IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 18, 2019, accepted February 11, 2019, date of publication February 19, 2019, date of current version March 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2900071

A Biological Mechanism Based Structure Self-Adaptive Algorithm for Feedforward Neural Network and Its Engineering Applications

GUANG HAN[1], QI CHENG[2], XIAOYUN SUN[1], LI LI[1], AND WEIGUO DI[1]
[1]School of Electrical and Electronic Engineering, Shijiazhuang Tiedao University, Shijiazhuang 050043, China
[2]College of Automation, Harbin Engineering University, Harbin 150001, China

Corresponding author: Xiaoyun Sun (sunxyheb@stdu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 51674169.

ABSTRACT Feedforward neural network (FNN) is an information processing system that simulates human brain function to a certain extent by referring to the structure of a biological neural network and the working mechanism of biological neurons. As the most important part, the architecture of FNN essentially influences its application performance. This paper proposed a structure self-adaptive algorithm for FNN (SSAFNN) based on biological mechanism. First, self-adaptive neuron growing and pruning indexes are proposed based on the idea of biological neuron grow factor and neuron competition, respectively. The FNN structure is dynamically adjusted according to the growing and pruning indexes of hidden neurons. Second, the connect weights of FNN are automatically adjusted during the self-organizing process and trained by gradient descent method during the learning process. Third, the theoretical analysis is given that this proposed algorithm not only optimized the network structure but also ensured the convergence and performance of FNN. The proposed SSAFNN is tested on the benchmark problems in the field of classification and prediction and is applied in the engineering problems of anchor bolt non-destructive testing and wastewater effluent ammonia nitrogen predicting. The experiment results reveal the good performance and potentiality of SSAFNN in industrial applications.

INDEX TERMS Feedforward neural network, neuron growing index, neuron pruning index, structure self-adaptive.

I. INTRODUCTION

The structure and learning procedure of feedforward neural network is designed via the conception of structure of biological neural network and the working mechanism of biological neurons [1], [2]. FNN, due to its learning and universal approximating ability, has been widely applied in many fields, such as data classification, soft measurement, and nonlinear system modeling [3]–[6].

Traditionally, the FNN structure is designed by human experience and sufficient data. Once determined, it will not be adjusted, and only the connection weights for various tasks will be adjusted. On one hand, the FNN with large scale can learn the training samples well, but often lead to the occurrence of ''over-fitting'' phenomenon and the increase of calculation and storage [7], [8]. On the other hand, the FNN with small scale has better generalization ability, but has insufficient information processing capability for complex problems [9], [10]. Thus, how to dynamically adjust the structure of FNN, especially how to adjust the parameters in the FNN while dynamically adjusting the network structure, has always been an open problem that needs to be solved.

The FNN structure self-organizing problem mainly has the following two aspects: structure growing and structure pruning. Start with a small scale of neural network, the growing algorithm can add hidden layer neurons and their connection weights with other layers, for enhancing the approximating ability of neural networks. Several researches focus on this issue. A growing grid structure is proposed by Fritzke [11], which can be regarded as a growth self-organizing feature map. Wu and Chow [12] proposed a growth neural network based on genetic algorithm. Li *et al.* [13] proposed a forward

The associate editor coordinating the review of this manuscript and approving it for publication was Sabah Mohammed.

neural network structure growing algorithm based on the hidden layer activation function and its derivative function. Pruning algorithm can delete the redundant neurons and their connections weights in large scale neural network, to make the FNN a more suitable structure and improve the efficiency of FNN. For example, optimal brain surgeon (OBS) method for neuron pruning is proposed by Hassibi and Stork [14]; Zeng *et al.* [15] proposed the sensitivity analysis of neurons to prune hidden neurons. Jiang *et al.* [16] implemented a nonlinear inversion of resistivity imaging using pruning Bayesian neural network. There are also several neural network self-adaptive or self-organizing researches combining the advantages of growing algorithm and pruning algorithm. Islam *et al.* [17] proposed a new algorithm, called adaptive merging and growing algorithm (AMGA), in designing artificial neural networks (ANNs). Hsu *et al.* [18] proposed an adaptive growing-and-pruning neural network control (AGPNNC) system for modeling piezoelectric ceramic motor. Han and Qiao [19] and Qiao *et al.* [20] proposed two kind of self-organizing algorithm called constructing and pruning (CP) and hybrid constructing and pruning strategy (HCPS) for optimizing the structure of FNN.

Some existing self-organizing FNN algorithms are analyzed in the above, which overcomes some shortcomings during the FNN structure self-organizing, such as how to stop structural adjustment [17], [19]. But there are mainly some disadvantages that need to be solved. (1) The theoretical link between structural self-organizing algorithms and biological mechanisms has not been explained; (2) The ability to restructure FNN to solve complex problems that needs to be improved; (3) There are often several computation parameters to be select for different problems, the universality of structure self-organizing algorithms needs to improve.

In this paper, we proposed a new algorithm, called structure self-adaptive algorithm for feedforward neural network (AASFNN). For neuron growing phase, an adaptive neuron growing index is proposed by the concept of biological neuron growing factor; for neuron pruning phase, an adaptive neuron pruning index is proposed by the conception of biological neuron competition. The connection weights of FNN are automatically adjusted during the self-organizing process and trained by gradient descent method with adaptive learning rates during the learning process. The contributions of this work are as follows:

(1) Both the adaptive growing and pruning algorithm are proposed by the conception of biological mechanism. The difference between SSAFNN and other methods is the completely independence of growing mechanism and pruning mechanism. This design helps improving the ability of FNN for solving complex problems.

(2) The adaptive design of growing index, pruning index, and learning rate improved the universality of FNN for solving different problems.

(3) The theoretical analysis is given for the convergence guarantee of both the SSAFNN structure self-organizing process and learning process.

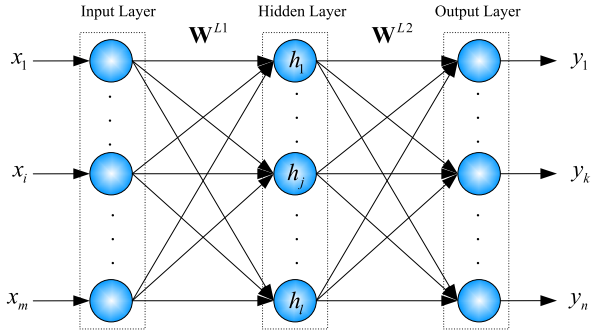


FIGURE 1. The basic structure of three layer FNN.

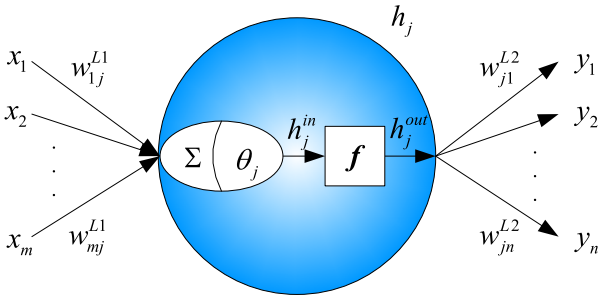


FIGURE 2. The structure of a artificial neuron.

The subsequent sections are organized as follows: Section II gives the basic conceptions of FNN; the adaptive growing and pruning index is designed in Section III; Section IV describes the structure self-adaptive process and learning process of SSAFNN in detail; Section V presents experiment results and discussions of SSAFNN in benchmark testing and engineering applications; Finally, Section VI draws the conclusions.

II. BASIC CONCEPTIONS OF FEEDFORWARD NEURAL NETWORK

The basic structure of the feedforward neural network consists of three or more layers, each of which is fully connected to the neurons of the previous layer and the latter layer, but there is no connection between the neurons in the same layer. A three layer feedforward neural network can be represented by Fig. 1.

According to the Fig. 1, the number of neurons in the input layer, the hidden layer, and the output layer are respectively: m, l, n. Assume that the training sample set is $\mathbf{x} = [x_1, x_2, \ldots, x_i, \ldots, x_m]^T$, the expected objective is $\mathbf{d} = [d_1, d_2, \ldots, d_k, \ldots, d_n]^T$, and the actual output is $\mathbf{y} = [y_1, y_2, \ldots, y_k, \ldots, y_n]^T$. The weight matrix connecting the input layer and the hidden layer is $\mathbf{W}^{L1}$, and the weight matrix connecting the hidden layer and the output layer is $\mathbf{W}^{L2}$.

