Tsinghua Science and Technology
Doi: 10.26599/TST.2025.9010048

Research Article

Design and Validation of Optimizer Parameter Critical Conditions via Coupled Transfer Functions and Phase Trajectories

Biyuan Yao†, Guiqing Li†, Xiaokang Wang*, Qingchen Zhang, Yongwei Nie, and Jing Chen

Abstract The design of optimizer parameters affects model performance and is widely applied in fields such as image analysis, autonomous driving, and security monitoring. However, the interpretability and generalizability of optimizers are insufficient, limiting their practical applications. To address these challenges, we introduce a novel approach using transfer function and phase trajectory methods to design the parameters and critical conditions for Stochastic Gradient Descent with Momentum (SGD-M) and Nesterov Accelerated Gradient (NAG). The proposed theory is verified through numerical examples and image recognition experiments. First, using the phase trajectory method, a qualitative analysis of the responses of SGD-M and NAG to initial states is conducted, revealing the influence of parameters on the phase trajectory. Then, through the transfer function method, a quantitative analysis of the unit step response of SGD-M and NAG is performed to explain the impact of parameters on system response. Finally, numerical examples and image recognition experiments verify the significant impact of the momentum control parameter $g(\mu)$ and momentum parameter α on optimizer performance, stability, and time-domain characteristics. Experimental results show that adjusting $g(\mu)$ or α improves image classification accuracy on the Modified National Institute of Standards and Technology (MNIST) and Canadian Institute for Advanced Research (CIFAR-10) datasets. It reduces the loss value, validating the effectiveness of the proposed theory.

Keywords Optimizers; transfer function; stability; parameter selection; time-domain features; image recognition

• Biyuan Yao† and Guiqing Li† contribute equally to this paper.
• Biyuan Yao, Xiaokang Wang, Qingchen Zhang, and Jing Chen are with the School of Computer Science and Technology, Hainan University, Haikou 570100, China. E-mail: {byyao, xkwang, zhangqingchen, chenjing724}@hainanu.edu.cn.
• Guiqing Li and Yongwei Nie are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China. E-mail: {ligq, nieyongwei}@scut.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2024-11-22; revised: 2025-03-22; accepted: 2025-03-28

1 Introduction

With the rapid development of Artificial Intelligence (AI) technologies, Deep Learning (DL), as its core engine, has been widely applied in fields such as image recognition [1–3], control [4], and more. Optimizers play a crucial role in model training, impacting convergence speed, and performance. They improve convergence, shorten time in computer vision, and enhance understanding in sentiment analysis. Thus, optimizers are vital for both theoretical exploration and practical applications.

Optimizers, crucial for parameter adjustment in DL to minimize loss functions [5], yet face challenges (see Fig. 1). First, suboptimal settings lead to gradient issues, oscillations, and inefficiency [6]. Second, research often relies on experience and extensive experimental tuning, lacking systematic qualitative analysis [7]. This results in blind and unexplained parameter settings, severely affecting the efficiency and performance of model training. Again, most studies focus on data-driven and statistical analysis, lacking precise quantitative analysis methods [8], which limits the application of optimizers in complex tasks. Finally, although the optimizer reduces the value of the loss function through a finite number of iterations, studies for gradient descent optimizer order are less concerned with the effect of the modulation coefficients on the training process, and lack the necessary tools for preselection of parameter ranges [9]. These issues collectively underscore the need for an interpretable, quantifiable, and controller-oriented theoretical framework for optimizer parameter design.

Second-order system optimizers like SGD-M and NAG mitigate slow convergence and local optima issues in Gradient Descent (GD) through momentum. Control-based optimizers, such as An et al. [10], applied Proportional-Integral-Derivative (PID) controllers to optimizers, drawing a connection between momentum terms and integral control, correction terms and differential control. However, the impact of control parameters remains underexplored. Wang et al. [11] analyzed the influence of PID parameters on optimizer performance based on Ref. [10] and proposed parameter setting strategies. However,

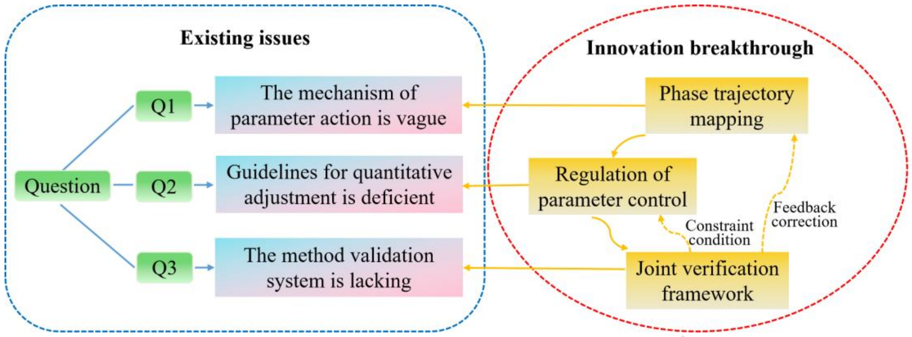


Fig. 1 Existing issues and innovation breakthrough.

this experiment was limited to small-scale datasets. Wu et al. [12] explained the conditions for gradient descent optimizers to achieve stable learning dynamics from the perspective of dynamic differential equations. Yet, there is a lack of systematic research on parameter settings for specific optimizers. Ref. [13, 14] introduced the phase plane method to analyze the effect of optimizer parameters (e.g., learning rate) on dynamic performance. However, the applicability and effectiveness of this method in different types of optimizers and network architectures still need further verification, and there is no coupling analysis of multivariate parameters such as momentum and correction coefficients.

To address the above issues, we propose a dual-mode control-theoretic framework integrating phase trajectory analysis (qualitative) and transfer function modeling (quantitative) to analyze the impact of second-order system optimizer parameters (such as learning rate, momentum coefficient, and correction coefficient) and their combinations on DL performance. This shifts optimizer research from statistical methods to control theory, providing new insights and practical tools for parameter design to improve the interpretability and controllability of model. The contributions (see Fig. 1) are as follows.

(1) Using the phase trajectory analysis method, an explanatory mapping between optimizer parameters and phase trajectory shapes for the first time is established, overcoming empirical tuning limitations and providing theoretical guidance for network parameter tuning.

(2) The transfer function method is used to quantify the regulatory mechanism of parameter coupling on system performance, transforming empirical parameter combinations into mathematical constraints, thereby providing mathematical criteria for parameter selection.

(3) Using the multi-modal modeling framework of phase trajectory and frequency domain response analysis, we establish a mapping between parameter settings, dynamic responses, and model convergence. Through cross-domain validation using both numerical simulations and real datasets, the completeness of the theoretical framework is validated, achieving the transition from theory to practice.

The paper is organized as follows. Section 2 outlines the evolution of optimizers, parameter tuning, and control theory in parameter design. Section 3 analyzes optimizer parameters' impact on system response and stability using transfer functions, phase trajectories, and validates Corollaries 1 to 4 via phase-plane plots, unit step responses. Section 4 tests these insights on MNIST and CIFAR-10 datasets with varying values of $g(\mu)$ and α. Section 5 concludes the paper.

2 Related Work

The rapid development of deep learning is closely related to the improvement of optimizers. Optimizers adjust network parameters to minimize the loss function, and their selection and configuration directly impact the stability and efficiency of the gradient descent process. This subsection describes the current development status of DL optimizers, methods of optimizer parameter tuning, and applications of control theory to optimizer parameter design.

2.1 Deep learning optimizer

Deep learning uses complex neural networks with many parameters. Optimizers are essential for finding optimal solutions in high-dimensional parameter spaces. They are mainly classified into two types: first-order and higher-order. First-order optimizers, like SGD [15], AdaGrad [16], RMSProp [17], and Adam [18], use gradient information of the loss function to update parameters. Higher-order optimizers, such as Newton's method, utilize second-order derivatives for parameter updating. Recently, frontier optimizers like RAdam [19] and AdaBelief [20] have emerged to address stability and convergence issues in Adam. RAdam [19] enhances convergence and stability through dynamic correction. Lookahead [21] improves convergence speed and stability by iteratively updating "external" parameters to guide the "internal" optimizer's search direction. Ranger combines RAdam's dynamic correction with Lookahead's updates for better performance. Additionally, AdamW [22] introduces a weight decay term to improve generalization, while NovoGrad [23] merges momentum and adaptive learning rates to enhance optimization efficiency and stability.

2.2 Optimizer parameter tuning

In deep learning, optimizer parameter tuning is one of the key steps to improve model performance and training efficiency. These parameters include learning rate, momentum factor and decay rate, etc., which directly affect the convergence speed and stability. The main methods for optimizer parameter tuning include: empirical and experimental methods, automated tuning frameworks, and learning rate scheduling strategies. Empirical and experimental based methods, such as grid search, random search and manual tuning, are suitable for small-scale parameter spaces and limited computational resources [24]. Modern automated hyperparameter tuning frameworks (e.g., Hyperopt, Optuna, Ray Tune, etc.) combine methods such as Bayesian optimization, genetic algorithms, and stochastic search to automate the selection of optimizer parameters [25]. These frameworks are suitable for large-scale parameter searches and complex model architectures to find better-performing parameter settings through an efficient exploration process [26]. Learning rate scheduling strategy, dynamically adjusts the learning rate during the training to balance convergence speed and stability of the model.

2.3 Control theory to optimizer parameter design

Control theory provides a new perspective for optimizer parameter design, by modelling the optimization process as a dynamic system and applying control strategies to adjust the parameters to achieve the stability of the optimization process. Methods based on control theory to study optimizer parameters include frequency domain control theory, feedback control strategy and proportional-integral-derivative control methods. Yang et al. [27] regarded the optimizer as a control system, and analyzed the frequency response characteristics of the gradient descent optimizer by using frequency domain control theory to design a more robust optimizer. Grefenstette et al applied frequency domain control theory applied to the feedback control strategy to dynamically adjust the optimizer parameters to improve the stability and efficiency of training.

3 Parametric Analysis and Damping of SGD-M and NAG

This section analyzes the impact of parameter from a control theory perspective, comparing SGD-M (zero-free) and NAG (zero-included) optimizers through four integrated approaches. First, identifying convergence-critical parameters through characteristic equations of their transfer functions, revealing dynamic properties. Second, the phase trajectory method is employed to qualitatively examine the performance characteristics of the SGD-M and NAG systems. By analyzing different parameter combinations, the damping characteristics of the system are assessed to evaluate their impact on system response and stability. Third, the transfer function method quantitatively analyzes the step response to determine system order and optimize parameters. Finally, qualitative (phase trajectory) and quantitative (step response) analyses are combined to validate the proposed parameter optimization criteria (Corollaries 1–4).

