Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.061913

ARTICLE

Deterministic Convergence Analysis for GRU Networks via Smoothing Regularization

Qian Zhu[1], Qian Kang[1], Tao Xu[2], Dengxiu Yu[3,*] and Zhen Wang[1]

[1]School of Cybersecurity, Northwestern Polytechnical University, Xi'an, 710072, China
[2]Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an, 710072, China
[3]School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an, 710072, China

*Corresponding Author: Dengxiu Yu. Email: yudengxiu@126.com

Received: 06 November 2024; Accepted: 11 February 2025; Published: 16 April 2025

ABSTRACT: In this study, we present a deterministic convergence analysis of Gated Recurrent Unit (GRU) networks enhanced by a smoothing L_1 regularization technique. While GRU architectures effectively mitigate gradient vanishing/exploding issues in sequential modeling, they remain prone to overfitting, particularly under noisy or limited training data. Traditional L_1 regularization, despite enforcing sparsity and accelerating optimization, introduces non-differentiable points in the error function, leading to oscillations during training. To address this, we propose a novel smoothing L_1 regularization framework that replaces the non-differentiable absolute function with a quadratic approximation, ensuring gradient continuity and stabilizing the optimization landscape. Theoretically, we rigorously establish three key properties of the resulting smoothing L_1-regularized GRU (SL1-GRU) model: (1) monotonic decrease of the error function across iterations, (2) weak convergence characterized by vanishing gradients as iterations approach infinity, and (3) strong convergence of network weights to fixed points under finite conditions. Comprehensive experiments on benchmark datasets-spanning function approximation, classification (KDD Cup 1999 Data, MNIST), and regression tasks (Boston Housing, Energy Efficiency)-demonstrate SL1-GRUs superiority over baseline models (RNN, LSTM, GRU, L1-GRU, L2-GRU). Empirical results reveal that SL1-GRU achieves 1.0%–2.4% higher test accuracy in classification, 7.8%–15.4% lower mean squared error in regression compared to unregularized GRU, while reducing training time by 8.7%–20.1%. These outcomes validate the method's efficacy in balancing computational efficiency and generalization capability, and they strongly corroborate the theoretical calculations. The proposed framework not only resolves the non-differentiability challenge of L_1 regularization but also provides a theoretical foundation for convergence guarantees in recurrent neural network training.

KEYWORDS: Gated recurrent unit; regularization; convergence

1 Introduction

Recurrent Neural Networks (RNN) have emerged as a powerful class of neural networks, particularly adept at modeling sequential data due to their ability to retain and utilize temporal dependencies [1]. These networks have demonstrated remarkable success across various domains, including natural language processing, speech recognition, and time-series forecasting [2]. However, the application of RNN is not without challenges. One of the primary issues is the vanishing and exploding gradient problem, which can significantly hinder the training of deep RNN [3,4]. To address this, several variants of RNN have been proposed, such as Long Short-Term Memory Networks (LSTM) and Gated Recurrent Units (GRU) [5,6]. These architectures incorporate gating mechanisms to selectively retain or forget information, effectively mitigating gradient-related issues and improving performance [7]. LSTM, for instance, uses a combination of input, forget, and output gates to control the flow of information, allowing the network to retain relevant information over extended sequences [8]. Similarly, GRU simplifies the gating mechanism while maintaining comparable performance, making them computationally more efficient [9].

Despite the advancements in RNN architectures, the issue of overfitting remains a significant challenge, particularly when dealing with limited or noisy data [10]. Overfitting occurs when a model learns to memorize the training data instead of generalizing to unseen samples, leading to poor performance on test data [11,12]. Regularization techniques have been introduced to address this, aiming to improve the generalization ability of neural networks by controlling their complexity. Common regularization methods, such as L_2 regularization and dropout, have shown efficacy in various settings [13–15].

L_2 regularization penalizes large weights by adding their squared magnitude to the loss function, thereby encouraging simpler models [16,17]. Dropout, on the other hand, randomly deactivates neurons during training, preventing the network from relying too heavily on specific features [18–20]. L_1 regularization can suppress weight growth to enhance model performance and increase parameter sparsity to improve computational efficiency [21–23]. Building on these methods, researchers have analyzed the theoretical properties of regularized networks. Zhang et al. [24] investigate a penalized batch backpropagation algorithm for training feedforward neural networks. They establish the boundedness, as well as the weak and strong convergence properties of the algorithm, using mathematical methods. Similarly, Wang et al. [25] prove the boundedness of backpropagation neural networks (BPNN) with L_2 regularization and provide convergence results based on this. Kang et al. [26] incorporate an adaptive momentum term into the iterative error function when training the group lasso-regularized Sigma Pi Sigma neural network, thus boosting the algorithm's convergence speed and reducing the model's training time. Yu et al. [27] optimize a generalized learning system using $L_{1/2}$ regularization, further examining its theoretical properties and performance. However, there are significant difficulties in the theoretical analysis of L_1 regularization [28]. The L_1 regularization term is often written as (1):

$$\Omega(w) = \|w\|_1 = \sum_i |w_i| \tag{1}$$

where $\|\cdot\|_1$ represents 1-$norm$. Obviously, the L_1 regularization term lacks a derivative at the origin [29,30]. Therefore, it is necessary to introduce smoothing approximation functions to solve the non-differentiability problem of L_1 regularization [31].

In this research, we propose the use of GRU networks with smoothing L_1 regularization to address the aforementioned challenges. Unlike traditional L_1 regularization, which can introduce non-differentiable points, the smoothed variant ensures a more stable optimization process, making it better suited for modern neural network architectures.

This research primarily focuses on analyzing the monotonicity, weak convergence, and strong convergence properties of GRU networks with smoothing L_1 regularization (referred to as SL1-GRU), including theoretical proofs and simulation experiments. This paper makes the following contributions:

(1) The smoothing L_1 regularization is integrated into the network, effectively overcoming the oscillation phenomenon caused by traditional L_1 regularization. At the same time, the redundant weight values in the network are trimmed, further optimizing the network structure and improving its sparsity.

(2) Under given conditions and assumptions, the monotonicity, weak convergence, and strong convergence of SL1-GRU are theoretically demonstrated. The network's error function decreases monotonically with the increasing number of iterations. As iterations approach infinity, weak convergence is demonstrated

by the error function's gradient approaching zero. Strong convergence means network weights can converge to a fixed point under defined conditions.

(3) The theoretical results are validated through experiments on approximation, classification, and regression tasks. The experimental results show that GRU networks with smoothing L_1 regularization achieve excellent performance in solving various machine learning problems, with high sparsity generated during the network weights optimization process, which is conducive to optimizing the network structure, reducing the possibility of overfitting and improving the generalization ability of the network.

The rest of this paper is structured as follows: Section 2 explores the GRU network structure and the parameter iteration mechanism after introducing smoothing L_1 regularization. Section 3 discusses the principal theoretical achievements. Section 4 confirms the theoretical findings and the practical performance of SL1-GRU through simulation experiments. Lastly, Section 5 encapsulates the research content and discusses possible directions for future investigations. The detailed proofs of theorems and corollaries are included in the Appendix A.

2 GRU Based on Regularization Method

2.1 Network Structure of GRU

As a streamlined variant of LSTM, GRU features just two gate mechanisms: an update gate and a reset gate [32]. The internal configuration of GRU, shown in Fig. 1, together with the standard forward propagation equations, is detailed below:

$$z_t = \sigma\left(W_z\left[h_{t-1}, x_t\right] + b_z\right) \tag{2}$$

$$r_t = \sigma\left(W_r\left[h_{t-1}, x_t\right] + b_r\right) \tag{3}$$

$$\tilde{h}_t = \tanh\left(W_{\tilde{h}}\left[r_t \circ h_{t-1}, x_t\right] + b_{\tilde{h}}\right) \tag{4}$$

$$h_t = z_t \circ \tilde{h}_t + \left(1 - z_t\right) \circ h_{t-1} \tag{5}$$

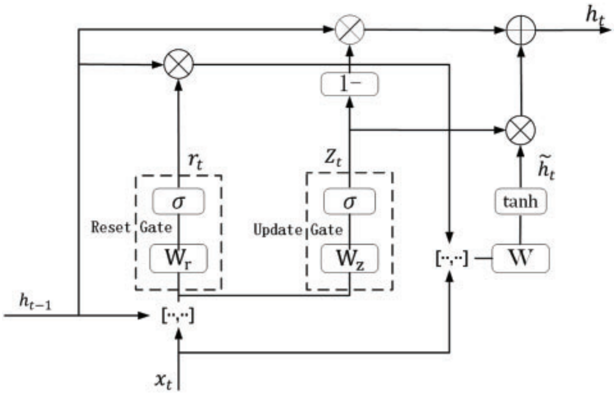


Figure 1: Structure of GRU

The following are the explanations for the related symbols:

- The symbol ∘ stands for the Hadamard product, which refers to element-wise multiplication.
- $[\cdot, \cdot]$ denotes the concatenation of two vectors into a longer vector.
- x_t denotes the input to the network at time t.
- z_t and r_t correspond to the update outputs and reset gate outputs at time t, respectively.

- At time t, $\tilde{h}_t$ denotes the candidate output, while h_t represents the output of hidden layer.
- The symbols W_r, W_z, and $W_{\tilde{h}}$ respectively signify the weight matrices associated with the reset gate, the update gate, and the candidate output.
- σ represents the sigmoid function, a nonlinear activation mapping real-valued inputs to the range $(0, 1)$. Similarly, tanh is a nonlinear function that maps inputs to the range $(-1, 1)$.
- b_r, b_z, and $b_{\tilde{h}}$ correspond to the biases for the respective weight matrices.

In (2), W_z denotes the weight matrix associated with the update gate. In fact, W_z is formed by concatenating two matrices: $W_{z,h}$, which corresponds to the input vector h_{t-1}, $W_{z,x}$, which corresponds to the input vector. Therefore, Eq. (2) can be written as:

$$\begin{aligned}[W_z]\begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} &= \begin{bmatrix} W_{z,h} & W_{z,x} \end{bmatrix}\begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \\ &= W_{z,h}h_{t-1} + W_{z,x}x_t \end{aligned} \tag{6}$$

Obviously, the weight matrix in other Eqs. (3)–(5) can be also rewritten in the same form as (6). For the convenience of subsequent analysis, we set the biases b_r, b_z, and $b_{\tilde{h}}$ to 0 and get new expressions as follows:

$$z_t = \sigma(W_{z,h} \cdot h_{t-1} + W_{z,x} \cdot x_t) \tag{7}$$

$$r_t = \sigma(W_{r,h} \cdot h_{t-1} + W_{r,x} \cdot x_t) \tag{8}$$

$$\tilde{h}_t = \tanh(W_{\tilde{h}} \cdot (r_t \circ h_{t-1}) + W_{\tilde{h}} \cdot x_t) \tag{9}$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t \tag{10}$$

If $\{x^n, T^n\}_{n=1}^{N} \subset \mathbb{R}^N \times \mathbb{R}^N$ as the given set of the training samples, where the x^n represent the n-th input sample and the T^n is the label, respectively. Let $y_t^n = \sigma\left(w_{\text{out}} \cdot h_t^n\right) \in \mathbb{R}$ be the actual output for each input X^n, and $y_t^0 = w_{\text{out}} \cdot h_t$. Thereby, the error function is defined by the following formula:

$$\begin{aligned}\tilde{E}(W) &= \frac{1}{2N}\sum_{n=1}^{N}\left(y_t^n - T^n\right)^2 \\ &= \frac{1}{2N}\sum_{n=1}^{N}\left(\sigma\left(w_{\text{out}} \cdot h_t^n\right) - T^n\right)^2 \\ &= \frac{1}{N}\sum_{n=1}^{N}\sigma_n\left(w_{\text{out}} \cdot h_t^n\right)\end{aligned} \tag{11}$$

where $\sigma_n(r) = \frac{1}{2}\left(\sigma(r) - T^n\right)^2, r \in \mathbb{R}, 1 \leq n \leq N.$

2.2 Gradient Learning Method with Smoothing L_1 Regularization for GRU

The standard approach to achieve L_1 regularization entails incorporating a penalty term within the error function, expressed as:

$$E(W) = \tilde{E}(W) + \lambda \left\|w_{\text{out}}\right\|_1 \tag{12}$$

This can be written as:

$$E(W) = \sum_{n=1}^{N}\sigma_n\left(w_{\text{out}} \cdot h_t^n\right) + \lambda\left\|w_{\text{out}}\right\|_1 \tag{13}$$

where $\lambda > 0$ is the penalty parameter, while $||\cdot||_1$ indicates L_1-*norm*.

However, there is no derivative of the L_1 regularization term at the origin [33,34]. To tackle the non-differentiable problem of the L_1 regularization term, a smoothing approximation function is introduced. Smoothing approximation is essentially the use of continuous differentiable functions instead of absolute value functions. In this paper, a quadratic form smoothing approximation function is used, which means:

$$h(x) = \begin{cases} |x|, |x| \geq \alpha \\ \dfrac{|x|^2}{2\alpha} + \dfrac{\alpha}{2}, |x| < \alpha \end{cases} \tag{14}$$

The smoothing coefficient α is a constant greater than zero. Fig. 2 illustrates the effect of α on the degree of approximation. It is easy to see that when the smoothing coefficient α tends to zero, the approximation function increasingly resembles the absolute function. Therefore, in practical applications, the smaller the smoothing coefficient, the closer the actual effect of the regularization term is to the L_1 regularization method.

