IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 13, 2019, accepted May 15, 2019, date of publication May 17, 2019, date of current version May 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2917526

A Novel Neural Network Classifier Using Beetle Antennae Search Algorithm for Pattern Classification

QING WU[1], ZHEPING MA[1], GANG XU[1], SHUAI LI[2], (Senior Member, IEEE), AND DECHAO CHEN[1], (Member, IEEE)

[1]Department of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China
[2]Department of Computing, The Hong Kong Polytechnic University, Hong Kong

Corresponding authors: Shuai Li (shuaili@polyu.edu.hk) and Dechao Chen (chdchao@hdu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61401385 and Grant 61702146, in part by the Hong Kong Research Grants Council Early Career Scheme under Grant 25214015, in part by the Departmental General Research Fund of The Hong Kong Polytechnic University under Grant G.61.37.UA7L and in part by the PolyU Central Research Grant under Grant G-YBMU.

ABSTRACT Traditional training algorithms in artificial neural networks (ANNs) show some inherent weaknesses, such as the possibility of falling into local optimum, slow learning speed, and the inability to determine the optimal neuronal structure. To remedy the deficiencies of traditional neural networks, this paper proposes a novel neural network classifier (NNC) using the beetle antennae search (BAS) algorithm, termed BASNNC. The BAS algorithm is explored to optimize the weights of the NNC. The network of the proposed BASNNC adopts three-layer structure, including an input layer, a hidden layer, and an output layer. Quite differing from the traditional training algorithm using a principle of gradient descent, the weights between the hidden and output layers are optimized by the BAS algorithm, which effectively improves the computational speed of the classifier. The numerical studies, applications to pattern classification and comparisons with an error back-propagation neural network model, show that the proposed BASNNC has faster computational speed and higher classification accuracy.

INDEX TERMS Beetle antennae search (BAS) algorithm, pattern classification, artificial neural networks (ANNs), neural network classifier (NNC), training algorithms.

I. INTRODUCTION

The complexity of the system conflicts with the accuracy required. By simulating human learning and adaptive capabilities, people have proposed the idea of intelligent control. Fuzzy logic [1], expert system [2], and neural network [3], [4] are three typical control methods. Usually, expert systems are based on expert experience and are not based on operational data generated by industrial processes. It is difficult for domain experts to grasp the inaccuracies and uncertainties of general complex systems, which makes it very difficult to establish expert systems. Fuzzy logic and neural networks, as two typical intelligent control methods, have their own advantages and disadvantages.

Artificial neural networks (ANNs) are an intelligent system that people imitate the information processing functions of the human brain nervous system. At present, the field of artificial intelligence research is very focused on the fusion of intelligent identification methods. In view of the many excellent capabilities of neural network technology such as knowledge storage and uncertain information processing, the application of neural networks in the fields of pattern recognition [5], [6], signal processing [7], intelligent control [8]–[11] and intelligent optimization [12], [13] can make up for the shortcomings and deficiencies in the original technology field. Therefore, the application of neural networks in many different fields of engineering and science has become a research focus. ANNs have successfully solved many practical problems that are difficult to solve with modern computers, and have shown good intelligence characteristics. Note that ANNs have become powerful tools for pattern classification. Various ANN models are universal function approximators [14]. They can adjust themselves to the data without any explicit specification of functionality, which is called data-driven adaptive capabilities.

The associate editor coordinating the review of this manuscript and approving it for publication was Yan-Jun Liu.

Common-used classification methods mainly include naive bayes (NB), decision trees, support vector machine (SVM), k-nearest neighbors (k-NN), ANNs and so on [15]–[17]. We do not need to learn parameters in NB like what we do in ANNs. However, NB assumes that attribute values are independent of each other given the class. It does not apply to data with a large number of attributes or a large correlation between attributes, but ANNs do not have this problem. The advantage of decision trees over ANNs is that they can handle non-numeric data. ANNs are better at processing continuous data than decision trees. The SVM classifier has no general solution to nonlinear problems and must be carefully chosen to handle the kernel function. And there are certain difficulties for multi-label classification problems. k-NN is computationally intensive. For each sample to be classified, k-NN calculates the distance to all known samples in order to find its k nearest neighbors. ANNs classifies the samples to be classified directly into the pre-trained model. There's no such thing as a free lunch. In other words, there is no algorithm that solves all problems perfectly. They are subject to many factors, such as the size and structure of the data set. The selected ANNs can achieve better results than the other classifiers when it is suitable for the corresponding problem.

ANNs are nonlinear models [18]–[20]. Applications of neural networks for classification problems include handwriting recognition [21], speech recognition [22], product inspection [23], fault detection [24], medical diagnosis [25], etc. For example, deep neural network (DNN) [26], [27], convolutional neural network (CNN) [28] and recurrent neural network (RNN) [29]–[31] are all the research hotspots of ANN. Neural networks based on error back-propagation (BP) training algorithm are one of the most widely applied and matured ANN models [32], [33]. However, the training speed of BP algorithm is much slower than people expect. The key factor accounting for this phenomenon is the training of neural networks. Most of them use the gradient descent algorithm (GD) [34], thus the training speed of this algorithm is limited. Besides, network structure influences the network performance significantly. Therefore, to determine the optimal weights and structure of ANNs is a challenging issue.

In order to overcome the above-mentioned weakness of traditional neural networks and improve the performance of neural networks, we propose and investigate a novel multi-input beetle antennae search neural network classifier (BASNNC), which differing from the algorithm in the BP iterative training process. It uses the BAS training algorithm to optimize the weights, which remedies the weaknesses of the traditional gradient-based of the BP algorithm. BAS is a single-body intelligent search algorithm without complex gradient solution. Since there is only one individual when searching, the search speed is fast. In addition to the optimization of weights, according to the research of Zhang and Tan [35], the structure determination is likewise important for neural networks. Note that the structure corresponds to the number of hidden neurons. Therefore, the number of hidden layer neurons is skillfully determined to construct a network structure for the proposed BASNNC in this paper. Numerical results further confirm the efficiency and validity of the proposed BASNNC. The main work of this paper is based on the improvement and application of bio-heuristic algorithm. As an expansive application, we apply it to neural network optimization and pattern classification.

A. RELATED WORK

There are many applications of ANN, and many people have studied it. For example, Liu *et al.* [3] proposed an adaptive neural network (NN) control scheme for a quarter-car model, which uses neural networks to approximate unknown mass of car-body. In the data categorization method, ANNs have stronger learning ability than the traditional statistical classification program, which greatly elevate the exact ratio. Neural networks based on the BP algorithm are widely used in ANNs. The traditional gradient-based BP neural network learning process includes two processes: the forward propagation of information and the back propagation of errors. The training is repeated, and the changes of the network weights and deviations are continuously calculated in the direction of the gradient of the relative error function gradient. The target is gradually approached. However, traditional gradient-based BP neural networks have some inherent shortcomings, for instance, the convergence speed is slow, easy to fall into local optimum, etc.

