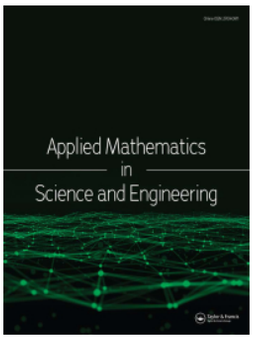

Applied Mathematics in Science and Engineering

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/gipe21

Learning rate selection in stochastic gradient methods based on line search strategies

Giorgia Franchini, Federica Porta, Valeria Ruggiero, Ilaria Trombini & Luca Zanni

To cite this article: Giorgia Franchini, Federica Porta, Valeria Ruggiero, Ilaria Trombini & Luca Zanni (2023) Learning rate selection in stochastic gradient methods based on line search strategies, Applied Mathematics in Science and Engineering, 31:1, 2164000, DOI: 10.1080/27690911.2022.2164000

To link to this article: https://doi.org/10.1080/27690911.2022.2164000

Published online: 09 Jan 2023.

APPLIED MATHEMATICS IN SCIENCE AND ENGINEERING
2023, VOL. 31, NO. 1, 2164000
https://doi.org/10.1080/27690911.2022.2164000

Taylor & Francis
Taylor & Francis Group

Learning rate selection in stochastic gradient methods based on line search strategies

Giorgia Franchini[a], Federica Porta[a], Valeria Ruggiero[b], Ilaria Trombini[b,c] and Luca Zanni[a]

[a]Department of Physics, Informatics and Mathematics, University of Modena and Reggio Emilia, Modena, Italy; [b]Department of Mathematics and Computer Science, University of Ferrara, Ferrara, Italy; [c]Department of Mathematical, Physical and Computer Sciences, University of Parma, Parma, Italy

ABSTRACT

Finite-sum problems appear as the sample average approximation of a stochastic optimization problem and often arise in machine learning applications with large scale data sets. A very popular approach to face finite-sum problems is the stochastic gradient method. It is well known that a proper strategy to select the hyperparameters of this method (i.e. the set of a-priori selected parameters) and, in particular, the learning rate, is needed to guarantee convergence properties and good practical performance. In this paper, we analyse standard and line search based updating rules to fix the learning rate sequence, also in relation to the size of the mini batch chosen to compute the current stochastic gradient. An extensive numerical experimentation is carried out in order to evaluate the effectiveness of the discussed strategies for convex and non-convex finite-sum test problems, highlighting that the line search based methods avoid expensive initial setting of the hyperparameters. The line search based approaches have also been applied to train a Convolutional Neural Network, providing very promising results.

ARTICLE HISTORY
Received 27 July 2022
Accepted 26 December 2022

KEYWORDS
Stochastic gradient methods; variance reduced methods; learning rate selection; mini batch size selection; convolutional neural networks

MATHEMATICS SUBJECT CLASSIFICATIONS
68T07; 68T05; 46N10; 65K10

1. Introduction

In this paper, we consider the following optimization problem

$$\min_{x\in\mathbb{R}^d} F(x) \equiv \frac{1}{N}\sum_{i=1}^{N} f_i(x), \tag{1}$$

where each $f_i : \mathbb{R}^d \to \mathbb{R}$ is differentiable with L-Lipschitz continuous gradient. We are especially interested in the case where the number of components N is very large, and, hence, the adoption of stochastic gradient methods is convenient since they exploit either a single gradient ∇f_i or a very limited number of them at each iteration, rather than the entire gradient ∇F. The study of the minimization problem (1) is relevant since it often arises in machine learning applications where it is known as empirical risk minimization.

CONTACT Giorgia Franchini giorgia.franchini@unimore.it Department of Physics, Informatics and Mathematics, University of Modena and Reggio Emilia, Via Campi 213/B, 41125 Modena (MO), Italy

In this framework, N represents the number of samples and f_i is the single cost function corresponding to the ith sample.

A classical stochastic approach to solve (1) is the mini batch stochastic gradient (SG) method [1,2], which, given $x^{(0)} \in \mathbb{R}^d$, is defined as

$$x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)}_{\mathcal{N}_k}, \tag{2}$$

with $g^{(k)}_{\mathcal{N}_k} = \frac{1}{N_k} \sum_{i \in \mathcal{N}_k} \nabla f_i(x^{(k)})$. Here $\mathcal{N}_k$ is a randomly chosen subset of $\{1, \ldots, N\}$ whose cardinality is denoted by N_k and α_k is a positive learning rate. The mini batch size N_k can be either a priori fixed or allowed to vary during the iterations.

It is well known in the literature that both the convergence properties and the practical performance of the algorithm (2) are strongly influenced by the learning rate selection rule. In this paper, we discuss some techniques to choose the sequence of the learning rate $\{\alpha_k\}$ in combination with proper strategies to fix the mini batch size along the iterative process. We compare standard learning rate selection rules to line search based approaches which exploit a convenient increase of the mini batch size aimed at imposing some useful properties on the stochastic directions. An extensive numerical experimentation enables to evaluate the effectiveness of the considered strategies for convex and non-convex test problems. The behaviour of the discussed approaches is also evaluated on the problem arising in training a Convolutional Neural Network (CNN) for multi-classification, highlighting the stability properties of the line search strategies with respect to the hyperparameter setting.

2. Learning rate selection rules for stochastic gradient method

In this section, we discuss several possibilities to choose the learning rate in the SG scheme (2) for both the fixed and the variable mini batch size cases.

2.1. Fixed mini batch size

For a deep survey on the convergence results of the SG scheme (2) with fixed mini batch size $N_k = \overline{N}$, the reader is referred to [2]. However, it is worth to recall that, under the assumption that the gradient of the objective function is Lipschitz continuous with parameter L and some additional conditions on the first and second moments of the stochastic gradient, when the positive learning rate satisfies $\alpha_k = \overline{\alpha} \leq \alpha_{\max}$, for a constant $\alpha_{\max}$ depending on L, the expected optimality gap for strongly convex objective functions, or the expected sum of gradients for general objective functions, asymptotically converges to values proportional to $\overline{\alpha}$. Roughly speaking, if the learning rate is sufficiently small, the method generates iterates in the neighbourhood of the optimum or the stationary point. Nevertheless, the constants related to the above mentioned assumptions, such as the Lipschitz parameter L, are either unknown or not easy to approximate. For this reason, these results do not give an idea of how to select the learning rate. Moreover a too small value of this hyperparameter can give rise to a very slow learning process. As a consequence, the learning rate is often manually tuned in the practice by means of an expensive trial and error procedure. Furthermore, we recall that, under the selection of a suitable diminishing learning rate $\alpha_k = \mathcal{O}(1/k)$ and a fixed mini batch size, the expected value of the optimality gap generated by the SG method (2) for strongly convex objective functions, or the expected

sum of gradients for general objective functions, converges to 0 at a sublinear rate $\mathcal{O}(1/k)$ [2]. Unfortunately, to select a diminishing sequence $\{\alpha_k\}$ is not efficient as well and the starting value of the learning rate α_0 has to satisfy suitable assumptions.

An attempt to overcome this difficulty has been made in [3,4] where the Barzilai-Borwein (BB) rule, very often exploited to select the learning rate for the deterministic gradient methods, has been adapted to the stochastic setting. However, unlike in the deterministic framework, the generalized BB update requires either proper smoothing technique [4] to diminish the current learning rate α_k while running the algorithm or a thresholding procedure on it, based on user dependent bounds $[\alpha_{\min}/k, \alpha_{\max}/k]$ [3] to avoid instability and ensure the convergence.

2.2. Variable mini batch size

For twice differentiable objective function such that $\mu I \preceq \nabla^2 F \preceq LI$, $\mu > 0$, a way to obtain the linear convergence for the SG method (2) consists in increasing the size of the current mini batch N_k at a geometric rate [5], provided that the learning rate is fixed as a positive value bounded from above by $\frac{1}{L}$. This approach has two drawbacks: the size of the mini batch increases too rapidly and the constant L is typically not known, as already said before. In order to overcome these drawbacks we report in the following two strategies developed very recently [6,7].

