International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Training of the Artificial Neural Networks using States of Matter Search Algorithm

Saban GULCU[1]

Submitted: 23/06/2020 Accepted: 26/08/2020

Abstract: In recent years, technology has been developing very rapidly in the field of artificial intelligence. In this development, Artificial neural networks (ANNs) have taken a huge place. The human brain has an excellent understanding structure. The brain makes this understanding through neuron cells. ANN aims to solve some complex problems by establishing the perception structure of human over neurons in the computer environment. A multilayer perceptron (MLP) is a class of artificial neural networks. MLP has the ability to learn using inputs and expected outputs. In order to do this, weight values in MLP are constantly updated according to the inputs and expected outputs. Thus, weight values are tried to be kept at an optimum level. Therefore, this problem is an optimization problem. In this study, the State of Matter Search meta-heuristic algorithm was used to optimize the weight values in MLP, called SMS-MLP. In the experiments, five classification datasets (xor, balloon, iris, breast cancer, heart) were used. The SMS-MLP algorithm was compared with the previous six algorithms (GWO-MLP, ACO-MLP, GA-MLP, PBIL-MLP, PSO-MLP and ES-MLP) in the literature. The experimental study shows that the SMS-MLP algorithm is more efficient than the other six algorithms.

Keywords: *Artificial neural networks, training of artificial neural networks, states of matter search algorithm, optimization, feed-forward artificial neural networks.*

1. Introduction

In recent years, technology has been developing very rapidly in the field of artificial intelligence. In this development, Artificial neural networks (ANN) have taken a huge place. In 1943, the first neural network model was proposed by Warren McCullosh, a neural doctor, and Walter Pitts, a mathematician. They named the neural network which is a simple electrical circuit modelled by inspiring the processing ability of the human brain [1]. ANN is the system that can teach the processing and learning structure of the human brain to a computer according to certain rules. Table 1 shows the elements of the biological nervous system and their counterparts in ANN. ANN has been actively used in many different areas. Civil engineering [2], banking [3], chemistry [4], medicine [5], energy [6, 7] are some of these fields.

Table 1. Biological nervous system and their counterparts in ANN

Biological nervous system	ANN system
Neuron	Processor element
Dendrite	Aggregation function
Cell body	Transfer function
Axons	Output of the processor element
Synapses	Weights

A multilayer perceptron (MLP) is a class of feed-forward artificial neural network. MLP has the ability to learn using inputs and expected outputs. In order to do this, the weight values are constantly updated according to inputs and expected outputs. Thus, the weight values are tried to be kept at an optimum level. Therefore, this problem is an optimization problem. In this study, the State of Matter Search meta-heuristic algorithm (SMS) is used to optimize the weight values in MLP.

There are many studies on the training of ANN using meta-heuristic algorithms. Jaddi, Abdullah and Hamdan [8] optimize the architecture and weights of the ANN with the improved bat optimization algorithm to increase the success of the ANN algorithm. They combine the bat algorithm with local search algorithms and population-based algorithms. To adjust the parameters of the developed method, they also use the Taguchi method. They use two time series and 6 classifications datasets to demonstrate the success of the method. Besides, they apply the proposed algorithm to the real-world problem, the prediction of future values of precipitation dataset. Askarzadeh and Rezazadeh [9] define finding the optimum weights of ANN as a complex continuous optimization problem. They solve this problem using the bird mating optimization algorithm [10]. In [11], the particle swarm optimization method is used for ANN training. In the experimental studies, 4 datasets from the UCI machine learning database are used. In [12], the weights of ANN are optimized by the kidney optimization algorithm to increase the success of the ANN algorithm. The rain precipitation prediction problem is solved using the developed method. Piotrowski et al [13] optimize the weights of ANN using the different meta-heuristic algorithms and solve the temperature prediction problem using the developed method. Mirjalili [14] uses the Grey Wolf Optimizer algorithm for training MLP, called GWO-MLP. The developed algorithm is compared with some of the most well-known meta-heuristic trainers: Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Ant Colony Optimization (ACO), Evolution Strategy (ES) and Population-based Incremental Learning (PBIL). According to the experimental results, the GWO-MLP algorithm has a high level accuracy in classification.

