A DELTA RULE ALGORITHM USING DOUBLE HYSTERESIS THRESHOLDS FOR RECURRENT ASSOCIATIVE MEMORY

Kenji NAKAYAMA **Katsuaki NISHIMURA**

Dept. of Electrical and Computer Eng., Faculty of Tech., Kanazawa Univ.
2-40-20, Kodatsuno, Kanazawa 920 JAPAN
E-Mail: nakayama@haspnn1.ec.t.kanazawa-u.ac.jp

ABSTRACT An associative memory using fixed and variable hysteresis thresholds in learning and recalling processes, respectively, has been proposed by authors. This model can achieve a large memory capacity and very low noise sensitivity. However, a relation between weight change Δ w and the hysteresis threshold $\pm$ T has not been well discussed. In this paper, a new learning algorithm is proposed, which is based on a delta rule. However, in order to stabilize the learning process, a method of using double hysteresis thresholds is proposed. Unit states are updated using $\pm$ T. The error, used for adjusting weights, is evaluated using $\pm$ (T+dT). This means 'over correction'. Stable and fast convergence can be obtained. Relations between η =dT/T and convergence rate and noise sensitivity are discussed, resulting the optimum selection for η. Furthermore, the order of presenting training data is optimized taking correlation, into account. In the recalling process, a threshold control method is further proposed in order to achieve fast recalling from noisy patterns.

I INTRODUCTION

An associative memory is one of hopeful applications of artificial neural networks (NNs). Connection weights are adjusted so that patterns are memorized on equilibrium states. Conventional methods, autocorrelation methods and orthogonal methods [1]-[6], assume symmetrical weights, and are effective only for lineally independent patterns or orthogonal patterns. Therefore, memory capacity and noise insensitivity are strictly limited.

Authors proposed an associative memory, and its learning and recalling algorithms [7]-[9]. Fixed and variable hysteresis thresholds were effectively employed in the learning and recalling processes, respectively. It can drastically improve recalling ability from noisy pattens. However, a relation between connection weight change and the threshold was not well discussed. It was determined by experience. Furthermore, control of the variable hysteresis threshold in the recalling process was not optimized.

In this paper, new learning and recalling algorithms are proposed in order to solve the above remaining problems, and to achieve fast convergence, low noise sensitivity and large memory capacity.

II ASSOCIATIVE MEMORY WITH HYSTERESIS THRESHOLD

The associative memory proposed in [7]-[9] is briefly described here. A unit is connected with all the other units. The weights are not always

0-7803-1901-X/94 $4.00 ©1994 IEEE

symmetrical. A self-loop is not used. Let the input and output of the ith unit at the nth transition cycle be $u_i(n)$ and $v_i(n)$, respectively. The connection weight from the ith unit to the jth unit is expressed w_{ij}. Network transition is formulated as follows:

$$u_j(n) = \sum_{i=1}^{N} w_{ij} v_i(n), \quad w_{ii}=0 \qquad (1)$$

$$v_j(n+1) = 1, \qquad u_j(n) \geqq T(n) \qquad (2a)$$

$$v_j(n+1) = v_j(n), \quad | u_j(n) | < T(n) \qquad (2b)$$

$$v_j(n+1) = 0, \qquad u_j(n) \leqq -T(n) \qquad (2c)$$

III LEARNING ALGORITHM FOR CONNECTION WEIGHTS

3.1 Delta Rule Algorithm with Double Hysteresis Thresholds

The proposed learning algorithm is based on a delta rule [10]. However, the ordinary error correcting method is very poor in training the mutually connected NNs. The learning process is very unstable and oscillation easily occurs. Therefore, in order to prevent such unstable behavior and to achieve stable and fast convergence, double hysteresis threshold is proposed. The learning algorithm is described in the following.

Let P(m), m=1~M, be patterns to be memorized. $p_i(m)$ expresses the ith element of P(m), which takes a binary value, that is 1 or 0.

(1) Initial weights are set to zero.

(2) The network state is set to one of the patterns P(m).