In [6] the authors suggest to increase the mini batch size in (2) on the basis of two conditions imposed on the stochastic directions. These conditions, called inner product test and orthogonality test, guarantee that the search directions computed on a mini batch of suitable size are descent directions with high probability:

$$\mathbb{E}\left[\left(g_{\mathcal{N}_k{}^{(k)}}^T \nabla F(x^{(k)}) - \left\|\nabla F(x^{(k)})\right\|^2\right)^2\right] \leq \theta^2 \left\|\nabla F(x^{(k)})\right\|^4, \tag{3}$$

$$\mathbb{E}\left[\left\|g_{\mathcal{N}_k}^{(k)} - \frac{g_{\mathcal{N}_k}^{(k)\,T} \nabla F(x^{(k)})}{\left\|\nabla F(x^{(k)})\right\|^2} \nabla F(x^{(k)})\right\|^2\right] \leq \nu^2 \left\|\nabla F(x^{(k)})\right\|^2, \tag{4}$$

where θ and ν are prefixed positive values. For twice differentiable objective functions such that $\mu I \preceq \nabla^2 F \preceq LI$, if the inner product test and the orthogonality test are fulfilled and the learning rate is fixed at each iteration and bounded from above by a constant depending on L, θ and ν, then the SG scheme (2) is linearly convergent. Under the same assumptions on the mini batch size and the learning rate, weaker theoretical convergence properties hold for more general objective functions. Since these results strongly depend on the knowledge of the Lipschitz parameter L, a line search procedure has been devised for its numerical estimation. The resulting algorithm is called the Adaptive Sampling Method (ASM).

In the deterministic setting [8–11], the BB rule has been proved to give a local estimate of the inverse of the Lipschitz constant of ∇F. Motivated by this observation, we decide to also consider a modified version of ASM, called ASM-BB1, where the learning rate is no more fixed by means of the line search procedure but through the BB rule developed for the stochastic setting in [4]. The value for α_k is kept fixed until the size of the mini batch changes in according to the conditions (3) and (4). When the size of the mini batch

is increased by the inner product test and the orthogonality test, a new BB learning rate is computed. This version of ASM is denoted by ASM-BB1 in the following. In [12] a similar approach to that of ASM-BB1 is adopted. The authors developed two SG methods of the form (2) where the increase of the mini batch size is controlled by means of (3) and (4) and the learning rate is computed through two different generalized BB rules based on the so called Ritz-like and harmonic Ritz-like values, respectively. Hereafter, these approaches will be denoted by ASM-A-R and ASM-AA-R, respectively.

An idea similar to that on which ASM is based, has been followed in [7]. Here the authors developed a proximal stochastic gradient algorithm to face a regularized version of problem (1); however, if the function to minimize is defined as in (1), such algorithm belongs to the class of mini batch SG methods (2) with variable mini batch size. In more detail, the mini batch $\mathcal{N}_k$ is selected such that the variance of the stochastic gradients is dynamically reduced along the iterative process, namely,

$$\mathbb{E}_k[\|g_{\mathcal{N}_k}^{(k)} - \nabla F(x^{(k)})\|^2] \leq \varepsilon_k, \quad \varepsilon_k \geq 0, \ \forall\, k \quad \text{and} \quad \sum_k \varepsilon_k < \infty \quad \text{a.s.}, \tag{5}$$

where $\mathbb{E}_k[\cdot]$ denotes the conditional expected value with respect to the σ-algebra generated by the information collected before iteration k, i.e. assuming $x^{(0)}, \ldots, x^{(k)}$ given. Hereafter we provide two convergence results for the scheme (2) combined with condition (5): the first one is more general since it allows the objective function to be non-convex. This convergence analysis is different to the one developed in [7] and it is especially tailored for the SG method (2) applied to the class of optimization problems (1). Before presenting the main results, we need to recall a classical result from stochastic analysis.

Lemma 2.1 ([13, Lemma 11]): *Let v_k, u_k, α_k, β_k be nonnegative random variables and let*

$$\mathbb{E}_k[v_{k+1}] \leq (1+\alpha_k)v_k - u_k + \beta_k \quad \text{a.s.}$$
$$\sum_{k=0}^{\infty} \alpha_k < \infty \quad \text{a.s.}, \quad \sum_{k=0}^{\infty} \beta_k < \infty \quad \text{a.s.},$$

where $\mathbb{E}_k[v_{k+1}]$ denotes the conditional expectation for the given $v_0, \ldots, v_k$, $u_0, \ldots, u_k$, $\alpha_0, \ldots, \alpha_k$, $\beta_0, \ldots, \beta_k$. Then

$$v_k \longrightarrow v \quad \text{a.s,} \quad \sum_{k=0}^{\infty} u_k < \infty \quad \text{a.s,}$$

where $v \geq 0$ is some random variable.

Theorem 2.2: *Let $\{x^{(k)}\}$ be the sequence generated by (2) with $0 < \alpha_k = \alpha < \frac{2}{L}$. If $g_{\mathcal{N}_k}^{(k)}$ is an unbiased estimate of $\nabla F(x^{(k)})$ and condition (5) holds, then $\|\nabla F(x^{(k)})\| \to 0$ a.s.*

Proof: In view of (2) and the L-Lipschitz continuity of ∇F, we have that

$$\begin{aligned}
F(x^{(k+1)}) &\leq F(x^{(k)}) + \nabla F(x^{(k)})^T(x^{(k+1)} - x^{(k)}) + \frac{L}{2}\|x^{(k+1)} - x^{(k)}\|^2 \\
&= F(x^{(k)}) - \alpha \nabla F(x^{(k)})^T g_{\mathcal{N}_k}^{(k)} + \frac{L\alpha^2}{2}\|g_{\mathcal{N}_k}^{(k)}\|^2 \\
&= F(x^{(k)}) - \alpha \nabla F(x^{(k)})^T g_{\mathcal{N}_k}^{(k)} + \frac{L\alpha^2}{2}\|g_{\mathcal{N}_k}^{(k)} \pm \nabla F(x^{(k)})\|^2 \\
&= F(x^{(k)}) - \alpha \nabla F(x^{(k)})^T g_{\mathcal{N}_k}^{(k)} + \frac{L\alpha^2}{2}\|g_{\mathcal{N}_k}^{(k)} - \nabla F(x^{(k)})\|^2 + \\
&\quad + \frac{L\alpha^2}{2}\|\nabla F(x^{(k)})\|^2 + L\alpha^2 \left(g_{\mathcal{N}_k}^{(k)} - \nabla F(x^{(k)})\right)^T \nabla F(x^{(k)}). \qquad (6)
\end{aligned}$$

By taking the conditional expectation on both sides of inequality (6) and recalling that $\mathbb{E}_k[g_{\mathcal{N}_k}^{(k)}] = \nabla F(x^{(k)})$ and, hence, $\mathbb{E}_k[g_{\mathcal{N}_k}^{(k)} - \nabla F(x^{(k)})] = 0$, we obtain that

$$\begin{aligned}
\mathbb{E}_k[F(x^{(k+1)})] &\leq F(x^{(k)}) - \alpha\|\nabla F(x^{(k)})\|^2 + \frac{L\alpha^2}{2}\|\nabla F(x^{(k)})\|^2 \\
&\quad + \frac{L\alpha^2}{2}\mathbb{E}_k[\|g_{\mathcal{N}_k}^{(k)} - \nabla F(x^{(k)})\|^2] \\
&\quad + L\alpha^2\mathbb{E}_k\left[\left(g_{\mathcal{N}_k}^{(k)} - \nabla F(x^{(k)})\right)^T \nabla F(x^{(k)})\right] \\
&= F(x^{(k)}) - \left(\alpha - \frac{L\alpha^2}{2}\right)\|\nabla F(x^{(k)})\|^2 + \frac{L\alpha^2}{2}\mathbb{E}_k[\|g_{\mathcal{N}_k}^{(k)} - \nabla F(x^{(k)})\|^2].
\end{aligned}$$

Since $\alpha < \frac{2}{L}$ and condition (5) holds, Lemma 2.1 can be invoked. Therefore

$$\sum_k \|\nabla F(x^{(k)})\|^2 < +\infty \quad \text{a.s.}$$

and the theorem is proved. ■

The above theorem enables us to affirm that if $\{x^{(k)}\}$ has a limit point, then this point is a stationary point for F a.s.

