

Received May 2, 2020, accepted May 20, 2020, date of publication May 25, 2020, date of current version June 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2997010

# Adaptive Learning-Rate Backpropagation Neural Network Algorithm Based on the Minimization of Mean-Square Deviation for Impulsive Noises

DONG WOO KIM<sup>1</sup>, (Student Member, IEEE), MIN SU KIM<sup>1</sup>, (Graduate Student Member, IEEE), JAEHO LEE<sup>2</sup>, AND POOGYEON PARK<sup>1</sup>, (Senior Member, IEEE)

<sup>1</sup>Department of Electronic and Electrical Engineering, Pohang University of Science and Technology, Pohang 37673, South Korea

<sup>2</sup>AI Laboratory, LG Electronics Inc., Seoul 06763, South Korea

Corresponding author: Poogyeon Park (ppg@postech.ac.kr)

This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, through the ICT under Grant IITP-2019-20111-1-00783, and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) under Grant 2019R1A4A1029003.

**ABSTRACT** This paper presents a novel adaptive learning-rate backpropagation neural network (ALR-BPNN) algorithm based on the minimization of mean-square deviation (MSD) to implement a fast convergence rate and robustness to impulsive noises. The learning rates of the weights in each hidden layer are derived to minimize the upper bound of the MSD obtained by the analysis, which guarantees a fast convergence rate in a stable range. Moreover, by adopting the variance of the kind of the measurement noises in each layer through the variance of the error signals, the proposed scheme provides robustness to the impulsive noises. The performance of the proposed algorithm is evaluated on various sequential signals and industrial data including the impulsive noise and compared with conventional ALR-BPNN algorithms. Simulation results indicate that the proposed algorithm outperforms the existing algorithms.

**INDEX TERMS** Adaptive learning rate, neural network, mean-square deviation, impulsive noises.

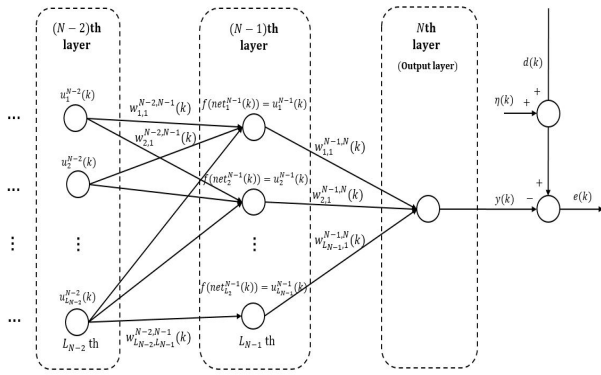
## I. INTRODUCTION

Neural networks are widely used to train various models. Training method of neural network is inundated with algorithms that focus on the application and real-time implementation of various problems [1], [2]. This paper is also concerned with the training algorithm of a multilayered feedforward neural network. A low steady-state error and fast convergence rate are important issues in the training algorithm [3], [4]. Backpropagation (BP) algorithm is a representative for training algorithm [5]–[7]. However, the main drawback of BP algorithm has a slow convergence rate. To be a fast convergence rate, some heuristic techniques such as novel BP algorithms with a momentum term and numerical optimization algorithms with quasi-Newton method were introduced [8], [9]. The limitation of the quasi-Newton methods is that the memory must be secured up to the square

of the network size. The conjugate-gradient and Newton method have been proposed to be fast convergence rate, but the algorithms require too much computational complexity [10]–[12]. Other algorithms such as recursive least-square [13], Levenberg-Marquardt [14], [15], and extended Kalman filtering [16] have been also improved to be a fast convergence, but these improvements deteriorate the simplicity and convenience of implementation of BP algorithm.

Another issue is its robustness to the impulsive noises [17], [18]. Specifically, the impulsive noises generated in real world, such as the sound of the door suddenly shutting, is a major challenge to be solved in training algorithms. To overcome the challenge, some robust adaptive algorithms have been developed, such as adaptive learning-rate saturation and sign algorithms [19], [20], but these algorithms cannot effectively train the nonlinear model. To suppress the impulsive noises in nonlinear model, various detection techniques of the impulsive noises using the neural network and other nonlinear filters have been introduced [21]–[23]. These algorithms

The associate editor coordinating the review of this manuscript and approving it for publication was Jingen Ni<sup>1</sup>.


**FIGURE 1.** Neural networks with  $N$  layers.

well detected the impulsive noises in nonlinear model, but applying these detection techniques to the training algorithms is another complex process. To mitigate this drawback, functional link neural network (FLNN) architectures have been proposed [24]–[26]. Specifically, a neural network based on sparse representations of functional links has been proposed to be robust in the impulsive noise environments [27]. This algorithm not only solved the problems of basic FLNN architectures, such as slow convergence rate and computational complexity, but also improved the robustness to the impulsive noises. However, as the algorithm contains the trigonometric functions and exponential calculations, implementing this algorithm on a DSP board or memory chip is not easy in real environment.

In order to address the aforementioned issues like the fast convergence rate and robustness to the impulsive noises, this paper proposes a novel adaptive learning-rate backpropagation neural network (ALR-BPNN) algorithm based on the minimization of mean-square deviation (MSD). Although advantages of the training algorithm based on the minimization of the MSD are well known in adaptive filtering problems [28], [29], such as fast convergence rate and low steady-state error, this algorithm has not been used to train neural networks as it is not feasible to know exact value of the MSD of the weights in each hidden layer. To take these advantages of this algorithm into neural network, the upper bound of the MSD of the weights in each hidden layer is analyzed and the learning rates of the weights is set to minimize the upper bound of the MSD to ensure a fast convergence rate in a stable range. In addition, the proposed algorithm provides robustness to the impulsive noises by adopting the variance of the kind of the measurement noises in each hidden layer through the variance of the error signals.

The rest of this paper are organized as follows. In Sections II, basic BP algorithm and notations are presented. Details of the proposed algorithm are described in Section III. Simulations results are discussed in Section IV and conclusion is provided in Section V.

## II. PRELIMINARY

Consider feedforward neural networks (FNNs) with  $L_n$  neurons in the  $n$ th layer, for  $n = 1, 2, \dots, N$ . The neural

networks represent based on the following equations:

$$net_j^{N-1}(k) = \sum_{i=1}^{L_{N-2}} w_{i,j}^{N-2,N-1}(k) u_i^{N-2}(k), \quad (1)$$

$$u_j^{N-1}(k) = f(net_j^{N-1}(k)), \quad (2)$$

where  $w_{i,j}^{N-2,N-1}(k)$  represents the weight from the  $i$ th neuron at the  $(N-2)$ th layer to the  $j$ th neuron at the  $(N-1)$ th layer.  $u_j^{N-1}(k)$  represents the output of the  $j$ th neuron that belongs to the  $(N-1)$ th layer.  $f(\cdot)$  is a rectified linear unit (ReLU) activation function and defined as

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The output of the neural network is expressed as

$$y(k) = \sum_{i=1}^{L_{N-1}} w_{i,1}^{N-1,N}(k) u_i^{N-1}(k). \quad (4)$$

To derive backpropagation method, cost function  $J(k)$  is defined as

$$J(k) = \frac{1}{2} \{d(k) - y(k)\}^2 = \frac{1}{2} e^2(k), \quad (5)$$

where  $d(k)$  and  $e(k)$  are target signals and error signals. Using gradient descent method, the update equation of the weight  $w_{i,j}^{n-1,n}(k)$  is expressed as

$$w_{i,j}^{n-1,n}(k+1) = w_{i,j}^{n-1,n}(k) - \mu \frac{\partial J(k)}{\partial w_{i,j}^{n-1,n}(k)}. \quad (6)$$

The backpropagation method is summarized in the following equations [6]

$$w_{i,j}^{n-1,n}(k+1) = w_{i,j}^{n-1,n}(k) + \mu \delta_j^n(k) u_i^{n-1}(k), \quad (7)$$

where  $\mu$  represents the learning rate. For  $n = N-1, \dots, 2$ ,  $\delta_j^n(k)$  is defined as

$$\text{If } net_j^n(k) > 0, \quad (8)$$

$$\delta_j^n(k) = \sum_{q=1}^{l_{n+1}} w_{j,q}^{n,n+1}(k) \delta_q^{n+1}(k),$$

else,

$$\delta_j^n(k) = 0, \quad (9)$$

end.

