Turkish Journal of Electrical Engineering and Computer Sciences

Volume 25 | Number 3 Article 6

1-1-2017

Training ANFIS by using the artificial bee colony algorithm

DERVİŞ KARABOĞA

EBUBEKİR KAYA

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 1669 – 1679

doi:10.3906/elk-1601-240

Training ANFIS by using the artificial bee colony algorithm

Derviş KARABOĞA[1], **Ebubekir KAYA**[2,*]
[1]Department of Computer Engineering, Faculty of Engineering, Erciyes University, Kayseri, Turkey
[2]Department of Computer Technologies, Nevşehir Vocational College, Nevşehir Hacı Bektaş Veli University, Nevşehir, Turkey

Received: 21.01.2016 • **Accepted/Published Online:** 09.06.2016 • **Final Version:** 29.05.2017

Abstract: In this study, a new adaptive network-based fuzzy inference system (ANFIS) training algorithm, the artificial bee colony (ABC) algorithm, is presented. Antecedent and conclusion parameters existing in the structure of ANFIS are optimized with the ABC algorithm and ANFIS training is realized. Identification of a set of nonlinear dynamic systems is performed in order to analyze the suggested training algorithm. The ABC algorithm is operated 30 times for each identification case and the average root mean square error (RMSE) value is obtained. Training RMSE values calculated for the four examples considered are 0.0325, 0.0215, 0.0174, and 0.0294, respectively. In addition, test error values for the same cases are respectively computed as 0.0270, 0.0186, 0.0167, and 0.0435. The results obtained are compared with those of known neuro-fuzzy-based methods frequently used in the literature in identification studies of nonlinear systems. It is shown that ANFIS can be trained successfully by using the ABC algorithm for the identification of nonlinear systems.

Key words: ANFIS, swarm intelligence, artificial bee colony algorithm, nonlinear system identification

1. Introduction

The adaptive network-based fuzzy inference system (ANFIS) is based on the idea of combining the learning ability of artificial neural networks and superiorities of fuzzy logic, such as humanlike decision-making and the easiness of providing expert knowledge [1]. Thus, although the learning and calculation power of artificial neural networks can be given to fuzzy logic inference systems, the ability of fuzzy logic inference systems for humanlike decision-making and provision of expert knowledge is gained by artificial neural networks. ANFIS uses artificial neural networks found in its internal structure to create the system structure and determine its variables [2]. Therefore, algorithms used in ANFIS training are important. In recent years, many studies have been conducted in this field and different algorithms have been suggested for this purpose. We can categorize these studies on the training of ANFIS into three groups: the first group is to develop a new learning algorithm for ANFIS; the second is to perform ANFIS training with existing optimization algorithms (at this stage, known optimization algorithms may be updated for ANFIS training); and the third is to perform ANFIS training by using known optimization algorithms to solve individual problems, although this group is very similar to the second. However, the second group aims to develop a more general training algorithm for ANFIS.

The main learning algorithm of ANFIS is the hybrid learning algorithm, created by jointly using the least squares method and the backpropagation learning algorithm. It is seen that the number of training

*Correspondence: ebubekir@nevsehir.edu.tr

algorithms used for ANFIS is increasing daily, together with the recently conducted studies. We can generally list training algorithms used for ANFIS as derivative-based ones and nonderivative heuristics. Since derivative-based algorithms trip over local minima in the determination of parameters belonging to membership functions, demand for global heuristic algorithms increases. Therefore, several researchers have recently proposed heuristic search-based training algorithms for ANFIS. Ho et al. [3] performed ANFIS training with the hybrid Taguchi-genetic algorithm to estimate the adequacy of vancomycin regimen. Nariman-Zadeh et al. [4] used a genetic algorithm and the singular value decomposition approach to determine antecedent and conclusion parameters of ANFIS. Chen [5] constructed a model with ANFIS for predicting business failures and used particle swarm optimization (PSO) for optimization of ANFIS parameters. Shoorehdeli et al. [6] suggested PSO and a Kalman filter-based training algorithm for ANFIS. In this study, whereas the parameters belonging to the membership functions found in the structure of ANFIS are optimized with PSO, a Kalman filter is used to find the values of the conclusion parameters. In another study conducted by Shoorehdeli et al. [7], a new hybrid learning algorithm is proposed, where PSO is used for training the antecedent part and the forgetting factor recursive least squares algorithm is employed for training the conclusion part. Jalali-Heravi and Asadollahi-Baboli [8] suggested modified ant colony algorithm-based ANFIS training for the prediction of the inhibitory activity of quinazolinone derivatives on serotonin. Khazraee et al. [9] trained ANFIS with a differential evolution for model reduction and optimization of reactive batch distillation. Priyadharsini et al. [10] proposed an artifact removal study based on ANFIS and used an artificial immune algorithm to optimize the parameters of ANFIS. In our previous study, we used the artificial bee colony (ABC) algorithm to train ANFIS for identifying nonlinear static and dynamic systems. In this study, we present an extended version of previous conference papers [11,12].

