Study on Evolutionary Neural Network Based on Ant Colony Optimization

Gao Wei
Wuhan Polytechnic University
Hubei, Wuhan 430023, P. R. China
wgaowh@hotmail.com

Abstract

The evolutionary neural network model can be generated combining the evolutionary optimization algorithm and neural network. Based on analysis of merits and demerits of previously proposed evolutionary neural network models, combining the continuous ant colony optimization proposed by author and BP neural network, a new evolutionary neural network whose architecture and connection weights evolve simultaneously is proposed. At last, through the typical XOR problem, the new ENN is compared and analyzed with BP neural network, traditional ENN based on genetic algorithm and evolutional programming. The computing results show that the precision and efficiency of the new ENN are all better.

1. Introduction

By combination of evolutionary algorithm and neural network, a new neural network-Evolutionary Neural Network (ENN) is generated [1]. In ENN, the auto-adaptability of evolutionary algorithm and learning capability of neural network is combined effectively. And then, almost all demerits of traditional neural network can be overcome. Someone has already predicated that the ENN is the next generation neural network [2]. Based on above ideas, a lot of researchers have studied the ENN and already proposed many ENNs [1-7]. For the complexity of ENN research, this study is not very good. So, it is very necessary to still do some work on this field. In this paper, combining the continuous ant colony optimization proposed by author and BP neural network, a new ENN whose architecture and connection weights evolve simultaneously is proposed.

2. Analysis of traditional ENN

Through carefully analyzing the existing ENNs, we can see that, the ENN based on genetic algorithm is the main one that has been studied for a long time and has obtained lot of achievements. But these ENNs have follow shortcomings.

2.1. Problems produced by coding operation

The binary code is generally used in genetic algorithm, which corresponds to coding individuals into discrete space. When coding the architecture of neural network, this operation will make the architecture fixed and make some good architecture lost. While coding connection weights, it will make the expressing precision low.

2.2. Problems produced by genetic operation

The crossover operation will likely destroy the produced good network architecture. And also the crossover operation can generate the “interconversion”, that is, two different genotypes are essentially the right-and-left interchange of same network architecture. While the mutation operation will produce large jump, and makes the searching process unstable.

2.3. Multi-multi mapping of genotype and representation type

First, it will generate estimating error when fitness of representation type approximates the fitness of genotype. Second, there will exist arraying problem, that is to say, one representation type is corresponding with many genotypes.

According to above analysis, Yao et al [4-5] proposed one kind of typical ENN called EP Net. In this model, evolutionary programming and modified BP network is combined. And then, the shortcomings of traditional ENN can be overcome effectively. But in this model, users must only use their experiences to estimate some parameters that have strong relation.
with the performance of ENN, which makes the robustness of ENN very poor.

To get a kind of good ENN, here the Continuous Ant Colony Optimization (CACO) is introduced and one new kind of ENN whose architecture and connection weights evolve simultaneously is proposed.

The theories of neural network have proved that, three-layer feed forward neural network can approximate any mapping from input space to output space with any precision [8]. Also, the previous studies have proved that [8], in order to improve the generalization of neural network, its architecture should be as simple as possible. So, in this new ENN, the three-layer feed forward neural network is applied. Also, to make the model as simple as possible, the full-linking network is applied.

3. New evolutionary neural network

3.1. Ant colony optimization

Ant Colony Optimization (ACO) is a new evolutionary optimization algorithm from mimic the behavior of ant colony, and proposed by Italy scholar M. Dorigo in 1990’s [9-10]. The original intention of ACO is to solve the complicated combination optimization problems, such as TSP, so the traditional ACO is a very good combination optimization method. Its basic principles are as follows.

As one kind of social insect, the behavior of ant is very simple, but the ant colony can represent very complicated behavior, and can complete very complicated task. The scientists have noticed this phenomenon for a long time. From a lot of researches, they found that, the information is delivered among ant colony by one kind of hormone. Through this kind of information exchange, the ant can cooperate and complete very complicated task. As the ant moves, the hormone is released on its path. The follow ants can perceive this hormone and recognize its density. And have a larger probability to move along the path that has the greater hormone density. So, this is a kind of information positive feedback: the more ants move along a road, the more probability of follow ants also move along this road.

