Desheng Li, Huibin Xu
Anhui Science and Technology University, Shanghai Normal University Tianhua College

Optimization of Neural Network Model Design: An Electoral Cooperative Particle Swarm Optimization Approach

***Abstract**. This paper proposes an electoral cooperative particle swarm optimization approach to optimize the model of neural network from both structure and linked weights. Different with other related research work, a new encoding method is adopted to divide the neural network into several modules, each of them corresponding to a sub-swarm. Based on the experiments on typical problems and classic dataset, the results show that the proposed algorithm outperforms all the compared ones in perspective of test error, correctness, connection number, and the CPU time of the training phase.*

***Streszczenie.** W przedstawionym artykule opisano zastosowanie metod optymalizacji roju cząstek do optymalizacji struktury i współczynników wagowych sieci neuronowej. Zaimplementowano nową metodę analizy, do dzielenia podzielenia sieci na moduły, reprezentujące mniejsze roje. Weryfikacja eksperymentalna i porównanie z metodami klasycznymi wykazały wysoką sprawność i skuteczność analizy. (**Optymalizacja modelu sieci neuronowej z zastosowaniem optymalizacji roju cząstek ze współdzieleniem grup**).*

Keywords: Particle Swarm Optimization, Cooperative Evolution, Neural Network, Back Propagation
Słowa kluczowe: optymalizacja roju cząstek, rozwój współdzielony, sieć neuronowa, propagacja wsteczna.

1. Introduction

Neural Network (NN), which reflects the nonlinear mapping relation of input and output with the advantage of self-adaptation, self-learning, and fault-tolerance, has been widely applied in complex system identification, pattern recognition, fault diagnosis, and other relative fields. However, a series of problems also exist in the application, such as NN's structure, training algorithm, linked weights and so forth, which are always be determined by a large number of experiments and consequently limited the NN's application.

Cooperative evolution, as a new technology in swarm Intelligence, has been widely applied in the optimization of neural network model design. Predrajas [1] developed a cooperative GA to optimize the structure and link weights of NN considering of the diversity of sub-swarms and performance. When the cooperative PSO was proposed, it had been used for NN training firstly in literature [2]. Then, Rui Mendes et al. [3] investigated the application of PSO in the feedforward NN training systematically. In the paper of Niu [4], a standard (real-coded) PSO is employed to training NN's free parameters (weights and bias) and binary-coded GA is used to find optimal NN's structure.

In this paper, a particle swarm optimization with electoral mechanism, called Electoral Cooperative Particle Swarm Optimization, is employed to optimize the model of neural network from structure and linked weights perspectives based on typical classification problems.

2. Electoral Cooperative Particle Swarm Optimization (ECPSO)

The electoral mechanism is on the basis of the multi-swarm and cooperative variants of PSO, Cooperative Particle Swarm Optimization (CPSO) proposed by Van den Bergh F. in [5], in which the high-dimension search space can be decompose into small scale ones similar to the idea of RELAX/CLEAN algorithm. However, its difference to it is that due to the imported information exchange mechanism among particles, the more accurate estimates did not need reduplicative iterations any more. Compared to basic single swarm PSO, both robustness and precision are improved and guarantied. The key idea of CPSO is to divide all the n-dimension vectors into k sub-swarms. So the front n/k swarms are $\lceil n/k \rceil$-dimensional, and the $k-(n/k)$ swarms behind have $\lfloor n/k \rfloor$-dimensional vectors.

In this paper, we use a cooperative swarm optimization algorithm named ECPSO in our previous work [6]. Firstly, we will discuss the dynamics of particles in the swarm, which is different with plain PSO and conventional cooperative PSO algorithms. The movement equation can be formalized as following equation set (1):

$$(1)\qquad \begin{cases} V_{id}^{new} = \omega \times V_{id} + C_1 \times Rand() \times (P_{id}^{best} - P_{id}) + \\ C_2 \times Rand() \times (P_{gid}^{best} - P_{id}) + C_3 \times Rand() \times (\hat{P}_{egid}^{best} \uparrow_{id} - P_{id}) \\ P_{id}^{new} = P_{id} + V_{id}^{new} \end{cases}$$

The function *b* shown in Equation (2) performs exactly this: it takes the best particle from each of the other sub-swarms, concatenates them, splicing in the current particle from the current sub-swarm *j* in the appropriate position.

$$(2)\qquad b(u, S_u.\hat{P}_{gid}^{best}) = argmin\ fitness(b(u, S_u.P_{id}^{best}), b(u, ES.P_{gid}^{best})),\quad 1 \le id \le s, 1 \le u \le k$$

3. Artificial neural network model design with ECPSO

3.1 Population division and representative individual selection

In cooperative co-evolution for NN optimization, it is important to divide the NN into several modules so that each module can be evolved by a sub-swarm.