When reviewing the literature, it is seen that swarm intelligence-based algorithms, such as PSO and ACO, are used for the training of ANFIS. In this study, ANFIS training is performed with a different algorithm, which is also based on swarm intelligence, known as the ABC algorithm. The ABC algorithm, invented by Karaboga in 2005, is an optimization algorithm showing the intelligent foraging behavior of honey bee swarms [13,14]. It found a wide application area and is used to solve real-world problems [15–17]. The ABC algorithm was used in many different applications and fields: neural networks, such as feed-forward neural networks; multilayer perception neural networks; RBF neural networks; and recurrent neural networks, which were trained by using the ABC algorithm [18]. At the same time, many studies have been conducted in the fields of electrical engineering, electronics engineering, civil engineering, software engineering, control engineering, industrial engineering, and mechanical engineering [15–18]. In addition, the ABC algorithm has been applied to different fields including data mining, sensor networks, image processing, numerical problems, and protein structure optimization [17–19]. Besides these studies, different versions of the ABC algorithm have been developed, namely continuous, combinatorial/discrete, hybrid, chaotic, binary/integer, multiobjective, constrained, parallel, and cooperative versions [18]. These studies show that ABC is quite a successful and popular heuristic optimization algorithm in solving difficult optimization problems. Since the training of ANFIS is accepted as a difficult problem, the ABC algorithm is used for the training of ANFIS in this study. Nonlinear dynamic systems are identified for performance analysis of the proposed method. Application results obtained are compared with those of fuzzy neural network, neural network, and ANFIS-based known methods, used frequently in studies of identification of nonlinear systems.

2. Artificial bee colony algorithm

In the ABC algorithm, foraging bees are classified into three groups: employed, onlooker, and scout bees. The first half of the colony consists of the employed artificial bees and the second half includes the onlookers. There is only one bee employed for each food source. In other words, the number of food sources near the hive corresponds to the number of employed bees. The position of a food source is a possible solution to the optimization problem. The quality of the problem concerned is related to the amount of nectar contained in a food source. The number of employed or onlooker bees is equal to the colony size.

The ABC algorithm has three fundamental control parameters: colony size, limit value, and value of the maximum number of cycles (MCN). The main steps of the ABC algorithm are given below:

Initialize the population of solutions x_i, i = 1 . . . SN

Evaluate the population

Cycle = 1

REPEAT

Produce new solutions v_i for the employed bees by using Eq. (2) and evaluate them

Apply the greedy selection process

Calculate the probability values p_i for solutions x_i with Eq. (3)

Produce the new solutions v_i for the onlookers from solutions x_i, selected depending on p_i, and evaluate them

Apply the greedy selection process

Determine the abandoned solution for the scout, if it exists, and replace it with a new randomly produced solution x_i with Eq. (1)

Memorize the best solution achieved so far

Cycle = Cycle + 1

UNTIL (Cycle = MCN)

In the ABC algorithm, the process starts with the random determination of initial food sources. The random position production process is conducted by generating a value between the lower and upper limits of each parameter, using Eq. (1).

$$x_{ij}=x_j^{\min}+rand\left(\ 0,\quad 1\ \right)\left(x_j^{\max}-x_j^{\min}\right) \tag{1}$$

Here, i = 1...SN and j = 1...PN. SN is the number of food sources and PN is the number of parameters to be optimized. $x_j^{\min}$ describes the lower limit of the j parameter, and $x_j^{\max}$ describes the upper limit of the j parameter.

