A HYBRID MULTILAYER NEURAL NETWORK FOR BINARY PATTERN CLASSIFICATION AND ITS LOW-BIT LEARNING ALGORITHM

Kenji NAKAYAMA Hiroshi KATAYAMA
Dept. of Electrical and Computer Eng., Faculty of Tech., Kanazawa Univ.
2-40-20, Kodatsuno, Kanazawa, 920 JAPAN

ABSTRACT This paper proposes a new hybrid multilayer neural network and its low-bit learning algorithm for binary pattern classification. In the training process, a single layer network is first employed. If the training does not converge, then a middle unit is assigned to a critical pattern. Connection weights are adjusted so that this unit responds to the critical pattern. The training is repeated by increasing the middle units. Thus, the number of middle units can be optimized. Connection weights are adjusted using a small number of bits, resulting in very simplified digital hardware. Since the outputs of the middle and the output layers are specified, a single layer low-bit learning algorithm is proposed. The training is very fast and is insensitive to initial weights and parameters. A divided form is proposed, in order to drastically save the middle units for a large number of the patterns. A bit sequence of the pattern is partitioned into several groups. They are independently processed in the early stage, and are combined in the final stage. The proposed method can be applied to a variety of binary pattern classification problems.

I INTRODUCTION

Multilayer neural networks are attractive for pattern recognition and classification. Back-propagation and its modified version [1] have been successful to some extent. However, many problems still remain. For example, there is no general rule to determine the optimum number of the hidden units. In complicated pattern classification, such as the parity problem, training convergence is highly dependent on the initial weights and parameters. We must try so many times by changing the the initial weights and the parameters. Furthermore, reductions in hardware are also important in actual applications. This, however, is rather difficult based on the BP algorithm.

A digital hardware realization is one hopeful approach. Variable connection weights, complicated learning algorithms and a large memory capacity can be easily realized, with desired accuracy.

In order to simplify digital hardware, it is necessary to decrease the number of units, connections and bits, while maintaining the desired performance. For this purpose, several low-bit learning algorithms have been proposed [2]-[6]. However, efficiency of these methods are rather limited to some extent.

In this paper, a new hybrid multilayer neural network is proposed for binary pattern classification. Middle units are used to isolate critical patterns. The low-bit learning algorithm [5] is basically employed. However, several improvements have been developed for the proposed hybrid form. Furthermore, a divided form is proposed in order to save the middle units.

0-7803-0559-0/92 $3.00 © 1992 IEEE

II A HYBRID MULTILAYER NEURAL NETWORK

2.1 Binary Pattern Classification

In this paper, binary patterns are taken into account. They are expressed with N_B bits. So, the number of all possible patterns is 2^{N_B}. Let the number of patterns to be considered be N_P, which satisfies $N_P \leqq 2^{N_B}$. N_P binary patterns are classified into N_C classes. N_C is equal to or less than N_P, $N_C \leqq N_P$. If N_C is equal to N_P, then the problem is referred as pattern recognition. On the other hand, if N_C is less than N_P, then it is referred as pattern classification. Both cases are categorized into pattern classification in this paper.

2.2 Hybrid Multilayer Neural Network

A blockdiagram of the proposed hybrid multilayer neural network is shown in Fig.1. It consists of the input layer, the middle layer and the output layer.

Let the connection weights from the ith input unit to the jth middle unit and the jth output unit be w_{IMij} and w_{IOij}, respectively. Furthermore, let the weight from the ith middle unit to the jth output unit be w_{MOij}. Relations in the blockdiagram shown in Fig.1 are as follows:

$$x_{Mj} = \sum_{i=1}^{NB} w_{IMij} y_{Ii} \tag{1a}$$

$$y_{Mj} = f(x_{Mj}) \tag{1b}$$

$$x_{Oj} = \sum_{i=1}^{NB} w_{IOij} y_{Ii} + \sum_{i=1}^{NM} w_{MOij} y_{Mi} \tag{2a}$$

$$y_{Oj} = f(x_{Oj}) \tag{2b}$$

$$f(x) = \begin{cases} 1, & x \geqq \theta + \alpha \\ 0, & x \leqq \theta - \alpha \end{cases} \tag{3}$$

where, x and y mean the input and the output of each unit, respectively. f(x) is a hysteresis threshold function.

