Artificial Neural Network Weights Optimization based on Imperialist Competitive Algorithm

Maryam, Tayefeh Mahmoudi

[1]School of Electrical and Computer Eng., University of Tehran,
[2]Knowledge Engineering & Intelligent Systems Group, Iran Telecom Research Center
Tehran,Iran
Mahmodi@itrc.ac.ir

Nafiseh, Forouzideh

Faculty of Engineering, Department of Computer Eng, Azad University, Central Tehran Branch
Tehran, Iran
n.forouzideh@gmail.com

Caro, Lucas

Control and Intelligent Processing Lab, University of Tehran,
Tehran, Iran
lucas@ipm.ir

Fattaneh, Taghiyareh

School of Electrical and Computer Eng. , College of Engineering, University of Tehran,
Tehran, Iran
ftaghiyar@ut.ac.ir

ABSTRACT

Imperialist Competitive Algorithm (ICA) is a new socio-politically motivated global search strategy that has recently been introduced for dealing with different optimization tasks In this paper, we adopt ICA to optimize the weights of Multilayer Perceptron (MLP) Artificial Neural Network to solve premature convergence problem of Genetic Algorithm and some other similar algorithms. For this purpose, ICA is applied on four known datasets (WINE, GLASS, PIMA, WDBC) which are used for classification problems and compared with three other training methods. In almost all datasets, the proposed method outperforms its competitors.

Keywords

Imperialist competition algorithm (ICA), weight optimization, artificial neural network, evolutionary algorithms.

1. INTRODUCTION

Artificial Neural Networks (ANNs) are well-established tools used with success in many problems such as pattern recognition, classification problems, regression problems differential equations, etc [1].

As the determination of network structure and parameters are very important, some evolutionary algorithms such as Genetic Algorithm (GA)[2], Back Propagation (BP) [3], Pruning Algorithm[4], Simulated Annealing [5] and Particle Swarm Optimization [6] have shown significant role in this regard.

These algorithms can evolve neural network at various levels: weight training, architecture adaptation (including number of hidden layers, number of hidden neurons and node transfer functions) and learning rules [7].

There are some unavoidable disadvantages of the above mentioned approaches, which lead us into more potential algorithms. For instance, BP (as a gradient descend method) has slow convergence speed, easily gets into partial extreme value and infirm global searching capability. While GA has its inherent disadvantages of the pre-maturity and the unpredictability of the result. Accordingly, the researching high efficiency algorithm has been one of the most important problems in ANN application [8].

As limited studies have been done only on weight optimization of ANN, we have decided to focus on this matter and propose an evolutionary algorithm to handle this problem. In this paper, we propose an evolutionary algorithm for optimizing the weights of Multilayer Perceptron (MLP) ANNs called Imperialist Competitive Algorithm. This optimization algorithm is inspired by imperialistic competition, which will be discussion in section3.

The proposed method is applied on four experimental datasets for classification purposes that are available on http://archive.ics.uci.edu/ml/datasets.html and have been evaluated by various algorithms in [1]. In order to reduce random variation of the proposed algorithm itself, each experiment has been run 30 times and the mean is presented. The method is tested against ANNs that are trained with various algorithms which are presented in [1], such as GA [2], RPROP [9], MinFinder [10] and ICA [11]. In all the mentioned algorithms, the goal is minimizing the train error of ANN. The comparison results are illustrated in section 4. In almost all cases, the proposed algorithm outperforms its competitors.

2. EXISTING APPROACHES TO ANN WEIGHT OPTIMIZATION

A survey on existing approaches to ANN optimization reveals some evolutionary algorithms which have been widely used in solving various problems. For instance a meta-learning evolutionary neural network is presented [7] to combine the learning of the connection weights and topology on predicting some time series problems. While in an intrusion detection method which is presented in [12], the evolutionary artificial neural network is used to find the optimal network topology and weights. In this research, GA is actually used o initialize the network's weights and the BP to perform a local search and train the network. In a breast cancer diagnosis system [13], a hybrid algorithm that uses BP to estimate the network's weights after the network has been constructed and initialized by the GA. In [14] the authors employ GA to estimate the weights and the number of neurons in the hidden layer of a Radial Basis Function (RBF) network and it is then used to predict time series data. In [1] a grammatical evolution method called Context Free Grammer (CFG), is presented to construct and train the Neural Network topology along with the network parameters (input vector, weights, bias). The combination of a CFG and a genetic algorithm is known as grammatical evolution and has the benefit of allowing easy shaping of the resulting search space. Mind Evolutionary Computation (MEC) is another method for weight optimization of a neural network which is based on simlartaxis and dissimilation operators of weights optimization [8].