In FNN, each neuron connects with the neurons from the previous and next layer. Let's take the hidden layer neuron h_j, shown in Fig. 2, as example.

In Fig. 2, the input of neuron of h_j is,

$$h_j^{in} = \sum_{i=1}^{m} w_{ij}^{L1} x_i + \theta_j \tag{1}$$

where, Σ represents the summation of the neuron input, θ_j represents the bias of the neuron. Then the output of h_j is

$$h_j^{out} = f\left(\sum_{i=1}^{m} w_{ij}^{L1} x_i + \theta_j\right) \tag{2}$$

where, the activation function f of the hidden layer neurons uses the sigmoid function.

$$f(x) = \frac{1}{1+e^{-x}} \tag{3}$$

The neurons of the input layer and the output layer adopt a linear function.

The work signal of FNN is transmitted from the input layer to the output layer via the procession of hidden layer. Thus we have

$$\begin{aligned} \mathbf{Y} &= \mathbf{H}^{out}\mathbf{W}^{L2} \\ &= f(\mathbf{W}^{L1}\mathbf{X} + \Theta)\mathbf{W}^{L2} \end{aligned} \tag{4}$$

where, $\mathbf{X}$ and $\mathbf{Y}$ represent the input and output of FNN respectively, $\mathbf{H}^{out}$ represents the hidden layer output matrix, $\mathbf{\Theta}$ represents the bias vector of all the hidden neurons.

$$\mathbf{X} = [x_1, x_2, \ldots, x_m]; \quad \mathbf{Y} = [y_1, y_2, \ldots, y_n];$$
$$\mathbf{H}^{out} = \left[h_1^{out}, h_2^{out}, \ldots, h_l^{out}\right]; \quad \mathbf{\Theta} = [\theta_1, \theta_2, \ldots, \theta_l].$$

$\mathbf{W}^{L1}$ and $\mathbf{W}^{L2}$ are the weight matrices connecting the neighboring layers; m, l, n represent the neuron number of input, hidden, output layer respectively.

$$\mathbf{W}^{L1} = \begin{bmatrix} w_{11}^{L1}, w_{12}^{L1}, \ldots, w_{1l}^{L1} \\ w_{21}^{L1}, w_{22}^{L1}, \ldots, w_{2l}^{L1} \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ w_{m1}^{L1}, w_{m2}^{L1}, \ldots, w_{ml}^{L1} \end{bmatrix}$$

$$\mathbf{W}^{L2} = \begin{bmatrix} w_{11}^{L2}, w_{12}^{L2}, \ldots, w_{1n}^{L2} \\ w_{21}^{L2}, w_{22}^{L2}, \ldots, w_{2n}^{L2} \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ w_{l1}^{L2}, w_{l2}^{L2}, \ldots, w_{ln}^{L2} \end{bmatrix}$$

III. THE BIOLOGICAL MECHANISM BASED GROWTH AND PRUNING FNN DESIGN AND ANALYSIS

As shown in Fig. 1, for a FNN, the number of input and output neurons in the network is determined by the training samples, but the selection of hidden layer neurons is not unique. The hidden layer neurons are all nonlinear nodes that satisfy the sigmoid function, which has a great influence on the network. Excessive neurons will increase the complexity of the network while too few neurons will reduce the nonlinear fitting ability of the network. From the above, the one key task of determining the network structure is to determine the number of hidden layer neurons in the network. However, before training, the relationship between input data and output data is unknown. Early determination of the number of neurons in the hidden layer leads to great blindness. Therefore, this paper proposes a structure self-adaptive algorithm for dynamically adjusting hidden layer nodes during training.

Although the research on the structure of artificial neural network has begun to approach the biological neural system, how to simulate the neuron growth and elimination of biological neural network is still an urgent problem. Some neurobiological studies have inspired us.

A. DESIGN OF ADAPTIVE NEURON GROWING INDEX

In terms of neuron growth, the survival of developing neurons depends on the supply of specific proteins called growth factors. The first identified growth factor is the neuron growth factor (NGF). Levi-Montalcini *et al.* [21] and Aloe *et al.* [22] first demonstrated in ground breaking experiments that NGF stimulates sensory and sympathetic neurons to protrude from the neuritis. They went on to show that neuron survival also requires NGF.

Inspired by the NGF conception, we define the neuron growing index (NGI) of FNN.

$$\mathrm{NGI}_j = \frac{1}{Tn}\sum_{t=1}^{T}\sum_{k=1}^{n}\frac{w_{jk}^{L2}(t)h_j^{out}(t)}{y_k(t)} \tag{5}$$

where, w_{jk}^{L2} is the connection weight of the jth node of the hidden layer to the kth node of the output layer, and h_j^{out} is the output of the jth neuron of the hidden layer. T is the total number of learning samples, for the large amount of samples or online learning, we choose the proper T with a slide window.

Together with NGI, an adaptive threshold of FNN neuron growing $\mathrm{G_{th}}$ is also designed:

$$\mathrm{G_{th}} = \frac{e^{l/\max l}}{\alpha l}\sum_{j=1}^{l}\mathrm{NGI}_j \tag{6}$$

where, l is the number of hidden neurons before growing, max l is the upper bound of hidden neurons and α is the adaptive growing coefficient, range form (0, 1]. In this paper, the upper bound of l is set to 60. We will discuss the value of α in section V.

In the growing process, if the NGI of the jth hidden neuron is larger than the $\mathrm{G_{th}}$, which means, the neuron growth factor of the jth neuron is satisfied. That is, if $\mathrm{NGI}_j > \mathrm{G_{th}}$, the jth neuron affords too heavy tasks, the growing condition is fulfilled, a new neuron will be added for helping neuron j.

B. DESIGN OF ADAPTIVE NEURON PRUNING INDEX

In terms of neuron pruning, once a population of neurons that dictate a particular target is defined by cell death, the surviving neurons compete with each other for synaptic territory. This competition usually leads to the loss of some terminal branches and previously formed synapses, which is the pruning process. Pruning provides a mechanism to ensure

that a particular group of neurons is properly and completely dictated by a target [23]. In some cases, pruning also provides a mechanism to correct errors [24].

Inspired by this point of view, we have designed the neuron pruning index. Unlike neuron growing, the pruning or eliminating of a biological neuron always begins with a competition among all the neurons.

According to the three-layer neural network structure diagram given in Fig. 1, the output of the output layer neurons is shown by (4). And for the kth output of FNN, the expression can be extended:

$$\begin{aligned} y_k(t) &= \sum_{j=1}^{l} w_{jk}^{L2}(t) h_j^{out}(t) \\ &= \sum_{j=1}^{l} w_{jk}^{L2}(t)(h_j^{out}(t) - \overline{h_j}) + \sum_{j=1}^{l} w_{jk}^{L2}(t)\overline{h_j} \end{aligned} \tag{7}$$

where, h_j^{out} is the output of the jth neuron in the hidden layer, and $\overline{h_j}$ is the average output of the jth neuron in the hidden layer,

$$\bar{h}_j = \frac{1}{T}\sum_{t=1}^{T} h_j^{out}(t) \tag{8}$$

where, T is the total number of learning samples, for the large amount of samples or online learning, the proper T can be chosen with a slide window.

In (7), the first term can be understood as the output fluctuation of the neuron, which can be approximated by the output standard deviation. The second term is the mean value of the output of the hidden layer neurons, which is determined by the performance of the network. The smaller the output means the smaller the sensitivity of the neuron to the network outputs, and this neuron lacks of competitiveness. It can be used as an effective feature to judge the sensitivity of the hidden layer neuron.

