International Journal of Advanced Technology and Engineering Exploration, Vol 11(116)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2024.111100125

Research Article

Convergence of batch gradient training-based smoothing L_1 regularization via adaptive momentum for feedforward neural networks

Khidir Shaib Mohamed[1, 2*] and Raed Muhammad Albadrani[3]
Department of Mathematics, College of Sciences, Qassim University, Qassim, KSA[1]
Department of Mathematics and Computer, College of Science, Dalanj University, Dalanj, Sudan[2]
Department of Computer Science, College of Computer, Qassim University, Qassim, KSA[3]

Received: 03-February-2024; Revised: 20-July-2024; Accepted: 23-July-2024

Abstract

Momentum has been extensively researched for regularization, and it is a widely used method to quicken the convergence of practical training. Unfortunately, no such effective acceleration method for L_1 regularization has yet to be presented. For the purpose of training and pruning feedforward neural networks, the convergence of batch gradient training-based smoothing L_1 regularization via adaptive momentum (BGSL$_1$AM) was developed. In doing so, the usual L_1 regularization is foremost nonsmooth; it generates oscillations in the computation and makes convergence analysis difficult. To overcome this issue, a smoothing function approximation L_1 regularizer at the origin is proposed. Numerical simulation results on a range of function approximations and pattern classification problems demonstrate the effectiveness of BGTSL$_1$AM algorithm. This process progressively reduces weights pending training and allows for their removal afterwards. Together with the significance of the suggested approach and some weak and strong convergence analyses, the convergence conditions are also offered. The suggested learning method has good convergence qualities and accuracy in function approximation, as demonstrated by the simulation results.

Keywords

Convergence, Smoothing L_1 regularization, Adaptive momentum, Feedforward neural network, Batch gradient training.

1.Introduction

Artificial neural networks are extensively employed in numerous fields, including intelligent data processing, pattern identification, feature extraction, and more [1–3]. For feedforward neural networks (FFNN), a popular learning algorithm is backpropagation (BP) [4, 5], which has since been widely used in neural network training. Gradient descent training (GDT), an optimization algorithm, minimizes the loss function [6], while BP computes the gradients of the loss function in relation to the model parameters. BP algorithm is a key component of GDT, as it provides the gradients requisited to update the model parameters. There are two main and widely used modes for implementing the GDT algorithm: the batch approach and the online method. In batch learning, the typical gradient approach refers to adjusting the weights of networks once, after the training samples have been presented in their entirety. Nonetheless, incremental learning is a variant of the conventional gradient approach, which modifies the weights once following the display of every training sample [7]. For FFNN training, GDT is a well-liked and straightforward learning technique. A few deterministic convergence findings for neural network gradient algorithms, both batch and online, have been proven in previous studies [8, 9].

Overfitting is the most common problem that disturbs networks during training. Overfitting occurs when the function class grows reasonably larger than the dataset [10] or the variables are inconsistent. There are many methods used to avoid this phenomenon, including regularization [11], pruning [12], early stopping [13], data augmentation [14], and ensembling [15]. In general, regularization is an effective strategy that can boost stability, encourage feature sharing, and lessen overfitting to dramatically enhance model performance. Other forms of regularization include weight decay [16], matrix regularization [17], and weight elimination [18].

[*] Author for correspondence

Generally, a L_τ $(0 \leq \tau \leq 2)$ regularizer can be defined as in Equation 1:

$$E(\theta) = \tilde{E}(\theta) + \lambda \|\theta\|_\tau^\tau \qquad (1)$$

A vector called w represents each weight in the network, $\lambda > 0$ is the regularization coefficient, $\|\theta\|_\tau$ is the τ-norm of vector $\theta = (\theta_1, \cdots, \theta_n)$ and

$$\|\theta\|_\tau := \left(\sum_{i=1}^{n} |\theta_i|^\tau\right)^{1/\tau}$$

The regularization terms L_0, $L_{1/2}$, L_1, and L_2 norms are correspondingly on the τ-value. The L_τ $(0 \leq \tau \leq 1)$ regularization can only generate spare solutions [19–21]. L_0 regularization has been used for high-dimensional regression with corrupted data [22] and for large-scale data [23]. L_0 regularization is an NP-hard optimization, because the L_0 norm is non-differential. In order to get around these problems, FFNNs, and extreme learning machines, respectively, employed a smoothing L_0 regularization [24, 25]. An $L_{1/2}$ regularizer was proposed in [21] which is a nonconvex penalty. The results of the studies indicate that weight elimination and weight decay are not as effective at pruning as $L_{1/2}$ penalty. Smoothing $L_{1/2}$ regularization is a popular way to acquire a decent generalization for a trained network by adding a smoothing function to the standard $L_{1/2}$ regularization method [26–28]. The absolute value of the coefficient is added as a penalty term in L_1 regularization. Although less sparse, the L_1 regularizer suggested in [29] is simpler to solve. L_1 regularization has also been proposed in [30] for federated learning. It is also utilized for electrical impedance tomography [31]. Since, L_1 regularization is not diffrerentble and the gradient method cannot directly used. This results is difficulty in theoretical investigation and may lead to poor convergence in computational application. In [32], a novel method was developed to prune FFNNs by modifying the usual L_1 regularization term using a smoothing technique. A regularizer for FFNN prune is commonly used for weight elimination in order to discourage the use of unnecessary connections. A term symmetrical to the L_2 norm of the weight vectors is one of the easiest penalties to add to the conventional cost function [33–35]. Its purpose is to deter the weights from taking excessive values. When training the FFNNs with the online gradient technique with L_2 regularization penalty, weights are automatically constrained [36, 37]. Referring to the Equation 1 the weights at time step t are altered in accordance with the steepest descent procedure, which is the most basic gradient descent algorithm as shown in Equation 2.

$$\Delta\theta_t = -\eta \nabla_\theta E(\theta_t) \qquad (2)$$

where ∇_θ is the gradient operator with respect to the weights and η is a little positive value known as the learning rate. It is well knowledge that learning programs can go quite slowly. A momentum term is commonly included in simulations of connectionist learning algorithms. Researchers discovered in [38] that the addition of a momentum component significantly raises the rate of convergence, despite the fact that it is commonly known that such a term significantly accelerates learning. Equation 2 becomes Equation 3 using this procedure.

$$\Delta\theta_t = -\eta \nabla_\theta E(\theta_t) + \mu \Delta\theta_{t-1} \qquad (3)$$

where μ is the momentum parameter. Frequently, add a momentum term to the weight increment formula to accelerate and stabilize the gradient method training iterations [39–41].

In this case, the current weight updating increment is determined by combining the previous weight updating increment with the error function's current gradient. Numerous scholars have expanded the applicability of momentum theory and developed it; see, for example, [42–44]. As a novel method, BP algorithms with adaptive momentum are provided in [45–47]. In [48], a novel method for dynamically choosing the momentum coefficient was given. Iteratively adjusting the momentum coefficient is done by taking the inner product of the last weight increment and the current direction of descent. L_1 regularization with momentum is utterly absent from machine learning literature, particularly when it comes to learning neural networks. Filling this gap is one of the key goals of this paper. Examining the batch gradient training method's deterministic convergence with both a momentum component and a smoothing L_1 regularization term is the aim of this paper. For simplicity's sake, a three-layer FFNN is considered here, with relatively simple values assigned to the parameters: the learning rate is a positive constant, the momentum coefficient is an adaptive variable, and both the regularization factor and smoothing parameter are positive constants. Enough criteria are provided to ensure that this novel algorithm converges both weakly and strongly.

Section 2 offers an overview of the relevant literature on the subject. In section 3, the learning algorithm approach, algorithm description, learning parameters, batch gradient-based smoothing L_1 regularization via adaptive momentum, network structure, proposition and theorem, and data analysis are covered. The experimental research and interpretation of the findings are presented in section 4. The analysis of the findings and their interpretation is covered in section 5. The paper finally comes to a close in section 6.

2.Literature review

In the literature, most previous studies dealt with regularization or momentum term. Very few studies combined regularization with momentum term. Zhang et al. [49] implemented an adaptive momentum and inner-product penalty based on online gradient method for FFNNs. Both two-layer and three-layer neural network models are looked at, with the assumption that the training samples are changed randomly in each cycle of iteration. Zhang et al. [50] presented a split-complex back-propagation algorithm with momentum and penalties for training a complex-valued neural network. The algorithm uses momentum to accelerate its convergence, while penalties regulate the magnitude of the network weights. Under this condition, demonstrate the theoretical proof of the network weights remaining bounded throughout the training process. In their study, Fan et al. [51] presented novel theoretical findings on the BP algorithm. They focused on FFNNs with a single hidden layer and explored the impact of smoothing regularization and adaptive momentum. Their research revealed that the gradient of the error function tends to decrease to zero, leading to a stabilization of the weight sequence. In their study, Kang et al. [52] utilized adaptive momentum and smoothing group lasso regularization techniques to showcase a fast and effective method for training a Sigma-Pi-Sigma neural network. With the addition of an adaptive momentum term, the learning convergence is accelerated during the iteration process. From a group perspective, it is evident that group sparsity highlights the advantage of maintaining a sparse network structure. Analyzing the theoretical results is a significant contribution. Researchers Fan et al. [53] recently investigated the sparse-response feed-forward algorithm for training pi-sigma neural network inference models. This approach employs gradient descent and integrates a self-adaptive momentum term and a group lasso regularizer to improve its performance.

The primary objective of this research is to improve the traditional group lasso regularization term by integrating a smoothing technique at its origin.

By employing this procedure, it is possible to get neural networks that possess both sparsity and efficiency, along with a comprehensive theoretical examination of the technique. Additionally, the iteration process incorporates an adaptive momentum term to enhance the network's learning speed. As well as the overparametrized models' implicit regularization of momentum gradient descent covered in [54, 55]. Lasso's outstanding performance led to its rapid development into a wide variety of models. In [33] exploited neural networks with smoothed L_1 regularization to achieve maximally sparse networks with minimal performance degradation. However, no one has yet proposed such a successful acceleration technique for smoothing L_1 regularization with adaptive momentum. To do this, this paper looks into how fast batch gradient training-based smoothing L_1 regularization works with adaptive momentum for FFNN. The algorithm uses momentum to accelerate its convergence, while regularization regulates the magnitude of the network weights.