3. The PROPOSED APPROACH

3.1. Imperialist Competitive Algorithm (ICA)

Imperialist Competitive Algorithm is a new evolutionary optimization method which is inspired by imperialistic competition [11]. Like other evolutionary algorithms, it starts with an initial population which is called country and is divided into two types of colonies and imperialists which together form empires. Imperialistic competition among these empires forms the proposed evolutionary algorithm. During this competition, weak empires collapse and powerful ones take possession of their colonies. Imperialistic competition converges to a state in which there exists only one empire and colonies have the same cost function value as the imperialist. The pseudo code of Imperialist competitive algorithm is as follows:

1) Select some random points on the function and initialize the empires.
2) Move the colonies toward their relevant imperialist (Assimilation).
3) Randomly change the position of some colonies (Revolution).
4) If there is a colony in an empire which has lower cost than the imperialist, exchange the positions of that colony and the imperialist.
5) Unite the similar empires.
6) Compute the total cost of all empires.
7) Pick the weakest colony (colonies) from the weakest empires and give it (them) to one of the empires (Imperialistic competition).
8) Eliminate the powerless empires.
9) If stop conditions satisfied, stop, if not go to 2.

After dividing all colonies among imperialists and creating the initial empires, these colonies start moving toward their relevant imperialist state which is based on assimilation policy [1]. Fig.1 shows the movement of a colony towards the imperialist. In this movement, θ and x are random numbers with uniform distribution as illustrated in formula (1) and d is the distance between colony and the imperialist.

$$x \sim U(0, \beta \times d), \theta \sim U(-\gamma, \gamma) \qquad (1)$$

where β and γ are parameters that modify the area that colonies randomly search around the imperialist. In our implementation β and γ are considered as 2 and 0.5 (Radian) respectively.

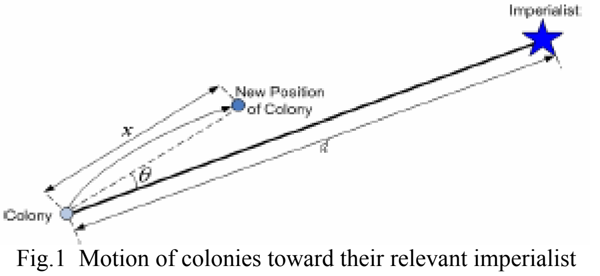


Fig.1 Motion of colonies toward their relevant imperialist

In ICA, revolution causes a country to suddenly change its socio-political characteristics. That is, instead of being assimilated by an imperialist, the colony randomly changes its position in the socio-political axis. The revolution increases the exploration of the algorithm and prevents the early convergence of countries to local minimums.

The total power of an empire depends on both the power of the imperialist country and the power of its colonies which is shown in formula (2).

$$T.C._n = Cost(imperialist_n) + \zeta \, mean\{Cost(colonoes\ of\ impire_n)\} \qquad (2)$$

In imperialistic competition, all empires try to take possession of colonies of other empires and control them. This competition gradually brings about a decrease in the power of weaker empires and an increase in the power of more powerful ones. This is modeled by just picking some of the weakest colonies of the weakest empires and making a competition among all empires to possess these colonies. Fig.2 shows a big picture of the modeled imperialistic competition. Based on their total power, in this competition, each of the empires will have a likelihood of taking possession of the mentioned colonies. The more powerful an empire, the more likely it will possess the colonies. In other words these colonies will not be certainly possessed by the most powerful empires, but these empires will be more likely to possess them. Any empire that is not able to succeed in imperialist competition and can not increase its power (or at least prevent decreasing its power) will be eliminated.

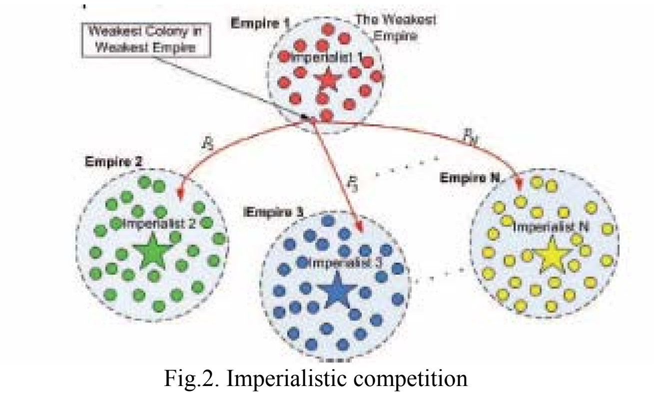


Fig.2. Imperialistic competition

ICA as a new evolutionary method which is used in several applications, such as designing PID controller [15], achieving Nash equilibrium point [16], characterizing materials properties [17], error rate beam forming [18], designing vehicle fuzzy controller [19], etc.

