Acta Technica 62 No. 2B/2017, 283–294

BP neural network based on wolf pack algorithm optimization

Xushan Peng[2], Yongping Li[2], Xiaoming Zhang[2], Shui Wang[2]

Abstract. BP neural network has a good ability to fit nonlinear functions. But the convergence speed of traditional BP algorithm is slow, so that it is difficult to achieve the desired results. Many scholars begin to study the initial weights and thresholds of BP neural networks, so as to strengthen the reliability of BP algorithm by optimizing initial weights and thresholds. In this paper, a BP neural network algorithm based on wolf pack optimization was proposed. Then the initial weights and thresholds were obtained by the wolf optimization algorithm. Finally, a nonlinear function was used, and the reliability of the algorithm was verified. The results show that the BP neural network based on wolf pack algorithm optimization has been greatly improved in terms of prediction accuracy and reliability, and it takes less time to process, which not only improves reliability, but also saves a lot of time.

Key words. Wolf pack algorithm, BP neural network, optimization.

1. Introduction

Traditional BP neural networks often suffer from slow convergence and local minimum. One of the reasons is that the connection weights and thresholds are arbitrarily taken between $< -1, 1 >$ [1]. Particle swarm optimization, genetic algorithm and ant colony algorithm have the function of global search and optimization Therefore, many researchers use these algorithms to study the optimization of initial

[1]This paper is supported by projects:

1) Zhejiang public welfare Technology Application Research Project, Study of chaotic synchronization and information flow in large scale coupled nonlinear systems (2017C35013).

2)Zhejiang Provincial Natural Science Foundation, Research on Multi-order Brillouin Scattering Mode Locked Laser in RoF systems (LQ13F01004).

3)Ningbo science projects, he development of Dendrobium automatic screening grading equipment of machine vision technology (2016C10056).

4)The scientific research project of Ningbo Dahongying University, Research on Frequent Itemsets Mining Based on block transaction data (1320133017).

[2]School of Information Engineering, Ningbo Dahongying University, Ningbo, Zhejiang, China, 315175

weights and thresholds of networks [2]. These algorithms have more or less defects, and the optimization results of genetic algorithms are not accurate [3]. In the wolf pack algorithm, based on individual ability difference, the wolves are divided into leader wolf, detective wolf and fierce wolf. And iterative optimization is performed by three intelligent behaviors. The choice of the leader wolf and the regeneration of population reduce the chances that individuals will fall into local minima. The wolf pack optimization method is a kind of intelligent algorithm which is summarized after an analysis of a large number of hunting behaviors [4]. The algorithm has a good advantage in optimizing the multidimensional parameter matrix of neural networks. At present, there is no wolf pack algorithm to optimize BP neural network in the existing research [5]. Therefore, the wolf pack algorithm and BP neural network are combined together to form an improved BP neural network model, so as to prevent the network from falling into local extremum, and to improve the network performance [6].

2. State of the art

BP neural network has a good ability to fit nonlinear functions, but the traditional BP algorithm is easy to diverge. Many scholars begin to study the initial weights and thresholds of BP neural networks, hoping to strengthen the reliability of BP algorithm by optimizing initial weights and thresholds [7]. Therefore, the improved BP neural network algorithm came into being. A BP neural network algorithm based on wolf pack optimization has been proposed, in which, the initial weights and thresholds can be obtained by the wolf pack optimization algorithm [8]. Artificial neural networks based on wolf pack optimization can also be used for spectrum sensing, and can realize the neural network spectrum sensing with the optimal structure of neural networks [9]. In the algorithm, on the basis of spectrum sensing algorithm including self-organizing neural networks, the training sample generation is described in detail. And after the training of neural network and the training phase of neural network, the weight matrix is further optimized by using the wolf pack optimization method [10]. Recently, a method of handwriting identification of probabilistic neural network (WAPNN) based on wolf pack optimization algorithm has been proposed [11]. The method contains the advantages of probabilistic neural networks (PNN) and wolf pack algorithm (WA), forming WAPNN. By using wolf pack algorithm, the optimal smoothing parameter α of PNN can be obtained, and the structure of probabilistic neural network can be optimized. The application shows that the method of probabilistic neural network based on wolf pack optimization algorithm can greatly improve the reliability of machine identification and save a lot of time, which can provide scientific theory support for further development of handwriting identification, and has certain application value [12]. Aiming at the problem of the extraction accuracy of fault signal frequency components and the accuracy of fault location, the Prony algorithm is used to extract the natural frequency of the fault voltage signal as a sample, and then the wolf pack algorithm is applied to optimize the structure of BP neural network and train it, thus to improve the defects that are easy to generate multiple local minima, enhance the training efficiency and convergence speed of the network, and make the distance measurement more accurate [13]. The BP neural network based on wolf pack algorithm optimization has been greatly improved in terms of prediction accuracy and reliability, and it takes less time to process, which not only improves reliability, but also saves a lot of time. ERNN is one of the most efficient feedforward neural networks learning algorithms. In the ERNN training process, the gradient descent technique is used, and therefore, there are no problems such as local minima and slow convergence. The new heuristic search algorithm, known as wolf search (WS) based on predatory behavior of wolves, can achieve faster convergence and avoid local minimization by training weights in ERNN [14].

