퍼지 및 지능시스템학회 논문지 2004, Vol. 14, No. 2, pp. 216-221

NETLA Based Optimal Synthesis Method of Binary Neural Network for Pattern Recognition

Joon-Tark Lee

School of Electrical, Electronic & Computer Engineering, Dong-A University
840, Hadan 2 Dong, Saha Ku, Busan, Korea, Zip Code:604-714
E-mail : jtlee@daunet.donga.ac.kr

Abstract

This paper describes an optimal synthesis method of binary neural network for pattern recognition. Our objective is to minimize the number of connections and the number of neurons in hidden layer by using a Newly Expanded and Truncated Learning Algorithm (NETLA) for the multilayered neural networks. The synthesis method in NETLA uses the Expanded Sum of Product (ESP) of the boolean expressions and is based on the multilayer perceptron. It has an ability to optimize a given binary neural network in the binary space without any iterative learning as the conventional Error Back Propagation (EBP) algorithm. Furthermore, NETLA can reduce the number of the required neurons in hidden layer and the number of connections. Therefore, this learning algorithm can speed up training for the pattern recognition problems. The superiority of NETLA to other learning algorithms is demonstrated by an practical application to the approximation problem of a circular region.

Key words : optimal synthesis, binary neural network, pattern recognition, Newly Expanded and Truncated Learning Algorithm (NETLA), perceptron

1. Introduction

The binary neural network is a simple feedforward network that include a hard limiter activation function, And it can realize a desired Boolean function with a sufficient number of hidden neurons[1,2]. Specially, the binary neural network based n-tuple method have been used in the image processing fields for the pig evisceration, the scene analysis, and the pattern recognition because of having many potential advantages in the knowledge manipulation. However, the problem of real-time processing in the binary neural network for huge quantities of data was not solved[3]. To solve these problems, various kinds of the learning algorithms have been proposed[4,5].

Recently, EBP algorithm has been applied to the many binary mapping problems.

However, EBP algorithm applied to binary to binary mapping problems results in long training time and inefficient performance. Since the number of neurons in the input layer and the output layer are determined by the dimensions of the input vectors and the output ones, respectively, the abilities of three layer binary neural network depend on the number of neurons in the hidden layer and the number of connections.

Therefore, one of the most important problems in application of three layer binary neural network is to determine the necessary number of neurons in the hidden layer. Recently, various learning algorithms has been proposed to determine the required number of neurons in the hidden layer for binary to binary mapping. Firstly, the learning algorithm called as Expanded and Truncated Learning(ETL) was proposed to train a three layer binary neural network for the generation of binary to binary mapping.

Its basic principle is to find a set of required separating hyperplanes through a geometrical analysis of given training inputs and to determine the number of neurons in the hidden layer. However, there are some disadvantages that requires the tedious sequential learning over all the patterns [6]. Secondly, in the Minimal Sum of Product(MSP) Term Grouping Algorithm(MTGA) the training patterns are represented only in MSP form of the boolean expressions satisfying the Unate property. However, there are also some defects that the not-Unate terms are to be given as additional neurons in the hidden layer [7].

In this paper, a Newly Expanded and Truncated Algorithm (NETLA) is proposed which can be applied even for the case of not-Unate terms. Here the

접수일자 : 2003년 12월 9일
완료일자 : 2004년 2월 5일

boolean expressions are represented as the ESP for an optimal synthesis. Therefore, NETLA does not require any special necessary condition as MTGA's Unate property and can automatically determine the required number of neurons in the hidden layer. Furthermore, by finding the redundancy part of given input patterns, it can reduce the number of connections between the input layer and the hidden layer. Here, the proposed geometrical learning algorithm called as NETLA is applied to train a three layer binary neural network and its superiority over other networks [8],[9],[10] is demonstrated through a practical pattern recognition problem for a circular region approximation.

2. Preliminaries

2.1 The basic unit structure of binary neural network

Assume that the binary training input patterns can be separated by the n-dimensional hyperplanes which is expressed as the net function of Eq.(1) for the n-inputs;

$$net(\boldsymbol{w},\boldsymbol{x},T)=\sum_{i=1}^{n} w_i x_i - T = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n - T = 0 \quad (1)$$

where, x_i is the i-th input, T is the threshold value and w_i is the connection weight between the i-th input and the neuron. In this case, the set for training inputs must be Linearly Separable(LS), and the n-dimensional hyperplanes can be established by n-inputs with the hard-limiter activation function as Eq.(2).

$$y=\begin{cases} 1: & for \ \sum_{i=1}^{n} w_i x_i - T \geq 0 \\ 0: & otherwise \end{cases} \quad (2)$$

where y is the output of neuron.