(3) Calculate the unit input by Eq.(1). $p_i(m)$ is used instead of $v_i(n)$.

$$u_j(n)=\sum_{i=1}^{N} w_{ij}(n) p_i(m), \quad p_i(m)=1 \text{ or } 0 \qquad (3)$$

(4) Letting ±T be hysteresis thresholds, the error at the jth unit input is evaluated by

$$\varepsilon_j(n) = p_j(m)[T-u_j(n)] + (1-p_j(m))[T+u_j(n)] \qquad (4)$$

(5) If $| u_j(n) |$ cannot exceed T, then $\varepsilon_j(n) > 0$. Thus, $\varepsilon_j(n) \leqq 0$ means the output is correct, that is $v_j(n+1)$ becomes $p_j(m)$. Therefore, the weights are updated following

$$w_{ij}(n+1) = w_{ij}(n) + \mu(n) \delta_j(n) p_i(m) S[\varepsilon_j(n)] \qquad (5)$$

$$\delta_j(n) = p_j(m)[T+dT-u_j(n)] + (p_i(m)-1)[T+dT+u_j(n)] \qquad (6)$$

$$S[\varepsilon_j(n)] = \begin{cases} 1, & \varepsilon_j(n) > \\ 0, & \varepsilon_j(n) \leqq 0 \end{cases} \qquad (7)$$

$$\mu(m)=\mu_0/(M(m)-1), \quad 0< \mu_0 \leqq 1 \qquad (8)$$

M(m) is the number of the units locate on P(m).

In the above equations, T+dT is used instead of T. dT serves as the hysteresis margin. A pair of T and dT is called "double hysteresis thresholds" in this paper. This method can stabilize and accelerate the learning process. In the later section, we will compare the learning behavior with dT and without dT through computer simulation. A ratio of dT and T is denoted η =dT/T.

(6) The connection weights are simultaneously updated for a pattern P(m) following Eqs.(5)-(8).

(7) By replacing P(m) by P(m+1), the above processes **(2)** through **(6)** are repeated.

Furthermore, Steps **(2)** through **(7)** are repeated until all unit inputs satisfy

$$\text{If } p_i(m) = 1, \text{ then } u_i(n) \geqq T \qquad (9a)$$

$$\text{If } p_i(m) = 0, \text{ then } u_i(n) \leqq -T \qquad (9b)$$

3.2 Relation between η =dT/T and Convergence Rates

dT is used to stabilize the learning process. If patterns P(i) and P(j) are conflict with each other, then adjusting of the connection weights for P(i) are easily broken by learning P(j) some other time. This causes oscillation, that is unstable learning and slow convergence. In order to avoid this unstable phenomena, dT is proposed. However, if η is very small, effect of dT is not sufficient. Therefore, η should be chosen to guarantee stable and fast convergence.

3.3 Relation between η =dT/T and Noise Sensitivity

When some noise is added to the pattern, the conditions Eqs.(9a) and (9b) are not satisfied any more. In this situation, the network can change its state. The additive noise and error responses caused by the noise are transmitted though the connection weights. Thus, large weights, which are not necessary to satisfy the conditions Eqs.(9a) and (9b), increase noise sensitivity. Therefore, it is important to decrease the connection weights, while the conditions Eq.(9) is satisfied.

On the other hand, distribution of the connection weights is determined by η =dT/T. By using small η, the distribution of the weights can be suppressed. This direction of η is opposite to that of stable and fast convergence. Therefore, η should be optimized taking both the convergence rate and the noise sensitivity into account. This will be further discussed in **Sec.V**.

3.4 Order of Presenting Training Patterns

In the mutually connected NNs, connections from common units for many patterns to the other units are not emphasized. On the contrary, connections from the units, not included in many patterns, to the other units are emphasized, and play an important role in the recalling process. In other words, patterns having high correlation with the other patterns are difficult to be memorized, and to be recalled from noisy patterns.