Theorem 2.3: *Let $\{x^{(k)}\}$ be the sequence generated by (2) with $0 < \alpha_k = \alpha < \frac{1}{L}$. If $g_{\mathcal{N}_k}^{(k)}$ is an unbiased estimate of $\nabla F(x^{(k)})$, condition (5) holds true, the function F is convex and the solution set X^* of problem (1) is not empty, then $\{x^{(k)}\}$ converges to a solution of (1) a.s.*

Proof: Let $x^* \in X^*$. We observe that

$$\begin{aligned}
\|x^{(k+1)} - x^*\|^2 &= \|x^{(k+1)} \pm x^{(k)} - x^*\|^2 \\
&= \|x^{(k+1)} - x^{(k)}\|^2 + \|x^{(k)} - x^*\|^2 + 2(x^{(k+1)} - x^{(k)})^T(x^{(k)} - x^*) \\
&= \|x^{(k)} - x^*\|^2 - 2\alpha {g_{\mathcal{N}_k}^{(k)}}^T(x^{(k)} - x^*) + \alpha^2\|g_{\mathcal{N}_k}^{(k)}\|^2 \\
&= \|x^{(k)} - x^*\|^2 - 2\alpha {g_{\mathcal{N}_k}^{(k)}}^T(x^{(k)} - x^*) + \alpha^2\|g_{\mathcal{N}_k}^{(k)} \pm \nabla F(x^{(k)})\|^2 \qquad (7)
\end{aligned}$$

where the third equality follows from the definition of $x^{(k+1)}$ in (2). By taking the conditional expectation on both sides of inequality (7), we can write that

$$\begin{aligned}\mathbb{E}_k[\|x^{(k+1)} - x^*\|^2] &= \|x^{(k)} - x^*\|^2 - 2\alpha \nabla F(x^{(k)})^T (x^{(k)} - x^*) + \alpha^2 \|\nabla F(x^{(k)})\|^2 \\ &\quad + \alpha^2 \mathbb{E}_k[\|g^{(k)}_{\mathcal{N}_k} - \nabla F(x^{(k)})\|^2] \\ &\leq \|x^{(k)} - x^*\|^2 - 2\alpha (F(x^{(k)}) - F(x^*)) + \alpha^2 \|\nabla F(x^{(k)})\|^2 \\ &\quad + \alpha^2 \mathbb{E}_k[\|g^{(k)}_{\mathcal{N}_k} - \nabla F(x^{(k)})\|^2] \\ &\leq \|x^{(k)} - x^*\|^2 - 2\alpha (1 - \alpha L)(F(x^{(k)}) - F(x^*)) + \\ &\quad + \alpha^2 \mathbb{E}_k[\|g^{(k)}_{\mathcal{N}_k} - \nabla F(x^{(k)})\|^2]\end{aligned}$$

where the first inequality follows from the convexity of F and the second inequality follows from the fact that the L-Lipschitz continuity of ∇F implies that $\|\nabla F(x)\|^2 \leq 2L(F(x) - F(x^*))$. From the hypotheses on α_k and the conditional expected value of the variance of the stochastic gradient, Lemma 2.1 can be applied and we can state that the sequence $\{\|x^{(k)} - x^*\|\}_{k\in\mathbb{N}}$ converges a.s.

Next we prove the almost sure convergence of the sequence $\{x^{(k)}\}$ by following a strategy similar to the one employed in [14, Theorem 2.1]. Let $\{x_i^*\}_i$ be a countable subset of the relative interior $\mathrm{ri}(X^*)$ that is dense in X^*. From the almost sure convergence of $\|x^{(k)} - x^*\|$, $x^* \in X^*$, we have that for each i, the probability $\mathrm{Prob}(\{\|x^{(k)} - x_i^*\|\} \text{ is not convergent}) = 0$. Therefore, we observe that

$$\begin{aligned}&\mathrm{Prob}(\forall\, i\, \exists b_i \text{ s.t. } \lim_{k\to+\infty} \|x^{(k)} - x_i^*\| = b_i) = 1 - \mathrm{Prob}(\{\|x^{(k)} - x_i^*\|\} \text{ is not convergent}) \\ &\geq 1 - \sum_i \mathrm{Prob}(\{\|x^{(k)} - x_i^*\|\} \text{ is not convergent}) = 1,\end{aligned}$$

where the inequality follows from the union bound, i.e. for each i, $\{\|x^{(k)} - x_i^*\|\}$ is a convergent sequence a.s. For a contradiction, suppose that there are convergent subsequences $\{u_{k_j}\}_{k_j}$ and $\{v_{k_j}\}_{k_j}$ of $\{x^{(k)}\}$ which converge to their limiting points u^* and v^* respectively, with $\|u^* - v^*\| = r > 0$. By Theorem 2.2, u^* and v^* are stationary; in particular, since F is convex, they are minimum points, i.e. $u^*, v^* \in X^*$. Since $\{x_i^*\}_i$ is dense in X^*, we may assume that for all $\epsilon > 0$, we have $x_{i_1}^*$ and $x_{i_2}^*$ are such that $\|x_{i_1}^* - u^*\| < \epsilon$ and $\|x_{i_2}^* - v^*\| < \epsilon$. Therefore, for all k_j sufficiently large,

$$\|u_{k_j} - x_{i_1}^*\| \leq \|u_{k_j} - u^*\| + \|u^* - x_{i_1}^*\| < \|u_{k_j} - u^*\| + \epsilon.$$

On the other hand, for sufficiently large j, we have

$$\|v_{k_j} - x_{i_1}^*\| \geq \|v^* - u^*\| - \|u^* - x_{i_1}^*\| - \|v_{k_j} - v^*\| > r - \epsilon - \|v_{k_j} - v^*\| > r - 2\epsilon.$$

This contradicts with the fact that $x^{(k)} - x_{i_1}^*$ is convergent. Therefore, we must have $u^* = v^*$, hence there exists $\bar{x} \in X^*$ such that $x^{(k)} \longrightarrow \bar{x}$. ■

The previous theorems show that the convergence of the sequence generated by algorithm (2) can be obtained provided that condition (5) holds and the learning rate α_k is

bounded by a constant depending on the Lipschitz constant of the gradient of the objective function. Since both these requirements seem difficult to be preserved in practice, we now detail how to select the stochastic gradients and the learning rate along the iterations in order to realize them. The resulting method is a Line search based Stochastic first order Algorithm (LISA) and it is stated in Algorithm 1. Some explanations are needed.

Algorithm 1 - LISA

Given $x^{(0)} \in \mathbb{R}^d$, $0 < N_0 < N$, $\beta \in (0,1)$, $0 < \alpha_{\min} < \alpha_{\max}$ and a nonnegative sequence $\{\varepsilon_k\}_{k\in\mathbb{N}}$, $\sum_{k=0}^{+\infty} \varepsilon_k < \infty$.

For $k = 0, 1, 2, \ldots$

Step 1. Choose a sample $\mathcal{N}_k$ of size N_k and compute $g_{\mathcal{N}_k}(x^{(k)})$.

If

$$V^{(k)} = \frac{1}{N_k(N_k-1)} \sum_{i\in\mathcal{N}_k} \|\nabla f_i(x^{(k)}) - g_{\mathcal{N}_k}(x^{(k)})\|^2 \leq \varepsilon_k \quad \text{or} \quad N_k \geq N$$

Then go to Step 2.

Else set $N_k = \min\left\{N, \max\left\{\frac{N_k V^{(k)}}{\varepsilon_k}, N_k + 1\right\}\right\}$ and go to Step 1.

Step 2. Compute $F_{\mathcal{N}_k}(x^{(k)}) = \sum_{i\in\mathcal{N}_k} f_i(x^{(k)})$ and set $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$.

Step 3. Set $\bar{x}^{(k)} = x^{(k)} - \alpha_k g_{\mathcal{N}_k}(x^{(k)})$.

If

$$F_{\mathcal{N}_k}(\bar{x}^{(k)}) \leq F_{\mathcal{N}_k}(x^{(k)}) - \frac{\alpha_k}{2}\|g^{(k)}_{\mathcal{N}_k}\|^2 \tag{8}$$

Then go to Step 4.

Else set $\alpha_k = \beta\alpha_k$ and go to Step 3.

Step 4. Set $x^{(k+1)} = \bar{x}^{(k)}$.