Employed bees determine a new neighbor food source by using the equality given in Eq. (2) in the neighborhood of the food source and evaluate its quality. Here, j is an integer produced randomly in the interval of [1, D]. ϕ_{ij} is a random value taken in the interval of [–1, 1]. In Eq. (2), the v_i source is found by changing only one parameter of x_i.

$$v_{ij}=x_{ij}+\phi_{ij}\left(x_{ij}-x_{kj}\right) \tag{2}$$

If v_i is better than x_i, the employed bee deletes that address from its memory. Otherwise, it continues to go to the former address. A probabilistic selection process by onlooker bees in the ABC algorithm is performed by using the quality value corresponding to the nectar amount. A selection process based on the quality value was performed by using the roulette wheel method in the basic ABC algorithm. According to the roulette wheel method, the selection probability of sources is computed by Eq. (3). Here, fitness_i shows the quality of the ith source, and SN represents the number of employed bees.

$$p_i = \frac{fitness_i}{\sum_{j=1}^{SN} fitness_j} \tag{3}$$

Once onlooker bees choose a food source, each bee finds a new solution in the neighborhood of the food source, which is chosen by means of Eq. (3). A greedy selection is applied to the new solution and the food source chosen in order to improve the solution at hand. If predetermined cycles do not lead to the enhancement of a food source, which is called the limit, it is removed from the population and the employed bee of that food source becomes a scout. The scout bee looks for a position in the new random food source with Eq. (1).

2.1. Training ANFIS using the ABC algorithm

As mentioned above, ANFIS is a hybrid artificial intelligence algorithm generated by aggregating the learning ability of neural networks and the inference feature of fuzzy logic [1,2]. ANFIS is trained by using the existing input-output data pairs for the solution of available problems. Thus, IF-THEN rules in ANFIS are obtained. The structure of ANFIS consists of five layers. In the training of ANFIS, antecedent and conclusion parameters found in layers 1 and 4 are optimized. In this study, the mentioned parameters are optimized by using the ABC algorithm. The parameters that belong to the ANFIS structure used in training are seen in Figure 1. The number of parameters found in the structure of ANFIS and used in training is equal to the total number of antecedent and conclusion parameters. As mentioned earlier, the position of a food source represents a possible solution to the addressed problem. Therefore, a set of antecedent and conclusion parameters of ANFIS corresponds to a food source in the ABC algorithm. Thus, the ABC algorithm operates for finding the best food source around the hive or the best antecedent and conclusion parameter set in the search space. A representative figure showing the food source position is given in Figure 2.

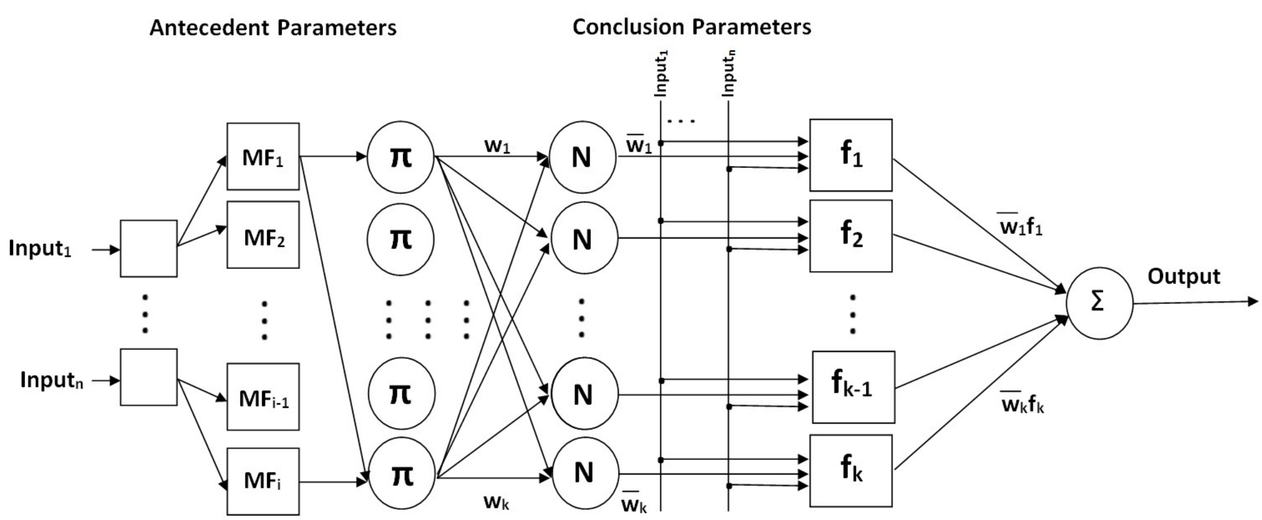

Figure 1. The antecedent and conclusion parts of ANFIS.