In this paper, the neuron pruning index (NPI) is designed according to the output variance and output mean of hidden layer nodes:

$$\mathrm{NPI}_j = \sqrt{\left(\frac{1}{T}\sum_{t=1}^{T}(h_j^{out}(t) - \overline{h_j})^2\right)}\left(\frac{1}{T}\sum_{n=1}^{T} h_j^{out}(t)\right) \tag{9}$$

Together with NPI, an adaptive threshold of FNN neuron pruning $\mathrm{P_{th}}$ is also designed:

$$\mathrm{P_{th}} = e^{l/\max l}\frac{\beta}{l}\left(\sum_{j=1}^{l}\mathrm{NPI}_j\right) \tag{10}$$

where, l and $max\ l$ are defined in a manner same to the adaptive growth index. β is the adaptive pruning coefficient, range form (0, 1]. We will discuss the value of β in section V.

In neuron pruning process, if the NPI of the jth hidden neuron is less than the $\mathrm{P_{th}}$, which means the jth neuron lacks of competitiveness among all the neurons. That is, if $\mathrm{NPI}_j < \mathrm{P_{th}}$, the jth neuron failed in the competition and contributed little to the network. Then this neuron can be pruned or eliminated.

IV. THE LEARNING PROCESS AND STRUCTURE SELF-ADAPTIVE OF SSAFNN

A. THE LEARNING PROCESS OF UNCHANGING STRUCTURE PHASE

In structure unchanging phase, the FNN is trained by gradient descent method. Define the training index J as,

$$J = \frac{1}{2}\sum_{k=1}^{n} e_k^2(t) = \frac{1}{2}\sum_{k=1}^{n}(d_k(t) - y_k(t))^2 \tag{11}$$

The update of the connect weights between the hidden layer and the output layer can be express as,

$$w_{jk}^{L2}(t+1) = w_{jk}^{L2}(t) + \eta(t)\Delta w_{jk}^{L2}(t) \tag{12}$$

where,

$$\Delta w_{jk}^{L2}(t) = \frac{\partial J}{\partial w_{jk}^{L2}(t)} = -e_k(t)h_j^{out}(t) \tag{13}$$

Similarly, the update of the connect weights between the input layer and the hidden layer can be express as,

$$w_{ij}^{L1}(t+1) = w_{ij}^{L1}(t) + \eta(t)\Delta w_{ij}^{L1}(t) \tag{14}$$

where,

$$\begin{aligned} \Delta w_{ij}^{L1}(t) &= \frac{\partial J}{\partial w_{ij}^{L1}(t)} \\ &= -\mathbf{e}(t)\Psi(t)x_j(t) \\ \mathbf{e}(t) &= [e_1(t), e_2(t), \ldots, e_k(t), \ldots, e_n(t)] \\ \Psi(t) &= [\psi_1(t), \psi_2(t), \ldots, \psi_k(t), \ldots, \psi_n(t)]^{\mathrm{T}} \\ \psi_k(t) &= \sum_{j=1}^{l} w_{jk}^{L2}(t)h_j^{out}(t)(1 - h_j^{out}(t)) \end{aligned} \tag{15}$$

One important key to guarantee the convergence of FNN learning process and improve the learning ability of FNN is properly choosing the learning rate. Kramer [25] proved that when the Euclidean norm of the gradient vector reaches a sufficiently small gradient threshold, the gradient descent algorithm has converged. Or we can say, when the absolute rate of mean square error variation for each round is small enough, the gradient descent algorithm has converged. Some researches including our former research [26], [27] have also proved that properly choose learning rate η can ensure the convergence of structure unchanged FNN learning process.

Therefore, an adaptive learning rate is chosen via (16).

$$\eta(t) = \frac{1}{\sum_{j=1}^{l}(h_j^{out}(t))^2} \tag{16}$$

The detailed analysis of (16) can be find in [27], and we will not expended discuss it here.

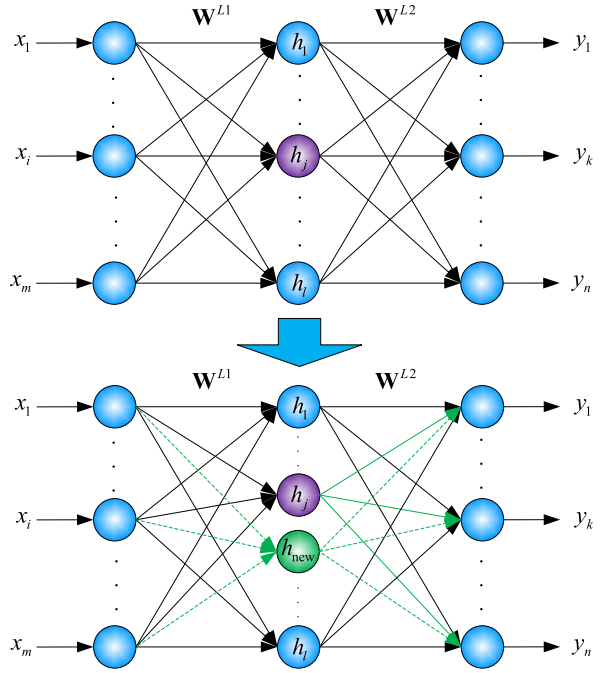


FIGURE 3. The growing procedure of FNN.

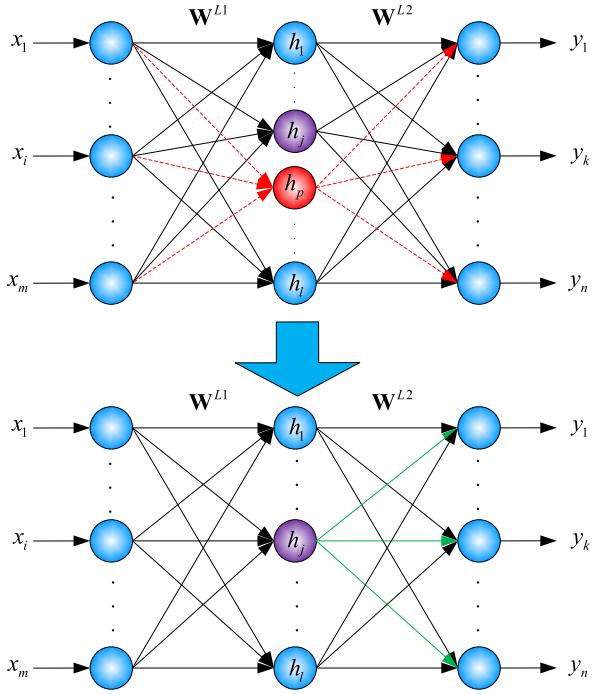


FIGURE 4. The pruning procedure of FNN.

B. THE DESIGN AND ANALYSIS OF GROWING PROCESS

The growing procedure of SSAFNN can be illustrated in Fig. 3.

In the growing procedure, if the NGI of jth hidden neuron (the purple neuron in Fig. 3) is over the G_{th}, then a new neuron (the green neuron in Fig. 3) will be added to share the contribution. The key to the success of the neuron growth algorithm is to ensure the convergence of network errors during the growth process. Here we would like to discuss this issue.

Let $\mathbf{w}_{new}^{L1}$ and $\mathbf{w}_{new}^{L2}$ represent the connect weights vector of the new added neuron, ζ represents the split coefficient of jth neuron.