3.Methodology

3.1Neural network structure (FFNN)

Three different types of layers make up the architecture of an FFNN: output, hidden, and input. A three-layer FFNN is considered. This network's architecture consists of one output layer, M input units, and N hidden units. The transfer function for the hidden and output layers is denoted as $h : R \rightarrow \mathrm{R}$. Generally, although not invariably, this is a sigmoid function. Let $w_0 = (w_{01}, w_{02}, \dots, w_{0N})^T \in R^N (n = 1,2,\dots,N)$

represent the weight vector connecting the hidden layers to the output layer. The weight vector $w_l = (w_{l1}, w_{l2}, \dots, w_{lM})^T \in R^M (m = 1,2,\dots,M)$

represents the connections between each input layer and the hidden layer. The weight factors are stated in a concise manner, denoted as $\mathcal{W} = (w_0^T, w_0^T, \dots, w_N^T) \in R^{N+MN}$, for the purpose of classifying the offer. The specific matrix U, denoted as $(w_1, w_2, \dots, w_N)^T$ and belonging to the set of real numbers $R^{M \times N}$, was taken into account. Subsequently, Equation 4 establishes the definition of a vector function.

$$\mathrm{H(x) = (h(x_1), h(x_2), \dots, h(x_M))^T} \qquad (4)$$

Here, $\boldsymbol{H}: \boldsymbol{R}^{\boldsymbol{N}} \rightarrow \boldsymbol{R}^{\boldsymbol{M}}$, for $\boldsymbol{x} = (\boldsymbol{x}_{\boldsymbol{1}}, \boldsymbol{x}_{\boldsymbol{2}}, \dots, \boldsymbol{x}_{\boldsymbol{M}})^{\boldsymbol{T}} \in \boldsymbol{R}^{\boldsymbol{M}}$. For any given input $\boldsymbol{\xi} \in \boldsymbol{R}^{\boldsymbol{N}}$. The hidden neuron's output is $\boldsymbol{H}(\boldsymbol{U}\,\boldsymbol{\xi})$, while the network's final output is shown in Equation 5.

$$\mathrm{y} = \mathrm{h}(w_0 \cdot \mathrm{H(U\,\xi)}) \qquad (5)$$

The expression $\boldsymbol{w_0} \cdot \boldsymbol{H}(\boldsymbol{U}\boldsymbol{\xi}^{\boldsymbol{l}})$ denotes the inner product of the vectors $\boldsymbol{w}^{\boldsymbol{0}}$ and $\boldsymbol{H}(\boldsymbol{U}\boldsymbol{\xi}^{\boldsymbol{l}})$.

3.2.Batch gradient training based smoothing L_1 regularizer via adaptive momentum (BGTSL_1AM)

3.2.1.Modified error function based L_1 regularizer via adaptive momentum (BGTL_1AM)

The training set consists of pairs of input-output examples, denoted as $\{\xi^l, O^l\}_{l=1}^{L} \subset R^M \times R$, where $\boldsymbol{\xi^l}$ represents the input and O^l represents the desired output. Many of the regularization terms have exponential increase in weights which makes to increase the order of the network. The regular error function $\boldsymbol{\mathcal{C}}(\mathcal{W})$ with L_1 regularization as shown in Equation 6.

$$\mathcal{C}(\mathcal{W}) = \frac{1}{2}\sum_{l=1}^{L}\left[O^l - h\left(w_0 \cdot H(U\xi^l)\right)\right]^2$$

$$+\lambda\sum_{n=1}^{N}\sum_{m=1}^{M}|w_{nm}|$$

$$= \sum_{l=1}^{L} h_l\left(w_0 \cdot H(U\xi^l)\right) + \lambda\sum_{n=1}^{N}\sum_{m=1}^{M}|w_{nm}| \qquad (6)$$

The function $\boldsymbol{h_l(t)}$ is defined as $\frac{1}{2}(\boldsymbol{O^l - h_l(t)})^2$, where $\boldsymbol{h'_l(t)}$ is also defined as $(\boldsymbol{O^l - h_l(t))h'(t)}$. The values of $\boldsymbol{l}$ range from 1 to L, and t belongs to the set of real numbers. λ is the regularization coefficient, and $|\cdot|$ represents the absolute value of the weights. Equation 7 displays the gradient of the error function mentioned earlier.

$$\mathcal{C}_w(\mathcal{W}) =$$
$$\left(\mathcal{C}_{w_0}^T(\mathcal{W}), \mathcal{C}_{w_1}^T(\mathcal{W}), \mathcal{C}_{w_2}^T(\mathcal{W}), \cdots, \mathcal{C}_{w_N}^T(\mathcal{W})\right)^T \quad (7)$$

where

$$\mathcal{C}_{w_0}^T(\mathcal{W}) = \left(\mathcal{C}_{10}(\mathcal{W}), \mathcal{C}_{20}(\mathcal{W}), \cdots, \mathcal{C}_{N0}(\mathcal{W})\right)^T$$
$$\mathcal{C}_{w_1}^T(\mathcal{W}) = \left(\mathcal{C}_{11}(\mathcal{W}), \mathcal{C}_{12}(\mathcal{W}), \cdots, \mathcal{C}_{1M}(\mathcal{W})\right)^T$$
$$\mathcal{C}_{w_2}^T(\mathcal{W}) = \left(\mathcal{C}_{21}(\mathcal{W}), \mathcal{C}_{22}(\mathcal{W}), \cdots, \mathcal{C}_{2M}(\mathcal{W})\right)^T$$
$$\cdots$$
$$\mathcal{C}_{w_M}^T(\mathcal{W}) = \left(\mathcal{C}_{N1}(\mathcal{W}), \mathcal{C}_{N2}(\mathcal{W}), \cdots, \mathcal{C}_{NM}(\mathcal{W})\right)^T$$

The gradient of $\boldsymbol{\mathcal{C}_w(\mathcal{W})}$ in Equation 7 with estimate to $\boldsymbol{w_{n0}}$ and $\boldsymbol{w_{nm}}$ $(\boldsymbol{n = 1, 2, \ldots, N, m = 1, 2, \ldots, M})$ are given, respectively, by Equation 8 and 9.

$$\mathcal{C}_{w_{n0}}(\mathcal{W}) = \sum_{l=1}^{L} h'_l\left(w_0 \cdot H(U\xi^l)\right)$$
$$\times h(w_n\xi^l) + \lambda sgn(w_{n0}) \qquad (8)$$

$$\boldsymbol{\mathcal{C}_{w_{nm}}(\mathcal{W}) = \sum_{l=1}^{L} h'_l\left(w_0 \cdot H\left(U\xi^l\right)\right) w_{n0}}$$
$$\times h'(w_l \cdot \xi^l)\xi_m^l + \lambda sgn(w_{nm}) \qquad (9)$$

Momentum is a vector quantity that has both magnitude and direction. Even though momentum has a direction, it can be used to forecast the orientation and rate of motion of objects after a collision. Below is a one-dimensional description of the fundamental characteristics of momentum. The detailed of proposed method is presented as follows. The error function $\mathcal{C}(\mathcal{W}^{\boldsymbol{k}})$ with L_1 regularization weights $\{\mathcal{W}^k\}$ is updated iteratively starting from an initial value $\mathcal{W}^{\boldsymbol{0}}$ using Equations 10 and 11, given the initial values $\mathcal{W}^0$ and $\mathcal{W}^1$.

$$\mathcal{W}^{\boldsymbol{k+1}} = \mathcal{W}^{\boldsymbol{k}} + \Delta\mathcal{W}^{\boldsymbol{k}}, \quad k = 0,1,2, \qquad (10)$$
$$\Delta\mathcal{W}^k = -\eta\boldsymbol{\mathcal{C}}_w(\mathcal{W}^k), \quad k = 0,1,2, \qquad (11)$$

and error function $\mathcal{C}(\mathcal{W}^{\boldsymbol{k}})$ with L_1 regularization and adaptive momentum weights $\{\mathcal{W}^{\mathrm{k}}\}$ updated iteratively by Equation 12.

$$\Delta\mathcal{W}^k = -\eta\mathcal{C}_w(\mathcal{W}^k) + r_w^k\Delta\mathcal{W}^{k-1}, \; k = 0,1,2, \quad (12)$$

where $\boldsymbol{\eta > 0}$ the learning is rate, and $\boldsymbol{r_w^k} = (\boldsymbol{r_{w_0}^k, r_{w_1}^k, \cdots, r_{w_N}^k})$ represented the $\boldsymbol{k - th}$ training's momentum coefficient vector. It is composed of $\boldsymbol{r_{w_n}^k}$ coefficients for every. It consists of coefficients $\boldsymbol{r_{w_n}^k}$ for each $\boldsymbol{\Delta w_{nm}^k}$ $(\boldsymbol{n = 1, 2, \cdots, N}$, $\boldsymbol{m = 1, 2, \cdots, M})$ for each $\boldsymbol{\Delta w_{n0}^k}$ $(\boldsymbol{n = 1, 2, \cdots, N})$ and $\boldsymbol{\Delta w_{nm}^k}$it is adjusted after each training epoch by Equation 13.

$$r_{w_n}^k =$$
$$\begin{cases} r \cdot \dfrac{-\eta\mathcal{C}_{w_n}(\mathcal{W}^k)\cdot\Delta w_n^k}{\|\Delta w_n^{k-1}\|^2}, & if\ \mathcal{C}_{w_n}(\mathcal{W}^k)\cdot\Delta w_n^k < 0 \\ 0, & otherwise \end{cases} \qquad (13)$$

where momentum factor $r \in (0,1)$.