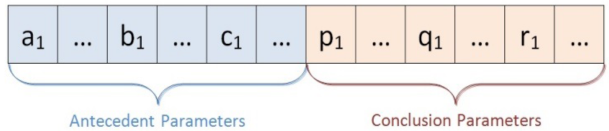

Figure 2. Representation of a solution (food source) in string form.

The nectar amount of a source shows the fitness of a possible solution. The RMSE function is used as a fitness function in order to calculate the fitness of a solution. In order to compute the RMSE error value, the output of ANFIS and its real output are used as given in Eq. (4):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \overline{y}_i)}{N}} \tag{4}$$

Here, $\overline{y}_i$ is the predicted and y_i is the known value and N is the number of samples. The RMSE error value between the output of ANFIS and the output of the nonlinear dynamic system must be minimum. Therefore, the most appropriate model is formed by optimizing the parameters of ANFIS with the ABC algorithm to obtain the lowest RMSE error value. The block diagram showing this structure is given in Figure 3. The number of antecedent parameters found in the structure is directly related to the input number of ANFIS, membership function type, and number. The number of antecedent parameters in the structure of ANFIS is calculated by using Eq. (5). The value of a_i depends on the type of membership function. For example, in the case of a generalized bell function (gbell), its value is 3, since the function is defined by 3 parameters. The number of conclusion parameters is directly related to the number of inputs and rules and can be calculated with Eqs. (6) and (7).

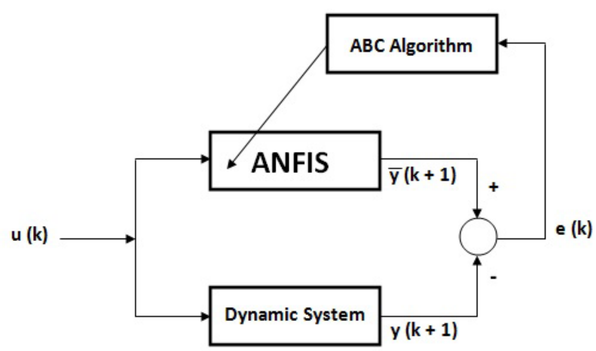

Figure 3. Block diagram of the system identification using ANFIS and ABC.

$$a_p = \sum_{i=1}^{n} a_i \tag{5}$$

$$nr = \prod_{i=1}^{k} A_i \tag{6}$$

$$cp = nr\,(k+1) \tag{7}$$

Here, nr represents the rule number, A_i is the membership function number used for the input i, and k is the input number. cp represents the total conclusion parameter number.

3. Simulation results

This section presents applications of ANFIS, trained by using the ABC algorithm, and their results. In applications, the identification of 4 nonlinear dynamic systems is carried out. The examples, inputs of ANFIS, and number of train/test data used are given in Table 1. Here, u(t) indicates the previous input and y(t) indicates the previous output. In Example 1, Eq. (8) is used to provide the previous input. Eq. (9) is utilized for Examples 2 and 3. A generalized bell function is selected as the membership function type in the applications. The value of the colony size, limit, and maximum cycle control parameters used for the ABC algorithm are 20, 250, and 5000, respectively. The parameter values given here are the most effective parameter set for the ABC algorithm. The results obtained are compared with different fuzzy-neuro-based models taken from [20,21].

Table 1. Nonlinear dynamic systems and parameter values used in training ANFIS.

