electronics

MDPI

Article

Incorporating Derivative-Free Convexity with Trigonometric Simplex Designs for Learning-Rate Estimation of Stochastic Gradient-Descent Method

Emre Tokgoz [1], Hassan Musafer [2,*], Miad Faezipour [3] and Ausif Mahmood [2]

[1] School of Computing & Engineering, Quinnipiac University, Hamden, CT 06518, USA
[2] Department of Computer Science & Engineering, University of Bridgeport, Bridgeport, CT 06604, USA
[3] School of Engineering Technology, Electrical and Computer Engineering Technology, Purdue University, West Lafayette, IN 47907, USA
* Correspondence: hmusafer@bridgeport.edu

Abstract: This paper proposes a novel mathematical theory of adaptation to convexity of loss functions based on the definition of the condense-discrete convexity (CDC) method. The developed theory is considered to be of immense value to stochastic settings and is used for developing the well-known stochastic gradient-descent (SGD) method. The successful contribution of change of the convexity definition impacts the exploration of the learning-rate scheduler used in the SGD method and therefore impacts the convergence rate of the solution that is used for measuring the effectiveness of deep networks. In our development of methodology, the convexity method CDC and learning rate are directly related to each other through the difference operator. In addition, we have incorporated the developed theory of adaptation with trigonometric simplex (TS) designs to explore different learning rate schedules for the weight and bias parameters within the network. Experiments confirm that by using the new definition of convexity to explore learning rate schedules, the optimization is more effective in practice and has a strong effect on the training of the deep neural network.

Keywords: derivative-free convexity; trigonometric simplex design; stochastic gradient descent; adaptive learning rate; deep neural network

Citation: Tokgoz, E.; Musafer, H.; Faezipour, M.; Mahmood, A. Incorporating Derivative-Free Convexity with Trigonometric Simplex Designs for Learning-Rate Estimation of Stochastic Gradient-Descent Method. *Electronics* **2023**, *12*, 419. https://doi.org/10.3390/electronics12020419

Academic Editor: Prasan Kumar Sahoo

Received: 13 December 2022
Revised: 9 January 2023
Accepted: 10 January 2023
Published: 13 January 2023

1. Introduction

The nature of convexity of many machine learning models has not been addressed properly in literature, particularly when it comes to train parameters of neural networks. The first-order methods such as the stochastic gradient descent (SGD) and its variants are the preferred techniques for optimizing neural networks and many other machine learning algorithms. However, these methods do not consider the learning activity of the parameters in the different layers of neural networks. Therefore, there is a need to calculate learning rates mathematically for the individual parameters in a deep neural network and better understand the learning hierarchy of the different layers of the network. In machine learning applications including deep learning, a number of different convexity definitions have been presented in the literature (e.g., see a recent review by [1]). In addition, based on the definition of convexity, SGD for empirical risk minimization is utilized to converge to a global optimum for convex loss and non-convex loss of objective functions [2]. The nature of strong convexity requires the objective function $f : \mathbb{R}^n \to \mathbb{R}$ to be twice differentiable (i.e., C^2) for all of its variables, and the classical Hessian matrix given in Equation (1) can be used for determining convexity results for the function.

$$H_f = \left[\frac{\partial^2 f}{\partial w_{jk}}\right]_{d\times d} \tag{1}$$

The iterative SGD method applies the gradient operator $\nabla = \frac{\partial}{\partial w}$ to the function f (i.e., ∇f) with the stochastic variables ξ and $w \in \mathbb{R}^d$ and calculates:

$$w_{t+1} = w_t - \eta_t \nabla f(w_t, \xi_t) \tag{2}$$

where η_t is the learning rate.

