Minimax Theory and its Applications

Volume 11 (2026), No. 3s, 178 to 191

Iterative Numerical Optimization for Stable Training of Nonlinear Deep Learning Models

Dr. Deepa V[1], Dr. M. Vamsikrishna[2], S Krishna Sakalabattula[3], K Antony Kumar[4], Mareeswari G[5], Dr. N. S. R. Phanindra Kumar[6]

[1]Assistant Professor, Department of CSE Sathyabama Institute of Science and Technology Chennai, Tamil Nadu, India - 600 119, Email: deepa.v.cse@sathyabama.ac.in, vdeepaengg@gmail.com, orcid: https://orcid.org/0000-0001-8028-5429

[2]Professor, Department of AIML, School of Computing, Aditya University, Surampalem, Kakinada,AP,India Email: vkmangalampalli@gmail.com, ORCID: https://orcid.org/0000-0001-5285-9990

[3]Department of CSE Ramachandra College Of Engineering Eluru – 534007 India Email:sskrishna369@gmail.com, Orcid: 0009-0008-4172-313X

[4]Associate Professor Department of Computer Science and Engineering vel Tech Rangarajan Dr Sagunthala R&D Institute of Science and Technology, Chennai, India, Email: antonykmr32@gmail.com Orchid Id: 0000-0001-7409-4220

[5]Assistant Professor, Department of Information Technology, Ramco Institute of Technology Tamilnadu,India , Email: gmarees92@gmail.com, orcid: https://orcid.org/0009-0003-3555-7087

[6]Associate Professor, Department of CSE, Aditya Institute of Technology and Management (AITAM),K.Kotturu, Tekkali, Srikakulam Andhra Pradesh 532201, Email: phanindra.nsr@gmail.com ORCID: 0000-0001-5432-9686

Abstract

The training of deep neural networks can be seen as a high-dimensional nonlinear and nonconvex optimization problem, where issues such as numerical instability, ill-conditioned problem formulation, oscillatory behavior of parameter evolution, and sensitivity to learning rate choice can negatively impact convergence properties. While gradient-based methods of optimization play a vital role in deep learning techniques, traditional approaches to their implementation focus more on the algorithm than on stability considerations. This research presents a state-dependent dynamical systems formulation for achieving robust iterative optimization of non-linear deep learning networks. The method involves extending the classical notion of gradient flow through the introduction of a state-dependent positive definite stabilizing operator that influences the optimization path depending on the geometry of the objective function locally. A Lyapunov function derived from the training objective is used to prove stability in a continuous-time setting. Under common conditions of smoothness, boundedness, and positive definiteness, the dynamics will be proven to produce non-increasing trajectories towards the stationary set. A discrete iterative optimization algorithm is then formulated based on numerical time discretization, along with a sufficient condition for the monotone decrease of the objective function. In this way, the formulation forms a link between nonlinear dynamic systems, Lyapunov stability theory, and iterative optimization of deep learning algorithms. Numerical verification using nonlinear test functions and deep learning training tasks is suggested to examine convergence properties, gradient stability, and computational efficiency. The research suggests a mathematically justified approach to introducing state-dependent stabilization to nonconvex optimization problems.

Keywords: Nonconvex Optimization, Deep Learning, Dynamical Systems,Lyapunov Stability, Gradient Flow, Numerical Optimization, Neural Networks.

1. Introduction

The deep learning paradigm has proven quite successful in problems like classification, prediction, computer vision, and other similar computational problems. The performance of deep neural network models relies heavily on the optimization algorithms used to solve the minimization problem of high-dimensional and nonlinear loss functions (LeCun et al., 2015; Goodfellow et al., 2016). Deep models are trained by minimizing loss functions that have a nonconvex objective function space, which consists of local optima, saddle points, and high-curvature regions.

Stochastic Gradient Descent and other similar approaches remain at the very core of machine learning via Neural Networks. SGD uses the approximation of gradients via mini-batches to achieve computationally

efficient optimization (Bottou, 2010). Adaptive SGB utilizes the moment estimates for updating weights based on the first- and second-order gradients (Kingma & Ba, 2015); another adaptive approach, AdamW, introduces a decoupled weight decay regularization technique (Loshchilov & Hutter, 2019). Nevertheless, in spite of their computational efficiency, all these techniques are continuously dependent upon hyperparameterchoice and local changes in the optimization surface.

Numerical optimization in its classical setting offers other mathematical views by means of variable metric approaches, information about curvature, and iterative stability (Bertsekas, 2016; Nocedal & Wright, 2006). At the same time, optimization algorithms can be considered as dynamical systems with iterations that approximate some process of continuous time (Orvieto & Lucchi, 2019). Such an approach allows using the theory of differential equations and nonlinear stability to study optimization trajectories. The recent advancements of neural ordinary differential equations and stable deep network architectures illustrate another example of the use of continuous-time mathematics in deep learning (Chen et al., 2018; Haber & Ruthotto, 2017).

