Qinwei Fan
School of Mathematics and Information Science, Guangzhou University, Guangzhou, 510006, CHINA

Qianru Huang
School of Science, Xi'an Polytechnic University, Xi'an, 710048, CHINA

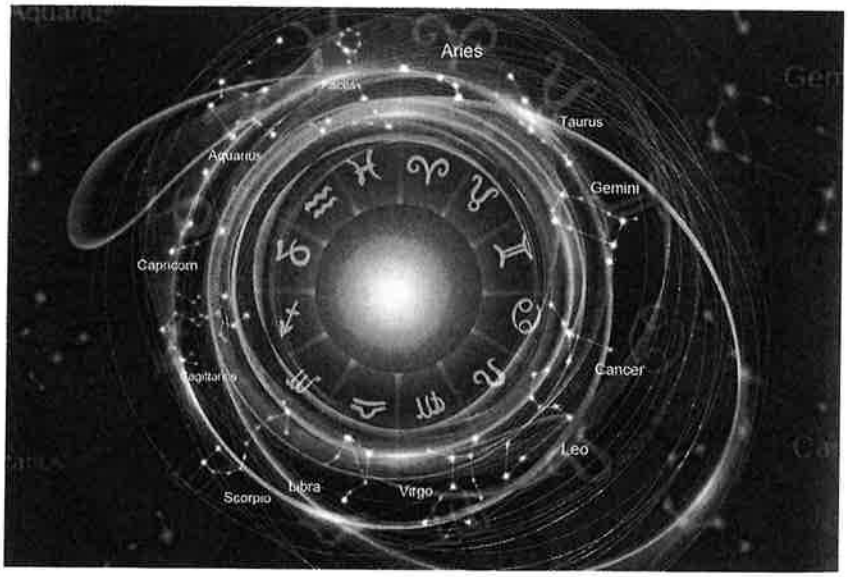
Jacek M. Zurada
Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY, 40292, USA and also University of Social Sciences, 90-113, Lodz, POLAND

Jigen Peng and Haiyang Li
School of Mathematics and Information Science, Guangzhou University, Guangzhou, 510006, CHINA

Graph Regularization for High Order Neural Networks: Boundedness, Convergence, and Enhanced Performance With Entropy Error Function

Abstract

As a kind of high order neural network, Sigma-Pi-Sigma has strong nonlinear mapping ability, but the complexity of its network structure often leads to low application efficiency and lack of theoretical results. Although traditional regularization methods are more effective in sparsity and preventing overfitting, they do not take into account the topology and connection information between the data. Therefore, in order to sparsely optimize the network structure for improved efficiency and to enable rigorous theoretical analysis of complex inter-node relationships, graph-structured data must be processed more effectively. In this paper, the graph regularization component is integrated into the Sigma-Pi-Sigma neural network (SPSNN) learning framework, and the SPSNN online gradient algorithm based on the Turlaplacian regularization is studied, considering the entropy error measure. Under reasonable conditions, the boundedness of the weights, the monotonicity of the error function, and the convergence of the algorithm are strictly proved. Subsequently, the effectiveness of this method is verified through extensive experimental data. The simulation results are in good agreement with the theoretical results.



©SHUTTERSTOCK/ AUI KHEMMANAT

I. Introduction

Wide applications of neural networks in various fields [1], [2], [3], [4] such as healthcare, autonomous driving, and agriculture have triggered widespread attention. However, neural network models are constrained by issues such as limited memory capacity when solving complex problems. To mitigate these limitations, higher-order interactions between neurons have been explored

Digital Object Identifier 10.1109/MCI.2026.3655677
Date of current version: 7 April 2026

This article was recommended for publication by Associate Editor Danil Prokhorov. Corresponding author: Qinwei Fan (e-mail: qinweifan@xpu.edu.cn).

1556-603X © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies.

to enhance computational, storage, and learning capabilities [5]. Higher-order neural networks are known to capture complex patterns and features more effectively in data by introducing more complex connections. Compared to traditional lower-order networks, higher-order neural networks also increased memory capacity, quicker learning convergence and enhanced fault tolerance [6], [7].

The Sigma-Pi-Sigma Neural Network (SPSNN), a higher-order neural network distinguished by its Pi-Sigma units, belongs to a unique class of neural networks [8]. The Pi-Sigma Neural Network (PSNN), a distinct category within higher-order neural networks, features a sum-product structure in its output, which enhances its modeling capabilities and makes it well-suited for addressing the challenge of mapping high-dimensional input spaces [9]. SPSNN is an extension of PSNN that retains the advantages of the latter, while also enabling a static mapping similar to that of radial basis function (RBF) neural networks [10], [11]. Additionally, SPSNN is capable of handling the symmetry problem commonly encountered in general networks [12].

The gradient descent method is widely used for training networks, resulting in iterative weight adjustments in the direction opposite to the error function's gradient. The gradient-based methods are divided into batch gradient and online gradient approaches. Compared to the disadvantages of batch gradient methods that require high accuracy of gradient estimation and high computational cost, the online gradient methods require less memory, reducing training time while enhancing efficiency [13], [14], [15]. Nevertheless, when training networks using online gradient methods, the complex structure of SPSNNs increases the risk of local optima and overfitting. On one hand, the error function is a key determinant of a neural network's convergence, stability, and generalization capability. For most feedforward neural networks, the mean squared error function is typically chosen as the error function for training [16], [17], [18]. However, in actual training, neural networks based on the mean squared error function converge slowly and are prone to getting stuck in local optima. Research indicates that the entropy error function exhibits a lower density of local minima and steeper gradients compared to the squared error function, enabling more efficient model parameter adjustment [19], [20], [21]. Furthermore, the entropy error function directly measures the discrepancy between the predicted probability distribution and the actual class distribution. This facilitates better handling of probability distributions and provides more effective gradient signals, thereby accelerating model training and enhancing classification performance [22], [23]. Therefore, replacing the squared error function with the cross-entropy (CE) function can be considered for network optimization. On the other hand, the training of the network involves updating the weights iteratively. However, this can result in tremendous weight values, further causing overfitting in the network. Moreover, having too few or too many neurons in the hidden layers can result in underfitting or overfitting, impacting the model's performance and generalization. Thus, finding an optimal architecture to enhance efficiency remains a significant challenge.

A regularization term is commonly added to the error function as an approach to tackle this issue. Various regularization techniques have been developed to enhance robustness and reliability of neural network training. The L_2 regularization is a widely employed method to mitigate overfitting [24], [25], [26]. The L_1 regularization can be used for feature selection as it tends to reduce or eliminate the coefficients of unimportant features [27], [28]. Group Lasso regularization, introduced in [29], [30], [31], seeks to disregard groups of coefficients while diminishing the weights at the group level within the network, thus increasing the network's sparseness and potentially enhancing its performance. Furthermore, $L_{1/2}$ regularization was proposed in [32], [33], to force the network to be sparser than the L_1 regularization. In [34], the authors proposed a resilient network combining L_2 and L_1 regularization, which combines the feature selection power of L_1 regularization with the stability of L_2 regularization.

Although traditional regularization terms have found extensive application in numerous models, they typically consider only the features themselves, overlooking the importance of neighbourhood information between data nodes. How, then, can the information between data points be more effectively exploited? In recent years, deep neural networks for graphs (DNNGs) have emerged as a promising field [35], [36], [37], owing to the graph's capacity as a powerful and flexible tool for representing complex information in the form of patterns and their relationships. Reference [38] introduces a novel approach for multi-scale feature extraction by constructing a Hal-type graph wavelet with permutation invariance, efficiency, and sparsity, thereby offering fresh insights for graph deep learning tasks. Reference [39] delves into the stability and generalization properties of deep graph convolutional neural networks (GCNs), revealing that these capabilities are influenced by certain critical factors. This contributes significantly to a deeper understanding of GCN stability and generalization. Consequently, effectively incorporating and utilising this a priori knowledge of graph structures during model optimization has become a central challenge for enhancing model generalization and robustness.

Since graph regularization can directly mine features on manifolds from input data, it is one of the most representative works in recent years, integrating with various machine learning methods [40], [41]. Compared with traditional regularization terms, graph regularization not only has the properties of sparsity and overfitting prevention of traditional regularization terms, it also takes into account the special structure of graph data, i.e., the connectivity between nodes. While traditional

regularization terms usually only consider the features themselves, graph regularization also exploits the neighborhood information of the nodes, thus better capturing the local and global patterns of the data. Graph Regularized Nonnegative Matrix Factorization (GNMF), as presented in [42], integrates local invariance and captures the geometric properties of the data space by constructing a nearest-neighbor graph. [43] introduces an efficient spatial regularization method using hyperpixel segmentation and graph Laplacian regularization, which uses graph Laplacian regularization to study spatial similarity. In [44], the unsupervised feature selection problem is framed using graph regularization-based data reconstruction. This method's feature selection preserves the intrinsic local patterns of the data through graph regularization, minimizes the reconstruction error, and effectively retains both the similarity and discriminative information from the original data. A BP algorithm based on graph regularization (BPGR) is proposed in [45]. This method refines its parameters by incorporating a graph regularization term into the error function. The core concept is to encourage greater concentration of the hidden layer's latent features, thereby enhancing generalization performance.

This paper aims to conduct a thorough analysis of weight boundedness, as well as to evaluate weak and strong convergence for the SPSNN online gradient method, which incorporates graph regularization and an entropy error function.

(1) To determine an appropriate architecture, the unique geometric and topological relationships of a multidimensional environment are mapped onto the data elements of a low-dimensional space, and manifold learning is employed to preserve these relationships by capturing the associations among sample points on the high-dimensional manifold. The algorithm captures the structure and features of the nodes by introducing graph Laplacian regularization, which drives a greater concentration of potential features in the hidden layer belonging to the same class, thus improving generalization performance and sparsity.

(2) The CE-based algorithm quantifies the discrepancy between the model's output distribution and the true label distribution, featuring a lower density of local minima and a steeper gradient. This solves the problems of the traditional error function of falling into the local optimum easily and converging slowly.

(3) In addition to demonstrating benefits through numerical experiments, this paper provides a straightforward and efficient proof of weight boundedness under more relaxed conditions, thereby removing boundedness as a necessary criterion for ensuring convergence.

(4) Both the strong convergence theorem and the weak convergence theorem for the new algorithm are established and demonstrated in this paper.

Section II presents related work on Laplace matrices as well as graph regularization. Section III describes SPSNN's online gradient method based on entropy error functions with graph regularization terms. The primary theoretical results are outlined in Section IV, followed by a discussion of numerical simulations in Section V, with the paper concluding in Section VI. The Appendix includes comprehensive proofs for the theorems related to the proposed method.

II. Related Work

A. Graph Laplacian Matrix

Before constructing the Laplacian matrix, the weighted graph is first built from the training data, and the Laplacian matrix is then generated based on the weighted graph G. There are many ways to construct the weighted graph G. In this case, a straightforward approach is employed by connecting only nodes with the same category labels and assigning edge weights based on the similarity between training examples. For a given training set $x_i, y_{i=1}^N$, the similarity between pairs of instances x_i and x_j is quantified using a Gaussian kernel.

$$w_{ij} = \begin{cases} e^{-\|x_i - x_j\|^2/\sigma^2} & if\ i \neq j \\ 0, & if\ otherwise \end{cases} \quad (II.1)$$

where

$$\sigma^2 = \frac{1}{N}\sum_{j=1}^{N} \| x^j \|^2 \quad (II.2)$$

Therefore, the normalised Laplace matrix L corresponding to the weighted graph G is described as

$$L = (l_{ij})_{N\times N} = \begin{cases} 1 - w_{ij}/d_i, & if\ i = j\ and\ d_i \neq 0 \\ -w_{ij}/d_i d_j, & if\ i \neq j\ and\ y_i = y_j \\ 0, & if\ otherwise \end{cases} \quad (II.3)$$

where $d_i = \sum_{j\neq i} w_{ij}$.