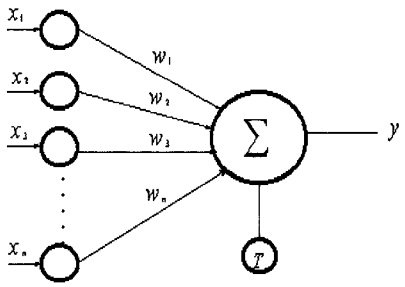

Fig. 1 The basic unit structure of binary neural network

2.2 Decision of the weights and the threshold values in the hidden layer

To decide the weights and the threshold values, we used the Reference HyperSphere(RHS) method. RHS means the separating hyperspheres which enclose all the training patterns and SHS (Separating HyperSphere)

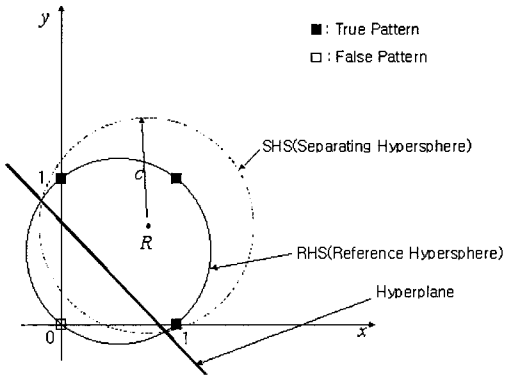

Fig. 2 Example of RHS method for two dimensions

includes only the true patterns. Hyperplanes are obtained by the two intersecting hyperspheres. By using RHS method, we can find the necessary hyperplanes into which the training inputs can be partitioned into the n-dimensional unit hypercube with center, $(1/2,1/2,\cdots,1/2)$ and radius, $\sqrt{n}/2$.

The equation for the RHS is written as Eq.(3).

$$(x_1-\frac{1}{2})^2+(x_2-\frac{1}{2})^2+\cdots+(x_n-\frac{1}{2})^2=\frac{n}{4} \quad (3)$$

We assume the SHS with its center, $(c_1/c_0, c_2/c_0, \cdots, c_i/c_0)$ and radius, R, inside the SHS is written as Eq.(4).

$$(x_1-\frac{c_1}{c_0})^2+(x_2-\frac{c_2}{c_0})^2+\cdots+(x_n-\frac{c_i}{c_0})^2=R^2 \quad (4)$$

Therefore, when these two n-dimensional hyperspheres are intersected, the n-dimensional hyperplanes can be found. The hyperplane equation obtained by subtracting Eq.(4) from Eq.(3) and multiplying by c_0 as follows

$$(2c_1-c_0)x_1+(2c_2-c_0)x_2+\cdots+(2c_n-c_0)x_n-T=0 \quad (5)$$

where, $(c_1\cdots c_n)$ are each binary variables of quantized values. c_0 is the total number in set of included true vertices. These two classes of training inputs can be separated by an n-dimensional hyperplane as Eq.(6)

$$\sum_{i=1}^{n} w_i x_i - T = 0 \quad (6)$$

If the Eq.(5) is equal to the Eq.(6), then each of w_i and T are $2c_i-c_0$ and $R^2c_0-\sum_{i=1}^{n} c_i^2/c_0$. Therefore, the weight values can be calculated as

$$w_i = 2c_i - c_0 \quad (7)$$

And the threshold values calculated as

$$T = R^2 c_0 - \sum_{i=1}^{n} c_i^2 / c_0 \qquad (8)$$

Where, the number of hyperplanes is equal to the number of neurons in the hidden layer.

2.3 Example of decision on weights and threshold values

Suppose, a set of n-bit training input vectors is given and the desired binary output is assigned to each training input vector. Also, suppose two classes of training input vectors (i.e., vertices) which can be separated by an n-dimensional hyperplane as in Eq. (1). Of course, where T is equal to $(t_{\min} + f_{\max})/2$ and $[x_i.]$ is defined as the smallest integer greater than or equal to x_i. And also, we can infer as following.

If $f(x_1, x_2, x_3)$, then there is a separating hyperplane.
If $t_{\min} \le f_{\max}$, then there is no separating hyperplane.

