Appropriate Learning Rates of Adaptive Learning Rate Optimization Algorithms for Training Deep Neural Networks

Hideaki Iiduka

***Abstract*—This paper deals with nonconvex stochastic optimization problems in deep learning. Appropriate learning rates, based on theory, for adaptive-learning-rate optimization algorithms (e.g., Adam, AMSGrad) to approximate the stationary points of such problems are provided. These rates are shown to allow faster convergence than previously reported for these algorithms. Specifically, the algorithms are examined in numerical experiments on text and image classification, and are shown in experiments to perform better with constant learning rates than algorithms using diminishing learning rates.**

***Index Terms*—Adam, adaptive-learning-rate optimization algorithm, AMSGrad, deep neural network, learning rate, nonconvex stochastic optimization.**

I. Introduction

DEEP learning as a field is mainly concerned with determining appropriate methods for training deep neural networks [1], [2], [3]. One aspect of this is devising useful methods for finding the model parameter values of deep neural networks that reduce certain cost functions called the expected risk and empirical risk (Section 2 in [4]). Accordingly, optimization methods are needed for minimizing the expected (or empirical) risk, i.e., for solving stochastic optimization problems in deep learning.

The classical method for solving a convex stochastic optimization problem is the Stochastic Approximation (SA) method [5], [6], which is a first-order method using the stochastic (sub)gradient of an observed function at each iteration. Modifications of the SA method, such as the mirror descent SA method [6] and the accelerated SA method [7], have been presented.

Within the field of deep learning, practical algorithms based on the SA method and incremental methods [8] for adjusting the *learning rates* of the model parameters have been developed. Such algorithms are referred to as *adaptive-learning-rate optimization algorithms* (Subchapter 8.5 in [9]). Examples include ones that use momentum (Subchapter 8.3.2 in [9]) or Nesterov's accelerated gradients (Subchapter 2.2 in [10] and Subchapter 8.3.3 in [9]). The Adaptive Gradient (AdaGrad) algorithm [11] is a modification of the mirror descent SA method, while the Root Mean Square Propagation (RMSProp) algorithm (Algorithm 8.5 in [9]) is in turn based on AdaGrad, both using element-wise squared values of the stochastic (sub)gradient.

H. Iiduka is with the Department of Computer Science, Meiji University, Kanagawa 214-8571, Japan (e-mail: iiduka@cs.meiji.ac.jp). This work was supported by JSPS KAKENHI Grant Numbers JP18K11184 and 21K11773.

The Adaptive Moment Estimation (Adam) algorithm [12], which is based on momentum and RMSProp, is a powerful algorithm for training deep neural networks. The performance measure of adaptive-learning-rate optimization algorithms is called the regret (see (2) for the definition of regret), and the main objective is to achieve low regret. However, there is an example of a convex optimization problem for which Adam does not minimize the regret (Theorems 1–3 in [13]).

The Adaptive Mean Square Gradient (AMSGrad) algorithm [13] guarantees that the regret is minimized and preserves the practical benefits of Adam. Theorem 4 and Corollary 1 in [13] show that AMSGrad achieves an $\mathcal{O}(\sqrt{(1+\ln n)/n})$ convergence rate for *convex* optimization, where n is the number of iterations. However, since the primary goal of training deep models is to solve nonconvex stochastic optimization problems [14], [15], [16] in deep learning by using optimization algorithms, we need to develop algorithms that can in theory be applied to nonconvex stochastic optimization. The convergence of AMSGrad for *nonconvex* optimization was recently studied in [17] (see [18, Theorem 3], [19, Section 4], [20, Section 3], and [21, Sections 3.5 and 3.6] for convergence analyses of Stochastic Gradient Descent (SGD) methods for nonconvex optimization). The results in [17] show that AMSGrad can be applied to nonconvex optimization in deep learning. In particular, Corollary 3.1 in [17] indicates that AMSGrad with diminishing learning rates for nonconvex optimization achieves an $\mathcal{O}(\ln n/\sqrt{n})$ convergence rate (see [17, Corollary 3.2] for a convergence analysis of the AdaGrad with First Order Momentum (AdaFom) algorithm for nonconvex optimization).

In complicated stochastic optimizations, the learning rates are approximately zero for some iterations of adaptive-learning-rate optimization algorithms if diminishing learning rates are adopted, indicating that diminishing learning rates are not practical. Even if this basic problem were overcome, the problem of empirically selecting appropriate learning rates for obtaining sufficient convergence speed would still remain, and setting these rates in advance in such a way that the convergence speed is guaranteed is prohibitively difficult, due to the rates significantly affecting the model parameters (see Subchapter 8.5 in [9]). We avoid both problems by using constant learning rates [22].

The first motivation behind this work is to identify whether adaptive-learning-rate optimization algorithms (Algorithm 1), including Adam and AMSGrad, with constant learning rates can in theory be applied to nonconvex optimization in deep

learning. This is significant from the viewpoint of practice since using constant learning rates would make adaptive-learning-rate optimization algorithms truly implementable.

The second motivation is to identify whether Algorithm 1 can achieve a better convergence rate than the previous results. We look in particular at the case of AMSGrad—which is included in Algorithm 1 (Section II)—with diminishing learning rates to see whether (for the convex setting) it surpasses the $\mathcal{O}(\sqrt{(1+\ln n)/n})$ convergence rate reported in [13] or (for the nonconvex setting) it does better than the $\mathcal{O}(\ln n/\sqrt{n})$ convergence rate reported in [17].

The results of this study include the sufficient conditions for both constant and diminishing learning rates in order that Algorithm 1 is guaranteed to solve a nonconvex stochastic optimization problem. We consider this to be one of the two contributions of this paper (see Theorems III.1 and III.2). In particular, we show that Algorithm 1 with a constant learning rate has approximately $\mathcal{O}(1/n)$ convergence (Theorem III.1), which is superior to the convergence rates reported in [17] for diminishing learning rates. Because the learning rate never becomes zero, the analysis for constant learning rates (Theorem III.1) should be of theoretical interest as well as being useful for practical applications.

The second contribution is to show that Algorithm 1 with diminishing learning rates achieves an $\mathcal{O}(1/\sqrt{n})$ convergence rate (Theorem III.2), which improves on the results in [13] and [17]. In the special case where cost functions are convex, our analyses guarantee that Algorithm 1 can solve the convex stochastic optimization problem (Propositions III.1 and III.2), in contrast to the previously reported results in [12] and [13] showing Adam and AMSGrad achieving low regret.

To supplement the convergence analysis reports (Theorems III.1 and III.2), Algorithm 1 is applied to the stochastic optimization of tasks in text and image classification. As a result, it was found numerically that the algorithm with constant learning rates is superior to the same algorithm with diminishing learning rates (Section IV).

The remainder of the paper is as follows. First, the mathematical preliminaries and the main problem are laid out in Section II, among with related problems and a more detailed discussion of the motivations behind the present study. These preliminaries include the notation used in this paper, which is summarized in Table I. Then, the adaptive-learning-rate optimization algorithm (Algorithm 1) for solving the main problem is presented in Section III and its convergence is analyzed. These analyses are then compared with previously reported results, summarized in Table II. A numerical comparison of the proposed algorithm with constant versus diminishing learning rates follows in Section IV. Finally, a brief summary is presented in Section V.

II. Stationary Point Problem for Nonconvex Optimization

We consider the following stationary point problem (for a detailed discussion of stationary point problems, see, e.g., Subchapter 1.3.1 in [23]):

Problem II.1 *Let*

(A1) $X \subset \mathbb{R}^d$ *be a closed convex set such that projection onto* X *can be easily computed; and*

(A2) $f\colon \mathbb{R}^d \to \mathbb{R}$ *be defined such that* $f(\boldsymbol{x}) := \mathbb{E}[F(\boldsymbol{x}, \boldsymbol{\xi})]$ *is well defined for all* $\boldsymbol{x} \in \mathbb{R}^d$ *for some* $F(\cdot, \boldsymbol{\xi})$ *that is continuously differentiable for all* $\boldsymbol{\xi} \in \Xi$.

Find any stationary point $\boldsymbol{x}^\star$ *for the problem of minimizing* f *over* X, *i.e.,*

$$\boldsymbol{x}^\star \in X^\star := \{\boldsymbol{x}^\star \in X\colon \langle \boldsymbol{x} - \boldsymbol{x}^\star, \nabla f(\boldsymbol{x}^\star)\rangle \geq 0 \ (\boldsymbol{x} \in X)\}.$$

If $X = \mathbb{R}^d$, then $X^\star = \{\boldsymbol{x}^\star \in \mathbb{R}^d\colon \nabla f(\boldsymbol{x}^\star) = \boldsymbol{0}\}$. A point $\boldsymbol{x}^\star \in \mathbb{R}^d$ satisfying $\nabla f(\boldsymbol{x}^\star) = \boldsymbol{0}$ is a local minimizer of f over $\mathbb{R}^d$. In the case that f is convex, all $\boldsymbol{x}^\star \in X^\star$ are also global minimizers of f over X.

We will examine Problem II.1 under the following conditions [6, Assumptions (A1) and (A2)].

(C1) For the random vector $\boldsymbol{\xi}$, there exists an independent identically distributed (i.i.d.) sample of realizations $\boldsymbol{\xi}_0, \boldsymbol{\xi}_1, \ldots$;

(C2) There exists an oracle that for input $(\boldsymbol{x}, \boldsymbol{\xi}) \in \mathbb{R}^d \times \Xi$ returns stochastic gradient $\mathsf{G}(\boldsymbol{x}, \boldsymbol{\xi})$ such that $\mathbb{E}[\mathsf{G}(\boldsymbol{x}, \boldsymbol{\xi})] = \nabla f(\boldsymbol{x})$;

(C3) A positive scalar M exists such that $\mathbb{E}[\|\mathsf{G}(\boldsymbol{x}, \boldsymbol{\xi})\|^2] \leq M^2$ for all $\boldsymbol{x} \in X$.

A. Related work

The main objective of adaptive-learning-rate optimization algorithms is to solve Problem II.1 with $f(\boldsymbol{x}) = \mathbb{E}[F(\boldsymbol{x}, \xi)] = (1/T)\sum_{t=1}^T f_t(\boldsymbol{x})$ under (A1) and (A2) and (C1)–(C3), i.e.,

$$\text{minimize} \sum_{t\in\mathcal{T}} f_t(\boldsymbol{x}) \text{ subject to } \boldsymbol{x} \in X, \tag{1}$$

where $\mathcal{T} := \{1, 2, \ldots, T\}$ is the index set of training examples and $f_t(\cdot) = F(\cdot, t)\colon \mathbb{R}^d \to \mathbb{R}$ $(t \in \mathcal{T})$ is a differentiable loss function.

1) Convex case: The measure of the performance of adaptive-learning-rate optimization algorithms in solving problem (1) when f_t $(t \in \mathcal{T})$ is convex is called the *regret* and is defined as follows:

$$R(T) := \sum_{t\in\mathcal{T}} f_t(\boldsymbol{x}_t) - f^\star, \tag{2}$$

where $f^\star$ denotes the optimal value for problem (1) and $(\boldsymbol{x}_t)_{t\in\mathbb{N}} \subset X$ is the sequence generated by a learning algorithm. Adam [12] is useful for training deep neural networks. In particular, as indicated in Theorem 4.1 in [12], using Adam guarantees the existence of a positive real number D such that $R(T)/T \leq D/\sqrt{T}$. However, Theorem 1 in [13] shows a counter-example that disproves Theorem 4.1 in [12].