At  $N$ th layer (output layer),  $\delta_j^N(k)$  is defined as [30]

$$\delta_j^N(k) = e(k). \quad (10)$$

The update equation of the weight vector  $\mathbf{w}_j^{n-1,n}(k)$  from all neurons at the  $(n-1)$ th layer to the  $j$ th neuron at the  $n$ th layer is derived as

$$\mathbf{w}_j^{n-1,n}(k+1) = \mathbf{w}_j^{n-1,n}(k) + \mu \delta_j^n(k) \mathbf{u}^{n-1}(k), \quad (11)$$

where  $\mathbf{w}_j^{n-1,n}(k) = [w_{1,j}^{n-1,n}(k) w_{2,j}^{n-1,n}(k) \dots w_{L_{n-1},j}^{n-1,n}(k)]^T \in \mathcal{R}^{L_{n-1} \times 1}$  and  $\mathbf{u}^{n-1}(k) = [u_1^{n-1}(k) u_2^{n-1}(k) \dots u_{L_{n-1}}^{n-1}(k)]^T \in \mathcal{R}^{L_{n-1} \times 1}$ .

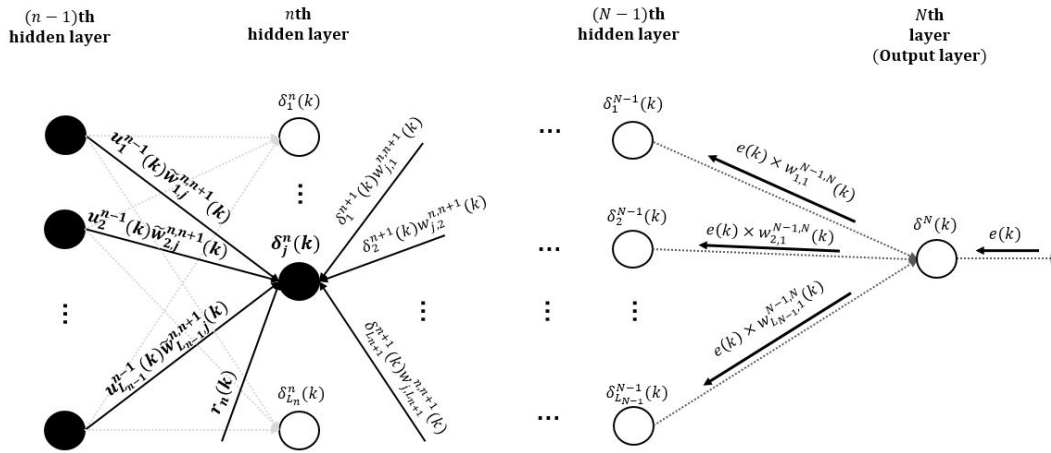


FIGURE 2. Backpropagation error of the  $j$ th neuron at the  $n$ th hidden layer.

### III. PROPOSED ALR-BPNN ALGORITHM

#### A. ADAPTIVE LEARNING-RATE EQUATION OF WEIGHT VECTOR AT HIDDEN LAYER

The update equations of the weights are modified into normalized form to analyze the MSD of each weight. This normalized update equation has advantages in defining the stability range of the learning rate over the original update equation (11).

$$\mathbf{w}_j^{n-1,n}(k+1) = \mathbf{w}_j^{n-1,n}(k) + \mu_j^n(k) \delta_j^n(k) \frac{\mathbf{u}^{n-1}(k)}{\|\mathbf{u}^{n-1}(k)\|^2}, \quad (12)$$

where  $\delta_j^n(k)$  represents the backpropagation error of the  $j$ th neuron at the  $n$ th hidden layer [6].

To derive the MSD of the weights vector from all neurons at the  $(n-1)$ th layer to the  $j$ th neuron at the  $n$ th hidden layer, in this paper,  $\delta_j^n(k)$  is defined in terms of deviation of the weights vector as follows,

$$\delta_j^n(k) = \mathbf{u}_t^{n-1,T} \mathbf{w}_{j,t}^{n-1,n} - \mathbf{u}^{n-1}(k)^T \mathbf{w}_j^{n-1,n}(k) + \eta_n(k), \quad (13)$$

$$= \mathbf{u}^{n-1}(k)^T \mathbf{w}_{j,t}^{n-1,n} - \mathbf{u}^{n-1}(k)^T \mathbf{w}_j^{n-1,n}(k) + r_n(k), \quad (14)$$

$$= \mathbf{u}^{n-1}(k)^T \tilde{\mathbf{w}}_j^{n-1,n}(k) + r_n(k), \quad (15)$$

where  $\eta_n(k)$  is a kind of a measurement noise that is independent of  $\mathbf{u}^{n-1}(k)$  and assumed to be stationary and zero-mean.  $\mathbf{u}_t^{n-1}$  and  $\mathbf{w}_{j,t}^{n-1,n}$  represent target outputs vector at the  $(n-1)$  layer and the weights vector, respectively, which are used to generate the final output signals converged to the target signals  $d(k)$ .  $\tilde{\mathbf{w}}_j^{n-1,n}(k) = \mathbf{w}_{j,t}^{n-1,n} - \mathbf{w}_j^{n-1,n}(k)$  represents the deviation of the weights vector  $\mathbf{w}_j^{n-1,n}(k)$ .  $r_n(k) = \eta_n(k) + (\mathbf{u}_t^{n-1,T} - \mathbf{u}^{n-1}(k)^T) \mathbf{w}_{j,t}^{n-1,n}$  called perturbations means a kind of the measurement noise at the hidden layer. Fig. 2 graphically shows that the backpropagation error  $\delta_j^n(k)$  is defined in terms of  $\tilde{\mathbf{w}}_j^{n-1,n}(k)$ .