End For

The variance of the search directions is controlled through a dynamic increase of the size of the mini batch as described in Step 1. In more detail, under the assumption that $\mathbb{E}_k[\nabla f_i(x^{(k)})] = \nabla F(x^{(k)})$, $\forall\ i$, there exists a constant value $C \geq 0$ such that $\mathbb{E}_k(\|\nabla f_i(x^{(k)}) - \nabla F(x^{(k)})\|^2) \leq C$, $\forall\ i$. Hence, for an arbitrary $i \in \mathcal{N}_k$, we have that

[15, p. 183]

$$\mathbb{E}_k[\|g_{\mathcal{N}_k}(x^{(k)}) - \nabla F(x^{(k)})\|^2] \leq \frac{\mathbb{E}_k[\|\nabla f_i(x^{(k)}) - \nabla F(x^{(k)})\|^2]}{N_k} \leq \frac{C}{N_k}.$$

This bound, when combined with a suitable rate of increase in N_k, enables to guarantee condition (5). Indeed, it is sufficient that

$$\frac{\mathbb{E}_k[\|\nabla f_i(x^{(k)}) - \nabla F(x^{(k)})\|^2]}{N_k} \leq \varepsilon_k$$

with $N_k = \frac{C}{\varepsilon_k}$. Following a standard strategy [2,5,6], the first term of the above condition can be approximated by the sample variance which, at the kth iteration, is defined as $\frac{1}{N_k - 1} \sum_{i \in \mathcal{N}_k} \|\nabla f_i(x^{(k)}) - g_{\mathcal{N}_k}(x^{(k)})\|^2$. Hence, as a practical counterpart of condition (5), at each iteration we force that

$$V^{(k)} \equiv \frac{1}{N_k(N_k - 1)} \sum_{i \in \mathcal{N}_k} \|\nabla f_i(x^{(k)}) - g_{\mathcal{N}_k}(x^{(k)})\|^2 \leq \varepsilon_k, \tag{9}$$

where $\{\varepsilon_k\}_{k \in \mathbb{N}}$ is any nonnegative sequence such that $\sum_{k=0}^{\infty} \varepsilon_k < \infty$. In view of inequality (9), the variance can be monitored by a proper increase of the sample size N_k: whenever condition (9) is not satisfied, the sample size N_k is increased. As for the selection of the learning rate, since the Lipschitz constant of ∇F is often not known, a line search procedure on the sampled objective function is adopted in order to estimate it. Indeed, given $F_{\mathcal{N}_k}(z) = \frac{1}{N_k} \sum_{i \in \mathcal{N}_k} f_i(z)$ and by assuming that all the f_i have Lipschitz-continuous gradients with Lipschitz constant L, the gradient estimate $g_{\mathcal{N}_k}(x)$ is Lipschitz continuous with the same Lipschitz parameter and it holds that

$$F_{\mathcal{N}_k}(y) \leq F_{\mathcal{N}_k}(x) + g_{\mathcal{N}_k}(x)^T(y - x) + \frac{L}{2}\|y - x\|^2, \qquad \forall\, x, y \in \mathbb{R}^d. \tag{10}$$

In view of (10) and denoted by $\bar{x}^{(k)} = x^{(k)} - \alpha_k g_{\mathcal{N}_k}(x^{(k)})$, we require that α_k satisfies the following inequality

$$F_{\mathcal{N}_k}(\bar{x}^{(k)}) \leq F_{\mathcal{N}_k}(x^{(k)}) + g_{\mathcal{N}_k}(x^{(k)})^T(\bar{x}^{(k)} - x^{(k)}) + \frac{1}{2\alpha_k}\|\bar{x}^{(k)} - x^{(k)}\|^2,$$

which can be rewritten as in (8). If the value of α_k does not guarantee the validity of (8), then it is reduced by a factor $\beta < 1$. The line search strategy (8) is well defined: indeed as soon as $\alpha_k \leq \frac{1}{L}$, condition (8) is automatically satisfied. Finally, we point out that the ASM method and the LISA one shares the same line search requirement on the learning rate.

3. Numerical experiments

In this section, we present two different kinds of numerical experiments. In the first one, we consider a binary classification problem, using both convex and non-convex loss functions with several literature datasets. In the second one, we train an artificial neural network tailored for a multiple classification problem.

Table 1. Features of each data set.

Data set	d	#train set (N)	#test set
MNIST	784	60,000	10,000
w8a	300	44,774	4975
CHINA0	132	16,033	1604
IJCNN	22	49,990	91,701

3.1. Binary classification

This section is devoted to a comparison of the previously discussed strategies for solving binary classification problems. In particular, we consider the following algorithms:

- the standard SG method with a fixed size for the mini batch and the learning rate initialized by an optimal hand-tuned value and then properly decreased during the iterations;
- the SG method with a fixed size for the mini batch and the learning rate selected by means of the BB updating rule as defined in [3]; hereafter this approach has been denoted by BB1;
- the ASM method developed in [6];
- the ASM-BB1 method discussed above;
- the LISA method described in Algorithm 1.

To evaluate the effectiveness of the methods under analysis in solving problem (1), we build a binary classifier in the case of four data sets. Table 1 shows the details of these data sets and the cardinality of the train and the test sets. The datasets W8A, IJCNN1 are downloadable from https://www.csie.ntu.edu.tw/cjlin/libsvmtools/, whereas MNIST is available at https://yann.lecun.com/exdb/mnist/ and CHINA0 at https://www.causality.inf.ethz.ch/home.php.

To consider different instances of the problem (1), two convex loss functions and two non-convex loss functions are used as objective function $F(x)$; in particular, by denoting with $a_i \in \mathbb{R}^d$ the feature vector and with $b_i \in \{1, -1\}$ the class label of the ith sample, $F(x)$ assumes one of the following forms:

- logistic regression (LR) loss:

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

- smooth hinge (SH) loss:

$$F(x) = \frac{1}{N}\sum_{i=1}^{N} \begin{cases} \frac{1}{2} - b_i a_i^T x, & \text{if } b_i a_i^T x \leq 0; \\ \frac{1}{2}(1 - b_i a_i^T x)^2, & \text{if } 0 < b_i a_i^T x < 1; \\ 0, & \text{if } b_i a_i^T x \geq 1; \end{cases}$$

Table 2. Best tuned values of α_{opt} for the considered test problems.

	MNIST	*w8a*	*CHINA0*	*IJCNN*
LR	1e-3	1e-1	1e-2	1e-2
SH	1e-3	5e-2	1e-2	1e-2
NN	1e-2	1e-1	1e-1	1e-1
LD	1e-2	1	1e-1	1

Table 3. Values of $\alpha_{\min}$ and $\alpha_{\max}$ for the BB1 method; in the thresholding procedure, these values are divided by the counter of the current epoch.

	MNIST		*w8a*		*CHINA0*		*IJCNN*	
	$\alpha_{\min}$	$\alpha_{\max}$	$\alpha_{\min}$	$\alpha_{\max}$	$\alpha_{\min}$	$\alpha_{\max}$	$\alpha_{\min}$	$\alpha_{\max}$
LR	1e-6	1e-3	1e-6	1e-3	1e-6	1e-3	1e-6	1e-3
SH	1e-6	1e-3	1e-6	1e4	1e-6	1e-3	1e-6	1e4
NN	1e-8	1e-6	1e-8	1e-6	1e-8	1e-6	1e-8	1e-6
LD	1e-8	1e-6	1e-8	1e-6	1e-8	1e-6	1e-6	1e4

Table 4. Values of $\alpha_{\min}$ and $\alpha_{\max}$ for the ASM-BB1 method.

	MNIST		*w8a*		*CHINA0*		*IJCNN*	
	$\alpha_{\min}$	$\alpha_{\max}$	$\alpha_{\min}$	$\alpha_{\max}$	$\alpha_{\min}$	$\alpha_{\max}$	$\alpha_{\min}$	$\alpha_{\max}$
LR	1e-6	1e-3	1e-5	1	1e-5	1	1e-5	1
SH	1e-5	1e-2	1e-5	1	1e-5	1	1e-5	1
NN	1e-5	1e-1	1e-5	1	1e-5	1e-1	1e-5	1
LD	1e-5	1	1e-5	1	1e-5	1e-1	1e-5	1

- nonconvex loss in 2-layer neural networks (NN):

$$F(x) = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{1}{1+\mathrm{e}^{-b_i a_i^T x}}\right)^2;$$

- logistic difference (LD) loss:

$$F(x) = \frac{1}{N}\sum_{i=1}^{N}\left(\ln(1+\mathrm{e}^{-b_i a_i^T x}) - \ln(1+\mathrm{e}^{-b_i a_i^T x - 1})\right).$$

3.1.1. Hyperparameters setting

In the SG method, the mini batch has a fixed size equal to $\overline{N} = 50$; the learning rate is selected with the rule $\alpha_j = \frac{100\overline{\alpha}_0}{100+j}$, where $\overline{\alpha}_0$ is the initial learning rate and j is the counter of the epochs. In particular, the initial learning rate is defined by $\overline{\alpha}_0 = \alpha_{opt} \cdot \overline{N}$, where α_{opt} is the optimal tuned value computed through a trial and error process by using $\overline{N} = 1$. The values of α_{opt} for the considered test problems are shown in Table 2.