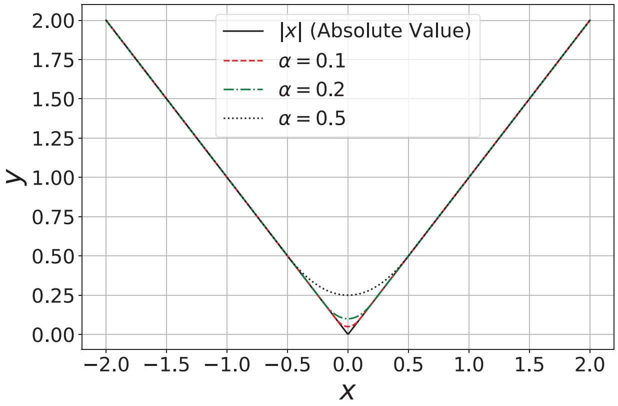


Figure 2: Influence of smoothing coefficient on fitting degree

By incorporating a smooth approximation function into the error propagation mechanism of L_1 regularized GRU, the issue of non-differentiability at the origin is overcome, providing a basis for a rigorous analysis of the error function's monotonicity. Specifically, the error function expression of the smoothed SL1-GRU model is derived as follows by replacing the L_1 regularization term with the smoothed approximation function $L_1(W_{out})$:

$$E = \sum_{n=1}^{N} \sigma_n \left(W_{\text{out}} \cdot h_t^n \right) + \lambda L_1 \left(W_{\text{out}} \right), \lambda > 0 \tag{15}$$

The element $L_1(W_{\text{out}}^{i,j})$ is positioned in the i-th row and j-th column of the matrix $L_1(W_{\text{out}})$. Specifically, $L_1(W_{\text{out}}^{i,j})$ is defined as follows:

$$L_1(W_{\text{out}}^{i,j}) = \begin{cases} |W_{\text{out}}^{i,j}|, & \text{if } |W_{\text{out}}^{i,j}| \geq \alpha \\ \dfrac{|W_{\text{out}}^{i,j}|^2}{2\alpha} + \dfrac{\alpha}{2}, & \text{if } |W_{\text{out}}^{i,j}| < \alpha \end{cases} \tag{16}$$

here, α is a given bounded constant.

The optimization algorithm, Stochastic Gradient Descent (SGD), is frequently used to train GRU. To achieve the fastest reduction of the error function E, the direction of weight changes should be the same as the negative gradient of E in the weight matrix. The learning rate, symbolized by η, is a scalar hyperparameter that determines the step increment for each iteration in the optimization algorithm. $\nabla_W E$ represents the partial derivative of the error function E with respect to the weight W. If W^k and W^{k+1} denote the weight matrices for the k-th and (k+1)-th iterations, respectively, and ΔW^k represents the change in the weight matrix from W^k to W^{k+1}. The weight update rule for the SGD algorithm is defined as:

$$W^{k+1} = W^k + \Delta W^k = W^k - \eta \nabla_W E \tag{17}$$

This equation indicates that during each iteration, SL1-GRU updates the weights by deducting the result of multiplying the learning rate η by the gradient of the error function with respect to the weights from the current weight matrix W_k, causing the weights to change in a direction that reduces the error function. By iteratively applying this rule, the weights are adjusted to minimize the error.

Define $\delta_{h,t}^k$ as the partial derivative of E over h_t^k, and it is given by:

$$\delta_{h,t}^k = \frac{\partial E}{\partial h_t^k} \tag{18}$$

Similarly, define:

$$\delta_{z,t}^k = \frac{\partial E}{\partial z_t^k} \circ z_t^k \circ (1 - z_t^k) = \delta_{h,t}^k \circ (\tilde{\boldsymbol{h}}_t^k - \boldsymbol{h}_{t-1}^k) \circ z_t^k \circ (1 - z_t^k) \tag{19}$$

$$\delta_{r,t}^k = \frac{\partial E}{\partial \boldsymbol{r}_t^k} \circ \boldsymbol{r}_t^k \circ \left(1 - \boldsymbol{r}_t^k\right) = \boldsymbol{h}_{t-1}^k \circ [(\delta_{h,t}^k \circ z_t^k \circ (1 - (\tilde{\boldsymbol{h}}_t^k)^2 t)\boldsymbol{W}_h^k] \circ \boldsymbol{r}_t^k \circ (1 - \boldsymbol{r}_t^k) \tag{20}$$

$$\delta_{\tilde{h},t}^k = \frac{\partial E}{\partial \tilde{\boldsymbol{h}}_t^k} \circ (1 - (\tilde{\boldsymbol{h}}_t^k)^2) = \delta_{h,t}^k \circ z_t^k \circ (1 - (\tilde{\boldsymbol{h}}_t^k)^2) \tag{21}$$

For each weight matrix, the partial derivatives of E are as follows:

$$\nabla_{W_{z,h}}^k E = \frac{\partial E}{\partial \boldsymbol{W}_{z,h}^k} = \sum_{i=1}^{t} \boldsymbol{\delta}_{z,i}^k \boldsymbol{h}_{i-1}^k, \tag{22}$$

$$\nabla_{W_{z,x}}^k E = \frac{\partial E}{\partial \boldsymbol{W}_{z,x}^k} = \sum_{i=1}^{t} \boldsymbol{\delta}_{z,i}^k \boldsymbol{x}_i^k, \tag{23}$$

$$\nabla_{W_{r,h}}^k E = \frac{\partial E}{\partial \boldsymbol{W}_{r,h}^k} = \sum_{i=1}^{t} \boldsymbol{\delta}_{r,i}^k \boldsymbol{h}_{i-1}^k, \tag{24}$$

$$\nabla_{W_{r,x}}^k E = \frac{\partial E}{\partial W_{rx}} = \sum_{i=1}^{t} \boldsymbol{\delta}_{r,i}^k \boldsymbol{x}_i^k, \tag{25}$$

$$\nabla_{W_{\tilde{h},r}}^k E = \frac{\partial E}{\partial \boldsymbol{W}_h^k} = \sum_{i=1}^{t} \boldsymbol{\delta}_i^k \left(\boldsymbol{r}_i^k \circ \boldsymbol{h}_{i-1}^k\right), \tag{26}$$

$$\nabla_{W_{\tilde{h},x}}^k E = \frac{\partial E}{\partial W_x^k} = \sum_{i=1}^{t} \boldsymbol{\delta}_i^k \boldsymbol{x}_i^k, \tag{27}$$

For the output weight matrix W_{out}, the partial derivative of E specifically is:

$$\frac{\partial E}{\partial W_{\text{out}}^{k}} = \sum_{n=1}^{N} \sigma_n' \left(W_{\text{out}}^{k} \cdot h_t^{k,n} \right) \Delta W_{\text{out}}^{k} \cdot h_t^{k,n} + \lambda \frac{\partial L_1 \left(W_{\text{out}} \right)}{\partial W_{\text{out}}} \tag{28}$$

where $\lambda > 0$.

According to (17) and (22) to (28), the weights are updated iteratively by:

$$W_{z,h}^{k+1} = W_{z,h}^{k} + \Delta \boldsymbol{W}_{z,h}^{k} = W_{z,h}^{k} - \eta \frac{\partial E}{\partial \boldsymbol{W}_{z,h}^{k}} \tag{29}$$

$$W_{z,x}^{k+1} = W_{z,x}^{k} + \Delta \boldsymbol{W}_{z,x}^{k} = W_{z,x}^{k} - \eta \frac{\partial E}{\partial \boldsymbol{W}_{z,x}^{k}} \tag{30}$$

$$W_{r,h}^{k+1} = W_{r,h}^{k} + \Delta \boldsymbol{W}_{r,h}^{k} = W_{r,h}^{k} - \eta \frac{\partial E}{\partial \boldsymbol{W}_{r,h}^{k}} \tag{31}$$

$$W_{r,x}^{k+1} = W_{r,x}^{k} + \Delta \boldsymbol{W}_{r,x}^{k} = W_{r,x}^{k} - \eta \frac{\partial E}{\partial \boldsymbol{W}_{r,x}^{k}} \tag{32}$$

$$W_{\tilde{h},r}^{k+1} = W_{\tilde{h},r}^{k} + \Delta \boldsymbol{W}_{\tilde{h},r}^{k} = W_{\tilde{h},r}^{k} - \eta \frac{\partial E}{\partial \boldsymbol{W}_{\tilde{h},r}^{k}} \tag{33}$$

$$W_{\tilde{h},x}^{k+1} = W_{\tilde{h},x}^{k} + \Delta \boldsymbol{W}_{\tilde{h},x}^{k} = W_{\tilde{h},x}^{k} - \eta \frac{\partial E}{\partial \boldsymbol{W}_{\tilde{h},x}^{k}} \tag{34}$$

$$W_{\text{out}}^{k+1} = W_{\text{out}}^{k} + \Delta \boldsymbol{W}_{\text{out}}^{k} = W_{\text{out}}^{k} - \eta \frac{\partial E}{\partial \boldsymbol{W}_{\text{out}}^{k}} \tag{35}$$

Based on the above analysis, the SL1GRU algorithm flow is presented in Algorithm 1.

Algorithm 1: SL1-GRU algorithm

```
   Input:
     A training set of sequences and targets $\{(x^{(m)}, y^{(m)})\}_{m=1}^{M}$
     GRU parameters: $\{W_{z,x}, W_{z,h}, b_z, W_{r,x}, W_{z,h}, b_r, W_h, W_{\tilde{h}}, b_h\}$
     Output weight matrix $W_{\text{out}}$
     Initial hidden state $h_0$
     Learning rate $\eta$, smoothed $L_1$ coefficient $\lambda$, threshold $\alpha$
     Number of training epochs $E$
   Function: SmoothL_{1}Regularization $(W_{\text{out}}, \alpha)$
     Initialize: $r \leftarrow 0$
     For each element $W_{\text{out}}^{i,j}$:
       if $|W_{\text{out}}^{i,j}| \geq \alpha$:
         $r \leftarrow r + |W_{\text{out}}^{i,j}|$
       else:
         $r \leftarrow r + \left(\frac{|W_{\text{out}}^{i,j}|^2}{2\alpha} + \frac{\alpha}{2}\right)$
     return $r$
  for epoch = 1 to $E$ do
    Initialize $epoch\_loss \leftarrow 0$
```

(Continued)

Algorithm 1 (continued)

```
    for $m$ = 1 to $M$ do
        /* Forward Pass for sample $m$ */
        $h_0 \leftarrow \mathbf{0}$ (if not provided otherwise)
        for $t$ = 1 to $T$ do
            $z_t \leftarrow \sigma\left(W_{z,x}\, x_t^{(m)} + W_{z,h}\, h_{t-1} + b_z\right)$
            $r_t \leftarrow \sigma\left(W_{r,x}\, x_t^{(m)} + W_{z,h}\, h_{t-1} + b_r\right)$
            $\tilde{h}_t \leftarrow \tanh\left(W_h\, x_t^{(m)} + W_{\tilde{h}}\,(r_t \circ h_{t-1}) + b_h\right)$
            $h_t \leftarrow (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t$
        end for
        $\hat{y}^{(m)} \leftarrow W_{\text{out}} \cdot h_T$
        $\ell_{\text{data}}^{(m)} \leftarrow \text{Loss}\left(\hat{y}^{(m)}, y^{(m)}\right)$
        $\ell_{SL1} \leftarrow \lambda \times \textsc{Smooth } L_1 \textsc{ Regularization}\left(W_{\text{out}}, \alpha\right)$
        $L^{(m)} \leftarrow \ell_{\text{data}}^{(m)} + \ell_{SL1}$
        $\nabla_\theta L^{(m)} \leftarrow \text{BackwardPass}\left(L^{(m)}\right)$
        $\theta \leftarrow \theta - \eta \cdot \nabla_\theta L^{(m)}$
        $epoch\_loss\ epoch\_loss + L^{(m)}$
    end for
end for
return Trained parameters $(W_{z,x}, W_{z,h}, b_z, W_{r,x}, W_{z,h}, b_r, W_h, W_{\tilde{h}}, b_h, W_{\text{out}})$
```

3 Convergence Analysis

This section presents the theoretical findings of GRU networks with smoothing L_1 regularization, with detailed proofs available in Appendix A. To ensure the validity and correctness of the proposed statements and conclusions, the following mild assumptions are made:

(A1) For $r \in \mathbb{R}$, $|\sigma(r)|$, $|\sigma'(r)|$, $|\sigma''(r)|$, $|\tanh(r)|$, $\left|\tanh'(r)\right|$, and $|\tan''(r)|$ are uniformly bounded.

(A2) λ and η are chosen to meet the conditions of $0 < \eta < \frac{2(1+D_4)}{\lambda C + 4D_4 + 2D_5}$, where D_4 and D_5 are constants defined in below.

(A3) There exists a bounded region $\Omega \subset \mathbb{R}^n$ such that $\left\{w_{\text{out}}^k\right\}_{k=0}^{\infty} \subset \Omega$.

(A4) A compact set ϕ_0 exists where $W^k \in \phi_0$, and the set $\phi_1 = \left\{W \in \phi_0 : \frac{\partial E}{\partial W} = 0\right\}$ includes only a finite number of points.