Let us consider a function of three input variables, $f(x_1, x_2, x_3)$. If the inputs are {000, 010, 011, 111}, then roduces '1'. If the inputs are {001, 100}, then $f(x_1, x_2, x_3)$ produces '0'. The inputs {101, 110} are don't care. With the training inputs {000, 010, 011, 111, 001, 100}, $f(x_1, x_2, x_3)$ produces '1' or '0'. And, let $t_{\min}$ be the minimum value of $\sum_{i=1}^{n}(2c_i - c_0)v_t^i$ among all the vertices in (set of included true vertices) SITV, and $f_{\max}$ be the maximum of $\sum_{i=1}^{n}(2c_i - c_0)v_r^i$ among the rest vertices. Where, v_t^i is i-th bit of the true vertex in the set and v_r^i is i-th bit of the training vertex in the set. For this example, $t_{\min} = \min[-3x_1 + x_2 - x_3]$ for the set of included true vertices {000, 010, 011}, and $t_{\min} = 0$. $f_{\max} = \max[-3x_1 + x_2 - x_3]$ for vertices {001, 100 111}, thus $f_{\max} = -1$.

Since $t_{\min} > f_{\max}$ and $T = 0$, the hyperplane $-3x_1 + x_2 - x_3 = 0$ separates the vertices in the set of included true vertices {000, 010, 011} from the rest vertices. For the given example, it turns out that the set of included true vertices are {000, 010, 011}. If all the true vertices are included in the set of included true vertices, then the given function is a linearly separable function and only one neuron is required for the function. Since, the set of included true vertices of the first neuron includes only {000, 010, 011}, the remaining vertices are converted to expand into the first hypersphere. That is, the false vertices {001, 100} are converted into the true vertices, and the remaining true vertex {111} is converted into a false vertex.

Choose one true vertex as {001} and test if the new vertex can be added to the set of included true vertices. It turns out that the set of included true vertices includes all the currently declared true vertices {000, 010, 011, 001, 100}.

Therefore, two neurons are required in the hidden layer. The second required hyperplane is defined as Eq.(5). Where, $c_0 = 5$, $c_1 = 1$, $c_2 = 2$, and $c_3 = 2$. That is, $-3x_1 - x_2 - x_3 - T = 0$. Hence, $t_{\min} = -3$ and $f_{\max} = -5$. Since $t_{\min} > f_{\max}$ and $T = -4$, the required hyperplane is given as $-3x_1 - x_2 - x_3 + 4 = 0$.

Resultantly, two separating hyperplanes are found from the algorithm;that is, two neurons in the hidden layer are set.

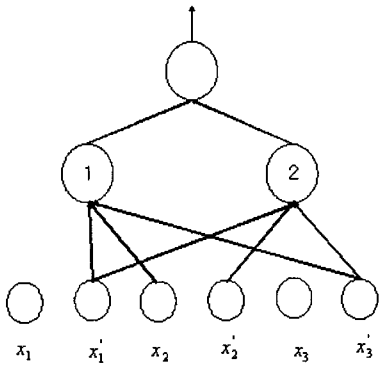

Fig. 3 The structure of three layer binary neural network for given example

Table 2 Weights and threshold values in hidden layer for given example

	1	-6	2	-2	-1
	2	-3	-1	-1	-4

2.4 The advantages of NETLA and Its synthesis process

The synthesis process for NETLA is shown in figure 4. NETLA can synthesize regardless of the given input pattern orders for the binary neural network. The advantages of NETLA are as follows :

i) Since the weights and threshold values can be directly calculated from false and true patterns, so any iterative learning is not needed.

ii) There are no special necessary condition as the Unate property.

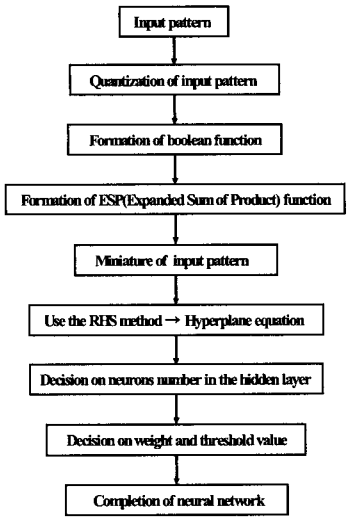

Fig. 4 The block diagram of NETLA

3. Simulations

ETL, MTGA and NETLA are applied to the approximation problem of one circular region as Fig. 5 and the results are stated in the following subsections. Through these applications, we want to show the reduction in number of hidden neurons and connections.