3.1 Time-domain response analysis of SGD-M

To identify the key parameters affecting the convergence characteristics and stability of the SGD-M and to analyze the dynamic characteristics of the SGD-M using the damping properties in classical control theory. The iterative expression of SGD-M is [28]

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^{t} - r\nabla L\left(\boldsymbol{\theta}^{t}\right) - r\sum_{i=1}^{t-1}\alpha^{t-i}\nabla L\left(\boldsymbol{\theta}^{i}\right). \quad (1)$$

where θ^t is network parameters θ at time t, r is the learning rate, α is momentum parameter, and $\nabla L\left(\theta'\right)$ is gradient of θ of the loss function at t.

Definition 1. An automatic control system is a second-order system if its transfer function's denominator is of order 2 [29]. The general differential equation is $a_2\frac{d^2y(t)}{dt^2} + a_1\frac{dy(t)}{dt} + a_0y(t) = b_1\frac{du(t)}{dt} + b_0u(t)$. Here, at a specific moment t, $y(t)$ and $u(t)$ are the output and input signals, respectively, constant coefficients a_i (i = 0, 1, 2) are structural parameter of system, and constant coefficients b_j (j = 0, 1) are structural parameter of the input function. Let $Y(s)$=$L[y(t)]$, $U(s)$=$L[u(t)]$, according to the differential theorem, when the initial condition is 0, the closed-loop transfer function is

$$\phi(s) = \frac{Y(s)}{U(s)} = \frac{b_1 s + b_0}{a_2 s^2 + a_1 s + a_0}. \quad (2)$$

Definition 2. A second-order system without zeros occurs when the numerator of the transfer function in Eq. (2) is a constant [29], i.e., $\phi(s) = \frac{b_0}{a_2\cdot s^2 + a_1\cdot s + a_0}$. If a_1 and a_2 have the same sign, then $\phi(s) = \frac{K_0 \cdot \omega_n^2}{s^2+2\xi\omega_n \cdot s+\omega_n^2}$. The characteristic equation and roots are given by

$$s^2+2\xi\omega_n s+\omega_n^2=0,\ s_{1,2}=-\xi\omega_n \pm \omega_n\sqrt{\xi^2-1}. \quad (3)$$

where $\omega_n = \sqrt{\frac{a_0}{a_2}}$ is the undamped natural oscillation frequency, $\xi = \frac{a_1}{2\sqrt{a_0 a_2}}$ is the damping ratio, and $K_0 = \frac{b_0}{a_0}$ is the amplification coefficient for amplifying the input signal. Ref. [13] gives the SGD-M closed-loop transfer function:

$$\phi_{\mathrm{SGD-M}} = \frac{K_0 r}{g(\mu)s^2+s+K_0 r}. \quad (4)$$

where K_0 is the proportionality coefficient, and $g(\mu)$ is the momentum control parameter, equal to α in the SGD-M iterative format. K_0 is the amplification coefficient of the autotuning amplifier given according to the actual problem [29], which accelerates system response and improves regulation accuracy. Transfer function in Eq. (4) has a denominator of order 2. From Definition 2, SGD-M is a second-order control model without zeros.

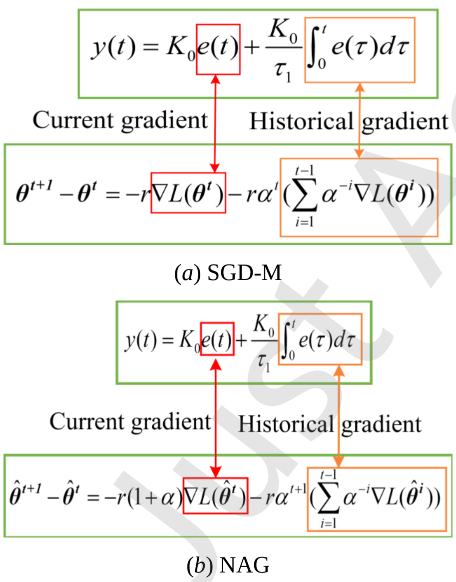


Fig. 2 Linkage of SGD-M and NAG with proportional integrators.

The parameters ξ and ω_n define the dynamic response [29]. The SGD-M process is governed by $\omega_n = \sqrt{\frac{K_0 r}{g(\mu)}}$ and $\xi = \frac{1}{\sqrt{4K_0 r g(\mu)}}$, controlling convergence rate and stability through K_0, r and $g(\mu)$. Fig. 2(a) shows the relationship between the SGD-M transfer function and the iterative expression, where $K_0 = -r$ and $\frac{K_0}{\tau_1} = r\alpha^t$, then $K_0 = -r$ and $\tau_1 = \frac{1}{\alpha^t}$.

The inertial link is defined as $\phi(s) = \frac{1}{g(\mu)s+1}$. The inverse Laplace transform of its response to a unit step signal is $C(s) = \frac{1}{g(\mu)s+1} \cdot \frac{1}{s} = \frac{1}{s} - \frac{1}{s+\frac{1}{g(\mu)}}$, resulting in $c(t) = L^{-1}[C(s)] = \left(1-e^{-\frac{t}{g(\mu)}}\right) \cdot 1(t)$. The inertial link consists of two parts, $c(t) = c_1(t) - c_2(t)$, and the transient part

$$\begin{aligned} c_2(t) = e^{-\frac{t}{g(\mu)}} &= 1+\left(-\frac{t}{g(\mu)}\right)+\frac{\left(-\frac{t}{g(\mu)}\right)^2}{2!} \\ &+\frac{\left(-\frac{t}{g(\mu)}\right)^3}{3!}+\cdots+\frac{\left(-\frac{t}{g(\mu)}\right)^n}{n!}+\mathrm{O}\left(\frac{\left(-\frac{t}{g(\mu)}\right)^{n+1}}{(n+1)!}\right) \quad (5) \\ &\approx 1-\frac{t}{g(\mu)}+\frac{1}{2!}\frac{t^2}{g^2(\mu)}-\frac{1}{3!}\frac{t^3}{g^3(\mu)}+\cdots+\frac{1}{n!}\frac{t^n}{g^n(\mu)} \end{aligned}$$

Some terms are $\sum_{i=1}^{t-1}\alpha^{t-i}\nabla L(\boldsymbol{\theta}^t) = \alpha^0\nabla L(\boldsymbol{\theta}^1) + \alpha^1\nabla L(\boldsymbol{\theta}^2)+\cdots+\alpha^{t-1}\nabla L(\boldsymbol{\theta}^t)$, showing that α corresponds to a gradual decay of the gradient, reducing transient terms as computations increase. This leads to the relationships $\alpha = \frac{1}{n!}\frac{t^n}{g^n(\mu)}$, $g(\mu) = \sqrt[n]{\frac{1}{n!\alpha}}t$. Ultimately, K_0 and $g(\mu)$ in the SGD-M control model correspond to the iterative format with $K_0 = -r$, $g(\mu) = \sqrt[n]{\frac{1}{n!\alpha}}t$, and $K_0 r g(\mu) = -\left(r^2 t\right)\sqrt[n]{\frac{1}{n!\alpha}}$.

Definition 3. The time response is the output variation over time to an input signal [29]. The system is underdamped for $0 < \xi < 1$, critically damped for $\xi = 1$, and overdamped for $\xi > 1$.

Then, the response of the SGD-M system to the initial state is discussed qualitatively using the phase trajectory; that is, considering the case of no input, $u(t) = 0$. This is mainly reflected in the categorical discussion of the effects of different values of the parameter $K_0 r g(\mu)$ on the damping characteristics, convergence rate and stability of the system.

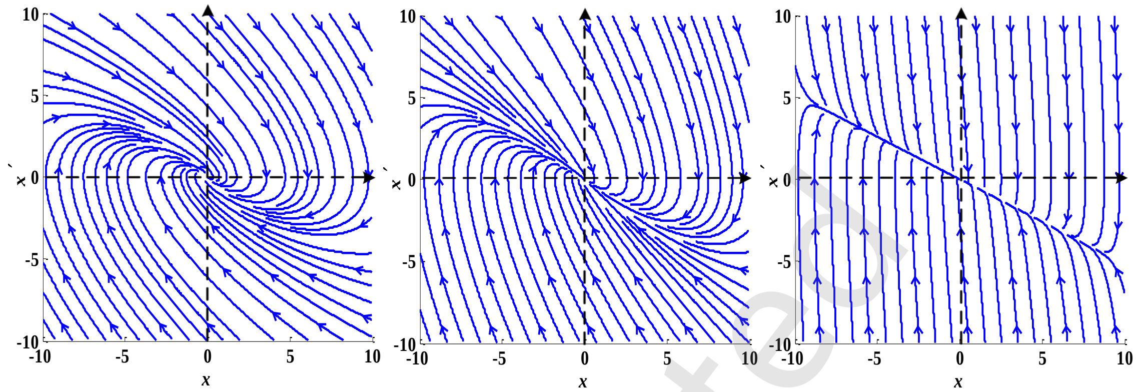
Corollary 1. When $K_0 r g(\mu) > 0.25$, SGD-M behaves as an underdamped system with fast convergence, but may oscillate and is less stable. When $K_0 r g(\mu) = 0.25$, SGD-M is critically damped system with fast convergence and stability, which is ideal for optimization. When $0 < K_0 r g(\mu) < 0.25$, SGD-M is over-damped system with high stability but slow convergence.

Proof. (1) When $K_0 r g(\mu) > 0.25$, the damping ratio $0 < \xi < 1$, SGD-M is an underdamped system, the system converges faster, but due to the presence of oscillations, the system may cross the target value several times. The characteristic equation is $g(\mu)s^2 + s + K_0 r = 0$, and $\sqrt{\xi^2-1} = j\sqrt{1-\xi^2}$, and eigenvalues are $s_1 = -\frac{1}{2g(\mu)} + \frac{j\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)}$, and $s_2 = -\frac{1}{2g(\mu)} - \frac{j\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)}$.

