2008 International Symposium on Nonlinear Theory and its Applications
NOLTA'08, Budapest, Hungary, September 7-10, 2008

A Learning Algorithm of Binary Neural Networks Based on Real-Coded GA

Yu Akedo† , Hidehiro Nakano† and Arata Miyauchi†

†Musashi Institute of Technology,
1-28-1,Tamazutsumi,Setagaya-ku,Tokyo,158-8557 Japan
Email: akedo@ic.cs.musashi-tech.ac.jp

Abstract—In this paper, we propose a learning algorithm of Binary Neural Networks (BNNs) based on real-coded genetic algorithm. The algorithm encodes parameters of hidden layer neurons as individuals consisting of real number vectors. The proposed algorithm can reduce the number of hidden layer neurons and has high recognition performance, comparing with the conventional algorithm. Through basic numerical experiments, we verify these advantages.

1. Introduction

A Binary Neural Network (BNN) is a kind of feedforward neural networks whose input and output are binary. The BNN can express any desired Boolean functions if a sufficient number of hidden layer neurons are provided [1][2]. Application examples of the BNN have been discussed for pattern classification [3] and error correcting codes [4]. As the learning algorithm of the BNN, many methods have been proposed ,e.g., ETL [1], IETL [2], and GAETL which are simple geometrical learning methods. In these methods, GAETL can reduce the number of hidden layer neurons and can suppress parameter dispersion. GAETL uses GA operations to decide hidden layer neurons. The BNN can be implemented on simple digital hardware consisting of parameter registers, AND gates, adders, and comparators [5]. Also, GA can be implemented on simple digital hardware consisting of logical gates. Therefore, GAETL can be expected to realize small-scale digital hardware such as embedded systems. However, the conventional GAETL uses a simple bit coding GA. In the bit coding GA operations, continuity of decision parameters can not be guaranteed. It might cause creation of redundant hidden layer neurons.

In this paper, we propose Real-Coded GAETL (RC-GAETL) which uses Real-Coded GA (RCGA), and consider its learning performance for the BNNs. The RC-GAETL encodes parameters of hidden layer neurons as individuals which consist of real number vectors considering continuity of decision parameters. We then apply BLX-α [7] and SPX [8] to RCGAETL as crossover operators for individuals. The RCGAETL can reduce the number of hidden layer neurons and has high recognition performance, comparing with the conventional algorithm. Through basic numerical experiments, we verify these advantages.

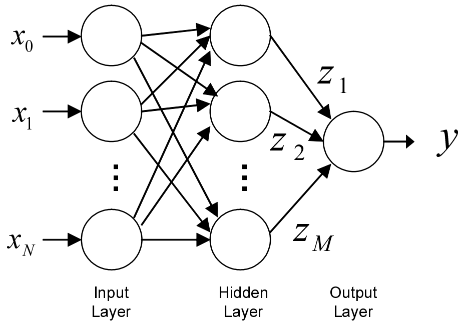

Figure 1: Three layer BNN.

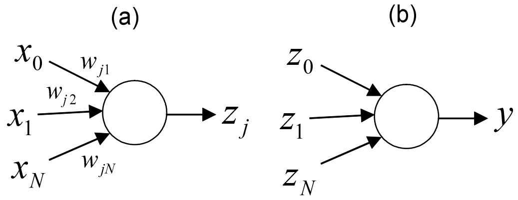

Figure 2: (a) Hidden layer neuron. (b) Output layer neuron.

2. BNN and GAETL

Figure 1 shows a three layer feedforward BNN. Figure 2 shows hidden layer neuron and output layer neuron consisting of the BNN. The model of BNN is described by the following equations.

$$z_i = S\left(\sum_{j=1}^{N} w_{ij}x_j - T_i\right) \tag{1}$$

$$y = S\left(\sum_{i=1}^{M} z_i - 1\right) = \begin{cases} 0 & if\ \ z_i = 0\ \ for\ all\ i \\ 1 & otherwise \end{cases} \tag{2}$$

$$S(X) = \begin{cases} 1 & (X \geq 0) \\ 0 & (X < 0) \end{cases} \tag{3}$$

where w_{ij} is a weight parameter between the i-th hidden layer neuron and the j-th input layer neuron, and T_i is a threshold parameter of the i-th hidden layer neuron. N is the number of input layer neurons and M is the number of hidden layer neurons. As the learning algorithm of the BNN, we use GAETL [6]. Figure 3 shows the flowchart of

