An Accelerated DFO Algorithm for Finite-sum Convex Functions

Yuwen Chen[1] **Antonio Orvieto**[1] **Aurelien Lucchi**[1]

Abstract

Derivative-free optimization (DFO) has recently gained a lot of momentum in machine learning, spawning interest in the community to design faster methods for problems where gradients are not accessible. While some attention has been given to the concept of acceleration in the DFO literature, existing stochastic algorithms for objective functions with a finite-sum structure have not been shown theoretically to achieve an accelerated rate of convergence. Algorithms that use acceleration in such a setting are prone to instabilities, making it difficult to reach convergence. In this work, we exploit the finite-sum structure of the objective in order to design a variance-reduced DFO algorithm that provably yields acceleration. We prove rates of convergence for both smooth convex and strongly-convex finite-sum objective functions. Finally, we validate our theoretical results empirically on several tasks and datasets.

1. Introduction

While gradient-based techniques are extremely popular in machine learning, there are applications where derivatives are too expensive to compute or might not even be accessible (black-box optimization). In such cases, an alternative is to use derivative-free methods which rely on function values instead of explicitly computing gradients. These methods date to the 1960's, including e.g. (Matyas, 1965; Nelder & Mead, 1965) and have recently gained more attention in machine learning in areas such as black-box adversarial attacks (Chen et al., 2017), reinforcement learning (Salimans et al., 2017), online learning (Bubeck et al., 2012), etc.

We focus our attention on optimizing finite-sum objective functions which are commonly encountered in machine learning and which can be formulated as:

$$\min_{x\in\mathbb{R}^d} \left[f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) \right], \qquad (1)$$

where each function $f_i : \mathbb{R}^d \to \mathbb{R}$ is convex and differentiable, but its derivatives are not directly accessible.

The problem of optimizing Eq. (1) has been addressed in a seminal work by (Nesterov & Spokoiny, 2011) who introduced a deterministic random[1] gradient-free method (RGF) using a two-point Gaussian random gradient estimator. The authors derived a rate of convergence for RGF for both convex and strongly-convex functions and they also introduced a variant with a provably accelerated rate of convergence. Subsequently, (Ghadimi & Lan, 2013) developed a stochastic variant of RGF, proving a nearly[2] optimal rate of convergence for convex functions.

In the field of first-order gradient-based methods, gradient descent has long been known to achieve a suboptimal convergence rate. In a seminal paper, Nesterov (1983) showed that one can construct an optimal – i.e. accelerated – algorithm that achieves faster rates of convergence for both convex and strongly-convex functions. Accelerated methods have attracted a lot of attention in machine learning, pioneering some popular momentum-based methods such as Adam (Kingma & Ba, 2014) which is commonly used to train deep neural networks. It therefore seems natural to ask whether *provably accelerated* methods can be designed in a derivative-free setting. While this question has been considered in a deterministic setting in (Nesterov & Spokoiny, 2011) as well as in a stochastic setting (Gorbunov et al., 2018; 2019), none of these works provably derived an accelerated rate of convergence for the finite-sum setting presented in Eq. (1).

The inherent difficulty of designing a stochastic algorithm with an accelerated rate of convergence is due to the instability of the momentum term (Allen-Zhu, 2017; Orvieto et al., 2019). One way to reduce instabilities is to rely on stochastic variance reduction (Johnson & Zhang, 2013; Defazio et al., 2014) which allows to achieve a linear rate of convergence for smooth and strongly convex functions in

[1] Computer Science Department, ETH Zürich, Switzerland. Correspondence to: Chen, Yuwen <aaronchenyuwen@gmail.com>, Orvieto, Antonio <antonio.orvieto@inf.ethz.ch>, Lucchi, Aurelien <aurelien.lucchi@inf.ethz.ch>.

Proceedings of the 37th International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

[1] The term random refers to the use of randomly sampled directions to estimate derivatives.

[2] For a precise definition, see (Ghadimi & Lan, 2013).

a gradient-based setting and then extended to nonconvex functions (Fang et al., 2018; Zhou et al., 2020). This rate is however still suboptimal (see e.g. (Lan & Zhou, 2018)) and there has been some recent effort to design an optimal variance-reduced method, including (Lin et al., 2015; Allen-Zhu, 2017; Lan & Zhou, 2018; Lan et al., 2019). We will build on the approach of (Lan et al., 2019) as it relies on less restrictive assumptions than other methods (see discussion in Section 2). We design a novel algorithm that estimates derivatives using the Gaussian smoothing approach of (Nesterov & Spokoiny, 2011) as well as the coordinate-wise approach of (Ji et al., 2019). We prove an accelerated rate of convergence for this algorithm in the case of convex and strongly-convex functions. Our experimental results on several datasets support our theoretical findings.

2. Related Work

Momentum in gradient-based setting. The first accelerated proof of convergence for the deterministic setting dates back to Polyak (1964) who proved a local linear rate of convergence for Heavy-ball (with constant momentum) for twice continuously differentiable, τ-strongly convex and L-smooth functions, with a constant of geometric decrease which is smaller than the one for gradient descent. A similar method, Nesterov's Accelerated Gradient (NAG), was introduced by (Nesterov, 1983). It achieves the optimal $\mathcal{O}(1/t^2)$ rate of convergence for convex functions and, with small modifications, an accelerated linear convergence rate for smooth and strongly-convex functions.

Prior work has shown that vanilla momentum methods lack stability in stochastic settings, where the evaluation of the gradients is affected by noise (see e.g. motivation in (Allen-Zhu, 2017)). Various solutions have been suggested in the literature, including using a regularized auxiliary objective that enjoys a better condition number than the original objective (Lin et al., 2015) or applying variance-reduction to obtain more stable momentum updates (Allen-Zhu, 2017; Lan et al., 2019). We here build on the Varag approach presented in (Lan et al., 2019) as it presents several advantages over prior work, including the ability to accelerate for smooth convex finite-sum problems as well as for strongly-convex problems without requiring an additional strongly-convex regularization term. Unlike Katyusha (Allen-Zhu, 2017), Varag also only requires the solution of one, rather than two, subproblems per iteration (discussed in (Lan et al., 2019)).