Our main results are as follows:

Theorem 1. *Monotonicity*

Assume the error function $E(W)$ is given as in Eq. (15). Consider the sequence of weights W^k produced by the iterative algorithm detailed in Eq. (17), with an arbitrary initial weight W^0. Under the assumptions (A1)–(A3), the following monotonicity property holds:

$$E\left(W^{k+1}\right) \leq E\left(W^k\right), \quad \text{for } k = 0, 1, 2, \ldots. \tag{36}$$

Theorem 2. *Weak Convergence*

Assuming that conditions (A1)–(A3) hold, then the weight sequence W_k generated by (17) is weak convergent, as evidenced by the following equation:

$$\lim_{k\to+\infty}\left\|\frac{\partial E}{\partial W^k}\right\| = 0 \tag{37}$$

Theorem 3. *Strong Convergence*

Furthermore, if assumption (A4) also holds, the subsequent strong convergence outcome can be derived:

$$\lim_{k\to\infty}\left(W^k\right) = W^* \tag{38}$$

where $W^* \in \phi_0$.

For clarity and convenience, certain notations will be introduced for future reference.

$$\begin{aligned} D_0 &= \max\nolimits_{1\le n\le N}\left\{\left\|x_t^n\right\|, \left\|h_{t-1}^n\right\|\right\} \\ D_1 &= \max\left\{\sup_{r\in\mathbb{R}}|\sigma(r)|, \sup_{r\in\mathbb{R}}|\sigma'(r)|, \sup_{r\in\mathbb{R}}|\sigma''(r)|, \sup_{r\in\mathbb{R},1\le n\le N}|\sigma_n'(r)|, \sup_{r\in\mathbb{R}}|\tanh(r)|, \sup_{r\in\mathbb{R}}\left|\tanh'(r)\right|,\right. \\ &\qquad \left.\sup_{r\in\mathbb{R}}\left|\tanh''(r)\right|, \sup_{r\in\mathbb{R},1\le n\le N}\left|\tanh_n'(r)\right|\right\}, \\ D_2 &= \max\left\{\left\|w_{out}^k\right\|\right\}. \end{aligned} \tag{39}$$

4 Experimental Results and Analysis

The experiment is divided into three distinct parts. The initial part involves an analysis of theoretical outcomes through the approximation of function. Subsequently, the generalization capability and sparsity of the model are evaluated using regression and classification datasets from the UCI Machine Learning Repository.

4.1 Function Approximation

To demonstrate the generalization capabilities of SL1-GRU, we approximate a one-dimensional function $f(x)$ and a two-dimensional function $q(x,y)$ in this section. The mathematical expressions of these functions are as follows:

Nonlinear oscillatory function:

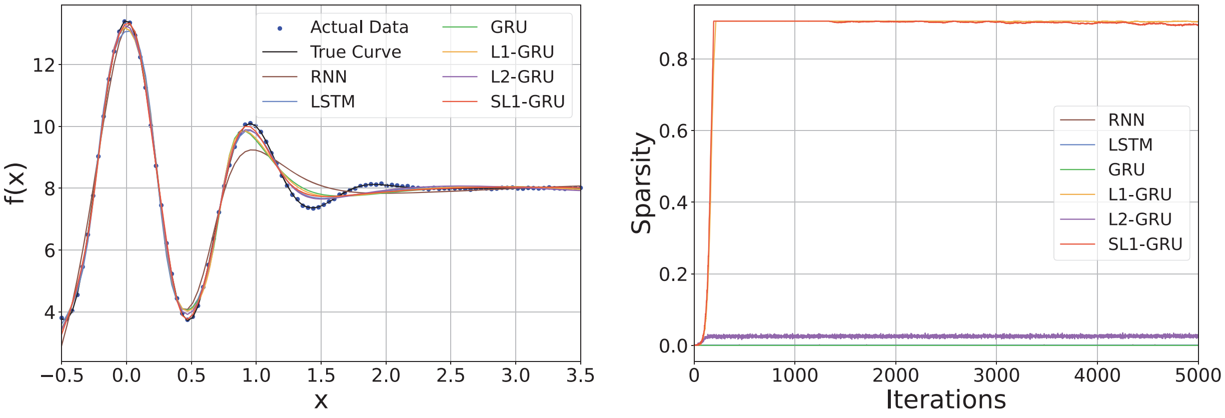
$$f(x) = 8 + 2e^{1-x^2}\cos(2\pi x), \quad x \in [-0.5, 3.5] \tag{40}$$

The peaks function, commonly used in numerical experiments, defined as:

$$q(x,y) = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10\left(\frac{x}{5} - x^3 - y^5\right)e^{-x^2-y^2} - \frac{1}{3}e^{-(x+1)^2-y^2}, \quad x, y \in [-2.5, 2.5] \tag{41}$$

For the nonlinear oscillatory function (40), 100 points are uniformly distributed in the interval $[-0.5, 3.5]$ and denoted as x_i for $i = 1, 2, \ldots, 100$, serving as inputs. The corresponding outputs are given by $f(x_i) + \varepsilon_i$, where $\varepsilon_i \sim N(0, 0.01)$. For the peaks function (41), a two-dimensional grid is generated with x, y uniformly sampled within $[-2.5, 2.5]$, resulting in 100 sample points. The outputs are perturbed by Gaussian noise $\varepsilon_{i,j} \sim N(0, 0.01)$, yielding $q(x_i, y_j) + \varepsilon_{i,j}$. The network weights of six models (RNN, LSTM, GRU, L1-GRU, L2-GRU, SL1-GRU) are initialized randomly in $[-0.5, 0.5]$, with the learning rate η set to 0.001. Regularization coefficients for L1-GRU, L2-GRU and SL1-GRU are $\lambda = 0.0005$, while the smoothing parameter for SL1-GRU is $\alpha = 0.01$.

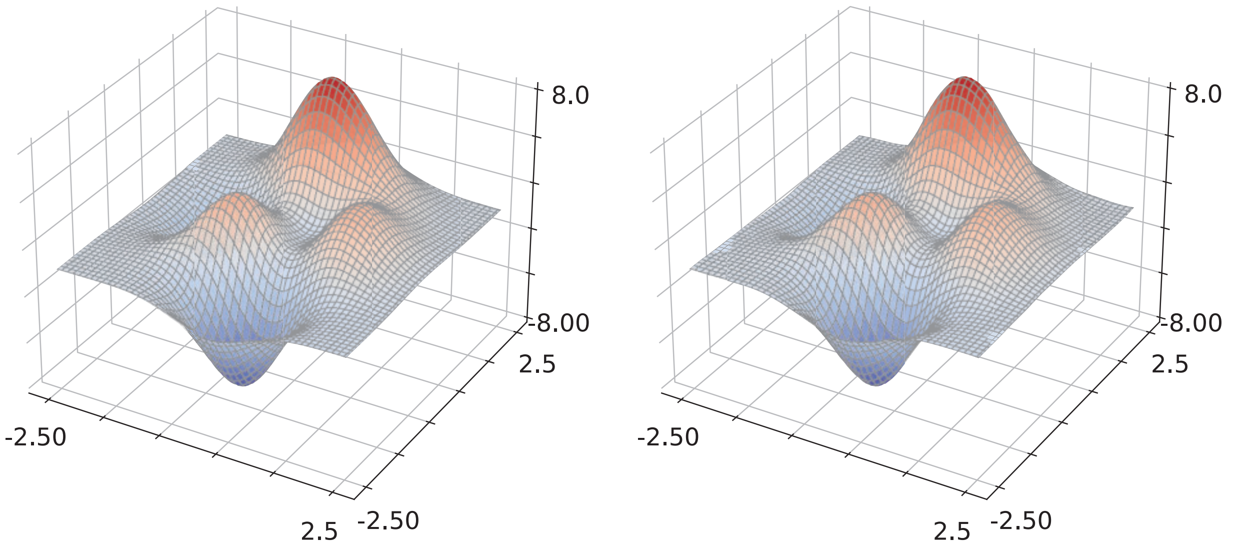
Fig. 3a shows the approximation performance of RNN, LSTM, GRU, L1-GRU, L2-GRU, and SL1-GRU for the nonlinear target function $f(x)$ in $[0.5, 3.5]$. Regularized models (L1-GRU, L2-GRU, SL1-GRU) align more closely with the actual curve, with SL1-GRU achieving the best accuracy in oscillatory regions, highlighting its robustness in capturing nonlinear dynamics. Fig. 3b illustrates the sparsity evolution over training iterations. L1-GRU and SL1-GRU achieve significantly higher sparsity, stabilizing around 0.8 after 2000 iterations, while GRU and LSTM show lower sparsity, reflecting greater parameter complexity. These results demonstrate the effectiveness of regularization in promoting model sparsity.



(a) (b)

Figure 3: Approximation perfomance for one-dimensional function (a) results of approximation (b) sparsity of models

Similarly, we approximate the two-dimensional function using the same approaches, with the approximation results of SL1-GRU presented in Fig. 4. The results highlight SL1-GRU's ability to effectively capture global trends and local variations.



(a) (b)

Figure 4: Approximation result of SL1-GRU for two-dimensional function (a) two-dimensional function (b) approximation function

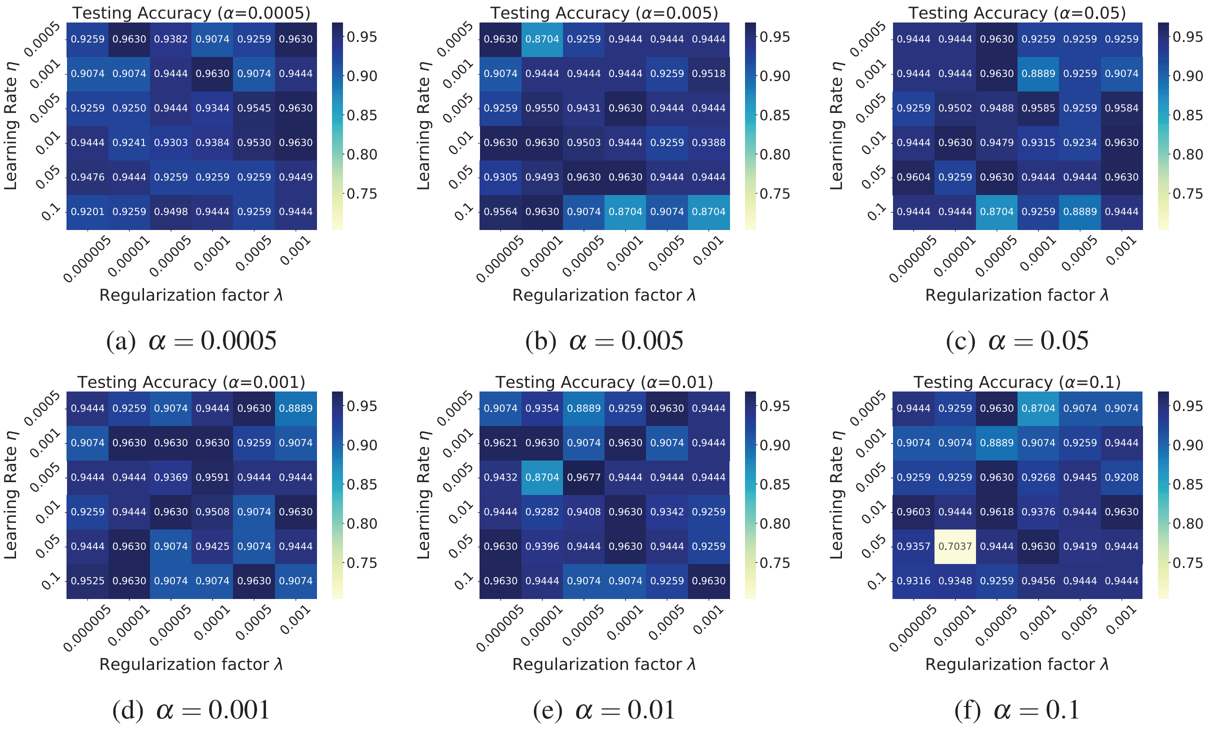
4.2 Classification Problem

This part presents an evaluation and comparison of the classification efficacy for RNN, LSTM, GRU, L1-GRU, L2-GRU, and SL1-GRU. Table 1 is a summary of the dataset utilized in the simulation experiment. The network weights are randomly initialized in $[-0.5, 0.5]$. Each network is set up with a hidden layer of 32 nodes. The dataset's features determine the input layer's node count, while the number of output layer nodes equals the count of classes.

Table 1: Details of the classification data sets

Dataset	Instances	Features	Classes	Training set	Test set
Iris	150	4	3	105	45
Wine	178	13	3	125	53
Breast cancer	286	9	2	200	86
KDD Cup 1999 Data (10%)	494019	22	23	444617	49402
MNIST	70000	784	10	60000	10000

As shown in Fig. 5, we use grid search to explore the hyperparameter space by testing combinations of learning rate η, regularization factor λ, and smoothing coefficient α within predefined ranges. Each combination of these hyperparameters is evaluated using k-fold cross-validation to ensure robust and reliable performance metrics. The evaluation criterion is based on the test accuracy achieved by SL1-GRU, aiming to identify the parameter set that maximizes accuracy while maintaining generalization. It is determined that $\{\alpha = 0.01, \lambda = 0.00005, \eta = 0.005\}$ constitutes the optimal parameter combination for the wine dataset, achieving the highest test accuracy for SL1-GRU. This approach is similarly applied to other datasets, and the results, summarized in Table 2, highlight the effectiveness of grid search in identifying optimal hyperparameters.