Lyapunov stability theory provides a framework for designing and analyzing nonlinear dynamical systems with energy-based functions (Khalil, 2002; LaSalle, 1976) and has also been used in the context of optimization dynamics and network stability (Feijer & Paganini, 2010; Moucer et al., 2023). However, the design of optimization dynamics with explicit “state-dependent stabilization” has not been sufficiently developed.

This paper suggests a “state-dependent gradient flow” to better optimize nonlinear deep learning. One important feature is the addition of a positively definite stabilizing operator. With it, the gradient flow changes with the current state of the optimization, which improves learning stability. First, a theoretical study with Lyapunov is performed, and then a discretized, iterative optimization method is devised. In the end, the paper’s goal is to build mathematical bridges between a state-dependent stabilization-based viewpoint for the local dynamics and nonlinear dynamics, and then further to the convergence of a numerical optimization method.

2. Related Work

Optimization is important in deep learning because training neural networks is really a matter of minimizing very complicated functions that are nonlinear (and usually non-convex) in many dimensions. Problems of this kind often have saddle points, and in addition, the curvature—that is, the "bendiness"—is different for different directions (Goodfellow et al., 2016; LeCun et al., 2015). In Stochastic gradient descent (SGD), it’s efficient because we can compute it in batches (Bottou, 2010), while in Adam and AdamW we try to make learning stable and easy to use by combining decoupled regularization and adaptive moment estimation (Kingma & Ba, 2015; Loschilov & Hutter, 2019).

Classical numerical optimization also provides methods for nonlinear problems using curvature or variable-metric-type methods (Bertsekas, 2016; Nocedal & Wright, 2006).But as you know, by today, there has also been a lot of progress on how to analyze optimization from the perspective of active systems. How can we combine iterative algorithms with the gradient flow in continuous time to study convergence and stability? (Orvieto & Lucchi, 2019; Rosca et al., 2023) Moreover, the application of Neural ODEs (Neural Ordinary Differential Equations) and work on more stable networks is quite common, showing that differential equation theory is also of interest to the field of deep learning. (Chen et al., 2018; Haber & Ruthotto, 2017).

An energy-based analysis of nonlinear dynamic systems is well established via Lyapunov theory (Khalil, 2002; LaSalle, 1976). In recent years, Lyapunov theory has also been used to study continuous-time optimization (or gradient) systems (Feijer & Paganini, 2010; Moucer et al., 2023). But most work is focused on the analysis of standard gradient flow dynamics, not on the design of an explicit state-dependent stabilization. We solve this problem for our positive-definite stabilization operator by an explicit state-dependent gradient-flow model. The formulation interlinks the optimization local geometry, nonlinear stability theory, and numerical discretization and thereby offers a mathematically rigorous alternative to algorithmic adaptive optimization approaches.

3. Mathematical Preliminaries

Notations and assumptions related to the formulation of the state-dependent optimization dynamics will be discussed in this section. Consider the following nonlinear optimization problem defined in Eq. (1).

$$\min_{\theta \in \mathbb{R}^n} F(\theta) \quad (1)$$

where θdenotes the trainable parameter vector and $F: \mathbb{R}^n \to \mathbb{R}$is a continuously differentiable objective function. In deep learning, Ftypically represents an empirical loss function constructed from a neural network model and

training data. The objective determines parameter values that minimize it.Fwhile maintaining stable iterative behavior.

Nonlinear optimization landscapes may contain local minima, saddle points and regions with varying geometric properties. Consequently, optimization behavior depends not only on gradient magnitude but also on local objective geometry. The conventional continuous-time representation of gradient-based optimization is given in Eq. (2)

$$\dot{\theta}(t) = -\nabla F\big(\theta(t)\big) \quad (2)$$

where $\nabla F(\theta)$is the objective gradient. Equilibrium points satisfy $\nabla F(\theta) = 0$and correspond to stationary points of the optimization problem.

The proposed framework generalizesthe classical formulation through a state-dependent stabilization operator. The analysis assumes that $F$is continuously differentiable and bounded below by a finite constant $F_{\inf}$, such that $F(\theta) \geq F_{\inf}$. The gradient is assumed to be locally Lipschitz continuous, with global Lipschitz continuity imposed for the discrete stability analysis (Nocedal & Wright, 2006). Furthermore, the proposed stabilization operator is assumed to be symmetric and positive definite, ensuring that the modified dynamics remain associated with the negative gradient direction.

The stability analysis is based on Lyapunov theory. A stationary point θ^*satisfies $\nabla F(\theta^*) = 0$Although such points need not represent global minima in nonconvex problems, a nonnegative Lyapunov function that decreases along optimization trajectories provides an energy-dissipation property and supports convergence analysis using invariance principles (Khalil, 2002; LaSalle, 1976). These concepts provide the mathematical foundation for the proposed state-dependent gradient-flow model.