In Table 1, the real part of the eigenvalues of the system state control matrix A determines the stability of the equilibrium point, while the imaginary part of the eigenvalues determines the oscillatory nature of the system [30].The eigenvalues of the SGD-M state matrix are complex with negative real parts, indicating that SGD-M is stable but has weak stability due to oscillations and overshoot during convergence, making

Table 1 Equilibrium types and stability based on eigenvalues of **A** [30].

$s_{1,2} = \sigma + \omega j$	Classification and explanation of s_1, s_2		Equilibrium point	Stability
$s_{1,2}$ are real numbers	$s_1 s_2 > 0$ and $s_1 + s_2 < 0$	$s_{1,2}$ are both negative	stable node	asymptotically stable
$s_{1,2}$ are complex numbers	$s_{1,2} = \sigma + \omega j (\sigma < 0)$	real part is less than 0	stable focus	asymptotically stable



(a) $K_0 r g(\mu)$>0.25, $g(\mu)$=0.93 (b) $K_0 r g(\mu)$=0.25, $g(\mu)$=0.5 (c) 0<$K_0 r g(\mu)$<0.25, $g(\mu)$=0.07

Fig. 3 SGD-M phase trajectory curves with different $g(\mu)$.

it susceptible to fluctuations. The imaginary parts in s_1 and s_2 causes oscillation in the SGD-M system, with the phase trajectory converging to the stable focus $(0, 0)$ in an oscillatory, helical manner [30], showing both oscillation and decay, as shown in Fig. 3(a).

(2) When $K_0 r g(\mu) = 0.25$, then $\xi = 1$, the SGD-M system becomes critically damped, enabling faster convergence to the target value without overshoot. This critical damping state ensures quick stabilization without oscillations, achieving both fast convergence and good stability. The eigenvalues are calculated as $s_1 = s_2 = -\sqrt{\frac{K_0 r}{g(\mu)}}$, the SGD-M is a stable system whose equilibrium point is a stable node. The phase trajectory will converge non-oscillatory to the stable node $(0, 0)$ along the parabola in Fig. 3(b).

(3) When $0 < K_0 r g(\mu) < 0.25$ and $\xi > 1$, SGD-M is an overdamped system. At this time the system tends to steady state slowly because there is no oscillation. Since $\xi > \sqrt{\xi^2 - 1} > 0$, from Eq. (3), we know that $s_1 < 0$, $s_2 < 0$, $s_1 = -\frac{1}{2g(\mu)} + \frac{\sqrt{1-4K_0 r g(\mu)}}{2g(\mu)}$, $s_2 = -\frac{1}{2g(\mu)} - \frac{\sqrt{1-4K_0 r g(\mu)}}{2g(\mu)}$. Since the eigenvalues of the SGD-M state matrix are all negative (Table 1), the system is stable, avoiding oscillations and overshoots. However, this stability comes at the expense of the convergence speed, and its equilibrium is a stable node. The phase trajectory of the SGD-M system converges non-oscillatory to the stable node $(0, 0)$ along the parabola, as depicted in Fig. 3(c).

Next, the effect of $g(\mu)$ on the performance and its stability is investigated by means of $x - x'$ phase trajectory curves. By setting various initial conditions, where x represents position and x' denotes velocity, multiple phase trajectories are generated in the $x - x'$ plane. The differential equation for the motion is $g(\mu)s^2 + s + K_0 r = 0$. Different initial conditions yield distinct x and x' solutions, creating various curves in the phase plane (Fig. 3). Each trajectory reflects the system's kinematic state, with points indicating the system's status at specific times, while the trajectories illustrate how the state evolves over time, with arrows showing the direction of change as time progresses.

In Fig. 3, K_0 and r are fixed to K_0=50 and r=0.01, while $g(\mu)$ is taken to be 0.93, 0.5, and 0.07, respectively, which corresponds to an underdamped (Fig. 3(a)), critically damped (Fig. 3(b)), and overdamped (Fig. 3(c)). In the underdamped case (Fig. 3(a)), the phase trajectory consists of converging logarithmic spirals that oscillate around the target value, indicating rapid convergence with some overshooting. The critically damped case (Fig. 3(b)) shows a parabolic trajectory that converges smoothly to the target without oscillation, reflecting fast convergence. In the overdamped scenario (Fig. 3(c)), the curve is smoother and converges slowly to the equilibrium, indicating the slowest convergence. Different values of $g(\mu)$ result in varying damping states, affecting convergence speed and stability, as predicted in Corollary 1.

Lemma 1. The auxiliary angle formula $a \sin\alpha + b \cos\alpha = \sqrt{a^2 + b^2} \cdot \sin(a + \varphi)$, and $\tan\varphi = \frac{b}{a}$.

Proof. Since $\tan\varphi = \frac{b}{a}$, $\sin\varphi = \frac{b}{\sqrt{a^2+b^2}}$, and $\cos\varphi = \frac{a}{\sqrt{a^2+b^2}}$, thus $a \sin\alpha + b \cos\alpha = \sqrt{a^2 + b^2} \cdot \left(\frac{a \cdot \sin\alpha}{\sqrt{a^2+b^2}} + \frac{b \cdot \cos\alpha}{\sqrt{a^2+b^2}}\right) = \sqrt{a^2 + b^2} \cdot (\cos\varphi \cdot \sin\alpha + \sin\varphi \cdot \cos\alpha) = \sqrt{a^2 + b^2} \cdot \sin(\alpha + \varphi)$. □

The unit step response is the response of a system to an input that changes abruptly from zero to a constant value. In SGD-M, it reveals the speed and stability of the system's response to changes in the loss function. Next, we use a transfer function approach to quantitatively analyze the dynamic properties exhibited by the unit step response of the SGD-M under different circumstances (when $K_0rg(\mu)$ is greater than, equal to, or less than 0.25), which affects the parameter updating speed and stability in deep learning training.

Corollary 2. When $K_0rg(\mu) > 0.25$, the unit step response of SGD-M shows an oscillatory decay curve. This indicates that in deep learning training, the parameter update speed is faster but easy to cause larger fluctuations. When $K_0rg(\mu) = 0.25$ or $0 < K_0rg(\mu) < 0.25$, the unit step response of SGD-M shows a monotonically increasing curve. This indicates that under the critically damped system, the response is faster and without oscillation, while in the over-damped system, the response is slower but more stable.

Proof. When the parameters K_0, r, and $g(\mu)$ take different values, the time response curve of SGD-M has different shapes.

(1) When $K_0rg(\mu) > 0.25$, the second-order system is essentially an oscillatory link. When the input to the SGD-M system is a unit step function, the Laplace transform of the input is $U(s) = \frac{1}{s}$. Substituting this into $\phi(s) = \frac{Y(s)}{U(s)}$, the Laplace transform of time response of the SGD-M system becomes $Y(s) = \frac{\frac{K_0 r}{g(\mu)}}{s^2 + \frac{s}{g(\mu)} + \frac{K_0 r}{g(\mu)}} \frac{1}{s}$. Let the denominator of $Y(s)$ be equal to 0, and then the three poles of the SGD-M are as follows: s_1=0, $s_2 = -\frac{1}{2g(\mu)} + \frac{j\sqrt{4K_0rg(\mu)-1}}{2g(\mu)}$, and $s_3 = -\frac{1}{2g(\mu)} - \frac{j\sqrt{4K_0rg(\mu)-1}}{2g(\mu)}$. where s_1 comes from the input, and s_2 and s_3 are poles of the transfer function of the system. Assuming that the three undetermined coefficients are A, B, and C, and substituting them into $Y(s)$, we obtain

$$Y(s) = \frac{A}{(s-s_1)} + \frac{B}{(s-s_2)} + \frac{C}{(s-s_3)}. \tag{6}$$

Comparing the numerators on both sides of the equation after generalizing Eq. (6) yields

$$\begin{aligned} \frac{K_0 r}{g(\mu)} =& A(s-s_2)(s-s_3) + B(s-s_1)(s-s_3) \\ & + C(s-s_1)(s-s_2), \end{aligned} \tag{7}$$

Let $s = s_1 = 0$. Then, $\frac{K_0 r}{g(\mu)} = As_2s_3 = A\frac{K_0 r}{g(\mu)}$, and $A = 1$. Let $s = s_2 = -\frac{1}{2g(\mu)} + \frac{\sqrt{4K_0rg(\mu)-1}}{2g(\mu)}j$, then

$$\begin{aligned} \frac{K_0 r}{g(\mu)} &= B\left(-\frac{1}{2g(\mu)} + \frac{j\sqrt{4K_0rg(\mu)-1}}{2g(\mu)}\right)\left(\frac{j\sqrt{4K_0rg(\mu)-1}}{g(\mu)}\right) \\ &= B\left(-\frac{j\sqrt{4K_0rg(\mu)-1}}{2g(\mu)^2} - \frac{4K_0rg(\mu)-1}{2g(\mu)^2}\right) \\ &= B\left(\frac{-2K_0 r}{g(\mu)}\right)\left(\frac{j}{2\sqrt{K_0rg(\mu)}}\sqrt{1-\frac{1}{4K_0rg(\mu)-1}} + 1 - \frac{1}{4K_0rg(\mu)}\right) \end{aligned} \tag{8}$$

Since $\xi = \frac{1}{\sqrt{4K_0\,\mathrm{rg}(\mu)}}$, Eq. (8) is transformed into $1 = -2B\left(\xi\sqrt{1-\xi^2}j + 1 - \xi^2\right)$. This yields

$$\begin{aligned} B &= -\frac{1}{2}\frac{1}{(1-\xi^2)+\xi\sqrt{1-\xi^2}j}\frac{(1-\xi^2)-\xi\sqrt{1-\xi^2}j}{(1-\xi^2)-\xi\sqrt{1-\xi^2}j} \\ &= -\frac{1}{2}\left(1-\frac{j}{\sqrt{4K_0rg(\mu)-1}}\right). \end{aligned} \tag{9}$$

Similarly, let $s = s_3 = -\frac{1}{2g(\mu)} - \frac{\sqrt{4K_0rg(\mu)-1}}{2g(\mu)}j$, then

$$\frac{K_0 r}{g(\mu)} = C\left(\frac{2K_0 r}{g(\mu)}\right)\left(\frac{1}{2\sqrt{K_0rg(\mu)}}\sqrt{1-\frac{1}{4K_0rg(\mu)}j + 1 - \frac{1}{4K_0rg(\mu)}}\right) \tag{10}$$