(a) $\alpha = 0.0005$

(b) $\alpha = 0.005$

(c) $\alpha = 0.05$

(d) $\alpha = 0.001$

(e) $\alpha = 0.01$

(f) $\alpha = 0.1$

Figure 5: Test accuracy of SL1-GRU on the wine dataset under different parameter combinations; α is regularization coefficient

Table 2: Training parameters for the classification data sets

Dataset	η	λ	α	Batch size
Iris	5×10^{-3}	5×10^{-4}	1×10^{-2}	32
Wine	5×10^{-4}	1×10^{-5}	1×10^{-2}	32
Breast cancer	5×10^{-5}	5×10^{-4}	1×10^{-2}	32
KDD Cup 1999 Data (10%)	1×10^{-3}	5×10^{-4}	1×10^{-2}	128
MNIST	1×10^{-3}	5×10^{-4}	1×10^{-2}	32

Table 3 compares the training accuracy, test accuracy, sparsity, and training time of different models on the same dataset. These experimental results represent the average values obtained over 10 trials. Sparsity, defined as the ratio of elements in the neural network's weight matrix that are less than 1×10^{-5} to the total number of elements in the weight matrix, is used as an indicator of network sparsity. Mathematically, it can be expressed as:

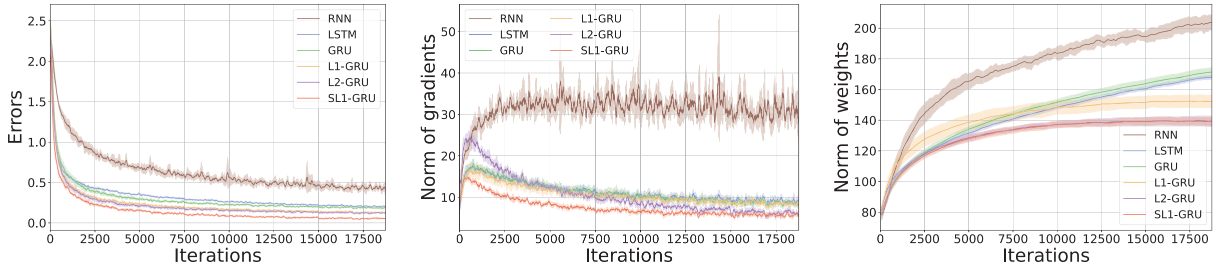
$$\text{Sparsity} = \frac{Num_0}{Num_n} \tag{42}$$

where the number of elements in the weight matrix that are less than 1×10^{-5} is denoted by Num_0, and Num_n represents the overall element count of the matrix. It can be observed in Table 3 that although the training accuracy of SL1-GRU may not be the highest, its test accuracy is consistently the best across all datasets, highlighting its excellent generalization ability. Moreover, both L1-GRU and SL1-GRU exhibit significantly higher sparsity compared to other models. Except for one dataset, SL1-GRU achieves the highest sparsity, demonstrating that the proposed method effectively enhances network sparsity. Additionally, benefiting from its superior sparsity, SL1-GRU requires the shortest training time, indicating that it significantly improves computational efficiency.

Table 3: Evaluation indicators of classification result (The bold entries indicate the optimal performance)

Dataset	Model	Training accuracy	Test accuracy	Sparsity	Training time (s)
Iris	RNN	0.9609	0.9420	3.2000×10^{-6}	19.42
	LSTM	0.9877	0.9510	3.2124×10^{-5}	17.33
	GRU	**0.9896**	0.9556	9.7653×10^{-5}	16.58
	L1-GRU	0.9772	0.9623	0.0676	13.96
	L2-GRU	0.9798	0.9633	0.0034	15.96
	SL1-GRU	0.9827	**0.9656**	**0.0681**	**13.24**
Wine	RNN	0.9545	0.9202	9.2020×10^{-7}	25.35
	LSTM	0.9739	0.9487	5.2028×10^{-5}	25.63
	GRU	0.9739	0.9441	3.9004×10^{-5}	21.35
	L1-GRU	**0.9829**	0.9630	**0.0220**	19.62
	L2-GRU	0.9801	0.9599	0.0030	20.84
	SL1-GRU	0.9798	**0.9677**	0.0194	**18.09**
Breast cancer	RNN	0.9589	0.9265	1.3650×10^{-5}	34.02
	LSTM	0.9874	0.9559	3.5896×10^{-5}	33.25
	GRU	**0.9896**	0.9556	4.2955×10^{-5}	29.56
	L1-GRU	0.9732	0.9623	0.1177	25.76
	L2-GRU	0.9754	0.9633	0.0098×10^{-5}	29.03
	SL1-GRU	0.9739	**0.9634**	**0.1072**	**24.89**
KDD Cup 1999 Data (10%)	RNN	0.9689	0.9499	1.9428×10^{-7}	2465.62
	LSTM	0.9787	0.9687	3.0010×10^{-6}	2438.51
	GRU	0.9784	0.9659	2.3743×10^{-6}	2203.45
	L1-GRU	0.9813	0.9716	0.1843	1984.63
	L2-GRU	**0.9843**	0.9772	0.0125	2179.00
	SL1-GRU	0.9802	**0.9775**	**0.1895**	**1907.63**
MNIST	RNN	0.9583	0.9230	2.2541×10^{-6}	303.60
	LSTM	0.9698	0.9532	4.3253×10^{-6}	278.94
	GRU	0.9781	0.9542	4.3650×10^{-6}	243.77
	L1-GRU	0.9748	0.9627	0.0458	211.77
	L2-GRU	0.9795	0.9598	0.0002	241.40
	SL1-GRU	**0.9798**	**0.9689**	**0.0489**	**201.50**

From Fig. 6, it can be observed that the loss function curve of SL1-GRU monotonically decreases and gradually stabilizes at zero as the number of iterations increases, which verifies Theorem 1. Meanwhile, in Fig. 6b, the gradient curve of SL1-GRU decreases the fastest, and as the number of iterations approaches infinity, its gradient also tends to zero, consistent with Theorem 2. Fig. 6c shows that the weight curves of L1-GRU and SL1-GRU do not grow indefinitely, indicating that both regularization methods effectively suppress weight growth. Among them, SL1-GRU is more effective in constraining network weights, stabilizing them around a constant value of approximately 140, aligning with Theorem 3.



(a) Curves of loss (b) Curves of norm of gradient (c) Curves of norm of weight

Figure 6: The performance of RNN, LSTM, GRU, L1-GRU, L2-GRU and SL1-GRU on MNIST dataset; the shaded area presents the mean ± the standard deviation over 10 trials

4.3 Regression Problem

The performance of SL1-GRU in regression tasks is also considered. The dataset utilized in this part is detailed in Table 4. For RNN, LSTM, GRU, L1-GRU, L2-GRU, and SL1-GRU, the hidden layer is designed with 32 nodes. The nodes in both the input and output layers are configured based on the dataset's features and labels, respectively. The learning rate is established at $\eta = 1 \times 10^{-3}$, the regularization factor at $\lambda = 3 \times 10^{-4}$, and the smoothing coefficient at $\alpha = 0.01$. The initial weight range is $[-0.5, 0.5]$ as in the previous part.

Table 4: Details of the regression data sets

Dataset	Instances	Features	Training set	Test set
Boston housing	506	13	354	152
Diabetes	442	10	309	133
Wine quality	6497	11	5198	1299
Energy efficiency	768	8	614	154
Student performance	649	30	519	130

In the evaluation of regression models, the standard metric used is Mean Squared Error (MSE), which is calculated using the following formula:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(pred_i - true_i)^2, \tag{43}$$

where $pred_1, pred_2, \ldots, pred_n$ indicate the predicted values, and the set of actual values is denoted by $true_1, true_2, \ldots, true_n$.

Table 5 shows that the Test MSE of SL1-GRU is consistently the smallest, indicating that it performs the best on the test set and has the strongest generalization ability. From the perspective of sparsity, the network weights of SL1-GRU remain the sparsest, which suggests that it eliminates unimportant parameters to enhance the computational efficiency of the model while maintaining its excellent performance.

Table 5: Evaluation indicators of regression result (The bold entries indicate the optimal performance)

Dataset	Models	Training MSE	Test MSE	Sparsity	Training time (s)
Boston housing	RNN	3.3659×10^{-4}	0.5001	8.5728×10^{-5}	36.53
	LSTM	1.8025×10^{-4}	0.1720	2.0008×10^{-5}	34.32
	GRU	1.5778×10^{-4}	0.1749	1.2787×10^{-5}	31.70
	L1-GRU	$\mathbf{2.6969 \times 10^{-5}}$	0.1446	0.4379	29.32
	L2-GRU	4.7778×10^{-5}	0.1639	2.1054×10^{-4}	31.54
	SL1-GRU	5.9140×10^{-5}	**0.1446**	**0.4382**	**28.98**
Diabetes	RNN	0.1023	3.2653	3.2451×10^{-7}	33.65
	LSTM	0.0398	0.0602	2.5778×10^{-6}	31.96
	GRU	0.0482	0.0641	1.8440×10^{-6}	31.75
	L1-GRU	0.0481	0.0534	5.3077×10^{-3}	28.33
	L2-GRU	0.0385	0.0632	1.2382×10^{-4}	30.88
	SL1-GRU	**0.0364**	**0.0528**	$\mathbf{5.4721 \times 10^{-3}}$	**28.32**
Wine quality	RNN	0.1895	0.7521	1.0078×10^{-6}	189.96
	LSTM	0.0229	0.6002	5.5248×10^{-6}	180.23
	GRU	0.0253	0.5969	8.9947×10^{-6}	168.41
	L1-GRU	**0.0207**	0.5902	0.4729	157.33
	L2-GRU	0.0222	0.5898	0.0342	167.39
	SL1-GRU	0.0235	**0.5862**	**0.4784**	**152.63**
Energy efficiency	RNN	1.7029	5.6548	3.7448×10^{-7}	62.63
	LSTM	0.4961	3.7296	1.2036×10^{-5}	58.66
	GRU	0.5154	3.9230	8.9947×10^{-6}	53.02
	L1-GRU	0.4543	3.1730	0.4729	48.50
	L2-GRU	0.4652	3.2356	4.8124×10^{-5}	53.29
	SL1-GRU	**0.4359**	**2.9928**	**0.4770**	**46.58**
Student performance	RNN	0.0447	0.1085	5.3696×10^{-6}	60.38
	LSTM	0.0201	0.0742	3.2778×10^{-4}	57.20
	GRU	0.0213	0.0765	4.1212×10^{-5}	51.90
	L1-GRU	**0.0180**	0.0325	0.1034	45.32
	L2-GRU	0.0185	0.0478	0.0321	50.23
	SL1-GRU	0.0192	**0.0331**	**0.1298**	**43.30**

5 Conclusions

This article proposes a GRU with smoothing L_1 regularization to address the issue of non-differentiability at the origin inherent in traditional L_1 regularization. This approach also aims to enhance the network sparsity and generalization capability. We theoretically demonstrate the monotonicity, weak convergence, and strong convergence of SL1-GRU in backpropagation algorithms and design simulation experiments to compare SL1-GRU with RNN, LSTM, GRU, L1-GRU, and L2-GRU. The simulation results align with the theoretical analysis, demonstrating that SL1-GRU effectively curbs excessive weight growth, reduces the risk of overfitting, and enhances the network's generalization capability. In addition, SL1-GRU also performs well in handling classification and regression problems on real-world datasets, indicating its usability in practical problems. Future work will focus on conducting theoretical analysis under more relaxed assumptions. Furthermore, we will investigate whether dynamically adjusting the smoothing coefficients can further optimize model performance. For example, the smoothing coefficients could be adaptively adjusted based on gradient changes during training.

Acknowledgement: We would like to thank the editors and reviewers for their valuable work.

Funding Statement: This work was supported by the National Science Fund for Distinguished Young Scholarship (No. 62025602), National Natural Science Foundation of China (Nos. U22B2036, 11931015), the Fok Ying-Tong Education Foundation China (No. 171105), the Fundamental Research Funds for the Central Universities (No. G2024WD0151) and in part by the Tencent Foundation and XPLORER PRIZE.

Author Contributions: Qian Zhu: Conceptualization, Software, Writing—review & editing. Qian Kang: Data curation, Writing—review. Tao Xu: Conceptualization, Validation, Methodology. Dengxiu Yu: Methodology, Supervision, Validation. Zhen Wang: Supervision, Validation. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Appendix A Detailed Proof

Lemma A1. *The function $f(x)$ is specified over a closed and bounded $[a, b]$, with its derivative $f'(x)$ being Lipschitz continuous, constant $c > 0$. Then the following equationas holds:*

$$f(x) \leq f(x_0) + f'(x_0)(x - x_0) + \frac{c}{2}(x - x_0)^2, \forall x_0, x \in [a, b] \tag{A1}$$

Proof of Lemma A1. □

A new function is constructed as:

$$g(x) = f(x) - f(x_0) - f'(x_0)(x - x_0) - \frac{c}{2}(x - x_0)^2 \tag{A2}$$

where c denotes a positive constant.