Using the Eq. (12), the deviation of  $\mathbf{w}_j^{n-1,n}(k)$  can be rewritten as

$$\begin{aligned} \tilde{\mathbf{w}}_j^{n-1,n}(k+1) &= \tilde{\mathbf{w}}_j^{n-1,n}(k) - \mu_j^n(k) \delta_j^n(k) \frac{\mathbf{u}^{n-1}(k)}{\|\mathbf{u}^{n-1}(k)\|^2}, \quad (16) \\ &= \tilde{\mathbf{w}}_j^{n-1,n}(k) - \mu_j^n(k) \end{aligned}$$

$$\times \{\mathbf{u}^{n-1}(k)^T \tilde{\mathbf{w}}_j^{n-1,n}(k) + r_n(k)\} \frac{\mathbf{u}^{n-1}(k)}{\|\mathbf{u}^{n-1}(k)\|^2}, \quad (17)$$

$$\begin{aligned} &= \left[ I - \mu_j^n(k) \frac{\mathbf{u}^{n-1}(k) \mathbf{u}^{n-1}(k)^T}{\|\mathbf{u}^{n-1}(k)\|^2} \right] \tilde{\mathbf{w}}_j^{n-1,n}(k) \\ &\quad - \mu_j^n(k) r_n(k) \frac{\mathbf{u}^{n-1}(k)}{\|\mathbf{u}^{n-1}(k)\|^2}, \quad (18) \end{aligned}$$

$$\begin{aligned} &= \mathbf{F}_j^{n-1,n}(k) \tilde{\mathbf{w}}_j^{n-1,n}(k) \\ &\quad - \mu_j^n(k) r_n(k) \frac{\mathbf{u}^{n-1}(k)}{\|\mathbf{u}^{n-1}(k)\|^2}, \quad (19) \end{aligned}$$

where  $\mathbf{F}_j^{n-1,n}(k) = \left[ I - \mu_j^n(k) \frac{\mathbf{u}^{n-1}(k) \mathbf{u}^{n-1}(k)^T}{\|\mathbf{u}^{n-1}(k)\|^2} \right]$  represents a transition matrix of  $\mathbf{w}_j^{n-1,n}(k)$ . The MSD of  $\mathbf{w}_j^{n-1,n}(k)$  is defined as

$$\begin{aligned} \text{MSD}(k)_{\mathbf{w}_j^{n-1,n}} &= E[\tilde{\mathbf{w}}_j^{n-1,n}(k)^T \tilde{\mathbf{w}}_j^{n-1,n}(k) | \mathbf{U}^{n-1}(k)], \\ &= \text{Tr}\{\mathbf{P}_j^{n-1,n}(k)\}, \quad (20) \end{aligned}$$

where  $E(\cdot)$  and  $\text{Tr}(\cdot)$  represent expectation and trace, respectively.  $\mathbf{P}_j^{n-1,n}(k) = E[\tilde{\mathbf{w}}_j^{n-1,n}(k) \tilde{\mathbf{w}}_j^{n-1,n}(k)^T | \mathbf{U}^{n-1}(k)]$ , and  $\mathbf{U}^{n-1}(k) = \{\mathbf{u}^{n-1}(i) | 0 \leq i < k\}$ . Using the Eq. (19) and the assumption that  $\tilde{\mathbf{w}}_j^{n-1,n}(k)$  and  $r_n(k)$  are uncorrelated, the recursive equation of  $\mathbf{P}_j^{n-1,n}(k)$  is derived as

$$\begin{aligned} \mathbf{P}_j^{n-1,n}(k+1) &= \mathbf{F}_j^{n-1,n}(k) \mathbf{P}_j^{n-1,n}(k) \mathbf{F}_j^{n-1,n}(k)^T \\ &\quad + \{\mu_j^n(k)\}^2 \sigma_{r_n}^2(k) \frac{\mathbf{u}^{n-1}(k) \mathbf{u}^{n-1}(k)^T}{\|\mathbf{u}^{n-1}(k)\|^4}, \quad (21) \end{aligned}$$

$$\mathbf{P}_j^{n-1,n}(0) = E[\tilde{\mathbf{w}}_j^{n-1,n}(0) \tilde{\mathbf{w}}_j^{n-1,n}(0)^T | \mathbf{U}^{n-1}(0)], \quad (22)$$

**Algorithm 1** Adaptive Learning-Rate Backpropagation Neural Network

Parameters:  $N$  = number of layer,  $L_n$  = number of neuron in  $n$ th hidden layer,  $\lambda = 0.99$ ,  $\alpha_n, \beta_n, \epsilon$  : predefined

Initialization:  $\mathbf{w}_j^{n-1,n}(1) = \epsilon$ ,  $Tr\{\mathbf{P}_j^{n-1,n}(1)\} = 10$ ,  $\sigma_e^2(0) = 0$ ,  $n = N, N - 1, \dots, 2$ ,  $j = 1, 2, \dots, L_n$

- 1) For  $k = 1$  : end
- 2)  $e(k) = d(k) - y(k)$
- 3)  $\sigma_e^2(k) = (1 - \lambda)\sigma_e^2(k - 1) + \lambda e^2(k)$
- 4) For  $n = N$  : 2
- 5)  $\sigma_{r_n}^2(k) = \beta_n \sigma_e^2(k)$
- 6) For  $j = 1 : L_n$
- 7)  $\mu_j^n(k) = \frac{Tr\{\mathbf{P}_j^{n-1,n}(k)\}/(\alpha_{n-1}L_{n-1})}{Tr\{\mathbf{P}_j^{n-1,n}(k)\}/(\alpha_{n-1}L_{n-1}) + \sigma_{r_n}^2(k)/(\|\mathbf{u}^{n-1}(k)\|^2 + \epsilon)}$
- 8)  $Tr\{\mathbf{P}_j^{n-1,n}(k + 1)\} = \left[1 - \frac{2\mu_j^n(k) - \{\mu_j^n(k)\}^2}{\alpha_{n-1}L_{n-1}}\right] Tr\{\mathbf{P}_j^{n-1,n}(k)\} + \frac{\{\mu_j^n(k)\}^2 \sigma_{r_n}^2(k)}{\|\mathbf{u}^{n-1}(k)\|^2 + \epsilon}$
- 9)  $\mathbf{w}_j^{n-1,n}(k + 1) = \mathbf{w}_j^{n-1,n}(k) + \mu_j^n(k) \delta_j^n(k) \frac{\mathbf{u}^{n-1}(k)}{\|\mathbf{u}^{n-1}(k)\|^2}$
- 10) end
- 11) end
- 12) end

where  $\sigma_{r_n}^2$  represents the variance of the perturbations. The recursive equation of the MSD of  $\mathbf{w}_j^{n-1,n}(k)$  is defined as

$$Tr\{\mathbf{P}_j^{n-1,n}(k + 1)\} = Tr\{\mathbf{F}_j^{n-1,n}(k)^T \mathbf{F}_j^{n-1,n}(k) \mathbf{P}_j^{n-1,n}(k)\} + \frac{\{\mu_j^n(k)\}^2 \sigma_{r_n}^2(k)}{\|\mathbf{u}^{n-1}(k)\|^2}. \quad (23)$$

Using Lemma 1. in [31],  $Tr\{\mathbf{F}_j^{n-1,n}(k)^T \mathbf{F}_j^{n-1,n}(k) \mathbf{P}_j^{n-1,n}(k)\}$  can be bounded as

$$\begin{aligned} & Tr\{\mathbf{F}_j^{n-1,n}(k)^T \mathbf{F}_j^{n-1,n}(k) \mathbf{P}_j^{n-1,n}(k)\} \\ &= Tr \left[ \mathbf{P}_j^{n-1,n}(k) + [-2\mu_j^n(k) + \{\mu_j^n(k)\}^2] \right. \\ & \quad \times \left. \frac{\mathbf{u}^{n-1}(k) \mathbf{u}^{n-1}(k)^T}{\|\mathbf{u}^{n-1}(k)\|^2} \mathbf{P}_j^{n-1,n}(k) \right], \\ &\leq Tr\{\mathbf{P}_j^{n-1,n}(k)\} + [-2\mu_j^n(k) + \{\mu_j^n(k)\}^2] \lambda_{\min}\{\mathbf{P}_j^{n-1,n}(k)\}. \end{aligned} \quad (24)$$