3. Methodology

3.1. Wolf algorithm principle

The wolf pack optimization algorithm is an intelligent algorithm derived from the natural behavior of the wolf race in nature, which has high accuracy and good reliability, and can optimize other algorithms accurately. Based on the behavior of wolf colony hunt, the bottom-up design concept was adopted. As shown in Fig. 1, after a great deal of behavioral analysis, the specific process of information exchange and responsibility allocation among wolf groups is summed up. Thus, the wolf algorithm is defined, and the specific wolf hunting model is shown below. The specific wolf hunting model is shown below [15].

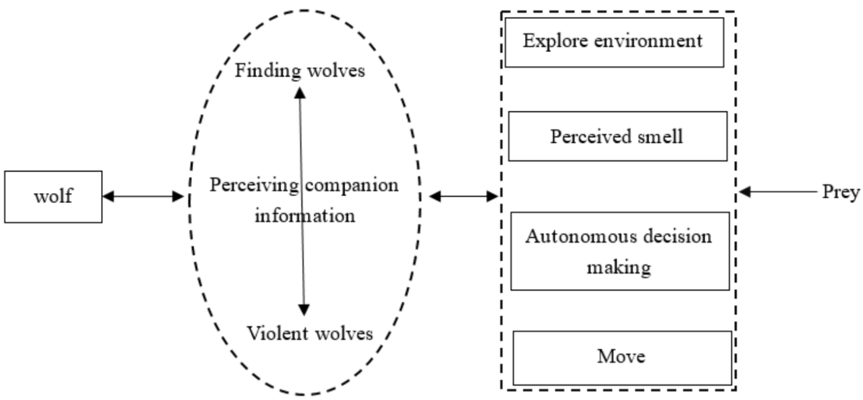

Fig. 1. Wolves hunting model

A resource for wolves to hunt is an $N \times D$ space. Here, N is the number of wolves in a wolf pack, and D is a hunted quantity. The state of an artificial wolf i can be expressed as $X_i = (x_{i1}, x_{i2}, \cdots, x_{id})$, where x_{id} is the position of the ith artificial wolf in the d-dimensional ($d = 1, 2, \cdots, D$) variable space. The concentration of artificial wolf hunting can be perceived as $Y = f(X)$, Y being the expected value. The distance between the wolf p and wolf q is defined the Manhattan distance between their state vectors $L(p,q)=\sum_{d=1}^{D}|x_{pd}-x_{qd}|$. Other methods may be used to define distance without affecting actual results and research steps.

The specific steps of the wolf algorithm are as follows.

Step 1: Initially, the data is computed to obtain some values. The wolf positions X_i and their number N are initialized in the wolves. The maximum allowable number of iterations is $k_{\max}$, the wolf scale factor is α, the maximum number of trips is $T_{\max}$, the distance determination factor is ω, the step size factor is S, and the update scaling factor is β.

Step 2: As the optimal individual in the group, the position of the leader wolf is extremely important. It is necessary to iterate to determine the relationship between the odor concentration Y_i of detective wolf i and Y_{lead}. Then, Step 3 can be executed.