3.2.2. BGTSL_1AM Algorithm

Due to the presence of the absolute value in Equation 6, the optimization problem becomes difficult to solve. Consequently, the GDT cannot be efficiently applied to reduce this type of optimization problem. A unique training technique was proposed to address the challenges of calculating the gradients of the L_1 regularizer by smoothing the non-differential penalty at the origin. Equation 14 can be used to modify the error function $E(\mathcal{W})$ with L_1 regularization smoothing.

$$E(\mathcal{W}) = \frac{1}{2}\sum_{l=1}^{L}\left[O^l - h\left(w_0 \cdot \mathrm{H}(U\xi^l)\right)\right]^2 +$$
$$\lambda\sum_{n=1}^{N}\sum_{m=1}^{M} f(w_{nm}) = \sum_{l=1}^{L} h_l\left(w_0 \cdot \mathrm{H}(U\xi^l)\right) +$$
$$\lambda\sum_{n=1}^{N}\sum_{m=1}^{M} f(w_{nm}) \qquad (14)$$

where any continuous and differentiable function is represented by $f(x)$. The incrementally polynomial function that follows is designated as in Equation 15:

$$f(x) = \begin{cases} |x| \quad if \ |x| \geq \gamma \\ -\frac{x^4}{8\gamma^3} + \frac{3x^2}{4\gamma} + \frac{3}{8}\gamma \ , if \ |x| < \gamma, \end{cases} \qquad (15)$$

where an appropriate smoothing fixed, γ is utilized. Next, Equation 13 is gradient is provided via Equation 16:

$$f(x) \rightarrow [\tfrac{3}{8}\gamma, +\infty), \ f'(x) \rightarrow [-1, 1], \ f''(x) \rightarrow [0, \tfrac{3}{2\gamma}] \qquad (16)$$

Then gradient of Equation 13 is $\boldsymbol{E_w(\mathcal{W})}$ with assessment to $\boldsymbol{w_{n0}}$ and $\boldsymbol{w_{nm}(n = 1, 2, \ldots, N, m = 1, 2, \ldots, M)}$ are given, respectively, by Equation 17 and 18.

$$E_{w_{n0}}(\mathcal{W}) = \sum_{l=1}^{L} h_l'\left(w_0 \cdot H(U\xi^l)\right) \times h(w_n\xi^l) + \lambda f'(w_{n0}) \qquad (17)$$

$$\boldsymbol{E_{w_{nm}}(\mathcal{W}) = \sum_{l=1}^{L} h_l'\left(w_0 \cdot H\left(U\xi^l\right)\right) w_{n0} \times h'\left(w_l \cdot \xi^l\right)\xi_m^l + \lambda f'(w_{nm})} \qquad (18)$$

by an premier given $\mathcal{W}^{\mathbf{0}}$ and $\mathcal{W}^{\mathbf{1}}$, error function $E(\mathcal{W}^k)$ with smoothing L_1 regularization weights $\{\mathcal{W}^{\mathrm{k}}\}$ updated frequently beginning from an premier value $\mathcal{W}^0$ by Equations 19 and 20.

$$\mathcal{W}^{\boldsymbol{k+1}} = \mathcal{W}^{\boldsymbol{k}} + \Delta\mathcal{W}^{\boldsymbol{k}}, \quad k = 0,1,2, \qquad (19)$$

$$\Delta\mathcal{W}^k = -\eta E_w(\mathcal{W}^k), \qquad k = 0,1,2, \qquad (20)$$

and error function $E(\mathcal{W}^k)$ based smoothing L_1 regularization with adaptive momentum (BGTSL_1AM) weights $\{\mathcal{W}^{\mathrm{k}}\}$ updated frequently using Equation 21.

$$\Delta\mathcal{W}^k = -\eta E_w(\mathcal{W}^k) + r_w^k \Delta\mathcal{W}^{k-1}, k = 0,1,2, (21)$$

where $\boldsymbol{\eta} > \mathbf{0}$ the learning is rate. The momentum coefficient vector of the k-th training is represented by the term denoted by $\boldsymbol{r_w^k = (r_{w_0}^k, r_{w_1}^k, \cdots, r_{w_N}^k}$. It consists of coefficients $\boldsymbol{r_{w_{\mathrm{n}}}^{\mathrm{k}}}$ for each $\boldsymbol{\Delta w_{nm}^k}$ $\boldsymbol{(n = 1, 2, \cdots, N, \ m = 1, 2, \cdots, M\ )}$ for every $\boldsymbol{\Delta w_{n0}^k}$ $\boldsymbol{(n = 1, 2, \cdots, N)}$ and $\boldsymbol{\Delta w_{\mathrm{nm}}^{\mathrm{k}}}$ it is amendment after each training period by Equation 22.

$$r_{w_n}^k = \begin{cases} \frac{-\eta E_{w_n}(\mathcal{W}^k) r \Delta w_n^k}{\left\|\Delta w_n^{k-1}\right\|^2}, & if \ E_{w_n}(\mathcal{W}^k)\Delta w_n^k < 0 \\ 0, & otherwise \end{cases} \qquad (22)$$

where momentum factor $r \in (0,1)$.

3.3.Proposition and theorem

Equations 19 and 21 present the convergence theorems of the BGTSL_1AM in this section. *Appendix II* provides the proofs. The following conditions are sufficient for convergence:

Proposition 1 Let $|h(t)|, |h'(t)|, |h(t)|, |h''(t)|$ and $|f(t)|, |f'(t)|$ and $|f''(t)|$ for $\mathrm{t} \in \mathrm{R}$, they are uniformly bounded such as

Proposition 2 There exists a uniformly bounded like$\left\|w_0^k\right\|$ $(k = 0,1, \cdots)$

Proposition 3 The learning rate (η) and penalty coefficient (λ) should satisfy the condition $0 < \eta < 1/(\lambda\mathrm{M} + \mathrm{C}_1)(1 + r)^2$, where M is equal to $3/2\gamma$ and C_1 is a constant defined in Equation 23.

Proposition 4 The set $\Theta = \{\mathcal{W} | E_w(\mathcal{W}) = 0\}$ has definitive points.

Then there are some positive constants such that

$$\begin{aligned}
C_1 &= C_7 + C_8 + C_9, \\
C_2 &= ma\,x\left\{\sqrt{N}C_3, (C_3C_4)^2\right\} \\
C_3 &= max\begin{Bmatrix} \sup_{t\in\mathbb{R}}|h(t)|, \sup_{t\in\mathbb{R}}|h'(t)|, \sup_{t\in\mathbb{R}}|h''(t)|, \\ \sup_{t\in\mathbb{R},1\leq l\leq L}\left|h_j'(t)\right|, \sup_{t\in\mathbb{R},1\leq l\leq L}\left|h_j''(t)\right| \end{Bmatrix}, \\
C_4 &= \min_{1\leq l\leq L}\|\xi^l\|, \\
C_5 &= \sup_{k\in\mathbb{N}}\left\|w_0^k\right\|, \\
C_6 &= C_2 max\{C_2^2, C_5^2\} \\
C_7 &= JC_6(1 + C_2), \\
C_8 &= \frac{1}{2}J(1 + C_2)C_3, \\
C_9 &= \frac{1}{2}LJC_3^2C_4^2C_5.
\end{aligned} \qquad (23)$$

Theorem 1 *Assume that the weight sequence $\{\mathcal{W}^k\}$ is produced for any initial value $\mathcal{W}^0$ and $\mathcal{W}^1$ by the new BGTSL₁AM method,* ***Equation*** *21, that the error function $E(\mathcal{W})$ has been established by* ***Equation*** *14, and that the propositions 1–3 are true. The outcome of weak convergence is as follows:*

$$\lim_{k\to\infty}\|E_{\boldsymbol{w}}(\mathcal{W}^k)\| = 0$$

Moreover, if premise 4 is also valid, then there is a strong convergence, meaning that there exists $\mathcal{W}^k \in \Theta$ such that

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

3.4.How does gradient descent work?

The GDT process involves two steps: calculating the training direction and determining an appropriate training rate. The algorithm evaluates performance by determining the derivative from the initial point and utilizing a tangent line to gauge the slope. As new parameters are generated, the steepness of the initial slope decreases until it reaches the point of convergence. The target of gradient descent is to minimize the cost function, or error between anticipated and actual values of the output. Two data points are necessary: a direction and a learning rate.

These elements determine the partial derivative computations of future iterations, enabling the process to approach the local or global minimum.

3.5.Algorithm description

In order to determine whether a specific neuron in the hidden cells undergoes training or is eliminated, the straightforward approach of calculating the norm of the combined unused weights from the neuron is used. The literature does not specify a threshold value for removing redundant neurons and redundant weighted connections from the first proposed neural network architecture. To determine the learning algorithm's sparsity, the number of weights with absolute values of ≤0.0099 and ≤0.01, respectively, was employed. The study's threshold value, randomly selected at 0.00099, is lower than the thresholds previously established in the literature. For each of the two methods are carried out fifty trails performed. Algorithm 1 explain how the experiment is conducted. The learning rate ($\boldsymbol{\eta}$), regularization parameter (λ), and smoothing coefficient ($\boldsymbol{\gamma}$) for each problem were chosen to be constants.

Algorithm 1 Project of Batch gradient training-based Smoothing L_1 regularization via adaptive momentum (BGTSL_1AM)

Input	Training set is given by $\{\xi^l, O^l\}_{l=1}^L \subset R^M \times R$, the error function in Equation 13, Learning rate ($\boldsymbol{\eta}$), regularization parameter (λ), smoothing coefficient (γ)
Initialization	Weight vectors w_0^0 *and* w_n^0 randomly
Training steps	For $k = 1,2,3,\cdots,K$ Do Calculate the cost error function in Equation 14. Calculate the gradients in Equation 21. Update the weights w_0^0 *and* w_n^0 by applying Equation 19. End

4. Results

This section displays the simulation results used to evaluate the effectiveness of the suggested BGTSL_1AM algorithm. In higher-order Gaussian function approximation and the 5-dimensional parity problem. Compared the performance of BGTSL_1AM with two typical regularization algorithms: batch gradient training-based smoothing $L_{1/2}$ regularization via adaptive momentum (BGTS$L_{1/2}$AM) and batch gradient training-based L_2 regularization via adaptive momentum (BGTL_2AM). Learning parameters play a crucial role in controlling the learning process. Before training a model, these parameters determine the algorithm's search for the optimal solution. *Table 1* carefully selects the learning parameters for both exams' examples. *Table 2* presents the input samples for the 5-dimensional parity problem.

