A Train Algorithm of Multi-Aggregation Process Neural Networks Based on Chaos Genetic Optimization

PANG Yue-Wu
School of Computer & Information technology, Northeast Petroleum University, Daqing, 163318 China

XU Shao-Hua
School of Computer & Information technology, Northeast Petroleum University, Daqing, 163318 China

***Abstract* — Aiming at the learning problem of multi-aggregation process neural networks, an optimization train method based on chaos genetic algorithm (CGA) with lowest mean square algorithm (LMS) is proposed in the paper. The learning problem of the network weight functions is converted to the training of basis expansion coefficients through adopting the function orthogonal basis expansion method. The lowest mean square error as the train objective function, the global optimal solution of network parameters is solved in feasible solution space using the chaos rail to traverse the search of CGA. The application in multi-variant dynamic signal identification proved that the algorithm proposed in the paper improved the training efficiency and stability greatly.**

Key Words-process neural networks, train algorithm, chaos genetic algorithm, mixed optimization strategy

I. Introduction

Aiming at the Information processing problem of spatio-temporal dimension signal space, a Multi-Aggregation Process Neural Networks (MAPNN) model is proposed by HE and XU [1]. The inputs, activation thresholds and connection weights of the model can be multivariate process functions, of which the spatio-temporal aggregation operations are composed of Multi-process input signals, spatial weighted aggregation, process accumulation and activation outputs. It has good adaptability for data processing problems of nonlinear system which includes time and space information. However, since the information processing mechanism of MAPNN is complex, which includes many parameters to be determined and large amounts of data in Network training sample set, the learning efficiency based on gradient descent method is reduced. At the same time, constraint between the network parameters has a great degree of freedom, which affects the stability of the algorithm and generalization ability of network models.

In recent years, optimization techniques based on evolutionary computation are widely used to solve the optimal solution in engineering field, which are effectively applied to the fields [2- 4] of calculation with complex functions, process optimization, system identification and system control. Chaos-genetic algorithm (CGA) combined developmental mechanism in genetic algorithms with chaos search strategy, has properties of groups search and track traversing, which shows advantages [5, 6] in algorithm theory and optimization strategies, such as constrained multi-objective optimization, extremal problems solving in high-dimensional space, etc. Therefore, if the training combines CGA optimization search mechanism with MAPNN, the learning properties of MAPNN will be greatly improved.

Aiming at the learning problem of multi-aggregation process neural networks, an optimization train method based on chaos genetic algorithm (CGA) with lowest mean square algorithm (LMS) is proposed in the paper. The learning problem of the network weight functions and threshold functions is converted to the training of basis expansion coefficients through adopting the function orthogonal basis expansion method. And multi-time aggregation operators of Multi-Aggregation Process neurons are converted to multivariate numerical integration. The lowest mean square error as the train objective function, the global optimal solution of network parameters is solved in feasible solution space using the chaos rail to traverse the search of the CGA. The optimizational learning steps of MAPNN are given in this paper, which are applied to the research of multivariate dynamic signal recognition, and obtain good results.

II. The model of Multi-Aggregation Process Neural Network

A Multi-Aggregation Process Neuron

Suppose that the common input form of spatio-temporal nonlinear system is $x_i(t_1,t_2,...,t_P)$ for $i=1,2,...,n$; $t_p\in[0,T_p]$ for $p=1, 2,...,P$ in which some T_p can be 0. The structure and information transfer process of multi-aggregation process neuron is shown in Fig. 1.

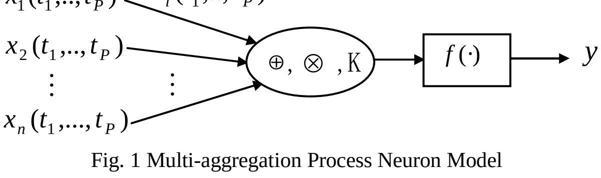

Fig. 1 Multi-aggregation Process Neuron Model

In Fig.1, $w_i(t_1,t_2,...,t_P)$ is multivariate process connection weight function, for $i=1, 2,...,n$; $t_p\in[0,T_p]$. "⊕" is the spatial aggregation operator of n multivariate process input functions; "⊗" is the accumulation operator of multivariate process; $K(\cdot)$ is the multivariate aggregation kernel function; $f(\cdot)$ is the activation function.