Step 3: If the odor concentration perceived by fierce wolf $Y_i > Y_{\text{lead}}$, then$Y_{\text{lead}} = Y_i$, and then the convening behavior can be carried out; on the contrary, if $Y_i < Y_{\text{lead}}$, Step 4 cannot be executed until $d_{is} \le d_{\text{near}}$.

Step 4: With the formula in Step 3 as standard, the location update and execution of group siege are carried out.

Step 5: According to the mechanism of survival of the fittest, the position of the leader wolf should be constantly updated. In this way, the entire wolf pack can be renewed to ensure overall consistency.

Step 6: It is necessary to determine whether the maximum allowable number of iterations has been reached.

The flow chart of the algorithm is depicted in Fig. 2.

3.2. The principle of fitting function of BP neural network

Neural network is a new method to imitate the working mode of human brain. In recent years, many scholars have done a lot of research and simplification on it. The traditional neural network learning experience shows that one layer is good enough. However, recent depth learning denies the claim. Just like every update value of the gradient descent function, every time a sample is updated, the cost function becomes smaller and smaller. Similarly, weights are firstly given with random initial values. Then, the calculations are carried out until the last layer (output layer). If there is an error between the output and the actual value (which is certain in the normal case), then the error back propagation algorithm is used to optimize the value of each layer (the weight value).

(1) Node output model

Hidden node output model:

$$O_j = f(\sum W_{ij} \times X_i - q_j)\,. \tag{1}$$

Output model of output node:

$$Y_k = f(\sum T_{jk} \times O_j - q_k)\,. \tag{2}$$

(2) Interaction function model

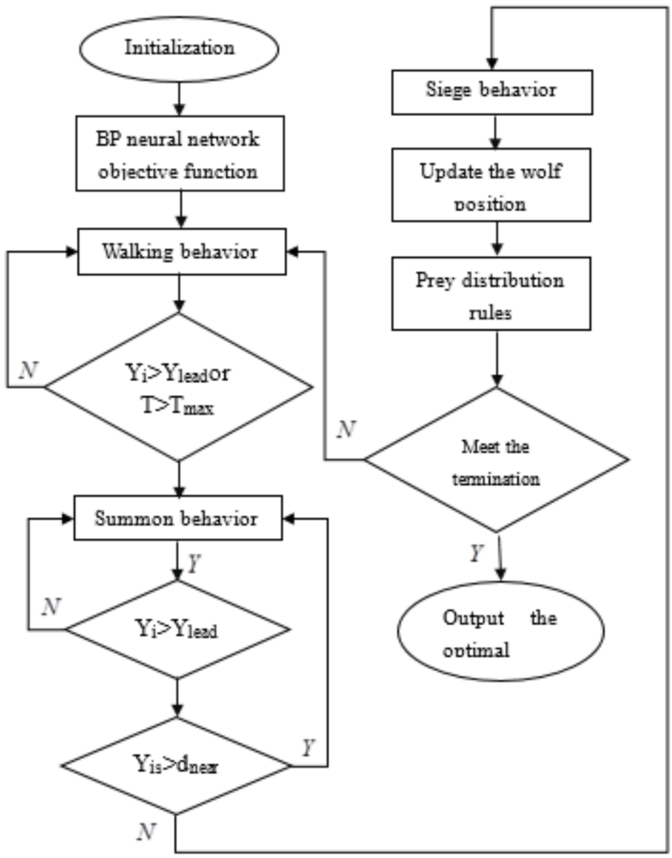

Fig. 2. Flow chart of WPA

The activation function is generally a continuously valued Sigmoid function in (0,1):

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

(3) Error calculation model

$$E_p = 1/2 \sum (t_{pi} - O_{pi})^2\,, \tag{4}$$

where t_{pi} is the theoretical output value of the ith node and O_{pi} is the actual result of the ith node.