Example	Inputs of ANFIS	Number of training/testing data	System
1	$u(k)$, $y_p(k)$	900/100	$y_p(k+1) = \frac{y_p(k)y_p(k-1)y_p(k-2)u(k-1)(y_p(k-2)-1)+u(k)}{1+y_p(k-1)^2+y_p(k-2)^2}$
2	$u(k)$, $y_p(k)$	900/100	$y_p(k+1) = 0.72y_p(k) + 0.025y_p(k-1)u(k-1) + +0.01u^2(k-2) + 0.2u(k-3)$
3	$u(k)$, $y_p(k)$, $y_p(k-1)$	200/50	$y_p(k+1) = \frac{y_p(k)y_p(k-1)(y_p(k)+2.5)}{1+y_p(k)^2+y_p(k-1)^2} + u(k)$
4	$y_p(k)$, $y_p(k-1)$	100/100	$y_p(k+1) = -1.4y_p(k)^2 + 0.3y_p(k-1) + 1.0$

The ABC algorithm is operated 30 times for each ANFIS structure, and, in each case, the algorithm starts with different initial populations. The error value in the application is calculated in RMSE, as described in Eq. (6).

$$u(k) = \begin{cases} \sin(\pi k/25) & k < 250 \\ 1 & 250 \le k < 500 \\ -1 & 500 \le k < 750 \\ 0.3\sin(\pi k/25) + 0.1\sin(\pi k/32) + 0.6\sin(\pi k/10) & 750 \le k < 1000 \end{cases} \tag{8}$$

$$u(k) = \sin(2\pi k/25) \tag{9}$$

In this study, different ANFIS structures are used for each example in the applications. The training and test results, obtained as RMSE_{Mean} and standard deviation (SD), are given in Table 2. In Examples 1, 2, and 4, different ANFIS structures having 4, 9, and 16 rules are used. In Example 3, rules 8 and 18 are utilized. The best result is obtained using 16 rules for Examples 1, 2, and 4. The best training error values are 0.0325, 0.0215, and 0.0294, respectively. The best test error values are 0.0270, 0.0186, and 0.0435. In Example 3, the best result is found utilizing 18 rules. The best training and test error values for Example 3 are 0.0174 and 0.0167, respectively .

Table 2. Simulations results obtained for Examples 1–4.

Example	Number of MFs for each input	Number of rules	Number of parameters	Train		Test	
				Mean (RMSE)	SD	Mean (RMSE)	SD
1	2	4	24	0.0386	0.0015	0.0281	0.0039
	3	9	45	0.0345	0.0019	0.0271	0.0026
	4	16	72	0.0325	0.0014	0.0270	0.0047
2	2	4	24	0.0244	0.0004	0.0223	0.0029
	3	9	45	0.0225	0.0009	0.0212	0.0017
	4	16	72	0.0215	0.0005	0.0186	0.0014
3	2-2-2	8	50	0.0224	0.0058	0.0219	0.0059
	3-3-2	18	96	0.0174	0.0039	0.0167	0.0039
4	2	4	24	0.0714	0.0112	0.0662	0.0134
	3	9	45	0.0374	0.0064	0.0478	0.0141
	4	16	72	0.0294	0.0044	0.0435	0.0104

As the number of rules and parameters in the ANFIS structure employed in Examples 1–4 increases, better training and test error values are obtained. It is suitable to use a less parameterized ANFIS structure in studies where speed and accuracy performance is in the forefront for solution to problems and to use a more parameterized ANFIS structure in cases where the accuracy performance result is more important.

Low SD values are obtained, particularly in Examples 1–3 and partially in Example 4. A low SD value shows that the majority of RMSE values found at different times are very close to each other. Additionally, low SD values show that the results obtained are reliable.

Another advantage provided by the training of ANFIS using the ABC algorithm is that test error values are as good as training error values. The training of ANFIS is conducted with a known dataset. In the testing of ANFIS, a different dataset, which was not used in the training of ANFIS, is used. In all applications, good results are obtained in the testing of ANFIS. This shows that learning and generalization are effective in the training of ANFIS conducted with the ABC algorithm. When the convergence graphic of RMSE, given in Figure 4, is evaluated, it is seen, for example, that the ABC algorithm is able to satisfactorily train approximately 1000 epochs. In addition, when 'the actual output-obtained ANFIS output' graphics shown in Figure 5 are analyzed, the obtained and actual outputs are observed to be close to each other.

Application results are compared with different approaches in Tables 3 and 4. Here, the number of parameters (NP), training, and test error values are given. The best test error values for Examples 1 and 2 are obtained with the method proposed. The lowest number of parameters is again found in the proposed method, as well as obtaining the best RMSE value. Obtaining good RMSE values in training and test results for these examples shows that effective training is realized by using the ABC algorithm. According to these results, the method proposed for Examples 1 and 2 is more successful. It is seen that the result obtained by using an ANFIS structure having 50 parameters for Example 3 is the best result after the GDFNN method. For Example 4, the best training error value is obtained using 72 parameters with the proposed method. When comparison results are examined for these four examples, it is concluded that the method proposed is quite successful in the identification of considered systems and training ANFIS.