Theorem 1: The network error will remain unchanged if the (17) is satisfied during the neuron growing process.

$$\begin{cases} \mathbf{w}_j^{L1}(t+1) = \mathbf{w}_j^{L1}(t) \\ \mathbf{w}_j^{L2}(t+1) = \zeta \mathbf{w}_j^{L2}(t) \\ \mathbf{w}_{new}^{L1}(t+1) = \mathbf{w}_j^{L1}(t) \\ \mathbf{w}_{new}^{L2}(t+1) = (1-\zeta)\mathbf{w}_j^{L2}(t) \end{cases} \tag{17}$$

Proof : Assume $\mathbf{e}_G(t+1)$ is the error vector after growing, $\mathbf{e}_G(t)$ is the error vector before growing and $\mathbf{d}$is the training objective, then we have

$$\mathbf{e}_G(t) = \mathbf{d} - \mathbf{y}(t) \tag{18}$$

$$\mathbf{e}_G(t+1) = \mathbf{d} - \mathbf{y}(t+1) \tag{19}$$

Notice that,

$$\begin{aligned} \mathbf{y}(t+1) &= \mathbf{W}^{L2}(t+1)\mathbf{H}(t+1) \\ &= \mathbf{w}_j^{L2}(t+1)h_j^{out}(t+1) + \mathbf{w}_{new}^{L2}(t+1)h_{new}^{out}(t+1) \\ &\quad + \sum_{i=1,i\neq j}^{l} \mathbf{w}_i^{L2}(t+1)h_i^{out}(t+1) \end{aligned} \tag{20}$$

According to (17),

$$\begin{aligned} \mathbf{y}(t+1) &= \zeta \mathbf{w}_j^{L2}(t)h_j^{out}(t) + \sum_{i=1,i\neq j}^{l} \mathbf{w}_i^{L2}(t)h_i^{out}(t) \\ &\quad + (1-\zeta)\mathbf{w}_j^{L2}(t)h_{new}^{out}(t) \\ &= \mathbf{W}^{L2}(t)\mathbf{H}(t) \\ &= \mathbf{y}(t) \end{aligned} \tag{21}$$

So,

$$\mathbf{e}_G(t+1) = \mathbf{d} - \mathbf{y}(t+1) = \mathbf{d} - \mathbf{y}(t) = \mathbf{e}_G(t) \tag{22}$$

$\therefore$ The network error remains unchanged, which means the network error convergence of FNN growing process is ensured.

Remark 1: According to the section III, the growth algorithm is similar to the neuron cell division in the biological systems. From biology, neuron cell division belongs to the mitosis that makes every mother cell splits into two daughter cells with similar size and function. Inspired by this, the paper defines the range of the ζ as [0.5, 0.7], which ensures that the two daughter nodes after growth algorithm have similar contributions.

C. THE DESIGN AND ANALYSIS OF PRUNING PROCESS

The pruning procedure of SSAFNN is illustrated in Fig. 4.

In the pruning procedure, if the NPI of hidden neuron p (the red neuron in Fig. 4) is less than P_{th}, it means neuron p

failed in the neuron competition, it's now a redundant neuron and will be pruned. The contribution of neuron p will be added on the neuron j (the purple neuron in Fig. 4), which contributes the second last among all the neurons. The output error convergence analysis is given below.

Assume neuron p is the redundant neuron and neuron j contribute the second last of all neurons, let $\mathbf{w}_i^{L1}$, $\mathbf{w}_j^{L2}$ and $\mathbf{w}_p^{L1}$, $\mathbf{w}_p^{L2}$ represent the connect weights vectors, respectively.

Theorem 2: The network error will remain unchanged if the (23) is satisfied during the neuron pruning process.

$$\begin{cases} \mathbf{w}_j^{L1}(t+1) = \mathbf{w}_j^{L1}(t) \\ \mathbf{w}_j^{L2}(t+1) = \dfrac{\mathbf{w}_j^{L2}(t)h_j^{out}(t) + \mathbf{w}_p^{L2}(t)h_p^{out}(t)}{h_j^{out}(t)} \end{cases} \tag{23}$$

Proof: Assume $\mathbf{e}_P(t+1)$ is the error vector after growing, $\mathbf{e}_P(t)$ is the error vector before growing and $\mathbf{d}$is the training objective, then we have

$$\mathbf{e}_P(t) = \mathbf{d} - \mathbf{y}(t) \tag{24}$$

$$\mathbf{e}_P(t+1) = \mathbf{d} - \mathbf{y}(t+1) \tag{25}$$

Notice that after pruning,

$$\begin{aligned} \mathbf{y}(t+1) &= \mathbf{W}^{L2}(t+1)\mathbf{H}(t+1) \\ &= \mathbf{w}_j^{L2}(t+1)h_j^{out}(t+1) \\ &\quad + \sum_{i=1, i\neq j, i\neq p}^{l} \mathbf{w}_i^{L2}(t+1)h_i^{out}(t+1) \end{aligned} \tag{26}$$

According to (23),

$$\begin{aligned} \mathbf{y}(t+1) &= \mathbf{w}_j^{L2}(t+1)h_j^{out}(t) + \sum_{i=1, i\neq j, i\neq p}^{l} \mathbf{w}_i^{L2}(t)h_i^{out}(t) \\ &= \mathbf{w}_j^{L2}(t)h_j^{out}(t) \\ &\quad + \mathbf{w}_p^{L2}(t)h_p^{out}(t) + \sum_{i=1, i\neq j, i\neq p}^{l} \mathbf{w}_i^{L2}(t)h_i^{out}(t) \\ &= \sum_{i=1}^{l} \mathbf{w}_i^{L2}(t)h_i^{out}(t) \\ &= \mathbf{y}(t) \end{aligned} \tag{27}$$

So,

$$\mathbf{e}_G(t+1) = \mathbf{d} - \mathbf{y}(t+1) = \mathbf{d} - \mathbf{y}(t) = \mathbf{e}_G(t) \tag{28}$$

$\therefore$ The network error remains unchanged, which means the network error convergence of FNN pruning process is ensured.

The main step of the learning process of SSAFNN can be summarized in Fig. 5.

Remark 2: The epoch τ_1, τ_2, τ_3 and the accuracy rate Ar_f are specified by the users. After many experiments, epoch τ_2, τ_3 often choosing similar numbers and τ_3 should choose a bit longer for pre-training; Ar_f usually used as the stop condition of the algorithm.

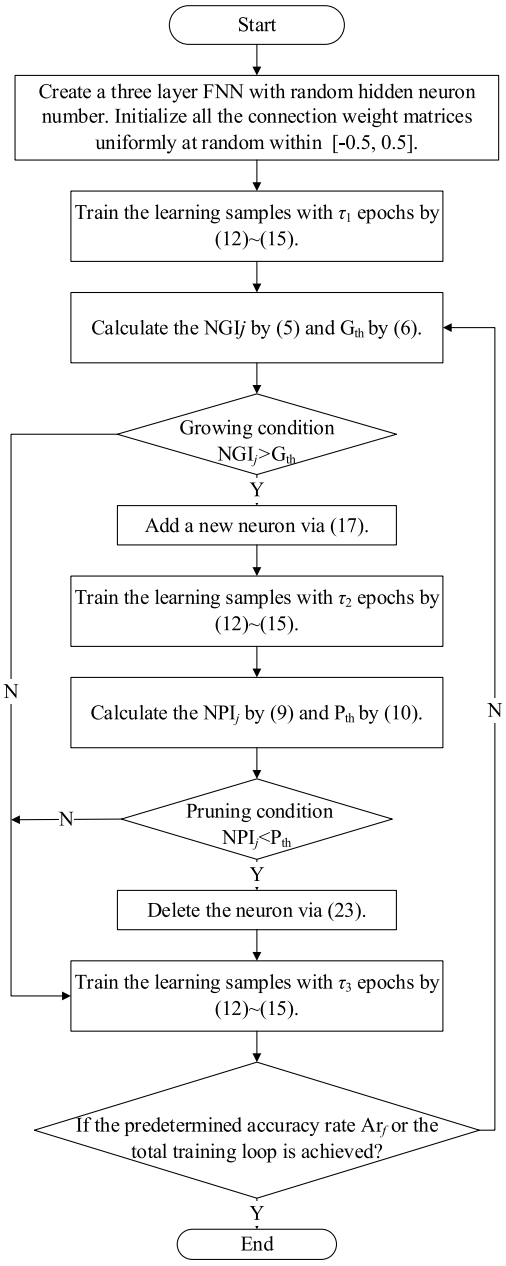


FIGURE 5. Learning flow chart of SSAFNN.

V. PERFORMANCE EVALUATIONS

In this section, the proposed SSAFNN is tested and applied into two kinds of problems: the classification problem and the prediction problem. In section A.1, three benchmark classification data sets with various characters are used for the proposed method effectiveness testing, and the method is applied in engineering problem: A.2 bolts nondestructive testing. In section B.1, the Mackey–Glass time series is used for the proposed method prediction effectiveness testing, and the method is applied in engineering problem: B.2 effluent ammonia nitrogen concentration (S_{NH}) prediction in wastewater treatment process. All the experiments are programmed with Matlab version 8.6, and run on a PC with Core I3

TABLE 1. Characteristic of three benchmark classification probelms.