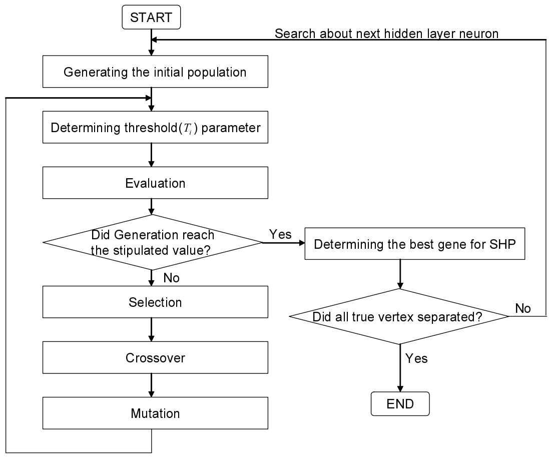

Figure 3: The flowchart of GAETL.

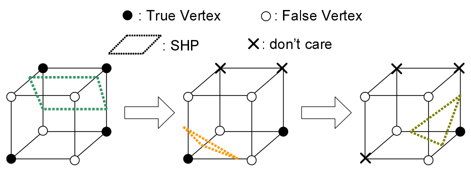

Figure 4: Learning process of BNN.

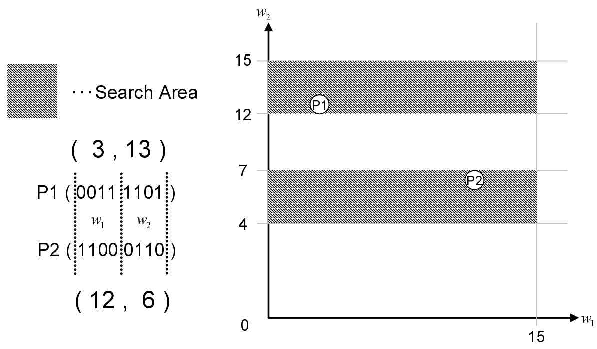

Figure 5: Search area in binary coding.

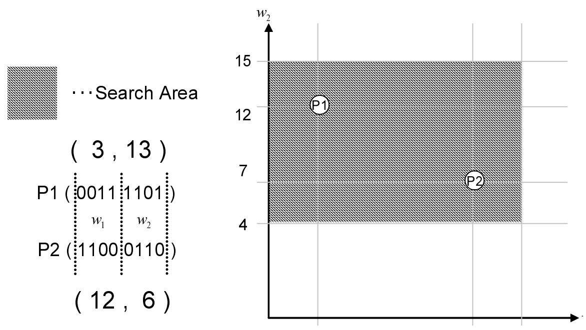

Figure 6: Search area in BLX-α.

GAETL. First, we consider that teacher signals are vertices of hypercube in N dimensional space. "true vertex" and "false vertex" are the teacher signals for which output of BNN are "1" and "0", respectively. Next, we find Separating Hyper Planes (SHP) to divide these vertices. Separated vertices are regarded as "don't care" in searching the next SHP. Found SHP is expressed as follows

$$\sum_{j=1}^{N} w_{ij}x_j - T_i = 0 \tag{4}$$

The formula of SHP (4) corresponds to hidden layer neuron. Therefore, finding SHP leads to the learning of BNN. As SHPs separate all vertices, learning finishes (see Figure 4).

GAETL uses GA to find normal vectors of SHPs as shown in Figure 3. In the conventional GAETL, a normal vector is coded as individuals consisting of binary strings. It should be noted that binary coding might decrease learning efficiency. Because, the binary coding and uniform crossover do not consider decision parameter continuity. Figure 5 shows an example of such a problem. As shown in the figure, parent individuals P1 and P2 can generate child individuals in the discontinuous gray area. Such a characteristic might decrease learning efficiency for BNN by finding normal vectors of SHPs.

3. Proposed Method

In order to overcome the above problem, we propose Real-Coded GAETL (RCGAETL) which uses Real-Coded GA (RCGA) for the learning of BNN.

The gene in RCGA is coded by real number. Crossover operator has a significantly different characteristic between RCGA and GA. RCGA makes the child individuals based on the position of the parent individuals in the real number space. As shown in Figure 6, parent individuals P1 and P2 generate child individuals in the continuous gray area. Such a decision parameter continuity can improve learning efficiency for BNN. As the operator which has continuity, various methods such as BLX-α[7] and SPX[8], UNDX-m [9] have been proposed. In this paper, we select BLX-α and SPX as operators and compare learning performance for BNN.

Next, we explain a gene expression method to adapt RCGA to GAETL. In the convensional GAETL, individuals consist of simply connected binary strings of w_{ij}. In the same way, in the proposed method, individuals consist of simply connected real number of w_{ij} (see Figure 7).

