

An Application of Multi-layers FNN with GA by MATLAB in The Air-Conditioning Temperature Control System Identification

Yahui Wang, Xuejun Hao, Xiao Liu

Beijing Institute of Civil Engineering & Architecture, Beijing, China

Abstract: This paper introduces a kind of system identification method based on multi-layers FNN, and it also use GA to improve that method to some degree. We can use MATLAB in the programming process in order to achieve that method. We have applied this vindication method into the temperature controlling mode of air-condition system. The facts illustrate that the idea is very successful.

Keywords: System Identification, Fuzzy Neural Network, Genetic Algorithm, The Air-Conditioning Temperature Control

1. MULTI-LAYERS NEURAL NETWORK

Connecting two or more single neural network we can produce multi-layer neural network. As the structure of the there neural network.

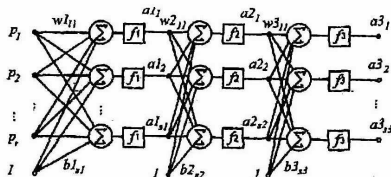


Figure 1. Three layers neural network framework

The structure of the network has r input vector. In the first level it has s_1 nerve cell, and in the second layer it has s_2 nerve cell. Generally speaking, different level maintains different nerve cell number, furthermore, each nerve cell contains a error which the value of its input is 1.

2. USING MATLAB TO CALCULATE OUTPUT OF ARTIFICIAL NEURAL NETWORK (ANN)

In MATLAB environment of ANN's tool box, when single net is composed of input matrix P , weight matrix W and error matrix B , we may easily conclude the output matrix A by using corresponding the activation function. All the calculating process can be accomplished by matrix mode.

3. THE DATA MODEL IDENTIFICATION FOR OBJECT CONTROLLED

3.1 About Input/Output Data and Object Controlled.

In this experiment the controlling subject of neural

network mode should be the air-condition system. This controller is rightly the output of fuzzy neural network in the real process this controller could be the power consumed by chilled water system, turnon magnitude of fan-coil system, turnon magnitude of valve in the air-conditioning system. Anyhow, it represents the total consumption volume or the total refrigerating output of the whole air-condition system.

The input of fuzzy neural network is the temperature measured at the spot in a particular time span. To be more specific, the temperature is selected from 40 continuous data of the auto-recording device (figures which has been taken in 400 minutes). Reasons for this activity as follows: 1) the efficiency of the acting program, for the speed of the training of neural network is very slow therefore the date will be increased consequently the time will also rise at a double rate level. 2) the record of any time period can be prolonged to a day, a month, a season or even a year. Similarly, the temperature date selected in any location can be equally applied to any other location.

The output expected by fuzzy neural network is just the output control of fuzzy neural network. This expecting output is a fictitious date, its specific date is determined by the former input records. The input value is the temperature and the assumed control value is the consumption P of the air-condition system.

3.2 The Application of Neural Network for the Model Identification

In the established neural network, the input vector and the output vector has been already confirmed, the dimension of this two vector are same and each element of this two vector are relatively corresponding. Because of the non-linear change of its input, the output of the network also maintains no regulation. Considering the system which consist of this kind of input and output situation, according to those general analyzing method, the functional relationship of its input and output cannot be established. However,

neural network rightly holds strong advantage in this respect. The input and output of neural network dose not have a direct functional relationship but have reflecting connection. It has a highly non-linear total effect. Each nerve cell of the neural network are receiving a huge number of input data from other nerve cell. The mutual confrontation and mutual affects among the nets realize the reflection relationship between input and output. The special competitiveness of neural network has attributed it in playing a significant role in model identifying process.

3.3 The Identification Program and Explanation on the Object Controlled

The model identification program: plant2.m

Explanation	Program Name
%<the input/output mode of the controlled subject>	the brief of the program
PauseTime=0.1;	make the pause to be 0.1s
load('input.mat')	put in input vector and output vector of the system
x11=a(1:1:40)/100;	normalization of the input vector and put it into "x11"
x21=a(1:1:40)/100;	normalization of the expected output and put it into "x21"
P=x11';	the matrix P equals to the transpose matrix "x11"
T=x21';	the matrix T equals to the transpose matrix "x21"
for i=1:40	the total data number is 40 and circulate 40 times
T(i)=i;	designing the algorithm needed for drawing, make a 40 timing point
end	ending the circulation
plot(T,T,P);	drawing input and expected output curve
title('training data');	the title of the drawing
xlabel('input P');	the name of abscissa
ylabel('expectation output T');	the name of the Y-coordinate
pause	
[R,Q]=size(P);	gaining the dimension of the input matrix 'P'
[S2]=size(T);	gaining the dimension of the expected output matrix 'T'
s1=5;	using 5 hidden points
W20=randn(s2,s1)*0.5;	establishing the weight vector W20, and initial value should be random number