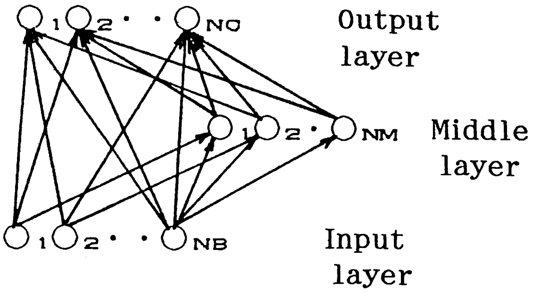

Fig.1 Blockdiagram of hybrid multilayer neural network.

2.3 Role of Middle Units

A single middle unit responds only to one of critical patterns, which are defined in Sec.2.4. An output unit also responds to only one of the categories. Connection weights are determined so as to make the above conditions possible.

A network, in which N_M is equal to N_P, is referred to a full middle unit model. In this network, all patterns have the corresponding middle unit. Therefore, binary pattern classification problems can be always solved. Because all binary patterns can be completely separated in the middle layer, as will be described in Sec.2.5. On the other hand, the simplest model is a single layer network. The hybrid model locates between them. Although we can find the same structure in the reference [1], there is essential differences between the proposed and the conventional methods. That is, in the proposed method, the middle unit is assigned to one of the patterns. Furthermore, it is added in the training process. Thus, the number of the middle units is optimized.

2.4 Critical Patterns

If a set of patterns is linearly separable, then no middle unit is required. On the other hand, if linearly non-separable patterns are included, then the middle units are required.

Figure 2 shows an example of a critical pattern. In this example, a pattern is a two dimensional vector (x,y). A pattern $\Diamond_C$ is surrounded by different patterns ○. It is difficult to separate this pattern $\Diamond_C$ from the other using a multilayer neural network, in which the unit input is determined by a sum of products, and the output is determined through some logistic function. Therefore, this kind of pattern is called 'critical pattern' in this paper.

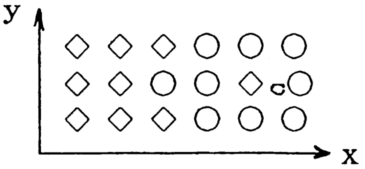

Fig.2 An example of critical pattern $\Diamond_C$.

This pattern, however, can be easily isolated by assigning a middle unit. That is, this middle unit responds only to the critical pattern. After adding the middle unit for $\Diamond_C$, pattern classification can be essentially carried out except for the critical pattern $\Diamond_C$.

In the parity problem, for instance, all possible patterns are taken into account. Therefore, there are many critical patterns. After removing one critical pattern, another critical pattern is selected, except for previous critical patterns. As a result, the selected critical patterns locate far from each other.

On the other hand, when N_P is much less than 2^{NB}, the number of critical patterns is very small. For example, alphabet and digit patterns with 16x16 dots do not need any middle unit. This can be rephrased as, these patterns can be classified using a single layer neural network.

2.5 Complete Separation of Binary Patterns

Here, we consider separation of all binary patterns: $p=(p_1,p_2,\dots,p_{NB})$, using a single layer network. In this network, N_P is equal to 2^{NB}. Thus, *the* middle unit is prepared for all patterns, that is the full middle unit model.