AMSGrad [13] was proposed as a way to guarantee the convergence of Adam. The AMSGrad algorithm is as follows:

$$\begin{aligned}
&\boldsymbol{m}_n := \beta_n \boldsymbol{m}_{n-1} + (1-\beta_n)\mathsf{G}(\boldsymbol{x}_n, \boldsymbol{\xi}_n),\\
&\boldsymbol{v}_n := \delta \boldsymbol{v}_{n-1} + (1-\delta)\mathsf{G}(\boldsymbol{x}_n, \boldsymbol{\xi}_n) \odot \mathsf{G}(\boldsymbol{x}_n, \boldsymbol{\xi}_n),\\
&\hat{\boldsymbol{v}}_n = (\hat{v}_{n,i}) := \left(\max\{\hat{v}_{n-1,i}, v_{n,i}\}\right),\\
&\mathsf{H}_n := \mathsf{diag}\left(\sqrt{\hat{v}_{n,i}}\right),\\
&\boldsymbol{x}_{n+1} := P_{X,\mathsf{H}_n}\left(\boldsymbol{x}_n - \alpha_n \mathsf{H}_n^{-1}\boldsymbol{m}_n\right),
\end{aligned} \tag{3}$$

TABLE I: Notation List

Notation	Description
$\mathbb{N}$	The set of all positive integers and zero
$\mathbb{R}^d$	A d-dimensional Euclidean space with inner product $\langle\cdot,\cdot\rangle$, which induces the norm $\|\cdot\|$
$\mathbb{S}^d$	The set of $d\times d$ symmetric matrices, i.e., $\mathbb{S}^d=\{M\in\mathbb{R}^{d\times d}\colon M=M^\top\}$
$\mathbb{S}^d_{++}$	The set of $d\times d$ symmetric positive-definite matrices, i.e., $\mathbb{S}^d_{++}=\{M\in\mathbb{S}^d\colon M\succ O\}$
$\mathbb{D}^d$	The set of $d\times d$ diagonal matrices, i.e., $\mathbb{D}^d=\{M\in\mathbb{R}^{d\times d}\colon M=\mathsf{diag}(x_i),\ x_i\in\mathbb{R}\ (i=1,2,\ldots,d)\}$
$A\odot B$	The Hadamard product of matrices A and B ($\boldsymbol{x}\odot\boldsymbol{x}:=(x_i^2)\in\mathbb{R}^d$ ($\boldsymbol{x}:=(x_i)\in\mathbb{R}^d$))
$\langle\boldsymbol{x},\boldsymbol{y}\rangle_H$	The H-inner product of $\mathbb{R}^d$, where $H\in\mathbb{S}^d_{++}$, i.e., $\langle\boldsymbol{x},\boldsymbol{y}\rangle_H:=\langle\boldsymbol{x},H\boldsymbol{y}\rangle$
$\|\boldsymbol{x}\|_H^2$	The H-norm, where $H\in\mathbb{S}^d_{++}$, i.e., $\|\boldsymbol{x}\|_H^2:=\langle\boldsymbol{x},H\boldsymbol{x}\rangle$
P_X	The metric projection onto a nonempty, closed convex set X ($\subset\mathbb{R}^d$)
$P_{X,H}$	The metric projection onto X under the H-norm
$\mathbb{E}[Y]$	The expectation of a random variable Y
$\boldsymbol{\xi}$	A random vector whose probability distribution P is supported on a set $\Xi\subset\mathbb{R}^{d_1}$
$F(\cdot,\boldsymbol{\xi})$	A function from $\mathbb{R}^d$ to $\mathbb{R}$ continuously differentiable for all $\boldsymbol{\xi}\in\Xi$
f	The objective function defined by $f(\boldsymbol{x}):=\mathbb{E}[F(\boldsymbol{x},\boldsymbol{\xi})]$ for all $\boldsymbol{x}\in\mathbb{R}^d$
∇f	The gradient of f
$\mathsf{G}(\boldsymbol{x},\boldsymbol{\xi})$	The stochastic gradient for a given $(\boldsymbol{x},\boldsymbol{\xi})\in\mathbb{R}^d\times\Xi$ which satisfies $\mathbb{E}[\mathsf{G}(\boldsymbol{x},\boldsymbol{\xi})]=\nabla f(\boldsymbol{x})$
$X^\star$	The set of stationary points of the problem of minimizing f over X

where $\boldsymbol{x}_0,\boldsymbol{m}_{-1}\in\mathbb{R}^d$, $\boldsymbol{v}_{-1}=\hat{\boldsymbol{v}}_{-1}=\boldsymbol{0}\in\mathbb{R}^d$, and $\delta\in[0,1)$. The AMSGrad algorithm has the following property (see Corollary 4.2 in [12] and Theorem 4 and Corollary 1 in [13]): suppose that $\beta_n:=\nu\lambda^n$ ($\nu,\lambda\in(0,1)$), $\theta:=\nu/\sqrt{\delta}<1$, and $\alpha_n:=\alpha/\sqrt{n}$ ($\alpha>0$). For AMSGrad (3), some positive real number $\hat{D}$ exists such that the following bound holds:

$$\frac{R(T)}{T}=\frac{1}{T}\left(\sum_{t=1}^{T}f_t(\boldsymbol{x}_t)-f^\star\right)\leq\hat{D}\sqrt{\frac{1+\ln T}{T}}. \quad (4)$$

Two algorithms based on Adam and AMSGrad were presented in [24]. One of the two algorithms is called AMSGWDC (named for AMSGrad with weighted gradient and dynamic bound of learning rate), which was obtained by modifying H_n in (3) with a clip function. For AMSGWDC, taking $\alpha_n:=\alpha/\sqrt{n}$ and $\beta_n:=\beta_1e^{-\beta_2 n}$, where $\beta_1,\beta_2>0$, guarantees the following [24, Appendix B]: there exists some positive real number $\tilde{D}$ such that the bound $R(T)/T\leq\tilde{D}/\sqrt{T}$ holds.

AMSGrad where $\beta_n=0$ and H_n is the identity matrix, i.e.,

$$\boldsymbol{x}_{n+1}=\boldsymbol{x}_n-\alpha_n\mathsf{G}(\boldsymbol{x}_n,\boldsymbol{\xi}_n), \quad (5)$$

is the corresponding SGD method. A parallel SGD algorithm [22, Algorithm 3] was presented for solving problem (1) when $X=\mathbb{R}^d$ and $f_t(\boldsymbol{x}):=(\lambda/2)\|\boldsymbol{x}\|^2+L_t(\boldsymbol{x})$ ($\boldsymbol{x}\in\mathbb{R}^d$), where $\lambda>0$ and $L_t\colon\mathbb{R}^d\to\mathbb{R}$ ($t\in\mathcal{T}$) is convex. Theorem 12 in [22] shows that, under certain assumptions [22, (6)], the sequence $(\boldsymbol{w}_n)_{n\in\mathbb{N}}$ generated by the parallel SGD algorithm with k machines and a constant learning rate α satisfies that, for $n:=(\ln k-(\ln\alpha+\ln\lambda))/(2\alpha\lambda)$,

$$\mathbb{E}\left[f(\boldsymbol{w}_n)\right]-f^\star\leq\frac{8\alpha G^2\sqrt{\|\nabla f\|}}{\sqrt{k\lambda}}+\frac{8\alpha G^2\|\nabla f\|}{k\lambda}+2\alpha G^2, \quad (6)$$

where $G>0$ is a constant. See [25] for two natural variants of SGD (5), greedy deploy and lazy deploy, for optimizing jointly smooth, strongly convex loss functions.

2) Nonconvex case: Let us next consider the following stationary point problem [17, (1), (2), Theorem 3.1] associated with a nonconvex optimization problem (1) in the case that f_t ($t\in\mathcal{T}$) is nonconvex and $X=\mathbb{R}^d$: find a point $\boldsymbol{x}^\star\in\mathbb{R}^d$ such that

$$\boldsymbol{x}^\star\in X^\star=\left\{\boldsymbol{x}^\star\in\mathbb{R}^d\colon\nabla f(\boldsymbol{x}^\star)=\boldsymbol{0}\right\}. \quad (7)$$

Under certain assumptions [17, p.4], AMSGrad (3) has the following property (see Theorem 3.1 and Corollary 3.1 in [17]): Let $(\mathsf{g}_{n,i}):=\mathsf{G}(\boldsymbol{x}_n,\boldsymbol{\xi}_n)$. Suppose that there exists $c>0$ such that, for all $i=1,2,\ldots,d$, $|\mathsf{g}_{0,i}|\geq c$, $(\beta_n)\subset[0,1)$ is non-increasing, and $\alpha_n:=1/\sqrt{n}$. Using AMSGrad (3) for problem (7) ensures that there are positive real numbers Q_1 and Q_2 such that, for all $n\in\mathbb{N}$,

$$\min_{k\in[n]}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right]\leq\frac{1}{\sqrt{n}}\left(Q_1+Q_2\ln n\right), \quad (8)$$

where $[n]:=\{1,2,\ldots,n\}$. AdaBelief (named for adapting stepsizes by the belief in observed gradients) was presented in [26], which uses $\boldsymbol{v}_n:=\delta\boldsymbol{v}_{n-1}+(1-\delta)(\mathsf{G}(\boldsymbol{x}_n,\boldsymbol{\xi}_n)-\boldsymbol{m}_n)\odot(\mathsf{G}(\boldsymbol{x}_n,\boldsymbol{\xi}_n)-\boldsymbol{m}_n)$ in place of $\boldsymbol{v}_n$ in (3). If $v_{n+1,i}\geq v_{n,i}$ holds, then AdaBelief with $\alpha_n:=\alpha/\sqrt{n}$ has convergence satisfying (8) [26, Theorem 2.2].

Recently, useful results for SGD (5) were presented for solving problem (7). The convergence analyses of SGD (5) with both constant and diminishing learning rates were presented in [19]. In particular, for a constant learning rate $\alpha_n=1/\beta$, where $\beta>0$ is the Lipschitz constant of ∇f, the following was reported [19, Theorem 12]: there exist positive real numbers M_1 and M_2 such that, for all $n\in\mathbb{N}$, almost surely

$$\frac{1}{n}\sum_{k=1}^{n}\|\nabla f(\boldsymbol{x}_k)\|^2\leq\frac{M_1}{n}+M_2.$$

See [19, Theorem 11] for the convergence rate of SGD being $\mathcal{O}(1/\sqrt{n})$ with a diminishing learning rate $\alpha_n=\min\{\mathcal{O}(1/\sqrt{n}),1/\beta\}$ (this result is also listed in Table II).

Theorem 3.8 in [21] indicates that, under certain assumptions, using SGD (5) with the stochastic Polyak step-size (SPS) defined by $\alpha_n = \min\{(f_t(\boldsymbol{x}_n) - f_t^*)/(c\|\nabla f_t(\boldsymbol{x}_n)\|^2), \gamma_b\}$ guarantees that there exist positive real numbers M_3 and M_4 such that, for all $n \in \mathbb{N}$, $\min_{k\in[n]} \mathbb{E}[\|\nabla f(\boldsymbol{x}_k)\|^2] \leq M_3/n + M_4$, where $c, \gamma_b > 0$ and $f_t^* := \inf_{\boldsymbol{x}\in\mathbb{R}^d} f_t(\boldsymbol{x})$.

A minibatch SGD with a diminishing learning rate $\alpha_n = \mathcal{O}(1/n)$ was presented in [20], and Theorem 3.2 in [20] shows that, under certain assumptions, using minibatch SGD guarantees that there exists $C_1 > 0$ such that, with high probability, $\|\nabla f(\boldsymbol{x}_n)\|^2 \leq C(C_1/n + m^{-1/2+\epsilon})$, where m ($> C$) is the minibatch size and $\epsilon \in (0, C\ln(\ln m)/\ln m)$. See [27] for an appropriate minibatch setting for achieving an ϵ-approximation of Stochastic Path Integrated Differential EstimatoR (SPIDER).

B. Motivation

It was shown in Section II-A that the existing adaptive-learning-rate optimization algorithms are suitable for both convex and nonconvex stochastic optimization problems. Here, we discuss two motivations for the present work that are related to the results in [13], [17], [19], [22]. In the cases of [13], [17], only a convergence analysis of AMSGrad with a diminishing learning rate was reported. Since such diminishing learning rates became approximately zero after a large number of iterations, an algorithm using such rates is not practical. In the case of [22], a convergence analysis of a parallel SGD algorithm with a constant learning rate, which does not have this problem, was presented. However, those results were limited to convex optimization. In the case of [19], although a useful convergence analysis of SGD with a constant learning rate was presented for nonconvex optimization, it remains necessary to develop adaptive-learning-rate optimization algorithms with constant learning rates for nonconvex optimization. Accordingly, the first motivation of the present study was to identify whether AMSGrad with constant learning rates would be such an algorithm in the context of deep learning.

The second motivation was to identify whether AMSGrad can achieve a better convergence rate than previous results. In particular, we would like to show that, for the convex setting, AMSGrad with a diminishing learning rate (e.g., $\alpha_n = 1/\sqrt{n}$) achieves a better convergence rate than (4) and, for the nonconvex setting, AMSGrad with a diminishing learning rate (e.g., $\alpha_n = 1/\sqrt{n}$) achieves a better convergence rate than (8).

