A DIGITAL MULTILAYER NEURAL NETWORK WITH LIMITED BINARY EXPRESSIONS

Kenji NAKAYAMA Satoru INOMATA Yukou TAKEUCHI
Dept. of Electrical and Computer Eng., Faculty of Tech., Kanazawa Univ.
2-40-20, Kodatsuno, Kanazawa, 920 JAPAN

ABSTRACT This paper presents a design methodology for digital multilayer neural networks with limited binary expressions. An error back propagation algorithm is modified as follows: The numbers of binary bits used for connections and unit ouputs, are decreased step by step in the training process. In order to express unit outputs with 2-level values, differential of the logistic function is replaced by small positive constant, used in weight change equations. After the above training is completed, binary expressions for connections and unit outputs can be reduced to the several bits and 2-level values, respectively. Therefore, any multiplier and nonlinear function are not required in the resulting network, which will be used for pattern recognition. Furthermore, memory capacity and adder circuit hardware can be reduced. The network performance is also insensitive to noisy patterns.

I INTRODUCTION

Artificial neural networks, consisting of massively connected nonlinear units, have been attractive [1],[2]. Particularly, multilayer neural networks, trained by an error back propagation algorithm, have demonstrated good pattern recognition facility [2],[3].

There are several approaches to hardware realization. One hopeful approach is digital realization. Because high accuracy, complicated training control, and highly integrated density can be easily achieved. On the other hand, neural networks usually require a great number of units and their connections. Therefore, only small size neural networks have been integrated on a single chip, even though the present LSI technology is employed [4],[5].

In order to miniaturize digital neural network, the numbers of units and connections should be decreased. Reductions in the number of binary bits, used for connections and unit outputs, are also important. Nonlinear functions should be simplified.

In this paper, digital multilayer neural networks, trained by the error back propagation algorithm, are taken into account. We propose an improved learning algorithm, which can drastically save the number of binary bits, while maintaining good pattern recognition facility. Furthermore, we discuss relations between reductions in the above parameters and a training convergence rate and the resulting network performance.

II MULTILAYER NEURAL NETWORKS

A two layer neural network is briefly explained in this section. Let w_{ij} and w_{jk} be connections from input units to hidden units, and from hidden units to output units, respectively. In the training process, the connection values are changed as follows: When a pattern Φ (p) is set on the input layer, the output of the ith input unit and the input and output of the jth hidden unit are denoted by u_{pi}, v_{pj} and u_{pj}, respectively. They are related by

$$v_{pj}=\sum_i w_{ij}u_{pi} \tag{1}$$

$$u_{pj} = f(v_{pj}) = \frac{1}{1 + \exp[-(v_{pj} + \theta_j)]} \tag{2}$$

where θ_j is a bias similar to a threshold. In the output layer, the same relation are held. The rule for changing weights is given by [2]

$$\Delta w_{ij}(n+1) = \mu\, \delta_j u_i + \alpha\, \Delta w_{ij}(n) \tag{3}$$

$$\Delta w_{jk}(n+1) = \mu\, \delta_k u_j + \alpha\, \Delta w_{jk}(n) \tag{4}$$

Furthermore, δ_k and δ_j are given by

$$\delta_k = (t_k-u_k)f'(v_k) = (t_k-u_k)(1-u_k)u_k \tag{5}$$

$$\delta_j = (\sum_k w_{jk}\delta_k)f'(v_j) = (\sum_k w_{jk}\delta_k)(1-u_j)u_j \tag{6}$$

where t_k is the target input for the kth output unit, and f'() indicates the first differential of f().

III REDUCTIONS IN NUMBER OF BITS FOR CONNECTIONS

3.1 Learning Algorithms

We will investigate the following learning algorithms for reductions in the number of binary bits.

Method I : In a training process, connections and unit outputs are always expressed with a limited number of bits.

Method II : In the training process, the numbers of bits for connections and unit outputs are decreased step by step.