An alternative representation of the Laplacian matrix L can be given as

$$L = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \quad (II.4)$$

where $D = diag(d_1, d_2, \ldots, d_N)$ and $W = (w_{ij})$ represent the diagonal matrix.

B. Feedforward Neural Network Algorithm Enhanced With Graph Regularization

The objective of the backpropagation learning process incorporating graph-based regularization in feed-forward neural networks (FNNs) is to exploit the intrinsic relationships among neurons in the hidden layer during training. Consider a three-layer FNN, where the input layer consists of P neurons, the hidden layer contains N neurons, and the output layer comprises Q neurons. W_0 and V are defined as the neural network parameters, where V corresponding to the weights linking the input and hidden layers, and W_0 denotes the weights that connect the hidden layer to the output layer. Let $\{X^j, O^j\}_{j=1}^J$ represent the training sample set. The networks output can then be defined as $y^j = g(W_0 \cdot f(V \cdot X^j))$, where both g and f are activation functions.

Let $\widetilde{E}(W)$ denote the error function. The error function, when including the regularization term, is given by

$$\widetilde{E}(W)=\frac{1}{2}\sum_{j=1}^{J}(y^j-O^j)^2+\frac{\lambda}{2}\phi(W_0,V) \tag{II.5}$$

where, $\phi(W_0, V)$ denotes the penalty term, with λ as its coefficient.

It is enforced that the outputs of the hidden layer corresponding to the same class are as similar as possible in order to enhance prediction accuracy. The penalty is defined as follows:

$$\begin{aligned}\phi(W_0,V)&=\frac{1}{2}\sum_{n=1}^{N}w(x_i,x_j)\\&\left\|\frac{1}{\sqrt{d_i}}f(Vx_i)-\frac{1}{\sqrt{d_j}}f(Vx_j)\right\|^2\\&=tr(f(VX)(I-D^{-\frac{1}{2}}WD^{-\frac{1}{2}})f(VX)^T)\\&=tr(f(VX)Lf(VX)^T)\end{aligned} \tag{II.6}$$

Thus, graph regularization is applied to the objective function to penalize the weights between the input and hidden layers. The revised error function is given as follows:

$$\begin{aligned}E(W)&=\widetilde{E}(W)+\frac{\lambda}{2}\phi(W_0,V)\\&=\widetilde{E}(W)+\frac{\lambda}{2}tr(f(VX)Lf(VX)^T)\end{aligned} \tag{II.7}$$

III. Algorithm Description

Within this section, the essential architecture of the SPSNN is delineated, which encompasses the initial input layer, the $\sum_1$, the $\prod$ and the $\sum_2$. The number of neurons across these layers is specified below: P for the input layer; N for the internal layer of summation nodes ($\sum_1$); Q for the internal layer of product nodes ($\prod$); and 1 for the output layer of summation nodes ($\sum_2$) (refer to Figure 1).

Consider a set of training samples, denoted as $\{X^j, O^j\}_{j=1}^{J} \subset \mathbb{R}^P \times \mathbb{R}$, where X^j is the input for the j-th sample and O^j is its corresponding target output. The weight vector between the input layer and the $\sum_1$ layer for $i = 1, 2, \ldots, N$ is labeled as $W_i = (w_{i1}, w_{i2}, \ldots, w_{iP}) \in \mathbb{R}^P$, while the weight vector linking the $\sum_2$ layer to the $\prod$ layers is referred to as $W_0 = (w_{01}, w_{02}, \ldots, w_{0Q}) \in \mathbb{R}^Q$. All weights are represented in a compact form as $W = (W_0^T, W_1^T, \ldots, W_N^T)^T \in \mathbb{R}^{Q+NP}$, and additionally define a matrix $V = (W_1^T, W_2^T, \ldots, W_Q^T)^T \in \mathbb{R}^{N\times P}$, with $g(x) = 1/(1 + e^{-x})$ denoting the predefined activation function mentioned in the text.

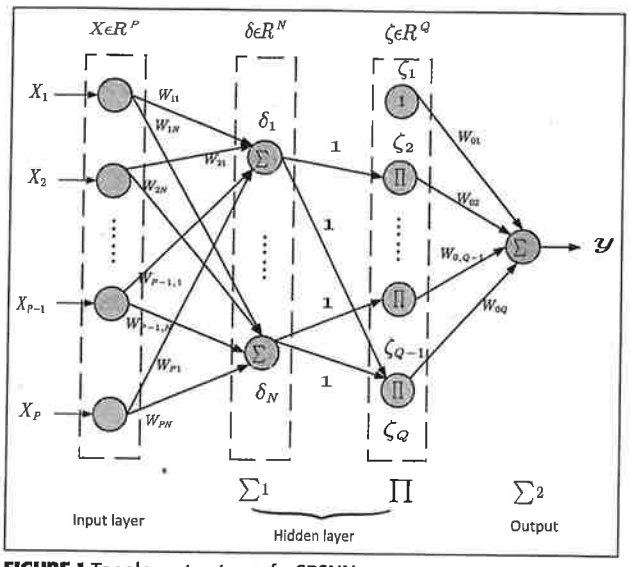


FIGURE 1 Topology structure of a SPSNN.

Then, it can be expressed as follows:

$$g'(x) = (1 - g(x))g(x)$$

and

$$0 \le g'(x) \le 1/4$$

To simplify calculations, the output of the Π layer is denoted by ζ^j, and it can be articulated as delineated below:

$$\begin{aligned}\zeta^j&=(\zeta_1^j,\zeta_2^j,\cdots\zeta_Q^j)\\&=\left(\prod_{i\in\Lambda_1}\sigma_i,\prod_{i\in\Lambda_2}\sigma_i,\cdots\prod_{i\in\Lambda_Q}\sigma_i\right)\\&=\left(\prod_{i\in\Lambda_1}g(W_i\cdot X^j),\prod_{i\in\Lambda_2}g(W_i\cdot X^j),\right.\\&\left.\cdots\prod_{i\in\Lambda_Q}g(W_i\cdot X^j)\right)\end{aligned} \tag{III.1}$$

In the equation, Λ_q represents a set formed by the indices of elements from $\sigma = \{\sigma_1, \sigma_1, \ldots, \sigma_N\}$, where $0 \le i \le N$.

Next, $y^j = g(W_0 \cdot \zeta^j) \in R$ is set to represent the actual output corresponding to each input X^j.

A. Entropy-Error-Function-Based SPSNN Model

This section explains the SPSNN model, which is founded on the entropy error function. This function can be expressed as $\widetilde{E}(W)$.

$$\widetilde{E}(W)=-\sum_{j=1}^{J}\left[\left(O^j\ln\left(\frac{y^j}{O^j}\right)+(1-O^j)\ln\left(\frac{1-y^j}{1-O^j}\right)\right)\right] \tag{III.2}$$

By applying (III.1), $\widetilde{E}(W)$ can alternatively be formulated as below:

$$\begin{aligned}\widetilde{E}(W)=&-\sum_{j=1}^{J}\\&\left[\left(O^j\ln\left(\frac{g(\sum_{q=1}^{Q}w_{0q}\prod_{i\in\Lambda_q}g(W_i\cdot X^j))}{O^j}\right)\right.\\&+(1-O^j)\\&\left.\ln\left(\frac{1-g(\sum_{q=1}^{Q}w_{0q}\prod_{i\in\Lambda_q}g(W_i\cdot X^j))}{1-O^j}\right)\right]\end{aligned} \tag{III.3}$$

The partial derivative of $\widetilde{E}(W)$ is given by:

$$\begin{aligned}
&\widetilde{E}_{w_0}(W) \\
&= -\sum_{j=1}^{J}\left[\left(\frac{O^j}{g\left(\sum_{q=1}^{Q} w_{0q}\cdot\xi_q^j\right)}\right.\right. \\
&\quad\left.\cdot g'\left(\sum_{q=1}^{Q} w_{0q}\cdot\xi_q^j\right)\xi^j\right. \\
&\quad -\frac{1-O^j}{1-g\left(\sum_{q=1}^{Q} w_{0q}\cdot\xi_q^j\right)} \\
&\quad\left.\left.\cdot g'\left(\sum_{q=1}^{Q} w_{0q}\cdot\xi_q^j\right)\xi^j\right]\right. \\
&= \sum_{j=1}^{J}\left(g\left(\sum_{q=1}^{Q} w_{0q}\cdot\xi_q^j\right)-O^j\right)\xi^j
\end{aligned} \tag{III.4}$$

Furthermore, based on $\xi_q^j = \prod_{i\in\Lambda_q} g(W_i\cdot X^j)$, we have:thus, according to the preceding definitions and (III.1), it follows that

$$\begin{aligned}
&\widetilde{E}_{W_n}(W) \\
&= -\sum_{j=1}^{J}\left[\left(\frac{O^j}{g(w_0\cdot\xi^j)}\cdot g'(w_0\cdot\xi^j)\right.\right. \\
&\quad\left.\left.-\frac{1-O^j}{1-g(w_0\cdot\xi^j)}\cdot g'(w_0\cdot\xi^j)\right]\right. \\
&\quad\sum_{q\in V_n}(w_{0q})\frac{\partial\xi_q^j}{\partial W_n} \\
&= \sum_{j=1}^{J}(g(w_0\cdot\xi^j)-O^j)\sum_{q\in V_n}(w_{0q})\frac{\partial\xi_q^j}{\partial W_n}
\end{aligned} \tag{III.6}$$

B. Graph Regularization Enhanced Entropy-Based Online Gradient Algorithm for SPSNN

To enhance the stability and generalization capability of neural networks, an entropy-based online gradient algorithm with graph regularization for SPSNN (*EEOGR*) is proposed. The proposed method encourages stronger intra-class feature compactness in the hidden layers, leading to improved generalization performance. In contrast to traditional regularization terms, which fail to fully exploit the intrinsic structure of the data, EEOGR effectively incorporates data-driven information to refine the network topology, thereby achieving more robust and discriminative feature representations.

First, a weighted graph is constructed using the categorized data and their associated information. Based on this graph, the corresponding Laplacian matrix is then computed. Finally, a graph-based regularization term is derived from the Laplacian matrix and incorporated into the error function (see Algorithm 1).

Subsequently, the error function incorporating graph regularization can be expressed as:

$$\begin{aligned}
&E(W) \\
&= \widetilde{E}(W)+\frac{\lambda}{2}\phi(W_0,V) \\
&= \widetilde{E}(W)+\frac{\lambda}{2}tr(f(VX)Lf(VX)^T)
\end{aligned} \tag{III.7}$$

where, the regularization coefficient is represented by $\lambda > 0$.

For easier calculations, the penalty term is reformulated as:

$$E(W) = \widetilde{E}(W)+\frac{\lambda}{2}tr((VX)L(VX)^T) \tag{III.8}$$

Building on this form of graph regularization, the gradient of the regularization term can be computed efficiently, as it avoids the need to evaluate the gradient of the activation function.