At each evaluation, SGD selects a random training sample from the training dataset; then, the network output is computed to perform the sub-gradient of the loss function over the selected sample, and the algorithm adjusts the network parameters [3]. Therefore, determining an efficient learning rate η_t is crucial for successfully solving machine learning problems as a part of the corresponding SGD algorithm. Strong convexity and w-convexity presented recently by [1] are two of the convexity definitions introduced in the literature for solving stochastic optimization problems. In this work, we use the SGD method to solve the well-known stochastic optimization problem:

$$\min_{w \in \mathbb{R}^d} \{F(w) = E[f(w; \xi)]\} \tag{3}$$

where ξ is a random variable of a stochastic distribution [4]. One can define

$$f_i(w) := f(w; \xi_i) \tag{4}$$

for a given training set $\{w; \xi_i\}$, where ξ_i is a random variable that is defined by a single random sample $\{x_i; y_i\}_{i=1}^m$ pulled uniformly from the training set. The empirical risk minimization reduces to

$$\min_{w \in \mathbb{R}^d} \left\{ F(w) = \frac{1}{m} \sum_{i=1}^{m} f_i(w) \right\} \tag{5}$$

The existence of the unbiased gradient estimator (i.e., $\nabla F = E_{\xi}[\nabla f(w; \xi)]$) is required for any fixed w to apply the SGD in its form of Equation (3).

We present a brief glance at gradient descent optimization algorithms that mainly contributed to the development of the learning-rate schedulers. Duchi et al. introduced the "AdaGrad" method, an adaptive learning-rate based on previous knowledge gained from observing the accumulative sum of squared gradients in earlier iterations [5]. The proposed subgradient-based learning has improved the robustness of the SGD algorithm by controlling the gradient steps of the algorithm [6]. "AdaDelta" is an enhanced version of "AdaGrad", restricting the past accumulated gradients to be a fixed window size [7]. This window is implemented as an exponentially decaying average of the squared gradients. The new implementation ensures that a separate dynamic learning rate is computed on a per-dimension basis. The adaptive-moment estimation, "Adam" [8], is designed to combine the heuristics of the exponential decaying average of past gradients "AdaGrad" with the root mean square prop or "RMSprop" of the exponential average of square of gradients [9]. Adam is observed to be robust and particularly well-suited for non-convex optimization problems. A new variant of the Adam method is "AMSGrad" [10], which relies on the long-term memory of past gradients. The AMSGrad is proposed to develop a new principled exponential moving average because it has been shown that the reliance on only the past few gradients to update the learning rate can result in poor convergence rates. A successful contribution to SGD with diminishing learning rates is performed for convex objective functions by [1]. The defined framework is characterized by a core property, called curvature. Based on the curvature, a new inequality is derived to find an optimal sequence of learning rates by solving a differential equation.

2. CDC and Optimization

CDC is introduced as a nonlinear real extensible closed form function $f : \mathbb{Z}^d \to \mathbb{R}$ by [11]. For the sake of completeness, we summarize the CDC results relevant to SGD optimization.

The first difference of an integer variable function $f : \mathbb{Z}^d \to \mathbb{R}$ is defined by:

$$\triangle_j f(w) = f(w + e_j) - f(w) \tag{6}$$

where e_j represents the integer vector of the unit length at the jth position of the function f towards the direction of jth dimension. The difference of the first difference, namely, the second difference of f is defined by Equation (7) below.

$$\triangle_{jk} f(w) = f(w + e_j + e_k) - f(w + e_j) - f(w + e_k) + f(w) \tag{7}$$

The $d \times d$ discrete Hessian matrix corresponding to the function f contains the second differences and this matrix is presented in Equation (8) as follows:

$$A_f = \left[\triangle_{jk} f\right]_{d\times d} = \begin{bmatrix} \triangle_{11} f & \triangle_{12} f & \cdots & \triangle_{1d} f \\ \triangle_{21} f & \triangle_{22} f & \cdots & \triangle_{2d} f \\ \vdots & \vdots & \ddots & \vdots \\ \triangle_{d1} f & \triangle_{d2} f & \cdots & \triangle_{dd} f \end{bmatrix}_{d\times d} \tag{8}$$

This Hessian matrix is introduced in local settings, and the convexity results are obtained for condense-discrete convex functions similar to the convexity results obtained in real convex analysis. The discrete Hessian matrix A_f is shown to be symmetric and linear, and it vanishes when the condense discrete function is affine. The coefficient matrix A_f of $f : D \to \mathbb{R}$ is shown to satisfy the properties of the Hessian matrix corresponding to real convex functions. That is, A_f is linear with respect to the condense discrete functions, symmetric, and vanishes when f is discrete affine. It is also shown that a function $f : D \to \mathbb{R}$ is condense-discrete convex if and only if the corresponding discrete Hessian matrix is positive definite in D.