plot(t1,P,t1,purelin(W20*tansig(W10*P,B10),B20));	drawing the input /output Figure of the initial weight value
title('training data');	the title of the drawing
Pause2(pauseTime);	0.1s pause
h=plot(gca,'children');	gaining the object of the sub-handle
h(1:100)	
disp_freq=3;	the displayed the frequency of the study process
max_epoch=2000;	Set training parameter s.
err_goal=0.0005;	the index of the error
tr_err=1; tr_err_goal=0.01;	the changing frequency of the error
err_ratio=1.04;	
IP=plot(t1,P,t1,purelin(W20*tansig(W10*P,B10),B20));	setting the initial value of the floating-point to be 0
flops(0);	calculating the number of the floating-point operation
totalflops=flops;	
%delete(h)	deleting the handle
h=plot(t1,purelin(W20*tansig(W10*P,B10),B20),m');	drawing the training result m');
title('training Result');	the title of the drawing
pause	Pausing
hold on;	adding the presentation on the graph
plot(tr(2,:),t1);title('Study Speed');	drawing the curve of the record of study speed
plot(tr(2,:),t1);title('Study Speed');	adding the error recording curve
plot(t1,P,t1,purelin(W20*tansig(W10*P,B10),B20));	drawing the curve of the input/expected output
h=legend('input expected output,real output');	setting the legend
SSE=sumsqr(T-purelin(W20*tansig(W10*P,B10),B20));	calculating the error quadratic sum of the system

fprintf('\n neural network final: \n')	printing the weight vector W1, B1, W2, B2'
W1,B1,W2,B2,P	Display the weight vector
save chen2 W1 W2 B1 B2 P T;	preserving the weight vector after the calculation
fprintf('The number of training step : %0f.\n',epoch);	printing the training step number
fprintf('floating-point operation times : %0f.\n',flops);	printing the number of the floating-point operation
fprintf('the number of the each average floating-point operation : %0f.\n',round(totalflops/epoch));	printing the number of the each floating-point operation
fprintf('the quadratic sum of the error index : %g.\n',err_goal);	printing the quadratic sum of the error index
fprintf('the real error quadratic sum : %g.\n',SSE);	printing the real error quadratic sum
fprintf('the conclusion of the NN training:'); if SSE<err_goal	printing the conclusion of the NN training. if the real error quadratic sum less than the quadratic sum of error target
disp('Adequately')	the conclusion ought to be suitable
else	Otherwise
disp('Inadequately')	the conclusion ought to be unsuitable
end	ending the judging process

3.4 The Analyse to Model Identification Program

We apply BP network in the nerve cell net of the identification program of the controller model, in addition, we use S and the linear activation function in this network's internal structure, in terms of experiment result we select 5 implicit points.

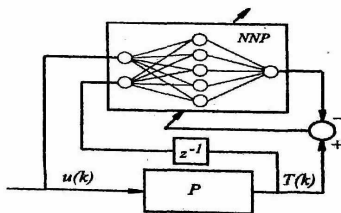


Figure 2. The structure used for identification by ANN

As showing in Figure 2, the relationship of the network as follows:

$$O_i = f(w_i^{(1)} \cdot P + B^{(1)}) = f(\sum_{j=1}^5 w_{ij}^{(1)} \cdot x_j + b_j^{(1)}), i=1,2,\dots,5$$

$$a(k) = f(\sum_{i=1}^5 w_i^{(2)} \cdot o_i + b_i^{(2)})$$

Execute and explain the program as follows: In the identification program plant2.m of the object controlled model, from line 6 to line 9 represent the reading and pre-processing of the data. 'x11=a(1:1:40)/100;' and 'x21=f(1:1:40)/100;' means normalization data. According to this neural network which can only deal with data which contains within [-1, 1], thereby, we ought to divide each element from vector a to f by 100, and then put them into vector X1 and X2. vector P and T are the transpose matrix of x1 and x2 respectively, this two vector respectively symbolize the input of the processed network and expected output of the system. The number of input/expected output data is 40.

In the line13 'plot(ti,T,ti,P);' , we draw input and expected output curve(ti symbolize a vector which consists of 1 to 40, in order to represent 40 abscissa of the time point). As Figure 3 shows, the curve of input and expected output follows.