The partial derivatives of $E(W)$ are derived from the error function (III.8) as follows:

$$\frac{\partial\xi_q^j}{\partial w_n} = \begin{cases} \left(\prod_{i\in\Lambda_q\setminus n}\sigma_i\right)\cdot g'(W_i\cdot X^j)X^j, & \text{if } q\neq 1 \text{ and } n\in\Lambda_q \\ 0, & \text{if } q\equiv 1 \text{ or } n\notin\Lambda_q \end{cases} \tag{III.5}$$

Algorithm 1. The EEOGR algorithm for SPSNN.

1: **Input:** samples $\{X^j, O^j\}_{j=1}^{J}$, initial weight W^0, learning rate η, the regularization coefficient λ.
2: The total number of iterations T, the number of iterations k.
3: **for** $j = 1 : J$ **do**
4: Calculate Gaussian kernel σ^2 by Eq (II.2).
5: Measure similarity by Eq (II.1).
6: **end for**
7: Compute Laplacian matrix L by Eq (II.3).
8: **while** $k \leq T$ **do**
9: Obtain network outputs y^j and weights W^{k-1} by SPSNN.
10: Calculate graph regularization term $\phi(W_0, V)$ by Eq (II.6).
11: Calculate error function $E(W)$ by Eq (III.8).
12: Update weights W^k by Eq (III.11), Eq (III.12) and Eq (III.13).
13: **end while**
14: **Output:** weights W^k.

$$\begin{aligned}
&E_{w_0}(W) \\
&= \sum_{j=1}^{J}\left(g\left(\sum_{q=1}^{Q} w_{0q}\cdot\xi_q^j\right)-O^j\right)\xi^j
\end{aligned} \tag{III.9}$$

$$\begin{aligned}
&E_{w_n}(W) \\
&= \sum_{j=1}^{J}(g(w_0\cdot\xi^j)-O^j)\sum_{q\in V_n}(w_{0q})\frac{\partial\xi_q^j}{\partial W_n} \\
&\quad +\lambda\sum_{n=1}^{N} W_n XLX^T
\end{aligned} \tag{III.10}$$

For any initial guess W^0, the weight updates of the *EEOGR* algorithm during the k-th $(k = 1, 2, \cdots)$ training epoch are determined as follows:

$$\begin{aligned}
&W^{kJ+j} = W^{kJ+j-1} + \triangle W^{kJ+j-1} \\
&k = 1, 2, \cdots
\end{aligned} \tag{III.11}$$

where η_k denotes the learning rate at the $k-th$ training epoch.

IV. Main Results

This section presents the convergence results for the constructed network model and details the assumptions required to validate these results.

$(A1)$ $g(r)$ is defined on its domain R with the property of being twice continuously differentiable. Additionally, the absolute values of $|g(r)|$, $|g'(r)|$ and $|g''(r)|$ are uniformly bounded for every $r \in R$.

$(A2)$ A finite closed region $\Omega \in R^{Q+NP}$ exists such that the sequence $\{W^k\}_{k=0}^{\infty} \subset \Omega$, and the set $\Omega_0 = \{W \in \Omega : \nabla E(W) = 0\}$ contains finite points.

$(A3)$ The learning rate η_k satisfies (a) $0 < \eta_k < 1$,

(b) $\sum_{k=1}^{\infty} \eta_k = \infty$, (c) $\sum_{k=1}^{\infty} \eta_k^2 < \infty$.

For ease of understanding, we define

$$\begin{aligned} C_0 &= \max_{1\le j\le J}\{\|X^j\|, \|O^j\|\}, \\ C_1 &= \max\{\sup_{t\in\mathbb{R}}|g(t)|, \sup_{t\in\mathbb{R}}|g'(t)|, \\ &\quad \sup_{t\in\mathbb{R}}|g''(t)|\}, \\ C_2 &= \max_{1\le q\le Q}|(C_1 - C_0)C_1^q|, \\ C_3 &= \max\{C_1, C_1^N\}. \end{aligned} \tag{IV.1}$$

Under these assumptions, the main results concerning the monotonicity of the error function and the convergence of W^k throughout the gradient-based learning iterations are established using the entropy-based error metric.

Theorem IV.1. *Given that assumptions $(A1)$ and $(A3)$ hold, the weight sequence $\{W^k\}$ remains uniformly bounded.*

Theorem IV.2. *Considering that suppositions $(A1)$, and $(A3)$ are met, the weight sequence $\{W^k\}$ be produced through the iteration algorithm (III.11) from any original value. Additionally, the following property of feeble convergence is maintained*

$$\lim_{k\to\infty} \|E_W(W^k)\| = 0. \tag{IV.2}$$

Theorem IV.3. *Presume the hypotheses $(A1)$, $(A2)$ and $(A3)$ are fulfilled, the weight sequence $\{W^k\}$ be produced through the iteration algorithm (III.11) for an arbitrary starting value, exhibits the following strong convergence property.*

$$\lim_{k\to\infty} W^k = W^*. \tag{IV.3}$$

Remark. Weight boundedness is an essential requirement for establishing convergence, and while most existing studies treat weight boundedness as an assumption, this paper rigorously and effectively demonstrates the boundedness of weights in the graph regularization approach for the SPSNN online gradient learning algorithm using the entropy error function. Moreover, this study reveals that assuming weight boundedness as a prerequisite is no longer necessary, thus simplifying and easing the theoretical analysis compared to conventional approaches.

The proofs of weak and strong convergence mentioned in the text correspond to mathematical concepts. Both convergence theorems in the text refer to the weak and strong convergence of the weight sequence W^k.

TABLE I Overview of benchmark data sets for classification and the main attributes.

DATA SET	INSTANCES	TRAINING SAMPLES	TESTING SAMPLES	INPUT FEATURES	CLASS
Divorce	170	119	51	54	2
parkinson	756	529	227	754	2
diabetes	520	364	156	16	2
Wine	178	124	54	13	3
winequality-red	1599	1119	480	11	10
winequality-white	4898	3428	1470	11	10

$$\triangle W_0^{kJ+j-1} = -\eta_k\left[\left(g\left(\sum_{q=1}^{Q} w_{0q}^{kJ+j-1}\cdot\zeta_q^{kJ+j-1}\right) - O^j\right)\zeta^{kJ+j-1}\right] \tag{III.12}$$

$$\triangle W_n^{kJ+j-1} = -\eta_k\left[\left(g(w_0^{kJ+j-1}\cdot\zeta^{kJ+j-1}) - O^j\right)\sum_{q\in V_n}\left(w_{0q}^{kJ+j-1}\right)\frac{\partial\zeta_q^{kJ+j-1}}{\partial W_n^{kJ+j-1}}\right] + \frac{\lambda}{J}\sum_{n=1}^{N} W_n X L X^T \tag{III.13}$$

V. Numerical Experimental Results

In this section, the numerical performance of the SPSNN with graph regularization is evaluated. To assess the effectiveness of the proposed method, experiments were conducted on 15 classification datasets and 6 regression datasets, which were obtained from the UCI Machine Learning Repository and Kaggle. Tables I and II summarize the characteristics of the classification datasets, while Table III provides detailed information on the regression datasets. Each dataset was randomly divided into a training set comprising 70% of the samples and a testing set containing the remaining 30%. For all datasets, feature values were linearly normalized to the range $[-1, 1]$, and the same normalization parameters were applied to the corresponding test data.

The initial network architecture was configured as $P - 4 - 16 - 1$, with the sigmoid function used as the activation function in the hidden layers. All algorithms were implemented in MATLAB R2021a and executed on a workstation equipped with an Intel Core i5-8300H processor operating at 2.30 GHz and 16 GB of RAM.

A. Classification Problem

Tables I and II provide comprehensive detail about the 15 benchmark real-world categorical datasets. The network is trained using 10-fold cross-validation in each trial. The performance of three online gradient methods was evaluated: the first uses the mean-square error for updates (*SOGM*), the second employs

TABLE II Overview of benchmark data sets for classification and the main attributes.

DATA SET	INSTANCES	TRAINING SAMPLES	TESTING SAMPLES	INPUT FEATURES	CLASS
Wdbc	569	398	171	30	2
Breast	277	193	84	9	2
Pima	768	537	231	8	2
German	1000	700	300	24	2
Phoneme	5404	3782	1622	5	2
Dry bean	13611	9527	4084	16	7
Loangeles	22899	16029	6870	19	6
Health indicators	100000	70000	30000	30	2
Health lifestyle	100000	70000	30000	15	2

TABLE III Overview of benchmark data sets for regression and the main attributes.

DATA SET	INSTANCES	TRAINING SAMPLES	TESTING SAMPLES	INPUT FEATURES
Qsar aquatic toxicity	546	382	164	8
Enb2012	768	538	230	9
Qsar fish toxicity	908	635	273	6
Concrete compressive strength	1030	721	309	8
Air quality	9357	6550	2807	14
Combined cycle power plant	9568	6698	2870	4

an entropy-based error for the same purpose (*EEOGM*), and the third integrates a graph regularization component with the entropy error function (*EEOGR*). Table IV presents the performance comparison results of various methods. Compared to the SOGM algorithm, the EEOGM algorithm demonstrates higher classification accuracy on both the training and test sets, indicating that the entropy-based error function outperforms the mean squared error function in handling classification tasks, thereby enhancing model stability and classification performance. Furthermore, the EEOGR algorithm, which incorporates a graph regularization term based on EEOGM, achieves a significant further improvement in classification accuracy. This demonstrates that the introduction of graph regularization effectively enhances the model's generalization capability.

In addition, comparisons were made among the online gradient algorithm featuring original group lasso regularization (*OOGL*), the one with a smoothed group lasso approach (*OSGL*), the variant employing $L_{1/2}$ group regularization ($OGL_{1/2}$), and the algorithm that includes graph regularization with the entropy error function. Table V displays the performance comparison results, where the *EEOGR* algorithm and the other algorithms are evaluated on the same dataset for training and testing. The *EEOGR* algorithm clearly surpasses the other algorithms on nearly all of the tested classification datasets. Moreover, it is clear that the *EEOGR* algorithm finds more redundant nodes and has a more greater effect on sparsifying weights compared to other algorithms on most datasets, and forces a greater concentration of potential features in the hidden layer, leading to better optimization of the network architecture while maintaining training and testing accuracy.

Parameter Setting: In this section, to assess the model's robustness, the sensitivity of the parameters η and λ is tested in this experiment, and Figure 2 shows the classification accuracies achieved by applying multiple learning rates and penalty term coefficients on the divorce dataset. We can see that there is not much difference in the classification results of scanning the two parameters η and λ over the whole range, and the results are at a high level of accuracy. In addition, the horizontal axis in Figure 2 is the learning rate and the vertical axis is the graph regularization coefficient. The highest accuracy is clearly achieved when the learning rate is $\eta = 0.0002$ and the graph regularization coefficient is set to $\lambda = 0.0001$. The settings and ranges of the parameters for the subsequent step are determined based on their effects on accuracy as illustrated in the graph.

B. Regression Problems

In the following analysis, the generalization capabilities of the four

TABLE IV Performance comparison of SOGM, EEOGM and EEOGR algorithms in classification problems.

DATA SET	TRAINING ACCURACY			TESTING ACCURACY		
	SOGM	EEOGM	EEOGR	SGM	EEOGM	EEOGR
Divorce	0.9832	0.9831	0.9915	0.9412	0.9607	0.9803
Parkinson	0.7415	0.7787	0.7713	0.7556	0.7591	0.7657
Diabetes	0.8216	0.8368	0.8857	0.8122	0.8105	0.8763
Wine	0.9183	0.9561	0.9533	0.9122	0.9263	0.9317
Winequality-red	0.6833	0.7065	0.7458	0.6727	0.6800	0.7267
Winequality-white	0.6462	0.6870	0.6953	0.6599	0.6699	0.6731

TABLE V Performance comparison of OGL, OSGL, $OGL_{1/2}$ and EEOGR algorithms in classification problems.