In BB1 method, the mini batch size is fixed as $\overline{N} = 50$ and the first learning rate is set as $\alpha_0 = 1$. Furthermore, the values of $\alpha_{\min}$ and $\alpha_{\max}$ are listed in Table 3. Similarly, the initial mini batch size and the initial learning rate of ASM-BB1 method are $N_0 = 50$ and $\alpha_0 = 1$, whereas Table 4 shows the values of $\alpha_{\min}$ and $\alpha_{\max}$ for ASM-BB1.

Table 5. Results for the LR and SH loss functions.

		Logistic regression				*Smooth hinge*			
Method		*MNIST*	*w8a*	*CHINA0*	*IJCNN*	*MNIST*	*w8a*	*CHINA0*	*IJCNN*
SG	$\lvert F(\bar{x}) - F^*\rvert$	0.0067	0.0010	0.0081	0.0002	0.0045	0.0010	0.0030	0.0003
	$\pm STD$	$\pm$0.0007	$\pm$0.0008	$\pm$0.0002	$\pm 5.79e^{-5}$	$\pm$0.0041	$\pm$0.0007	$\pm$0.0018	$\pm$0.0001
	$A(\bar{x})$	0.8985	0.9061	0.9206	0.9197	0.8994	0.9068	0.9213	0.9226
	$\pm STD$	$\pm$0.0009	$\pm$0.0011	$\pm$0.0019	$\pm$0.0004	$\pm$0.0021	$\pm$0.0014	$\pm$0.0017	$\pm$0.0006
ASM-BB1	$\lvert F(\bar{x}) - F^*\rvert$	0.0538	0.0228	0.0277	0.0007	0.0221	0.0057	0.0104	0.0002
	$\pm STD$	$\pm$0.0125	$\pm$0.0006	$\pm$0.0004	$\pm$0.0002	$\pm$0.0140	$\pm$0.0004	$\pm$0.0001	$\pm$0.0002
	$A(\bar{x})$	0.8912	0.9007	0.9164	0.9183	0.8977	0.9056	0.9148	0.9206
	$\pm STD$	$\pm$0.0039	$\pm$0.0005	$\pm$0.0009	$\pm$0.0004	$\pm$0.0017	$\pm$0.0003	$\pm$0.0009	$\pm$0.0007
BB1	$\lvert F(\bar{x}) - F^*\rvert$	0.0025	0.0191	0.0132	0.0025	0.0028	0.0088	0.0033	$7.51e^{-5}$
	$\pm STD$	$\pm$0.0008	$\pm 6.92e^{-5}$	$\pm$0.0002	$\pm 4.32e^{-5}$	$\pm$0.0019	$\pm$0.0176	$\pm$0.0001	$\pm 4.63e^{-5}$
	$A(\bar{x})$	0.8987	0.9103	0.9190	0.9166	0.8998	0.9067	0.9207	0.9213
	$\pm STD$	$\pm$0.0030	$\pm$0.0012	$\pm$0.0012	$\pm$0.0009	$\pm$0.0009	$\pm$0.0002	$\pm$0.0008	$\pm$0.0001
ASM	$\lvert F(\bar{x}) - F^*\rvert$	0.0332	0.0061	0.0262	0.0003	0.0686	0.0133	0.0588	0.0027
	$\pm STD$	$\pm$0.0010	$\pm$0.0007	$\pm$0.0007	$\pm 7.14e^{-5}$	$\pm$0.0041	$\pm$0.0009	$\pm$0.0153	$\pm$0.0009
	$A(\bar{x})$	0.8861	0.9057	0.9163	0.9192	0.8496	0.9012	0.8967	0.9170
	$\pm STD$	$\pm$0.0010	$\pm$0.0010	$\pm$0.0011	$\pm$0.0005	$\pm$0.0026	$\pm$0.0006	$\pm$0.0199	$\pm$0.0006
LISA	$\lvert F(\bar{x}) - F^*\rvert$	0.0055	0.0007	0.0060	0.0004	0.0034	0.0002	0.0026	0.0010
	$\pm STD$	$\pm$0.0018	$\pm$0.0004	$\pm$0.0073	$\pm$0.0001	$\pm$0.0013	$\pm$0.0002	$\pm$0.0025	$\pm$0.0006
	$A(\bar{x})$	0.8978	0.9061	0.9218	0.9204	0.8999	0.9071	0.9221	0.9217
	$\pm STD$	$\pm$0.0014	$\pm$0.0009	$\pm$0.0036	$\pm$0.0006	$\pm$0.0012	$\pm$0.0009	$\pm$0.0015	$\pm$0.0015

In the ASM method, the initial mini batch size is set as $N_0 = 3$ whereas the initial learning rate is $\alpha_0 = 10$; using the same notation of [6], the setting of the other hyperparameters is $\theta = 0.7$, $\nu = 5.84$, $r = 10$, $\gamma = 0.38$, $\eta = 2$ and $\zeta_k = \zeta = 2$. For the loss SH, θ is fixed as 0.9 for the data sets *IJCNN* and *CHINA0*.

In the LISA method, we set $N_0 = 3$, $\alpha_0 = 10$, $\beta = \frac{1}{2}$ and the attempt value of α_k to start the line search procedure (8) has been chosen as $\min(\alpha_0, \alpha_{k-1}\frac{1}{\beta})$, for the following iterations. In addition, the rule $\varepsilon_k = 100 \cdot 0.999^k$ guarantees the consistency with respect to the theoretical formulation.

It is worth to emphasize that both the ASM and the LISA algorithms are free from the expensive tuning of either an optimal value for the learning rate or proper bounds on it, unlike the SG, the BB1 and the ASM-BB1 approaches.

3.1.2. Results

The numerical experiments are carried out in Matlab® on a 1.8 GHz Intel Core i7 processor. All the runs are carried out by varying random number generator and performing 10 trials keeping fixed the aforementioned hyperparameters. In order to estimate the optimality gap, namely $|F(x) - F^*|$, for any test problem an approximate value F^* of the minimum is computed by a huge number of iterations of one of the considered methods.

For any numerical test, the following results are reported:

- average and STandard Deviation (STD) of the optimality gap $|F(\bar{x}) - F^*|$ evaluated on the train set, where $\bar{x}$ is the iterate at the end of the 30th epoch;
- average and STD of the accuracy $A(\bar{x})$ evaluated on the test set, at the end of the 30th epoch.

Table 6. Results for the NN and LD loss functions.