3.2 Simulation

The minimum number of bits, required for training convergence in both Method I and II, is investigated. Connection values are rounded off after their maximum value is normalized to unity. Numeral patterns 「0」 ～ 「9」, shown in Fig.1, are used as training patterns. Black and white points correspond to 1 and 0, respectively. Table 1 shows target patterns A and B, and A is used in this section. The initial values for connections are given by random numbers, uniformly distributed in the range -0.5 ～ 0.5. The number of hidden units is fixed to 15. The parameters μ and α used in Eqs.(3) and (4) are chosen to be

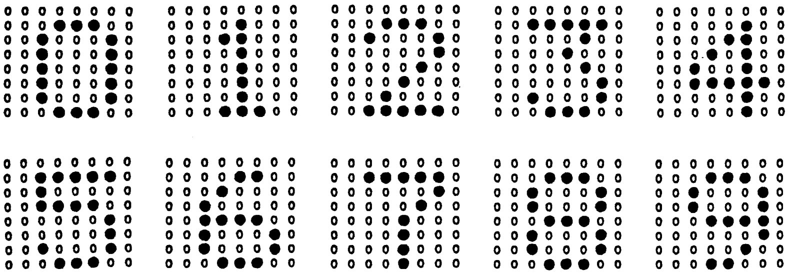


Fig.1 Numeral training patterns 「0」 ～ 「9」.

Table 1 Target patterns.

Numeral patterns	A 10units	B 4units
0	0000000001	0 0 0 0
1	0000000010	0 0 0 1
2	0000000100	0 0 1 0
3	0000001000	0 0 1 1
4	0000010000	0 1 0 0
5	0000100000	0 1 0 1
6	0001000000	0 1 1 0
7	0010000000	0 1 1 1
8	0100000000	1 0 0 0
9	1000000000	1 0 0 1

2.5 and 0.6, respectively.

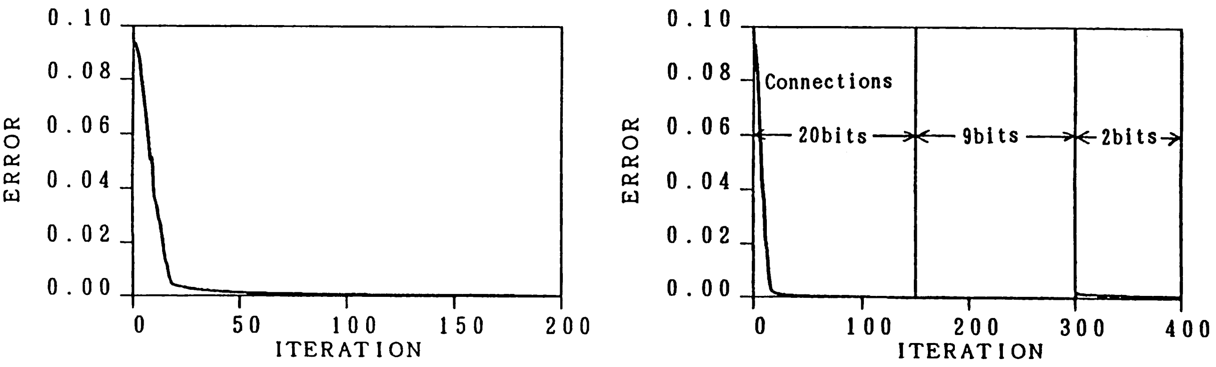
(1) Method I

The minimum number of bits, with which a training process converges, was 1(sign)+5(mag.)=6bits. A learning curve is illustrated in Fig.2(a). One iteration corresponds weight changing for all patterns 「0」 ∼ 「9」. The vertical axis indicates the output error defined by

$$\text{ERROR} = \frac{1}{P}\sum_{p=1}^{P}\left[\frac{1}{K}\sum_{K=1}^{K}(t_{pK}-u_{pK})^{2}\right]. \qquad (7)$$

(2)Method Ⅱ

The number of bits is reduced in three stages, as shown in Fig.2(b). The final number of bits is 1(sign)+1(mag.)=2bits. The quantization steps of the binary code, used in our simulation, are slightly different from the ordinary binary expression due to normalization. For example, 2 bit binary code expresses five steps, that is -1, -0.5, 0. 0.5, 1.