Taking the derivative of x

$$g'(x) = f'(x) - f'(x_0) - c(x - x_0) \tag{A3}$$

$$|f(x) - f'(x_0)| \leq c|x - x_0| \tag{A4}$$

$$\begin{cases} \mathrm{g}'(x) \leq 0, x \geq x_0 \\ \mathrm{g}'(x) \geq 0, x < x_0 \end{cases} \tag{A5}$$

then,

$$g(x) \leq g(x_0) = 0 \tag{A6}$$

Proof of Theorem 1. □

By (15), the errors at the k-th and (k+1)-th iterations are given as:

$$E^{k+1} = \sum_{n=1}^{N} \sigma_n(W_{out}^{k+1} \cdot h_t^{k+1,n}) + \lambda L_1(W_{out}^{k+1}) \tag{A7}$$

$$E^k = \sum_{n=1}^{N} \sigma_n(W_{out}^{k} \cdot h_t^{k,n}) + \lambda L_1(W_{out}^{k}) \tag{A8}$$

and the difference between them is:

$$\begin{aligned} E^{k+1} - E^k &= \sum_{n=1}^{N} \sigma_n(W_{out}^{k+1} \cdot h_t^{k+1,n}) + \lambda L_1(W_{out}^{k+1}) - [\sum_{n=1}^{N} \sigma_n(W_{out}^{k} \cdot h_t^{k,n}) + \lambda L_1(W_{out}^{k})] \\ &= \sum_{n=1}^{N} [\sigma_n(W_{out}^{k+1} \cdot h_t^{k+1,n}) - \sigma_n(W_{out}^{k} \cdot h_t^{k,n})] + \lambda[L_1(W_{out}^{k+1}) - L_1(W_{out}^{k})] \\ &= \sum_{n=1}^{N} [\sigma'_n(W_{out}^{k} \cdot h_t^{k,n})(W_{out}^{k+1} \cdot h_t^{k+1,n} - W_{out}^{k} \cdot h_t^{k,n})] + R_1 + \lambda[L_1(W_{out}^{k+1}) - L_1(W_{out}^{k})] \\ &= \sum_{n=1}^{N} \sigma'_n(W_{out}^{k} \cdot h_t^{k,n})\Delta W_{out}^{k} \cdot h_t^{k,n} + \sum_{n=1}^{N} \sigma'_n(W_{out}^{k} \cdot h_t^{k,n}) W_{out}^{k} \cdot \Delta h_t^{k,n} \\ &\quad + \sum_{n=1}^{N} \sigma'_n(W_{out}^{k} \cdot h_t^{k,n})\Delta W_{out}^{k} \cdot \Delta h_t^{k,n} + R_1 + \lambda[L_1(W_{out}^{k+1}) - L_1(W_{out}^{k})] \\ &= \sum_{n=1}^{N} \sigma'_n(W_{out}^{k} \cdot h_t^{k,n})\Delta W_{out}^{k} \cdot h_t^{k,n} + \lambda[L_1(W_{out}^{k+1}) - L_1(W_{out}^{k})] \\ &\quad + \sum_{n=1}^{N} \sigma'_n(W_{out}^{k} \cdot h_t^{k,n}) W_{out}^{k} \cdot \Delta h_t^{k,n} + \sum_{n=1}^{N} \sigma'_n(W_{out}^{k} \cdot h_t^{k,n})\Delta W_{out}^{k} \cdot \Delta h_t^{k,n} + R_1 \end{aligned} \tag{A9}$$

where Lagrange remainder

$$R_1 = \frac{1}{2} \sum_{n=1}^{N} \sigma''(s_{k,n})(W_{out}^{k+1} h_t^{k+1,n} - W_{out}^{k} h_t^{k,n})^2 \tag{A10}$$

in the above equation, $s_{k,n}$ is a constant between $W_{out}^{k+1} h_t^{k+1,n}$ and $W_{out}^{k} h_t^{k,n}$.

To simplify, we use the following notation:

$$A_1 = \sum_{n=1}^{N} \sigma'_n(W_{out}^{k} \cdot h_t^{k,n})\Delta W_{out}^{k} \cdot h_t^{k,n} + \lambda[L_1(W_{out}^{k+1}) - L_1(W_{out}^{k})] \tag{A11}$$

$$A_2 = \sum_{n=1}^{N} \sigma'_n(W_{out}^{k} \cdot h_t^{k,n}) W_{out}^{k} \cdot \Delta h_t^{k,n} \tag{A12}$$

$$A_3 = \sum_{n=1}^{N} \sigma'_n(W_{out}^{k} \cdot h_t^{k,n})\Delta W_{out}^{k} \cdot \Delta h_t^{k,n} \tag{A13}$$

According to Lemma A1,

$$\begin{aligned} \lambda[L_1(W_{out}^{k+1}) - L_1(W_{out}^{k})] &\leq \lambda[L_1{}'(W_{out}^{k})[L_1(W_{out}^{k+1}) - L_1(W_{out}^{k})] + \frac{c}{2}[L_1(W_{out}^{k+1}) - L_1(W_{out}^{k})]^2] \\ &\leq \lambda L_1{}'(W_{out}^{k})\Delta L_1(W_{out}^{k}) + \frac{\lambda c}{2}[\Delta L_1(W_{out}^{k})]^2 \end{aligned} \tag{A14}$$

Using transition variables

$$\begin{aligned} \Delta h_t^{k,n} &= h_t^{k+1,n} - h_t^{k,n} \\ &= [(1 - z_t^{k+1,n}) \circ h_{t-1}^n + z_t^{k+1,n} \circ \tilde{h}_t^{k+1,n}] - [(1 - z_t^{k,n}) \circ h_{t-1}^n + z_t^{k,n} \circ \tilde{h}_t^{k,n}] \\ &= z_t^{k+1,n} \circ \tilde{h}_t^{k+1,n} - z_t^{k,n} \circ \tilde{h}_t^{k,n} + (1 - z_t^{k+1,n}) \circ h_{t-1}^n - (1 - z_t^{k,n}) \circ h_{t-1}^n \\ &= (z_t^{k+1,n} - z_t^{k,n}) \circ (\tilde{h}_t^{k+1,n} - \tilde{h}_t^{k,n}) + (z_t^{k+1,n} - z_t^{k,n}) \circ \tilde{h}_t^{k,n} + z_t^{k,n} \circ (\tilde{h}_t^{k+1,n} - \tilde{h}_t^{k,n}) \\ &\quad - (z_t^{k+1,n} - z_t^{k,n}) \circ h_{t-1}^n \\ &= [\sigma(W_{z,h}^{k+1,n} \cdot h_{t-1}^n + W_{z,x}^{k+1,n} \cdot x_t^n) - \sigma(W_{z,h}^{k,n} \cdot h_{t-1}^n + W_{z,x}^{k,n} \cdot x_t^n)] \\ &\quad \circ [\tanh(W_{\tilde{h},r}^{k+1,n} \cdot (r_t^{k+1,n} \circ h_{t-1}^n) + W_{\tilde{h},x}^{k+1,n} \cdot x_t^n) - \tanh(W_{\tilde{h},r}^{k,n} \cdot (r_t^{k,n} \circ h_{t-1}^n) + W_{\tilde{h},x}^{k,n} \cdot x_t^n)] \\ &\quad + [\sigma(W_{z,h}^{k+1,n} \cdot h_{t-1}^n + W_{z,x}^{k+1,n} \cdot x_t^n) - \sigma(W_{z,h}^{k,n} \cdot h_{t-1}^n + W_{z,x}^{k,n} \cdot x_t^n)] \circ \tilde{h}_t^{k,n} \\ &\quad + z_t^{k,n} \circ [\tanh(W_{\tilde{h},r}^{k+1,n} \cdot (r_t^{k+1,n} \circ h_{t-1}^n) + W_{\tilde{h},x}^{k+1,n} \cdot x_t^n) - \tanh(W_{\tilde{h},r}^{k,n} \cdot (r_t^{k,n} \circ h_{t-1}^n) + W_{\tilde{h},x}^{k,n} \cdot x_t^n)] \\ &\quad - [\sigma(W_{z,h}^{k+1,n} \cdot h_{t-1}^n + W_{z,x}^{k+1,n} \cdot x_t^n) - \sigma(W_{z,h}^{k,n} \cdot h_{t-1}^n + W_{z,x}^{k,n} \cdot x_t^n)] \circ h_{t-1}^n \\ &= [\sigma'(\xi_{zh})(W_{z,h}^{k+1,n} \cdot h_{t-1}^n - W_{z,h}^{k,n} \cdot h_{t-1}^n) + \sigma'(\xi_{zx})(W_{z,x}^{k+1,n} \cdot x_t^n - W_{z,x}^{k,n} \cdot x_t^n)] \\ &\quad \circ [\tanh'(\xi_{\tilde{h}r})(W_{\tilde{h},r}^{k+1,n} \cdot (r_t^{k+1,n} \circ h_{t-1}^n) - W_{\tilde{h},r}^{k,n} \cdot (r_t^{k,n} \circ h_{t-1}^n)) + \tanh'(\xi_{\tilde{h}x})(W_{\tilde{h},x}^{k+1,n} \cdot x_t^n \\ &\quad - W_{\tilde{h},x}^{k,n} \cdot x_t^n)] + [\sigma'(\xi_{zh})(W_{z,h}^{k+1,n} \cdot h_{t-1}^n - W_{z,h}^{k,n} \cdot h_{t-1}^n) + \sigma'(\xi_{zx})(W_{z,x}^{k+1,n} \cdot x_t^n - W_{z,x}^{k,n} \cdot x_t^n)] \circ \tilde{h}_t^{k,n} \\ &\quad + z_t^{k,n} \circ [\tanh'(\xi_{\tilde{h}r})(W_{\tilde{h},r}^{k+1,n} \cdot (r_t^{k+1,n} \circ h_{t-1}^n) - W_{\tilde{h},r}^{k,n} \cdot (r_t^{k,n} \circ h_{t-1}^n)) + \tanh'(\xi_{\tilde{h}x})(W_{\tilde{h},x}^{k+1,n} \cdot x_t^n \\ &\quad - W_{\tilde{h},x}^{k,n} \cdot x_t^n)] - [\sigma'(\xi_{zh})(W_{z,h}^{k+1,n} \cdot h_{t-1}^n - W_{z,h}^{k,n} \cdot h_{t-1}^n) + \sigma'(\xi_{zx})(W_{z,x}^{k+1,n} \cdot x_t^n - W_{z,x}^{k,n} \cdot x_t^n)] \circ h_{t-1}^n \\ &= [\sigma'(\xi_{zh})(\Delta W_{z,h}^{k,n} \cdot h_{t-1}^n) + \sigma'(\xi_{zx})(\Delta W_{z,x}^{k+1,n} \cdot x_t^n)] \circ [\tanh'(\xi_{\tilde{h}r})(\Delta W_{\tilde{h},r}^{k,n} \cdot \Delta(r_t^{k,n} \circ h_{t-1}^n) \\ &\quad + W_{\tilde{h},r}^{k,n} \cdot \Delta(r_t^{k,n} \circ h_{t-1}^n) + \Delta W_{\tilde{h},r}^{k,n} \cdot (r_t^{k,n} \circ h_{t-1}^n)) + \tanh'(\xi_{\tilde{h}x})(\Delta W_{\tilde{h},x}^{k,n} \cdot x_t^n)] \\ &\quad + [\sigma'(\xi_{zh})(\Delta W_{z,h}^{k,n} \cdot h_{t-1}^n) + \sigma'(\xi_{zx})(\Delta W_{z,x}^{k,n} \cdot x_t^n)] \circ \tilde{h}_t^{k,n} + z_t^{k,n} \\ &\quad \circ [\tanh'(\xi_{\tilde{h}r})(\Delta W_{\tilde{h},r}^{k,n} \cdot \Delta(r_t^{k,n} \circ h_{t-1}^n) \\ &\quad + W_{\tilde{h},r}^{k,n} \cdot \Delta(r_t^{k,n} \circ h_{t-1}^n) + \Delta W_{\tilde{h},r}^{k,n} \cdot (r_t^{k,n} \circ h_{t-1}^n)) + \tanh'(\xi_{\tilde{h}x})(\Delta W_{\tilde{h},x}^{k,n} \cdot x_t^n)] \\ &\quad - [\sigma'(\xi_{zh})(\Delta W_{z,h}^{k,n} \cdot h_{t-1}^n) + \sigma'(\xi_{zx})(\Delta W_{z,x}^{k+1,n} \cdot x_t^n)] \circ h_{t-1}^n \\ &= A_4 + A_5 + A_6 + A_7 \end{aligned} \tag{A15}$$

where

$$A_4 = (z_t^{k+1,n} - z_t^{k,n}) \circ (\tilde{h}_t^{k+1,n} - \tilde{h}_t^{k,n}) \tag{A16}$$