To obtain minimization results for a given condense-discrete convex function, we require the given condense-discrete convex function to be C^1.

Assuming $f : \mathbb{Z}^d \to \mathbb{R}$ is a C^1 strict condense-discrete convex function, the set of local minimums of f form a set of global minimums and vice versa. The importance of applying CDC to stochastic gradient-descent calculations is the elimination of the second differential operator for determining convexity. Given a function without knowing its convexity structure, CDC determines the convexity within the domain without calculating derivatives of the function.

2.1. SGD and CDC Functions

Given a function $f : B \to \mathbb{R}$ such that $B \subseteq \mathbb{R}^d$, the condensed convexity of f can be checked by showing that the corresponding discrete Hessian matrix H_f is positive definite; therefore, CDC allows for convexity calculations by using simple mathematical operations. The existence of w_* depends on the assumption $\nabla f = 0$. In this section, we use the definition of condense-discrete convexity as a part of the SGD algorithm and its application. Using the iterative procedure

$$w_{j(t+1)} = w_{jt} - e_j \tag{9}$$

where $|e_j| = 1$ for and using the definition of the first difference of f, we introduce:

$$e_j = \eta_{jt} \nabla f(w_{jt}; \xi_{jt}) \tag{10}$$

Noting that the iterative procedure follows the directional method, we use the jth entry $\frac{\partial f}{\partial w}$ of the gradient vector ∇f; therefore, we attain:

$$\eta_{jt} = \frac{e_j}{\frac{\partial f(w_{jt};\xi_{jt})}{\partial w_j}} \tag{11}$$

in the vector form and

$$\eta_{jt} = \frac{1}{\frac{\partial f(w_{jt};\xi_{jt})}{\partial w_j}} \tag{12}$$

in the scalar form towards the jth direction satisfying the first difference $\triangle_j f$. By using the function differentiation definition, the differential of f can be approximated by choosing sufficiently small γ_{jt} such that

$$\frac{\partial f(w_{jt};\xi_{jt})}{\partial w_j} \simeq \frac{f(w_{jt}+\gamma_{jt};\xi_{jt}) - f(w_{jt};\xi_{jt})}{\gamma_{jt}} \tag{13}$$

indicating

$$\eta_{jt} = \frac{\tau_t \gamma_{jt}}{f(w_{jt}+\gamma_{jt};\xi_{jt}) - f(w_{jt};\xi_{jt})} \tag{14}$$

where η_{jt} is the step size of the iterative procedure in the jth dimension of the directional derivative and τ_t is a non-negative scalable parameter. The use of τ_t is a key tuning component that is essential for defining the step size for adjusting it based on the algorithmic solution.

2.2. CDC Examples

This section presents the condense-discrete convexity of logistic regression examples that are shown to be convex, $w-convex$, and strongly convex by [1]. These examples are going to be used for attaining experimental results in Section 3.