Problem	Input attributes	Output classes	patterns	Training patterns	Testing patterns
Breast cancer	9	2	683	563	120
Iris	4	3	150	110	40
Yeast	8	10	1484	1112	372

TABLE 2. Detailed information of yeast dataset.

Name of classes	Data numbers
CYT (cytosolic or cytoskeletal)	461
NUC (nuclear)	429
MIT (mitochondrial)	244
ME3 (membrane protein, no N-terminal signal)	163
ME2 (membrane protein, uncleaved signal)	51
ME1 (membrane protein, cleaved signal)	44
EXC (extracellular)	35
VAC (vacuolar)	30
POX (peroxisomal)	20
ERL (endoplasmic reticulum lumen)	5

3.6 GHz and 3.68 GB of RAM, under a Microsoft Windows 10 operating system.

A. DATA CLASSIFICATION PROBLEMS

1) BENCHMARK CLASSIFICATION EVALUATIONS

Three generally classification problems from the UCI Machine Learning Repository [28] are used for evaluating the performance of the proposed methods, namely they are Breast cancer, Iris and Yeast. The characteristic of the three problems are illustrated in Table 1.

The Breast cancer data included 683 case, with 444 benign and 239 malignant. In this experiment, 364 benign, 199 malignant were selected as training data, and the remaining were used as test data. The input for each case is the nine feature quantities in the sampled tissue, with both benign and malignant results.

The Iris data consisted of 150 data sets, including Iris versicolour, Iris setosa, and Iris virginica, each of which has 50 groups. 110 groups were selected as training data in each of them, and the remaining 40 groups were used as test data.

The Yeast data consisted of 1484 data sets with 8 attributes and 10 classes distributes, as Table 2. The choosing of training and testing samples obeys the ratio 4:1.

These three data sets included various characteristics. The Breast cancer dataset has 9 attributes for 2 classes, but the two categories are relatively vague and hard to classify according to the former research; The Iris dataset has small amount of data and easy to classify; The difficulties of Yeast dataset classification are the lack of attributes information and the unbalanced distribution of different classes.

For all experiments, the learning rate was chosen adaptively according to (16). The connection weights of ANNs were initialized to random values within the range $[-0.5, 0.5]$, and the total training loop is set to 100. The training parameters are chosen according to Table 3.

TABLE 3. SSAFNN parameters for classification problems.

Problem	τ_1	τ_2	τ_3	α	β	Ar_f
Breast cancer	50	5	5	0.6	0.3	0.90
Iris	50	5	5	0.5	0.15	0.95
Yeast	50	10	10	0.7	0.1	0.70

TABLE 4. Performance comparison of different algorithms (initial 2 nodes).

Dataset	Algorithms	Average No. of Hidden nodes	Range of hidden nodes	Recognition rate
Breast cancer	**SSAFNN**	**12.60**	**9~16**	**90.67**
	AMGA	27.62*	25~31*	63.64*
	AGPNNC	-	27~37*	61.62*
	CPNN	21*	18~24*	66.71*
	HCPS	7*	-	74.97*
Iris	**SSAFNN**	**18.60**	**11~25**	**93.64**
	AMGA	22.48*	21~25*	93.53*
	AGPNNC	-	21~25*	94.17*
	CPNN	21*	19~23*	95.35*
	HCPS	2*		98.67*
Yeast	**SSAFNN**	**14.46**	**11~19**	**64.02**
	AMGA	57.60*	49~65*	47.01*
	AGPNNC	-	54~70*	49.33*
	CPNN	42*	38~48*	52.10*
	HCPS	19*	-	58.21*

- The results are non-significant
* the results are same as the original papers

TABLE 5. Performance comparison of different algorithms (INITIAL 50 NODES).

Dataset	Algorithms	Average No. of Hidden nodes	Range of hidden nodes	Recognition rate
Breast cancer	**SSAFNN**	**12.40**	**8~14**	**85.59**
	AMGA	26.62*	23-31*	63.64*
	AGPNNC	32*	27~37*	61.76*
	CPNN	22*	20~24*	66.41*
	HCPS	6*	-	76.17*
Iris	**SSAFNN**	**31.42**	**23~36**	**93.33**
	AMGA	22.88*	21~25*	94.33*
	AGPNNC	-	22~26*	93.27*
	CPNN	19*	17~21*	95.57*
	HCPS	4*	-	98.33*
Yeast	**SSAFNN**	**21.38**	**15~26**	**64.15**
	AMGA	55.56*	51~63*	48.55*
	AGPNNC	-	53~65*	47.89*
	CPNN	42*	37~47*	52.81*
	HCPS	18*	-	58.80*

- The results are non-significant
* the results are same as the original papers

Table 4 and Table 5 reveal the classification performance of SSAFNN with initial 2 and 50 nodes by 50 times independent runs, together with the comparisons of other FNN structure adjustments algorithms, such as AMGA in [17], AGPNNC in [18], CPNN in [19] and HCPS in [20]. Need to point out that the results of AMGA are acquired from [19].

One can see that all the three benchmark classification problems are well solved by SSAFNN. In Breast cancer and Yeast classification problems, the recognition rate of SSAFNN is better than the other method, and SSAFNN has a

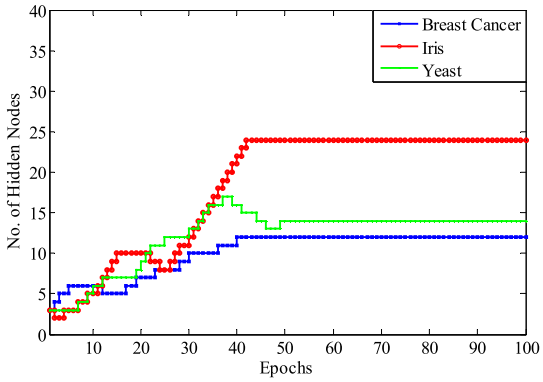


FIGURE 6. Updating process of SSAFNN initial with 2 nodes.

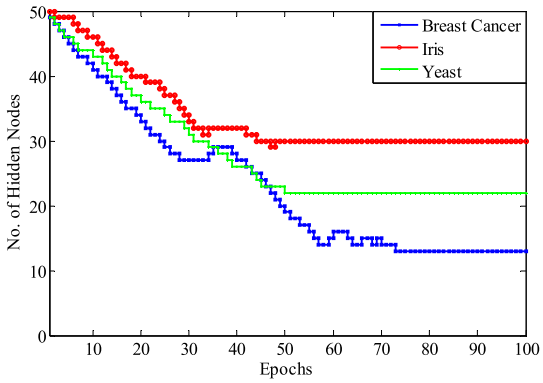


FIGURE 7. Updating process of SSAFNN initial with 50 nodes.

more compact structure. For Iris classification, SSAFNN has similar classification performance and structure compared with other methods.

Fig. 6 and Fig. 7 dedicate the updating process of hidden nodes during training. By analyzing Table 4, Table 5, Fig. 6 and Fig. 7, the following specific analysis can be made:

(1) The SSAFNN increased the ability for solving complex classification problems without losing the efficiency in solving easy classification problems. Choosing a larger α value will help solve the problems with large amount of data, and a smaller β value and longer training steps in structure adjusting process will help solve the problems with more complex characteristics. The Ar_f for each problems are set to be high values which are not ease to achieve.

(2) The number of FNN hidden neurons can be well self-adaptive, and the final structure of SSAFNN can be more compact compare with other methods, especially for complex problems.

(3) For each problem, the appropriate structure of FNN should be a range rather than a fixed value. From the experiment results, one can see that the appropriate range from initial 2 and 50 nodes are not the same. It is because on one hand the growing mechanism and pruning mechanism are designed separately in SSAFNN, on the other hand, the initial value of connection weight matrices are randomly generated in each run.