We also analyze the following modified Adam algorithm based on the definition of $\hat{\boldsymbol{v}}_n$ in AMSGrad (3):

$$\begin{aligned}
&\boldsymbol{m}_n := \beta_n \boldsymbol{m}_{n-1} + (1-\beta_n)\mathsf{G}(\boldsymbol{x}_n, \boldsymbol{\xi}_n),\\
&\boldsymbol{v}_n := \delta \boldsymbol{v}_{n-1} + (1-\delta)\mathsf{G}(\boldsymbol{x}_n, \boldsymbol{\xi}_n) \odot \mathsf{G}(\boldsymbol{x}_n, \boldsymbol{\xi}_n),\\
&\bar{\boldsymbol{v}}_n := \frac{\boldsymbol{v}_n}{1-\delta^{n+1}},\\
&\hat{\boldsymbol{v}}_n = (\hat{v}_{n,i}) := (\max\{\hat{v}_{n-1,i}, \bar{v}_{n,i}\}),\\
&\mathsf{H}_n := \mathsf{diag}\left(\sqrt{\hat{v}_{n,i}}\right),\\
&\boldsymbol{x}_{n+1} := P_{X,\mathsf{H}_n}\left(\boldsymbol{x}_n - \alpha_n \mathsf{H}_n^{-1} \boldsymbol{m}_n\right),
\end{aligned} \tag{9}$$

where $\boldsymbol{x}_0, \boldsymbol{m}_{-1} \in \mathbb{R}^d$, $\boldsymbol{v}_{-1} = \hat{\boldsymbol{v}}_{-1} = \boldsymbol{0} \in \mathbb{R}^d$, and $\delta \in [0, 1)$. We can see that Adam [12] uses $\mathsf{H}_n = \mathsf{diag}(\bar{v}_{n,i}^{1/2})$. We use $\hat{\boldsymbol{v}}_n = (\hat{v}_{n,i}) := (\max\{\hat{v}_{n-1,i}, \bar{v}_{n,i}\})$ in (9) so as to guarantee the convergence of algorithm (9).

To analyze both AMSGrad (3) and the modified Adam (9), we study the following algorithm which includes both. Throughout this paper, we refer to the parameters α_n and β_n as *sub-learning rates*.

Algorithm 1 Adaptive learning rate optimization algorithm for solving Problem II.1

Require: $(\alpha_n)_{n\in\mathbb{N}} \subset (0,1)$, $(\beta_n)_{n\in\mathbb{N}} \subset [0,1)$, $\gamma \in [0,1)$

1: $n \leftarrow 0$, $\boldsymbol{x}_0, \boldsymbol{m}_{-1} \in \mathbb{R}^d$, $\mathsf{H}_0 \in \mathbb{S}_{++}^d \cap \mathbb{D}^d$
2: **loop**
3: $\quad \boldsymbol{m}_n := \beta_n \boldsymbol{m}_{n-1} + (1-\beta_n)\mathsf{G}(\boldsymbol{x}_n, \boldsymbol{\xi}_n)$
4: $\quad \hat{\boldsymbol{m}}_n := \dfrac{\boldsymbol{m}_n}{1-\gamma^{n+1}}$
5: $\quad \mathsf{H}_n \in \mathbb{S}_{++}^d \cap \mathbb{D}^d$ (see (3) and (9) for examples of H_n)
6: $\quad$ Find $\mathbf{d}_n \in \mathbb{R}^d$ that solves $\mathsf{H}_n \mathbf{d} = -\hat{\boldsymbol{m}}_n$
7: $\quad \boldsymbol{x}_{n+1} := P_{X,\mathsf{H}_n}(\boldsymbol{x}_n + \alpha_n \mathbf{d}_n)$
8: $\quad n \leftarrow n+1$
9: **end loop**

III. Convergence Analyses of Algorithm 1

In the convergence analyses of Algorithm 1 presented here, we adopt the following set of conditions as assumptions.

Assumption III.1 *For Algorithm 1, first, with the decomposition* $\mathsf{H}_n := \mathsf{diag}(h_{n,i})$, *sequence* $(\mathsf{H}_n)_{n\in\mathbb{N}} \subset \mathbb{S}_{++}^d \cap \mathbb{D}^d$ *satisfies the following two conditions:*

(A3) $h_{n+1,i} \geq h_{n,i}$ *almost surely for all* $n \in \mathbb{N}$ *and all* $i = 1, 2, \ldots, d$;

(A4) *For all* $i = 1, 2, \ldots, d$, *a positive number* B_i *exists such that* $\sup\{\mathbb{E}[h_{n,i}] \colon n \in \mathbb{N}\} \leq B_i$.

Second, with the decomposition $\boldsymbol{x}_n = (x_{n,i})$, *the generated sequence* $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ *satisfies the following condition: for* $\boldsymbol{x} = (x_i) \in X$,

(A5) $D := \max_{i=1,2,\ldots,d} \sup\{(x_{n+1,i} - x_i)^2 \colon n \in \mathbb{N}\} < +\infty$.

Assumption (A5) holds if X is bounded, which was assumed in [6, p.1574], [12, Theorem 4.1], and [13, p.2]. Here, we show that $(\mathsf{H}_n)_{n\in\mathbb{N}}$ defined for either AMSGrad (3) or Adam (9) satisfies (A3) and (A4) given that X is bounded (i.e., (A5) holds).

To show this, we first consider H_n and $\boldsymbol{v}_n$ ($n \in \mathbb{N}$) defined for Adam (9). The definitions of $\hat{\boldsymbol{v}}_n$ and $\mathsf{H}_n = \mathsf{diag}(h_{n,i}) = \mathsf{diag}(\hat{v}_{n,i}^{1/2}) \in \mathbb{S}_{++}^d \cap \mathbb{D}^d$ in (9) obviously satisfy (A3). Step 7 in Algorithm 1 implies that $(\boldsymbol{x}_n)_{n\in\mathbb{N}} \subset X$. Accordingly, the boundedness of X and (A2) ensure that $(\mathsf{G}(\boldsymbol{x}_n, \boldsymbol{\xi}_n))_{n\in\mathbb{N}}$ is almost surely bounded, i.e., $M_1 := \sup\{\|\mathsf{G}(\boldsymbol{x}_n, \boldsymbol{\xi}_n) \odot \mathsf{G}(\boldsymbol{x}_n, \boldsymbol{\xi}_n)\| : n \in \mathbb{N}\} < +\infty$. Moreover, from the definition of $\boldsymbol{v}_n$ and the triangle inequality, we have, for all $n \in \mathbb{N}$, $\|\boldsymbol{v}_n\| \leq \delta\|\boldsymbol{v}_{n-1}\| + (1-\delta)M_1$. Induction thus shows that, for all $n \in \mathbb{N}$, $\|\boldsymbol{v}_n\| = (\sum_{i=1}^d |v_{n,i}|^2)^{1/2} \leq M_1$ almost surely, which, together with the definition of $\bar{\boldsymbol{v}}_n$, implies that $\|\bar{\boldsymbol{v}}_n\| = (\sum_{i=1}^d |\bar{v}_{n,i}|^2)^{1/2} \leq M_1/(1-\delta)$. Accordingly, we have, for all $n \in \mathbb{N}$ and all $i = 1, 2, \ldots, d$,

$|v_{n,i}|^2, |\bar{v}_{n,i}|^2 \leq M_1^2/(1-\delta)^2$. The definition of $\hat{\boldsymbol{v}}_n$ and $\hat{\boldsymbol{v}}_{-1} = \boldsymbol{0}$ ensure that, for all $n \in \mathbb{N}$ and all $i = 1, 2, \ldots, d$,

$$\mathbb{E}[h_{n,i}] := \mathbb{E}\left[\sqrt{\hat{v}_{n,i}}\right] \leq \frac{M_1}{1-\delta},$$

which implies that (A4) holds.

Next, we consider H_n and $\boldsymbol{v}_n$ ($n \in \mathbb{N}$) defined for AMSGrad (3). A discussion similar to the one showing that H_n and $\boldsymbol{v}_n$ defined by (9) satisfy (A3) and (A4) ensures that H_n and $\boldsymbol{v}_n$ defined by (3) also satisfy (A3) and (A4); i.e., for all $n \in \mathbb{N}$ and all $i = 1, 2, \ldots, d$,

$$\mathbb{E}[h_{n,i}] := \mathbb{E}\left[\sqrt{\hat{v}_{n,i}}\right] \leq M_1.$$

A. Constant sub-learning rate case

Here we present a convergence analysis of Algorithm 1 for constant sub-learning rates. The proof of the following theorem is given in Appendix A.

Theorem III.1 *Under assumptions (A1)–(A5) and (C1)–(C3) and supposing that sequence $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ was generated by Algorithm 1 using $\alpha_n := \alpha$ and $\beta_n := \beta$ ($n \in \mathbb{N}$), the following relation holds for all $\boldsymbol{x} \in X$:*

$$\limsup_{n\to+\infty} \mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_n, \nabla f(\boldsymbol{x}_n)\rangle\right] \geq -\frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2}\alpha - \frac{\tilde{M}\sqrt{Dd}}{\tilde{b}\tilde{\gamma}}\beta,$$

where $\tilde{B} := \sup\{\max_{i=1,2,\ldots,d} h_{n,i}^{-1/2} \colon n \in \mathbb{N}\} < +\infty$, $\tilde{M}^2 := \max\{\|\boldsymbol{m}_{-1}\|^2, M^2\}$ and $\tilde{\gamma} := 1-\gamma$, $\tilde{b} := 1-\beta$ (Note that D was defined in (A5)). Furthermore, the following two relations hold for all $\boldsymbol{x} \in X$ and all $n \in \mathbb{N}$:

$$\begin{aligned}
&\frac{1}{n}\sum_{k=1}^{n} \mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_k, \nabla f(\boldsymbol{x}_k)\rangle\right] \\
&\quad \geq -\frac{D\sum_{i=1}^{d} B_i}{2\tilde{b}\alpha n} - \frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2}\alpha - \frac{\tilde{M}\sqrt{Dd}}{\tilde{b}}\beta, \\
&\max_{k\in[n]} \mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_k, \nabla f(\boldsymbol{x}_k)\rangle\right] \\
&\quad \geq -\frac{D\sum_{i=1}^{d} B_i}{2\tilde{b}\alpha n} - \frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2}\alpha - \frac{\tilde{M}\sqrt{Dd}}{\tilde{b}}\beta.
\end{aligned}$$

Theorem III.1 leads to the following proposition.

Proposition III.1 *Under assumptions (A1)–(A5) and (C1)–(C3) and supposing that $F(\cdot, \boldsymbol{\xi})$ is convex for any fixed $\boldsymbol{\xi} \in \Xi$ and that $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ is the sequence generated by Algorithm 1 using $\alpha_n := \alpha$ and $\beta_n := \beta$ ($n \in \mathbb{N}$), the following relation holds:*

$$\liminf_{n\to+\infty} \mathbb{E}\left[f(\boldsymbol{x}_n) - f^\star\right] \leq \frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2}\alpha + \frac{\tilde{M}\sqrt{Dd}}{\tilde{b}\tilde{\gamma}}\beta,$$

where $f^\star$ denotes the optimal objective function value for the problem of minimizing f over X, and $\tilde{\gamma}$, $\tilde{b}$, $\tilde{M}$, D, and $\tilde{B}$ are as defined in Theorem III.1. Furthermore, for all $n \in \mathbb{N}$, the following relation holds:

$$\min_{k\in[n]} \mathbb{E}\left[f(\boldsymbol{x}_k) - f^\star\right] \leq \frac{D\sum_{i=1}^{d} B_i}{2\tilde{b}\alpha n} + \frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2}\alpha + \frac{\tilde{M}\sqrt{Dd}}{\tilde{b}}\beta.$$

If we additionally define the elements of the sequence $(\tilde{\boldsymbol{x}}_n)_{n\in\mathbb{N}}$ by $\tilde{\boldsymbol{x}}_n := (1/n)\sum_{k=1}^{n} \boldsymbol{x}_k$, then the following relation holds:

$$\mathbb{E}\left[f(\tilde{\boldsymbol{x}}_n) - f^\star\right] \leq \frac{D\sum_{i=1}^{d} B_i}{2\tilde{b}\alpha n} + \frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2}\alpha + \frac{\tilde{M}\sqrt{Dd}}{\tilde{b}}\beta.$$

Finally, for problem (1)*, the regret for Algorithm 1 satisfies the following relation:*

$$\frac{R(T)}{T} \leq \frac{D\sum_{i=1}^{d} B_i}{2\tilde{b}\alpha T} + \frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2}\alpha + \frac{\tilde{M}\sqrt{Dd}}{\tilde{b}}\beta.$$

B. Diminishing sub-learning rate case

Here we give a convergence analysis of Algorithm 1 for diminishing sub-learning rates. The proof of the following theorem is also presented in Appendix A.