If "⊕" picks spatial weighed summary, "⊗" multiple integral on multidimensional process interval

$[0,T_1]\times[0,T_2]\times...\times[0,T_P]$, and the kernel function $K(\cdot)$ is 1, then the input-output mapping relationship of multi-aggregation process neuron is

$$y=f(\int_0^{T_1}\int_0^{T_2}...\int_0^{T_p}\sum_{i=1}^{n}w_i(t_1,t_2,...t_p)x_i(t_1,t_2,...t_p)dt_1dt_2..dt_p-\theta) \tag{1}$$

As known from Eq.(1), the multi-aggregation process neuron model whose the inputs and outputs are both process functions denotes simultaneously spatial weighted aggregation of several multivariate process input signals and phase effect accumulation for already inputted process signals at time $(t_1,t_2,...,t_p)$ and can implement synchronous mapping relationship between inputs and outputs. The model has good adaptability in mechanism for the description of problems of spatio-temporal signal identification and diagnostic analysis of dynamic process.

B The model of Multi-Aggregation Process Neural Network

The Multi-Aggregation Process Neural Network is the Network model, which consists of multi-aggregation process neurons and other types of neurons according to certain structure and information transfer flow are referred as multi-aggregation process neural networks. For briefness of discussion, consider a multi-input-single-output feedback multi-aggregation process neural network with single multi-aggregation process neuron hidden layer defined by Eq.(1). The network structure to be discussed is shown in Fig.2.

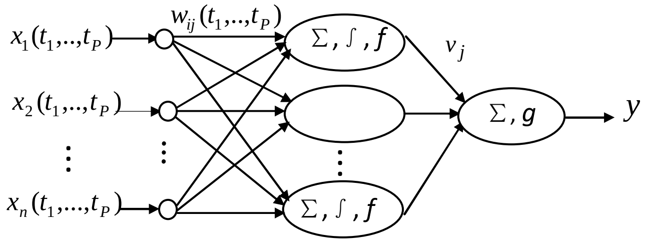

Fig. 2 Multi-aggregation Process Neural Network Model

In Fig.2, $x_i(t_1,t_2,...,t_P)$ $(i=1,2,...,n\ ;\ t_p\in[0,T_p])$ is a *P*-variable process function; $w_{ij}(t_1,t_2,...,t_P)$ ($i=1,2,...,n; j=1,2,...,m$) is the P-variable connection weight function between input layer nodes and hidden layer nodes, m is the number of hidden layer nodes; (Σ, *g*) is a time-invariant neuron in which *g* is the activation function of this neuron; v_j is the connection weight from the hidden layer to the output node.

The input-output relationship of MAPNN denoted by Fig.2 is:

$$y=g(\sum_{i=1}^{m}v_j f(\int_0^{T_1}\int_0^{T_2}...\int_0^{T_p}\sum_{i=1}^{n}w_{ij}(t_1,t_2,...,t_p) \\ x_i(t_1,t_2,...,t_p)dt_1dt_2...dt_p-\theta_j)-\theta) \tag{2}$$

In Eq. (2), θ_j is the activation threshold of nodes of hidden layer neurons, θ is the activation threshold of nodes of output layer neurons.

The neural networks consisting of multi-aggregation process neuron and other types of neurons according to certain structure and information transfer flow are referred as multi-aggregation process neural networks. The same type of neurons has the same structure, shares theory and learning algorithm and carries out the same aggregation operation in the network. At the same time, the information transfer between various types of neuron hidden layers should satisfy the definition of each kind of neuron input/output signal type in multi-aggregation process neural networks. Multi-aggregation process neural networks have parallelity in nature, are an extension of process neural networks only with time dimension, and have quite broad application in practical problems.

The information transformation process between inputs and outputs of Multi-aggregation process neural networks requires multi-aggregation operations in multi-variable and multi-level space and time. However, we know from the information transfer process, the model of information processing has parallelity in nature.

C The objective function of MAPNN training

Suppose that the input function space $\tilde{H}$ of multi-aggregation process neural networks belongs to $(C([0,T_1]\times[0,T_2]\times\cdots\times[0,T_P]))^n$, $b_1(t_1,\ t_2,...,t_P),\ b_2(t_1,t_2,...,t_P),\ ...,b_l(t_1,t_2,...,t_P),...$ are a group of basis functions in $C([0,T_1]\times[0,T_2]\times\cdots\times[0,T_P])$, $X(t_1,t_2,...,t_P)=(x_1(t_1,t_2,...,t_P),\ x_2(t_1,t_2,...,t_P),...,$

$x_n(t_1,t_2,...,t_P))$ is an arbitrary function in $\tilde{H}$ Under the given fitting error accuracy, $x_i(t_1,t_2,...,t_P)$ ($i=1,2,\ ...,n$) is expanded into the finite series form of basis functions [7]:

$$x_i(t_1,t_2,...,t_P)=\sum_{l=1}^{L}a_{il}b_l(t_1,t_2,...,t_P),i=1,2,...,n \tag{3}$$

where ail is the expansion coefficient of $x_i(t_1,t_2,...,t_P)$ corresponding to the basis function $b_l(t_1,t_2,...,t_p)$, L is the number of basis function terms satisfying fitting error accuracy demand.