(4) Self-learning model

The network output error is reduced to an acceptable level (or to the preset number of studies). The error between the output and expected output is obtained through the output layer to adjust the weight of the hidden layer indirectly. Self-learning model is

$$\Delta W_{ij}(n+1) = h \times \Phi_i \times O_j + a \times \Delta W_{ij}(n)\,. \tag{5}$$

The optimization strategy of BP network model includes:

(1) Optimization of learning factor h

The step size can be adjusted to make the target value the best, and then the number of operations can be reduced and the efficiency of calculation can be improved.

$$h = h + a \times (E_{p(n)} - E_{p(n-1)})/E_{p(n)}\,, \tag{6}$$

where a is the adjustment step size, and its value is 0∼1.

(2) Optimization of node number in hidden layer

The number of hidden nodes has a great influence on the performance of the algorithm, and the empirical formula can be used to determine the number of hidden nodes. In the BP algorithm, the weights and thresholds are adjusted once every training session. Firstly, the weights are assigned with a random initial value, and then the calculation is run until the last layer (output layer). If the output result is in error with the actual value, it is necessary to continue to judge; if there is no error, the discriminant can be terminated. It is difficult to determine the optimal value among the neurons in the actual hidden layer, which is usually determined by trial and error method or optimization algorithm. The number of nodes in the input layer is 3, the number of nodes in the output layer is 1, and the number of nodes in the hidden layer can be within 2–8. The empirical formula of the best hidden node number L is

$$L = (m+n)^{1/2} + c\,. \tag{7}$$

(3) Algorithm optimization

In the BP algorithm, a gradient descent method is used. Weights are firstly given with random initial values. Then, the calculations are carried out until the last layer (output layer). If there is an error between the output and the actual value (which is certain in the normal case), then the error back propagation algorithm is used to overcome the shortcomings.

3.3. BP neural network based on wolf pack algorithm optimization

The basic idea of the BP neural network based on wolf pack algorithm optimization is derived from the natural behavior of the wolf race in nature. The algorithm has high precision and good reliability, and can optimize the BP neural network algorithm accurately. Based on the behavior of wolves during hunting, a bottom-up design concept is adopted. According to the specific process of information exchange and responsibility allocation among wolf groups, an algorithm can be obtained to optimize the BP neural network. In other words, the state of the artificial wolf is used to represent the weights and thresholds of the BP neural network. Then through the process of information exchange and responsibility allocation among wolf groups, the best weights and thresholds can be found as initial values. After screening the initial values in the above steps, the BP neural network can be trained to achieve the prediction effect. Because its initial value is derived from the wolf pack optimization algorithm, instead of artificial substitution according to experience, this method is more accurate and rigorous compared with the traditional prediction model. After the Wolf algorithm optimization, training was carried out. Under the same conditions, the BP neural network based on wolf pack algorithm optimization has been greatly improved in terms of prediction accuracy and reliability, and it takes less time to process, which not only improves reliability, but also saves a lot of time. Thus, this method is rigorous in theory and is clear in logic, and is suitable for theoretical research and practical application in engineering. The main steps of the algorithm are as follows:

(1) Initialization parameter, including the number of individual wolves, the maximum number of iterations, the wolf scale factor, the maximum number of trips, the distance determination factor, the step factor, and the update scaling factor.

(2) Determination of odor concentration function. This method can obtain two values at the same time, one is the expected value, and the other is the predictive value. The size of the two is generally unequal, and the odor concentration function is the sum of the absolute error between the expected value and the predicted value. Formula (8) shows the concrete operations.

$$F = \sum_{i=1}^{n} |y_i - o_i| \,. \tag{8}$$

(3) Detective wolf's wandering behavior. As the optimal individual in the group, the position of the leader wolf is extremely important. It is necessary to iterate to determine the relationship between the odor concentration Y_i of detective wolf i and Y_{lead}. Then, steps can be executed.

$$x_{id}^{p} = x_{id} + \sin(2\pi p/h)\text{step}_a^d \tag{9}$$

In the formula, step_a is the wandering length.