DATA SET	ALGORITHM	TRAINING ACCURACY	TESTING ACCURACY	SURVIVING NODES	SURVIVING WEIGHTS
Wdbc	OGL	0.8643	0.8187	13	89
	OSGL	0.8655	0.8367	12	89
	$OGL_{1/2}$	0.8889	0.8492	10	91
	EEOGR	0.8772	0.8645	10	83
Breast	OGL	0.7470	0.7145	14	43
	OSGL	0.7229	0.7113	12	39
	$OGL_{1/2}$	0.7526	0.7108	10	37
	EEOGR	0.7680	0.7320	12	37
Pima	OGL	0.7151	0.6883	11	47
	OSGL	0.7505	0.7403	11	36
	$OGL_{1/2}$	0.7430	0.7316	13	33
	EEOGR	0.7579	0.7319	9	33
German	OGL	0.7100	0.6871	13	78
	OSGL	0.7465	0.6957	11	75
	$OGL_{1/2}$	0.7244	0.7033	9	74
	EEOGR	0.7257	0.7167	10	74
Phoneme	OGL	0.7763	0.7565	11	32
	OSGL	0.7696	0.7591	9	32
	$OGL_{1/2}$	0.7496	0.7312	11	30
	EEOGR	0.7840	0.7686	10	29
Dry bean	OGL	0.8741	0.8457	11	62
	OSGL	0.8719	0.8671	11	64
	$OGL_{1/2}$	0.8856	0.8670	11	62
	EEOGR	0.8856	0.8723	10	58
Loangeles	OGL	0.8031	0.7974	10	82
	OSGL	0.8096	0.8050	9	74
	$OGL_{1/2}$	0.8004	0.7996	11	77
	EEOGR	0.8131	0.8109	9	72
Health indicators	OGL	0.7951	0.7143	12	151
	OSGL	0.8431	0.7479	12	149
	$OGL_{1/2}$	0.8039	0.7311	13	152
	EEOGR	0.8235	0.7647	12	149
Health lifestyle	OGL	0.7513	0.7500	11	142
	OSGL	0.7532	0.7486	12	138
	$OGL_{1/2}$	0.7534	0.7479	14	139
	EEOGR	0.7570	0.7527	14	138

algorithms are evaluated across a range of regression tasks, with relevant details summarized in Table III. For each dataset, the number of input nodes corresponds to its features, and the model produces a single output. The network was trained using 10-fold cross-validation in each trial.

As can be seen from Table VI, the *EEOGR* algorithm has less training error and testing error compared to *EEOGM*, which indicates that the *EEOGR* algorithm has more generalization ability and better stability. The housing and QSAR fish toxicity datasets were utilized to exemplify our analysis. Figure 3 and Figure 5 highlight the trends in absolute errors for both training and testing samples with the *EEOGR* and *EEOGM* algorithms applied to these datasets. It is noted that the absolute errors of the *EEOGR* algorithm for the training and test samples of the two datasets are mainly

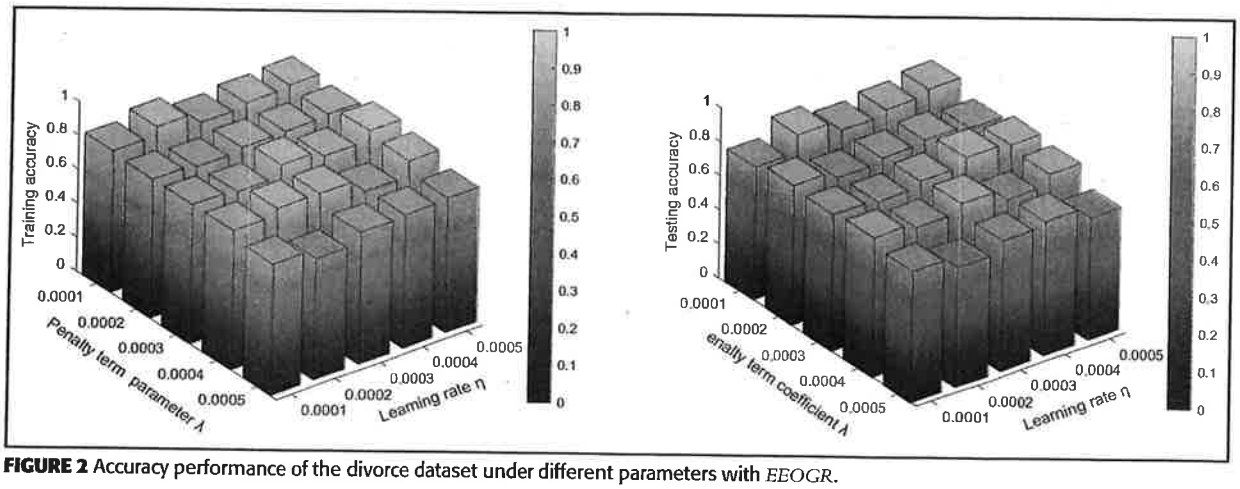


FIGURE 2 Accuracy performance of the divorce dataset under different parameters with *EEOGR*.

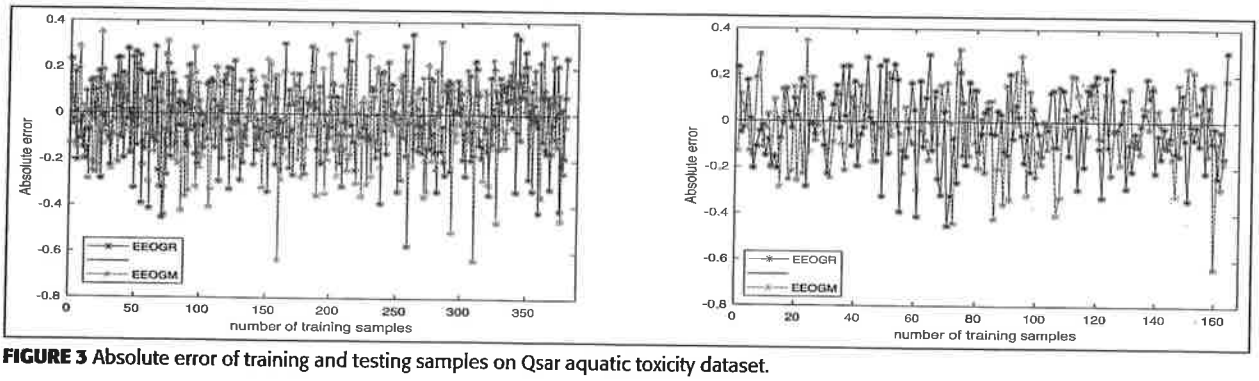


FIGURE 3 Absolute error of training and testing samples on Qsar aquatic toxicity dataset.

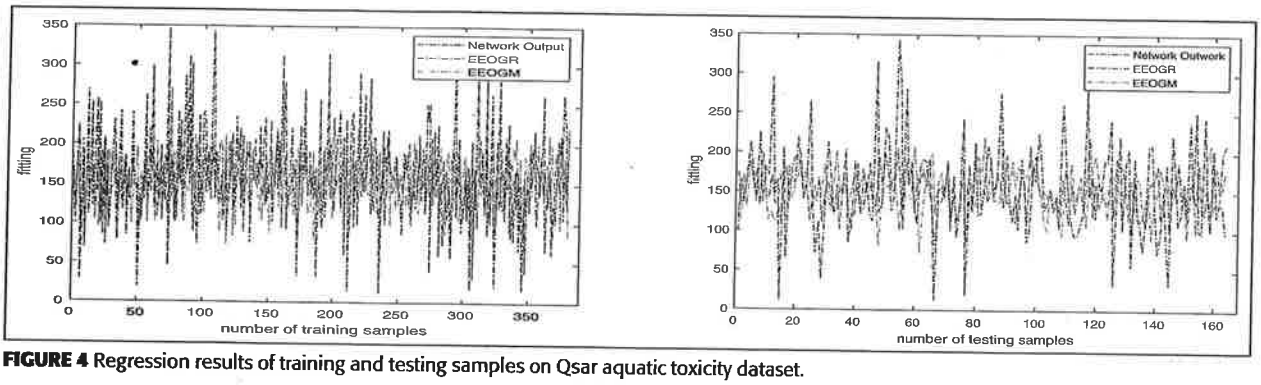


FIGURE 4 Regression results of training and testing samples on Qsar aquatic toxicity dataset.

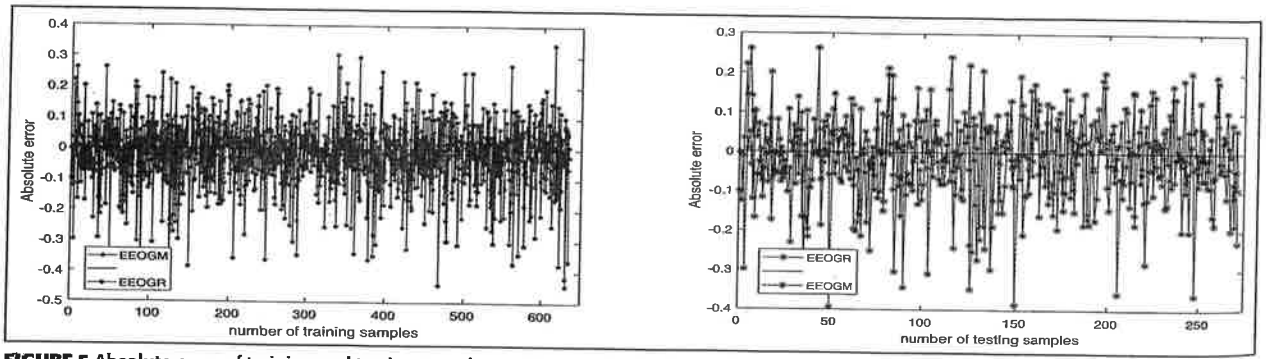


FIGURE 5 Absolute error of training and testing samples on Qsar fish toxicity dataset.

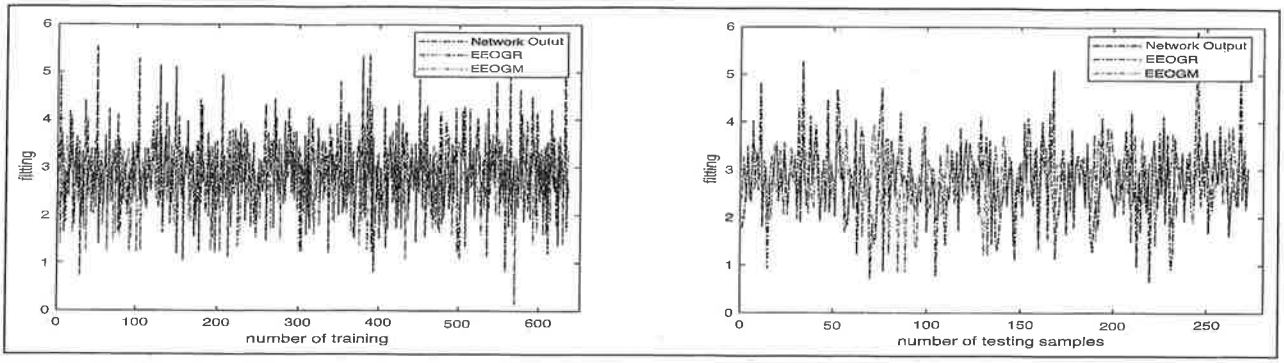


FIGURE 6 Regression results of training and testing samples on Qsar fish toxicity dataset.

concentrated in $[-0.4, 0.3]$ and $[-0.3, 0.2]$, with a relatively concentrated and stable distribution, indicating that the algorithms of this paper have good effectiveness and generalization performance. Figure 4 and Figure 6 demonstrate the fitting performance of the *EEOGR* and *EEOGM* algorithms across both datasets. We can clearly see that, compared with *EEOGM*, *EEOGR* algorithm has higher overlap, which means that *EEOGR* algorithm has superior data fitting capability.