4. Experiments

In order to confirm effectivity of the proposed method, we perform numerical experiments. We compare four

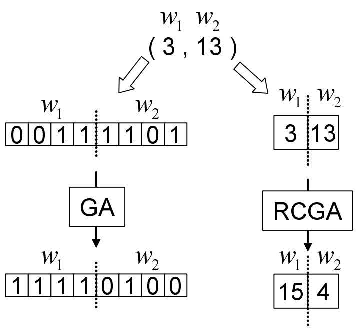

Figure 7: Gene expression of GAETL and RCGAETL.

Table 1: Parameters of each method.

	Generation	Population
GA,binary coding	1000	50
GA,gray coding	1000	50
RCGA,BLX-α	1000	75
RCGA,SPX	1000	180

learning methods. Table 1 shows the methods and their parameters. Binary coding and gray coding use uniform crossover and 1 % mutation. We show average values for 100 trials as experimental values.

We use MGG [10] as a generation alternation model for all the methods. The number of child individuals for MGG is set to 120.

4.1. A Number of Hidden Layer Neurons

In order to confirm the learning efficiency, we evaluate the number of hidden layer neurons in completing learning of BNN. We regard the number of hidden layer neurons as an evaluated values, because the reduction of hidden layer neurons corresponds to the reduction of circuit scale in implementing hardware of BNN.

Figure 8 shows 12bit Circular Region which is a high-level quantization problem and easy to separate vertices. In the black and white regions, teacher signals are true and false vertices, respectively. This problem has 12 dimensional intputs, and does not have “don’t care”. Therefore, the number of true and false vertices are 4096.

Table 2 shows the results for the number of hidden layer neurons in 12bit Circular Region.

BLX-α and SPX show almost the same results. As we compare proposed method with conventional method, the proposed method can reduce the number of hidden layer neurons.

Figure 9 shows Gaussian Problem which is a high-level quantization and difficult to separate vertices. True and false vertices are distributed in 2 dimensional space with a normal distribution. This problem has 12 dimensional inputs. The number of true vertices are 512, and the number of false vertices are 512. The other vertices are “don’t care”.

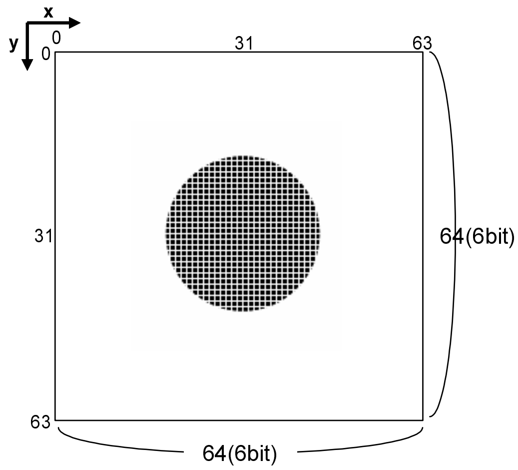

Figure 8: 12bit Circular Region Problem.

Table 2: Number of hidden layer neurons in 12bit Circular Region.

	Ave	Min	Max
GA,binary coding	16.13	13	20
GA,gray coding	12.48	12	15
RCGA,BLX-α	12.37	12	14
RCGA,SPX	12.39	12	14

Table 3 shows the results for the number of hidden layer neurons in Gaussian Problem.
The number of hidden layer neurons in the proposed method are smaller than that in the conventional method. In all the methods, the number of hidden layer neurons in binary coding is the largest and that in BLX-α is the smallest.

4.2. Generalization Ability

In order to confirm generalization ability, we evaluate recognition rate for training data. For the Gaussian Problem, let 50% vertices be training data, and let the other 50%

Table 3: Number of hidden layer neurons in Gaussian Problem.

	Ave	Min	Max
GA,binary coding	10.84	10	12
GA,gray coding	10.40	9	13
RCGA,BLX-α	10.17	9	11
RCGA,SPX	10.35	9	12

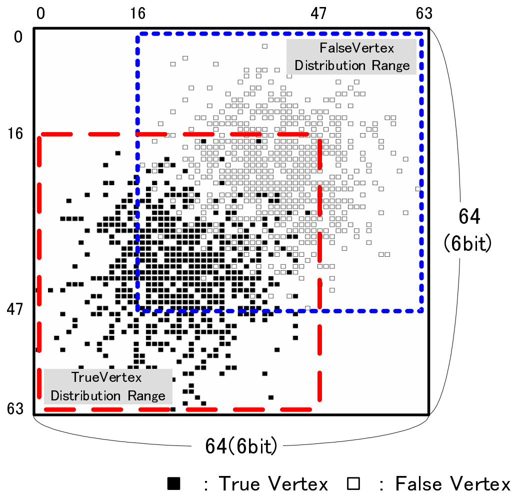

Figure 9: Gaussian Problem.