		Nonconvex loss				*Logistic difference*			
Method		*MNIST*	*w8a*	*CHINA0*	*IJCNN*	*MNIST*	*w8a*	*CHINA0*	*IJCNN*
SG	$\lvert F(\bar{x}) - F^* \rvert$	0.0015	0.0015	0.0015	0.0001	0.0022	0.0002	0.0020	0.0265
	$\pm STD$	$\pm$0.0015	$\pm$5.66e^{-5}	$\pm$0.0012	$\pm$4.54e^{-5}	$\pm$0.0003	$\pm$0.0001	$\pm$0.0003	$\pm$0.0075
	$A(\bar{x})$	0.9023	0.9048	0.9220	0.9360	0.9014	0.9067	0.9205	0.9087
	$\pm STD$	$\pm$0.0011	$\pm$0.0005	$\pm$0.0020	$\pm$0.0010	$\pm$0.0007	$\pm$0.0005	$\pm$0.0013	$\pm$0.0118
ASM-BB1	$\lvert F(\bar{x}) - F^* \rvert$	0.0058	0.0097	0.0298	0.0024	0.0060	0.0145	0.0691	0.0306
	$\pm STD$	$\pm$0.0008	$\pm$0.0001	$\pm$0.0005	$\pm$0.0002	$\pm$0.0010	$\pm$0.0002	$\pm$0.0012	$\pm$1.57e^{-5}
	$A(\bar{x})$	0.8994	0.8994	0.9107	0.9221	0.8993	0.8992	0.8913	0.9050
	$\pm STD$	$\pm$0.0007	$\pm$0.0002	$\pm$0.0017	$\pm$0.0004	$\pm$0.0006	$\pm$0.0004	$\pm$0.0006	$\pm$0.0000
BB1	$\lvert F(\bar{x}) - F^* \rvert$	0.0018	0.0097	0.0071	0.0062	0.0067	0.0161	0.0234	0.0230
	$\pm STD$	$\pm$3.80e^{-5}	$\pm$2.92e^{-5}	$\pm$4.69e^{-5}	$\pm$4.31e^{-5}	$\pm$9.36e^{-5}	$\pm$5.35e^{-5}	$\pm$7.89e^{-5}	$\pm$0.0105
	$A(\bar{x})$	0.9027	0.8993	0.9161	0.9166	0.8987	0.8989	0.9167	0.9121
	$\pm STD$	$\pm$0.0007	$\pm$0.0003	$\pm$0.0010	$\pm$7.83e^{-5}	$\pm$0.0006	$\pm$0.0004	$\pm$0.0007	$\pm$0.0148
ASM	$\lvert F(\bar{x}) - F^* \rvert$	0.0136	0.0008	0.0096	0.0006	0.0200	0.0011	0.0164	*
	$\pm STD$	$\pm$0.0004	$\pm$0.0002	$\pm$0.0004	$\pm$8.19e^{-5}	$\pm$0.0004	$\pm$0.0004	$\pm$0.0005	*
	$A(\bar{x})$	0.8406	0.9058	0.9159	0.9303	0.8926	0.9064	0.9151	*
	$\pm STD$	$\pm$0.0011	$\pm$0.0005	$\pm$0.0011	$\pm$0.0011	$\pm$0.0009	$\pm$0.0004	$\pm$0.0015	*
LISA	$\lvert F(\bar{x}) - F^* \rvert$	0.0038	0.0024	0.0016	6.07e^{-5}	0.0006	0.0053	0.0006	0.0302
	$\pm STD$	$\pm$0.0015	$\pm$0.0003	$\pm$0.0005	$\pm$2.39e^{-5}	$\pm$0.0007	$\pm$0.0002	$\pm$0.0005	$\pm$8.84e^{-7}
	$A(\bar{x})$	0.9005	0.9049	0.9246	0.9398	0.9003	0.9027	0.9211	0.9050
	$\pm STD$	$\pm$0.0030	$\pm$0.0008	$\pm$0.0016	$\pm$0.0004	$\pm$0.0014	$\pm$0.0005	$\pm$0.0015	$\pm$0.0000

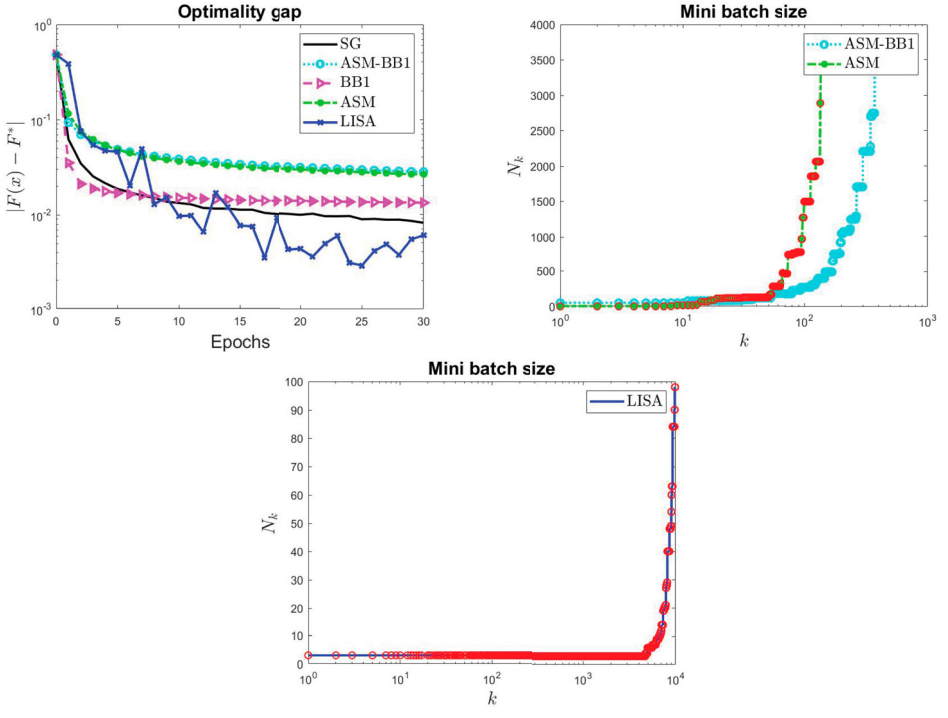

Figure 1. *CHINA0* data set with LR loss: optimality gap (top left panel), increase of mini batch size in ASM and ASM-BB1 (top right panel) and increase of mini batch size in LISA (bottom panel).

In Tables 5 and 6 the results obtained for all the test problems are shown. We can conclude that the LISA method is competitive with respect to the other methods in terms of both optimality gap and accuracy.

For a subset of the considered test problems, in Figures 1–4 we report the behaviour of the optimality gap with respect to the epochs for all the considered methods (top left panels); in the top right and bottom panels, the increase of the mini batch size is shown for ASM, ASM-BB1 methods and for LISA respectively. The mini batch size for ASM, ASM-BB1, and LISA is always significantly lower than the size of the considered data set. The red circles represent the learning rate reductions performed by the line search in ASM and LISA methods.

3.2. A case study: image classification via a convolutional neural network

A meaningful case study is the training of a multiclassifier on a 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 (see Figure 5). In particular, the first convolutional layer is composed by 64 filters (each of them of size 5×5), the second convolutional layer is composed by 32 filters (each of them of size 5×5) and both the max-pooling layers are 2×2. At the end of each inner convolutional layer, there is a sigmoid activation function, whereas the output layer has a softmax function and the loss is given by the cross entropy function. By this CNN, we build a 10-class classifier for the *MNIST* data set. To avoid the

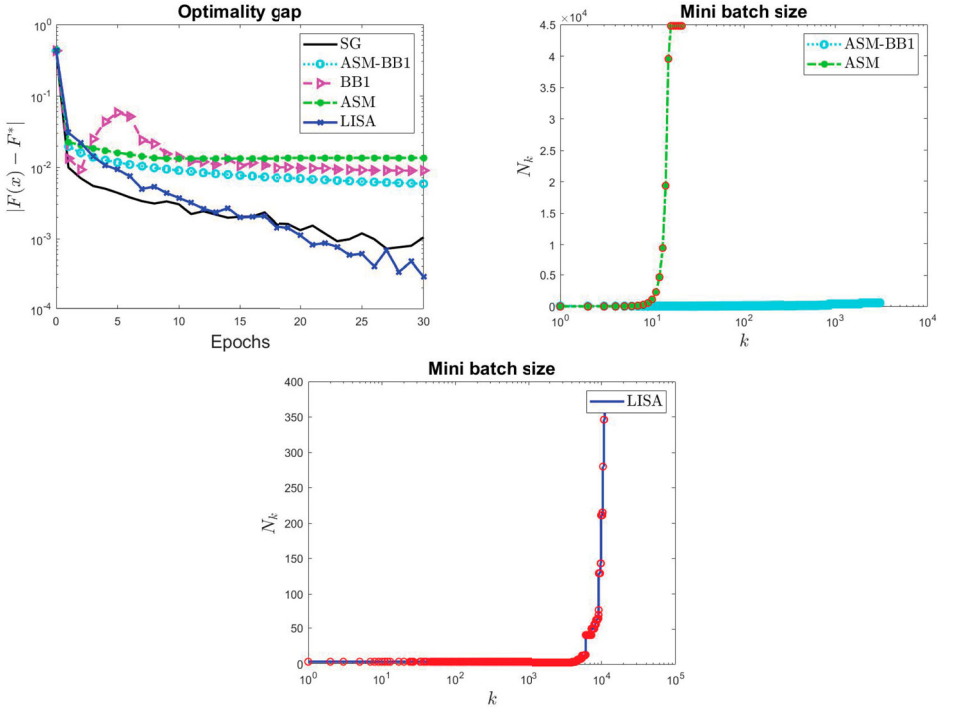

Figure 2. *w8a* data set with SH loss: optimality gap (top left panel), increase of mini batch size in ASM and ASM-BB1 (top right panel) and increase of mini batch size in LISA (bottom panel).

overfitting phenomenon, a ridge regularization is added to the loss, with a regularization parameter equal to $\delta = 10^{-4}$. In addition to the simple SG scheme, we also present a comparison with the ASM-A-R and ASM-AA-R methods recalled in Section 2.2 and developed in [12]. For the SG method, the size of the mini batch, is set as $\overline{N} = 50$; in ASM-A-R and ASM-AA-R methods, the size of the initial mini batch is $N_0 = 3$.