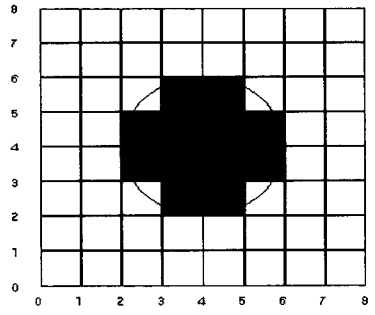

Fig. 5 Circular region obtained by 6-bit quantization

Consider the separation problem of a circular region in 2-dimensional space within a square with sides of length 8 with origin in the lower left corner. A circle of diameter 4 is placed within the square, at (4,4), and then the space is sampled with 64 grid points located at the center of 64 identical squares covering the large square. Of these points, 52 fall outside of the circle (desired output '0') , and 12 fall within the circle(desired output '1'). After two axes (cross and vertical) are quantized as 3 bits, the resultant function before synthesis for the given problem is described as follows.

$$f=(x_1,x_2,x_3,x_4,x_5,x_6)=x_1'x_2x_4'x_5x_6+x_1'x_2x_3x_4'x_5+x_1'x_2x_3x_4x_5'+x_1'x_2x_4x_5'x_6$$
$$+x_1x_2'x_4x_5x_6'+x_1x_2'x_3'x_4'x_5+x_1x_2'x_3'x_4x_5'+x_1x_2'x_4x_5'x_6'$$

where, the $(x_1\cdots x_6)$ means the true patterns and the $(x_1'\cdots x_6')$ means the false patterns.

3. 1 Synthesis result for ETL

The synthesis result for ETL is given as follows

$$f=(x_1,x_2,x_3,x_4,x_5,x_6)=x_1'x_2x_3'x_4'x_5x_6+x_1'x_2'x_3'x_4'x_5x_6+x_1'x_2'x_3'x_4'x_5x_6+x_1'x_2'$$

ETL requires the five neurons in hidden layer as Fig. 6. Their weights and threshold values in the hidden layer are shown in Table 3.

Table 3 Weights and threshold values in hidden layer for ETL

Layer	Unit	Weight						Threshold
Hidden	1	-3	3	1	-3	3	1	7
	2	-29	-3	-1	-3	3	1	-5
	3	-29	-3	-1	-3	3	1	-25
	4	-19	-13	-1	-3	3	1	-20
	5	-16	-16	0	0	0	0	-24

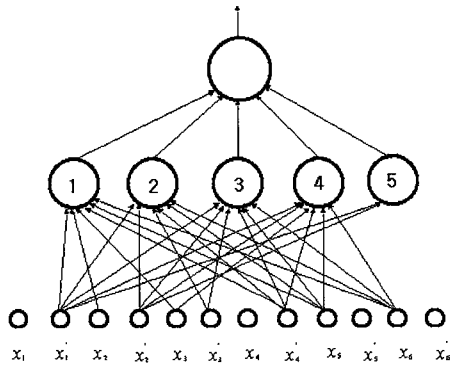

Fig. 6 Structure of three layer neural network for ETL

3.2 Synthesis result for MTGA

MTGA synthesis method represents the function in the MSP form and then checks for the Unate's property. The Unate's property means that a complemented and un complemented variable x_i should not appear simultaneously in the MSP form of the function. The Unate property is explained using the following example. Assume two function $f_1(x_1,x_2,x_3)$ and $f_2(x_1,x_2,x_3)$

$$f_1(x_1,x_2,x_3)=x_1'x_2+x_2'x_3+x_1x_3 \quad : \text{not- Unate}$$
$$f_2(x_1,x_2,x_3)=x_1'x_2+x_2x_3+x_1'x_3 \quad : \text{Unate}$$

The synthesis result for MTGA is as follows

$$f=(x_1,x_2,x_3,x_4,x_5,x_6)=(x_1'x_2x_4'x_5x_6+x_1'x_2x_3x_4'x_5)$$
$$+(x_1'x_2x_3x_4x_5'+x_1x_2x_4x_5'x_6)+(x_1x_2'x_4x_5'x_6'+x_1x_2'x_3'x_4'x_5)$$
$$+(x_1x_2'x_3'x_4x_5'+x_1x_2'x_4x_5x_6')$$

MTGA requires four neurons in the hidden layer as in Fig. 7. Their weights and threshold values in the hidden layer for MTGA are shown in Table 4.

