Soft Computing (2020) 24:17573–17588
https://doi.org/10.1007/s00500-020-05219-6

FOCUS

Ritz-like values in steplength selections for stochastic gradient methods

Giorgia Franchini[1] · Valeria Ruggiero[2] · Luca Zanni[1]

Published online: 18 August 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

The steplength selection is a crucial issue for the effectiveness of the stochastic gradient methods for large-scale optimization problems arising in machine learning. In a recent paper, Bollapragada et al. (SIAM J Optim 28(4):3312–3343, 2018) propose to include an adaptive subsampling strategy into a stochastic gradient scheme, with the aim to assure the descent feature in expectation of the stochastic gradient directions. In this approach, theoretical convergence properties are preserved under the assumption that the positive steplength satisfies at any iteration a suitable bound depending on the inverse of the Lipschitz constant of the objective function gradient. In this paper, we propose to tailor for the stochastic gradient scheme the steplength selection adopted in the full-gradient method knows as limited memory steepest descent method. This strategy, based on the Ritz-like values of a suitable matrix, enables to give a local estimate of the inverse of the local Lipschitz parameter, without introducing line search techniques, while the possible increase in the size of the subsample used to compute the stochastic gradient enables to control the variance of this direction. An extensive numerical experimentation highlights that the new rule makes the tuning of the parameters less expensive than the trial procedure for the efficient selection of a constant step in standard and mini-batch stochastic gradient methods.

Keywords Stochastic gradient methods · Learning rate selection rule · Ritz-like values · Adaptive subsampling strategies · Reduction variance techniques

1 Introduction

The problem we consider is the unconstrained minimization of the form

$$\min_{x\in\mathbb{R}^d} F(x) \equiv \mathbb{E}[f(x,\xi)], \tag{1}$$

where ξ is a multi-value random variable, f represents a cost function, and the mathematical expectation $\mathbb{E}$ is defined with respect to ξ in the probability space $(\Xi, \mathcal{F}, \mathcal{P})$. It is assumed that the function $f : \mathbb{R}^d \times \Xi \to \mathbb{R}$ is known analytically or it is provided by a black box oracle within a prefixed accuracy. In practice, since the probability distribution of ξ is unknown, we seek the solution of a problem that involves an estimate of the objective function $F(x)$. The most common approximation is the Sample Average Approximation, defined as

$$\min_{x\in\mathbb{R}^d} F_n(x) \equiv F_n(x,\xi^{(n)}), \tag{2}$$

where the objective function is the empirical risk

$$F_n(x,\xi^{(n)}) = \frac{1}{n}\sum_{i=1}^{n} f(x,\xi_i^{(n)}) = \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{3}$$

based on a random sample $\xi^{(n)} = \{\xi_1^{(n)},\ldots,\xi_n^{(n)}\}$ of size n of the variable ξ. In the machine learning framework, each $f_i(x) \equiv f(x,\xi_i^{(n)})$ denotes the loss function related to the instance $\xi_i^{(n)}$ of the training set. In the big data frame-

Communicated by Yaroslav D. Sergeyev.

✉ Giorgia Franchini
giorgia.franchini@unimore.it

Valeria Ruggiero
valeria.ruggiero@unife.it

Luca Zanni
luca.zanni@unimore.it

[1] Department of Physics, Informatics and Mathematics, University of Modena and Reggio Emilia, Modena, Italy

[2] Department of Mathematics and Computer Science, University of Ferrara, Ferrara, Italy

work, since n can be a very large number, it is prohibitively expensive to deal with the objective function $F_n(x, \xi^{(n)})$, its gradient or its Hessian matrix. A common approach to address the problem (1) or its approximation (2)–(3) is the Stochastic Gradient (SG) method and its variants, requiring only the gradient of one or few terms of $F_n(x)$ at each iteration, so that the cost of the overall optimization procedure is limited.

Starting from a vector $x^{(0)} \in \mathbb{R}^d$, the basic iteration of the SG method can be written as

$$x^{(k+1)} = x^{(k)} - \alpha_k g(x^{(k)}, \xi^{(n_k)}), \tag{4}$$

where $\xi^{(n_k)}$ denotes a set of n_k realizations of the random variable ξ, randomly chosen from the sample data $\xi^{(n)}$, $g(x^{(k)}, \xi^{(n_k)})$ is the stochastic gradient vector at the current iterate $x^{(k)}$ and α_k is a positive steplength, known also as learning rate. The main strategies for the choices of $\xi^{(n_k)}$ give rise to the standard SG method, when $n_k = 1$ for all k, and its mini-batch version, for $n_k > 1$. In particular, given a randomly chosen subset $S_k \subset \{1, \ldots, n\}$ of $|S_k| = n_k$ indices, $n_k \geq 1$, and a subsample of the training set $\xi^{(n_k)} = \{\xi_i^{(n_k)}\}_{i \in S_k}$, the stochastic gradient is defined as

$$g_k^{(n_k)} \equiv g(x^{(k)}, \xi^{(n_k)}) = \frac{1}{n_k} \sum_{i \in S_k} \nabla f_i(x^{(k)}). \tag{5}$$

As concerns the convergence results of the standard SG method (4) and its variant with fixed subsample size n_k, a very deep survey is given in Bottou et al. (2018). The results provided in Bottou et al. (2018) hold in the case of the solution of both problems (1) and (2). Under the crucial assumption that the gradient of the objective function is L-Lipschitz continuous and some additional conditions on the first and second moments of the stochastic gradient, when the positive steplength α_k is bounded from above by a constant $\alpha_{\max}$, the expected optimality gap for strongly convex objective functions, or the expected sum of gradients for general objective functions, asymptotically converge to values proportional to $\alpha_{\max}$. In practice, if the steplength is sufficiently small and $k \to \infty$, the method generates iterates in the neighborhood of the optimal or stationary value.

Nevertheless, since the constants related to the assumptions, such as the Lipschitz parameter or the parameters involved in the bounds of the moments of the stochastic directions, are unknown and not easy to approximate, there is no guidance on the specific choice of the steplength. A selection of a too small value of a steplength without an accurate tuning, can give rise to a very slow learning process. In the literature, there exist a number of different proposals to overcome this drawback without resorting to second-order methods or introduce line search techniques.

In particular, we refer to Tan et al. (2016) and Yang et al. (2018), where the updating rule of the steplength is borrowed from the Barzilai's rules, well known in the deterministic context. In the case of strongly convex objective functions, in order to obtain a linear convergence in expectation to zero for the optimality gap (Yang et al. 2018) or to a solution for the sequence of the iterates (Tan et al. 2016), the updating rules are inserted in variance reducing schemes as SVRG (Johnson and Zhang 2013) or SAGA (Defazio et al. 2014). These methods require to periodically compute the full gradient or to store the last computed term of each gradient in the sum (3).

Another way to obtain the linear convergence for strongly convex objective functions consists in increasing n_k at a geometric rate (Byrd et al. 2012) (see also Friedlander and Schmidt 2012). Despite this very strong condition, from the practical point of view, a procedure based on the so-called *norm test*, enables to control the sample size n_k so that

$$\mathbb{E}\left[\|g_k^{(n_k)} - \nabla F(x^{(k)})\|^2\right] \leq \zeta \|\nabla F(x^{(k)})\|^2$$

for some $\zeta > 0$ (Hashemi et al. 2014). In the practical implementation, the left side of the last inequality can be approximated with the sample variance and the gradient $\nabla F(x^{(k)})$ on the right side with a sample gradient (Byrd et al. 2012; Bottou et al. 2018). Similar techniques are developed in Cartis and Scheinberg (2015), relaxing the norm test by the use of a line search technique based on the true value of the objective function.

A recent proposal suggested in Bollapragada et al. (2018) is to increase the sample on the basis of an *inner product test*, combined with an *orthogonality test*. These conditions guarantee that the negatives of the stochastic gradients based on subsamples of suitable size are descent directions in expectation. Numerical evidence highlights that the mechanism gives rise to an increase in n_k slower than the one induced by the norm test; on the other hand, linear rate of convergence for objective functions satisfying the Polyak–Lojasiewicz (P–L) condition is preserved and other theoretical convergence features hold for general problems. These results strongly depend on the knowledge of the Lipschitz parameter L or on its suitable (local) estimate. Consequently, motivated by the numerical experiences shown in Franchini et al. (2020), in this paper we propose to tailor the steplength selection rule adopted in the Limited Memory Steepest Descent (LMSD) method (Fletcher 2012) to give a local estimate of the inverse of L in the SG framework, combining this strategy with the technique for increasing the subsample size detailed in Bollapragada et al. (2018) to adaptively control the variance of the stochastic directions.

The paper is organized as follows. In Sect. 2, we briefly recall the inner product test, the orthogonality test and the related theoretical convergence results. Section 3 is devoted to address the Ritz-like values in the context of the LMSD deterministic method and to tailor this technique to the stochastic framework; more precisely, the SG iteration based on the steplength defined by a Ritz-like value is combined with the adaptive subsampling technique proposed in Bollapragada et al. (2018). In Sect. 4 we describe the results of an extensive numerical experimentation. The conclusions are drawn in Sect. 5.

2 Theoretical results on SG method equipped with inner product and orthogonality tests

The convergence results on SG iteration (4) require that the stochastic gradient $g_k^{(n_k)}$ based on the subsample $\xi^{(n_k)}$ is a descent direction sufficiently often, i.e., assuming that S_k is chosen uniformly at random from $\{1, \dots, n\}$ and $g_k^{(n_k)}$ is an unbiased estimate of $\nabla F(x^{(k)})$, we can write

$$\mathbb{E}\left[{g_k^{(n_k)}}^{\mathrm{T}} \nabla F(x^{(k)}) \right] = \|\nabla F(x^{(k)})\|^2, \tag{6}$$

for all $k \geq 0$. The variance of the term on the left-hand side of (6) can be controlled by determining the sample size n_k at the k-th iteration so that the stochastic gradient is guaranteed to be a suitable estimate of the corresponding gradient. In particular, the following condition can be imposed on the sample size n_k of $\xi^{(n_k)}$:

$$\mathbb{E}\left[\left({g_k^{(n_k)}}^{\mathrm{T}} \nabla F(x^{(k)}) - \|\nabla F(x^{(k)})\|^2\right)^2 \right] \leq \theta^2 \|\nabla F(x^{(k)})\|^4, \tag{7}$$

for some $\theta > 0$. Furthermore, the inner product test can be combined with the orthogonality test, guaranteeing that the step direction is bounded away from orthogonality with $\nabla F(x^{(k)})$:

$$\mathbb{E}\left[\|g_k^{(n_k)} - \frac{{g_k^{(n_k)}}^{\mathrm{T}} \nabla F(x^{(k)})}{\|\nabla F(x^{(k)})\|^2} \nabla F(x^{(k)})\|^2 \right] \leq \nu^2 \|\nabla F(x^{(k)})\|^2, \tag{8}$$

for some $\nu > 0$. The combination of the two tests (7) and (8) is also known as *augmented inner product test*.