(a) Method I with 6 bits. (b) Method Ⅱ with 2 bits in final stage.

Fig.2 Learning curves with limited binary expressions for connections.

Ⅳ REDUCTIONS IN NUMBER OF BITS FOR UNIT OUTPUTS

In order to remove any multiplier, required in Eq.(1), that is $w_{ij}u_{pi}$ or $w_{jK}u_{pj}$, our discussion is focused on 2-level expression for unit outputs.

4.1 Modified Delta Rule

Two-level expression corresponds to using a threshold function instead of the logistic function defined by Eq.(2). Therefore, the first-order differential cannot be calculated. For this reason, we replace the differential by small positive constant f_o and f_h as follows:

$$\delta_K=(t_K-u_K)f'(v_K) \quad \rightarrow \quad (t_K-u_K)f_o \qquad (8)$$

$$\delta_j=(\sum_K w_{jK}\delta_K)f'(v_j) \rightarrow \quad (\sum_K w_{jK}\delta_K)f_h \qquad (9)$$

4.2 Simulation

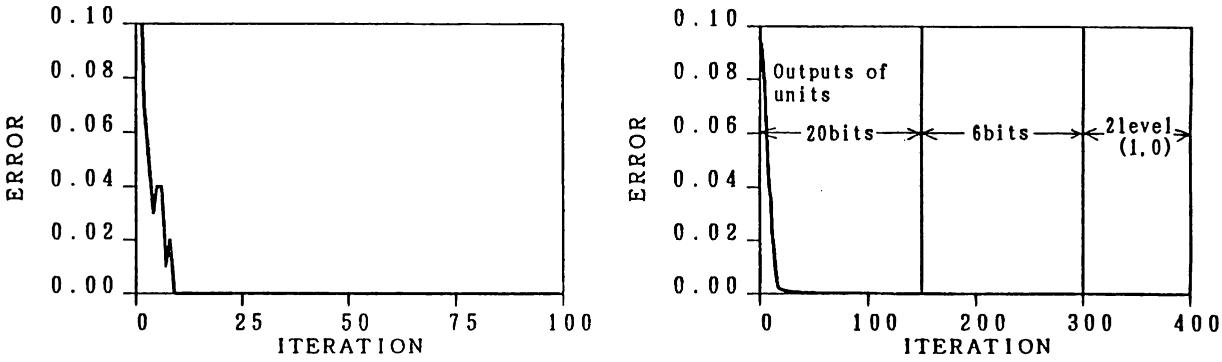
In the simulation, f_o and f_h are chosen to be 0.05, and connection values are not quantized. Another condition is the same as in Sec. Ⅲ.

(1) Method I

The unit output are expressed by 2-level values, that is 1 and 0, during the training process. The learning curve is illustrated in Fig.3(a). The network training converges, even though some vibrations occur.

(2) Method II

The number of bits is gradually decreased in three stages as shown in Fig.3(b). The learning curve demonstrates no vibration. Because, after 150 iteration, the network mostly converges, and the outputs of both hidden and output units approach to 1 or 0. Therefore, quantization error is very small.



(a) Method I. (b) Method II.

Fig.3 Learning curves with 2-level value expressions for unit outputs.

V BIT REDUCTION IN BOTH CONNECTIONS AND UNIT OUTPUTS

First, the number of the hidden units is fixed to 15. Another condition is the same as in the previous sections. In Method I, the learning process converges down to 1(sign)+4(mag.)=5bits for connections. The unit outputs are expressed with 2-level values. Method II can decrease the number of bits for the connections into 1(sign)+1(mag.)=2bits. Learning curves in both methods are mostly the same as in Figs.3(a) and 3(b), respectively.