Variance-reduced DFO. In the finite-sum setting introduced in Eq. (1), variance-reduction techniques have become popular in machine learning. These techniques were originally developed for gradient-based methods and later adapted to the derivative-free setting in (Liu et al., 2018b) and (Liu et al., 2018a). Various improvements were later made in (Ji et al., 2019) such as allowing for a larger constant stepsize, as well as extending the analysis of (Fang et al., 2018) to a broader class of functions in a DFO setting. Finally, (Ji et al., 2019) introduced a coordinate-wise approach to estimate the gradients instead of the Gaussian smoothing method. This yields a more accurate estimation of the gradient at the price of a higher computational complexity. We rely on this technique to estimate the gradient at the pivot point in our analysis (see details in Section 4).

Momentum in DFO. As mentioned earlier, (Nesterov & Spokoiny, 2011) proved a rate in a deterministic setting. (Gorbunov et al., 2018), analyzes acceleration in a stochastic setting for general objective functions without explicitly exploiting any finite-sum structure (hence, assuming finite variance). Closer to our setting, (Gorbunov et al., 2019) analyzes a stochastic momentum DFO method based on the three point estimation technique proposed in (Bergou et al., 2019). Although they do theoretically analyze the convergence of such algorithms, they only prove a suboptimal rate of convergence instead of the accelerated rate of convergence derived in our work.

3. Background and Notation

In this paper we work in $\mathbb{R}^d$ with the standard Euclidean norm $\|\cdot\|$ and scalar product $\langle\cdot,\cdot\rangle$. Our goal, as stated in the introduction, is to minimize a convex function $f = \frac{1}{n}\sum_{i=1}^n f_i$ (with $f_i : \mathbb{R}^d \to \mathbb{R}$ for each $i = 1 \dots n$) without using gradient information. For our theoretical analysis, we will need the following standard assumption.

(A1) Each f_i is convex, differentiable and L-smooth[a]. Hence, also $f = \frac{1}{n}\sum_{i=1}^n f_i$ is convex and L-smooth.

[a] for all $x, y \in \mathbb{R}^d$ we have $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x-y\|$.

To estimate gradients, we will use and combine two different gradient estimation techniques, with different properties.

Estimation by Gaussian smoothing. This technique was first presented by Nesterov & Spokoiny (2011): let μ be the smoothing parameter, then $f_\mu : \mathbb{R}^d \to \mathbb{R}$, the smoothed version of f, is defined to be such that for all $x \in \mathbb{R}^d$

$$f_\mu(x) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} f(x+\mu u) e^{-\frac{1}{2}\|u\|^2} du.$$

In our setting, it is easy to see that (see discussion in the appendix) f_μ is still convex and L-smooth. Crucially, the integral in the definition of f_μ can be approximated by sampling random directions $u \in \mathbb{R}^d$ with a Gaussian distribution: $f_\mu(x) = \mathbb{E}_u[f(x+\mu u)]$. Note that, for $\mu \ll 1$, we have $f_\mu \approxeq f$.

The gradient of f_μ can be written as

$$\nabla f_\mu(x) = \mathbb{E}_u\left[\frac{(f(x+\mu u) - f(x))u}{\mu}\right] =: \mathbb{E}_u[g_\mu(x,u)].$$

A stochastic estimate of $g_\mu(x,u)$ using data-point i, which we denote by $g_\mu(x,u,i)$, can be then calculated as follows:

$$g_\mu(x,u,i) := \frac{f_i(x+\mu u) - f_i(x)}{\mu} u. \tag{2}$$

As we will see more in more detail in the next section, this cheap estimate is not appropriate if we seek a solid approximation. Fortunately, for such a task, we can use the coordinate-wise finite difference method.

Estimation by coordinate-wise finite difference. This approach, introduced in (Ji et al., 2019), estimates $\nabla f_i(x)$ without introducing a smoothing distortion, by directly evaluating the function value in each coordinate:

$$g_\nu(x,i) = \sum_{j=1}^{d} \frac{f_i(x+\nu \mathrm{e}_j) - f_i(x-\nu \mathrm{e}_j)}{2\nu} \mathrm{e}_j, \tag{3}$$

where e_j is the unit vector with only one non-zero entry 1 at its j^{th} coordinate. Note that, g_ν is d times more expensive to compute compared to g_μ. Besides, the coordinate-wise estimator of $\nabla f(x)$ is denoted as $g_\nu(x)$ where we remove the subscript i from Eq. (3).

4. Algorithm and Analysis

The method we propose is presented as Algorithm 1 (ZO-Varag), and is an adaptation of Varag (Lan et al., 2019) to the DFO setting. At it's core, ZO-Varag has the same structure of SVRG (Johnson & Zhang, 2013), but profits from the mechanism of accelerated stochastic approximation (Lan, 2012) combined with the two different zero-order gradient estimators presented in the last section. We highlight some important details below:

1. At the beginning of epoch s, we compute a *full* zero-order gradient $\tilde{g}^s$ at the *pivotal point* $\tilde{x}^{s-1}$ (i.e. the approximation of the solution provided by the preceding epoch). Since the accuracy in $\tilde{g}^s$ drastically influences the progress made in the epoch, we choose for its approximation the coordinate-wise estimator in Eq. (3). The estimate $\tilde{g}^s$ will then be used to perform T_s inner-iterations and to compute the next approximation $\tilde{x}^s$ to the problem solution.

2. Each inner-iteration (within an epoch) uses three sequences: $\{x_t\}, \{\underline{x}_t\}, \{\bar{x}_t\}$. Each of these sequences play an important role in the acceleration mechanics (see discussion in (Lan et al., 2019)).

3. In the inner loop, at iteration t, a cheap *variance-reduced gradient estimate* of $\nabla f_\mu(\underline{x}_t)$ is computed using the same technique as SVRG (Johnson & Zhang, 2013) combined with Gaussian smoothing (see Eq. (2))

$$G_t := g_\mu(\underline{x}_t, u_t, i_t) - g_\mu(\tilde{x}, u_t, i_t) + \tilde{g}^s, \tag{4}$$

   where u_t is a sample from a standard multivariate Gaussian, as required by the estimator definition and $\tilde{x}$ is the pivotal point for this inner loop (epoch s).

4. The choice of the additional parameters $\{T_s\}$, $\{\gamma_s\}$, $\{\alpha_s\}$, $\{p_s\}$, $\{\theta_t\}$ will be specified in the convergence theorems depending on each function class being considered (smooth, convex or strongly-convex).