The common improvement methods used by researchers include modifying the network structure, trying various activation functions and improving weight-definite method. For example, Weight-and-structure-determination (WASD) algorithms, the method of using linearly independent or orthogonal polynomials as activation functions and some other methods have been proposed by Zhang *et al.* [36]–[38]. The heuristic random search algorithm has a strong global search ability due to the relative gradient descent method. Therefore, some researchers have combined heuristic random search algorithms with neural networks. For example, Han *et al.* [39] combined APSO algorithm with neural network and APSO to optimize network parameters. Salcedo-Sanz and Yao [40] mixed Hopfield neural network and GA algorithm to tackle the terminal assignment (TA) problem. Zhang *et al.* [41] GA is applied to the hidden layer structure and parameter optimization of neural networks. Ren *et al.* [42] proposed a Back Propagation neural network based on Particle Swam Optimization that have studied how to select input parameters carefully to achieve desired results. Beyond that, there are improved algorithms based on numerical optimization, such as the conjugate gradient method [43], least squares method [44] and so on. The BP algorithm with simulated annealing algorithm [45], genetic algorithm [46], and ant colony algorithm [47] are hybrid algorithms. It also includes an additional momentum algorithm, adaptive learning rate method [48] and other improved algorithms.

B. ORGANIZATION AND CONTRIBUTIONS

The remaining structure of this article is as follows. In Section II, the preliminaries of the BASNNC are presented together with the theoretical basis shown. In Section III, we introduce the BAS algorithm and the detailed process of determining the number of single hidden layer neurons for our neural network model. The numerical studies, applications and comprehensive comparisons are shown in Section IV to testify the effectiveness of the proposed BASNNC approach. Section V concludes the paper with final remarks. Before ending this subsection, the main contributions of this paper are summarized as follows.

- In this paper, a novel BASNNC is proposed and investigated for the first time, which is different from the traditional BP neural networks;
- The BAS training algorithm is used to optimize the weights. To construct the network structure of proposed BASNNC, this step process of determining the number of single hidden layer neurons is performed;
- Numerical studies, applications and comprehensive comparisons are conducted to substantiate the efficacy and superiority of the proposed BASNNC approach to pattern classification.

II. PRELIMINARIES

As the preliminaries, the neural network model is first constructed in this section. The proposed BASNNC adopts a three-layer structure (i.e., the input layer, the hidden layer and the output layer), the structure is shown in Figure 1. Each neuron in the input layer is connected to all neurons in the hidden layer; Each neuron in the hidden layer is connected to all neurons in the output layer.

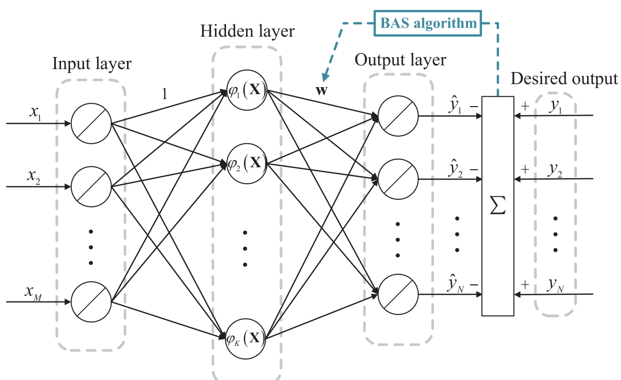


FIGURE 1. Model structure of the proposed BASNNC.

Assuming that the input layer has M neurons, the input vector is $\mathbf{X} = [x_1, x_2, \ldots, x_M]^{\mathrm{T}}$, where the superscript T denotes the transposition of the matrix or vector, M is determined by the number of attributes. Note that we normalize raw data first. The number of hidden layer neurons is K. The activation function of a hidden layer is

$$\begin{cases} \varphi_1(\mathbf{X}) = 1 \\ \varphi_2(\mathbf{X}) = x_1 + x_2 + x_3 + \cdots + x_M \\ \vdots \\ \varphi_K(\mathbf{X}) = x_1^{K-1} + x_2^{K-1} + x_3^{K-1} + \cdots + x_M^{K-1}. \end{cases} \tag{1}$$

Remark 1: The sigmoid function is a common activation function that has the disadvantage of being too supersaturated to lose gradients. This has a great influence on the neural network using the gradient descent algorithm (GD). However, the proposed neural network training uses a heuristic search algorithm, and there is no gradient problem. So when both the neural network using GD and our proposed neural network use sigmoid function, our model has advantages. The tanh function also has soft saturation, which causes the gradient to disappear. It can also be applied to our neural network to avoid the problem of gradient disappearance. There are also commonly used activation functions such as ReLU, Leaky ReLU function and PReLU function, which can also be applied in our proposed network model.

Thereby the output of a hidden layer is

$$\mathbf{A} = [\varphi_1(\mathbf{X}), \varphi_2(\mathbf{X}), \ldots, \varphi_K(\mathbf{X})]^{\mathrm{T}}. \tag{2}$$

There are N neurons in the output layer and the output vector is $\hat{\mathbf{y}} = \left[\hat{y}_1, \hat{y}_2, \hat{y}_3, \ldots, \hat{y}_N\right]^{\mathrm{T}}$, and the number of neurons in the output layer is determined by the number of sample types.

Remark 2: In order to simplify the established neural network and reduce the computational complexity, all neuronal thresholds in the BAS neural network classification model are set to be 0, and the weights between input and hidden layers are set to be 1. It has no influence on the current research, and is general. In future research, we will continue to explore non-zero and one.

And the connection weights between hidden and output layers are

$$\mathbf{w} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{K1} & w_{K2} & \cdots & w_{KN} \end{bmatrix} \in \mathbb{R}^{K \times N}. \tag{3}$$

The weights are optimized by the BAS training algorithm. We use error between the predicted and desired values as an optimization criterion. Note that the detail of BAS algorithm will be described in Section III. Then, the output of the n-th neuron in the network output layer is

$$\hat{y}_n = \sum_{i=1}^{K} w_{in}\varphi_i(\mathbf{X}). \tag{4}$$

Consequently, the output vector can be written as follows:

$$\hat{\mathbf{y}} = \left[\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n, \ldots, \hat{y}_N\right]^{\mathrm{T}}$$
$$= \left[\sum_{i=1}^{K} w_{i1}\varphi_i(\mathbf{X}), \sum_{i=1}^{K} w_{i2}\varphi_i(\mathbf{X}), \ldots,\right.$$

$$\sum_{i=1}^{K} w_{in}\varphi_i(\mathbf{X}), \ldots, \sum_{i=1}^{K} w_{iN}\varphi_i(\mathbf{X})\Bigg]^{\mathrm{T}}$$
$$= \mathbf{w}^{\mathrm{T}}\mathbf{A}. \tag{5}$$

III. DESIGN OF BASNNC AND THEORETICAL ANALYSES

A. NEURAL NETWORK CLASSIFIER DESIGN

To construct a complete BASNNC, we obtain the structure of the hidden layer (or the best structure of BASNNC), and then obtain the connection weight between the hidden layer and the output layer after determining the hidden layer structure. The BAS algorithm is applied to optimize the connection weights between hidden and output layers, replacing the traditional low-convergence speed and high computational complexity gradient back propagation.

The number of hidden-layer neurons can greatly affect the overall performance of neural networks. In particular, when the number of hidden layer neurons is too small, the learning and approximation ability of neural networks are insufficient, the expected training accuracy may not be achieved. Excessive hidden layer neurons may produce overfitting phenomena and higher computational complexity [35]. On the basis of the above analysis, it is meaningful and important to obtain the optimal number of hidden layer neurons (that is, the optimal structure of neural networks).