(4) Fierce wolf approaches prey by formula (9), if $Y_i > Y_{\text{lead}}$, and then $Y_{\text{lead}} = Y_i$, and the fierce wolf can replace the leader wolf; if $Y_i < Y_{\text{lead}}$, then fierce wolf continues to approach the prey until $d_{is} \leq d_{\text{near}}$. The distance between individual wolves a and b is defined as Manhattan, and the distance is $L(a,b) = \sum_{d=1}^{D} |x_{ad} - x_{bd}|\, d_{\text{new}} = \frac{1}{D_{gw}} \sum_{d=1}^{D} |\text{max}_d - \text{min}_d|$. The range of variable d to be optimized is $[\text{min}_d, \text{max}_d]$. The value step_b is the step size of fierce wolf approximating leader wolf.

$$x_{id}^{k+1} = x_{id}^{k} + \text{step}_b^d g(g_d^k - x_{id}^k) / \left|g_d^k - x_{id}^k\right| \,. \tag{10}$$

(5) Using the above formula, the status of the besieged individual wolf is updated:

$$x_{id}^{k+1} = x_{id}^{k} + \lambda g \text{step}_b^d g \left|G_d^k - x_{id}^k\right| \,. \tag{11}$$

In the formula, λ is the random number between <-1, 1>, and step_c is the siege step of the ith wolf.

(6) According to the mechanism of survival of the fittest, the position of the leader wolf should be constantly updated. In this way, the entire wolf pack can be renewed to ensure overall consistency.

(7) The state of the artificial wolf is used to represent the weights and thresholds of the BP neural network. Then through the process of information exchange and responsibility allocation among wolf groups, the best weights and thresholds can be found as initial values. After screening the initial value.s in the above steps, the BP neural network is trained to achieve the prediction effect.