TABLE VI Performance comparison of EEOGM and EEOGR algorithms in regression problems.

DATA SET	TRAINING ERROR		TESTING ERROR	
	EEOGM	EEOGR	EEOGM	EEOGR
qsar aquatic toxicity	0.1301	0.0678	0.0675	0.0452
ENB2012	0.0846	0.0693	0.0375	0.0282
qsar fish toxicity	0.0382	0.0332	0.0229	0.0195
concrete compressive strength	0.0657	0.0695	0.0282	0.0192
Air Quality	0.0506	0.0334	0.0307	0.0283
combined cycle power plant	0.0587	0.0557	0.0470	0.0374

VI. Conclusion

This paper presents a novel graph regularization approach for the online gradient learning algorithm of the Sigma-Pi-Sigma Neural Network (SPSNN), utilizing the entropy error function. The proposed methodology involves constructing a weighted graph based on input data and their associated class labels. Based on this weighted graph, the Laplacian matrix is computed to derive the graph regularization, which exploits the graph's topological structure to enhance the network's capacity for modeling graph-structured data. This regularization promotes greater concentration of latent features in the hidden layers, thereby improving overall model performance.

In addition, an entropy-based error function is introduced to further optimize the training process, enhancing the models ability to capture graph-structured data. The boundedness of the weights is established in a simple and efficient manner, and rigorous proofs are provided for both weak and strong convergence of the algorithm. Compared to existing methods, the proposed approach reduces computational complexity and improves learning efficiency.

In future work, we aim to explore the extension of this graph regularization method to more complex neural network architectures, as well as investigate its application to dynamic graph-structured data. Additionally, further optimization techniques could be developed to improve the scalability and robustness of the algorithm for large-scale data sets.

Appendix

Before delving into the proof of the primary theorem, we need to establish some preliminary notations and lemmas. Some of the symbols used in this paper are shown in Table VII.

Lemma A.1. *Suppose the existence of sequences D_t, E_t and F_t, where E_t is nonnegative and D_t is bounded for all t. If*

$$D_{t+1} \leq D_t - E_t + F_t, t = 0, 1, \ldots, \tag{A.1}$$

and the series $\sum_{t=0}^{\infty} F_t$ is convergent, Then, D_t converges to a finite value and $\sum_{t=0}^{\infty} E_t < \infty$.

Lemma A.2. *Assuming that the learning rate η_k satisfies assumption (A3), and the sequence a_k $k \in \mathbf{N}$ fulfils $a_k \geq 0$, $\sum_{k=0}^{\infty} \eta_k a_k^{\beta} < \infty$ and $|a_{k+1} - a_k| \leq \mu\eta_k$ for positive constants β and μ, then we get*

$$\lim_{k \to \infty} a^k = 0. \tag{A.2}$$

Lemma A.3. *Assume that $f: R^P \to R$ is continuous and differentiable on a compact set $\Omega \in R^P$, and that $\Omega_0 = \{W \in \Omega \mid \nabla f(W) = 0\}$ consists of a finite number of points. Consider a sequence $\{W^k\}_{k=0}^{\infty} \subset \Omega$ fulfills*

$$\lim_{m\to\infty} \|W^{k+1} - W^k\| = 0, \text{and } \lim_{k\to\infty} \| \nabla f(W^k)\| = 0, \quad \text{(A.3)}$$

In that case, there is a $W^* \in \Omega_0$ for which $\lim_{k\to\infty} W^k = W^*$ holds.

Proof of Theorem IV.1 Under the assumptions (A3), the sequence $S_k = \eta_0 + \eta_1 + \cdots + \eta_k$ is convergent. According to the Cauchy convergence test, for any $\forall \varepsilon > 0$, there exists a positive integer $N_1 \in \mathbb{N}$, $\forall k > N_1$ and $b \in \mathbb{N}$, we have

$$|S_{k+b} - S_k| = \eta_k + \eta_{k+1} + \cdots + \eta_{k+b-1} < \varepsilon \quad \text{(A.4)}$$
$$|S_{k+b+1} - S_k| = \eta_{k+1} + \eta_{k+2} + \cdots + \eta_{k+b} < \varepsilon \quad \text{(A.5)}$$

By assumption (A1) and updating (III.11),(III.12) and (III.13), we get

$$\begin{aligned} &|w_{oq}^{kJ+j} - w_{oq}^{kJ+j-1}| \\ &= \eta_k |\Delta_j^k w_{oq}^{kJ+j-1}| \\ &= \eta_k \left| \left(g\left(\sum_{q=1}^{Q} w_{0q}^{kJ+j-1} \cdot \zeta_q^{kJ+j-1} \right) - O^j \right) \zeta_q^{kJ+j-1} \right| \\ &\le \eta_k C_2 \end{aligned} \quad \text{(A.6)}$$

where $C_2 = \sup |(g(\sum_{q=1}^{Q} w_{0q}^{kJ+j-1} \cdot \zeta_q^{kJ+j-1}) - O^j)\zeta_q^{kJ+j-1}|$.

$$\begin{aligned} &|w_{oq}^{(k+1)J+j} - w_{oq}^{kJ+j}| \\ &\le |w_{oq}^{(k+1)J+j} - w_{oq}^{kJ+j-1}| + |w_{oq}^{(k+1)J+j-1} - w_{oq}^{kJ+j-2}| \\ &\quad + \cdots + |w_{oq}^{(k+1)J+j} - w_{oq}^{kJ+J-1}| + |w_{oq}^{kJ+J-1} - w_{oq}^{kJ+J-2}| \\ &\quad + \cdots + |w_{oq}^{(kJ+j+1} - w_{oq}^{kJ+j}| \\ &\le j\eta_k A_1 + (J-j)\eta_k C_2 \\ &\le (j\eta_k + (J-j)\eta_k) C_2 \end{aligned} \quad \text{(A.7)}$$

$$\begin{aligned} &|w_{oq}^{(k+p)J+j} - w_{oq}^{kJ+j}| \\ &\le |w_{oq}^{(k+p)J+j} - w_{oq}^{(k+p-1)J+j}| + |w_{oq}^{(k+p-1)J+j} - w_{oq}^{(k+p-2)J+j}| \\ &\quad + \cdots + |w_{oq}^{((k+1)J+j} - w_{oq}^{kJ+j}| \\ &\le C_2 j(\eta_{k+p} + \eta_{k+p-1} + \cdots + \eta_{k+1}) \\ &\quad + C_2(J-j)(\eta_{k+p-1} + \eta_{k+p-2} + \cdots + \eta_k) \\ &\le JC_2\varepsilon \end{aligned} \quad \text{(A.8)}$$

Thus, the sequence of weights w_{oq}^{kJ+j} is a convergent sequence, and by the nature of convergent sequences, we know that w_{oq}^{kJ+j} is a bounded sequence. A constant $M_1 > 0$ can be found such that

$$\| w_{oq}^{kJ+j} \| \le M_1 \quad \text{(A.9)}$$

In the same way, we can obtain that the sequence of weights w_{np}^{kJ+j} is bounded. In conclusion, it can be established that the sequence of weights W^{kJ+j} is uniformly bounded, implying that there is a constant $M > 0$ for which

$$\| W^{kJ+j} \| \le M \quad \text{(A.10)}$$

TABLE VII Symbols and definitions used.

NOTATION	DEFINITIONS
X^j	The given sample.
O^j	The target output.
y^j	The final actual output.
$\tilde{E}(w)$	The error function.
$\phi(W_0, V)$	The given sample.
V	The hidden layer weight matrix.
w_i	The weight vector.
L	The Laplace matrix.
ζ_q	The output of the $\prod$ layer.
η^k	The learning rate.
λ	The Penalty coefficient.

It is apparent that there is also a constant $\overline{M} > 0$ for which

$$\| \Delta_j^k W^{kJ+j} \| \le \overline{M} \quad \text{(A.11)}$$

Theorem IV.1 has been proven. □

Proof of Theorem IV.2

For ease of reference, we establish the following notations:

$$\begin{aligned} &\psi^{n,j,j} = \zeta^{kJ+j-1,j} - \zeta^{kJ,j}, \quad \psi^{n,J+1,j} = \zeta^{(k+1)J,j} - \zeta^{kJ,j} \\ &\varphi_n^{kJ} = W_n^{kJ} \cdot X^j, d_n^{k,j} = w_n^{kJ+j-1} - w_n^{kJ} \\ &r_n^{k,j} = \Delta_j^k w_n^{kJ+j-1} - \Delta_j^k w_n^{kJ} \end{aligned} \quad \text{(A.12)}$$

For every $n = 0, 1, \ldots, N$ and $k = 0, 1, 2, \cdots$

The Taylor mean value theorem is used to derive:

$$\begin{aligned} &E(W^{(k+1)J}) - E(W^{kJ}) \\ &= -\sum_{j=1}^{J} \left[\left(O^j \ln \left(\frac{g(w_0^{(k+1)J} \cdot \zeta^{(k+1)J})}{O^j} \right) \right. \right. \\ &\quad \left. \left. + (1 - O^j) \ln \left(\frac{1 - g(w_0^{(k+1)J} \cdot \zeta^{(k+1)J})}{1 - O^j} \right) \right] \right. \\ &\quad + \sum_{j=1}^{J} \left[\left(O^j \ln \left(\frac{g(w_0^{kJ} \cdot \zeta^{kJ})}{O^j} \right) \right. \right. \\ &\quad \left. \left. + (1 - O^j) \ln \left(\frac{1 - g(w_0^{kJ} \cdot \zeta^{kJ})}{1 - O^j} \right) \right] \right. \\ &\quad + \frac{\lambda}{2} \sum_{n=1}^{N} (W_n^{(k+1)J} X L X^T (W_n^{(k+1)J})^T - W_n^{kJ} X L X^T (W_n^{kJ})^T) \\ &= -\sum_{j=1}^{J} \left[\left(O^j \ln \left(\frac{g(w_0^{(k+1)J} \cdot \zeta^{(k+1)J})}{O^j} \right) \right. \right. \\ &\quad \left. \left. - \left(O^j \ln \left(\frac{g(w_0^{kJ} \cdot \zeta^{kJ})}{O^j} \right) \right) \right] \right. \\ &\quad - \sum_{j=1}^{J} [(1 - O^j) \ln \left(\frac{1 - g(w_0^{(k+1)J} \cdot \zeta^{(k+1)J})}{1 - O^j} \right) \\ &\quad \left. - (1 - O^j) \ln \left(\frac{1 - g(w_0^{kJ} \cdot \zeta^{kJ})}{1 - O^j} \right) \right] \\ &\quad + \frac{\lambda}{2} \sum_{n=1}^{N} (W_n^{(k+1)J} X L X^T (W_n^{(k+1)J})^T - W_n^{kJ} X L X^T (W_n^{kJ})^T) \end{aligned}$$