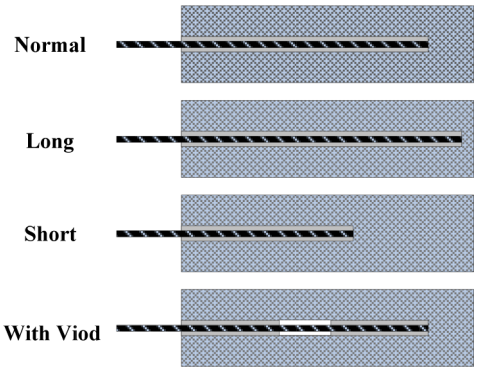


FIGURE 8. Schematic diagram of anchor bolt types.

FIGURE 9. Physical map of anchor bolt models.

2) ENGINEERING CLASSIFICATION PROBLEM: BOLT NON-DESTRUCTIVE TESTING

The anchor bolt technology can fully excavate the strength and self-stability of the rock and soil, strengthen and support the surrounding rock. Thus, it can reduce the weight of the structure, save engineering materials, and reduce the area of the site. However, the anchor bolt construction technology is highly concealed, making it difficult to find quality problems. Once quality problems occur, it will cause huge losses of people and property. Therefore, the nondestructive detection of anchor bolt system becomes an important task. The bolt non-destructive detection problem can transform into classification problem. There are mainly four types anchor bolt models in actual engineering, including normal anchor bolt, long anchor bolt, short anchor bolt and anchor bolts with voids. Fig. 8 dedicated the four anchor bolt types and the actual anchor bolt model is shown in Fig. 9.

The stress wave reflection method is used to obtain the excitation response signal of the bolt. The head of the anchor is tapped with a hammer excitation signal, and the anchor tip acceleration curve is received by an acceleration sensor

TABLE 6. SSAFNN parameters for bolt non-destructive testing.

τ_1	τ_2	τ_3	α	β	Ar_f
50	**5**	**5**	**0.4**	**0.3**	**0.95**

TABLE 7. SSAFNN performance for bolt non-destructive testing.

Initial No. of Hidden Nodes	Average No. of Hidden nodes	Range of hidden nodes	Recognition Rate(%)	Recognition Accuracy(%)
2	**17.30**	**13~23**	**93.78**	**100**
50	**22.88**	**17~26**	**90.29**	**100**
Fixed 2	-	-	71.92	58.50
Fixed 50	-	-	88.28	100

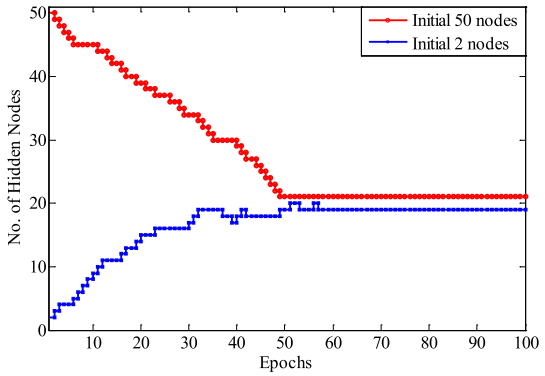


FIGURE 10. Nodes updating process of bolt nondestructive testing.

mounted at the free end. Since the energy carried by the anchor reflection acceleration signal in different states is different, the wavelet packet decomposition is used to decompose the reflected signal into different frequency bands, and the energy values corresponding to the different frequency bands are used as feature vectors of the system. Here are the wavelet packet decomposition steps.

Step 1: Perform three layers of wavelet packet decomposition on four types of acceleration signals to obtain low frequency signals and high frequency signals of each layer;

Step 2: Reconstructing the wavelet coefficients of the four signals from low frequency to high frequency, and calculating the reconstructed energy;

Step 3: Construct and normalize the feature vectors.

Thus, the bolt non-destructive classification data is constructed with 8 input attributes and 4 output classes. We compared SSAFNN with fix structure FNN. According to the analysis in benchmark data simulation, the parameters selections of SSAFNN are shown in Table 6. The total training loop is set to 100.

The performance of SSAFNN for solving bolt nondestructive testing are demonstrated in Table 7, Fig. 10 and Fig. 11.

The results show that the bolt classification problem is well solved and the recognition rate has been effectively increased by self adjusting the FNN structure. Here are some comments by analyzing Table 7, Fig. 10 and Fig. 11:

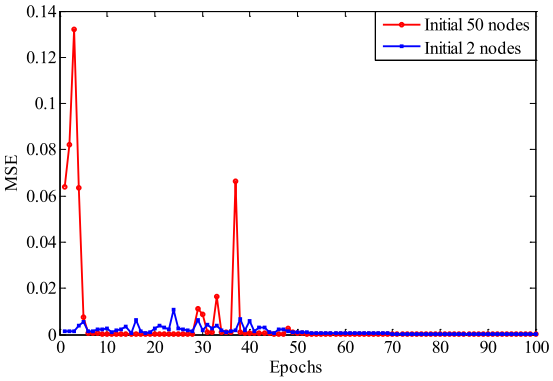


FIGURE 11. Training MSE of bolt nondestructive testing.

TABLE 8. SSAFNN parameters for mackey-glass time series prediction.

τ_1	τ_2	τ_3	α	β	Ar_f
50	5	5	0.3	0.3	0.98

(1) The appropriate structure of FNN is range from 13∼26. Too few and too many hidden nodes will cause lower recognition accuracy, and too few hidden nodes sometimes cause the failure of recognition.

(2) The MSE will oscillation during the structure adjusts process. Too many hidden nodes have an obviously higher MSE, and this means start with a small scale, SSAFNN will achieve a higher recognition rate, and start with a large scale of neurons will caused the significant increase of local minimums.

B. PREDICTION PROBLEMS

1) MACKEY–GLASS TIME SERIES PREDICTION

The Mackey–Glass time series prediction is a classical benchmark problem, which is constructed with the following equation,

$$x(t+1) = (1-a)\,x(t) + \frac{bx(t-\tau)}{1+x^c(t-\tau)} \tag{29}$$

where, the parameters we defined are $a = 0.1$, $b = 0.2$, $c = 10$, $\tau = 17$, $x(0)= 1.2$.

The prediction model is as follows:

$$\hat{x}(t+4) = \hat{f}(x(t), x(t+1), x(t+2), x(t+3)) \tag{30}$$

As shown in (30), there are 4 inputs and 1 output in this network. And the training parameters we set for Mackey–Glass time series prediction is shown in Table 8. The total training loop is set to 200.

The experiment results are as follows:

There are 500 samples used for performance testing. Fig. 12 exhibits that SSAFNN can predict Mackey-Glass time series well with both initial 2 and 50 nodes. Fig. 13 exhibits the training process of SSAFNN with 200 training epochs. The appropriate hidden node range for this problem is 12∼21 with initial 2 nodes and 15∼19 with initial 50 nodes. Compared with the fixed structure and other structure

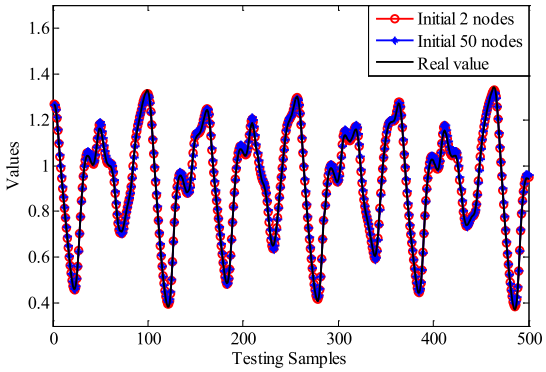


FIGURE 12. Results comparison of Mackey-Glass time series prediction.

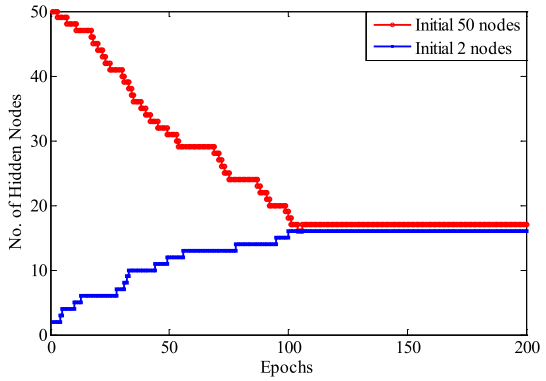


FIGURE 13. Nodes updating process of Mackey-Glass time series prediction.