Using  $\lambda_{\min}\{\mathbf{P}_j^{n-1,n}(k)\} \approx \frac{Tr\{\mathbf{P}_j^{n-1,n}(k)\}}{\alpha_{n-1}L_{n-1}}$ , the Eq. (23) is bounded as

$$\begin{aligned} & Tr\{\mathbf{P}_j^{n-1,n}(k + 1)\} \\ &\leq Tr\{\mathbf{P}_j^{n-1,n}(k)\} + [-2\mu_j^n(k) + \{\mu_j^n(k)\}^2] \\ & \quad \times \frac{Tr\{\mathbf{P}_j^{n-1,n}(k)\}}{\alpha_{n-1}L_{n-1}} + \frac{\{\mu_j^n(k)\}^2 \sigma_{r_n}^2(k)}{\|\mathbf{u}^{n-1}(k)\|^2}, \end{aligned} \quad (25)$$

$$= Tr\{\mathbf{P}_j^{n-1,n}(k)\} + \Delta(\mu_j^n(k)), \quad (26)$$

where  $\alpha_{n-1} \geq 1$  and  $\Delta(\mu_j^n(k)) \triangleq [-2\mu_j^n(k) + \{\mu_j^n(k)\}^2] \frac{Tr\{\mathbf{P}_j^{n-1,n}(k)\}}{\alpha_{n-1}L_{n-1}} + \frac{\{\mu_j^n(k)\}^2 \sigma_{r_n}^2(k)}{\|\mathbf{u}^{n-1}(k)\|^2}$ . By setting the partial differential of  $\Delta(\mu_j^n(k))$ , the proposed learning-rate of the weights vector from all neurons at the  $(n - 1)$ th layer to the  $j$ th neuron at the  $n$ th layer,  $\mu_j^n(k)$  is obtained as

follows

$$\begin{aligned} & \mu_j^n(k) \\ &= \frac{Tr\{\mathbf{P}_j^{n-1,n}(k)\}/(\alpha_{n-1}L_{n-1})}{Tr\{\mathbf{P}_j^{n-1,n}(k)\}/(\alpha_{n-1}L_{n-1}) + \sigma_{r_n}^2(k)/(\|\mathbf{u}^{n-1}(k)\|^2 + \epsilon)}, \end{aligned} \quad (27)$$

$$\begin{aligned} & Tr\{\mathbf{P}_j^{n-1,n}(k + 1)\} \\ &= \left[1 - \frac{2\mu_j^n(k) - \{\mu_j^n(k)\}^2}{\alpha_{n-1}L_{n-1}}\right] Tr\{\mathbf{P}_j^{n-1,n}(k)\} \\ & \quad + \frac{\{\mu_j^n(k)\}^2 \sigma_{r_n}^2(k)}{\|\mathbf{u}^{n-1}(k)\|^2 + \epsilon}, \end{aligned} \quad (28)$$

where  $\epsilon$  is set to small value to prevent the denominator from becoming zero. A specific algorithm is shown in Algorithm (1).

**B. STABILITY ANALYSIS**

To guarantee the stability of the proposed ALR-BPNN algorithm, *a posteriori* backpropagation error  $\delta_{j,post}^n$  is defined as

$$\delta_{j,post}^n = \mathbf{u}_t^{n-1,T} \mathbf{w}_{j,t}^{n-1,n} - \mathbf{u}^{n-1}(k)^T \mathbf{w}_j^{n-1,n}(k + 1) + r_n(k). \quad (29)$$

From the Eqs. (12) and (14),  $\delta_{j,post}^n$  can be rewritten as

$$\delta_{j,post}^n = \left(1 - \mu_j^n(k) \frac{\|\mathbf{u}^{n-1}(k)\|^2}{\|\mathbf{u}^{n-1}(k)\|^2}\right) \delta_j^n(k). \quad (30)$$

As  $\delta_{j,post}^n$  represents the backpropagation error obtained through the new updated weights and it should be less than  $\delta_j^n(k)$ ,  $\mu_j^n(k)$  should be satisfied as following,

$$|1 - \mu_j^n(k)| < 1. \quad (31)$$

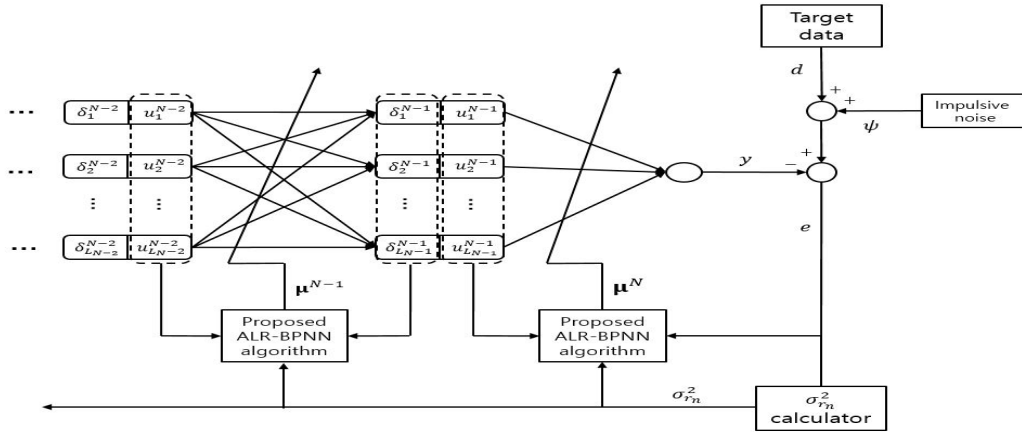


FIGURE 3. Feedforward neural networks using the proposed ALR-BPNN algorithm.

Using the Eq. (27) and (31), the stability condition of the proposed ALR-BPNN can be derived as

$$\left| \frac{\sigma_{r_n}^2(k) / (\|\mathbf{u}^{n-1}(k)\|^2 + \epsilon)}{\text{Tr}\{\mathbf{P}_j^{n-1,n}(k)\} / (\alpha_{n-1} L_{n-1}) + \sigma_{r_n}^2(k) / (\|\mathbf{u}^{n-1}(k)\|^2 + \epsilon)} \right| < 1. \quad (32)$$

From above equation, it can be seen that the proposed ALR-BPNN algorithm always satisfies the stability condition if  $\text{Tr}\{\mathbf{P}_j^{n-1,n}(k)\} > 0$ , which can be easily confirmed by the Eq. (28).

### C. PRACTICAL CONSIDERATIONS

#### 1) ESTIMATION OF $\sigma_{R_N}^2(k)$

As the variance of the perturbation  $\sigma_{r_n}^2(k)$  is not measurable value in the Eqs. (27) and (28), it is adopted by the variance of the error signals which are also obtained using a moving average method as follows

$$\sigma_e^2(k) = (1 - \lambda)\sigma_e^2(k - 1) + \lambda e^2(k), \quad (33)$$

$$\sigma_{r_n}^2(k) = \beta_n \sigma_e^2(k), \quad (34)$$

where  $\lambda$  and  $\beta_n$  were set to 0.99 and [0.01 0.5]. By choosing the variance of the perturbation in each hidden layer through the variance of the error signals, the proposed algorithm can be robustly updated even when the impulsive noises are suddenly generated. Specifically, the variance of the perturbation is rapidly increased by the error signals with the impulsive noises, which makes the learning rate small and prevents erroneous updates of the weights.