4.1. Variance of the Gradient Estimators

From our discussion above, it is clear the following *error term* δ_t will heavily influence the analysis: the error in the estimation of the per-iteration direction G_t.

$$\delta_t := G_t - \nabla f_\mu(\underline{x}_t) \quad \text{(iteration gradient error).}$$

The expectation of δ_t, over u_t, i_t, is e^s defined below:

$$e^s := \tilde{g}^s - \nabla f_\mu(\tilde{x}) \quad \text{(pivotal gradient error).}$$

This is different from the standard SVRG as the pivotal gradient error e^s vanishes for gradient-based methods. The rest of this section is dedicated to the fundamental properties of this error.

Pivotal gradient error bound. Crucially, note that e^s is measured with respect to f_μ (the smoothed version of f). This provides consistency with δ_t, at the price of a well-behaved additional error coming from the smoothing distortion. A necessary first step to start our analysis is to bound $\|e^s\|^2$ *uniformly* by a problem-dependent constant E:

$$\begin{aligned}\|e^s\|^2 &\le \|\tilde{g}^s - \nabla f_\mu(\tilde{x}^{s-1})\|^2 \\ &\le 2\big[\|\tilde{g}^s - \nabla f(\tilde{x}^{s-1})\|^2 \\ &\quad + \|\nabla f(\tilde{x}^{s-1}) - \nabla f_\mu(\tilde{x}^{s-1})\|^2\big] \\ &\le 2L^2 d\nu^2 + \frac{\mu^2 L^2 (d+3)^3}{2} =: E,\end{aligned} \tag{5}$$

where the last inequality is a combination of Lemma 3 from (Ji et al., 2019) and Lemma 3 from (Nesterov & Spokoiny, 2011). Note that a similar inequality would not be possible by using an estimate obtained from sampling a random direction for pivotal $\tilde{g}^s$ — as the strength of the error would depend on the gradient magnitude, i.e. cannot be uniformly bounded (see Theorem 3 in (Nesterov & Spokoiny, 2011)).

Iteration gradient error bound. Unfortunately, as ZO-Varag is a DFO algorithm, the expectation of δ_t is *not vanishing* (in contrast to standard SVRG and Varag). However, the next lemma shows that it is still possible to bound the (trace of the) variance of G_t.

Algorithm 1 ZO-Varag

```
Require: $x^0 \in \mathbb{R}^d, \{T_s\}, \{\gamma_s\}, \{\alpha_s\}, \{p_s\}, \{\theta_t\}$.
  Set $\tilde{x}^0 = \bar{x}^0 = x^0$.
  for $s = 1, 2, \ldots, S$ do
    Option I : $\tilde{x} = \tilde{x}^{s-1}$
    Option II: $\tilde{x} = \bar{x}^{s-1}$
    Set $x_0 = x^{s-1}$, $\bar{x}_0 = \tilde{x}$.
    Pivotal ZO gradient $\tilde{g}^s = g_\nu(\tilde{x})$ using the coordinate-wise approach by Eq. (3).
    for $t = 1, 2, \ldots, T_s$ do
      $\underline{x}_t = \big[(1+\tau\gamma_s)(1-\alpha_s-p_s)\bar{x}_{t-1} + \alpha_s x_{t-1} + (1+\tau\gamma_s)p_s\tilde{x}\big]/[1+\tau\gamma_s(1-\alpha_s)]$.
      Pick $i_t \in \{1, \ldots, m\}$ uniformly and generate $u_t$ from $\mathcal{N}(0, I_d)$.
      $G_t = g_\mu(\underline{x}_t, u_t, i_t) - g_\mu(\tilde{x}, u_t, i_t) + \tilde{g}^s$.
      $x_t = [\gamma_s G_t + \gamma_s\tau\underline{x}_t + x_{t-1}]/[1+\gamma_s\tau]$    $\diamond = \arg\min_{x\in\mathbb{R}^d}\left\{\gamma_s\left[\langle G_t, x\rangle + \frac{\tau}{2}\|\underline{x}_t - x\|^2\right] + \frac{1}{2}\|x_{t-1} - x\|^2\right\}$.
      $\bar{x}_t = (1-\alpha_s-p_s)\bar{x}_{t-1} + \alpha_s x_t + p_s\tilde{x}$.
    end for
    Set $x^s = x_{T_s}, \bar{x}^s = \bar{x}_{T_s}$ and $\tilde{x}^s = \sum_{t=1}^{T_s}(\theta_t\bar{x}_t)/\big(\sum_{t=1}^{T_s}\theta_t\big)$.
  end for
Output: $\tilde{x}^S$
```

Lemma 1. *(Variance of G_t) Assume **(A1)**. Then, at any epoch $s \geq 1$ and iteration $1 \leq t \leq T_s$ we have*

$$\begin{aligned}&\mathbb{E}_{u_t,i_t|\mathcal{F}_{t-1}}\left[\|G_t - \mathbb{E}_{u_t,i_t|\mathcal{F}_{t-1}}[G_t]\|^2\right] \\ &\leq\ 18\mu^2L^2(d+6)^3 \\ &\quad + 8(d+4)L\big[f_\mu(\tilde{x}) - f_\mu(\underline{x}_t) - \langle\nabla f_\mu(\underline{x}_t), \tilde{x} - \underline{x}_t\rangle\big],\end{aligned} \tag{6}$$

and

$$\mathbb{E}_{u_t,i_t|\mathcal{F}_{t-1}}\big[\delta_t\big] = \tilde{g}^s - \nabla f_\mu(\tilde{x}) \neq 0, \tag{7}$$

where $\mathcal{F}_t$ is the σ-algebra generated by the previous iterates in the current epoch, i.e. $\mathcal{F}_t := \{u_t, i_t, \ldots, u_1, i_1\}$, and $\tilde{x}$ is the pivotal point for the epoch s.

Compared to Lemma 3 in (Lan et al., 2019), the bound on the variance of the gradient in the DFO case is dimension-dependent and has an extra error $18\mu^2L^2(d+6)^3$ due to Gaussian smoothing (it comes from the fact that we also take the expectation over u_t).