Algorithm 1 Determination of Hidden-Layer Neurons

```
Input: dataset;
Output: K;
Initialize the number of hidden layer neurons K;
Normalize the dataset attributes;
for p = 1; p ≤ 500; p++ do
    for q = 1; q ≤ 10; q++ do
        Divide the dataset into training set and test set;
        Calculate A_tra by K, Equation (1) and (2);
        Optimize w_tra by the BAS algorithm;
        Calculate A_tes by K, Equation (1) and (2);
        Calculate
        ŷ_tes = [ŷ_1, ŷ_2, ŷ_3, ···, ŷ_N]^T = w^T A_tes;
        Calculate ε_tes;
    end
    Calculate ε_avg for 10 cycles;
    K ← K + 1;
end
Find ε_min in 500 cycles;
Return K corresponds to ε_min;
```

In order to clearly describe the determination of hidden-layer neurons, the basic steps of the process are presented in the following Algorithm 1. According to the algorithm, we obtained the appropriate number of neuron nodes through a large number of experiments. First of all, the variables involved are explained as follows: $\mathbf{A}_{\mathrm{tra}}$ represents the output of hidden layer of the training set, $\mathbf{w}_{\mathrm{tra}}$ corresponds to the connection weights between hidden and output layers of the training set, the output of hidden layer of the test set is denoted by $\mathbf{A}_{\mathrm{tes}}$, $\hat{\mathbf{y}}_{\mathrm{tes}}$ is the output vector of the test set, $\varepsilon_{\mathrm{tes}}$ and $\varepsilon_{\mathrm{avg}}$ are the test error and the average test error, respectively. $\varepsilon_{\min}$ can be described as the minimum value of the average test error.

For the connection weights, traditional approach is based on negative gradients, which is obtained through iterative learning. However, using the iterative method to train neural networks needs more computational time [49]. For the sake of time-consuming problem, we use the BAS algorithm to determine the connection weights.

Note that the algorithm is a heuristic algorithm that is suitable for multi-objective function optimization, which was introduced and proposed by Jiang and Li [50], and then improve it [51], [52]. The BAS algorithm does not require the specific form of the function and the gradient information. This method mimics the detecting and searching behaviors of beetles. It can automate the optimization process and the optimal speed is significant. In this paper, the BAS algorithm is modified to find the optimal connection weights between hidden and output layers.

On the basis of Equation (5), the objective function to be optimized is

$$\epsilon_n = \sqrt{\frac{\sum_{i=1}^{S}\left(\hat{y}_{ni} - y_{ni}\right)^2}{S}} = \frac{\left\|\mathbf{w}_n^{\mathrm{T}}\mathbf{A} - \mathbf{y}_n\right\|_2}{\sqrt{S}}, \tag{6}$$

where S is the number of training samples, respectively. $\hat{y}_{ni}$ represents the predicted value of the n-th output of the i-th sample, the true value of the n-th output of the i-th sample is denoted by y_{ni}, $\mathbf{y}_n$ stands for the true value of the n-th output of all samples, where $\mathbf{w}_n$ is the n-th column of the connection weights matrix $\mathbf{w}$, as we already mentioned, $\mathbf{A}$ is the output of the hidden layer, ϵ_n indicates the deviation rate of the n-th output of all samples.

For the sake of illustration, we define the connection weights value iteration t times for the beetle's position $\mathbf{p}_n^t$ at time t, where $\mathbf{p}_n^t$ represents $\mathbf{w}_n$, and by Equation (6), the fitness function is defined as

$$f\left(\mathbf{p}_n^t\right) = \sqrt{\frac{\sum_{i=1}^{S}\left(\sum_{j=1}^{K}\left(\mathbf{p}_n^t\right)_j \varphi_j\left(\mathbf{X}_i\right) - y_{ni}\right)^2}{S}}$$
$$= \frac{\left\|\left(\mathbf{p}_n^t\right)^{\mathrm{T}}\mathbf{A} - \mathbf{y}_n\right\|_2}{\sqrt{S}}. \tag{7}$$

The random direction of the beetle search is

$$\mathbf{b} = \frac{\mathrm{rands}\,(K, 1)}{\|\mathrm{rands}\,(K, 1)\|_2}, \tag{8}$$

where rands $(\cdot)$ indicates a random function that produces a K-dimensional column vector, K is equal to the number of hidden layer neurons. Depending on the direction of the beetle search and the length of the antennae, calculate the position of the left and right antennae of beetle at time t:

$$\begin{aligned}\mathbf{p}_r &= \mathbf{p}_n^t + d^t\mathbf{b},\\ \mathbf{p}_l &= \mathbf{p}_n^t - d^t\mathbf{b},\end{aligned} \tag{9}$$

where

$$\begin{aligned}d^t &= \delta^t c,\\ \delta^t &= \delta^{t-1}\eta,\end{aligned} \tag{10}$$

where d^t is the antenna length of the beetle at time t, it should be large enough to cover the appropriate search area. δ^t stands for the step size at time t, c and η are the basic parameters of BAS algorithm. After debugging parameter, δ^0 stands for the initial step size and initialize it to 0.5, η and c are initialized to 1 and 0.5, respectively. The beetle's location is updated by

$$\mathbf{p}_n^t = \mathbf{p}_n^{t-1} + \delta^t\mathbf{b}\,\mathrm{sign}\left(f\left(\mathbf{p}_r\right) - f\left(\mathbf{p}_l\right)\right), \tag{11}$$

where $\mathbf{p}_n^{t-1}$ indicates the beetle's position at the previous moment of time t, sign $(\cdot)$ is a symbolic function:

$$\mathrm{sign}\left(x\right) = \begin{cases}1, & x > 0,\\ 0, & x = 0,\\ -1, & x < 0.\end{cases} \tag{12}$$

Algorithm 2 corresponds the BAS algorithm described in this subsection. It demonstrates the improvement based on the original BAS algorithm which is adopted in the research to design a feasible approach to solve the weight optimization problem. And introduce simply some variables of Algorithm 2. The label of instance is denoted by $\mathbf{y}$, ε_{tra} represents training error, d^0 indicates the sensing diameter at first, $\mathbf{p}_n^0 = \left[\left(\mathbf{p}_n^0\right)_1, \left(\mathbf{p}_n^0\right)_2, \cdots, \left(\mathbf{p}_n^0\right)_K\right]^{\mathrm{T}}$ can express the beginning value of weights, f_{bes} means the best f, $\mathbf{p}_{\text{bes}}$ is equivalent to the best position of beetle. The remainder of variables as described above.

Algorithm 2 BAS Algorithm for Optimal Weights Searching

Input: K, $\mathbf{y}$, M, $\mathbf{A}$;
Output: ε_{tra}, $\mathbf{w}$;
for $n = 1$ to N **do**
  Initialize c, η, δ^0, d^0, $\mathbf{p}_n^0 = \left[\left(\mathbf{p}_n^0\right)_1, \left(\mathbf{p}_n^0\right)_2, \cdots, \left(\mathbf{p}_n^0\right)_K\right]^{\mathrm{T}}$;
  Select the n-th row of data from $\mathbf{y}$ as $\mathbf{y}_n$;
  while $(t < T_{\max})$ **do**
    Generate $\mathbf{b}$ according to Equation (8);
    Calculate $\mathbf{p}_r$ and $\mathbf{p}_l$ according to Equation (9);
    Update $\mathbf{p}_n^t$ according to Equation (11);
    if $f\left(\mathbf{p}_n^t\right) < f_{bes}$ **then**
      $f_{\text{bes}} = f\left(\mathbf{p}_n^t\right)$;
      $\mathbf{p}_{\text{bes}} = \mathbf{p}_n^t$;
    end
    Update d^t and δ^t according to Equation (10);
  end
  $\mathbf{w}_n = \mathbf{p}_{\text{bes}}$;
end
Calculate $\hat{\mathbf{y}} = \left[\hat{y}_1, \hat{y}_2, \hat{y}_3, \cdots, \hat{y}_N\right]^{\mathrm{T}} = \mathbf{w}^{\mathrm{T}}\mathbf{A}$;
Find out the maximum value of $\hat{\mathbf{y}}$, and set it to be 1 with the rest being 0. Then a new vector $\hat{\mathbf{y}}$ consisting of 0 and 1 can be obtained;
Calculate the number of samples different from $\hat{\mathbf{y}}$ and $\mathbf{y}$, and divide the total number of samples to get ε_{tra};
Return ε_{tra}, $\mathbf{w}$;

B. THEORETICAL ANALYSES

In this section, we analyze the convergence of the BAS algorithm [51]. For the sake of explanation, define $\mathbf{p}_{bes}^t$ as the position of the optimal solution of the beetle at time t and f_{bes}^t is the corresponding optimal solution. $\mathbf{p}^*$ represents the position of the theoretical optimal solution of the problem.