#### 2) SET OF $\alpha_N$

To choose an appropriate  $\alpha_n$  in the proposed algorithm, the normalized mean-square error (NMSE) curves according to  $\alpha_1$  and  $\alpha_2$  were compared for two types of input. As can be seen Figs. 4 and 5, when using multi-tonal sinusoidal signals as input signals,  $\alpha_1$  closer to the input layer had a greater effect on performance than  $\alpha_2$ . In that, it should be always adjusted larger than 1. However, this tendency depends on

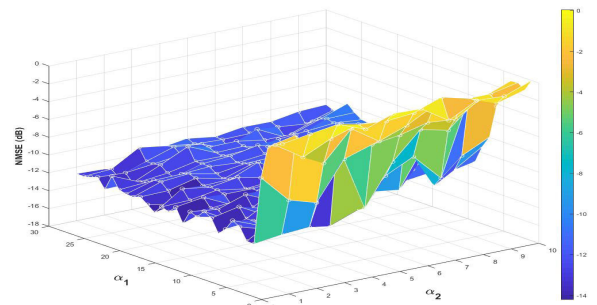


FIGURE 4. NMSE of the proposed algorithm according to  $\alpha_n$  (uncorrelated multi-tonal signals).

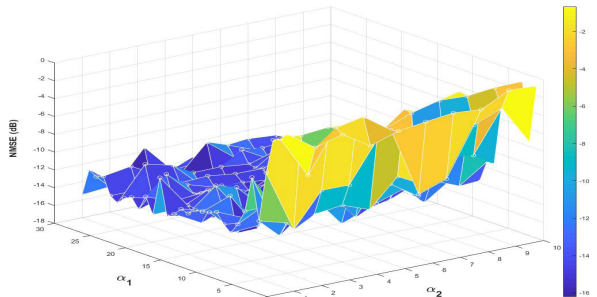


FIGURE 5. NMSE of the proposed algorithm according to  $\alpha_n$  (correlated multi-tonal signals by  $G(z)$ ).

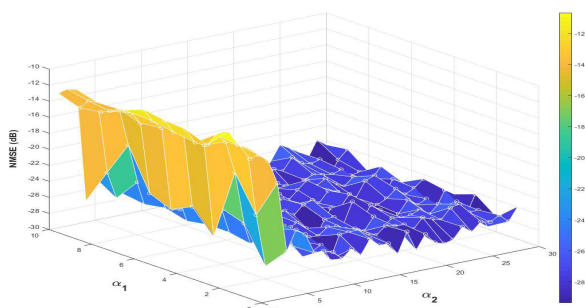
the characteristics of the inputs. For example, when using a highly correlated data features such as exchange rates by country, SHANGHAI index, and etc. as input data,  $\alpha_2$  closer to the output layer had a greater effect on performance than  $\alpha_1$  as can be seen Figs. 6 and 7. In this paper, simulations were performed by setting  $\alpha_1$  and  $\alpha_2$  based on this analysis.

### IV. SIMULATION

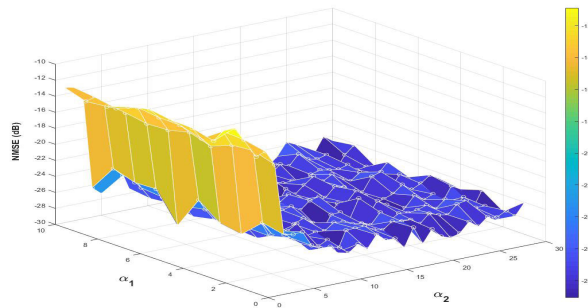
In order to evaluate the performance of the proposed algorithm, two simulation scenarios were performed in this paper. In the first case, multi-tonal sinusoidal signals dependent on various frequencies were sequentially used as input signals.

**TABLE 1.** Number of multiplication (No. of. mul.) operations per one update (◦ represents an element-wise).

Algorithm	Update rule	No. of. mul.
Adagrad	$\Delta \mathbf{w}_j^{n-1,n}(k) = -\frac{\mu_0}{\sqrt{\mathbf{G}_j^{n-1,n}(k) + \epsilon}} \circ \mathbf{g}_j^{n-1,n}(k),$ $\mathbf{G}_j^{n-1,n}(k) = \sum_{\tau=1}^k \{\mathbf{g}_j^{n-1,n}(\tau) \circ \mathbf{g}_j^{n-1,n}(\tau)\}, \mathbf{g}_j^{n-1,n}(k) = -\delta_j^n(k) \mathbf{u}^{n-1}(k)$	$4L_{n-1} + 1$
Adam	$\Delta \mathbf{w}_j^{n-1,n}(k) = -\frac{\mu_0}{\sqrt{\hat{\mathbf{v}}_j^{n-1,n}(k) + \epsilon}} \circ \hat{\mathbf{m}}_j^{n-1,n}(k),$ $\hat{\mathbf{m}}_j^{n-1,n}(k) = \frac{\mathbf{m}_j^{n-1,n}(k)}{1 - \beta_1^k}, \hat{\mathbf{v}}_j^{n-1,n}(k) = \frac{\mathbf{v}_j^{n-1,n}(k)}{1 - \beta_2^k}, \mathbf{g}_j^{n-1,n}(k) = -\delta_j^n(k) \mathbf{u}^{n-1}(k)$ $\mathbf{m}_j^{n-1,n}(k) = \beta_1 \mathbf{m}_j^{n-1,n}(k-1) + (1 - \beta_1) \mathbf{g}_j^{n-1,n}(k), \mathbf{v}_j^{n-1,n}(k) = \beta_2 \mathbf{v}_j^{n-1,n}(k-1) + (1 - \beta_2) \{\mathbf{g}_j^{n-1,n}(k) \circ \mathbf{g}_j^{n-1,n}(k)\}$	$10L_{n-1} + 1$
Proposed ALR	$\Delta \mathbf{w}_j^{n-1,n}(k) = -\mu_j^n(k) \frac{\mathbf{g}_j^{n-1,n}(k)}{\ \mathbf{u}^{n-1}(k)\ ^2},$ $\mu_j^n(k) = \frac{\text{Tr}\{\mathbf{P}_j^{n-1,n}(k)\}/(\alpha_{n-1}L_{n-1})}{\text{Tr}\{\mathbf{P}_j^{n-1,n}(k)\}/(\alpha_{n-1}L_{n-1}) + \sigma_{r_n}^2(k)/(\ \mathbf{u}^{n-1}(k)\ ^2 + \epsilon)}, \mathbf{g}_j^{n-1,n}(k) = -\delta_j^n(k) \mathbf{u}^{n-1}(k)$ $\text{Tr}\{\mathbf{P}_j^{n-1,n}(k+1)\} = \left[ 1 - \frac{2\mu_j^n(k) - \mu_j^n(k)^2}{\alpha_{n-1}L_{n-1}} \right] \text{Tr}\{\mathbf{P}_j^{n-1,n}(k)\} + \frac{\{\mu_j^n(k)\}^2 \sigma_{r_n}^2(k)}{\ \mathbf{u}^{n-1}(k)\ ^2 + \epsilon}$	$8L_{n-1} + 10$



**FIGURE 6.** NMSE of the proposed algorithm according to  $\alpha_n$  (Highly correlated input data).



**FIGURE 7.** NMSE of the proposed algorithm according to  $\alpha_n$  (Highly correlated input data with 10% bad data).

Target signals were set by passing the input signals through a specific nonlinear model. This simulation was performed to confirm how quickly and robustly the output signals generated by the proposed algorithm can converge the target signals in impulsive noise environments.