[1] *Computer Eng. Dept., Necmettin Erbakan Unv., Konya, Turkey*
ORCID ID: 0000-0001-7714-8861
** Corresponding Author Email: sgulcu@erbakan.edu.tr*

In this study, the optimum weights of the ANN algorithm were found using the SMS algorithm. The SMS algorithm [15] developed in 2014 by Cuevas, Echavarría and Ramírez-Ortegón is a meta-heuristic algorithm. Its inspiration is by the states of matter. In the experiments, xor, balloon, iris, breast cancer, heart datasets were used. Balloon, iris, breast cancer, heart datasets are taken from the public UCI database (https://archive.ics.uci.edu/ml/index.php).

The sections of the article is as follows. In Section 1, the history of the ANN is briefly explained and the literature review is given about the meta-heuristic algorithms and training of the ANN. Section 2 presents the ANN, the SMS algorithm and the training of Artificial Neural Networks using the SMS algorithm. Section 3 presents the experimental results and analysis of the developed method on five datasets. Finally, the article is concluded in Section 4.

2. Material and Method

2.1. Artificial Neural Networks

Today, there are different ANN models developed for a specific purpose. Some of these are feed-forward neural networks, feed backward neural networks, SOM self-organizing maps neural networks. A multilayer perceptron (MLP) is a class of feed-forward artificial neural network. MLP can contain one or more hidden layers. In this study, MLP is used and its structure is shown in Fig. 1. The task of MLP is to take the information given to it as an input value and to produce outputs thanks to the weight values, bias values and activation functions. The artificial neural network has one input layer, one or more hidden layers and one output layer. The layers are made up of artificial nerve cells (neurons). As seen in Fig. 1, the neurons are connected to each other by the weights. An exemplary artificial nerve cell is shown in Fig. 2. The total input value of a neuron is calculated by Equation (1). Then, the output value of the neuron is calculated by passing this *net* value through an activation function [16].

$$net = \sum_{i=1}^{m} w_i x_i + b \quad (1)$$

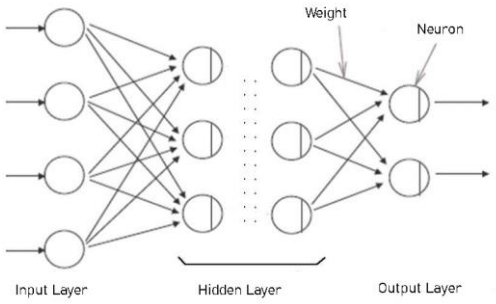


Fig. 1 The structure of a multilayer perceptron [17]

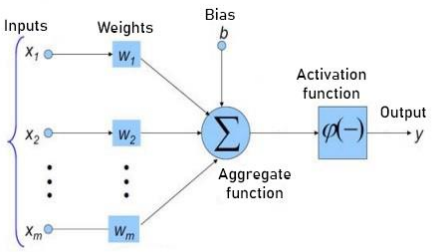


Fig. 2 An artificial neuron [18]

2.2. State of Matter Search Algorithm (SMS)

The State of Matter Search (SMS) algorithm [15] developed in 2014 by Cuevas, Echavarría and Ramírez-Ortegón is a meta-heuristic algorithm. Its inspiration is by the states of matter. In the SMS algorithm, individuals mimic molecules that interact with each other using the evolutionary operations based on the laws of physics. These operations increase the diversity of population and they reduce the number of particles trapped into the local optima. The evolutionary process in the algorithm is divided into 3 levels and the levels mimic 3 states of matter: gas, liquid and solid. The individuals behave differently in each state. The algorithm intensifies the search beginning with the gas level until it reaches the solid-state. Thus, it maintains the balance between the global and local search.

The substance is present in nature in 3 states. These are the gas, liquid and solid states. The characteristic distinctions between these 3 states are the applied forces between the particles (molecules) that make up a material [19]. Fig. 3 shows the movements of the particles in the different states of a matter. In the gas state, adjacent molecules have a bigger distance between them than the size of the molecules. Therefore, the effect of the forces between molecules is small. As seen in Fig. 3, particles in the gas state can move to the distant areas and the maximum permitted displacement between these particles is represented by ρ_1 [20]. In liquid form, the forces between molecules are more restrictive than the forces between molecules in gas form. Fig. 3 shows the movement ρ_2 of a particle in liquid form. The motion of a particle in the liquid form is smaller than that of a particle in the gas form but greater than that of a particle in the solid form [21]. In the solid-state, the particles look like a packaged structure. So the particles cannot freely move, they just vibrate. Therefore, the movement ρ_3 between the particles is minimal.