4.1.Higher-order gaussian function approximation problem

Gaussian functions are widely used in image processing to create Gaussian blurring and in statistics to characterize normal distributions. Higher- order Gaussian functions, like in Equation 24, are frequently employed in the formulation of Gaussian beams. By increasing the exponent's content to a power p, one can formulate a Gaussian function with a flat top and Gaussian fall-off in a more general way:

$$f(x) = A\,exp\left(\frac{-(x-x_0)^2}{2\sigma_x}\right)^p, x \in [-4,4] \qquad (24)$$

Table 3 lists the average error function, gradient norm, and average time across the five trails. Furthermore, to assess the accuracy of the three algorithms, present a representative performance from one of the five experiments. *Figures 1* and *2* show the approximation error surfaces and norm of gradient, respectively, for each of the three algorithms, including BGTSL_1AM. The training results reveal that BGTSL_1AM outperforms BGTL_2AM and BGTS$L_{1/2}$AM, resulting in smaller errors and a more noticeable pruning time. *Figures 1* and *2* show that BGTSL_1AM has better approximation performance than BGTL_2AM and BGTS$L_{1/2}$AM. In addition, as shown in *Table 3*, the BGTSL_1AM consumes time to complete the process, which means it is the fastest.

Table 1 The numbers of parameters

Learning methods	Learning parameters	Gaussian function	5-bitparity
BGTL_2AM	η	0.005	0.006
	λ	0.002	0.003
	r	0.0006	0.005

BGTS$L_{1/2}$AM	η λ γ r	0.005 0.002 0.0006 0.003	0.006 0.003 0.0005 0.005
BGTSL_1AM	η λ γ r	0.005 0.002 0.0006 0.003	0.006 0.003 0.0005 0.005
Network structure		2-6-1	6-10-1
Weight size		$[-0.4,0.4]$	$[-0.5,0.5]$
Max iteration		1400	800

Table 2 Samples of 5-dimensional parity problem

Input	Output						Input						Output
-1	1	1	1	1	-1	0	1	1	1	1	1	-1	1
1	1	1	1	-1	-1	0	-1	1	1	1	-1	-1	1
1	-1	-1	-1	1	-1	0	-1	-1	-1	-1	1	-1	1
-1	-1	-1	-1	-1	-1	0	1	-1	-1	-1	-1	-1	1
-1	-1	1	-1	-1	-1	1	1	-1	1	-1	-1	-1	0
-1	1	-1	1	1	-1	1	1	1	-1	1	1	-1	0
1	1	-1	-1	1	-1	1	-1	1	-1	-1	1	-1	0
1	-1	-1	-1	1	-1	0	-1	-1	-1	-1	1	-1	1
-1	-1	1	1	1	-1	1	1	-1	1	1	1	-1	0
1	-1	1	1	-1	-1	1	-1	-1	1	1	-1	-1	0
-1	1	-1	-1	1	-1	0	1	1	-1	-1	1	-1	1
1	1	-1	-1	-1	-1	0	-1	1	-1	-1	-1	-1	1
-1	1	1	-1	-1	-1	0	1	1	1	-1	-1	-1	1
1	-1	-1	1	1	-1	1	-1	-1	-1	1	1	-1	0
1	-1	-1	-1	1	-1	0	-1	-1	-1	-1	1	-1	1
1	1	-1	-1	1	-1	1	-1	1	-1	-1	1	-1	0

Table 3 Results of the simulation for the Results of the simulation for the 5-bit parity classification problem

Learning methods	Average error function	Average norm of gradient	Average time (s)
BGTSL_1AM	1.2296e-06	0.0040	4.498593
BGTS$L_{1/2}$AM	9.4936e-06	0.0058	4.826789
BGTL_2AM	2.0190e-06	0.0107	4.890718

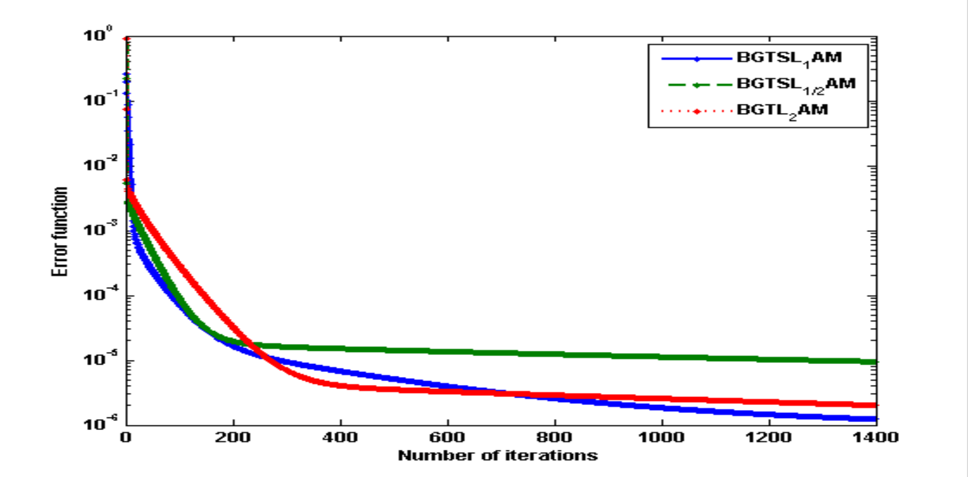

Figure 1 Comparative results of the error function for the Gaussian function problem

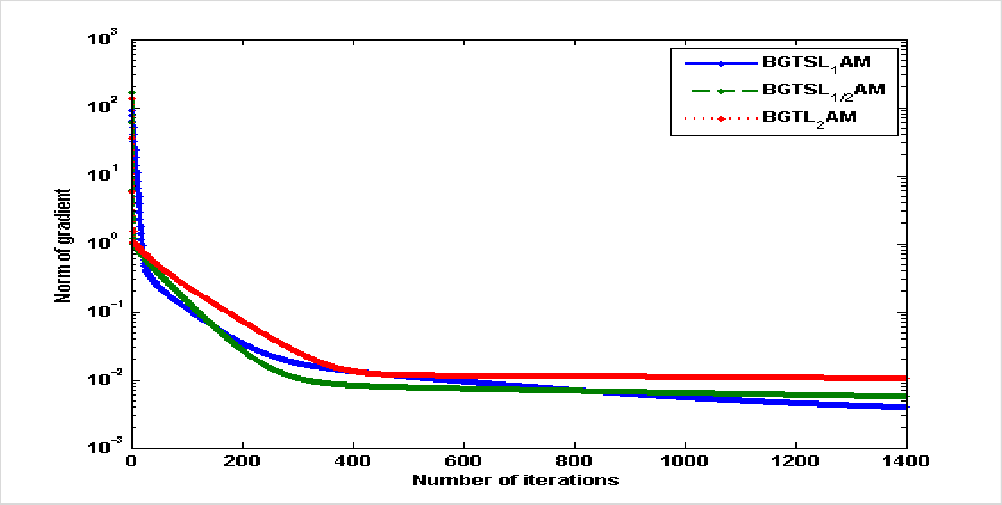

Figure 2 Comparative results of the norm of gradient for the Gaussian function problem

4.2.K-dimensional parity problem

The K-dimensional parity challenge is a widely recognized classification problem, whereas the XOR problem is a 2-dimensional parity problem. Perform numerical experiments on a 5-dimensional parity problem to illustrate the monotonicity and convergence of the suggested technique. Within the k-dimensional vector space, there are a total of 2^k input patterns. Based on the findings, it is evident that BGTSL_1AM outperforms both BGTL_2AM and BGTS$L_{1/2}$AM, as illustrated in *Figures 3* and *4*. The error function $E(\mathcal{W})$ and the norm of its gradient in BGTSL_1AM both exhibit a monotonically decreasing trend and converge to 0, aligning with the expected outcome stated in Theorem 1. *Table 4* presents the mean errors and norms of training pattern gradients obtained from five tests for every teaching algorithm. Additionally, *Table 4* displays the mean duration for the five assessments. The comparison provides strong evidence that BGTSL_1AM is superior in efficiency and possesses more resilient sparsity-promoting characteristics compared to BGTL_2AM and BGTS$L_{1/2}$AM.

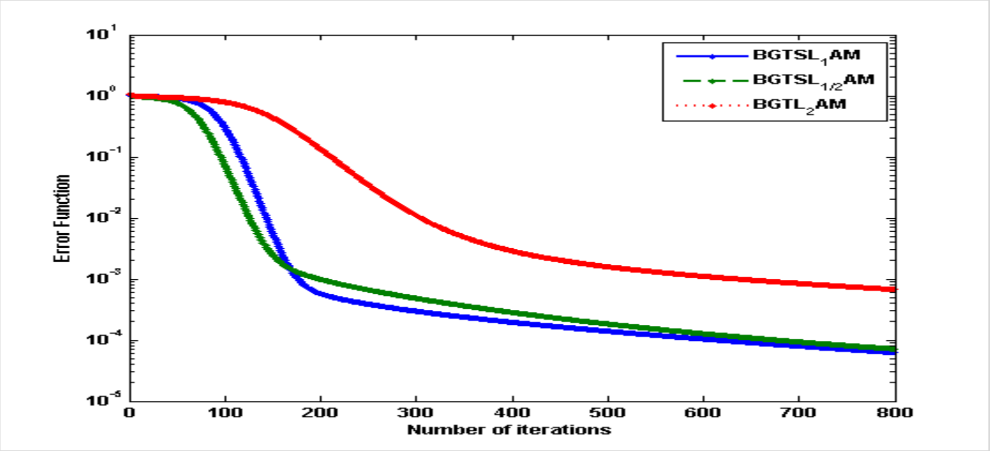

Figure 3 Comparative results of the error function for the 5-bit parity classification problem