In this paper, we have applied this algorithm for optimizing the weights of ANN and compared the results with other optimization methods which have previously used in this regard.

3.2 ANN weights evolution using ICA

Optimal connection weights can be formulated as a global search problem wherein the architecture of the neural network is pre-defined and fixed during the evolution. Connection weights may be represented as being strings with certain length and the whole network is encoded by concatenation of all connection weights of the network in the chromosome. A heuristic concerning the order of the concatenation is to put connection weights to the same node together.

Evolutionary search of connection weights can be formulated as follows:

(1) Generate an initial population of N weight chromosomes. Evaluate the fitness of each EANN depending on the problem.
(2) Depending on the fitness and using suitable selection methods reproduce a number of children for each individual in the current generation.
(3) Apply genetic operators to each child individual generated above and obtain the next generation.
(4) Check whether the network has achieved the required error rate or the specified number of generations has been reached then goes to step 2.
(5) End.

In order to express the ANN, consider a two layered network which is formulated as formula (3):

$$\sum_{i=1}^{H} w_i \, \sigma\left(\sum_{j=1}^{d} w_j x_j + b \right) \qquad (3)$$

where H denotes the number of neurons in the hidden layer, w denotes the weights of the network, b denotes the bias value and σ denotes the activation function of each neuron which can be considered as sigmoid, tanh (x), coth (x),….

Each chromosome is constructed from the weights of ANN and the fitness of the chromosome is the performance of the neural network on a selected train dataset. The fitness/cost function is considered as formula (4):

$$f(x) = \frac{1}{Q}\sum_{i=1}^{Q}(y_{real} - y_{net})^2 \quad (4)$$

which is the same as MSE(Minimum Squared Error) that should be minimized by this algorithm. Q denotes the number of samples, while y_{real} is the real output for each input x and y_{net} is the desired network output. In this approach, in order to overcome the problem of local minimum and fast convergence, the fitness/cost function has been changed into formula (5)

$$f(x) = \left[\frac{1}{Q}\sum_{i=1}^{Q}(y_{real} - y_{net})^6\right]^{\frac{1}{6}} \quad (5)$$

3.3 Dataset Description

The datasets used for evaluating the proposed approach are known datasets that are available for download from http://archive.ics.uci.edu/ml/datasets.html and refer to classification problems. We have chosen four datasets as follows:

- **WINE** includes data from wine chemical analysis that belong to 3 classes. It contains 178 samples with 13 attributes.
- **GLASS** includes glass component analysis for glass pieces that belong to 6 classes. It contains 214 samples with 10 attributes.
- **PIMA** includes Pima Indians diabetes analysis that belongs to classes of healthy and diabetic. It contains 768 samples with 8 attributes.
- **WDBC** includes Wisconsin Diagnostic Breast Cancer that belongs to 2 classes. It contains 569 samples with 30 attributes.

4. Experimental Results

In this section, the results from the application of ICA against the methods GA, RPROP and Min Finder are listed. Each method was tested for different topologies of the ANN and the topology with the best results was selected. The topologies that we have experienced are as follows:

(1) One hidden layer with 7 neurons.
(2) Two hidden layers with variant neurons between 2 to 20 at each.
(3) Tree hidden layers with variant neurons between 2 to 20 at each.

The best topology among the above experienced ones for WINE was [7,4,3] and for PIMA was [12,5,2]. For WDBC [8,8,2] was appropriate. Moreover, the experiments for WDBC with [3,2,3,2] topology represents very close results to [8,8,2] topology .

With respect to GLASS data set, several experiments have been done with various topologies. The best topologies among more than 15 experiments (with various hidden layers and neurons) were [12,5,6]. Training and Test performance (Percent of corrected & false classified data) of ICA for mentioned topologies were evaluated which are presented in Table1.

Various cost functions, network parameters and activation functions have also been experimented and as a resultant the appropriate ones were selected as are illustrated in Table1.

Fig.3 shows the Test error (false classification percent) for each of the 4th compared optimization methods in [1] (smaller is better).

From the experimental results, it can be seen that in all cases the ICA performed better. ICA method is followed by MinFinder, GA, while RPROP method gives the worst results. For each classification problem, best topologies with best parameters have been selected and minimum cost function value and Mean cost value versus epochs/ decades are presented. Fig.4 illustrates ICA convergence for WINE, for instance. As it is seen, WDBC shows a better performance with least error with respect to the other classification problems. This is followed by WINE, then PIMA and finally Glass classification problems, respectively. WDBC and WINE converge faster than PIMA and GLASS.