TABLE 9. Perfrmance comparison for mackey-glass time series prediction.

Algorithms	Initial No. of Hidden nodes	Average No. of Hidden nodes	Range of hidden nodes	Testing RMSE
2	2	16.40	12~21	0.0083
50	50	17.40	12~19	0.0143
Fixed	2	-	-	0.0409
Fixed	50	-	-	0.0451
AGPNNC	2*	10*	-	0.0123*
AGPNNC	20*	10*	-	0.0171*
AMGA	2*	12*	-	0.0112*
AMGA	20*	12*	-	0.0134*

self- adjust algorithms, Table 9 shows SSAFNN achieved better testing results. Notice that AGPNN and AMGA do the Mackey-Glass time series prediction with initial 2 and 20 nodes. Here are some comments:

(1) SSAFNN has good performance in Mackey-Glass time series prediction. That means SSAFNN can be applied to soft measurements in practical problems.

(2) Same as the conclusions obtained in the classification problems: SSAFNN with few initial hidden nodes will achieve a better testing performance. We can also obtain this result when analyze other structure self-adjusting methods.

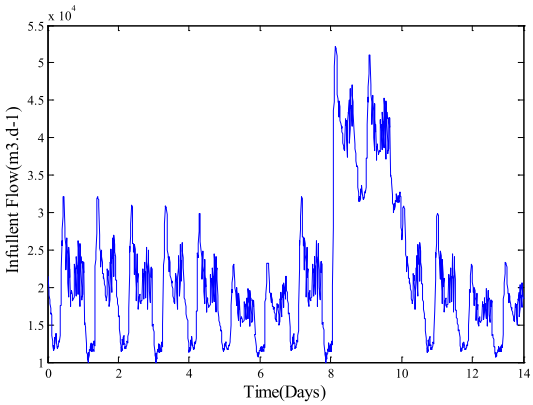


FIGURE 14. Rain weather influent.

TABLE 10. SSAFNN parameters for effluent S_{NH} prediction.

τ_1	τ_2	τ_3	α	β	Ar_f
50	5	5	0.3	0.3	0.95

2) ENGINEERING PREDICTION PROBLEM: AMMONIA NITROGEN CONCENTRATION (S_{NH}) PREDICTION

With the development of society and the increase of domestic and industrial wastewater discharge, water pollution has seriously affected people's living environment. Wastewater treatment has become one of the urgent problems to be solved nowadays. One key to wastewater treatment is nitrogen removal, but the ammonia nitrogen concentration (S_{NH}) is hard to measure in real time, which seriously affects the efficiency of wastewater treatment.

In this section, the issue of S_{NH} prediction is computed by SSAFNN. The 14 days data of rain weather is used for prediction, total 623 groups of data, where 459 groups for training and 164 groups for testing. The influent flow of rain weather is shown in Fig. 14. The data is acquired by the Benchmark Simulation Model No.1 (BSM1). BSM1 [29] was announced by European Cooperation in the Field of Science and Technology (COST). The benchmark development has been undertaken in Europe by working groups of COST Action 682 and 624 and continues under the umbrella of an IWA Task Group.

The input variables of FNN include the influent flow and the concentration of Dissolved Oxygen (So) at present, the concentration of influent S_{NH} one hour ago and effluent S_{NH} one hour ago, the output variable was the concentration of effluent S_{NH} at present. The training parameter is choosing the same of Mackey-glass time series prediction, see Table 10. The total training loop is set to 200.

Fig. 15 exhibits the good S_{NH} concentration prediction performance of SSAFNN with initial 4 and 50 nodes. Fig.16 exhibits the nodes change during the training process. The MSE comparisons are shown in Fig. 17 and Table 11 shows the detailed results. By analyzing with Fig. 15∼ Fig. 17 and Table 11, here are some comments:

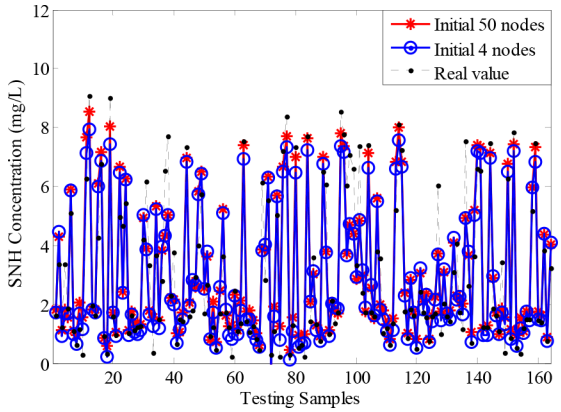


FIGURE 15. Results comparison of effluent S_{NH} prediction.

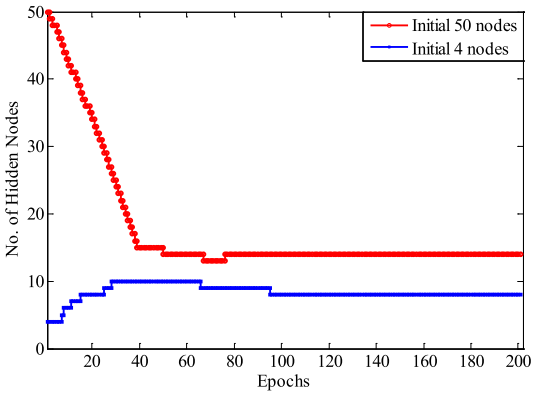


FIGURE 16. Nodes updating process of effluent S_{NH} prediction.

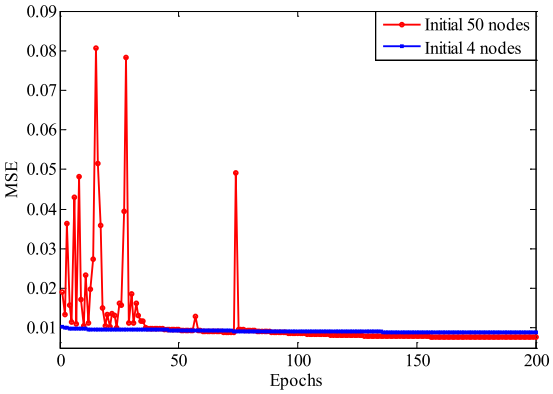


FIGURE 17. Training MSE of effluent S_{NH} prediction.

TABLE 11. Performance comparison for effluent S_{NH} prediction.

Algorithms	Average No. of Hidden nodes	Range of hidden nodes	Testing RMSE
4	8.6	7~10	0.0749
50	14.2	13~15	0.0732
Fixed 4	-	-	0.0818
Fixed 50	-	-	0.0970

(1) Whether starting with a small scale or a large scale, SSAFNN showing its good performance in solving S_{NH} prediction problem.

(2) The appropriate hidden node range for this problem is 7∼10 with initial 2 nodes and 13∼15 with initial 50 nodes. And we also find that SSAFNN cannot solve the SNH prediction problem initial with 3 or less hidden nodes.

(3) When the initial number of nodes is small, the initial value of MSE will be lower. This conclusion is the same with Mackey-Glass time series prediction. But we also find a contrary result: the MSE value will achieve a lower value at the end when initial with 50 nodes. This conclusion reflexes the difference character of Mackey-Glass time series and effluent S_{NH} prediction, and the proposed SSAFNN will have different effects when solving different problems.

VI. CONCLUSIONS

In this paper, a biological mechanism based self-adaptive algorithm for feedforward neural network structure is proposed. The adaptive neuron growing and pruning index are designed based on the concepts of biological neuron grow factor and neuron competition, respectively. The effectiveness of SSAFNN has been demonstrated by both the theoretical analysis and several benchmark tests of classification and prediction problems. The proposed SSAFNN is also applied in two engineering problems, bolt non-destructive testing and effluent ammonia nitrogen predicting. All the benchmark and engineering experiments show that the SSAFNN effectively optimized the network structure and had good performance. The improvements of SSAFNN can be summarized as follows:

(1) According to the idea of biological neural network, two different mechanisms are designed for neuron growing and pruning, respectively. These mechanisms effectively optimized the network structure, and improved the FNN performance in solving different problems.