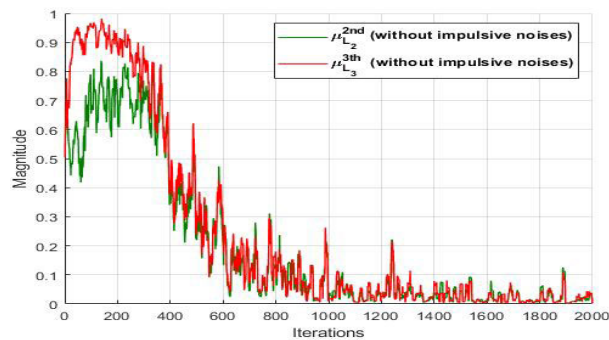
The second simulation was conducted to predict the NASDAQ index. Target and input data were set to NASDAQ index data and 18 data features for about 10 years. This simulation also evaluated how robustly the proposed algorithm trains the neural network in the environment where bad training target data is randomly generated.

**A. CASE 1**

Sinusoidal wave was used as the original signals  $s(k)$  and defined as

$$s(k) = \sqrt{2} \sin\left(\frac{2\pi f k}{F_s}\right), \tag{35}$$

where  $f$  and  $F_s$  are center frequency and sampling frequency of the original signal, respectively. A multi-tonal signal was generated through the sum of the sinusoidal waves  $s(k)$  having the center frequencies of 200, 400, ..., 1200Hz and sampling frequency of 2000Hz. Correlated input signals  $x(k)$



**FIGURE 8.**  $\mu_j^n$  curve of the proposed algorithm without the impulsive noises in case 1.

were obtained from passing the multi-tonal signals through the succeeding filters as

$$G(z) = \frac{1}{1 - 0.9z^{-1}}. \tag{36}$$

Nonlinear target signals to be estimated were set as

$$d(k) = x(k - 1) + 0.8x(k - 1)^2 - 0.6x(k - 1)^3 + 0.4x(k - 2)^2 - 0.1x(k - 3)^2. \tag{37}$$

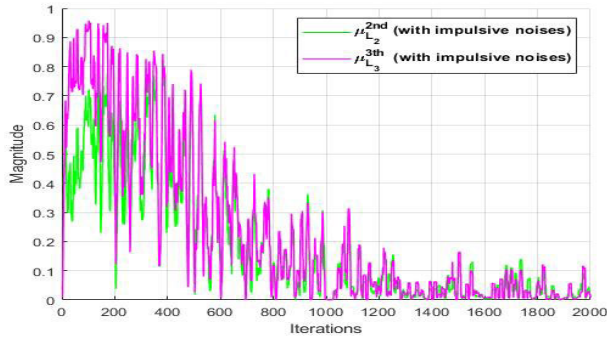


FIGURE 9.  $\mu_j^n$  curve of the proposed algorithm with the impulsive noises in case 1.

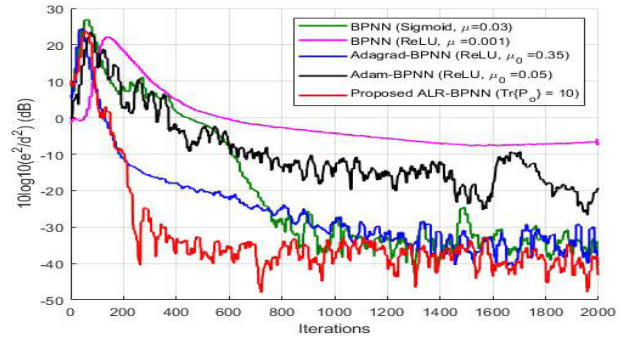


FIGURE 11. Normalized error curve without the impulsive noise (correlated multi-tonal signals by  $G(z)$ ,  $\alpha_1 = 25$ ,  $\alpha_2 = 2$ ).

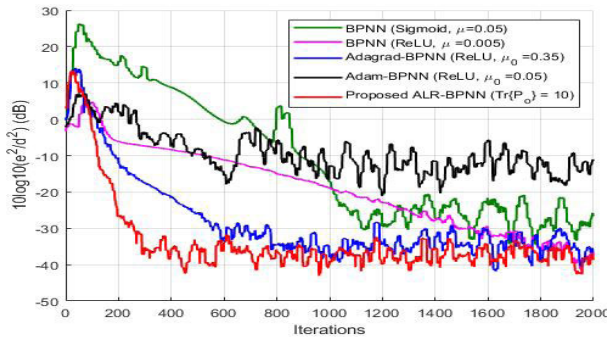


FIGURE 10. Normalized error curve without the impulsive noise (uncorrelated multi-tonal signals,  $\alpha_1 = 10$ ,  $\alpha_2 = 1$ ).

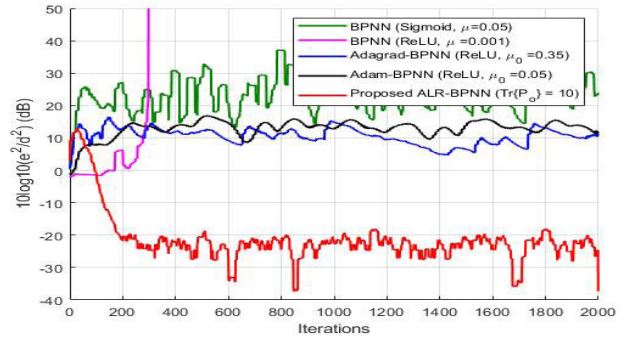


FIGURE 12. Normalized error curve with the impulsive noises (uncorrelated multi-tonal signals,  $\alpha_1 = 10$ ,  $\alpha_2 = 1$ ).

The impulsive noises  $\psi(k)$  were generated as  $\psi(k) = \omega(k)G(k)$ , where  $\omega(k)$  is Bernoulli process with  $\Pr(\omega(k) = 1) = p$  and  $p$  is set to 0.005 in this paper.  $G(k)$  is zero-mean Gaussian with power  $\sigma_G^2 = 1000\sigma_y^2$ . Basic BPNN algorithms based on the sigmoid and ReLU activation functions, Adagrad-BPNN [32], Adam-BPNN [33], and proposed ALR-BPNN algorithms are simulated to compare the performance. The parameters,  $\beta_1$  and  $\beta_2$  used in the Adam-BPNN algorithm were set to 0.99. Number of layers, including to input and output layer, was set to 3 and number of neurons at the input and 2th layers in all algorithms were set to 20, respectively. All simulation results were presented by averaging 50 independent simulations.

In this simulation, the proposed ALR-BPNN algorithm performed very well for not only the uncorrelated inputs but also the correlated inputs. As the proposed algorithm has larger learning rate by the MSD analysis compared to other algorithms in the beginning, the initial normalized error was larger than other algorithms. However, it is intended to ensure that the proposed algorithm has a fast convergence rate in a stable range, so it does not negatively affect the overall performance. As can be seen in Figs 10 and 11, the proposed ALR-BPNN algorithm had a fast convergence rate over the compared algorithms even in environments where the impulse noises are not generated. The value of the proposed algorithm is much more exerted in environments where the impulse noises are generated. As can be seen in Figs 12 and 13, regardless of whether the inputs are the uncorrelated or

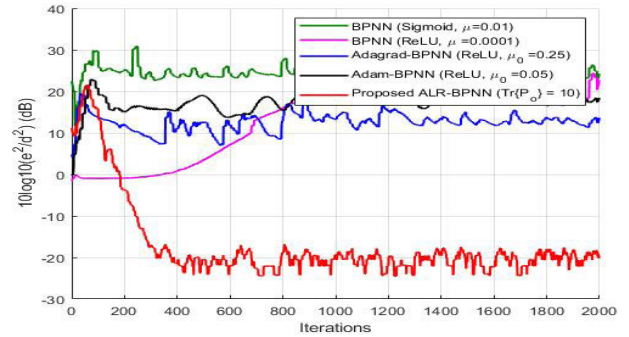


FIGURE 13. Normalized error curve with the impulsive noises (correlated multi-tonal signals by  $G(z)$ ,  $\alpha_1 = 30$ ,  $\alpha_2 = 2$ ).

correlated signals, other comparison algorithms have failed to maintain a low steady-state errors in the impulse noise environments. On the other hand, the proposed algorithm showed a good performance even for all inputs mixed with the impulsive noises as the learning rate automatically decreases when the error signals rapidly increases due to the impulsive noises.