Table 4 Results of the simulation for the 5-bit parity classification problem

Learning methods	Average error function	Average norm of gradient	Average time (s)
BGTSL_1AM	6.2432e-05	0.0199	3.507659
BGTS$L_{1/2}$AM	7.1534e-05	0.0216	3.576783
BGTL_2AM	6.7535e-05	0.0450	3.551802

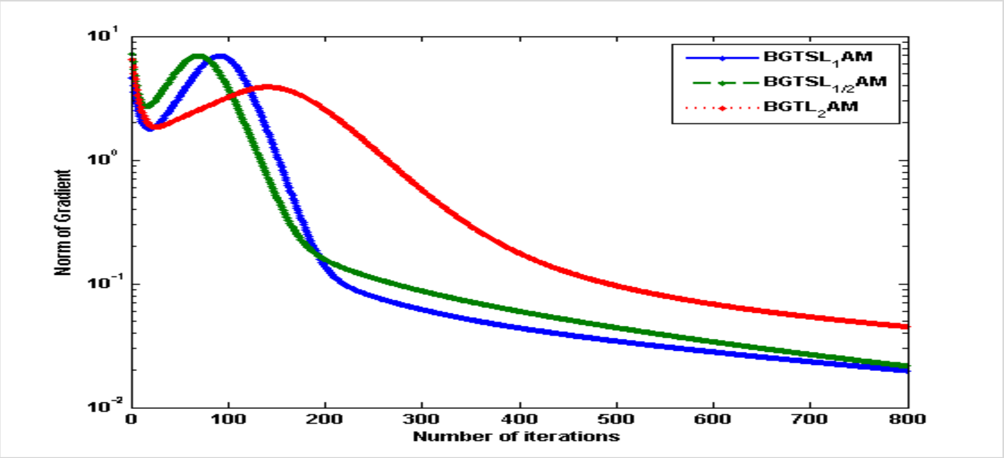

Figure 4 Comparative results of the norm of the gradient for the 5-bit parity classification problem

5. Discussion

The convincing comparison presented in [33] demonstrates that the L_1 regularized smoothing method is more efficient and has better sparsity-promoting properties compared to the $L_{1/2}$ regularized, L_2 regularized, and other regularized methods. Furthermore, it outperforms all numerical results in terms of speed.

This work builds on a previously demonstrated study in [33], utilizing an extension smoothing L_1 regularized method with adaptive momentum (BGTSL_1AM) to further control excessive weights and accelerate the process. *Tables 3* and *4* display the results of the five trials. They compare the average error and the average norms of gradients for BGTSL_1AM, BGTS$L_{1/2}$AM, and BGTL_2AM. Table 3 displays the results of the higher-order Gaussian function with the same parameters for all three algorithms. *Table 4* displays the results of the 5−dimensional parity problem with the same parameters for all three algorithms. *Tables 3* and *4* convincingly demonstrate that the BGTSL_1AM outperforms the BGTS$L_{1/2}$AM and BGTL_2AM in terms of efficiency and sparsity-promoting properties. In addition to that, *Tables 3* and *4* show that the BGTSL_1AM is faster. *Figures 1 to 4* display the convergence performance of BGTSL_1AM, BGTS$L_{1/2}$AM, and BGTL_2AM. *Figures 1* and *3* show that the square error function goes down steadily until it becomes constant, and *Figures 2* and *4* show that the gradient function tends to be zero.

The complete list of abbreviations is shown in *Appendix I*. The theoretical outcome findings are shown in *Appendix II*.

6. Conclusion and future work

The batch gradient training approach for FFNN utilizing adaptive momentum through smoothing L_1 regularizer was examined. The outcomes were displayed for both strong and weak convergence: Strong convergence is shown in this instance as $\mathcal{W}^k \rightarrow \mathcal{W}^*$ as $k \rightarrow \infty$ and the weak convergence is defined as $\|E_{\mathcal{w}}(\mathcal{W}^k)\| \rightarrow 0$ as $k \rightarrow \infty$. A further need is that the set of zero points of $E_{\mathcal{w}}(\mathcal{W})$ is finite, where $\mathcal{W}^*$ represents a local minimum point of $E(\mathcal{W})$. To validate our theoretical results, numerical examples are provided which demonstrate that BGTSL_1AM outperforms ordinary BGTS$L_{1/2}$AM, and BGTL_2AM in terms of generalization capability and convergence rate. As an alternative to batch gradient descent, which updates the dataset just once, mini-batch gradient descent (MBGD) splits the dataset into smaller batches and performs updates for each batch. That's the reason our upcoming research will focus on MBGD. Gradient boosting and artificial deep neural networks are two examples of sophisticated machine learning algorithms that hope to integrate with smoothing techniques to enhance the pruning of pi-sigma networks. It is anticipated that these advancements would lead to an increase in training and pruning neural network performance and efficiency.

Acknowledgment

The Researchers would like to thank the Deanship of Graduate Studies and Scientific Research at Qassim University for financial support (QU-APC-2024-9/1)

Conflicts of interest

The authors have no conflicts of interest to declare.

Data availability

None.

Author's contribution statement

Khidir Shaib Mohamed: Conceptualization, writing original draft and editing, analysis and interpretation of results, review and supervision and **Raed Muhammad Albadrani:** Conceptualization, writing- review and editing.

References

[1] Chen T, Lu W, Amari SI. Global convergence rate of recurrently connected neural networks. Neural Computation. 2002; 14(12):2947-57.

[2] Haykin S. Neural networks: a comprehensive foundation. Prentice Hall PTR; 1998.

[3] Magoulas GD, Vrahatis MN, Androulakis GS. Improving the convergence of the backpropagation algorithm using learning rate adaptation methods. Neural Computation. 1999; 11(7):1769-96.

[4] Plaut DC. Experiments on learning by back propagation. Reports-Research. 1986.

[5] Wilson DR, Martinez TR. The general inefficiency of batch training for gradient descent learning. Neural Networks. 2003; 16(10):1429-51.

[6] Amari SI. Backpropagation and stochastic gradient descent method. Neurocomputing. 1993; 5(4-5):185-96.

[7] Nakama T. Theoretical analysis of batch and on-line training for gradient descent learning in neural networks. Neurocomputing. 2009; 73(1-3):151-9.

[8] Wu W, Shao H, Li Z. Convergence of batch BP algorithm with penalty for FNN training. In neural information processing: 13th international conference, ICONIP, Hong Kong, China. 2006 (pp. 562-9). Springer Berlin Heidelberg.

[9] Zhang H, Wu W, Liu F, Yao M. Boundedness and convergence of online gradient method with penalty for feedforward neural networks. IEEE Transactions on Neural Networks. 2009; 20(6):1050-4.

[10] Ying X. An overview of overfitting and its solutions. In journal of physics: conference series 2019 (pp. 1-6). IOP Publishing.

[11] Santos CF, Papa JP. Avoiding overfitting: a survey on regularization methods for convolutional neural networks. ACM Computing Surveys. 2022; 54(10):1-25.

[12] He Z, Xie Z, Zhu Q, Qin Z. Sparse double descent: where network pruning aggravates overfitting. In international conference on machine learning 2022 (pp. 8635-59). PMLR.

[13] Li M, Soltanolkotabi M, Oymak S. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In international conference on artificial intelligence and statistics 2020 (pp. 4313-24). PMLR.

[14] Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. Journal of Big Data. 2019; 6(1):1-48.

[15] Sollich P, Krogh A. Learning with ensembles: how overfitting can be useful. Advances in Neural Information Processing Systems. 1995 :1-7.

[16] Zhou L, Fan Q, Huang X, Liu Y. Weak and strong convergence analysis of Elman neural networks via weight decay regularization. Optimization. 2023; 72(9):2287-309.

[17] Tian Y, Zhang Y. A comprehensive survey on regularization strategies in machine learning. Information Fusion. 2022; 80:146-66.

[18] Weigend AS, Rumelhart DE, Huberman BA. Back-propagation, weight-elimination and time series prediction. In connectionist models 1991 (pp. 105-16). Morgan Kaufmann.

[19] Moudafi A. On an extension of a spare regularization model. Mathematics. 2023; 11(20):1-12.

[20] Vidaurre D, Bielza C, Larranaga P. A survey of L1 regression. International Statistical Review. 2013; 81(3):361-87.

[21] Xu Z, Zhang H, Wang Y, Chang X, Liang Y. L 1/2 regularization. Science China Information Sciences. 2010; 53:1159-69.

[22] Zhang J, Li Y, Zhao N, Zheng Z. L 0-regularization for high-dimensional regression with corrupted data. Communications in Statistics-Theory and Methods. 2024; 53(1):215-31.

[23] Ming H, Yang H. L0 regularized logistic regression for large-scale data. Pattern Recognition. 2024; 146:110024.

[24] Zhang H, Tang Y. Online gradient method with smoothing ℓ0 regularization for feedforward neural networks. Neurocomputing. 2017; 224:1-8.

[25] Fan Q, Liu T. Smoothing l0 regularization for extreme learning machine. Mathematical Problems in Engineering. 2020; 2020(1):9175106.

[26] Wu W, Fan Q, Zurada JM, Wang J, Yang D, Liu Y. Batch gradient method with smoothing L1/2 regularization for training of feedforward neural networks. Neural Networks. 2014; 50:72-8.

[27] Li W, Chu M. A pruning feedforward small-world neural network by dynamic sparse regularization with smoothing l1/2 norm for nonlinear system modeling. Applied Soft Computing. 2023; 136:110133.

[28] Li W, Li Z, Qiao J. A fast feedforward small-world neural network for nonlinear system modeling. IEEE Transactions on Neural Networks and Learning Systems. 2024.

[29] Huang W, Li S, Fu X, Zhang C, Shi J, Zhu Z. Transient extraction based on minimax concave regularized sparse representation for gear fault diagnosis. Measurement. 2020; 151:107273.