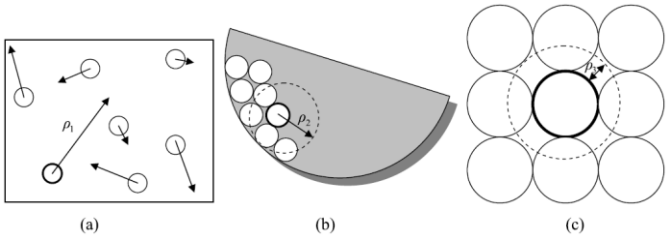


Fig. 3 The movement of particles (a) gas, (b) liquid, (c) solid [15].

The velocity and direction of molecular movements are determined by collision applied by molecule sets, force and random events [22]. To implement these behaviours in the algorithm, three operators are used in the SMS: direction vector, collision and random positions.

The direction vector operator is used to change the position of the molecule. A direction vector $\mathbf{d}_i$ with *n* dimension is assigned for a molecule $\mathbf{p}_i$ with *n* dimension. This vector controls the motion of the particle. The initial values of all direction vectors $\mathbf{D} = \{\mathbf{d_1}, \mathbf{d_2}, \ldots, \mathbf{d}_{N_P}\}$ are generated randomly between -1 and 1. N_P represents the population size.

The new direction of an individual is calculated by Equation (2).

$$\boldsymbol{d_i^{k+1}} = \boldsymbol{d_i^k}\, x \left(1 - \frac{k}{gen}\right) x\, 0.5 + \boldsymbol{a_i} \quad (2)$$

where $\boldsymbol{a_i}$ is a vector and it is calculated by $\boldsymbol{a_i} = (\boldsymbol{p}^{best} - \boldsymbol{p}_i)/\left\|\boldsymbol{p}^{best} - \boldsymbol{p}_i\right\|$.

To calculate the new position of a molecule, the velocity $\boldsymbol{v}_i$ of each molecule is firstly calculated by Equation (3) and Equation

(4).

$$\boldsymbol{v}_i = \boldsymbol{d}_i \; x \; v_{init} \tag{3}$$

$$v_{init} = \frac{\sum_{j=1}^{n}(b_j^{high} - b_j^{low})}{n} \; x \; \beta \tag{4}$$

Where v_{init} is the initial velocity, b_i^{high} and b_i^{low} are the upper bound and the lower bound, respectively. β is a value between 0 and 1. Then using Equation (5), the positions of the individuals are updated.

$$p_{i,j}^{k+1} = p_{i,j}^{k} + \; v_{i,j} \; x \; rand(0,1) \; x \; \rho \; x \left(b_j^{high} - b_j^{low}\right) \tag{5}$$

The collision operator simulates a collision taking place by molecules interacting with each other. If the range between 2 individuals is less than a specified value, a collision occurs. Hence, if $\|\boldsymbol{p}_i - \boldsymbol{p}_a\| < r$, then a collision is assumed between the molecules *i* and *q*. If the collision occurs, the direction vectors of the colliding particles are replaced as seen in Equation (6).

$$\boldsymbol{d}_i = \boldsymbol{d}_q \text{ and } \boldsymbol{d}_q = \boldsymbol{d}_i \tag{6}$$

Equation (7) shows the radius of the collision.

$$r = \frac{\sum_{j=1}^{n}(b_j^{high} - b_j^{low})}{n} \; x \; \alpha \tag{7}$$

where α is a value between 0 and 1.
At the end of this process, each individual has a *r* value and the collisions occur among individuals. The collision operators and *r* value control variety during the search process. Namely, the rate of increasing or decreasing variety is predetermined for each level.
To simulate the random behaviours of the individuals, the SMS algorithm creates random positions within the search space by Equation (8). Firstly, the random value r_m is generated in the range [0, 1]. If r_m is less than a limit value *H*, the random position of the individual is recreated, otherwise the position stays as it is.

$$p_{i,j}^{k+1} \begin{cases} b_j^{low} + rand(0,1) \; x \left(b_j^{high} - b_j^{low}\right) & \text{with probability } H \\ p_{i,j}^{k+1} & \text{otherwise} \end{cases} \tag{8}$$

where $i \in \{1, \ldots, N_p\}$ and $j \in \{1, \ldots, n\}$.
The flowchart of the SMS algorithm is presented in [15].