Arbitrary pattern **p**(m) can be isolated by using the following N_B-dimensional hyperplane.

$$\sum_I p_I - \sum_J (p_J - 1) - \theta = 0 \qquad (4)$$

$$p_I = 0,\ p_J = 1 \qquad (5)$$

The above hyperplane is replaced by connection weights in the single layer network shown in Fig.3. The upper layer is referred to a middle layer here. Letting the mth middle unit respond to p(m), connection weights from the offset unit, the ith unit and the jth unit in the input layer to the mth unit in the middle layer, denoted by w_{Offm}, w_{Im} and w_{Jn}, respectively, are determined by

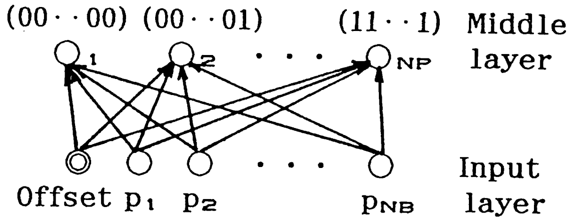

Fig.3 Complete separation of binary patterns.

$$w_{Offm} = \theta - N_J, \qquad (6a)$$

N_J: the number of p_J, $0 < \theta < 1$

$$w_{Im} = -1 \qquad (6b)$$

$$w_{Jm} = 1 \qquad (6c)$$

Threshold is zero.

2.6 Examples of Hybrid Multilayer Neural Network

(1) XOR Problem

Figure 4 shows location of a binary vector (x,y). If a vector (x,y)=(1,1) can be removed by adding the middle unit, then odd and even vectors can be separated by a straight line, as shown in Fig.4.

Two cases are shown below, in which a vector (1,1) or (1,0) is employed as the critical pattern. Figure 5 shows the hybrid form, which solves the XOR problem. The middle unit responds only to $(x,y)_M$=(1,1) or (1,0). Examples of the weights are listed in Table 1. Threshold is 0. That is, f(x)=1, $x \geqq 0$ and f(x)=0, $x<0$.

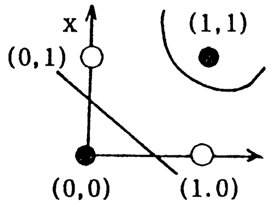

Fig.4 Location of binary vector (x,y).

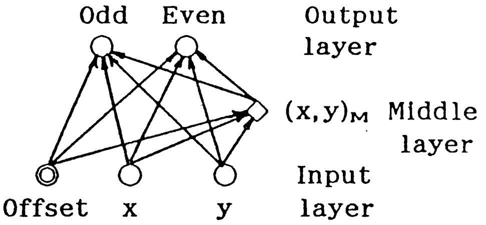

Fig.5 An example of hybrid multilayer neural network to solve XOR problem.

Table 1 Examples of connection weights in the hybrid form in Fig.5.

(a) $(x,y)_M$=(1,1)

<To>	Odd	Even	Middle
<From>			
Offset	-0.5	0.5	-1.5
x	1	-1	1
y	1	-1	1
Middle	-2	2	/

(b) $(x,y)_M$=(1,0)

<To>	Odd	Even	Middle
<From>			
Offset	-0.5	0.5	-0.5
x	-1	1	1
y	1	-1	-1
Middle	2	-2	/

(2) Three Bit Parity Problem

Three dimensional vectors (x,y,z) locate on the vertex of the cube as shown in Fig.5. In this case, by isolating the vectors (000) and (111) with two middle units, the rest of vectors can be separated into the odd and even categories, using a plane given by x+y+z=1.5, as shown in the same figure.

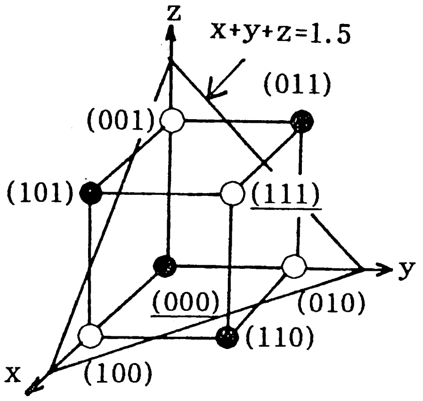

Fig.6 Locations of binary vectors (x,y,z).