$$A_5 = (z_t^{k+1,n} - z_t^{k,n}) \circ \tilde{h}_t^{k,n} \tag{A17}$$

$$A_6 = z_t^{k,n} \circ (\tilde{h}_t^{k+1,n} - \tilde{h}_t^{k,n}) \tag{A18}$$

$$A_7 = (z_t^{k+1,n} - z_t^{k,n}) \circ h_{t-1}^n \tag{A19}$$

continuing from the previous step and according to assumption (A1),

$$\begin{aligned}
A_4 &= [\sigma(W^{k+1,n}_{z,h}\cdot h^n_{t-1} + W^{k+1,n}_{z,x}\cdot x^n_t) - \sigma(W^{k,n}_{z,h}\cdot h^n_{t-1} + W^{k,n}_{z,x}\cdot x^n_t)]\\
&\quad \circ [\tanh(W^{k+1,n}_{\tilde{h},r}\cdot(r^{k+1,n}_t \circ h^n_{t-1}) + W^{k+1,n}_{\tilde{h},x}\cdot x^n_t) - \tanh(W^{k,n}_{\tilde{h},r}\cdot(r^{k,n}_t \circ h^n_{t-1}) + W^{k,n}_{\tilde{h},x}\cdot x^n_t)]\\
&= [\sigma'(\xi_{zh})(W^{k+1,n}_{z,h}\cdot h^n_{t-1} - W^{k,n}_{z,h}\cdot h^n_{t-1}) + \sigma'(\xi_{zx})(W^{k+1,n}_{z,x}\cdot x^n_t - W^{k,n}_{z,x}\cdot x^n_t)]\\
&\quad \circ [\tanh'(\xi_{\tilde{h}r})(W^{k+1,n}_{\tilde{h},r}\cdot(r^{k+1,n}_t \circ h^n_{t-1}) W^{k,n}_{\tilde{h},r}\cdot(r^{k,n}_t \circ h^n_{t-1})) + \tanh'(\xi_{\tilde{h}x})(W^{k+1,n}_{\tilde{h},x}\cdot x^n_t - W^{k,n}_{\tilde{h},x}\cdot x^n_t)]\\
&= [\sigma'(\xi_{zh})(\Delta W^{k,n}_{z,h}\cdot h^n_{t-1}) + \sigma'(\xi_{zx})(\Delta W^{k,n}_{z,x}\cdot x^n_t)] \circ [\tanh'(\xi_{\tilde{h}r})(\Delta W^{k,n}_{\tilde{h},r}\cdot(r^{k+1,n}_t \circ h^n_{t-1})\\
&\quad + W^{k,n}_{\tilde{h},r}\cdot\Delta(r^{k,n}_t \circ h^n_{t-1})) + \tanh'(\xi_{\tilde{h}x})(\Delta W^{k,n}_{\tilde{h},x}\cdot x^n_t)]\\
&= [\sigma'(\xi_{zh})(\Delta W^{k,n}_{z,h}\cdot h^n_{t-1}) + \sigma'(\xi_{zx})(\Delta W^{k,n}_{z,x}\cdot x^n_t)] \circ [\tanh'(\xi_{\tilde{h}r})(\Delta W^{k,n}_{\tilde{h},r}\cdot(\sigma(W^{k+1,n}_{r,h}\cdot h^n_{t-1}\\
&\quad + W^{k+1,n}_{r,x}\cdot x^n_t) \circ h^n_{t-1}) + W^{k,n}_{\tilde{h},r}\cdot((\sigma(W^{k+1,n}_{r,h}\cdot h^n_{t-1} + W^{k+1,n}_{r,x}\cdot x^n_t) - \sigma(W^{k,n}_{r,h}\cdot h^n_{t-1} + W^{k,n}_{r,x}\cdot x^n_t))\\
&\quad \circ h^n_{t-1})) + \tanh'(\xi_{\tilde{h}x})(\Delta W^{k,n}_{\tilde{h},x}\cdot x^n_t)]\\
&= [\sigma'(\xi_{zh})(\Delta W^{k,n}_{z,h}\cdot h^n_{t-1}) + \sigma'(\xi_{zx})(\Delta W^{k,n}_{z,x}\cdot x^n_t)] \circ [\tanh'(\xi_{\tilde{h}r})(\Delta W^{k,n}_{\tilde{h},r}\cdot(\sigma(W^{k+1,n}_{r,h}\cdot h^n_{t-1}\\
&\quad + W^{k+1,n}_{r,x}\cdot x^n_t) \circ h^n_{t-1}) + W^{k,n}_{\tilde{h},r}\cdot((\sigma'(\xi_{rh})(\Delta W^{k,n}_{r,h}\cdot h^n_{t-1}) + \sigma'(\xi_{rx})(\Delta W^{k,n}_{r,x}\cdot x^n_t)) \circ h^n_{t-1}))\\
&\quad + \tanh'(\xi_{\tilde{h}x})(\Delta W^{k,n}_{\tilde{h},x}\cdot x^n_t)]\\
&\le [D_0(\Delta W^{k,n}_{z,h}\cdot D_1) + D_0(\Delta W^{k,n}_{z,x}\cdot D_1)] \circ [D_0(\Delta W^{k,n}_{\tilde{h},r}\cdot(D_0 \circ D_1)\\
&\quad + W^{k,n}_{\tilde{h},r}\cdot((D_0(\Delta W^{k,n}_{r,h}\cdot D_1) + D_0(\Delta W^{k,n}_{r,x}\cdot D_1)) \circ D_1)) + D_0(\Delta W^{k,n}_{\tilde{h},x}\cdot D_1)]
\end{aligned} \tag{A20}$$

and

$$\begin{aligned}
A_5 &= [\sigma(W^{k+1,n}_{z,h}\cdot h^n_{t-1} + W^{k+1,n}_{z,x}\cdot x^n_t) - \sigma(W^{k,n}_{z,h}\cdot h^n_{t-1} + W^{k,n}_{z,x}\cdot x^n_t)]\\
&\quad \circ \tanh(W^{k+1,n}_{\tilde{h},r}\cdot(r^{k+1,n}_t \circ h^n_{t-1}) + W^{k+1,n}_{\tilde{h},x}\cdot x^n_t)\\
&= [\sigma(W^{k+1,n}_{z,h}\cdot h^n_{t-1} + W^{k+1,n}_{z,x}\cdot x^n_t) - \sigma(W^{k,n}_{z,h}\cdot h^n_{t-1} + W^{k,n}_{z,x}\cdot x^n_t)] \circ \tanh(W^{k+1,n}_{\tilde{h},r}\\
&\quad \cdot(\sigma(W^{k+1,n}_{r,h}\cdot h^n_{t-1} + W^{k+1,n}_{r,x}\cdot x^n_t) \circ h^n_{t-1}) + W^{k+1,n}_{\tilde{h},x}\cdot x^n_t)\\
&= [\sigma'(\xi_{zh})(\Delta W^{k,n}_{z,h}\cdot h^n_{t-1}) + \sigma'(\xi_{zx})(\Delta W^{k,n}_{z,x}\cdot x^n_t)]\\
&\quad \circ \tanh(W^{k+1,n}_{\tilde{h},r}\cdot(\sigma(W^{k+1,n}_{r,h}\cdot h^n_{t-1} + W^{k+1,n}_{r,x}\cdot x^n_t) \circ h^n_{t-1}) + W^{k+1,n}_{\tilde{h},x}\cdot x^n_t)\\
&\le [D_0(\Delta W^{k,n}_{z,h}\cdot D_1) + D_0(\Delta W^{k,n}_{z,x}\cdot D_1)] \circ D_0
\end{aligned} \tag{A21}$$

and

$$\begin{aligned}
A_6 &= z^{k,n}_t \circ [\tanh(W^{k+1,n}_{\tilde{h},r}\cdot(r^{k+1,n}_t \circ h^n_{t-1}) + W^{k+1,n}_{\tilde{h},x}\cdot x^n_t) - \tanh(W^{k,n}_{\tilde{h},r}\cdot(r^{k,n}_t \circ h^n_{t-1}) + W^{k,n}_{\tilde{h},x}\cdot x^n_t)]\\
&= \sigma(W^{k,n}_{z,h}\cdot h^n_{t-1} + W^{k,n}_{z,x}\cdot x^n_t) \circ [\tanh'(\xi_{\tilde{h}r})(W^{k+1,n}_{\tilde{h},r}\cdot(r^{k+1,n}_t \circ h^n_{t-1}) - W^{k,n}_{\tilde{h},r}\cdot(r^{k,n}_t \circ h^n_{t-1}))\\
&\quad + \tanh'(\xi_{\tilde{h}x})(W^{k+1,n}_{\tilde{h},x}\cdot x^n_t - W^{k,n}_{\tilde{h},x}\cdot x^n_t)]\\
&= \sigma(W^{k,n}_{z,h}\cdot h^n_{t-1} + W^{k,n}_{z,x}\cdot x^n_t) \circ [\tanh'(\xi_{\tilde{h}r})(\Delta W^{k,n}_{\tilde{h},r}\cdot(r^{k+1,n}_t \circ h^n_{t-1}) + W^{k,n}_{\tilde{h},r}\cdot\Delta(r^{k,n}_t \circ h^n_{t-1}))\\
&\quad + \tanh'(\xi_{\tilde{h}x})(\Delta W^{k,n}_{\tilde{h},x}\cdot x^n_t)]\\
&= \sigma(W^{k,n}_{z,h}\cdot h^n_{t-1} + W^{k,n}_{z,x}\cdot x^n_t) \circ [\tanh'(\xi_{\tilde{h}r})(\Delta W^{k,n}_{\tilde{h},r}\cdot(\sigma(W^{k+1,n}_{r,h}\cdot h^n_{t-1} + W^{k+1,n}_{r,x}\cdot x^n_t) \circ h^n_{t-1})\\
&\quad + W^{k,n}_{\tilde{h},r}\cdot((\sigma(W^{k+1,n}_{r,h}\cdot h^n_{t-1} + W^{k+1,n}_{r,x}\cdot x^n_t) - \sigma(W^{k,n}_{r,h}\cdot h^n_{t-1} + W^{k,n}_{r,x}\cdot x^n_t)) \circ h^n_{t-1}))
\end{aligned}$$

$$\begin{aligned}
&+\tanh'(\xi_{\tilde{h}x})(\Delta W_{\tilde{h},x}^{k,n}\cdot x_t^n)] \\
&=\sigma(W_{z,h}^{k,n}\cdot h_{t-1}^n+W_{z,x}^{k,n}\cdot x_t^n)\circ[\tanh'(\xi_{\tilde{h}r})(\Delta W_{\tilde{h},r}^{k,n}\cdot(\sigma(W_{r,h}^{k+1,n}\cdot h_{t-1}^n+W_{r,x}^{k+1,n}\cdot x_t^n)\circ h_{t-1}^n) \\
&\quad+W_{\tilde{h},r}^{k,n}\cdot((\sigma'(\xi_{rh})(\Delta W_{r,h}^{k,n}\cdot h_{t-1}^n)+\sigma'(\xi_{rx})(\Delta W_{r,x}^{k,n}\cdot x_t^n))\circ h_{t-1}^n))+\tanh'(\xi_{\tilde{h}x})(\Delta W_{\tilde{h},x}^{k,n}\cdot x_t^n)] \\
&\le D_0\circ[D_0(\Delta W_{\tilde{h},r}^{k,n}\cdot(D_0\circ D_1)+W_{\tilde{h},r}^{k,n}\cdot((D_0(\Delta W_{r,h}^{k,n}\cdot D_1)+D_0(\Delta W_{r,x}^{k,n}\cdot D_1))\circ D_1)) \\
&\quad+D_0(\Delta W_{\tilde{h},x}^{k,n}\cdot D_1)]
\end{aligned} \tag{A22}$$

further, we have

$$\begin{aligned}
A_7&=[\sigma(W_{z,h}^{k+1,n}\cdot h_{t-1}^n+W_{z,x}^{k+1,n}\cdot x_t^n)-\sigma(W_{z,h}^{k,n}\cdot h_{t-1}^n+W_{z,x}^{k,n}\cdot x_t^n)]\circ h_{t-1}^n \\
&=[\sigma'(\xi_{zh})(W_{z,h}^{k+1,n}\cdot h_{t-1}^n-W_{z,h}^{k,n}\cdot h_{t-1}^n)+\sigma'(\xi_{zx})(W_{z,x}^{k+1,n}\cdot x_t^n-W_{z,x}^{k,n}\cdot x_t^n)]\circ h_{t-1}^n \\
&=[\sigma'(\xi_{zh})(\Delta W_{z,h}^{k,n}\cdot h_{t-1}^n)+\sigma'(\xi_{zx})(\Delta W_{z,x}^{k+1,n}\cdot x_t^n)]\circ h_{t-1}^n \\
&\le[D_0(\Delta W_{z,h}^{k,n}\cdot D_1)+D_0(\Delta W_{z,x}^{k+1,n}\cdot D_1)]\circ D_1
\end{aligned} \tag{A23}$$