Theorem III.2 *Under assumptions (A1)–(A5) and (C1)–(C3) and supposing that sequence $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ was generated by Algorithm 1 using α_n and β_n ($n \in \mathbb{N}$)[1] such that $\sum_{n=0}^{+\infty} \alpha_n = +\infty$, $\sum_{n=0}^{+\infty} \alpha_n^2 < +\infty$, and $\sum_{n=0}^{+\infty} \alpha_n\beta_n < +\infty$, the following relation holds for all $\boldsymbol{x} \in X$:*

$$\limsup_{n\to+\infty} \mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_n, \nabla f(\boldsymbol{x}_n)\rangle\right] \geq 0. \tag{10}$$

For the case of $\alpha_n := 1/n^\eta$ ($\eta \in [1/2, 1)$)[2] and $\beta_n := \lambda^n$ ($\lambda \in (0,1)$), Algorithm 1 has convergences such that, for all $\boldsymbol{x} \in X$ and all $n \in \mathbb{N}$,

$$\begin{aligned}
&\frac{1}{n}\sum_{k=1}^{n} \mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_k, \nabla f(\boldsymbol{x}_k)\rangle\right] \\
&\quad \geq -\frac{D\sum_{i=1}^{d} B_i}{2\tilde{b}n^{1-\eta}} - \frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2(1-\eta)n^{1-\eta}} - \frac{\tilde{M}\lambda\sqrt{Dd}}{\tilde{b}(1-\lambda)n}, \\
&\max_{k\in[n]} \mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_k, \nabla f(\boldsymbol{x}_k)\rangle\right] \\
&\quad \geq -\frac{D\sum_{i=1}^{d} B_i}{2\tilde{b}n^{1-\eta}} - \frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2(1-\eta)n^{1-\eta}} - \frac{\tilde{M}\lambda\sqrt{Dd}}{\tilde{b}(1-\lambda)n},
\end{aligned}$$

where D, $\tilde{b}$, $\tilde{M}$, $\tilde{B}$, and $\tilde{\gamma}$ are the same as in Theorem III.1.

Theorem III.2 leads to the following.

Proposition III.2 *Under assumptions (A1)–(A5) and (C1)–(C3), suppose that $F(\cdot, \boldsymbol{\xi})$ is convex for any fixed $\boldsymbol{\xi} \in \Xi$ and that $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ is the sequence generated by Algorithm 1 using $\alpha_n := 1/n^\eta$ ($\eta \in [1/2, 1]$) and $\beta_n := \lambda^n$ ($n \in \mathbb{N}$; $\lambda \in (0,1)$). If $\eta \in (1/2, 1]$, then the following limit holds:*

$$\liminf_{n\to+\infty} \mathbb{E}\left[f(\boldsymbol{x}_n) - f^\star\right] = 0,$$

where $f^\star$ denotes the optimal objective function value for the problem of minimizing f over X. If $\eta \in [1/2, 1)$, any accumulation point of $(\tilde{\boldsymbol{x}}_n)_{n\in\mathbb{N}}$, defined by $\tilde{\boldsymbol{x}}_n := (1/n)\sum_{k=1}^{n} \boldsymbol{x}_k$, al-

[1]The sub-learning rates $\alpha_n := 1/n^\eta$ ($\eta \in (1/2, 1]$) and $\beta_n := \lambda^n$ ($\lambda \in (0,1)$) satisfy $\sum_{n=0}^{+\infty} \alpha_n = +\infty$, $\sum_{n=0}^{+\infty} \alpha_n^2 < +\infty$, and $\sum_{n=0}^{+\infty} \alpha_n\beta_n < +\infty$ (by $\lim_{n\to+\infty}(\alpha_{n+1}\beta_{n+1})/(\alpha_n\beta_n) = \lambda \in (0,1)$).

[2]Algorithm 1 with $\alpha_n := 1/\sqrt{n}$ and $\beta_n := \lambda^n$ does not satisfy (10). However, Algorithm 1 with $\alpha_n := 1/\sqrt{n}$ and $\beta_n := \lambda^n$ achieves a convergence rate of $\mathcal{O}(1/\sqrt{n})$.

most surely belongs to $X^\star$, and Algorithm 1 has convergences such that, for all $n \in \mathbb{N}$,

$$\mathbb{E}\left[f(\tilde{\boldsymbol{x}}_n) - f^\star\right] \leq \frac{D\sum_{i=1}^d B_i}{2\tilde{b}n^\theta} + \frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2\theta n^\theta} + \frac{\tilde{M}\tilde{\lambda}\sqrt{Dd}}{\tilde{b}n},$$

$$\min_{k\in[n]} \mathbb{E}\left[f(\boldsymbol{x}_k) - f^\star\right] \leq \frac{D\sum_{i=1}^d B_i}{2\tilde{b}n^\theta} + \frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2\theta n^\theta} + \frac{\tilde{M}\tilde{\lambda}\sqrt{Dd}}{\tilde{b}n},$$

where D, $\tilde{b}$, $\tilde{M}$, $\tilde{B}$, and $\tilde{\gamma}$ are the same as in Theorem III.1, $\tilde{\lambda} := \lambda/(1-\lambda)$, and $\theta := 1-\eta$. Finally, for problem (1)*, the regret for Algorithm 1 satisfies the following relation:*

$$\frac{R(T)}{T} \leq \frac{D\sum_{i=1}^d B_i}{2\tilde{b}T^\theta} + \frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2\theta T^\theta} + \frac{\tilde{M}\tilde{\lambda}\sqrt{Dd}}{\tilde{b}T}.$$

C. Comparison between Algorithm 1 and existing algorithms

Table II summarizes the results of the existing algorithms discussed in Section II-A and our results presented in Sections III-A and III-B for convex and nonconvex optimization. The following are detailed comparisons for adaptive-learning-rate optimization algorithms, such as AMSGrad and Algorithm 1.

1) Convex case: AMSGrad [13] with $\alpha_n := 1/\sqrt{n}$ has convergence satisfying (4). In comparison, Propositions III.1 and III.2 indicate that AMSGrad (i.e., Algorithm 1 with H_n defined by (3)) has convergence satisfying

$$\frac{R(T)}{T} \leq \begin{cases} \mathcal{O}\left(\frac{1}{T}\right) + C_1\alpha + C_2\beta & \text{(Proposition III.1)}, \\ \mathcal{O}\left(\frac{1}{T^{1-\eta}}\right) & \text{(Proposition III.2)}, \end{cases}$$

where C_1 and C_2 are constants independent of T. In particular, AMSGrad with $\alpha_n := 1/\sqrt{n}$ (i.e., $\eta = 1/2$) has convergence satisfying

$$\frac{R(T)}{T} \leq \mathcal{O}\left(\frac{1}{\sqrt{T}}\right),$$

which is better than (4). Proposition III.1 shows that Algorithm 1 with constant sub-learning rates α and β satisfies

$$\liminf_{n\to+\infty} \mathbb{E}\left[f(\boldsymbol{x}_n) - f^\star\right] \leq C_1\alpha + C_2\beta.$$

This result resembles (6) for the parallel SGD, indicating that there exists $C > 0$ such that, for some n, $\mathbb{E}[f(\boldsymbol{w}_n) - f^\star] \leq C\alpha$.

2) Nonconvex case: AMSGrad [17] with $\alpha_n := 1/\sqrt{n}$ satisfies that, for all $n \in \mathbb{N}$,

$$\min_{k\in[n]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] = \mathcal{O}\left(\frac{\ln n}{\sqrt{n}}\right) \tag{11}$$

(see also (8)). Meanwhile, Theorem III.2 indicates AMSGrad (i.e., Algorithm 1 with H_n defined by (3)) with $\alpha_n := 1/n^\eta$, where $\eta \in [1/2, 1]$, satisfies that, if $\eta \in (1/2, 1]$, then, for all $\boldsymbol{x} \in X$,

$$\limsup_{n\to+\infty} \mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_n, \nabla f(\boldsymbol{x}_n)\rangle\right] \geq 0, \tag{12}$$

and if $\eta \in [1/2, 1)$, then, for all $\boldsymbol{x} \in X$ and all $n \in \mathbb{N}$,

$$\frac{1}{n}\sum_{k=1}^n \mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_k, \nabla f(\boldsymbol{x}_k)\rangle\right] \geq -\mathcal{O}\left(\frac{1}{n^{1-\eta}}\right),$$

$$\max_{k\in[n]} \mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_k, \nabla f(\boldsymbol{x}_k)\rangle\right] \geq -\mathcal{O}\left(\frac{1}{n^{1-\eta}}\right). \tag{13}$$

Unlike (11), $\eta = 1/2$ is not allowed for (12) to hold. However, (12) guarantees that Algorithm 1 with diminishing sub-learning rates converges to a point in $X^\star$ in the sense that an accumulation point of $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ belonging to $X^\star$ exists. If $X = \mathbb{R}^d$, then for all $\eta \in [1/2, 1)$ and all $n \in \mathbb{N}$, (13) implies that AMSGrad (3) satisfies

$$\min_{k\in[n]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] = \mathcal{O}\left(\frac{1}{n^{1-\eta}}\right), \tag{14}$$

which, for the case of $\eta = 1/2$, is better than (11).

IV. Numerical Experiments

We examined the behavior of Algorithm 1 for different sub-learning rates. The adaptive-learning-rate optimization algorithms with $\delta = 0.999$ [12], [13] and the default values in `torch.optim`[3] were as follows, where the initial points initialized automatically by PyTorch were used.

Algorithm 1 with constant sub-learning rates:

- ADAM-C1: Algorithm 1 with (9), $\gamma = 0.9$, $\alpha_n = 10^{-3}$, and $\beta_n = 0.9$
- ADAM-C2: Algorithm 1 with (9), $\gamma = 0.9$, $\alpha_n = 10^{-3}$, and $\beta_n = 10^{-3}$
- ADAM-C3: Algorithm 1 with (9), $\gamma = 0.9$, $\alpha_n = 10^{-2}$, and $\beta_n = 10^{-2}$
- AMSG-C1: Algorithm 1 with (3), $\gamma = 0$, $\alpha_n = 10^{-3}$, and $\beta_n = 0.9$
- AMSG-C2: Algorithm 1 with (3), $\gamma = 0$, $\alpha_n = 10^{-3}$, and $\beta_n = 10^{-3}$
- AMSG-C3: Algorithm 1 with (3), $\gamma = 0$, $\alpha_n = 10^{-2}$, and $\beta_n = 10^{-2}$
- MAMSG-C1: Algorithm 1 with (3), $\gamma = 0.1$, $\alpha_n = 10^{-3}$, and $\beta_n = 0.9$
- MAMSG-C2: Algorithm 1 with (3), $\gamma = 0.1$, $\alpha_n = 10^{-3}$, and $\beta_n = 10^{-3}$
- MAMSG-C3: Algorithm 1 with (3), $\gamma = 0.1$, $\alpha_n = 10^{-2}$, and $\beta_n = 10^{-2}$

Algorithm 1 with diminishing sub-learning rates:

- ADAM-D1: Algorithm 1 with (9), $\gamma = 0.9$, $\alpha_n = 1/\sqrt{n}$, and $\beta_n = 1/2^n$
- ADAM-D2: Algorithm 1 with (9), $\gamma = 0.9$, $\alpha_n = 1/n^{3/4}$, and $\beta_n = 1/2^n$
- ADAM-D3: Algorithm 1 with (9), $\gamma = 0.9$, $\alpha_n = 1/n$, and $\beta_n = 1/2^n$
- AMSG-D1: Algorithm 1 with (3), $\gamma = 0$, $\alpha_n = 1/\sqrt{n}$, and $\beta_n = 1/2^n$
- AMSG-D2: Algorithm 1 with (3), $\gamma = 0$, $\alpha_n = 1/n^{3/4}$, and $\beta_n = 1/2^n$
- AMSG-D3: Algorithm 1 with (3), $\gamma = 0$, $\alpha_n = 1/n$, and $\beta_n = 1/2^n$
- MAMSG-D1: Algorithm 1 with (3), $\gamma = 0.1$, $\alpha_n = 1/\sqrt{n}$, and $\beta_n = 1/2^n$
- MAMSG-D2: Algorithm 1 with (3), $\gamma = 0.1$, $\alpha_n = 1/n^{3/4}$, and $\beta_n = 1/2^n$
- MAMSG-D3: Algorithm 1 with (3), $\gamma = 0.1$, $\alpha_n = 1/n$, and $\beta_n = 1/2^n$

[3]https://pytorch.org/docs/stable/optim.html

TABLE II: Convergence rates of stochastic optimization algorithms for convex and nonconvex optimization

	Convex optimization		Nonconvex optimization	
	Constant learning rate	Diminishing learning rate	Constant learning rate	Diminishing learning rate
SGD [19]	$\mathcal{O}\left(\frac{1}{T}\right)+C$	$\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$	$\mathcal{O}\left(\frac{1}{n}\right)+C$	$\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$
SGD with SPS [21]	———	$\mathcal{O}\left(\frac{1}{T}\right)+C$	———	$\mathcal{O}\left(\frac{1}{n}\right)+C$
Minibatch SGD [20]	———	$\mathcal{O}\left(\frac{1}{T}\right)+C$	———	$\mathcal{O}\left(\frac{1}{n}\right)+C$
Adam [12]	———	$\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)^{(*)}$	———	———
AMSGrad [13]	———	$\mathcal{O}\left(\sqrt{\frac{1+\ln T}{T}}\right)$	———	———
GWDC [24]	———	$\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$	———	———
AMSGWDC [24]	———	$\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$	———	———
AMSGrad [17]	———	$\mathcal{O}\left(\frac{\ln T}{\sqrt{T}}\right)$	———	$\mathcal{O}\left(\frac{\ln n}{\sqrt{n}}\right)$
AdaBelief [26]	———	$\mathcal{O}\left(\frac{\ln T}{\sqrt{T}}\right)$	———	$\mathcal{O}\left(\frac{\ln n}{\sqrt{n}}\right)$
Algorithm 1 (presented herein)	$\mathcal{O}\left(\frac{1}{T}\right)+C_1\alpha+C_2\beta$	$\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$	$\mathcal{O}\left(\frac{1}{n}\right)+C_1\alpha+C_2\beta$	$\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$

Note: C, C_1, and C_2 are constants independent of learning rates α, β, number of training examples T, and number of iterations n. The convergence rate for convex optimization is measured in terms of regret as $R(T)/T$, and the convergence rate for nonconvex optimization is measured as the expectation of the squared gradient norm $\min_{k\in[n]} \mathbb{E}[\|\nabla f(\boldsymbol{x})\|^2]$. In the case of using constant learning rates, SGD [19] and Algorithm 1 can be applied to not only convex but also nonconvex optimization. In the case of using diminishing learning rates, SGD [19] and Algorithm 1 had the best convergence rates, $\mathcal{O}(1/\sqrt{n})$. (*) Theorem 1 in [13] shows that a counter-example to the [12] results exists.