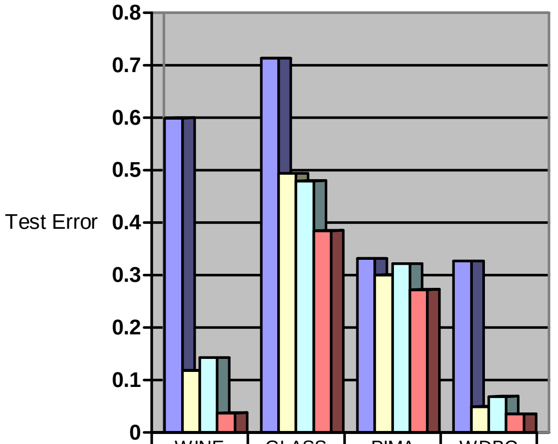
Table1. Selected Parameters for Different Data Bases

Data set	Best Topology	No of Countries (Population size)	No of Initial Imperialists	No of Decades (Generations)	Revolution Rate (Mutation Rate)	Assimilation Coefficient (β)	Assimilation Angle Coefficient (γ)	Zeta	Damped Ratio	Uniting Threshold	Activation function	MSE (Test)	MSE (Train)	Corrected Classified (test) (%)	Corrected Classified (train) (%)	False Classified - test(%)
WINE	[7,4,3]	250	20	250	0.5	2	0.5	0.02	0.99	0.02	Tanh	0.0394	0.0382	0.9629	0.9677	0.0370
WDBC	[8,8,2]	250	20	300	0.5	2	0.5	0.02	0.99	0.02	Tanh	0.0509	0.0498	0.9649	0.9673	0.0351
PIMA	[12,5,2]	250	20	200	0.5	2	0.5	0.02	0.99	0.02	Tanh	0.0767	0.0743	0.7272	0.7728	0.2727
GLASS	[12,5,6]	250	20	2000	0.5	2	0.5	0.02	0.99	0.02	Tanh	0.0406	0.0403	0.6153	0.6174	0.3846

5. Concluding Remarks

The method proposed in this paper, Imperialist Competitive Algorithm (ICA) uses an evolutionary algorithm in order to optimize the weights of a MLP neural network. The proposed method encodes the network parameters. The ICA method is evaluated on four known classification problems and compared against the state of the art methods: GA, RPROP, MinFinder. An accurate comparison of the four methods is presented that uses 30-fold experiment replication. The experimental results show that the proposed ICA method outperforms the other methods. This evolutionary optimization strategy has shown great performance in both convergence rate and better global optima achievement.

In continuing this work, the proposed algorithm could be extended to include time series prediction, feature selection and/or feature construction, ANN topology and weight optimization simultaneously.



	WINE	GLASS	PIMA	WDBC
RPROP	0.5993	0.7137	0.3319	0.3271
MINIFINDER	0.1178	0.4941	0.3004	0.0489
GENETIC	0.1426	0.4801	0.3216	0.0687
ICA	0.037	0.3846	0.2727	0.0351

Fig.3 shows the Test error (false classification percent) for each of the 4 compared optimization methods

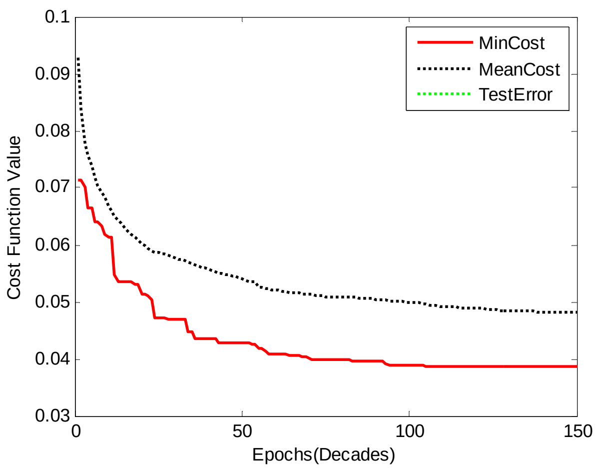


Fig.4 ICA Convergence for WINE Classification problem (with topology [7,4,3], 250 population, 20 imperialist, 250 generation, 0.5 evolution rate, 0.0185 error)

6. Acknowledgment

Authors gratefully acknowledge Mr. Esmaeil Atashpaz, who has invented ICA algorithm and presented the related useful documents on his website. His support and help is truly appreciated.