Lemma 1: If the proposed BASNNC with the output vector $\hat{y}$ depicted in Equation (5) using the BAS algorithm, then f_{bes}^t is not increased.

Proof: According to the solution method of f_{bes}^t in Algorithm 2: at time t, if $f\left(\mathbf{p}^t\right) < f_{\text{bes}}$, then $f_{bes}^t = f\left(\mathbf{p}^t\right)$. The optimal solution is the minimum value, then the initial value of f_{bes}^t is set to be large. It is thus proved that the BAS algorithm guarantees that f_{bes}^t does not increase.

Theorem 1: Provided that the parameters are set correctly, the BAS algorithm for the proposed BASNNC with the output vector $\hat{y}$ depicted in Equation (5) converges with probability 1.

Proof: Assume that the parameters of the BAS algorithm are set correctly. Let P_t denote the probability that $\mathbf{p}^t$ is not on $\mathbf{p}^*$ at time t, where $0 \leqslant P_t < 1$. Then, $P\left(p_{bes}^t = p*\right) \geqslant 1 - P_0P_1\cdots P_t$. Therefore, $\lim\limits_{t\to+\infty}\left(1 - P_0P_1\cdots P_t\right) = 1 - \lim\limits_{t\to+\infty} P_0P_1\cdots P_t = 1$. Note that $P\left(p_{bes}^t = p*\right) \leqslant 1$. According to the squeeze theorem, $\lim\limits_{t\to+\infty} P\left(p_{bes}^t = p*\right) = 1$. The proof is thus completed. □

By this point, the convergence of the BAS algorithm is proved, and the effectiveness of the algorithm is proved theoretically.

IV. NUMERICAL STUDIES, APPLICATIONS AND COMPARISONS

Two numerical examples running in the Matlab R2016a environment to prove the validity of BASNNC. First of all, two multivariable objective functions are selected to verify the effectiveness of the BAS algorithm in finding the optimal solution. Through this numerical experiment, it can be proved that when the variable to be optimized is replaced by the weight, it is also capable of finding the optimal solution of the weight. Afterwards, nine UCI classification sample datasets are selected for the proposed BASNNC classification experiments (Table 1). In addition, with the intention of further demonstrating the superiority of BASNNC, comprehensive comparisons with traditional BP neural network model under the same conditions are presented.

TABLE 1. Features of different real-world classification datasets.

Dataset	Number of attributes	Number of classes	Number of instances
Iris	4	3	150
Zoo	16	7	101
Breast Cancer Wisconsin (BCW)	9	2	699
Banknote Authentication (BA)	4	2	1372
Blood Transfusion Service Center (BTSC)	4	2	748
Cryotherapy	6	2	90
Haberman's Survival (HS)	3	2	306
Hayes-Roth	4	3	132
Indian Liver Patient Dataset (ILPD)	10	2	583

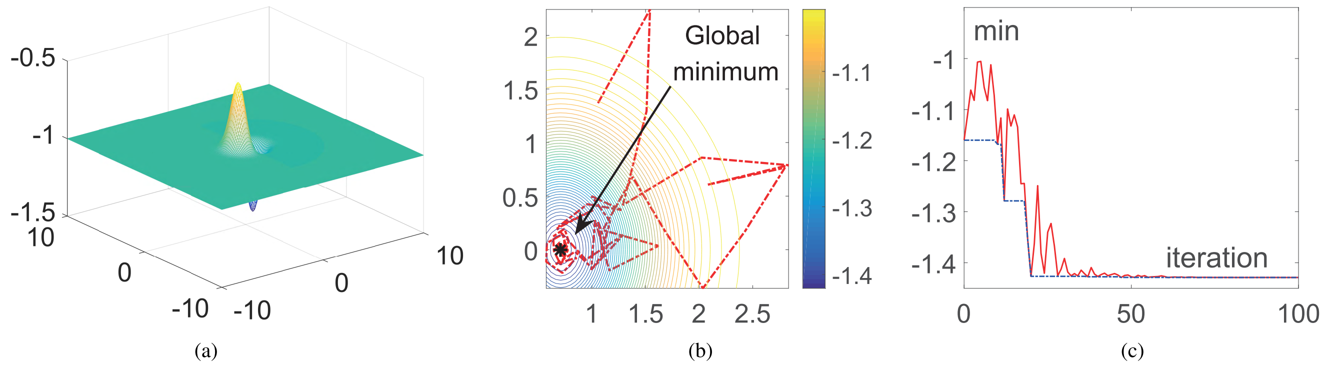


FIGURE 2. **The BAS algorithm to search the global optimum of the function (13) through 100 iteration steps. (a) It's a three-dimensional display. (b) The searching trajectory of the BAS algorithm. (c) Convergence of the minimum value along iteration step *t*.**

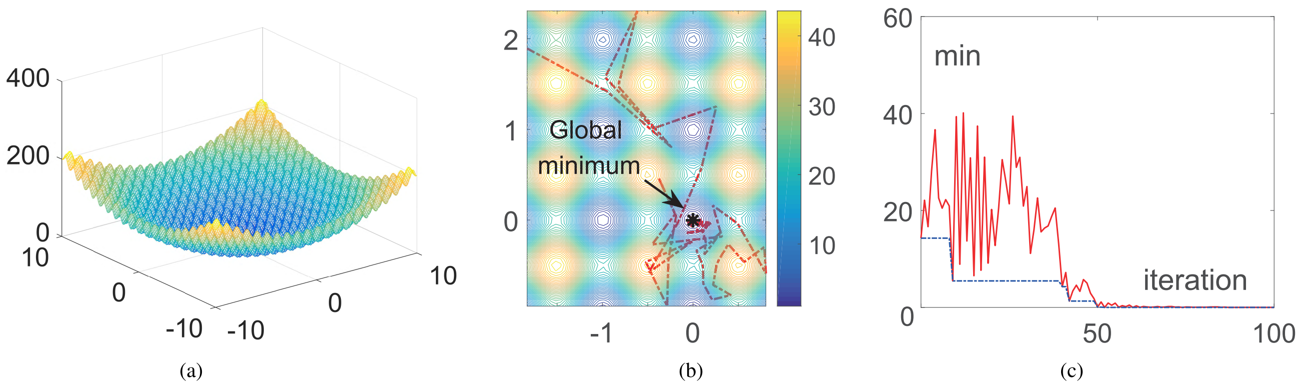


FIGURE 3. **The BAS algorithm to search the global optimum of the function (14) through 100 iteration steps. (a) It's a three-dimensional display. (b) The searching trajectory of the BAS algorithm. (c) Convergence of the minimum value along iteration step *t*.**

A. NUMERICAL VERIFICATION OF BAS ALGORITHM

In this subsection, two test functions are selected to validate the efficacy of BAS algorithm. For ease of presentation, we choose two-variable (i.e., $\mathbf{x} = [x_1, x_2]$) functions as test functions.