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

Since we are working with a CNN, the choices of both the learning rate and the mini batch size are particularly important and critical. On the one hand, the choice of the mini batch size is subject to memory constraints. Particularly, taking into account the memory resources on the architecture, the maximum possible number of examples for a single sample of the considered dataset is near 8000. On the other hand, to select a good learning rate is crucial to obtain a fast learning phase and to avoid divergence phenomena.

On the left panel of Figure 6, we show the accuracy obtained by the SG method on the test set in the first 5 epochs with different values of the learning rate. It is evident that the choice of the learning rate is critical. In more detail, a too small learning rate (dashed black line) leads to a very slow learning phase, while a too large learning rate (dashed magenta line with solid dot) makes the algorithm divergent; indeed an accuracy of 0.1 in a 10-class case is total randomness. Especially in neural networks context, this initial process of finding a good learning rate involves a high number of attempts, which are computationally expensive. On the other hand, the adaptive methods present a good performances in both

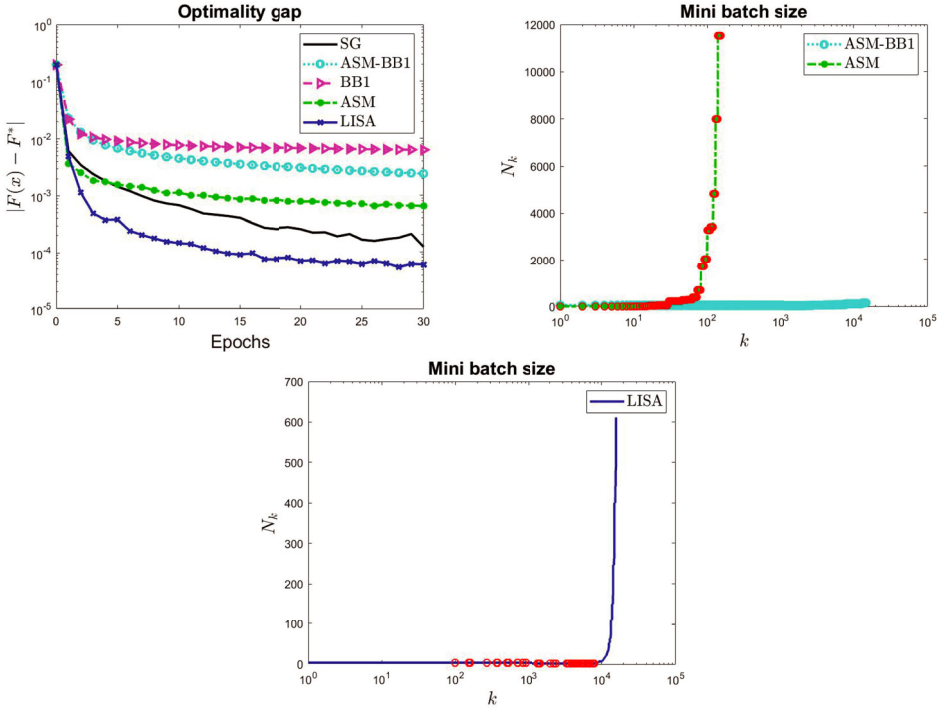

Figure 3. *IJCNN* data set with NN loss: optimality gap (top lef panel), increase of mini batch size in ASM and ASM-BB1 (top right panel) and increase of mini batch size in LISA (bottom panel).

Table 7. Possible different configurations for the LISA method.

Parameter	Conf. 1	Conf. 2	Conf. 3	Conf. 4
τ	1.5	1.5	1.5	1.1
β	1/2	1/2	1/3	1/3
C	10	100	100	100

cases, as highligthed on the right panel of Figure 6. This behaviour also occurs in the case of the LISA method. Indeed, in the following we can show that very stable and accurate results can be obtained even with different settings of the hyperparameters. The initial mini batch size N_0 and the initial learning rate α_0 are always set equal to 10 and 1, respectively. Moreover, we remark that when the condition (8) is not satisfied, the value of α_k is reduced by a factor $\beta < 1$; on the contrary, the starting value of the learning rate for the successive line search is incremented by a factor τ, with $1 < \tau < 1/\beta$. The condition $\tau < 1/\beta$ avoids an unnecessary number of line searches. For the mini batch size increasing, the rule $\varepsilon_k = C \cdot 0.999^k$ guarantees the consistency with respect to the theoretical formulation and, by means of the C value, the increase rate of the sample is driven. Numerical experiments were conducted with different combinations of the above hyperparameters. Table 7 reports their values for the considered configurations.

As shown in Figure 7, the LISA scheme is very robust toward the choice of the hyperparameters. Indeed, for all the configurations explored, it achieves an accuracy comparable to

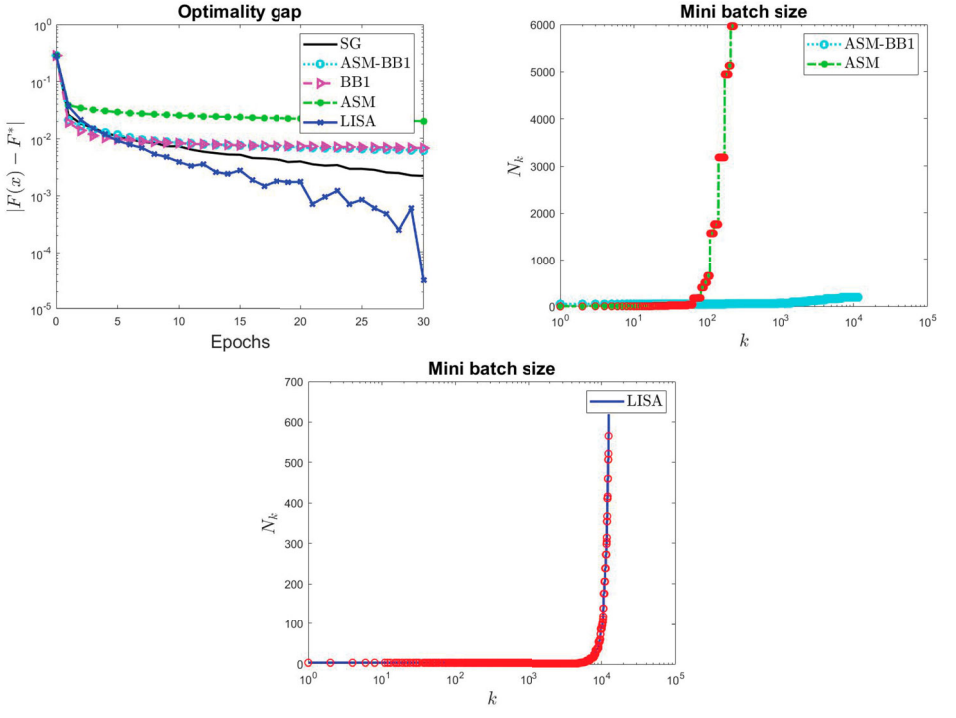

Figure 4. *MNIST* data set with LD loss: optimality gap (top left panel), increase of mini batch size in ASM and ASM-BB1 (top right panel) and increase of mini batch size in LISA (bottom panel).

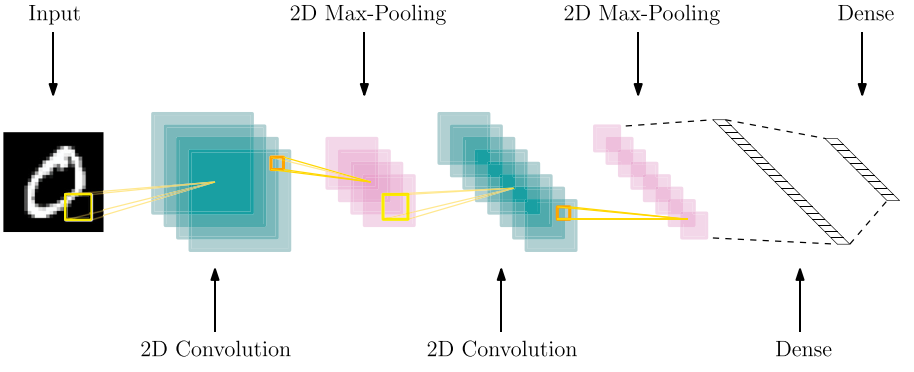

Figure 5. Artificial neural network structure.

that obtained by SG equipped with the optimal learning rate and constant mini batch size. For completeness we report in Table 8 the accuracy achieved for each of the five epochs by the LISA method (measured on the test set) and the corresponding loss values on the train set in Table 9.