Therefore, C=$-\frac{1}{2}\left(1+\frac{j}{\sqrt{4K_0rg(\mu)-1}}\right)$. Substituting the solved A, B and C into Eq. (7) yields

$$\begin{aligned} Y(s) =& \frac{1}{(s-s_1)} - \frac{1}{2(s-s_2)}\left(1-\frac{j}{\sqrt{4K_0rg(\mu)-1}}\right) \\ & - \frac{1}{2(s-s_3)}\left(1+\frac{j}{\sqrt{4K_0rg(\mu)-1}}\right). \end{aligned} \tag{11}$$

The Laplace transform is applied to both sides of Eq. (10),

$$\begin{aligned} y(t) =& e^{s_1 t} - \frac{1}{2}\left(1-\frac{j}{\sqrt{4K_0rg(\mu)-1}}\right)e^{s_2 t} \\ & - \frac{1}{2}\left(1+\frac{j}{\sqrt{4K_0rg(\mu)-1}}\right)e^{s_3 t} \\ =& 1 - e^{-\frac{t}{2g(\mu)}}\left[\frac{\left(e^{\frac{jt\sqrt{4K_0rg(\mu)-1}}{2g(\mu)}} + e^{-\frac{jt\sqrt{4K_0rg(\mu)-1}}{2g(\mu)}}\right)}{2}\right. \\ & \left. + \frac{j\left(e^{-\frac{jt\sqrt{4K_0rg(\mu)-1}}{2g(\mu)}} - e^{\frac{jt\sqrt{4K_0rg(\mu)-1}}{2g(\mu)}}\right)}{2\sqrt{4K_0\,\mathrm{rg}(\mu)-1}}\right] \end{aligned} \tag{12}$$

According to Euler's formula $e^{j\alpha} = \cos\alpha + j\sin\alpha$, there are

$$e^{\frac{jt\sqrt{4K_0rg(\mu)-1}}{2g(\mu)}} = \cos\left(\frac{t\sqrt{4K_0rg(\mu)-1}}{2g(\mu)}\right)$$

$$+ j\sin\left(\frac{t\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)}\right), \tag{13}$$

$$\begin{aligned} e^{-\frac{jt\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)}} &= \cos\left(\frac{t\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)}\right) \\ &\quad - j\sin\left(\frac{t\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)}\right). \end{aligned} \tag{14}$$

Thus, we have

$$\begin{aligned} &\frac{1}{2}\left(e^{\frac{jt\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)}} + e^{-\frac{jt\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)}}\right) \\ &= \cos\left(\frac{t\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)}\right), \end{aligned} \tag{15}$$

$$\begin{aligned} &\frac{j}{2}\left(e^{-\frac{jt\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)}} - e^{\frac{jt\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)}}\right) \\ &= \sin\left(\frac{t\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)}\right), \end{aligned} \tag{16}$$

$$\begin{aligned} y(t) =& 1 - e^{-\frac{t}{2g(\mu)}}\left(\cos\frac{t\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)}\right. \\ &\left. + \frac{1}{\sqrt{4K_0 r g(\mu)-1}}\sin\frac{t\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)}\right). \end{aligned} \tag{17}$$

By Lemma 1, we have

$$\begin{aligned} y(t) =& 1 - e^{-\frac{t}{2g(\mu)}}\sqrt{\frac{4K_0 r g(\mu)}{4K_0 r g(\mu)-1}} \\ &\cdot \sin\left(\frac{t\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)} + \varphi\right), \\ &\tan\varphi = \sqrt{4K_0 r g(\mu)-1} \end{aligned} \tag{18}$$

Since $s_1 = 0$, the system output contains a constant, 1 from the system input. $\left(e^{-\frac{t}{2g(\mu)}}\sqrt{\frac{4K_0 r g(\mu)}{4K_0 r g(\mu)-1}}\right)$ decreases with time and tends to zero, but $\sin\left(\frac{t\sqrt{4K_0 r g(\mu)-1}}{2g(\mu)} + \varphi\right)$ is the period of $\frac{4\pi g(\mu)}{\sqrt{4K_0 r g(\mu)-1}}$. The initial phase is a trigonometric function of φ. Eq. (18) shows the unit step response of the SGD-M system as a function of time for $K_0 r g(\mu) > 0.25$, which is an oscillating and decaying curve. This indicates that the system fluctuates briefly in the face of sudden changes, but eventually converges to a steady state. In deep learning training, this means that the model parameters will experience large update fluctuations at the beginning, and these fluctuations will gradually decrease as the training progresses, and the parameters will eventually stabilize near the optimal solution. The momentum term plays a key role in this process by introducing a certain degree of fluctuation while speeding up convergence, thus helping the model to find a better solution on the complex loss surface.

(2) When $K_0 r g(\mu) = 0.25$, the poles are s_1=0, $s_2 = s_3 = -\sqrt{\frac{K_0 r}{\mu}}$, and $Y(s) = \frac{\omega_n^2}{(s^2+2\omega_n s+\omega_n^2)s} = \frac{1}{s} - \frac{1}{s+\omega_n} - \frac{\omega_n}{(s+\omega_n)^2}\left(\omega_n = \sqrt{\frac{K_0 r}{g(\mu)}}\right)$. The Laplace transform is performed on both sides of the equation,

$$\begin{aligned} y(t) &= 1 - e^{t\sqrt{\frac{K_0 r}{g(\mu)}}} - t\sqrt{\frac{K_0 r}{g(\mu)}}e^{-t\sqrt{\frac{K_0 r}{g(\mu)}}} \\ &= 1 - \left(1 + t\sqrt{\frac{K_0 r}{g(\mu)}}\right)e^{-t\sqrt{\frac{K_0 r}{g(\mu)}}} \end{aligned} \tag{19}$$

Eq. (19) shows the unit step response of the SGD-M system in the case of critical damping. $-\left(1+t\sqrt{\frac{K_0 r}{g(\mu)}}\right)e^{-t\sqrt{\frac{K_0 r}{g(\mu)}}}$ decreases with time, the system response exhibits a monotonically increasing curve without overshoot. During the DL training process, this indicates that the model parameter updates can move towards the optimal solution in a smooth and continuous manner, reducing excessive fluctuations and repeated adjustments, thus improving the convergence speed and the stability of parameter updates.

(3) When $0 < K_0 r g(\mu) < 0.25$, the poles are $s_1 = 0$, $s_2 = -\frac{1}{2g(\mu)} + \frac{\sqrt{1-4K_0 r g(\mu)}}{2g(\mu)}$, and $s_3 = -\frac{1}{2g(\mu)} - \frac{\sqrt{1-4K_0 r g(\mu)}}{2g(\mu)}$. Let $s = s_1 = 0$, then $A = 1$. Let $s = s_2 = -\frac{1}{2g(\mu)} + \frac{\sqrt{1-4K_0 r g(\mu)}}{2g(\mu)}$, then $B = \frac{2K_0 r g(\mu)}{(1-4K_0 r g(\mu)) - \sqrt{1-4K_0 r g(\mu)}}$. Let $s = p_3 = -\frac{1}{2g(\mu)} - \frac{\sqrt{1-4K_0 r g(\mu)}}{2g(\mu)}$, then $C = \frac{2K_0 r g(\mu)}{(1-4K_0 r g(\mu)) + \sqrt{1-4K_0 r g(\mu)}}$. Substituting A, B, and C into Eq. (6) yields

$$\begin{aligned} Y(s) =& \frac{1}{(s-s_1)} + \frac{2K_0 r g(\mu)}{\sqrt{1-4K_0 r g(\mu)}} \\ &\left[\frac{1}{(s-s_2)\left(\sqrt{1-4K_0 r g(\mu)}-1\right)} + \right. \\ &\left. \frac{1}{(s-s_3)\left(\sqrt{1-4K_0 r g(\mu)}+1\right)}\right] \end{aligned} \tag{20}$$

Taking the Laplace transform of Eq. (20) produces

$$y(t) = 1 + \frac{2K_0 r g(\mu)}{\sqrt{1-4K_0 r g(\mu)}} \left[\frac{e^{\frac{\sqrt{1-4K_0 r g(\mu)}-1}{2g(\mu)}t}}{\sqrt{1-4K_0 r g(\mu)}-1} + \frac{e^{\frac{-\sqrt{1-4K_0 r g(\mu)}-1}{2g(\mu)}t}}{\sqrt{1-4K_0 r g(\mu)}+1} \right] \tag{21}$$

where 1 comes from system inputs. This property ensures the stability of model parameter updates during deep learning training and helps to reduce large fluctuations. The slower but more stable response of the overdamped system is reflected in the training by ensuring that the model converges to the optimal solution faster by reducing fluctuations and instability during parameter updates. And the $\frac{2K_0 r g(\mu)}{\sqrt{1-4K_0 r g(\mu)}}[\frac{e^{\frac{\sqrt{1-4K_0 r g(\mu)}-1}{2g(\mu)}t}}{\sqrt{1-4K_0 r g(\mu)}-1} + \frac{e^{\frac{-\sqrt{1-4K_0 r g(\mu)}-1}{2g(\mu)}t}}{\sqrt{1-4K_0 r g(\mu)}+1}]$ of Eq. (21) is the sum of two monotonically decaying and zero-converging curves, which ensures that each parameter update is made towards the optimal solution and reduces iterative adjustments, thus improving training efficiency.

The coefficients of the characteristic equation, which is a sufficient condition for the stability of the second-order system, are all positive. Therefore, $K_0 > 0$, $r > 0$, and $0 < g(\mu) < 1$. Corollaries 1 and 2 do not consider the cases $K_0 r g(\mu) = 0$ and $K_0 r g(\mu) < 0$, because the SGD-M system will be unstable.

3.2 Time domain response analysis of the NAG

This subsection aims to identify the key parameters affecting the convergence characteristics and stability of NAG, and analyze its dynamic properties (e.g., convergence, stability, etc.) using damping from classical control theory to guide the problem of applying in engineering.

Definition 4. When the numerator of the transfer function of Eq. (2) is not equal to a constant, the system is a second-order system with zeros [29], i.e., $\phi(s) = \frac{\omega_n^2(\tau s+1)}{s^2+2\xi\omega_n \cdot s+\omega_n^2} = \frac{\omega_n^2(s+z)}{z(s^2+2\xi\omega_n \cdot s+\omega_n^2)} = \frac{K_g(s+z)}{(s+p_1)(s+p_2)}$. Among them, $s = -z = \frac{1}{\tau}$, and $K_g = \frac{\omega_n^2}{z} = \frac{p_1 p_2}{z}$.