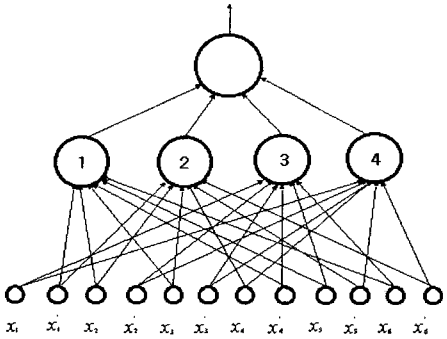

Fig. 7 Structure of three layer neural network for MTGA

Table 4 Weight and threshold values in hidden layer for MTGA

Layer	Unit	Weight						Threshold
Hidden	1	-3	3	1	-3	3	-1	6.5
	2	-3	-3	1	3	-3	1	5.5
	3	3	-3	-1	-3	3	1	5.5
	4	3	-3	-1	3	-3	1	4.5

3.3 Optimal synthesis result for the proposed NETLA

Using NETLA, the resultant synthesis will be given as

$$f(x_1,x_2,x_3,x_4,x_5,x_6) = (x_1'x_2x_3x_4 + x_1'x_2x_4x_6 + x_1x_2'x_3'x_4 + x_1x_2'x_4'x_6')\cdot x_5' + (x_1'x_2x_4'x_6 + x_1'x_2x_3x_4' + x_1x_2'x_4'x_6' + x_1x_2'x_3'x_4')\cdot x_5$$

And each additional hyperplane for the 1st and the 2nd term are extended as follows

$$(x_1'x_2x_3x_4 + x_1'x_2x_4x_6 + x_1x_2'x_3'x_4 + x_1x_2'x_4'x_6')\cdot 1$$

and

$$(x_1x_2'x_4'x_6' + x_1x_2'x_3'x_4' + x_1'x_2x_4'x_6 + x_1'x_2x_3x_4')\cdot 1$$

Therefore, we can obtain the synthesis function as follows

$$\begin{aligned} f(x_1,x_2,x_3,x_4,x_5,x_6) &= (x_1'x_2x_3x_4 + x_1'x_2x_4x_6 + x_1x_2'x_3'x_4 + x_1x_2'x_4'x_6')\cdot x_5' \\ &+ (x_1'x_2x_3x_4 + x_1'x_2x_4x_6 + x_1x_2'x_3'x_4 + x_1x_2'x_4'x_6')\cdot 1 \\ &+ (x_1x_2'x_4'x_6' + x_1x_2'x_3'x_4' + x_1'x_2x_4'x_6 + x_1'x_2x_3x_4')\cdot x_5 \\ &+ (x_1x_2'x_4x_6' + x_1x_2'x_3'x_4' + x_1'x_2x_4'x_6 + x_1'x_2x_3x_4')\cdot 1 \end{aligned}$$

and then,

$$\begin{aligned} f(x_1,x_2,x_3,x_4,x_5,x_6) &= (x_1'x_2x_3x_4 + x_1'x_2x_4x_6 + x_1x_2'x_3'x_4 + x_1x_2'x_4'x_6')\cdot(x_5'+1) \\ &+ (x_1x_2'x_4'x_6' + x_1x_2'x_3'x_4' + x_1'x_2x_4'x_6 + x_1'x_2x_3x_4')\cdot(x_5+1) \end{aligned}$$

where, $(x_5'+1)$ and (x_5+1) become '1'. Consequently, the resultant synthesis function $f(x_1,x_2,x_3,x_4,x_5,x_6)$ is rearranged as follows

$$\begin{aligned} &f(x_1,x_2,x_3,x_4,x_5,x_6) \\ &= (x_1'x_2x_3x_4 + x_1'x_2x_4x_6 + x_1x_2'x_3'x_4 + x_1x_2'x_4'x_6') \\ &\quad + (x_1x_2'x_4'x_6 + x_1'x_2x_3x_4' + x_1x_2'x_3'x_4' + x_1x_2'x_3'x_4') \\ &= (x_1'x_2x_3x_4 + x_1'x_2x_4x_6) + (x_1x_2'x_3'x_4 + x_1x_2'x_4'x_6') \\ &\quad + (x_1'x_2x_4'x_6 + x_1'x_2x_3x_4') + (x_1x_2'x_4'x_6' + x_1x_2'x_3'x_4') \end{aligned}$$

NETLA needs only four neurons in the hidden layer as in Fig. 8. Their weights and threshold values in the hidden layer are shown in Table 5. Thus, we can conclude that NETLA requires the minimum number of connections comparing to ETL and MTGA. Also, in the case of NETLA, the number of neurons in hidden layer is less than the one of ETL.