A further remark is related to the sample size reached at the end of five epochs. As shown in Table 10, the final sample size is only slightly over a hundred, thus remaining far from

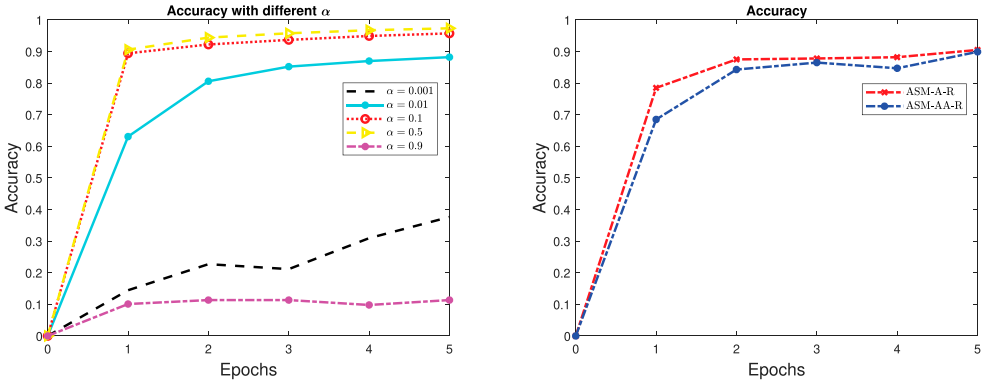

Figure 6. CNN accuracy in the SG case.

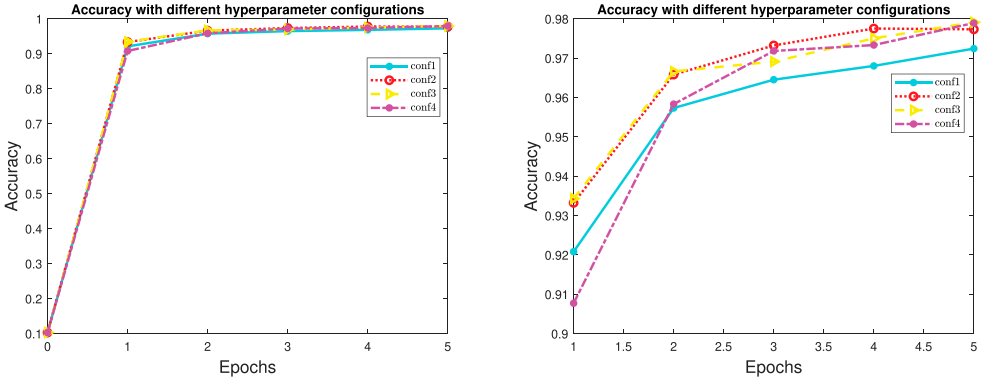

Figure 7. Accuracies for LISA method in the CNN case, with different hyperparameter configuration.

Table 8. Accuracies obtained by LISA method in the CNN case, with different hyperparameter configurations.

Epoch	Conf. 1	Conf. 2	Conf. 3	Conf. 4	Conf. 5
0	0.1028	0.1028	0.1028	0.1028	0.1028
1	0.8907	0.9208	0.9332	0.9343	0.9077
2	0.9457	0.9573	0.9658	0.9666	0.9583
3	0.9512	0.9645	0.9732	0.9691	0.9718
4	0.9682	0.968	0.9775	0.975	0.9733
5	0.973	0.9724	0.9773	0.9791	0.9789

the hardware memory constraint. Furthermore, for comparison with ASM-A-R and ASM-AA-R methods, we notice that the sample size increases up to a maximum of 204 and 182 respectively.

4. Conclusions

In this paper, we discussed several updating rules to choose the learning rate in stochastic gradient methods to face finite-sum problems. We considered standard and line search based learning rate selection techniques and we compared them in combination with

Table 9. Loss values for LISA method in the CNN case, with different hyperparameter configurations.

Epoch	Conf. 1	Conf. 2	Conf. 3	Conf. 4	Conf. 5
0	2.7872	2.7872	2.7872	2.7872	2.7872
1	0.4089	0.3045	0.2687	0.2558	0.3127
2	0.2403	0.1897	0.1697	0.1622	0.1890
3	0.2097	0.1576	0.1431	0.1534	0.1446
4	0.1581	0.1424	0.1285	0.1328	0.1370
5	0.1476	0.1457	0.1273	0.1202	0.1228

Table 10. Final mini batch cardinality for LISA method in the CNN case, with different hyperparameter configurations.

	Conf. 1	Conf. 2	Conf. 3	Conf. 4	Conf. 5
Size of final minibatch	113	121	106	105	106

proper strategies to fix the mini batch size along the iterative process. The stochastic gradient algorithms which exploit a line search procedure to determine the learning rate have performance comparable to that of the standard stochastic gradient methods but they avoid the computational expensive trial and error phase to manually adjust the learning rate (and other hyperparameters) needed instead by the latter ones. Moreover the line search based schemes do not even require the setting of proper bounds on the learning rate. For these reasons, the line search approaches appear preferable and they are worthy of further investigation.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was also supported by the Gruppo Nazionale per il Calcolo Scientifico (GNCS-INdAM) . The publication was created with the co-financing of the European Union-FSE-REACT-EU, PON Research and Innovation 2014–2020 [grant number DM1062/2021] .

References

[1] Robbins H, Monro S. A stochastic approximation method. Ann Math Stat. 1951;22(3):400–407.
[2] Bottou L, Curtis FE, Nocedal J. Optimization methods for large-scale machine learning. SIAM REV. 2018;60(2):223–311.
[3] Liang J, Xu Y, Bao C, et al. Barzilai–Borwein-based adaptive learning rate for deep learning. Pattern Recognit Lett. 2019;128:197–203.
[4] Tan C, Ma S, Dai YH, et al. Barzilai–Borwein step size for stochastic gradient method. In: Lee D, Sugiyama M, Luxburg U, Guyon I, Garnett R, editors. Advances in neural information processing systems (NIPS2016), Barcelona, Spain; 2016. p. 29.
[5] Byrd RH, Chin GM, Nocedal J, et al. Sample size selection in optimization methods for machine learning. Math Program. 2012;134(1):127–155.
[6] Bollapragada R, Byrd R, Nocedal J. Adaptive sampling strategies for stochastic optimization. SIAM J Optim. 2018;28(4):3312–3343.
[7] Franchini G, Porta F, Ruggiero V, et al. A line search based proximal stochastic gradient algorithm with dynamical variance reduction. Optimization online; 2022.

[8] Barzilai J, Borwein JM. Two-point step size gradient methods. IMA J Numer Anal. 1988;8:141–148.
[9] Dai YH, Liao LZ. R-linear convergence of the Barzilai and Borwein gradient method. IMA J Numer Anal. 2002;22(1):1–10.
[10] Dai YH, Fletcher R. On the asymptotic behaviour of some new gradient methods. Math Program. 2005;103:541–559.
[11] di Serafino D, Ruggiero V, Toraldo G, et al. On the asymptotic behaviour of some new gradient methods. Appl Math Comput. 2018;318:176–195.
[12] Franchini G, Ruggiero V, Zanni L. Ritz-like values in steplength selections for stochastic gradient methods. Soft Comput. 2020;24:17573–17588.
[13] Polyak BT. Introduction to optimization. New York: Optimization Software; 1987.
[14] Poon C, Liang J, Schoenlieb C. Local convergence properties of SAGA/Prox-SVRG and acceleration. In: Dy J and Krause A, editors. Proceedings of the 35th International Conference on Machine Learning. PMLR; 2018. Vol. 80. p. 4124–4132.
[15] Freund JE. Mathematical statistics. Englewood Cliffs, NJ, USA: Prentice-Hall; 1962.