2.3. Training ANN using the SMS algorithm (SMS-MLP)

ANN has the ability to learn using inputs and expected outputs. In order to do this, the weight values are constantly updated according to inputs and expected outputs. Thus, the weight values are tried to be kept at an optimum level. In this study, the SMS meta-heuristic algorithm is used to optimize the weight values in ANN. Fig. 4 shows this process.

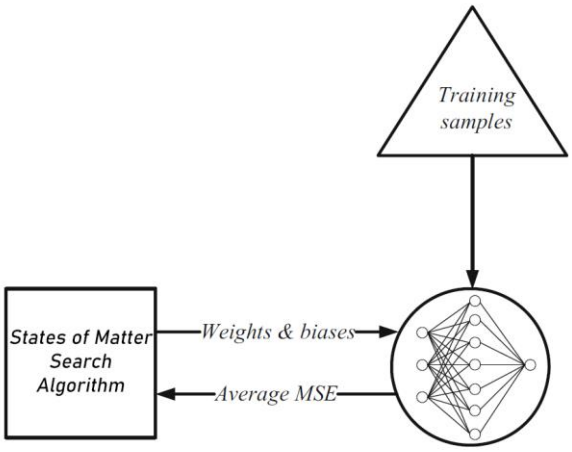


Fig. 4 The process of the optimization of the weights in ANN

To optimize the weights of ANN using the SMS algorithm, these weights must firstly be represented by the particles in the SMS algorithm. For this purpose, the representation vector consisting of the values of the weights and biases is used. This vector is shown in Equation (9). The SMS algorithm updates the weights at each iteration using the training dataset and uses the Mean Square Error (MSE) value obtained from ANN for the fitness value. The aim of the SMS-MLP algorithm is to minimize the MSE value. The MSE value is calculated by Equation (10).

$$Vector = \{w_{i,j}, \theta_k\} \tag{9}$$

where $w_{i,i}$ represents the weight between the neuron *i* and neuron *j*. θ_k represents the bias coming to the neuron *k*.

$$MSE = \sum_{i=1}^{N} \frac{\sum_{k=1}^{m}(g_k^i - t_k^i)^2}{N} \tag{10}$$

where *N* represents the number of the training instances and *m* represents the number of the neurons in the output layer. g_k^i and t_k^i are respectively the real output value and the estimated output value.

3. Experimental Results

In the experiments, 5 different datasets were used: xor, balloon, iris, breast cancer, heart. Balloon, iris, breast cancer and heart datasets are taken from the publicly available UCI database (https://archive.ics.uci.edu/ml/index.php). Table 2 shows the properties of these datasets, the vector size and the ANN architecture. The number of attributes and classes in the datasets affects the architecture of ANN. The number of the attributes in the dataset is equal to the number of the neurons in the input layer of the ANN architecture. The number of the neurons in the output layer of the ANN architecture is determined by the number of the classes in the dataset. The ANN architecture determines the size of the representation vector in the SMS-MLP algorithm. The size of the representation vector used for each dataset is shown in Table 2. The numbers of the instances used in training and test are presented in Table 2. The numbers of the instances are the same as the numbers of the instances in [14] to fairly compare the results of the algorithms. Besides, the sigmoid function shown in Equation (11) was used in the ANN architecture as the activation function.

$$S(x) = \frac{1}{1+e^{-x}} \quad (11)$$

The experiments were carried out on the computer with the Windows 10 operating system, i5 3.0 GHz processor, 4 GB memory. The parameters of the SMS-MLP algorithm are presented in Table 3. The initial values of the weights and biases were randomly generated between [-10, 10]. For each dataset, the algorithm was run 30 times independently. The average classification rate, the average and standard deviation of the MSE values and the average training time, which are the results of the experiments, are presented in Table 4. Also, the convergence graph of the average MSE values of the SMS-MLP algorithm is presented in Fig. 5 for each dataset.