From the previous equation (A15) to (A23), it follows that

$$\begin{aligned}
\Delta h_t^{k,n}&=A_4+A_5+A_6+A_7 \\
&\le[D_0(\Delta W_{z,h}^{k,n}\cdot D_1)+D_0(\Delta W_{z,x}^{k,n}\cdot D_1)]\circ[D_0(\Delta W_{\tilde{h},r}^{k,n}\cdot(D_0\circ D_1) \\
&\quad+W_{\tilde{h},r}^{k,n}\cdot((D_0(\Delta W_{r,h}^{k,n}\cdot D_1)+D_0(\Delta W_{r,x}^{k,n}\cdot D_1))\circ D_1))+D_0(\Delta W_{\tilde{h},x}^{k,n}\cdot D_1)]+[D_0(\Delta W_{z,h}^{k,n}\cdot D_1) \\
&\quad+D_0(\Delta W_{z,x}^{k,n}\cdot D_1)]\circ D_0+D_0\circ[D_0(\Delta W_{\tilde{h},r}^{k,n}\cdot(D_0\circ D_1)+W_{\tilde{h},r}^{k,n}\cdot((D_0(\Delta W_{r,h}^{k,n}\cdot D_1)+D_0(\Delta W_{r,x}^{k,n} \\
&\quad\cdot D_1))\circ D_1))+D_0(\Delta W_{\tilde{h},x}^{k,n}\cdot D_1)]+[D_0(\Delta W_{z,h}^{k,n}\cdot D_1)+D_0(\Delta W_{z,x}^{k+1,n}\cdot D_1)]\circ D_1 \\
&\le[D_0D_1(\Delta W_{z,h}^{k,n}+\Delta W_{z,x}^{k,n})]\circ[D_0^2D_1\Delta W_{\tilde{h},r}^{k,n}+D_0^2D_1^2D_2(\Delta W_{r,h}^{k,n}+\Delta W_{r,x}^{k,n})+D_0D_1\Delta W_{\tilde{h},x}^{k,n}] \\
&\quad+D_0D_1^2(\Delta W_{z,h}^{k,n}+\Delta W_{z,x}^{k,n})+D_0^3D_1\Delta W_{\tilde{h},r}^{k,n}+D_0^3D_1^2D_2(\Delta W_{r,h}^{k,n}+\Delta W_{r,x}^{k,n})+D_0^2D_1\Delta W_{\tilde{h},x}^{k,n} \\
&\quad+D_0D_1^2(\Delta W_{z,h}^{k,n}+\Delta W_{z,x}^{k,n}) \\
&\le D_3[(\Delta W_{z,h}^{k,n}+\Delta W_{z,x}^{k,n})(\Delta W_{\tilde{h},r}^{k,n}+\Delta W_{\tilde{h},x}^{k,n})+(\Delta W_{z,h}^{k,n}+\Delta W_{z,x}^{k,n})(\Delta W_{r,h}^{k,n}+\Delta W_{r,x}^{k,n}) \\
&\quad+2(\Delta W_{z,h}^{k,n}+\Delta W_{z,x}^{k,n})+(\Delta W_{\tilde{h},r}^{k,n}+\Delta W_{\tilde{h},x}^{k,n})+(\Delta W_{r,h}^{k,n}+\Delta W_{r,x}^{k,n})] \\
&\le D_3[(-\eta\frac{\partial E}{\partial W_z^k})(-\eta\frac{\partial E}{\partial W_{\tilde{h}}^k})+(-\eta\frac{\partial E}{\partial W_z^k})(-\eta\frac{\partial E}{\partial W_r^k}) \\
&\quad+2(-\eta\frac{\partial E}{\partial W_z^k})+(-\eta\frac{\partial E}{\partial W_{\tilde{h}}^k})+(-\eta\frac{\partial E}{\partial W_r^k})] \\
&\le D_3[(-\eta\frac{\partial E}{\partial W_z^k})(-\eta\frac{\partial E}{\partial W_{\tilde{h}}^k})+(-\eta\frac{\partial E}{\partial W_z^k})(-\eta\frac{\partial E}{\partial W_r^k})+2(-\eta\frac{\partial E}{\partial W_z^k})+(-\eta\frac{\partial E}{\partial W_{\tilde{h}}^k})+(-\eta\frac{\partial E}{\partial W_r^k})] \\
&\le D_3[\frac{1}{2}\eta^2(||\frac{\partial E}{\partial W_z^k}||^2+||\frac{\partial E}{W_{\tilde{h}}^k}||^2)+\frac{1}{2}\eta^2(||\frac{\partial E}{\partial W_z^k}||^2+||\frac{\partial E}{W_r^k}||^2) \\
&\quad+(-\eta)||\frac{\partial E}{\partial W_z^k}||^2+(-\eta)\frac{1}{2}||\frac{\partial E}{W_{\tilde{h}}^k}||^2)+(-\eta)\frac{1}{2}||\frac{\partial E}{\partial W_r^k}||^2]
\end{aligned} \tag{A24}$$

then

$$\begin{aligned}(\Delta h_t^{k,n})^2 &\le D_3^2[(\Delta W_{z,h}^{k,n} + \Delta W_{z,x}^{k,n})(\Delta W_{\tilde{h},r}^{k,n} + \Delta W_{\tilde{h},x}^{k,n}) + (\Delta W_{z,h}^{k,n} + \Delta W_{z,x}^{k,n})(\Delta W_{r,h}^{k,n} + \Delta W_{r,x}^{k,n}) \\ &\quad + 2(\Delta W_{z,h}^{k,n} + \Delta W_{z,x}^{k,n}) + (\Delta W_{\tilde{h},r}^{k,n} + \Delta W_{\tilde{h},x}^{k,n}) + (\Delta W_{r,h}^{k,n} + \Delta W_{r,x}^{k,n})]^2 \\ &\le D_3^2[4(\Delta W_{z,h}^{k,n} + \Delta W_{z,x}^{k,n})^2 + 2(\Delta W_{\tilde{h},r}^{k,n} + \Delta W_{\tilde{h},x}^{k,n})^2 + 2(\Delta W_{r,h}^{k,n} + \Delta W_{r,x}^{k,n})^2] \\ &\le \eta^2 D_3^2(4||\frac{\partial E}{\partial W_z^k}||^2 + 2||\frac{\partial E}{W_{\tilde{h}}^k}||^2 + 2||\frac{\partial E}{\partial W_r^k}||^2)\end{aligned} \tag{A25}$$

where $D_3 = max\{D_0^3 D_1^2, D_0^3 D_1^3 D_2, D_0^2 D_1^2, D_0 D_1^2, D_0^3 D_1, D_0^3 D_1^2 D_2, D_0^2 D_1\}$.

The next step is to focus on deriving (A11) to (A13):

$$\begin{aligned}A_1 &= \sum_{n=1}^{N} \sigma'_n(W_{out}^k \cdot h_t^{k,n})\Delta W_{out}^k \cdot h_t^{k,n} + \lambda[L_1(W_{out}^{k+1}) - L_1(W_{out}^k)] \\ &\le \sum_{n=1}^{N} \sigma'_n(W_{out}^k \cdot h_t^{k,n})\Delta W_{out}^k \cdot h_t^{k,n} + \lambda L_1'(W_{out}^k)\Delta L_1(W_{out}^k) + \frac{\lambda C}{2}[\Delta L_1(W_{out}^k)]^2 \\ &\le \frac{\partial E^k}{\partial W_{out}^k}\Delta W_{out}^k + \frac{\lambda C}{2}[\Delta L_1(W_{out}^k)]^2 \\ &\le \frac{\partial E^k}{\partial W_{out}^k}(-\eta\frac{\partial E^k}{\partial W_{out}^k}) + \frac{\lambda C}{2}|\Delta W_{out}^k|^2 \\ &\le -\eta(\frac{\partial E^k}{\partial W_{out}^k})^2 + \eta^2\frac{\lambda C}{2}|\frac{\partial E^k}{\partial W_{out}^k}|^2\end{aligned} \tag{A26}$$

and

$$\begin{aligned}A_2 &= \sum_{n=1}^{N} \sigma'_n(W_{out}^k \cdot h_t^{k,n})\Delta W_{out}^k \cdot h_t^{k,n} \\ &\le \sum_{n=1}^{N} D_0 D_2 D_3[(\Delta W_{z,h}^{k,n} + \Delta W_{z,x}^{k,n})(\Delta W_{\tilde{h},r}^{k,n} + \Delta W_{\tilde{h},x}^{k,n}) + (\Delta W_{z,h}^{k,n} + \Delta W_{z,x}^{k,n})(\Delta W_{r,h}^{k,n} + \Delta W_{r,x}^{k,n}) \\ &\quad + 2(\Delta W_{z,h}^{k,n} + \Delta W_{z,x}^{k,n}) + (\Delta W_{\tilde{h},r}^{k,n} + \Delta W_{\tilde{h},x}^{k,n}) + (\Delta W_{r,h}^{k,n} + \Delta W_{r,x}^{k,n})] \\ &\le N D_0 D_2 D_3[\frac{1}{2}(||\frac{\partial E}{\partial W_z^k}||^2 + ||\frac{\partial E}{W_{\tilde{h}}^k}||^2) + \frac{1}{2}(||\frac{\partial E}{\partial W_z^k}||^2 + ||\frac{\partial E}{W_r^k}||^2) + (-\eta)||\frac{\partial E}{\partial W_z^k}||^2 \\ &\quad + (-\eta)\frac{1}{2}||\frac{\partial E}{W_{\tilde{h}}^k}||^2) + (-\eta)\frac{1}{2}||\frac{\partial E}{\partial W_r^k}||^2]\end{aligned} \tag{A27}$$

and

$$\begin{aligned}A_3 &= \sum_{n=1}^{N} \sigma'_n(W_{out}^k \cdot h_t^{k,n})\Delta W_{out}^k \cdot \Delta h_t^{k,n} \\ &\le N D_0 D_2 D_3[\frac{1}{2}(||\frac{\partial E}{\partial W_z^k}||^2 + ||\frac{\partial E}{W_{\tilde{h}}^k}||^2) + \frac{1}{2}(||\frac{\partial E}{\partial W_z^k}||^2 + ||\frac{\partial E}{W_r^k}||^2) + (-\eta)||\frac{\partial E}{\partial W_z^k}||^2 \\ &\quad + (-\eta)\frac{1}{2}||\frac{\partial E}{W_{\tilde{h}}^k}||^2) + (-\eta)\frac{1}{2}||\frac{\partial E}{\partial W_r^k}||^2]\end{aligned} \tag{A28}$$

next,

$$
\begin{aligned}
R_1 &= \frac{1}{2}\sum_{n=1}^{N}\sigma''(s_{k,n})(W_{out}^{k+1}h_t^{k+1,n} - W_{out}^{k}h_t^{k,n})^2 \\
&= \frac{1}{2}\sum_{n=1}^{N}\sigma''(s_{k,n})\left[\Delta W_{out}^{k}\left(z_t^{k+1,n}\circ\tilde{h}_t^{k+1,n} + (1-z_t^{k+1,n})\circ h_{t-1}^{n}\right) - W_{out}^{k}\Delta h_t^{k,n}\right]^2 \\
&= \frac{1}{2}\sum_{n=1}^{N}\sigma''(s_{k,n})[\Delta W_{out}^{k}(\sigma(W_{z,h}^{k+1,n}\cdot h_{t-1}^{n} + W_{z,x}^{k+1,n}\cdot x_t^{n})\circ\tanh(W_{\tilde{h}^{k+1,n}}\cdot(r_t^{k+1,n}\circ h_{t-1}^{n}) + W_{\tilde{h}}^{k+1,n}\cdot x_t^{n}) \\
&\quad + (1-\sigma(W_{z,h}^{k+1,n}\cdot h_{t-1}^{n} + W_{z,x}^{k+1,n}\cdot x_t^{n}))\circ h_{t-1}^{n}) - W_{out}^{k}\Delta h_t^{k,n})]^2 \\
&\le \frac{1}{2}\sum_{n=1}^{N}D_0[\Delta W_{out}^{k}(D_0D_0 + (1-D_0)D_1) - D_2\Delta h_t^{k,n})]^2 \\
&\le D_0\frac{1}{2}\sum_{n=1}^{N}[\Delta W_{out}^{k}(D_0^2 + D_1 - D_0D_1) - D_2\Delta h_t^{k,n}]^2 \\
&\le D_0\frac{1}{2}\sum_{n=1}^{N}2[(\Delta W_{out}^{k})^2(D_0^2 + D_1 - D_0D_1)^2 + D_2^2(\Delta h_t^{k,n})^2] \\
&\le \frac{1}{2}D_0\sum_{n=1}^{N}[(\Delta W_{out}^{k})^2(D_0^2 + D_1 - D_0D_1)^2 + D_2^2D_3^2[4(\Delta W_{z,h}^{k,n} + \Delta W_{z,x}^{k,n})^2 + 2(\Delta W_{\tilde{h},r}^{k,n} + \Delta W_{\tilde{h},x}^{k,n})^2 \\
&\quad + 2(\Delta W_{r,h}^{k,n} + \Delta W_{r,x}^{k,n})^2]] \\
&\le \frac{1}{2}D_0N[\eta^2||\frac{\partial E}{\partial W_{out}^{k}}||^2(D_0^2 + D_1 - D_0D_1)^2 + D_2^2D_3^2\eta^2(4||\frac{\partial E}{\partial W_z^{k}}||^2 + 2||\frac{\partial E}{W_{\tilde{h}}^{k}}||^2 + 2||\frac{\partial E}{\partial W_r^{k}}||^2)]
\end{aligned}
\tag{A29}
$$