$$f_i(w) = \log\left(1 + e^{-y_i x_i^T w}\right) \quad (convex) \tag{15}$$

$$f_i^{(a)}(w) = f_i(w) + \lambda||w|| \quad (w-convex) \tag{16}$$

$$f_i^{(b)}(w) = f_i(w) + \lambda G(w) \quad (w-convex) \tag{17}$$

$$f_i^{(c)}(w) = f_i(w) + \frac{\lambda}{2}||w||^2 \quad (strongly\ convex) \tag{18}$$

where $f_i(w)$ is a convex function, $f_i^{(a)}(w)$ and $f_i^{(b)}(w)$ are w-convex functions, and $G(w) = e^w + e^{-w} - 2 - w^2$. The following calculations prove that $f_i(w)$ is a CDC:

$$\triangle_{11} f(w) = \log\left(1 + e^{-y_i x_i^T (w+2)}\right) - 2\log\left(1 + e^{-y_i x_i^T (w+1)}\right) + \log\left(1 + e^{-y_i x_i^T w}\right)$$

$$\triangle_{11} f(w) = \log\left(e^{y_i x_i^T} \frac{\left(e^{y_i x_i^T (w+2)} + 1\right)\left(e^{y_i x_i^T w} + 1\right)}{\left(e^{y_i x_i^T (w+1)} + 1\right)^2}\right)$$

For simplicity we let $c_i = e^{y_i x_i^T}$, then

$$\triangle_{11} f(w) = \log\left(c_i \frac{\left(c_i^2 c_i^w + 1\right)\left(c_i^w + 1\right)}{\left(c_i c_i^w + 1\right)^2}\right)$$

$$\triangle_{11} f(w) = \log\left(c_i \frac{c_i^2 c_i^{2w} + \left(c_i^2 + 1\right) c_i^w + 1}{c_i^2 c_i^{2w} + 2 c_i c_i^w + 1}\right)$$

The second difference $\triangle_{11} f$ is non-negative for $c_i^2 + 1 > 2c_i$ which holds for $c_i > 1$. Therefore, $\triangle_{11} f$ is non -negative for $y_i x_i^T > 0$ that naturally holds in a data set for non-negative input x and output y.

Now, we explain the condense-discrete convexity of the function $f_i^{(a)}$:

$$\triangle_{11} f_i^{(a)}(w) = \triangle_{11} f_i(w) + \lambda \triangle_{11} ||w||$$

It is shown by [12] that the $2 - norm$ (i.e., $||w||$) is a condense-discrete convex function; noting that $f_i(w)$ is also CDC, the summation of the two functions, $f_i^{(a)}(w)$, is also a CDC. Next, we show that $G(w)$ is a CDC function:

$$\Delta_1 G(w) = e^{w+1} + e^{-(w+1)} - 2 - (w+1)^2 - \left(e^w + e^{-w} - 2 - w^2\right)$$

$$= (e-1)e^w + \left(e^{-1} - 1\right)e^{-w} - (2w+1)$$

$$\Delta_{11} G(w) = (e-1)e^{w+1} + e^{-(w+1)}\left(e^{-1} - 1\right) - (2$$

$$(w+1)+1) - \left[(e-1)e^w + e^{-w}\left(e^{-1} - 1\right) - (2w+1)\right]$$

$$= \left(e^2 - 2e^1 + 1\right)\left(e^w + e^{-(w+2)}\right) - 2 > 0 \text{ for } w > 1$$

Therefore, the convex, $w - convex$, and strongly convex examples we examined are condense-discrete convex functions.

2.3. Learning-Rate Estimation

In this work, we utilize the Hassan–Nelder–Mead algorithm (HNM) to tune the hyperparameters of Equation (14) and help in estimating a set of optimal learning rates for the different weights and biases of the loss functions [13–15]. The HNM algorithm is a variant of the famous Nelder–Mead algorithm [16], which allows the k-dimensional simplex to break down into a set of trigonometric simplex designs that work sequentially to locate a minimum of a nonlinear function. In addition, the HNM algorithm has delivered a higher accuracy than a famous Matlab function, known as "fminsearch", for handling unconstrained optimization problems. To create k-trigonometric simplex designs of the HNM algorithm, we need to generate 5 vertices that reflect 5 different initialization points in k-dimensional space. The 5 vertices of the standard HNM algorithm are the points $(p_1, ..., p_5) \in \mathbb{R}^k$. In this particular case, the vertex parameters are the parameters of the neural network, including weights and biases. After creating the vertices of the HNM algorithm, we need to arrange them in ascending order according to the values of the objective function.