(2) The proposed SSAFNN effectively improved the performance in solving complex problems without losing the efficiency in solving simple problems.

Notice that the difference of the growing and pruning mechanisms has led to the inconsistency of the appropriate range of hidden nodes, and the initialization of the neural network still affects the problem solving performance. Further researches along these directions are being expected.

REFERENCES

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," *Neurocomput., Found. Res.*, pp. 339–421, 1988.

[2] A. A. Zare and S. H. Zahiri, "Recognition of a real-time signer-independent static Farsi sign language based on Fourier coefficients amplitude," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 5, pp. 727–741, 2016.

[3] J. Wu, S. Wang, and F.-L. Chung, "Positive and negative fuzzy rule system, extreme learning machine and image classification," *Int. J. Mach. Learn. Cybern.*, vol. 2, no. 4, pp. 261–271, 2011.

[4] H. G. Han, L. D. Wang, and J. F. Qiao, "Efficient self-organizing multilayer neural network for nonlinear system modeling," *Neural Netw.*, vol. 43, no. 7, pp. 22–32, 2013.

[5] J. Farajzadeh, A. F. Fard, and S. Lotfi, "Modeling of monthly rainfall and runoff of Urmia lake basin using 'feed-forward neural network' and 'time series analysis' model," *Water Resour. Ind.*, vols. 7–8, pp. 38–48, Sep. 2014.

[6] X. Sun, S. Gong, G. Han, M. Wang, and A. Jin, "Pruning Elman neural network and its application in bolt defects classification," *Int. J. Mach. Learn. Cybern.*, pp. 1–16, Sep. 2018. doi: 10.1007/s13042-018-0871-0.
[7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[8] H.-G. Han and J.-F. Qiao, "Adaptive dissolved oxygen control based on dynamic structure neural network," *Appl. Soft Comput. J.*, vol. 11, no. 4, pp. 3812–3820, 2011.
[9] W.-C. Chen, L.-Y. Tseng, and C.-S. Wu, "A unified evolutionary training scheme for single and ensemble of feedforward neural network," *Neurocomputing*, vol. 143, no. 143, pp. 347–361, 2014.
[10] F. Li, J. Qiao, H. Han, and C. Yang, "A self-organizing cascade neural network with random weights for nonlinear system modeling," *Appl. Soft Comput.*, vol. 42, pp. 184–193, May 2016.
[11] B. Fritzke, "Growing grid—A self-organizing network with constant neighborhood range and adaptation strength," *Neural Process. Lett.*, vol. 2, no. 5, pp. 9–13, 1995.
[12] S. Wu and T. W. S. Chow, "Self-organizing and self-evolving neurons: A new neural network for optimization," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 385–396, Mar. 2007.
[13] F. J. Li, J. F. Qiao, and H. G. Han, "Incremental constructive extreme learning machine," *Control Theory Appl.*, vol. 31, no. 5, pp. 638–643, 2014.
[14] B. Hassibi and D. G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 5, 1993, pp. 164–171.
[15] X. Zeng, J. Shao, Y. Wang, and S. Zhong, "A sensitivity-based approach for pruning architecture of Madalines," *Neural Comput. Appl.*, vol. 18, no. 8, pp. 957–965, 2009.
[16] F. B. Jiang, Q. W. Dai, and L. Dong, "Nonlinear inversion of electrical resistivity imaging using pruning Bayesian neural networks," *Appl. Geophys.*, vol. 13, no. 2, pp. 267–278, 2016.
[17] M. M. Islam, M. A. Sattar, M. F. Amin, X. Yao, and K. Murase, "A new adaptive merging and growing algorithm for designing artificial neural networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 3, pp. 705–722, Jun. 2009.
[18] C.-F. Hsu, T.-T. Lee, C.-M. Lin, and B. K. Lee, "Design of adaptive growing-and-pruning neural control for LPCM drive system," in *Proc. Int. Multiconf. Eng. Comput. Scientists*, Mar. 2007.
[19] H.-G. Han and J.-F. Qiao, "A structure optimisation algorithm for feedforward neural network construction," *Neurocomputing*, vol. 99, nos. 2–3, pp. 347–357, 2013.
[20] J. Qiao, S. Li, H. Han, and D. Wang, "An improved algorithm for building self-organizing feedforward neural networks," *Neurocomputing*, vol. 5, no. 7, pp. 28–40, 2017.
[21] R. Levi-Montalcini, S. D. Skaper, T. R. Dal, and L. Petrelli, "Nerve growth factor: From neurotrophin to neurokine," *Trends Neurosci.*, vol. 19, no. 11, pp. 514–520, 1996.
[22] L. Aloe, E. Alleva, A. Böhm, and R. Levi-Montalcini, "Aggressive behavior induces release of nerve growth factor from mouse salivary gland into the bloodstream," *Proc. Nat. Acad. Sci. USA*, vol. 83, no. 16, pp. 6184–6187, 1986.
[23] D. D. M. O'Leary, "Development of connectional diversity and specificity in the mammalian brain by the pruning of collateral projections," *Current Opinion Neurobiol.*, vol. 2, no. 1, pp. 70–77, 1992.
[24] H. Nakamura and D. D. O'Leary, "Inaccuracies in initial growth and arborization of chick retinotectal axons followed by course corrections and axon remodeling to develop topographic order," *J. Neurosci.*, vol. 9, no. 11, pp. 3776–3795, 1989.
[25] A. H. Kramer, "Efficient parallel learning algorithms for neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 1, 1988, pp. 40–48.
[26] R. Zhang, Z.-B. Xu, D. Wang, and G.-B. Huang, "Global convergence of online BP training with dynamic learning rate," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 2, pp. 330–341, Feb. 2012.
[27] J.-F. Qiao, G. Han, and H.-G. Han, "Neural network on-line modeling and controlling method for multi-variable control of wastewater treatment processes," *Asian J. Control*, vol. 16, no. 4, pp. 1213–1223, 2014.
[28] D. Dua and E. K. Taniskidou, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, Tech. Rep., 2017. [Online]. Available: http://archive.ics.uci.edu/ml/citation_policy.html
[29] J. B. Copp, "The COST Simulation Benchmark-Description and Simulator Manual," in *Proc. Office Off. Publications Eur. Community*, Luxembourg City, Luxembourg, 2001, pp. 123–133.

GUANG HAN received the Ph.D. degree in pattern recognition and intelligent system from the Beijing University of Technology, in 2014.

He is currently a Lecturer with the School of Electrical and Electronics Engineering, Shijiazhuang Tiedao University. His current research interests include machine learning, artificial neural networks, and its engineering applications on process control and non-destructive testing.

QI CHENG received the B.S. degree from the School of Electrical and Electronic Engineering, Shijiazhuang Tiedao University, China, in 2018. He is currently pursuing the M.S. degree with the College of Automation, Institute of Intelligent Systems and Biological Information, Harbin Engineering University, Harbin, China.

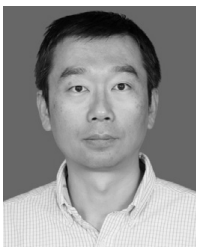
His current research interests include pattern identification, deep learning, and brain wave analysis.

XIAOYUN SUN received the Ph.D. degree in theory and new technology of electrical engineering from Xi'an Jiaotong University, China, in 2000.

She is currently a Professor and the Dean of the School of Electrical and Electronics Engineering, Shijiazhuang Tiedao University. Her current research interests include intelligent control and non-destructive testing.

LI LI received the B.S. degree from the School of Electrical and Electronic Engineering, Shijiazhuang Tiedao University, China, in 2017, where she is currently pursuing the M.S. degree.

Her current research interests include machine learning, generative adversarial networks, and its engineering applications on non-destructive testing.



WEIGUO DI received the master's degree from the Hebei University of Technology, China, in 2003. He is studying for a Ph.D. degree in civil engineering at Shijiazhuang Tiedao University.

He is currently an Associate Professor with the School of Electrical and Electronics Engineering, Shijiazhuang Tiedao University. His current research interests include neural networked control systems, electromagnetic acoustic transducer testing, and non-destructive testing.