4.2. Analysis for Smooth and Convex Functions

For our final complexity result in this section to hold, we need all the sequences generated by Algorithm 1 to be bounded *in expectation.*

(A2$_{\boldsymbol{\mu}}$) Let $x_\mu^* \in \text{argmin}_x f_\mu(x)$ and consider the sequence of approximations $\{\tilde{x}^s\}$ returned by Algorithm 1. There exists a *finite* constant $Z < \infty$, potentially dependent on L and d, such that, for μ small enough,

$$\sup_{s\geq 0}\mathbb{E}\left[\|\tilde{x}^s - x_\mu^*\|\right] \leq Z.$$

Using an argument similar to (Gadat et al., 2018), it is possible to show that this assumption holds under the requirement that f is coercive, i.e. $f(x) \to \infty$ as $\|x\| \to \infty$. We are ready to state the main theorem of this section.

Theorem 2. *Assume **(A1)** and **(A2$_{\boldsymbol{\mu}}$)**. If we define $s_0 := \lfloor\log(d+4)n\rfloor + 1$ and set $\{T_s\}$, $\{\gamma_s\}$ and $\{p_s\}$ as*

$$T_s = \begin{cases}2^{s-1}, & s \leq s_0\\ T_{s_0}, & s > s_0\end{cases}, \qquad \gamma_s = \tfrac{1}{12(d+4)L\alpha_s}, \qquad p_s = \tfrac{1}{2}, \tag{8}$$

with

$$\alpha_s = \begin{cases}\frac{1}{2}, & s \leq s_0\\ \frac{2}{s-s_0+4}, & s > s_0\end{cases}. \tag{9}$$

If we set

$$\theta_t = \begin{cases}\frac{\gamma_s}{\alpha_s}(\alpha_s + p_s) & 1 \leq t \leq T_s - 1\\ \frac{\gamma_s}{\alpha_s} & t = T_s.\end{cases} \tag{10}$$

we obtain

$$\begin{aligned}&\mathbb{E}\big[f_\mu(\tilde{x}^s) - f_\mu^*\big] \leq \\ &\begin{cases}\dfrac{(d+4)D_0}{2^{s+1}} + 2\varsigma_1 + 3\varsigma_2, & 1 \leq s \leq s_0,\\ \dfrac{16D_0}{n(s-s_0+4)^2} + \delta_s\cdot(\varsigma_1 + \varsigma_2), & s > s_0,\end{cases}\end{aligned}$$

where $\varsigma_1 = \mu^2L(d+4)^2$, $\varsigma_2 = Z\sqrt{E}$, $\delta_s = \mathcal{O}(s - s_0)$ and D_0 is defined as

$$D_0 := \frac{2}{(d+4)}[f_\mu(x^0) - f_\mu(x_\mu^*)] + 6L\|x^0 - x_\mu^*\|^2, \tag{11}$$

where x_μ^* is any finite minimizer of f_μ.

Compared to the gradient-based analysis of (Lan et al., 2019), *two additional errors terms appear* because of the DFO framework: ς_1 is the error due to the Gaussian smooth estimation and ς_2 is an error due to the approximation made at the pivot point. It is essential to note that, in the bound for $s > s_0$, the error $\delta_s(\varsigma_1 + \varsigma_2)$ grows linearly with the number of epochs. In Corollary 3 we show how it is possible to tune our zeroth-order estimators to make these errors vanish by choosing sufficiently small smoothing parameters μ and ν, with an argument similar to the one used in Theorem 9 from Nesterov & Spokoiny (2011).

Based on Theorem 2, we obtain the following complexity bound.

Corollary 3. *Assume **(A1)** and **(A2$_\mu$)**. The total number $\bar{N}_\epsilon$ of function queries performed by Algorithm 1 to find a stochastic ϵ-solution, i.e. a point $\bar{x} \in \mathbb{R}^d$ s.t. $\mathbb{E}[f(\bar{x}) - f^*] \le \epsilon$, can be bounded by*

$$\bar{N}_\epsilon := \begin{cases} \mathcal{O}\left\{dn \log \frac{dD_0}{\epsilon}\right\}, & n \ge D_0/\epsilon \\ \mathcal{O}\left\{dn \log dn + d\sqrt{\frac{nD_0}{\epsilon}}\right\}, & n < D_0/\epsilon. \end{cases}$$

The reasoning behind the proof is quite standard in the DFO literature (Nesterov & Spokoiny, 2011), yet it contains some important ideas. Hence we include a proof sketch in order to give some additional intuition to the reader, who might wonder how to control the error terms from the theorem. Details can be found in the appendix.

Proof sketch. The procedure consists in deriving three bounds, that combined give the desired suboptimality ϵ:

1. In general, $x^*_\mu \neq x^* \in \text{argmin}_x f(x)$. Yet, Theorem 2 gives us a procedure to approximate x^*_μ. Hence we have to show that $f_\mu(\tilde{x}^s) - f_\mu(x^*_\mu)$ and $f(\tilde{x}^s) - f(x^*)$ are close enough. In particular, we have $f_\mu(\tilde{x}^s) - f_\mu(x^*_\mu) \ge f(\tilde{x}^s) - f(x^*) - \mu^2 Ld$, directly from Theorem 1 in (Nesterov & Spokoiny, 2011). Hence, we get the following sufficient condition: $\frac{\epsilon}{4} \ge \mu^2 Ld$. Therefore, the desired bound holds for $\mu^2 \le \frac{\epsilon}{4Ld}$. Since μ is a design parameter, *which does not affect the convergence speed but just the error*, we can choose it small enough so that this requirement is satisfied.

2. Next, assume $\varsigma_1 = \varsigma_2 = 0$ — we will deal with these terms at the end of the proof. We can then follow the proof of Theorem 1 in (Lan et al., 2019), but with the requirement of $\frac{\epsilon}{2}$ accuracy. This gives us the desired number $\bar{N}_\epsilon$ of function queries, which correspond to $\bar{s}_\epsilon$ epochs.

3. Last, we spend the last $\frac{\epsilon}{4}$ accuracy to bound the error terms, now that we know we need to be running the algorithm only for $\bar{s}_\epsilon$ epochs. First, we group together the error terms in $\varsigma = (1 + \delta_{\bar{s}_\epsilon})(\varsigma_1 + \varsigma_2)$. We recall that, by Eq. (5), $(\varsigma_1 + \varsigma_2) \propto \mu^2 + \sqrt{\mu^2 + \nu^2}$. Hence, again as for the first point of this proof, we can choose μ and ν small enough such that $\varsigma \le \frac{\epsilon}{4}$. Note that it is exactly in this step that we need **(A2$_\mu$)**.