[30] Shi Y, Zhang Y, Zhang P, Xiao Y, Niu L. Federated learning with ℓ1 regularization. Pattern Recognition Letters. 2023; 172:15-21.

[31] Tehrani JN, Mcewan A, Jin C, Van SA. L1 regularization method in electrical impedance tomography by using the L1-curve (pareto frontier curve). Applied Mathematical Modelling. 2012; 36(3):1095-105.

[32] Yashwanth M, Nayak GK, Rangwani H, Singh A, Babu RV, Chakraborty A. Minimizing layerwise activation norm improves generalization in federated learning. In proceedings of the winter conference on applications of computer vision 2024 (pp. 2287-2296). IEEE.

[33] Mohamed KS. Batch gradient learning algorithm with smoothing L 1 regularization for feedforward neural networks. Computers. 2022; 12(1):1-15.

[34] Lillo WE, Loh MH, Hui S, Zak SH. On solving constrained optimization problems with neural networks: a penalty method approach. IEEE Transactions on Neural Networks. 1993; 4(6):931-40.

[35] Setiono R. A penalty-function approach for pruning feedforward neural networks. Neural Computation. 1997; 9(1):185-204.

[36] Fan Q, Peng J, Li H, Lin S. Convergence of a gradient-based learning algorithm with penalty for ridge polynomial neural networks. IEEE Access. 2020; 9:28742-52.

[37] Hong-mei S, Wei W, Li-jun L. Convergence of online gradient method with penalty for BP neural networks. Communications in Mathematical Research. 2010; 26(1):67-75.

[38] Attoh-okine NO. Analysis of learning rate and momentum term in backpropagation neural network algorithm trained to predict pavement performance. Advances in Engineering Software. 1999; 30(4):291-302.

[39] Qian N. On the momentum term in gradient descent learning algorithms. Neural Networks. 1999; 12(1):145-51.

[40] Gitman I, Lang H, Zhang P, Xiao L. Understanding the role of momentum in stochastic gradient methods. Advances in Neural Information Processing Systems. 2019; 32.

[41] Hagiwara M, Sato A. Analysis of momentum term in back-propagation. IEICE Transactions on Information and Systems. 1995; 78(8):1080-6.

[42] Choi J. Physical approach to price momentum and its application to momentum strategy. Physica A: Statistical Mechanics and its Applications. 2014; 415:61-72.

[43] Kuhl D, Ramm E. Constraint energy momentum algorithm and its application to non-linear dynamics of shells. Computer Methods in Applied Mechanics and Engineering. 1996; 136(3-4):293-315.

[44] Chang SY. Application of the momentum equations of motion to pseudo–dynamic testing. Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences. 2001; 359(1786):1801-27.

[45] Hameed AA, Karlik B, Salman MS. Back-propagation algorithm with variable adaptive momentum. Knowledge-Based Systems. 2016; 114:79-87.

[46] Han X, Dong J. Applications of fractional gradient descent method with adaptive momentum in BP neural networks. Applied Mathematics and Computation. 2023; 448:127944.

[47] Alkhairi P, Wanayumini W, Hayadi BH. Analysis of the adaptive learning rate and momentum effects on prediction problems in increasing the training time of the backpropagation algorithm. In AIP conference proceedings 2024. AIP Publishing.

[48] Majda AJ, Stechmann SN. A simple dynamical model with features of convective momentum transport. Journal of the Atmospheric Sciences. 2009; 66(2):373-92.

[49] Zhang H, Wu W, Yao M. Boundedness and convergence of batch back-propagation algorithm with penalty for feedforward neural networks. Neurocomputing. 2012; 89:141-6.

[50] Zhang H, Xu D, Zhang Y. Boundedness and convergence of split-complex back-propagation algorithm with momentum and penalty. Neural Processing Letters. 2014; 39:297-307.

[51] Fan Q, Wu W, Zurada JM. Convergence of batch gradient learning with smoothing regularization and adaptive momentum for neural networks. Springer Plus. 2016; 5:1-7.

[52] Kang Q, Fan Q, Zurada JM. Deterministic convergence analysis via smoothing group lasso regularization and adaptive momentum for sigma-Pi-sigma neural network. Information Sciences. 2021; 553:66-82.

[53] Fan Q, Liu L, Kang Q, Zhou L. Convergence of batch gradient method for training of pi-sigma neural network with regularizer and adaptive momentum term. Neural Processing Letters. 2023; 55(4):4871-88.

[54] Wang L, Fu Z, Zhou Y, Yan Z. The implicit regularization of momentum gradient descent in overparametrized models. In proceedings of the AAAI conference on artificial intelligence 2023 (pp. 10149-56).

[55] Jelassi S, Li Y. Towards understanding how momentum improves generalization in deep learning. In international conference on machine learning 2022 (pp. 9965-10040). PMLR.

Khidir Shaib Mohamed received his M.S. degree in Applied Mathematics from Jilin University, Changchun, China, in 2011, and his Ph.D. degree in Computational Mathematics from Dalian University of Technology, Dalian, China, in 2018. His research interests include Theoretical Analysis and Regularization Methods for Neural Networks and Machine Learning.

Email: k.idris@qu.edu.sa

Raed Muhammad Albadrani received the B.Sc. and M.S. degrees in computer science from Al Qassim University, KSA in 2008 and 2023, respectively. His research interests include Artificial Intelligence, Image Processing, Machine Learning And Computer Vision.

Email: rbdrany@qu.edu.sa

Appendix I

S.No.	Abbreviation	Description
1	BP	Backpropagation
2	BGTSL$_1$AM	Batch Gradient Traning-Based Smoothing L$_1$ Regularization Via Adaptive Momentum
3	BGTSL$_{1/2}$AM	Batch Gradient Traning-Based Smoothing L$_{1/2}$ Regularization Via Adaptive Momentum
4	BGTL$_2$AM	Batch Gradient Traning-Based L$_2$ Regularization Via Adaptive Momentum
5	FFNN	Feedforward Neural Networks
6	GDT	Gradient Descent Training
7	MBGD	Mini-Batch Gradient Descent

Appendix II

To facilitate explanation, we present the subsequent notations:

$$H^{k,l} = H(U^k \xi^l)$$
$$\psi^{k,l} = H^{k+1,l} - H^{k,l}$$
$$\Delta w_n^k = w_n^{k+1} - w_n^k,\ \ for\, n = 1,2,\cdots,N$$
$$\Delta w_0^k = w_0^{k+1} - w_0^k \tag{25}$$

As a result of using error function Equation 25, we have

$$E(\mathcal{W}^{k+1}) = \textstyle\sum_{l=1}^{L} h_l\left(w_0^{k+1} \cdot \mathrm{H}(U^{k+1}\xi^l)\right) + \lambda \sum_{n=1}^{N}\sum_{m=1}^{M} f(w_{nm}^{k+1}) \tag{26}$$

$$E(\mathcal{W}^{k}) = \textstyle\sum_{l=1}^{L} h_l\left(w_0^{k} \cdot \mathrm{H}(U^{k}\xi^l)\right) + \lambda \sum_{n=1}^{N}\sum_{m=1}^{M} f(w_{nm}^{k}) \tag{27}$$

Regarding the deterministic convergence results mentioned above, let us say the following three lemmas:

Lemma I. *Assuming proposition 1 is true, and then there are*

$$(a)\ \|\mathrm{H}(x)\| \ \le C_2\,,\ \ \forall x \in R \tag{28}$$

$$(b)\ \ \left\|\psi^{k,l}\right\|^2 \le C_2 \textstyle\sum_{n=1}^{N}\left\|\Delta w_n^k\right\|^2 \le\ C_2\eta^2(1+r)^2 \sum_{n=1}^{N}\left\|E_{w_n}\left(\mathcal{W}^k\right)\right\|^2\ \ , l = 1,2,\cdots,L. \tag{29}$$

Proof. By the proposition 1, Equation 18 and definition of $\mathrm{H}(x)$. We have

$$\|\mathrm{H}(x)\| = \sqrt{h^2\left(w_1^k\xi^l\right) + h^2\left(w_2^k\xi^l\right) + \cdots + h^2\left(w_N^k\xi^l\right)}$$
$$\le \sqrt{C_3^2 + C_3^2 + \cdots + C_3^2} \le \sqrt{N}\sup_{t\in\mathbb{R}}|h(t)| \le C_3\sqrt{N} \le C_2 \tag{30}$$

This leads to obtain Inequality Equation 28. Similarly, by the proposition 1, Equation 20 and Lagrange mean value theorem of $h_j(t)$. We have

$$\|\psi^{k,l}\|^2 = \left\|\begin{pmatrix} h(w_1^{k+1}\cdot\xi^l) - h(w_1^k\cdot\xi^l) \\ \vdots \\ h(w_n^{k+1}\cdot\xi^l) - h(w_n^k\cdot\xi^l)\end{pmatrix}\right\|^2 = \left\|\begin{pmatrix} g'\left(\tilde{t}_{1,l,k}\right)(w_1^{k+1} - w_1^k)\cdot\xi^l \\ \vdots \\ g'\left(\tilde{t}_{n,l,k}\right)(w_n^{k+1} - w_n^k)\cdot\xi^l\end{pmatrix}\right\|$$
$$\le (\sup_{t\in\mathbb{R}}|h'(t)|\cdot\max_{1\le l\le L}\|\xi^l\|)^2 \textstyle\sum_{n=1}^{N}\|w_n^{k+1} - w_n^k\|^2 \le C_2\sum_{n=1}^{N}\|\Delta w_n^k\|^2 \tag{31}$$

where $C_2 = \max\{\sqrt{N}\sup_{t\in\mathbb{R}}|h'(t)|, (\sup_{t\in\mathbb{R}}|h'(t)|\max_{1\le l\le L}\|\xi^l\|)^2\}$ and $\tilde{t}_{n,l,k} \in \mathbb{R}$ is between $w_n^{k+1}\cdot\xi^l$ and $w_n^k\cdot\xi^l$. With reference to two Equation 14 and Equation 18, we get

$$\|\Delta w_n^k\| \le \eta[(1+r)]\left\|E_{w_n}(\mathcal{W}^k)\right\|,\ \ n = 1,2,\cdots,N \tag{32}$$

Inequality Equation 25 results from combining Equation 26 and Equation 27, and completes this lemma's proof.