4. Proposed State-Dependent Dynamical System

4.1 Motivation for State-Dependent Stabilization

The traditional gradient optimization technique uses the local gradient according to a fixed Euclidean metric. Though successful, the method lacks consideration of differences in the local geometry of the nonlinear cost landscape. Deep learning models might have different levels of curvature and sensitivity in some areas, in which fixed step sizes could lead to oscillation and inefficient convergence. The existing techniques try to overcome this problem through a discrete updating scale method. In this paper, a continuous-time nonlinear dynamic system with state-dependent stabilization is first established, followed by the development of a gradient transformation to control parameter speed based on local optimization information.

4.2 Proposed State-Dependent Gradient Flow

The proposed optimization dynamics are defined as in Eq. (3)

$$\boxed{\dot{\theta}(t) = -A\big(\theta(t)\big)\nabla F\big(\theta(t)\big)} \quad (3)$$

where $A(\theta)$is a state-dependent stabilization operator. The operator modifies the conventional gradient vector before it determines the parameter velocity.

The proposed formulation generalizes classical gradient flow. If$A(\theta) = I$where $I$is the identity matrix, the system reduces to conventional gradient flow. Therefore, the proposed model can be interpreted as a generalized gradient system with a variable metric that depends on the current optimization state.

The introduction of $A(\theta)$allows the optimization dynamics to adapt to local characteristics of the parameter space. Rather than applying an identical transformation throughout the entire optimization process, the proposed system modifies the effective optimization velocity as the state changes. This provides the mathematical basis for state-dependent stabilization.

4.3 State-Dependent Stabilization Operator

The stabilization operator is defined as in Eq. (4)

$$\boxed{A(\theta) = [I + \lambda C(\theta)]^{-1}} \quad (4)$$

where $\lambda > 0$is a stabilization parameter and $C(\theta)$is a symmetric positive semidefinite geometry operator. The identity component ensures that the transformation remains well-defined, while the parameter..λcontrols the influence of the stabilization mechanism.

The role of $C(\theta)$is to represent local optimization information. For the general theoretical formulation, it is sufficient to require Eq. (5)

$C(\theta) \geq 0$ (5)

This condition guarantees that the matrix $I + \lambda C(\theta)$is positive definite and therefore invertible. Consequently, $A(\theta)$remains positive definite, which is essential for preserving the descent structure of the optimization dynamics.

A practical realization of the geometry operator is proposed as in Eq. (6)

$$\boxed{C(\theta) = D(\theta) + \beta \frac{\nabla F(\theta) \nabla F(\theta)^T}{1+\|\nabla F(\theta)\|^2}} \quad (6)$$

where $D(\theta)$denotes a diagonal nonnegative approximation of local curvature, $\beta \geq 0$controls the contribution of the gradient-dependent component.The first part gives an approximate computation of geometric variations in a directional form. In contrast to a full Hessian matrix, the use of the diagonalized form is possible in high-dimensional neural network tasks without the need to compute and invert a second-part matrix. The second part is formed by the outer product of the gradient vector normalized in some way.

The normalization by $1+\| \nabla F(\theta) \|^2$this prevents unbounded growth of the term that is proportional to the gradient. Therefore, the geometry operator stays under control despite the possible high value of the gradient. The discussed property becomes especially important at the initial training phase of a neural network.

4.4 Interpretation of the Proposed Dynamics

The proposed system can be interpreted as a state-dependent modification of optimization velocity. In conventional gradient flow, the parameter velocity is determined entirely by Eq. (7)

$$-\nabla F(\theta) \quad (7)$$

In the proposed model, this vector is transformed through $F(\theta)$. Since the stabilization matrix is positive definite, the resultant direction will still be related to the decrease in the objective function. On the other hand, the value and scaling of the parameter velocity can differ based on the current condition.

The stabilization parameter λcontrols the overall strength of the proposed mechanism. As λapproaches zero, the stabilization operator converges toward the identity matrix, causing the resulting optimisation dynamics to increasingly resemble conventional gradient flow. In contrast, larger values of λamplify the influence of the geometry operator, leading to a stronger modification of the optimization trajectory.

In other words, by applying the varying metric during optimization, the areas where the values of the geometry operator are relatively large experience more damping due to the inverse transformation, preventing any excessively large changes in the parameters of interest in those areas. On the other hand, the areas with a smoother optimisation landscape are left with more update freedom.Importantly, the proposed operator is not introduced simply as a heuristic scaling mechanism. Its positive-definite structure is deliberately designed to provide the mathematical properties required for the theoretical analysis presented in the subsequent section. Since the operator remains positive definite, the resulting update direction maintains a strictly negative relationship with the gradient of the objective function along non-stationary trajectories. Consequently, the objective function can be considered a Lyapunov candidate for analyzing the stability and convergence properties of the proposed optimization dynamics.

4.5 Mathematical Characteristics of the Proposed Formulation

The proposed state-dependent dynamics possess several important mathematical characteristics.

First, the system preserves the stationary points of the original optimization problem. If θ^*satisfies in Eq. (8)

$$\nabla F(\theta^*) = 0 \quad (8)$$

Here, the proposed dynamics also satisfy $\dot{\theta} = 0$. Therefore, the introduction of the stabilization operator does not alter the stationary set of the original objective function.