Borrowing the results stated in Bollapragada et al. (2018), we perform the following additional assumptions:

A. ∇F is L-Lipschitz continuous;

B. the Polyak–Lojasiewicz (P–L) condition holds

$$\|\nabla F(x)\|^2 \geq 2c(F(x) - F_*), \quad \forall x \in \mathbb{R}^d, \tag{9}$$

where c is a positive constant and $F_* = \inf_{x \in \mathbb{R}^d} F(x)$;

C. $\alpha_k \in (\alpha_{\min}, \alpha_{\max}]$, and $\alpha_{\max} \leq \frac{1}{(1+\theta^2+\nu^2)L}$, for given positive constants θ, ν in (7) and (8).

We remark that assumption B holds when F is c-strongly convex, but it is also satisfied for other functions that are not convex (see Karimi and Nutini 2016). In addition we observe that assumptions A and B do not guarantee the existence of a stationary point for F; nevertheless, under the two assumptions, any stationary point for F is a global minimizer.

Furthermore, in view of the assumption C, the iteration (4) can be equipped by a variable steplength, as long as it belongs to the interval $(\alpha_{\min}, \alpha_{\max}]$, where $\alpha_{\max}$ is proportional to the inverse of L.

Following the arguments of Bollapragada et al. (2018), the following theorems can be stated.

Theorem 1 *Suppose the assumptions A and B hold. Let $\{x^{(k)}\}$ be the sequence generated by (4), where the size n_k of any subsample is chosen so that the conditions (7) and (8) are satisfied and α_k satisfies the assumption C. Then, we have that*

$$\mathbb{E}\left[F(x^{(k)}) - F_* \right] \leq \rho^k \left(F(x^{(0)}) - F_* \right), \tag{10}$$

where $\rho = 1 - c\,\alpha_{\min}$. In particular, for a constant steplength $\alpha_k \equiv \alpha_{\max} = \frac{1}{(1+\theta^2+\nu^2)L}$, for all $k \geq 0$, we have $\rho = 1 - \frac{c}{(1+\theta^2+\nu^2)L}$.

The proof follows as in Theorem 3.2 of Bollapragada et al. (2018), by using the (P–L) condition instead of the strongly convexity of F and the inequality $\alpha_k \leq \alpha_{\max} \leq \frac{1}{(1+\theta^2+\nu^2)L}$.

In the case of a convex function F such that the P-L condition does not hold, we can state the following theorem, whose proof runs as the one of Theorem 3.3 of Bollapragada et al. (2018), taking account of $\alpha_k \leq \alpha_{\max}$ and of the additional strict bound on $\alpha_{\max}$.

Theorem 2 *Suppose the assumption A holds. Let $\{x^{(k)}\}$ be the sequence generated by (4), where the size n_k of any subsample is chosen so that the conditions (7) and (8) are satisfied and α_k satisfies the assumption C, with $\alpha_{\max} < \frac{1}{(1+\theta^2+\nu^2)L}$. Assume that $X^* = argmin_x F(x) \neq \emptyset$. Then, we have that*

$$\min_{0 \leq k \leq K} \mathbb{E}\left[F(x^{(k)}) - F_* \right] \leq \frac{1}{2\alpha_k \gamma K} \|x^{(0)} - x^*\|^2, \tag{11}$$

where $x^* \in X^*$ and $\gamma = 1 - \alpha_{\max} L(1 + \theta^2 + \nu^2)$.

Finally, along the lines of Theorem 3.4 in Bollapragada et al. (2018), taking into account that $\alpha_k > \alpha_{\min}$, we state the following proposition for the case of a general non-convex objective function F. In this case, $\{\nabla F(x^{(k)})\}$ converges to zero in expectation, with a sub-linear rate of convergence of the smallest gradients arising after K iterations.

Theorem 3 *Suppose the assumption A holds and F is bounded below from F_*. Let $\{x^{(k)}\}$ be the sequence generated by (4), where the size n_k of any subsample is chosen so that the conditions (7) and (8) are satisfied and α_k satisfies the assumption C. Assume that $X^* = argmin_x F(x) \neq \emptyset$. Then, we have that*

$$\lim_{k\to\infty} \mathbb{E}\left[\|\nabla F(x^{(k)})\|^2\right] = 0. \tag{12}$$

Furthermore, for any $K > 0$, we have

$$\min_{0\le k\le K-1} \mathbb{E}\left[\|\nabla F(x^{(k)})\|^2\right] \le \frac{1}{2\alpha_{\min} K}\left(F(x^{(0)}) - F_*\right). \tag{13}$$

To make a robust implementation of the iteration (4), Bollapragada et al. propose to determine the current steplength by a backtracking line search, aimed at providing a (local) estimate of Lipschitz parameter.

Exploiting the assumption of α_k belonging to a suitable bounded interval, we propose an updating rule for the definition of the current α_k, based on a stochastic version of the LMSD computation of Ritz-like values. In the next section we recall the deterministic procedure and we describe the stochastic version in detail.

3 Steplength selection via Ritz and harmonic Ritz values

Among the state-of-the-art steplength selection strategies for deterministic gradient methods, the limited memory rule proposed in Fletcher (2012) is one of the most effective ideas for capturing second-order information on the objective function. In order to describe our strategy for extending this approach to the stochastic gradient methods, in the following we recall the basic details on the rule (Fletcher 2012).

3.1 The deterministic framework

The limited memory rule (Fletcher 2012) provides the steplengths for performing groups of $m \geq 1$ iterations, where m is a small number (generally not larger than 7). After each group of m iterations, called sweep, a symmetric tridiagonal $m \times m$ matrix is defined by exploiting the gradients computed within the sweep. The m eigenvalues of the tridiagonal matrix are interpreted as approximations of the eigenvalues of the Hessian of the objective function at the current iteration and their inverses define the m steplengths for the new sweep. The crucial point of this approach consists in building the tridiagonal matrix in an inexpensive way, starting from the information acquired in the last sweep. To this end, in Fletcher (2012) the following strategy is proposed: suppose that the iterate $x^{(j)}$ and m steplengths $\alpha_{j+k}, k = 0, \ldots, m-1$, are available for performing a new sweep and store the gradients and the steplengths used within the sweep in the following way:

$$G_j = [g_j,\ g_{j+1}, \ldots, g_{j+m-1}], \tag{14}$$

$$J_j = \begin{pmatrix} \frac{1}{\alpha_j} & & \\ -\frac{1}{\alpha_j} & \ddots & \\ & \ddots & \frac{1}{\alpha_{j+m-1}} \\ & & -\frac{1}{\alpha_{j+m-1}} \end{pmatrix}. \tag{15}$$

From the $d \times m$ matrix G_j, an upper triangular $m \times m$ matrix R_j such that $G_j^{\mathrm{T}} G_j = R_j^{\mathrm{T}} R_j$ can be obtained, for example by means of the Cholesky factorization of $G_j^{\mathrm{T}} G_j$; the matrix R_j is non-singular if G_j is full rank. By using G_j, J_j and R_j define the matrix

$$T_j = R_j^{-T} G_j^{\mathrm{T}} [G_j \quad g_{j+m}] J_j R_j^{-1} = [R_j \quad r_j] J_j R_j^{-1} \tag{16}$$

where r_j is the solution of the linear system $R_j^{\mathrm{T}} r_j = G_j^{\mathrm{T}} g_{j+m}$. In the case of quadratic strictly convex objective function, T_j is the symmetric tridiagonal matrix provided by m steps of the Lanczos process applied to the Hessian matrix of the objective function, with starting vector $g_j/\|g_j\|$; this means that its eigenvalues, called Ritz values, are special approximations of the Hessian eigenvalues. In the general non-quadratic case, T_j is upper Hessenberg and a symmetric tridiagonal matrix $\overline{T}_j$ can be obtained as

$$\overline{T}_j = \mathrm{tril}(T_j) + \mathrm{tril}(T_j, -1)', \tag{17}$$

where the Matlab notation is used for denoting the lower triangular of T_j. The limited memory steplength rule (Fletcher 2012) proposes to use the eigenvalues of $\overline{T}_j$, $\lambda_i, i = 1, \ldots, m$, as approximations of the eigenvalues of the Hessian of the objective function at the iteration $(j+m)$, and to exploit the inverses of these approximations as steplengths for the next sweep:

$$\alpha_{j+m-1+i} = \frac{1}{\lambda_i}, \quad i = 1, \ldots, m. \tag{18}$$

Following the terminology used in the quadratic case, we call Ritz-like values the eigenvalues of $\overline{T}_j$.

In Fletcher (2012) another idea is also introduced for defining the steplengths for the sweeps, based on a similar strategy. In the strictly convex quadratic case, this idea consists in obtaining the steplengths as eigenvalues of the matrix $P_j^{-1} T_j$, where

$$P_j = R_j^{-T} J_j^{\mathrm{T}} \begin{pmatrix} R_j & r_j \\ 0 & \rho_j \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} R_j & r_j \\ 0 & \rho_j \end{pmatrix} J_j R_j^{-1}$$
$$= \begin{pmatrix} T_j^{\mathrm{T}} & t_j \end{pmatrix} \begin{pmatrix} T_j \\ t_j^{\mathrm{T}} \end{pmatrix}, \qquad (19)$$

$\rho_j = \sqrt{g_{j+m}^{\mathrm{T}} g_{j+m} - r_j^{\mathrm{T}} r_j}$ and t_j is the solution of the linear system $R_j^{\mathrm{T}} t_j = J_j^{\mathrm{T}} \begin{pmatrix} 0 \\ \rho_j \end{pmatrix}$. The reciprocals of the eigenvalues of $P_j^{-1} T_j$ are called harmonic Ritz.