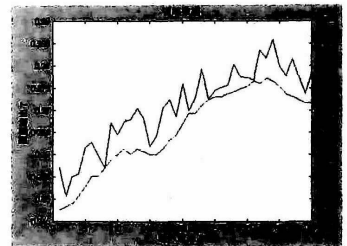


Figure 3. The curve of input and expected output

There is the initialization of weight vector in the programming line of 18 to 23. In these lines W10 and B10 represent implicit layer's weight vector, W20 and B20 represent the output layers' weight vector. As the Figure 4 shows.

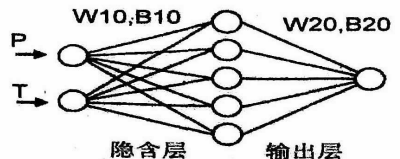


Figure 4. The structure of BP network about the program plant2.m

In Figure 4, weight vector W10 and B10 use N-W initialization condition, the function of this initialization efficiency is to produce Nguyen-Widrow random number from tansig nerve cell. The goal of initialization of N-W is to speedup the training process. And the object of the initialization of tansig nerve cell is to focus the linear zone onto the input space which the input

frequently visited. When using the data of current input /expected output W10, we apply random initialization condition and the training step number is 58; when using N-W initialization condition, training step number is 47 that means the later one's speed increases up to 19 percent comparing with the former one. The initialization condition of weight vector W20 and B20 is random number, because this group of weight vector related to output layer, and the function of output layer is linear activity function: purelin. The line 24 of the program contains the input/output graph when drawing initialization weight, as it is shown in Figure 5. In the picture, we can see that when we do not using optimization method, the output<green> of neural network by using initialization(random) weight vector is quite different from the output(blue) of the network.

Line 30 to line 38 maintains the training program of BP network. Line 6 to line 10 are setting the training factor, which is 'disp_freq=3;' frequency during study period 'max_epoch=2000;' the maximal training times 'err_goal=0.0005;' error target; 'lr=0.01;' , correcting learning speed for weight; 'lr_inc=1.05;' , the increase coefficient of the learning speed; 'lr_dec=0.7;' , the decrease coefficient of the learning speed ; 'err_ratio=1.04;' , error variance ratio.

The factors are designed for the training order 'trainbpa' of BP network. The line 37 '[W1,B1,W2,B2,epoch,TR]=trainbpa(W10,B10,' tansig ', W20,B20,'purelin',P,T,TP);' is the training of the weight of BP with adaptive rate. The left side of the formula are new weight vector W1、B1、W2 and B2; the real training times of network training epoch and error quadratic sum vector tr of the network. The right side of formula are BP network's training order trainbpa and its factors. W10,B10,tansig means that the first layer of the network(implicit layer) use tangent S transform function, and its initialization weight vector. The 36th line 'totalflops=flops' means that putting the times of float operation into variable—totalflops, the times of operation is one of the reference in judging the effectiveness of the algorithm.

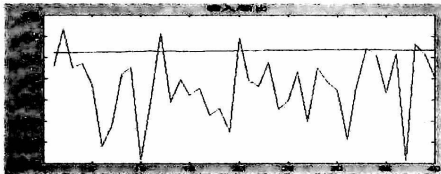


Figure 5. The input/output characteristic diagram of initialization weight

During the training process the change of error will be real-time presented at the map, then the speed rate will be determined by factor disp_freq. After the training it will also show the changes of error quadratic sum (Figure6), the training result (Figure7), the change of learning speed (Figure8), the input/expected output compare with real network output (Figure9) during

the training period.

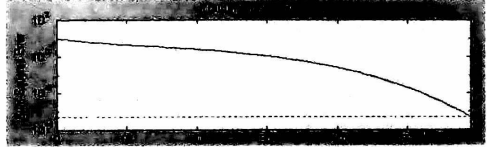


Figure 6. The changes of error quadratic sum about training

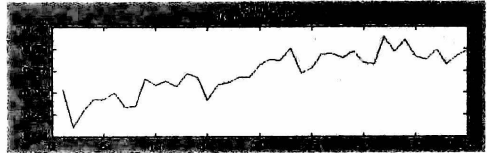


Figure 7. The training result



Figure 8. The change of learning speed

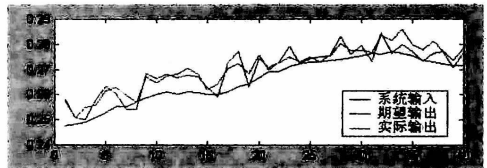


Figure 9. The input/expected output compare with real network output

Noticing from Figure9, the output curve(red) of the network is very approaching to the expected output of the system.