First, consider the following objective function:

$$F(\mathbf{x}) = -x_1 e^{-x_1^2 - x_2^2} - 1. \tag{13}$$

Theoretically, objective function (13) has a global minimum point. When $\mathbf{x}^* = \left[\sqrt{1/2}, 0\right]$, $F(\mathbf{x})$ reaches a global minimum $F^* = -\sqrt{1/2}e^{-\frac{1}{2}} - 1 \approx -1.4289$. Figure 2(a) is a three-dimensional display of function (13). The performance of the BAS algorithm is shown in Figure 2(b) and Figure 2(c). Along the time step t from 0 to 100, under the parameter configuration, the step size δ is updated according to rule (10) with initialization $\delta^0 = 2$, the parameters of η and c are initialized to 0.95. As can be observed from Figure 2, the BAS algorithm can find the global minimum of the function (13). Numerically, the solution of function (13) $F_{\text{bes}} = -1.4289$ approximates theoretically at the corresponding point $\mathbf{x}_{\text{bes}} = [0.7066, -0.0013]$.

We also consider the Rastrigin function:

$$F(\mathbf{x}) = \left(x_1^2 - 10\cos(2\pi x_1) + 10\right) + \left(x_2^2 - 10\cos(2\pi x_2) + 10\right). \quad (14)$$

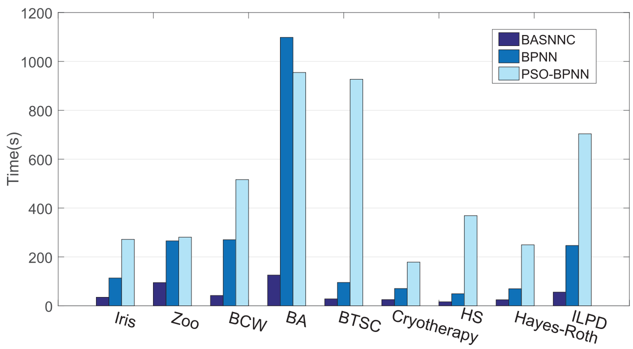


FIGURE 4. Histogram of the run time. The ordinate is Time. The abscissa is the dataset.

Figure IV is a three-dimensional rendering of the Rastrigin function. It can be seen from the Figure IV that the function (14) has multiple local minimums, but it has a global minimum $F^* = 0$ at $\mathbf{x}^* = [0, 0]$. Adopting the same time step t and parameter configuration as the function (13), the simulation results of the BAS algorithm as shown in Figure IV and Figure IV are obtained. Figure IV indicates a search trajectory of a beetle which seeks a global optimum. Figure IV shows the convergence of the minimum value along iteration step t. The optimization result of the function (14) by the BAS algorithm is when $\mathbf{x}_{\text{bes}} = [-0.0017, -0.0012]$, the optimal solution of the function is $F_{\text{bes}} = 8.8670\text{e-}04 \approx 0$. The above two example function (13) and function (14) have demonstrated that the BAS algorithm is effective for solving multi-objective function problems by visualization results and numerical solutions.

B. APPLICATION TO PATTERN CLASSIFICATION

This subsection verifies the classification performance of the proposed BASNNC by numerical experiments. For supervised learning situations, the effectiveness of the classifier can be assessed by classification accuracy. We use several published datasets in the standard UCI database for experiments (see Table 1). Iris Plants Database contains 3 classes of 150 instances each, where each class refers to a type of iris plant. A simple database of Zoo is artificial and contains 7 types of animals. Breast Cancer Wisconsin (BCW) was obtained from the University of Wisconsin Hospitals, Madison from Wolberg and Mangasarian [53]. Banknote Authentication (BA) was extracted from images that were taken from genuine and forged banknote-like specimens. Blood Transfusion Service Center (BTSC) adopted the donor database of Blood Transfusion Service Center in Hsin-Chu City in Taiwan. Cryotherapy dataset contains information about wart treatment results of 90 patients using cryotherapy. Haberman's Survival (HS) Dataset involves cases from study conducted on the survival of patients who had undergone surgery for breast cancer. Topic of Hayes-Roth is human subjects study. Indian Liver Patient Dataset (ILPD) contains 416 liver patient records and 167 non liver patient records. For convenience, we use above abbreviations. These datasets all have a known number of categories and the category to which each sample belongs.

In the experiment, we divide data samples into three parts by category, two for training set and one for the test set. For example, Breast Cancer Wisconsin dataset has two categories, the first category has 458 samples and the second category has 241 samples. Divide the first class of 458 samples into three parts and take two of them as training set 1, one for test set 1. Similarly, the second class of 241 samples are divided into three parts, two of which are used for training set 2, and one for test set 2. The test set consists of test set 1 and 2. Similarly, the training set is achieved by combining training set 1 and 2. In the end, each class is evenly distributed in both the training set and the test set. To avoid the fact that a certain class is only distributed in the test set, and there is no such class in the training set, thus affecting the classification. Note that the above method of dividing data set with regular ratio is commonly used in neural networks [54], [55]. In the previous section we mentioned the process of determining the hidden layer neurons. Here we use the datasets in Table 1 and write program code according to the Algorithm 1 of determination of hidden-layer neurons. Table 2 shows the experimental results, including the number of suitable hidden layer neurons corresponding to each dataset and the accuracy obtained with the number of neurons. After the number of hidden neurons was determined, the corresponding number of hidden neurons was placed in BASNNC. At this point, our network structure is determined.

TABLE 2. Classification results of Determination of hidden-layer neurons include the number of suitable hidden-layer neurons corresponding to each dataset and the accuracy obtained with the number of neurons.

Dataset	Number of neurons	Average test accuracy (%)
Iris	286	94.9990
Zoo	476	71.0260
BCW	254	97.5831
BA	508	88.8913
BTSC	140	76.3274
Cryotherapy	439	81.6667
HS	44	74.8039
Hayes-Roth	73	81.3636
ILPD	509	72.0725

TABLE 3. Classification results of BASNNC for datasets.

Dataset	Training accuracy (%)	Test accuracy (%)	Run time (s)	Run time per step/10000 (s)
Iris	93.0841	91.7362	34.8913	3.49e-5
Zoo	64.0448	62.8728	95.0023	9.50e-5
BCW	96.7721	96.6871	42.1671	4.22e-5
BA	88.5284	87.9291	125.7478	1.26e-6
BTSC	76.2960	76.0782	28.2534	2.83e-5
Cryotherapy	80.1667	73.9667	25.3829	2.54e-5
HS	74.3382	72.7549	16.3268	1.63e-5
Hayes-Roth	78.1705	76.5909	24.6370	2.46e-5
ILPD	71.3834	71.0207	55.9830	5.60e-5

The program of BASNNC is, written according to Algorithm 2. The corresponding dataset is placed in BASNNC. We repeat 100 times, and get the corresponding training accuracy, test accuracy, running time and time required for iterating once. Note that other classifiers via the WASD algorithm in [38], [56], [57] is based on the traditional matrix pseudo-inverse (PI) resolution. In contrast, the proposed BASNNC based on the BAS optimization strategy can avoid solving for matrix PI, and thus it costs considerably lower computation burden to solve the problem than the PI based classifiers in other studies. Matlab simulation results are shown in Table 3. As can be seen from the chart, forasmuch as our classification accuracy which includes training accuracy and test accuracy exceeds 60% for all datasets with two or more categories. Training accuracy and test accuracy are similar and there is almost no overfitting. Each dataset has a different run time per step and may be affected by the number and type of samples in the dataset itself. In conclusion, BASNNC is effective indeed.