$$
\begin{aligned}
&= -\sum_{j=1}^{J}\left[\left(O^j\frac{g'(w_0^{kJ}\cdot\zeta^{kJ})}{g(w_0^{kJ}\cdot\zeta^{kJ})}-(1-O^j)\frac{g'(w_0^{kJ}\cdot\zeta^{kJ})}{1-g(w_0^{kJ}\cdot\zeta^{kJ})}\right]\right.\\
&\quad\cdot(w_0^{(k+1)J}\cdot\zeta^{(k+1)J}-w_0^{kJ}\cdot\zeta^{kJ})\\
&\quad+\frac{1}{2}g''(w_0^{kJ}\cdot\zeta^{kJ})\cdot(w_0^{(k+1)J}\cdot\zeta^{(k+1)J}-w_0^{kJ}\cdot\zeta^{kJ})^2\\
&\quad\left.+\frac{\lambda}{2}\sum_{n=1}^{N}(W_n^{(k+1)J}XLX^T(W_n^{(k+1)J})^T-W_n^{kJ}XLX^T(W_n^{kJ})^T\right)\\
&=\sum_{j=1}^{J}(g(w_0^{kJ}\cdot\zeta^{kJj})-O^j)\cdot d_0^{kJ+1}\cdot\zeta^{kJj}\\
&\quad+\sum_{j=1}^{J}(g(w_0^{kJ}\cdot\zeta^{kJj})-O^j)w_0^{kJ}\cdot\psi^{nJ+1j}+\sigma_1+\sigma_2\\
&\quad+\frac{\lambda}{2}\sum_{n=1}^{N}(W_n^{(k+1)J}XLX^T(W_n^{(k+1)J})^T-W_n^{kJ}XLX^T(W_n^{kJ})^T)
\end{aligned}
\tag{A.13}
$$

where $\sigma_1=\sum_{j=1}^{J}(g(w_0^{kJ}\cdot\zeta^{kJj})-O^j)\cdot d_0^{kJ+1}\cdot\psi^{nJ+1j}$, $\sigma_2=\frac{1}{2}g''(w_0^{kJ}\cdot\zeta^{kJ})\cdot(w_0^{(k+1)J}\cdot\zeta^{(k+1)J}-w_0^{kJ}\cdot\zeta^{kJ})^2$

From (III.12), it follows that

$$
\begin{aligned}
&\sum_{j=1}^{J}(g(w_0^{kJ}\cdot\zeta^{kJj})-O^j)\cdot d_0^{kJ+1}\cdot\zeta^{kJj}\\
&=\sum_{j=1}^{J}(g(w_0^{kJ}\cdot\zeta^{kJj})-O^j)\cdot\sum_{q=1}^{Q}d_{0q}^{kJ+1}\cdot\zeta_q^{kJj}\\
&=\sum_{j=1}^{J}\sum_{q=1}^{Q}(g(w_0^{kJ}\cdot\zeta^{kJj})-O^j)\cdot\zeta_q^{kJj}\cdot d_{0q}^{kJ+1}\\
&=\sum_{j=1}^{J}\Delta_j^k w_0^{kJ}\cdot d_0^{kJ+1}
\end{aligned}
\tag{A.14}
$$

Then, according to (III.1), (IV.1), there exists a constant $C_3=\max\{C_1,C_1^N\}$ such as

$$
\|\zeta_q\|\le C_3,\ 1\le q\le Q \tag{A.15}
$$

To complete the proof outlined above, the following identity is required

$$
\Pi_{n=1}^{N}a_n-\Pi_{n=1}^{N}b_n=\Sigma_{n=1}^{N}(\Pi_{s=1}^{n-1}a_s)(\Pi_{r=n+1}^{N}b_r)(a_n-b_n) \tag{A.16}
$$

where we have made the convention that $\Pi_{s=1}^{0}a_s\equiv 1$ and $\Pi_{r=N+1}^{N}b_r\equiv 1$. This can be readily established through mathematical induction. Thus, the proof is omitted.

Then, employing (A.12) and (A.16), we can obtain

$$
\begin{aligned}
\psi_q^{kJ+1j}&=\zeta_q^{(k+1)Jj}-\zeta_q^{kJj}\\
&=\prod_{n\in\Lambda_q}g(W_n^{(k+1)J}\cdot X^j)-\prod_{n\in\Lambda_q}g(W_n^{kJ}\cdot X^j)\\
&=\prod_{n\in\Lambda_q}g(\varphi_n^{(k+1)Jj})-\prod_{n\in\Lambda_q}g(\varphi_n^{kJj})\\
&=\sum_{n\in\Lambda_q}\left(\prod_{s\in\Lambda'_{q,n}}g(\varphi_s^{kJj})\right)\left(\prod_{r\in\Lambda''_{q,n}}g(\varphi_r^{kJj})\right)\\
&\quad\times(g(\varphi_n^{(k+1)Jj})-g(W_n^{kJ}\cdot X^j))\\
&=\sum_{n\in\Lambda_q}\rho_{q,n}^{kJ,(k+1)Jj}\times(g(\varphi_n^{(k+1)Jj})-g(\varphi_n^{kJj}))
\end{aligned}
\tag{A.17}
$$

where $\rho_{q,n}^{kJ,(k+1)Jj}=(\prod_{s\in\Lambda'_{q,n}}g(\varphi_s^{kJj}))(\prod_{r\in\Lambda''_{q,n}}g(\varphi_r^{kJj}))$, $\Lambda'_{q,n}=\{u|u<n,u\in\Lambda_q\}$ and $\Lambda''_{q,n}=\{u|u>n,u\in\Lambda_q\}$.

Based on assumption $A(1)$ and (A.17), we can get

$$
|\rho_{q,n}^{kJ,(k+1)Jj}|\le(C_1^{N-1}),\forall q,n,k,j \tag{A.18}
$$

Then, by using (A.17), we can derive

$$
\begin{aligned}
&\sum_{j=1}^{J}(g(w_0^{kJ}\cdot\zeta^{kJj})-O^j)w_0^{kJ}\cdot\psi^{nJ+1j}\\
&=\sum_{j=1}^{J}(g(w_0^{kJ}\cdot\zeta^{kJj})-O^j)\sum_{q=1}^{Q}w_{oq}^{kJ}\left(\sum_{n\in\Lambda_q}\rho_{q,n}^{kJ,(k+1)Jj}\right.\\
&\quad\left.\times(g(\varphi_n^{(k+1)Jj})-g(\varphi_n^{kJj}))\right)\\
&=\sum_{j=1}^{J}(g(w_0^{kJ}\cdot\zeta^{kJj})-O^j)\sum_{q=1}^{Q}w_{oq}^{kJ}\left(\sum_{n\in\Lambda_q}\rho_{q,n}^{kJ,(k+1)Jj}\right.\\
&\quad\left.\times\left[g'(\varphi_n^{kJj})d_n^{kJ+1}\cdot X^j+\frac{1}{2}g''(\varphi_n^{kJj})(d_n^{kJ+1}\cdot X^j)^2\right]\right.\\
&=\sum_{j=1}^{J}(g(w_0^{kJ}\cdot\zeta^{kJj})-O^j)\sum_{q=1}^{Q}w_{oq}^{kJ}\left(\sum_{n\in\Lambda_q}\left(\prod_{i\in\Lambda_{q\setminus n}}\delta_i^{kJj}\right)\right.\\
&\quad\left.\times g'(\varphi_n^{kJj})d_n^{kJ+1}\cdot X^j+\sigma_3\right.\\
&=\sum_{j=1}^{J}\sum_{n=1}^{N}\sum_{p=1}^{P}\Delta_j^k w_n^{kJ}d_{np}^{kJ+1}+\sigma_3
\end{aligned}
\tag{A.19}
$$

where

$$
\begin{aligned}
\sigma_3=&\sum_{j=1}^{J}(g(w_0^{kJ}\cdot\zeta^{kJj})-O^j)\frac{1}{2}g''(\varphi_n^{kJj})(d_n^{kJ+1}\cdot X^j)^2\\
&\sum_{q=1}^{Q}w_{oq}^{kJ}\sum_{n\in\Lambda_q}\rho_{q,n}^{kJ,(k+1)Jj}
\end{aligned}
\tag{A.20}
$$

Using (III.11), we have

$$
\begin{aligned}
d_n^{kj}&=w_n^{kJ+j-1}-w_n^{kJ}\\
&=-\eta_k\Delta_j^k w_n^{kJ+j-2}-\eta_k\Delta_j^k w_n^{kJ+j-3}\\
&\quad-\cdots-\eta_k\Delta_j^k w_n^{kJ+1}-\eta_k\Delta_j^k w_n^{kJ}\\
&=-\eta_k\sum_{l=1}^{j}(\Delta_j^k w_n^{kJ}+r_n^{k,t})
\end{aligned}
\tag{A.21}
$$

where $n=0,1,\ldots,N$.

And

$$
\begin{aligned}
&W_n^{(k+1)J}XL(W_n^{(k+1)J}X)^T-W_n^{kJ}XL(W_n^{kJ}X)^T\\
&\quad=2W_n^{kJ}XLX^T(d_n^{kJ+1})^T+d_n^{kJ+1}XLX^T(d_n^{kJ+1})^T\\
&\quad\le 2W_n^{kJ}XLX^T(d_n^{kJ+1})^T+C_4\parallel d_n^{kJ+1}\parallel^2
\end{aligned}
\tag{A.22}
$$

where $C_4=\sup(XLX^T)$.

Then, we can obtain

$$
\begin{aligned}
&E(W^{(k+1)J})-E(W^{kJ})\\
&=\sum_{j=1}^{J}\Delta_j^k w_o^{kJ}\cdot d_0^{kJ+1}+\sum_{j=1}^{J}\sum_{n=1}^{N}\Delta_j^k w_n^{kJ}d_n^{kJ+1}\\
&\quad+\frac{\lambda}{2}\sum_{n=1}^{N}W_n^{(k+1)J}XL(W_n^{(k+1)J}X)^T-\frac{\lambda}{2}\sum_{n=1}^{N}W_n^{kJ}XL(W_n^{kJ}X)^T\\
&\quad+\sigma_1+\sigma_2+\sigma_3
\end{aligned}
$$

$$= -\eta_k \|E_w(w^{kJ})\|^2 - \eta_k \left(\sum_{j=1}^{J} \Delta_j^k w_0 \cdot \sum_{j=1}^{J} r_0^{kj} \right) - \sum_{n=1}^{N} \eta_k \left(\sum_{j=1}^{J} \Delta_j^k w_n \cdot \sum_{j=1}^{J} r_n^{kj} \right) + \frac{C_4 \lambda}{2} \sum_{n=1}^{N} \| d_n^{kJ+1} \|^2 + \sigma_1 + \sigma_2 + \sigma_3 \tag{A.23}$$

We showed that $\Delta_j^k w_n^{kJ+j-1}$ is bounded, and based on the boundedness of the weights, we can derive

$$\begin{aligned} \| d_0^{kJ+1} \|^2 &\le \sum_{q=1}^{Q} (d_{0q}^{kJ+1})^2 \\ &\sum_{q=1}^{Q} \left(-\eta_k \sum_{j=1}^{J} \right) \Delta_j^k w_{0q}^{kJ+j-1})^2 \\ &\le QJ^2 \overline{M}^2 (\eta_k)^2 \\ &\le C_5 (\eta_k)^2 \end{aligned} \tag{A.24}$$

$$\begin{aligned} \| d_n^{kJ+1} \|^2 &\le \sum_{n=1}^{N} \sum_{j=1}^{J} (d_{np}^{kJ+1})^2 \\ &\sum_{n=1}^{N} \sum_{j=1}^{J} \left(-\eta_k \sum_{j=1}^{J} \right) \Delta_j^k w_{np}^{kJ+j-1})^2 \\ &\le NPJ^2 \overline{M}^2 (\eta_k)^2 \\ &\le C_6 (\eta_k)^2 \end{aligned} \tag{A.25}$$

where $C_5 = QJ^2\overline{M}^2$, $C_6 = NPJ^2\overline{M}^2$.