Recurring expression is [28] $\hat{\boldsymbol{\theta}}^{t+1} = \hat{\boldsymbol{\theta}}^t - r(1+\alpha)\nabla L\left(\hat{\boldsymbol{\theta}}^t\right) - r\alpha\sum_{i=1}^{t-1}\alpha^{t-i}\nabla L\left(\hat{\boldsymbol{\theta}}^i\right)$. The closed-loop transfer function is [13] $\phi_{\mathrm{NAG}}(s) = \frac{K_0 r g(\mu)s+K_0\alpha r}{g(\mu)s^2+s+K_0 r g(\mu)s+K_0\alpha r}$, where α is the leading correction factor. The denominator is of order 2, making NAG a second-order control model with specific dynamics governed by [13] $\omega_n = \sqrt{\frac{K_0\alpha r}{g(\mu)}}$ and $\xi = \frac{K_0 r g(\mu)+1}{2\sqrt{K_0\alpha r g(\mu)}}$. The convergence speed and smoothness are influenced by K_0, r, $g(\mu)$, and α. Fig. 2(b) shows the relationship is $K_0 = -(1+\alpha)r$, $g(\mu) = \sqrt[n]{\frac{1}{n!\alpha}}t$, and $\frac{K_0 r g(\mu)+1}{2\sqrt{K_0\alpha r g(\mu)}} = \frac{\left[r^2(1+\alpha)t\right]\sqrt[n]{\frac{1}{n!\alpha}}+1}{2r\sqrt{\left[(1+\alpha)\alpha t\sqrt[n]{\frac{1}{n!\alpha}}\right.}}$.

Subsequently, the effects of different parameter values on the damping characteristics, convergence rate and stability of the NAG system are qualitatively analyzed using a phase-trajectory approach.

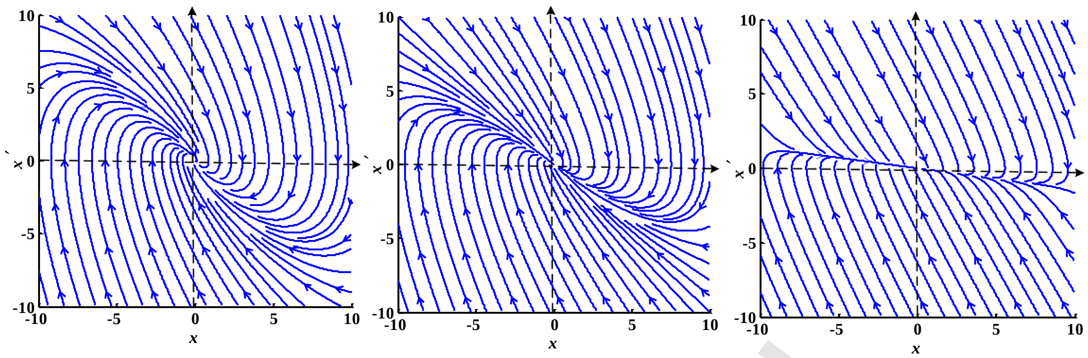
Corollary 3. NAG exhibits greater stability than the SGD-M system when it incorporates a zero point. For $0 < \frac{K_0 r g(\mu)+1}{2\sqrt{K_0\alpha r g(\mu)}} < 1$, NAG operates as an underdamped system, characterized by oscillations and significant fluctuations in model parameter updates, ultimately stabilizing. When $\frac{K_0 r g(\mu)+1}{2\sqrt{K_0\alpha r g(\mu)}} = 1$, it becomes critically damped, resulting in rapid convergence and stability in model parameter updates. For $\frac{K_0 r g(\mu)+1}{2\sqrt{K_0\alpha r g(\mu)}} \geqslant 1$, NAG is overdamped, leading to a slow but stable system response.

Proof. NAG demonstrates greater stability than SGD-M when zeros are added, as this optimizes the system's frequency response and enhances phase margin. This means smoother and more controllable parameter updates during deep learning training. These improvements ensure that the NAG performs more consistently on complex loss surfaces, thereby increasing the speed at which the model converges to an optimal solution.

(1) When $0 < \frac{K_0 r g(\mu)+1}{2\sqrt{K_0\alpha r g(\mu)}} < 1$, the system is underdamped, described by $\mathrm{g}(\mu)s^2 + \mathrm{s} + K_0 r g(\mu)s + K_0\alpha r = 0$, then $s_1 = -\frac{K_0 r g(\mu)+1}{2g(\mu)} + \frac{j\sqrt{4K_0\alpha r g(\mu)-(K_0 r g(\mu)+1)^2}}{2g(\mu)}$, $s_2 = -\frac{K_0 r g(\mu)+1}{2g(\mu)} - \frac{j\sqrt{4K_0\alpha r g(\mu)-(K_0 r g(\mu)+1)^2}}{2g(\mu)}$. s_1 and s_2 are complex with a real part of 1, indicating stability through oscillation and decay. The phase trajectory in Fig. 4(a) converges to the stable focus (0, 0) in a helical manner, leading to large fluctuations in parameter updates that ultimately stabilize.

(2) When $\frac{K_0 r g(\mu)+1}{2\sqrt{K_0\alpha r g(\mu)}} = 1$, the NAG is a critically damped system, then $s_1 = s_2 = -\xi\omega_n = -\frac{K_0 r g(\mu)+1}{2g(\mu)}$. Since s_1 and s_2 are two equal negative real numbers, the NAG is a stable system whose phase trajectory converges to the stable node (0, 0) in a non-positive oscillatory manner, as shown in Fig. 4(b). This reflects that the model parameter updates converge fastest, without oscillations and stable during the deep learning training process compared to the underdamped and overdamped cases.

(3) When $\frac{K_0 r g(\mu)+1}{2\sqrt{K_0\alpha r g(\mu)}} > 1$, the NAG is an overdamped system. Since $\xi > \sqrt{\xi^2-1} > 0$, $s_1 < 0$, $s_2 < 0$, then $s_1 = -\frac{K_0 r g(\mu)+1}{2g(\mu)} + \frac{\sqrt{(K_0 r g(\mu)+1)^2-4K_0\alpha r g(\mu)}}{2g(\mu)}$, $s_2 = -\frac{K_0 r g(\mu)+1}{2g(\mu)} - \frac{\sqrt{(K_0 r g(\mu)+1)^2-4K_0\alpha r g(\mu)}}{2g(\mu)}$. Since s_1 and s_2



(a) $0<\frac{K_0 rg(\mu)+1}{2\sqrt{K_0\alpha rg(\mu)}}<1$, α=1.5 (b) $\frac{K_0 rg(\mu)+1}{2\sqrt{K_0\alpha rg(\mu)}}=1$, α=1 (c) $\frac{K_0 rg(\mu)+1}{2\sqrt{K_0\alpha rg(\mu)}}>1$, α=0.25

Fig. 4 NAG phase trajectory curves with different α values.

are both negative, and the NAG is a stable system whose phase trajectory converges to the steady point node (0, 0) in a non-positive oscillatory manner, as shown in Fig. 4(c). This reflects that during the deep learning training process, the model parameters are updated slowly but are able to converge steadily to the optimal solution due to the absence of oscillations.

The effect of α on the performance and stability of NAG is analyzed using phase trajectory curves with fixed parameters $K_0 = 100$, $g(\mu) = 0.9$, and $r = 0.01$. The differential equation is derived from Fig. 4. When α=1.5 (Fig. 4(a)), the phase trajectory shows converging logarithmic spirals, indicating oscillation and stabilization at (0, 0). This verifies that the model parameter updates for the underdamped system in Corollary 3 are fluctuating but stable. When α=1 (Fig. 4(b)), the trajectory appears as a parabola that quickly converges to (0, 0), verifying fast and stable updates for the critically damped system. When α=0.25, Fig. 4(c) shows that the phase trajectory behaves as a parabola that slowly converges to the stable node (0, 0). This verifies the slower but stable response of the overdamped system in Corollary 3.

The control system is desired to be designed as an underdamped system with a fast time response, then the transfer function is used to quantitatively analyze the dynamic properties exhibited by the unit step response of the NAG system at time.

Corollary 4. When $0 < \frac{K_0 rg(\mu)+1}{2\sqrt{K_0\alpha rg(\mu)}} < 1$, the unit step response of the NAG system is oscillating and decaying curve. This indicates that with an underdamped system, the system responds faster and eventually stabilizes despite experiencing oscillations.

Proof. When the input to the NAG system is a unit step, substituting $U(s) = \frac{1}{s}$ into the transfer function yields the Laplace transform of its system time response as

$$\begin{aligned}\mathrm{Y}(s) &= \frac{K_0 r}{s^2 + (K_0 r + \frac{1}{g(\mu)})s + \frac{K_0\alpha r}{g(\mu)}} \\ &+ \frac{\frac{K_0 r\alpha}{g(\mu)}}{s^2 + (K_0 r + \frac{1}{g(\mu)})s + \frac{K_0\alpha r}{g(\mu)}}\frac{1}{s} \\ &= Y_1(s) + Y_2(s). \end{aligned} \tag{22}$$

Let $\omega_d = \omega_n\sqrt{1-\xi^2}$. Since $\xi = \frac{K_0 rg(\mu)+1}{2\sqrt{K_0\alpha rg(\mu)}}$ and $\omega_n = \sqrt{\frac{K_0\alpha r}{g(\mu)}}$, then

$$\begin{aligned}Y_1(s) &= \frac{K_0 r}{(s+\xi\omega_n - j\omega_d)(s+\xi\omega_n + j\omega_d)}, \\ Y_2(s) &= \frac{\omega_n^2}{(s+\xi\omega_n - j\omega_d)(s+\xi\omega_n + j\omega_d)}\frac{1}{s}. \end{aligned} \tag{23}$$