### B. CASE 2

2000 training data and 500 test data were used to train the FNNs predicting the NASDAQ index. Number of the input features, including exchange rates by country, SHANGHAI index, Goldman index, etc., were 18. 5 simulations were performed according to the ratio of bad data among the training target data and the ratio of bad data was divided from

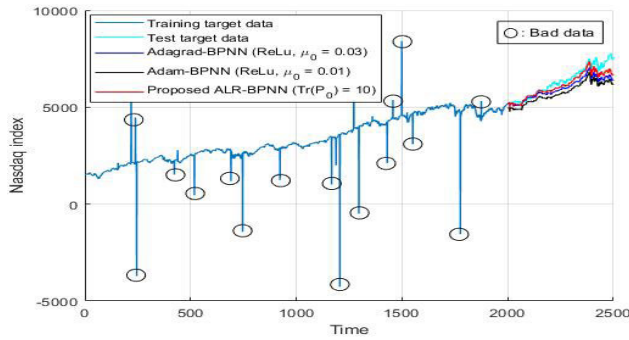


FIGURE 14. Training target data with 1% bad training data.

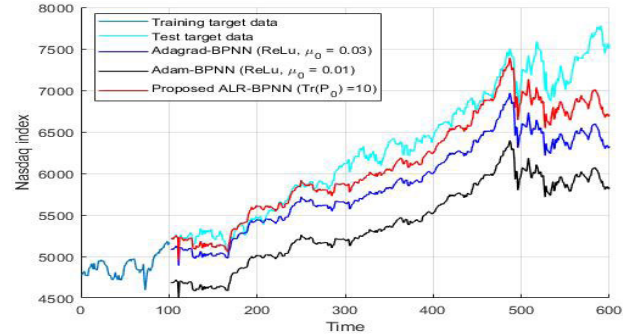


FIGURE 17. Test outputs of the algorithms with 20% bad training data ( $\alpha_1 = 5, \alpha_2 = 15$ ).

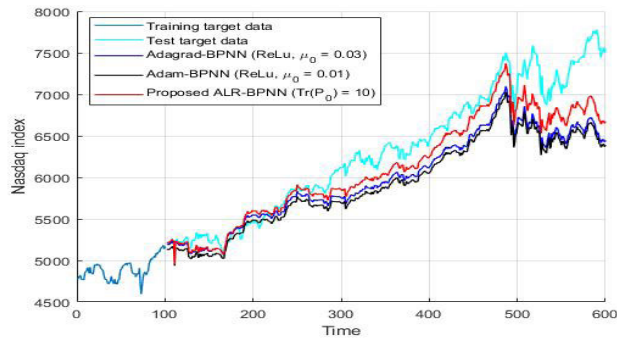


FIGURE 15. Test outputs of the algorithms without bad training data ( $\alpha_1 = 5, \alpha_2 = 15$ ).

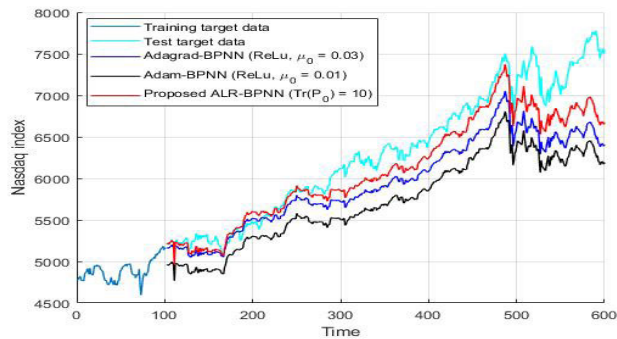


FIGURE 16. Test outputs of the algorithms with 10% bad training data ( $\alpha_1 = 5, \alpha_2 = 15$ ).

0 to 40%. As can be seen Fig. 14, bad data was randomly set to 2 to 6 times larger or smaller than the original training target data. The prediction accuracy of the algorithm was calculated by root-mean-square error (RMSE) using test data. Number of layers was also set to 3, and number of neurons at the input and 2th layers was set to 18 and 20. Comparison algorithms were set to Adagrad-BPNN and Adam-BPNN algorithms. The parameters,  $\beta_1$  and  $\beta_2$  used in the Adam-BPNN algorithm were also set to 0.99. All simulation results were presented by averaging 30 independent simulations.

In this simulation case, the proposed algorithm showed good prediction accuracy regardless of the ratio of bad data among the training data. As can be seen Fig. 15, 16, and 17, the proposed ALR-BPNN algorithm not only provided good prediction accuracy even in the absence of bad data, but also maintained good accuracy in the presence of bad data.

TABLE 2. RMSE of comparison algorithms for the ratio of bad training data.

	Ratio of bad training data (%)				
	0	10	20	30	40
RMSE of Adagrad	431.1	461.0	556.8	600.1	1006.7
RMSE of Adam	480.4	640.4	585.4	959.7	916.6
RMSE of proposed ALR	298.2	298.4	288.2	285.9	285.7

This prediction accuracy is specifically shown in TABLE 2. While the comparison algorithms had less accurate as the ratio of bad data increases, the proposed algorithm maintained good accuracy regardless of the ratio of bad data. Moreover, TABLE 1 showing number of multiplication of Adagrad-BPNN, Adam-BPNN, and proposed ALR-BPNN algorithms indicates that the proposed algorithm does not increase the computational complexity compared to the Adagrad and Adam algorithms commonly used in adaptive learning-rate method.

## V. CONCLUSION

This paper proposed a novel ALR-BPNN algorithm updating the learning rate in the direction of minimizing MSD at each hidden layer and showed for the first time that minimizing the MSD can effectively reduce the overall error of the neural network. The problem that exact value of the MSD at each hidden layer is not feasible was solved by setting the upper bound of the MSD. In addition, to be robustness to the impulsive noises, the proposed algorithm adopted the variance of the perturbation at each hidden layer through the variance of the error signals. The results of the two simulations, estimating nonlinear model target signals through the sequential input signals and estimating the actual NASDAQ data, showed how robust and excellent the proposed ALR-BPNN algorithm was even in the case of the impulsive noise. The proposed BPNN algorithm based on the MSD analysis will be utilized in various BPNN algorithms in the future.

## REFERENCES

[1] M. K. Weir, "A method for self-determination of adaptive learning rates in back propagation," *Neural Netw.*, vol. 4, no. 3, pp. 371–379, Jan. 1991.