ADAM-C1 (Algorithm 1 with (9)) is a modification of Adam [12], using the same parameters, that guarantees convergence. AMSG-C1 coincides with AMSGrad [13]. We implemented ADAM-Ci (resp. AMSG-Ci) ($i = 2, 3$) so that we could compare ADAM-C1 (resp. AMSG-C1) with the proposed algorithms with small constant sub-learning rates. Their performances are compared to those using diminishing sub-learning rates as specified in Theorem III.2, referred to as ADAM-Di and AMSG-Di. Finally, MAMSG-Ci (resp. MAMSG-Di) ($i = 1, 2, 3$) with $\gamma = 0.1$ is a modification of AMSG-Ci (resp. AMSG-Di) with $\gamma = 0$.

All experiments were performed on a fast scalar computation server running at Meiji University. The experimental environment is as follows: two Intel(R) Xeon(R) Gold 6148 at 2.4 GHz CPUs with 20 cores, 16 GB NVIDIA Tesla V100 at 900 Gbps GPU, Red Hat Enterprise Linux 7.6. The code was all written in Python 3.8.2 using the NumPy 1.17.3 and PyTorch 1.3.0 packages.

A. Text classification

Text classification was performed using long short-term memory (LSTM), which is an artificial recurrent neural network (RNN) architecture developed in the field of deep learning for natural language processing. The LSTM implementation included an affine layer and at the output employed a sigmoid function as the activation function. For the text classification tasks, the IMDb dataset[4] was used. This dataset comprises 50,000 movie reviews and the associated binary sentiment polarity labels, and these were split evenly (i.e., 25,000 and 25,000 movie reviews) into training and test sets. The classification of the dataset employed a multilayer neural network, and binary cross-entropy (BCE) was used as the loss function.

In the experiment, constant sub-learning rates yielded better performance with Algorithm 1 than did diminishing sub-learning rates for both training loss and accuracy score (panels (a) and (b), respectively, of Figures 1 and 2). For classification scores, a box-plot comparison is included as Figure 5, where the boxes represent the upper and lower quartiles and the interior horizontal lines are the medians. Looking at the median, we can see that constant sub-learning rates yielded better results than did diminishing sub-learning rates. We believe that the reason for the inferior performance with diminishing sub-learning rates was they became approximately zero after a number of iterations, versus with constant sub-learning rates, for which the learning rates never became zero.

B. Image classification

Image classification was performed using a Residual Network (ResNet), which is a relatively deep model based on a convolutional neural network (CNN). Specifically, a 20-layer ResNet (ResNet-20) was employed and comprised 19 convolutional layers with 3×3 filters and a single 10-way fully connected layer with a softmax function. Following common practice in image classification, for fitting ResNet, the cross-entropy was used as the loss function. The model on ResNet-20 used batch normalization (`nn.BatchNorm2d`) based on `torchvision.models.ResNet`[5] [28]. As the database this task, the CIFAR-10 dataset[6] was used, which is considered

[4]https://datasets.imdbws.com/

[5]https://github.com/pytorch/vision/blob/master/torchvision/models/resnet.py

[6]https://www.cs.toronto.edu/~kriz/cifar.html

a benchmark and is commonly used for image classification. This dataset comprises 60,000 32×32 color images assigned equally to 10 classes (i.e., 6,000 images to each class). In the experiment, 50,000 images were used as the training set and the remaining 10,000 images constituted the test set. The test batch comprised 1,000 randomly selected images from each class.

In the experiment, constant sub-learning rates again yielded better performance with Algorithm 1 than did diminishing sub-learning rates for both training loss and accuracy score (panels (a) and (b), respectively, of Figures 3 and 4). For classification scores, a box-plot comparison is included as Figure 6. In this case, diminishing sub-learning rates yielded mixed results relative to constant sub-learning rates. In particular, as shown in Figure 6(a), Algorithm 1 with constant sub-learning rates yielded good classification scores in terms of the median, the same as in the case of text classification (Figure 5). Figure 6(b) shows that the accuracy scores for the image classification task were less than 20% with diminishing sub-learning rates for all algorithms except AMSG-D1 and AMSG-D2. Therefore, we conclude that using of constant sub-learning rates is superior for training neural networks.

V. Conclusion

Looking at a particular adaptive-learning-rate optimization algorithm used for solving the stationary point problems associated with nonconvex stochastic optimization problems in the field of deep learning, the present study examined constant versus diminishing sub-learning rates by performing separate convergence and convergence rate analyses. For the algorithm with constant sub-learning rates, it was found that the algorithm can solve the problem. In the case of the algorithm with diminishing sub-learning rates, $\mathcal{O}(1/\sqrt{n})$ convergence can be achieved. In the numerical experiments on the stochastic optimization of text and image classification tasks, it was found that the algorithm was successful, whereas Adam and AMSGrad with diminishing sub-learning rates were not. In particular, it was found that the algorithm with constant sub-learning rates is sufficiently suitable for the training of neural networks.

Appendix A
Proofs of Theorems III.1 and III.2 and Propositions III.1 and III.2

Lemma A.1 *Under assumptions (A1), (A2), (C1), and (C2), if $\boldsymbol{x} \in X$ and $n \in \mathbb{N}$, then*

$$\begin{aligned}&\mathbb{E}\left[\left\|\boldsymbol{x}_{n+1}-\boldsymbol{x}\right\|_{\mathsf{H}_n}^2\right]\\ &\leq \mathbb{E}\left[\left\|\boldsymbol{x}_{n}-\boldsymbol{x}\right\|_{\mathsf{H}_n}^2\right]+2\alpha_n\left\{\frac{1-\beta_n}{1-\gamma^{n+1}}\mathbb{E}\left[\langle \boldsymbol{x}-\boldsymbol{x}_n, \nabla f(\boldsymbol{x}_n)\rangle\right]\right.\\ &\quad\left.+\frac{\beta_n}{1-\gamma^{n+1}}\mathbb{E}\left[\langle \boldsymbol{x}-\boldsymbol{x}_n, \boldsymbol{m}_{n-1}\rangle\right]\right\}+\alpha_n^2\mathbb{E}\left[\left\|\mathbf{d}_n\right\|_{\mathsf{H}_n}^2\right].\end{aligned}$$

Proof: For any $\boldsymbol{x} \in X$ and $n \in \mathbb{N}$, the definition of $\boldsymbol{x}_{n+1}$ and nonexpansivity of P_{X,H_n} (i.e., $\|P_{X,\mathsf{H}_n}(\boldsymbol{x}) - P_{X,\mathsf{H}_n}(\boldsymbol{y})\|_{\mathsf{H}_n} \leq \|\boldsymbol{x}-\boldsymbol{y}\|_{\mathsf{H}_n}$ $(\boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^d)$) imply that, almost surely,

$$\begin{aligned}&\left\|\boldsymbol{x}_{n+1}-\boldsymbol{x}\right\|_{\mathsf{H}_n}^2\\ &\leq \left\|\boldsymbol{x}_{n}-\boldsymbol{x}\right\|_{\mathsf{H}_n}^2+2\alpha_n\left\langle \boldsymbol{x}_n-\boldsymbol{x}, \mathbf{d}_n\right\rangle_{\mathsf{H}_n}+\alpha_n^2\left\|\mathbf{d}_n\right\|_{\mathsf{H}_n}^2.\end{aligned}$$

Moreover, the definitions of $\mathbf{d}_n$, $\boldsymbol{m}_n$, and $\hat{\boldsymbol{m}}_n$ ensure that

$$\begin{aligned}&\left\langle \boldsymbol{x}_n-\boldsymbol{x}, \mathbf{d}_n\right\rangle_{\mathsf{H}_n}=\frac{1}{\tilde{\gamma}_n}\left\langle \boldsymbol{x}-\boldsymbol{x}_n, \boldsymbol{m}_n\right\rangle\\ &=\frac{\beta_n}{\tilde{\gamma}_n}\left\langle \boldsymbol{x}-\boldsymbol{x}_n, \boldsymbol{m}_{n-1}\right\rangle+\frac{1-\beta_n}{\tilde{\gamma}_n}\left\langle \boldsymbol{x}-\boldsymbol{x}_n, \mathsf{G}(\boldsymbol{x}_n,\boldsymbol{\xi}_n)\right\rangle,\end{aligned}$$

where $\tilde{\gamma}_n := 1-\gamma^{n+1}$. Hence, almost surely,

$$\begin{aligned}&\left\|\boldsymbol{x}_{n+1}-\boldsymbol{x}\right\|_{\mathsf{H}_n}^2\\ &\leq \left\|\boldsymbol{x}_{n}-\boldsymbol{x}\right\|_{\mathsf{H}_n}^2+2\alpha_n\left\{\frac{\beta_n}{\tilde{\gamma}_n}\left\langle \boldsymbol{x}-\boldsymbol{x}_n, \boldsymbol{m}_{n-1}\right\rangle\right.\\ &\quad\left.+\frac{1-\beta_n}{\tilde{\gamma}_n}\left\langle \boldsymbol{x}-\boldsymbol{x}_n, \mathsf{G}(\boldsymbol{x}_n,\boldsymbol{\xi}_n)\right\rangle\right\}+\alpha_n^2\left\|\mathbf{d}_n\right\|_{\mathsf{H}_n}^2.\end{aligned}\tag{15}$$

Denote the history of process $\boldsymbol{\xi}_0, \boldsymbol{\xi}_1, \ldots$ to time step n by $\boldsymbol{\xi}_{[n]} = (\boldsymbol{\xi}_0, \boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_n)$. The condition $\boldsymbol{x}_n = \boldsymbol{x}_n(\boldsymbol{\xi}_{[n-1]})$ $(n \in \mathbb{N})$, (C1), and (C2) guarantee that

$$\begin{aligned}\mathbb{E}\left[\left\langle \boldsymbol{x}-\boldsymbol{x}_n, \mathsf{G}(\boldsymbol{x}_n,\boldsymbol{\xi}_n)\right\rangle\right]&=\mathbb{E}\left[\mathbb{E}\left[\left\langle \boldsymbol{x}-\boldsymbol{x}_n, \mathsf{G}(\boldsymbol{x}_n,\boldsymbol{\xi}_n)\right\rangle|\boldsymbol{\xi}_{[n-1]}\right]\right]\\ &=\mathbb{E}\left[\left\langle \boldsymbol{x}-\boldsymbol{x}_n, \mathbb{E}\left[\mathsf{G}(\boldsymbol{x}_n,\boldsymbol{\xi}_n)|\boldsymbol{\xi}_{[n-1]}\right]\right\rangle\right]\\ &=\mathbb{E}\left[\left\langle \boldsymbol{x}-\boldsymbol{x}_n, \nabla f(\boldsymbol{x}_n)\right\rangle\right].\end{aligned}$$

Therefore, the lemma follows by taking the expectation of (15). □

Lemma A.2 *Under assumption (C3), $\mathbb{E}[\|\boldsymbol{m}_n\|^2] \leq \tilde{M}^2 := \max\{\|\boldsymbol{m}_{-1}\|^2, M^2\}$ for all $n \in \mathbb{N}$. Under the additional assumption (A3), $\mathbb{E}[\|\mathbf{d}_n\|_{\mathsf{H}_n}^2] \leq \tilde{B}^2\tilde{M}^2/(1-\gamma)^2$ for all $n \in \mathbb{N}$, where $\tilde{B} := \sup\{\max_{i=1,2,\ldots,d} h_{n,i}^{-1/2} : n \in \mathbb{N}\} < +\infty$.*