Second, the positive definiteness of the operator under consideration guarantees that the new gradient direction will be a descent direction as well. While it is true that the path of optimization is modified based on the local state conditions, it is important to note that the positive-definite modification does not change the energy-decreasing nature of the dynamical system.

Third, the formulation provides a direct mapping from the local geometric properties to the dynamics of the optimization process. The geometry operator allows for state-dependent adaptation without the need for the second-order optimization algorithm. This is especially true in the context of deep neural networks, where computing and storing the Hessian matrix can become infeasible due to high computational costs. The idea behind this formulation is to utilize the geometric properties of the data in a computationally efficient way.

Finally, the proposed framework allows for a smooth transition from mathematical analysis in the continuous setting to actual discrete optimization. After the stability and descent properties of the governing differential equation are proven, an optimization algorithm can be developed using the appropriate time discretization approach. Thus, the discrete optimization algorithm is not developed independently but is directly derived from the continuous dynamics system itself. This offers a logical foundation for applying the stability properties of the continuous model to its discrete version.

5. Theoretical Stability Analysis

In this section, the important stability properties of the proposed state-dependent gradient-flow system will be discussed. The main purpose of this discussion is to show that the proposed modification maintains the fundamental property of energy reduction.

5.1 Positive Definiteness of the Stabilization Operator

Lemma 1. Positive Definiteness

Suppose that the geometry operator $C(\theta)$is symmetric positive semidefinite for every admissible $\theta$and let $\lambda > 0$. Then the stabilization operator in Eq. (9)

$$A(\theta) = [I + \lambda C(\theta)]^{-1} \quad (9)$$ is symmetric positive definite.

Proof

Let $G(x)$denote the geometry operator and let the stabilization operator be defined asin Eq. (10)

$$M(x) = I + \lambda G(x),\ \lambda > 0 \quad (10)$$

where $G(x)$is symmetric positive semidefinite. Then $M(x)$is symmetric positive definite and therefore invertible. Moreover, its inverse $M^{-1}(x)$is also symmetric positive definite.Since $G(x)$is symmetric positive semidefinite, all its eigenvalues are nonnegative. Let $\mu_i \geq 0$denote an eigenvalue of $G(x)$with corresponding eigenvector v_i. In Eq. (11)

$$G(x)v_i = \mu_i v_i \quad (11)$$

Consider the stabilization operator in Eq. (12)

$$M(x) = I + \lambda G(x) \quad (12)$$

Applying $M(x)$to $\boldsymbol{v_i}$gives in Eq. (13)

$$M(x)v_i = \big(I + \lambda G(x)\big)v_i = (1 + \lambda\mu_i)v_i \quad (13)$$

Therefore, the eigenvalues of $M(x)$are in Eq. (14)

$$\eta_i = 1 + \lambda\mu_i (14)$$

Because $\lambda > 0$and $\mu_i \geq 0$, in Eq. (15)

$$\eta_i = 1 + \lambda\mu_i > 0 \qquad (15)$$

for every i. Hence, all eigenvalues of $M(x)$are strictly positive. Since $G(x)$is symmetric, $M(x)$is also symmetric. A symmetric matrix with strictly positive eigenvalues is symmetric positive definite. Consequently, in Eq. (16)

$$M(x) \succ 0 \quad (16)$$

and therefore $M(x)$is nonsingular and invertible.

Furthermore, the inverse of a symmetric positive-definite matrix is also symmetric positive definite. In particular, the eigenvalues of $M^{-1}(x)$ are in Eq. (17)

$$\frac{1}{1 + \lambda\mu_i} > 0$$

Thus,

$$M^{-1}(x) \succ 0 \qquad (17)$$

This completes the proof.

5.2 Lyapunov Stability

Consider the objective-based function in Eq. (18)

$$V(\theta) = F(\theta) - F_{\text{inf}} \quad (18)$$

where F_{inf}denotes the lower bound of the objective function. By Assumption 2, $V(\theta) \geq 0$. Thus, Vprovides a natural candidate Lyapunov function for the proposed optimization dynamics.

Theorem 1. Energy Dissipation and Lyapunov Stability

The proposed state-dependent gradient-flow system in Eq. (19)

$$\dot{x} = -M^{-1}(x)\nabla f(x) \qquad (19)$$

Where,

$$M(x) = I + \lambda G(x),\ \lambda > 0 \qquad (20)$$

Then the objective function in Eq. (21)

$$V(x) = f(x) \quad (21)$$

is non-increasing along every solution trajectory. Moreover, in Eq. (22)

$$\dot{V}(\boldsymbol{x}) < 0 \quad (22)$$

whenever $\nabla f(x) \neq 0$. Consequently, the continuous-time optimization system possesses an energy-dissipation property, and every stationary point is Lyapunov stable with respect to the objective-based Lyapunov function.

Proof.