In the above design of the learning rates scheduler, we have noticed that from Equation (14), if the SGD algorithm proceeds successfully to the next iteration, then a good set of γ_t vector has to be extracted from exploring the solution space of any of the convex objective functions defined in the previous section. So, if we assume the starting vector of the randomly initialized parameters of a deep network is the initial vertex of the HNM algorithm, then we can generate the other vertices using Pfefferś method [17] and run the HNM algorithm to explore the neighborhood around w_0. For example, suppose that the objective function is convex and defined as in Equation (15). For a given training set $(x_i, y_i)_{i=1}^n$, we allow the HNM simplex optimization to explore the solution space and extract different features of non-isometric reflections for the next vector w_1, which has a lower function evaluation than w_0. After the vector $(w_1 = w_0 + \gamma_t)$ is determined, the optimal values of γ_t and the constant value of δ can be found to adjust the learning rates scheduler for the next iteration. Hence, the values of τ_t are calculated for each iteration and set to $(\delta / Max(\gamma_t))$.

If we train the network to learn the characteristics of an objective function relative to a particular case or dataset by forcing the network to update its parameters with a single learning rate, then there is a possibility that the network can converge to a non-stationary point or fall into a local minimum. On the contrary, our solution is to examine the solution space for optimal step sizes that individual parameters in the network can perform and use to optimize the network to find an optimal point . Some parameters in the network can push the learning process faster than others; therefore, they need larger learning rates, while others need to slow down and thus need smaller learning rates . The advantage of using the HNM algorithm is that it generates independent trigonometric simplex designs that can extract distinct non-isometric reflections to sequentially estimate different and adaptive learning rates for the parameters of the network.

3. Discussion

This section is devoted to experimental results by testing the CDC results and the learning-rate estimation introduced in the previous section, using the HNM algorithm. In addition, the test is designed to examine the performance of the proposed learning-rate scheduler on a logistic regression dataset, "mushroom", introduced by [18]. The proposed framework includes various modules for data cleaning up, preprocessing, and normalization. In order to provide a comprehensive evaluation of the performance of the proposed learning-rate scheduler, we conduct four experiments on the "mushroom" dataset from the UCI machine learning repository, which is a binary classification problem. We test Equation (14) for the convex logistic regression examples given in the previous section that are shown to be CDC. The additional tests compare the proposed learning-rate scheduler for the adaptive SGD algorithm to state-of-the-art models such as [1] .

The results given in Figure 1 indicate that the proposed learning-rate scheduler has helped the hidden layers of the network to adapt efficiently to an optimal solution. It is also observed that the network shows fast convergence rates as the different weight and bias parameters of the network are characterized by different learning rates. Our solution confirms the known results in the literature such as the previous study introduced by [1], which indicates that the optimal performance of training a neural network is obtained by a diminishing step size scheduler as the network progresses in terms of evaluations. In [1], a new definition of curvature of convex objective functions is presented, and the value of the curvature property determines the optimal learning rates for deep networks. The best step size that is determined for Equation (15) is $\eta_t = 0.1/\sqrt{t}$ [1]. In this work, however, the trigonometric simplex designs explore the solution space of the loss function around the neighborhood with respect to the values of the network parameters and determine the optimal sequence of the learning-rate scheduler based on the CDC definition.

The experimental results on Equations (16) and (17) are shown in Figures 2 and 3, reflecting the computational results on regularized and unregularized logistic regression examples. These test results prove the successful contribution of the CDC definition to estimate a vector of optimal learning rates for different weights and biases and have resulted in developing an efficient deep learning network architecture. When comparing our results to those of [1], the pattern of the learning rates results is almost similar when exploring w-convex loss functions (Equations (16) and (17)); however, the scale of diminishing learning rates presented in this study is more efficient than [1] in stabilizing training and accelerating the convergence rate due to the use of the simplex optimization method to explore the properties of the objective functions. Thus, the adaptive step sizes provide different learning activities for the parameters of the network compared to the use of one diminishing step size to update the network parameters. In particular, the optimal step sizes proposed by [1] for Equations (16) and (17) are $\eta_t = 0.1/t^{1/1.25}$ and $\eta_t = 0.1/t^{1/1.5}$, respectively.