The connection weight function $w_{ij}(t_1,t_2,...,t_P)$ in Eq.(2) is denoted into the series form of basis functions $b_1(t_1,\ t_2,...,t_P),\ b_2(t_1,t_2,...,t_P),\ ...,b_L(t_1,t_2,...,t_P)$, that is

$$w_{ij}(t_1,t_2,...,t_P)=\sum_{l=1}^{L}w_{ij}^{(l)}b_l(t_1,t_2,...,t_P) \\ ,\ i=1,2,...,n; j=1,2,...,m \tag{4}$$

where $w_{ij}^{(l)}$ is the expansion coefficient of $w_{ij}(t_1,t_2,...,t_P)$ corresponding to the basis function $b_l(t_1,t_2,...,t_P)$ and is an adjustable non-process parameter.

Substitute Eqs.(3) and (4) into Eq.(2), then the input-output transformation relationship of the MAPNN can be expressed as:

$$y = g(\sum_{i=1}^{m} v_j f(\int_0^{T_1}\int_0^{T_2}\cdots\int_0^{T_p}\sum_{i=1}^{n}(\sum_{l=1}^{L} w_{ij}^{(l)} b_l(t_1,t_2,...,t_P)) \\ (\sum_{l=1}^{L} a_{il} b_l(t_1,t_2,...,t_p))dt_1 dt_2 ... dt_P - \theta_j) - \theta) \quad (5)$$

which can be organized as:

$$y = g(\sum_{j=1}^{m} v_j f(\sum_{i=1}^{n}\sum_{l=1}^{L}\sum_{s=1}^{L} a_{il} w_{ij}^{(l)} \int_0^{T_1}\int_0^{T_2}\cdots\int_0^{T_p} b_l(t_1,t_2,...,t_P) \\ \times b_s(t_1,t_2,...,t_P)dt_1 dt_2 ... dt_P - \theta_j) - \theta) \quad (6)$$

Denote

$$B_{ls} = \int_0^{T_1}\int_0^{T_2}\cdots\int_0^{T_p} b_l(t_1,t_2,...,t_P) b_s(t_1,t_2,...,t_p) dt_1 dt_2 ... dt_P \\ , \ l,s = 1,2,...,L \quad (7)$$

Thus, Eq.(6) can be written as

$$y = g(\sum_{j=1}^{m} v_j f(\sum_{i=1}^{n}\sum_{l=1}^{L}\sum_{s=1}^{L} a_{il} B_{ls} w_{ij}^{(l)} - \theta_j) - \theta) \quad (8)$$

Give K learning samples $(x_{k1}(t_1,t_2,...,t_P), x_{k2}(t_1,t_2,...,t_P),..., x_{kn}(t_1,t_2,...,t_P); d_k)$, for $k = 1,\ 2,\ ...,K$;. Assume that the network real output corresponding to the *k*-th learning sample is y_k, the network error function is defined as:

$$E = \sum_{k=1}^{K}(y_k - d_k)^2 = \sum_{k=1}^{K}(g(\sum_{j=1}^{m} v_j f(\sum_{i=1}^{n}\sum_{l=1}^{L}\sum_{s=1}^{L} a_{il}^{(k)} B_{ls} w_{ij}^{(l)} - \theta_j) - \theta) - d_k)^2 \quad (9)$$

where $a_{il}^{(k)}$ is the coefficient of $x_{ki}(t_1,t_2,...,t_P)$ corresponding to $b_l(t_1,t_2,...,t_P)$ in basis function expansion.

In determining the multi-process function of the input range and function-based as the basis function and the input interval have already been given, B_{ls} and $a_{il}^{(k)}$ are constants. E of the objective function of MAPNN training is the function of variable parameters $w_{ij}^{(l)}, v_j, \theta_j, \theta$.