Building on the previous equations and Assumption (A3),

$$
\begin{aligned}
E^{k+1} - E^{k} &= A_1 + A_2 + A_3 + R_1 \\
&\le \frac{\partial E^k}{\partial W_{out}^{k}}(-\eta\frac{\partial E^k}{\partial W_{out}^{k}}) + \frac{\lambda C}{2}|\Delta W_{out}^{k}|^2 \\
&\le -\eta||\frac{\partial E^k}{\partial W_{out}^{k}}||^2 + \eta^2\frac{\lambda C}{2}||\frac{\partial E^k}{\partial W_{out}^{k}}||^{2} + ND_0D_2D_3[\frac{1}{2}\eta^2(||\frac{\partial E}{\partial W_z^{k}}||^2 + ||\frac{\partial E}{W_{\tilde{h}}^{k}}||^2) \\
&\quad + \frac{1}{2}\eta^2(||\frac{\partial E}{\partial W_z^{k}}||^2 + ||\frac{\partial E}{W_r^{k}}||^2) + (-\eta)||\frac{\partial E}{\partial W_z^{k}}||^2 + (-\eta)\frac{1}{2}||\frac{\partial E}{W_{\tilde{h}}^{k}}||^2 + (-\eta)\frac{1}{2}||\frac{\partial E}{\partial W_r^{k}}||^2] \\
&\quad + ND_0D_2D_3[\frac{1}{2}(||\frac{\partial E}{\partial W_z^{k}}||^2 + ||\frac{\partial E}{W_{\tilde{h}}^{k}}||^2) + \frac{1}{2}(||\frac{\partial E}{\partial W_z^{k}}||^2 + ||\frac{\partial E}{W_r^{k}}||^2) + (-\eta)||\frac{\partial E}{\partial W_z^{k}}||^2 \\
&\quad + (-\eta)\frac{1}{2}||\frac{\partial E}{W_{\tilde{h}}^{k}}||^2) + (-\eta)\frac{1}{2}||\frac{\partial E}{\partial W_r^{k}}||^2] + \frac{1}{2}D_0N[(D_0^2 + D_1 - D_0D_1)^2\eta^2||\frac{\partial E}{\partial W_{out}^{k}}||^2 \\
&\quad + D_2^2D_3^2\eta^2(4||\frac{\partial E}{\partial W_z^{k}}||^2 + 2||\frac{\partial E}{W_{\tilde{h}}^{k}}||^2 + 2||\frac{\partial E}{\partial W_r^{k}}||^2)] \\
&\le [-\eta + \eta^2\frac{\lambda C}{2} + 4ND_0D_2D_3(\eta^2 - \eta) + \frac{1}{2}D_0N[(D_0^2 + D_1 - D_0D_1)^2\eta^2 + 8D_2^2D_3^2\eta^2]]||\frac{\partial E}{\partial W^{k}}||^2 \\
&\le [-\eta + \eta^2\frac{\lambda C}{2} + 4ND_0D_2D_3\eta^2 - 4ND_0D_2D_3\eta
\end{aligned}
$$

$$
\begin{aligned}
&\quad + \frac{1}{2}D_0N(D_0^2 + D_1 - D_0D_1)^2\eta^2 + 4ND_0D_2^2D_3^2\eta^2]||\frac{\partial E}{\partial W^k}||^2 \\
&\le [-\eta + \eta^2\frac{\lambda C}{2} + 4ND_0D_2D_3\eta^2 - 4ND_0D_2D_3\eta + \frac{1}{2}D_0N(D_0^2 + D_1 - D_0D_1)^2\eta^2 \\
&\quad + 4ND_0D_2^2D_3^2\eta^2]||\frac{\partial E}{\partial W^k}||^2 \\
&\le \eta[-1 - 4ND_0D_2D_3 + (\frac{\lambda C}{2} + 4ND_0D_2D_3 + \frac{1}{2}D_0N(D_0^2 + D_1 - D_0D_1)^2 \\
&\quad + 4ND_0D_2^2D_3^2)\eta]||\frac{\partial E}{\partial W^k}||^2 \\
&\le -\eta[1 + 4ND_0D_2D_3 - \eta(\frac{\lambda C}{2} + 4ND_0D_2D_3 + \frac{1}{2}D_0N(D_0^2 + D_1 - D_0D_1)^2 \\
&\quad + 4ND_0D_2^2D_3^2)]||\frac{\partial E}{\partial W^k}||^2 \\
&\le -\eta[1 + D_4 - \eta(\frac{\lambda C}{2} + 2D_4 + D_5)]||\frac{\partial E}{\partial W^k}||^2 \\
&\le 0
\end{aligned}
\tag{A30}
$$

where $D_4 = max\{4ND_0D_2D_3, 4ND_0D_2^2D_3^2\}$ and $D_5 = \frac{1}{2}D_0N(D_0^2 + D_1 - D_0D_1)^2$.

This completes the proof of Theorem 1.

Proof of Theorem 2. □

Let $D_6 = \eta[1 + D_4 - \eta(\frac{\lambda C}{2} + 2D_4 + D_5)]$. According to assumptions (A2) and (A3), there is obviously $D_6 > 0$. Using the result from Eq. (A30), we have

$$
\begin{aligned}
E^{k+1} &\le E^k - D_6\left\|\frac{\partial E^k}{\partial W^k}\right\|^2 \\
&\le E^{k-1} - \left(D_6\left\|\frac{\partial E^{k-1}}{\partial W^{k-1}}\right\|^2 + D_6\left\|\frac{\partial E^k}{\partial W^k}\right\|^2\right) \\
&\le \cdots \\
&\le E^0 - D_6\sum_{i=0}^{k}\left\|\frac{\partial E}{\partial W^i}\right\|^2
\end{aligned}
\tag{A31}
$$

with $E^{k+1} \le 0$, we can get

$$
0 \le E^0 - D_6\sum_{i=0}^{k}\left\|\frac{\partial E}{\partial W^i}\right\|^2 \tag{A32}
$$

when $k \to +\infty$,

$$
\sum_{i=0}^{k}\left\|\frac{\partial E}{\partial W^i}\right\|^2 \le \frac{E^0}{D_6} < +\infty \tag{A33}
$$

$$
\lim_{k\to+\infty}\left\|\frac{\partial E}{\partial W^k}\right\|^2 = 0 \tag{A34}
$$

Consequently

$$\lim_{k\to+\infty}\left\|\frac{\partial E}{\partial W^k}\right\| = 0 \tag{A35}$$

This concludes the proof of Theorem 2.

Proof of Theorem 3. □

Lemma A2. *Consider $U \subset \mathbb{R}^Q$ as a compact set, where the function F: $\mathbb{R}^Q \to \mathbb{R}$ is both continuous and differentiable. Assume that $\bar{\Omega} = \left\{x \in U \,\middle|\, \frac{\partial F(x)}{\partial x}\right\} = 0$ includes only a finite number of points. If a sequence $\left\{x^k\right\} \subset U$ satisfies*

$$\lim_{k\to\infty}\left\|x^{k+1} - x^k\right\| = 0, \lim_{k\to\infty}\left\|\frac{\partial F\left(x^k\right)}{\partial x}\right\| = 0 \tag{A36}$$

then, there has $x^* \in \bar{\Omega}$ such that $\lim\limits_{k\to\infty} x_k = x^*$.

According to assumption (A4), Lemma A2 and (A35), a point $W^* \in \phi_1$ exists such that

$$W^* = \lim_{k\to\infty} W^k \tag{A37}$$

Thus the proof to Theorem 3 is completed.

References

1. Agarap AFM. A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data. In: Proceedings of the 2018 10th International Conference on Machine Learning and Computing; 2018; Macau, China. p. 26–30.
2. Liang X, Wang J. A recurrent neural network for nonlinear optimization with a continuously differentiable objective function and bound constraints. IEEE Transact Neural Netw. 2000;11(6):1251–62. doi:10.1109/72.883412.
3. Hochreiter S. Untersuchungen zu dynamischen neuronalen Netzen. Diploma, Technische Universität München. 1991;91(1):31.
4. Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Transact Neural Netw. 1994;5(2):157–66. doi:10.1109/72.279181.
5. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80. doi:10.1162/neco.1997.9.8.1735.
6. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv:14061078. 2014.
7. Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge, MA, USA: MIT Press; 2016.
8. Shewalkar A, Nyavanandi D, Ludwig SA. Performance evaluation of deep neural networks applied to speech recognition: rNN, LSTM and GRU. J Artif Intell Soft Comput Res. 2019;9(4):235–45. doi:10.2478/jaiscr-2019-0006.
9. Zaman U, Khan J, Lee E, Hussain S, Balobaid AS, Aburasain RY, et al. An efficient long short-term memory and gated recurrent unit based smart vessel trajectory prediction using automatic identification system data. Comput Mater Contin. 2024;81(1):1789–808. doi:10.32604/cmc.2024.056222.
10. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data. 2021;8:1–74.doi:10.1186/s40537-021-00444-8.
11. Bejani MM, Ghatee M. A systematic review on overfitting control in shallow and deep neural networks. Artif Intel Rev. 2021;54(8):6391–438. doi:10.1007/s10462-021-09975-1.

12. Schittenkopf C, Deco G, Brauer W. Two strategies to avoid overfitting in feedforward networks. Neural Netw. 1997;10(3):505–16. doi:10.1016/S0893-6080(96)00086-X.
13. Li H, Kadav A, Durdanovic I, Samet H, Graf HP. Pruning filters for efficient convnets. arXiv:160808710. 2016.
14. Girosi F, Jones M, Poggio T. Regularization theory and neural networks architectures. Neural Comput. 1995;7(2):219–69. doi:10.1162/neco.1995.7.2.219.
15. Quasdane M, Ramchoun H, Masrour T. Sparse smooth group $L_1{}^{\circ}L_{1/2}$ regularization method for convolutional neural networks. Knowl Based Syst. 2024;284:111327.
16. Van Laarhoven T. L2 regularization versus batch and weight normalization. arXiv:170605350. 2017.
17. Santos CFGD, Papa JP. Avoiding overfitting: a survey on regularization methods for convolutional neural networks. ACM Comput Surv . 2022;54(10s):1–25. doi:10.1145/3510413.
18. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014;15(1):1929–58.
19. Wu L, Li J, Wang Y, Meng Q, Qin T, Chen W, et al. R-drop: regularized dropout for neural networks. Adv Neural Inform Process Syst. 2021;34:10890–905.
20. Israr H, Khan SA, Tahir MA, Shahzad MK, Ahmad M, Zain JM. Neural machine translation models with attention-based dropout layer. Comput Mater Contin. 2023;75(2):2981–3009. doi:10.32604/cmc.2023.035814.
21. Park MY, Hastie T. L_1-regularization path algorithm for generalized linear models. J Royal Statist Soc Ser B: Statist Method. 2007;69(4):659–77. doi:10.1111/j.1467-9868.2007.00607.x.
22. Salehi F, Abbasi E, Hassibi B. The impact of regularization on high-dimensional logistic regression. Adv Neural Inf Process Syst. 2019;32:1310–20.
23. Shi X, Kang Q, An J, Zhou M. Novel L1 regularized extreme learning machine for soft-sensing of an industrial process. IEEE Transact Indust Inform. 2021;18(2):1009–17. doi:10.1109/TII.2021.3065377.
24. Zhang H, Wu W, Yao M. Boundedness and convergence of batch back-propagation algorithm with penalty for feedforward neural networks. Neurocomputing. 2012;89(3):141–6. doi:10.1016/j.neucom.2012.02.029.
25. Wang J, Wu W, Zurada JM. Computational properties and convergence analysis of BPNN for cyclic and almost cyclic learning with penalty. Neural Netw. 2012;33(4):127–35. doi:10.1016/j.neunet.2012.04.013.
26. Kang Q, Fan Q, Zurada JM. Deterministic convergence analysis via smoothing group Lasso regularization and adaptive momentum for Sigma-Pi-Sigma neural network. Inform Sci. 2021;553(1):66–82. doi:10.1016/j.ins.2020.12.014.
27. Yu D, Kang Q, Jin J, Wang Z, Li X. Smoothing group $L_{1/2}$ regularized discriminative broad learning system for classification and regression. Pattern Recognit. 2023;141(10–11):109656. doi:10.1016/j.patcog.2023.109656.
28. Wang J, Wen Y, Ye Z, Jian L, Chen H. Convergence analysis of BP neural networks via sparse response regularization. Appl Soft Comput. 2017;61:354–63. doi:10.1016/j.asoc.2017.07.059.
29. Fan Q, Kang Q, Zurada JM, Huang T, Xu D. Convergence analysis of online gradient method for high-order neural networks and their sparse optimization. IEEE Trans Neural Netw Learn Syst. 2023;35(12):18687–701. doi:10.1109/TNNLS.2023.3319989.
30. Kang Q, Fan Q, Zurada JM, Huang T. A pruning algorithm with relaxed conditions for high-order neural networks based on smoothing group $L_{1/2}$ regularization and adaptive momentum. Knowl Based Syst. 2022;257:109858. doi:10.1016/j.knosys.2022.109858.
31. Fan Q, Peng J, Li H, Lin S. Convergence of a gradient-based learning algorithm with penalty for ridge polynomial neural networks. IEEE Access. 2021;9:28742–52. doi:10.1109/ACCESS.2020.3048235.
32. Yang S, Yu X, Zhou Y. LSTM and GRU neural network performance comparison study: taking yelp review dataset as an example. In: 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI); 2020. Shanghai, China: IEEE. p. 98–101.
33. Ma R, Miao J, Niu L, Zhang P. Transformed L_1 regularization for learning sparse deep neural networks. Neural Netw. 2019;119:286–98. doi:10.1016/j.neunet.2019.08.015.
34. Campi MC, Caré A. Random convex programs with L_1-regularization: sparsity and generalization. SIAM J Cont Optimiza. 2013;51(5):3532–57. doi:10.1137/110856204.