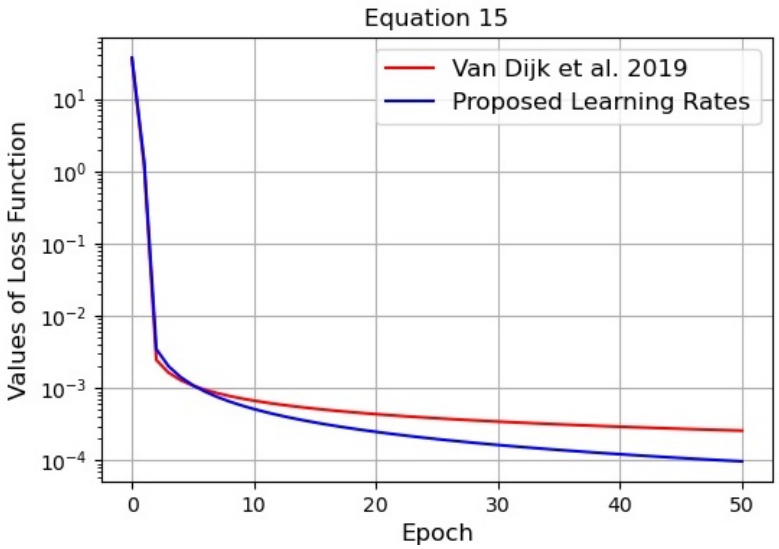

Figure 1. Values of loss function and convergence rate for convex binary problem [1].

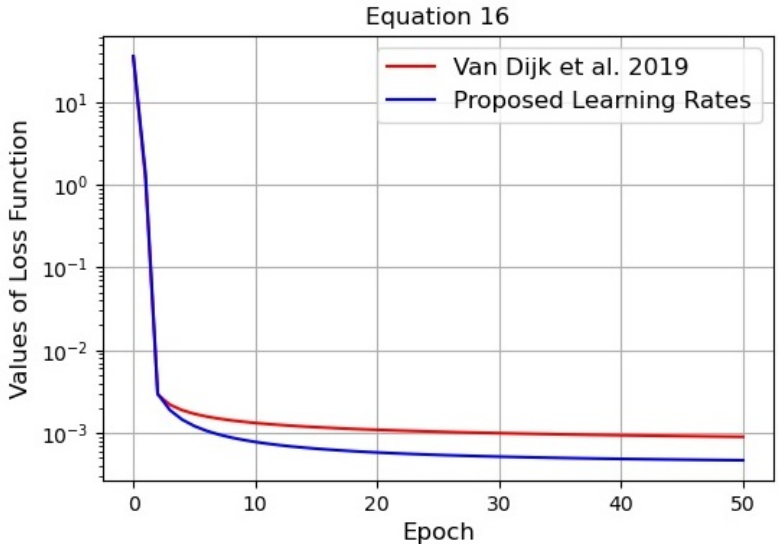

Figure 2. Values of loss function and convergence rate for w-convex binary problem (Equation (16)) [1].

Figure 4 displays the experimental results using our theoretical framework on the function given in Equation (18). The empirical results show that the proposed learning-rate scheduler achieves remarkable success in obtaining a faster convergence rate for optimizing the SGD method. In addition, the idea of adapting network parameters to various levels of learning enhances the effectiveness of the neural network for analyzing convex optimization problems. CDC-based updates on learning rates proved to perform better than a single rate-based method to adjust the network parameters. The main problem of adjusting network parameters based on a single adaptive rate comes from the fact that the objective function for the multilayer network is not an explicit function of the weights and biases in the hidden layers. The effectiveness of our framework is characterized by allowing the parameters of the objective function to be tuned with respect to the architecture and performance of the deep neural network.