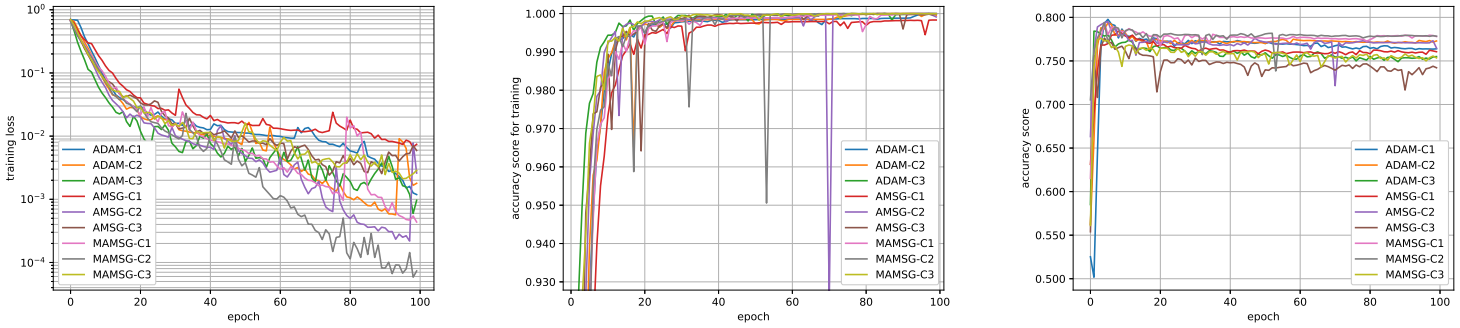
Proof: The convexity of $\|\cdot\|^2$, together with the definition of $\boldsymbol{m}_n$ and (C3), guarantees that, for all $n \in \mathbb{N}$, $\mathbb{E}[\|\boldsymbol{m}_n\|^2] \leq \beta_n\mathbb{E}[\|\boldsymbol{m}_{n-1}\|^2] + (1-\beta_n)M^2$. Induction thus ensures that, for all $n \in \mathbb{N}$,

$$\mathbb{E}\left[\left\|\boldsymbol{m}_n\right\|^2\right] \leq \tilde{M}^2 := \max\left\{\left\|\boldsymbol{m}_{-1}\right\|^2, M^2\right\} < +\infty.\tag{16}$$

For $n \in \mathbb{N}$, $\mathsf{H}_n \succ O$ guarantees the existence of a unique matrix $\overline{\mathsf{H}}_n \succ O$ such that $\mathsf{H}_n = \overline{\mathsf{H}}_n^2$ [29, Theorem 7.2.6]. The relation $\|\boldsymbol{x}\|_{\mathsf{H}_n}^2 = \|\overline{\mathsf{H}}_n\boldsymbol{x}\|^2$ for all $\boldsymbol{x} \in \mathbb{R}^d$ and the definitions of $\mathbf{d}_n$ and $\hat{\boldsymbol{m}}_n$ imply that $\mathbb{E}[\|\mathbf{d}_n\|_{\mathsf{H}_n}^2] = \mathbb{E}[\|\overline{\mathsf{H}}_n^{-1}\mathsf{H}_n\mathbf{d}_n\|^2] \leq (1/\tilde{\gamma}_n^2)\mathbb{E}[\|\overline{\mathsf{H}}_n^{-1}\|^2\|\boldsymbol{m}_n\|^2]$, for all $n \in \mathbb{N}$, where $\|\overline{\mathsf{H}}_n^{-1}\| = \|\mathsf{diag}(h_{n,i}^{-1/2})\| = \max_{i=1,2,\ldots,d} h_{n,i}^{-1/2}$ and $\tilde{\gamma}_n := 1-\gamma^{n+1} \geq 1-\gamma$. Bound (16) and $\tilde{B} := \sup\{\max_{i=1,2,\ldots,d} h_{n,i}^{-1/2} : n \in \mathbb{N}\} \leq \max_{i=1,2,\ldots,d} h_{0,i}^{-1/2} < +\infty$ (by (A3)) imply $\mathbb{E}[\|\mathbf{d}_n\|_{\mathsf{H}_n}^2] \leq \tilde{B}^2\tilde{M}^2/(1-\gamma)^2$, for all $n \in \mathbb{N}$, completing the proof. □

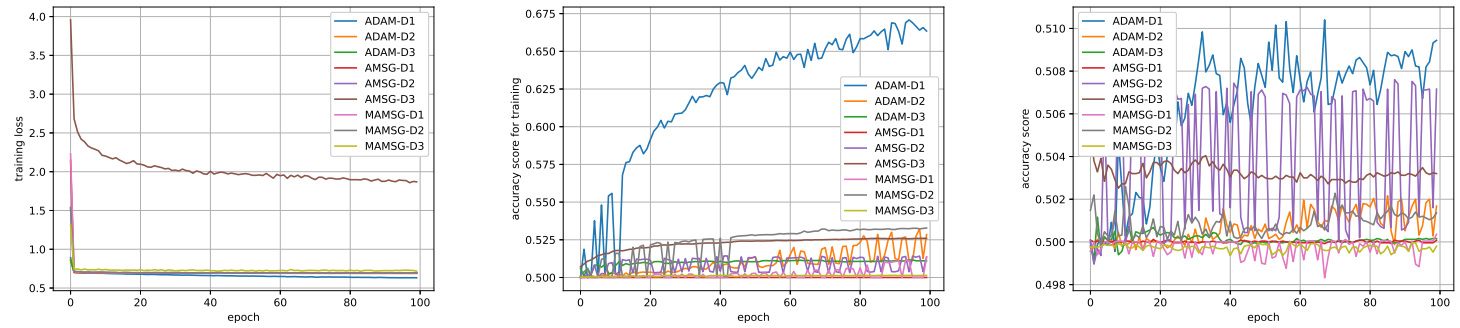
The following theorem is a convergence rate analysis of Algorithm 1.

Theorem A.1 *Under assumptions (A1)–(A5) and (C1)–(C3), suppose that $(\gamma_n)_{n\in\mathbb{N}}$ defined by $\gamma_n := \alpha_n(1-\beta_n)/(1-\gamma^{n+1})$ and $(\beta_n)_{n\in\mathbb{N}}$ satisfy the relations $\gamma_{n+1} \leq \gamma_n$ $(n \in \mathbb{N})$ and*



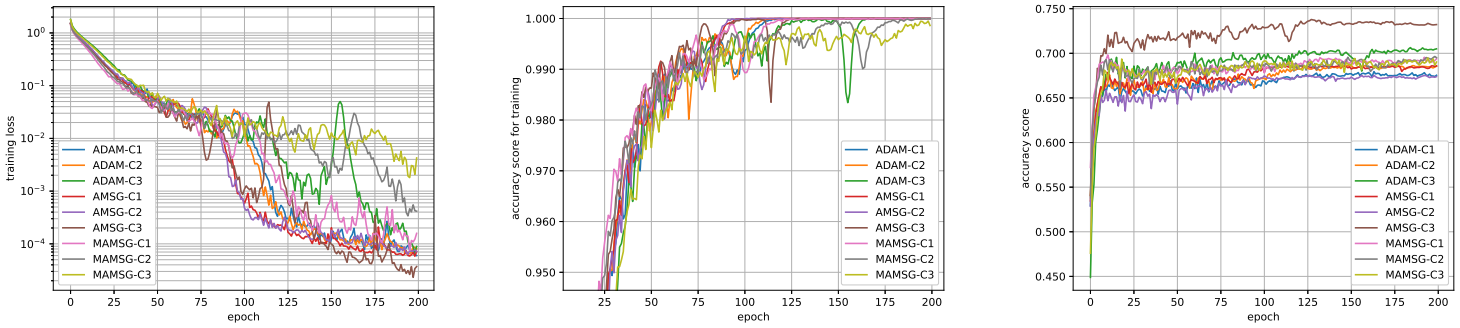
(a) (b) (c)

Fig. 1: (a) Training loss function value, (b) training classification accuracy score, and (c) test classification accuracy score for Algorithm 1 with constant sub-learning rates versus number of epochs on the IMDb dataset



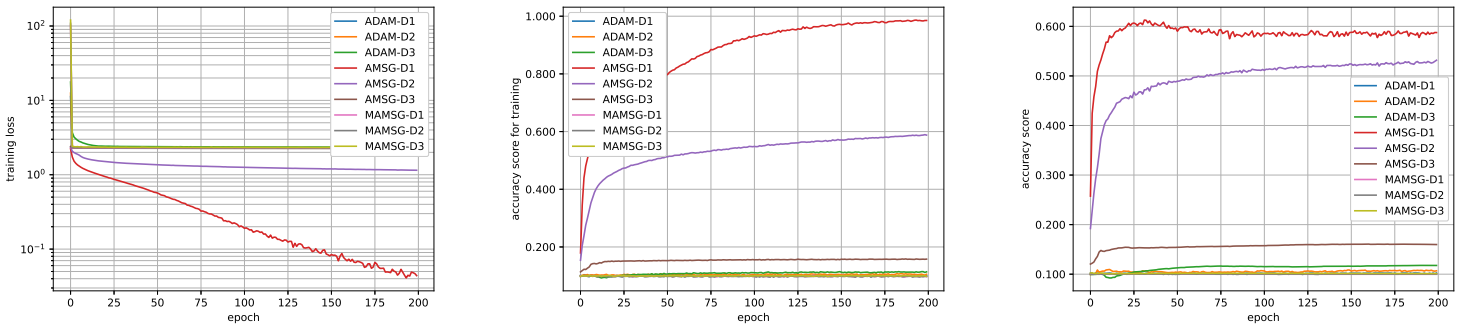
(a) (b) (c)

Fig. 2: (a) Training loss function value, (b) training classification accuracy score, and (c) test classification accuracy score for Algorithm 1 with diminishing sub-learning rates versus number of epochs on the IMDb dataset



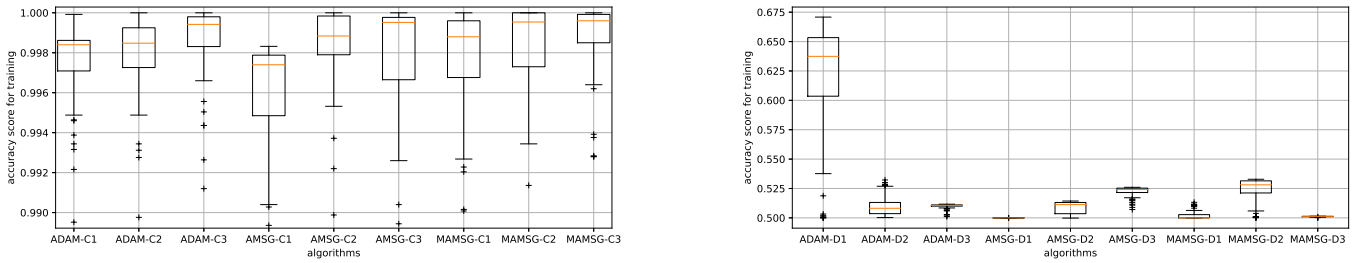
(a) (b) (c)

Fig. 3: (a) Training loss function value, (b) training classification accuracy score, and (c) test classification accuracy score for Algorithm 1 with constant sub-learning rates versus number of epochs on the CIFAR-10 dataset



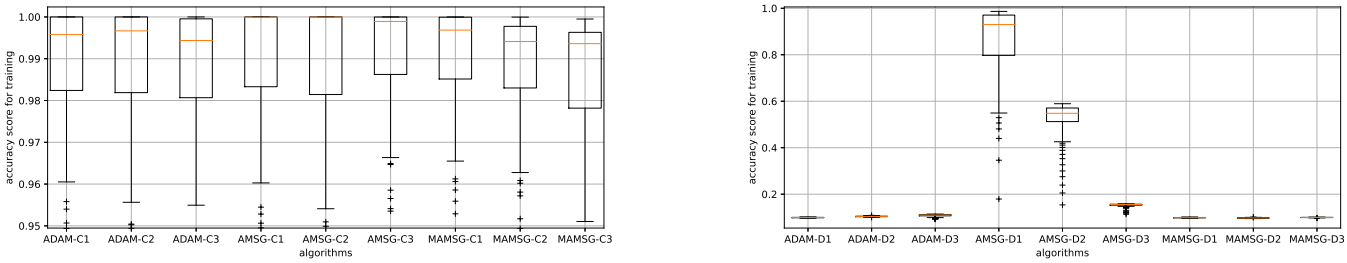
(a) (b) (c)

Fig. 4: (a) Training loss function value, (b) training classification accuracy score, and (c) test classification accuracy score for Algorithm 1 with diminishing sub-learning rates versus number of epochs on the CIFAR-10 dataset



(a) (b)