Next, effects of the number of hidden units is investigated. Method I is employed. Both target patterns A and B are taken into account. Five sets of random numbers are used for the initial guess of the connections. The simulation results are illustrated in Table 2. Numerical data in the table indicate the numbers of random number sets, with which the training process converges. In the shaded portion, the training always converges for all five sets. From this table, the numbers of the hidden units, and bits for the connections are inversely proportional to each other. Therefore, there exists some lower bound for hardware

Table 2 Effects of numbers of hidden units and binary bits for connections.

Number of hidden units	3	4	5	6	7	8	9	10
12	4 1	3 4	3 3	5 5 ▨	▨	▨	▨	▨
11	3 2	3 2	4 2	5 5 ▨	▨	▨	▨	▨
10	3 0	2 0	3 3	4 4	5 4	5 5 ▨	▨	▨
9		1 0	1 2	4 3	5 5 ▨	▨	▨	▨
8			2 0	5 4	5 4	5 5 ▨	▨	▨
7				3 2	2 2	3 5	3 4	4 3
6					2 2	2 2	3 1	4 2
5							1 2	3 2
4								0 0

Target patterns A
B

Number of bits for connections

reductions.

Furthermore, the numbers of bits, required for the unit inputs are investigated. In the above simulation, the maximum values for w_{ij} and v_j were 1.74 and 7.74, and for w_{jk} and v_k were 3.42 and 16.1, respectively. Therefore, 3 bits are further required. In the Method Ⅱ case, for example, 5 bits including a sign bit, are required for the unit inputs.

Ⅵ STABILITY ANALYSIS FOR NOISY PATTERN RECOGNITION

6.1 Comparison between Methods Ⅰ and Ⅱ

After the network is trained using the numeral patterns shown in Fig.1, it is used to recognize noisy patterns, whose examples are shown in Fig.4. The target pattern A in Table 1 is taken into account. The number of the hidden units is fixed to 15. In this simulation, a standard method, in which values of the connections and the unit outputs are not rounded off, is also evaluated for comparison. Actually, single precision floating point is used.

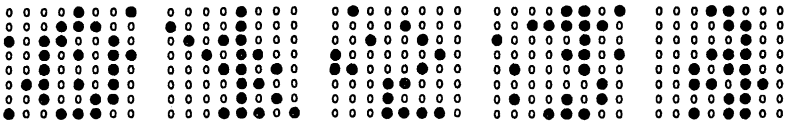


Fig.4 Examples of noisy patterns for 「0」 ~ 「4」. Ten unit outputs, randomly selected, are changed.

In Method Ⅰ, the connections are expressed with 7 bits including a sign bit. The unit outputs are expressed with 2-level values. In Method Ⅱ, the number of bits is decreased in three stages as shown in Figs.2 and 3. Three final bits, 6, 4 and 2 bits are considered for the connections. The number of noises, correspond to reverse of the outputs of the input units, is increased by 1.

The simulation results are shown in Table 3. Numerical data in this table mean the maximum number of noises, up to which the exact training pattern is recollected. Thirty sets of noise patterns are used in the simulation, and average value for the maximum number of noises is listed in this table.

Table 3 Comparison of three learning methods based on recognition rates for noisy patterns.

Numeral patterns		0	1	2	3	4	5	6	7	8	9
Standard		16	17	18	18	22	17	16	14	14	17
Method Ⅰ		1	5	2	3	2	2	3	3	2	2
Method Ⅱ	6 bits	13	15	17	14	18	15	15	14	14	16
	4 bits	13	15	14	15	19	15	15	13	14	15
	2 bits	7	15	17	8	17	10	6	8	8	16

From these results, the network obtained by Method Ⅰ is very sensitive to noisy pattern. On the contrary, the recognition rates of Method Ⅱ are not so decreased from that of the standard method, down to 4 bit binary expression.