4. TRAINING THE FNN CONTROLLER'S WEIGHT WITH GA

4.1 The Output of FNN Controller

First, the output of FNN is calculated with follow formula:

$$P = [e, ec]^T$$

$$a_{ij}^{(1)} = \exp[-(W_{ij}^{(1)} \cdot x_j + b_{ij})^2] \quad (1)$$

$$a_k^{(2)} = a_{n1}^{(1)} \cdot a_{n2}^{(1)} = \exp[-(W_{n1}^{(1)} \cdot x_1 + b_{n2})^2 + (W_{n2}^{(1)} \cdot x_2 + b_{n2})^2]$$

(2)

$$u = W^{(2)} \cdot A^{(2)} = \sum_{k=1}^{49} a_k^{(2)} \cdot W_k^{(2)} \quad (3)$$

Where $j=1,2; l,m,n=1,2,\dots,r1; k=1,2,\dots,r1 \times r1$, $r1$ is fuzzy flag number. If we put apart of the structure of neural network and directly use theorization and normal distribution membership function by means of multiplication-addition-bary center method, we can also receive the same result. The reason for that matter is that through reflection and combination of fuzzy logic and neural network we can let the structure and location of parameter-membership function which is very essential to fuzzy logic and fuzzy rule to convert into neural network weight. In this way, we can use the self-study and self-adjusting ability of neural network to modify and optimize them, furthermore to achieve the goal of improving the controlling effects of fuzzy logic.

As there is no practical program can be used in the toolbox of neural network, it is necessary to make it into sub-program in order to be selected timely(as confining with the space of the paper, the program don't appear here, readers who interested in this aspect can contact with the author).

4.2 The Calculating Process of FNN Controller Output

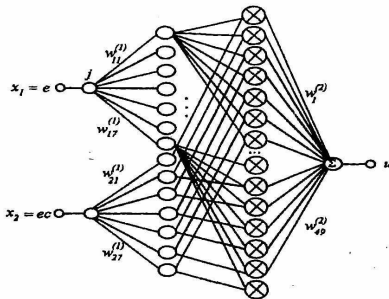


Figure 10. The structure of fuzzy neural network controller

The output vector is:

$$A = f(W \times P + b) = f\left(\sum_{j=1}^r w_j p_j + b\right) \quad (4)$$

This function can be applied to any nerve cell output, that is the output of nerve cell can be write as the input vector P of nerve cell multiply the weight vector W , after the summation, add variation b of weight vector, and then put the result into a transform function f .

4.3 The Result of Optimization by Self-organizing Competition Network

While optimizing training, the saturation of output tracing goal vector shows in Figure11 & Figure12.

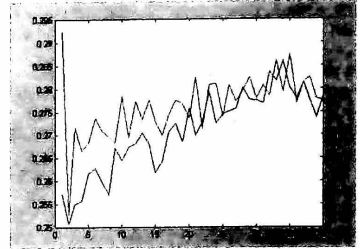


Figure11. The output of network and expect output in system when training in 45th generation.

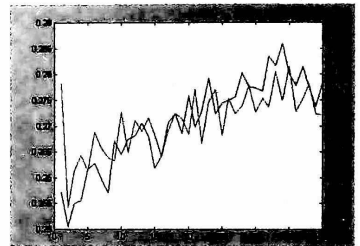


Figure12. The output of network and expect output in system when training in 100th generation.

Judging from the above several Figures we can easily come to the conclusion that along with the processing of the improvement of the training, the output of the network become increasingly approach to the expected output of the system.

5. CONCLUSION

This paper discussed the primitive result of the fuzzy neural network applied in the air-condition controlling system according to the program designing aspect. By comparing the deference between the output through the 40 input data and the expected output of the system we can adjust the network's weight vector furthermore to improve the goal and let the output be closer to the expected result. Concerning the result of identification training method, the advantage of the nerve cell net is obviously distinctive.

REFERENCES

- [1] G.Cembrano Optimal control of a water distribution network in a supervisory control system, *Control Engineering Practice*, March, 2000
- [2] Y.H. Wang etc. Simulation Study of Structure Optimization in Fuzzy Neural Network (FNN) with genetic algorithm (GA) *The Fourth World Congress on Intelligent Control and Automation(WCICA2002)* IEEE Press Shanghai, China, June 10-14, 2002
- [3] Yahui Wang Xuejun Hao Yang Zhao etc. The Study on Automatic Measure Technique in Lab for The Inside of Green Buildings *Journal of Apparatus and Meter*, No.4, 2003