Replacing T_j in (19) by the non-singular tridiagonal matrix $\overline{T}_j$, a pentadiagonal matrix $\overline{P}_j$ is obtained. The matrices $\overline{T}_j$ and $\overline{T}^{-1}\overline{P}_j$ can have nonpositive eigenvalues. This phenomenon is due firstly to the non-quadratic features of the objective function and secondly to the presence of negative curvature. The first situation can arise also for convex objective functions, whereas the second one concerns the minimization of general functions. To overcome these drawbacks, there are different strategies. In Fletcher (2012), di Serafino et al. (2018), the authors suggest to simply discard these values, hence providing fewer than m steplengths for the next sweep; if no positive eigenvalues are available, any tentative steplength can be adopted for a sweep of length 1. In addition it can be convenient to discard also the oldest back gradients. Another strategy, aimed at handling non-positive curvature, is to adopt a local cubic model, that reduces to a standard quadratic model when only positive eigenvalues are computed (Curtis and Guo April 2016).

3.2 The Stochastic framework

The strategy suggested by the LMSD method for an adaptive update of the steplength in the full gradient method performs as well as an L-BFGS method also for the minimization of non-quadratic and non-convex objective functions (Fletcher 2012 and di Serafino et al. 2018). In a stochastic framework, where the computation of the Hessian matrix is very expansive, even when it is based on a subsampling, the LMSD approach can inspire a strategy for defining a selecting rule of the steplength at the current iteration of SG. The main difference with respect to the deterministic case is in the construction of the matrix G_j, where we have to replace the full gradients computed at the m most recent iterations ($m \geq 1$) with the corresponding stochastic gradients at the iterates $x^{(j+i)}$, obtained by using different samples of data $\{\xi^{(n_{j+i})}\}$, $i = 0, .., m-1$:

$$\overline{G}_j = \left[g_j^{(n_j)}, g_{j+1}^{(n_{j+1})}, \ldots, g_{j+m-1}^{(n_{j+m-1})} \right]. \qquad (20)$$

Following the procedure developed in the deterministic case combined with the approximation (17), the matrices $\overline{T}_j$ and $\overline{P}_j$ can be computed, by replacing G_j in (16) with $\overline{G}_j$.

When the collected stochastic gradients are suitable approximations of the full gradients, i.e., they are in expectation suitable descent directions at the current iterate with a reduced variance, it is quite likely that from the inverses of the eigenvalues of the matrices $\overline{T}_j$ and $\overline{T}_j^{-1}\overline{P}_j$, that are the Ritz-like and harmonic Ritz-like values, useful approximations of the inverse of the local Lipschitz constant of ∇F can be obtained for the new sweep of iterations. For simplicity, we refer in the following to the Ritz-like values, but, similarly, the same considerations hold for harmonic Ritz-like values. We observe that, in addition to the drawbacks highlighted in the deterministic context, in this case $\overline{G}_j$ is only an approximation of G_j and, as a consequence, non-positive Ritz-like values can arise. As in the deterministic case, these values can be discarded, by removing also the oldest back stochastic gradients from $\overline{G}_j$. As a consequence, fewer than m Ritz-like values λ_i, $i = 1, \ldots, m_R$, $m_R \leq m$, can be available.

Furthermore, in order to avoid line search techniques, it is convenient to consider only the values λ_i belonging to a prefixed range $[\frac{1}{\alpha_{\max}}, \frac{1}{\alpha_{\min}})$, where $\alpha_{\max} > \alpha_{\min} > 0$. In particular, we redefine

$$\lambda_i \leftarrow \max\left(\frac{1}{\alpha_{\max}}, \min\left(\lambda_i, \frac{1}{\alpha_{\min}}\right)\right), \quad i{=}1, \ldots, m_R, \qquad (21)$$

and we eliminate the values $\lambda_i = 1/\alpha_{\min}$, reducing again m_R and discarding all the stochastic gradients giving rise to these values. If $m_R = 0$, a tentative steplength $\overline{\alpha} \in (\alpha_{\min}, \alpha_{\max}]$ can be adopted for a sweep of length 1. This reference value is also used at the first iterate. If $m_R > 0$, the steplengths in the next sweep are defined as

$$\alpha_{j+m+i} = \frac{1}{\lambda_i}, \quad i = 1, \ldots, m_R. \qquad (22)$$

A similar procedure involving the harmonic Ritz-like values enables us to define alternatively the steplengths in the next sweep as

$$\alpha_{j+m+i} = \frac{1}{\tilde{\lambda}_i}, \quad i = 1, \ldots, m_H. \qquad (23)$$

We remark that, in view of Theorem 3.3 in Curtis and Guo (April 2016), the positive harmonic Ritz-like values are greater or equal than the corresponding Ritz-like values; as a consequence, the rule (23) generates shorter steplengths with respect to the ones defined by (22). The alternate use of different rules to generate long and short stepsizes in the full gradient methods has been deeply investigated (see, for example, Dai and Yuan 2003; Zhou et al. 2006; Frassoldati et al. 2008), showing a large increase in their practical performance. Also in the stochastic framework, we can explore an alternate use of the Ritz-like and harmonic Ritz-like values. A first approach can be to simply toggle the use of the Ritz-like values to the one of the harmonic Ritz-like values at each sweep (Alternate Ritz-like values or A-R).

A second strategy may be to link the choice between Ritz-like and harmonic Ritz-like values to the selection of the size of the current subsample. We discuss in detail this selection. A crucial point is how to check when the stochastic gradients assembling the matrix $\overline{G}_j$ can be considered acceptable estimates of the corresponding gradients. Inspired by the adaptive sampling technique in Bollapragada et al. (2018), the variance can be monitored by a suitable increase in the sample size n_k. More precisely, the inner test condition (7) can be imposed on the sample size n_k. Since the left-hand-side term of (7) is bounded from above by the true variance of individual gradient, the condition (7) holds when the following *exact variance inner product test* is satisfied:

$$\frac{\mathbb{E}[(\nabla f_i(x^{(k)})^{\mathrm{T}}\nabla F(x^{(k)}) - \|\nabla F(x^{(k)})\|^2)^2]}{n_k} \le \theta^2\|\nabla F(x^{(k)})\|^4. \tag{24}$$

In order to implement condition (24), the variance can be approximate with the sample variance

$$\mathrm{Var}_{i\in S_k}(\nabla f_i(x^{(k)})^{\mathrm{T}}\nabla F(x^{(k)}))$$

and the gradient $\nabla F(x^{(k)})$ on the right side with a sample gradient, so that the *approximate inner product test* is given by the following condition

$$\frac{(\sum_{i\in S_k}(\nabla f_i(x^{(k)})^{\mathrm{T}} g_k^{(n_k)} - \|g_k^{(n_k)}\|^2)^2}{n_k(n_k-1)} \le \theta^2\|g_k^{(n_k)}\|^4. \tag{25}$$

When this condition is not satisfied by the current sample size, the sample size is increased so that (25) is satisfied. With regard to the orthogonality test, a sufficient condition for (8) is the following *exact variance orthogonality test*:

$$\frac{\mathbb{E}[\|\nabla f_i(x^{(k)}) - \frac{\nabla f_i(x^{(k)})^{\mathrm{T}}\nabla F(x^{(k)})}{\|\nabla F(x^{(k)})\|^2}\nabla F(x^{(k)})\|^2]}{n_k} \le \nu^2\|\nabla F(x^{(k)})\|^2. \tag{26}$$

As for the previous test (24), a practical variant, named *approximate variance orthogonality test*, based on the sample approximation can be formulated as follows

$$\frac{\sum_{i\in S_k}\|\nabla f_i(x^{(k)}) - \frac{\nabla f_i(x^{(k)})^{\mathrm{T}} g_k^{(n_k)}}{\|g_k^{(n_k)}\|^2} g_k^{(n_k)}\|^2}{n_k(n_k-1)} \le \nu^2\|g_k^{(n_k)}\|^2. \tag{27}$$

In order to choose a new sample size n_k when the conditions (25) and (27) are not satisfied, we can compute

$$Z_1 = \frac{\mathrm{Var}_{i\in S_k}(\nabla f_i(x^{(k)})^{\mathrm{T}} g_k^{(n_k)}}{\theta^2\|g_k^{(n_k)}\|^4},$$
$$Z_2 = \frac{\mathrm{Var}_{i\in S_k}(\nabla f_i(x^{(k)}) - \frac{\nabla f_i(x^{(k)})^{\mathrm{T}} g_k^{(n_k)}}{\|g_k^{(n_k)}\|^2} g_k^{(n_k)})}{\nu^2\|g_k^{(n_k)}\|^2} \tag{28}$$

and set $n_k = \min(\lceil\max(Z_1, Z_2)\rceil, n)$. We observe that, when at the iteration k the size of the sample increases, the stochastic gradients previously stored are related to subsamples of lower size; then, we propose to discard the available Ritz-like values and to exploit the current stored stochastic gradients to determine a set of harmonic Ritz-like values. Indeed this strategy, named Adaptive Alternation of Ritz-like values (AA-R), leads to shorter steplengths in this transition phase.

The approximated version (25)–(27) of the augmented inner test is a way to choose the size of the subsample at the current iteration with the aim to control the goodness of the estimate $g_k^{(n_k)}$ of the full gradient $\nabla F(x^{(k)})$, but other approaches in the literature can be found, as for example the *norm test* in Byrd et al. (2012), Cartis and Scheinberg (2015), Hashemi et al. (2014), or the rule based on the matrix Bernstein inequality (Tropp 2015, Th. 6.1.1, Cor. 6.2.1) (see Cartis and Scheinberg 2015; Bellavia et al. 2019). We followed the strategy based on the conditions (25) and (27) since numerical experience highlights that they are not too restrictive, slowly increasing the sequence $\{n_k\}$.

4 Numerical experiments

In order to evaluate the effectiveness of the proposed steplength rule for SG methods, we consider the optimization problems arising in training binary and multi-labels classifiers for the following well-known data sets:

- the *MNIST* data set of handwritten digits, categorized in 10 classes (downloadable from http://yann.lecun.com/exdbmnist), commonly used for testing different systems that process images; the images in gray-scale [0, 255] are normalized in the interval [0, 1] and centered in a box of 28 × 28 pixels; the data set contains 60,000 images for training, whereas further 10, 000 images can be used for testing purposes;
- the web data set *w8a* downloadable from https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html, containing 49,749 examples, partitioned in 44,774 samples for training and 4975 for testing; each example is described by 300 binary features.