The algorithm was tested on the test datasets and the average classification rate is given in Table 4. When Table 4 is examined, the highest classification rate was achieved with 99.17% on the balloon dataset. The lowest classification rate was obtained with 68.57% on the heart dataset. The classification rates on other datasets are above 85% and these rates can be considered successful.

When the results are examined in terms of average training time, the training time of the SMS-MLP on the xor dataset is 5.5 seconds. The problem with the longest training time is the breast cancer dataset. The SMS-MLP algorithm solved the breast cancer dataset in 2830.3 seconds. Furthermore, when the convergence graphs of the SMS-MLP on the datasets are examined in Fig. 5, it is seen that the SMS-MLP algorithm tries to reach the global optimum successfully during the iterations.

Equations (12)-(15) show the performance metrics (sensitivity, specificity, precision and F_1-score) based on true positive (TP), true negative (TN), false positive (FP) and false negative (FN). The sensitivity metric measures the ratio of correctly classified positive instances over all instances belonging to the positive class. The specificity metric measures the ratio of correctly classified negative instances over all instances belonging to the negative class. The precision metric measures the ratio of correctly classified positive instances over all instances being classified as positive. The F_1-score determines the balance between sensitivity and precision [23, 24]. The results of the performance metrics for each dataset are presented in Table 5. According to the results, there is a balance between sensitivity and specificity on xor, balloon and iris datasets. But the algorithm has a low sensitivity percentage and a high specificity percentage on the breast cancer dataset. On the other hand, the algorithm has a high sensitivity percentage and a low specificity percentage on the heart dataset. The algorithm has a high precision percentage on all the datasets. Overall, the SMS-MLP algorithm is successful according to the performance metrics.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (12)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (13)$$

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (14)$$

$$F_1 - score = 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \quad (15)$$

The proposed SMS-MLP algorithm is compared with the previous six algorithms (GWO-MLP, ACO-MLP, GA-MLP, PBIL-MLP, PSO-MLP and ES-MLP) in the literature to demonstrate the performance of the SMS-MLP algorithm. The GWO-MLP, ACO-MLP, GA-MLP, PBIL-MLP, PSO-MLP and ES-MLP algorithms are based on the Grey Wolf Optimization, Ant Colony Optimization (ACO), Genetic Algorithm (GA), Population-based Incremental Learning (PBIL), Particle Swarm Optimization (PSO) and Evolution Strategy (ES), respectively. The results of these six algorithms are taken from reference [14]. The parameters of these six algorithms are presented in [14] and the parameters of the SMS-MLP are the same as in [14] to compare in a fair situation. Namely, the population size for xor and balloon datasets is 50 in the GWO-MLP algorithm and also in the others. The population size for other datasets is 200 in the GWO-MLP algorithm and also in the others. The maximum number of iterations is also the same, namely 250. The initial values of the biases and weights are randomly generated between [-10, 10]. Table 6 shows the comparison results of the algorithms according to the best classification accuracy. As seen in the table, the SMS-MLP has the best results on 4 datasets and the GWO-MLP has the best results on 3 datasets. Also, the GA-MLP has the best results on 2 datasets and the others have the best results on only 1 dataset. The experimental results show that the SMS-MLP algorithm is more efficient than the other six algorithms.

Table 2. Properties of the datasets

Dataset	Attribute number	Training	Test	Class number	ANN architecture	Vector size
Xor	3	8	8	2	3-7-1	36
Balloon	4	20	20	2	4-9-1	55
Iris	4	150	150	3	4-9-3	75
Breast Cancer	9	599	100	2	9-19-1	210
Heart	22	80	187	2	22-45-1	1081

Table 3. Parameters of the SMS-MLP algorithm

Parameter	Value
α, β, H, ρ (gas state)	0.3, 0.9, 0.9, 0.85
α, β, H, ρ (liquid state)	0.05, 0.5, 0.2, 0.35
α, β, H, ρ (solid state)	0, 0.1, 0, 0.1
Population size	50 for xor and balloon, 200 for the other datasets
Maximum number of iterations	250