According to the Lagrangian median theorem, assuming (A1), (IV.1), (A.11), (A.21), we have

$$\begin{aligned} &\| \psi^{k,l,j} \| \\ &= \| \zeta^{kJ+l-1,j} - \zeta^{kJ,j} \| \\ &= \| \sum_{n \in \Lambda_q} \rho^{\vec{0},kJ+l-1,j} \times (g(w_n^{kJ+l-1} \cdot X^i) - g(w_n^{kJ} \cdot X^i)) \| \\ &\le C_1^{N-1} \sum_{n \in \Lambda_q} \| (g(w_n^{kJ+l-1} \cdot X^i) - g(w_n^{kJ} \cdot X^i)) \| \\ &\le C_1^{N-1} \sum_{n \in \Lambda_q} \| g'(w_n^{kJ} \cdot X^i)(w_n^{kJ+l-1} \cdot X^i - w_n^{kJ} \cdot X^i) \| \\ &\le \eta_k C_1^N C_0 \sum_{n \in \Lambda_q} \sum_{t=1}^{l-1} \Delta_t^k w_n^{kJ+t-1} \\ &\le C_1^N C_0 N (l-1) \overline{M} \eta_k \end{aligned} \tag{A.26}$$

And by the continuity of $g'(t)$,(A.12) and (A.15), we obtain

$$\begin{aligned} &| g(w_0^{kJ+j-1} \cdot \zeta^{kJ+j-1,j}) - O^j - (g(w_0^{kJ} \cdot \zeta^{kJ+j-1,j}) - O^j) | \\ &\le | g'(w_0^{kJ} \cdot \zeta^{kJ+J-1,j})(w_0^{kJ+j-1} - w_0^{kJ}) \zeta^{kJ+J-1,j} | \\ &\le C_1 C_2 \| d_0^{kj} \| \end{aligned} \tag{A.27}$$

$$\begin{aligned} &| g(w_0^{kJ} \cdot \zeta^{kJ+j-1,j}) - O^j - (g(w_0^{kJ} \cdot \zeta^{kJ,j}) - O^j) | \\ &\le | g'(w_0^{kJ} \cdot \zeta^{kJ+J-1,j}) w_0^{kJ} (\zeta^{kJ+J-1,j} - \zeta^{kJ,j}) | \\ &\le C_1 \overline{M} \| \psi^{k,j,j} \| \end{aligned} \tag{A.28}$$

By (A.12), we can obtain

$$\begin{aligned} r_0^{kj} &= |\Delta^{kj} w_0^{kJ+j-1} - \Delta^{kj} w_0^{kJ}| \\ &= |(g(w_0^{kJ+j-1} \cdot \zeta^{kJ+j-1,j}) - O^j) \zeta^{kJ+j-1,j} \\ &\quad - g(w_0^{kJ} \cdot \zeta^{kJ,j}) - O^j) \zeta^{kJ,j}| \\ &= |(g(w_0^{kJ+j-1} \cdot \zeta^{kJ+j-1,j}) - O^j) \cdot (\zeta^{kJ+j-1,j} - \zeta^{kJ,j}) \\ &\quad + (g(w_0^{kJ+j-1} \cdot \zeta^{kJ+j-1,j}) - g(w_0^{kJ} \cdot \zeta^{kJ+j-1,j})) \times \zeta^{kJ,j}| \\ &= (g(w_0^{kJ+j-1} \cdot \zeta^{kJ+j-1,j}) - O^j) \\ &\quad \times \left(\prod_{n \in \Lambda_q} g(w_n^{kJ+j-1} \cdot X^i) - \prod_{n \in \Lambda_q} g(w_n^{kJ} \cdot X^j) \right) \\ &\quad + (g(w_0^{kJ+j-1} \cdot \zeta^{kJ+j-1,j} - g(w_0^{kJ} \cdot \zeta^{kJ+j-1,j}) \\ &\quad \times \prod_{n \in \Lambda_q} g(w_n^{kJ} \cdot X^j) \\ &\quad + (g(w_0^{kJ} \cdot \zeta^{kJ+j-1,j} - g(w_0^{kJ} \cdot \zeta^{kJ,j})) \times \prod_{n \in \Lambda_q} g(w_n^{kJ} \cdot X^j) \\ &\le (C_1 + C_0) \| \psi^{k,j,j} \| + C_1^{\Lambda_q+1} C_2 \| d_0^{kj} \| + C_1^{\Lambda_q+1} \overline{M_1} | \psi^{k,j,j} \| \\ &\le C_1^N C_0 (C_1 + C_0) N (J-1) \overline{M} \eta_k + (j-1) C_1^{\Lambda_q+2} C_2 \eta_k \\ &\quad + (j-1) C_1^{N+\Lambda_{q+1}} C_0 N \overline{M} \eta_k \\ &\le C_7 \eta_k \end{aligned} \tag{A.29}$$

where $C_7 = C_1^N C_0 (C_1 + C_0) N (J-1) \overline{M} + (j-1) C_1^{\Lambda_q+2} C_2 + (j-1) C_1^{N+\Lambda_{q+1}} C_0 N \overline{M}$

Similar to the calculation of (A.28), we can obtain

$$r_n^{kj} \le C_8 \eta_k \tag{A.30}$$

where $C_8 = C_0^2 C_1^2 C_3 Q M_1 \overline{M} (j-1) + C_0^3 + C_1 C_2 Q M_1 \overline{M} (j-1) + \frac{2\lambda}{J} C_4 M \overline{M} N (j-1) + N C_4 \overline{M} (j-1)$

Then, according to (A.11), (A.29), (A.30), we have

$$\begin{aligned} &- \eta_k \left(\sum_{j=1}^{J} \Delta_j^k w_0^{kJ} \cdot r_0^{kj} \right) \\ &\le -\eta_k J^2 C_7 \overline{M} \eta_k \\ &\le J^2 C_7 \overline{M} (\eta_k)^2 \\ &\le C_9 (\eta_k)^2 \end{aligned} \tag{A.31}$$

and

$$\begin{aligned} &- \eta_k \left(\sum_{j=1}^{J} \Delta_j^k w_n^{kJ} \cdot r_n^{kj} \right) \\ &\le -\eta_k J^2 C_7 \overline{M} \eta_k \\ &\le J^2 C_8 \overline{M} (\eta_k)^2 \\ &\le C_{10} (\eta_k)^2 \end{aligned} \tag{A.32}$$

where $C_9 = J^2 C_7 \overline{M}$, $C_{10} = J^2 C_8 \overline{M}$.

By applying assumption $A(1)$, (IV.1), (A.24), (A.26) and Cauchy-Schwartz inequality, we obtain

$$
\begin{aligned}
|\sigma_2| &= \left|\sum_{j=1}^{J}(g(w_0^{kJ}\cdot\zeta^{kJj})-O^j)\cdot d_0^{kJ+1}\psi^{kJ+1,j}\right| \\
&\le (C_1+C_0)\left|\sum_{j=1}^{J} d_0^{kJ+1}\psi^{kJ+1,j}\right| \\
&\le \frac{C_1+C_0}{2}\sum_{j=1}^{J}(\|d_0^{kJ+1}\|^2+\|\psi^{kJ+1,j}\|^2) \\
&\le \frac{1}{2}J(C_1+C_0)(C_5+C_0C_1^{2N}C_6N^2J)\eta_k^2 \\
&\le C_{11}\eta_k^2
\end{aligned} \tag{A.33}
$$

where $C_{11}=\frac{1}{2}J(C_1+C_0)(C_5+C_0C_1^{2N}C_6N^2J)$.

By the same token, we can obtain

$$
\begin{aligned}
|\sigma_1| &= \left|\frac{1}{2}\sum_{j=1}^{J} g''(w_0^{kJ}\cdot\zeta^{kJj})(w_0^{(k+1)J}\cdot\zeta^{k+1)Jj}-w_0^{kJ}\cdot\zeta^{kJj})^2\right| \\
&\le \frac{C_1}{2}\left|\sum_{j=1}^{J}(d_0^{kJ+1}\cdot\zeta^{k+1)Jj}+w_0^{kJ}\cdot\psi^{kJ+1,j})^2\right| \\
&\le \frac{C_1}{2}\sum_{j=1}^{J}(C_2^2\|d_0^{kJ+1}\|^2+\overline{M_1}^2\|\psi^{kJ+1,j}\|^2) \\
&\le C_{12}\sum_{j=1}^{J}(\|d_0^{kJ+1}\|^2+\|\psi^{kJ+1,j}\|^2) \\
&\le C_{11}C_{12}J\eta_k^2 \\
&\le C_{13}\eta_k^2
\end{aligned} \tag{A.34}
$$

where $C_{12}=\max\{C_1C_2^2, C_1\overline{M_1}^2\}$, $C_{13}=C_{11}C_{12}J$.

And

$$
\begin{aligned}
|\sigma_3| &= \left|\frac{1}{2}\sum_{j=1}^{J} g''(\varphi^{kJj})(d_n^{kJ+1}\cdot X^j)^2(g(w_0^{kJ}\cdot\zeta^{kJj})-O^j)\right. \\
&\quad \left.\cdot\sum_{q=1}^{Q} w_{0q}^k J\sum_{n=1}^{N}\rho^{kJ,(k+1)Jj}\right| \\
&\le (C_1+C_0)C_1^{N-1}JQ\overline{M_1}\cdot\frac{1}{2}C_1C_0^2\sum_{n=1}^{N}(d_n^{kJ+1})^2 \\
&\le \frac{1}{2}C_0^2C_1^N(C_1+C_0)C_5Q\overline{M_1}\eta_k^2 \\
&\le C_{14}\eta_k^2
\end{aligned} \tag{A.35}
$$

where $C_{14}=\frac{1}{2}C_0^2C_1^N(C_1+C_0)C_5Q\overline{M_1}$.

Consequently, we can ascertain

$$
\begin{aligned}
&E(W^{(k+1)J})-E(W^{kJ}) \\
&= -\eta_k\|E_W(W^{kJ})\|^2-\eta_k\sum_{j=1}^{J}\Delta_j^k w_0^{kJ}\cdot\sum_{j=1}^{J} r_0^{kJ} \\
&\quad -\eta_k\sum_{n=1}^{N}\sum_{j=1}^{J}\Delta_j^k w_n^{kJ}\cdot\sum_{j=1}^{J} r_n^{kJ}+\sigma_1+\sigma_2+\sigma_3 \\
&\le -\eta_k\|E_W(W^{kJ})\|^2+C_9\eta_k^2+C_{10}\eta_k^2+C_{11}\eta_k^2+C_{12}\eta_k^2 \\
&\quad +C_{13}\eta_k^2+C_{14}\eta_k^2+\frac{\lambda}{2}C_4C_6\eta_k^2 \\
&\le -\eta_k\|E_W(W^{kJ})\|^2+(C_9+C_{10}+C_{11}+C_{12}+C_{13} \\
&\quad +C_{14}+\frac{\lambda}{2}C_4C_6)\eta_k^2 \\
&\le -\eta_k\|E_W(W^{kJ})\|^2+C_{15}\eta_k^2
\end{aligned} \tag{A.36}
$$

where $C_{15}=C_9+C_{10}+C_{11}+C_{12}+C_{13}+C_{14}+\frac{\lambda}{2}C_4C_6$.