C. COMPARISON WITH OTHER ALGORITHMS

We compare BASNNC with traditional BP neural network (BPNN) model and PSO-BPNN. Table 4 and Table 5 show the training accuracy and test accuracy of BASNNC and other methods for classifying datasets. Note that the accuracy in the table refers to the average training accuracy and average test accuracy of 100 experiments.

TABLE 4. Comparison of the training accuracy between the proposed BASNNC and other methods.

Dataset	Training accuracy (%)		
	BASNNC	BPNN	PSO-BPNN
Iris	93.0841	92.5743	98.0400
Zoo	64.0448	58.3484	99.9692
BCW	96.7721	97.2002	64.9123
BA	88.5284	88.6666	55.5191
BTSC	76.2960	76.3120	78.2952
Cryotherapy	80.1667	80.4333	98.3333
HS	74.3382	74.2843	76.5490
Hayes-Roth	78.1705	68.1932	94.7159
ILPD	71.3834	71.1528	72.2098
Average rank	1.89	2.33	1.44

TABLE 5. Comparison of the test accuracy between the proposed BASNNC and other methods.

Dataset	Test accuracy (%)		
	BASNNC	BPNN	PSO-BPNN
Iris	91.7362	91.5565	92.0000
Zoo	62.8728	58.9979	96.0556
BCW	96.6871	96.9684	65.1982
BA	87.9766	88.1100	55.5799
BTSC	76.0782	76.0221	77.5200
Cryotherapy	73.9667	75.2333	84.3667
HS	72.7549	72.8431	72.7255
Hayes-Roth	76.5909	65.7273	67.2727
ILPD	71.0207	70.6788	67.1399
Average rank	1.89	2.11	2

In Table 4, PSO-BPNN ranked first for the average ranking of the nine datasets, and in Table 5, our method ranked first. Moreover, the training accuracy rate of the PSO-BPNN method differs greatly from the test accuracy. For example, the training accuracy of Hayes-Roth differs from the test accuracy by 27%. This shows that PSO-BPNN is easier to overfit. In Table 4, for five datasets, the training accuracy of our method is much higher than that of traditional BPNN model. The training accuracy of four datasets is lower than BP, however, the difference is below 0.5%. Table 5 is show that there are five datasets where the accuracy of BASNNC are higher than that of BP, and the highest difference is about 11%. The test accuracy of four datasets is below the BPNN model in our method, but the difference is less than 1.5%. In other words, our algorithm may perform worse than BPNN model in some datasets, however, it won't be too bad, and for some datasets, BASNNC can be significantly better than BPNN model. As shown in Table 4 and Table 5, our method can achieve the classification effect of BPNN model.

TABLE 6. Comparison of the run time between the proposed BASNNC and BPNN.

Dataset	Run time (s)	
	BASNNC (λ)	BPNN
Iris	34.8913 (↑**225.80%**)	113.6755
Zoo	95.0023 (↑**179.55%**)	265.5779
BCW	42.1671 (↑**541.09%**)	270.3288
BA	125.7478 (↑**773.48%**)	1098.3779
BTSC	28.2534 (↑**237.91%**)	95.4722
Cryotherapy	25.3829 (↑**178.40%**)	70.6672
HS	16.3268 (↑**201.17%**)	49.1707
Hayes-Roth	24.6370 (↑**182.50%**)	69.5993
ILPD	55.9830 (↑**340.79%**)	246.7653

TABLE 7. Comparison of the run time between the proposed BASNNC and PSO-BPNN.

Dataset	Run time (s)	
	BASNNC (λ)	PSO-BPNN
Iris	34.8913 (↑**678.79%**)	271.7290
Zoo	95.0023 (↑**195.58%**)	280.8060
BCW	42.1671 (↑**1124.59%**)	516.3720
BA	125.7478 (↑**659.22%**)	954.7000
BTSC	28.2534 (↑**3180.37%**)	926.8160
Cryotherapy	25.3829 (↑**604.06%**)	178.7110
HS	16.3268 (↑**2159.15%**)	368.8470
Hayes-Roth	24.6370 (↑**912.25%**)	249.3880
ILPD	55.9830 (↑**1156.87%**)	703.6340

Table 6, Table 7 and Figure 4 show the time required for BASNNC and comparison methods to finish classification. Table 6 and Table 7 is the numerical result, and Figure 4 is the histogram of Table 6. The formula for value-added rate λ of time in Table 6:

$$\lambda = \frac{\max(A, B) - \min(A, B)}{\min(A, B)} \times 100\%, \tag{15}$$

where A and B represent the running time of different methods. Combined with Table 1, we can find that the running time of the program is affected by the number of samples of the dataset, the number of attributes, the number of categories, and so on. It can be seen from the table and figure that the time required by our method is greatly improved compared with other model run time. This is the biggest advantage of our approach. We can express our advantage more intuitively through Figure 4.

Furthermore, to clearly and fully show the superiorities of BASNNC, Table 8 in the revised manuscript qualitatively compares our approach with other NN algorithms, e.g., the WASD algorithm in existing literatures. Due to the fact that the proposed BASNNC requires no matrix inversion. Therefore, it has low computational complexity.

TABLE 8. Comparison of performance between the proposed BASNNC and other BP neural network models.

Method	Performance		
	Speed	Gradient	Inversion
The proposed	**Fast**	**No**	**No**
[56]	Fast	No	Yes
[38]	Fast	No	Yes
[58]	Slow	Yes	No
[59]	Slow	No	No
[57]	Fast	No	Yes
[60]	Slow	Yes	No

Moreover, the proposed method is no need seeking the grads of function, and as a consequence it can run fast. As shown in the table, the BAS neural network classifier is effective. We can solve some of the inherent defects of traditional BP algorithms, such as slow learning.

V. CONCLUSION

We have proposed and investigated a novel neural network classifier (NNC) using the beetle antennae search (BAS) algorithm which termed BASNNC. With the purpose of determining a relatively proper neural network structure, we have chosed a way to select the number of neural network hidden layer neurons. To further verify the classification ability of this model, we have selected several different classification datasets to train and test the network. The results of numerical studies, applications and comparisons have illustrated the effectiveness of the algorithm. It has shown that the proposed method can quickly and effectively classify the classification datasets, overcome some inherent defects of traditional back-propagation (BP) algorithms. Future work can improve the accuracy of the classification. The BAS algorithm is improved, and the BAS is applied to optimize the number of nodes in the hidden layer of the neural network. It is also possible to increase the number of layers of the neural network and expand the hypothesis space of the neural network, and apply the BAS to more complex network structure research.

REFERENCES

[1] L. Liu, Y.-J. Liu, and S. Tong, "Fuzzy based multi-error constraint control for switched nonlinear systems and its applications," *IEEE Trans. Fuzzy Syst.*, to be published.

[2] Z. Dong, J. Zhao, J. Duan, M. Wang, and H. Wang, "Research on agricultural machinery fault diagnosis system based on expert system," in *Proc. 2nd IEEE Adv. Inf. Manage., Communicates, Electron. Automat. Control Conf. (IMCEC)*, May 2018, pp. 2057–2060.