The monotonicity of the sequence is examined in the lemma that follows. It is required for the BGTSL_1AM weak convergence proof that is given in the next theorem (Theorem I).

Lemma II. *The following estimate appears under propositions 1*−3.

$$E(\mathcal{W}^{k+1}) \le E(\mathcal{W}^k),\ \ k = 0,1,2 \tag{33}$$

Proof. According to the definition of w_0^k and $\psi^{k,l}$. We have

$$w_0^k\psi^{k,l} = \textstyle\sum_{n=1}^{N} w_{n0}^k \cdot [h(w_n^{k+1}\cdot\xi^l) - h(w_n^k\cdot\xi^l)] \tag{34}$$

Using the Lagrange remainder and the Taylor mean value theorem, $h(t)$ at the point $w_n^k\cdot\xi^l$. We have

$$h(w_n^{k+1}\cdot\xi^l) - h(w_n^k\cdot\xi^l) = h'^{(w_n^k\cdot\xi^l)}\cdot\Delta w_n^k\cdot\xi^l + \tfrac{1}{2}h''\left(t_{n,l,k}\right)(\Delta w_n^k\cdot\xi^l)^2 \tag{35}$$

where each $t_{n,l,k}$ lies on the segment between $w_n^{k+1}\cdot\xi^l$ and $w_n^k\cdot\xi^l$, $n = 1,2,\cdots,N, m = 1,2,\cdots,M.$

This together with Equation 17 and Equation 23. We have

$$\sum_{l=1}^{L} h_l'\left(w_0^k\cdot\mathrm{H}^{k,l}\right)w_0^k\psi^{k,l}$$
$$= \sum_{n=1}^{N}\sum_{l=1}^{L} h_l'\left(w_0^k\cdot\mathrm{H}^{k,l}\right)w_{n0}^k h'(w_n^k\cdot\xi^l)\cdot\Delta w_n^k\cdot\xi^l$$
$$+\frac{1}{2}\sum_{n=1}^{N}\sum_{l=1}^{L} h_l'\left(w_0^k\cdot\mathrm{H}^{k,l}\right) w_{n0}^k h''\left(t_{n,l,k}\right)(\Delta w_n^k\cdot\xi^l)^2$$

$$\begin{aligned}
&= \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{l=1}^{L} h_l'\left(w_0^k \cdot \mathrm{H}^{k,l}\right) w_{n0}^k h'(w_n^k \cdot \xi^l) \cdot \Delta w_n^k \cdot \xi_m^l \\
&+ \frac{1}{2} \sum_{n=1}^{N} \sum_{l=1}^{L} h_l'(w_0^k \cdot \mathrm{H}^{k,l})\, w_{n0}^k h''\left(t_{n,l,k}\right)(\Delta w_n^k \cdot \xi^l)^2 \\
&= \sum_{m=1}^{M} \sum_{n=1}^{N} E_{\boldsymbol{w_{nm}}}(\mathcal{W}^k) \cdot \Delta w_{nm}^k \; - \lambda \sum_{m=1}^{M} \sum_{n=1}^{N} f'(w_{nm}^k) \cdot \Delta w_{nm}^k + \mu_1 \\
&+ \frac{1}{2} \sum_{n=1}^{N} \sum_{l=1}^{L} h_l'(w_0^k \cdot \mathrm{H}^{k,l})\, w_{n0}^k h''\left(t_{n,l,k}\right)(\Delta w_n^k \cdot \xi^l)^2 \\
&= \sum_{n=1}^{N} E_{\boldsymbol{w_n}}(\mathcal{W}^k) \cdot \Delta w_n^k - \lambda \sum_{m=1}^{M} \sum_{n=1}^{N} f'(w_{nm}^k) \cdot \Delta w_{nm}^k \\
&+ \frac{1}{2} L C_3^2 C_4^2 C_5 \sum_{n=1}^{N} \|\Delta w_n^k\|^2 \\
&= \sum_{n=1}^{N} E_{\boldsymbol{w_n}}(\mathcal{W}^k)\left(-\eta E_{\boldsymbol{w_n}}(\mathcal{W}^k) + r_{w_n}^k \cdot \Delta w_n^{k-1}\right) \\
&-\lambda \sum_{m=1}^{M} \sum_{n=1}^{N} f'(w_{nm}^k) \cdot \Delta w_{nm}^k + \frac{1}{2} L C_3^2 C_4^2 C_5 \eta^2 (1+r)^2 \sum_{n=1}^{N} \left\|E_{\boldsymbol{w_n}}(\mathcal{W}^k)\right\|^2 \\
&= -\eta \sum_{n=1}^{N} \left\|E_{\boldsymbol{w_n}}(\mathcal{W}^k)\right\|^2 + \sum_{n=1}^{N} r_{w_n}^k \cdot \left(E_{\boldsymbol{w_n}}(\mathcal{W}^k)\right) \cdot \Delta w_n^{k-1} \\
&-\lambda \textstyle\sum_{m=1}^{M} \sum_{n=1}^{N} f'(w_{nm}^k) \cdot \Delta w_{nm}^k + \frac{1}{2} L C_3^2 C_4^2 C_5 \eta^2 (1+r)^2 \|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2
\end{aligned} \tag{36}$$

where $t_{n,l,k} \in R$ is between $w_n^k \cdot \xi^l$ and $w_n^{k+1} \cdot \xi^l$. Using Equations 17 and 22. We have

$$\begin{aligned}
&\sum_{l=1}^{L} h_l'\left(w_0^k \cdot \mathrm{H}^{k,l}\right)\mathrm{H}^{k,l} \Delta w_0^k \\
&= \sum_{n=1}^{N} \sum_{l=1}^{L} h_l'\left(w_0^k \cdot H^{k,l}\right) h\left(w_0^k \xi^l\right) \cdot \Delta w_{n0}^k \\
&= \sum_{n=1}^{N} \left(E_{\boldsymbol{w_{n0}}}(\mathcal{W}^k) - \lambda f'\left(w_{n0}^k\right)\right) \cdot \Delta w_{n0}^k \\
&= E_{\boldsymbol{w_{n0}}}(\mathcal{W}^k) \cdot \Delta w_{n0}^k - \lambda \sum_{n=1}^{N} f'\left(w_{n0}^k\right) \cdot \Delta w_{n0}^k \\
&= E_{\boldsymbol{w_{n0}}}(\mathcal{W}^k)\left(-\eta E_{\boldsymbol{w_n}}(\mathcal{W}^k) + r_{w_0}^k \cdot \Delta w_0^{k-1}\right) - \lambda \sum_{n=1}^{N} f'\left(w_{n0}^k\right) \cdot \Delta w_{n0}^k \\
&= -\eta \left\|E_{\boldsymbol{w_0}}(\mathcal{W}^k)\right\|^2 + r_{w_0}^k \cdot \left(E_{\boldsymbol{w_0}}(\mathcal{W}^k)\right) \cdot \Delta w_0^{k-1} - \lambda \textstyle\sum_{n=1}^{N} f'\left(w_{n0}^k\right) \cdot \Delta w_{n0}^k
\end{aligned} \tag{37}$$

Utilizing the Lagrange remainder and the Taylor mean value theorem, we arrive at

$$\begin{aligned}
&E(\mathcal{W}^{k+1}) - E(\mathcal{W}^k) \; = \sum_{l=1}^{L} h_j'(w_0^k \cdot \mathrm{H}^{k,l})[w_0^{k+1} \cdot \mathrm{H}^{k+1,l} - w_0^k \cdot \mathrm{H}^{k,l}] \\
&+\lambda \sum_{n=1}^{N} \sum_{m=1}^{M} [f(w_{nm}^{k+1}) - f(w_{nm}^k)] \\
&+ \frac{1}{2} \sum_{l=1}^{L} h_j''(s_{k,l})[w_0^{k+1} \cdot \mathrm{H}^{k+1,l} - w_0^k \cdot \mathrm{H}^{k,l}]^2 \\
&= \sum_{l=1}^{L} h_j'(w_0^k \cdot \mathrm{H}^{k,l})\, \mathrm{H}^{k,l} \Delta w_0^k + \sum_{l=1}^{L} h_j'(w_0^k \cdot \mathrm{H}^{k,l})\, w_0^k \psi^{k,l} \\
&+ \frac{1}{2} \sum_{l=1}^{L} h_j''\left(s_{k,l}\right)[w_0^{k+1} \cdot \mathrm{H}^{k+1,l} - w_0^k \cdot H^{k,l}]^2 \\
&+\lambda \textstyle\sum_{n=1}^{N} \sum_{m=1}^{M} [f(w_{nm}^{k+1}) - f(w_{nm}^k)]
\end{aligned} \tag{38}$$

where $s_{k,l} \in \mathrm{R}$ is a constant between $w_0^k \cdot \mathrm{H}^{k,l}$ and $w_0^{k+1} \cdot \mathrm{H}^{k+1,l}$.