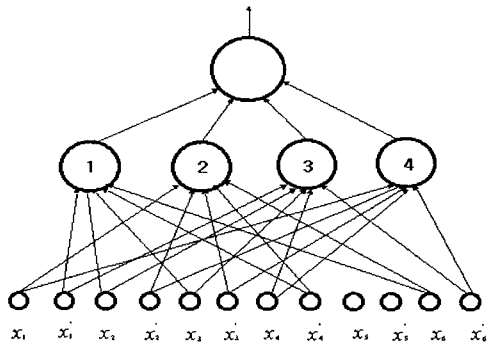

Fig. 8 Structure of three layer neural network for NETLA

Table 5 Weights and threshold values in hidden layer for NETLA

Layer	Unit	Weight					Threshold
Hidden	1	-3	3	1	-3	-1	6.5
	2	-3	-3	-1	3	1	5.5
	3	-3	3	-1	3	1	5.5
	4	3	-3	-1	3	1	4.5

4. Conclusion

Since the NETLA proposed in this paper can automatically calculates the weights and the threshold values from all the true patterns and the false ones, so it dose not require any iterative learning. And also, the proposed NETLA can reduce the number of the required neurons in the hidden layer and their connections. Therefore, the training speed of NETLA is much faster than those of other learning algorithms which requires the troublesome generations for the binary to binary mapping. The superiority of this NETLA to other algorithms was proved by the approximation problem of one circular region. In the future, we are going to apply this algorithm to fingerprint identification, digital image processing and etc.

References

[1] M. Shimada, T. Saito: A GA- Based Learning Algorithm for Binary Neural Networks, IEICE Trans. Fund., vol.E85-A, no.11, pp. 2544-2546, Nov. 2002

[2] N. S. Chaudhari, A. Tiwari: Extending ETL for multi-class output, International Conference on Neural Information Processing, pp.1777-1780, 2002

[3] A. W. Andersen, S. S.Christensen, T. M. Jorgensen: An Active Vision System for Robot Guidance using a Low Cost Neural Network Board. In European Robotics and Intelligent Systems Conference, pp480-488, 1994

[4] P. L. Bartlett, T. Downs: Using Random Weight to rain Multilayer Network of Hard-limiting Units. IEEE Trans. Neural Networks, pp.202-210, 1992

[5] M. L. Brady, R. Rayhavan, J. Slawny: Back Propagation fails to separate where Perceptrons Succeed. IEEE Trans. Circuits Systems, pp. 665-674, 1989

[6] J.H. Kim, S. K. Park, Han, H. Oh, M. S. Han: The Geometrical Learning of Binary Neural Network. IEEE Trans. Neural Networks. vol. 6.no. 1. pp 237-247, 1995

[7] Z. Kohavi.: Switching and Finite Automata Theory 2nd Ed. McGraw-Hill, 1986

[8] S. Park, J. H. Kim, H. Chang: A Learning Algorithm for Discrete Multilayer Perceptron. in Proc. Int. Sysp. Circuit Systems. Singapore, 1991

[9] Donald L. Gray, Anthony N. Michel: A Training Algorithm for Binary Feedforward Neural Networks. IEEE Trans. Neural Networks),pp.176-194, 1992

[10] J.T. Lee, S.K. Sung: Optimal Synthesis Method for Binary Neural Network using NETRA, Advances in Soft Computing AFSS 2002, pp.236-244, 2002

저 자 소 개

Joon-Tark Lee[李 浚 栻]

He received the B.S. and M.S. degrees in electrical engineering from Dong-A University in 1979 and 1981. And he received the Ph. D. Degree in electrical engineering from Chung-Ang University in 1988. From 1983 to 1985, he was a researcher in LG Electronic Co. Ltd. Since 1985, he has been a Professor of School Electrical, Electronic and Computer Engineering at Dong-A University. From 1997 to 1998, and 2002, he was a Visiting Professor of Tsukuba University in Japan. His research interests are in Fuzzy Theory, Neuro-Science based Neural Network, GA and Intelligent Controller Design including Stability Analysis.