Table 4. The results of the SMS-MLP algorithm on datasets

Dataset	MSE (Avg±Std)	Classification Rate (%)	Training Time (s)
Xor	1.33E-01±3.39E-02	83.75	5.5
Balloon	1.11E-02±1.31E-02	99.17	20.2
Iris	2.58E-01±5.12E-02	86.78	881.6
Breast Cancer	2.27E-02±4.33E-03	85.67	2830.3
Heart	1.18E-01±3.13E-02	68.57	892.4

Table 5. The results of the performance metrics

Dataset	Sensitivity	Specificity	Precision	F_1-score
Xor	81.7%	86.7%	88.9%	0.8349
Balloon	99.2%	99.2%	98.8%	0.9894
Iris	82.1%	91.1%	81.0%	0.8002
Breast Cancer	50.2%	95.1%	83.7%	0.5828
Heart	76.2%	55.6%	82.5%	0.7856

Table 6. The comparison of the algorithms

Dataset	Best Classification Accuracy (%)						
	SMS-MLP	GWO-MLP	PSO-MLP	GA-MLP	ACO-MLP	ES-MLP	PBIL-MLP
Xor	**100.00**	**100.00**	37.50	**100.00**	62.50	62.50	62.50
Balloon	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
Iris	**93.33**	91.33	37.33	89.33	32.66	46.66	86.66
Breast Cancer	93.00	**99.00**	11.00	98.00	40.00	06.00	07.00
Heart	**77.54**	75.00	68.75	58.75	00.00	71.25	45.00