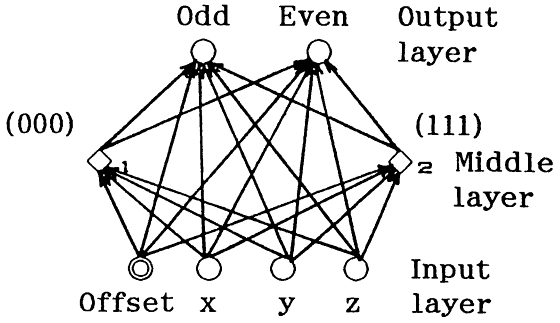

Fig.7 Hybrid form to solve 3 bit parity problem, using two middle units.

The hybrid multilayer neural network, in which the middle units are assigned to the vectors (000) and (111), is illustrated in Fig.7. Examples of the connection weights are listed in Table 2. Threshold is 0.

We have tried a 4bit parity problem on computer simulation by the proposed learning algorithm. Seven middle units were needed at least.

Table 2 Example of connection weights of the hybrid form shown in Fig.7.

<To>	◇₁	◇₂	Odd	Even
<From>				
x	-1	1	-1	1
y	-1	1	-1	1
z	-1	1	-1	1
Offset	0.5	-2.5	1.5	-1.5
◇₁	/	/	0	1
◇₂	/	/	2	-2

III A LEARNING ALGORITHM

3.1 General Follow Chart

A single layer network is used at first. The connection weights are adjusted through a low-bit learning algorithm, which will be described in the next section. If the training process does not converge, then the critical pattern is selected. A middle unit is assigned to this pattern, and adjusting weights is repeated. The above process is further repeated by increasing the middle units until the training converges.

3.2 Low-Bit Learning Algorithm

In the proposed hybrid model, the outputs of the middle units and the output units can be specified before training. Therefore, the low-bit learning algorithm, using hidden layer target [5], can be basically employed. The weights w_{IMIJ} are first adjusted. After that, the weights w_{IOIJ} and w_{MOIJ} are adjusted, using fixed w_{IMIJ}.

(1) Adjusting w_{IMIJ}:

Let the critical pattern be $p_C(m)$, which is applied to the input layer. The input of the middle unit is calculated by Eq.(1a). Let the mth middle unit be assigned to $p_C(m)$. The weights are adjusted as follows:

$$w_{IMIJ}(n+1) = w_{IMIJ}(n) + \Delta w_{IM} \tag{7}$$

$$j=m,\quad \text{If } x_{MJ}(n) \geqq \theta+\alpha \quad \text{then} \quad \Delta w_{IM} = 0 \tag{8a}$$

$$\text{If } x_{MJ}(n) < \theta+\alpha \quad \text{then} \quad \Delta w_{IM} = \mu \cdot q \cdot y_{II} \tag{8b}$$

$$j \neq m,\quad \text{If } x_{MJ}(n) > \theta-\alpha \quad \text{then} \quad \Delta w_{IM} = -\mu \cdot q \cdot y_{II} \tag{9a}$$

$$\text{If } x_{MJ}(n) \leqq \theta-\alpha \quad \text{then} \quad \Delta w_{IM} = 0 \tag{9b}$$

where q is a quantization step, and η is a step size, which takes an integer number. α is hysteresis threshold, which is gradually increased in the training process [5]. The initial guess of w_{IMIJ} is zero.

The above process is repeated for all critical patterns until Δw_{IM} becomes 0.

(2) Adjusting w_{IOIJ} and w_{MOIJ}:

All patterns are taken into account in this step. A pattern p(m) is applied to the input layer. The input of the output unit is calculated by Eq.(2a). Let the mth output unit is assigned to p(m). The weights are adjusted as follows:

$$w_{IOIJ}(n+1) = w_{IOIJ}(n) + \Delta w_{IO} \tag{10a}$$

$$w_{MOIJ}(n+1) = w_{MOIJ}(n) + \Delta w_{MO} \tag{10b}$$

$$j=m,\quad \text{If } x_{OJ}(n) \geqq \theta+\alpha \quad \text{then} \quad \Delta w_{IO}=0,\ \Delta w_{MO}=0 \tag{11a}$$