6.2 Discussions on Stability for Noisy Patterns

In the multilayer networks under the error back propagation training, the hidden unit outputs and the connections from the hidden units to the ouput units are simultaneously adjusted so as to discriminate the input patterns. As a result, differences on these parameters are emphasized.

In Method Ⅰ, however, the hidden unit outputs are highly dependent on initial guess for the connections, and cannot be optimized. In other word, difference of the input patterns is concentrated on limited hidden units. The connections from this limited hidden units are emphasized too much. This kind of pattern discrimination is easily broken by noise. Furthermore, noise margin is also decreased due to a threshold function.

On the contrary, in Method Ⅱ, a network is trained using a large number of bits in the first stage. After the network mostly converged, the binary expression is limited. In the error back propagation process, the unit outputs tend to converge on 0 or 1 due to differential of the logistic function. Therefore, quantization errors in the unit outputs are very small. As a result, the optimized network, as in the standard method, can be obtained.

In order to confirm the above discussions, the resulting hidden unit outputs for the input patterns 「1」 and 「5」 are shown in Table 4. The hidden unit outputs in Method Ⅱ are very close to that in the standard method.

Table 4 Examples of hidden unit outputs by three learning methods.

		Outputs of hidden units							
「1」	Standard	0.89	0.15	0.97	0.80	0.97	0.06	0.01	1.00
		0.00	0.95	0.98	0.96	0.96	0.08	0.03	
	Method Ⅰ	1.00	0.00	0.00	0.00	1.00	0.00	1.00	1.00
		0.00	1.00	1.00	1.00	0.00	0.00	1.00	
	Method Ⅱ	1.00	0.00	1.00	1.00	1.00	0.00	0.00	1.00
		0.00	1.00	1.00	1.00	1.00	0.00	0.00	
「5」	Standard	0.83	0.05	0.03	0.92	0.98	0.11	0.44	0.02
		0.80	0.01	0.18	0.02	0.00	0.96	1.00	
	Method Ⅰ	0.00	0.00	0.00	0.00	1.00	0.00	1.00	0.00
		0.00	0.00	0.00	0.00	0.00	1.00	0.00	
	Method Ⅱ	1.00	0.00	0.00	1.00	1.00	0.00	0.00	0.00
		1.00	0.00	0.00	0.00	0.00	1.00	1.00	

Ⅵ CONCLUSIONS

Discussions in this paper are summarized as follows: The number of binary bits should be decreased step by step in the training process. After the training is completed, the connections and the unit outputs can be expressed with the several bits and 2-level values, respectively. For 2-level expression of unit outputs, the differential of the logistic function is replaced by small positive constant. As a result, any multiplier and nonlinear function are not required, and also memory capacity and adder circuits are reduced. Although, in the training process, a general purpose computer is desirable, the obtained digital neural network, used for actual pattern recognition, can be drastically simplified. The recognition facility is stable for noisy patterns.

REFERENCES

[1]J.J.Hopfield,"Neural networks and physical systems with emergent collective computational abilities," Proc. Natl. Acad. Sci. USA, vol.79, pp.2554-2558, April 1982.

[2]D.E.Rumelhart and J.L.McClelland, Parallel Distributed Processing, MIT Press, MA 1986.

[3]T.Sejnowski and C.R.Rosenberg,"NETtalk: A parallel network that learns to read aloud," Tohns Hopkins Univ. Tech. Rep., JHU/EECS-86/01, 1986.

[4]J.Tomberg et al,"Fully digital neural network implementation based on pulse density modulation," Proc. IEEE CICC, pp.12.7.1-12.7.4, 1989.

[5]Y.Hirai et al,"A digital neuro-chip with unlimited connectability for large scale neural networks," Proc. IJCNN, vol.2, pp.163-169 1989.