REFERENCES

[1] I.Tsoulos, D.Gavrilis, E.Glavas, "Neural network construction and training using grammatical evolution", Science Direct Neurocomputing Journal, Vol.72, Issues 1-3, December 2008, pp. 269-277.

[2] X. Qu1, J. Feng, W. Sun, "Parallel Genetic Algorithm Model Based on AHP and Neural Networks for Enterprise Comprehensive Business", IEEE Intl. Conf. on Intelligent Information Hiding and Multimedia Signal Processing, 2008, pp.897-900.

[3] P.Tang and Z. Xi, "The Research on BP Neural Network Model Based on Guaranteed Convergence Particle Swarm Optimization", Second Intl. Symp. on Intelligent Information Technology Application, IITA '08, Vol.2, Dec. 2008, pp.13 – 16.

[4] R.Reed, "Pruning algorithms—a survey", IEEE Trans. Neural Networks Vol.4, 1993, pp.740–747.

[5] M.C.P.de Souto, A.Yamazaki, T.B.Ludernir, "Optimization of neural network weights and architecture for odor recognition using simulated annealing", Proc. 2002 Intl. Joint Conf. on Neural Networks,Vol.1, 2002, pp.547–552.

[6] C.zhang, H.Shao, Y.Li, "Particle swarm optimization for evolving artificial neural network", 2000 IEEE Intl. Conf. on Systems, Man, and Cybernetics, Vol. 4, Oct. 2000, pp.2487 – 2490.

[7] A.Abraham, "Meta learning evolutionary artificial neural networks", Neurocomputing, Vol. 56, 2004, pp.1–38.

[8] X.He, J.Zeng, J.Jie, " Artificial neural network weights optimization design based on MEC algorithm", Proc. of 2004 Intl. Conf. on Machine Learning and Cybernetics, Vol. 6, Aug. 2004, pp. 3361 – 3364.

[9] M.Riedmiller, H.Braun, "A direct adaptive method for faster back propagation learning (RPROP}, Proc. of the IEEE Intl. Conf. on Neural Networks,1993, pp. 586–591.

[10] I.G.Tsoulos, I.E.Lagaris, "MinFinder: locating all the local minima of a function", Comput. Phys. Commun. Journal, Vol.174, No.2, January 2006, pp.166–179.

[11] E.Atashpaz Gargari, C. Lucas, "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition", IEEE Congress on Evolutionary Computation, 2007.

[12] S.-J.Han, S.-B. Cho, "Evolutionary neural networks for anomaly detection based on the behavior of a program, IEEE Trans. Systems, Man Cybernetics, Vol.36, No.3, 2006.

[13] H.A.Abbass, "An evolutionary artificial neural networks approach for breast cancer diagnosis, Artif. Intell. Med., Vol.25, No.3, July2002, pp.265–281.

[14] V.M.Rivas, J.J.Merelo, P.A.Castillo, M.G.Arenas,J.G. Castellano, "Evolving RBF neural networks for time-series forecasting with EvRBF", Inform. Sci. Journal, Vol.165, No. 3-4, Oct 2004, pp.207–220.

[15] E. Atashpaz, F. Hashemzadeh, R. Rajabioun, C. Lucas, "Colonial competitive algorithm: a novel approach for PID controller design in MIMO distillation column process", Intl. Journal of Intelligent Computing and Cybernetics, Vol.1, Issue 3, 2008, pp.337 – 355.

[16] Ramin Rajabioun, E.Atashpaz, C.Lucas, **"**Colonial Competitive Algorithm as a Tool for Nash Equilibrium Point Achievement", Lecture Notes In Computer Science; Vol. 5073, Proc. of the Intl. conf. on Computational Science and Its Applications, Part II, 2008, pp.680-695.

[17] B.Oskouyi, E. Atashpaz-Gargari, N. Soltani, C. Lucas, "Application of Imperialist Competitive Algorithm for Materials Property Characterization from Sharp Indentation Test", to be appeared in International Journal of Engineering Simulation.

[18] A. Khabbazi, E. Atashpaz and C. Lucas, "Imperialist Competitive Algorithm for Minimum Bit Error Rate Beamforming", To be appeared in International Journal of Bio-Inspired Computation (IJBIC)

[19] A. M. Jasour, E. Atashpaz, C. Lucas, "Vehicle Fuzzy Controller Design Using Imperialist Competitive Algorithm", Second First Iranian Joint Congress on Fuzzy and Intelligent Systems, Tehran, Iran, 2008.