- [2] S. Iranmanesh and M. A. Mahdavi, "A differential adaptive learning rate method for back-propagation neural networks," *World Acad. Sci., Eng. Technol.*, vol. 50, no. 1, pp. 285–288, 2009.
- [3] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural Networks for Perception*. Amsterdam, The Netherlands: Elsevier, 1992, pp. 65–93.
- [4] R. J. Erb, "Introduction to backpropagation neural network computation," *Pharmaceutical Res.*, vol. 10, no. 2, pp. 165–170, 1993.
- [5] H. Leung and S. Haykin, "The complex backpropagation algorithm," *IEEE Trans. Signal Process.*, vol. 39, no. 9, pp. 2101–2104, Sep. 1991.
- [6] S. Abid, F. Fnaiech, and M. Najim, "A fast feedforward training algorithm using a modified form of the standard backpropagation algorithm," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 424–430, Mar. 2001.
- [7] X. Yu, M. O. Efe, and O. Kaynak, "A general backpropagation algorithm for feedforward neural networks learning," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 251–254, Jan. 2002.
- [8] D. Sarkar, "Methods to speed up error back-propagation learning algorithm," *ACM Comput. Surv. (CSUR)*, vol. 27, no. 4, pp. 519–544, Dec. 1995.
- [9] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ, USA: Prentice-Hall, 1994.
- [10] M. F. Møller, *A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning*. Aarhus, Denmark: Aarhus Univ., Computer Science Department, 1990.
- [11] T. Gao, X. Gong, K. Zhang, F. Lin, J. Wang, T. Huang, and J. M. Zurada, "A recalling-enhanced recurrent neural network: Conjugate gradient learning algorithm and its convergence analysis," *Inf. Sci.*, vol. 519, pp. 273–288, May 2020.
- [12] R. Setiono and L. C. K. Hui, "Use of a quasi-Newton method in a feedforward neural network construction algorithm," *IEEE Trans. Neural Netw.*, vol. 6, no. 1, pp. 273–277, Jan. 1995.
- [13] S. Ouyang, Z. Bao, and G.-S. Liao, "Robust recursive least squares learning algorithm for principal component analysis," *IEEE Trans. Neural Netw.*, vol. 11, no. 1, pp. 215–221, Jan. 2000.
- [14] J. Shawash and D. R. Selviah, "Real-time nonlinear parameter estimation using the Levenberg–Marquardt algorithm on field programmable gate arrays," *IEEE Trans. Ind. Electron.*, vol. 60, no. 1, pp. 170–176, Jan. 2013.
- [15] H. Yu and B. M. Wilamowski, "Levenberg–Marquardt training," *Ind. Electron. handbook*, vol. 5, no. 12, p. 1, 2011.
- [16] J. Sum, C.-S. Leung, G. H. Young, and W.-K. Kan, "On the Kalman filtering method in neural network training and pruning," *IEEE Trans. Neural Netw.*, vol. 10, no. 1, pp. 161–166, Jan. 1999.
- [17] T. Gansler, S. L. Gay, G. M. M. Sondhi, and J. Benesty, "Double-talk robust fast converging algorithms for network echo cancellation," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 6, pp. 656–663, Nov. 2000.
- [18] M. Shao and C. L. Nikias, "Signal processing with fractional lower order moments: Stable processes and their applications," *Proc. IEEE*, vol. 81, no. 7, pp. 986–1010, Jul. 1993.
- [19] J. W. Hur, M. Lee, D. W. Kim, and P. G. Park, "A variable step-size robust saturation algorithm against impulsive noises," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, early access, Nov. 21, 2019, doi: 10.1109/TCSII.2019.2954918.
- [20] J. Yoo, J. Shin, and P. Park, "Variable step-size sign algorithm against impulsive noises," *IET Signal Process.*, vol. 9, no. 6, pp. 506–510, Aug. 2015.
- [21] P. P. Gandhi and V. Ramamurti, "Neural networks for signal detection in non-Gaussian noise," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2846–2851, Nov. 1997.
- [22] M. Ciolek and M. Niedzwiecki, "Detection of impulsive disturbances in archive audio signals," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 671–675.
- [23] S. P. V. Ebenezzer, "Detection of acoustic impulse events in voice applications," U.S. Patent 10242696, Mar. 26 2019.
- [24] J. C. Patra, R. N. Pal, B. N. Chatterji, and G. Panda, "Identification of nonlinear dynamic systems using functional link artificial neural networks," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 29, no. 2, pp. 254–262, Apr. 1999.
- [25] H. Zhao and J. Zhang, "Adaptively combined FIR and functional link artificial neural network equalizer for nonlinear communication channel," *IEEE Trans. Neural Netw.*, vol. 20, no. 4, pp. 665–674, Apr. 2009.
- [26] L. Xu, D. D. Huang, and Y. J. Guo, "Robust blind learning algorithm for nonlinear equalization using input decision information," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3009–3020, Dec. 2015.
- [27] S. Zhang and W. X. Zheng, "Recursive adaptive sparse exponential functional link neural network for nonlinear AEC in impulsive noise environment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4314–4323, Sep. 2018.
- [28] J. Shin, J. Yoo, and P. Park, "Variable step-size sign subband adaptive filter," *IEEE Signal Process. Lett.*, vol. 20, no. 2, pp. 173–176, Feb. 2013.
- [29] J.-H. Seo, S. M. Jung, and P. Park, "A diffusion subband adaptive filtering algorithm for distributed estimation using variable step size and new combination method based on the MSD," *Digit. Signal Process.*, vol. 48, pp. 361–369, Jan. 2016.
- [30] J. Leonard and M. A. Kramer, "Improvement of the backpropagation algorithm for training neural networks," *Comput. Chem. Eng.*, vol. 14, no. 3, pp. 337–341, Mar. 1990.
- [31] D. W. Kim, J. Hur, and P. Park, "Two-stage active noise control with online secondary-path filter based on an adapted scheduled-stepsize NLMS algorithm," *Appl. Acoust.*, vol. 158, Jan. 2020, Art. no. 107031.
- [32] A. Lydia and S. Francis, "Adagrad—An optimizer for stochastic gradient descent," *Int. J. Inf. Comput. Sci.*, vol. 6, no. 5, May 2019.
- [33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: <http://arxiv.org/abs/1412.6980>



**DONG WOO KIM** (Student Member, IEEE) received the B.S. and M.S. degrees in electronic and electrical engineering from the Pohang University of Science and Technology, Pohang, South Korea, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree in electronic and electrical engineering. His research interests include adaptive filtering algorithm and active noise control systems.



**MIN SU KIM** (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in electronic and electrical engineering from the Pohang University of Science and Technology, Pohang, South Korea, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree in electronic and electrical engineering. His research interests include deep learning based modeling and state estimation.



**JAHO LEE** received the B.S. degree in electronic and electrical engineering from Sungkyunkwan University, Suwon, South Korea, in 2017, and the M.S. degree in electronic and electrical engineering from the Pohang University of Science and Technology, Pohang, South Korea, in 2019. He is currently an Associate Researcher with the AI Laboratory, LG Electronics Inc., Seoul, implementing on-device intelligence on various home appliances. His research interests include reinforcement learning, time-series prediction, and neural network optimization.



**POOGEON PARK** (Senior Member, IEEE) received the B.S. and M.S. degrees in control and instrumentation engineering from Seoul National University, Seoul, South Korea, in 1988 and 1990, respectively, and the Ph.D. degree from Stanford University, Stanford, CA, USA, in 1995.

From 1996 to 2000, he was an Assistant Professor with the Pohang University of Science and Technology. Since 2006, he has been a Professor with the Electronic Electrical Engineering Department, Pohang University of Science and Technology. He has authored over 170 articles and the total citation for his articles is 9712. His research interest includes control and signal processing.

• • •