Hence, we can reach accuracy $\epsilon = \frac{\epsilon}{4} + \frac{\epsilon}{2} + \frac{\epsilon}{4}$. □

We make two important remarks.

Remark (Error terms)**.** *As we discussed in the proof of Corollary 3, smaller smoothing parameters yields smaller additional errors. Thus, in line with the previous literature (Nesterov & Spokoiny, 2011; Ji et al., 2019) we can choose the smoothing parameters μ, ν arbitrarily small as long as they are less than the upper bounds derived in ”Proof of Corollary 3” in the appendix. Theoretically, μ, ν relies on a good estimation of Z in A2$_\mu$. However, from a more practical side, we note in our experimental results in Section 6 that the worst-case guarantees are not necessarily tight since we do not observe any significant error accumulation.*

Remark (Dependency on the problem dimension)**.** *The overall dependency of $\bar{N}_\epsilon$ on the problem dimension is $\mathcal{O}(d\log(d))$. This complexity is comparable[3] to the usual $\mathcal{O}(d)$ found in the classical literature (Nesterov & Spokoiny, 2011; Ghadimi & Lan, 2013).*

We conclude by observing that, in the case $n \ge D_0/\epsilon$, Algorithm 1 achieves a linear rate of convergence when the desired accuracy is low (ϵ has a large value) and/or n is large. In the other case $n < D_0/\epsilon$ (i.e. high accuracy), Algorithm 1 achieves *acceleration*.

4.3. Analysis for Smooth and Strongly-convex Functions

We now analyze the case where f is strongly-convex.

(A3) $f = \frac{1}{n}\sum_{i=1}^n f_i$ is τ-strongly convex. That is, for all $x, y \in \mathbb{R}^d$, $f(y) \ge f(x) + \langle \nabla f(x), y - x\rangle + \frac{\tau}{2}\|y - x\|^2$.

For this case, we do not need **(A2$_\mu$)** since we will leverage on strong convexity (which implies coercivity) to include Z directly into our analysis.

Theorem 4. *Assume **(A1)** and **(A3)**. Let us denote $s_0 := \lfloor \log(d+4)n \rfloor + 1$ and assume that the weights $\{\theta_t\}$ are set to Eq. (10) if $1 \le s \le s_0$. Otherwise, they are set to*

$$\theta_t = \begin{cases} \Gamma_{t-1} - (1 - \alpha_s - p_s)\Gamma_t, & 1 \le t \le T_s - 1, \\ \Gamma_{t-1}, & t = T_s, \end{cases} \tag{12}$$

where $\Gamma_t = \left(1 + \frac{\tau\gamma_s}{2}\right)^t$. If the parameters $\{T_s\}$, $\{\gamma_s\}$ and

[3] As an interesting side-note, if d is the ratio between the diameter of the universe and the diameter of a proton (i.e. $\approx 10^{42}$), we have $\log(d) < 100$.

$\{p_s\}$ are set to Eq. (8) with

$$\alpha_s = \begin{cases} \frac{1}{2}, & s \leq s_0, \\ \min\{\sqrt{\frac{n\tau}{24L}}, \frac{1}{2}\}, & s > s_0, \end{cases} \tag{13}$$

we obtain

$$\mathbb{E}[f_\mu(\tilde{x}^s) - f_\mu^*] \leq$$
$$\begin{cases} \frac{1}{2^{s+1}}(d+4)D_0 + 2\varsigma_1 + 0.5\varsigma_2, & 1 \leq s \leq s_0 \\ (4/5)^{s-s_0}\frac{D_0}{n} + 12\varsigma_1 + 5\varsigma_2, & s > s_0 \text{ and } n \geq \frac{6L}{\tau} \\ \left(1 + \frac{1}{4}\sqrt{\frac{n\tau}{6L}}\right)^{-(s-s_0)}\frac{D_0}{n} & s > s_0 \text{ and } n < \frac{6L}{\tau} \\ \quad + \left(8\sqrt{\frac{6L}{n\tau}} + 4\right)\varsigma_1 + 5\varsigma_2, & \end{cases}$$

where $\varsigma_1 = \mu^2 L(d+4)^2$, $\varsigma_2 = E/\tau$ and D_0 is defined as in Eq. (11).

Remark. *Unlike the result in Theorem 2 for convex functions, the error term in Theorem 4 for the strongly-convex case does not increase with the epoch s. This is consistent with Theorem 9 in (Nesterov & Spokoiny, 2011).*

Using the same technique as for the proof of Corollary 3, we get the following complexity bound.

Corollary 5. *Assume **(A1)** and **(A3)**. The total number $\bar{N}_\epsilon$ of function queries performed by Algorithm 1 to find a stochastic ϵ-solution, i.e. a point $\bar{x} \in \mathbb{R}^d$ s.t. $\mathbb{E}[f(\bar{x}) - f^*] \leq \epsilon$, can be bounded by*

$$\bar{N}_\epsilon := \begin{cases} \mathcal{O}\Big\{dn\log\left(\frac{dD_0}{\epsilon}\right)\Big\}, & n{\geq}D_0/\epsilon \text{ or } n{\geq}6L/\tau, \\ \mathcal{O}\Big\{dn\log(dn) & n{<}D_0/\epsilon \textit{ and } n{<}6L/\tau \\ +d\sqrt{\frac{nL}{\tau}}\log\left(\frac{D_0}{n\epsilon}\right)\Big\}, & \end{cases}$$

We conclude this subsection by commenting on the optimality of this complexity result, following the discussion in (Lan et al., 2019). When τ and ϵ are small enough (i.e. the second case in Corollary 5, ill-conditioned), ZO-Varag exhibits an accelerated linear rate of convergence which depends on the square root of the condition number $\sqrt{L/\tau}$. Else, if ϵ or τ are relatively large (first case), ZO-Varag treats the problem as if it was not strongly convex and retrieves the complexity bound of Corollary 3. Again, similarly to the smooth convex case, the dependency of $\bar{N}_\epsilon$ on the problem dimension is $\mathcal{O}(d\log(d))$.