Table 1 Values of the best-tuned steplength α_{OPT} in 20 epochs for the standard SG method in the case of the two data sets and the three loss functions

	MNIST			*w8a*		
$F_n(x)$	LR	SL	SH	LR	SL	SH
α_{OPT}	10^{-3}	10^{-4}	10^{-3}	10^{-1}	10^{-3}	$5\ 10^{-2}$

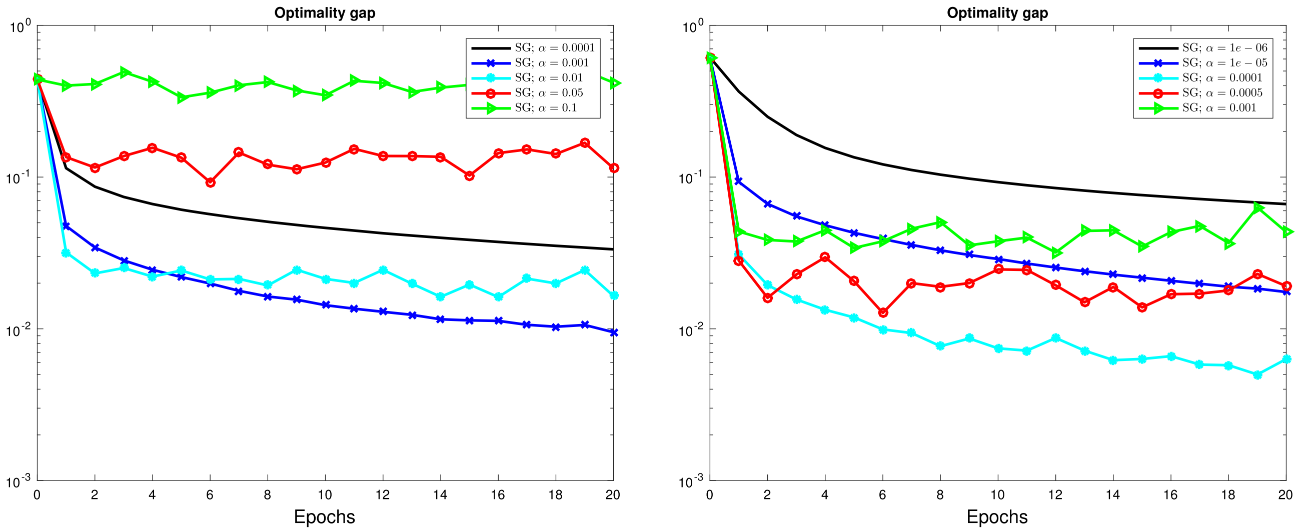


Fig. 1 Behavior of standard SG in 20 epochs on the *MNIST* data set, with logistic regression (on the left panel) and square loss (in the right panel)

We consider two kinds of problems, the first relating to convex objective functions and the second involving a non-convex objective function. In the case of convex minimization problems, a binary classifier is searched for the data sets *MNIST* and *w8a*. For the *MNIST* data set, the two classes are the even and odd digits. In the non-convex case, the objective function arises from the design of a multi-class classifier for the *MNIST* data set. In both kinds of problems, a regularization term was added to the loss function to avoid overfitting.

4.1 Convex problems

We built linear classifiers corresponding to three different convex loss functions. Thus the minimization problem has the form

$$\min_{x\in\mathbb{R}^d} F_n(x) + \frac{\delta}{2}\|x\|_2^2, \tag{29}$$

where $\delta > 0$ is the regularization parameter. By denoting as $a_i \in \mathbb{R}^d$ and $b_i \in \{1, -1\}$ the feature vector and the class label of the i-th sample, respectively, the loss function $F_n(x)$ assumes one of the following forms:

- logistic regression (LR) loss:

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} \log\left[1 + e^{-b_i a_i^{\mathrm{T}} x}\right];$$

- square loss (SL):

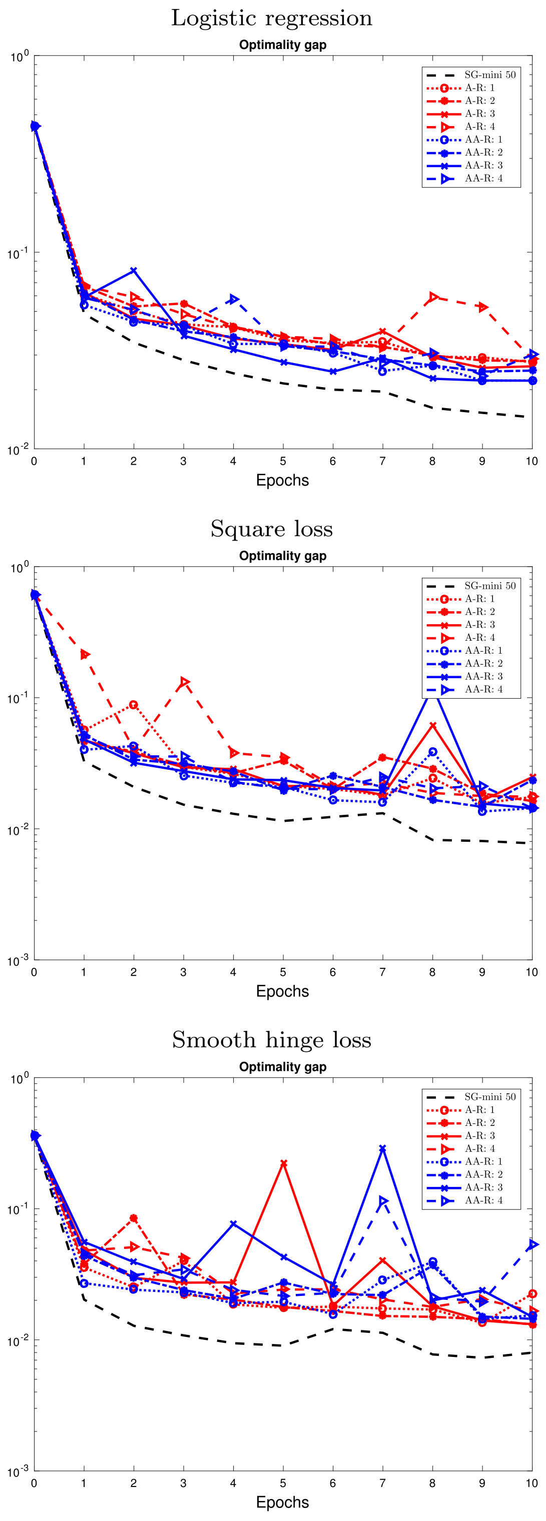
$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n}(1 - b_i a_i^{\mathrm{T}} x)^2;$$

- smooth hinge loss (SH):

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n}\begin{cases}\frac{1}{2} - b_i a_i^{\mathrm{T}} x, & \text{if } b_i a_i^{\mathrm{T}} x \le 0\\ \frac{1}{2}(1 - b_i a_i^{\mathrm{T}} x)^2, & \text{if } 0 < b_i a_i^{\mathrm{T}} x < 1\\ 0, & \text{if } b_i a_i^{\mathrm{T}} x \ge 1.\end{cases}$$

We compare the effectiveness of the following schemes:

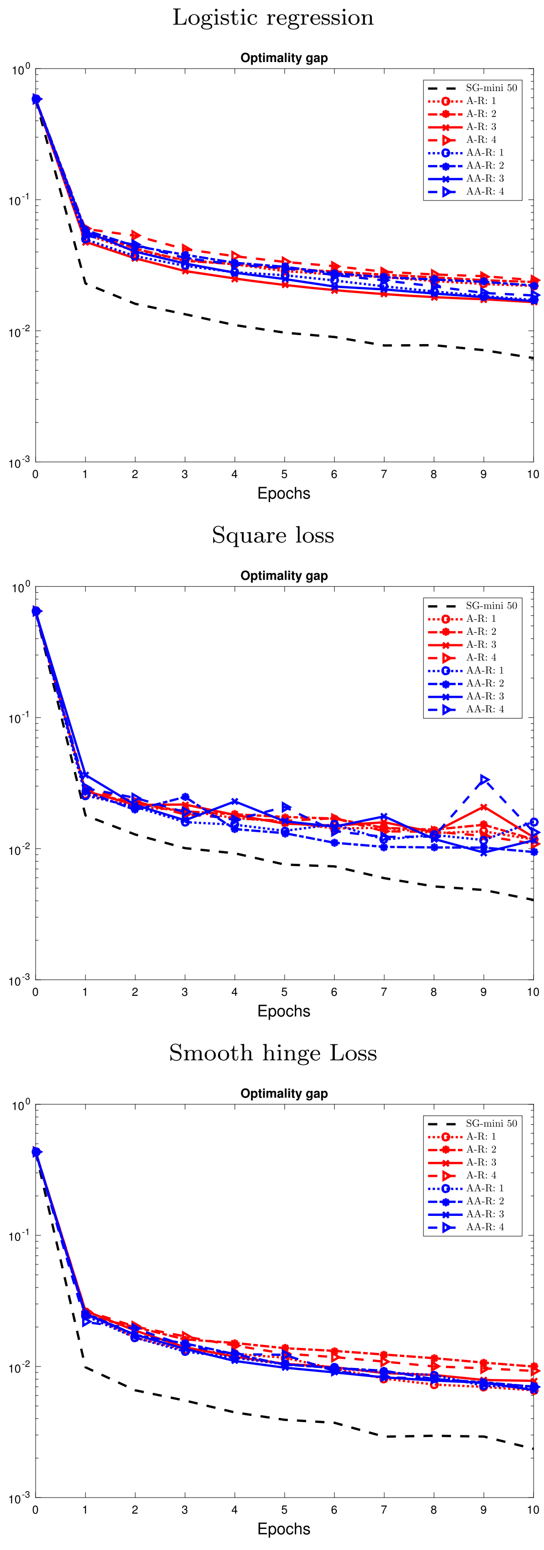
- SG with a fixed mini-batch size in the version with fixed steplength, denoted by **SG mini**;
- methods using Ritz-like values to adaptively select a suitable steplength; in particular, we consider:


Logistic regression

Square loss

Smooth hinge loss

Fig. 2 Behavior of the optimality gap in 10 epochs for **SG mini**, **A-R** and **AA-R** methods in the case of the *MNIST* data set


Logistic regression

Square loss

Smooth hinge Loss

Fig. 3 Behavior of the optimality gap in 10 epochs for **SG mini**, **A-R** and **AA-R** methods in the case of the *w8a* data set

- Alternate Ritz-like values in the scheme denoted by **A-R**, which toggles the use of the Ritz-like values to the one of the harmonic Ritz-like values at each sweep;
- Adaptive Alternation of Ritz-like values in the scheme denoted by **AA-R**; in this method, when at the iteration k the size of the sample increases, we discard the available Ritz-like values and we exploit the current stored stochastic gradients to determine a set of harmonic Ritz-like values.

For both methods, an adaptive strategy is used for increasing the mini-batch size, as detailed in Sect. 3.2. In all the numerical simulations, we set $\theta = 0.7$ in (25) and $\nu = 7$ in (27).