III. Chaos genetic algorithm

Chaos is a nonlinear phenomenon common in nature, which has properties of intrinsic random, orbit ergodicity and implicit rules that can traverse all states repeatedly according to the laws of the system itself within a certain range. Consider a chaotic search strategy based on the insect population model, of which logistic map is a chaotic sequence generator [8, 9], and chaotic state is introduced to the optimization variables. The Iterative equation is as follows:

$$\delta^{j+1} = u\delta^j(1 - \delta^j) \quad (10)$$

where u is a chaotic attractor. When $u = 4$, the system becomes into a chaotic state, a chaos variable of δ^j is emerged, which changes in the interval of $[0,1]$.

Chaotic search is divided into two stages. At first, analyze the track in the whole solution space for ergodic tape tracks produced by Chaotic iterative. When a certain termination condition is satisfied, we consider that the best state found in the search process has been close to the optimal solution, and as the starting point of the second stage of the search. In the second stage, consider the results obtained in the first stage as the center, fine search by additional minor disturbances within the local area until the algorithm satisfies the termination criterion.

Chaos genetic algorithm is to combine the chaotic variables with parameters to be optimized, and to make the optimization search process have property of orbit ergodicity. Then, encode the chaotic variables by the evolutionary mechanism of genetic algorithm, which are represented as chromosomes and are placed in the environment of the problem to select, copy, chaotic cross and chaotic mutation according to the principle of survival of the fittest. According to the evolving of evolutionary iterations repeatedly, the optimal solution is obtained, which is converged to the individual on the most suitable environment finally.

IV. The optimization training of MAPNN based on CGA-LMS algorithm

Based on the above analysis, the training of MAPNN is the process to minimize the parameters of the objective function $E = E(w_{ij}^{(l)}, v_j, \theta_j, \theta)$ and search the optimal solution.

The training algorithm of MAPNN based on CGA-LMS is described as follows.

Step1 Determine the population size N .Generate the initial population G^0 randomly, and encode the chromosomes with decimal numbers, on which the number of genes is the number of variables to be optimized. Set the maximum iterative times M and the error accuracy ε.

Step 2 Build the fitness function. Since the training of MAPNN is the problem to search the minimum value of the objective function, the fitness function is chosen as $fit = e^{-E}$.

Step 3 Select Operations: select the chromosome with runner rules in G^1, of which the selection probability is proportional to its fitness value.

Step 4 Cross of chaos: combine two chromosomes as follows: $W_1' = \vec{\lambda} W_1 + (\vec{1} - \vec{\lambda})W_2$, $W_2' = \vec{\lambda} W_2 + (\vec{1} - \vec{\lambda})W_1$, where λ_i is the chaotic variable and changes in the interval of $(-1, 1)$. Define a cross-amplitude as λ_k . λ_i is determined according to Eq. (11).

In order to show the bidirectional traversing, the chaotic variable of δ_i^{j+1} is determined according to Eq. (12).

$$\lambda_i = \lambda_k \delta_i^{j+1} \tag{11}$$

$$\delta_i^{j+1} = 8\delta_i^j(1-\delta_i^j)-1 \tag{12}$$

Step 5 Mutation operations: Set the mutated gene as w_i , the following as $w_i^{'} = w_i + \beta(w_i^U - w_i)$ or $w_i^{'} = w_i + \beta(w_i - w_i^L)$, where w_i^U is the upper bound, w_i^L is the lower bound, and β is the chaotic variable that changes in the interval of $(-1,1)$. Mutation operations need to define variation amplitude of $\tilde{\lambda}_k$ at first, and then chaos variables are introduced. For *m* individuals elected to be variated, sort in ascending order of fitness. For the *k t*h individual, variation amplitude of $\tilde{\lambda}_k$ can be chosen as Eq. (13), where $\tilde{\lambda}_0$ is the parameter that controls the disturbance size, and β searches in the interval of $[-\tilde{\lambda}_k, \tilde{\lambda}_k]$.

$$\tilde{\lambda}_k = \tilde{\lambda}_0 \exp((m-k)/m) \tag{13}$$

If satisfies the termination condition, save the optimal solution and stop. Otherwise, select the particles whose adaptation degree is before 50%, supply the others to the population in the feasible solution space randomly, make the population reach the given size of N, and generate a new population G^1. Go to Step 3.