Without loss of generality, suppose that the target NN includes N_p+1 layers, i.e., one input layer, N_p-1 hidden layers and one output layer. Whereas the node number and the linked weights of hidden layers are not determined in advance, but need to be optimized. For clarity, the layers from front to behind are called the 1st, 2nd, and the N_p+1 layer.

In our research, we divide an NN into several modules. As illustrated in Fig.1, we can see that the N_p module is defined as the part between the N_p layer (the N_p-1 hidden layer) to the N_p+1 layer (output layer). For each module, a corresponding sub-swarm is deployed to optimize the NN's structure and the linked weights, which are denoted as P_1, P_2, …,P_{Np}.

In Potter's paper [7], he proposed two individual selection schemes: one is selecting the current best individual of other sub-swarms; the other is choosing a best individual and a random one of other sub-swarms. But these two methods need to consider the correlation among the sub-swarms and also not notable in the promotion of efficiency.

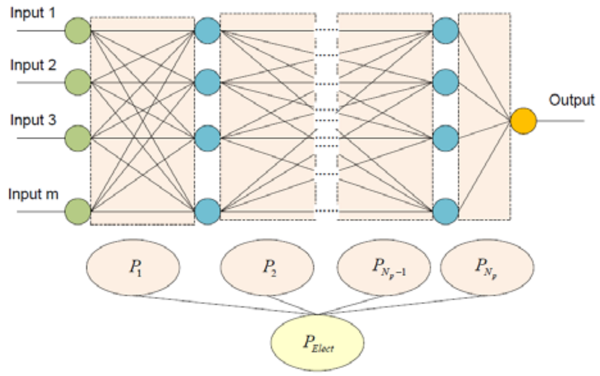

Fig.1. Structure of NN and evolutionary population division

In our previous work in literature [6], an approach employing the electoral mechanism with the dynamic voting has used to improve the CPSO. Hence, in this paper we also follow this technological path and compare its performance with Potter's methods, which will be introduced in the next section in detail.

3.2 Decision variable encoding

Now, considering the p-th module, as it is the part between layer p and p+1, so it is optimized by the P_{Np}. In our algorithm, an approach of binary encoding is adopted to represent the structure of ANN, i.e., the connective relations between nodes, while real number encoding for the linked weights.

Let the node number of layer p is N^p, then the connective relations can be denoted by a matrix $S^p_{N^{p+1}\times N^p}$. If $S^p_{(i,j)}=1$, then it represents there exists a link between the node i in layer $p+1$ to the node j in layer p; otherwise, if $S^p_{(i,j)}=0$, then there exists no any link between them. Consequently, the structure encoding of individual in the p-th sub-swarm's can be denoted by $S^p_{N^{p+1}\times N^p}$. Taking the NN in Fig.2 for instance, as the network includes one input layer, two hidden layers and one output layer, so the node numbers of perspecitive layers are 2,4,4,4,1. Then the network can be divided into 3 parts corresponding to 3 sub-swarms.

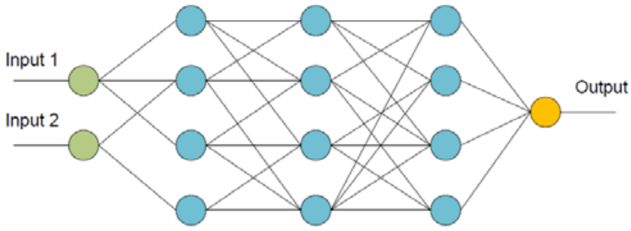

Fig.2. An example of NN for the structure encoding

Table.1. Structure encoding of NN in Fig.2

Structure encoding	P_1	P_2	P_3	P_4
$S^p_{N^{p+1}\times N^p}$	$\begin{bmatrix}10\\11\\10\\01\end{bmatrix}$	$\begin{bmatrix}1100\\1110\\1011\\0111\end{bmatrix}$	$\begin{bmatrix}1101\\1111\\1011\\0111\end{bmatrix}$	$[1111]$

Obviously, if there exists no connection relation between two nodes, then the link weight is zero. Hence, the encoding of link weights reflect the structure in this case. When the structure code is one, the link weights are only considered, whose lenghts equal to the number of "1" in structure codes.