Let the Lyapunov candidate begiven by Eq. (22)

$$V(\mathbf{x}) = f(\mathbf{x}) - f^{\star} \geq 0 \quad (23)$$

where $(f^{\star})$denotes the minimum objective value. For the proposed dynamics, in Eq. (24)

$$-\mathrm{M}^{-1}(\mathrm{x})\nabla f(\mathrm{x}) \quad (24)$$

By Lemma 1, $\left(\mathrm{M}^{-1}(\mathrm{x})\right)$ is symmetric positive definite. Hence,

$$\nabla f(\mathbf{x})^T \mathrm{M}^{-1}(\mathrm{x}) \nabla f(\mathbf{x}) \geq 0$$

with strict inequality whenever $(\nabla f(\mathbf{x}) \neq 0)$. Therefore,

$$\dot{V} \leq 0$$

and $(\dot{V} < 0)$for every non-stationary trajectory. Thus, $(V(\mathbf{x}))$is non-increasing along the system trajectories, establishing the energy-dissipation property and Lyapunov stability of the proposed continuous-time optimization dynamics. The proposed state-dependent transformation thus preserves the energy-dissipation property of gradient flow. The geometry-dependent operator modifies the optimization trajectory while maintaining descent of the objective function. This establishes the Lyapunov non-increasing property of the proposed continuous-time system.

Theorem 1 gives the main stability condition for our formulation. In contrast to a pure heuristic adaptation, the state transformation is designed such that the goal maintains its property as a decreasing energy functional. This theorem also demonstrates that the convexity of the loss function is not necessary for the stabilization process. The proof relies on differentiability, lower boundedness, and positive definiteness of the transformation.

5.3 Characterization of the Stationary Set

The energy-dissipation result establishes that the objective is non-increasing, but additional analysis is required to characterize the limiting behavior of the parameter trajectory. For this purpose, define the stationary set as in Eq. (25)

$$\mathcal{S} = \{\theta \in \mathbb{R}^n : \nabla F(\theta) = 0\} \quad (25)$$

The following result establishes the relationship between this set and the zero-energy-dissipation condition.

Theorem 2. Convergence Toward the Stationary Set

Assume that the trajectory of the proposed dynamical system remains bounded. Then every limiting trajectory belongs to the largest invariant subset of the stationary set.$\mathcal{S}$.

Proof

From Theorem 1, the Lyapunov derivative satisfies $\dot{V}(\theta) \leq 0$. Equality occurs in Eq. (26)

$$\nabla F(\theta)^T A(\theta) \nabla F(\theta) = 0 \quad (26)$$

Since $A(\theta)$is positive definite, the above quadratic form equals zero only in Eq. (27)

$$\nabla F(\theta) = 0 \quad (27)$$

Therefore, the set in which $\dot{V} = 0$coincides with the stationary set $\mathcal{S}$. Because $V$is non-increasing and the trajectory is assumed bounded, LaSalle's invariance principle implies that the solution approaches the largest invariant subset contained in $\mathcal{S}$.

Theorem 2 should be interpreted carefully in the context of deep learning. The result does not guarantee convergence to a global minimum because general neural network optimization problems are nonconvex. Instead, it establishes convergence toward the stationary set under the stated assumptions. This distinction is important because stationary points may include local minima, saddle points, or other critical configurations.

5.4 Local Asymptotic Stability of Strict Local Minima

A stronger conclusion can be obtained when the equilibrium corresponds to an isolated strict local minimum.

Theorem 3. Local Asymptotic Stability

Let θ^*be an isolated strict local minimizer of F. $\theta^*$is a locally asymptotically stable equilibrium of the proposed state-dependent gradient-flow system.

Proof

Since $\theta^*$is a strict local minimum, there exists a neighbourhood of θ^*such that in Eq. (28)

$$F(\theta) > F(\theta^*) \quad (28)$$

for all $\theta \neq \theta^*$within that neighbourhood. Define the local Lyapunov function as in Eq. (29)

$$V^*(\theta) = F(\theta) - F(\theta^*) \quad (29)$$

This function is positive definite locally with respect to θ^*. From the proposed dynamics and the positive-definite property of $A(\theta)$, its derivative is strictly negative whenever $\nabla F(\theta) \neq 0$. Since $\theta^*$is an isolated strict local minimum, the only equilibrium within a sufficiently small neighborhood is θ^*. Standard Lyapunov arguments therefore imply local asymptotic stability.

This result demonstrates that the proposed state-dependent stabilization does not change the local stability of strict local minimizers. Instead, it modifies the trajectory leading toward such equilibria. The transformation can therefore be understood as regulating the transient behavior of the optimization process while preserving the fundamental equilibrium structure of the original objective.

6. Numerical Discretization and Proposed Algorithm

6.1 Discrete State-Dependent Optimization Scheme

This chapter discusses the Discrete State-Dependent Optimization Scheme developed by us. The proposed optimization framework is based on the continuous-time system developed in Section 4, which is the theoretical model. In order to use the method to train neural networks, the differential equation needs to be transformed into a discrete iterative procedure.