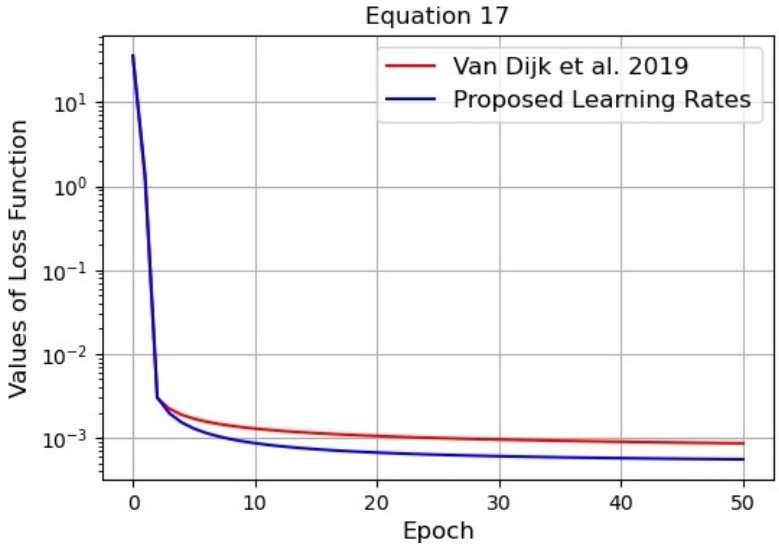

Figure 3. Values of loss function and convergence rate for w-convex binary problem (Equation (17)) [1].

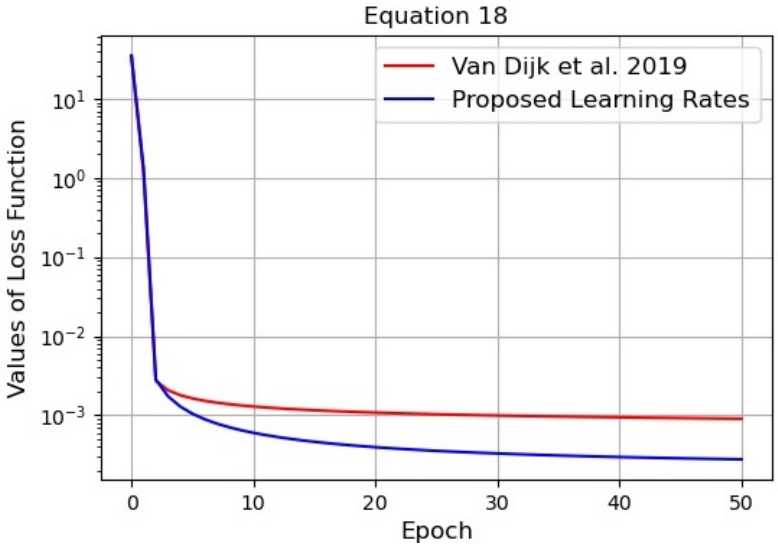

Figure 4. Values of loss function and convergence rate for strongly convex binary problem (Equation (18)) [1].

4. Conclusions

In this work, we introduced a new convexity definition to calculate the learning-rate scheduler for the SGD method. This convexity method and learning rate are directly related to each other, and this work is the first time such a relationship between the convexity and learning rate has been introduced, to the best of our knowledge; learning rate calculations follow from the first difference of a given function with a modification by using a tuning parameter, while the condense-discrete convexity determination follows from the second difference of the given function. The developed theory incorporates CDC with a sequence of trigonometric simplex designs to explore various characteristics of convex, w-convex, and strongly convex loss functions and determine an optimal vector of learning rates for SGD to adjust the network parameters. In fact, the proposed learning-rate scheduler can be used for other convex optimization applications pertaining to deep learning and pattern analysis. The four functions used by [1] for computational results are also used for attaining the numerical results in this work after showing that they are CDC. The computational results proved that the different parameters of the network could increase their adaption at various levels of the learning hierarchy when they are characterized by different step sizes. Finally, the proposed optimization solution has an advantage over the solution attained

by [1]: being able to work on a given problem without knowing its curvature conditions, while requiring statistical estimate tests based on trigonometric simplex evaluations .