3.3 Fitness function

In this research, the target is to design a proper NN under the condition of a given group of input/output pairs to make the output of NN approximate the given data as close as possible.

Let note the input/output pairs as (x_i,y_i), $i=1,2,\ldots,N_{tr}$, and when the input is x_i, the real output of NN is $\hat{y}_i$. Considering the p-th sub-swarm P_p, the j-th individual x^p_j, the representative ones from other sub-swarms, x^q_r, $q=1,2,\ldots,p-1,p+1,\ldots,N_p$, then the fitness value of the x^p_j can be defined as below:

$$F(x^p_j)=1/(\frac{1}{N_{tr}}\sum_{i=1}^{N_{tr}}(y_i-\hat{y}_i(x^1_r,x^2_r,\ldots,x^{p-1}_r,x^p_j,\ldots,x^{N_p}_r))^2+\varepsilon \tag{3}$$

3.4 Algorithms

According to the principal and encoding method above, we have developed an algorithm to optimize the structure and link weights of a NN, which is shown in Alg.1.

Algorithm 1. ECPSO-NN
Input: N_p modules of NN, fitness function.
Output: optimized NN.
Procedure*:*
Divide the Network into N_p modules.
Let t=0, and initialize the N_p sub-swarms: $P_1(t)$, $P_2(t)$, …, $P_{Np}(t)$.
Do
Select the representative individual of respective sub-swarms by electoral mechanism.
Composite the cooperative swarm by the selected particles.
Decode and form the NN.
Evaluate the fitness function.
Update the velocities and positions of particles in sub-swarms.
$t=t+1$.
While (terminate condition not satisfied)
Return the optimized NN.
End

For the feedforward neural network, BP algorithm, with the strong local search ability, is the traditional training algorithm. To improve the optimization effect, a two-stage algorithm hybriding the ECPSO and BP is developed as shown in Alg.2, which has both considerabe trait of high exploration and exploit.

Algorithm 2. ECPSO-BP-NN
Input: N_p modules of NN, fitness function.
Output: optimized NN.
Procedure*:*
Divide the Network into N_p modules.
Let t=0, and initialize the N_p sub-swarms: $P_1(t)$, $P_2(t)$, …, $P_{Np}(t)$.
Do
Select the representative individual of respective sub-swarms by electoral mechanism.
Composite the cooperative swarm by the selected particles.
Decode and form the NN.
Evaluate the fitness function.
Update the velocities and positions of particles in sub-swarms.
$t=t+1$.
While ($t<T_{max\text{-}CPSO}$)
$gbest_= gbest_{CPSO}$, t=0.

Do

Utilize the BP to search around the $gbest_{CPSO}$ loclally.

If ($gbest_{BP}$<$gbest_{CPSO}$)

Then $gbest$=$gbest_{BP}$.

While (t<$T_{max\text{-}BP}$)

Return the optimized NN.

End

4. Computational results on NN for classification

4.1 Experiment 1: Bi-spire problem

To verify the availability of the proposed algorithms, Bi-spire problem using NN is investigated compared with the Potter's method [8]. Fig.3 shows the landscape of a Bi-spire problem, which is a typical classification problem that can hardly find a satisfied NN by gradient descent methods. In this experiment, we have executed two groups of testing data and training data with the NN. The comparison of related optimization algorithms is illustrated in Table 2.

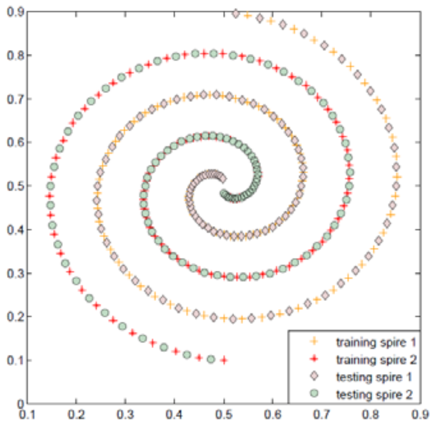

Fig.3. Bi-spire curves under the testing and training data

4.2 Experiment 2: Iris, Ionosphere, and Breast Cancer

In experiment 2, the classic data sets of Iris, Ionosphere, and Breast Cancer [9] are used to investigate the NNs' performance optimized by BP-NN, ECPSO-NN, and ECPSO-BP-NN hybrid algorithm.