Fig. 5: (a) Box-plot comparison of Algorithm 1 with constant sub-learning rates and (b) box-plot comparison of Algorithm 1 with diminishing sub-learning rates in terms of training classification accuracy scores on the IMDb dataset



(a) (b)

Fig. 6: (a) Box-plot comparison of Algorithm 1 with constant sub-learning rates and (b) box-plot comparison of Algorithm 1 with diminishing sub-learning rates in terms of training classification accuracy scores on the CIFAR-10 dataset

$\limsup_{n\to+\infty}\beta_n < 1$. *For* $\boldsymbol{x}\in X$ *and* $n\in\mathbb{N}$, *define* $V_n(\boldsymbol{x}) = V_n := \mathbb{E}\left[\langle \boldsymbol{x}_n - \boldsymbol{x}, \nabla f(\boldsymbol{x}_n)\rangle\right]$ *over all* X *and* $\mathbb{N}$. *Then*

$$\frac{1}{n}\sum_{k=1}^{n} V_k \leq \frac{D\sum_{i=1}^{d} B_i}{2\tilde{b}n\alpha_n} + \frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2 n}\sum_{k=1}^{n}\alpha_k + \frac{\tilde{M}\sqrt{Dd}}{\tilde{b}n}\sum_{k=1}^{n}\beta_k,$$

for all $\boldsymbol{x}\in X$ *and all* $n\geq 1$, *where* $\tilde{b} := 1-b$, $\tilde{\gamma} := 1-\gamma$, $(\beta_n)_{n\in\mathbb{N}}\subset(0,b]\subset(0,1)$, $\tilde{M}$ *and* $\tilde{B}$ *are as defined in Lemma A.2, and* D *and* B_i *are as defined in Assumption III.1.*

Proof: Fix $\boldsymbol{x}\in X$ arbitrarily. Lemma A.1 guarantees that, for all $n\geq 1$,

$$\sum_{k=1}^{n} V_k \leq \underbrace{\frac{1}{2}\sum_{k=1}^{n}\frac{1}{\gamma_k}\left\{\mathbb{E}\left[\|\boldsymbol{x}_k-\boldsymbol{x}\|_{\mathsf{H}_k}^2\right]-\mathbb{E}\left[\|\boldsymbol{x}_{k+1}-\boldsymbol{x}\|_{\mathsf{H}_k}^2\right]\right\}}_{\Gamma_n} + \underbrace{\sum_{k=1}^{n}\frac{\beta_k}{\tilde{\beta}_k}\mathbb{E}\left[\langle \boldsymbol{x}-\boldsymbol{x}_k, \boldsymbol{m}_{k-1}\rangle\right]}_{B_n} + \underbrace{\frac{1}{2\tilde{b}}\sum_{k=1}^{n}\alpha_k\mathbb{E}\left[\|\mathbf{d}_k\|_{\mathsf{H}_k}^2\right]}_{A_n}, \tag{17}$$

where $\tilde{\beta}_n := 1-\beta_n$ $(n\in\mathbb{N})$. From the definition of Γ_n and $\mathbb{E}[\|\boldsymbol{x}_{n+1}-\boldsymbol{x}\|_{\mathsf{H}_n}^2]/\gamma_n\geq 0$,

$$\Gamma_n \leq \frac{\mathbb{E}\left[\|\boldsymbol{x}_1-\boldsymbol{x}\|_{\mathsf{H}_1}^2\right]}{\gamma_1} + \underbrace{\sum_{k=2}^{n}\left\{\frac{\mathbb{E}\left[\|\boldsymbol{x}_k-\boldsymbol{x}\|_{\mathsf{H}_k}^2\right]}{\gamma_k}-\frac{\mathbb{E}\left[\|\boldsymbol{x}_k-\boldsymbol{x}\|_{\mathsf{H}_{k-1}}^2\right]}{\gamma_{k-1}}\right\}}_{\tilde{\Gamma}_n}. \tag{18}$$

Since $\overline{\mathsf{H}}_k \succ O$ exists such that $\mathsf{H}_k = \overline{\mathsf{H}}_k^2$, we have $\|\boldsymbol{x}\|_{\mathsf{H}_k}^2 = \|\overline{\mathsf{H}}_k\boldsymbol{x}\|^2$ for all $\boldsymbol{x}\in\mathbb{R}^d$. Accordingly, we have

$$\tilde{\Gamma}_n = \mathbb{E}\left[\sum_{k=2}^{n}\left\{\frac{\left\|\overline{\mathsf{H}}_k(\boldsymbol{x}_k-\boldsymbol{x})\right\|^2}{\gamma_k}-\frac{\left\|\overline{\mathsf{H}}_{k-1}(\boldsymbol{x}_k-\boldsymbol{x})\right\|^2}{\gamma_{k-1}}\right\}\right].$$

Assumption III.1 ensures that we can express H_k as $\mathsf{H}_k = \mathsf{diag}(h_{k,i})$, where $h_{k,i}>0$ $(k\in\mathbb{N}, i=1,2,\ldots,d)$. Hence, for all $k\in\mathbb{N}$ and all $\boldsymbol{x} := (x_i)\in\mathbb{R}^d$,

$$\overline{\mathsf{H}}_k = \mathsf{diag}\left(\sqrt{h_{k,i}}\right) \text{ and } \left\|\overline{\mathsf{H}}_k\boldsymbol{x}\right\|^2 = \sum_{i=1}^{d} h_{k,i}x_i^2. \tag{19}$$

Hence, for all $n\geq 2$,

$$\tilde{\Gamma}_n = \mathbb{E}\left[\sum_{k=2}^{n}\sum_{i=1}^{d}\left(\frac{h_{k,i}}{\gamma_k}-\frac{h_{k-1,i}}{\gamma_{k-1}}\right)(x_{k,i}-x_i)^2\right].$$

From $\gamma_k\leq\gamma_{k-1}$ $(k\geq 1)$ and (A3), we have $h_{k,i}/\gamma_k - h_{k-1,i}/\gamma_{k-1}\geq 0$ $(k\geq 1, i=1,2,\ldots,d)$. Moreover, from (A5), $D := \max_{i=1,2,\ldots,d}\sup\{(x_{n,i}-x_i)^2 : n\in\mathbb{N}\} < +\infty$. Accordingly, for all $n\geq 2$, $\tilde{\Gamma}_n \leq D\mathbb{E}[\sum_{i=1}^{d}(h_{n,i}/\gamma_n - h_{1,i}/\gamma_1)]$. Therefore, (18), $\mathbb{E}[\|\boldsymbol{x}_1-\boldsymbol{x}\|_{\mathsf{H}_1}^2]/\gamma_1 \leq D\mathbb{E}[\sum_{i=1}^{d}h_{1,i}/\gamma_1]$, and (A4) imply, for all $n\in\mathbb{N}$,

$$\Gamma_n \leq \frac{D}{\gamma_n}\mathbb{E}\left[\sum_{i=1}^{d}h_{n,i}\right] \leq \frac{D}{\gamma_n}\sum_{i=1}^{d}B_i,$$

which, together with $\gamma_n := \alpha_n(1-\beta_n)/(1-\gamma^{n+1})$ and $\tilde{b} := 1-b$, implies

$$\Gamma_n \leq \frac{D\sum_{i=1}^{d}B_i}{\tilde{b}\alpha_n}. \tag{20}$$

From the Cauchy-Schwarz inequality and bounds $D := \max_{i=1,2,\dots,d} \sup\{(x_{n,i} - x_i)^2 \colon n \in \mathbb{N}\} < +\infty$ (by (A5)) and $\mathbb{E}[\|\boldsymbol{m}_n\|] \leq \tilde{M}$ $(n \in \mathbb{N})$ (by Lemma A.2), we have that

$$B_n \leq \frac{\sqrt{Dd}}{\tilde{b}} \sum_{k=1}^{n} \beta_k \mathbb{E}\left[\|\boldsymbol{m}_{k-1}\|\right] \leq \frac{\tilde{M}\sqrt{Dd}}{\tilde{b}} \sum_{k=1}^{n} \beta_k, \quad (21)$$

for all $n \in \mathbb{N}$. It follows from $\mathbb{E}[\|\mathbf{d}_n\|_{\mathsf{H}_n}^2] \leq \tilde{B}^2\tilde{M}^2/(1-\gamma)^2$ $(n \in \mathbb{N})$ (by Lemma A.2) that

$$A_n := \sum_{k=1}^{n} \alpha_k \mathbb{E}\left[\|\mathbf{d}_k\|_{\mathsf{H}_k}^2\right] \leq \frac{\tilde{B}^2\tilde{M}^2}{(1-\gamma)^2} \sum_{k=1}^{n} \alpha_k, \quad (22)$$

for all $n \in \mathbb{N}$. Then the assertion of Theorem A.1 follows from (17) and (20)–(22), completing the proof. □

Lemmas A.1 and A.2 and Theorem A.1 lead to Theorem III.1, as follows.

Proof of Theorem III.1: Let $\boldsymbol{x} \in X$, $\alpha_n := \alpha \in (0,1)$, and $\beta_n := \beta = b \in (0,1)$. We show that, for all $\epsilon > 0$,

$$\liminf_{n\to+\infty} V_n \leq \frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2}\alpha + \frac{\tilde{M}\sqrt{Dd}}{\tilde{b}\tilde{\gamma}}\beta + \frac{Dd\epsilon}{2\tilde{b}} + \epsilon. \quad (23)$$

If (23) does not hold for all $\epsilon > 0$, then there exists $\epsilon_0 > 0$ such that

$$\liminf_{n\to+\infty} V_n > \frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2}\alpha + \frac{\tilde{M}\sqrt{Dd}}{\tilde{b}\tilde{\gamma}}\beta + \frac{Dd\epsilon_0}{2\tilde{b}} + \epsilon_0. \quad (24)$$

Because (A3) and (A4) hold, there exists $n_0 \in \mathbb{N}$ such that $n \in \mathbb{N}$ with $n \geq n_0$ implies

$$\mathbb{E}\left[\sum_{i=1}^{d}(h_{n+1,i} - h_{n,i})\right] \leq \frac{d\alpha\epsilon_0}{2}. \quad (25)$$

From (19), (25), (A3), and (A5), for all $n \geq n_0$,

$$X_{n+1} - \mathbb{E}\left[\|\boldsymbol{x}_{n+1} - \boldsymbol{x}\|_{\mathsf{H}_n}^2\right] \leq \frac{Dd\alpha\epsilon_0}{2}, \quad (26)$$

where the bounds $X_n := \mathbb{E}[\|\boldsymbol{x}_n - \boldsymbol{x}\|_{\mathsf{H}_n}^2] \leq D\sum_{i=1}^{d} B_i < +\infty$ hold for all $n \in \mathbb{N}$ from (A4) and (A5). Also, since $\gamma \in [0,1)$, it follows that there exists $n_1 \in \mathbb{N}$ such that $n \in \mathbb{N}$, $n \geq n_1$, implies

$$X_{n+1}\gamma^{n+1} \leq \frac{Dd\alpha\epsilon_0}{2}. \quad (27)$$

The definition of the limit inferior of $(V_n)_{n\in\mathbb{N}}$ guarantees the existence of $n_2 \in \mathbb{N}$ such that $\liminf_{n\to+\infty} V_n - \epsilon_0/2 \leq V_n$, for all $n \geq n_2$. By combining this with (24), we obtain that

$$V_n > \frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2}\alpha + \frac{\tilde{M}\sqrt{Dd}}{\tilde{b}\tilde{\gamma}}\beta + \frac{Dd\epsilon_0}{2\tilde{b}} + \frac{1}{2}\epsilon_0, \quad (28)$$

for all $n \geq n_1$. Thus, (26) taken with Lemmas A.1 and A.2 implies that the following holds for all $n \geq n_3 := \max\{n_0, n_1, n_2\}$:

$$\begin{aligned} X_{n+1} &\leq X_n + \frac{Dd\alpha\epsilon_0}{2} - \frac{2\alpha\tilde{b}}{1-\gamma^{n+1}}V_n + \frac{2\tilde{M}\sqrt{Dd}}{\tilde{\gamma}}\alpha\beta \\ &\quad + \frac{\tilde{B}^2\tilde{M}^2}{\tilde{\gamma}^2}\alpha^2, \end{aligned}$$

where $\tilde{b} := 1-b$ and $\tilde{\gamma} := 1-\gamma$. Hence, from (27), $1-\gamma^{n+1} \leq 1$, and $(X_{n+1} - X_n)\gamma^{n+1} \leq X_{n+1}\gamma^{n+1}$ $(n \in \mathbb{N})$, we have that, for all $n \geq n_3$,

$$\begin{aligned} X_{n+1} &\leq X_n + Dd\alpha\epsilon_0 - 2\alpha\tilde{b}V_n + \frac{2\tilde{M}\sqrt{Dd}}{\tilde{\gamma}}\alpha\beta \\ &\quad + \frac{\tilde{B}^2\tilde{M}^2}{\tilde{\gamma}^2}\alpha^2. \end{aligned} \quad (29)$$

Therefore, (28) ensures that, for all $n \geq n_3$,

$$\begin{aligned} X_{n+1} &< X_n + Dd\alpha\epsilon_0 - 2\alpha\tilde{b}\left\{\frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2}\alpha + \frac{\tilde{M}\sqrt{Dd}}{\tilde{b}\tilde{\gamma}}\beta \right. \\ &\quad \left. + \frac{Dd\epsilon_0}{2\tilde{b}} + \frac{1}{2}\epsilon_0\right\} + \frac{2\tilde{M}\sqrt{Dd}}{\tilde{\gamma}}\alpha\beta + \frac{\tilde{B}^2\tilde{M}^2}{\tilde{\gamma}^2}\alpha^2 \\ &= X_n - \alpha\tilde{b}\epsilon_0 < X_{n_3} - \alpha\tilde{b}\epsilon_0(n+1-n_3). \end{aligned}$$

Note that the right-hand side of the final inequality approaches minus infinity as n approaches positive infinity, producing a contradiction. It follows that (23) holds for all $\epsilon > 0$. Given this arbitrariness of ϵ, we have that $\liminf_{n\to+\infty} V_n \leq (\tilde{B}^2\tilde{M}^2/2\tilde{b}\tilde{\gamma}^2)\alpha + (\tilde{M}\sqrt{Dd}/\tilde{b}\tilde{\gamma})\beta$. Theorem A.1 implies the assertions in Theorem III.1. This completes the proof. □

Lemmas A.1 and A.2 and Theorem A.1 lead to Theorem III.2, as follows.