By applying the Lagrange remainder and substationing Equations 37 and 38 we may obtain Equation 36. Then

$$\begin{aligned}
&E(\mathcal{W}^{k+1}) - E(\mathcal{W}^k) = -\eta \left\|E_{\boldsymbol{w_n}}(\mathcal{W}^k)\right\|^2 - \lambda \sum_{n=1}^{N} \sum_{m=0}^{M} f'(w_{nm}^k) \cdot \Delta w_{nm}^k \\
&+ \sum_{n=1}^{N} \left(r_{w_n}^k \cdot \left(E_{\boldsymbol{w_n}}(\mathcal{W}^k)\right) \cdot \Delta w_n^{k-1} - r_{w_0}^k \cdot \left(E_{\boldsymbol{w_0}}(\mathcal{W}^k)\right) \cdot \Delta w_0^{k-1}\right) \\
&+ \frac{1}{2} \sum_{n=1}^{N} \sum_{l=1}^{L} h_l'\left(w_0^k \cdot \mathrm{H}^{k,l}\right) w_{n0}^k h''\left(t_{n,l,k}\right)(\Delta w_n^k \cdot \xi^l)^2 \\
&+ \frac{1}{2} \sum_{l=1}^{L} h_j''(s_{k,l})\left[w_0^{k+1} \cdot \mathrm{H}^{k+1,l} - w_0^k \cdot \mathrm{H}^{k,l}\right]^2 \\
&+ \sum_{l=1}^{L} h_j'(w_0^k \cdot \mathrm{H}^{k,l})\, \Delta w_0^k \psi^{k,l} \; + \lambda \sum_{n=1}^{N} \sum_{m=1}^{M} [f(w_{nm}^{k+1}) - f(w_{nm}^k)] \\
&\leq -\eta \left\|E_{\boldsymbol{w_n}}(\mathcal{W}^k)\right\|^2 + \lambda f''\left(t_{n,l,k}\right) \sum_{n=1}^{N} \sum_{m=1}^{M} (\Delta w_{nm}^k)^2 \\
&+ \frac{1}{2} \sum_{l=1}^{L} h_j''(s_{k,l})\left[w_0^{k+1} \cdot \mathrm{H}^{k+1,l} - w_0^k \cdot \mathrm{H}^{k,l}\right]^2
\end{aligned}$$

$$+\sum_{l=1}^{L} h_j'\left(w_0^k \cdot \mathrm{H}^{k,l}\right) \Delta w_0^k \psi^{k,l} \tag{39}$$

Proposition 1, Equations 23, 26 and 29, based on the Cauchy-Schwarz inequality. As we've

$$\left|\frac{1}{2}\sum_{l=1}^{L} h_l''(s_{k,l})[w_0^{k+1} \cdot \mathrm{H}^{k+1,l} - w_0^k \cdot \mathrm{H}^{k,l}]^2\right|$$
$$\leq \frac{C_3}{2}\sum_{l=1}^{L}(\Delta w_0^k H^{k+1,l} + w_0^k \psi^{k,l})^2$$
$$\leq \frac{C_3}{2}\sum_{l=1}^{L}(C_2\|\Delta w_0^k\| + C_5\|\psi^{k,l}\|)^2$$
$$\leq C_3 max\{C_2^2, C_5^2\}\sum_{l=1}^{L}(\|\Delta w_0^k\|^2 + \|\psi^{k,l}\|^2)$$
$$\leq C_3 max\{C_2^2, C_5^2\}\sum_{l=1}^{L}\left(\|\Delta w_0^k\|^2 + \sum_{n=1}^{N}\|\Delta w_n^k\|^2\right)$$
$$\leq \frac{1}{2}LC_3 max\{C_2^2, C_5^2\}(1 + C_2)\left(\|\Delta w_0^k\|^2 + \sum_{n=1}^{N}\|\Delta w_n^k\|^2\right)$$
$$\leq \tfrac{1}{2}LC_6(1 + C_2)\eta^2(1 + r)^2\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2 \tag{40}$$

where $C_6 = C_3 max\{C_2^2, C_5^2\}$. Similarly, we can obtain

$$\sum_{l=1}^{L} h_j'(w_0^k \cdot \mathrm{H}^{k,l}) \Delta w_0^k \psi^{k,l}$$
$$\leq \frac{C_3}{2}\sum_{l=1}^{L}(\|\Delta w_0^k\| + \|\psi^{k,l}\|)^2$$
$$\leq C_7\sum_{l=1}^{L}\left(\|\Delta w_0^k\|^2 + C_2\sum_{n=1}^{N}\|\Delta w_n^k\|^2\right)$$
$$\leq \frac{1}{2}LC_3(1 + C_2)\left(\|\Delta w_0^k\|^2 + \sum_{n=1}^{N}\|\Delta w_n^k\|^2\right)$$
$$\leq \tfrac{1}{2}LC_3(1 + C_2)\eta^2(1 + r)^2\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2 \tag{41}$$

According to Equations 32, 38 - 41. We have

$$E(\mathcal{W}^{k+1}) - E(\mathcal{W}^k) \leq -\eta\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2 + \lambda C_1'\sum_{n=1}^{N}\sum_{m=0}^{M}(\Delta w_{nm}^k)^2$$
$$+\frac{1}{2}LC_3^2C_4^2C_5\eta^2(1 + r)^2\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2 + \frac{1}{2}LC_6(1 + C_2)\eta^2(1 + r)^2\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2$$
$$+\frac{1}{2}LC_3(1 + C_2)\eta^2(1 + r)^2\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2$$
$$\leq -\eta\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2 + \lambda C_1'\left(\sum_{n=0}^{N}\|\Delta w_n^k\|^2 + \|\Delta w_0^k\|^2\right)$$
$$+[C_7\eta^2(1 + r)^2 + C_8\eta^2(1 + r)^2C_9\eta^2(1 + r)^2]\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2$$
$$\leq -\eta\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2 + \lambda C_1'\eta^2(1 + r)^2\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2$$
$$+[C_7\eta^2(1 + r)^2 + C_8\eta^2(1 + r)^2C_9\eta^2(1 + r)^2]\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2$$
$$\leq -\eta[1 - C_1'\lambda\eta(1 + r)^2]\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2$$
$$+[C_7\eta^2(1 + r)^2 + C_8\eta^2(1 + r)^2C_9\eta^2(1 + r)^2]\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2 \tag{42}$$

where $C_7 = \frac{1}{2}LC_3^2C_4^2C_5, C_8 = \frac{1}{2}LC_6(1 + C_2)$, and $C_9 = \frac{1}{2}LC_3(1 + C_2)$. Using Equation 20, and from Equation 37. We have

$$E(\mathcal{W}^{k+1}) - E(\mathcal{W}^k) \leq -\eta[1 - \lambda\eta^2(1 + r)^2C_1']\sum_{n=1}^{N}\sum_{m=0}^{M}\left\|E_{\boldsymbol{w}_{nm}}(\mathcal{W}^k)\right\|^2$$
$$+[C_7\eta^2(1 + r)^2 + C_8\eta^2(1 + r)^2C_9\eta^2(1 + r)^2]\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2$$
$$\leq -\eta[1 - \lambda\eta^2(1 + r)^2C_1']\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2$$
$$+[C_7\eta^2(1 + r)^2 + C_8\eta^2(1 + r)^2C_9\eta^2(1 + r)^2]\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2$$
$$\leq -\eta[1 - C_1'\lambda\eta(1 + r)^2 - C_1\eta(1 + r)^2]\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2 \leq 0 \tag{43}$$

It is holds $E(\mathcal{W}^{k+1}) \leq E(\mathcal{W}^k)$ for $k = 0,1,2 \cdots$and this completes the proof.

Lemma III. *(See Lemma 3 in [49]) For a bounded closed region Γ, let $\mathcal{J}: \Phi \subset \mathbb{R}^n \to \mathbb{R}$ be continuous and let $\Gamma_0 = \{\mathbb{u} \in \Gamma: \mathcal{J}(\mathbb{u}) = 0\}$. There are no interior points in the projection of Γ_0 on any coordinate axis. Allow the sequence $\{\mathbb{u}^\tau\}$ to fulfill:*

$$\lim_{\tau\to\infty}\mathcal{J}(\mathbb{u}^\tau) = 0\,;\quad \lim_{\tau\to\infty}\|\mathbb{u}^{\tau+1} - \mathbb{u}^\tau\| = 0$$

Then, there present a unique $\mathbb{u}^* \in \Gamma_0$ like

$$\lim_{\tau\to\infty}\mathbb{u}^\tau = \mathbb{u}^*$$

This lemma is important in proving strong convergence. May now give the primary finding of this work, the BGTSL_1AM convergence Theorem I, thanks to the preceding three lemmas.

Proof. According to Equation 36, taking $\mathcal{A} = \eta - (\lambda C_1' + C_1)\eta^2(1 + r)^2$, it follows from proposition 3 that $\mathcal{A} > 0$.

$$\mathcal{B}^k = \sum_{n=1}^{N}\sum_{m=0}^{M}\left(E_{\boldsymbol{w}_{nm}}(\mathcal{W}^k)\right)^2 = \|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2 \tag{44}$$

We have

$$E(\mathcal{W}^{k+1}) \leq E(\mathcal{W}^k) - \mathcal{A}\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2 \leq \cdots \leq E(\mathcal{W}^0) - \mathcal{A}\sum_{m=0}^{M}\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2 \tag{45}$$

Since $E(\mathcal{W}^{k+1}) \geq 0$, it gives that

$$\mathcal{A}\sum_{m=0}^{M}\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2 \leq E(\mathcal{W}^0) \tag{46}$$

Let $k \to \infty$. We have

$$\sum_{m=0}^{M}\|E_{\boldsymbol{w}}(\mathcal{W}^k)\|^2 \leq \tfrac{1}{\mathcal{A}}E(\mathcal{W}^0) < \infty \tag{47}$$

These results in

$$\lim_{\mathrm{k}\to\infty}\left\|E_{\boldsymbol{w}_{\mathbf{n}}}(\mathcal{W}^{\mathrm{k}})\right\| = 0\,, \quad n = 0.1.2 \cdots, N \tag{48}$$

Namely

$$\lim_{k\to\infty}\|E_{\boldsymbol{w}}(\mathcal{W}^{k})\| = 0 \quad) \tag{49}$$

Thus, weak convergence is proved.

Now demonstrate the robust convergence. According to Equations 25 and 39, there is

$$\lim_{k\to\infty}\|\Delta\mathcal{W}^{k}\| = 0 \tag{50}$$

Recall Lemma III and take $\mathcal{X} = \mathcal{W}$ and $G(\mathcal{X}) = E_{\mathcal{X}}(\mathcal{X})$. This together with proposition 4, Equations 44, 50 and Lemma III strong convergence occurs immediately, i.e., there $\mathcal{W}^{k} \in \Theta$ such that

$\lim_{k\to\infty}\mathcal{W}^{k} = \mathcal{W}^{*}$.