In the learning process given by Eqs.(5) through (8), the connection weights are adjusted so that the unit inputs satisfy the threshold. This adjusting affects the patterns early presented in both positive and negative directions. This negative effects will be readjusted in the next learning. The positive effects will remain. By repeating this learning, the early presented patterns can gain noise margin.

Taking the above discussions into account, highly correlated patterns are early presented to the NN. By this method, noise sensitivity is averaged over all patterns. The correlation is evaluated by the Hamming distance as follows:

$$d_H(i,j) = \sum_{K=1}^{M} | p_K(i) - p_K(j) | \qquad (10)$$

$$d_H(i) = \frac{1}{M} \sum_{J=1}^{M} d_H(i,j) \qquad (11)$$

IV RECALLING FROM INCOMPLETE PATTERNS

4.1 Variable Hysteresis Threshold

After the training completed, all units satisfy Eqs.(9a) and (9b). By adding noise, these conditions are destroyed, and the network changes its state. State changes are transferred through connections to the other units, and cause another state transition. The wrong state change tend to cause another wrong state changes. As a result, the NN fails in recalling the correct memory. Therefore, it is important to select the units, whose input are probably correct, and to change these units first.

For this purpose, we proposed variable hysteresis threshold $\pm T(n)$ in the association process [7]-[9]. Let $e_i(n)$ be an error added to the ith unit. It takes ± 1. In the noisy pattern, the unit input is expressed using $e_i(n)$ as follows:

$$u_j(n) = \sum_i w_{ij}[p_i(m)+e_i(n)] = \sum_i w_{ij}p_i(m) + \sum_i w_{ij}e_i(n) \qquad (12)$$

The first term is the correct component, satisfies Eq.(9). The second term is the error component. If the following condition is held, inaccurate transition is caused. The first and second terms are denoted $U_j(n)$ and $E_j(n)$, respectively.

$$p_j(m)=1: U_j(n) < -T(n), \qquad (13a)$$
$$p_j(m)=0: U_j(n) > T(n) \qquad (13b)$$

If we assume for $p_j(m)=1$ and 0, $U_j(n)$ takes T and -T, respectively, the above conditions can be rewritten as,

$$p_j(m)=1: E_j(n) < -T-T(n) \qquad (14a)$$
$$p_j(m)=0: T+T(n) < E_j(n) \qquad (14b)$$

$E_j(n)$ is uniformly distributed. Probability of Eq.(14) can be decreased by setting T(n) to much larger than T. Finally, T(n) approaches to T. This is an idea behind the variable hysteresis threshold [7]-[9].

T(n) is chosen to be large enough to T, and is gradually decreased toward T. In the previous work, T(n) was determined by

$$T(n) = T(0) - \alpha n, \quad \alpha: \text{constant} \qquad (15)$$

T(0) is chosen to larger than T. T(0) and α are also determined by experience.

4.2 Optimum Control of Variable Hysteresis Threshold

In this paper, an improved version of controlling T(n) is proposed. The method is described in the following step by step.

(1) The first threshold is determined by

$$T(0) = \max_i \{ | u_i(0) | \} \qquad (16)$$

$u_i(0)$ is the input of the ith unit at the initial state. The operation $| x |$ means absolute value of x.

(2) The units, whose input satisfy

$$| u_i(0) | = T(0) \qquad (17)$$

are updated following Eqs.(1) and (2). ± T(0) are used until the network state does not change any more.

(3) The next threshold T(1) is determined in the same way as Eq.(16).

$$T(1) = \max_i \{ | u_i(n) | \} \qquad (18)$$

The same processes in Step**(2)** are repeated.

Thus, after the network reaches to some sate, the maximum input is adopted as the next threshold. Finally, T(n) can reach T.

V SIMULATION RESULTS

5.1 Convergence Properties

A mutually connected NN, having 8x8=64 units, is used. Training data are generated as random patterns. Half of the units take 1, and the other units take 0. Hamming distances among patterns form normal distribution with mean of 32 and covers from 22 to 44. The learning coefficient μ_0 in Eq.(8) is unity.