In all the numerical experiments, carried out in Matlab® on 1.6 GHz Intel Core i5 processor, we use the following setting:

- the regularization parameter δ is equal to 10^{-8};
- in **SG mini** the size of the mini-batch is set as $|S_k| = |S| = 50$ for all $k \geq 0$;
- in **A-R** and **AA-R** methods, the size of the initial mini-batch is $|n_0| = 3$; furthermore the maximum length of the sweep is set as $m = 3$;
- each method is stopped after 10 epochs, i.e., after a time interval equivalent to 10 evaluations of a full gradient of F_n or 10 visits of the whole data set; in this way we compare the behavior of the methods in a time equivalent to 10 iterations of a full gradient method applied to $F_n(x)$.

In the following, we report the results obtained by the considered methods on the *MNIST* and *w8a*, by using the three loss functions (logistic regression, square and smooth hinge functions). For any numerical simulation we perform 10 runs with the same parameters, but leaving the possibility to the random number generator to vary. Indeed, due to the stochastic nature of the methods, the average values in different simulations provide more reliable outcomes. In particular, for any numerical test, we report the following results:

- the average value of the optimality gap $F_n(\overline{x}) - F_*$, where $\overline{x}$ is the iterate obtained at the end of the 10 epochs and F_* is an estimate of the optimal objective value; this value is obtained by a full gradient method with a huge number of iterations;
- the related average accuracy $A(\overline{x})$ at the end of the 10 epochs with respect to the testing set, i.e., the percentage of well-classified examples.

First of all, we determine by a trial procedure the best steplength α_{OPT} for the standard SG method, i.e., the

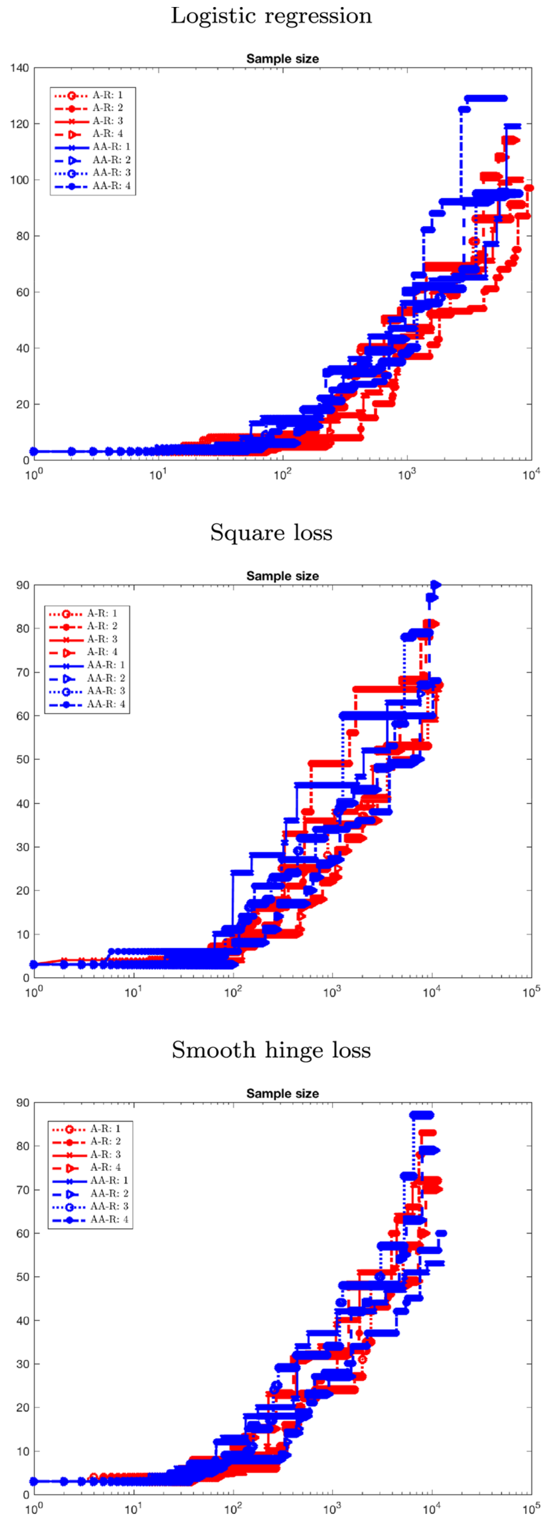


Fig. 4 Mini-batch size in **A-R** and **AA-R** methods on the *MNIST* data set with respect to the iterations

Table 2 Values of the setting providing the best results for **A-R** method

	MNIST			*w8a*		
$F_n(x)$	LR	SL	SH	LR	SL	SH
$\overline{\alpha}$	10^{-2}	10^{-3}	10^{-2}	1	10^{-2}	$5\ 10^{-1}$
$\alpha_{\min}$	10^{-5}	10^{-7}	10^{-6}	10^{-3}	10^{-6}	$5\ 10^{-4}$
$\alpha_{\max}$	1	$5\ 10^{-2}$	$5\ 10^{-1}$	100	1	25

Table 3 Values of the setting providing the best results for **AA-R** method

	MNIST			*w8a*		
$F_n(x)$	LR	SL	SH	LR	SL	SH
$\overline{\alpha}$	10^{-2}	10^{-3}	10^{-2}	1	10^{-2}	$5\ 10^{-1}$
$\alpha_{\min}$	10^{-5}	10^{-6}	10^{-5}	10^{-3}	10^{-6}	$5\ 10^{-4}$
$\alpha_{\max}$	1	10^{-1}	1	100	$5\ 10^{-1}$	50

Table 4 Numerical results of the considered methods with $F_n(x)$ given by the logistic regression after 10 epochs

Method	*MNIST*		*w8a*	
	$F_n(\overline{x}) - F_*$	$A(\overline{x})$	$F_n(\overline{x}) - F_*$	$A(\overline{x})$
SG mini	0.0145	0.890	0.0062	0.901
A-R	0.0263	0.890	0.0165	0.903
AA-R	0.0222	0.893	0.0168	0.903

Table 5 Numerical results of the considered methods with $F_n(x)$ given by the square loss after 10 epochs

Method	*MNIST*		*w8a*	
	$F_n(\overline{x}) - F_*$	$A(\overline{x})$	$F_n(\overline{x}) - F_*$	$A(\overline{x})$
SG mini	0.0078	0.892	0.0041	0.890
A-R	0.0163	0.888	0.0109	0.888
AA-R	0.0144	0.890	0.0094	0.888

Table 6 Numerical results of the considered methods with $F_n(x)$ given by the smooth hinge loss after 10 epochs

Method	*MNIST*		*w8a*	
	$F_n(\overline{x}) - F_*$	$A(\overline{x})$	$F_n(\overline{x}) - F_*$	$A(\overline{x})$
SG mini	0.0079	0.897	0.0024	0.907
A-R	0.013	0.896	0.0067	0.904
AA-R	0.0149	0.896	0.0067	0.904

steplength corresponding to the best obtained results. Indeed, following (Bottou et al. 2018), a suitable steplength for **SG mini** is $\alpha_{\text{SG mini}} = |S|\alpha_{OPT}$, where α_{OPT} is a fixed steplength for the standard SG method. We have tried five different steplengths for each combination of standard SG and data set. In Table 1, we report the value of the steplength α_{OPT} corresponding to the best performance of standard SG in 20 epochs. Furthermore, in order to highlight the trouble to define a suitable learning rate, in Fig. 1 we show the trend of the optimality gap for five values of the steplengths in the case of *MNIST* data set with logic regression and square loss functions. The instability of the standard SG method behavior with respect to the selection of the steplength motivates the expensive trial process that produces Table 1. In the following, we report the numerical results of the comparison between **SG mini** and **A-R** and **AA-R** methods. In particular, in **A-R** and **AA-R** methods, different settings of the bounds $\alpha_{\max}$ and $\alpha_{\min}$ are used:

1. $\alpha_{\min} = \alpha_{OPT}\ 10^{-2}$, $\alpha_{\max} = \alpha_{OPT}\ 500$;
2. $\alpha_{\min} = \alpha_{OPT}\ 10^{-3}$ and $\alpha_{\max} = \alpha_{OPT}\ 500$;
3. $\alpha_{\min} = \alpha_{OPT}\ 10^{-2}$ and $\alpha_{\max} = \alpha_{OPT}\ 1000$;
4. $\alpha_{\min} = \alpha_{OPT}\ 10^{-3}$ and $\alpha_{\max} = \alpha_{OPT}\ 1000$.

The tentative value of the steplength $\overline{\alpha}$ is set as $10\alpha_{OPT}$. Figures 3 and 4 show the behavior of the optimality gap with respect to the first 10 epochs for *MNIST* and *w8a*, respectively, in the case of the three loss functions. In particular, the dashed black line refers to **SG-mini**, whereas the red and the blues lines are related to **A-R** and **AA-R** methods, respectively, in the above specified four settings. We observe that the results obtained with the **A-R** and **AA-R** methods are comparable with the ones obtained with the **SG mini** equipped with the best tuned steplength. Indeed, the adaptive steplength rules in **A-R** and **AA-R** methods seem to be slightly dependent on the values of $\alpha_{\max}$ and $\alpha_{\min}$, making the choice of a suitable learning rate a less difficult task with respect to the selection of a good constant value in standard SG and **SG mini** methods. In Tables 2 and 3, we summarize the setting that provides the best results for **A-R** and **AA-R** methods. In Tables 4, 5 and 6, we show the final optimality gap (with respect to the training set) and accuracy (with respect to the testing set) obtained at the end of 10 epochs for the logistic regression, square and smooth hinge loss functions, respectively, for the best setting. The final accuracy of the three methods differs at most to the third decimal digit. This observation can also be extended to the simulations obtained for A-R and AA-R methods with the other settings. In Figs. 4 and 5, we show the increase in the subsample size in **A-R** and **AA-R** methods in the case of all the convex loss functions.

Starting with $n_0 = 3$, the size of current subsample is at least 120 in the case of *MNIST* data set and 900 in the case of *w8a* data set at the end of the 10 epochs, much smaller than the number of sample n of the training set.

Finally, in Figs. 6 and 7 , we compare the behavior of **SG mini** and **A-R** and **AA-R** methods when the parameter α_{SGmini} is not the best-tuned value, as in the previous experiments. In particular, **SG mini** method in Fig. 6 is carried out with α_{OPT} replaced by $\alpha = 10^{-5}$ for logistic regression function and $\alpha = 10^{-5}$ for smooth hinge loss function, that is $\alpha_{SGmini} = \alpha|S|$. In Fig. 7, **SG mini** is equipped with $\alpha = 1$ for logistic regression function and $\alpha = 10^{-5}$ for square loss function. **A-R** and **AA-R** methods are executed using the four previously specified settings, with α_{OPT} set as above.