As a result, we can infer that

$$
E(W^{(k+1)J}) \le E(W^{kJ})-\eta_k\|E_W(W^{kJ})\|^2+C_{15}\eta_k^2 \tag{A.37}
$$

Hence, the sequence $\{E(W^{kJ})\}$, being nonnegative, is characterized by a monotonic decrease and is bounded.

By the Lemma7.1 and (A.37), we can get

$$
\sum_{k=0}^{\infty}\eta_k\|E_W(W^{kJ})\|^2<\infty \tag{A.38}
$$

Naturally, we can get

$$
\sum_{k=0}^{\infty}\eta_k\|E_{w_0}(W^{kJ})\|^2<\infty
$$

According to (III.9), (III.10), (III.12) and (III.13), it can be derived that

$$
\begin{aligned}
&|E_{w_0}(W^{(k+1)J})-E_{w_0}(W^{kJ})| \\
&\quad =\sum_{j=1}^{J}|r_0^{kJ+1}|\le JC_7\eta_k
\end{aligned} \tag{A.41}
$$

and

$$
\begin{aligned}
&|E_{w_n}(W^{(k+1)J})-E_{w_n}(W^{kJ})| \\
&\quad =\sum_{j=1}^{J}|r_n^{kJ+1}|\le JC_8\eta_k
\end{aligned} \tag{A.42}
$$

Consequently

$$
|E_w(W^{(k+1)J})-E_w(W^{kJ})|\le C_{16}\eta_k \tag{A.43}
$$

where $C_{16}=J\max\{C_7,C_8\}$.

From (A.40), (A.43) and assumption $A(3)$, *Lemma*7.2, we can obtain

$$
\lim_{k\to\infty}E_w(W^{kJ})=0 \tag{A.44}
$$

The weak convergence result (IV.3) has be proved. □

Proof of Theorem IV.3 Review (A.11), we have

$$
\lim_{k\to\infty}\|W^{(k+1)J}-W^{kJ}\|=0 \tag{A.45}
$$

A point W^* exists such that, from (A.44), (A.45), assumption $A(3)$ and *Lemma*7.3, it is straightforward to see that

$$
\lim_{k\to\infty}W^{kJ+j}=W^* \tag{A.46}
$$

the strong convergence result (IV.3) has been established.

This proof is completed. □

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant 12471482, Grant 12571492, Grant 12031003, and Grant 12271117.

Disclosure Statement

The authors declare that they have no conflicts of interest.

Data Availability Statement

Data will be made available on reasonable request.

References

[1] W. Pei et al., "Saliency-aware regularized graph neural network," *Artif. Intell.*, vol. 328, 2024, Art. no. 104078.

[2] H. Chen, F. He, S. Lei, and D. Tao, "Spectral complexity-scaled generalization bound of complex-valued neural networks," *Artif. Intell.*, vol. 322, 2023, Art. no. 103951.

[3] Y. T. Wang and Q. Wang, "Intelligent traffic monitoring and traffic diagnosis analysis based on neural network algorithm," *Neural Comput. Appl.*, vol. 33, pp. 8107–8117, 2021.

[4] D. Maximov and V. I. Goncharenko, "Multi-valued neural networks I: A multi-valued associative memory," *Neural Comput. Appl.*, vol. 33, pp. 1018–10198, 2021.

[5] C. L. Giles and T. Maxwell, "Learning, invariance, and generalization in high-order neural networks," *Appl. Opt.*, vol. 26, no. 23, pp. 4972–4978, 1987.

[6] J. X. Li, H. Peng, and Y. W. Cao, "Higher-order attribute-enhancing heterogeneous graph neural networks," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 560–574, Jan. 2023.

[7] B. Hu, X. G. Wang, and Z. L. Feng, "HSDN: A high-order structural semantic disentangled neural network," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 8742–8756, Sep. 2023.

[8] C. K. Li, "A sigma-pi-sigma neural network," *Neural Process. Lett.*, vol. 17, pp. 1–19, 2003.

[9] E. Bas, E. Egrioglu, and E. Kolemen, "A novel intuitionistic fuzzy time series method based on bootstrapped combined pi-sigma artificial neural network," *Eng. Appl. Artif. Intell.*, vol. 114, pp. 0952–1976, 2022.

[10] V. Nedeljkovic, "A novel multilayer neural networks training algorithm that minimizes the probability of classificationerror," *IEEE Trans. Neural Netw.*, vol. 4, no. 4, pp. 650–659, Jul. 1993.

[11] M. Bianchini et al., "Learning without local minima in radial basis function networks," *IEEE Trans. Neural Netw.*, vol. 6, no. 3, pp. 749–756, May 1995.

[12] L. V. Gambuzza, M. Frasca, F. Sorrentino, L. M. Pecora, and S. Boccaletti, "Controlling symmetries and clustered dynamics of complex networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 282–293, Jan.–Mar. 2021.

[13] L. Gan and S. H. Low, "An online gradient algorithm for optimal power flow on radial networks," in *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 625–638, Mar. 2016.

[14] Q. Fan, Q. Kang, J. M. Zurada, T. Huang, and D. Xu, "Convergence analysis of online gradient method for high-order neural networks and their sparse optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 18687–18701, Dec. 2024.

[15] L. Xu, J. Chen, D. Huang, J. Lu, and L. Fang, "Analysis of boundedness and convergence of online gradient method for two-layer feedforward neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 8, pp. 1327–1338, Aug. 2013.

[16] L. Li and Z. Qiao, "A smoothing algorithm with constant learning rate for training two kinds of fuzzy neural networks and its convergence," *Neural Process. Lett*, vol. 51, pp. 1093–1109, 2020.

[17] Y. Xu and F. Deng, "Convergence of gradient method for training ridge polynomial neural network," *Neural Comput, Neural Comput. Appl.*, vol. 22, pp. 333–339, 2012.

[18] C. X. Huang and B. W. Liu, "Global convergence on asymptotically almost periodic SICNNs with nonlinear decay functions," *Neural Process. Lett*, vol. 49, pp. 625–641, 2019.

[19] K. Benyamin and L. Lina, "Infrared face recognition system using cross entropy error function based ensemble backpropagation neural networks," *Int. J. Comput. Theory Eng.*, vol. 8, pp. 161–166, 2016.

[20] H. Elsadek and E. Zahraa, "Improving error back propagation algorithm by using cross entropy error function and adaptive learning rate," *Int. J. Comput. Appl.*, vol. 161, pp. 5–9, 2017.

[21] P. Golik and P. Doetsch, "Cross-entropy vs. squared error training: A theoretical and experimental comparison," *Interspeech*, vol. 13, pp. 1756–1760, 2013.

[22] A. S. Bosman and A. Engelbrecht, "Visualising basins of attraction for the cross-entropy and the squared error neural network loss functions," *Neurocomputing*, vol. 400, pp. 113–136, 2020.

[23] Y. Xiong and X. Tong, "Convergence of batch gradient method based on the entropy error function for feedforward neural networks," *Neural Process. Lett.*, vol. 52 pp. 2687–2695,2020.

[24] K. Saito and R. Nakano, "Second-order learning algorithm with squared penalty term," *Neural Comput. Appl.*, vol. 12, no. 3, pp. 709–729, 2000.

[25] J. Portilla, A. Tristn-Vega, and I. W. Selesnick, "Efficient and robust image restoration using multiple-feature L_2-Relaxed sparse analysis priors," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5046–5059, Dec. 2015.

[26] Y. Liu and D. K. Yang, "Relaxed conditions for convergence analysis of online back-propagation algorithm with L_2 regularizer for sigma-pi-sigma neural network," *Neurocomputing*, vol. 272, pp. 163–169, 2018.

[27] L. He et al., "Seismic acoustic impedance inversion using reweighted L_1-norm sparse constraint," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 8027105.

[28] A. C. Cardall and R. C. Hales, "LASSO (L_1) regularization for development of sparse remote-sensing models with applications in optically complex waters using," *GEE Tools, Remote Sens.*, vol. 15, no. 6, 2023, Art. no. 1670.

[29] Q. W. Fan, L. Liu, and S. Zhao, "Boundedness and convergence analysis of pi sigma neural network based on online gradient method and its sparse optimization," *East Asian J. Appl. Math.*, vol. 14, no. 4, pp. 769–787, 2024.

[30] H. Zhang, J. Wang, Z. Sun, J. M. Zurada, and N. R. Pal, "Feature selection for neural networks using group lasso regularization," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 659–673, Apr. 2020.

[31] Q. Kang and Q. W. Fan, "Deterministic convergence analysis via smoothing group lasso regularization and adaptive momentum for sigma-pi-sigma neural network," *Inf. Sci.*, vol. 553, pp. 66–82, 2021.

[32] Q. W. Fan, J. M. Zurada, and W. Wu, "Convergence of online gradient method for feedforward neural networks with smoothing $L_{1/2}$ regularization penalty," *Neurocomputing*, vol. 131, pp. 208–216, 2014.

[33] F. Li and J. M. Zurada, "Smooth group $L_{1/2}$ regularization for input layer of feedforward neural networks," *Neurocomputing*, vol. 314, pp. 109–119, 2018.

[34] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Stat. Soc.*, vol. 67, no. 2, pp. 301–320, 2005.

[35] M. Li et al., "Guest editorial: Deep neural networks for graphs: Theory, models, algorithms, and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4367–4372, Apr. 2024.

[36] B. Jiang, B. Wang, S. Chen, J. Tang, and B. Luo, "Graph neural network meets sparse representation: Graph sparse neural networks via exclusive group lasso," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 10, pp. 12692–12698, Oct. 2023.

[37] F. Gama, E. Isufi, G. Leus, and A. Ribeiro, "Graphs, convolutions, and neural networks: From graph filters to graph neural networks," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 128–138, Nov. 2020.

[38] J. Li, R. Zheng, H. Feng, M. Li, and X. Zhuang, "Permutation equivariant graph framelets for heterophilous graph learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 11634–11648, Sep. 2024.

[39] G. Yang, M. Li, H. Feng, and X. Zhuang, "Deeper insights into deep graph convolutional networks: Stability and generalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 48, no. 2, pp. 1707–1719, Feb. 2026.

[40] L. Yang, J. Peng, Y. Wang, L. Xu, and W. Zhu, "Semisupervised hyperspectral image classification via superpixel-based graph regularization with local and nonlocal features," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 6645–6658, 2022.

[41] N. N. Zhang, J. X. Liu, C. H. Zheng, and J. Wang, "SLRRSC: Single-cell type recognition method based on similarity and graph regularization constraints," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 7, pp. 3556–3566, Jul. 2022.

[42] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[43] T. Ince, "Superpixel-based graph Laplacian regularization for sparse hyperspectral unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 5501305.

[44] Z. Zhao, X. He, D. Cai, L. Zhang, W. Ng, and Y. Zhuang, "Graph regularized feature selection with data reconstruction," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 689–700, Mar. 2016.

[45] Y. Fan and W. Yang, "A backpropagation learning algorithm with graph regularization for feedforward neural networks," *Inf. Sci.*, vol. 607, pp. 263–277, 2022.