Assuming that the three coefficients to be determined for $Y_2(s)$ are A_1, A_2 and A_3, factorizing $Y_2(s)$ yields $Y_2(s) = \frac{A_1 s + A_2}{(s+\xi\omega_n - j\omega_d)(s+\xi\omega_n + j\omega_d)} + \frac{A_3}{s}$. Comparing the numerators on both sides of the equation after dividing them by a common denominator gives $\omega_n^2 = (A_1 + A_3)s^2 + (A_2 + 2\xi\omega_n A_3)s + \omega_n^2 A_3$, so $A_1 = -1$, $A_2 = -2\xi\omega_n$, and $A_3 = 1$. Then, there is

$$\begin{aligned}Y_2(s) &= \frac{1}{s} - \frac{s + 2\xi\omega_n}{(s+\xi\omega_n - j\omega_d)\cdot(s+\xi\omega_n + j\omega_d)} \\ &= \frac{1}{s} - \frac{s+\xi\omega_n}{(s+\xi\omega_n)^2 + \omega_d^2} - \\ &\frac{\xi\omega_n}{\omega_d}\cdot\frac{\omega_d}{(s+\xi\omega_n)^2 + \omega_d^2}. \end{aligned} \tag{24}$$

Since $L\left(e^{-at}\sin\omega_n t\right) = \frac{\omega_n^2}{(s+a)^2+\omega_n^2}$ and $L\left(e^{-at}\cos\omega_n t\right) = \frac{s+a}{(s+a)^2+\omega_n^2}$, by Laplace inverse transform we get $y_2(t) = 1 - e^{-\xi\omega_n t}\cos\omega_d t - \frac{\xi\omega_n}{\omega_d}e^{-\xi\omega_n t}\sin\omega_d t = 1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}}\left[\sqrt{1-\xi^2}\cos\omega_d t + \xi\sin\omega_d t\right]$. Since $0 < \xi < 1$, let $\xi = \cos\varphi$, then $\sin\varphi = \sqrt{1-\xi^2}, \tan\varphi = \frac{\sqrt{1-\xi^2}}{\xi}$.

Therefore,

$$\begin{aligned} y_2(t) &= 1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} [\sin\varphi \cos\omega_d t + \cos\varphi \sin\omega_d t] \\ &= 1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \sin\left(\omega_d t + \arctan\frac{\sqrt{1-\xi^2}}{\xi}\right). \end{aligned} \tag{25}$$

The Laplace transform of the NAG system time response $y_2(s)$ is

$$\begin{aligned} y_2(s) =& 1 - e^{-\frac{K_0 r g(\mu) t}{2g(\mu)}} \sqrt{1 - \frac{(K_0 r g(\mu)+1)^2}{4K_0 \alpha r g(\mu)}} \\ &\cdot \sin\left[\frac{t\sqrt{4K_0\alpha r g(\mu) - (K_0 r g(\mu)+1)^2}}{2g(\mu)}\right. \\ &\left. + \arctan\left(\frac{2\sqrt{K_0\alpha r g(\mu)}}{K_0 r g(\mu)+1} - 1\right)\right] \end{aligned} \tag{26}$$

Eq. (26) indicates that the unit step response input is 1, with the term $e^{-\frac{K_0 r g(\mu) t}{2g(\mu)}} \sqrt{1 - \frac{(K_0 r g(\mu)+1)^2}{4K_0 \alpha r g(\mu)}}$ decreasing towards 0 over time, signaling the system's gradual convergence to a steady state. This characteristic indicates that the fluctuation of parameter updates will gradually decrease during the deep learning training process and eventually reach stability. In addition, the response function $\sin[\frac{t\sqrt{4K_0\alpha r g(\mu) - (K_0 r g(\mu)+1)^2}}{2g(\mu)} + \arctan(\frac{2\sqrt{K_0\alpha r g(\mu)}}{K_0 r g(\mu)+1} - 1)]$ is a trigonometric function, and the multiplication of these two parts forms an oscillating and decreasing curve. This oscillating and decreasing indicates that NAG parameter updates will experience significant initial fluctuations, which will gradually diminish over time, leading to stabilization.

Assuming that the two coefficients to be determined for $Y_1(s)$ are A_4 and A_5, factorizing $Y_1(s)$ yields $Y_1(s) = \frac{K_0 r}{s^2 + (K_0 r + \frac{1}{g(\mu)})s + \frac{K_0 \alpha r}{g(\mu)}} = \frac{K_0 r}{\omega_n} \frac{\omega_n}{s^2 + 2\xi\omega_n + \omega_n^2}$. Since $L\left(\frac{\omega_n e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \sin\omega_n t\sqrt{1-\xi^2}\right) = \frac{\omega_n}{s^2+2\xi\omega_n+\omega_n^2}$, the inverse Laplace transform gives $y_1(s) = \frac{K_0 r e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \sin\omega_n t\sqrt{1-\xi^2}$. Thus, the Laplace transform of system time response $y_1(s)$ is

$$\begin{aligned} y_1(t) =& K_0 r \sqrt{1 - \frac{(K_0 r g(\mu)+1)^2}{4K_0 \alpha r g(\mu)}} e^{-\frac{K_0 r g(\mu) t}{2g(\mu)}} \cdot \\ & \sin\left[\frac{t\sqrt{4K_0\alpha r g(\mu) - (K_0 r g(\mu)+1)^2}}{2g(\mu)}\right]. \end{aligned} \tag{27}$$

Eq. (27) shows that the response of the NAG system consists of the multiplication of two terms: the term $K_0 r \sqrt{1 - \frac{(K_0 r g(\mu)+1)^2}{4K_0\alpha r g(\mu)}} e^{-\frac{K_0 r g(\mu) t}{2g(\mu)}}$ decreases with time and tends to 0, and $\sin[\frac{t\sqrt{4K_0\alpha r g(\mu) - (K_0 r g(\mu)+1)^2}}{2g(\mu)}]$ is a delta function. Due to the multiplication of these two parts, the unit step response of the NAG system shows an oscillating and decreasing curve. The Laplace transform can be expressed as the addition of two terms, i.e., $y(s) = y_1(s) + y_2(s)$, where $y_1(s)$ and $y_2(s)$ represent the different response components of the system. the unit step response curve of the NAG system has the characteristics of oscillating and decaying, which indicates that the system responds faster in the underdamped condition, and although there are oscillations in the initial stage, these oscillations will gradually decrease and eventually tend to stabilize. This means that during the deep learning training process, although the parameter update will experience large fluctuations at the initial stage, such fluctuations will gradually decrease and eventually stabilize.

Next, the effect of different values of $g(\mu)$ in the SGD-M system and different values of α in the NAG system on the dynamic performance of the system is investigated by numerical examples.

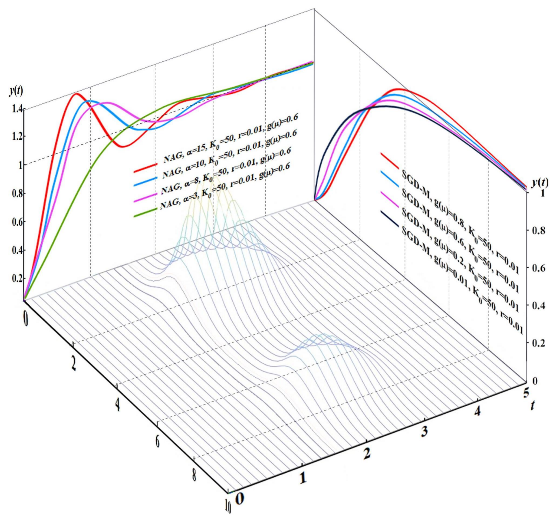


Fig. 5 Unit step response of the SGD-M and NAG system with different parameters.

Fig. 5 shows the different values of $g(\mu)$ and α affect the optimizer's step response curve, influencing response speed and oscillation. In Fig. 5(a), with $g(\mu)$=0.8 and $g(\mu)$=0.6, both K_0=50 and r=0.01, the response exceeds 1, indicating an oscillating and decaying curve, consistent with $K_0 r g(\mu) > 0.25$. Conversely, with $g(\mu)$=0.2 and $g(\mu)$=0.01 (same K_0 and r), the response is monotonic, aligning with $0 < K_0 r g(\mu) < 0.25$ and Corollary 2. In Fig. 5, NAG exhibits underdamping with large oscillations, as predicted by Corollary 4.

Table 2 Dynamic performance indices of the SGD-M and NAG.

Optimizer	SGD-M				NAG			
Parameter	K_0=50, r=0.01				K_0=50, r=0.01, $g(\mu)$=0.6			
	$g(\mu)$=0.8	$g(\mu)$=0.6	$g(\mu)$=0.2	$g(\mu)$=0.01	α=15	α=10	α=8	α=3
Maximum overshoot (%)	**1.73**	0.09	0	0	**36.8**	28.5	23.9	5.56
Rise time (s)	**3.08**	3.22	3.95	4.37	**0.371**	0.483	0.563	1.24
Settling time (s)	**4.66**	5.3	7.18	7.79	**3.13**	3.18	3.59	3.63

The step response metrics for SGD-M and NAG, shown in Fig. 5 and Table 2, indicate that varying $g(\mu)$ (0.6, 0.2, and 0.01) with fixed K_0 and r affects performance significantly. A comparison of parameter values reveals that $g(\mu)$ and α significantly impact the optimizer's dynamic performance, affecting response speed and oscillation. In the SGD-M system, higher $g(\mu)$ values result in faster responses but reduced stability. In the NAG system, increasing α speeds up the response but reduces smoothness.

4 Analysis of modulation coefficients

The experiment tests the validity of Corollaries 1 to 4 from Section 3. By comparing the effects of different $g(\mu)$ values (e.g., $g(\mu)$=0.5, 0.7, and 0.9) and α values (e.g., α=0.5, 0.7, and 0.9) on the recognition of the MNIST and CIFAR-10 datasets. The analysis evaluates their effects on convergence speed and stability, using metrics such as training loss, validation loss, testing loss, training accuracy, validation accuracy, testing accuracy, and convergence time.