5. A Coordinate-wise Variant

In this section, we study the effect of replacing the gradient estimator $g_\mu(x,u,i)$ in the inner-loop of Algorithm 1 with the coordinate wise variant $g_\nu(x,i)$ proposed in (Ji et al., 2019) and already used in the last section for the computation of the pivotal gradient $\tilde{g}^s$. More precisely, we consider the following modification (g_ν defined in Eq. (3)): at each inner-loop iteration t,

$$G_t = g_\nu(\underline{x}_t, i_t) - g_\nu(\tilde{x}, i_t) + \tilde{g}^s.$$

As we are not using a smoothed version of f anymore, we need to introduce a slight modification on **(A2$_{\boldsymbol{\mu}}$)**.

(A2$_{\boldsymbol{\nu}}$) Let $x^* \in \operatorname{argmin}_{x\in\mathbb{R}^d} f(x)$. For any epoch s of Algorithm 1, consider the inner-loop sequences $\{\underline{x}_t\}$ and $\{\bar{x}_t\}$. There exist a *finite* constant $Z < \infty$, potentially dependent on L and d, such that, for ν small enough,

$$\sup_{s\geq 0} \max_{x\in\{\bar{x}_t\}\cup\{\underline{x}_t\}} \mathbb{E}\left[\|x - x^*\|\right] \leq Z.$$

Again, as mentioned in the context of **(A2$_{\boldsymbol{\mu}}$)** in the last section, it is possible to show that this assumption holds under the requirement that f is coercive.

5.1. Modified Analysis for Smooth and Convex Functions

We follow the same proof procedure from last section, and comment the results with a remark at the end of this section.

Theorem 6. *Consider the coordinate-wise variant of Algorithm 1 we just discussed. Assume **(A1)** and **(A2$_{\boldsymbol{\nu}}$)**. Let us denote $s_0 := \lfloor \log n \rfloor + 1$. Suppose the weights $\{\theta_t\}$ are set as in Eq. (10) and parameters $\{T_s\}$, $\{\gamma_s\}$, $\{p_s\}$ are set as*

$$T_s = \begin{cases} 2^{s-1}, & s \leq s_0 \\ T_{s_0}, & s > s_0 \end{cases}, \ \gamma_s = \frac{1}{12L\alpha_s}, \quad p_s = \frac{1}{2}, \textit{ with} \tag{14}$$

$$\alpha_s = \begin{cases} \frac{1}{2}, & s \leq s_0 \\ \frac{2}{s-s_0+4}, & s > s_0 \end{cases}. \tag{15}$$

Then, we have

$$\mathbb{E}[f(\tilde{x}^s) - f^*] \leq$$
$$\begin{cases} \frac{D_0'}{2^{s+1}} + \varsigma_1 + 4\varsigma_2, & 1 \leq s \leq s_0 \\ \frac{16D_0'}{n(s-s_0+4)^2} + \delta_s \cdot (\varsigma_1 + \varsigma_2), & s > s_0 \end{cases}$$

where $\varsigma_1 = \nu^2 Ld$, $\varsigma_2 = L\sqrt{d}Z\nu$, $\delta_s = \mathcal{O}(s - s_0)$ and D_0' is defined as

$$D_0' := 2[f(x^0) - f(x^*)] + 6L\|x^0 - x^*\|^2, \tag{16}$$

where x^* is any finite minimizer of f.

Corollary 7. *Consider the coordinate-wise variant of Algorithm 1. Assume **(A1)** and **(A2$_\nu$)**. The total number $\bar{N}_\epsilon$ of function queries performed by Algorithm 1 to find a stochastic ϵ-solution, i.e. a point $\bar{x} \in \mathbb{R}^d$ s.t. $\mathbb{E}[f(\bar{x}) - f^*] \le \epsilon$, can be bounded by*

$$\bar{N}_\epsilon := \begin{cases} \mathcal{O}\left\{dn\log\frac{D_0'}{\epsilon}\right\}, & n \ge D_0'/\epsilon,\\ \mathcal{O}\left\{dn\log n + d\sqrt{\frac{nD_0'}{\epsilon}}\right\}, & n < D_0'/\epsilon.\end{cases}$$

5.2. Modified Analysis under Strong Convexity

Theorem 8. *Consider the coordinate-wise variant of Algorithm 1. Assume **(A1)**, **(A2$_\nu$)** and **(A3)**. Let us denote $s_0 := \lfloor \log n \rfloor + 1$ and assume that the weights $\{\theta_t\}$ are set to Eq. (10) if $1 \le s \le s_0$. Otherwise, they are set to*

$$\theta_t = \begin{cases} \Gamma_{t-1} - (1-\alpha_s - p_s)\Gamma_t, & 1 \le t \le T_s - 1,\\ \Gamma_{t-1}, & t = T_s,\end{cases} \tag{17}$$

where $\Gamma_t = \big(1 + \tau\gamma_s\big)^t$. If the parameters $\{T_s\}$, $\{\gamma_s\}$ and $\{p_s\}$ set to Eq. (14) with

$$\alpha_s = \begin{cases} \frac{1}{2}, & s \le s_0,\\ \min\{\sqrt{\frac{n\tau}{12L}}, \frac{1}{2}\}, & s > s_0,\end{cases} \tag{18}$$

We obtain

$$\mathbb{E}\big[f(\tilde{x}^s) - f^*\big] \le$$

$$\begin{cases} \dfrac{1}{2^{s+1}}D_0' + 1.5\varsigma_1 + 4\varsigma_2, & 1 \le s \le s_0\\[2ex] (4/5)^{s-s_0}\dfrac{D_0'}{n} + 9\varsigma_1 + 24\varsigma_2, & s > s_0 \text{ and } n \ge \frac{3L}{\tau}\\[2ex] \left(1 + \frac{1}{4}\sqrt{\frac{n\tau}{3L}}\right)^{-(s-s_0)}\dfrac{D_0'}{n} & \\ +\big(2\sqrt{\frac{3L}{n\tau}}+1\big)(3\varsigma_1 + 8\varsigma_2), & s > s_0 \text{ and } n < \frac{3L}{\tau}\end{cases}$$

where $\varsigma_1 = \nu^2 Ld$, $\varsigma_2 = L\sqrt{d}Z\nu$ and D_0' is defined as in Eq. (16).