Figures 6 and 7 highlight that a too small fixed steplength in SG mini produces a slow descent of the optimality gap; on the other hand, a steplength value larger than the best-tuned one can cause oscillating behavior of the optimality gap and, sometimes, it does not guarantee the convergence of **SG mini** method. As regards **A-R** and **AA-R** methods, these

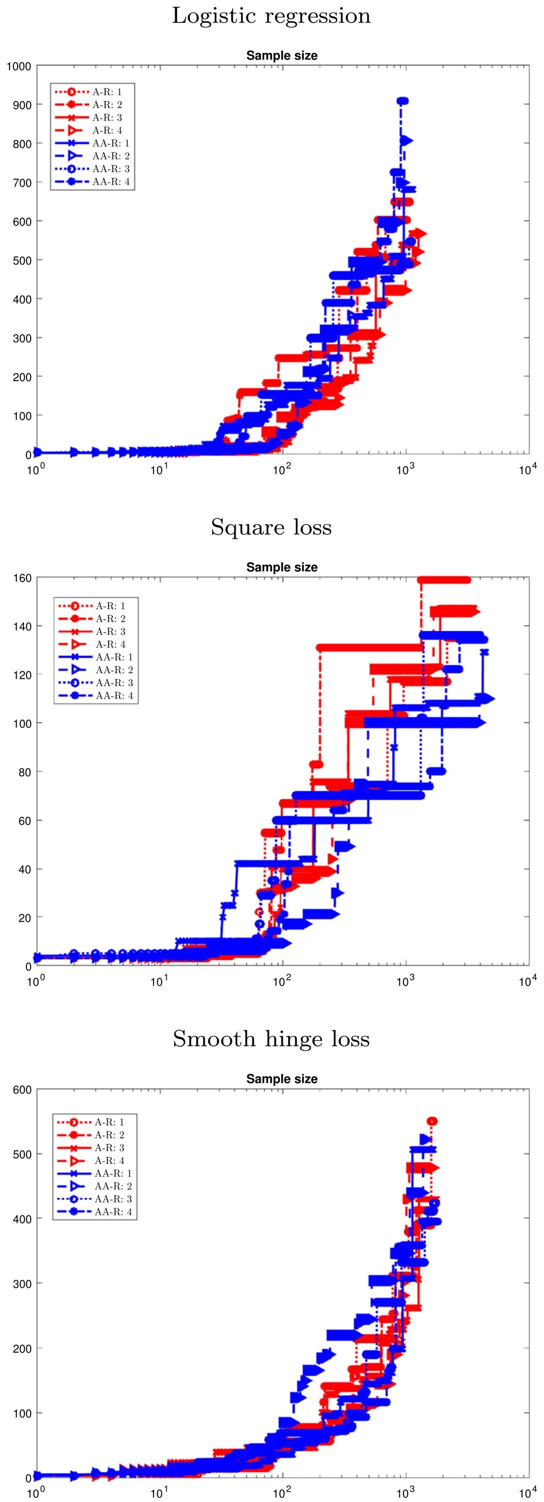
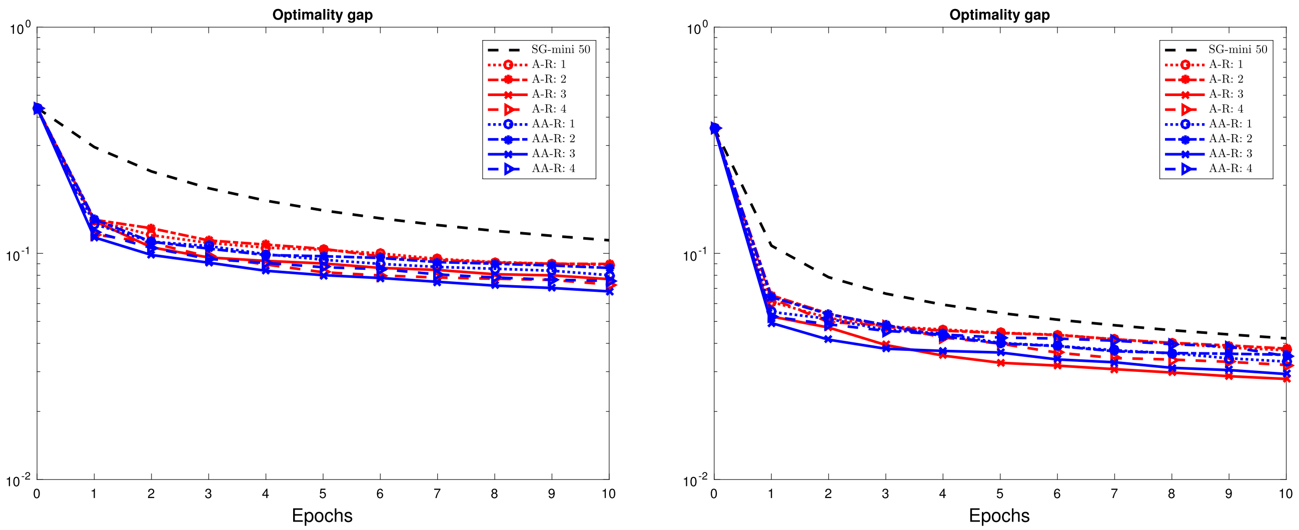


Fig. 5 Mini-batch size in **A-R** and **AA-R** methods on the *w8a* data set with respect to the iterations



(a) Logistic regression loss with smaller steplength than the best-tuned

(b) Smooth hinge loss with smaller steplength than the best-tuned

Fig. 6 Comparison between **SG-mini** with respect to **A-R** and **AA-R** in 10 epochs on the *MNIST* data set

approaches appear less dependent on an optimal setting of the parameters and they enable us to obtain smaller optimality gap values after the same number of epochs exploited by **SG mini**.

Furthermore, we observe that in **A-R** method, the behavior of the optimality gap is more stable than in **AA-R** method. Nevertheless, **AA-R** method can produce a smaller optimality gap at the end of 10 epochs.

4.2 Some further experiments and remarks on convex problems

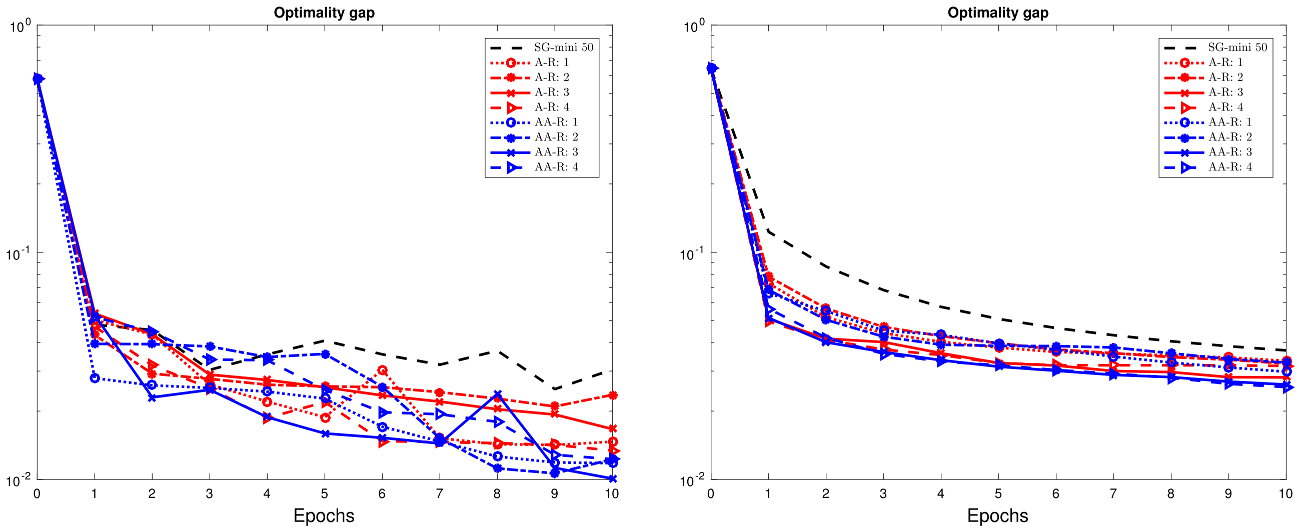
In the previous experiments, **A-R** and **AA-R** methods are equipped with the approximated version of the augmented inner test, based on sample statistics. For small samples, the conditions (25)–(27) may not be reliable enough in providing a sample size able to control the errors in the gradient estimates; indeed, in presence of noise, the norm of the current stochastic gradient $g_k^{(n_k)}$ can be greater than $\|\nabla F(x^{(k)}\|$, so that the conditions (25)-(27) are verified for many iterations before producing an increase in the sample size. To prevent this drawback, in Bollapragada et al. (2018), when the sample size does not change for at least r consecutive iterations, an average vector of the last r sample gradients is computed:

$$g_{\text{avg}} = \frac{1}{r} \sum_{j=k-r+1}^{k} g_j^{(n_j)}. \tag{30}$$

When $\|g_{\text{avg}}\| < \gamma \|g_k^{(n_k)}\|$, for a prefixed $\gamma \in (0, 1)$, the augmented inner product test is performed by replacing the current stochastic vector with the average vector g_{avg}; the possible consequence is an increase in the sample size. Typical values for r and γ are 10 and 0.38, respectively. For more details, also on this special setting, see Bollapragada et al. (2018); here, this practical procedure is viewed as a *recovery strategy* to improve the stability of **SG** method equipped with a line search rule for providing a suitable steplength. On the other hand, after some epochs, the effectiveness of the method can degrade for faster increase in the sequence $\{n_k\}$, although the adoption of the recovery procedure makes smaller the total number of backtracking steps.

In order to highlight this remark, in Fig. 8 we show the results obtained for *MNIST* when the problem (29) with logistic regression function is addressed by **SG** method equipped with a simple line search. In particular, we report the optimality gap with respect to 10 epochs when the augmented inner product test is coupled with the recovery procedure (magenta line) and without this recovery procedure (green line). In the latter case, the final sample size is 48 with a large number of backtracking steps (2700), while in the former one the sample size increases until 3300 with very few backtracking steps (110). As a consequence, the recovery procedure appears crucial for the control of the effectiveness of the line search and the sequence $\{n_k\}$.