4.1 Experimental design

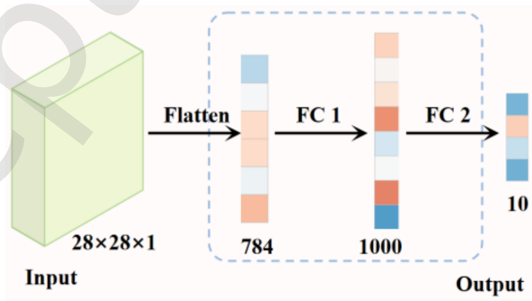
The experiment used the grayscale image dataset (MNIST) [31] and the color image dataset (CIFAR-10) [32], as shown in Table 3. Each sample of the MNIST dataset is a 28×28 pixel greyscale handwritten digit picture. It recognizes the digit picture as the corresponding digit, with digits ranging from 0 to 9. Each sample of the CIFAR-10 dataset is a 32×32 pixel color image, which identifies the color image into the corresponding categories, which are: aircraft, car, bird, cat, deer, dog, frog, horse, boat, and truck. The experiment utilized a fully connected neural network structure in Fig. 6. The network structures for both MNIST and CIFAR-10 dataset experiments have 2 fully connected layers, with the fc1 layer having inputs of $(28 \times 28 \times 1)$ and $(32 \times 32 \times 3)$, respectively, and the outputs of 1000 feature maps; and the fc2 layer having inputs of 1000, outputs of 10, and a total of 1000×10 parameters across both layers.

4.2 Experimental platforms

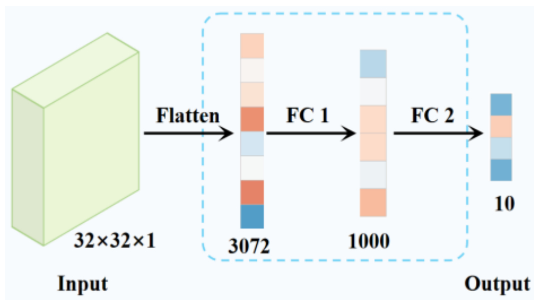
All the algorithms are implemented in Python 3.6 and MATLAB on the Supermicro 7048GR-TR deep learning server. The server processor is intel Xeon E5 2680V4*2 (28 cores and 56 threads, main frequency 2.4G, maximum frequency 3.3G), Supermicro comes with original air-cooled heat dissipation. Samsung RECCDDR4 32G 266 server memory, Samsung 500G high-speed solid-state drive, and supermicro X10DRG-Q workstation motherboard (supporting four GPUs).

Table 3 Introduction to the dataset.

Dataset	MNIST	CIFAR-10
Total samples	60000	50000
Training samples	55000	45000
Validation samples	5000	5000
Test samples	10000	10000



(*a*) MNIST



(*b*) CIFAR-10

Fig. 6 Network structure diagram for experiments on the MNIST and CIFAR-10 dataset.

Table 4 Classification results of datasets with different $g(\mu)$ values.

Datasets:	MNIST			CIFAR-10		
Number of steps:	100			50		
Parameter:	$g(\mu)=0.5$	$g(\mu)=0.7$	$g(\mu)=0.9$	$g(\mu)=0.5$	$g(\mu)=0.7$	$g(\mu)=0.9$
Training accuracy	98.40%	98.71%	98.90%	85.38%	93.49%	93.89%
Training loss	0.07	0.062	0.056	0.448	0.207	0.185
Validation accuracy	98.10%	98.32%	98.46%	68%	72.46%	76.34%
Validation loss	0.078	0.073	0.069	0.916	0.73	0.502
Test accuracy	97.69%	97.91%	98.07%	82.35%	73.92%	74.44%
Test loss	0.086	0.08	0.075	0.766	0.72	0.519
Time	1.51 h	1.14 h	**1.01 h**	2.17 h	1.82 h	**1.65 h**

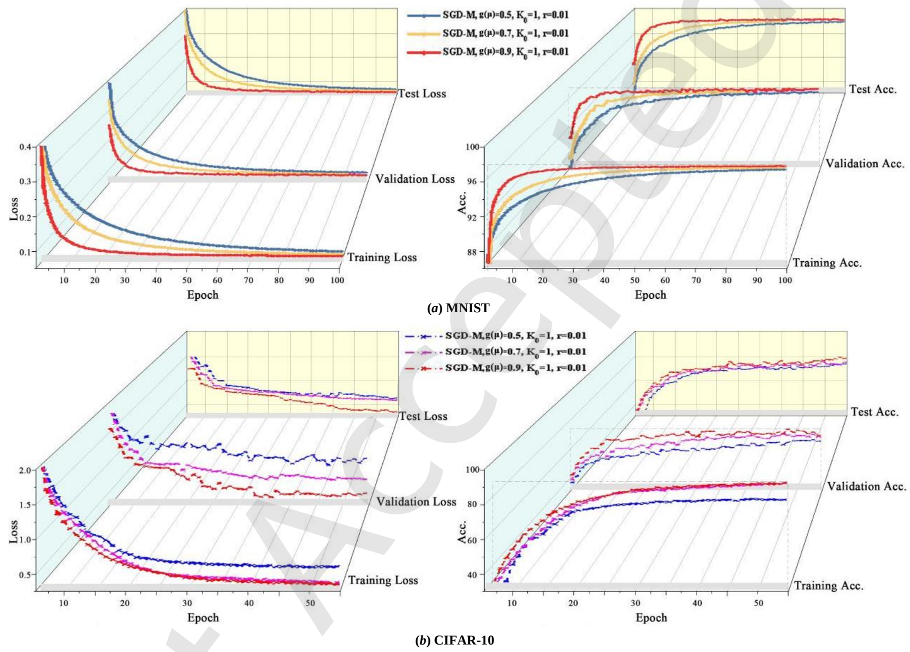


Fig. 7 Effects of different $g(\mu)$ values on MNIST and CIFAR10 classification.

4.3 Experimental results and analysis

4.3.1 *Effect of different $g(\mu)$ values on image recognition performance*

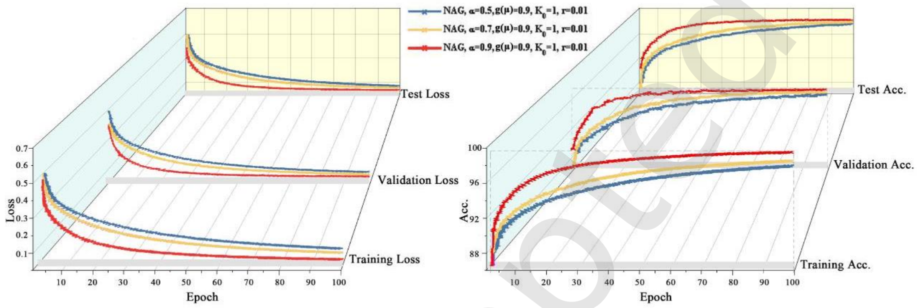
In the SGD-M method, experiments are performed using different values of $g(\mu)$ under the fully connected neural network structure and analyzed on MNIST and CIFAR-10 datasets for classification accuracy, speed of convergence and stability, the results are shown in Table 4 and Fig. 7.

In the SGD-M method, with K_0=1 and r=0.01, the chosen $g(\mu)$ values of 0.5, 0.7, and 0.9 result in $K_0rg(\mu)$ values of 0.005, 0.007, and 0.009, indicating an overdamped system per Corollary 1. The step response is monotonically increasing with no overshooting, characterized by regulation times. For $g(\mu)$ taken as 0.5, 0.7, and 0.9, the corresponding values of ξ are 7.07, 5.98, and 5.27, respectively. As the value of $g(\mu)$ increases, the regulation time of the SGD-M system decreases, which indicates a better rapidity of the system, which is consistent with Corollary 1. In addition, when $g(\mu)$ is taken as 0.5, 0.7 and 0.9, the solutions of its corresponding characteristic equations are $s_1 = -0.01$ and $s_2 = -1.99$, –0.01 and –1.41, –0.01 and –1.11. The characteristic roots of the system are negative real numbers for $g(\mu)$ values of 0.5, 0.7, and 0.9, confirming the monotonic stability of system. The unit step response remains stable and oscillation-free. This demonstrates consistent stability in the system's dynamic response across different values of $g(\mu)$, consistent with Corollary 2.

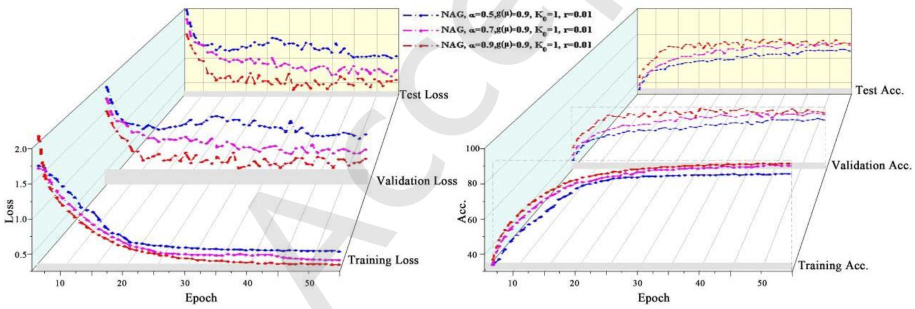
Table 4 shows that the shortest training times for the MNIST and CIFAR-10 datasets are 1.01 h and 1.65 h, respectively, when $g(\mu)$=0.9. This indicates that a larger value of $g(\mu)$ in the range of $0 < K_0rg(\mu) < 0.25$ significantly improves the rapidity of the system, while maintaining a high stability. According to Corollary 1, the SGD-M system behaves as an

Table 5 Classification results of datasets with different α values.

Datasets:	MNIST			CIFAR-10		
Number of steps:	100			50		
Parameter:	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.9$
Training accuracy	98.04%	98.71%	99.83%	86.58%	93.73%	93.85%
Training loss	0.076	0.052	0.015	0.413	0.201	0.182
Validation accuracy	97.90%	98.10%	98.44%	67.82%	73.22%	76.56%
Validation loss	0.082	0.072	0.06	1.158	0.932	0.717
Test accuracy	97.41%	97.73%	98.20%	66.78%	72.35%	75.31%
Test loss	0.092	0.077	0.059	1.206	0.943	0.747
Time	2.26 h	1.74 h	**1.51 h**	3.91 h	2.26 h	**1.82 h**



(a) MNIST



(b) CIFAR-10

Fig. 8 Effects of different α values on MNIST and CIFAR10 classification.

overdamped system with high stability when $0 < K_0 r g(\mu) < 0.25$. Experimental results support that larger $g(\mu)$ values improve speed and stability, corroborated by Fig. 7, which illustrates lower losses and higher accuracy. Corollary 2 further clarifies that the SGD-M system's unit step response in an overdamped state is a stable, monotonic curve. For the same values of K_0 and r, losses are lower and accuracy higher for $g(\mu)$=0.9 compared to $g(\mu)$=0.5 and $g(\mu)$=0.7. Thus, increasing $g(\mu)$ accelerates the system, reduces losses, and improves accuracy, leading to faster convergence without sacrificing stability, align with the theoretical predictions of Corollaries 1 and 2.