Corollary 9. *Consider the coordinate-wise variant of Algorithm 1. Assume **(A1)**, **(A2$_\nu$)** and **(A3)**. The total number $\bar{N}_\epsilon$ of function queries performed by Algorithm 1 to find a stochastic ϵ-solution, i.e. a point $\bar{x} \in \mathbb{R}^d$ s.t. $\mathbb{E}[f(\bar{x}) - f^*] \le \epsilon$, can be bounded by*

$$\bar{N} := \begin{cases} \mathcal{O}\big\{dn\log\big(\frac{D_0'}{\epsilon}\big)\big\}, & n \ge \frac{D_0'}{\epsilon} \text{ or } n \ge \frac{3L}{\tau},\\[2ex] \mathcal{O}\Big\{dn\log(n) & n < \frac{D_0'}{\epsilon} \text{ and } n < \frac{3L}{\tau}\\ +d\sqrt{\frac{nL}{\tau}}\log\big(\frac{D_0'}{n\epsilon}\big)\Big\}, & \end{cases}$$

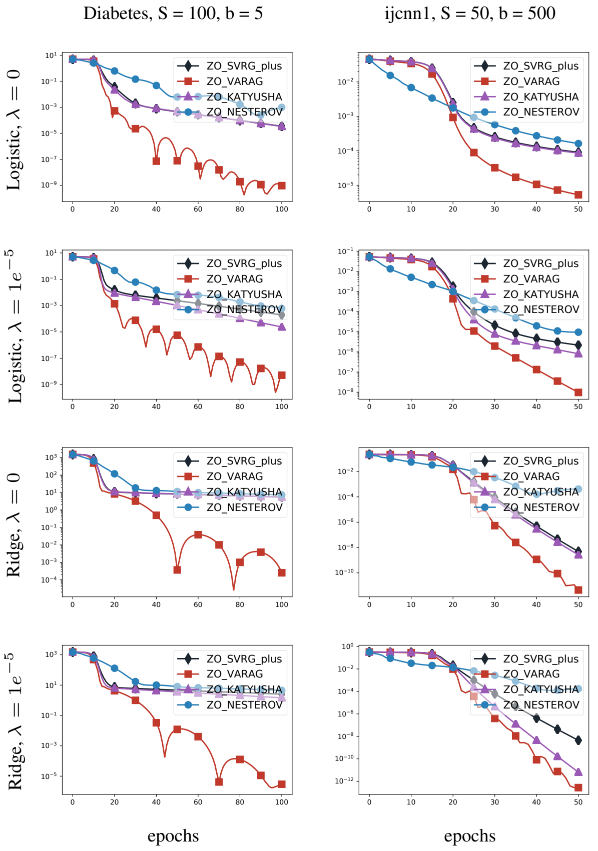

Figure 1. Loss $\log(f - f^*)$ over epochs. We show more results for various hyperparameters in the appendix.

Remark (Complexity using the coordinate-wise variant)**.** *Note that, the complexity results found in Corollary 7 and 9 are comparable to the ones for Gaussian smoothing (Corollary 3 and 5) while the dimensional dependency here is d rather than $d\log(d)$.*

6. Experiments

In this section, we compare the empirical performance of ZO-Varag with ZO-SVRG-Coord-Rand in (Ji et al., 2019) and a simplified ZO-Katyusha which is the ZO-version of the simplified Katyusha algorithm in (Shang et al., 2017), see Algorithm 2 in the appendix. We conduct experiments for both logistic regression and ridge regression [4] with and without ℓ_2 regularization on the diabete dataset ($n = 442, d = 10$) from sklearn and the ijcnn1 dataset ($n = 49990, d = 22$) from LIBSVM. The choice of the hyperparameters chosen for each algorithm is detailed in the appendix.

[4] Note that logistic regression with $\lambda = 0$ is not guaranteed to be coercive. However, this does not appear to be a problem in practice.

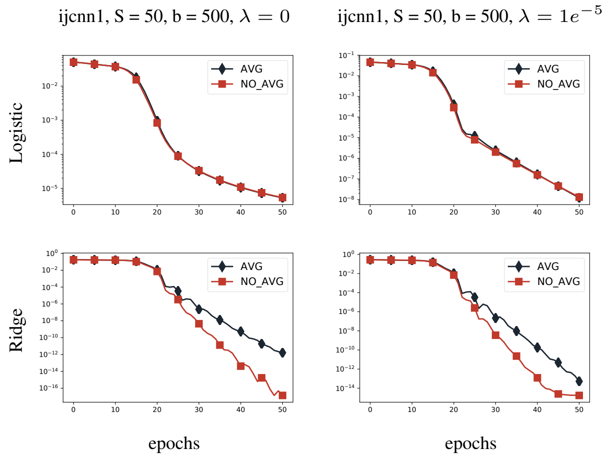

Figure 2. ZO-Varag, averaging vs. no-averaging

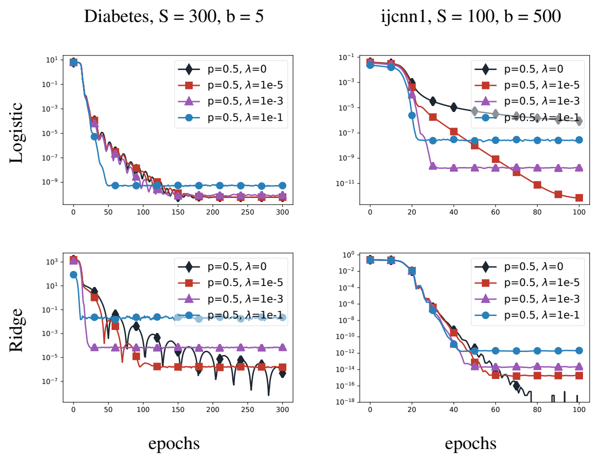

Figure 3. ZO-Varag, effect of varying the regularizer λ.

Based on our theoretical analysis, we require $\frac{(d+4)n}{2} \leq k \leq (d+4)n$ iterations per epoch for ZO-Varag. However, we can lower the computational cost by using a batch update with b samples per iteration and decrease the number of iterations to be b times smaller, i.e. $\frac{(d+4)n}{b}$ for each epoch.