3.2. Continuous ant colony optimization

For imitation information exchange among ant colony, the ACO can solve the complicated combination optimization problem, and its effect is better than that of traditional methods. So according to the information cooperation of ant colony, the continuous optimization method based on principles of ant colony should be feasible.

Based on above thought, some Continuous Ant Colony Optimizations have been proposed [11-13]. To study simply and effectively, here a new Continuous Ant Colony Optimization is proposed, which flow chart is as follow Figure 1.

![Figure 1. Basic flow chart of continuous ant colony optimization](image)

The process of CACO is introduced briefly as follows. Firstly, each ant is randomly distributed on the solution space. And the fitness of each ant is computed, also the position of each ant is recorded. Then, the internal cycle start, the ant individual is moved according probability. After the internal cycle end, the hormone of each ant individual is updated. And then the extrinsic cycle is continued until the optimum solution is found. Then the whole process is finished.

3.3. New ENN

The details of the new ENN are given as follows.

3.3.1. Individual expression. According to requirement of new ENN, the individual should include number of hidden neuron and linking weights and thresholds of whole network. In order to express network simply, the threshold of neuron is put into the matrix of linking weight. To express easily, the individual expression is taken structural data type. So, the pseudocode of individual is as follows.

```plaintext
Type geti
    Integer yinjiedianshu;
    Real W[i][j];
End type geti.
```
where, $W[i][j]$ is the matrix of linking weight, that include two matrices which are matrix of linking weight between input layer and hidden layer and matrix of linking weight between hidden layer and output layer. Parameter of $yinjiedianshu$ is the number of hidden neuron.

3.3.2. Creation of the ant initial population. Randomly generating certain number of neural networks as initial population, whose hidden neuron number and linking weights are generated in their initial scope. That is to say, the ant colony is randomly distributed in the solution space.

3.3.3. Expression of fitness function. In order to improve the generalization of ENN, the network individual is trained by "sample counterchanging method", and then its fitness is calculated. The sample counterchanging method is that, as to network individual training of each generation, not the whole training sample set is used, that is to say, a part of training sample set (the 80% of whole sample set) is randomly drawn out to train the individual of each generation. So, the used training sample set for neural network of each generation is changed, then the fitness of individual whose generalization capacity is poor is small while the fitness of individual whose generalization capacity is strong is large. Consequently, the performance of the whole ENN is improved through selection.

The error function of neural network is expressed as follows.

$$E = \frac{1}{2} \sum_{a=1}^{N} \sum_{k=1}^{M} [y_a(w_j; x^a) - t_a]^2$$

The individual fitness of neural network is expressed by follow transformation of error function of neural network.

$$f = \frac{1}{1+E}$$

3.3.4. Probability move of ant individual. The probability move of ant individual is the key operation of algorithm. Its move probability can be expressed as follows.

$$P_{ij} = \frac{[\tau_j]^\alpha [\eta_j]^\beta}{\sum_k [\tau_k]^\alpha [\eta_k]^\beta}$$

where, $\tau_j$ is hormone intensity of ant individual. At initial stage, it is a constant, which is $\tau_0 = c$. In this study, we take $c = 0.001$. $\tau_j$ is related with $f(i)$ through $\Delta \tau_j$. $\eta_j = f_i - f_j$, which express the modification quantity of objective function after the ant individual moves. When $f_i = f_j$ is occurred, the movement should be repeated, until they are not equal. The $\alpha$ and $\beta$ are two variables, which ranges are as follows, $1 \leq \alpha \leq 5$, $1 \leq \beta \leq 5$. Here, we take $\alpha = 1$ and $\beta = 5$.