4.4 Effect of different α-values on image recognition performance

This experiment investigates the damping features of the NAG system with varying values of α (0.5, 0.7, and 0.9) and their effects on response. With fixed parameters (K_0=1, r=0.01, $g(\mu)$=0.9), the damping ratio ξ is calculated, showing that higher ξ values result in shorter regulation times and improved system responsiveness. The system behaves as overdamped ($\xi > 1$), leading to slower yet stable responses, confirmed by negative characteristic roots, ensuring monotonic stability. The impact of different parameters α on dataset classification recognition is further analyzed in Table 5 and Fig. 8. Corollary 3 indicates that the NAG system's stability and response vary with damping states. Table 5 shows that α=0.9 results in the shortest training times for the MNIST (1.51 h) and CIFAR-10 (1.82 h) datasets, suggesting that higher α values reduce conditioning time and enhance response speed while maintaining stability. This aligns with Corollary 3, where a larger α value improves the response speed while keeping the system stable.

Fig. 8 shows that for the NAG-processed dataset, the model with α=0.9 outperforms those with α=0.5 and α=0.7 in both accuracy and speed of convergence, suggesting that increasing the value of α improves the accuracy and speed of the model identification, while keeping K_0, r, and $g(\mu)$ constant. According to Corollary 4, the unit step response is an oscillating and decaying in the underdamped state, whereas in the overdamped state, the unit step response is a monotonically increasing curve. This indicates that in the overdamped state, the system response is stable and without oscillation. Thus, as the value of α increases, the NAG system shows better fastness and stability in the overdamped state, with reduced loss during training and validation, improved accuracy, and faster convergence, consistent with Corollaries 3 and 4.

5 Conclusion

This paper addresses the weak interpretability and limited generalization of optimizer parameter settings. It proposes critical conditions for SGD-M and NAG parameters, combining transfer function and phase trajectory analysis. By mapping optimizer differential equations to phase plane projections and analyzing the initial response, the interpretability of parameter selection is improved. Transfer function modeling highlights the impact of parameter combinations on optimizer dynamics. Through experiments on MNIST/CIFAR-10, we show that the values of $g(\mu)$ and α significantly affect the optimizer's performance, stability, and time-domain characteristics. A univariate explanation method for parameter handling yields results consistent with multivariate analysis, offering new theoretical insights for optimization algorithms and practical applications like image analysis. However, the univariate method needs further development in multivariate collaborative regulation. Future work will focus on multi-objective optimization and exploring dynamic coupling effects in various optimizers using meta-learning strategies.

Acknowledgment

This work was supported by National Natural Science Foundation of China (Grant Nos. 62566020 and 62262014), the Hainan Provincial Natural Science Foundation of China (Grant Nos. 625QN269 and 825CXTD608), Hainan University Project (Grant Nos. XJ2500001761), and entrusted technology projects by enterprises and institutions (Grant No. HD-KYH-2024275).

Reference

[1] X. K. Wang, L. T. Yang, D. Huo, L. Ren, and M. J. Deen, A Tensor-Train-Based P2 Blockchain for Internet of Things Services, *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19394–19405, 2024.

[2] Y. Yu, M. Li, L. Liu, Y. Li, and J. Wang, Clinical big data and deep learning: Applications, challenges, and future outlooks, *Big Data Min. Anal.*, vol. 2, no. 4, pp. 288–305, 2019.

[3] N. Firdaus and Z. Raza, Federated learning based multi-head attention framework for medical image classification, *Concurr. Comput. Pract. Exp.*, vol. 36, no. 27, pp. e8280, 2024.

[4] I. Nti, J. Quarcoo, J. Aning, and G. Fosu, A mini-review of machine learning in big data analytics: Applications, challenges, and prospects, *Big Data Min. Anal.*, vol. 5, no. 2, pp. 81–97, 2022.

[5] I. Goodfellow, Y. Bengio and A. Courville, Deep learning, (in Chinese), Beijing: Posts & Telecom Press, 2017.

[6] X. K. Wang, L. T. Yang, L. Ren, Y. H. Wang, and M. J. Deen, A tensor-based computing and optimization model for intelligent edge services, *IEEE Netw.*, vol. 36, no. 1, pp. 40–44, 2022.

[7] S. M. Melo, J. C. Carver, P. S. L. Souza, and S. R. S. Souza, Empirical research on concurrent software testing: A systematic mapping study, *Inf. Softw. Technol.*, vol. 105, pp. 226–251, 2019.

[8] A. K. Kar and Y. K. Dwivedi, Theory building with big data-driven research–moving away from the "What" towards the "Why", *Int. J. Inf. Manag.*, vol. 54, no. 3, pp. 102205, 2020.

[9] L. Yang and A. Shami, On hyperparameter optimization of machine learning algorithms: Theory and practice, *Neurocomputing*, vol. 415, pp. 295–316, 2020.

[10] W. An, H. Wang, Q. Sun, J. Xu, Q. Dai, and L. Zhang, A PID controller approach for stochastic optimization of deep networks, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, USA, 2018, pp. 8522–8531.

[11] H. Wang, Y. Luo, W. An, Q. Sun, J. Xu, and L. Zhang, PID controller-based stochastic optimization acceleration for deep neural networks, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5079–5091, 2020.

[12] W. Wu, X. Y. Jing, W. C. Du and G. L. Chen, Learning dynamics of gradient descent optimization in deep neural networks, *Sci. China Inf. Sci.*, vol. 64, no. 5, pp. 150102, 2021.

[13] B. Y. Yao, G. Q. Li, and W. Wu, State space representation and phase analysis of gradient descent optimizers, *Sci. China Inf. Sci.*, vol. 66, no. 4, pp. 142102, 2023.

[14] B. Y. Yao, Q. C. Zhang, R. N. Feng, and X. K. Wang, System response curve based first-order optimization algorithms for cyber-physical-social intelligence, *Concurr. Comput. Pract. Exp.*, vol. 36, no. 21, pp. e8197, 2024.

[15] S. Ruder, An overview of gradient descent optimization algorithms, arxiv preprint arXiv:1609.04747, 2017.

[16] J. Duchi, E. Hazan, and Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011.

[17] T. Tieleman, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, *COURSERA: Neural Networks for Machine Learning*, vol. 4, no. 2, pp. 26–31, 2012.

[18] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, in *Proc. Int. Conf. Learn. Represent.*, San Diego, USA, 2015, pp. 1–15.

[19] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, On the variance of the adaptive learning rate and beyond, in *Proc. Int. Conf. Learn. Represent.*, Virtual, 2020, pp. 1–14.

[20] J. Zhuang, T. Tang, Y. Ding, S. C. Tatikonda, N. Dvornek, X. Papademetris, and J. Duncan, AdaBelief optimizer: adapting stepsizes by the belief in observed gradients, in *Ann. Conf. Neural Inf. Process. Syst.*, Virtual, 2020, pp. 18795–18806.

[21] M. Zhang, J. Lucas, J. Ba and G. E. Hinton, Lookahead optimizer: k steps forward, 1 step back, in *Ann. Conf. Neural Inf. Process. Syst.*, Vancouver, Canada, 2019, pp. 9593–9604.

[22] I. Loshchilov and F. Hutter, Decoupled weight decay regularization, in *Proc. Int. Conf. Learn. Represent.*, New Orleans, USA, 2019, pp. 1–19.

[23] B. Ginsburg, P. Castonguay, O. Hrinchuk, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, H. Nguyen, Y. Zhang, and J. M. Cohen, Stochastic gradient methods with layer-wise adaptive moments for training of deep networks, arxiv preprint arxiv: 1905.11286, 2019, pp. 1–7.

[24] J. Bergstra and Y. Bengio, Random search for hyper-parameter optimization, *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.

[25] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, Hyperband: A novel bandit-based approach to hyperparameter optimization, *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 1–52, 2018.

[26] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, Optuna: a next-generation hyperparameter optimization framework, in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, New York, USA, 2019, pp. 2623–2631.

[27] X. Yang, X. Zheng, and H. Gao, SGD-based adaptive NN control design for uncertain nonlinear systems, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 5071–5083, 2018.

[28] H. Liu, J. Hu, and Y. Li, Optimization: modeling, algorithms, and theory, (in Chinese), Beijing: Higher Education Press, 2020.

[29] G. Gao and W. Yu, Principle of automatic control , (in Chinese), Guangzhou: South China University of Technology Press, 2013.

[30] T. Wang, The beauty of control (volume 1)–control theory from transfer function to state space, (in Chinese), Beijing: Tsinghua University Press, 2022.

[31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition. *Proc. IEEE*, vol. 86, pp. 2278–2324, 1998.

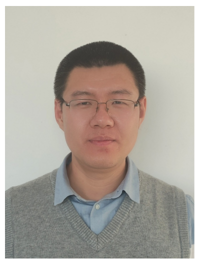
[32] A. Krizhevsky and G. Hinton, Learning multiple layers of features from tiny images, Technical Report, University of Toronto, 2009.

Author biography

Biyuan Yao is an associate researcher with School of Computer Science and Technology, Hainan University, China. She obtained her Ph.D. degree from South China University of Technology, China in 2023. Her research interests include cyber-physical-social systems, data analysis, and optimization computation.

Guiqing Li is a professor with School of Computer Science and Engineering, South China University of Technology, China. He obtained his Ph.D. degree from the University of Chinese Academy of Sciences, China in 2001. His research interests include digital and dynamic geometry processing in computer graphics.

Xiaokang Wang is a professor with School of Computer Science and Technology, Hainan University, China. He earned his Ph.D. degree from Huazhong University of Science and Technology, China in 2017. His research interests include parallel and distributed computing, big data, and tensor computing.



Qingchen Zhang is a Professor with School of Computer Science and Technology, Hainan University, China. He completed his Ph.D. at Dalian University of Technology, China in 2015. His research interests include machine learning, medical big data, blockchain.

Yongwei Nie is an associate professor with School of Computer Science and Engineering, South China University of Technology, China. He obtained his Ph.D. from Wuhan University, China in 2015. His research interests include computer graphics and computer vision.

Jing Chen completed his master's degree at the School of Computer Science and Technology, Hainan University, graduating, China in 2024. His research interests include artificial intelligence and computer vision.