$$\text{If } x_{OJ}(n) < \theta+\alpha \quad \text{then} \quad \Delta w_{IO}=\mu \cdot q \cdot y_{II},\ \Delta w_{MO}=\mu \cdot q \cdot y_{MI} \tag{11b}$$

$$j \neq m, \text{ If } x_{OJ}(n) > \theta - \alpha \quad \text{then} \quad \Delta w_{IO} = -\mu \cdot q \cdot y_{IJ},\ \Delta w_{MO} = -\mu \cdot q \cdot y_{MI} \quad (12a)$$

$$\text{If } x_{OJ}(n) \leqq \theta - \alpha \quad \text{then} \quad \Delta w_{IO} = 0,\ \Delta w_{MO} = 0 \quad (12b)$$

The initial guess of the weights is also zero.

The above process is repeated for all patterns until Δw_{IO} and Δw_{MO} become zero. In this step, w_{IMIJ} are fixed to the adjusted values. If the training does not converge, then another critical pattern is selected, and a middle unit is assigned to this pattern. After that, the above process is repeated.

IV DIVIDED FORM FOR HYBRID MULTILAYER NEURAL NETWORK

When all possible patterns are taken into account, $N_P = 2^{N_B}$, such as the parity problem, a large number of the middle units are required. In order to save the middle units, a divided form is proposed.

Figure 8 shows an example of the divided form, which is separated into two blocks. Generally speaking, it is possible to divide a network into more than two blocks. The input pattern $\mathbf{p}(m)$ is divided into $\mathbf{p}_1(m)$ and $\mathbf{p}_2(m)$ as follows:

$$\mathbf{p}(m) = [\mathbf{p}_1(m), \mathbf{p}_2(m)] \quad (13)$$

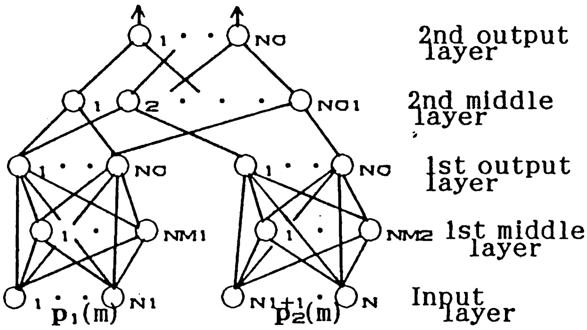

Fig.8 Divided form for hybrid multilayer neural network.

Partial patterns $\mathbf{p}_1(m)$ and $\mathbf{p}_2(m)$ are independently classified into the odd and even groups in the 1st output layer. The outputs of this layer are once distributed into all possible combinations in the 2nd middle layer. After that, they are finally classified into N_C categories in the 2nd output layer.

In the parity problem with a large N_B, the number of the middle units can be drastically saved.

V CONCLUSION

A new hybrid multilayer neural network and its low-bit learning algorithm have been proposed for binary pattern classification. The number of the middle units can be optimized, which are used to isolate the critical patterns. The single layer low-bit learning algorithm is stable and very fast, resulting in very simplified digital hardware.

REFERENCES

[1] D.E.Rumelhart and J.T.McClelland, Parallel Distributed Processing, MIT Press, 1986.

[2] K.Nakayama, S.Inomata and Y.Takeuchi,"A digital multilayer neural network with limited binary expressions", Proc. IJCNN'90, San Diego, pp. II-587-592, June 1990.

[3] K.Nakayama, et al,"Reductions in number of bits for digital realization of multilayer neural networks (in Japanese)", IEICE Trans. vol.J73-D-II, No.8, pp.1336-1345, August 1990.

[4] F.Rosenblatt, Principles of Neurodynamics, New York, Spartan Book 1962.

[5] K.Nakayama and H.Katanama, "A low-bit learning algorithm for digital multilayer neural networks applied to pattern recognition", Proc. IJCNN'91 Singapore, pp.1867-1872, Nov. 1991.

[6] S.Sakaue et al, "Reduction of necessary precision for the learning of pattern recognition", Proc. IJCNN'91 Singapore, pp.1795-1800, Nov. 1991.