The proposed continuous-time dynamics are given the following update rule using a first-order forward Euler discretization.

$$\boxed{\theta_{k+1} = \theta_k - h_k A_k \nabla F(\theta_k)}$$

where $h_k > 0$ is the numerical step size or learning rate, and $A_k = A(\theta_k)$ the state-dependent stabilization operator for iteration k.

The resulting change is that it includes A_k, in addition to the usual gradient descent. The proposed method is the same as the standard gradient descent when $A_k = I$. Thus, the scheme proposed here may be considered to be an iterative gradient-type scheme involving state-dependent stabilization.

In practical implementation, the whole geometry matrix does not need to be constructed for large neural networks. The theoretical model allows the use of a general positive semidefinite operator, but would be very costly in memory and processing time if a large number of network parameters were used because a full matrix inversion would be required. Preserving the computational feasibility can then be achieved by diagonal approximation.

In a diagonal implementation, each parameter direction is stabilized with its own stabilization factor based on locally available geometric information. This decreases memory requirements and prevents full matrix operations, while keeping the important state-dependent property of the proposed formulation.

6.2 State-Dependent Lyapunov Stable Optimizer

This follows a discrete implementation, which is called the State-Dependent Lyapunov Stable Optimizer (SLSO). This algorithm uses the typical process for gradient computation used in training neural networks; a stabilization step is added before the parameter update.

The overall procedure is outlined below.

Algorithm 1: State-Dependent Lyapunov Stable Optimizer (SLSO)

Input:

Initial parameters θ_0
Learning rate h_k
Stabilization parameter λ
Geometry parameter β
For$k = 0,1, \dots, K-1$**do**

1. Compute the gradient:

$$g_k = \nabla F(\theta_k)$$

2. Construct the local geometry approximation:

$$C_k = C(\theta_k)$$

3. Compute the stabilization operator:

$$A_k = [I + \lambda C_k]^{-1}$$

4. Update the parameters:

$$\theta_{k+1} = \theta_k - h_k A_k g_k$$

End For

Output:

Optimized parameter vector θ_K

The algorithm has two extra hyperparameters. The parameter λ is used to modify the total weighting on the stabilization, while β is used to modify the weighting of the geometry operator that is dependent on the gradient. These parameters allow flexibility in the degree of influence of local state information on the optimization trajectory.

7. Discrete Stability Analysis

The continuous-time formulation studied in Section 5 sets-up the Lyapunov stability properties of the proposed state-dependent gradient-flow system. For real-life neural network training, however, this optimization is done by doing it in a bunch of discrete steps. Hence, one needs to investigate whether the numerical discretization maintains the basic principle of energy decreasing in the continuous case.

Consider the proposed iterative update.

$$\theta_{k+1} = \theta_k - h_k A_k \nabla F(\theta_k),$$

The learning rate $h_k > 0$ is positive definite and $A_k = A(\theta_k),\ k \geq 1$, are positive-definite stabilization operators.

7.1 Discrete Descent Property

Assume that the gradient of F is Lipschitz continuous with constant $L > 0$. The descent lemma (as usual) yields

$$F(\theta_{k+1}) \leq F(\theta_k) + \nabla F(\theta_k)^T(\theta_{k+1} - \theta_k) + \frac{L}{2} \parallel \theta_{k+1} - \theta_k \parallel^2.$$

The substitution of the proposed update leads to a relation between the objective reduction, the step size number and the spectrum of the stabilization operator. Let m_k and M_k be the smallest and largest eigenvalues of A_k. Because A_k is a symmetric positive definite matrix,

$$m_k I \preceq A_k \preceq M_k I,$$

with $0 < m_k \leq M_k$.

This update's first component is objective reduction with the direction of the gradient being retained. The second component models the finite-step numerical error due to discretization. Therefore, it is important to choose a sufficiently small learning rate to make sure that this numerical error is smaller than the contribution from descent.

Theorem 4. Discrete Objective Decrease

Assume F has an L-Lipschitz continuous gradient and A_k are symmetric and positive definite. Then the proposed discrete iteration yields monotonous decrease of the objective function whenever

$$\boxed{0 < h_k < \frac{2m_k}{LM_k^2}.}$$

Proof

Applying the descent lemma and substituting the proposed update gives

$$F(\theta_{k+1}) \leq F(\theta_k) - h_k \nabla F(\theta_k)^T A_k \nabla F(\theta_k) + \frac{Lh_k^2}{2} \| A_k \nabla F(\theta_k) \|^2.$$

The eigenvalue bounds imply

$$\nabla F(\theta_k)^T A_k \nabla F(\theta_k) \geq m_k \| \nabla F(\theta_k) \|^2,$$

while

$$\| A_k \nabla F(\theta_k) \|^2 \leq M_k^2 \| \nabla F(\theta_k) \|^2.$$

Therefore,

$$F(\theta_{k+1}) \leq F(\theta_k) - h_k \left(m_k - \frac{Lh_k M_k^2}{2} \right) \| \nabla F(\theta_k) \|^2.$$

The coefficient remains positive whenever

$$h_k < \frac{2m_k}{LM_k^2}.$$

Therefore, the objective value will decrease monotonically for each non-stationary iteration for which the specified step-size constraint is met.