Figure 1 shows relation between the number of patterns memorized (horizontal axis) and the number of iterations (vertical axis). Adjusting connection weights using one set of patterns is counted as one iteration. dT/T=0 means the ordinary delta rule [10]. The graph with a symbol ◇ indicates that order of presenting training patterns is fixed. The other graph with a symbol + means that the order is randomized at each iteration. dT/T=0.1 and 1 indicate the proposed method.

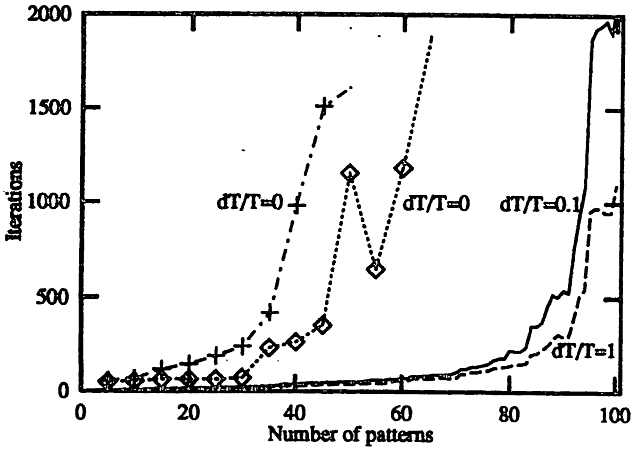

Fig.1 relations between the number of patterns memorized and the number of iterations in learning.

From these results, the delta rule algorithm without dT is very poor for training the mutual connected NN. Patterns moré than 45(+) and 60(◇) cannot be memorized due to unstable behavior. On the contrary, the proposed method is very efficient. As discussed in Sec.**3.2**, a large η =dT/T can provide fast convergence. Memory capacity can be also increased.

5.2 Memory Capacity

The memory capacity is dependent on correlation among the patterns. In this paper, random patterns are used. The results of Fig.1 are used for this discussion. The number of iterations gradually increases up to about 80 patterns. After that, it quickly increases. This is a very peculiar phenomenon. The training converged until 100 patterns. The number of the patterns could be increased a little more. However, from the very sharp slope, it is almost limited near by 100 patterns. Thus, the memory capacity is about 100/64≒1.56 times as large as N. When the correlation among the patterns in different sets is invariant, this relation can be held for an arbitrary number of units. This memory capacity is much higher than the other.

5.3 Recalling Accuracy from Noisy Patterns

Noisy patterns are generated by adding random error. Units are randomly selected, and their state are reversed. Thirty sets of random numbers are used. Association rates are evaluated in average. Figure 2 shows the simulation results. These results also support the previous discussion given in Sec.**3.3**. Association rates are inversely proportional to η =dT/T. Roughly speaking, around η =0.2 is desirable for both convergence speed and recalling accuracy.