Table.2. Comparison results on Bi-spire problem

Algorithms	Hidden layer nodes			Succ	CPU Time
	Mean	Max	Min		
Potter	13.7	18	12	70	429.4
CGA	11.0	15	8	90	77.2
ECPSO	9.4	12	7	88.4	89.7
ECPSO-BP	7.5	10	6	93.1	125.4

For Iris problem, the connection thresholds θ_{ih} and θ_{ho} are set to 0.5, while the number of hidden nodes n_h is 8. In comparison, the connection thresholds θ_{ih} and θ_{ho} of Ionosphere are set to 0.4, and n_h is 15. The setting of Breast Cancer is same as Iris problem except for n_h=15. The performance of optimized NNs, such as test error, correctness, connection number, and the CPU time of the training phase are illustrated in the Table 3.

From Table 3, we can draw a conclusion that the ECPSO-NN and ECPSO-BP-NN algorithms have faster convergence speed and higher accuracy than the pure BP appoach. To be specific, the structure optimization from ECPSO makes the NN enhance the pattern classification performance, which indicates that the deleted network connections are redundancy ones, and also verifies that the redundancy links' influences on the NNs' performance, especially on the overtraining problem.

5. Conclusion

This paper proposes an electoral cooperative particle swarm optimization approach to optimize the NN's structure and linked weights. Different with other related research work, we adopts a new encoding methods that divide the NN into several modules, each one corresponding to a sub-swarm. Based on the experiments of classification problems, the results show the proposed algorithms outperform the compared ones in perspecitve of test error, correctness, connection number, and the CPU time.

Table.3. Comparison results on Iris, Ionosphere, and Breast Cancer

Problem	ECPSO-NN				ECPSO-BP-NN				BP-NN			
	Error	Correct	links	Time	Error	Correct	links	Time	Error	Correct	links	Time
Iris	1.147	96.829	30	18.225	1.283	96.328	25	32.1	2.234	95.179	56	60.34
Ionosphere	1.864	97.040	268	202.13	2.025	94.549	258	378.4	4.640	94.583	530	863.47
Breast Cancer	0.658	97.995	64	97.424	0.714	96.25	55	219.2	1.573	98.510	125	391.58

Acknowledgments

This work was supported by the Talent Introduction Special Fund of Anhui Science and Technology University (No.ZRC2011304), and Science and Technology Plan Project of Chuzhou City (No.201236).

REFERENCES

[1] Pedrajas N.G., Martinez C.H., Perez J.M.: Multi-objective cooperative coevolution of artificial neural network, *Neural Network*, 15 (2002):1259-1278.

[2] van den Bergh F., Engelbrecht A.P.: Cooperative Learning in Neural Networks using Particle Swarm Optimizers, South African Computer Journal, 26(2000): 84-90

[3] Mendes R., Cortez P., Rocha M., Neves J.: Particle Swarms for Feedforward Neural Network Training, In Proceedings of The 2002 International Joint Conference on Neural Networks (2002):1895-1899

[4] Niu B., Zhu Y., Hu K., Li S., He X.: A Cooperative Evolutionary System for Designing Neural Networks, Lecture Notes in Computer Science, 4113(2006): 12-21

[5] van den Bergh F., Engelbrecht A.P.: A cooperative approach to particle swarm optimization, *IEEE Transactions on In Evolutionary Computation*, 8(2004):1-15

[6] Li D., He Q.: A Version of Cooperative Multi-Swarm PSO Using Electoral Mechanism to Solve Hybrid Flow Shop Scheduling Problem[J]. *Przeglad Elektrotechniczny*, 88(2012):22-26

[7] Potter M.A.: The design and analysis of a computational model of cooperative coevolution, Fairfax: George Mason University, 1997.

[8] Potter M.A., De Jong K.A.: Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents, *Evolutionary Computation*, 8(2002):1-29

[9] UC Irvine Machine Learning Repository, http://archive.ics.uci.edu/ml

***Authors**: dr inż, Li Desheng, Anhui Science and Technology University, School of Science, ul. Donghua Road 9, 233100, Fengyang, Anhui, China, E-mail: ldsyy2006@126.com. Huibin Xu, Shanghai Normal University Tianhua College, 1661, North Shengxin Rd, Jiading District Shanghai, 201815, China, xuhuibin188@163. com.*