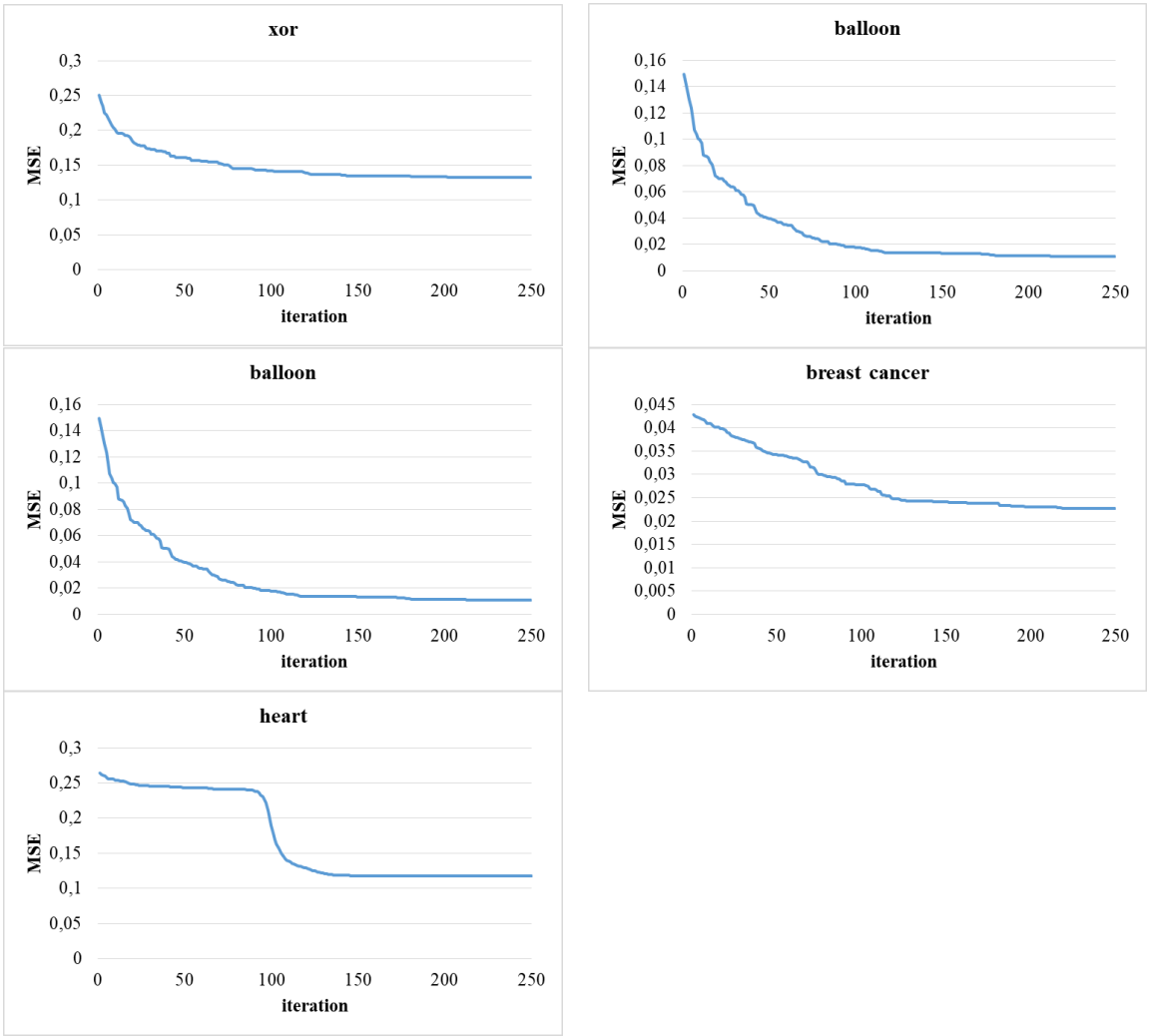


Fig 5. The convergence graphs of the SMS-MLP on the datasets

4. Conclusions

In this study, the ANN algorithm is trained using the State of Matter Search algorithm (SMS). The SMS algorithm is a meta-heuristic algorithm inspired by the states of matter. In the SMS algorithm, the individuals mimic molecules that interact with each other using evolutionary operations based on the laws of physics. These operations increase the diversity of population and reduce the number of particles trapped into the local optima. The evolutionary process in the algorithm is divided into 3 levels and the levels mimic the states of matter: gas, liquid and solid. The individuals behave differently in each state.

The ANN has the ability to learn using inputs and expected outputs. In order to do this, the weight values are constantly updated according to inputs and expected outputs. Thus, the weight values are tried to be kept at an optimum level. In this study, the proposed SMS-MLP algorithm optimizes the weights of the ANN using the SMS algorithm. Firstly, these weights must be represented by molecules in the SMS algorithm. For this purpose, the representation vector consisting of the values of the weights and biases is used. In the SMS-MLP algorithm, the vector is updated on each iteration using the training dataset and the SMS-MLP algorithm uses the MSE fitness value to evaluate the suitability of the molecules. At the end of the SMS-MLP algorithm, the most appropriate values for the training dataset are assigned to the weights.

In the experiments, 5 different datasets (xor, balloon, iris, breast cancer, heart) were used. Balloon, iris, breast cancer and heart datasets are taken from the public UCI database. The SMS-MLP algorithm was compared with 6 algorithms (GWO-MLP, ACO-MLP, GA-MLP, PBIL-MLP, PSO-MLP and ES-MLP) in the literature. The experimental results have shown that the SMS-MLP algorithm has a better performance than the other algorithms. As a conclusion, the SMS algorithm can be used successfully with ANN.

As a future study, the SMS-MLP method can be applied to different datasets. In addition, hybrid methods can be used to increase the success of the SMS-MLP algorithm.

References

[1] K. Öztürk, M.E. Şahin, Yapay Sinir Ağları ve Yapay Zekâ'ya Genel Bir Bakış, Takvim-i Vekayi, 6 25-36.

[2] E. Madenci, Ş. Gülcü, Optimization of flexure stiffness of FGM beams via artificial neural networks by mixed FEM, Structural Engineering and Mechanics, 75 (2020) 633-642.

[3] E. Kaya, A. Yasar, I. Saritas, Banknote classification using artificial neural network approach, International Journal of Intelligent Systems and Applications in Engineering, 4 (2016) 16-19.

[4] A. Tümer, S. Edebali, Ş. Gülcü, Modeling of Removal of Chromium (VI) from Aqueous Solutions Using Artificial Neural Network, Iranian Journal of Chemistry and Chemical Engineering (IJCCE), 39 (2020) 163-175.