6.1. Overall Performance

We first compare the performance of ZO-Varag to the baselines for two different regularizers: $\lambda = \{0, 1e^{-5}\}$ (i.e. adding $\lambda\|x\|^2$ to the loss). In this part, we set the Katyusha momentum to a constant p_0 such that $p_0 + \alpha_0 = 1$. We then set the Katyusha momentum to $p_0 = 0.5$ (see additional results for different values of p_0 in the appendix). The results shown in Figure 1 demonstrate that ZO-Varag does achieve an accelerated rate for all settings. While the zero-th order adaptation of simplified Katyusha does seem to be faster than the other two approaches, its performance is still close to the ZO-SVRG-Coord-Rand introduced in (Ji et al., 2019). Finally, we note that Nesterov's ZO (ZO-Nesterov) method (Nesterov & Spokoiny, 2011) is a deterministic approach and it therefore has a much higher complexity per step. Indeed, while one step of ZO-Nesterov requires $2n$ queries, all the other methods require $2b$ queries. In order to establish a fair comparison, we plot the results of ZO-Nesterov with the nearest functional queries w.r.t. the results at the pivotal points for other stochastic methods.

6.2. Options for Pivotal Point

As in (Johnson & Zhang, 2013), we consider two options for specifying the pivot point: i) $\tilde{x} = \tilde{x}^{s-1}$ (as used in our analysis), or ii) $\tilde{x} = \bar{x}^{s-1}$. The comparison of these two options is shown in Fig. 2 as well as in the appendix. Although option ii) does not have any theoretical guarantee, it empirically converges at a slightly faster rate than i) for logistic regression and significantly more for ridge regression.

6.3. Effect of the Regularizer λ

We vary the strength of the regularizer to understand the behavior of the algorithms for objectives with stronger convexity constants and also to observe the convergence of the algorithm to the optimal solution. These results are shown in Figure 3 for increasing values of λ. ZO-Varag is faster in the initial stage but for all values of λ, we observe that it converges to a ball around the optimum. At first, one could expect that this might be due to the DFO errors ς_1, ς_2 shown in our convergence theorems, which would only appears to be a problem in high-accuracy regimes. However, the reason may come from other two additional sources: 1) the non-vanishing SVRG variance problem raised in the SARAH paper (see Fig. 1 in (Nguyen et al., 2017) and our discussion of Fig. 7 in the appendix) and 2) the fact that stronger convexity constants increase the approximation error of Gaussian smoothing.

7. Conclusion

We presented a derivative-free algorithm that achieves the first accelerated rate of convergence for stochastic optimization of a convex finite-sum objective function. We also extended our analysis to the case of strongly-convex functions and included a variant of our algorithm for a coordinate-wise estimation of the gradient based on (Ji et al., 2019). Besides, a proximal variant of our approach could probably be derived, to deal with non-smooth problems as in the original Varag algorithm (Lan et al., 2019). Finally, we conducted experiments on several datasets demonstrating that our algorithm performs better than all existing non-accelerated DFO algorithms.

References

Allen-Zhu, Z. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1200–1205. ACM, 2017.

Bergou, E. H., Gorbunov, E., and Richtárik, P. Stochastic three points method for unconstrained smooth minimization. *arXiv preprint arXiv:1902.03591*, 2019.

Bubeck, S., Cesa-Bianchi, N., et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5 (1):1–122, 2012.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. 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*, pp. 15–26. ACM, 2017.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pp. 1646–1654, 2014.

Fang, C., Li, C. J., Lin, Z., and Zhang, T. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pp. 689–699, 2018.

Gadat, S., Panloup, F., Saadane, S., et al. Stochastic heavy ball. *Electronic Journal of Statistics*, 12(1):461–529, 2018.

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Gorbunov, E., Dvurechensky, P., and Gasnikov, A. An accelerated method for derivative-free smooth stochastic convex optimization. *arXiv preprint arXiv:1802.09022*, 2018.

Gorbunov, E., Bibi, A., Sener, O., Bergou, E. H., and Richtárik, P. A stochastic derivative free optimization method with momentum. *arXiv preprint arXiv:1905.13278*, 2019.

Ji, K., Wang, Z., Zhou, Y., and Liang, Y. Improved zeroth-order variance reduced algorithms and analysis for non-convex optimization. In *International Conference on Machine Learning*, pp. 3100–3109, 2019.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pp. 315–323, 2013.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lan, G. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

Lan, G. and Zhou, Y. An optimal randomized incremental gradient method. *Mathematical programming*, 171(1-2): 167–215, 2018.

Lan, G., Li, Z., and Zhou, Y. A unified variance-reduced accelerated gradient method for convex optimization. *arXiv preprint arXiv:1905.12412*, 2019.

Lin, H., Mairal, J., and Harchaoui, Z. A universal catalyst for first-order optimization. In *Advances in neural information processing systems*, pp. 3384–3392, 2015.

Liu, L., Cheng, M., Hsieh, C.-J., and Tao, D. Stochastic zeroth-order optimization via variance reduction method. *arXiv preprint arXiv:1805.11811*, 2018a.

Liu, S., Kailkhura, B., Chen, P.-Y., Ting, P., Chang, S., and Amini, L. Zeroth-order stochastic variance reduction for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pp. 3727–3737, 2018b.

Matyas, J. Random optimization. *Automation and Remote control*, 26(2):246–253, 1965.

Nelder, J. A. and Mead, R. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

Nesterov, Y. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Publishing Company, Incorporated, 1 edition, 2014. ISBN 1461346916.

Nesterov, Y. and Spokoiny, V. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2011.

Nesterov, Y. E. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pp. 543–547, 1983.

Nguyen, L. M., Liu, J., Scheinberg, K., and Takáč, M. Sarah: A novel method for machine learning problems using stochastic recursive gradient, 2017.

Orvieto, A., Kohler, J., and Lucchi, A. The role of memory in stochastic optimization. In *UAI*, 2019.

Polyak, B. T. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Shang, F., Liu, Y., Cheng, J., and Zhuo, J. Fast Stochastic Variance Reduced Gradient Method with Momentum Acceleration for Machine Learning. *arXiv e-prints*, Mar 2017.

Zhou, D., Xu, P., and Gu, Q. Stochastic nested variance reduction for nonconvex optimization. *Journal of Machine Learning Research*, 2020.