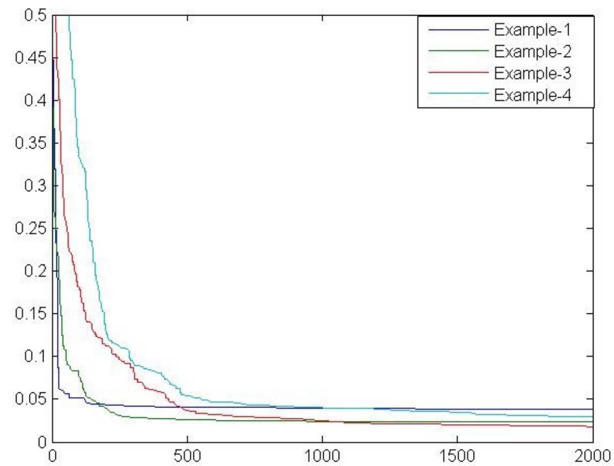

Figure 4. RMSE-epoch chart obtained for Examples 1–4.

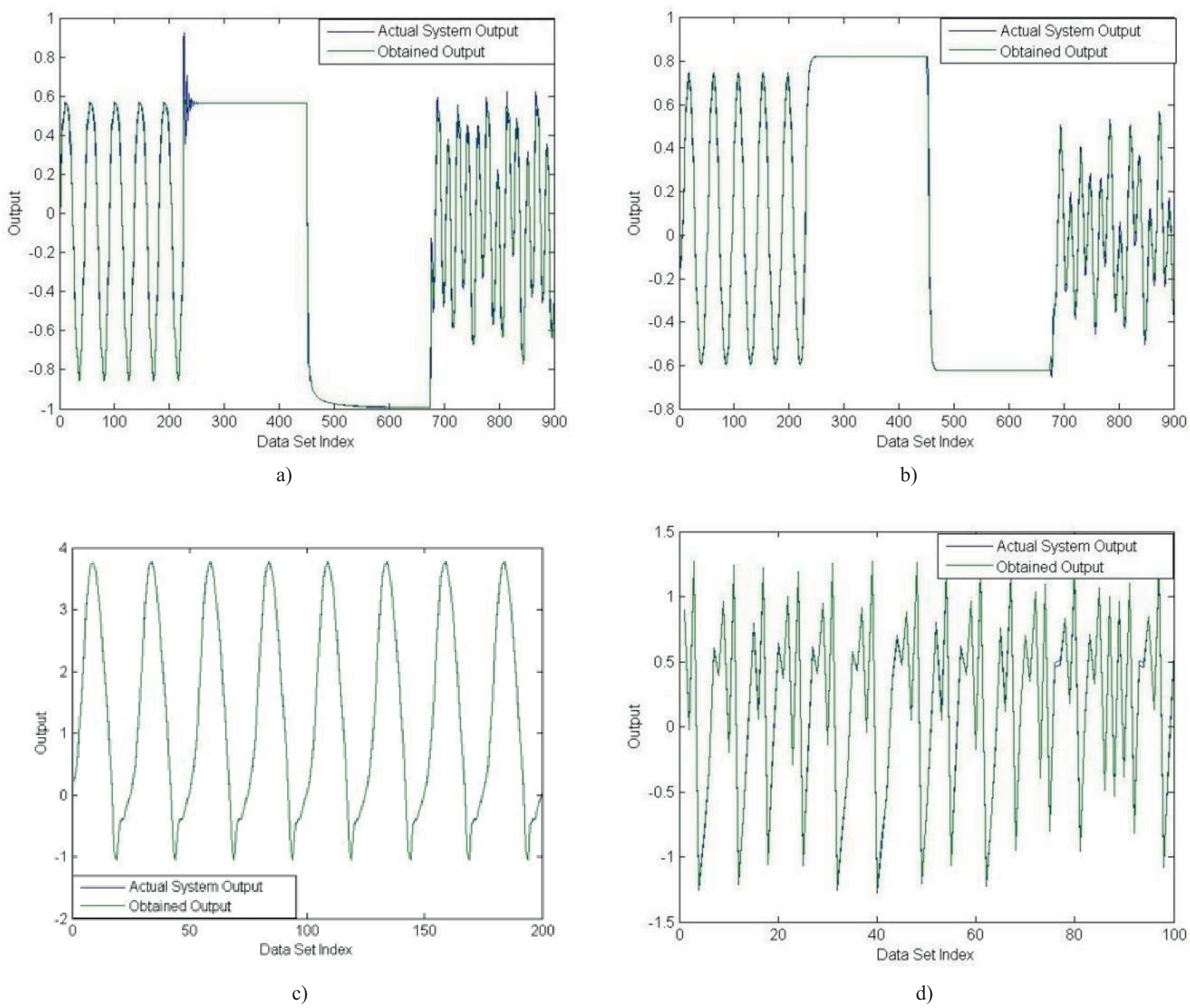

Figure 5. Actual system and obtained output chart for: a) Example 1; b) Example 2; c) Example 3; d) Example 4.

Table 3. Comparison of the proposed method with other algorithms for Examples 1 and 2 [20,21].

Algorithms	Example 1			Example 2		
	NP	Train	Test	NP	Train	Test
RSONFIN	36	0.0248	0.078	36	0.03	0.06
RFNN	112	0.114	0.0575	49	0.072	0.128
TRFN-S	33	0.0084	0.0346	33	0.0067	0.0313
DFNN	39	-	0.05	-	-	-
HO-RNFS	45	0.0542	0.0815	-	-	-
RSEFNN-LF$_{zero}$	34	0.0246	0.03	28	0.0221	0.0383
RSEFNN-LF$_{first}$	32	0.0199	0.0397	30	0.0156	0.0279
WRFNN	-	-	-	55	0.0574	0.083
ANFIS (PSO)	24	0.0415	0.0314	24	0.0246	0.0245
ANFIS (GA)	24	0.0413	0.0322	24	0.0252	0.0236
Proposed (ANFIS-ABC)	24	0.0386	0.0281	24	0.0244	0.0223

Table 4. Comparison of the proposed method with other algorithms for Examples 3 and 4 [20,21].

Algorithms	Example 3			Example 4		
	NP	Train	Test	NP	Train	Test
RFNN	-	-	-	60	0.463	0.469
TRFN-S	-	-	-	66	0.0362	0.032
RSEFNN-LF$_{zero}$	-	-	-	48	0.0568	0.0409
RSEFNN-LF$_{first}$	-	-	-	35	0.0320	0.0209
WRFNN	-	-	-	70	0.191	0.188
RBF-AFS	208	0.1384	-	-	-	-
OLS	326	0.0288	-	-	-	-
GDFNN	56	0.0108	-	-	-	-
FAOS-PFNN	25	0.0252	-	-	-	-
GOSFNN	72	0.0228	-	-	-	-
ANFIS (PSO)	50	0.0553	0.0531	72	0.0352	0.0488
ANFIS (GA)	50	0.0367	0.0382	72	0.0387	0.0498
Proposed (ANFIS-ABC)	50	0.0224	0.0219	72	0.0294	0.0435

4. Conclusion

The training of ANFIS is accepted as a difficult problem; it is also known that the ABC algorithm is quite successful in solving difficult problems. In the light of this information, the ABC algorithm is used for the training of ANFIS in this study. In this study, nonlinear dynamic systems are used for performance analysis and these systems are modeled and identified with ANFIS.

The optimization process of all parameters belonging to the ANFIS structure for the modeling of nonlinear dynamic systems is conducted with the ABC algorithm. The results obtained are compared with known fuzzy neural network, neural network, and ANFIS-based methods, which are used frequently in identification studies of nonlinear systems. When comparison results are examined, it is observed that the ABC algorithm is quite successful in the training of ANFIS. Nevertheless, it is also seen that it can be used in the modeling of many systems, notably in the identification of nonlinear systems. In this study, a simple structure is created by using a basic ANFIS structure and the ABC algorithm. The advantages of the method suggested are shown by comparing this model with different neural network and fuzzy neural network-based models that have a more complicated structure.

This study is important because it is the first study in the literature in which ANFIS training is performed by using the ABC algorithm and nonlinear systems are identified with this method. Moreover, this study is an example for future ANFIS-based studies, because it uses different heuristic algorithms, apart from genetic algorithms and PSO algorithms.