[5] O. İnan, A.E. Tümer, S. Koçer, Ş. Gülcü, Diagnosis of lung cancer disease using artificial neural networks, in: 4th International conference on computational and experimental science and engineering (iccesen-2017), Antalya, Turkey, 2017.

[6] D.-K. Bui, T.N. Nguyen, T.D. Ngo, H. Nguyen-Xuan, An artificial neural network (ANN) expert system enhanced with the electromagnetism-based firefly algorithm (EFA) for predicting the energy consumption in buildings, Energy, 190 (2020) 116370.

[7] M. Kim, S. Jung, J.-w. Kang, Artificial Neural Network-Based Residential Energy Consumption Prediction Models Considering Residential Building Information and User Features in South Korea, Sustainability, 12 (2020) 109.

[8] N.S. Jaddi, S. Abdullah, A.R. Hamdan, Optimization of neural network model using modified bat-inspired algorithm, Applied Soft Computing, 37 (2015) 71-86.

[9] A. Askarzadeh, A. Rezazadeh, Artificial neural network training using a new efficient optimization algorithm, Applied Soft Computing, 13 (2013) 1206-1213.

[10] A. Askarzadeh, Bird mating optimizer: an optimization algorithm inspired by bird mating strategies, Communications in Nonlinear Science and Numerical Simulation, 19 (2014) 1213-1228.

[11] F. Erdoğan, Ş. Gülcü, Training of Artificial Neural Networks using Meta Heuristic Algorithms, in: The International Aluminium-Themed Engineering and Natural Sciences Conference (IATENS19), Konya, Turkey, 2019, pp. 124-128.

[12] N.S. Jaddi, S. Abdullah, Optimization of neural network using kidney-inspired algorithm with control of filtration rate and chaotic map for real-world rainfall forecasting, Engineering Applications of Artificial Intelligence, 67 (2018) 246-259.

[13] A.P. Piotrowski, M. Osuch, M.J. Napiorkowski, P.M. Rowinski, J.J. Napiorkowski, Comparing large number of metaheuristics for artificial neural networks training to predict water temperature in a natural river, Computers & Geosciences, 64 (2014) 136-151.

[14] S. Mirjalili, How effective is the Grey Wolf optimizer in training multi-layer perceptrons, Applied Intelligence, 43 (2015) 150-161.

[15] E. Cuevas, A. Echavarría, M.A. Ramírez-Ortegón, An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation, Applied intelligence, 40 (2014) 256-272.

[16] O. Kaynar, S. Taştan, Zaman serisi analizinde mlp yapay sinir ağları ve arıma modelinin karşılaştırılması, Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, (2009) 161-172.

[17] A. Küçükyağlıoğlu, Ş. Gülcü, Training of Artificial Neural Network using Moth-Flame Optimization Algorithm, in: The International Aluminium-Themed Engineering and Natural Sciences Conference (IATENS19), Konya, Turkey, 2019, pp. 129-134.

[18] M.F. Keskenler, E.F. Keskenler, Geçmişten günümüze yapay sinir ağları ve tarihçesi, Takvim-i Vekayi, 5 (2017) 8-18.

[19] M.G. Ceruti, S.H. Rubin, Infodynamics: Analogical analysis of states of matter and information, Information Sciences, 177 (2007) 969-987.

[20] D. Chowdhury, D. Stauffer, Principles of equilibrium statistical mechanics, Principles of Equilibrium Statistical Mechanics, by Debashish Chowdhury, Dietrich Stauffer, pp. 564. ISBN 3-527-40300-0. Wiley-VCH, September 2000., (2000) 564.

[21] D.S. Betts, R.E. Turner, Introductory statistical mechanics, Addison-Wesley, 1992.

[22] Y. Cengel, M. Boles, An Engineering Approach Thermodynamics, McGrow Hill, (2005).

[23] D.L. García, À. Nebot, A. Vellido, Intelligent data analysis approaches to churn as a business problem: a survey, Knowledge and Information Systems, 51 (2017) 719-774.

[24] T. Rong, H. Gong, W.W. Ng, Stochastic sensitivity oversampling technique for imbalanced data, International conference on machine learning and cybernetics, Springer, 2014, pp. 161-171.