Proof of Theorem III.2: By an argument similar to that which obtained (29), Lemmas A.1 and A.2 guarantee that

$$\begin{aligned} 2\alpha_k V_k &\leq X_k - X_{k+1} + D\mathbb{E}\left[\sum_{i=1}^{d}(h_{k+1,i} - h_{k,i})\right] + \frac{\tilde{B}^2\tilde{M}^2}{\tilde{\gamma}^2}\alpha_k^2 \\ &\quad + 2\left(\frac{\tilde{M}\sqrt{Dd}}{\tilde{\gamma}} + F\right)\alpha_k\beta_k + D\sum_{i=1}^{d} B_i\gamma^{k+1}, \end{aligned}$$

for all $k \in \mathbb{N}$, where $F := \sup\{|V_n| \colon n \in \mathbb{N}\} < +\infty$ follows from (A2) and (A5). By summing the above inequality from $k = 0$ to $k = n$, we obtain

$$\begin{aligned} 2\sum_{k=0}^{n}\alpha_k V_k &\leq X_0 + D\mathbb{E}\left[\sum_{i=1}^{d}(h_{n+1,i} - h_{0,i})\right] + \frac{\tilde{B}^2\tilde{M}^2}{\tilde{\gamma}^2}\sum_{k=0}^{n}\alpha_k^2 \\ &\quad + 2\left(\frac{\tilde{M}\sqrt{Dd}}{\tilde{\gamma}} + F\right)\sum_{k=0}^{n}\alpha_k\beta_k + D\hat{B}\sum_{k=0}^{n}\gamma^{k+1}, \end{aligned}$$

where $\hat{B} := \sum_{i=1}^{d} B_i$. Let $(\alpha_n)_{n\in\mathbb{N}}$ and $(\beta_n)_{n\in\mathbb{N}}$ satisfy $\sum_{n=0}^{+\infty}\alpha_n = +\infty$, $\sum_{n=0}^{+\infty}\alpha_n^2 < +\infty$, and $\sum_{n=0}^{+\infty}\alpha_n\beta_n < +\infty$. From (A4) and $\gamma \in [0,1)$, we have

$$\sum_{k=0}^{+\infty}\alpha_k V_k < +\infty. \quad (30)$$

Next we show by contradiction that $\liminf_{n\to+\infty} V_n \leq 0$. Suppose not. Then there must exist $\zeta > 0$ and $m_0 \in \mathbb{N}$ such that $V_n \geq \zeta$ for all $n \geq m_0$. It follows from (30) and $\sum_{n=0}^{+\infty}\alpha_n = +\infty$ that $+\infty = \zeta\sum_{k=m_0}^{+\infty}\alpha_k \leq \sum_{k=m_0}^{+\infty}\alpha_k V_k < +\infty$, which is the contradiction. Therefore, $\liminf_{n\to+\infty} V_n \leq 0$ must hold.

Letting $\alpha_n := 1/n^\eta$ $(\eta \in [1/2, 1))$ and $\beta_n := \lambda^n$ $(\lambda \in (0,1))$, it follows that $\gamma_{n+1} \leq \gamma_n$ $(n \in \mathbb{N})$ and $\limsup_{n\to+\infty}\beta_n < 1$. Also, $\lim_{n\to+\infty} 1/(n\alpha_n) =$

$\lim_{n\to+\infty} 1/n^{1-\eta} = 0$ and $(1/n)\sum_{k=1}^{n} \alpha_k \leq (1/n)(1 + \int_1^n \frac{\mathrm{d}t}{t^\eta}) \leq 1/((1-\eta)n^{1-\eta})$. Furthermore, $\sum_{k=1}^{n} \beta_k \leq \sum_{k=1}^{+\infty} \beta_k = \lambda/(1-\lambda)$. These with Theorem A.1 imply that $\min_{k\in[n]} V_k, (1/n)\sum_{k=1}^{n} V_k \leq \mathcal{O}(1/n^{1-\eta})$, completing the proof. □

Proof of Proposition III.1: It follows form the assumption that $F(\cdot, \boldsymbol{\xi})$ is convex for $\boldsymbol{\xi} \in \Xi$ that $\mathbb{E}[f(\boldsymbol{x}_n) - f^\star] \leq V_n$ and $\min_{k\in[n]} \mathbb{E}[f(\boldsymbol{x}_k) - f^\star]$, $\mathbb{E}[f(\tilde{\boldsymbol{x}}_n) - f^\star] \leq (1/n)\sum_{k=1}^{n} \mathbb{E}[f(\boldsymbol{x}_k) - f^\star] \leq (1/n)\sum_{k=1}^{n} V_k$. In the case of problem (1), the relation $R(T) = \sum_{t\in\mathcal{T}}(f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}^\star)) \leq \sum_{t\in\mathcal{T}}\langle \boldsymbol{x}_t - \boldsymbol{x}^\star, \mathsf{G}(\boldsymbol{x}_t, \boldsymbol{\xi}_t)\rangle$ also holds. Thus, Proposition III.1 follows from Theorems III.1 and A.1. □

Proof of Proposition III.2: We need only show that any accumulation point of $(\tilde{\boldsymbol{x}}_n)_{n\in\mathbb{N}}$ belongs to $X^\star$ almost surely. We obtain $\lim_{n\to+\infty} \mathbb{E}[f(\tilde{\boldsymbol{x}}_n) - f^\star] = 0$ from Theorem III.2 and the proof of Proposition III.1. If we let $\hat{\boldsymbol{x}} \in X$ be an arbitrary accumulation point of $(\tilde{\boldsymbol{x}}_n)_{n\in\mathbb{N}} \subset X$, then the existence of $(\tilde{\boldsymbol{x}}_{n_i})_{i\in\mathbb{N}} \subset (\tilde{\boldsymbol{x}}_n)_{n\in\mathbb{N}}$ is guaranteed such that $(\tilde{\boldsymbol{x}}_{n_i})_{i\in\mathbb{N}}$ converges almost surely to $\hat{\boldsymbol{x}}$. The continuity of f and the limit $\lim_{n\to+\infty} \mathbb{E}[f(\tilde{\boldsymbol{x}}_n) - f^\star] = 0$ give us the relation $\mathbb{E}\left[f(\hat{\boldsymbol{x}}) - f^\star\right] = 0$ and thus that $\hat{\boldsymbol{x}} \in X^\star$. □

Acknowledgment

I am sincerely grateful to Editor-in-Chief C. L. Philip Chen and the three anonymous reviewers for helping me improve the original manuscript. I also thank Hiroyuki Sakai for his input on the numerical examples.

References

[1] P. P. San, S. H. Ling, Nuryani, and H. Nguyen, "Evolvable rough-block-based neural network and its biomedical application to hypoglycemia detection system," *IEEE Transactions on Cybernetics*, vol. 44, no. 8, pp. 1338–1349, 2014.

[2] W. Xu, J. Cao, M. Xiao, D. W. C. Ho, and G. Wen, "A new framework for analysis on stability and bifurcation in a class of neural networks with discrete and distributed delays," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2224–2236, 2015.

[3] Y. Lou, Y. He, L. Wang, and G. Chen, "Predicting network controllability robustness: A convolutional neural network approach," *IEEE Transactions on Cybernetics*, 2021.

[4] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.

[5] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[6] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on Optimization*, vol. 19, pp. 1574–1609, 2009.

[7] S. Ghadimi and G. Lan, "Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: A generic algorithmic framework," *SIAM Journal on Optimization*, vol. 22, pp. 1469–1492, 2012.

[8] A. Nedić and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM Journal on Optimization*, vol. 12, pp. 109–138, 2001.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, Cambridge, 2016.

[10] Y. Nesterov, *Lectures on Convex Optimization*. Springer, Switzerland, second ed., 2018.

[11] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[12] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *Proceedings of The International Conference on Learning Representations*, pp. 1–15, 2015.

[13] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of Adam and beyond," *Proceedings of The International Conference on Learning Representations*, pp. 1–23, 2018.

[14] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *Proceedings of Machine Learning Research*, vol. 37, pp. 2048–2057, 2015.

[15] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *preprint*, arXiv:1701.07875 (2017).

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.

[17] X. Chen, S. Liu, R. Sun, and M. Hong, "On the convergence of a class of Adam-type algorithms for non-convex optimization," *Proceedings of The International Conference on Learning Representations*, pp. 1–30, 2019.

[18] B. Fehrman, B. Gess, and A. Jentzen, "Convergence rates for the stochastic gradient descent method for non-convex objective functions," *Journal of Machine Learning Research*, vol. 21, no. 136, pp. 1–48, 2020.

[19] K. Scaman and C. Malherbe, "Robustness analysis of non-convex stochastic gradient descent using biased expectations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1–11, 2020.

[20] H. Chen, L. Zheng, R. A. Kontar, and G. Raskutti, "Stochastic gradient descent in correlated settings: A study on Gaussian processes," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1–12, 2020.

[21] N. Loizou, S. Vaswani, I. Laradji, and S. Lacoste-Julien, "Stochastic polyak step-size for SGD: An adaptive learning rate for fast convergence," *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 130, pp. 1–11, 2021.

[22] M. Zinkevich, M. Weimer, L. Li, and A. Smola, "Parallelized stochastic gradient descent," *Advances in Neural Information Processing Systems*, vol. 23, pp. 1–9, 2010.

[23] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems I*. Springer, New York, 2003.

[24] D. Liang, F. Ma, and W. Li, "New gradient-weighted adaptive gradient methods with dynamic constraints," *IEEE Access*, vol. 8, pp. 110929–110942, 2020.

[25] C. Mendler-Dünner, J. C. Perdomo, T. Zrnic, and M. Hardt, "Stochastic optimization for performative prediction," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1–11, 2020.

[26] J. Zhuang, T. Tang, Y. Ding, S. Tatikonda, N. Dvornek, X. Papademetris, and J. S. Duncan, "AdaBelief optimizer: Adapting stepsizes by the belief in observed gradients," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1–29, 2020.

[27] C. Fang, C.-J. Li, Z. Lin, and T. Zhang, "SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator," *Advances in Neural Information Processing Systems*, vol. 31, pp. 1–11, 2018.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV*, pp. 770–778, 2016.

[29] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, Cambridge, 1985.

Hideaki Iiduka received a Ph.D. degree in mathematical and computing science from Tokyo Institute of Technology, Tokyo, Japan, in 2005. From 2005 to 2007, he was a Research Assistant in the Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, Japan. From 2007 to 2008, he was a Research Fellow (PD) of the Japan Society for the Promotion of Science. From October 2008 to March 2013, he was an Associate Professor in the Network Design Research Center, Kyushu Institute of Technology, Tokyo, Japan. From April 2013 to March 2019, he was an Associate Professor in the Department of Computer Science, School of Science and Technology, Meiji University, Kanagawa, Japan. Since April 2019, he has been a Professor in the same department. He was awarded the 4th Research Encourage Award of ORSJ in August 2014 and the 9th Research Award of ORSJ in September 2019. His research field is optimization theory and its applications to mathematical information science. He is a Fellow of ORSJ and a member of MOS and SIAM.