Table 4: Recognition rate. [%]

	True vertices	False vertices
binary coding	89.4	86.8
gray coding	90.2	86.7
BLX-α	90.5	86.9
SPX	89.8	87.0

vertices be test data. Table 4 shows the recognition rate for true and false vertices.
Recognition rate in proposed method is almost the same as that in conventional method, although the number of hidden layer neurons are reduced. All the rate for true vertices are higher than that for false vertices. These results are caused by the algorithm of GAETL; GAETL determins SHPs focusing on only true vertices.

5. Discussion

The results show that BLX-α is more effective learning method for BNN than SPX. But, it has been reported that searching ability in BLX-α decreases since this method might search ineffectual space of solution for a problem with variable dependence [11][12]. SPX does not have such an influence. However, SPX requires the larger number of population as the dimension of problems is higher. Increasing population causes decreasing evolutional speed. To get better performance by SPX, longer generation is required for determining each SHP. As learning method for BNN, BLX-α is simpler and has better learning performance than SPX. It can also contribute to hardware implementation of RCGAETL. Therefore, BLX-α is more suitable learning method of BNN than the other methods.

6. Conclusion

In this paper, we have proposed RCGAETL which considers decision parameter continuity. The results show that the proposed method can decrease the number of hidden layer neuron. In addition, they show that the proposed method has high performance in recognition rate although the number of hidden layer neurons are reduced. Thus, the proposed method is more effective learning method than the conventional method.
Future problems include (1) consideration of larger scale problems and multi-class recognition problems, and (2) hardware implementation of RCGAETL.

References

[1] J.H.KIM and S.K.Park,"The geometrical learning of binary neural networks" *IEEE Trans. Neural Networks*, vol.6, no.1, pp.237–247, 1995.

[2] A.Yamamoto and T.Saito,"A flexible learning algorithm for binary neural networks" *IEICE Trans. Fundamentals*, vol.E81-A, no.9, pp.1925–1930, Sept, 1998.

[3] S.Gazula and M.R.Kabuka,"Design of supervised classifiers using Boolean neural networks" *IEEE Trans. Pattern Anal. & Mach. Intell.*, Vor.17, No.12, pp1239–1246, 1995.

[4] X.-A. Wang and SB.Wicker,"An artifical neural net Viterbi decoder" *IEEE Trans. Commu,* Vol44. No2, pp.165-171, 1996.

[5] B.Amine,"A Compact 3-D VLSI Classifier Using Bagging Threshold Network Ensembles" *IEEE Transactions on Neural Networks*, Vol.14, No.5, pp.1097–1109, 2003.

[6] M.Shimada and T.Saito,"A GA-Based Learning Algorithm for Binary Neural Net-works" *IEICE Trans*,Fundamentals, vol.E85-A, No.11,p2544 2002.

[7] Eshleman, L. J. and Schaffer, J. D.,"Real-Coded Genetic Algorithms and Interval-Schemata" *Foundations of Genetic Algorithms 2*, pp.187–202, 1993.

[8] T.Higuchi, S.Tsutsui and M.Yamamura,"Simplex Crossover for Real-coded Genetic Algolithms" *The Japanese Society for Artificial Intelligence*, vol.16, No.1, pp.147–155, 2001. (in Japanese)

[9] H.Kita, I.Ono and S.Kobayashi,"Multi-Parental Extension of the Unimodal Normal Distribution Crossover for Real-Coded Genetic Algorithms" *T-SICE*, vol.36, No.10, pp.875–883, 2000. (in Japanese)

[10] H.Satoh, I.Ono and S.Kobayashi,"A New Generation Alternation Model of Genetic Algorithms and Its Assessment" *The Japanese Society for Artifical Intelligence*, vol.12, No.5, pp734–744, 1997. (in Japanese)

[11] I.Ono, M.Yamamura and S.Kobayashi,"A Genetic Algorithm with Characteristic Preservation for Function Optimization" *Proceedings of the 4th International Conference on Soft Computing (IIZUKA '96)*, pp511–514, 1996.

[12] R.Salomon,"Performance Degradation of Genetic Algorithms Under Coordinate Rotation" *Proceedings of 5th Annual Conference on Evolutionary Programming*, pp.155–161, 1996.