V. The application in multi-dynamic recognition of signals

Consider recognition problems of time-varying signals as the following forms of multi-dynamic system $\tilde{H}(t_1,t_2)$ based on MAPNN.

$$\tilde{H}(t_1,t_2) = \{(t_1\sin((a_1t_1 + a_2t_2)\pi), t_2\cos(a_3t_1t_2))\text{；}$$

$$(t_1\sin((b_1t_1 + b_2t_2)\pi), t_2\cos(b_3t_1t_2))\text{;}$$

$$(t_1\sin((c_1t_1 + c_2t_2)\pi), t_2\cos(c_3t_1t_2\pi)\}$$

where $(t_1,t_2) \in [0,1]\times[0,1]$. a_i, b_i, c_i are parameters of signals, where i can be chosen from 1 to 3. a_1, a_2 change in the interval of $[0.5,0.7]$ and a_3 changes in the interval of $[1.0,1.2]$, which build the first class of signals. b_1, b_2 change in the interval of $[0.75,0.95]$ and b_3 changes in the interval of $[1.2,1.3]$, which build the second class of signals. c_1, c_2 changes in the interval of $[1.0,1.2]$ and c_3 changes in the interval of $[1.3,1.5]$,which build the third class of signals.

Build the training sample set of MAPNN as follows: take 30 groups of values of signal parameters of a_i, b_i, c_i from the intervals defined respectively, and build a dynamic sample set of 90 groups. Select 60 groups as the training set, the other 30 groups as the test set. Suppose the output of the first class of signals be 0.0, the output of the second class of signals be 0.5, and the output of the third class of signals be 1.0. Classify the process signals of MAPNN expressed by Eq. (2), and select the network structure as 2-8-1, of which the number of input nodes is 2, the number of hidden nodes is 8, and the number of output nodes is 1. Select the basis function as Binary 5th order polynomial function, the size of population as 60, the Optimization times as 5000, and error accuracy of learning as 0.05. In contrast, train the network according to the learning algorithm based on gradient descent [1] at the same time. Run 10 times with each algorithm, and classify the 30 samples in the test set with the results. The average training time based on CGA-LMS is 57.63 seconds, and the average correct recognition rate is 90%. The average training time based on gradient descent is 94.02 seconds, and the average correct recognition rate is 83.33%. The results show that the algorithm based on LMS-CGA is workable to train MAPNN, and improved the learning properties of MAPNN greatly.

VI. Conclusion

A train method of MAPNN based on chaos genetic algorithm with lowest mean square algorithm is proposed in the paper. According to the aggregation operation and the mapping mechanism of MAPNN in spatio-temporal dimension signal space, design and analyze the objective function of network training.

The functional optimization problem of MAPNN has been transformed to the problem of multivariate function to solve extreme value. The global optimal solution of network parameters is solved in feasible solution space by using the chaos rail to traverse the search of the CGA, which has reference value on the learning of other machine models and the complex function optimization problems.

References

[1] XU Shao-Hua, HE Xin-Gui. The Multi-Aggregation Process Neural Networks and Learning Algorithm [J]. CHINESE JOURNAL OF COMPU TERS, 2007,30(1): 48-55.

[2] Feng Zhenping,Li Jun,Shen Zyda. Application of Genetic Algorithm To Design For Turbine machinery [J].Gas Turbine Technology, 1997,11(2):13-22

[3] Li Jun, Feng Zhenping. Aerodynamic Optimum Design of Transonic Turbine Cascades Using Genetic Algorithms [J]. Journal of Thermal Science，1997,6(2):364-368.

[4] Miao Mingfei, Zhang Yongliang, Ma Jiming. Multi-objective Evolutionary Optimization of Large Differential Surge Chamber [J]. Journal of hydroelectric engineering, 2010, 29 (1) :57-61.

[5] HU Bin,YANG Jing-shu,WANG Li-bin. Improved Chaos Genetic Algorithm and Its Application. Computer Engineering, 2010,36(5):170-172.

[6] WU Tie-bin,CHENG Yun, ZHOU Tao-yun,YUE Zhou. Optim ization Control ofPID Based on ChaosGenetic Algorithm [J]. Computer Simulation, 2009,26(5):202-205

[7] Xu Shaohua, He Xingui. Learning Algorithms of Process Neural Networks Based on Orthogonal Function Basis Expansion [J]. CHINESE JOURNAL OF COMPUTERS, 2004, 27 (5): 645-649

[8] Li Bing, Jiang Weisun. Chaos Optimization Method and Its Application [J]. CONTROL THEORY AND APPLICATIONS, 1997, 14(4): 613-615

[9] Chen Bingrui, Yang Chengxiang. Self-Adapting Chaos-Genetic Hybrid Algorithm and Sensitivity Analysis of Its Parameters [J]. Journal of Northeastern University, 2006, 27 (6) :689-692.