Author Contributions: Supervision E.T., A.M. and M.F.; writing—original draft preparation, H.M., E.T., M.F. and A.M.; writing—review & editing, A.M., E.T. and M.F.; conceptualization, H.M., E.T. and A.M.; methodology, H.M., E.T., M.F. and A.M.; software, H.M.; validation, E.T., M.F. and A.M.; formal analysis, H.M., E.T., M.F. and A.M.; investigation, A.M., E.T. and M.F.; resources, E.T., M.F. and A.M.; data curation, H.M.; visualization, H.M.; and project administration, A.M., E.T. and M.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AdaDelta	adaptive-learning-rate method
AdaGrad	adaptive-gradient algorithm
Adam	adaptive-moment estimation
CDC	condense-discrete convexity
HNM	Hassan–Nelder–Mead
SGD	stochastic gradient descent
TS	trigonometric simplex

References

1. Van Dijk, M.; Nguyen, L.; Nguyen, P.H.; Phan, D. Characterization of convex objective functions and optimal expected convergence rates for sgd. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
2. Kawaguchi, K.; Lu, H. Ordered sgd: A new stochastic optimization framework for empirical risk minimization. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Palermo, Italy, 26–28 August 2020.
3. Demuth, H.D.; Beale, M.H.; De Jess, O.; Hagan, M.T. The title of the cited contribution. In *Neural Network Design*; Martin Hagan: San Francisco, CA, USA, 2014; pp. 9-1–9-38.
4. Robbins, H.; Monro, S. A stochastic approximation method. *JSTOR* **1951**, 400–407. [CrossRef]
5. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 7.
6. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
7. Zeiler, M.D. Adadelta: An adaptive learning rate method. *arXiv* **2012**, arXiv:1212.5701.
8. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
9. Hinton, G. Neural networks for machine learning. *Coursera Video Lect.* **2012**, *264*, 1.
10. Reddi, S.J.; Kale, S.; Kumar, S. On the convergence of adam and beyond. *arXiv* **2019**, arXiv:1904.09237.
11. Tokgöz, E.; Nourazari, S.; Kumin, H. Convexity and optimization of condense discrete functions. In Proceedings of the International Symposium on Experimental Algorithms, Crete, Greece, 5–7 May 2011.
12. Tokgöz, E.; Trafalis, T.B. Optimization of an SVM QP Problem Using Mixed Variable Nonlinear Polynomial Kernel Map and Mixed Variable Unimodal Functions. *Wseas Trans. Syst. Control* **2012**, *7*, 16–25.
13. Musafer, H.; Mahmood, A. Dynamic Hassan–Nelder–Mead with simplex free selectivity for unconstrained optimization. *IEEE Access* **2018**, *6*, 39015–39026. [CrossRef]
14. Musafer, H.; Abuzneid, A.; Faezipour, M.; Mahmood, A. An Enhanced Design of Sparse Autoencoder for Latent Features Extraction Based on Trigonometric Simplexes for Network Intrusion Detection Systems. *Electronics* **2020**, *9*, 259. [CrossRef]
15. Musafer, H.; Tokgoz, E.; Mahmood, A. High-dimensional normalized data profiles for testing derivative-free optimization algorithms. *PeerJ Comput. Sci.* **2022**, *8*, e960. [CrossRef] [PubMed]
16. Nelder, J.A.; Mead, R. A simplex method for function minimization. *Comput. J.* **1965**, *7*, 308–313. [CrossRef]
17. Fan, E. Global Optimization of the Lennard-Jones Atomic Cluster. Master's Thesis, McMaster University, Hamilton, ON, USA, 2002.
18. Merz, C.J. UCI Repository of Machine Learning Databases. 1989. Available online: http://www.ics.uci.edu/~mlearn/MLRepository.html (accessed on 9 January 2023).