The flow chart of the above algorithm is depicted in Fig. 3

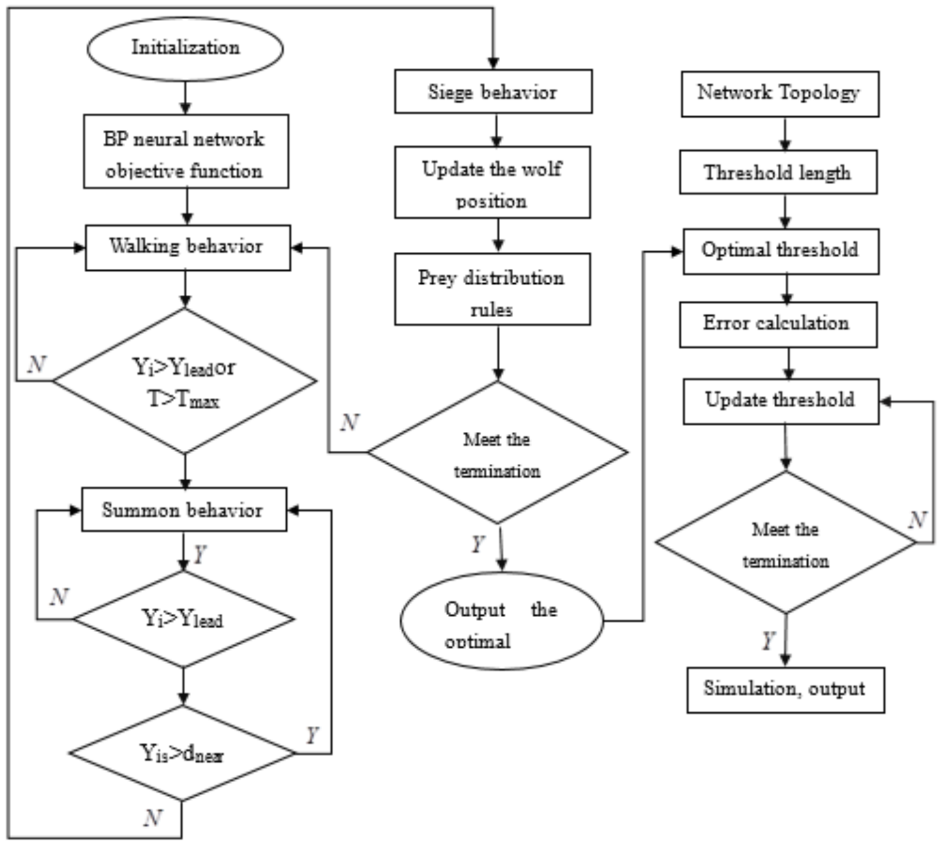

Fig. 3. Flow chart of algorithm

4. Result analysis and discussion

MATLAB software was used to carry out simulation experiments. MATLAB is the very powerful data processing software, which has a huge advantage in matrix computing, also known as the matrix laboratory. Thus, using MATLAB to solve such problems is a good method. MATLAB software is used to write the m file of these algorithms and then carry out nested calls. First of all, the state of the artificial wolf is used to represent the weights and thresholds of the BP neural network. Then through the process of information exchange and responsibility allocation among wolf groups, the best weights and thresholds can be found as initial values. After that, the initial weights and thresholds are optimized to enhance the reliability of the BP algorithm. After screening the initial values in the above steps, the BP neural network is trained to achieve the prediction effect.

The above algorithms were written in MATLAB2012b by using MATLAB language. MATLAB2012b is a classic MATLAB software series, which can carry out simulation, numerical analysis, and graphics processing at the same time. To verify the effectiveness of the BP neural network model based on wolf pack algorithm optimization (WPABP model) in fitting function, the BP neural network (BP model) without optimization and the BP neural network (GABP model) based on genetic algorithm optimization were compared in terms of the accuracy of function fitting. The following figure shows comparison results. The fitting error comparison is depicted in Fig. 4.

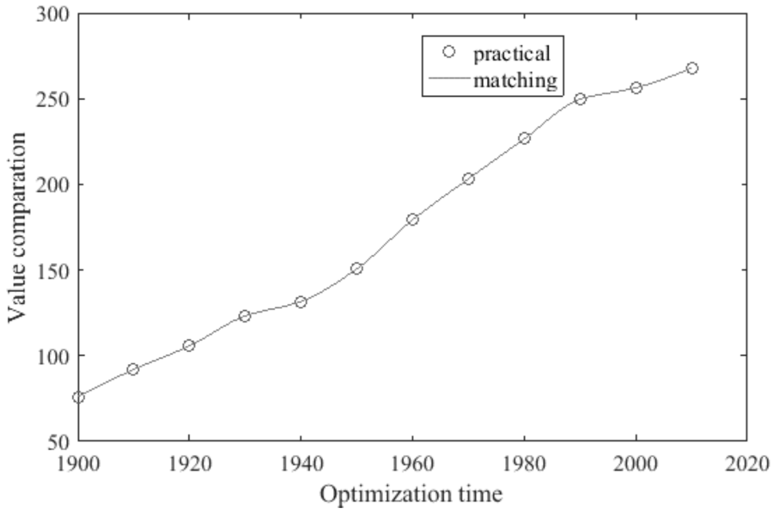

Fig. 4. Fitting error comparison

As can be seen from the diagram above, the predicted result fluctuates from top to bottom near the expected value, and as the number of runs increases, the error becomes bigger and bigger. This result is in line with the actual situation. In most cases, as the number of runs increases, the computational burden of the computer increases greatly, and a variety of problems such as storage space become prominent, thus affecting the accuracy of the results to a certain extent. The results are shown in Table 1.

Table 1. Results comparison

	Correct rate	Test probability	Time
Before optimization	0.8808	0.9903	7.7524
After optimization	0.9126	0.9966	7.6509

As can be seen from the table, after the Wolf algorithm was optimized, and then the training was carried out. Under the same conditions, the results of comparison were: in the case of no wolf algorithm optimization, the accuracy of the prediction was 0.8808, and the training reliability was 0.9903, and the time required from the start of the operation to the completion of the forecast was 7.7524 s; while in the case of wolf algorithm optimization, the accuracy of the prediction was 0.9126, and the training reliability was 0.9966, and the time required from the start of the operation to the completion of the forecast was 7.6509 s. According to these specific data, it can be found that the BP neural network based on wolf pack algorithm optimization has been greatly improved in terms of prediction accuracy and reliability, and it takes less time to process, which not only improves reliability, but also saves a lot of time.