3.3.5. Hormone update operation. The method of hormone update can be expressed as follows.

$$\tau_j^{new} = \rho \cdot \tau_j^{old} + \sum_k \Delta \tau_j^k$$

where, $\rho$ is volatile rate of hormone, which can be taken as about 0.3. $\Delta \tau_j$ is residual quantity of hormone. $\Delta \tau_j$ can be expressed as follow forms.

$$\Delta \tau_j = \begin{cases} Q & \text{ant moved along pathwayij} \\ 0 & \text{otherwise} \end{cases}$$

where, $Q$ is a constant.

4. Simulation experiment

In order to verify the new ENN proposed here, the typical problem of XOR is applied to test the new ENN.

The training sample set of XOR problem is as follow Table 1.

| X1 | 0 | 1 | 1 | 0 | 1 | 0 | -1 | -1 | -1 |
| X2 | 0 | 0 | 1 | 1 | -1 | -1 | -1 | 0 | 1 |
| Y  | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |

In new ENN, the controlling parameters are as follows. Number of input neuron is 2. Number of output neuron is 1. The number scope of hidden neuron is 1 to10. The initial value scope of linking weight is –1.0 to 1.0. The population size of ant colony algorithm is 100. The generation threshold is 200. At the same time, the maximum error of best individual is given as $10^{-5}$.

In order to compare the new ENN with BP network and ENNs in [6] and [7], the three networks are applied to solve the same XOR problem. In BP network, the controlling parameters are as follows, learning-rate $\nu = 0.6$, momentum $\lambda = 0.2$. The architecture of network is 2-2-1. In ENN in [6], the controlling parameters are as follows, $P_m = 0.05, P_e = 0.8, N = 30$. The evolutionary generation threshold is 200.

Training three networks with training set in Table 1, we can get the follow results.
The network architecture of new ENN is 2-2-1. The comparison results are in follow Table 2.

Table 2. Computation results of some neural network models

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0)</td>
<td>0.003864</td>
<td>0.000041</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0</td>
</tr>
<tr>
<td>(1, 0)</td>
<td>0.998074</td>
<td>0.999937</td>
<td>1.000000</td>
<td>1.000002</td>
<td>1</td>
</tr>
<tr>
<td>(1, 1)</td>
<td>1.000000</td>
<td>0.999745</td>
<td>0.999997</td>
<td>0.999996</td>
<td>1</td>
</tr>
<tr>
<td>(0, 1)</td>
<td>0.996921</td>
<td>0.999688</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1</td>
</tr>
<tr>
<td>(1, -1)</td>
<td>0.0000773</td>
<td>0.000013</td>
<td>0.000002</td>
<td>0.000001</td>
<td>0</td>
</tr>
<tr>
<td>(-1, -1)</td>
<td>0.999867</td>
<td>0.999945</td>
<td>1.000000</td>
<td>0.999998</td>
<td>1</td>
</tr>
<tr>
<td>(-1, 0)</td>
<td>0.997784</td>
<td>0.999991</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1</td>
</tr>
<tr>
<td>(-1, 1)</td>
<td>0.9997777</td>
<td>0.999795</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1</td>
</tr>
<tr>
<td>Training error</td>
<td>0.000014</td>
<td>0.000007</td>
<td>0.000000</td>
<td>0.000004</td>
<td></td>
</tr>
<tr>
<td>Iterative time</td>
<td>70000</td>
<td>41</td>
<td>14</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

From above experiment results we can see that, the computation effect of new ENN is obviously better than that of BP network and other ENNs. Its computation precision is higher and iterative time is fewer. So, the new ENN proposed in this paper is a very good ENN and can be applied in many complicated engineering problems.

6. Conclusion

ENN is a new kind of neural network combining evolutionary computation and neural network theory. Because in this model, the auto-adaptability of evolutionary computation and learning capability of neural network can be combined effectively, the ENN has become the inevitable tendency of neural network. As to the importance of this study, combining the continuous ant colony optimization proposed by author and BP neural network, a new evolutionary neural network whose architecture and connection weights evolve simultaneously is proposed in this paper. At last, this new ENN is verified by typical XOR problem, and is compared with BP network and other ENNs. The results show that, the new ENN can obviously improve the computation precision and computation efficiency and is a very good neural network.

7. References