[3] Y.-J. Liu, Q. Zeng, S. Tong, C. L. P. Chen, and L. Liu, "Adaptive neural network control for active suspension systems with time-varying vertical displacement and speed constraints," *IEEE Trans. Ind. Electron.*, to be published.

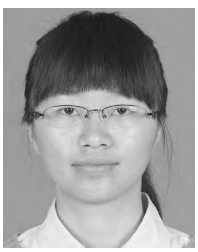
[4] L. Liu, Z. Wang, and H. Zhang, "Neural-network-based robust optimal tracking control for MIMO discrete-time systems with unknown uncertainty using adaptive critic design," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1239–1251.

[5] M. F. Mohammed and C. P. Lim, "An enhanced fuzzy min–max neural network for pattern classification," *IEEE Trans. neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 417–429, Mar. 2015.

[6] L. Jin, Z. Huang, Y. Li, Z. Sun, H. Li, and J. Zhang, "On modified multi-output Chebyshev-polynomial feed-forward neural network for pattern classification of wine regions," *IEEE Access*, vol. 7, pp. 1973–1980, 2019.
[7] J. P. Dominguez-Morales *et al.*, "Deep spiking neural network model for time-variant signals classification: A real-time speech recognition approach," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.
[8] D. Chen, S. Li, and Q. Wu, "Rejecting chaotic disturbances using a super-exponential-zeroing neurodynamic approach for synchronization of chaotic sensor systems," *Sensors*, vol. 19, no. 1, p. 74, 2019.
[9] L. Liu, Y.-J. Liu, and S. Tong, "Neural networks-based adaptive finite-time fault-tolerant control for a class of strict-feedback switched nonlinear systems," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2536–2545, Jul. 2019.
[10] D. Chen, Y. Zhang, and S. Li, "Zeroing neural-dynamics approach and its robust and rapid solution for parallel robot manipulators against superposition of multiple disturbances," *Neurocomputing*, vol. 275, pp. 845–858, Jan. 2018.
[11] L. Xiao, S. Li, F. Lin, Z. Tan, and A. H. Khan, "Zeroing neural dynamics for control design: Comprehensive analysis on stability, robustness, and convergence speed," *IEEE Trans. Ind. Inform.*, vol. 15, no. 5, pp. 2605–2616, May 2019.
[12] S. Zhang, Y. Xia, and J. Wang, "A complex-valued projection neural network for constrained optimization of real functions in complex variables," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3227–3238, Dec. 2015.
[13] D. Chen and Y. Zhang, "A hybrid multi-objective scheme applied to redundant robot manipulators," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 3, pp. 1337–1350, Jul. 2017.
[14] R. Kamesh and K. Y. Rani, "Novel formulation of adaptive MPC as EKF using ANN model: Multiproduct semibatch polymerization reactor case study," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 12, pp. 3061–3073, Dec. 2017.
[15] J. D. M. Rennie, L. Shih, J. Teevan, and D. R. Karger, "Tackling the poor assumptions of naive bayes text classifiers," in *Proc. 20th Int. Conf. Int. Conf. Mach. Learn. (ICML)*. Menlo Park, CA, USA: AAAI Press, 2003, pp. 616–623.
[16] X. Liu and J. Tang, "Mass classification in mammograms using selected geometry and texture features, and a new SVM-based feature selection method," *IEEE Syst. J.*, vol. 8, no. 3, pp. 910–920, Sep. 2014.
[17] L. Fan, "Accurate robust and efficient error estimation for decision trees," in *Proc. 33rd Int. Conf. Mach. Learn., Mach. Learn. Res. (PMLR)*, vol. 48, M. F. Balcan and K. Q. Weinberger, Eds. New York, NY, USA: Jun. 2016, pp. 239–247.
[18] M. Alfaro-Ponce, A. A. Cruz, and I. Chairez, "Adaptive identifier for uncertain complex nonlinear systems based on continuous neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 483–494, Mar. 2014.
[19] G. Ou and Y. L. Murphey, "Multi-class pattern classification using neural networks," *Pattern Recognit.*, vol. 40, no. 1, pp. 4–18, Jan. 2007.
[20] S. Duan, X. Hu, Z. Dong, L. Wang, and P. Mazumder, "Memristor-based cellular nonlinear/neural network: Design, analysis, and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1202–1213, Jun. 2015.
[21] X. Xiao, L. Jin, Y. Yang, W. Yang, J. Sun, and T. Chang, "Building fast and compact convolutional neural networks for offline handwritten chinese character recognition," *Pattern Recognit.*, vol. 72, pp. 72–81, Dec. 2017.
[22] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2013, pp. 8599–8603.
[23] K. W. Ko and H. S. Cho, "Solder joints inspection using a neural network and fuzzy rule-based classification method," *IEEE Trans. Electron. Packag. Manuf.*, vol. 23, no. 2, pp. 93–103, Apr. 2000.
[24] Y. Cheng and H. Zhao, "Fault detection and diagnosis for railway switching points using fuzzy neural network," in *Proc. IEEE 10th Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2015, pp. 860–865.
[25] C. Yao *et al.*, "A convolutional neural network model for online medical guidance," *IEEE Access*, vol. 4, pp. 4094–4103, 2016.
[26] M. Gong, J. Liu, H. Li, Q. Cai, and L. Su, "A multiobjective sparse feature learning model for deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3263–3277, Dec. 2015.
[27] J. Liao, T. Liu, M. Liu, J. Wang, Y. Wang, and H. Sun, "Multi-context integrated deep neural network model for next location prediction," *IEEE Access*, vol. 6, pp. 21980–21990, 2018.
[28] S. Lin and G. C. Runger, "GCRNN: Group-constrained convolutional recurrent neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4709–4718, Oct. 2018.
[29] S. Li, H. Wang, and M. U. Rafique, "A novel recurrent neural network for manipulator control with improved noise tolerance," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1908–1918, May 2018.
[30] D. Chen, Y. Zhang, and S. Li, "Tracking control of robot manipulators with unknown models: A jacobian-matrix-adaption method," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3044–3053, Jul. 2018.
[31] D. Chen and Y. Zhang, "Robust zeroing neural-dynamics and its time-varying disturbances suppression model applied to mobile robot manipulators," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4385–4397, Sep. 2018.
[32] S. Ding, C. Su, and J. Yu, "An optimizing BP neural network algorithm based on genetic algorithm," *Artif. Intell. Rev.*, vol. 36, no. 2, pp. 153–162, 2011.
[33] S. Liu, Z. Hou, and C. Yin, "Data-driven modeling for UGI gasification processes via an enhanced genetic BP neural network with link switches," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2718–2729, Dec. 2016.
[34] Y. Zhang, D. Chen, D. Guo, B. Liao, and Y. Wang, "On exponential convergence of nonlinear gradient dynamics system with application to square root finding," *Nonlinear Dyn.*, vol. 79, no. 2, pp. 983–1003, Jan. 2015.
[35] Y. Zhang and N. Tan, "Weights direct determination of feedforward neural networks without iterative bp-training," in *Intelligent Soft Computation and Evolving Data Mining: Integrating Advanced Technologies*. Hershey, PA, USA: IGI Global, 2010, pp. 197–225.
[36] Y. Zhang, X. Yu, D. Guo, Y. Yin, and Z. Zhang, "Weights and structure determination of multiple-input feed-forward neural network activated by Chebyshev polynomials of Class 2 via cross-validation," *Neural Comput. Appl.*, vol. 25, nos. 7–8, pp. 1761–1770, Dec. 2014.
[37] Y. Zhang, Y. Yin, D. Guo, X. Yu, and L. Xiao, "Cross-validation based weights and structure determination of chebyshev-polynomial neural networks for pattern classification," *Pattern Recognit.*, vol. 47, no. 10, pp. 3414–3428, 2014.
[38] Y. Zhang, D. Chen, L. Jin, Y. Wang, and F. Luo, "Twice-pruning aided WASD neuronet of Bernoulli-polynomial type with extension to robust classification," in *Proc. IEEE 11th Int. Conf. Dependable, Autonomic Secure Comput. (DASC)*, Dec. 2013, pp. 334–339.
[39] H. G. Han, W. Lu, Y. Hou, and J. F. Qiao, "An adaptive-PSO-based self-organizing RBF neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 104–117, Jan. 2018.
[40] S. Salcedo-Sanz and X. Yao, "A hybrid Hopfield network-genetic algorithm approach for the terminal assignment problem," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 6, pp. 2343–2353, Dec. 2004.
[41] R. Zhang, J. Tao, R. Lu, and Q. Jin, "Decoupled ARX and RBF neural network modeling using PCA and GA optimization for nonlinear distributed parameter systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 2, pp. 457–469, Feb. 2018.
[42] C. Ren, N. An, J. Wang, L. Li, B. Hu, and D. Shang, "Optimal parameters selection for BP neural network based on particle swarm optimization: A case study of wind speed forecasting," *Knowl.-Based Syst.*, vol. 56, pp. 226–239, Jan. 2014.
[43] C. B. Khadse, M. A. Chaudhari, and V. B. Borghate, "Conjugate gradient back-propagation based artificial neural network for real time power quality assessment," *Int. J. Electr. Power Energy Syst.*, vol. 82, pp. 197–206, Nov. 2016.
[44] L. Zhang, K. Li, E. W. Bai, and G. W. Irwin, "Two-stage orthogonal least squares methods for neural network construction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1608–1621, Aug. 2015.
[45] Y. Da and G. Xiurun, "An improved PSO-based ANN with simulated annealing technique," *Neurocomputing*, vol. 63, pp. 527–533, Jan. 2005.
[46] L. Li, Y. Chen, T. Xu, R. Liu, K. Shi, and C. Huang, "Super-resolution mapping of wetland inundation from remote sensing imagery based on integration of back-propagation neural network and genetic algorithm," *Remote Sens. Environ.*, vol. 164, pp. 142–154, Jul. 2015.
[47] M. Mavrovouniotis and S. Yang, "Training neural networks with ant colony optimization algorithms for pattern classification," *Soft Comput.*, vol. 19, no. 6, pp. 1511–1522, 2015.
[48] F. Yu and X. Xu, "A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network," *Appl. Energy*, vol. 134, no. 134, pp. 102–113, Dec. 2014.