References

[1] Jang JSR. ANFIS: Adaptive-network-based fuzzy inference system. IEEE T Syst Man Cyb 1993; 23: 665-685.

[2] Jang JSR, Sun CT, Mizutani E. Neuro fuzzy and soft computing a computational approach to learning and machine intelligence. IEEE T Automat Contr 1997; 42: 1482-1484.

[3] Ho WH, Chen JX, Lee I, Su HC. An ANFIS-based model for predicting adequacy of vancomycin regimen using improved genetic algorithm. Expert Syst Appl 2011; 38: 13050-13056.

[4] Nariman-Zadeh N, Darvizeh A, Dadfarmai MH. Design of ANFIS networks using hybrid genetic and SVD methods for the modeling of explosive cutting process. J Mater Process Tech 2004; 155: 1415-1421.

[5] Chen MY. A hybrid ANFIS model for business failure prediction utilizing particle swarm optimization and subtractive clustering. Inform Sci 2013; 220: 180-195.

[6] Shoorehdeli MA, Teshnehlab M, Sedigh AK. Training ANFIS as an identifier with intelligent hybrid stable learning algorithm based on particle swarm optimization and extended Kalman filter. Fuzzy Set Syst 2009; 160: 922-948.

[7] Shoorehdeli MA, Teshnehlab M, Sedigh AK, Khanesar MA. Identification using ANFIS with intelligent hybrid stable learning algorithm approaches and stability analysis of training methods. Appl Soft Comput 2009; 9: 833-850.

[8] Jalali-Heravi M, Asadollahi-Baboli M. Quantitative structure-activity relationship study of serotonin (5-HT7) receptor inhibitors using modified ant colony algorithm and adaptive neuro-fuzzy interference system (ANFIS). Eur J Med Chem 2009; 44: 1463-1470.

[9] Khazraee SM, Jahanmiri AH, Ghorayshi SA. Model reduction and optimization of reactive batch distillation based on the adaptive neuro-fuzzy inference system and differential evolution. Neural Comput Appl 2011; 20: 239-248.

[10] Priyadharsini SS, Rajan ES, Sheniha FS. A novel approach for the elimination of artefacts from EEG signals employing an improved artificial immune system algorithm. J Exp Theor Artif In 2016; 28: 239-259.

[11] Karaboga D, Kaya E. Training ANFIS using artificial bee colony algorithm. In: 2013 IEEE International Symposium on Innovations in Intelligent Systems and Applications; 19–21 June 2013; Albena, Bulgaria. New York, NY, USA: IEEE. pp. 1-5.

[12] Karaboga D, Kaya E. Training ANFIS using artificial bee colony algorithm for nonlinear dynamic systems identification. In: IEEE 2014 Conference on Signal Processing and Communications Applications; 23–25 April 2014; Trabzon, Turkey. New York, NY, USA: IEEE. pp. 493-496.

[13] Karaboga D. An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report-TR06. Kayseri, Turkey: Erciyes University, 2005.

[14] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Global Optim 2007; 39: 459-471.

[15] Kockanat S, Karaboga N. The design approaches of two-dimensional digital filters based on metaheuristic optimization algorithms: a review of the literature. Artif Intell Rev 2015; 44: 265-287.

[16] Kockanat S, Karaboga N. A novel 2D-ABC adaptive filter algorithm: a comparative study. Digit Signal Process 2015; 40: 140-153.

[17] Akay B, Karaboga D. A survey on the applications of artificial bee colony in signal, image, and video processing. SIViP 2015; 9: 967-990.

[18] Karaboga D, Gorkemli B, Ozturk C, Karaboga N. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artif Intell Rev 2014; 42: 21-57.

[19] Li B, Li Y, Gong L. Protein secondary structure optimization using an improved artificial bee colony algorithm based on AB off-lattice model. Eng Appl Artif Intel 2014; 27: 70-79.

[20] Juang CF, Lin YY, Tu CC. A recurrent self-evolving fuzzy neural network with local feedbacks and its application to dynamic system processing. Fuzzy Set Syst 2010; 161: 2552-2568.

[21] Ning W, Yue T, Shao-Man L. A generalized online self-organizing fuzzy neural network for nonlinear dynamic system identification. In: Proceedings of the 30th Chinese Control Conference; 22–24 July 2011; Yantai, China. New York, NY, USA: IEEE. pp. 2879-2883.