Proc. Machine Learning Research*, Apr. 16–18, 2019, vol. 89, pp. 2916–2925.

[83] A. Vemula, W. Sun, and J. A. Bagnell, "Contrasting exploration in parameter and action space: A zeroth-order optimization perspective," in *Proc. Machine Learning Research*, Apr. 16–18, 2019, vol. 89, pp. 2926–2935.

[84] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. 25th Conf. Neural Information Processing Systems (NeurIPS)*, 2012, pp. 2951–2959.

[85] Y. Ruan, Y. Xiong, S. Reddi, S. Kumar, and C.-J. Hsieh, "Learning to learn by zeroth-order oracle," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2020.

[86] Y. Cao, T. Chen, Z. Wang, and Y. Shen, "Learning to optimize in swarms," in *Proc. Advances Neural Information Processing Systems*, 2019, pp. 15,044–15,054.

[87] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[88] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *Proc. IEEE GlobalSIP*, 2013, pp. 245–248.

[89] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *JMLR*, vol. 18, no. 1, pp. 5595–5637, Jan. 2017.

[90] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Machine Learning (ICML)*, 2017, pp. 1126–1135.

[91] Fico Community, "Explainable machine learning challenge," 2020. [Online]. Available: https://community.fico.com/s/explainable-machine-learning-challenge

SP

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on July 18,2024 at 23:22:45 UTC from IEEE Xplore. Restrictions apply.