The numerical results of the previous section show that **A-R** and **AA-R** methods are less dependent on the lack of reliability of the augmented inner product test for small values of n_k. Nevertheless, we can introduce the recovery procedure when the computation of Ritz-like values gives rise to $m_R = 0$ and the steplength at the next iteration is set to a tentative value $\overline{\alpha}$. More precisely, when this situation occurs, if the sample size has not changed in the last r iterations, the novel sample size is determined by using the approximated



(a) Logistic regression loss with greater steplength than the best-tuned

(b) Square loss with smaller steplength than the best-tuned

Fig. 7 Comparison between **SG-mini** with respect to **A-R** and **AA-R** in 10 epochs on the *w8a* data set

augmented inner product test with $g_k^{(n_k)}$ replaced by the average vector (30). In Figs. 9 and 10 , we show the behavior of the optimality gap obtained by using the modified versions of **A-R** and **AA-R** methods and **SG** method equipped with the line search rule for *MNIST* and *w8a*, respectively, in the case of the three loss functions in the objective. The comparison with Figs. 2 and 3 allows to observe that the recovery procedure improves the stability of **A-R** and **AA-R** methods with respect to the setting of $\alpha_{\min}$ and $\alpha_{\max}$, preserving the effectiveness of the approach. Indeed the accuracy of the two versions at the end of 10 epochs differs at most to the third decimal digit. The final value of the sample size is at most 10 times the one obtained without the use of the recovery procedure. As already observed in the previous section, **AA-R** method allows to obtain better results with respect to **A-R** in most experiments.

Furthermore, we observe that the performance of our approach appears generally better with respect to **SG** with a line search procedure. The comparison is carried out by considering only the number of scalar products performed in all methods, that is n scalar products for each epoch. Indeed, for the considered loss functions, the computational cost of the evaluation of the stochastic gradient $g_k^{(n_k)}$ is essentially given by the n_k scalar products $a_i^{\mathrm{T}} x^{(k)}$, $i \in S_k$.

As regards **SG** method with a line search rule, we observe that, although the evaluation of an estimate of the objective function $\frac{1}{n_k}\sum_{i\in S_k} f_i(x^{(k)})$ at $x^{(k)}$ does not require additional scalar products and it is negligible, the computation of the same estimate at $x^{(k)} - \alpha g_k^{(n_k)}$ requires at least additional n_k scalar products. Thus, each iteration of **SG** with line search has a computational cost at least equal to two evaluations of the stochastic gradient on the same sample. Any backtracking step increases the count of total scalar products. This preliminary analysis appears to favor schemes that avoid a line search rule for the determination of the steplength, also in the case of a few epochs when the sample size remains low. This topic may be the subject of future investigations.

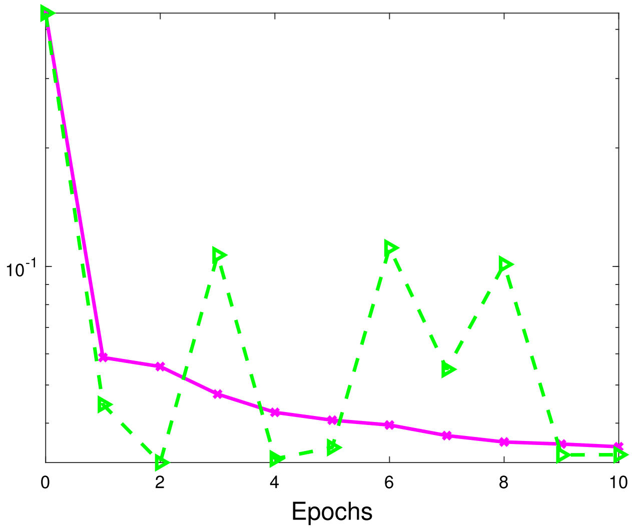


Fig. 8 Behavior of the optimality gap in 10 epochs for **SG** method equipped with a line search rule; magenta line is related to the version of the method combined with the recovery procedure while the green line is used for the version without this procedure. The parameters are chosen as in Bollapragada et al. (2018). In the experiment, logistic regression is the loss function and *MNIST* is the data set

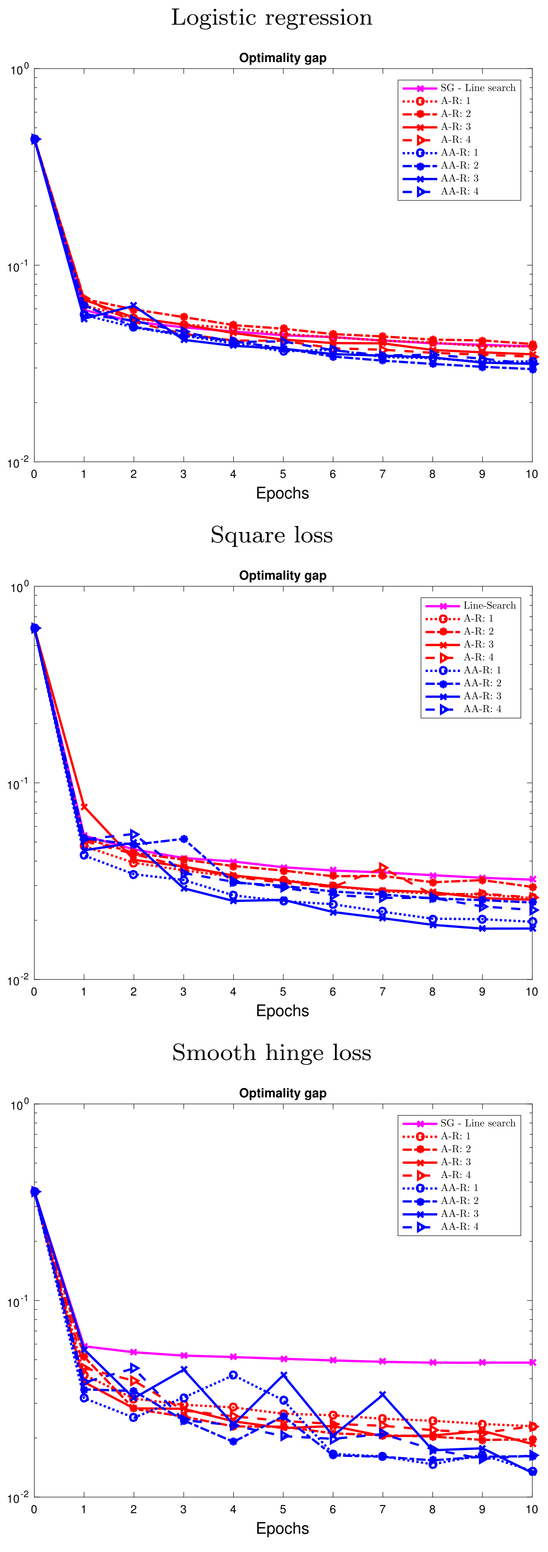


Fig. 9 Behavior of the optimality gap in 10 epochs for the versions of **A-R** and **AA-R** methods using with the recovery procedure and **SG** equipped with a line search rule in the case of the *MNIST* data set

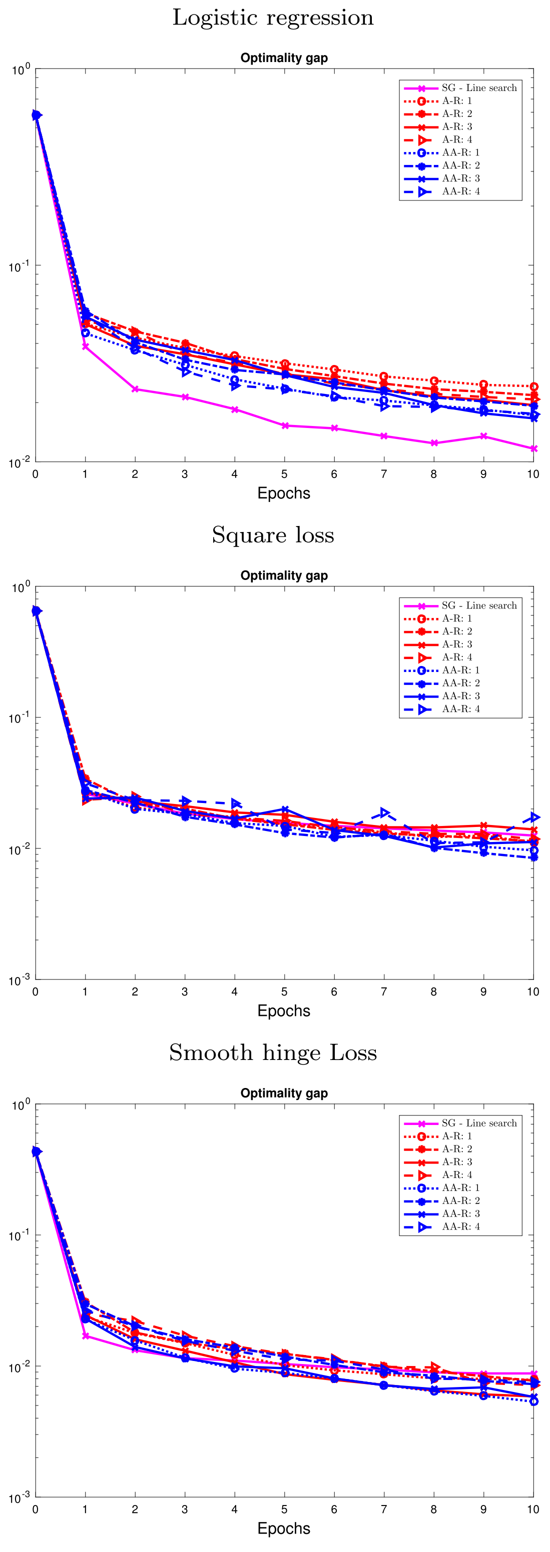


Fig. 10 Behavior of the optimality gap in 10 epochs for the versions of **A-R** and **AA-R** methods using with the recovery procedure and **SG** equipped with a line search rule in the case of the *w8a* data set

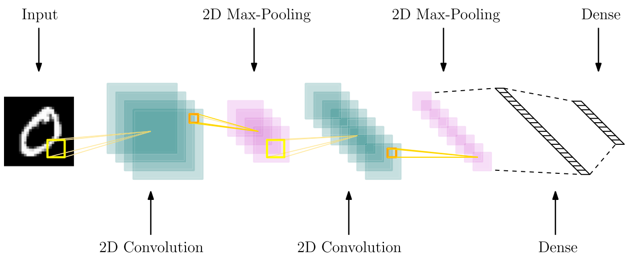


Fig. 11 Artificial neural network structure

4.3 A non-convex problem: a convolutional neural network

In the non-convex case, we consider as loss function an artificial neural network. In particular, dealing with image classification, we consider a convolutional neural network (CNN). The network is composed of an input layer, two sequences of convolutional and max-pooling layers, a fully connected layer and an output layer. We make use of rectified linear unit (ReLU) activations combined by a softmax function for the output layer and of a cross entropy as loss function (see Fig. 11). We consider the optimization problem arising in training a multi-class classifier for the *MNIST* data set.