[49] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural Networks for Perception*. Amsterdam, The Netherlands: Elsevier, 1992, pp. 65–93.
[50] X. Jiang and S. Li. (2017). "BAS: Beetle antennae search algorithm for optimization problems." [Online]. Available: https://arxiv.org/abs/1710.10724
[51] Y. Zhang, S. Li, and B. Xu, "Convergence analysis of beetle antennae search algorithm and its applications," 2019, *arXiv:1904.02397*. [Online]. Available: https://arxiv.org/abs/1904.02397
[52] Q. Wu *et al.*, "Intelligent beetle antennae search for UAV sensing and avoidance of obstacles," *Sensors*, vol. 19, no. 8, P. 1758, 2019.
[53] W. H. Wolberg and O. L. Mangasarian, "Multisurface method of pattern separation for medical diagnosis applied to breast cytology." *Proc. Nat. Acad. Sci.*, vol. 87, no. 23, pp. 9193–9196, 1990.
[54] J. M. D. Sá, L. A. Alexandre, W. Duch, and D. Mandic, *Artificial Neural Networks-ICANN 2007*, vol. 4669, no. 1. Porto, Portugal: Springer-Verlag, 2007, pp. 2409–2410.
[55] A. E. P. Villa, W. Duch, P. Érdi, F. Masulli, and G. Palm, *Artificial Neural Networks and Machine Learning–ICANN 2012*. 2012.
[56] Y. Zhang, Y. Wang, W. Li, Y. Chou, and Z. Zhang, "WASD algorithm with pruning-while-growing and twice-pruning techniques for multi-input euler polynomial neural network," *Int. J. Artif. Intell. Tools*, vol. 25, no. 2, 2016, Art. no. 1650007.
[57] Y. Zhang, J. Chen, S. Fu, L. Xiao, and X. Yu, "Weights and structure determination (WASD) of multiple-input hermit orthogonal polynomials neural network (MIHOPNN)," in *Proc. 24th Chin. Control Decis. Conf. (CCDC)*, May 2012, pp. 1106–1111.
[58] L. Wang, Y. Zeng, and T. Chen, "Back propagation neural network with adaptive differential evolution algorithm for time series forecasting," *Expert Syst. Appl.*, vol. 42, no. 2, pp. 855–863, 2015.
[59] G. Chen *et al.*, "The genetic algorithm based back propagation neural network for MMP prediction in CO_2-EOR process," *Fuel*, vol. 126, pp. 202–212, Jun. 2014.
[60] T. F. Burks, S. A. Shearer, J. R. Heath, and K. D. Donohue, "Evaluation of neural-network classifiers for weed species discrimination," *Biosyst. Eng.*, vol. 91, no. 3, pp. 293–304, 2005.

QING WU received the Ph.D. degree in computer science from Zhejiang University, in 2006. He is currently a Professor with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. His current research interests include machine learning, data mining, adaptive software, and ubiquitous computing.



ZHEPING MA received the B.S. degree in computer science and technology from Qingdao University, Qingdao, China, in 2015. She is currently pursuing the M.S. degree in computer technology with Hangzhou Dianzi University, Hangzhou, China. Her research interests include pattern classification, intelligent optimization algorithms, and machine learning.

GANG XU received the B.Sc. degree in computational mathematics from Shandong University, in 2003, and the Ph.D. degree in applied mathematics from Zhejiang University, in 2008. He was a Postdoctoral Researcher with INRIA Sophia-Antipolis, from 2008 to 2010. He is currently a Professor with the School of Computer Science and Technology, Hangzhou Dianzi University. His research interests include image processing and computer graphics. He received the Young Scholar Award on Geometric Design and Computing of China, in 2016.

SHUAI LI received the B.E. degree in precision mechanical engineering from the Hefei University of Technology, China, in 2005, the M.E. degree in automatic control engineering from the University of Science and Technology of China, China, in 2008, and the Ph.D. degree in electrical and computer engineering from the Stevens Institute of Technology, USA, in 2014. He is currently a Research Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His current research interests include dynamic neural networks, wireless sensor networks, robotic networks, machine learning, and other dynamic problems defined on a graph. He is currently on the Editorial Board of *Neural Computing and Applications* and the *International Journal of Distributed Sensor Networks*.

DECHAO CHEN received the B.S. degree in electronic information science and technology from the Guangdong University of Technology, Guangzhou, China, in 2013, and the Ph.D. degree in information and communication engineering from Sun Yat-sen University, Guangzhou, in 2018. He is currently a Postdoctoral Fellow with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. He is also an Associate Professor with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. His research interests include robotics, neural networks, dynamics systems, control systems, optimization, and machine learning.