5. Conclusion

In this paper, a BP neural network based on wolf pack algorithm optimization was proposed. The initial weights and thresholds were obtained by the wolf pack optimization algorithm. Finally, a nonlinear function was used to perform simulation experiments on MATLAB software, and then the reliability of the algorithm was verified. Under the same conditions, the results of comparison were: in the case of no wolf algorithm optimization, the accuracy of the prediction was 0.8808, and the training reliability was 0.9903, and the time required from the start of the operation to the completion of the forecast was 7.7524 s; while in the case of wolf algorithm optimization, the accuracy of the prediction was 0.9126, and the training reliability was 0.9966, and the time required from the start of the operation to the completion of the forecast was 7.6509 s. According to these specific data, it can be found that the BP neural network based on wolf pack algorithm optimization has been greatly improved in terms of prediction accuracy and reliability, and it takes less time to process, which not only improves reliability, but also saves a lot of time.

References

[1] R. Wang, N. Wang: *Research on BP neural network optimization based on ant colony algorithm.* Applied Mechanics and Materials *513–517* (2014), 1819–1821.

[2] H. Y. Li, X. X. Wu, T. Wang, D. Zhao: *A simulation and prediction method of the height of wheat based on BP neural networks and the ant colony algorithm.* Advanced Materials Research *659* (2013), 113–117.

[3] Q. D. Zhao, Y. Yu, M. M. Jia: *Applied-information technology in short-term wind

speed forecast model for wind farms based on ant colony optimization and BP neural network. Applied Mechanics and Materials *662* (2014), 259–262.

[4] E. Emary, H. M. Zawbaa, C. Grosan: *Experienced Gray Wolf optimization trough reinforcement learning and neural networks.* IEEE Transactions on Neural Networks and Learning Systems *PP* (2017), No. 99, 1–14.

[5] W. Jia, D. Zhao, T. Shen, S. Ding, Y. Zhao. C. HU: *An optimized classification algorithm by BP neural network based on PLS and HCA.* Applied Intelligence *43* (2015), No. 1, 176–191.

[6] G. Sun, T. Chen, Z. Wei, Y. Sun, H. Zhan, S. Chen: *A carbon price forecasting model based on variational mode decomposition and spiking neural networks.* Energies *9* (2016), No. 1, paper 54.

[7] W. Jia, D. Zhao, Y. Tang, C. Hu, Y. Zhao: *An optimized classification algorithm by neural network ensemble based on PLS and OLS.* Mathematical Problems in Engineering (2014), Article ID No. 395263, 1–8.

[8] K. Zhang, F. Yuan, J. Guo, G. Wang: *A novel neural network approach to transformer fault diagnosis based on momentum-embedded BP neural network optimized by genetic algorithm and fuzzy c-means.* Arabian Journal for Science and Engineering *41* (2015), No. 9, 3451–3461.

[9] M. Diao, R. Qian, H. Gao: *Spectrum sensing algorithm based on neural network with wolf pack optimization.* Computer Engineering and Applications *52* (2016), No. 19, 107–110.

[10] M. Madhiarasan, S. N. Deepa: *Long-term wind speed forecasting using spiking neural network optimized by improved modified grey wolf optimization algorithm.* International Journal of Advanced Research *4* (2016), No. 7, 356–368.

[11] D. Wang, H. Luo, O. Grunder, Y. Lin, H. Guo: *Multi-step ahead electricity price forecasting using a hybrid model based on two-layer decomposition technique and BP neural network optimized by firefly algorithm.* Applied Mechanics and Materials *190* (2017), 390–407.

[12] S. Amirsadri, S. J. Mousavirad, H. Ebrahimpour-Komleh: *A Levy flight-based grey wolf optimizer combined with back-propagation algorithm for neural network training.* Neural Computing and Applications (2017), Nos. 3–4, 1–14.

[13] W. Han, X. Shen, E. Hou, J. Xu: *Precision time synchronization control method for smart grid based on wolf colony algorithm.* International Journal of Electrical Power & Energy Systems *78* (2016), 816–822.

[14] Y. Yang, G. Wang, Y. Yang: *Parameters optimization of polygonal fuzzy neural networks based on GA-BP hybrid algorithm.* International Journal of Machine Learning and Cybernetics *5* (2014), No. 5, 815–822.

[15] S. Lahmiri: *Comparing variational and empirical mode decomposition in forecasting day-ahead energy prices.* IEEE Systems Journal *PP* (2015), No. 99, 1–4.

Received August 7, 2017