We compare the effectiveness of the same methods considered in the previous section, i.e., **SG mini**, **A-R** and **AA-R** methods. In all the numerical experiments we use the following setting:

- regularization parameter $\delta = 10^{-4}$;
- the first convolutive layer is composed by 64 filters, each filter has 5×5 dimension; after we apply a max-pooling of size 2×2;
- the second convolutive layer is composed by 32 filters, each filter has 5×5 dimension; after we apply a max-pooling of size 2×2;
- in **SG mini**, the size of the mini-batch, is set as $|S| = 50$;
- in **A-R** and **AA-R** methods, the length of any sweep is at most $m = 3$; furthermore, $\theta = 0.7$ in (25) and $\nu = 7$ in (27) for all the numerical simulations;

The numerical experiments were carried out in Matlab® on Intel(R) Core(TM) i7-6700 CPU @ 3.40 GHz with 8 CPUs.

In Fig. 12, we can observe the different accuracies (with respect to the testing set) provided by the CNN trained with **SG mini** in 5 epochs; the fixed steplength is set to values between 0.001 and 0.9. As we can see, the method provides effective results for $\alpha = 0.5$; a similar accuracy is obtained for $\alpha = 0.1$. In cases with smaller steplengths, the accuracy in 5 epochs is unsatisfactory, while a higher steplength can lead to the divergence of the method. Hence, in a more marked way than the convex case, for non-convex problems, finding an effective steplength requires a very expensive trial procedure. Conversely, using a random steplength without a prior trial phase can lead to inaccurate results due to slow convergence or divergence of the method.

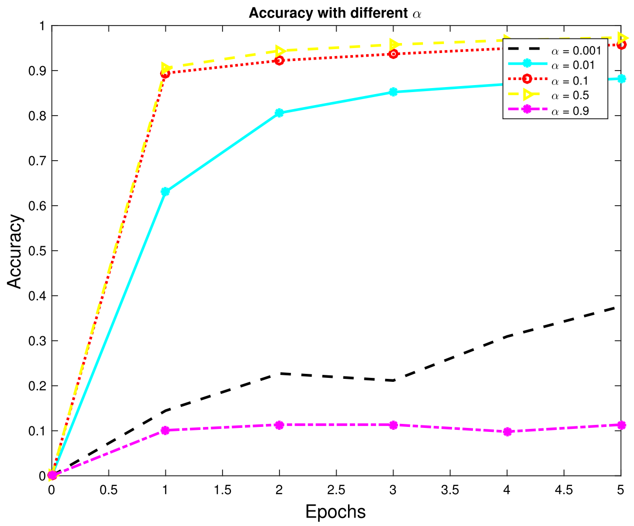


Fig. 12 CNN Accuracy in the SG mini case

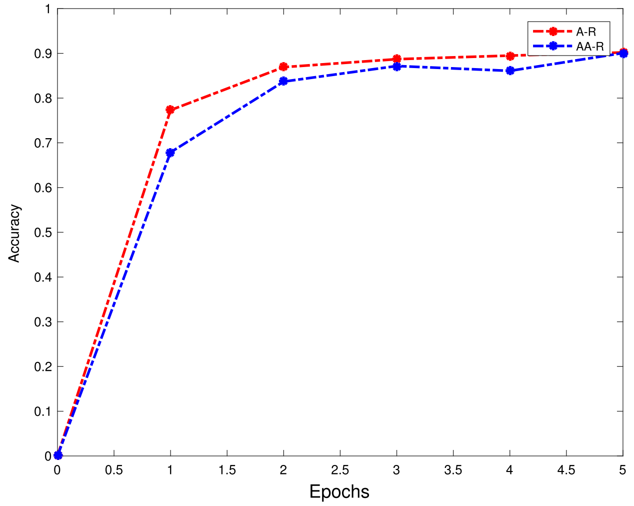


Fig. 13 Accuracy obtained by training the CNN with **A-R** and **AA-R** methods

In Fig. 13, we report the results obtained by training the CNN with the **A-R** and **AA-R** methods. In particular, we show the behavior of the accuracy with respect to the testing set in the first 5 epochs with the following settings:

- for **A-R** method, $\alpha_{\min} = 10^{-3}$, $\alpha_{\max} = 1$, $n_0 = 10$;
- for **AA-R** method, $\alpha_{\min} = 10^{-2}$, $\alpha_{\max} = 1$, $n_0 = 3$.

The parameter $\overline{\alpha}$ is set as 0.1 in all cases. We observe that **A-R** appears more robust with respect to the amplitude of the interval where α_k can belong. Furthermore, we notice that the subsample size increases up to a maximum of 204 and 182 in **A-R** and **AA-R** methods, respectively.

5 Conclusions

In this paper we proposed to tailor the steplength selection rule based on the Ritz-like values, used successfully in the deterministic gradient schemes, to a stochastic scheme, recently proposed by Bollapragada et al. (2018). This SG method includes an adaptive subsampling strategy, aimed to control the variance of the stochastic directions. We observed that the theoretical properties of this approach hold under the assumption that the steplength selection rule obeys to the assumption $\alpha_k \in (\alpha_{\min}, \alpha_{\max}]$, where $\alpha_{\max}$ is proportional to the inverse of the Lipschitz parameter of the objective function gradient. Consequently, we reformulate the procedure for obtaining the Ritz-like values in the stochastic framework, by using the stochastic gradients instead of the standard gradients. It is required that these stochastic directions, although based on different subsamples, satisfy two conditions (the inner product test and the orthogonality test), ensuring the descent property in expectation. In particular, we proposed two different ways to select the current steplength, by simply toggling the Ritz-like values with the harmonic Ritz-like values (**A-R** method) or using the harmonic Ritz-like values only when the size of the subsample is increased (**AA-R** method). The numerical experimentation highlighted that the proposed methods enable to obtain an accuracy similar to the one obtained with SG mini-batch with fixed best-tuned steplength. Although also in this case it is necessary to carefully select a thresholding range for the steplengths, the proposed approach appears slightly dependent on the bounds imposed on the steplengths, making the parameters setting less expensive with respect to the SG framework. In conclusion, the proposed technique provides a guidance on the learning rate selection and it allows to perform similarly to the SG approach equipped with the best-tuned steplength.

Acknowledgements We thank the anonymous reviewers for their careful reading of our manuscript and their useful comments and suggestions. This work has been partially supported by the INdAM research group GNCS. The authors are members of the INDAM research group GNCS.

Compliance with ethical standards

Conflict of interest Giorgia Franchini, Valeria Ruggiero and Luca Zanni declare that they have no conflict of interest.

Human and animal rights statement This article does not contain any studies with human participants or animals performed by any of the authors.

References

Bellavia S, Gurioli G, Morini B, Toint PL (2019) Adaptive regularization algorithms with inexact evaluations for nonconvex optimization. SIAM J Optim 29(4):2881–2915

Bollapragada R, Byrd R, Nocedal J (2018) Adaptive sampling strategies for stochastic optimization. SIAM J Optim 28(4):3312–3343

Bottou L, Curtis FE, Nocedal J (2018) Optimization methods for large-scale machine learning. SIAM Rev 60(2):223–311

Byrd RH, Chin GM, Nocedal J, Wu Y (2012) Sample size selection in optimization methods for machine learning. Math Program 1(134):127–155

Cartis C, Scheinberg K (2015) Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. Math Program 1:1–39

Curtis FE, Guo W (2016) Handling nonpositive curvature in a limited memory steepest descent method. IMA J Numer Anal 36(2):717–742. https://doi.org/10.1093/imanum/drv034

Dai YH, Yuan Y (2003) Alternate minimization gradient method. IMA J Numer Anal 23:377–393

Defazio A, Bach FR, Lacoste-Julien S (2014) SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives. In: NIPS

di Serafino D, Ruggiero V, Toraldo G, Zanni L (2018) On the steplength selection in gradient methods for unconstrained optimization. Appl Math Comput 318:176–195

Fletcher R (2012) A limited memory steepest descent method. Math Program Ser A 135:413–436

Franchini G, Ruggiero V, Zanni L (2020) On the steplength selection in Stochastic Gradient Methods. In: Sergeyev YD, Kvasov DE (eds) Numerical computations: theory and algorithms (NUMTA, 2019). Lecture notes in computer science, vol 11973. Springer, Berlin

Frassoldati G, Zanghirati G, Zanni L (2008) New adaptive stepsize selections in gradient methods. J Ind Manag Optim 4(2):299–312

Friedlander MP, Schmidt M (2012) Hybrid deterministic-stochastic methods for data fitting. SIAM J Sci Comput 34(3):A1380–A1405

Hashemi F, Ghosh S, Pasupathy R (2014) In adaptive sampling rules for stochastic recursions. In: Simulation conference (WSC) 2014, Winter, pp 3959–3970

Johnson R, Zhang T (2013) Accelerating stochastic gradient descent using predictive variance reduction. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ (eds) Advances in neural information processing systems, vol 26. Curran Associates Inc, Red Hook, pp 315–323

Karimi H, Nutini J, Schmidt M, (2016) Linear convergence of gradient and proximal-gradient methods under the Polyak–Łojasiewicz condition. In: Frasconi P, Landwehr N, Manco G, Vreeken J (eds) Machine learning and knowledge discovery in databases ECML PKDD 2016. Lecture notes in computer science, vol 9851. Springer, Berlin

Tan C, Ma S, Dai Y, Qian Y (2016) BB step size for SGD. In: Lee D, Sugiyama M, Luxburg U, Guyon I, Garnett R (eds) Advances in neural information processing systems 29 (NIPS 2016). Springer, Berlin

Tropp JA (2015) An introduction to matrix concentration inequalities. Found Trends Mach Lear 8(1–2):1–230. https://doi.org/10.1561/2200000048

Yang Z, Wang C, Zang Y, Li J (2018) Mini-batch algorithms with Barzilai–Borwein update step. Neurocomputing 314:177–185

Zhou B, Gao L, Dai YH (2006) Gradient methods with adaptive stepsizes. Comput Optim Appl 35(1):69–86

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.