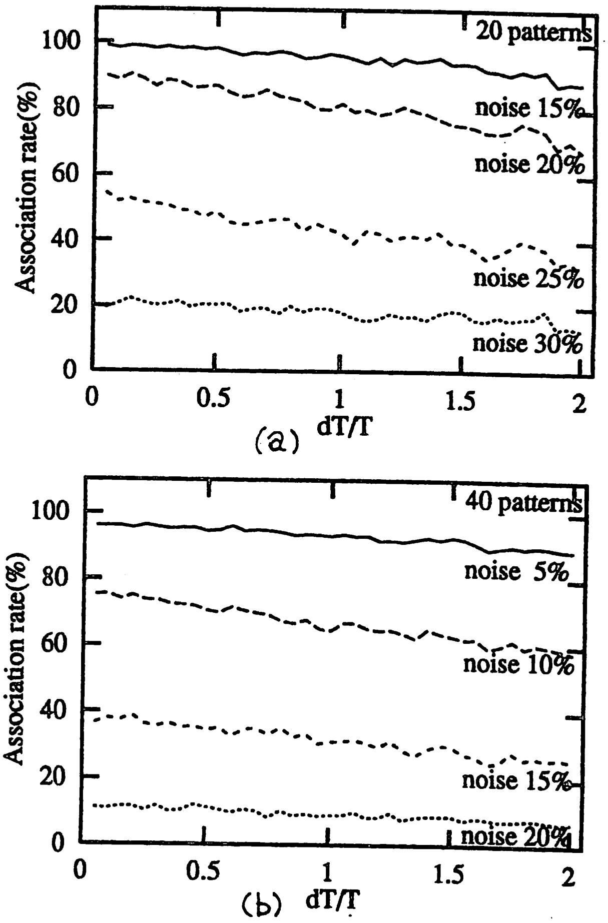

Fig.2 Relations between association rates and dT/T. (a) 20 patterns and (b) 40 patterns are memorized.

5.4 Improvement of Association Rates

Effects of the order of presenting the training data, and the control method of the hysteresis threshold are investigated. Since random patterns have almost the same correlation, alphabet patterns are employed for this purpose. The patterns are expressed with 16x16=256 dots. The network has also 256 units.

Table 1 lists association rates for noisy alphabet patterns. Method A is the original one [8], B improves the hysteresis threshold control, C orders the training patterns based on correlation, and BC combines Methods B and C. Association rate X is of the original pattens, that is 'correct answer', Y is of untrained patterns, that is spurious, and Z is of the other trained patterns.

The recalling accuracy from noisy patterns can be improved by 3~5% from the original version. The ordering of the training patterns is more efficient.

Table 1 Association rates for alphabet patterns with random noise.

(a) η =0.1, Noise=15%

Methods	Association rates		
	X	Y	Z
A	96.3	2.8	0.9 %
B	96.4	2.6	1.0
C	97.1	2.2	0.7
BC	97.1	2.3	0.6

(b) η =0.1, Noise=20%

Methods	Association rates		
	X	Y	Z
A	87.1	8.7	4.2 %
B	88.1	7.7	4.2
C	89.1	7.5	3.4
BC	89.5	7.1	3.3

(c) η =0.1, Noise=25%

Methods	Association rates		
	X	Y	Z
A	71.7	17.5	10.8%
B	73.3	17.1	9.7
C	76.6	13.9	9.5
BC	76.8	14.6	8.6

VI CONCLUSIONS

The delta rule algorithm using the double hysteresis thresholds has been proposed for the associative memory with hysteresis threshold. Stable and fast learning can be achieved. Large memory capacity is obtained. The proposed ordering the training patterns and the controlling the hysteresis threshold can further improve association rates.

REFERENCES

[1]T.Kohonen, Self-Organization and Associative Memory, 3rd Ed., Springer-Verlag 1989.

[2]K.Nakano, "Associatron-A model of associative memory",IEEE Trans vol. SMC-2, pp.380-388 1972.

[3]S.Amari,"Neural theory of association and concept-formation", Biol. Cybern., vol.26, pp.175-185, 1977.

[4]J.J.Hopfield, "Neural networks and physical system ~", Proc. Natl. Sci. USA, vol.79, pp.2554-2558, 1982.

[5]D.Amit et al, "Storing infinite number of patterns~", Phys. Rev. Lett.,pp.1530-1533,1985.

[6]S.Amari and K.Mginu,"Statistical neurodynamics of~",Neural Networks, vol.1,pp.63-73,1988.

[7]N.Mitsutani and K,Nakayama, IEICE Japan Rep. Tech. Meeting, vol. NC90-89, pp.125-130, March 1991.

[8]K.Nakayama and N.Mitsutani,"An adaptive hysteresis~", Proc. IJCNN'91 Seattle, p. II A-914, 1991.

[9]K.Nakayama et al.,"Memory capacity bound threshold~ ", Proc. IJCNN'93 Nagoya, pp.2603-2606, 1993.

[10]D.E.Rumelhart and J.L.McClell, Parallel Distributed Processing, MIT Press, 1986.