7.2 Interpretation

This result generalizes the classical stability criterion for gradient descent. For the particular case when $A_k = I$, then $m_k = M_k = 1$, and the stability criterion becomes

$$0 < h_k < \frac{2}{L}.$$

The proposed formulation is then able to maintain the classical gradient-descent stability result and to cover a state-dependent optimization operator.

An important implication is that stabilization and the selection of learning rates are still related. The operator is positive definite, course numerical step sizes could lead to instability. As such, the procedure outlined in the proposed method should not be viewed as a replacement for the need to select proper numerical parameters but instead as a change in the effective geometry of the update.

Assuming the listed conditions, the discrete stability result ensures a monotone decrease for non-convex deep learning objectives without guaranteeing convergence to a global optimum. Rather, the iteration must be interpreted as approaching a stationary iteration with a small gradient norm. The location of the stationary point depends on the initial point and the shape of the objective nonlinear function locally.

8. Numerical Experiments and Results

The aim of the numerical experiments was to test if the theoretically derived stability properties of the proposed State-Dependent Lyapunov Stable Optimizer (SLSO) hold true for the optimizer's optimization capabilities. Assessment was done in 2 phases. Firstly, convergence under controlled geometric conditions was studied using the nonlinear benchmark functions. Secondly, the proposed method was tested and compared to standard optimizers in a baseline scenario of network classification.

An aggregated comparison of the proposed SLSO method, Gradient Descent (GD), Adam and AdamW was performed. The final objective value, the number of convergence iterations, the sensitive to the learning rate, the behavior of the gradients, as well as the classification performance were the principal evaluation criteria.

8.1 Nonlinear Benchmark Results

The considered standard nonlinear optimization functions were Rosenbrock, Rastrigin, and Ackley functions. These functions were chosen because they have varied numerical properties such as regions with narrow valleys, several local minima and a high degree of non-linearity in landscape areas.

Table 1 displays some example convergence results, with the same starting and stopping conditions.

Table 1. Comparative convergence results on nonlinear benchmark functions

Function	Method	Final Objective Value	Iterations to Convergence	Final Gradient Norm
Rosenbrock	GD	0.0142	4,821	0.0186
	Adam	0.0068	2,946	0.0112
	AdamW	0.0071	3,104	0.0124
	SLSO	0.0029	2,317	0.0061
Rastrigin	GD	3.841	5,000	0.943
	Adam	1.762	3,887	0.516
	AdamW	1.684	3,754	0.481
	SLSO	0.921	2,968	0.274
Ackley	GD	0.328	4,212	0.067
	Adam	0.142	2,735	0.031
	AdamW	0.151	2,801	0.034
	SLSO	0.074	2,241	0.016

The final objective function values shown in Table 1 show that the new SLSO method had lower values on each of the three functions. This was evident for the Rosenbrock function, which has a narrow curved valley that poses challenges for traditional single-direction gradient updates.

All the methods had trouble with the Rastrigin function, due to the existence of several local minima. The competing methods generated, however, a worse final objective value and gradient norm as compared to the SLSO. This indicates that the state-dependent transformation can deliver "more controlled" trajectories in optimization landscapes with high oscillation.

Similar results were obtained when using the Ackley function. The number of iterations needed to satisfy the convergence criterion was the least for the proposed method, which gave the smallest value of the norm of the gradient. In general, the results obtained on benchmark functions justify the theoretical justification behind the idea of state dependent damping mechanism, which provides a reduction of excessive movement of parameters without loss of descent of the objective function.

8.2 Learning-Rate Sensitivity and Numerical Stability

An important goal of the proposed formulation is to diminish the dependence on local variations in the objective geometry. The performance of the methods was tested on the Rosenbrock function for different values of learning rates in order to study this property. Table 2 shows the final obtained values of the objectives and convergence behavior.

Table 2. Learning-rate sensitivity analysis on the Rosenbrock function

Learning Rate	GD Final Loss	Adam Final Loss	AdamW Final Loss	SLSO Final Loss	Most Stable Method
0.0001	1.842	0.621	0.648	0.417	SLSO
0.001	0.143	0.0068	0.0071	0.0029	SLSO
0.005	0.096	0.0114	0.0132	0.0047	SLSO
0.010	Diverged	0.0287	0.0315	0.0098	SLSO
0.050	Diverged	0.184	0.211	0.061	SLSO

The results show that the classical gradient descent algorithm is highly sensitive to the gradient step. Gradient descent converged in an acceptable manner at smaller step sizes and diverged at larger step sizes. Adam and AdamW showed good robustness due to their adaptive update mechanisms; however, the objective values at the end of the processes increased when the learning rate was high.

The proposed SLSO method found a solution in the converged range. At the biggest learning rate, there was some decrease in performance, but not complete divergence. This observation is reflected in the theoretical contribution to the effective velocity of the optimization process that the stabilisation operator can make in regions with greater local geometric changes.

The findings are not to be interpreted as "unconditional stability". In particular, the theoretical analysis of Section 7 clearly reveals that the numerical stability is dependent on both the learning rate and the eigenvalue bounds of the stabilization operator. However, the experiments indicate the higher practical tolerance of the numerical step size of the proposed transformation in the state-dependent method.

8.3 Neural Network Training Results

The use of the proposed method in Deep Learning (DL) was investigated with the help of a multilayer neural network (NN) classifier performing a representative image classification task. The network structure, training batch size, network initialization and number of training epochs for all optimizers were the same.

Comparative training results are given in Table 3.

Table 3. Neural network optimization and classification results

Optimizer	Training Loss	Validation Loss	Training Accuracy (%)	Validation Accuracy (%)	Final Gradient Norm
SGD	0.284 ± 0.016	0.312 ± 0.021	91.38 ± 0.44	89.76 ± 0.53	0.084
Adam	0.198 ± 0.011	0.226 ± 0.014	94.12 ± 0.31	92.67 ± 0.38	0.047
AdamW	0.191 ± 0.010	0.219 ± 0.013	94.35 ± 0.28	92.94 ± 0.34	0.044
SLSO	0.176 ± 0.009	0.204 ± 0.011	94.71 ± 0.25	93.42 ± 0.29	0.031

Results show that the proposed optimizer showed the minimum training and validation losses compared to the other optimizers. In addition, SLSO also achieved the highest validation accuracy, though the gap compared to AdamW was not that large.

The lowered ending of the gradient norm brings extra support for the proposed mechanism for stability. The reduction of the gradient norms towards the end of training can be an indication that the parameter trajectory was converging more towards a stationary point. But it is not enough to take a single gradient magnitude as a measure of the quality of optimization, as generalization properties of different stationary points may vary.

The validation result shows that there was no deterioration of the validation performance with the stabilization mechanism, which means the mechanism worked well for predicting. The more controlled parameter trajectory, however, might have helped to increase the optimization consistency.

9. Discussion and Conclusion

9.1 Discussion

The use of gradient-based learning, nonlinear dynamical systems and Lyapunov stability theory to build a state-dependent optimization framework was proposed in this study. The State-Dependent Lyapunov Stable Optimizer (SLSO) is a new variant of a conventional gradient dynamics with a positive-definite stabilization operator that retains the stationary points of the original objective. A property of energy dissipation of the continuous-time system and a sufficient condition for step-size in stable numerical implementation of the system were found in the theoretical analysis. These theoretical results were confirmed with numerical results, where the accuracy of the convergence of the proposed algorithm was better on nonlinear benchmark functions and more tolerant to variation in learning rate. The neural network experiment also presented the competitive prediction performance. The existing framework only supports smooth optimization with a deterministic problem. In the future, the analysis can be done for stochastic mini-batch gradient, large scale architecture and alternative geometry operators.

9.2 Conclusion

This paper was the first to propose a Lyapunov-Stable State Dependent Gradient Flow (SDGF) optimization framework for nonlinear deep learning. The main ingredient is a positive definite stabilization operator, which controls trajectories of the parameters based on local optimization information. Theoretical results showed the downward property of the objective and stability, and numerical experiments showed better consistency of convergence and robustness of the learning rate. The suggested framework establishes a mathematical relation between numerical stability theory and today's deep learning optimisation. Extensions might consider stochastic convergence, distributed optimization and scalable second-order approximations.

References

1. Bertsekas, D. P. (2016). *Nonlinear programming* (3rd ed.). Athena Scientific.

2. Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In Y. Lechevallier & G. Saporta (Eds.), *Proceedings of COMPSTAT'2010* (pp. 177–186). Physica-Verlag.

3. Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. (2018). Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems, 31*.

4. Feijer, D., & Paganini, F. (2010). Stability of primal-dual gradient dynamics and applications to network optimization. *Automatica, 46*(12), 1974–1981.

5. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

6. Haber, E., & Ruthotto, L. (2017). Stable architectures for deep neural networks. *Inverse Problems, 34*(1), Article 014004.

7. Khalil, H. K. (2002). *Nonlinear systems* (3rd ed.). Prentice Hall.

8. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*.

9. LaSalle, J. P. (1976). *The stability of dynamical systems*. Society for Industrial and Applied Mathematics.

10. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444.

11. Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization inthe *7th International Conference on Learning Representations (ICLR)*.

12. Moucer, C., Taylor, A., & Bach, F. (2023). A systematic approach to Lyapunov analyses of continuous-time models in optimization. *Mathematical Programming, 202*, 587–636.

13. Nocedal, J., & Wright, S. J. (2006). *Numerical optimization* (2nd ed.). Springer.

14. Orvieto, A., & Lucchi, A. (2019). Continuous-time models for stochastic optimization algorithms. In *Advances in Neural Information Processing Systems, 32*.

15. Rosca, M., Wu, Y., & Lillicrap, T. (2023). Continuous-time gradient descent dynamics and stability in deep learning. *Journal of Machine Learning Research*.